



608: Advanced configuration of XenApp and
XenDesktop 7.6 using the PowerShell SDK

Hands-on Lab Exercise Guide

Kimberly Ferrie & Mick Glover

January 2015

Contents

Contents	1
Overview.....	2
Exercise 1: A Good Start: Creating a PowerShell Profile.....	4
Exercise 2: A Little Invention: Manually Adding a Controller to a Site.....	11
Exercise 3: A Tricky Situation: Manually Updating FMA Service DB Schemas.....	26
Exercise 4: A Clever Idea: Controlling Access to Desktops through the SDK.....	39
Exercise 5: A Timely Reminder: Configuring Multistage Power Policy Rules.....	53
Exercise 6: A Final Step: Manually Removing a Controller from a Site (Optional)	61

Overview

Hands-on Training Module

Objective

This training will provide hands-on experience with carrying out XenApp and XenDesktop 7.6 advanced administration & configuration tasks.

Prerequisites

Basic XenDesktop 5.x/7.x and or XenApp 7.x administration and configuration knowledge would be beneficial but not essential.

Audience

Citrix Partners, Customers, Sales Engineers, Consultants, Technical Support.

Lab Environment Details

The Student Desktop is accessed remotely using Citrix Receiver running on your laptop. All windows applications such as XenCenter, (the XenServer GUI management tool) and Remote Desktop Manager are accessed from shortcuts on the Student Desktop.



For convenience, two methods can be used to access the lab VM's:

1. XenCenter
2. Remote Desktop Manager

Note: If you encounter the following message at any point during the lab exercises, click **Restart Later**:



Lab Guide Conventions

	This symbol indicates particular attention must be paid to this step
	Special note to offer advice or background information

List of Virtual Machines Used

VM Name	IP Address	Description / OS
AD.training.lab	192.168.10.11	Windows Server 2008R2 - Domain Controller, DNS, DHCP, Certificate Services
DC1	192.168.10.22	Windows Server 2008 R2 – XenApp / XenDesktop 7.6 Controller
DC2	192.168.10.23	Windows Server 2008 R2 - XenApp/XenDesktop 7.6 Controller
RDS2012	192.168.10.24	Windows Server 2012 R2 – RDS VDA 7.6
SF	192.168.10.25	Windows Server 2008 R2 - StoreFront 2.6 Server
SQLServer	192.168.10.26	Windows Server 2012 R2 - SQL Server 2012
Win7Client	192.168.10.27	Windows 7 Endpoint client.
Win7Master	192.168.10.28	Win7 MCS Golden Image with VDA 7.6
Win7VDA75	192.168.10.29	Win7 VDA 7.5 with Hotfixes
Win7VM	192.168.10.30	Win7 VDA v7.6
Win8VM	192.168.10.31	Win 8.1 VDA v7.6

Required Lab Credentials

The credentials required to connect to the environment and complete the lab exercises.

VM Name	Username	Password
AD.training.lab	training\administrator	Citrix123
DC1	training\administrator	Citrix123
DC2	training\administrator	Citrix123
RDS2012	training\administrator	Citrix123
SF2	training\administrator	Citrix123
SQLServer	training\administrator	Citrix123
Win7Client	training\administrator	Citrix123
Win7Master	training\administrator	Citrix123
Win7VDA75	training\administrator	Citrix123
Win7VM	training\administrator	Citrix123
Win8VM	training\administrator	Citrix123

Exercise 1


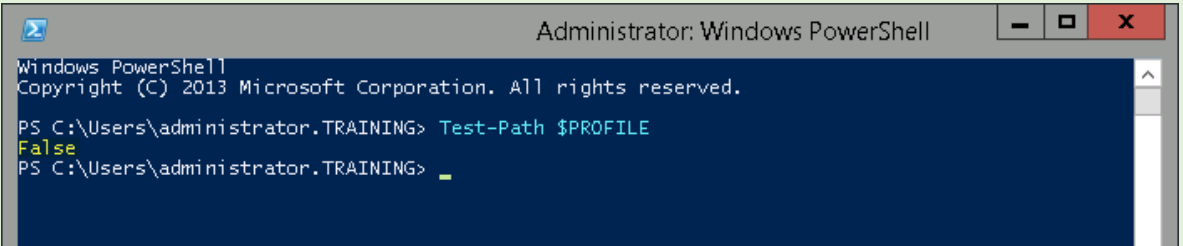
A Good Start: Creating a PowerShell Profile

Overview

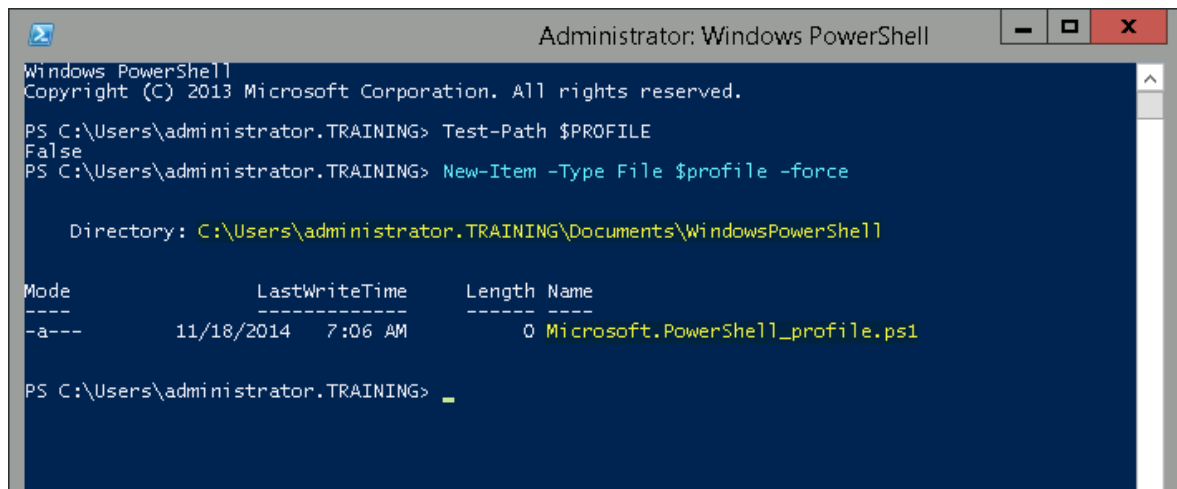
In this exercise we will create a customized PowerShell (PoSH) Profile so that the required functionality to interact with our XenDesktop Site is available each time we start a PoSH instance.

Step by step guidance

Estimated time to complete this lab: 20 minutes.

Step	
1.	<p>Login to DC1 as training\administrator with password Citrix123 and launch a PowerShell window by clicking on the PoSH icon on the taskbar:</p> 
2.	<p>Type <i>Test-Path \$PROFILE</i> to check for the existence of a previously configured Profile on DC1:</p>  <p>Note: A return value of False indicates that a PoSH Profile does not exist and therefore must be created. This is the expected behavior.</p>

3. Type ***New-Item -Type File \$profile -force*** to create a new PoSH Profile:



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2013 Microsoft Corporation. All rights reserved.

PS C:\Users\administrator.TRAINING> Test-Path $PROFILE
False
PS C:\Users\administrator.TRAINING> New-Item -Type File $profile -force

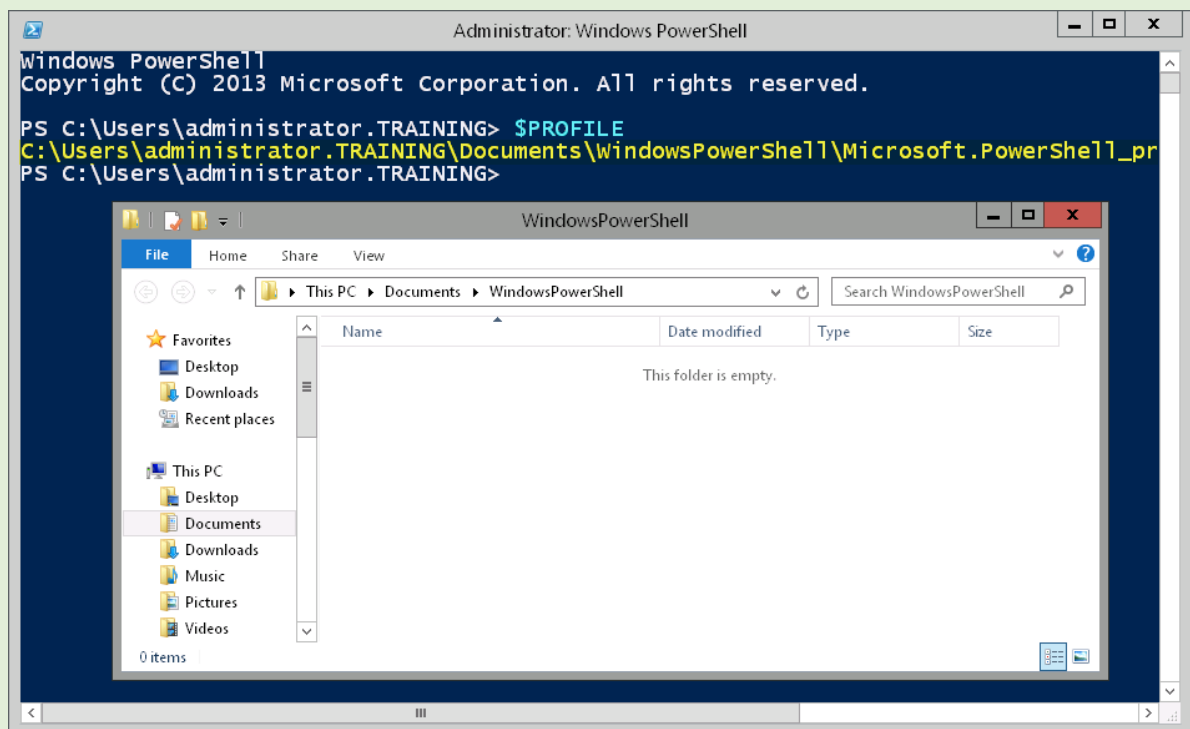
Directory: C:\Users\administrator.TRAINING\Documents\WindowsPowerShell

Mode                LastWriteTime         Length Name
----                -
-a---            11/18/2014   7:06 AM              0 Microsoft.PowerShell_profile.ps1

PS C:\Users\administrator.TRAINING>
```

Note: Although the built-in PoSH variable **\$PROFILE** can be queried at any time, the .ps1 file does not exist by default and must be created using the above command.

4. **Observation:** Prior to creating a PoSH Profile, if you type **\$PROFILE** within a PoSH instance and hit return you will notice that although the path to the ps1 file exists, the file itself does not exist by default. As per step 3 above, the ps1 file must be created by running ***New-Item -Type File \$profile -force***.



5. Type ***Notepad \$PROFILE*** and hit **RETURN** to begin editing the existing PoSH Profile:

 <http://blogs.citrix.com/2014/02/04/xd-tipster-creating-a-customized-xd-posh-profile/>

6. Paste the following text into Notepad & **Save** before closing:

```
asnp citrix*
```

```
$null = New-PSDrive -Name XDGPO -PSProvider CitrixGroupPolicy -Root \ -DomainGPO lab_policies
```

```
$null = New-PSDrive -Name XDSITEPOL -PSProvider CitrixGroupPolicy -Root \ -Controller DC1
```

```
" "
```

```
Write-Host "XD SUM608 Admin Profile loaded:" -foregroundcolor white -backgroundcolor DarkGreen
```

```
" "
```

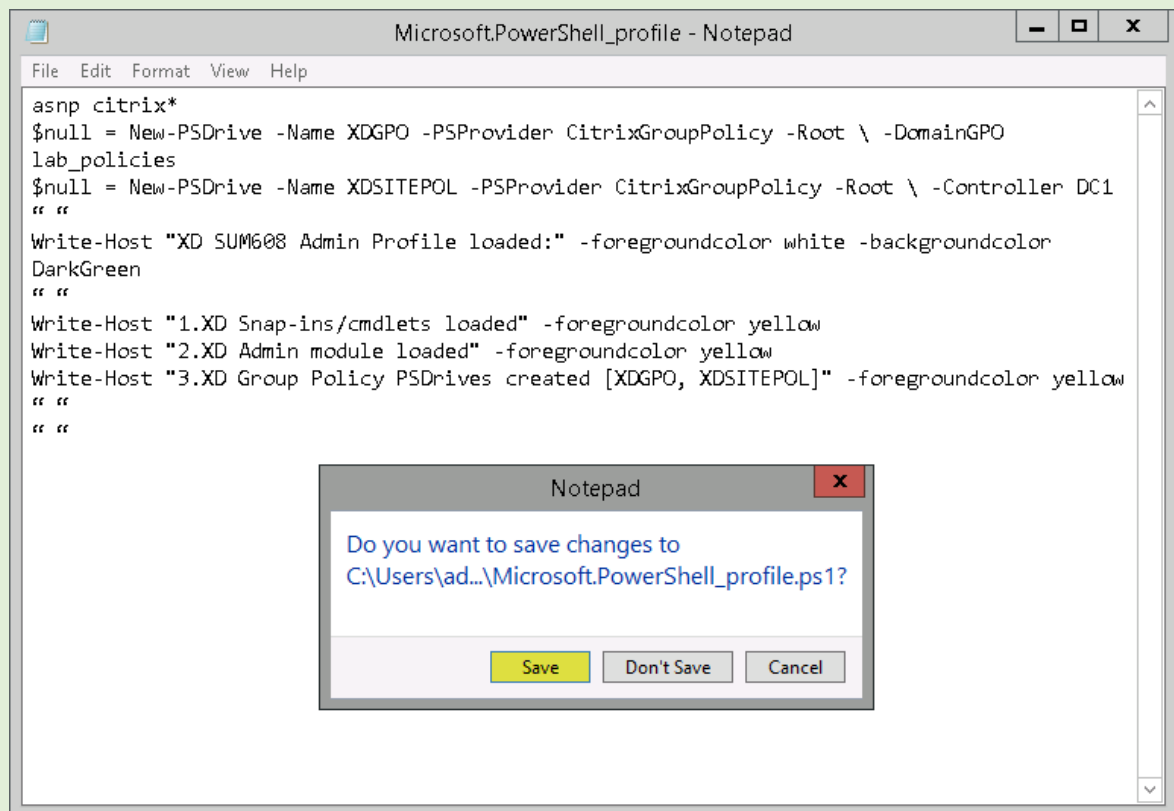
```
Write-Host "1.XD Snap-ins/cmdlets loaded" -foregroundcolor yellow
```

```
Write-Host "2.XD Admin module loaded" -foregroundcolor yellow
```

```
Write-Host "3.XD Group Policy PSDrives created [XDGPO, XDSITEPOL]" -foregroundcolor yellow
```

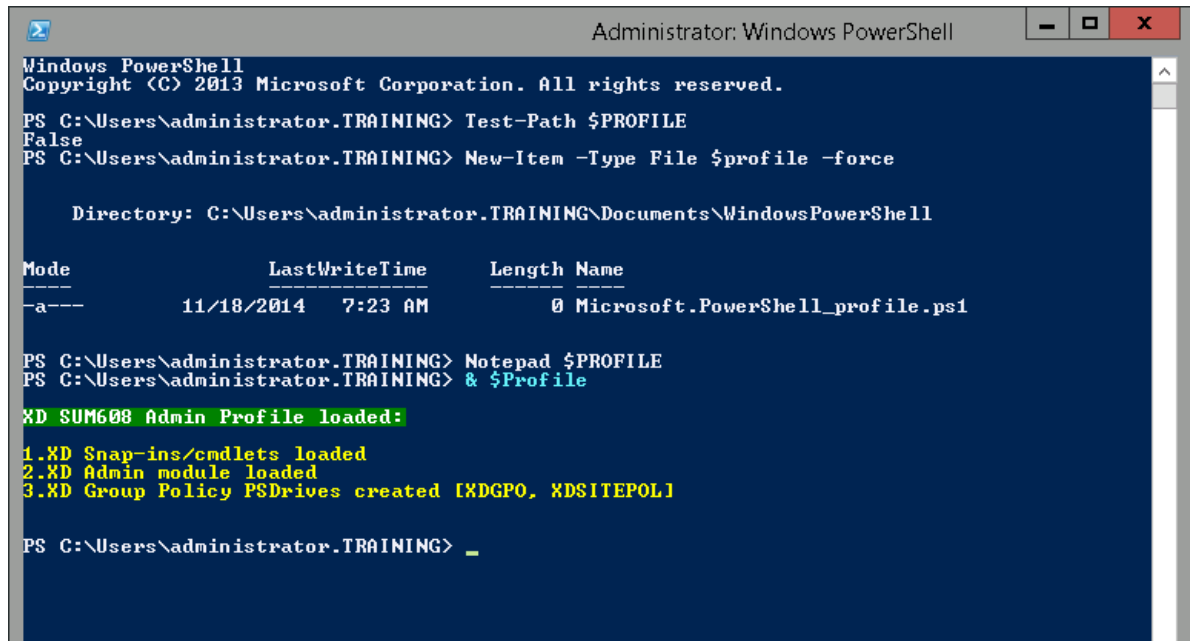
```
" "
```

```
" "
```



Note: Some lines above are nulled out using **\$null =** to hide the return output. This will ensure that the Citrix Group Policy PoSH drives (**PSDrives**) are available at any given time to create and manage AD & Site HDX policies.

7. Type **& \$Profile** and hit return to reload the newly created PoSH Profile:



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2013 Microsoft Corporation. All rights reserved.

PS C:\Users\administrator.TRAINING> Test-Path $PROFILE
False
PS C:\Users\administrator.TRAINING> New-Item -Type File $profile -force

Directory: C:\Users\administrator.TRAINING\Documents\WindowsPowerShell

Mode                LastWriteTime         Length Name
----                -
-a---             11/18/2014   7:23 AM              0 Microsoft.PowerShell_profile.ps1

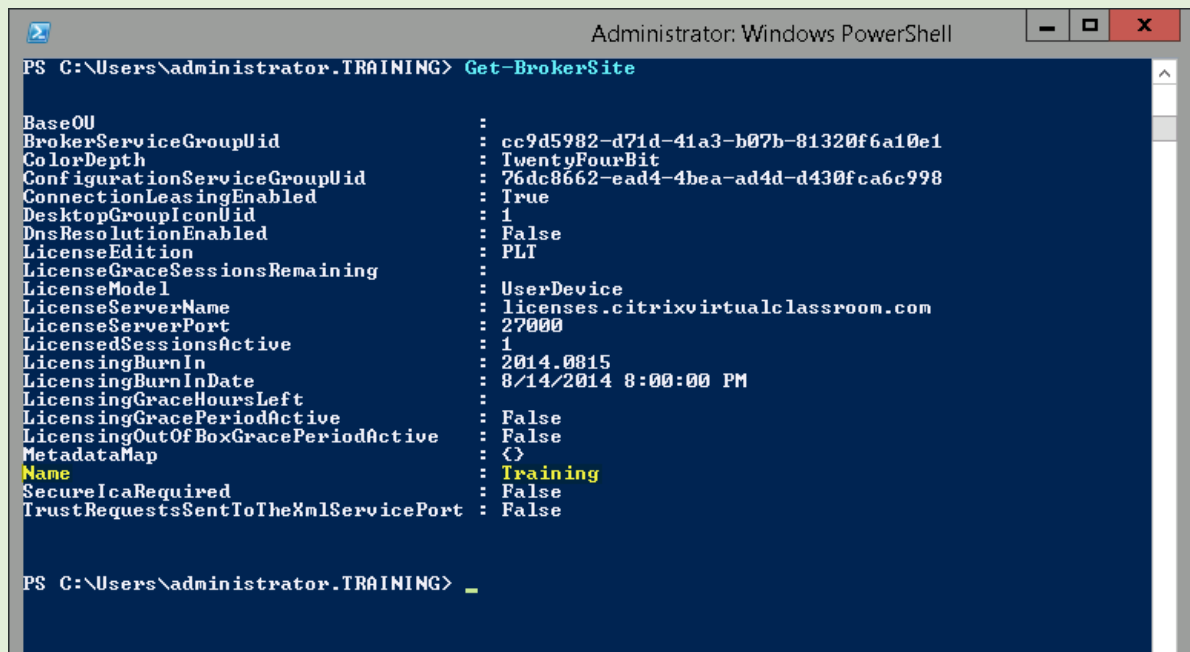
PS C:\Users\administrator.TRAINING> Notepad $PROFILE
PS C:\Users\administrator.TRAINING> & $Profile

XD SUM608 Admin Profile loaded:
1.XD Snap-ins/cmdlets loaded
2.XD Admin module loaded
3.XD Group Policy PSDrives created [XDGP0, XDSITEPOL]

PS C:\Users\administrator.TRAINING> _
```

Note: The call operator (&), also known as the "invocation operator," lets you run commands that are stored in variables and represented by strings.

8. Test your newly created PoSH Profile by typing **Get-BrokerSite** and hitting return:



```
Administrator: Windows PowerShell

PS C:\Users\administrator.TRAINING> Get-BrokerSite

BaseOU :
BrokerServiceGroupId : cc9d5982-d71d-41a3-b07b-81320f6a10e1
ColorDepth : TwentyFourBit
ConfigurationServiceGroupId : 76dc8662-ea4d-4bea-ad4d-d430fca6c998
ConnectionLeasingEnabled : True
DesktopGroupIconUid : 1
DnsResolutionEnabled : False
LicenseEdition : PLT
LicenseGraceSessionsRemaining :
LicenseModel : UserDevice
LicenseServerName : licenses.citrixvirtualclassroom.com
LicenseServerPort : 27000
LicensedSessionsActive : 1
LicensingBurnIn : 2014.0815
LicensingBurnInDate : 8/14/2014 8:00:00 PM
LicensingGraceHoursLeft :
LicensingGracePeriodActive : False
LicensingOutOfBoxGracePeriodActive : False
MetadataMap : {}
Name : Training
SecureIcaRequired : False
TrustRequestsSentToTheXmlServicePort : False

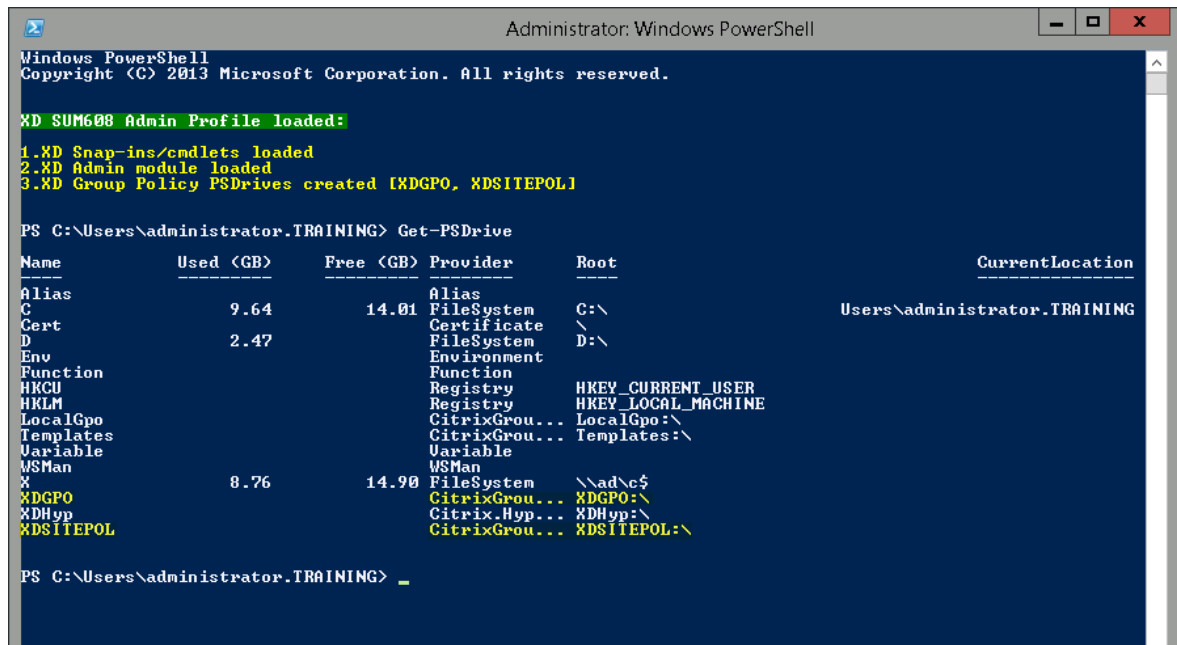
PS C:\Users\administrator.TRAINING> _
```

Note: All XenDesktop snap-ins are now loaded as per the [asnp citrix*](#) command saved within your PoSH Profile and therefore all XA/XD 7.6 **cmdlets** are available for use, each time a PoSH instance is launched.

Get-BrokerSite returns a number of key details about the **Training** site.

9. To validate the other functionality loaded through the PoSH Profile, close and reopen the existing PowerShell instance and type the following and hit return:

Get-PSDrive (Newly created Citrix Group Policy PoSH drives as per the newly created PoSH Profile & Citrix Group Policy PS-Provider will be displayed)



```
Administrator: Windows PowerShell

Windows PowerShell
Copyright (C) 2013 Microsoft Corporation. All rights reserved.

XD SUM608 Admin Profile loaded:
1.XD Snap-ins/cmdlets loaded
2.XD Admin module loaded
3.XD Group Policy PSDrives created [XDGP0, XDSITEPOL]

PS C:\Users\administrator.TRAINING> Get-PSDrive

Name          Used <GB>  Free <GB>  Provider      Root              CurrentLocation
-----
Alias          9.64      14.01     Alias         C:\              Users\administrator.TRAINING
C              2.47      FileSystem \
Cert           2.47      FileSystem D:\
Env            Environment
Function       Function
HKCU           Registry  HKEY_CURRENT_USER
HKLM           Registry  HKEY_LOCAL_MACHINE
LocalGpo       CitrixGrou... LocalGpo:\
Templates     CitrixGrou... Templates:\
Variable       Variable
WSMan          8.76      14.90     FileSystem    \\ad\c$
XDGP0         CitrixGrou... XDGP0:\
XDHyp         Citrix.Hyp... XDHyp:\
XDSITEPOL     CitrixGrou... XDSITEPOL:\

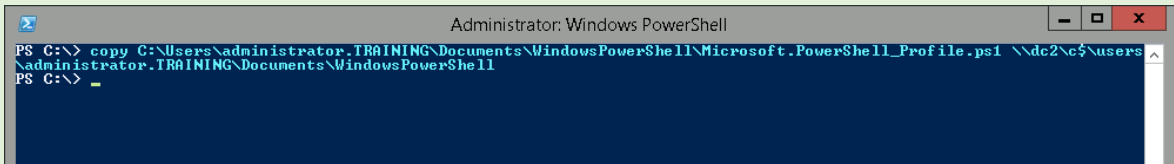
PS C:\Users\administrator.TRAINING> _
```

Note: To fully initiate the new Citrix Group Policy PoSH drives, you must restart PowerShell. Using the call operator i.e. **& \$Profile** will not suffice here.

 <http://blogs.citrix.com/2013/10/01/xd-tipster-creating-hdx-policies-through-posh/>

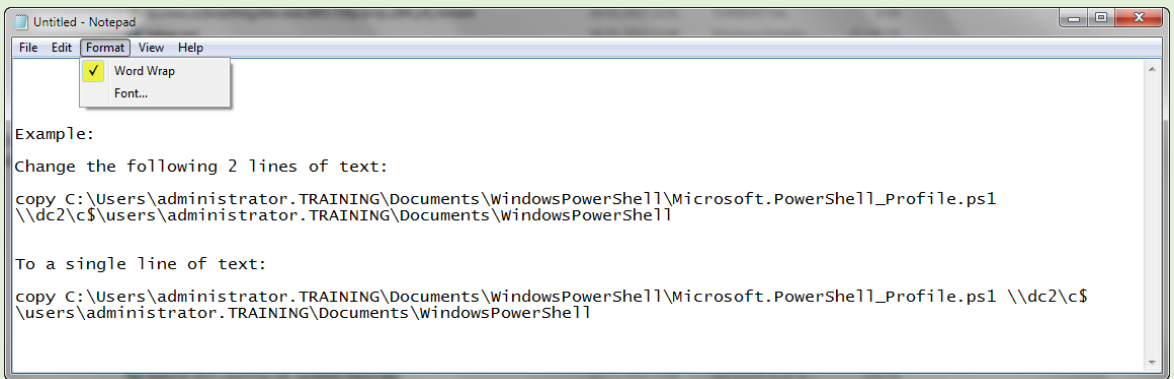
10. Copy the newly created PoSH Profile to **DC2** using the following string:

copy C:\Users\administrator.TRAINING\Documents\WindowsPowerShell\Microsoft.PowerShell_Profile.ps1
\\dc2\c\$\users\administrator.TRAINING\Documents\WindowsPowerShell



```
Administrator: Windows PowerShell
PS C:\> copy C:\Users\administrator.TRAINING\Documents\WindowsPowerShell\Microsoft.PowerShell_Profile.ps1 \\dc2\c$\users\administrator.TRAINING\Documents\WindowsPowerShell
PS C:\> _
```

IMPORTANT: If using Copy and paste you will need to use an intermediate application like Notepad.exe to re-format the above string and all subsequent strings transcending more than one line, so that they are treated as single lines of text within PowerShell. Please bear this in mind as you work through the remainder of the exercises...



Example:

Change the following 2 lines of text:

```
copy C:\Users\administrator.TRAINING\Documents\WindowsPowerShell\Microsoft.PowerShell_Profile.ps1  
\\dc2\c$\users\administrator.TRAINING\Documents\WindowsPowerShell
```

To a single line of text:

```
copy C:\Users\administrator.TRAINING\Documents\WindowsPowerShell\Microsoft.PowerShell_Profile.ps1 \\dc2\c$\users\administrator.TRAINING\Documents\WindowsPowerShell
```

11. Congratulations, you have finished this lab exercise.

Exercise Summary

Takeaways from this exercise:

- During this first exercise we learned how to customize a working XenDesktop PoSH environment through the use of a PoSH Profile. Similar to a User Profile, a PoSH Profile loads each time an instance of PoSH is launched and configures the PoSH env as per the user's unique needs.
- For more information about creating a customized XD PoSH Profile see the following XD Tipster blog post:
<http://blogs.citrix.com/2014/02/04/xd-tipster-creating-a-customized-xd-posh-profile/>
- For more cool info & tips on XenDesktop follow **@XDtipster** and **@XDInformer** on twitter.

Exercise 2

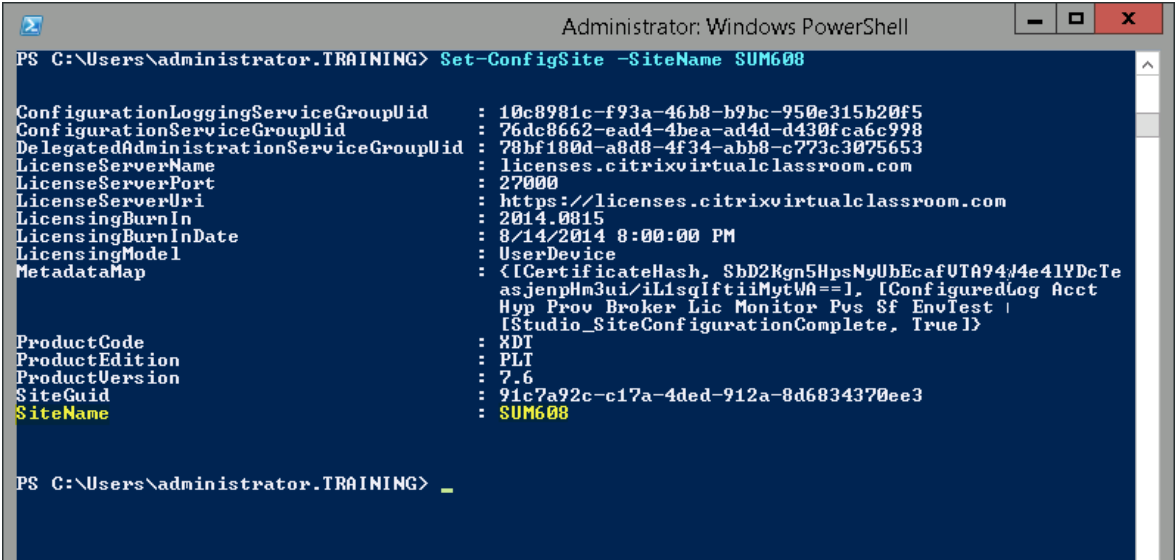
A Little Invention: Manually Adding a Controller to a Site

Overview

In this exercise you will learn how to manually add a controller to a site DB; simulating a situation where you do not have an existing Controller to facilitate the process. In the real world, this may be required when restoring a customer's DB in house or in Disaster Recovery situations.

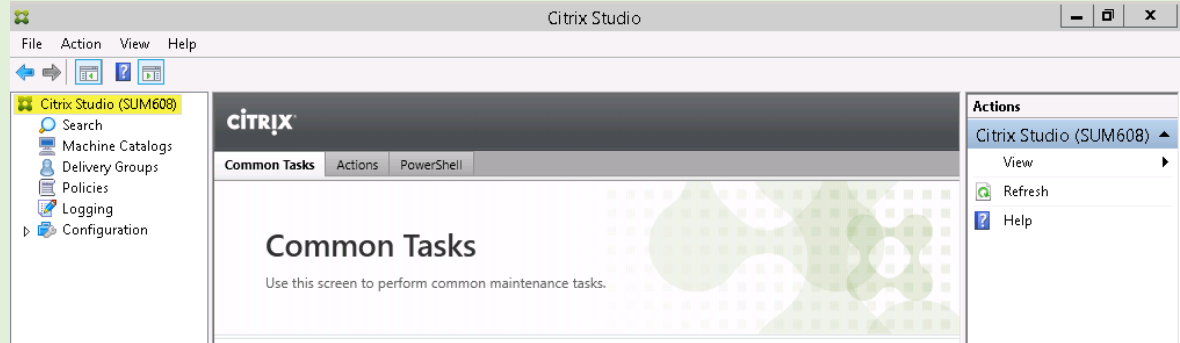
Step by step guidance

Estimated time to complete this lab: 45 minutes.

Step	Action
1.	<p>Before we carry out the steps to manually add a Controller to our XenDesktop Site, let's rename our existing site to something a little more appropriate...</p> <p>From within the existing PowerShell window on DC1, type Set-ConfigSite -SiteName SUM608 and hit return to change the existing site name from Training to SUM608:</p> 

2. **Observation:** Once changed through PowerShell, the new site name will also be reflected in Citrix Studio:

Go ahead and confirm this yourself by launching Citrix Studio...

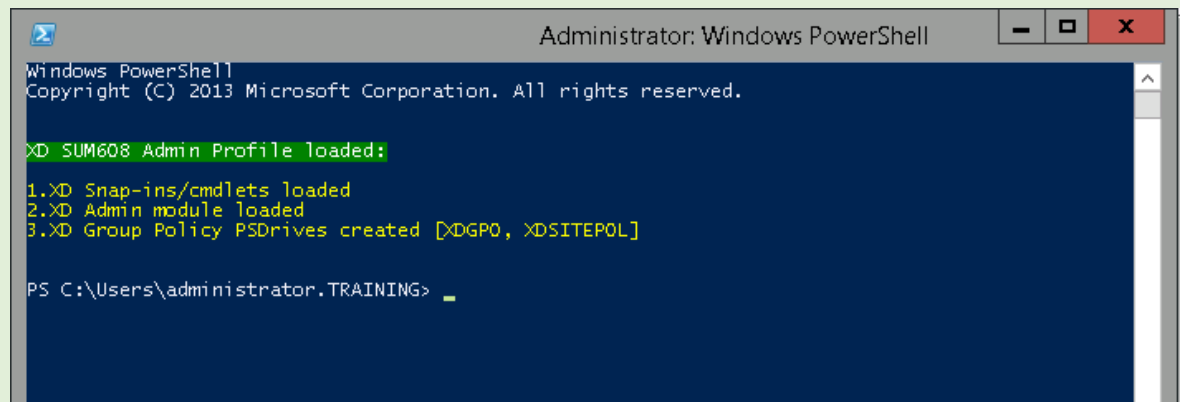


Note: Changing the Site name through PowerShell has no impact on the actual Site Database name itself which will remain the same. It is very important to make the distinction here, between the logical site name i.e. **SUM68** and the actual Site Database name i.e. **CitrixSummit608**.

3. Login to **DC2** as **training\administrator** with password: **Citrix123** and launch a PowerShell window by clicking on the **PoSH** icon on the taskbar:



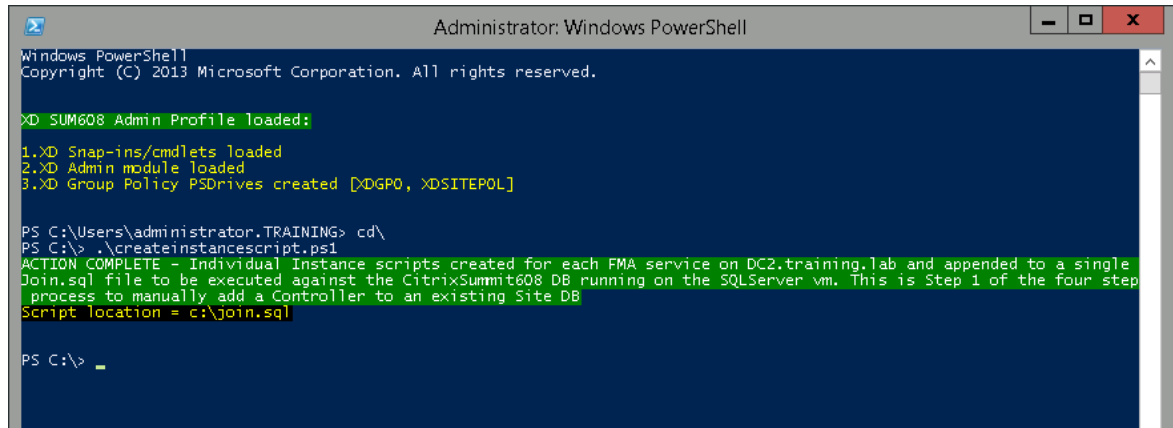
4. **Observation:** Our new PoSH Profile loads automatically as expected based on step 10 in exercise 1 where the newly created PoSH Profile was copied to **DC2**:



5. Run the following simple script from the root of PoSH: (To get to the root just type **cd** & hit return)

C:\> .\createinstancescript.ps1

Reminder: You should be running this script from DC2



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2013 Microsoft Corporation. All rights reserved.

> .\createinstancescript.ps1
XD SUM608 Admin Profile loaded:
1.XD Snap-ins/cmdlets loaded
2.XD Admin module loaded
3.XD Group Policy PSDrives created [XDGPO, XDSITEPOL]

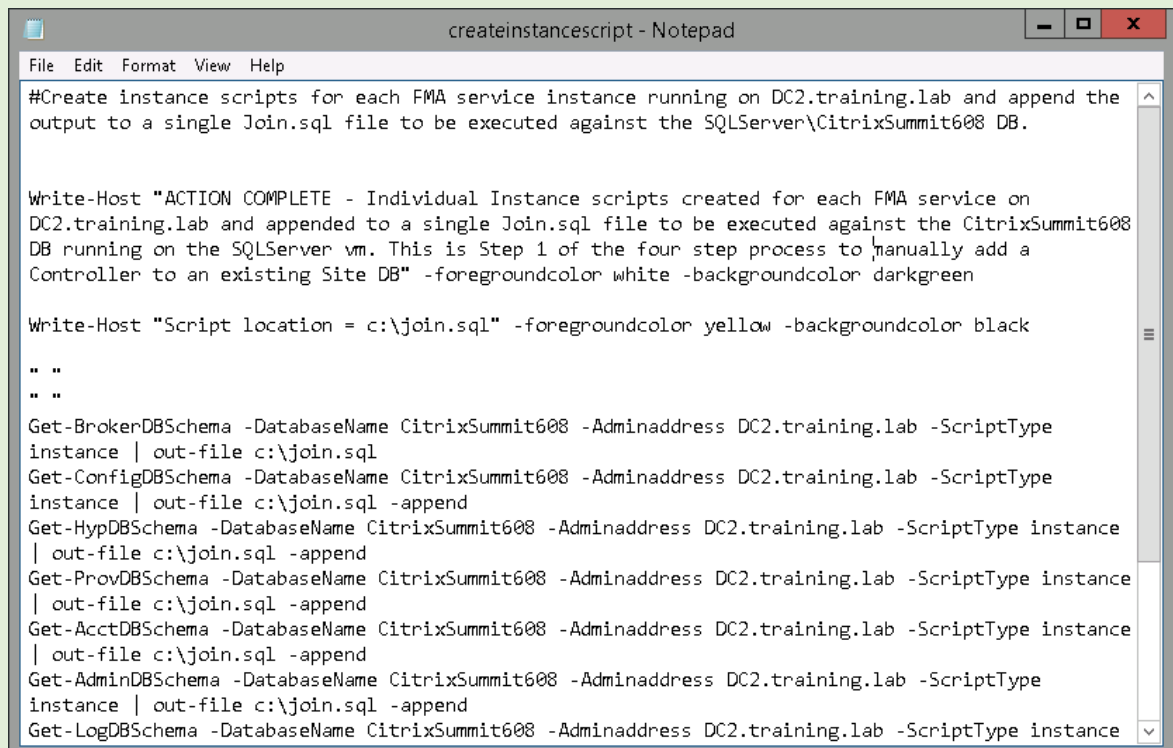
PS C:\Users\administrator.TRAINING> cd\
PS C:\> .\createinstancescript.ps1
ACTION COMPLETE - Individual Instance scripts created for each FMA service on DC2.training.lab and appended to a single
Join.sql file to be executed against the CitrixSummit608 DB running on the SQLServer vm. This is Step 1 of the four step
process to manually add a Controller to an existing Site DB
Script location = c:\join.sql

PS C:\> _
```

6. Browse to the root of **DC2** and open **createinstancescript.ps1** with notepad to view the contents and script structure:

Note: The **Get-<serviceALIAS>DBSchema** cmdlet can be used to generate an **instance** script to add each service instance running on **DC2** to the site DB (**CitrixSummit608**)

To optimize the process, you can see that we are using the **| out-file -append** switch so all service instance scripts are merged into a single .sql file. (Makes sense...☺)



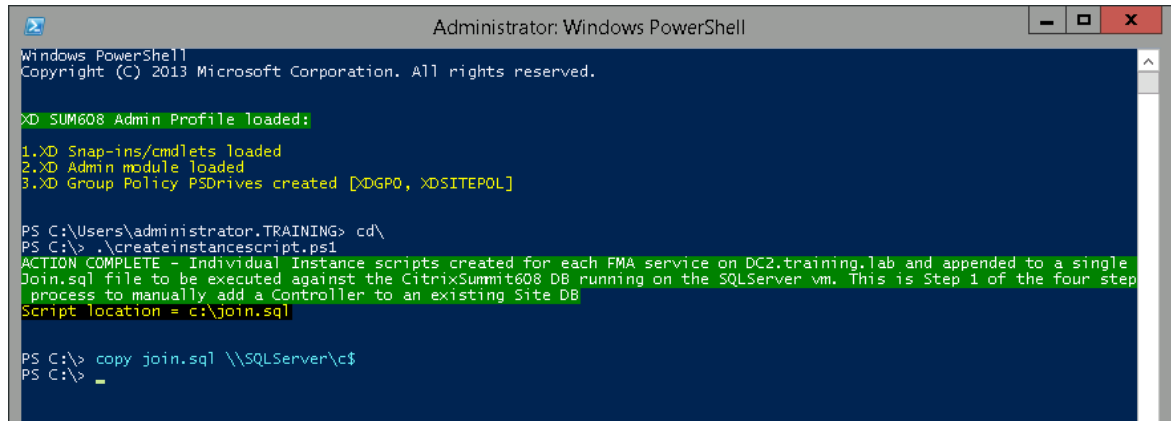
```
createinstancescript - Notepad
File Edit Format View Help
#Create instance scripts for each FMA service instance running on DC2.training.lab and append the
output to a single Join.sql file to be executed against the SQLServer\CitrixSummit608 DB.

Write-Host "ACTION COMPLETE - Individual Instance scripts created for each FMA service on
DC2.training.lab and appended to a single Join.sql file to be executed against the CitrixSummit608
DB running on the SQLServer vm. This is Step 1 of the four step process to manually add a
Controller to an existing Site DB" -foregroundcolor white -backgroundcolor darkgreen

Write-Host "Script location = c:\join.sql" -foregroundcolor yellow -backgroundcolor black

" "
" "
Get-BrokerDBSchema -DatabaseName CitrixSummit608 -Adminaddress DC2.training.lab -ScriptType
instance | out-file c:\join.sql
Get-ConfigDBSchema -DatabaseName CitrixSummit608 -Adminaddress DC2.training.lab -ScriptType
instance | out-file c:\join.sql -append
Get-HypDBSchema -DatabaseName CitrixSummit608 -Adminaddress DC2.training.lab -ScriptType instance
| out-file c:\join.sql -append
Get-ProvDBSchema -DatabaseName CitrixSummit608 -Adminaddress DC2.training.lab -ScriptType instance
| out-file c:\join.sql -append
Get-AcctDBSchema -DatabaseName CitrixSummit608 -Adminaddress DC2.training.lab -ScriptType instance
| out-file c:\join.sql -append
Get-AdminDBSchema -DatabaseName CitrixSummit608 -Adminaddress DC2.training.lab -ScriptType
instance | out-file c:\join.sql -append
Get-LogDBSchema -DatabaseName CitrixSummit608 -Adminaddress DC2.training.lab -ScriptType instance
```

7. Type ***copy join.sql \\SQLServer\c\$*** and hit return to copy the newly created join script to the root of the **SQLServer VM**:



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2013 Microsoft Corporation. All rights reserved.

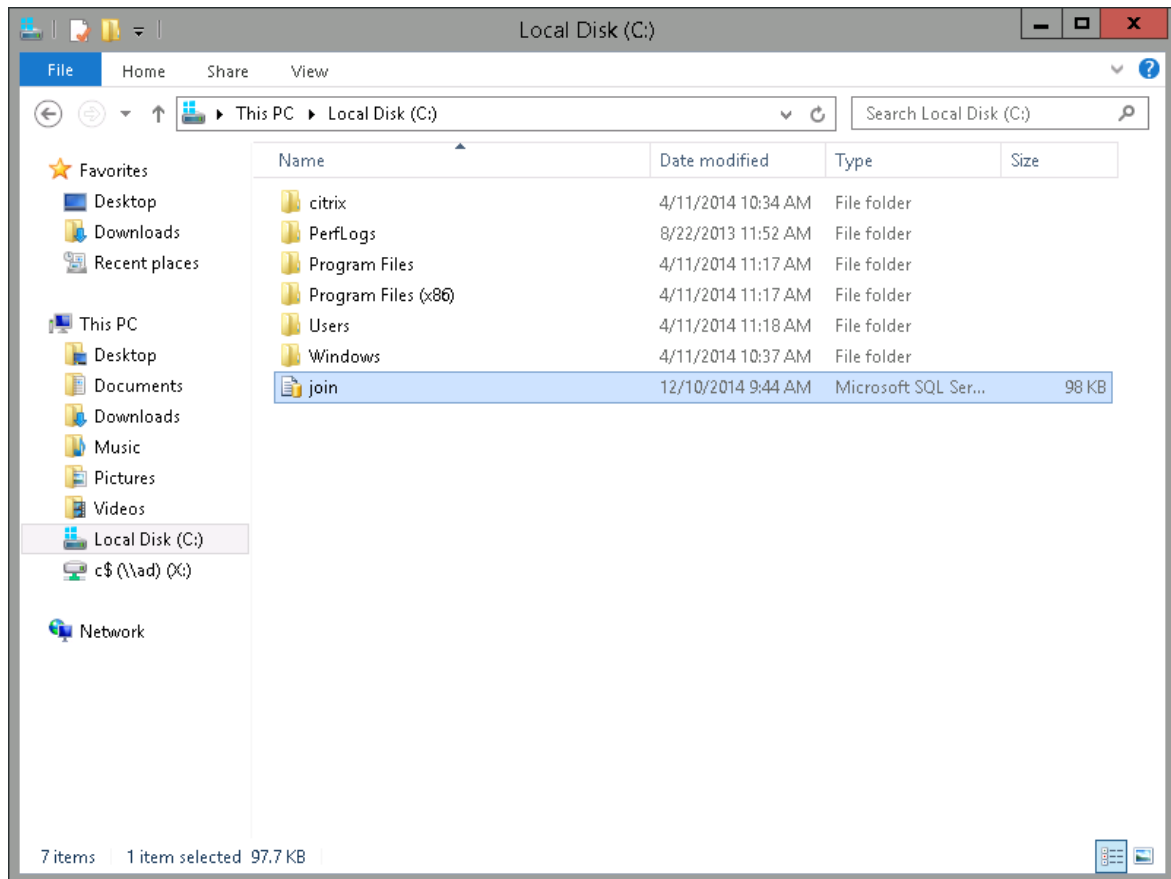
>> SUM608 Admin Profile loaded:
1.> Snap-ins/cmdlets loaded
2.> Admin module loaded
3.> Group Policy PSDrives created [DGPO, >SITEPOL]

PS C:\Users\administrator.TRAINING> cd\
PS C:\> .\createinstancescript.ps1
ACTION COMPLETE - Individual Instance scripts created for each FMA service on DC2.training.lab and appended to a single
Join.sql file to be executed against the CitrixSummit608 DB running on the SQLServer vm. This is Step 1 of the four step
process to manually add a Controller to an existing Site DB
Script location = c:\join.sql

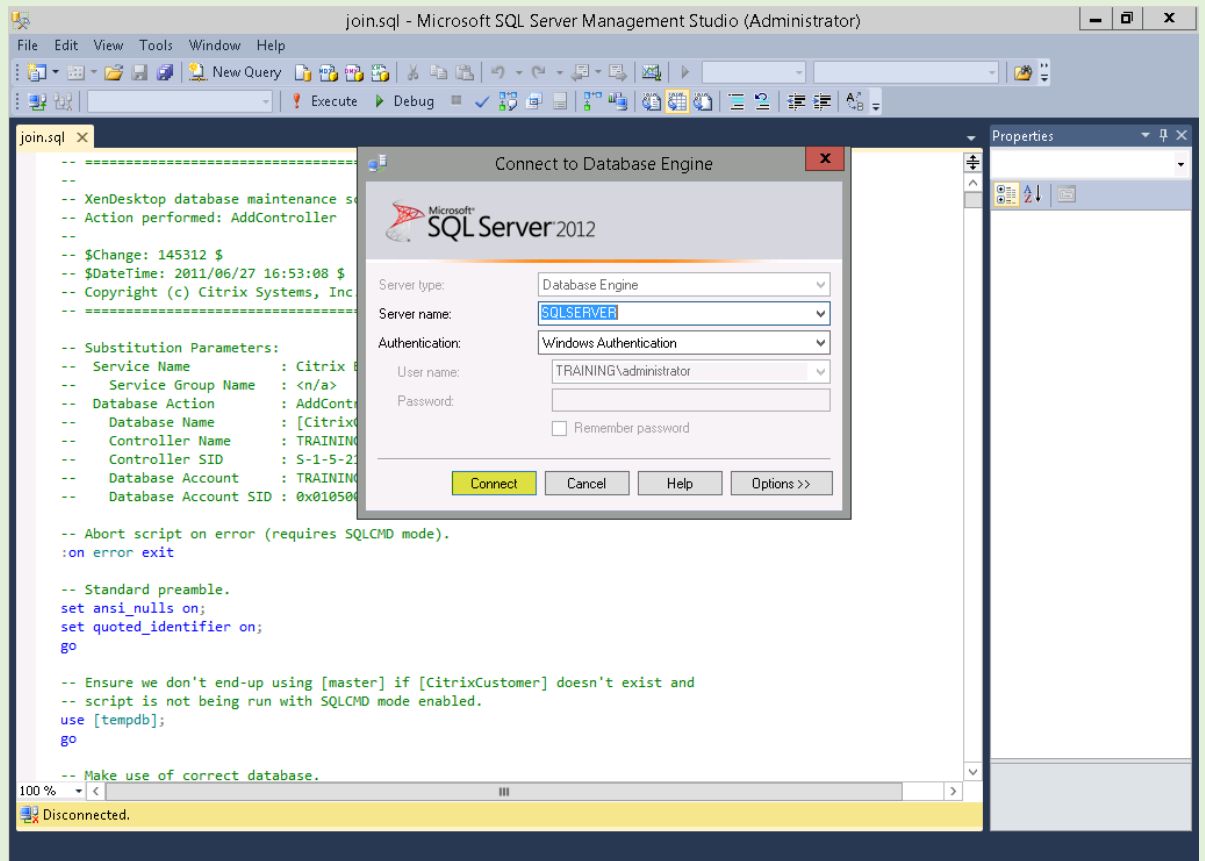
PS C:\> copy join.sql \\SQLServer\c$
PS C:\>
```

8. Login to **SQLServer** as **training\administrator** with password **Citrix123**

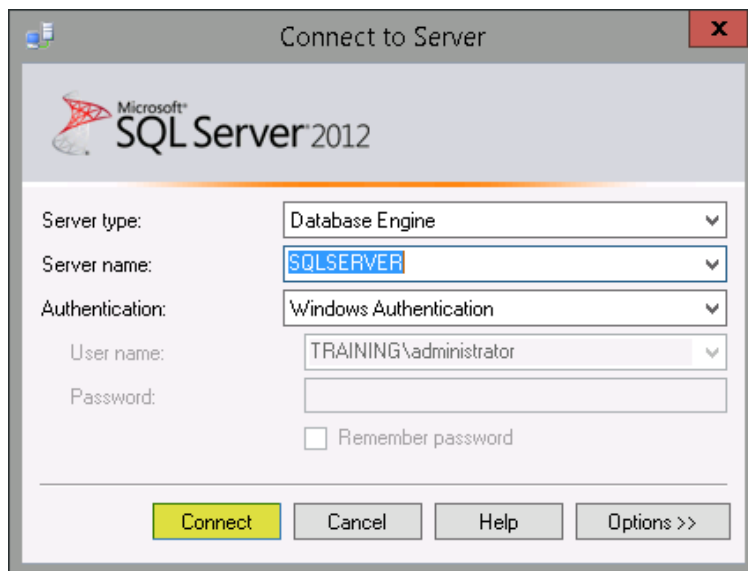
9. Browse the local disk on **SQLServer** and double click on the **Join.sql** file:



10. Click **Connect** on the Database engine authentication dialog box:



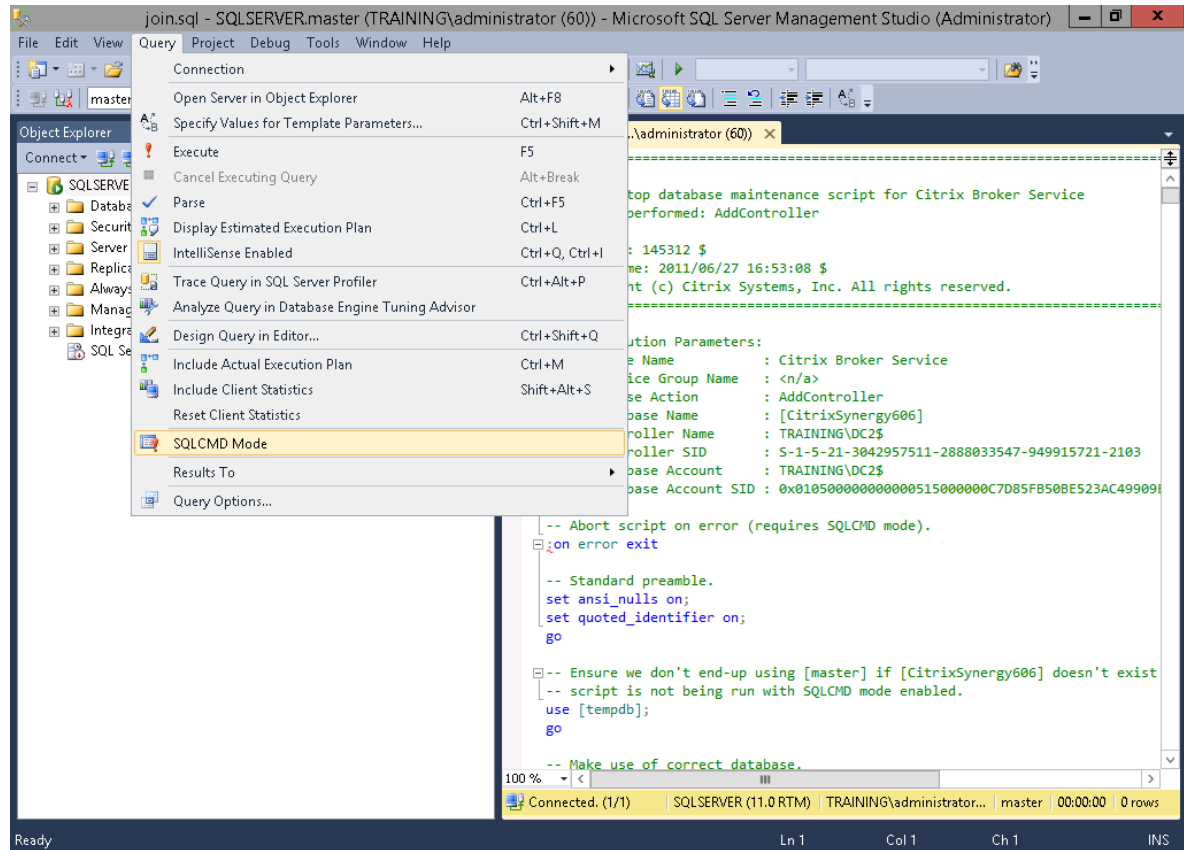
11. Click **Connect** again to authenticate to the Server:



12. Click inside the **Join.sql** script window.

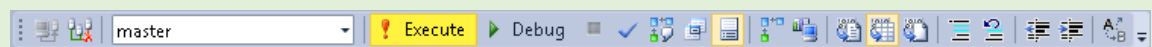
13. Click on **Query** and select **SQLCMD Mode**:

Note: If you review the comments in the script you will see that enabling **SQLCMD** mode is advised for the purposes of error handling.



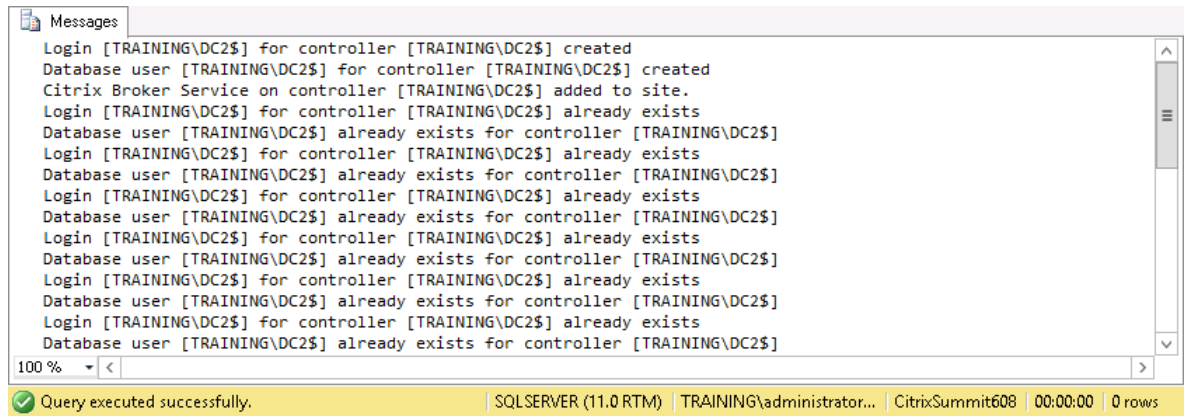
14. Press **F5** or the Execute Icon on the Toolbar to execute the **Join.sql** script

Note: There is no requirement to select the **CitrixSummit608** DB to run the script against as the script itself determines this.



15. **Observation:** The Script runs without errors and does the following:

1. Creates DB login for new Controller (Training\DC2\$)
2. Adds all FMA services to the SUM608 site



The screenshot shows the 'Messages' window in SQL Server Enterprise Manager. It displays a list of messages indicating that a script was executed successfully. The messages include: 'Login [TRAINING\DC2\$] for controller [TRAINING\DC2\$] created', 'Database user [TRAINING\DC2\$] for controller [TRAINING\DC2\$] created', 'Citrix Broker Service on controller [TRAINING\DC2\$] added to site.', and several 'Login [TRAINING\DC2\$] for controller [TRAINING\DC2\$] already exists' and 'Database user [TRAINING\DC2\$] already exists for controller [TRAINING\DC2\$]' messages. At the bottom, a yellow status bar indicates 'Query executed successfully.' and shows details like 'SQLSERVER (11.0 RTM)', 'TRAINING\administrator...', 'CitrixSummit608', '00:00:00', and '0 rows'.

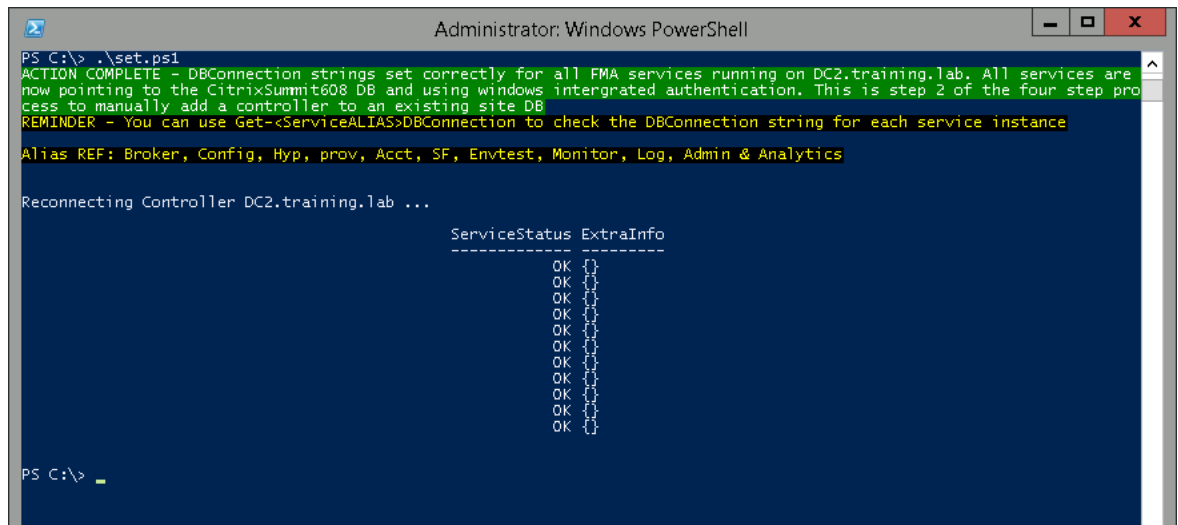
16. Close SQL management Studio.

17. Now that all the FMA services on **DC2** have been successfully added to the **SUM608** site DB, we need to set the **DBConnection** string for each so they point to the correct Database.

To do this, switch back to **DC2** and run the following simple script from the root of PoSH:
(To get to the root just type **cd** & hit return)

C:> .\set.ps1

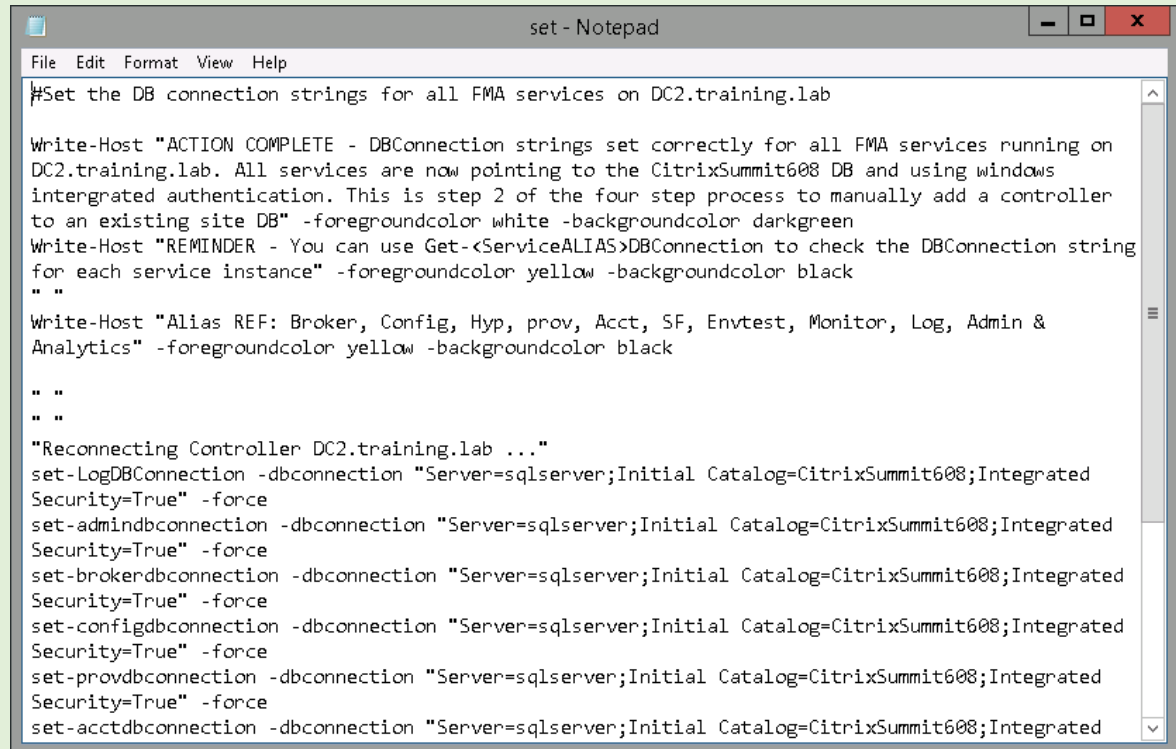
Reminder: You should be running this script from DC2



The screenshot shows a Windows PowerShell console window titled 'Administrator: Windows PowerShell'. The prompt is 'PS C:\>'. The user has entered '.\set.ps1'. The script output includes: 'ACTION COMPLETE - DBConnection strings set correctly for all FMA services running on DC2.training.lab. All services are now pointing to the CitrixSummit608 DB and using windows integrated authentication. This is step 2 of the four step process to manually add a controller to an existing site DB', 'REMINDER - You can use Get-ServiceALIAS>DBConnection to check the DBConnection string for each service instance', and 'Alias REF: Broker, Config, Hyp, prov, Acct, SF, Envtest, Monitor, Log, Admin & Analytics'. It then shows 'Reconnecting Controller DC2.training.lab ...' followed by a table with two columns: 'ServiceStatus' and 'ExtraInfo'. The table contains 10 rows, all with 'OK' in the 'ServiceStatus' column and empty 'ExtraInfo' columns. The prompt 'PS C:\>' is visible at the bottom.

18. Browse to the root of **DC2** and open **set.ps1** with notepad to view the contents and script structure:

Note: The **Set-*<serviceALIAS>*DBConnection** cmdlet is used again here but as before, rather than setting the value to **\$null**, we are setting a complete string to communicate with the **SUM608** site DB.



```
File Edit Format View Help
#Set the DB connection strings for all FMA services on DC2.training.lab

Write-Host "ACTION COMPLETE - DBConnection strings set correctly for all FMA services running on
DC2.training.lab. All services are now pointing to the CitrixSummit608 DB and using windows
intergrated authentication. This is step 2 of the four step process to manually add a controller
to an existing site DB" -foregroundcolor white -backgroundcolor darkgreen
Write-Host "REMINDER - You can use Get-<ServiceALIAS>DBConnection to check the DBConnection string
for each service instance" -foregroundcolor yellow -backgroundcolor black
" "
Write-Host "Alias REF: Broker, Config, Hyp, prov, Acct, SF, Envtest, Monitor, Log, Admin &
Analytics" -foregroundcolor yellow -backgroundcolor black
" "
"Reconnecting Controller DC2.training.lab ..."
set-LogDBConnection -dbconnection "Server=sqlserver;Initial Catalog=CitrixSummit608;Integrated
Security=True" -force
set-admindbconnection -dbconnection "Server=sqlserver;Initial Catalog=CitrixSummit608;Integrated
Security=True" -force
set-brokerdbconnection -dbconnection "Server=sqlserver;Initial Catalog=CitrixSummit608;Integrated
Security=True" -force
set-configdbconnection -dbconnection "Server=sqlserver;Initial Catalog=CitrixSummit608;Integrated
Security=True" -force
set-provdbconnection -dbconnection "Server=sqlserver;Initial Catalog=CitrixSummit608;Integrated
Security=True" -force
set-acctdbconnection -dbconnection "Server=sqlserver;Initial Catalog=CitrixSummit608;Integrated
```

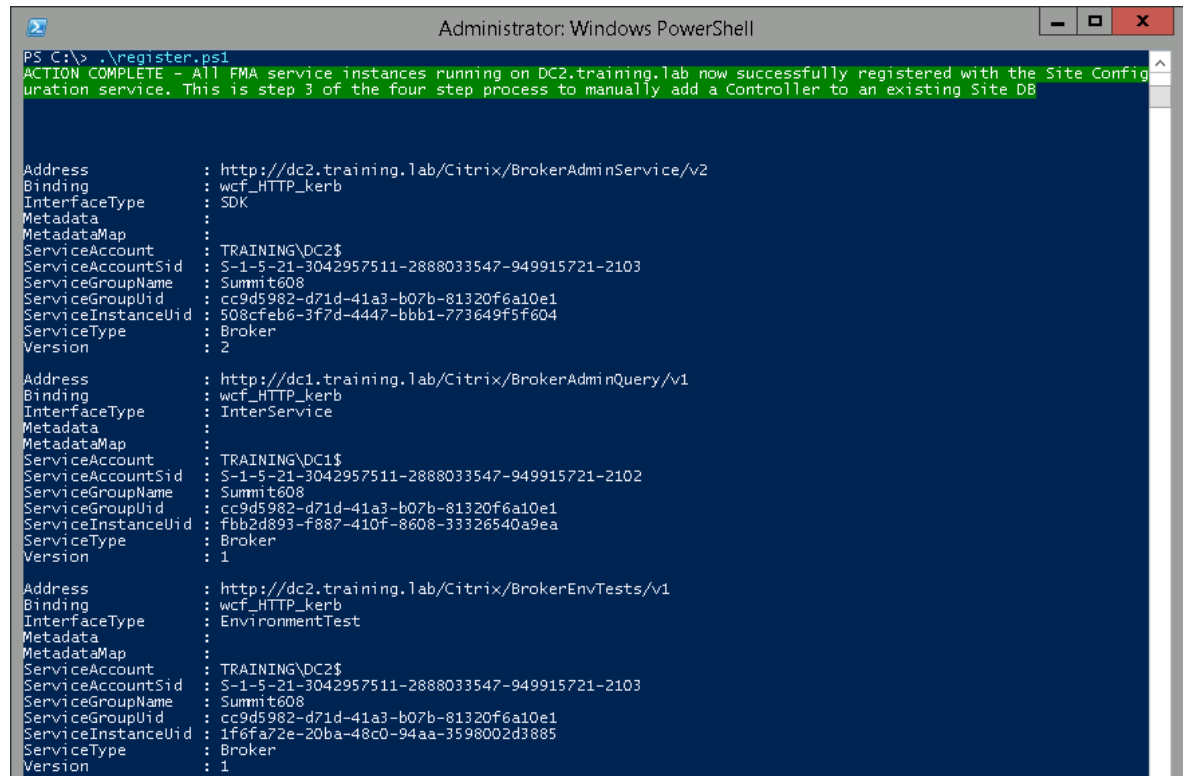
19. Now that we have added the services to the site DB and Set the DBConnection strings, we need to register each service instance on **DC2** with the site configuration service.

The configuration service acts as a directory services listing for the site and to be fully functional members of the site, each FMA service has to advertise its communication address(s) with the configuration service.

To do this just run the following simple script from the root of PoSH: (To get to the root just type **cd** & hit return)

C:\> .\register.ps1

Reminder: You should be running this script from DC2

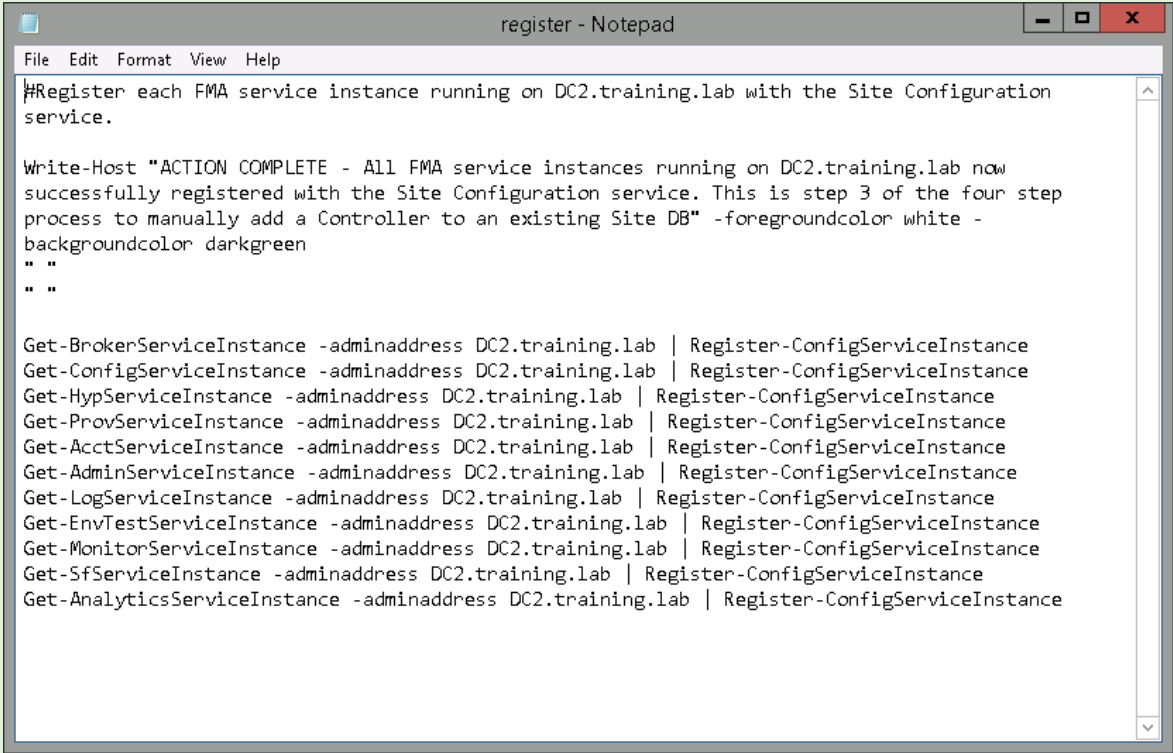


```
Administrator: Windows PowerShell
PS C:\> .\register.ps1
ACTION COMPLETE - All FMA service instances running on DC2.training.lab now successfully registered with the Site Configuration service. This is step 3 of the four step process to manually add a Controller to an existing Site DB

Address      : http://dc2.training.lab/Citrix/BrokerAdminService/v2
Binding      : wcf_HTTP_kerb
InterfaceType : SDK
Metadata     :
MetadataMap  :
ServiceAccount : TRAINING\DC2$
ServiceAccountSid : S-1-5-21-3042957511-2888033547-949915721-2103
ServiceGroupName : Summit608
ServiceGroupUid : cc9d5982-d71d-41a3-b07b-81320f6a10e1
ServiceInstanceUid : 508cfeb6-3f7d-4447-bbb1-773649f5f604
ServiceType   : Broker
Version       : 2

Address      : http://dc1.training.lab/Citrix/BrokerAdminQuery/v1
Binding      : wcf_HTTP_kerb
InterfaceType : InterService
Metadata     :
MetadataMap  :
ServiceAccount : TRAINING\DC1$
ServiceAccountSid : S-1-5-21-3042957511-2888033547-949915721-2102
ServiceGroupName : Summit608
ServiceGroupUid : cc9d5982-d71d-41a3-b07b-81320f6a10e1
ServiceInstanceUid : fbb2d893-f887-410f-8608-33326540a9ea
ServiceType   : Broker
Version       : 1

Address      : http://dc2.training.lab/Citrix/BrokerEnvTests/v1
Binding      : wcf_HTTP_kerb
InterfaceType : EnvironmentTest
Metadata     :
MetadataMap  :
ServiceAccount : TRAINING\DC2$
ServiceAccountSid : S-1-5-21-3042957511-2888033547-949915721-2103
ServiceGroupName : Summit608
ServiceGroupUid : cc9d5982-d71d-41a3-b07b-81320f6a10e1
ServiceInstanceUid : 1f6fa72e-20ba-48c0-94aa-3598002d3885
ServiceType   : Broker
Version       : 1
```

20.	<p>Browse to the root of DC2 and open register.ps1 with notepad to view the contents and script structure:</p> <p>Note: The Register-ConfigServiceInstance cmdlet can be used to register each service instance with the site configuration service. In the example below, we use the Get-<i><serviceALIAS></i>ServiceInstance cmdlet to return the specific service instance before piping the return value to the Register-ConfigServiceInstance cmdlet to register each specific service.</p>  <pre> #Register each FMA service instance running on DC2.training.lab with the Site Configuration service. Write-Host "ACTION COMPLETE - All FMA service instances running on DC2.training.lab now successfully registered with the Site Configuration service. This is step 3 of the four step process to manually add a Controller to an existing Site DB" -foregroundcolor white - backgroundcolor darkgreen " " " " Get-BrokerServiceInstance -adminaddress DC2.training.lab Register-ConfigServiceInstance Get-ConfigServiceInstance -adminaddress DC2.training.lab Register-ConfigServiceInstance Get-HypServiceInstance -adminaddress DC2.training.lab Register-ConfigServiceInstance Get-ProvServiceInstance -adminaddress DC2.training.lab Register-ConfigServiceInstance Get-AcctServiceInstance -adminaddress DC2.training.lab Register-ConfigServiceInstance Get-AdminServiceInstance -adminaddress DC2.training.lab Register-ConfigServiceInstance Get-LogServiceInstance -adminaddress DC2.training.lab Register-ConfigServiceInstance Get-EnvTestServiceInstance -adminaddress DC2.training.lab Register-ConfigServiceInstance Get-MonitorServiceInstance -adminaddress DC2.training.lab Register-ConfigServiceInstance Get-SfServiceInstance -adminaddress DC2.training.lab Register-ConfigServiceInstance Get-AnalyticsServiceInstance -adminaddress DC2.training.lab Register-ConfigServiceInstance </pre>
21.	<p>Taking Stock: Now that 3 of the 4 steps required to manually add a Controller to an existing site are complete we can carry out the final task. Before we do this, let's recap. The following 3 steps have now been completed:</p> <ol style="list-style-type: none"> 1. All FMA services on DC2 have been added to the SUM608 site DB. 2. DBConnection strings have been set for all FMA services on DC2. 3. All FMA services on DC2 have been registered against the site configuration service.

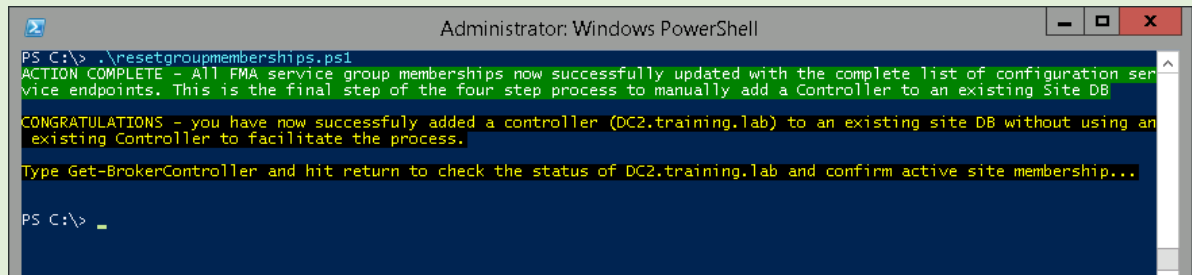
22.

To complete the task we need to finally update all FMA service group memberships in the Site with the complete and updated list of all configuration service endpoints. Essentially we need to let every FMA service group know that there is now more than one configuration service in the site. In fact, we now have two i.e. one running on **DC1** and one running on **DC2** (This is very important so as not to create a single point of failure in terms of the Configuration service instances available to each FMA service instance at run-time).

To do this just run the following simple script from the root of PoSH: (To get to the root just type **cd** & hit return)

C:\> .\resetgroupmemberships.ps1

Reminder: You should be running this script from DC2



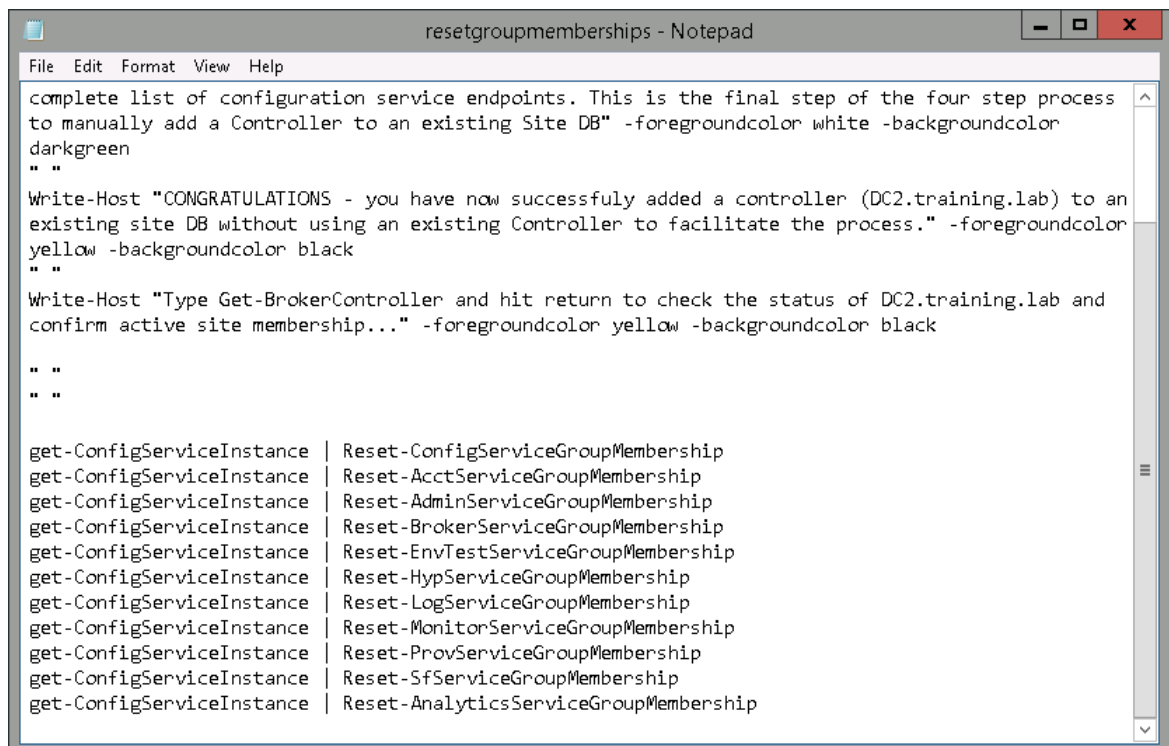
```

Administrator: Windows PowerShell
PS C:\> .\resetgroupmemberships.ps1
ACTION COMPLETE - All FMA service group memberships now successfully updated with the complete list of configuration service endpoints. This is the final step of the four step process to manually add a Controller to an existing Site DB
CONGRATULATIONS - you have now successfully added a controller (DC2.training.lab) to an existing site DB without using an existing Controller to facilitate the process.
Type Get-BrokerController and hit return to check the status of DC2.training.lab and confirm active site membership...
PS C:\>
  
```

23.

Browse to the root of **DC2** and open **resetgroupmemberships.ps1** with notepad to view the contents and script structure:

Note: The **Get-ConfigServiceInstance** cmdlet can be used to return a list of all available configuration service instances in the SUM608 site. In the example below, we pipe this information and reset each service group membership so that every service instance of the same type across all Controllers will have the full updated list of all configuration service instances in the site.



```

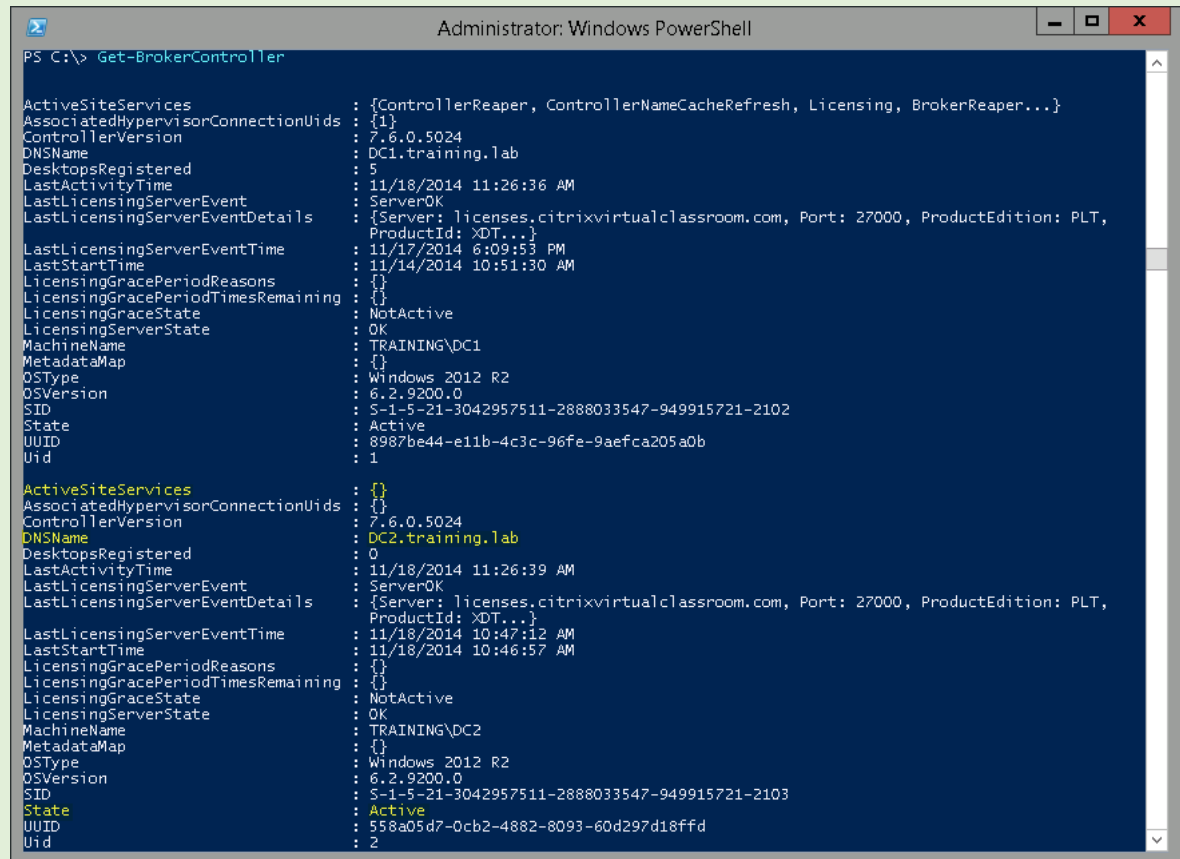
resetgroupmemberships - Notepad
File Edit Format View Help
complete list of configuration service endpoints. This is the final step of the four step process
to manually add a Controller to an existing Site DB" -foregroundColor white -backgroundColor
darkgreen
" "
Write-Host "CONGRATULATIONS - you have now successfully added a controller (DC2.training.lab) to an
existing site DB without using an existing Controller to facilitate the process." -foregroundColor
yellow -backgroundColor black
" "
Write-Host "Type Get-BrokerController and hit return to check the status of DC2.training.lab and
confirm active site membership..." -foregroundColor yellow -backgroundColor black

" "
" "

get-ConfigServiceInstance | Reset-ConfigServiceGroupMembership
get-ConfigServiceInstance | Reset-AcctServiceGroupMembership
get-ConfigServiceInstance | Reset-AdminServiceGroupMembership
get-ConfigServiceInstance | Reset-BrokerServiceGroupMembership
get-ConfigServiceInstance | Reset-EnvTestServiceGroupMembership
get-ConfigServiceInstance | Reset-HypServiceGroupMembership
get-ConfigServiceInstance | Reset-LogServiceGroupMembership
get-ConfigServiceInstance | Reset-MonitorServiceGroupMembership
get-ConfigServiceInstance | Reset-ProvServiceGroupMembership
get-ConfigServiceInstance | Reset-SfServiceGroupMembership
get-ConfigServiceInstance | Reset-AnalyticsServiceGroupMembership
  
```

24. As per the script instructions above, type **Get-BrokerController** to check Site membership and confirm that **DC2** appears as **ACTIVE**.

Note: As **DC2** has just been added to the **SUM608** site, **DC1** holds all the Site Service leases. This will remain as is, until both controllers or the Broker service itself is restarted on each Controller. At this point the leasing process will start again with an even distribution of Site Services (Specific house-keeping roles leased dynamically and evenly across all site Controllers at run-time).



```
Administrator: Windows PowerShell
PS C:\> Get-BrokerController

ActiveSiteServices           : {ControllerReaper, ControllerNameCacheRefresh, Licensing, BrokerReaper...}
AssociatedHypervisorConnectionUids : {1}
ControllerVersion            : 7.6.0.5024
DNSName                      : DC1.training.lab
DesktopsRegistered           : 5
LastActivityTime              : 11/18/2014 11:26:36 AM
LastLicensingServerEvent      : ServerOK
LastLicensingServerEventDetails : {Server: licenses.citrixvirtualclassroom.com, Port: 27000, ProductEdition: PLT,
                                   ProductId: XDT...}
LastLicensingServerEventTime   : 11/17/2014 6:09:53 PM
LastStartTime                 : 11/14/2014 10:51:30 AM
LicensingGracePeriodReasons    : {}
LicensingGracePeriodTimesRemaining : {}
LicensingGraceState            : NotActive
LicensingServerState           : OK
MachineName                   : TRAINING\DC1
MetadataMap                   : {}
OSType                        : Windows 2012 R2
OSVersion                     : 6.2.9200.0
SID                           : S-1-5-21-3042957511-2888033547-949915721-2102
State                          : Active
UUID                          : 8987be44-e11b-4c3c-96fe-9aefca205a0b
Uid                            : 1

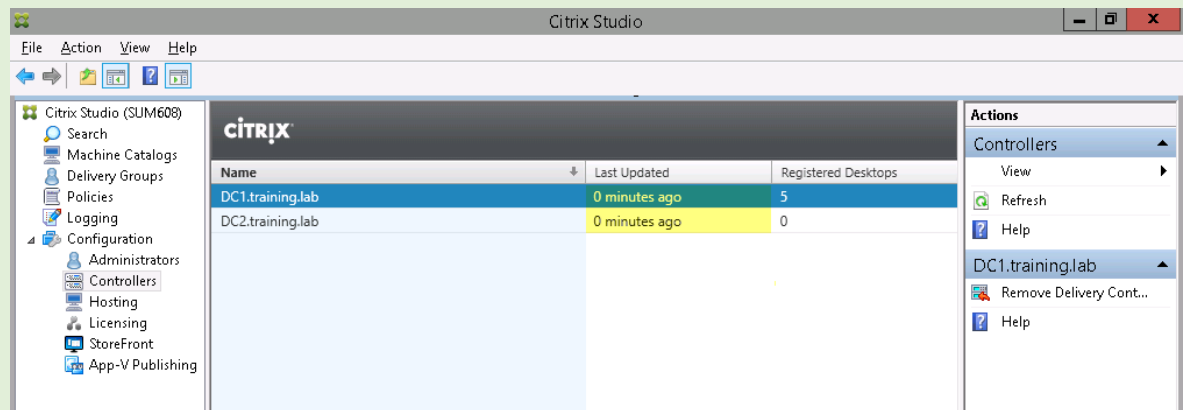
ActiveSiteServices           : {}
AssociatedHypervisorConnectionUids : {}
ControllerVersion            : 7.6.0.5024
DNSName                      : DC2.training.lab
DesktopsRegistered           : 0
LastActivityTime              : 11/18/2014 11:26:39 AM
LastLicensingServerEvent      : ServerOK
LastLicensingServerEventDetails : {Server: licenses.citrixvirtualclassroom.com, Port: 27000, ProductEdition: PLT,
                                   ProductId: XDT...}
LastLicensingServerEventTime   : 11/18/2014 10:47:12 AM
LastStartTime                 : 11/18/2014 10:46:57 AM
LicensingGracePeriodReasons    : {}
LicensingGracePeriodTimesRemaining : {}
LicensingGraceState            : NotActive
LicensingServerState           : OK
MachineName                   : TRAINING\DC2
MetadataMap                   : {}
OSType                        : Windows 2012 R2
OSVersion                     : 6.2.9200.0
SID                           : S-1-5-21-3042957511-2888033547-949915721-2103
State                          : Active
UUID                          : 558a05d7-0cb2-4882-8093-60d297d18ffd
Uid                            : 2
```

25. Launch **Citrix Studio** from the icon on the taskbar on **DC2**:



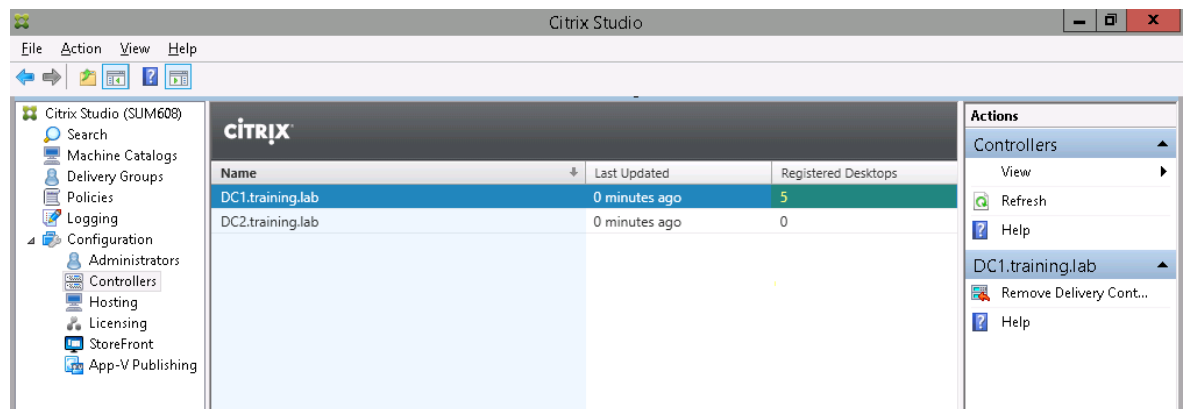
26. **Observation:** Both Controllers last update value = **0** as expected:

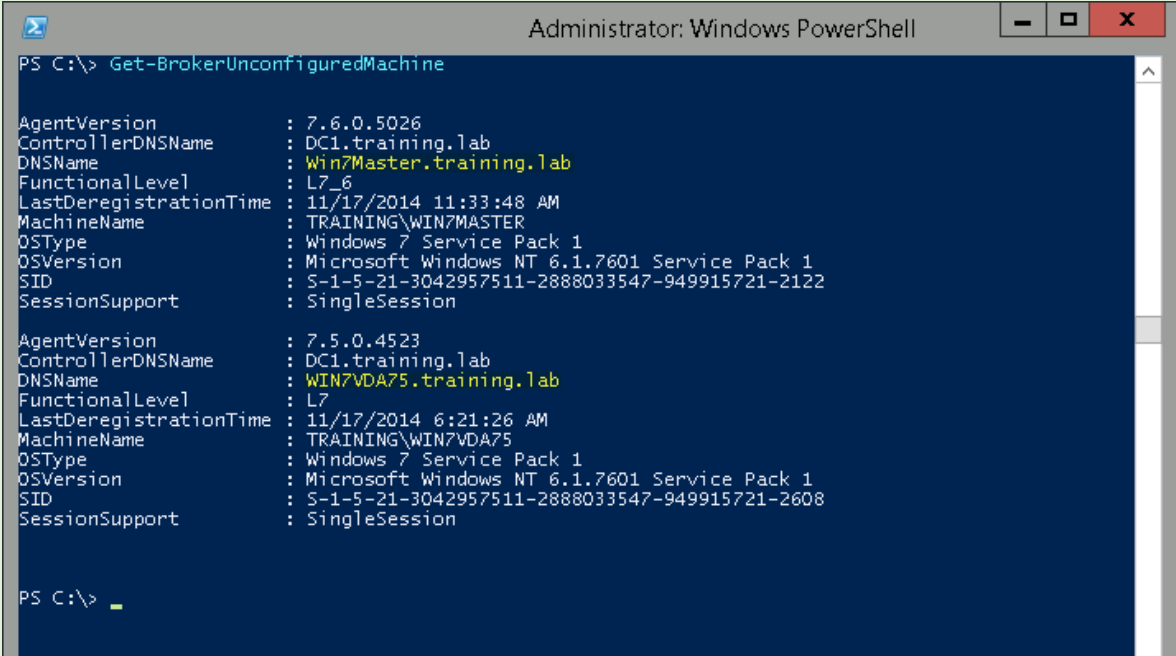
Go ahead and click on the controller's node to confirm...



Note: Fully functioning Controllers should always show a last update state of **0** minutes as they heartbeat by default with the Site DB every **20** seconds and will timeout after **40** seconds.

27. **Observation:** How many registered desktops are appearing against the **DC1** Controller? Haven't we only published 3 desktops as per the three Delivery Groups? If so, then why do you see **5** Registered desktops? See next step for explanation...



28.	<p>If you type <i>Get-BrokerUnconfiguredMachine</i> and hit return from within the already running instance of PoSH on DC2 you will find the answer.</p> <p>TIP: The answer is actually in the name of the cmdlet itself...</p>  <p>Note: The above cmdlet returns a list of any un-configured machines with the VDA installed within the site. Un-configured machines are machines which are considered to be Soft Registered only and not Hard registered.</p> <p>Soft Registered Machines: Machines with the VDA installed & communicating with at least one site Controller but have not been added to a Catalog and Delivery Group and therefore cannot be brokered for user connections.</p> <p>Hard registered Machines: Machines with the VDA installed & communicating with at least one site Controller but which are already members of a Catalog and Delivery Group and therefore are available to be brokered for user connections.</p>
29.	Close Citrix Studio.
30.	Congratulations, you have finished this lab exercise.

Exercise Summary

Takeaways from this exercise:

- To reduce the administration burden when carrying out complex repetitive tasks, simple PoSH scripts can and should be used.
- There are 4 key steps when manually adding a Controller to an existing Site DB which can essentially be broken as follows:
 1. Create Instance script for each FMA service on the controller to join the existing DB.
 2. Set DBConnection string for each of the FMA services on the controller to join the existing DB.
 3. Register all new FMA services on the controller to join the existing DB with the site configuration service.
 4. Reset all existing FMA service Group memberships so that they are aware of the new configuration service endpoint running on the new controller.
- For more cool info & tips on XenDesktop follow **@XDtipster** and **@XDInformer** on twitter.

Exercise 3

A Tricky Situation: Manually Updating FMA Service DB Schemas


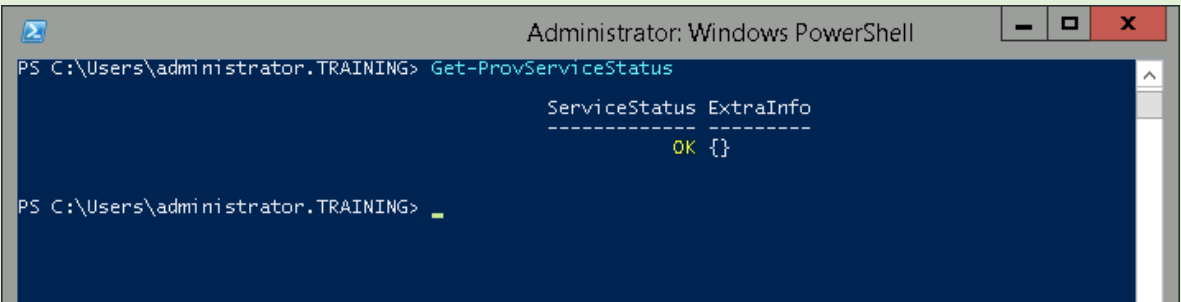
Overview

In this exercise we will learn how to check the true status of the key FMA services running on a XenDesktop site Controller and show how to manually change the DB Schema version through PoSH for troubleshooting purposes.

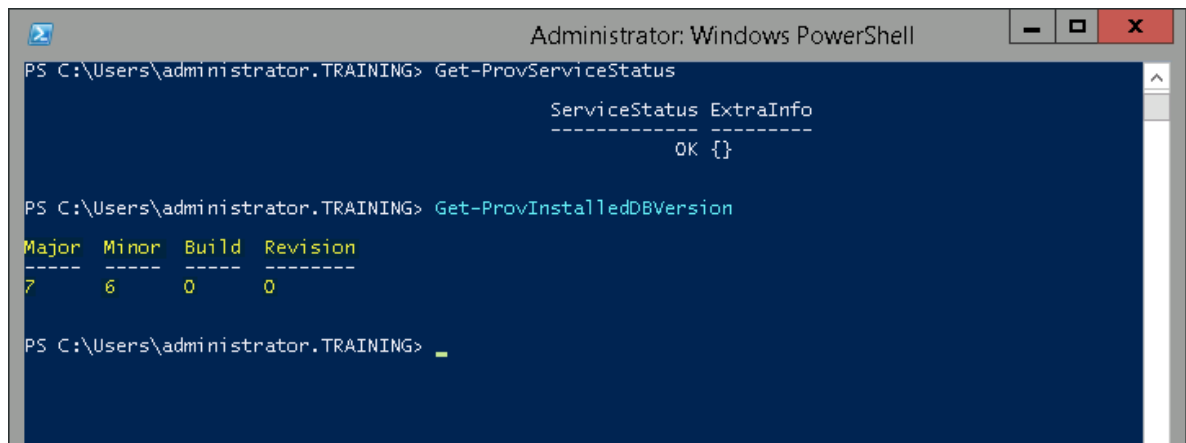
For the purpose of this exercise we will use the [Machine Creation Service](#) (Alias: **Prov**)

Step by step guidance

Estimated time to complete this lab: 30 minutes.

Step	Action
1.	<p>While logged into DC2 as training\administrator with password Citrix123.</p> <p>If not already running, launch a PowerShell window by clicking on the PoSH icon on the taskbar:</p> 
2.	<p>Type Get-ProvServiceStatus and hit return to check the status of the Broker service running on DC2:</p> <p>Expected Status = OK</p> 

3. Type ***Get-ProvInstalledDBVersion*** and hit return:



```
Administrator: Windows PowerShell
PS C:\Users\administrator.TRAINING> Get-ProvServiceStatus

ServiceStatus ExtraInfo
-----
OK {}

PS C:\Users\administrator.TRAINING> Get-ProvInstalledDBVersion

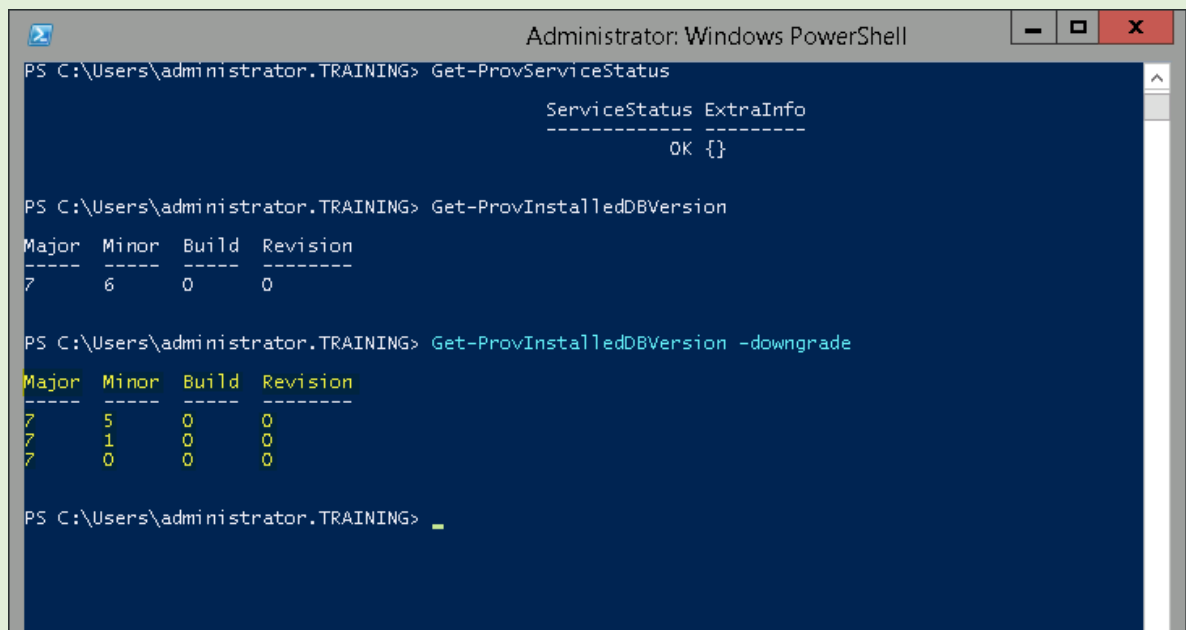
Major Minor Build Revision
-----
7      6      0      0

PS C:\Users\administrator.TRAINING> _
```

Note: The expected DBVersion for the Machine Creation service should appear as **7.6.0.0** indicating that the installed XenDesktop version is; you guessed it... **XenDesktop 7.6**.

4. Type ***Get-ProvInstalledDBVersion -downgrade*** and hit return:

TIP: Hit the up arrow on your keyboard to retrieve the last run command and just append ***-downgrade*** to the end.



```
Administrator: Windows PowerShell
PS C:\Users\administrator.TRAINING> Get-ProvServiceStatus

ServiceStatus ExtraInfo
-----
OK {}

PS C:\Users\administrator.TRAINING> Get-ProvInstalledDBVersion

Major Minor Build Revision
-----
7      6      0      0

PS C:\Users\administrator.TRAINING> Get-ProvInstalledDBVersion -downgrade

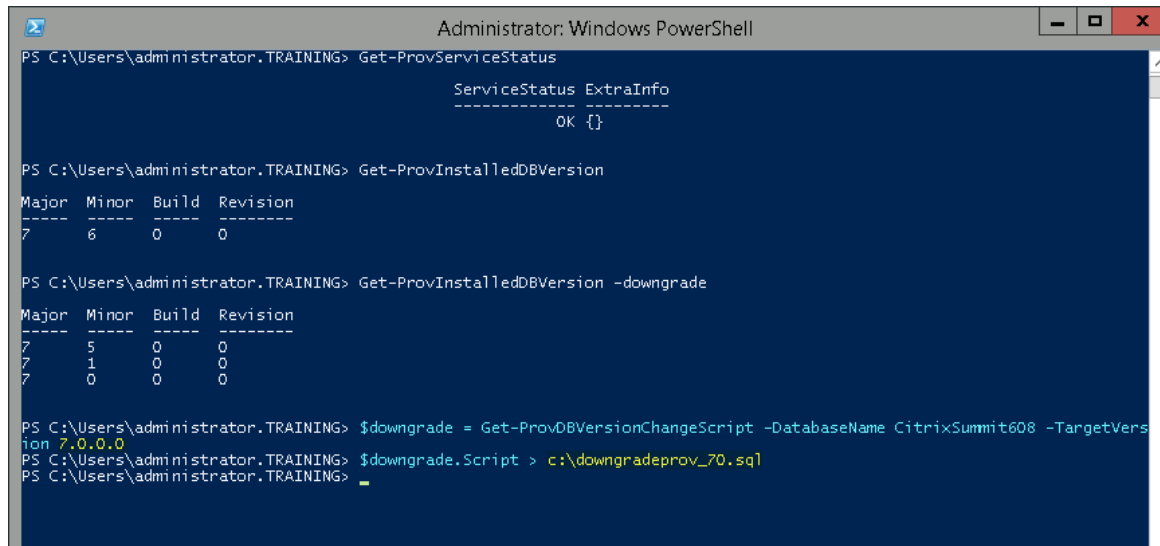
Major Minor Build Revision
-----
7      5      0      0
7      1      0      0
7      0      0      0

PS C:\Users\administrator.TRAINING> _
```

5. Paste the following two lines in order into the PoSH window and hit return after each, to create a script to downgrade the DB Schema for the Machine Creation service.

`$downgrade = Get-ProvDBVersionChangeScript -DatabaseName CitrixSummit608 -TargetVersion 7.0.0.0`

`$downgrade.Script > c:\downgradeprov_70.sql`



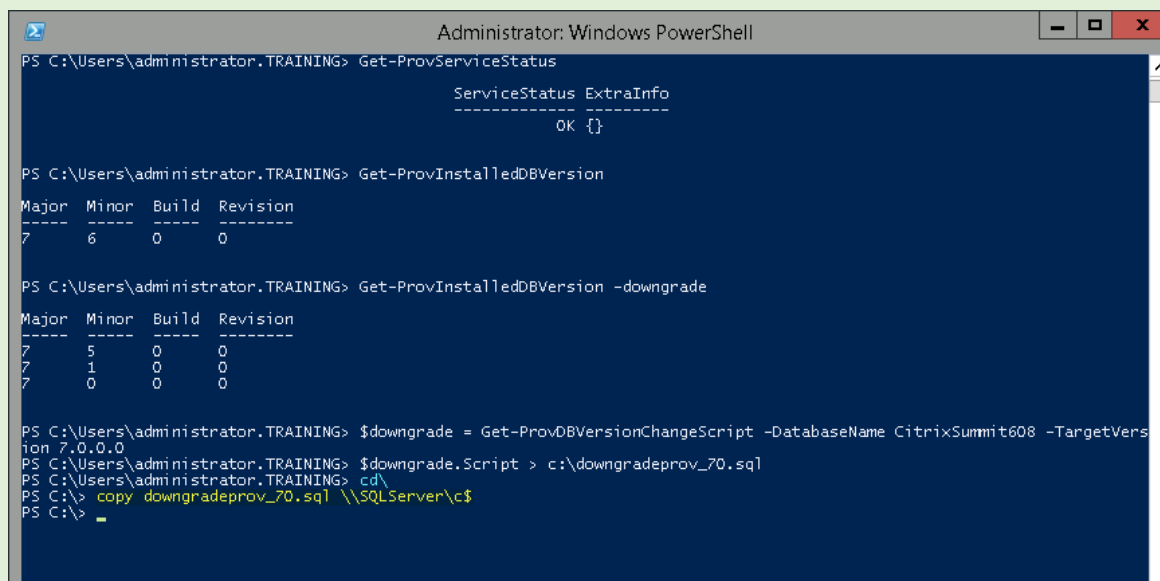
```
Administrator: Windows PowerShell
PS C:\Users\administrator.TRAINING> Get-ProvServiceStatus
ServiceStatus ExtraInfo
-----
OK {}

PS C:\Users\administrator.TRAINING> Get-ProvInstalledDBVersion
Major Minor Build Revision
-----
7      6      0      0

PS C:\Users\administrator.TRAINING> Get-ProvInstalledDBVersion -downgrade
Major Minor Build Revision
-----
7      5      0      0
7      1      0      0
7      0      0      0

PS C:\Users\administrator.TRAINING> $downgrade = Get-ProvDBVersionChangeScript -DatabaseName CitrixSummit608 -TargetVersion 7.0.0.0
PS C:\Users\administrator.TRAINING> $downgrade.Script > c:\downgradeprov_70.sql
PS C:\Users\administrator.TRAINING>
```

6. Change Directory to **`c:\`** and Type **`copy downgradeprov_70.sql \\SQLServer\c$`** and hit return to copy the Schema downgrade script to the root of the **SQLServer VM**:



```
Administrator: Windows PowerShell
PS C:\Users\administrator.TRAINING> Get-ProvServiceStatus
ServiceStatus ExtraInfo
-----
OK {}

PS C:\Users\administrator.TRAINING> Get-ProvInstalledDBVersion
Major Minor Build Revision
-----
7      6      0      0

PS C:\Users\administrator.TRAINING> Get-ProvInstalledDBVersion -downgrade
Major Minor Build Revision
-----
7      5      0      0
7      1      0      0
7      0      0      0

PS C:\Users\administrator.TRAINING> $downgrade = Get-ProvDBVersionChangeScript -DatabaseName CitrixSummit608 -TargetVersion 7.0.0.0
PS C:\Users\administrator.TRAINING> $downgrade.Script > c:\downgradeprov_70.sql
PS C:\Users\administrator.TRAINING> cd\
PS C:\> copy downgradeprov_70.sql \\SQLServer\c$
PS C:\>
```

7. **IMPORTANT: DO NOT BY PASS THIS STEP:**

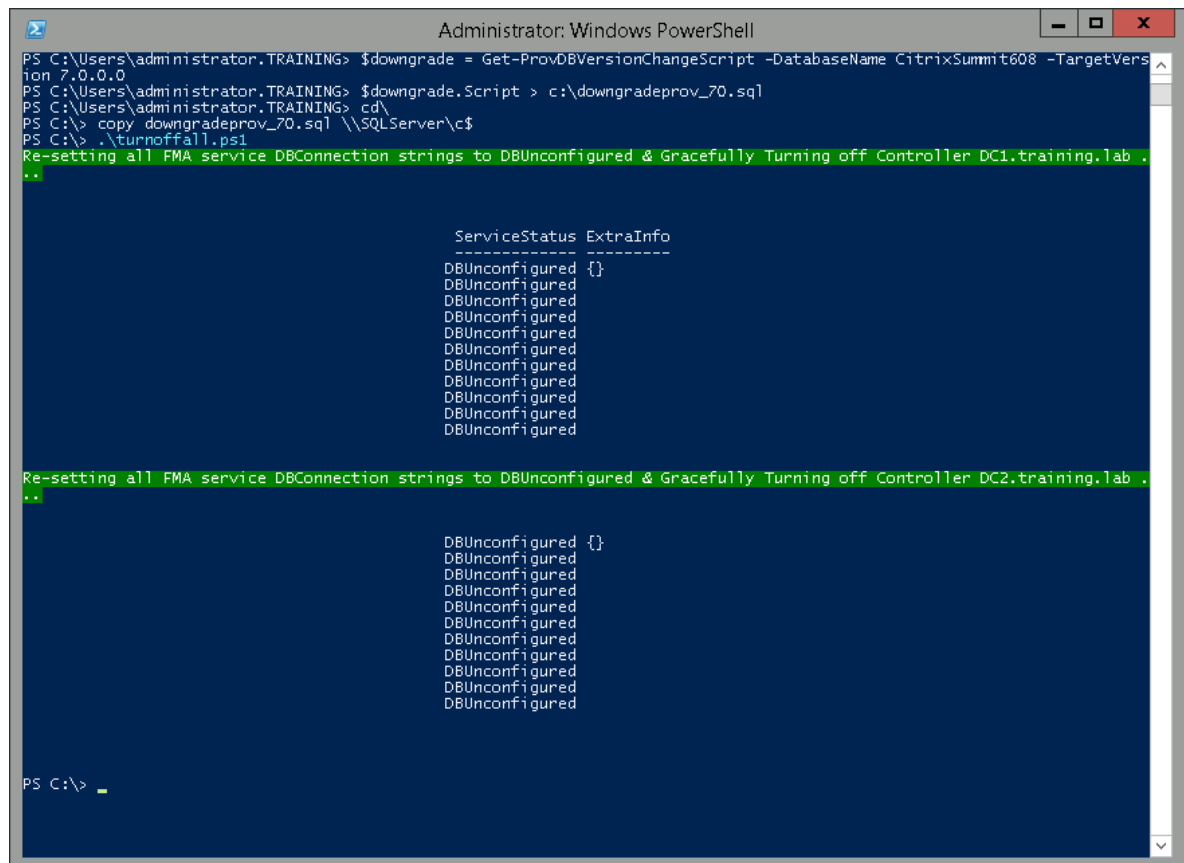
Before we can downgrade the existing Machine Creation service DB Schema we need to disconnect all controllers from the DB.

To do this just run the following simple script from the root of PoSH on **DC2**: (To get to the root just type **cd** & hit return)

C:\> .\turnoffall.ps1

Reminder: You should be running this script from DC2

Note: Setting all the service instance DBConnection strings to **DBUnconfigured** will turn off each controller gracefully and release all assigned Site Services (Specific house-keeping roles leased dynamically and evenly across all site Controllers at run-time).



```
Administrator: Windows PowerShell
PS C:\Users\administrator.TRAINING> $downgrade = Get-ProvDBVersionChangeScript -DatabaseName CitrixSummit608 -TargetVersion 7.0.0.0
PS C:\Users\administrator.TRAINING> $downgrade.Script > c:\downgradeprov_70.sql
PS C:\Users\administrator.TRAINING> cd\
PS C:\> copy downgradeprov_70.sql \\SQLServer\c$
PS C:\> .\turnoffall.ps1
Re-setting all FMA service DBConnection strings to DBUnconfigured & Gracefully Turning off Controller DC1.training.lab .
..

ServiceStatus ExtraInfo
-----
DBUnconfigured {}
DBUnconfigured
DBUnconfigured
DBUnconfigured
DBUnconfigured
DBUnconfigured
DBUnconfigured
DBUnconfigured
DBUnconfigured
DBUnconfigured
DBUnconfigured

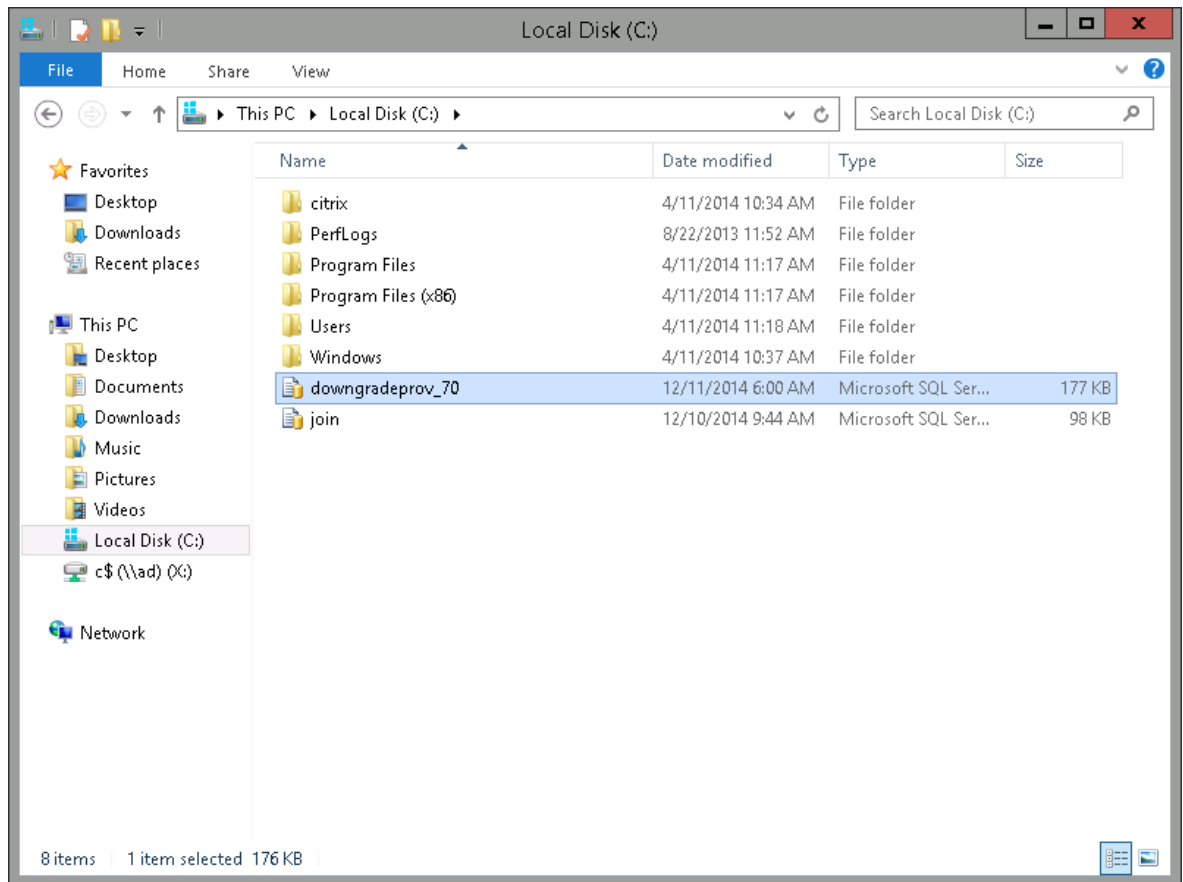
Re-setting all FMA service DBConnection strings to DBUnconfigured & Gracefully Turning off Controller DC2.training.lab .
..

DBUnconfigured {}
DBUnconfigured
DBUnconfigured
DBUnconfigured
DBUnconfigured
DBUnconfigured
DBUnconfigured
DBUnconfigured
DBUnconfigured
DBUnconfigured
DBUnconfigured

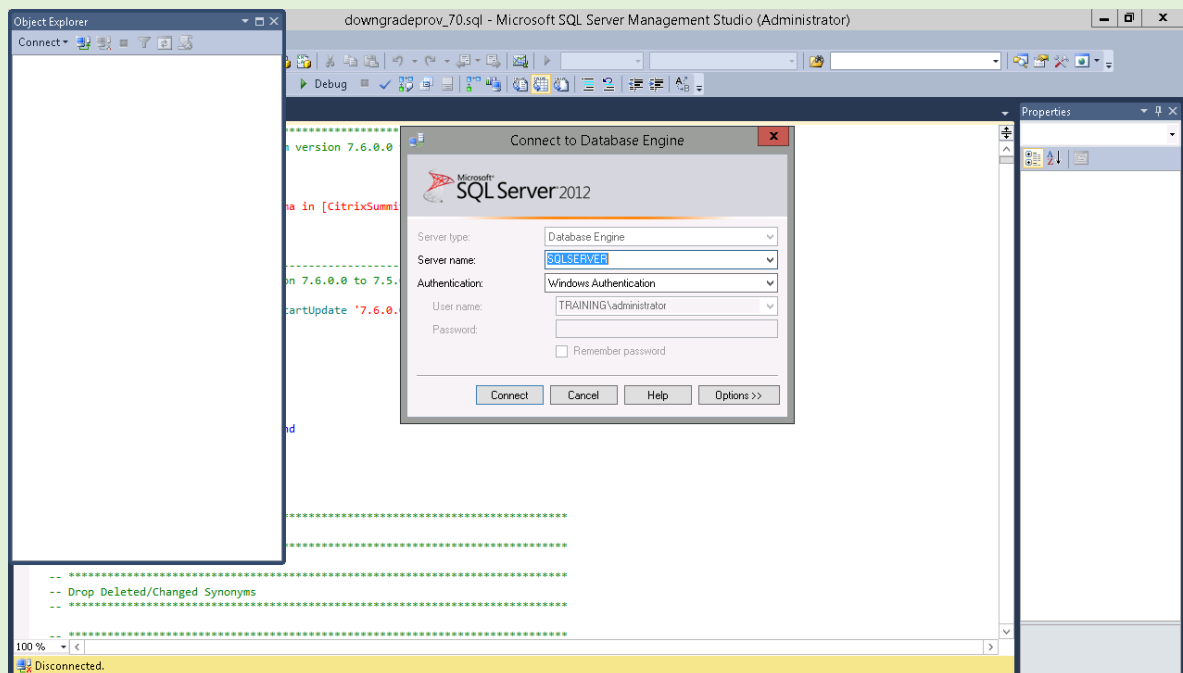
PS C:\> _
```

8. Login to **SQLServer** as **training\administrator** with password **Citrix123**

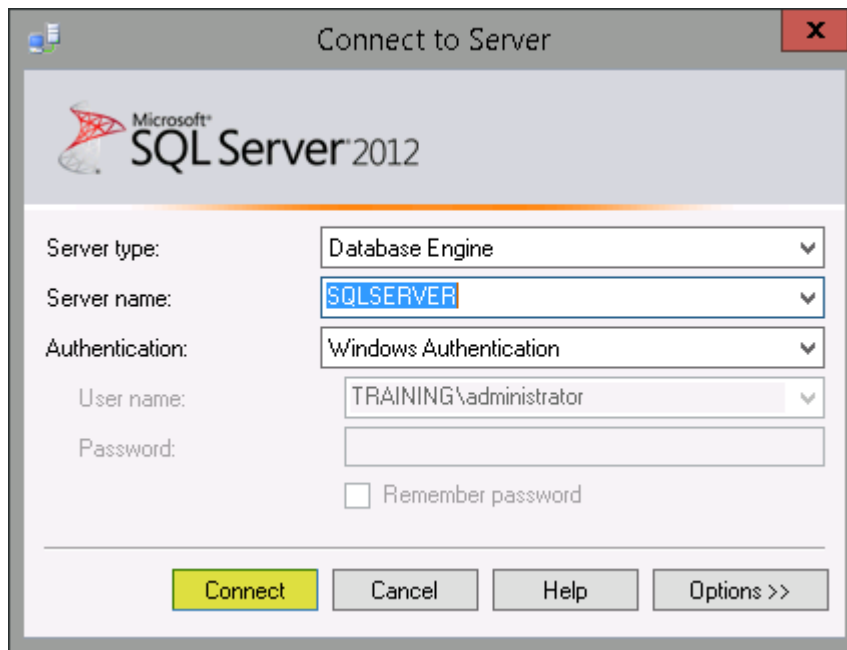
9. Browse the local disk and double click on the **downgradeprov_70.sql** file:



10. Click **Connect** on the Database engine authentication dialog box:

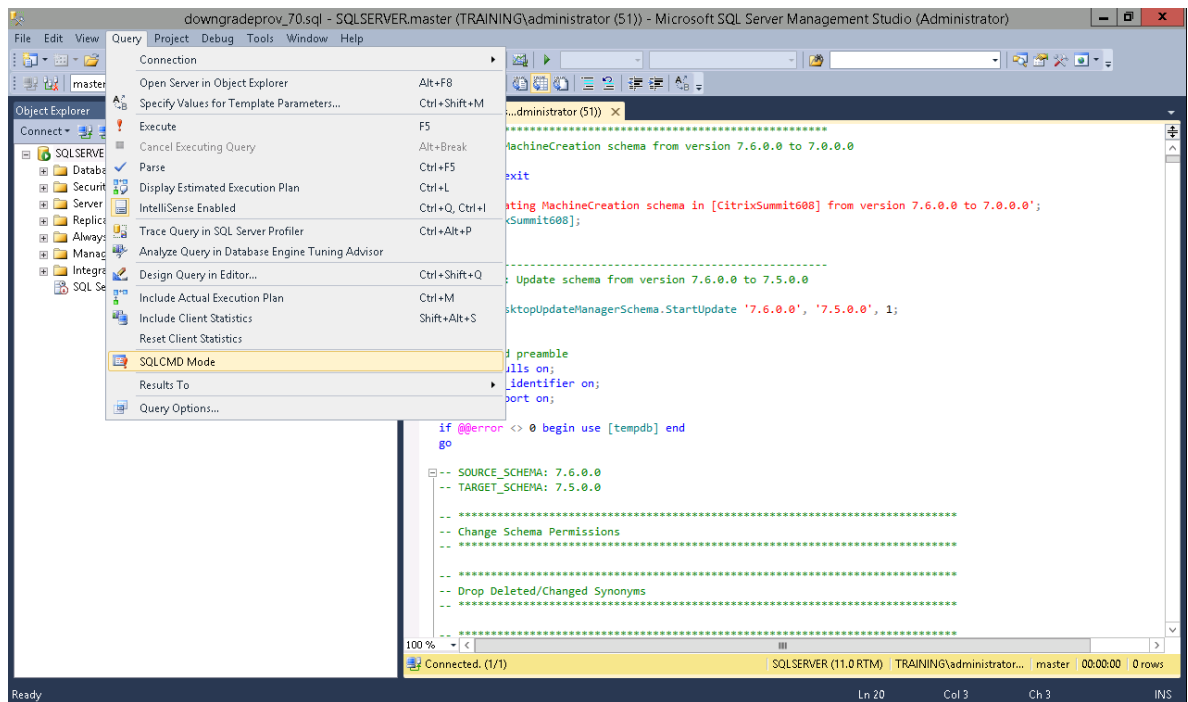


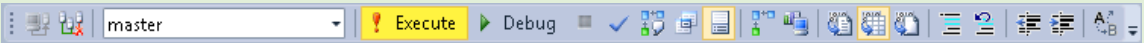
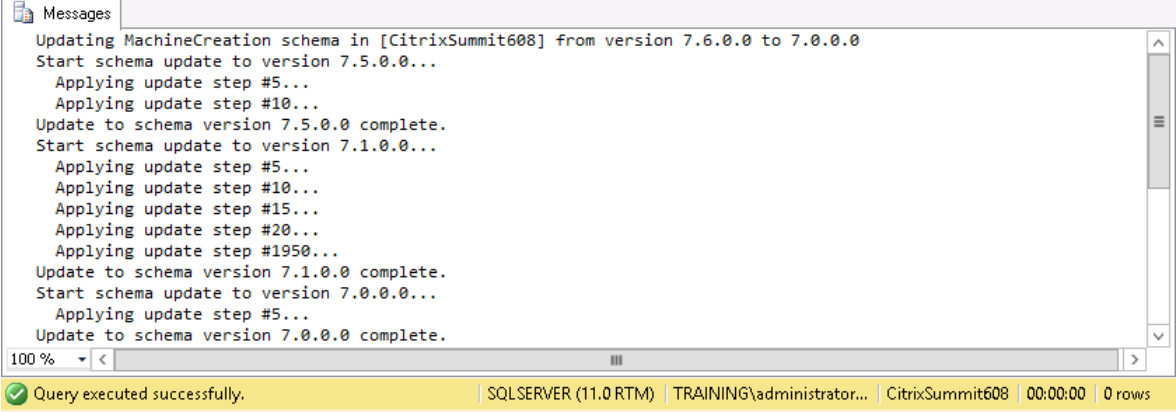

11. Click **Connect** again to authenticate to the Server:



12. Click inside the **downgradeprov_70.sql** script window.

13. Click on **Query** and select **SQLCMD Mode**:

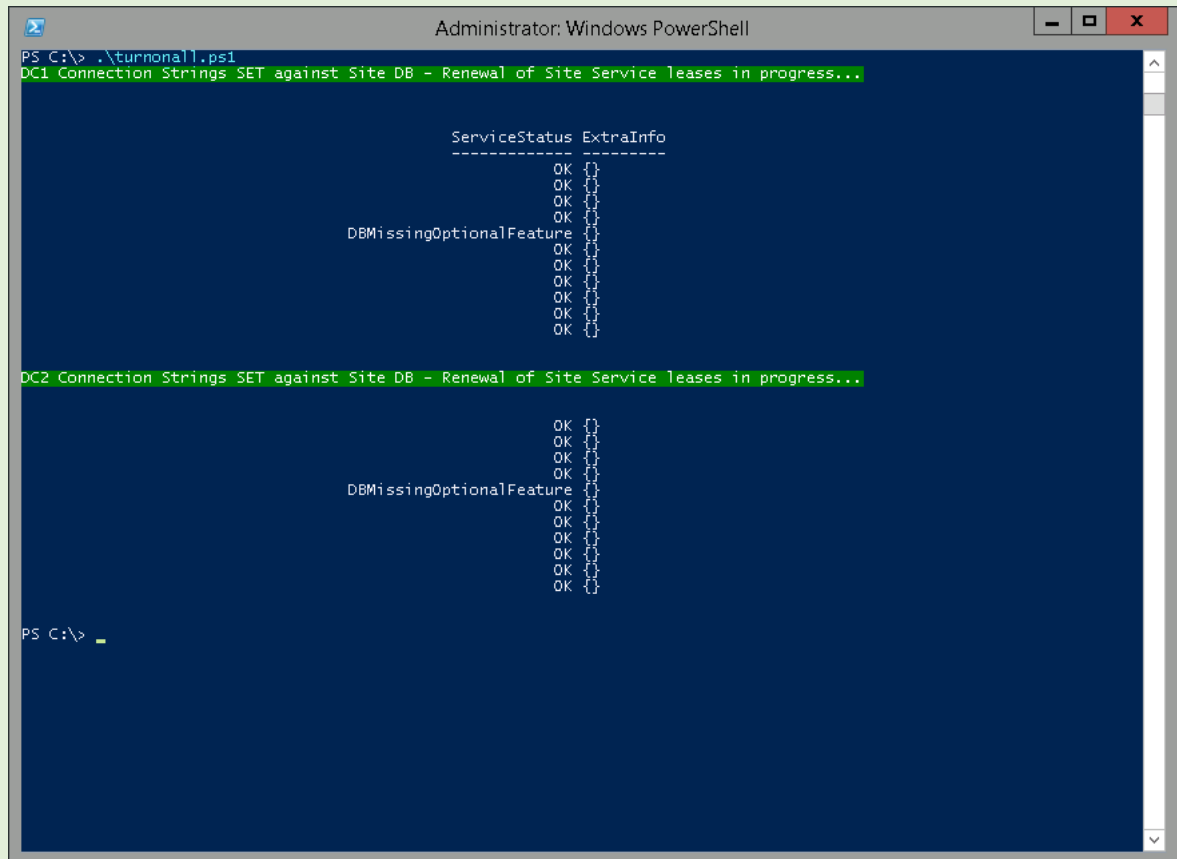


14.	<p>IMPORTANT: HAVE YOU COMPLETED STEP 7 ABOVE i.e. ran <code>turnoffall.ps1</code> to un-configure all site controller FMA services? It is essential that this is completed or you will receive an error when executing <code>downgradeprov_70.sql</code>.</p> <p>Press F5 or the Execute Icon on the Toolbar to execute the downgradeprov_70.sql script</p> <p>Note: There is no requirement to select the CitrixSummit608 DB to run the script against as the script itself determines this.</p> 
15.	<p>Observation: Script runs without errors and updates (i.e. Downgrades) the Machine Creation service Schema from 7.6.0.0 to 7.0.0.0.</p> 
16.	<p>Switch back to DC2 (Controller where you created the DB Schema downgrade script) and login again if needed as training\administrator with password Citrix123</p>
17.	<p>If needed launch a new PoSH window from the taskbar icon.</p> 

18. Before we can check the status of the Machine Creation service after the DB Schema has been downgraded we must reconnect all site Controllers to the **CitrixSummit608** DB
- To do this just run the following simple script from the root of PoSH: (To get to the root just type **cd** & hit return)

C:\> .\turnonall.ps1

Reminder: You should be running this script from DC2



```
Administrator: Windows PowerShell
PS C:\> .\turnonall.ps1
DC1 Connection Strings SET against Site DB - Renewal of Site Service leases in progress...

ServiceStatus ExtraInfo
-----
OK {}
OK {}
OK {}
OK {}
DBMissingOptionalFeature
OK {}
OK {}
OK {}
OK {}
OK {}
OK {}
OK {}

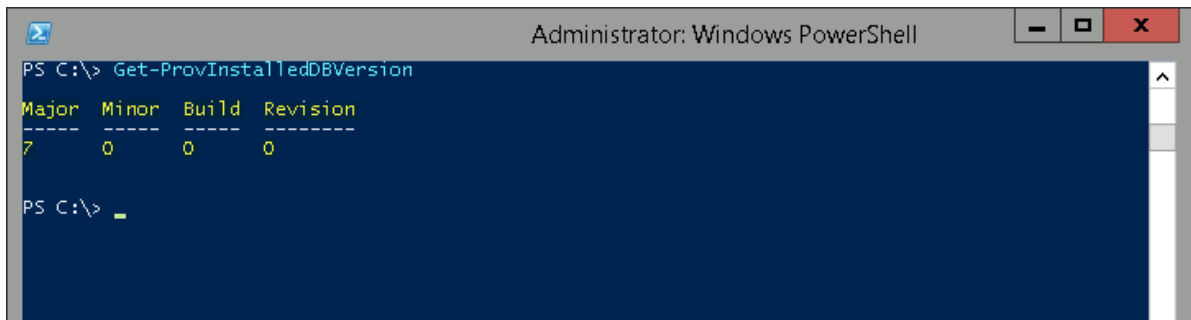
DC2 Connection Strings SET against Site DB - Renewal of Site Service leases in progress...

OK {}
OK {}
OK {}
OK {}
DBMissingOptionalFeature
OK {}
OK {}
OK {}
OK {}
OK {}
OK {}
OK {}

PS C:\> _
```

Note: Script indicates that the state of at least one FMA service running on each Controller is not returning a value of OK but in fact **DBMissingOptionalFeature**. Let's investigate...

19. Type ***Get-ProvInstalledDBVersion*** and hit return:



```
Administrator: Windows PowerShell
PS C:\> Get-ProvInstalledDBVersion

Major Minor Build Revision
-----
7      0      0      0

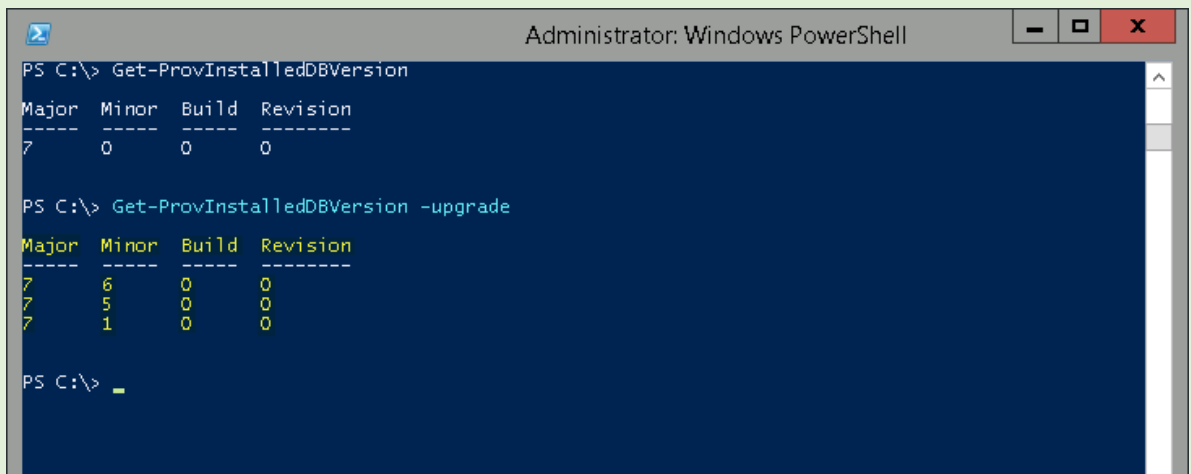
PS C:\> _
```

Note: After downgrading the Machine Creation service DB Schema the above cmdlet reports the installed DB version as 7.0.0.0 and not 7.6.0.0. As you may have already guessed. This is the reason for the above script to report a value of **DBMissingOptionalFeature** for a service instance on each controller. Based on the steps above and what we have seen so far, we can say with authority now that the affected service is the Machine Creation Service.

TIP: You can confirm this by simply running ***Get-ProvServiceStatus*** from within the same PoSH window.

Note: The **DBMissingOptionalFeature** return value in fact, indicates that the Machine Creation Service is connected to a database that is valid, but one that does not have the full functionality required for optimal performance. Upgrading the database is advisable. (**See Next Step...**)

20. Type ***Get-ProvInstalledDBVersion -upgrade*** to view the available upgrade options:



```
Administrator: Windows PowerShell
PS C:\> Get-ProvInstalledDBVersion

Major Minor Build Revision
-----
7      0      0      0

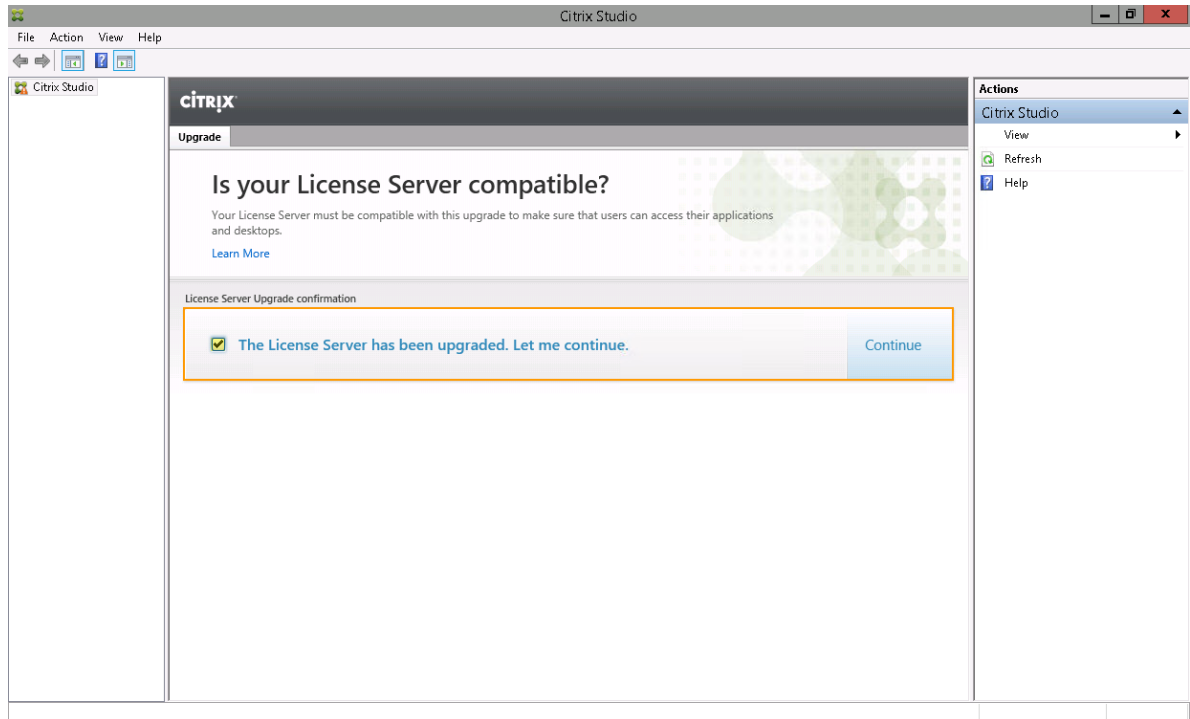
PS C:\> Get-ProvInstalledDBVersion -upgrade

Major Minor Build Revision
-----
7      6      0      0
7      5      0      0
7      1      0      0

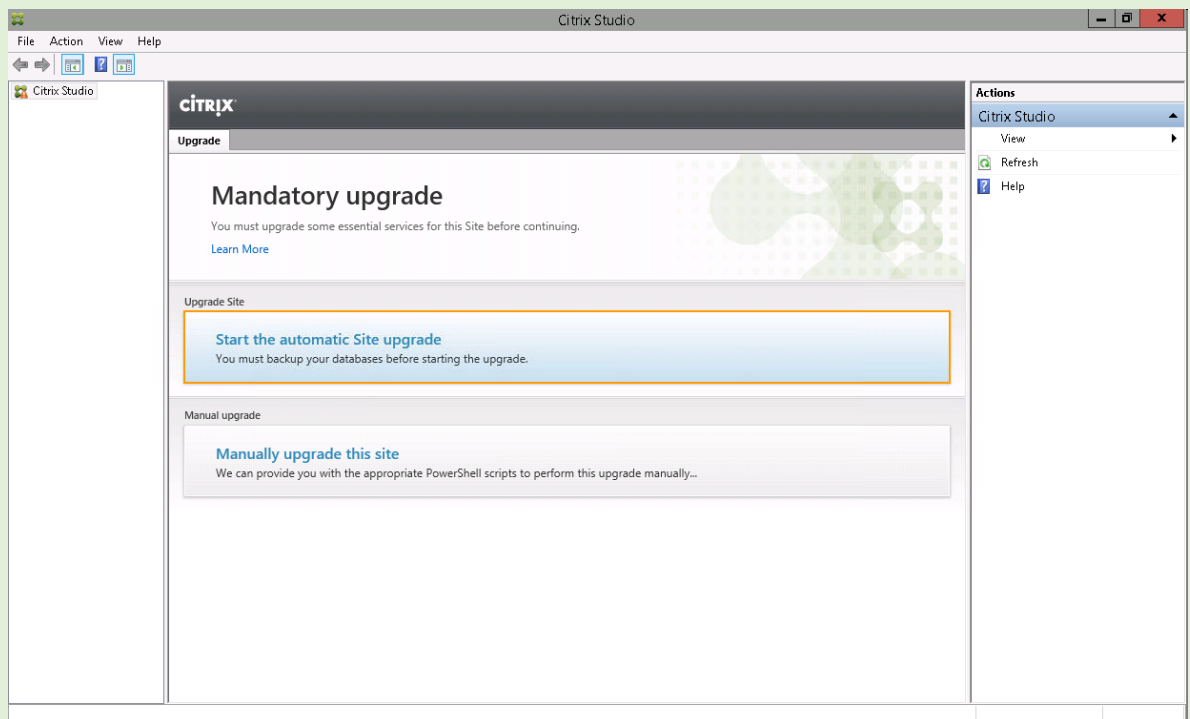
PS C:\> _
```

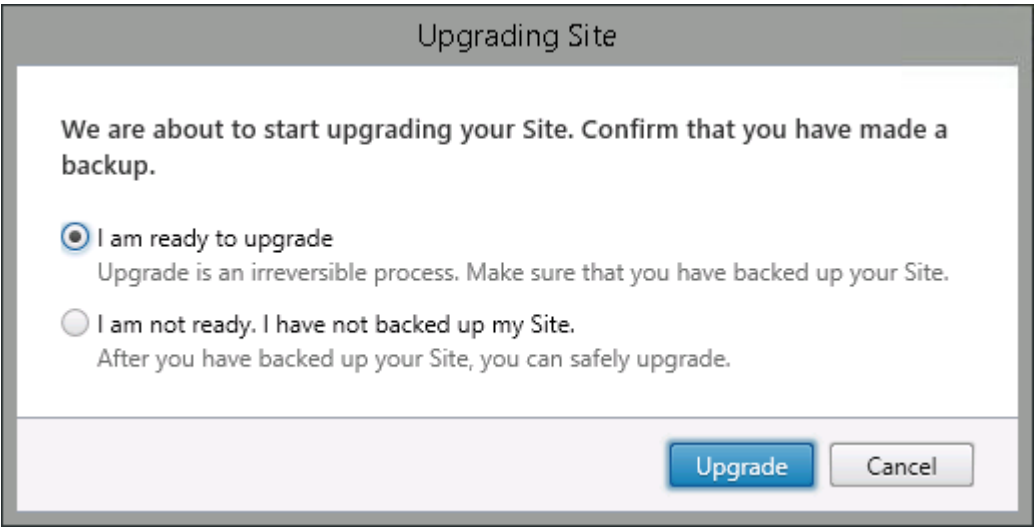
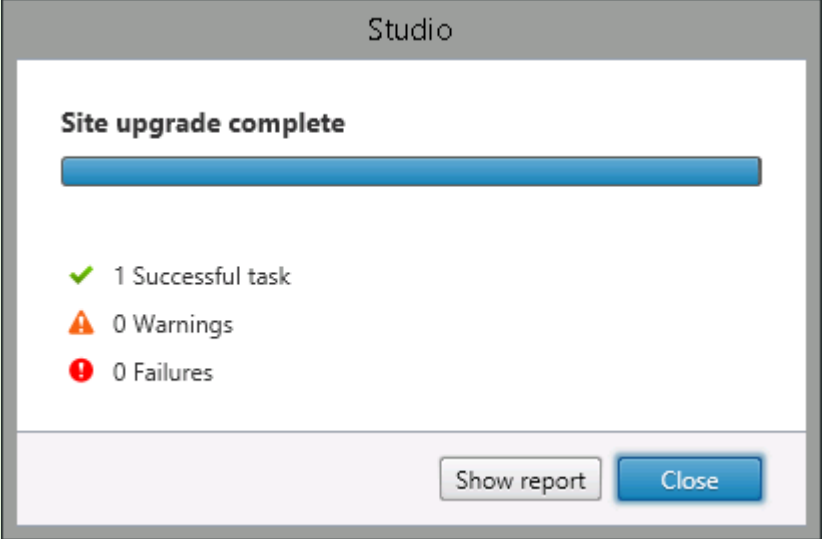
Note: At this point, you could (**But DON'T**) repeat Step 5-14 with a few little tweaks to manually upgrade the Machine Creation service Schema to target version 7.6.0.0. Let's take the easy option this time however...

21. Launch **Citrix Studio** and dismiss the license server message by placing a tick in the box and clicking **Continue**:

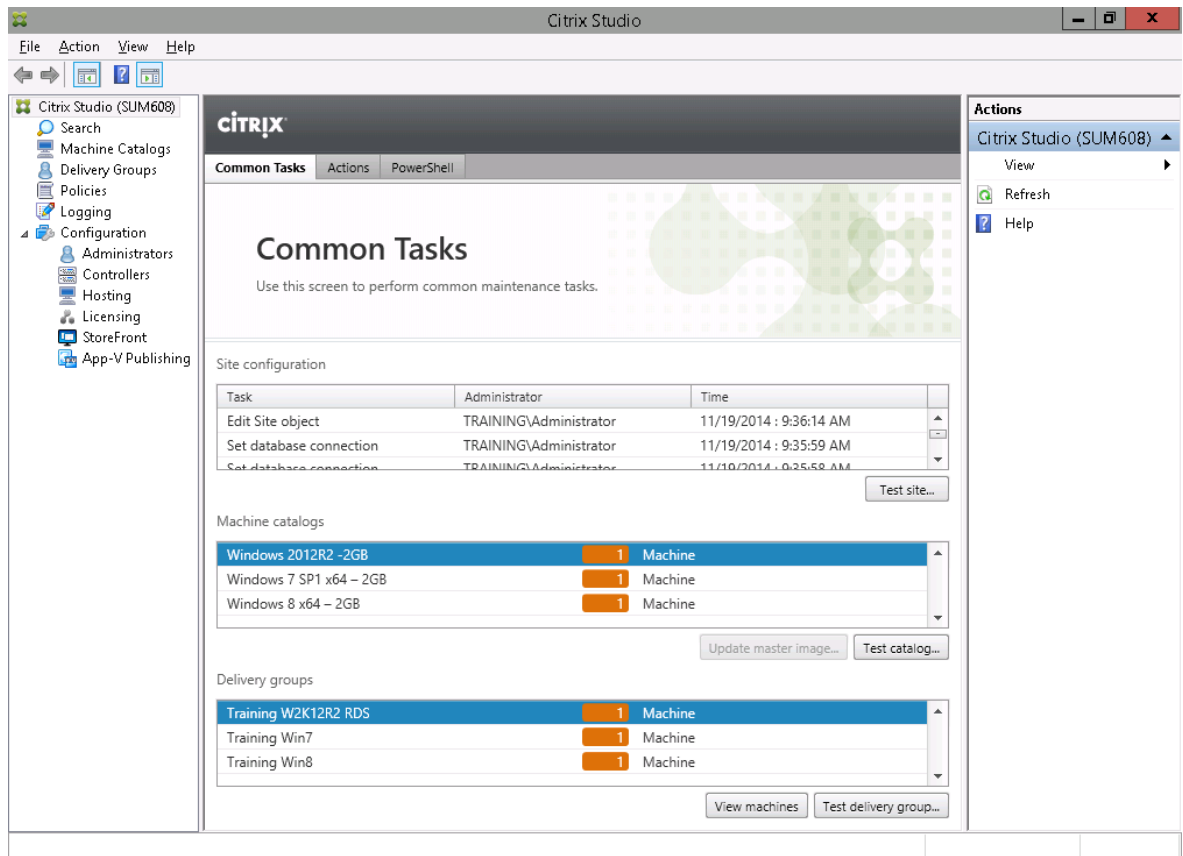


22. Click on the option to “**Start the automatic Site upgrade**”:

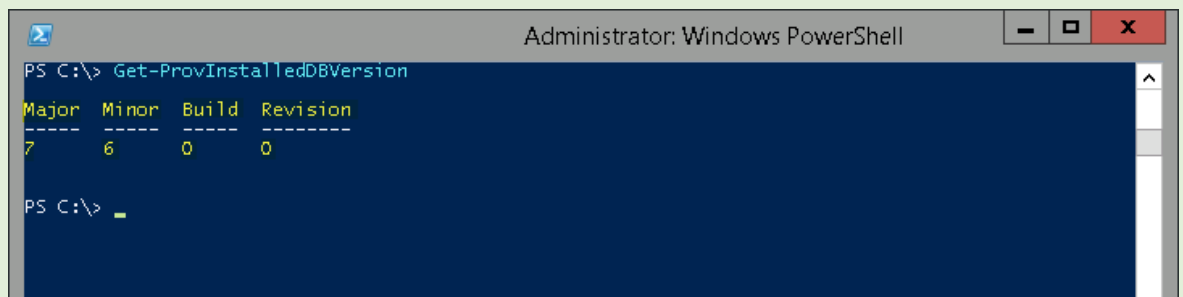


23.	<p>Select the radio button “I am ready to upgrade” and click Upgrade:</p> 
24.	<p>Click Close once the Site upgrade process completes:</p> 

25. **Observation:** Citrix Studio dashboard appears correctly after applying the mandatory DB Schema update:



26. From within the already active or a new PoSH window on **DC2**, Type ***Get-ProvInstalledDBVersion*** and hit return:



Note: After carrying out the mandatory upgrade of the Machine Creation Service DB Schema through Citrix Studio, the above cmdlet reports the installed DB version correctly as 7.6.0.0

27. Congratulations you have finished this exercise.

Exercise Summary

Takeaways from this exercise:

- FMA services should be running against the latest DB Schema versions to avoid possible inconsistent behavior.
- Citrix Studio will inform you if a mandatory DB upgrade is required and will also complete the task if instructed through helpful wizards
 - PoSH can be used to check for upgrade and downgrade Schema target versions and also generate an SQL script to apply directly against the site DB. This is an alternative method which can be used if Citrix Studio encounters issues.
- It is important to look beyond the windows services applet and check the true state of the FMA services using the PoSH SDK
 - E.G. `Get-<ServiceAlias>ServiceStatus`
- For more cool info & tips on XenDesktop follow **@XDtipster** and **@XDInformer** on twitter.

Exercise 4



A Clever Idea: Controlling Access to Desktops through the SDK

Overview

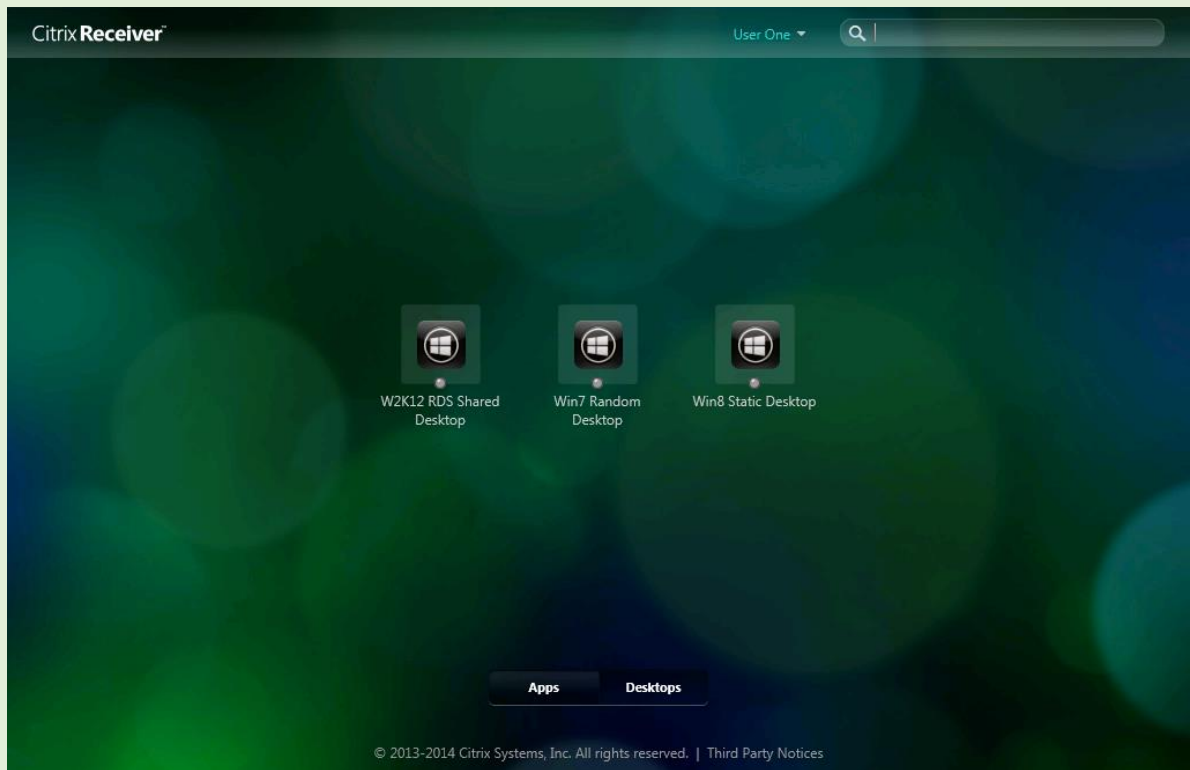
In this exercise we will use the PoSH SDK to explicitly control access to Desktops.

Step by step guidance

Estimated time to complete this lab: 25 minutes.

Step	Action
1.	Login to Win7Client as trainingadministrator with password Citrix123 .
2.	Launch Internet Explorer from the taskbar icon and enter the Receiver for Web URL in the address bar: http://sf.training.lab/Citrix/SummitWeb/ 
3.	Log on to StoreFront using Traininguser1 and Citrix123 . 

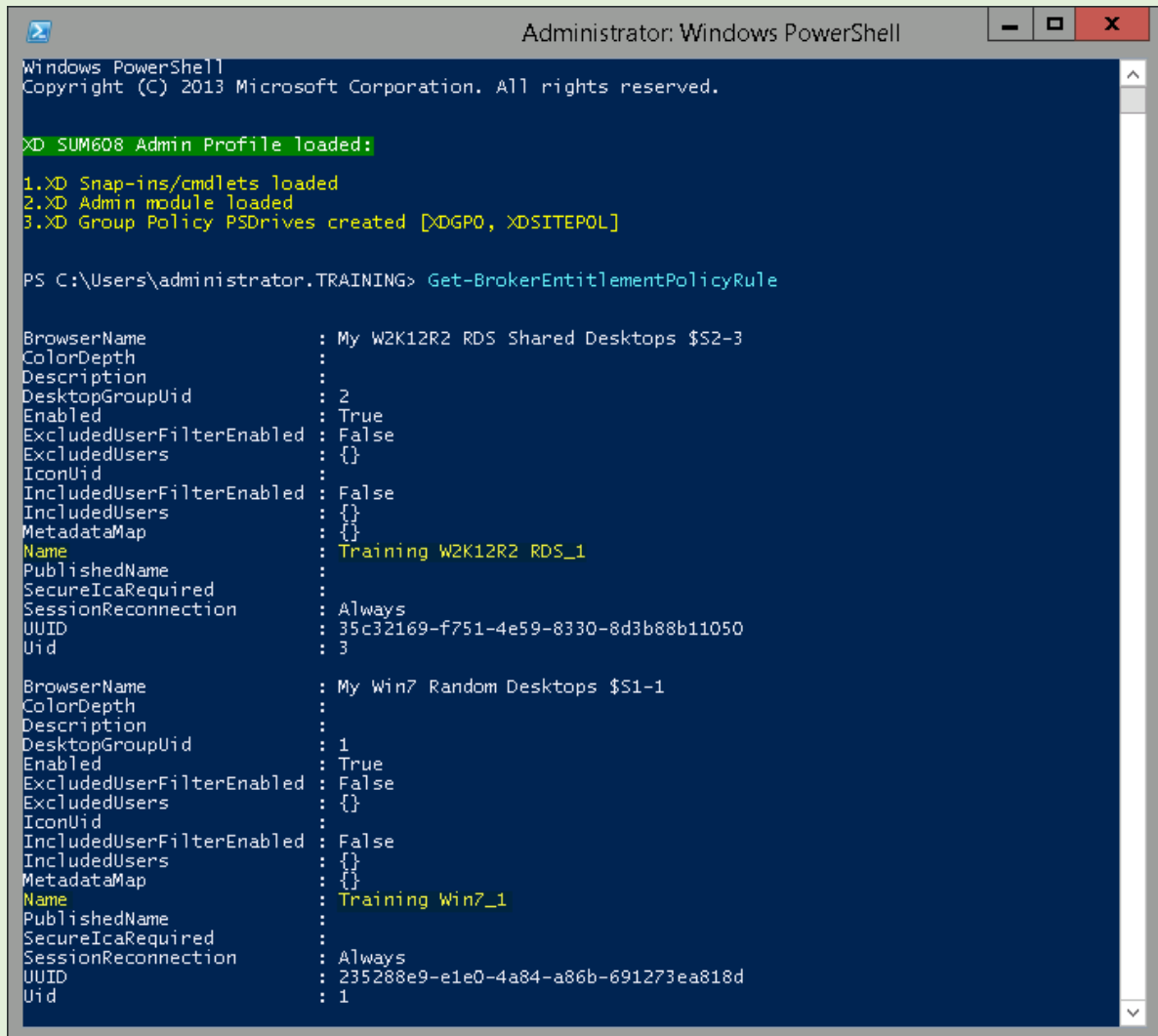
4. Based on the three existing Delivery Group user assignments, the following desktops types are available to **User1**:
1. W2K12 RDS Shared Desktop
 2. Win7 Random Desktop
 3. Win8 Static Desktop



5. Switch back to and login again if needed to **DC2** as **training\administrator** with password **Citrix123** and launch a PowerShell window by clicking on the **PoSH** icon on the taskbar:



6. Type ***Get-BrokerEntitlementPolicyRule*** and hit return:



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2013 Microsoft Corporation. All rights reserved.

XD SUM608 Admin Profile loaded:
1.XD Snap-ins/cmdlets loaded
2.XD Admin module loaded
3.XD Group Policy PSDrives created [XDGP0, XDITEPOL]

PS C:\Users\administrator.TRAINING> Get-BrokerEntitlementPolicyRule

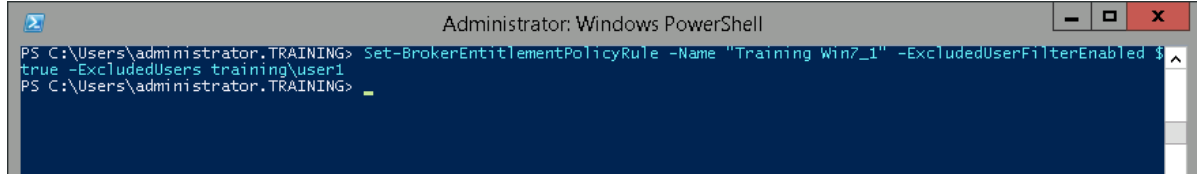
BrowserName      : My W2K12R2 RDS Shared Desktops $S2-3
ColorDepth       :
Description      :
DesktopGroupUid  : 2
Enabled          : True
ExcludedUserFilterEnabled : False
ExcludedUsers    : {}
IconUid          :
IncludedUserFilterEnabled : False
IncludedUsers    : {}
MetadataMap      : {}
Name             : Training W2K12R2 RDS_1
PublishedName    :
SecureIcaRequired :
SessionReconnection : Always
UUID             : 35c32169-f751-4e59-8330-8d3b88b11050
Uid              : 3

BrowserName      : My Win7 Random Desktops $S1-1
ColorDepth       :
Description      :
DesktopGroupUid  : 1
Enabled          : True
ExcludedUserFilterEnabled : False
ExcludedUsers    : {}
IconUid          :
IncludedUserFilterEnabled : False
IncludedUsers    : {}
MetadataMap      : {}
Name             : Training Win7_1
PublishedName    :
SecureIcaRequired :
SessionReconnection : Always
UUID             : 235288e9-e1e0-4a84-a86b-691273ea818d
Uid              : 1
```

Note: The Site Entitlement Policy contains a single rule by default for each Existing Random or Pooled Random (MCS) Delivery Group and can be used to explicitly control access to member desktops.

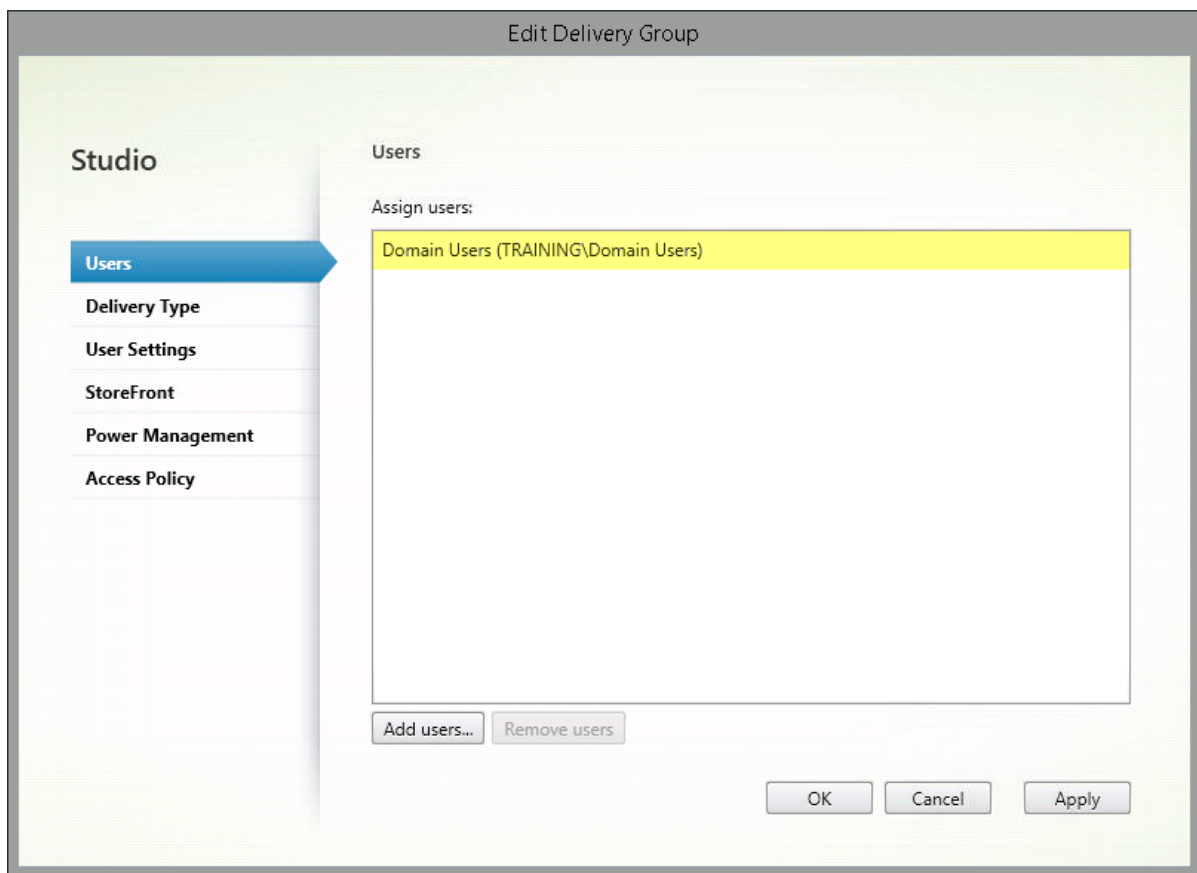
7. **Copy and paste** the following string into the **PoSH** window and hit return:

***Set-BrokerEntitlementPolicyRule -Name "Training Win7_1" -
ExcludedUserFilterEnabled \$true -ExcludedUsers training\user1***



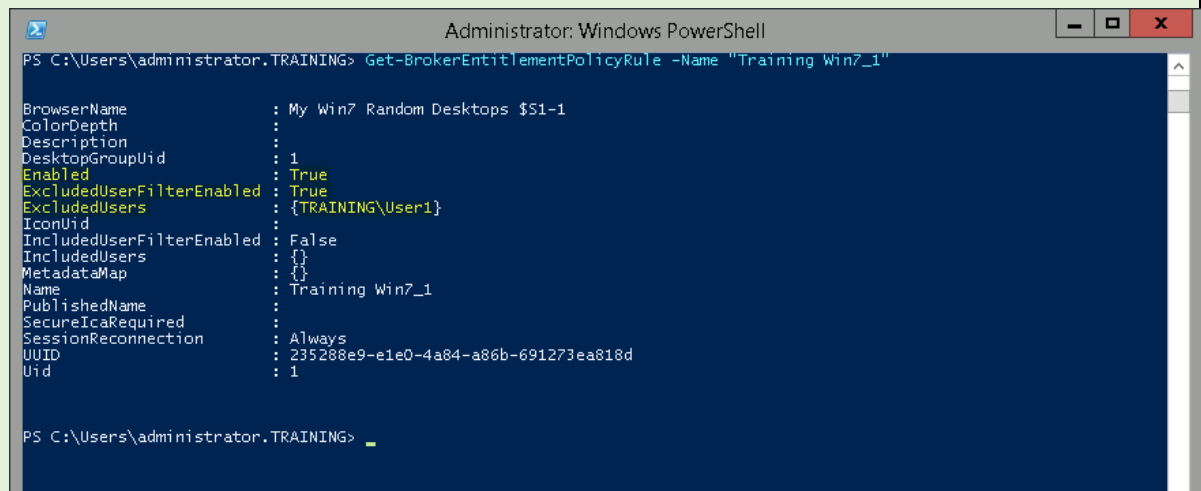
```
Administrator: Windows PowerShell
PS C:\Users\administrator.TRAINING> Set-BrokerEntitlementPolicyRule -Name "Training Win7_1" -ExcludedUserFilterEnabled $true -ExcludedUsers training\user1
PS C:\Users\administrator.TRAINING>
```

Note: The above string will enable the excluded users filter and exclude **training\user1** from accessing any desktops within the **Training Win7** Delivery Group even though user 1 is part of the Domain Users Group which in turn has been granted access through the UI (Citrix Studio) Entitlement & Access Policies will always take precedence over user associations through the UI.



8. Copy and paste the following string into **PoSH** and hit return:

Get-BrokerEntitlementPolicyRule -Name "Training Win7_1"



The screenshot shows a Windows PowerShell console window titled "Administrator: Windows PowerShell". The command prompt is "PS C:\Users\administrator.TRAINING> Get-BrokerEntitlementPolicyRule -Name "Training Win7_1"". The output is a list of properties and their values for the "Training Win7_1" policy rule.

```
BrowserName      : My Win7 Random Desktops $S1-1
ColorDepth       :
Description      :
DesktopGroupUid  : 1
Enabled          : True
ExcludedUserFilterEnabled : True
ExcludedUsers    : {TRAINING\User1}
IconUid          :
IncludedUserFilterEnabled : False
IncludedUsers    : {}
MetadataMap      : {}
Name             : Training Win7_1
PublishedName    :
SecureIcaRequired :
SessionReconnection : Always
UUID             : 235288e9-e1e0-4a84-a86b-691273ea818d
Uid              : 1
```

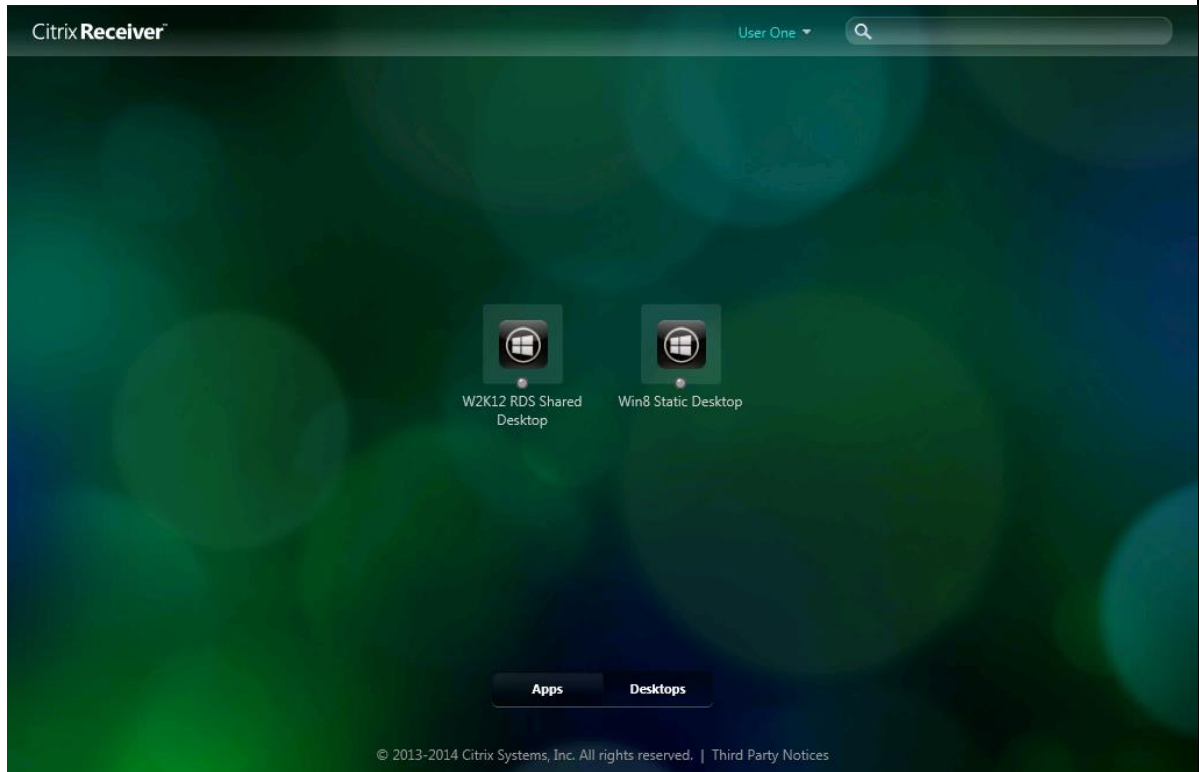
The prompt is "PS C:\Users\administrator.TRAINING> _".

Note: We can now confirm the exclusion just set in place.

9. Switch back to **Win7Client** and press **F5** or hit return to refresh StoreFront console. If your session has timed out, just log in again as **Training\user1** and **Citrix123**.

Based on the three existing Delivery Group user assignments and the explicit entitlement policy exclusion on the Training Win7 Delivery Group, only the following two desktops types are available to **User1**:

1. W2K12 RDS Shared Desktop
2. Win8 Static Desktop



Note: The Win7 Random Desktop is no longer available based on the entitlement policy exclusion which we just set in **Step 7** above.

10. Switch back to **DC2** and remove the exclusion by copying **one** of the following strings into the open **PoSH** window and hitting return:

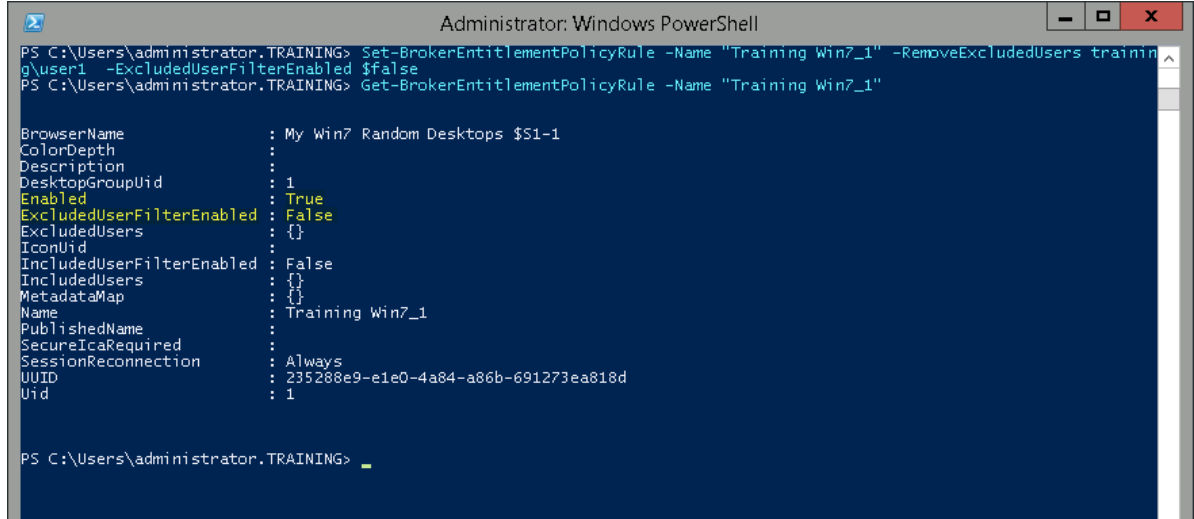
```
Set-BrokerEntitlementPolicyRule -Name "Training Win7_1" -RemoveExcludedUsers  
training\user1 -ExcludedUserFilterEnabled $false
```

Or

```
Set-BrokerEntitlementPolicyRule -Name "Training Win7_1" -  
ExcludedUserFilterEnabled $false -ExcludedUsers @()
```

11. **Copy and paste** the following string into the **PoSH** window and hit return to **confirm** that the exclusion has been removed:

Get-BrokerEntitlementPolicyRule -Name "Training Win7_1"



The screenshot shows a Windows PowerShell window titled "Administrator: Windows PowerShell". The command prompt shows the following commands and output:

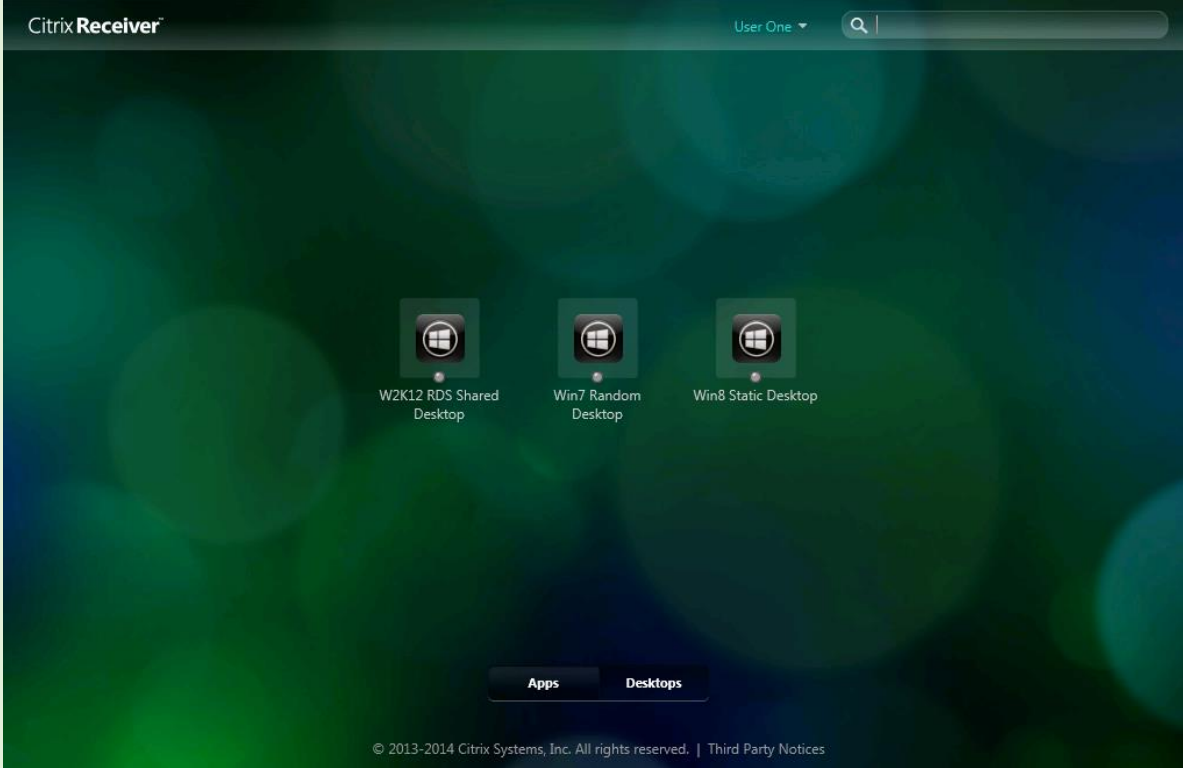
```
PS C:\Users\administrator.TRAINING> Set-BrokerEntitlementPolicyRule -Name "Training Win7_1" -RemoveExcludedUsers trainin
g\User1 -ExcludedUserFilterEnabled $false
PS C:\Users\administrator.TRAINING> Get-BrokerEntitlementPolicyRule -Name "Training Win7_1"
```

The output of the command is as follows:

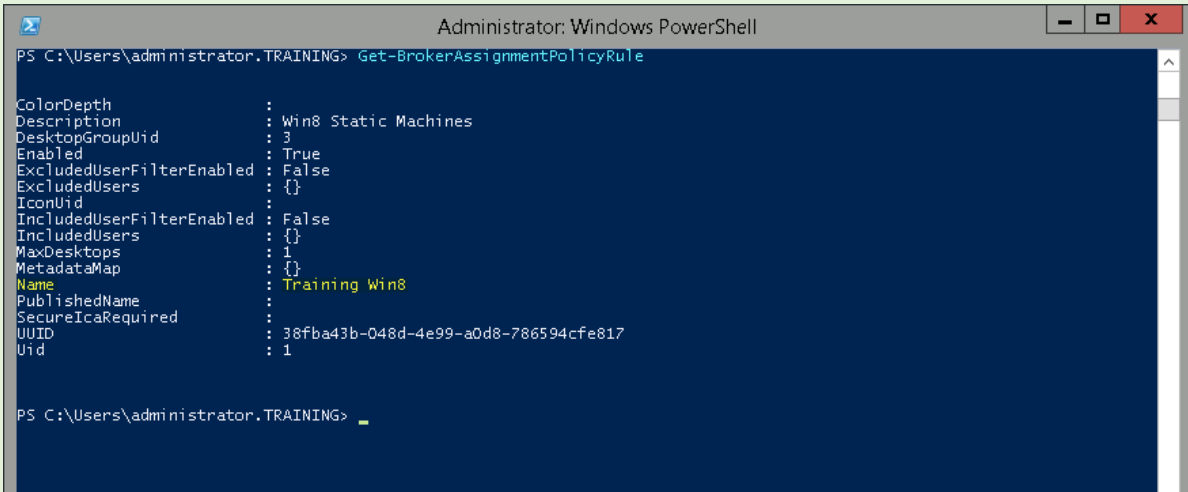
```
BrowserName      : My Win7 Random Desktops $S1-1
ColorDepth       :
Description      :
DesktopGroupUid  : 1
Enabled          : True
ExcludedUserFilterEnabled : False
ExcludedUsers    : {}
IconUid         :
IncludedUserFilterEnabled : False
IncludedUsers    : {}
MetadataMap     : {}
Name            : Training Win7_1
PublishedName    :
SecureIcaRequired :
SessionReconnection : Always
UUID            : 235288e9-e1e0-4a84-a86b-691273ea818d
Uid             : 1
```

The command prompt returns to the prompt: `PS C:\Users\administrator.TRAINING>`

Note: The exclusion has now been removed

12.	<p>Switch back to Win7Client and press F5 to refresh the StoreFront console. If your session has timed out, just log in again as Training\user1 and Citrix123.</p> <p>Once again we can see that based on the three existing Delivery Group user assignments the following three desktops types are available to User1:</p> <ol style="list-style-type: none"> 1. W2K12 RDS Shared Desktop 2. Win7 Random Desktop 3. Win8 Static Desktop 
13.	<p>Switch back to DC2 and login again if needed as training\administrator with password Citrix123</p>

14. Type ***Get-BrokerAssignmentPolicyRule*** into the open **PoSH** window and hit return:

A screenshot of a Windows PowerShell window titled "Administrator: Windows PowerShell". The command prompt shows the command "Get-BrokerAssignmentPolicyRule" being executed. The output is a list of properties and their values for a policy rule. The properties include ColorDepth, Description, DesktopGroupUid, Enabled, ExcludedUserFilterEnabled, ExcludedUsers, IconUid, IncludedUserFilterEnabled, IncludedUsers, MaxDesktops, MetadataMap, Name, PublishedName, SecureIcaRequired, UUID, and Uid. The Name property is highlighted in yellow and shows the value "Training Win8".

```
PS C:\Users\administrator.TRAINING> Get-BrokerAssignmentPolicyRule

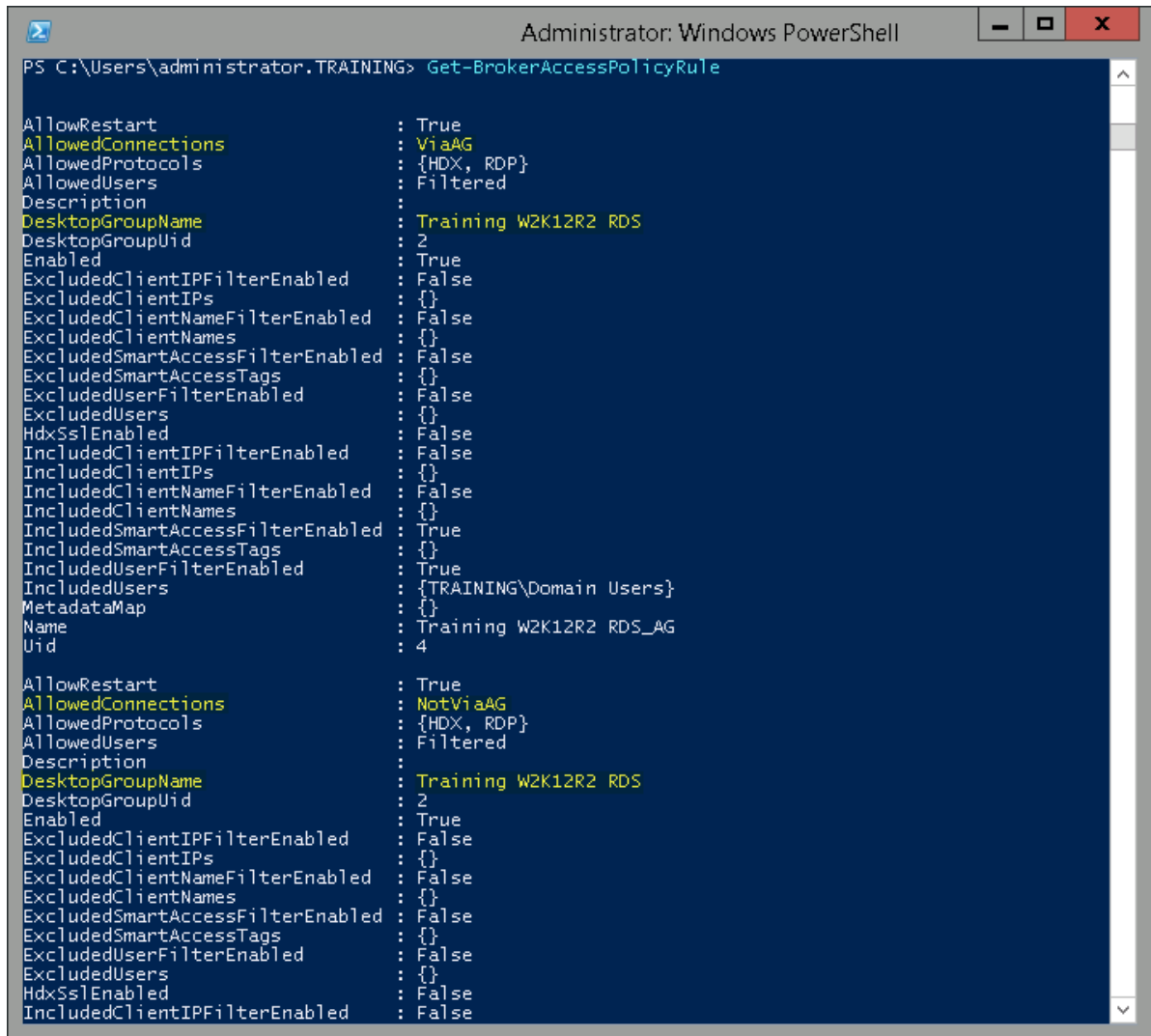
ColorDepth           :
Description           : Win8 Static Machines
DesktopGroupUid       : 3
Enabled              : True
ExcludedUserFilterEnabled : False
ExcludedUsers         : {}
IconUid              :
IncludedUserFilterEnabled : False
IncludedUsers         : {}
MaxDesktops          : 1
MetadataMap           : {}
Name                  : Training Win8
PublishedName         :
SecureIcaRequired     :
UUID                  : 38fba43b-048d-4e99-a0d8-786594cfe817
Uid                   : 1

PS C:\Users\administrator.TRAINING>
```

Note: The Site Assignment Policy contains a single rule by default for each Existing Static or Pooled Static (MCS) Delivery Group and can be used to explicitly control access to member desktops.

The same steps carried out above (Steps **7** through **10**) can also be followed to set and remove explicit exclusions against Static or Dedicated desktops with the **Set-BrokerAssignmentPolicyRule** cmdlet rather than the **Set-BrokerEntitlementPolicyRule** cmdlet.

15. Type ***Get-BrokerAccessPolicyRule*** and hit return:



```
Administrator: Windows PowerShell
PS C:\Users\administrator.TRAINING> Get-BrokerAccessPolicyRule

AllowRestart                : True
AllowedConnections           : ViaAG
AllowedProtocols              : {HDX, RDP}
AllowedUsers                  : Filtered
Description                  :
DesktopGroupName             : Training W2K12R2 RDS
DesktopGroupUid              : 2
Enabled                      : True
ExcludedClientIPFilterEnabled : False
ExcludedClientIPs            : {}
ExcludedClientNameFilterEnabled : False
ExcludedClientNames          : {}
ExcludedSmartAccessFilterEnabled : False
ExcludedSmartAccessTags      : {}
ExcludedUserFilterEnabled    : False
ExcludedUsers                : {}
HdxSslEnabled                : False
IncludedClientIPFilterEnabled : False
IncludedClientIPs            : {}
IncludedClientNameFilterEnabled : False
IncludedClientNames          : {}
IncludedSmartAccessFilterEnabled : True
IncludedSmartAccessTags      : {}
IncludedUserFilterEnabled    : True
IncludedUsers                 : {TRAINING\Domain Users}
MetadataMap                  : {}
Name                         : Training W2K12R2 RDS_AG
Uid                          : 4

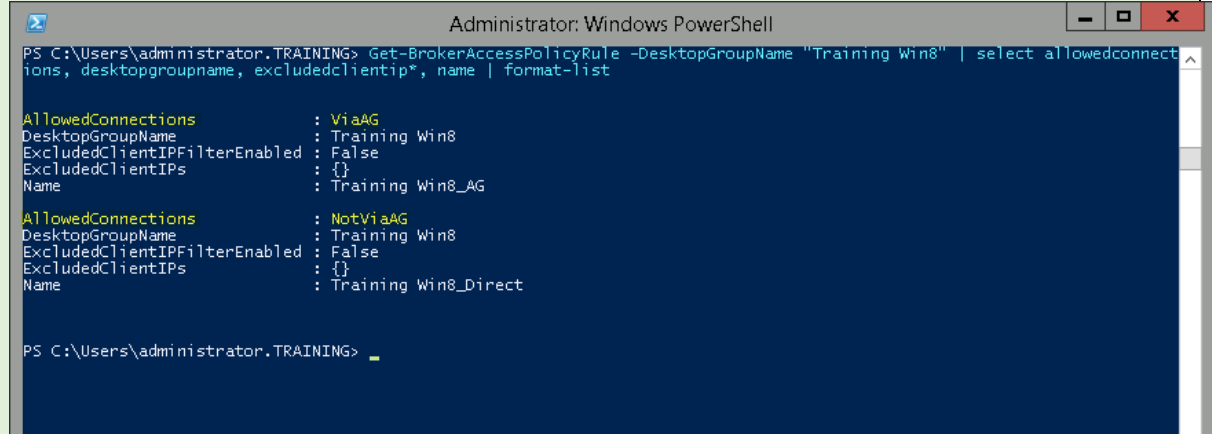
AllowRestart                : True
AllowedConnections           : NotViaAG
AllowedProtocols              : {HDX, RDP}
AllowedUsers                  : Filtered
Description                  :
DesktopGroupName             : Training W2K12R2 RDS
DesktopGroupUid              : 2
Enabled                      : True
ExcludedClientIPFilterEnabled : False
ExcludedClientIPs            : {}
ExcludedClientNameFilterEnabled : False
ExcludedClientNames          : {}
ExcludedSmartAccessFilterEnabled : False
ExcludedSmartAccessTags      : {}
ExcludedUserFilterEnabled    : False
ExcludedUsers                : {}
HdxSslEnabled                : False
IncludedClientIPFilterEnabled : False
```

Note: The Site Access Policy contains two rules by default for each existing Delivery Group regardless of type i.e. Existing Static, Pooled Static (MCS), Existing Random or Pooled Random (MCS) and controls the **conditions** for accessing resources (Desktops & Apps) contained within each Delivery Group.

By default, a Direct connection (**NotViaAG**) rule and a connection via Access Gateway (**ViaAG**) exist.

16. **Copy and paste** the following string into the **PoSH** window and hit return:

Get-BrokerAccessPolicyRule -DesktopGroupName "Training Win8" | select allowedconnections, desktopgroupname, excludedclientip*, name | format-list



```
Administrator: Windows PowerShell
PS C:\Users\administrator.TRAINING> Get-BrokerAccessPolicyRule -DesktopGroupName "Training Win8" | select allowedconnections, desktopgroupname, excludedclientip*, name | format-list

AllowedConnections      : ViaAG
DesktopGroupName        : Training Win8
ExcludedClientIPFilterEnabled : False
ExcludedClientIPs       : {}
Name                    : Training Win8_AG

AllowedConnections      : NotViaAG
DesktopGroupName        : Training Win8
ExcludedClientIPFilterEnabled : False
ExcludedClientIPs       : {}
Name                    : Training Win8_Direct

PS C:\Users\administrator.TRAINING> _
```

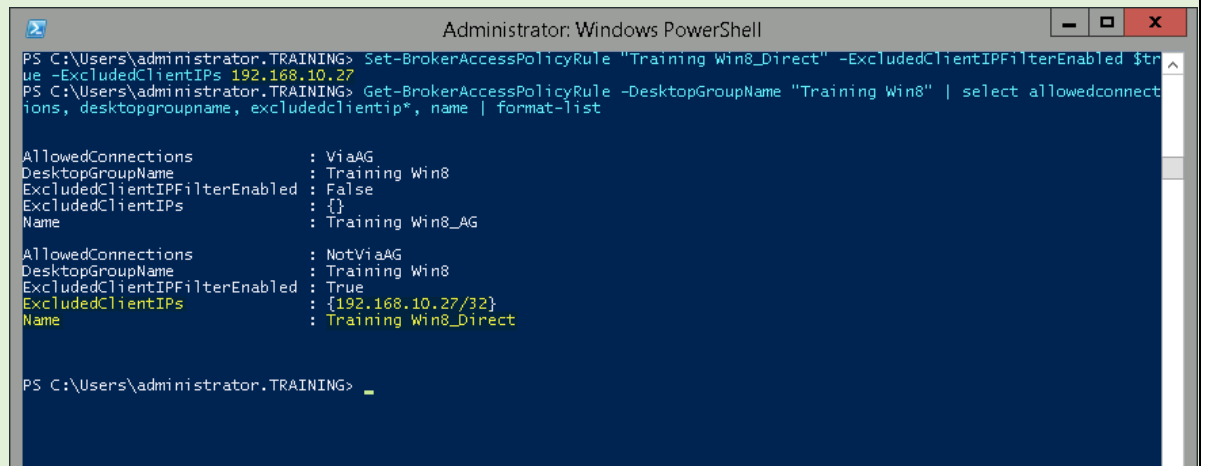
Note: Using the **select** function will allow us to view the relevant information on a single page without having to use the scroll bar. You should use this technique whenever possible for a better overall user experience.

17. **Copy and paste** the following string into the open **PoSH** window and hit return to set the new ClientIP access exclusion:

**Set-BrokerAccessPolicyRule "Training Win8_Direct" -
ExcludedClientIPFilterEnabled \$true -ExcludedClientIPs 192.168.10.27**

18. **Copy and paste** the following string once again into the **PoSH** window and hit return:

Get-BrokerAccessPolicyRule -DesktopGroupName "Training Win8" | select allowedconnections, desktopgroupname, excludedclientip*, name | format-list



```
Administrator: Windows PowerShell
PS C:\Users\administrator.TRAINING> Set-BrokerAccessPolicyRule "Training Win8_Direct" -ExcludedClientIPFilterEnabled $true -ExcludedClientIPs 192.168.10.27
PS C:\Users\administrator.TRAINING> Get-BrokerAccessPolicyRule -DesktopGroupName "Training Win8" | select allowedconnections, desktopgroupname, excludedclientip*, name | format-list

AllowedConnections      : ViaAG
DesktopGroupName        : Training Win8
ExcludedClientIPFilterEnabled : False
ExcludedClientIPs       : {}
Name                    : Training Win8_AG

AllowedConnections      : NotViaAG
DesktopGroupName        : Training Win8
ExcludedClientIPFilterEnabled : True
ExcludedClientIPs       : {192.168.10.27/32}
Name                    : Training Win8_Direct

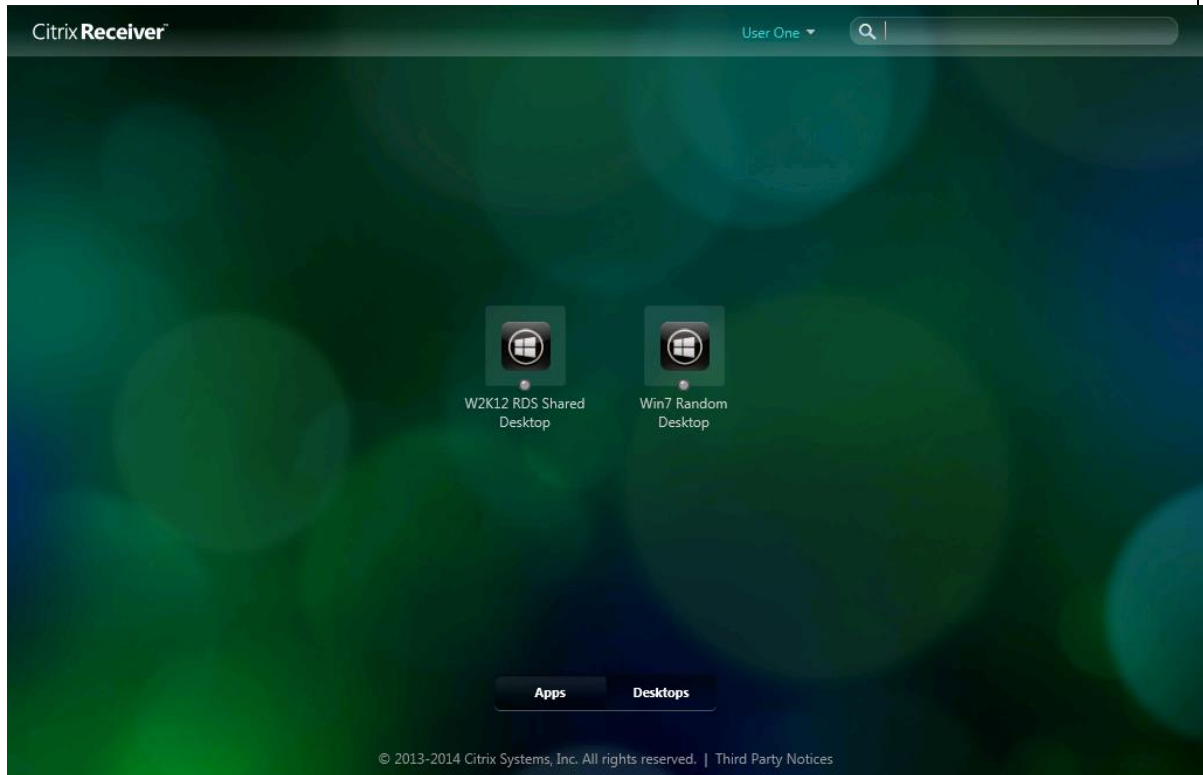
PS C:\Users\administrator.TRAINING> _
```

Note: The new ClientIP exclusion can be clearly seen and verified.

19. Switch back to **Win7Client** and press **F5** or return to refresh the StoreFront console. If your session has timed out, just log in again as **Training\user1** and **Citrix123**.

Based on the Three existing Delivery Group user assignments and the explicit access policy exclusion on the Training Win8 Delivery Group, only the following two desktops types are now available to **User1**:

1. W2K12 RDS Shared Desktop
2. Win7 Random Desktop



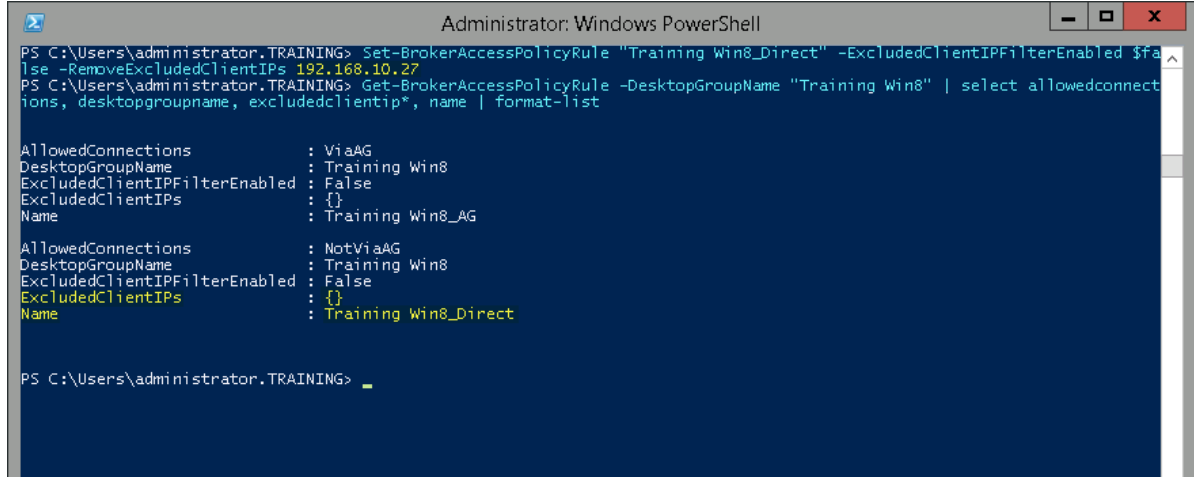
20. Switch back to **DC2** and remove the exclusion by copying **one** of the following strings into the open PoSH window and hitting return:

**Set-BrokerAccessPolicyRule "Training Win8_Direct" -
ExcludedClientIPFilterEnabled \$false -ExcludedClientIPs @()**

OR

**Set-BrokerAccessPolicyRule "Training Win8_Direct" -
ExcludedClientIPFilterEnabled \$false -RemoveExcludedClientIPs 192.168.10.27**

21. **Copy and paste** the following string once again into the **PoSH** window and hit return:
- Get-BrokerAccessPolicyRule -DesktopGroupName "Training Win8" | select allowedconnections, desktopgroupname, excludedclientip*, name | format-list**



```
Administrator: Windows PowerShell
PS C:\Users\administrator.TRAINING> Set-BrokerAccessPolicyRule "Training Win8_Direct" -ExcludedClientIPFilterEnabled $false -RemoveExcludedClientIPs 192.168.10.27
PS C:\Users\administrator.TRAINING> Get-BrokerAccessPolicyRule -DesktopGroupName "Training Win8" | select allowedconnections, desktopgroupname, excludedclientip*, name | format-list

AllowedConnections      : ViaAG
DesktopGroupName       : Training Win8
ExcludedClientIPFilterEnabled : False
ExcludedClientIPs       : {}
Name                   : Training Win8_AG

AllowedConnections      : NotViaAG
DesktopGroupName       : Training Win8
ExcludedClientIPFilterEnabled : False
ExcludedClientIPs       : {}
Name                   : Training Win8_Direct

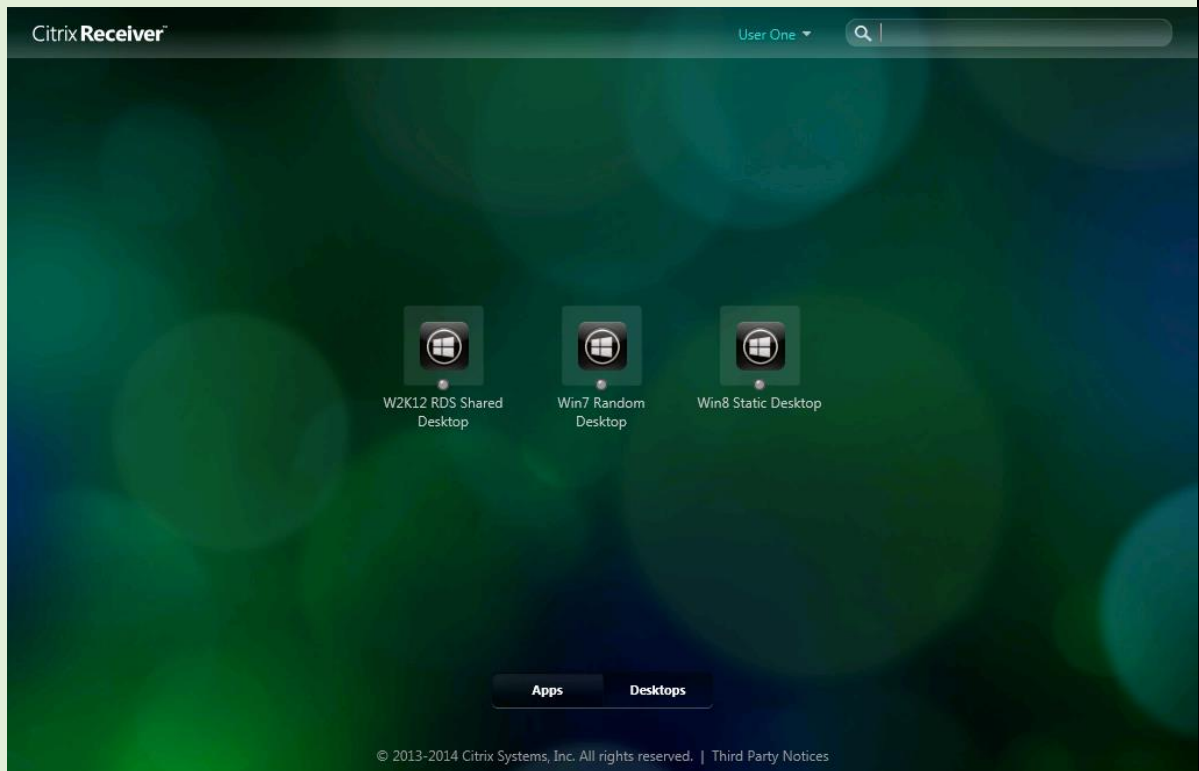
PS C:\Users\administrator.TRAINING> _
```

Note: The previously set ClientIP exclusions have been removed.

22. Switch back to Win7Client and press **F5** or return to refresh StoreFront console. If your session has timed out, just log in again as **Training\user1** and **Citrix123**.

Once again we can see that based on the three existing Delivery Group user assignments the following three desktops types are available to **User1**:

1. W2K12 RDS Shared Desktop
2. Win7 Random Desktop
3. Win8 Static Desktop



23.	Logout of StoreFront and close IE.
24.	Congratulations you have finished this exercise.

Exercise Summary

Takeaways from this exercise:

- XenDesktop includes a number of Site wide policies which can be used to explicitly control access to Desktops and the conditions under which Desktops are accessed. Some of those discussed during the above exercise include:
 - Broker Entitlement Policy (Get-BrokerEntitlementPolicyRule)
 - Controls access to Existing Random or Pooled Random (MCS) machines
 - Broker Assignment Policy (Get-BrokerAssignmentPolicyRule)
 - Controls access to Existing Static or Pooled Static (MCS) machines
 - Broker Access Policy (Get-BrokerAccessPolicyRule)
 - Controls the conditions for accessing resources contained within each Delivery Group
- XenDesktop also includes a number of site wide policies for controlling access to Applications and the conditions under which Applications are accessed. These include:
 - Broker Application Entitlement Policy (Get-Broker**App**EntitlementPolicyRule)
 - Controls access to applications on Existing Random or Pooled Random (MCS) machines
 - Broker Application Assignment Policy (Get-Broker**App**AssignmentPolicyRule)
 - Controls access to applications on Existing Static or Pooled Static (MCS) machines
- For complete granular control of access to site resources through site policies, the PoSH SDK must be used.
- For more cool info & tips on XenDesktop follow **@XDtipster** and **@XDInformer** on twitter.

Exercise 5


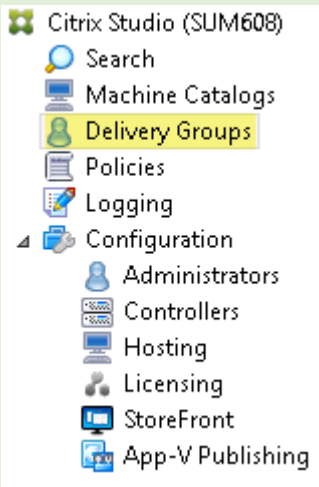
A Timely Reminder: Configuring Multistage Power Policy Rules

Overview

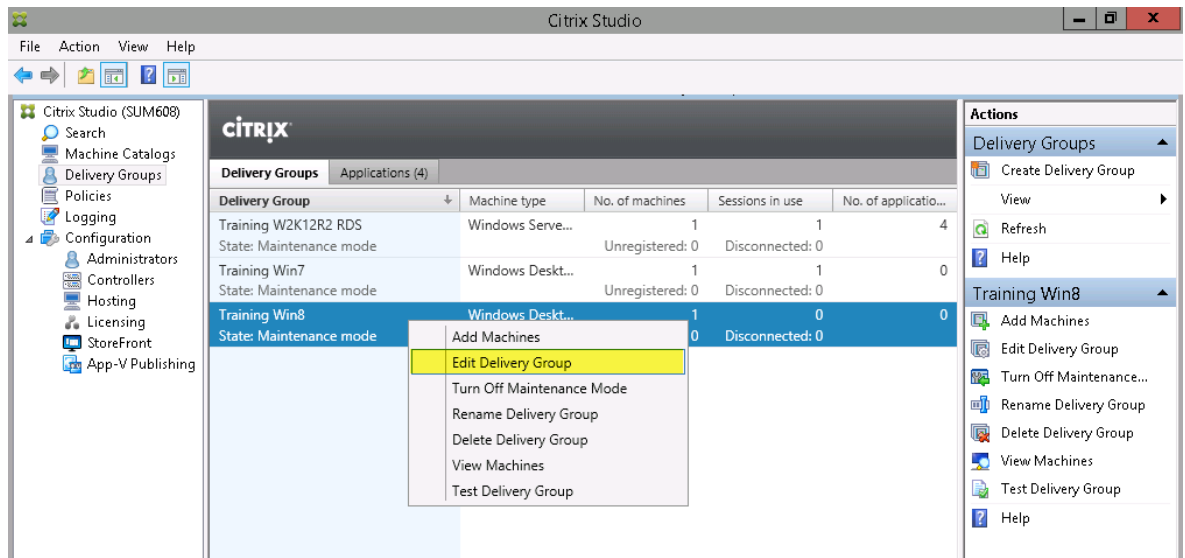
In this exercise we will take a look at the Extended Disconnect Power Policy Rules hidden within the SDK. Designed to control the actions to be performed after a second extended & configurable period of a user session disconnecting in peak and off-peak hours, extended disconnect policy rules along with the standard disconnect policy rules, allow administrators to apply multistage power policies settings such as initially suspending a machine shortly after a session disconnect occurs, and then later powering-off the machine if the session has not been reconnected.

Step by step guidance

Estimated time to complete this lab: 25 minutes.

Step	
1.	<u>IMPORTANT</u> : As a pre-req for this exercise, please ensure that you are running XenCenter and have connected to your assigned host as per the portal details. Note : XenCenter is required to monitor power state of VMs (namely Win8VM)
2.	Switch back to and login again if needed to DC2 as training\administrator with password: Citrix123 .
3.	If not already running. Launch Citrix Studio from the taskbar icon: 
4.	Click on the Delivery Group node within Citrix Studio: 

5. Right click on the Training Win8 Delivery Group and select **“Edit Delivery Group”**:



6. Click on the “**Power Management**” tab and set the Standard Power Policy Settings as follows and click **OK**:

During Peak Hours:

When disconnected: 5 mins

Action: Suspend

IMPORTANT: If the current time when working through this exercise, falls outside the defined peak hours below then make the adjustment as needed to the peak hour range and click **OK**.

Studio

Users

Machine allocation

Delivery Type

User Settings

StoreFront

Power Management

Access Policy

Power manage machines: Weekdays

Peak hours: 00:00 03:00 06:00 09:00 12:00 15:00 18:00 21:00 00:00

During peak hours:

When disconnected 5 mins Suspend

When logged off 0 mins No action

During off-peak hours:

When disconnected 0 mins No action

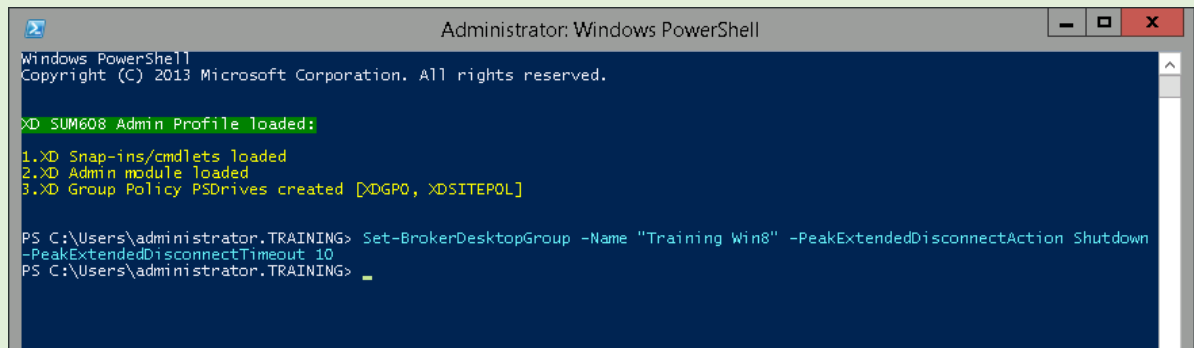
When logged off 0 mins No action

OK Cancel Apply

7. Launch a PowerShell window on **DC2** by clicking on the **PoSH** icon on the taskbar:



8. Type **Set-BrokerDesktopGroup -Name "Training Win8" – PeakExtendedDisconnectAction Shutdown –PeakExtendedDisconnectTimeout 10** and hit return:

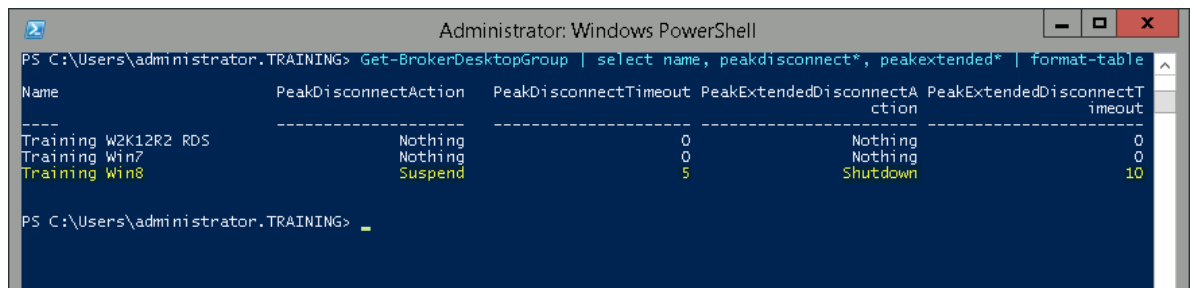


```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2013 Microsoft Corporation. All rights reserved.

XD SUM608 Admin Profile loaded:
1.XD Snap-ins/cmdlets loaded
2.XD Admin module loaded
3.XD Group Policy PSDrives created [XDGP0, XDSITEPOL]

PS C:\Users\administrator.TRAINING> Set-BrokerDesktopGroup -Name "Training Win8" -PeakExtendedDisconnectAction Shutdown
-PeakExtendedDisconnectTimeout 10
PS C:\Users\administrator.TRAINING>
```

9. Type **Get-BrokerDesktopGroup | select name, peakdisconnect*, peakextended* / format-table** to view the newly configured multistage power policy rule for the Training Win8 Delivery Group:



```
Administrator: Windows PowerShell
PS C:\Users\administrator.TRAINING> Get-BrokerDesktopGroup | select name, peakdisconnect*, peakextended* | format-table

Name                                PeakDisconnectAction  PeakDisconnectTimeout PeakExtendedDisconnectA PeakExtendedDisconnectT
-----                                -
Training W2K12R2 RDS                Nothing               0                    Nothing                 0
Training Win7                        Nothing               0                    Nothing                 0
Training Win8                        Suspend              5                    Shutdown                10

PS C:\Users\administrator.TRAINING>
```

10. Login to **Win7Client** as **training\administrator** with password **Citrix123**.

11. Launch Internet Explorer from the taskbar icon and enter the Receiver for Web URL in the address bar:

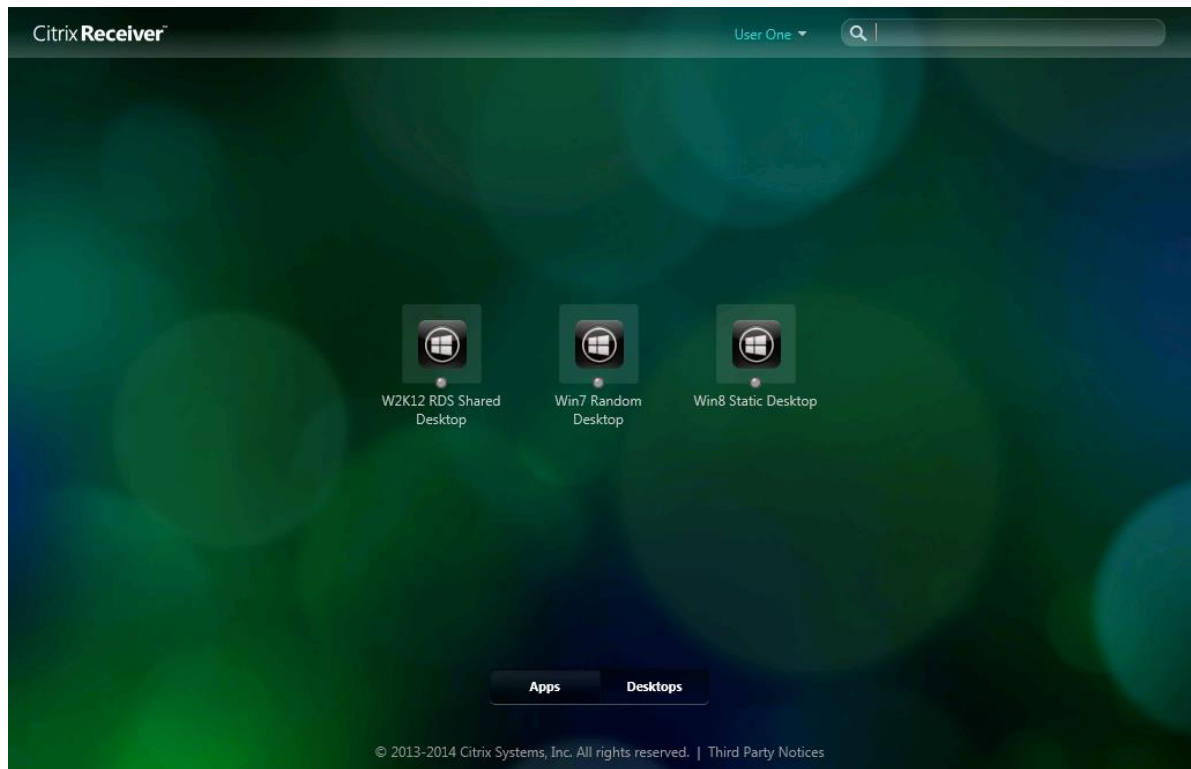
<http://sf.training.lab/Citrix/SummitWeb/>



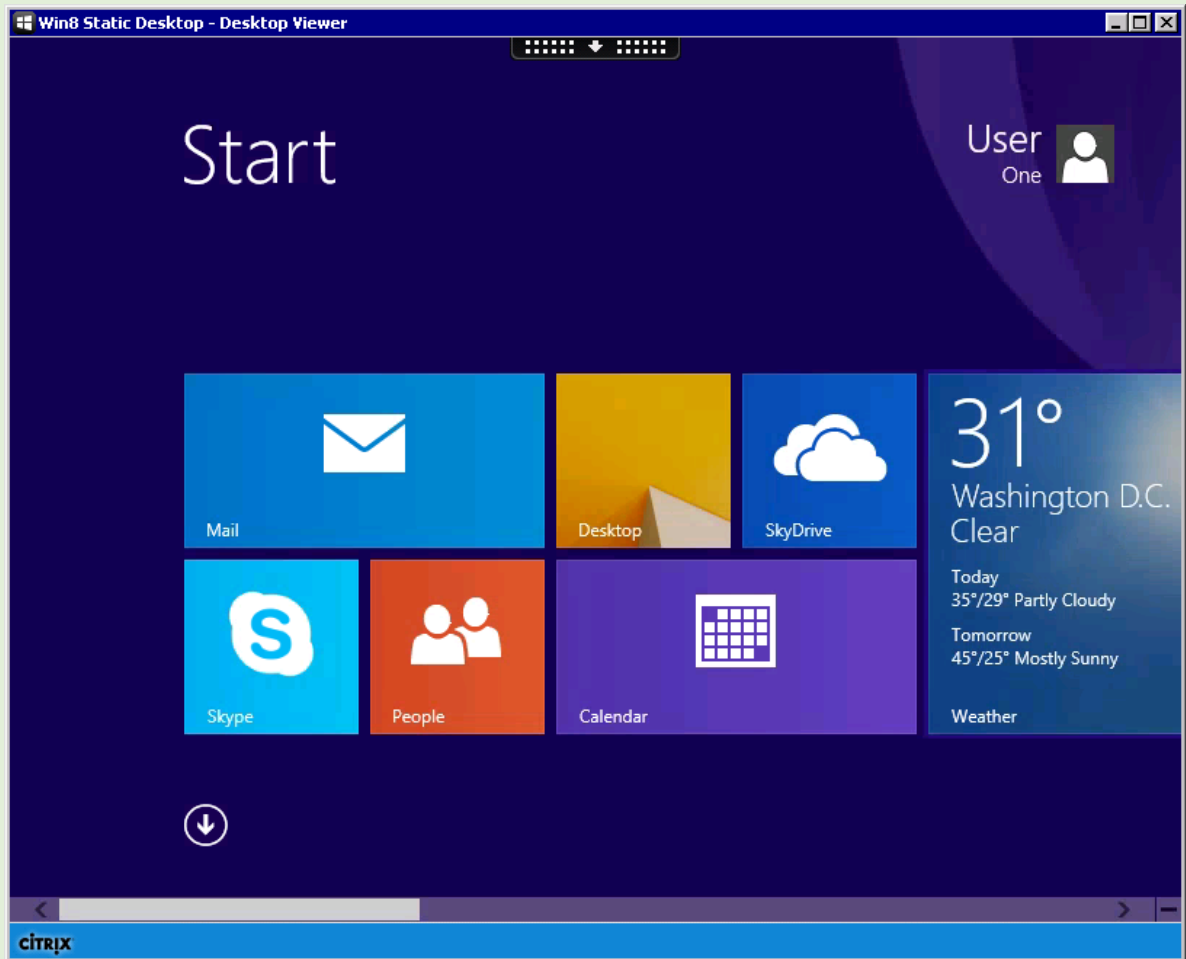
12. Log on to **StoreFront** using **Training\user1** and **Citrix123**.



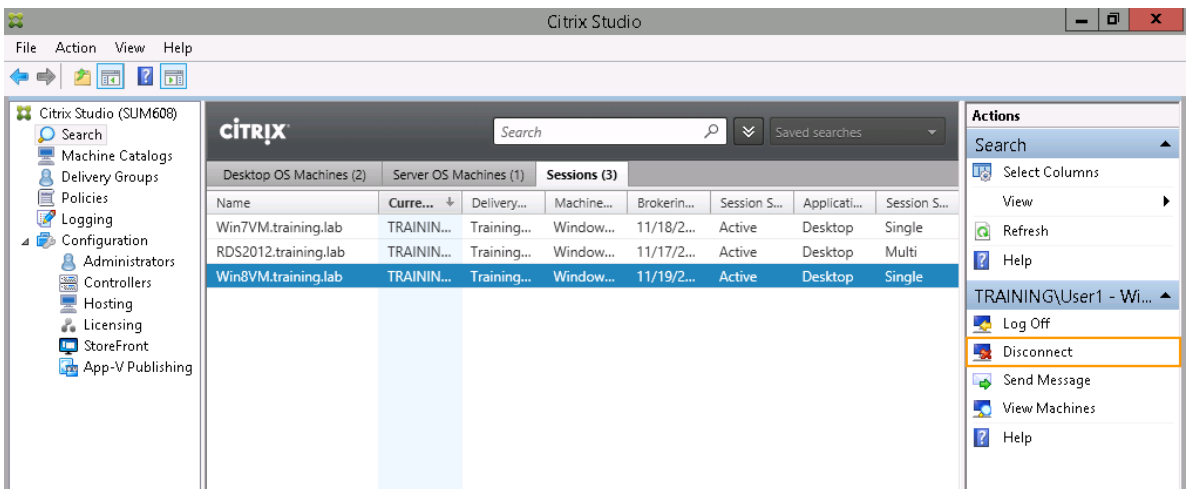
13. Once logged into StoreFront, launch the Win8 Static Desktop:



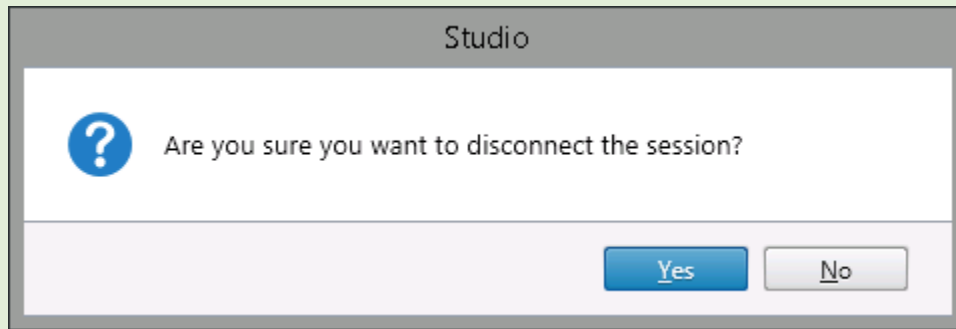
14. Confirm that you have connected to the Win8 Desktop before proceeding to the next step:



15. Switch back to and login again if needed to **DC2** as **training\administrator** with password **Citrix123** and disconnect the session now running on the Win8VM from within Citrix Studio:
- Click on the Search node → Sessions tab → Highlight the Win8VM → select Disconnect from the Actions menu.

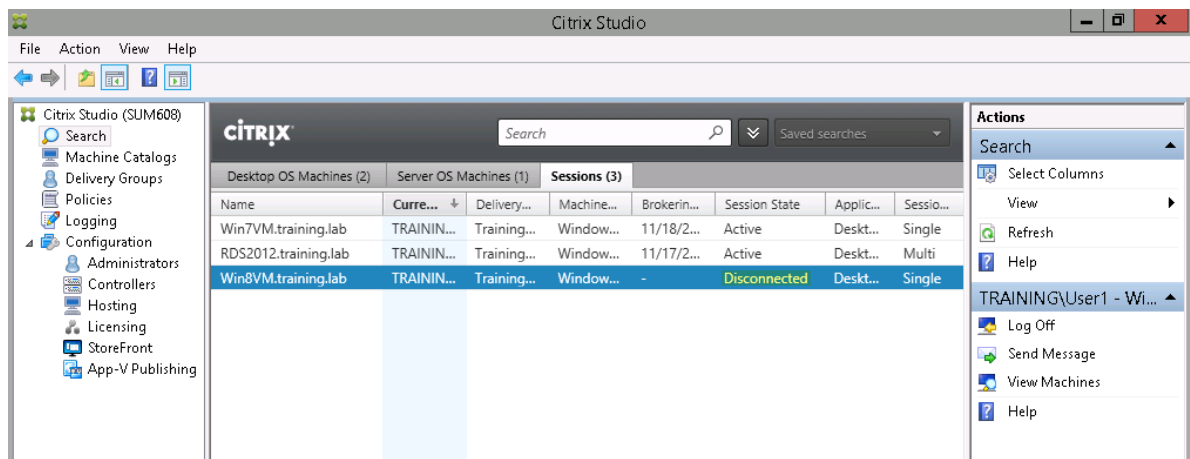


16. Click **Yes** when asked if you are sure you want to disconnect the session:

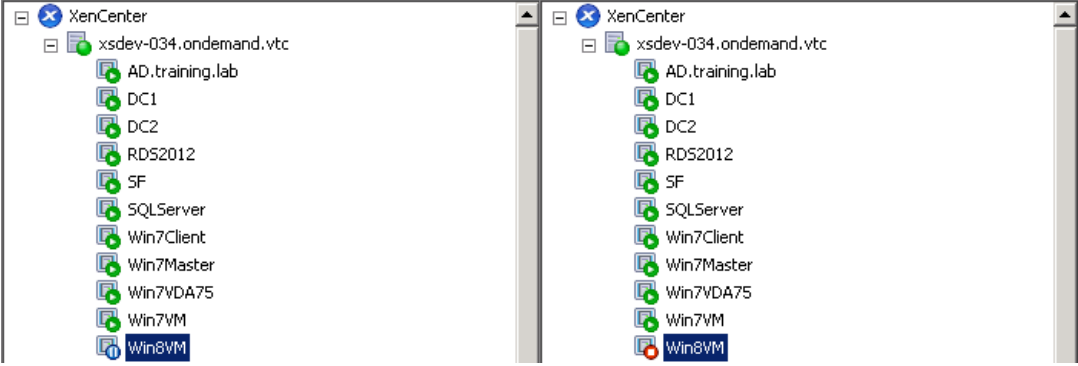


Did you know: By Default, the Session Idle timer and Interval HDX policy settings determine that a machine session will only be disconnected if idle for 24hrs.

17. Click **Refresh** on the Actions bar and ensure that the session state is now reporting as **Disconnected**:



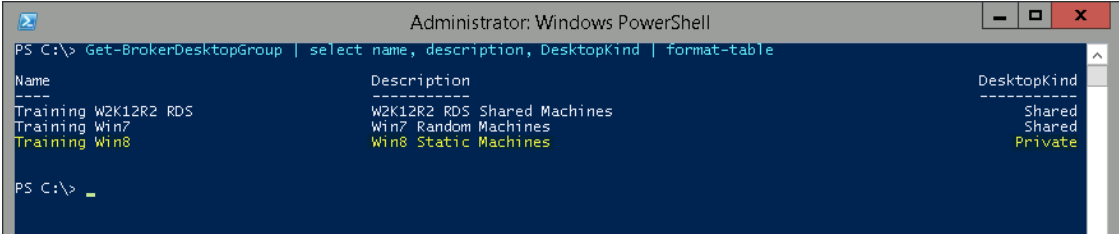
18. **IMPORTANT:** If using RDP manager to access the lab VMs, switch to XenCenter (connect to your assigned host) and monitor the behavior of the **Win8VM** (Now configured with a multistage disconnect power policy rule).

19.	<p>Observations Within XenCenter:</p> <p>After being disconnected for 5 mins: The Win8VM will suspend</p> <p>After being disconnected for a further 5 mins: The Win8VM will shutdown</p>  <p>Note: The extended disconnect setting timer starts from the initial disconnect and not after the initial disconnect action has taken place. For this reason. The Win8VM should shutdown 10 mins after the session has been disconnected and NOT 15 mins.</p>
20.	<p>Congratulations, you have finished this lab exercise.</p>

Exercise Summary

Takeaways from this exercise:

- Extended Disconnect settings for Peak and Off-Peak times can be configured through PoSH as part of a Multistage Power Policy Rule and applied to a supported Delivery Group
- Extended Disconnect settings and therefore Multistage Power Policy Rules are only supported with Private Deskops & Delivery Groups.



```

Administrator: Windows PowerShell
PS C:\> Get-BrokerDesktopGroup | select name, description, DesktopKind | format-table
-----
Name                Description                DesktopKind
-----
Training W2K12R2 RDS W2K12R2 RDS Shared Machines Shared
Training Win7        Win7 Random Machines      Shared
Training Win8        Win8 Static Machines      Private
PS C:\>

```

- By Default with XenApp & XenDesktop 7.6, idle machine sessions will be disconnected after a period of 24hrs.
- For more cool info & tips on XenDesktop follow [@XDtipster](#) and [@XDInformer](#) on twitter.

Exercise 6

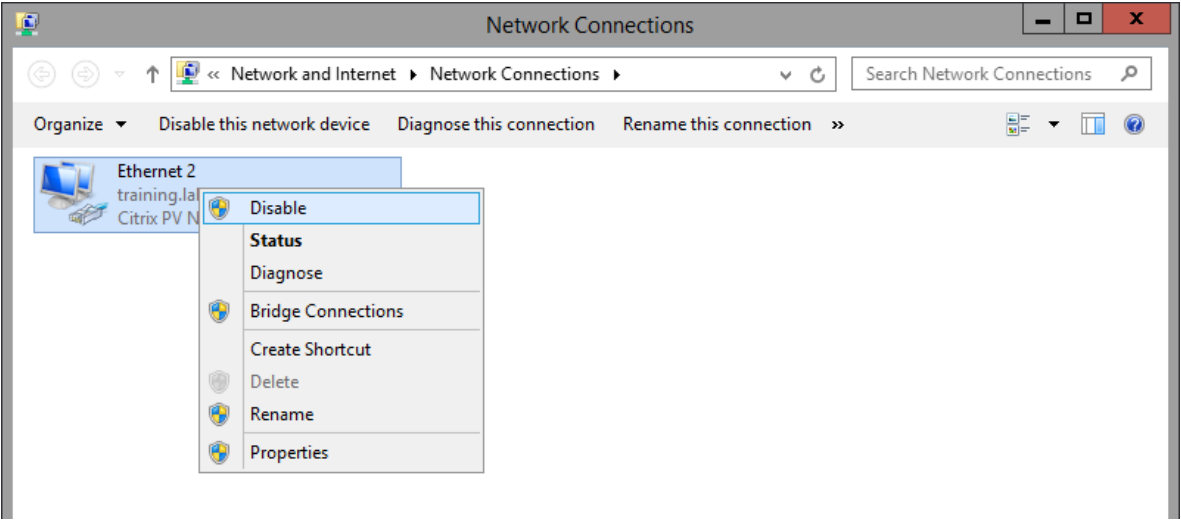

A Final Step: Manually Removing a Controller from a Site (Optional)

Overview

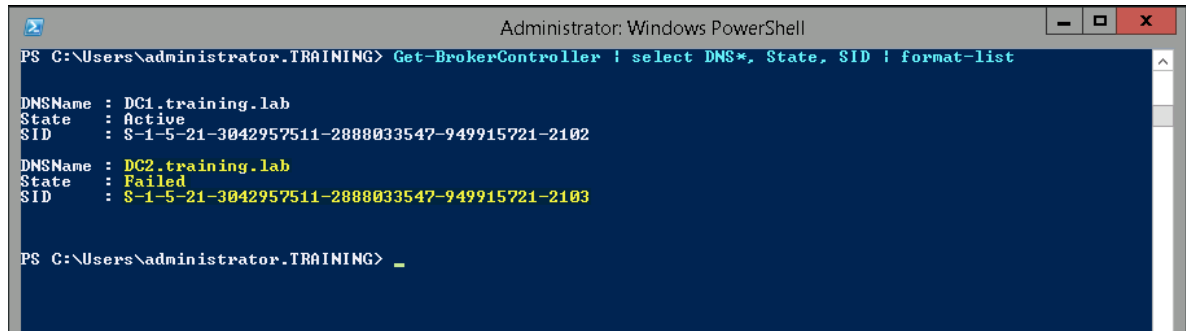
In this exercise you will learn how to manually remove a controller from a site DB using PowerShell, simulating a situation where you cannot gracefully remove an orphaned or misbehaving Controller through Citrix Studio.

Step by step guidance

Estimated time to complete this lab: 25 minutes.

Step	
1.	<p>To Simulate a failed Controller disable the NIC on DC2 through XenCenter:</p> <ol style="list-style-type: none">1. Logon to DC2 at the Console using the Default Desktop option in XenCenter.2. Right click on the Ethernet 2 adapter and select Disable. 
2.	<p>Minimize XenCenter and using Remote Desktop Manager, login to DC1 as training\administrator with password Citrix123 and launch a PowerShell window by clicking on the PoSH icon on the taskbar:</p> 

3. Run ***Get-BrokerController / select DNS*, State, SID / format-list*** to view the state of each Site Controller and take note of the SID for **DC2**:



The screenshot shows a Windows PowerShell window titled "Administrator: Windows PowerShell". The command prompt shows the command `Get-BrokerController | select DNS*, State, SID | format-list` being executed. The output displays the state of two Site Controllers: DC1 and DC2. DC1 is in an "Active" state, while DC2 is in a "Failed" state. The SID for DC2 is highlighted in yellow.

```
PS C:\Users\administrator.TRAINING> Get-BrokerController | select DNS*, State, SID | format-list

DNSName : DC1.training.lab
State   : Active
SID     : S-1-5-21-3042957511-2888033547-949915721-2102

DNSName : DC2.training.lab
State   : Failed
SID     : S-1-5-21-3042957511-2888033547-949915721-2103

PS C:\Users\administrator.TRAINING> _
```

4. Launch Notepad on **DC1** and copy in the following and save as **createevictionscript.ps1** on the root of C:\:


```
#Create eviction scripts for each FMA service running on DC2.training.lab and append the output to a
single evict.sql file to be executed against the SQLServer\CitrixSummit608 DB.

Write-Host "ACTION COMPLETE - Individual eviction scripts created for each FMA service on
DC2.training.lab and appended to a single evict.sql file to be executed against the CitrixSummit608
DB running on the SQLServer vm." -foregroundcolor white -backgroundcolor darkgreen

Write-Host "Script location = c:\evict.sql" -foregroundcolor yellow -backgroundcolor black

" "

" "

Get-BrokerDBSchema -DatabaseName CitrixSummit608 -ScriptType evict -sid S-1-5-21-
3042957511-2888033547-949915721-2103 | out-file c:\evict.sql

Get-ConfigDBSchema -DatabaseName CitrixSummit608 -ScriptType evict -sid S-1-5-21-
3042957511-2888033547-949915721-2103 | out-file c:\evict.sql -append

Get-HypDBSchema -DatabaseName CitrixSummit608 -ScriptType evict -sid S-1-5-21-3042957511-
2888033547-949915721-2103 | out-file c:\evict.sql -append

Get-ProvDBSchema -DatabaseName CitrixSummit608 -ScriptType evict -sid S-1-5-21-3042957511-
2888033547-949915721-2103 | out-file c:\evict.sql -append

Get-AcctDBSchema -DatabaseName CitrixSummit608 -ScriptType evict -sid S-1-5-21-3042957511-
2888033547-949915721-2103 | out-file c:\evict.sql -append

Get-AdminDBSchema -DatabaseName CitrixSummit608 -ScriptType evict -sid S-1-5-21-
3042957511-2888033547-949915721-2103 | out-file c:\evict.sql -append

Get-LogDBSchema -DatabaseName CitrixSummit608 -ScriptType evict -sid S-1-5-21-3042957511-
2888033547-949915721-2103 | out-file c:\evict.sql -append

Get-EnvTestDBSchema -DatabaseName CitrixSummit608 -ScriptType evict -sid S-1-5-21-
3042957511-2888033547-949915721-2103 | out-file c:\evict.sql -append

Get-MonitorDBSchema -DatabaseName CitrixSummit608 -ScriptType evict -sid S-1-5-21-
3042957511-2888033547-949915721-2103 | out-file c:\evict.sql -append

Get-SfDBSchema -DatabaseName CitrixSummit608 -ScriptType evict -sid S-1-5-21-3042957511-
2888033547-949915721-2103 | out-file c:\evict.sql -append

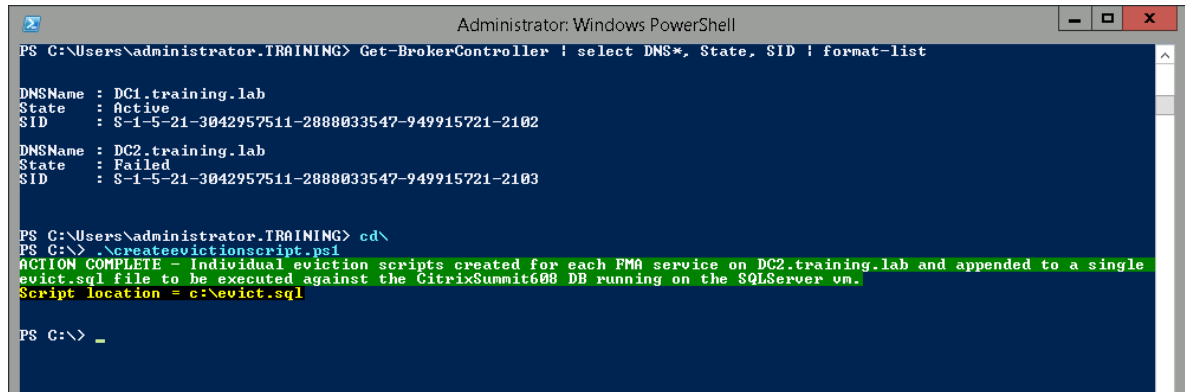
Get-AnalyticsDBSchema -DatabaseName CitrixSummit608 -ScriptType evict -sid S-1-5-21-
3042957511-2888033547-949915721-2103 | out-file c:\evict.sql -append
```

Note: A previously created version of this script can also be found at **\\AD\Software\Scripts**

5. Run the newly created eviction script from the root of PoSH: (To get to the root just type **cd** & hit return)

C:\> .\createevictionscript.ps1

Reminder: You should be running this script from DC1



```
Administrator: Windows PowerShell
PS C:\Users\administrator.TRAINING> Get-BrokerController | select DNS*, State, SID | format-list

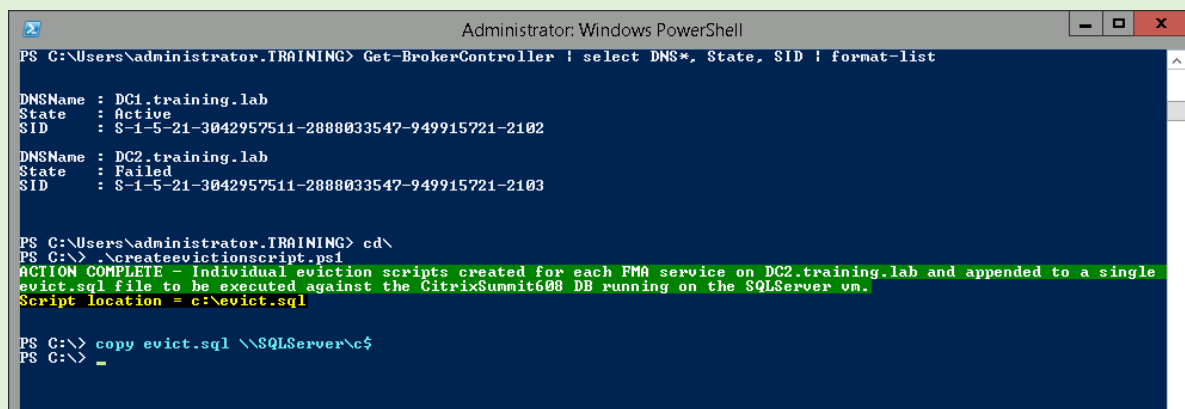
DNSName : DC1.training.lab
State   : Active
SID     : S-1-5-21-3042957511-2888033547-949915721-2102

DNSName : DC2.training.lab
State   : Failed
SID     : S-1-5-21-3042957511-2888033547-949915721-2103

PS C:\Users\administrator.TRAINING> cd\
PS C:\> .\createevictionscript.ps1
ACTION COMPLETE - Individual eviction scripts created for each FMA service on DC2.training.lab and appended to a single
evict.sql file to be executed against the CitrixSummit608 DB running on the SQLServer vm.
Script location = c:\evict.sql

PS C:\> _
```

6. Type **copy evict.sql \\SQLServer\c\$** and hit return to copy the single eviction script to the root of the **SQLServer** VM:



```
Administrator: Windows PowerShell
PS C:\Users\administrator.TRAINING> Get-BrokerController | select DNS*, State, SID | format-list

DNSName : DC1.training.lab
State   : Active
SID     : S-1-5-21-3042957511-2888033547-949915721-2102

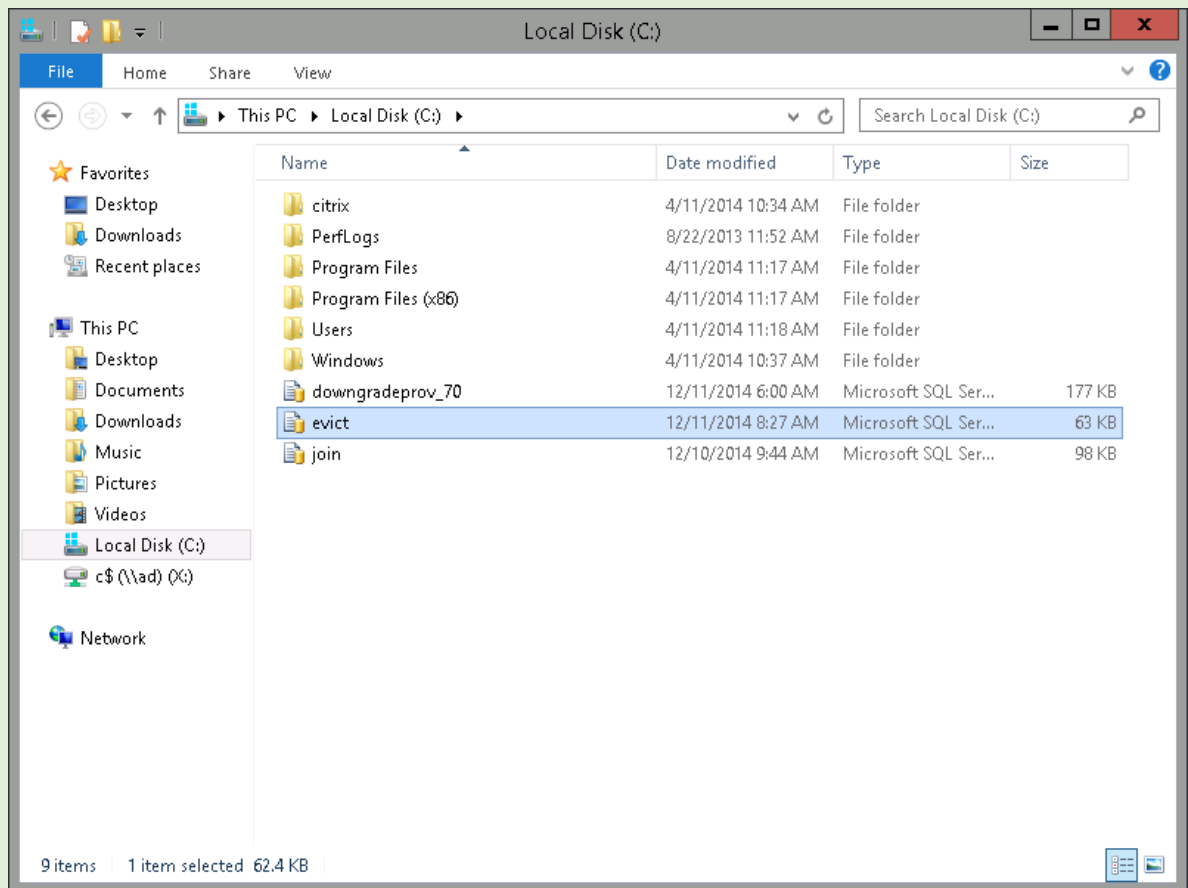
DNSName : DC2.training.lab
State   : Failed
SID     : S-1-5-21-3042957511-2888033547-949915721-2103

PS C:\Users\administrator.TRAINING> cd\
PS C:\> .\createevictionscript.ps1
ACTION COMPLETE - Individual eviction scripts created for each FMA service on DC2.training.lab and appended to a single
evict.sql file to be executed against the CitrixSummit608 DB running on the SQLServer vm.
Script location = c:\evict.sql

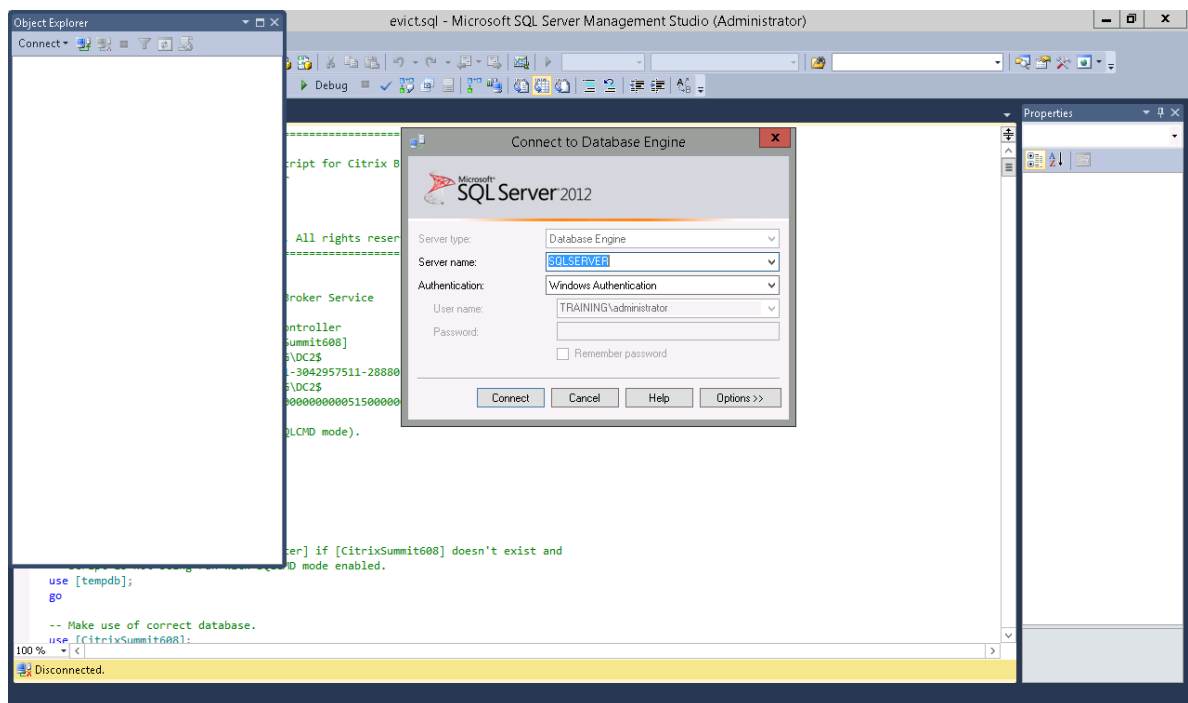
PS C:\> copy evict.sql \\SQLServer\c$
PS C:\> _
```

7. Login to **SQLServer** as **training\administrator** with password **Citrix123**.

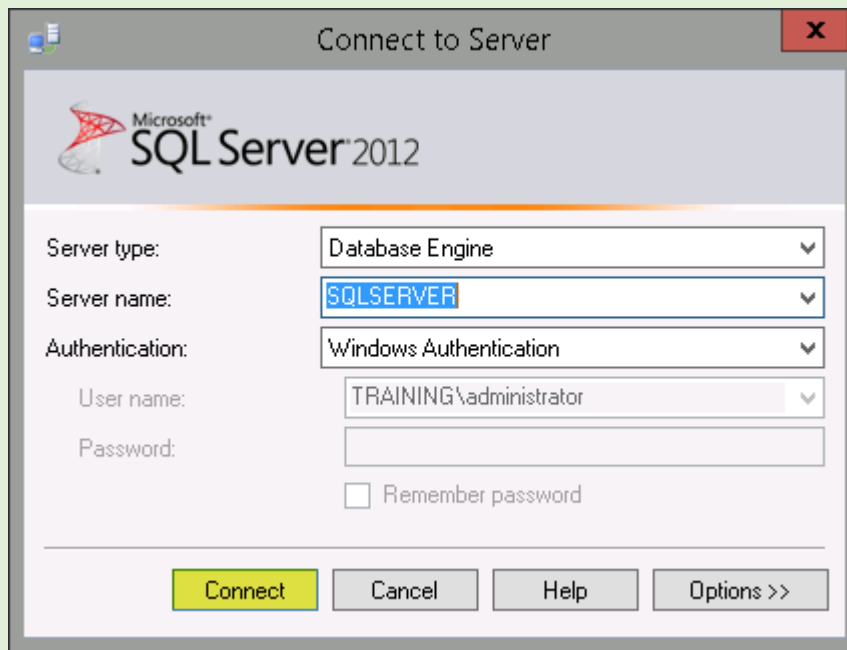
8. Browse the local disk and double click on the **evict.sql** file:



9. Click **Connect** on the Database engine authentication dialog box:

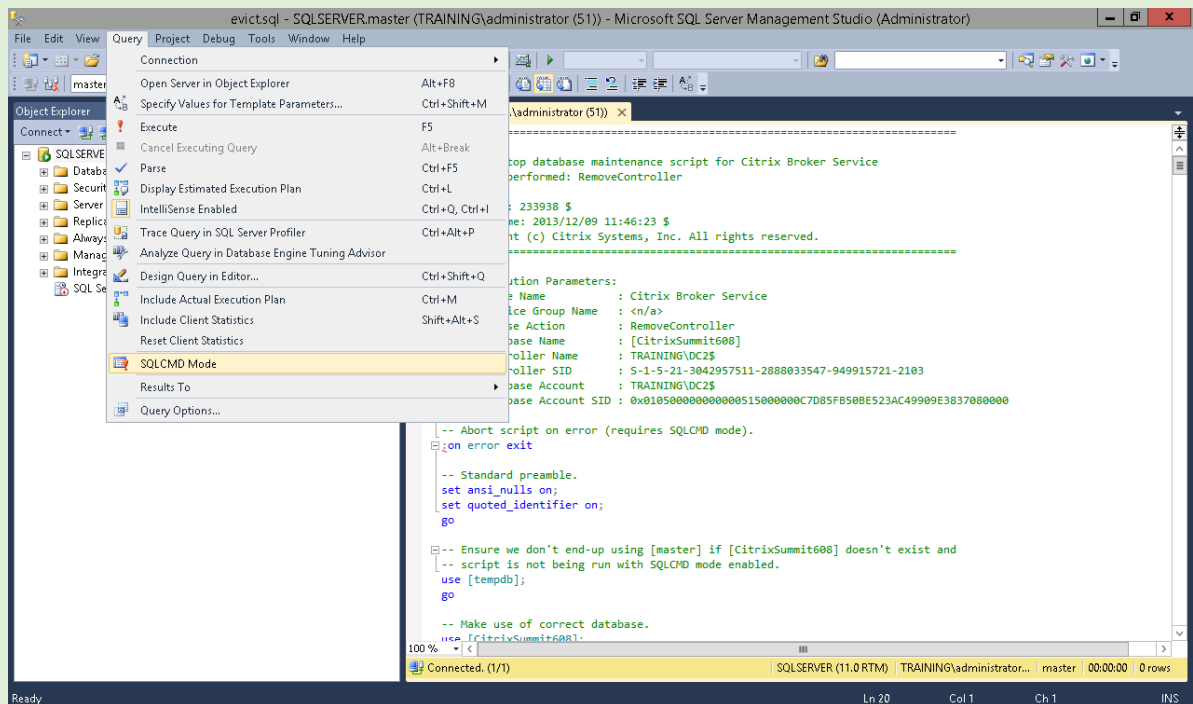


10. Click **Connect** again to authenticate to the Server:



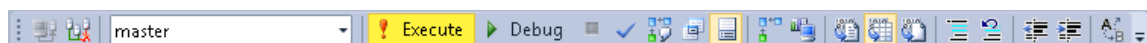
11. Click inside the **evict.sql** script window.

12. Click on **Query** and select **SQLCMD Mode**:

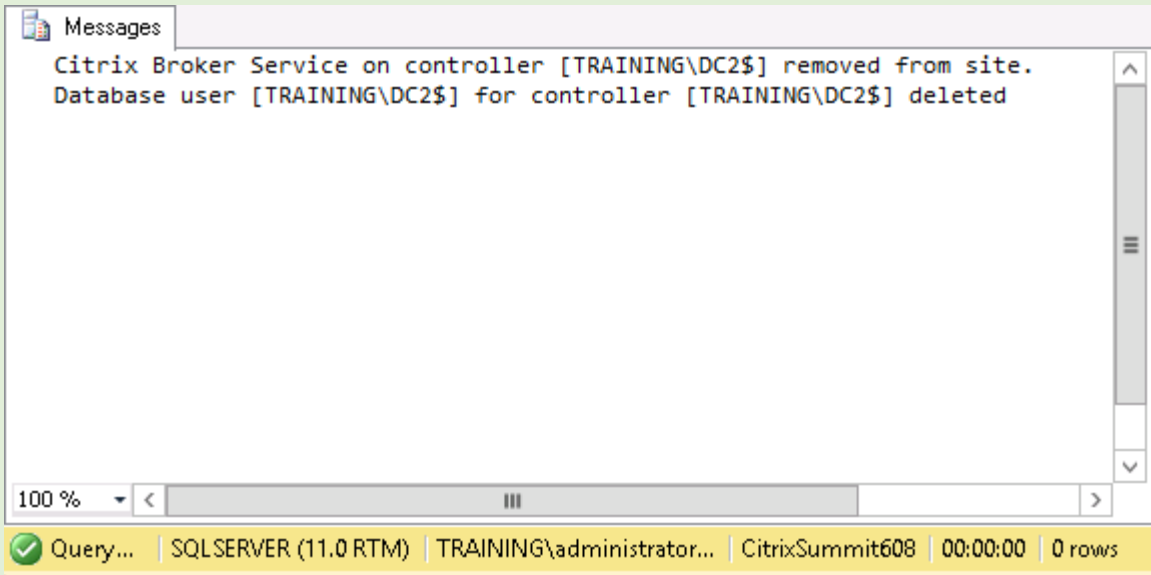


13. Press **F5** or the Execute Icon on the Toolbar to execute the **evict.sql** script

Note: There is no requirement to select the **CitrixSummit608** DB to run the script against as the script itself determines this.

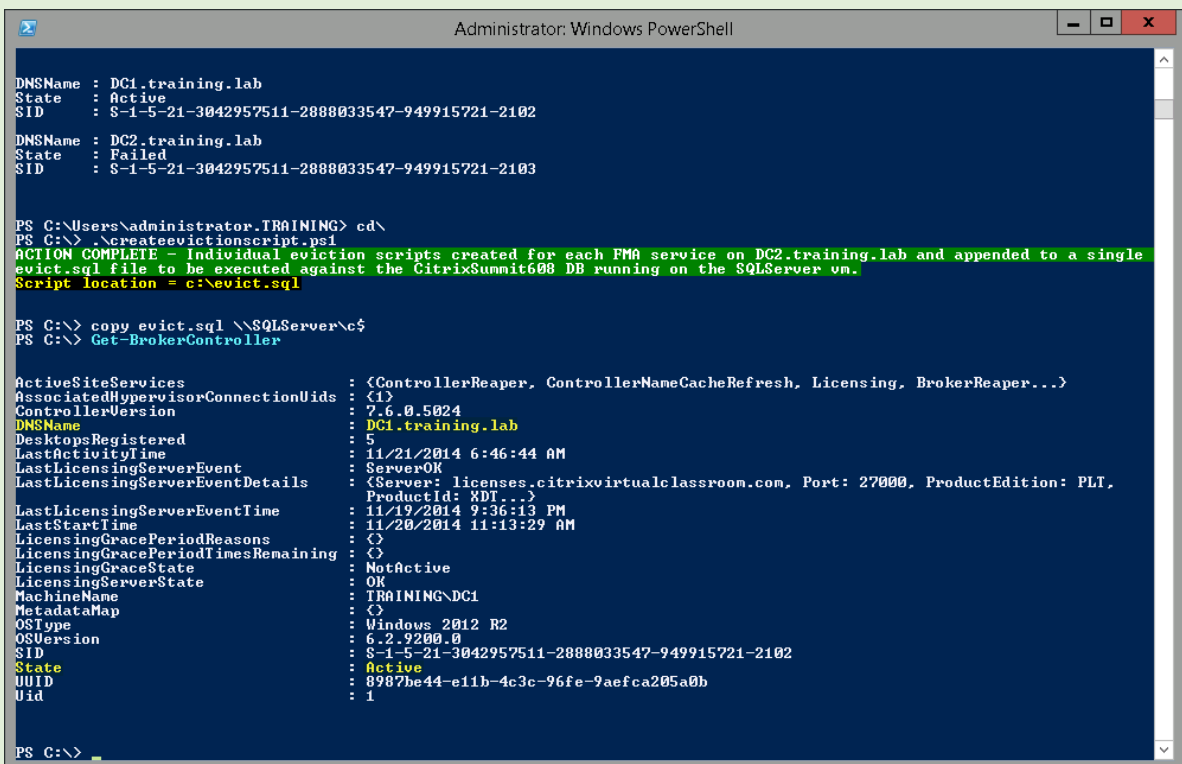


14. **Observation:** Script runs without errors and removes the Broker Service and Database user for a/c for DC2 from the Site DB:

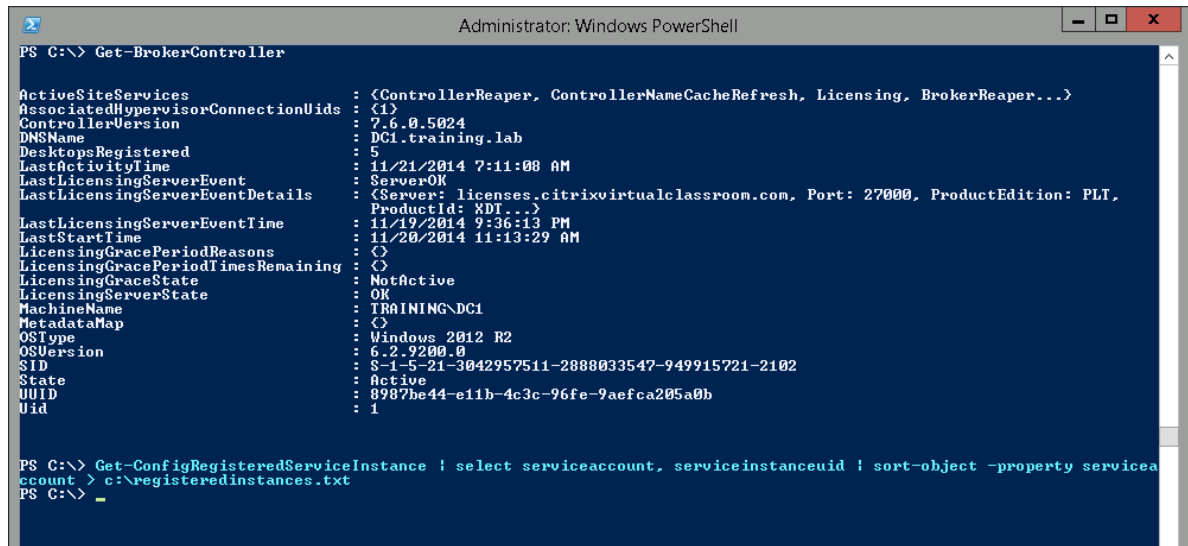


15. Switch back to **DC1** (Controller where you created the eviction script) and login again if needed as **training\administrator** with password **Citrix123**.

16. Type **Get-BrokerController** and verify that **DC2.training.lab** is no longer listed as a Site Controller:



17. Type **Get-ConfigRegisteredServiceInstance | select serviceaccount, serviceinstanceuid | sort-object -property serviceaccount > c:\registeredinstances.txt** and hit return to export the complete list of services registered against the Site Configuration Service:



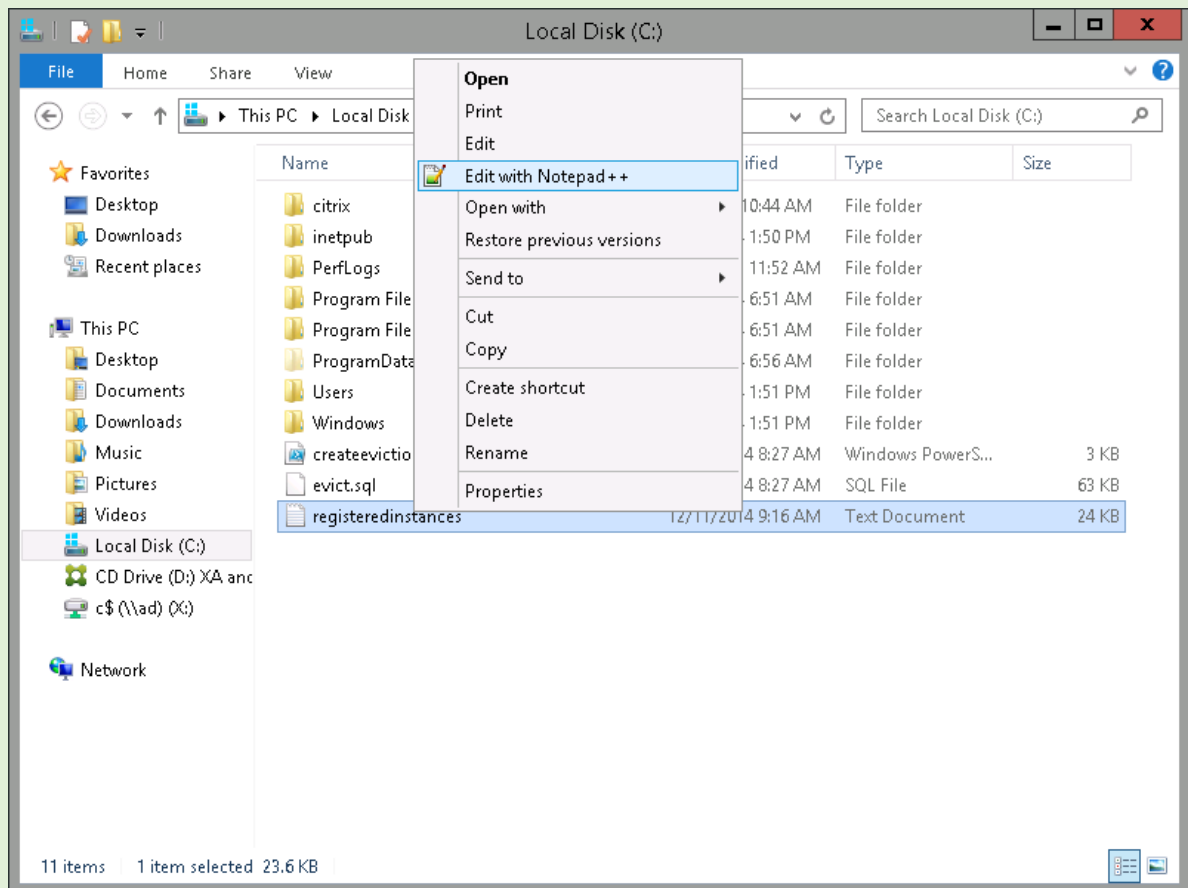
```
Administrator: Windows PowerShell

PS C:\> Get-ConfigRegisteredServiceInstance | select serviceaccount, serviceinstanceuid | sort-object -property serviceaccount > c:\registeredinstances.txt
PS C:\>
```

The screenshot shows a Windows PowerShell window with the following output:

```
ActiveSiteServices : <ControllerReaper, ControllerNameCacheRefresh, Licensing, BrokerReaper...>
AssociatedHypervisorConnectionUids : <1>
ControllerVersion : 7.6.0.5024
DNSName : DC1.training.lab
DesktopsRegistered : 5
LastActivityTime : 11/21/2014 7:11:08 AM
LastLicensingServerEvent : ServerOK
LastLicensingServerEventDetails : <Server: licenses.citrixvirtualclassroom.com, Port: 27000, ProductEdition: PLT, ProductId: XDI...>
LastLicensingServerEventTime : 11/19/2014 9:36:13 PM
LastStartTime : 11/20/2014 11:13:29 AM
LicensingGracePeriodReasons : <1>
LicensingGracePeriodLinesRemaining : <1>
LicensingGraceState : NotActive
LicensingServerState : OK
MachineName : TRAINING-DC1
MetadataMap : <1>
OSType : Windows 2012 R2
OSVersion : 6.2.9200.0
SID : S-1-5-21-3042957511-2888033547-949915721-2102
State : Active
UUID : 8987be44-e11b-4c3c-96fe-9aefca205a0b
Uid : 1
```

18. Browse the root of the local disk on **DC1** and open **registeredinstances.txt** using Notepad++:
- Note:** Just right click on **registeredinstances.txt** and select **Edit with Notepad++...**



19. Manually delete all reference to the service instances running on **DC1**:

Note: Make sure to also delete the column headings...

C:\registeredinstances.txt - Notepad++ [Administrator]

File Edit Search View Encoding Language Settings Macro Run Window ?

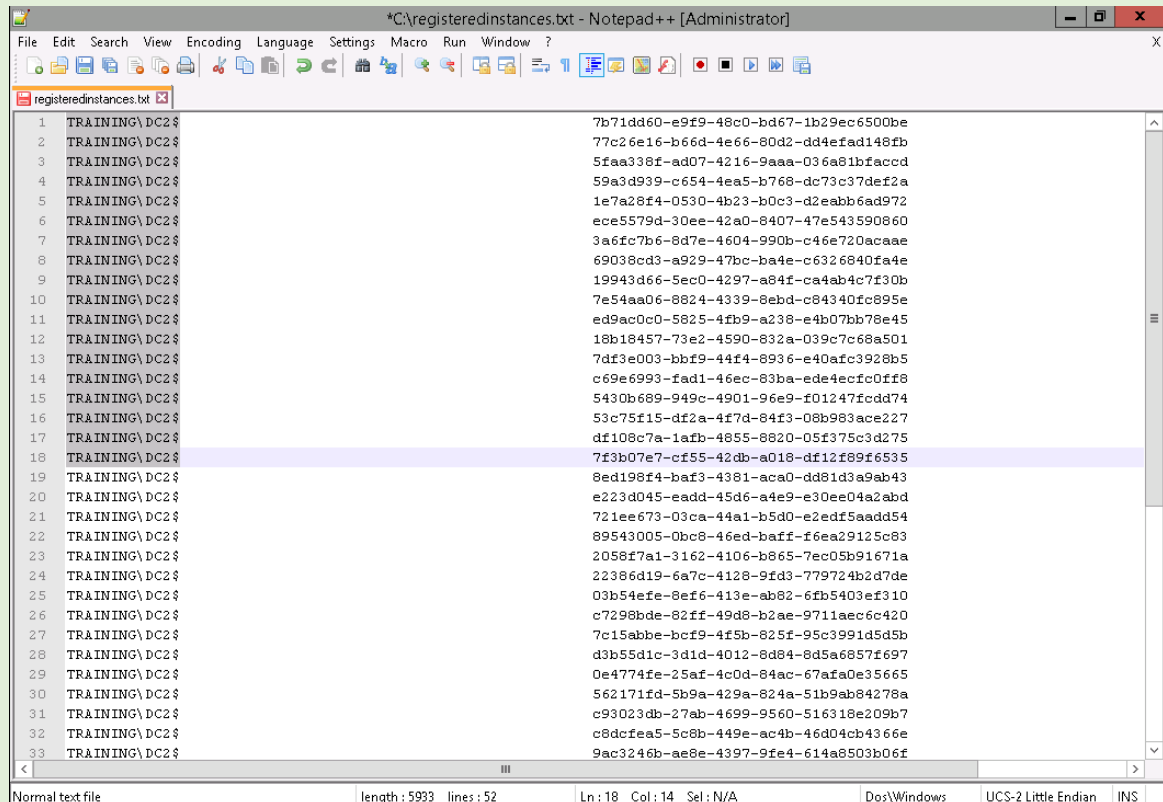
registeredinstances.txt

29	TRAINING\DC1\$	068db319-8f2a-4794-a54b-b262e98c35ba
30	TRAINING\DC1\$	c905c410-eed2-42f0-a4fa-b0aa97d40dc9
31	TRAINING\DC1\$	0d72e696-c14d-4747-8960-bd755f6e67de
32	TRAINING\DC1\$	3eaf6e74-c327-4800-b387-df74f5b8aed9
33	TRAINING\DC1\$	0f366523-7a16-47db-abb1-d88f1a9337b0
34	TRAINING\DC1\$	b1be95d0-d190-471e-8844-d6bbae958342
35	TRAINING\DC1\$	3b07c0db-3e65-4944-afe5-31ad12335550
36	TRAINING\DC1\$	930f4c4a-781e-4d39-9826-306056d5931b
37	TRAINING\DC1\$	5dc3137-1492-4a9e-a00e-31c1f3ed0118
38	TRAINING\DC1\$	5e0a294d-7a6c-4a30-b3c7-081ad1da2d48
39	TRAINING\DC1\$	f1bb2d893-f887-410f-8608-33326540a9ea
40	TRAINING\DC1\$	75602cd2-8968-4a2b-94f7-0ace7495e7fa
41	TRAINING\DC1\$	92e67b8c-7fc1-41d9-9dfe-1f28db5ad22e
42	TRAINING\DC1\$	01a3312c-dc8c-4b6c-9c90-1bd9137e6839
43	TRAINING\DC1\$	330daf12-996e-48cf-8cd9-1555fce9c377
44	TRAINING\DC1\$	7970972c-4a20-4e28-8def-0cf926de756e
45	TRAINING\DC1\$	680733b3-84ed-40a1-9533-21221b3fe993
46	TRAINING\DC1\$	678b6632-2749-4b97-a802-051ef100202e
47	TRAINING\DC1\$	1cbc3314-5e2e-4bb2-8408-40bf03315c03
48	TRAINING\DC1\$	9920ee5b-a6c3-4430-973c-3c57872120a8
49	TRAINING\DC1\$	ed7a575a-7d68-49f1-9037-4653ab8dbf88
50	TRAINING\DC1\$	a0a12a6f-237a-4a68-bb99-365b54446212
51	TRAINING\DC1\$	8298f27b-6bbc-4cd1-b401-4256e02c0c70
52	TRAINING\DC1\$	789574b0-421b-4749-8808-027d6c66fb4e
53	TRAINING\DC2\$	7b71dd60-e9f9-48c0-bd67-1b29ec6500be
54	TRAINING\DC2\$	77c26e16-b66d-4e66-80d2-dd4efad148fb
55	TRAINING\DC2\$	5faa338f-ad07-4216-9aaa-036a81bfaccd
56	TRAINING\DC2\$	59a3d939-c654-4ea5-b768-dc73c37def2a
57	TRAINING\DC2\$	1e7a28f4-0530-4b23-b0c3-d2eabb6ad972
58	TRAINING\DC2\$	ece5579d-30ee-42a0-8407-47e543590860
59	TRAINING\DC2\$	3a6fc7b6-8d7e-4604-990b-c46e720acaae
60	TRAINING\DC2\$	69038cd3-a929-47bc-ba4e-c6326840fa4e
61	TRAINING\DC2\$	19943d66-5ec0-4297-a84f-ca4ab4c7f30b
62	TRAINING\DC2\$	7e54aa06-8824-4339-8ebd-c84340fc895e

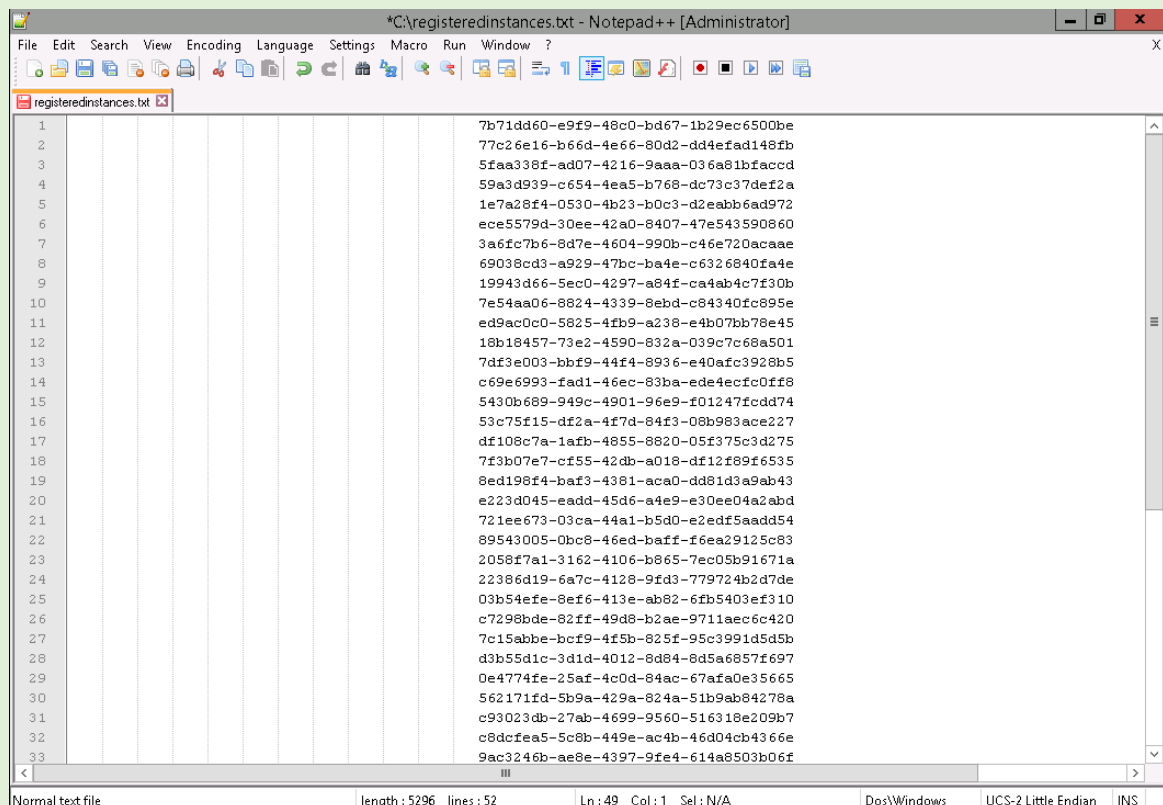
Normal text file length: 12106 lines: 104 Ln: 52 Col: 97 Sel: N/A Dos\Windows UCS-2 Little Endian INS

20. Using the **Alt** key (Hold down and highlight text) delete all **TrainingDC2\$** entries leaving just the service instance UID's behind:

Note: Notepad++ allows developers to manipulate text in a way that standard text editors like Notepad.exe can't and is very useful in this situation.



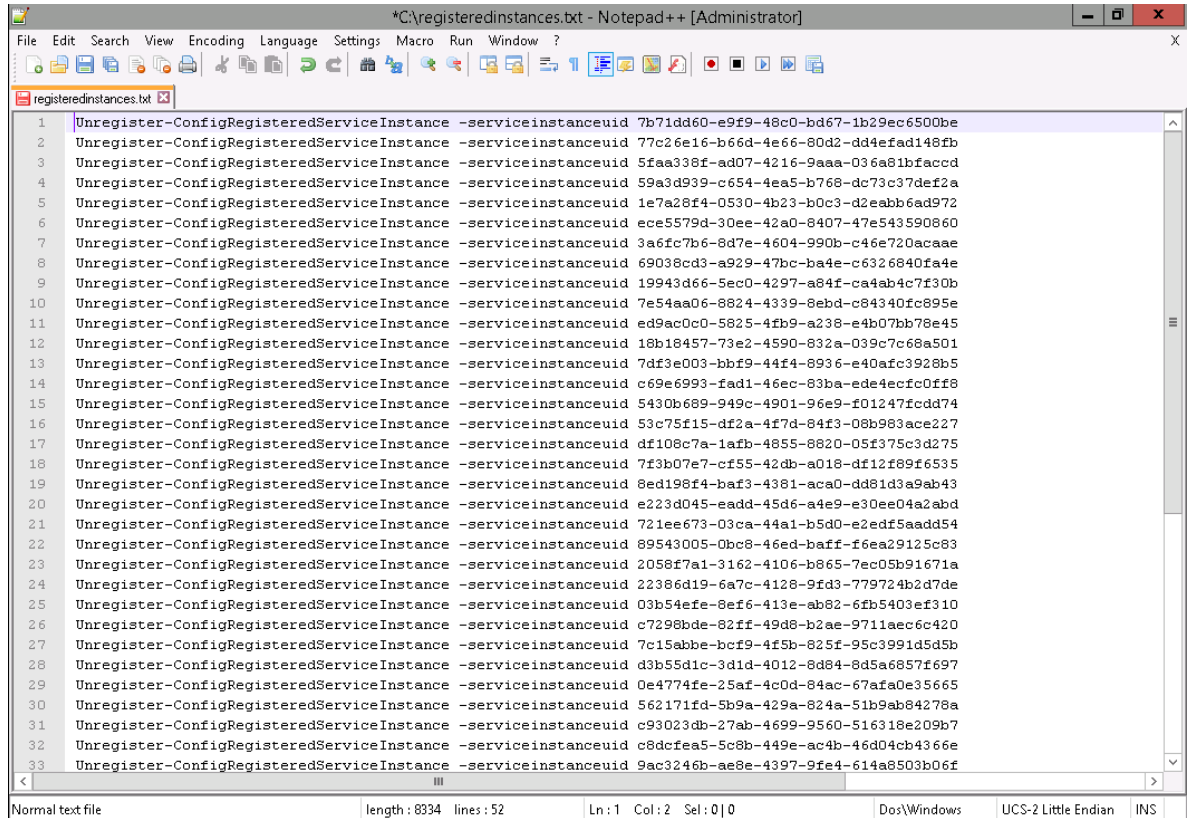
```
*C:\registeredinstances.txt - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Macro Run Window ?
registeredinstances.txt
1 TRAINING\DC2$ 7b71dd60-e9f9-48c0-bd67-1b29ec6500be
2 TRAINING\DC2$ 77c26e16-b66d-4e66-80d2-dd4efad148fb
3 TRAINING\DC2$ 5faa338f-ad07-4216-9aaa-036a81bfaccd
4 TRAINING\DC2$ 59a3d939-c654-4ea5-b768-dc73c37def2a
5 TRAINING\DC2$ 1e7a28f4-0530-4b23-b0c3-d2eabb6ad972
6 TRAINING\DC2$ ece5579d-30ee-42a0-8407-47e543590860
7 TRAINING\DC2$ 3a6fc7b6-8d7e-4604-990b-c46e720acaae
8 TRAINING\DC2$ 69038cd3-a929-47bc-ba4e-c6326840fa4e
9 TRAINING\DC2$ 19943d66-5ec0-4297-a84f-ca4ab4c7f30b
10 TRAINING\DC2$ 7e54aa06-8824-4339-8ebd-c84340fc895e
11 TRAINING\DC2$ ed9ac0c0-5825-4fb9-a238-e4b07bb78e45
12 TRAINING\DC2$ 18b18457-73e2-4590-832a-039c7c68a501
13 TRAINING\DC2$ 7df3e003-bbf9-44f4-8936-e40afc3928b5
14 TRAINING\DC2$ c69e6993-fad1-46ec-83ba-ede4ecfc0ff8
15 TRAINING\DC2$ 5430b689-949c-4901-96e9-f01247fcd74
16 TRAINING\DC2$ 53c75f15-df2a-4f7d-84f3-08b983ace227
17 TRAINING\DC2$ df108c7a-1afb-4855-8820-05f375c3d275
18 TRAINING\DC2$ 7f3b07e7-cf55-42db-a018-df12f89f6535
19 TRAINING\DC2$ 8ed198f4-baf3-4381-aca0-dd81d3a9ab43
20 TRAINING\DC2$ e223d045-eadd-45d6-a4e9-e30ee04a2abd
21 TRAINING\DC2$ 721ee673-03ca-44a1-b5d0-e2edf5aadd54
22 TRAINING\DC2$ 89543005-0bc8-46ed-baff-f6ea29125c83
23 TRAINING\DC2$ 2058f7a1-3162-4106-b865-7ec05b91671a
24 TRAINING\DC2$ 22386d19-6a7c-4128-9fd3-779724b2d7de
25 TRAINING\DC2$ 03b54efe-8ef6-413e-ab82-6fb5403ef310
26 TRAINING\DC2$ c7298bde-82ff-49d8-b2ae-9711aec6c420
27 TRAINING\DC2$ 7c15abbe-bcf9-4f5b-825f-95c3991d5d5b
28 TRAINING\DC2$ d3b55d1c-3d1d-4012-8d84-8d5a6857f697
29 TRAINING\DC2$ 0e4774fe-25af-4c0d-84ac-67afa0e35665
30 TRAINING\DC2$ 562171fd-5b9a-429a-824a-51b9ab84278a
31 TRAINING\DC2$ c93023db-27ab-4699-9560-516318e209b7
32 TRAINING\DC2$ c8dcfea5-5c8b-449e-ac4b-46d04cb4366e
33 TRAINING\DC2$ 9ac3246b-ae8e-4397-9fe4-614a8503b06f
Normal text file length: 5933 lines: 52 Ln: 18 Col: 14 Sel: N/A Dos\Windows UCS-2 Little Endian INS
```



```
*C:\registeredinstances.txt - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Macro Run Window ?
registeredinstances.txt
1 7b71dd60-e9f9-48c0-bd67-1b29ec6500be
2 77c26e16-b66d-4e66-80d2-dd4efad148fb
3 5faa338f-ad07-4216-9aaa-036a81bfaccd
4 59a3d939-c654-4ea5-b768-dc73c37def2a
5 1e7a28f4-0530-4b23-b0c3-d2eabb6ad972
6 ece5579d-30ee-42a0-8407-47e543590860
7 3a6fc7b6-8d7e-4604-990b-c46e720acaae
8 69038cd3-a929-47bc-ba4e-c6326840fa4e
9 19943d66-5ec0-4297-a84f-ca4ab4c7f30b
10 7e54aa06-8824-4339-8ebd-c84340fc895e
11 ed9ac0c0-5825-4fb9-a238-e4b07bb78e45
12 18b18457-73e2-4590-832a-039c7c68a501
13 7df3e003-bbf9-44f4-8936-e40afc3928b5
14 c69e6993-fad1-46ec-83ba-ede4ecfc0ff8
15 5430b689-949c-4901-96e9-f01247fcd74
16 53c75f15-df2a-4f7d-84f3-08b983ace227
17 df108c7a-1afb-4855-8820-05f375c3d275
18 7f3b07e7-cf55-42db-a018-df12f89f6535
19 8ed198f4-baf3-4381-aca0-dd81d3a9ab43
20 e223d045-eadd-45d6-a4e9-e30ee04a2abd
21 721ee673-03ca-44a1-b5d0-e2edf5aadd54
22 89543005-0bc8-46ed-baff-f6ea29125c83
23 2058f7a1-3162-4106-b865-7ec05b91671a
24 22386d19-6a7c-4128-9fd3-779724b2d7de
25 03b54efe-8ef6-413e-ab82-6fb5403ef310
26 c7298bde-82ff-49d8-b2ae-9711aec6c420
27 7c15abbe-bcf9-4f5b-825f-95c3991d5d5b
28 d3b55d1c-3d1d-4012-8d84-8d5a6857f697
29 0e4774fe-25af-4c0d-84ac-67afa0e35665
30 562171fd-5b9a-429a-824a-51b9ab84278a
31 c93023db-27ab-4699-9560-516318e209b7
32 c8dcfea5-5c8b-449e-ac4b-46d04cb4366e
33 9ac3246b-ae8e-4397-9fe4-614a8503b06f
Normal text file length: 5296 lines: 52 Ln: 49 Col: 1 Sel: N/A Dos\Windows UCS-2 Little Endian INS
```

21. Continue editing the registeredinstances.txt file placing **Unregister-ConfigRegisteredServiceInstance –serviceinstanceuid** before each service instance UID:

Once formatted correctly jump to the next step below.

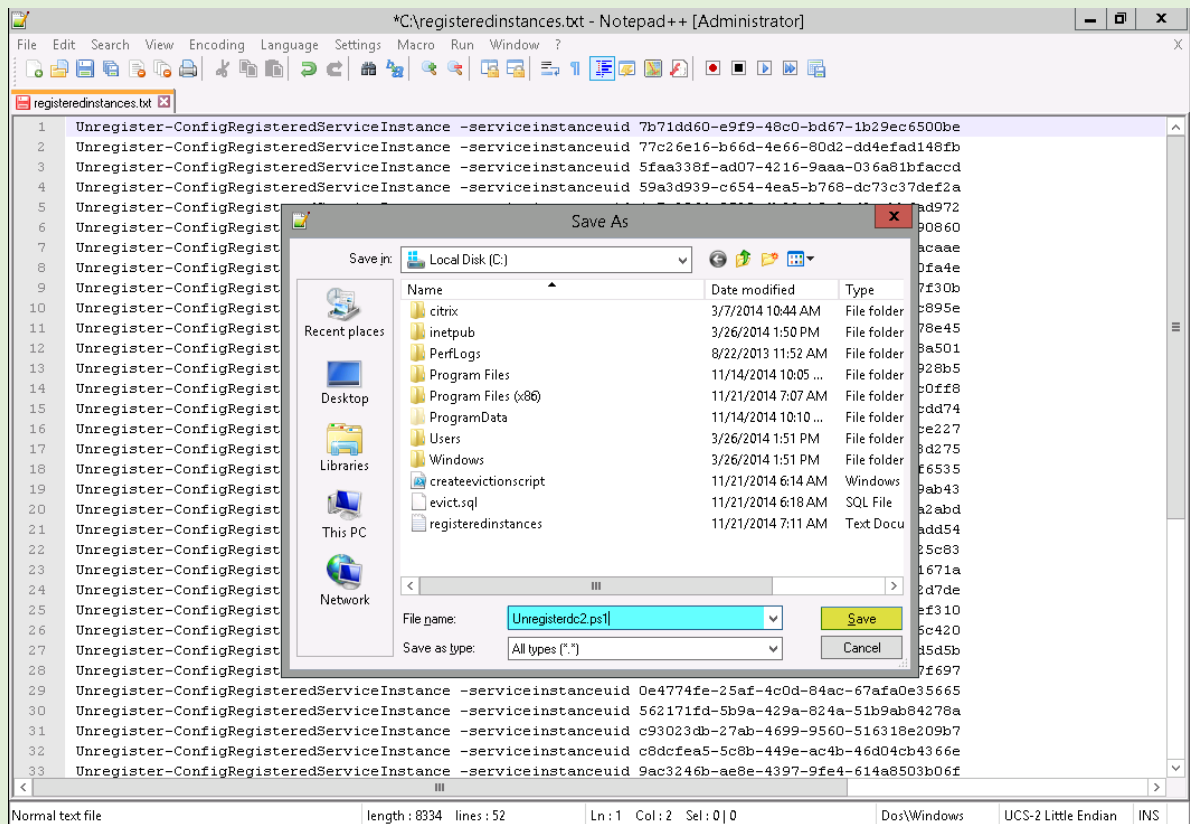


The screenshot shows a Notepad++ window titled "*C:\registeredinstances.txt - Notepad++ [Administrator]". The file contains 33 lines, each starting with "Unregister-ConfigRegisteredServiceInstance" followed by a hyphen and a long hexadecimal service instance UID. The status bar at the bottom indicates "Normal text file", "length: 8334", "lines: 32", "Ln: 1 Col: 2 Sel: 0 | 0", "Dos\Windows", "UCS-2 Little Endian", and "INS".

```
1 Unregister-ConfigRegisteredServiceInstance -serviceinstanceuid 7b71dd60-e9f9-48c0-bd67-1b29ec6500be
2 Unregister-ConfigRegisteredServiceInstance -serviceinstanceuid 77c26e16-b66d-4e66-80d2-dd4efad148fb
3 Unregister-ConfigRegisteredServiceInstance -serviceinstanceuid 5faa338f-ad07-4216-9aaa-036a81bfaccd
4 Unregister-ConfigRegisteredServiceInstance -serviceinstanceuid 59a3d939-c654-4ea5-b768-dc73c37def2a
5 Unregister-ConfigRegisteredServiceInstance -serviceinstanceuid 1e7a28f4-0530-4b23-b0c3-d2eabb6ad972
6 Unregister-ConfigRegisteredServiceInstance -serviceinstanceuid ece5579d-30ee-42a0-8407-47e543590860
7 Unregister-ConfigRegisteredServiceInstance -serviceinstanceuid 3a6fc7b6-8d7e-4604-990b-c46e720acaae
8 Unregister-ConfigRegisteredServiceInstance -serviceinstanceuid 69038cd3-a929-47bc-ba4e-c6326840fa4e
9 Unregister-ConfigRegisteredServiceInstance -serviceinstanceuid 19943d66-5ec0-4297-a84f-ca4ab4c7f30b
10 Unregister-ConfigRegisteredServiceInstance -serviceinstanceuid 7e54aa06-8824-4339-8ebd-c84340fc895e
11 Unregister-ConfigRegisteredServiceInstance -serviceinstanceuid ed9ac0c0-5825-4fb9-a238-e4b07bb78e45
12 Unregister-ConfigRegisteredServiceInstance -serviceinstanceuid 18b18457-73e2-4590-832a-039c7c68a501
13 Unregister-ConfigRegisteredServiceInstance -serviceinstanceuid 7df3e003-bbf9-44f4-8936-e40afc3928b5
14 Unregister-ConfigRegisteredServiceInstance -serviceinstanceuid c69e6993-fad1-46ec-83ba-ed4ecfc0ff8
15 Unregister-ConfigRegisteredServiceInstance -serviceinstanceuid 5430b689-949c-4901-96e9-f01247fcd74
16 Unregister-ConfigRegisteredServiceInstance -serviceinstanceuid 53c75f15-df2a-4f7d-84f3-08b983ace227
17 Unregister-ConfigRegisteredServiceInstance -serviceinstanceuid df108c7a-1afb-4855-8820-05f375c3d275
18 Unregister-ConfigRegisteredServiceInstance -serviceinstanceuid 7f3b07e7-cf55-42db-a018-df12f89f6535
19 Unregister-ConfigRegisteredServiceInstance -serviceinstanceuid 8ed198f4-baf3-4381-aca0-dd81d3a9ab43
20 Unregister-ConfigRegisteredServiceInstance -serviceinstanceuid e223d045-eadd-45d6-a4e9-e30ee04a2abd
21 Unregister-ConfigRegisteredServiceInstance -serviceinstanceuid 721ee673-03ca-44a1-b5d0-e2edf5aadd54
22 Unregister-ConfigRegisteredServiceInstance -serviceinstanceuid 89543005-0bc8-46ed-baff-f6ea29125c83
23 Unregister-ConfigRegisteredServiceInstance -serviceinstanceuid 2058f7a1-3162-4106-b865-7ec05b91671a
24 Unregister-ConfigRegisteredServiceInstance -serviceinstanceuid 22386d19-6a7c-4128-9fd3-779724b2d7de
25 Unregister-ConfigRegisteredServiceInstance -serviceinstanceuid 03b54efe-8ef6-413e-ab82-6fb5403ef310
26 Unregister-ConfigRegisteredServiceInstance -serviceinstanceuid c7298bde-82ff-49d8-b2ae-9711a6c6c420
27 Unregister-ConfigRegisteredServiceInstance -serviceinstanceuid 7c15abbe-bcf9-4f5b-825f-95c3991d5d5b
28 Unregister-ConfigRegisteredServiceInstance -serviceinstanceuid d3b55d1c-3d1d-4012-8d84-8d5a6857f697
29 Unregister-ConfigRegisteredServiceInstance -serviceinstanceuid 0e4774fe-25af-4c0d-84ac-67afa0e35665
30 Unregister-ConfigRegisteredServiceInstance -serviceinstanceuid 562171fd-5b9a-429a-824a-51b9ab84278a
31 Unregister-ConfigRegisteredServiceInstance -serviceinstanceuid c93023db-27ab-4699-9560-516318e209b7
32 Unregister-ConfigRegisteredServiceInstance -serviceinstanceuid c8dcfea5-5c8b-449e-ac4b-46d04cb4366e
33 Unregister-ConfigRegisteredServiceInstance -serviceinstanceuid 9ac3246b-ae8e-4397-9fe4-614a8503b06f
```


22. Click File → Save and save the text as **Unregisterdc2.ps1** on the root of C:\ on **DC1**:

Note: Ensure “All Types (*.*)” is selected under “Save as Type”



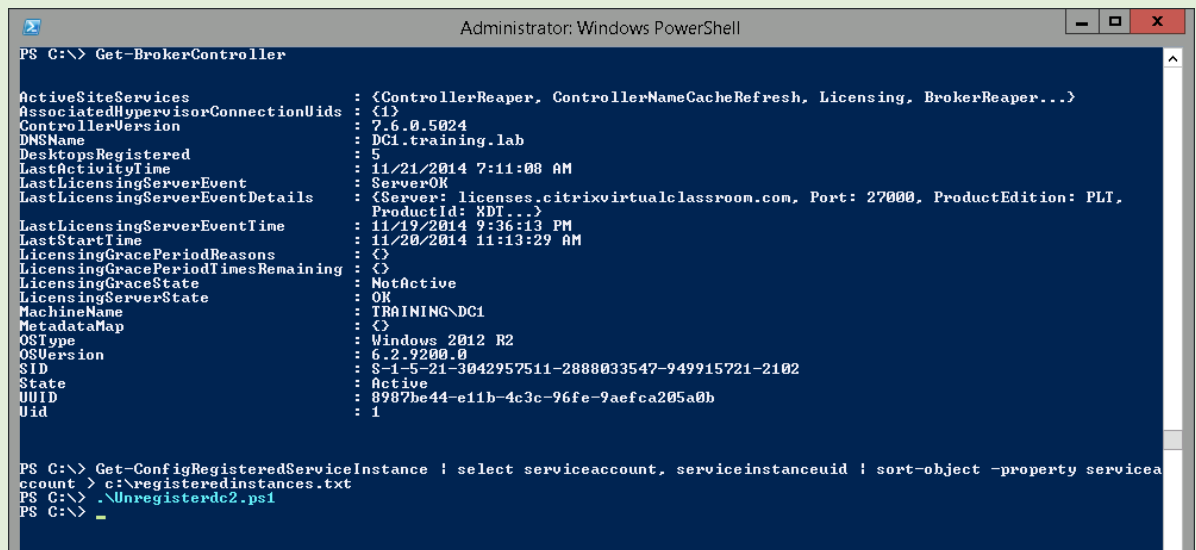
Note: A previously created and tested **Unregisterdc2.ps1** script can also be found at **\\ad\software\scripts\Exercise 6** for convenience and validation.

23. Close Notepad++ and switch back to the open instance of PoSH (on **DC1**).

24. Run the newly created **unregisterdc2.ps1** script from the root of PoSH: (To get to the root just type **cd** & hit return)

C:\> .\unregisterdc2.ps1

Reminder: You should be running this script from DC1

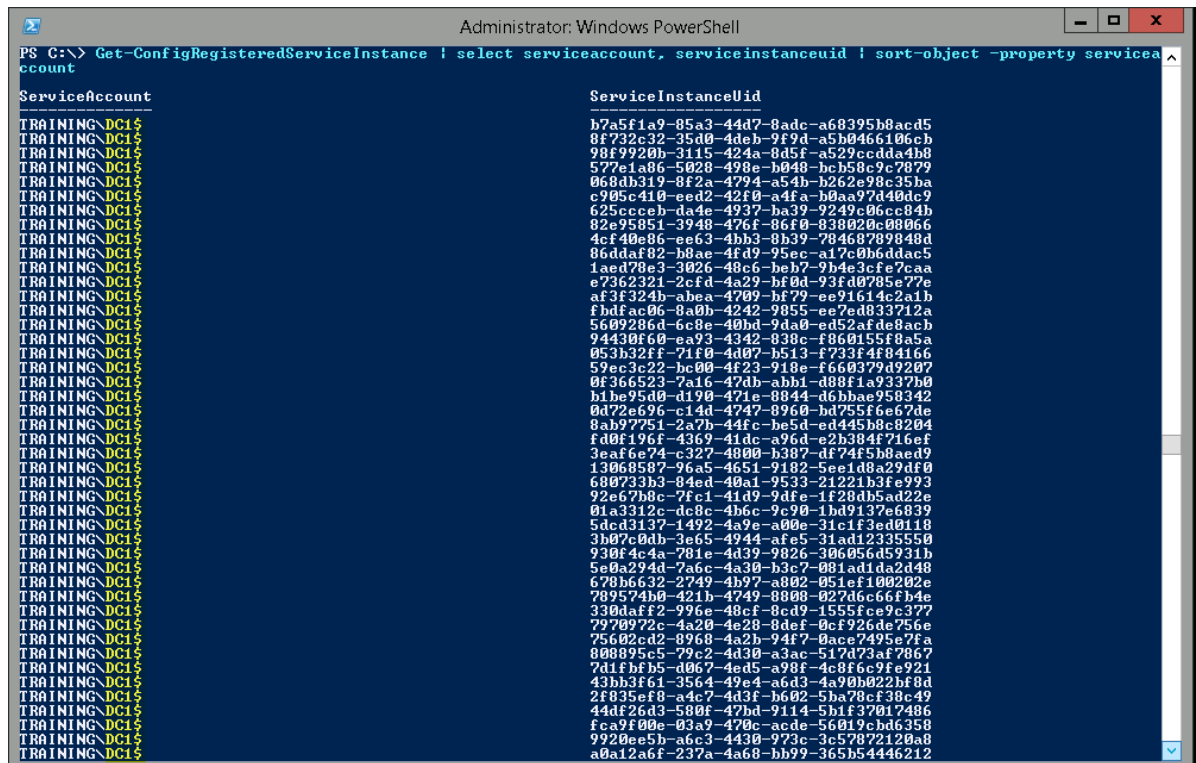


```
Administrator: Windows PowerShell
PS C:\> Get-BrokerController

ActiveSiteServices       : (ControllerReaper, ControllerNameCacheRefresh, Licensing, BrokerReaper...)
AssociatedHypervisorConnectionUids : (1)
ControllerVersion        : 7.6.0.5024
DNSName                  : DC1.training.lab
DesktopsRegistered       : 5
LastActivityTime         : 11/21/2014 7:11:00 AM
LastLicensingServerEvent : ServerOK
LastLicensingServerEventDetails : (Server: licenses.citrixvirtualclassroom.com, Port: 27000, ProductEdition: PLT, ProductId: KDT...)
LastLicensingServerEventTime : 11/19/2014 9:36:13 PM
LastStartTime            : 11/20/2014 11:13:29 AM
LicensingGracePeriodReasons : (0)
LicensingGracePeriodTimesRemaining : (0)
LicensingGraceState      : NotActive
LicensingServerState     : OK
MachineName              : TRAINING\DC1
MetadataNap              : (0)
OSType                   : Windows 2012 R2
OSVersion                : 6.2.9200.0
SID                      : S-1-5-21-3042957511-2888033547-949915721-2102
State                    : Active
UUID                     : 8987be44-e11b-4c3c-96fe-9aefca205a0b
Uid                      : 1


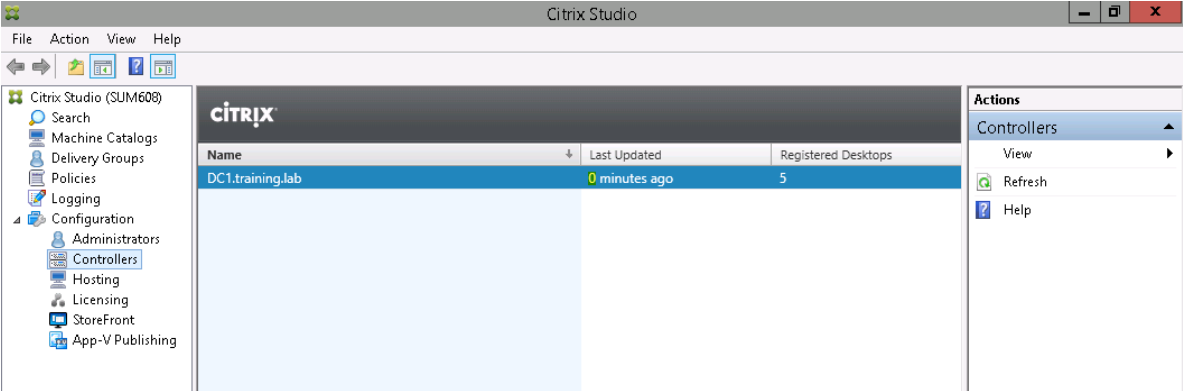
PS C:\> Get-ConfigRegisteredServiceInstance | select serviceaccount, serviceinstanceuid | sort-object -property serviceaccount > c:\registeredinstances.txt
PS C:\> .\Unregisterdc2.ps1
PS C:\> _
```

25. Type **Get-ConfigRegisteredServiceInstance | select serviceaccount, serviceinstanceuid | sort-object -property serviceaccount** and hit return to verify that only service instances running on **DC1** remain registered with the Site Configuration service:



```
Administrator: Windows PowerShell
PS C:\> Get-ConfigRegisteredServiceInstance | select serviceaccount, serviceinstanceuid | sort-object -property serviceaccount

ServiceAccount      ServiceInstanceUid
-----
TRAINING\DC1$      b7a5f1a9-85a3-44d7-8adc-a68395b8acd5
TRAINING\DC1$      8f732c32-35d0-4deb-9f9d-a5b0466106cb
TRAINING\DC1$      98f9920b-3115-424a-0d5f-a529ccdda4b8
TRAINING\DC1$      577e1a86-5028-498e-b048-bc5b58c9c7879
TRAINING\DC1$      068db319-8f2a-4794-a54b-b262e98c35ba
TRAINING\DC1$      c905c410-eed2-42f0-a4fa-b0aa97d40dc9
TRAINING\DC1$      625ccceb-da4e-4937-ba39-9249c06cc84b
TRAINING\DC1$      82e95851-3948-476f-86f0-930020c00066
TRAINING\DC1$      4cf40e86-ea63-4bb3-8b39-78468789948d
TRAINING\DC1$      86ddaf82-b8ae-4fd9-95ec-a17c0b6ddac5
TRAINING\DC1$      1aed78e3-3026-48c6-beb7-9b4e3cfe7caa
TRAINING\DC1$      e7362321-2cfd-4a29-bf0d-93fd0785e77e
TRAINING\DC1$      af3f324b-abea-4709-bf79-ee91614c2a1b
TRAINING\DC1$      fbdfac06-8a0b-4242-9855-ee7ed833712a
TRAINING\DC1$      5609286d-6c8e-40bd-9da0-ed52afde8ach
TRAINING\DC1$      94430ff60-ea93-4342-838c-f860155f8a5a
TRAINING\DC1$      053b32ff-71f0-4a07-b513-f733f4f84166
TRAINING\DC1$      59ec3c22-bc00-4f23-910e-f660379d9207
TRAINING\DC1$      0f366523-7a16-47db-abbi-488f1a9337b0
TRAINING\DC1$      b1be95d0-d190-471e-8844-d6bbae958342
TRAINING\DC1$      0d72e696-c14d-4747-8960-bd755f6e67de
TRAINING\DC1$      8ab97751-2a7b-44fc-be5d-ed445b8c8204
TRAINING\DC1$      fd0f196f-4369-41dc-a96d-e2b384f716ef
TRAINING\DC1$      3eaf6e74-c327-4800-b387-df74f5b8aed9
TRAINING\DC1$      13060587-76a5-4651-7182-5ee1d0a29df0
TRAINING\DC1$      680733b3-84ed-40a1-9533-21221b3fe993
TRAINING\DC1$      92e6708c-7fc1-41d9-9dfe-1f20db5ad22e
TRAINING\DC1$      01a3312c-dc8c-4b6c-9c90-1bd9137e6839
TRAINING\DC1$      5dc3137-1492-4a9e-a00e-31c1f3ed0118
TRAINING\DC1$      3b07c0db-3e65-4944-afe5-31ad12335550
TRAINING\DC1$      930f4c4a-781e-4d39-9826-306056d5931b
TRAINING\DC1$      5e0a294d-7a6c-4a30-b3c7-081ad1da2d48
TRAINING\DC1$      678b6632-2749-4b97-a802-051ef100202e
TRAINING\DC1$      789574b0-421b-7749-8808-027d6c66fb4e
TRAINING\DC1$      330daff2-996e-48cf-8cd9-1555f9c9c377
TRAINING\DC1$      2970972c-4a28-4e20-8def-0cf926de756e
TRAINING\DC1$      75602cd2-0960-4a2b-94f7-0acc7495e71a
TRAINING\DC1$      808895c5-79c2-4d30-a3ac-517d73af7867
TRAINING\DC1$      7d1fbfb5-d067-4ed5-a98f-4c8f6c9fe921
TRAINING\DC1$      43bb3f61-3564-49e4-a6d3-4a90b022bf8d
TRAINING\DC1$      2f835ef8-a4c7-4d3f-b602-5ba78cf38c49
TRAINING\DC1$      44df26d3-580f-47bd-9114-5b1f37017486
TRAINING\DC1$      fca9f00e-03a9-470c-acde-56019cbd6358
TRAINING\DC1$      9920ee5b-a6c3-4430-973c-3c57872120a8
TRAINING\DC1$      a0a12a6f-237a-4a68-bb99-365b54446212
```

26.	<p>If not already running, launch Citrix Studio from the taskbar icon on DC1:</p> 
27.	<p>Click on the Controllers node:</p>  <p>Observation: Only DC1.training.lab is now listed as an active Site Controller.</p> <p>Note: As mentioned previously, fully functioning Controllers should always show a last update state of 0 minutes as they heartbeat by default with the Site DB every 20 seconds and will timeout after 40 seconds.</p>
28.	<p>Congratulations, you have finished this lab exercise.</p>

Exercise Summary

Takeaways from this exercise:

- Although Citrix Studio provides the ability to gracefully remove Site Controllers this may not always be possible especially if the Controller state is returning as failed or unknown (Orphaned Controller, misbehaving Controller). In this situation it is important to understand how to approach this task manually using the SQL Service evict scripts.
- For more information re: How to manually remove Controllers from an XD site using PoSH take a look at the following XD Tipster blog:
 - <http://blogs.citrix.com/2013/10/22/xd-tipster-removing-controllers-from-an-xd-site-using-posh/>
- For more information and XenDesktop related tips follow **@XDtipster** and **@XDInformer**

Revision	Change Descriptions	Updated By	Date
1.0	Original Version	Kimberly Ferrie & Mick Glover	January 2015

About Citrix

Citrix (NASDAQ:CTXS) is a cloud company that enables mobile workstyles—empowering people to work and collaborate from anywhere, securely accessing apps and data on any of the latest devices, as easily as they would in their own office. Citrix solutions help IT and service providers build clouds, leveraging virtualization and networking technologies to deliver high-performance, elastic and cost-effective cloud services. With market-leading cloud solutions for mobility, desktop virtualization, networking, cloud platforms, collaboration and data sharing, Citrix helps organizations of all sizes achieve the speed and agility necessary to succeed in a mobile and dynamic world. Citrix products are in use at more than 330,000 organizations and by over 100 million users globally. Annual revenue in 2012 was \$2.59 billion. Learn more at www.citrix.com.