**CloudPlatform**



Self Service Desktops

Preparing a template with sysprep

  
www.citrix.com

Introduction

This document describes how to install the Citrix Self Service Desktops solution. The solution builds on original work by Christian Ferber. Before reading this document, we recommend that you first read the Citrix Self Service Desktops Installation guide [1]

Objectives

This document describes how to prepare a sysprep enabled template in CloudPlatform for the Citrix Self Service Desktops solution. A companion document [2] describes how to prepare an ISO and/or CloudPlatform template if you wish to stream desktops into CloudPlatform from Citrix Provisioning Services.

Overview

A major step for this integration is the preparation of a virtual desktop template in CloudPlatform. In this example Windows 7 Enterprise 64bit has been used, but the general process should be applicable to other client operating systems as well.

1. **Install Windows 7 Enterprise 64 bit VM**

Create a base Windows 7 instance in CloudPlatform. This could be done by any available method in CloudPlatform (e.g. iso installation, import and existing template…)

1. **Integrate Hypervisor tools**

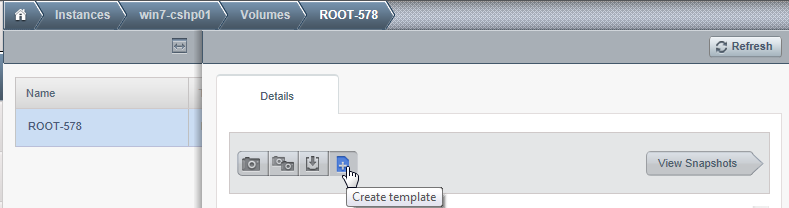
Install the relevant hypervisor tools (XenTools in this example)

1. **Install XenDesktop VDA**  
   This part is not covered in detail in this document. Please refer to XenDesktop documentation on how to install VDA agent on a Windows 7 desktop. Please ensure you use the Advanced Installation mode for the VDA and specify the XenDesktop server address during VDA installation.
2. **Configure Sysprep and domain join**

Please refer to the section below entitled Prepare Image for Sysprep and apply all relevant changes to the Windows 7 instance. Note that the PowerShell script for changing computer name enables self-service desktops to support user initiated revert of the virtual machine to its original state (through CloudPlatform UI).

1. **Create a template from this instance**

When sysprep has been executed according to Appendix A (after running C:\windows\system32\sysprep\sysprep.exe /oobe/generalize /shutdown) the instance is shutdown automatically and stays in stopped state. Now create a template from the volume related to this instance.



1. **Update Citrix Self Service Desktops configuration**

Once the new template has been created, then all that is required is to edit the Citrix Self Service Desktops configuration (see Installation Guide [1]) to insert the new template Id into a Desktop Offering.

Prepare Image for Sysprep

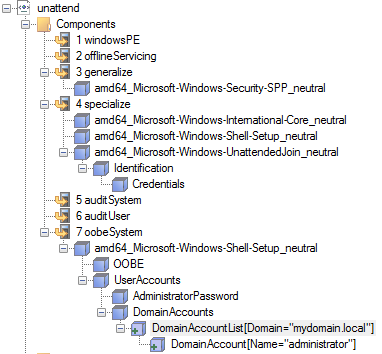
Windows VMs are typically integrated in the domain when used. In classical deployments with CloudPlatform this has to be done manually after the deployment task. This section describes how virtual machines can be added to the domain automatically after a VM instance has been deployed; using Windows 7 Enterprise 64bit as an example.

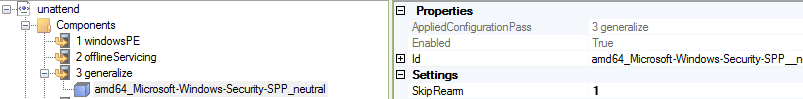
Integrating Sysprep in a Windows VM

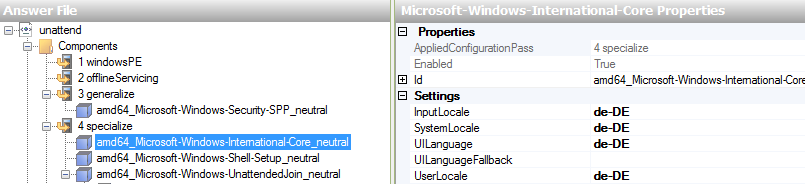
The prerequisite for this task is having a Windows VM instance which is prepared with the customer required applications and modifications. To generalize this virtual machine, Windows sysprep is used. The procedure how to do this is documented in the CloudPlatform Admin Guide (<http://support.citrix.com/article/CTX133615>) page 97 “Sysprep for Windows Server 2008 R2).

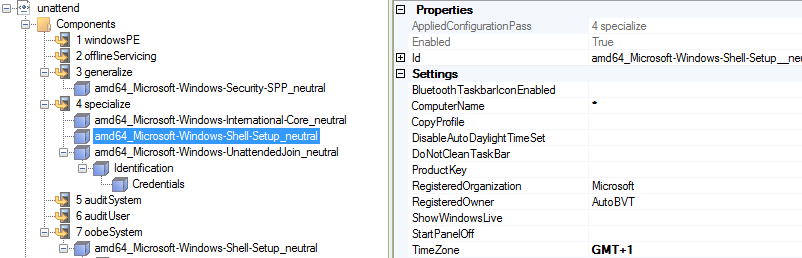
Follow this guideline until point 9. Don’t execute the sysprep.exe command yet.

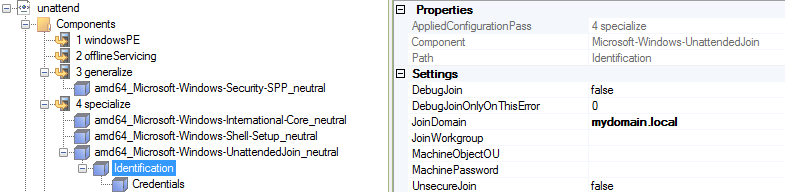
To enable domain join for sysprep you have to add additional parameters to your unattend.xml file using Windows System Image Manager. See an example of the parameters required below:

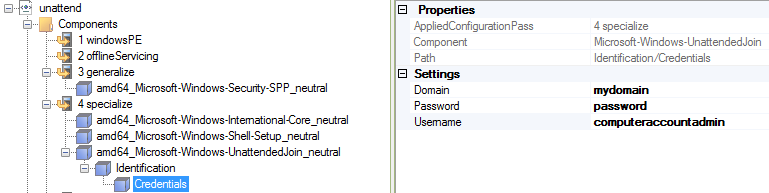


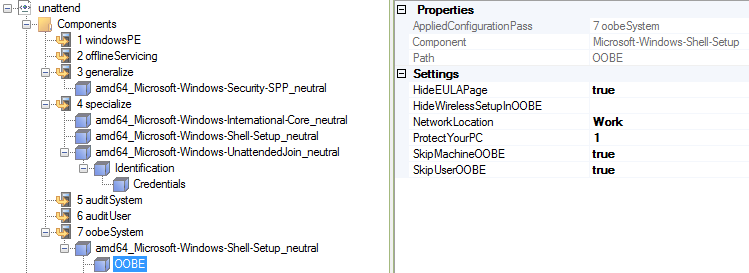


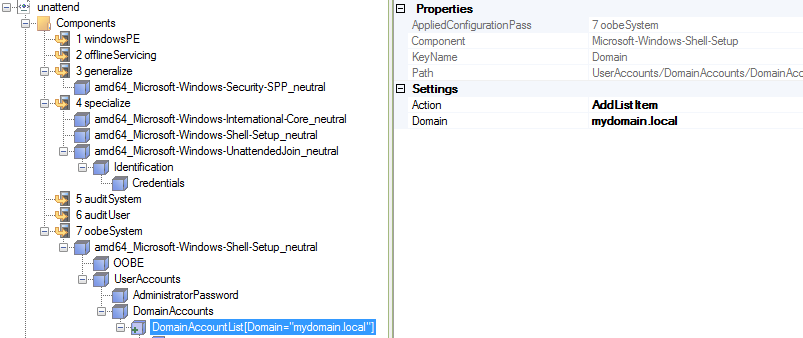


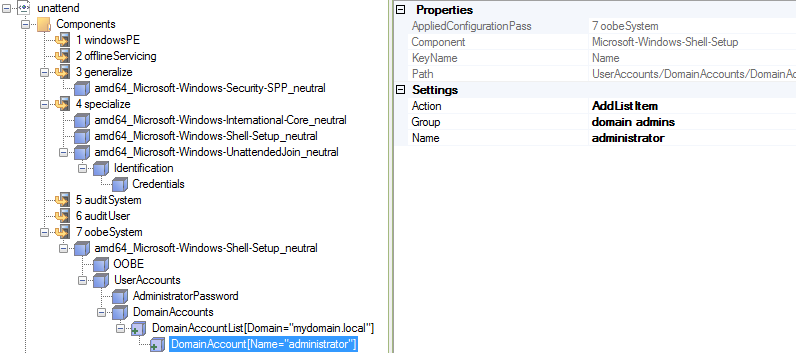












Appendix B includes an example unattend.xml file as well. If you don’t want to run through Windows System Image Manager you could use this file as an example, modify the relevant parts and copy it into your deployment.

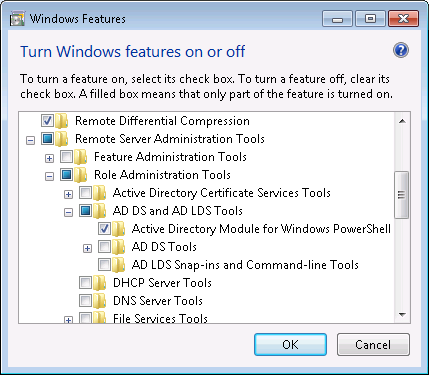
PowerShell script for changing computer name

This script uses an Active Directory PowerShell plugin in addition to WMI. It automatically overwrites the object in AD if the computer name already exists.

1.) Install Remote-Server-Administration-Tools (RSAT) on Windows 7.  
Download from: <http://www.microsoft.com/en-us/download/details.aspx?id=7887>

*Note: This requires Service Pack 1 to be installed on Windows 7. If you are running a Server OS these tools are part of the OS install and required components can be activated by adding features.*

2.) Activate “Active Directory Module for Windows PowerShell” (Windows 7)  
Control Panel 🡪 Programs 🡪 Turn Windows Features on and off



3.) Create directory c:\scripts in the VM.

4.) Create file c:\scripts\change-computername.ps1

5.) Edit change-computername.ps1 and insert below code

Import-Module ActiveDirectory

$username = 'mydomain\computeraccountadmin’

$password = 'password'

$cred = New-Object System.Management.Automation.PSCredential -ArgumentList @($username,(ConvertTo-SecureString -String $password -AsPlainText -Force))

$ErrorActionPreference = "SilentlyContinue"

$DHCPServers = Get-WmiObject Win32\_NetworkAdapterConfiguration -Filter "DHCPEnabled=TRUE AND DHCPServer IS NOT NULL" -Property DHCPServer

ForEach ($DHCPServer in $DHCPServers.DHCPServer) {

$URL = "http://$DHCPServer/latest/meta-data/local-hostname"

$metadatarequest=[System.Net.WebRequest]::Create($URL)

$resp=$metadatarequest.GetResponse()

$reqstream=$resp.GetResponseStream()

$sr=new-object System.IO.StreamReader $reqstream

$newhostname=$sr.ReadToEnd()

Write-Host $newhostname

$compaccount = Get-ADcomputer $newhostname -Credential $cred

Write-Host $compaccount

if ($compaccount)

{

Write-Host "Computeraccount exists"

Remove-ADComputer -identity $newhostname -Credential $cred -confirm:$false

}

else

{

Write-Host "Computeraccount does not exist"

}

Get-WmiObject Win32\_ComputerSystem

If ($newhostname.Length -gt 0) {$(Get-WmiObject Win32\_ComputerSystem).Rename($newhostname,$password,$username)}

}

6.) Replace the red marked sections by custom values from your deployment  
‘mydomain\computeraccountadmin’ = user account that has permission to create and change computer accounts in active directory

‘password’ = Password of above mentioned account

7.) Save the file

Configuring PowerShell script for autorun

1.) Create the file SetupComplete.cmd under c:\Windows\Setup\scripts   
*(you might need to create the sub dir “scripts” as it likely is not present)*

2.) Edit SetupComplete.cmd and insert below code

powershell Set-ExecutionPolicy RemoteSigned

powershell c:\scripts\change-computername.ps1

shutdown /r

3.) Save the file

Running Sysprep

When having done all preparation steps you are ready to run sysprep as explained in section 9, page 100 of the admin guide (the point we stopped above).

C:\windows\system32\sysprep\sysprep.exe /oobe/generalize /shutdown

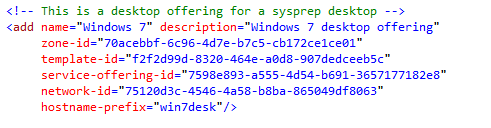
Now you can create a template from the VM. After deployment the new instance will use the computer name in Windows which was defined in CloudPlatform instance wizard. In addition the Windows VM will be part of the domain specified in the unattend.xml file.

Defining Self Service Desktops Configuration

The following is a sample desktop offering for a sysprepped desktop. The desktop offering has a name and description; the name must be unique.

The zone, network and service offering ids define the virtual machine size and deployment options. The template id should be the id of the sysprepped template you created above.

The hostname prefix must be unique amongst desktop offerings, and defines the base name of the desktops that will be created (e.g. this configuration will result in desktops with names win7desk0001, win7desk0002 etc.)



Appendix A

If you want to get up and running quickly with a default configuration, then please copy and edit the working unattend.xml file from here (for Windows 7 64bit). The local Administrator password is “Pa$$w0rd”. Simply copy the code into an unattend.xml file and store the file under c:\windows\system32\sysprep\. Make sure you adapt at least the red marked sections to match your deployment.

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <unattend xmlns="urn:schemas-microsoft-com:unattend">  <settings pass="specialize">  <component name="Microsoft-Windows-International-Core" processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35" language="neutral" versionScope="nonSxS" xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  <InputLocale>de-DE</InputLocale>  <SystemLocale>de-DE</SystemLocale>  <UILanguage>de-DE</UILanguage>  <UserLocale>de-DE</UserLocale>  </component>  <component name="Microsoft-Windows-Shell-Setup" processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35" language="neutral" versionScope="nonSxS" xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  <ComputerName>\*</ComputerName>  <TimeZone>GMT+1</TimeZone>  </component>  <component name="Microsoft-Windows-UnattendedJoin" processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35" language="neutral" versionScope="nonSxS" xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  <Identification>  <Credentials>  <Domain>mydomain</Domain>  <Password>password</Password>  <Username>comuteraccountadmin</Username>  </Credentials>  <JoinDomain>mydomain.local</JoinDomain>  </Identification>  </component>  </settings>  <settings pass="oobeSystem">  <component name="Microsoft-Windows-Shell-Setup" processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35" language="neutral" versionScope="nonSxS" xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  <OOBE>  <HideEULAPage>true</HideEULAPage>  <NetworkLocation>Work</NetworkLocation>  <ProtectYourPC>1</ProtectYourPC>  <SkipUserOOBE>true</SkipUserOOBE>  <SkipMachineOOBE>true</SkipMachineOOBE>  </OOBE>  <UserAccounts>  <AdministratorPassword>  <Value>UABhACQAJAB3ADAAcgBkAEEAZABtAGkAbgBpAHMAdAByAGEAdABvAHIAUABhAHMAcwB3AG8AcgBkAA==</Value>  <PlainText>false</PlainText>  </AdministratorPassword>  <DomainAccounts>  <DomainAccountList wcm:action="add">  <Domain>space.local</Domain>  <DomainAccount wcm:action="add">  <Group>domain admins</Group>  <Name>administrator</Name>  </DomainAccount>  </DomainAccountList>  </DomainAccounts>  </UserAccounts>  <TimeZone>GMT+1</TimeZone>  </component>  </settings>  <settings pass="generalize">  <component name="Microsoft-Windows-Security-SPP" processorArchitecture="amd64" publicKeyToken="31bf3856ad364e35" language="neutral" versionScope="nonSxS" xmlns:wcm="http://schemas.microsoft.com/WMIConfig/2002/State" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  <SkipRearm>1</SkipRearm>  </component>  </settings>  <cpi:offlineImage cpi:source="wim:c:/unattend/install.wim#Windows 7 ENTERPRISE" xmlns:cpi="urn:schemas-microsoft-com:cpi" />  </unattend> |

References

1. Citrix Self Service Desktops Installation Guide
2. Citrix Self Service Desktops with PVS