Chapter 4. Default User Profile Customization

In the previous chapter, we discussed the technique of collecting a reference build for Windows in a virtual machine on Hyper-V or physical box with boot media on a USB stick. In this chapter, we'll cover how to customize this base image as part of our process. Not from an application build perspective, but to make the image branded, remove or add features, remove the **out-of-box experience** (**OOBE**), and so forth. Now some general things one might want done to their image would be removing the games, setting the Internet Explorer default settings, customizing the background screen, and other branding components, removing the ability to access things in the UI by default, and so on, as well as customizing the image for a kiosk, tablet, cash register, ATM, exec's laptop, VDI image, and so on.

In this chapter, we'll go through the following topics:

- How to customize the Windows image
- Windows System Image Manager and `Unattend.xml`
- The differences between Windows 7 and 8 in UI customizations

Customizing the image

You may customize the image before running your reference task sequence. Customizations are done in many ways; some are script-based, some group policies, some are manual efforts, some are done in a tool known as **Windows System Image Manager** (**WSIM**), and others can be done with PowerShell scripts.

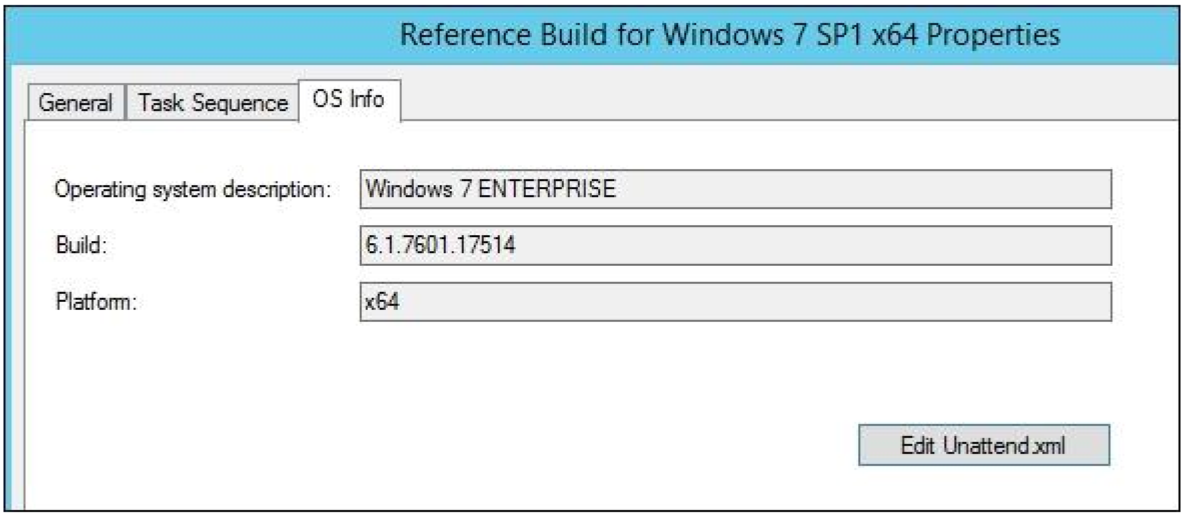
Checking out the customization documentation

The general caveats and principles of image customization are documented for Windows at this KB at http://support.microsoft.com/kb/973289 . The essential gist of this is that to customize the user profile of a future user of Windows 7 or 2008 R2, you must customize the default local user profile. When this is done, the settings in the default profile will become the settings of new user profiles on the system. Note that this is the only officially supported method of customizing the default user profile in Windows 7.

For Windows 8.x, things change somewhat. The steps are documented at http://technet.microsoft.com/en-us/library/hh825135.aspx and involve, generally speaking, the same steps as for Windows 7. However, a new functionality is provided to customize the Start menu, documented at http://technet.microsoft.com/en-us/library/jj134269.aspx .

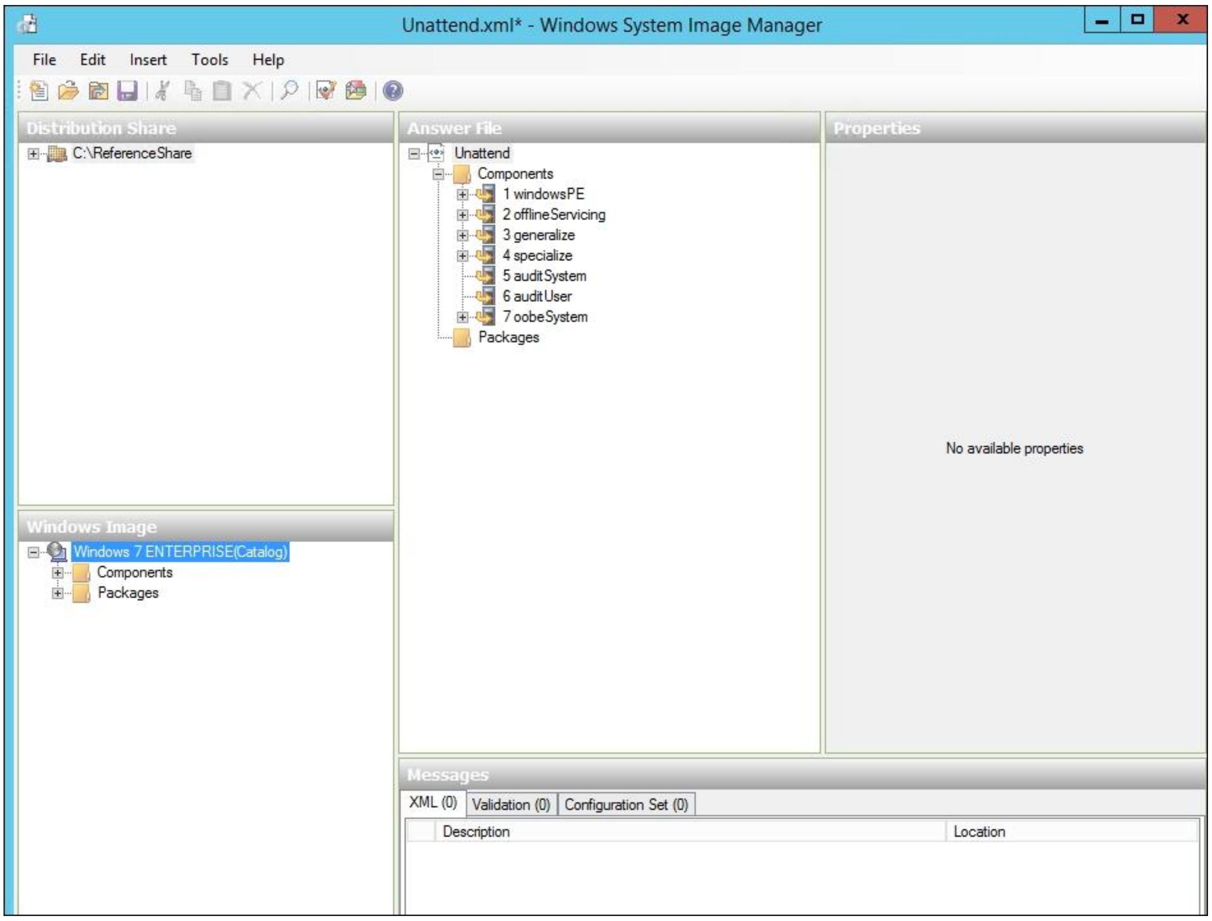
Accessing Windows System Information Manager

An easy way to use this customization, as mentioned previously, is **WSIM**. It's quite easy to access; simply right-click on your task sequence and select the **OS Info** tab, and click on **Edit Unattend.xml**:



Operating system description: Windows 7 ENTERPRISE
Build: 6.1.7601.17514
Platform: x64

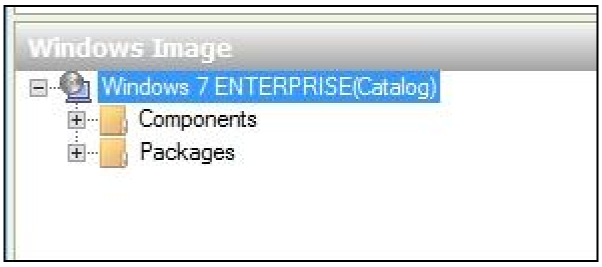
Then you'll be rewarded with a modified view of the typical MMC console layout for `Unattend.xml` editing, as shown in the following:



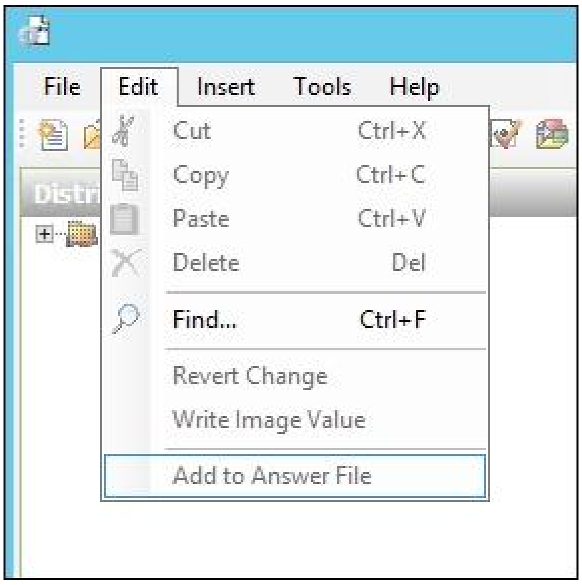
Adding games to our Enterprise image

This interface is a bit intimidating at first, and somewhat unclear about where things should be edited, as they will sometimes appear to be applicable in multiple locations of the XML. For those who have manually edited the `Unattend.xml` files in the past, this will make little sense perhaps. It may be best to learn by doing, so let's add games to our Windows 7 Enterprise image by performing the following exercise:

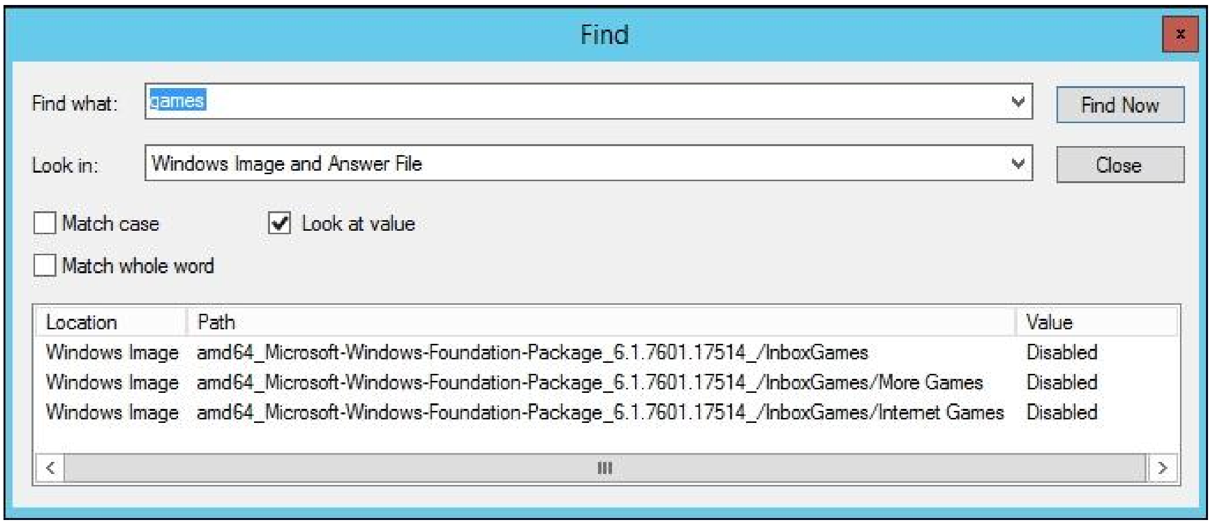
1. Ensure the focus is on the **Windows Image** area of the MMC, as shown in the following image:



2. Click on **Find...** in the **Edit** menu:



3. Now, simply type `games` here and observe that there are three responses to the query:



Find

Find what: games

Look in: Windows Image and Answer File

Find Now

Close

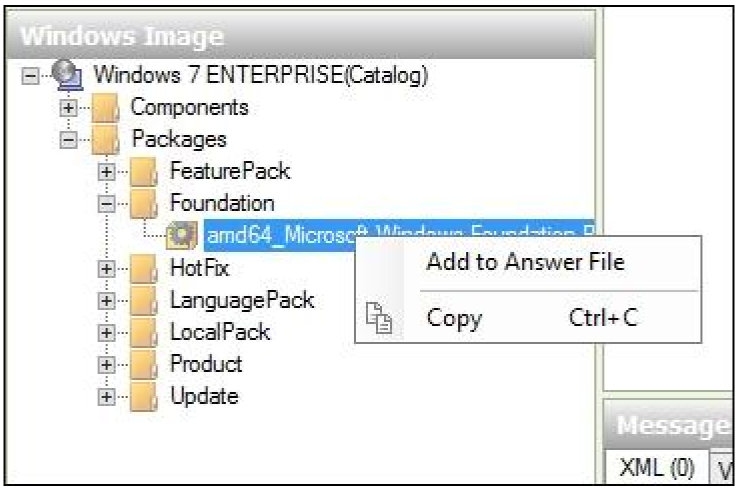
☐ Match case ☑ Look at value

☐ Match whole word

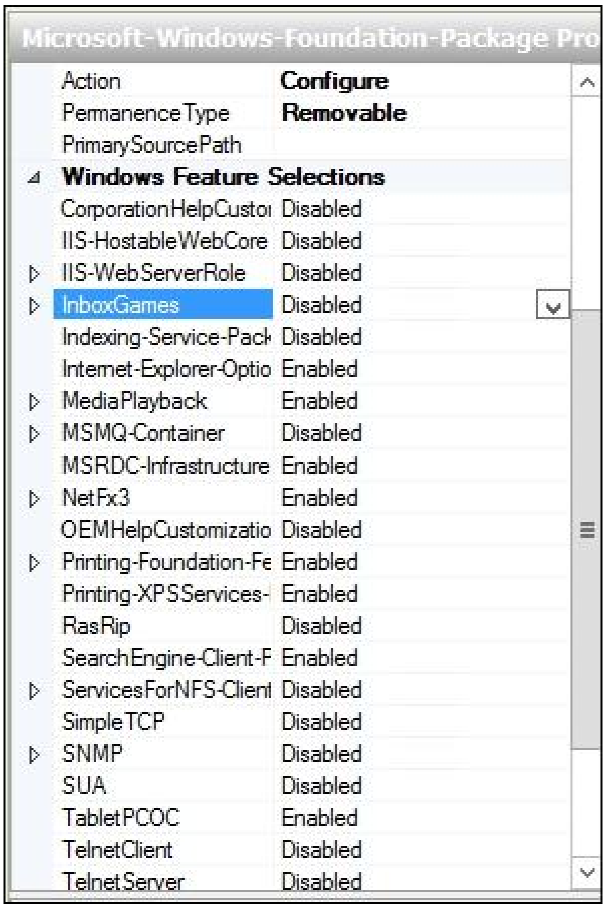
Location	Path	Value
Windows Image	amd64_Microsoft-Windows-Foundation-Package_6.1.7601.17514_/InboxGames	Disabled
Windows Image	amd64_Microsoft-Windows-Foundation-Package_6.1.7601.17514_/InboxGames/More Games	Disabled
Windows Image	amd64_Microsoft-Windows-Foundation-Package_6.1.7601.17514_/InboxGames/Internet Games	Disabled

4. In our case, we'll want them all, so select the top one, **InboxGames**. Double-click on it, and then note in the bottom-left pane, we will see it

highlighted. Right-click on it and then left-click on **Add to Answer File**:

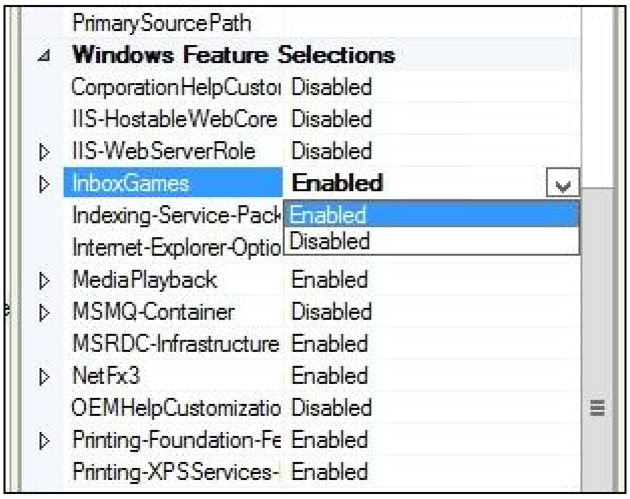


5. Note that the frame of reference in WSIM has changed. In the **Answer File** pane, there is **Packages** listed at the bottom, broken out to **Foundation** and then `amd64_Micrsooft-Windows-Foundation-Package_#Windows Version` ; on the far right, we have the break out of all the options for this entry listed. Nested in here is our **InboxGames**, and the property is **Disabled** by default. Each entry is a drop box on the far right, as shown in the following screenshot:



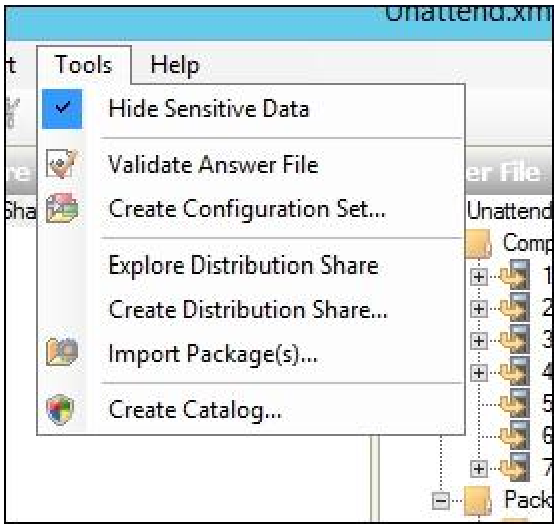
Action	Configure
PermanenceType	Removable
PrimarySourcePath	
Windows Feature Selections	
CorporationHelpCustoı	Disabled
IIS-HostableWebCore	Disabled
IIS-WebServerRole	Disabled
InboxGames	Disabled
Indexing-Service-Pack	Disabled
Internet-Explorer-Optio	Enabled
MediaPlayback	Enabled
MSMQ-Container	Disabled
MSRDC-Infrastructure	Enabled
NetFx3	Enabled
OEMHelpCustomizatio	Disabled
Printing-Foundation-Fe	Enabled
Printing-XPSServices-	Enabled
RasRip	Disabled
SearchEngine-Client-F	Enabled
ServicesForNFS-Client	Disabled
SimpleTCP	Disabled
SNMP	Disabled
SUA	Disabled
TabletPCOC	Enabled
TelnetClient	Disabled
TelnetServer	Disabled

Simply click the **Disabled** text and change to **Enabled**:

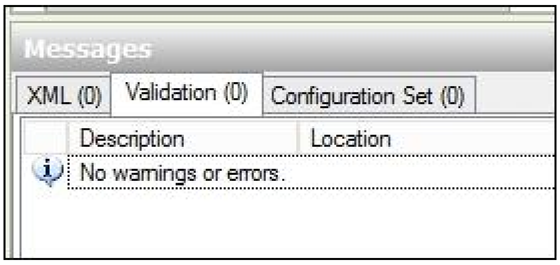


PrimarySourcePath	
Windows Feature Selections	
CorporationHelpCustoı	Disabled
IIS-HostableWebCore	Disabled
IIS-WebServerRole	Disabled
InboxGames	Enabled
Indexing-Service-Pack	Enabled
Internet-Explorer-Optio	Disabled
MediaPlayback	Enabled
MSMQ-Container	Disabled
MSRDC-Infrastructure	Enabled
NetFx3	Enabled
OEMHelpCustomizatio	Disabled
Printing-Foundation-Fe	Enabled
Printing-XPSServices-	Enabled

6. Now that we've specified that we want to enable **InboxGames** from our image, click on the **Tools** menu and select **Validate Answer File**:



7. Now, in the **Messages** area at the bottom of the screen, you should see that your validation answer file is good and has no warnings or errors:



Messages

XML (0) | Validation (0) | Configuration Set (0)

Description	Location
No warnings or errors.	

8. Finally, simply save the `Unattend.xml` file.

NOTE

Now, from this point forward, this task sequence for Windows 7 Enterprise in this deployment share will have **InboxGames** enabled.

TIP

It's an important concept, the `Unattend.xml` file follows the task sequence, not the operating system. Therefore, we can import Windows 7 Enterprise x64 edition once and make a multitude of different configurations that all use this WIM and modify it differently, based on the need.

Analyzing our changes

What has actually happened here is that the `Unattend.xml` file for our Windows 7 image in MDT has been modified. The image itself, the WIM file, is fine. It hasn't been modified; however, `Unattend.xml`, used in this task sequence, has been modified.

For example, `Unattend.xml` in the `Control Directory\1` directory now has a snippet that looks a bit similar to the following code:

```
<?xml version="1.0" encoding="utf-8"?>
<unattend xmlns="urn:schemas-microsoft-com:unattend">
  <servicing>
    <package action="configure">
      <assemblyIdentity
name="Microsoft-Windows-Foundation-Package"
version="6.1.7601.17514" processorArchitecture="amd64"
publicKeyToken="31bf3856ad364e35" language="" />
      <selection name="InboxGames" state="true" />
      <selection name="Solitaire" state="true" />
```

Leveraging the Audit mode

What if we wanted to do things outside of the options we find in `Unattend.xml`?

For this, you may want to leverage the audit mode. This is a little known mode of Windows boot, where the Windows installation is booted into the default administrator account (which, on Windows 8, is always disabled by default).

TIP

The current definition of audit mode on TechNet (http://technet.microsoft.com/en-us/library/cc722413(v=WS.10).aspx), when writing this text, is as follows:

> *"Audit Mode. Audit mode is used by OEMs and corporations to add customizations to their Windows images. Audit mode does not require settings in Windows Welcome to be applied. By bypassing Windows*

Welcome, you can get to the desktop quicker and perform your customizations. You can add additional device drivers, install applications, and test the validity of the installation. OEMs and corps should use audit mode to complete their manual customizations before shipping the computer to an end user."

In audit mode, settings in an unattended answer file in the `auditSystem` and `auditUser` configuration passes are processed. For more information about these configuration passes, see `auditSystem` and `auditUser`. If you are running in audit mode, to configure the installation to boot to Windows Welcome, run the `sysprep/oobe` command. For more information, see Sysprep Technical Reference. OEMs are required to run `sysprep/oobe` before shipping a computer to an end user.

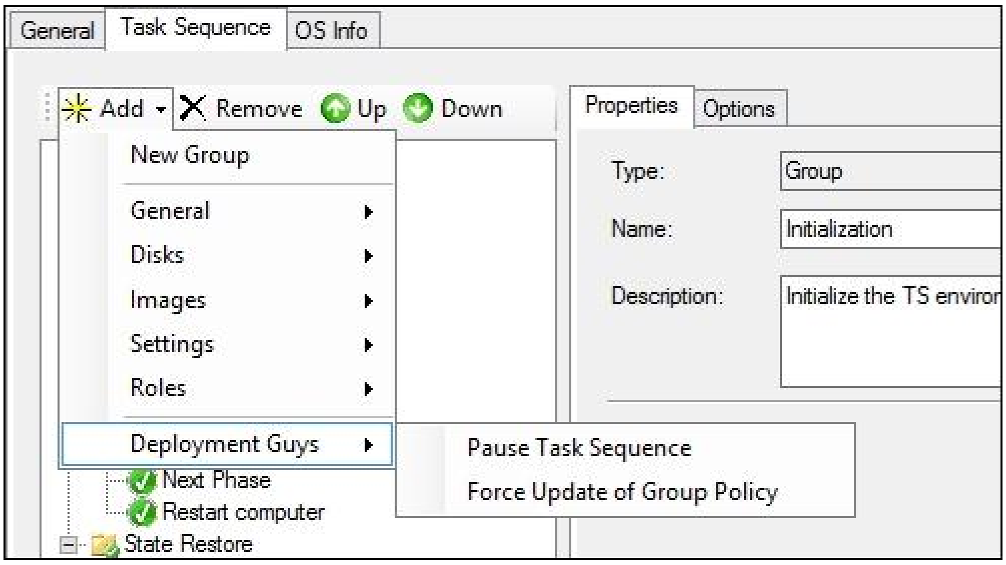
In the audit mode, all the changes to the **base administrator account** can become the default settings for other users, provided the `CopyProfile=true` switch is flipped in `Unattend.xml` (as we observed previously).

Now, MDT will enter the audit mode for us. However, if we want to customize the image further than just what WSIM provides, we can pause the Task Sequence and modify this logon for all sorts of customizations that WSIM does not expose to us.

There are a few methods of pausing the task sequence. My personal favorite is to modify the MDT Task Sequence Editor as documented at http://blogs.technet.com/b/deploymentguys/archive/2010/08/26/customising-the-mdt-task-sequence-editor.aspx .

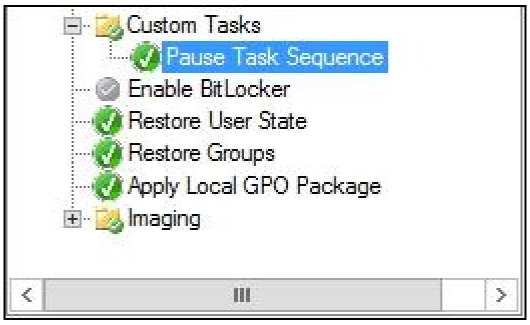
This method adds the following lines to the `actions.xml` XML file, located in the workbench machine:

```
<action divider="true" />
<action>
  <Category>Deployment Guys</Category>
  <Name>Pause Task Sequence</Name>

<Type>SMS_TaskSequence_RunCustomSuspendCommandLineAction</Type
>    <Assembly>Microsoft.BDD.Actions</Assembly>
  <Class>Microsoft.BDD.Actions.ActionRunCommandLine</Class>
  <Property type="string" name="CommandLine"
    default="cscript.exe %SCRIPTROOT%\LTISuspend.wsf" />
  <Property type="string" name="WorkingDirectory" />
  <Property type="string" name="SuccessCodes" default="0 3010"
/>
  <Property type="string" name="PackageID" />
  <Property type="string" name="RunAsUser" default="false" />
  <Property type="string" name="SMSTSRunCommandLineUserName"
/>
  <Property type="string"
name="SMSTSRunCommandLineUserPassword" />     <Property
type="boolean" name="LoadProfile" default="false" />
  <Property type="string" name="SupportedEnvironment"
default="WinPEandFullOS" />
</action>
<action>
  <Category>Deployment Guys</Category>
  <Name>Force Update of Group Policy</Name>

<Type>SMS_TaskSequence_RunCustomGPUpdateCommandLineAction</Typ
e>     <Assembly>Microsoft.BDD.Actions</Assembly>
<Class>Microsoft.BDD.Actions.ActionRunCommandLine</Class>
  <Property type="string" name="CommandLine"
    default="gpupdate.exe /force" />
  <Property type="string" name="WorkingDirectory" />
  <Property type="string" name="SuccessCodes" default="0 3010"
/>
  <Property type="string" name="PackageID" />
  <Property type="string" name="RunAsUser" default="false" />
  <Property type="string" name="SMSTSRunCommandLineUserName"
/>
  <Property type="string"
name="SMSTSRunCommandLineUserPassword" />
  <Property type="boolean" name="LoadProfile" default="false"
/>
  <Property type="string" name="SupportedEnvironment"
default="WinPEandFullOS" />
</action>
```
Now, my wizard in the MDT console appears as shown in the following image:



Type: Group
Name: Initialization
Description: Initialize the TS enviror

So what I would do here is place the **Pause Task Sequence** step in my **Custom Tasks** folder, as follows:

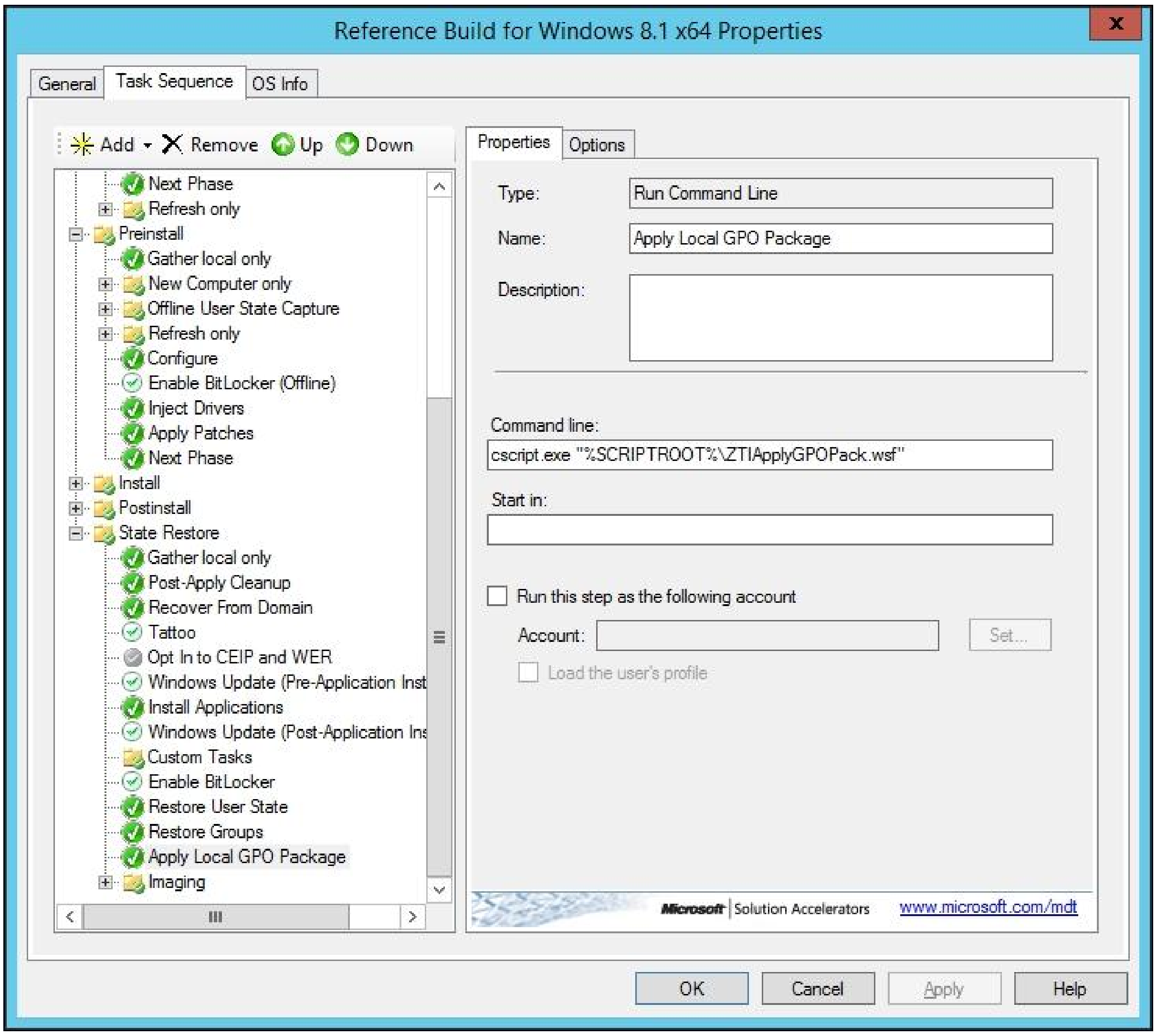


This will put a friendly link on the desktop to double-click when you choose to resume the task sequence. Thanks **Deployment Guys**!

Local Policy Object Customizations and SCM

In 2008, Aaron Margosis published the **Local Group Policy Object** (**LGPO**) toolset to manage and deploy local group policy objects. With these tools, it became easy to manage settings in local policy. You could insert these policies into your image with the scripts and tools that Aaron published and it worked fairly well.

Fast forward to today, **Security Compliance Manager** (**SCM**) (what LGPO became in essence) is now the tool for doing this. With SCM, you can manage policies, create and compare them, and then back them up. They are then (from a deployment perspective) GPO Packs, and then MDT can import them as part of the task sequence, as shown in the following image:



- Next Phase
- Refresh only
- Preinstall
  - Gather local only
  - New Computer only
  - Offline User State Capture
  - Refresh only
  - Configure
  - Enable BitLocker (Offline)
  - Inject Drivers
  - Apply Patches
  - Next Phase
- Install
- Postinstall
- State Restore
  - Gather local only
  - Post-Apply Cleanup
  - Recover From Domain
  - Tattoo
  - Opt In to CEIP and WER
  - Windows Update (Pre-Application Inst
  - Install Applications
  - Windows Update (Post-Application Ins
  - Custom Tasks
  - Enable BitLocker
  - Restore User State
  - Restore Groups
  - Apply Local GPO Package
  - Imaging

Type:	Run Command Line
Name:	Apply Local GPO Package
Description:	
Command line:	cscript.exe "%SCRIPTROOT%\ZTIApplyGPOPack.wsf"
Start in:	

☐ Run this step as the following account

Account:

☐ Load the user's profile

Using this suite of tools, we can then bake into the image customizations that we want to be the default for Windows. Many security-conscious organizations will implement this as a practice so that the machines that are built immediately have a default set of GPO settings applied. These can then be modified remotely via SCM and Active Directory Group Policy processing.

One must, however, consider the policy processing order when making these image modifications. According to http://technet.microsoft.com/en-us/library/cc785665(v=WS.10).aspx , the ordering of policy processing and precedence is as follows:

"Local Group Policy object—Each computer has exactly one Group Policy object that is stored locally. This processes for both computer and user Group Policy processing."

"Site—Any GPOs that have been linked to the site that the computer belongs to are processed next. Processing is in the order that is specified by the administrator, on the Linked Group Policy Objects tab for the site in Group Policy Management Console (GPMC). The GPO with the lowest link order is processed last, and therefore has the highest precedence."

"Domain-Processing of multiple domain—linked GPOs is in the order specified by the administrator, on the Linked Group Policy Objects tab for the domain in GPMC. The GPO with the lowest link order is processed last, and therefore has the highest precedence."

"Organizational units—GPOs that are linked to the organizational unit that is highest in the Active Directory hierarchy are processed first, then GPOs that are linked to its child organizational unit, and so on. Finally, the GPOs that are linked to the organizational unit that contains the user or computer are processed."

Therefore, changes we make in our LGPO will not be overridden by future updates of Active Directory policies, without modifying the local policy again, that is.

TIP

Another reason to utilize a tool such as SCM for configuration templates is that the work put into the templates by Microsoft Consulting Services can be leveraged as a baseline standard for your organization as well. Literally hundreds of man-hours of experience and case work have resulted in these templates for use in a security and management perspective.

Shell customizations

In Windows XP, in the old days, one meticulously set up the desktop and Start menu layout, and then copied the profile over the default profile by hand. This was never a supported method for user customization, and over the years, some problems occurred due to its use as a standard practice.

With the Windows Vista model of user profile customization, this has changed somewhat. Questions on Windows 7, 8, 8.1, and beyond, generally boil down to topics such as how to pin an internal application to the Start menu, how to pin it to the taskbar, and how to customize the Start screen in Windows 8.

Windows 7 Start menu and taskbar

Supported methods of Windows 7 customization are well documented in several blogs and support articles by Microsoft. In this text, we'll cover these modifications and the pros and cons of the mentioned methods.

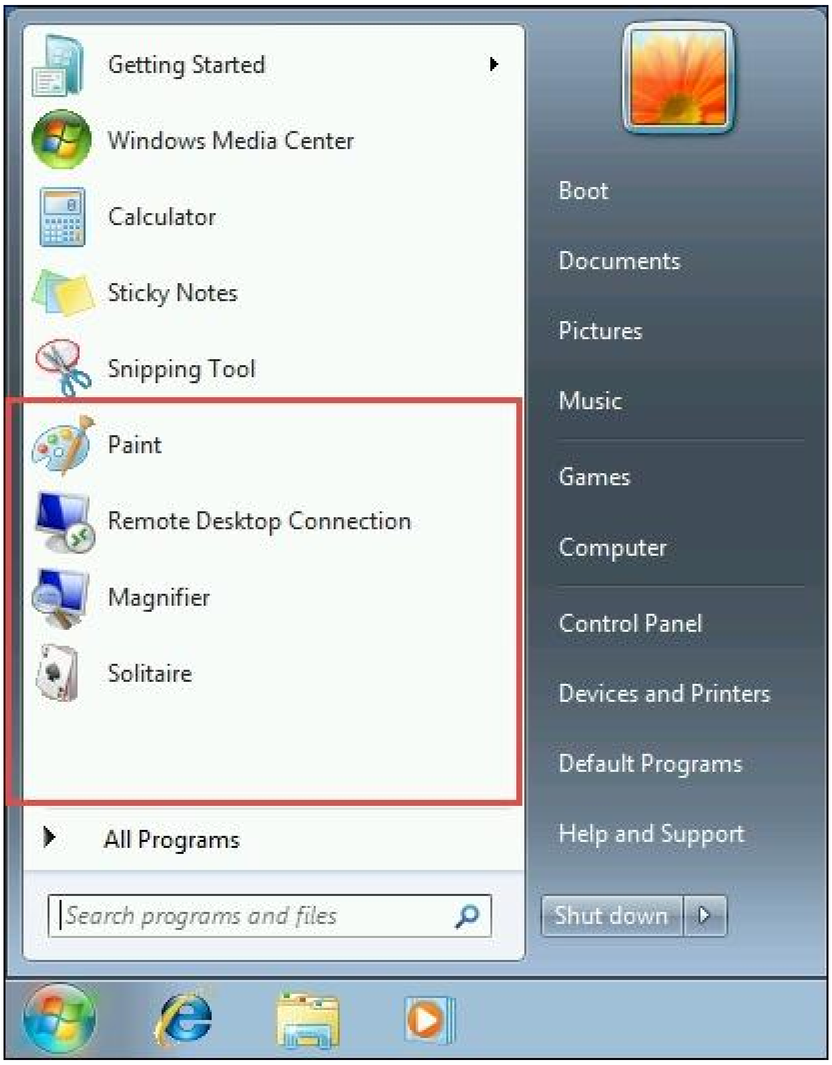
TIP

Note that the licensing of Windows 7 in some cases precludes image customization. The TechNet article at http://technet.microsoft.com/en-us/library/ff730914.aspx , in particular, discusses Windows 7 Professional and the customization of this image based on your particular licensing scenario. It is not supported in any case to take an OEM image that you get on a factory-built PC and capture and customize it. This image has already been syspreped, and therefore, could cause unpredictability in image creation and deployment, as well as violate the OEM license.

First, Scott McArthur reviewed ways of customizing the Start menu and taskbar in Windows 7 by way of `Unattend.xml`. This is a pretty straightforward method of image customization and is fully supported. The article was posted to AskCore on March 16th 2010, and is located at http://blogs.technet.com/b/askcore/archive/2010/03/16/how-to-customize-the-windows-7-start-menu-and-taskbar-using-unattend-xml.aspx .

In this blog, Scott goes over two items in `Unattend.xml`, `Microsoft-Windows-Shell-Setup\StartPanelLinks` and

Microsoft-Windows-Shell-Setup\TaskBarLinks. These settings should be placed in the oobeSystem part of the Unattend.xml file. The StartPanelLinks area is fairly straightforward; it's modifying the part of the Start menu shown in the following red box (the spot above is system-reserved):



To modify these, simply define them as links, as shown in the following example:

```
<StartPanelLinks>
<Link0>%ALLUSERSPROFILE%\Microsoft\Windows\Start
Menu\Programs\accessories\Notepad.lnk</Link0>
<Link1>%ALLUSERSPROFILE%\Microsoft\Windows\Start
Menu\Programs\accessories\Windows Explorer.lnk</Link1>
</StartPanelLinks>
```

This example will modify the menu to display Link0 as Notepad and Link1 as Windows Explorer.

To modify the TaskbarLinks area, the logic is the same:

```
<TaskbarLinks>
<Link0>%ALLUSERSPROFILE%\Microsoft\Windows\Start
Menu\Programs\accessories\Notepad.lnk</Link0>
<Link1>%ALLUSERSPROFILE%\Microsoft\Windows\Start
Menu\Programs\accessories\Windows Explorer.lnk</Link1>
</TaskbarLinks>
```

There are some caveats to this setup that one must consider:

- We need `CopyProfile=true` to be enabled in our `Unattend.xml`. This should be expected for the default profile configuration at this point.
- You cannot remove the icons in the Start menu above the red box that I detailed. If you do, they are simply recreated when a new user logs in. Microsoft does not support removing these icons in any manner.

Windows 7 background, logon screen, and user tiles

Modifying the Windows 7 logon screen is a fairly straightforward process. One simply modifies a `regkey` value, drops in a graphic, and it's done.

The `regkey` is located at `HKLM\Software\Microsoft\Windows\CurrentVersion\Authentication\LogonUI\Background`, and the value of `OEMBackground` is set to `1`.

Simply place your custom image in `%windir%\system32\oobe\info\backgrounds`. The desired logon wallpaper should have the `backgroundDefault.jpg` filename.

TIP

Images must be less than 256 KB in size. With regards to resolution of the image, set the resolution as you desire. If the monitor is set to an alternate resolution, the graphic will stretched to fit.

The Windows 7 wallpaper is pretty easy too, you can set it in the image prior to `sysprep/capture` through the normal UI; and as long as `CopyProfile=true`, it will keep it as the default setting. Another method might be to set the `HKEY_CURRENT_USER\Control Panel\Desktop` registry key value of wallpaper to the `C:\Windows\Web\Wallpaper\wallpaper.jpg` path, or even use the GPO documented at http://gpsearch.azurewebsites.net/#141 (Desktop Wallpaper User Policy).

To modify user tile to the company branding, the images are located in the following path:

- `%ProgramData%\Microsoft\User Account Pictures\Guest.bmp`
- `%ProgramData%\Microsoft\User Account Pictures\User.bmp`

However, I think that utilizing the picture of the user for the user tile is even more interesting. On a domain-joined Windows 7 machine, the user tiles are stored for the users in the `C:\ProgramData\Microsoft\User Account Pictures` path. There is an API to pull the tiles from Active Directory. So get to work!

Just kidding. Jacob Steenhagen was kind enough to create a `setUserTile.exe` utility that does the work for you. You can check it out at http://jacob.steenhagen.us/blog/2012/02/loading-a-windows-7-user-tile-using-the-picture-in-active-directory/.

You can use this tool (source documented in the blog post) to set this automatically from your Active Directory User Tile attribute that you have already populated.

Windows 8 customizations

Windows 8 brought us a new Start screen experience that changes the game somewhat with regards to image customization.

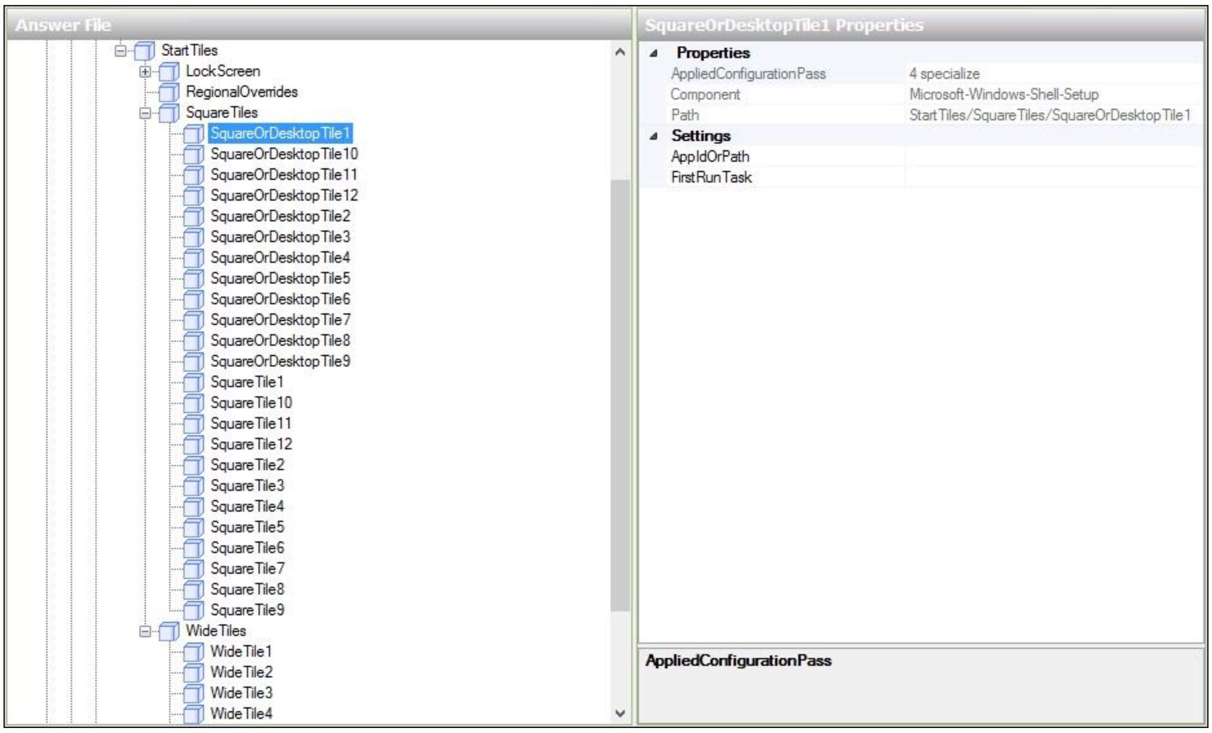
With regards to customizing the Start screen, Ben Hunter wrote (http://blogs.technet.com/b/deploymentguys/archive/2012/10/26/start-screen-customization-with-mdt.aspx):

> *"There are three approaches that you can use:*
>
> 1. *Use the Unattend.xml file to define which applications will appear in each "slot" on the Start Screen as detailed here.*
> 2. *Manually customize the Start Screen, then use CopyProfile to apply the customizations to the default user profile.*
> 3. *Manually customize the Start Screen and capture the layout file created, then copy this file to the default user during image deployment."*

We've already discussed `CopyProfile=true` to some extent in this text; the other two options are interesting, where new customization capabilities are present.

So, let's discuss using the `Unattend.xml` file to define apps and the slotting model in the `Unattend.xml` file. This will require the `AppID` key of the applications needed to be pinned by the way, and TechNet covers the how-to at http://technet.microsoft.com/en-us/library/jj134269.aspx#BKMK_StartTiles , but in essence, `Get-AppxPackage` will reveal this. Once you have your AppxPackage IDs, you can add the `StartTiles` setting to the `Unattend.xml` file with WSIM or Notepad. You will construct the layout using the Wide Tile##, Square Tile## and DesktopOrSquare Tile##.

In WSIM, these appear as follows:



```
Answer File
StartTiles
  LockScreen
  RegionalOverrides
  SquareTiles
    SquareOrDesktopTile1
    SquareOrDesktopTile10
    SquareOrDesktopTile11
    SquareOrDesktopTile12
    SquareOrDesktopTile2
    SquareOrDesktopTile3
    SquareOrDesktopTile4
    SquareOrDesktopTile5
    SquareOrDesktopTile6
    SquareOrDesktopTile7
    SquareOrDesktopTile8
    SquareOrDesktopTile9
    SquareTile1
    SquareTile10
    SquareTile11
    SquareTile12
    SquareTile2
    SquareTile3
    SquareTile4
    SquareTile5
    SquareTile6
    SquareTile7
    SquareTile8
    SquareTile9
  WideTiles
    WideTile1
    WideTile2
    WideTile3
    WideTile4
```

SquareOrDesktopTile1 Properties

Properties	
AppliedConfigurationPass	4 specialize
Component	Microsoft-Windows-Shell-Setup
Path	StartTiles/SquareTiles/SquareOrDesktopTile1
Settings	
AppIdOrPath	
FirstRunTask	

AppliedConfigurationPass

`AppsFolderLayout.bin` is actually discussed in the same TechNet article at http://technet.microsoft.com/en-us/library/jj134269.aspx#BKMK_AppFolder. To do this, create a user profile, set up the Start screen, and then copy this design back into your master image using **xcopy**. This works similar to the `CopyProfile=true` method, except you overwrite `AppsFolderLayout.bin` from the customized profile to the default user profile while in the audit mode. Then, all future profiles will load this customized layout.

NOTE

Windows 10

All the concepts shown in this chapter are still valid for Windows 10 image creation. In addition the following information is relevant to Windows 10 specifically:

- Windows 10 comes with a new **Windows Imaging and Configuration Designer** (**Windows ICD**). This Windows ICD is used for configuring features and policies by provisioning packages. This tool is designed specifically for Windows 10 customization and is discussed in Appendix, *Additional Enterprise Configuration Items*, of this book.
- MDT still needs modification of `Unattend.xml`. `Unattend.xml` can only be modified with **Windows System Image Manager** (**WSIM**), so you will still need WSIM.
- Currently (at the time of writing) `CopyProfile` on Windows 10 can create unexpected results. Therefore the recommendation is that you do not use `CopyProfile` with Windows 10 in your task sequences. This is subject to change with future updates to Windows 10.
- Start menu layout customization has been extended for Windows 10. It can be partially locked down. More information can be found at https://blogs.technet.microsoft.com/deploymentguys/2016/03/07/windows-10-start-layout-customization/.

Chapter 4. Default User Profile Customization

In the previous chapter, we discussed the technique of collecting a reference build for Windows in a virtual machine on Hyper-V or physical box with boot media on a USB stick. In this chapter, we'll cover how to customize this base image as part of our process. Not from an application build perspective, but to make the image branded, remove or add features, remove the **out-of-box experience** (**OOBE**), and so forth. Now some general things one might want done to their image would be removing the games, setting the Internet Explorer default settings, customizing the background screen, and other branding components, removing the ability to access things in the UI by default, and so on, as well as customizing the image for a kiosk, tablet, cash register, ATM, exec's laptop, VDI image, and so on.

In this chapter, we'll go through the following topics:

- How to customize the Windows image
- Windows System Image Manager and `Unattend.xml`
- The differences between Windows 7 and 8 in UI customizations

Customizing the image

You may customize the image before running your reference task sequence. Customizations are done in many ways; some are script-based, some group policies, some are manual efforts, some are done in a tool known as **Windows System Image Manager** (**WSIM**), and others can be done with PowerShell scripts.

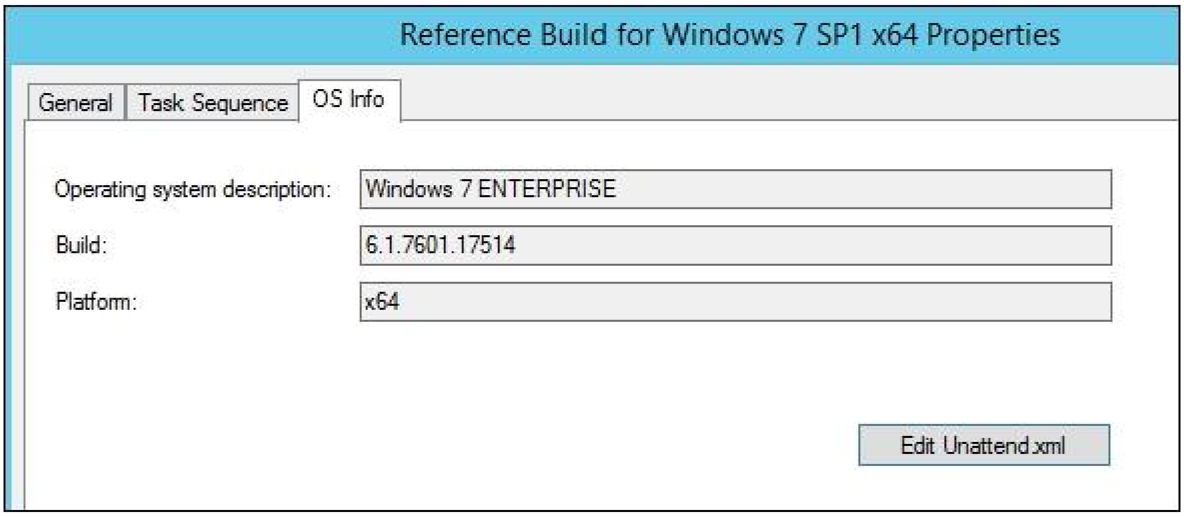
Checking out the customization documentation

The general caveats and principles of image customization are documented for Windows at this KB at http://support.microsoft.com/kb/973289 . The essential gist of this is that to customize the user profile of a future user of Windows 7 or 2008 R2, you must customize the default local user profile. When this is done, the settings in the default profile will become the settings of new user profiles on the system. Note that this is the only officially supported method of customizing the default user profile in Windows 7.

For Windows 8.x, things change somewhat. The steps are documented at http://technet.microsoft.com/en-us/library/hh825135.aspx and involve, generally speaking, the same steps as for Windows 7. However, a new functionality is provided to customize the Start menu, documented at http://technet.microsoft.com/en-us/library/jj134269.aspx .

Accessing Windows System Information Manager

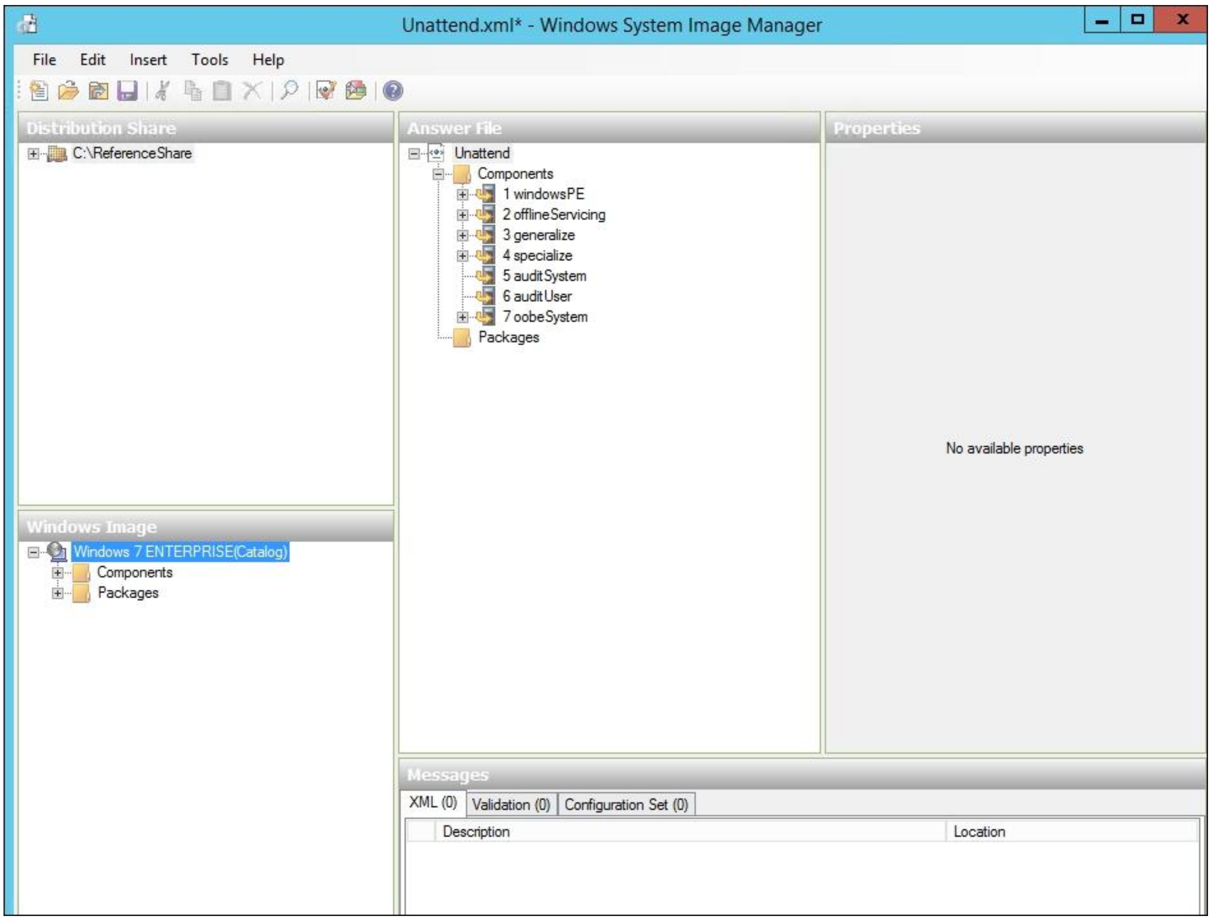
An easy way to use this customization, as mentioned previously, is **WSIM**. It's quite easy to access; simply right-click on your task sequence and select the **OS Info** tab, and click on **Edit Unattend.xml**:


Reference Build for Windows 7 SP1 x64 Properties

General | Task Sequence | OS Info

Operating system description: Windows 7 ENTERPRISE
Build: 6.1.7601.17514
Platform: x64

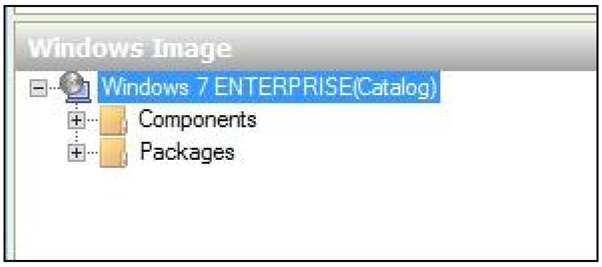
Then you'll be rewarded with a modified view of the typical MMC console layout for `Unattend.xml` editing, as shown in the following:



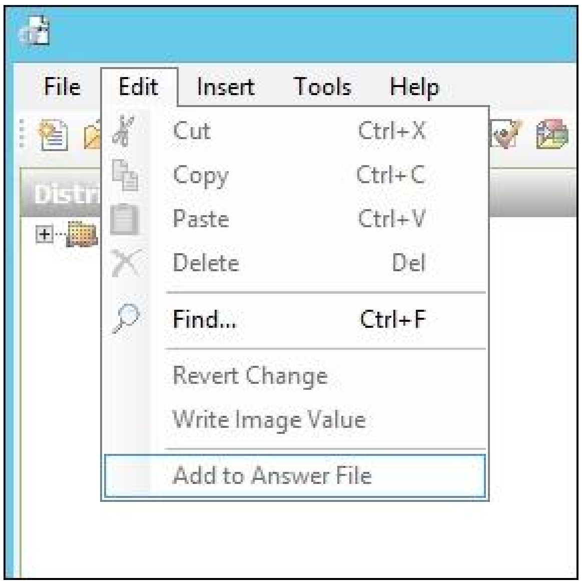
Adding games to our Enterprise image

This interface is a bit intimidating at first, and somewhat unclear about where things should be edited, as they will sometimes appear to be applicable in multiple locations of the XML. For those who have manually edited the `Unattend.xml` files in the past, this will make little sense perhaps. It may be best to learn by doing, so let's add games to our Windows 7 Enterprise image by performing the following exercise:

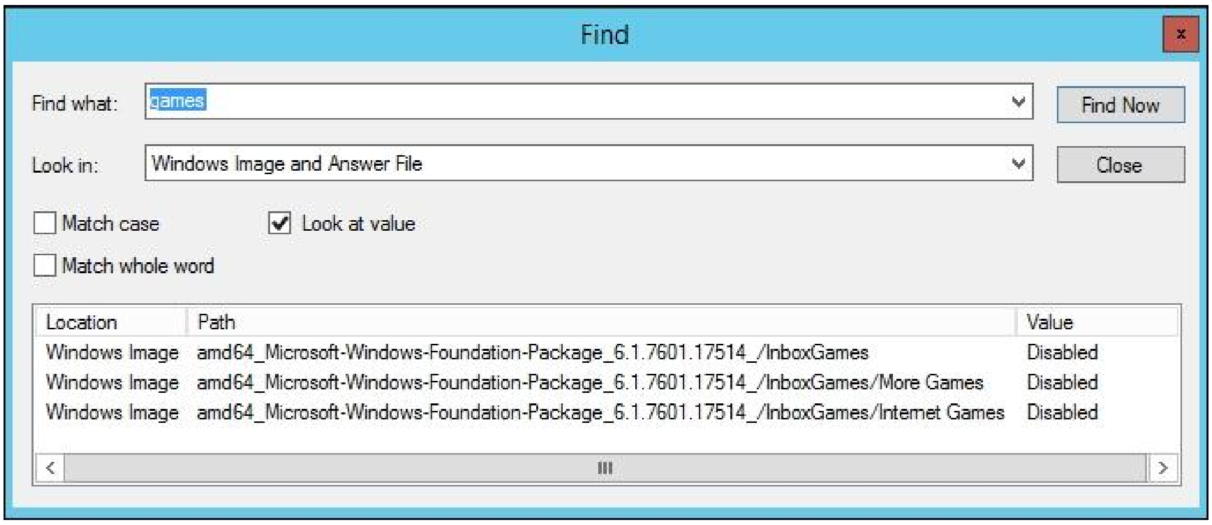
1. Ensure the focus is on the **Windows Image** area of the MMC, as shown in the following image:



2. Click on **Find...** in the **Edit** menu:



3. Now, simply type `games` here and observe that there are three responses to the query:


Find

Find what: games

Look in: Windows Image and Answer File

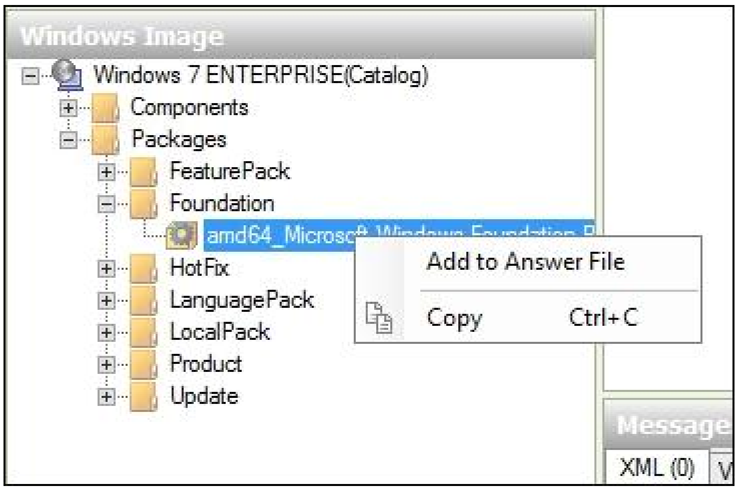
☐ Match case ☑ Look at value

☐ Match whole word

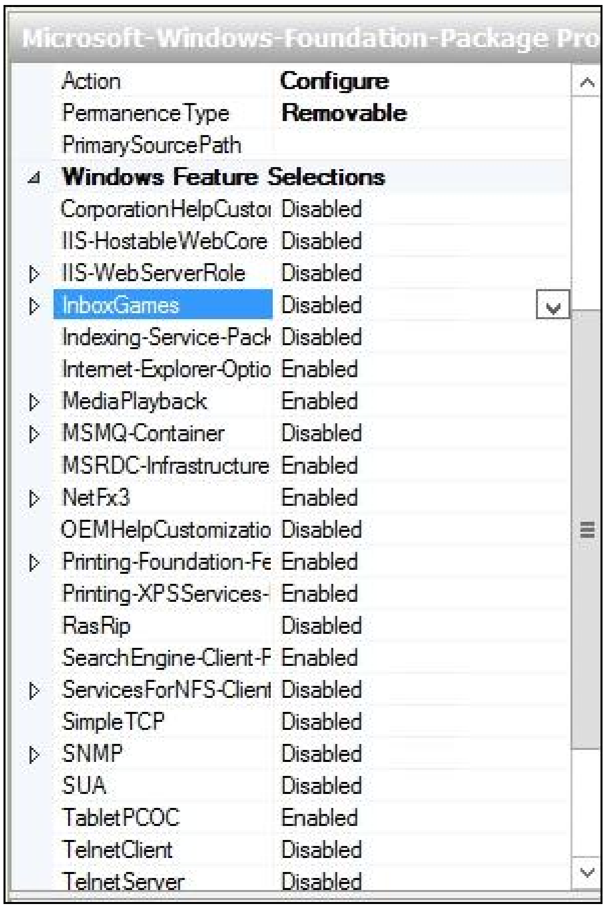
Location	Path	Value
Windows Image	amd64_Microsoft-Windows-Foundation-Package_6.1.7601.17514_/InboxGames	Disabled
Windows Image	amd64_Microsoft-Windows-Foundation-Package_6.1.7601.17514_/InboxGames/More Games	Disabled
Windows Image	amd64_Microsoft-Windows-Foundation-Package_6.1.7601.17514_/InboxGames/Internet Games	Disabled

4. In our case, we'll want them all, so select the top one, **InboxGames**. Double-click on it, and then note in the bottom-left pane, we will see it

highlighted. Right-click on it and then left-click on **Add to Answer File**:

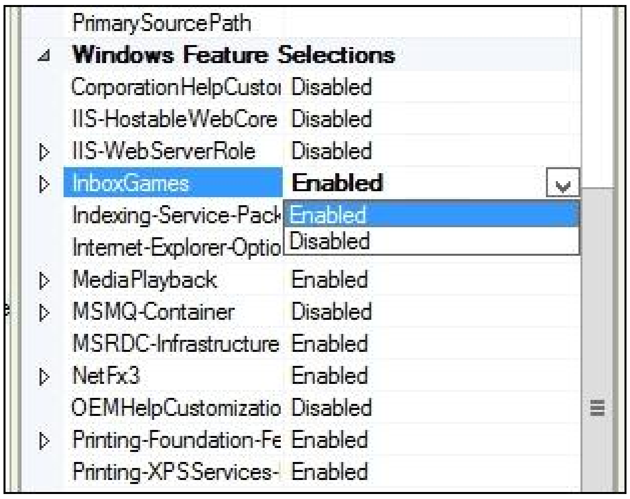


5. Note that the frame of reference in WSIM has changed. In the **Answer File** pane, there is **Packages** listed at the bottom, broken out to **Foundation** and then `amd64_Micrsooft-Windows-Foundation-Package_#Windows Version` ; on the far right, we have the break out of all the options for this entry listed. Nested in here is our **InboxGames**, and the property is **Disabled** by default. Each entry is a drop box on the far right, as shown in the following screenshot:


Microsoft-Windows-Foundation-Package Pro

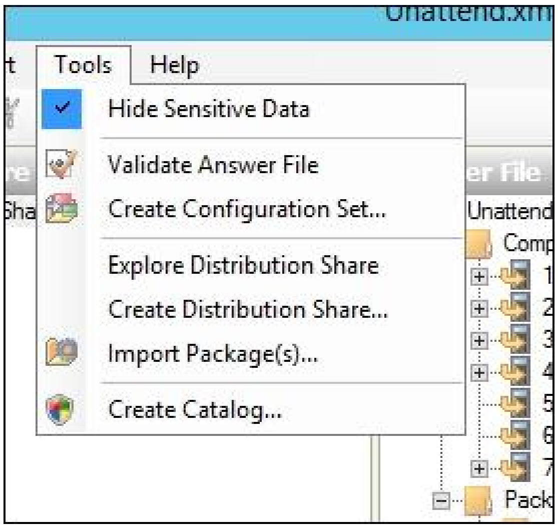
Action	**Configure**
PermanenceType	**Removable**
PrimarySourcePath	
Windows Feature Selections	
CorporationHelpCustor	Disabled
IIS-HostableWebCore	Disabled
IIS-WebServerRole	Disabled
InboxGames	Disabled
Indexing-Service-Pack	Disabled
Internet-Explorer-Optio	Enabled
MediaPlayback	Enabled
MSMQ-Container	Disabled
MSRDC-Infrastructure	Enabled
NetFx3	Enabled
OEMHelpCustomizatio	Disabled
Printing-Foundation-Fe	Enabled
Printing-XPSServices-	Enabled
RasRip	Disabled
SearchEngine-Client-F	Enabled
ServicesForNFS-Client	Disabled
SimpleTCP	Disabled
SNMP	Disabled
SUA	Disabled
TabletPCOC	Enabled
TelnetClient	Disabled
TelnetServer	Disabled

Simply click the **Disabled** text and change to **Enabled**:

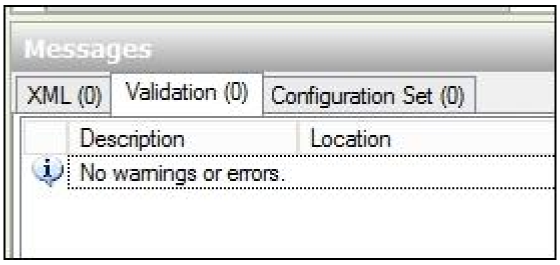


PrimarySourcePath	
Windows Feature Selections	
CorporationHelpCustoı	Disabled
IIS-HostableWebCore	Disabled
IIS-WebServerRole	Disabled
InboxGames	Enabled
Indexing-Service-Pack	Enabled
Internet-Explorer-Optio	Disabled
MediaPlayback	Enabled
MSMQ-Container	Disabled
MSRDC-Infrastructure	Enabled
NetFx3	Enabled
OEMHelpCustomizatio	Disabled
Printing-Foundation-Fe	Enabled
Printing-XPSServices-	Enabled

6. Now that we've specified that we want to enable **InboxGames** from our image, click on the **Tools** menu and select **Validate Answer File**:



7. Now, in the **Messages** area at the bottom of the screen, you should see that your validation answer file is good and has no warnings or errors:



Messages

XML (0) | Validation (0) | Configuration Set (0)

Description	Location
No warnings or errors.	

8. Finally, simply save the `Unattend.xml` file.

NOTE

Now, from this point forward, this task sequence for Windows 7 Enterprise in this deployment share will have **InboxGames** enabled.

TIP

It's an important concept, the `Unattend.xml` file follows the task sequence, not the operating system. Therefore, we can import Windows 7 Enterprise x64 edition once and make a multitude of different configurations that all use this WIM and modify it differently, based on the need.

Analyzing our changes

What has actually happened here is that the `Unattend.xml` file for our Windows 7 image in MDT has been modified. The image itself, the WIM file, is fine. It hasn't been modified; however, `Unattend.xml`, used in this task sequence, has been modified.

For example, `Unattend.xml` in the `Control Directory\1` directory now has a snippet that looks a bit similar to the following code:

```
<?xml version="1.0" encoding="utf-8"?>
<unattend xmlns="urn:schemas-microsoft-com:unattend">
  <servicing>
    <package action="configure">
      <assemblyIdentity
name="Microsoft-Windows-Foundation-Package"
version="6.1.7601.17514" processorArchitecture="amd64"
publicKeyToken="31bf3856ad364e35" language="" />
      <selection name="InboxGames" state="true" />
      <selection name="Solitaire" state="true" />
```

Leveraging the Audit mode

What if we wanted to do things outside of the options we find in `Unattend.xml`?

For this, you may want to leverage the audit mode. This is a little known mode of Windows boot, where the Windows installation is booted into the default administrator account (which, on Windows 8, is always disabled by default).

TIP

The current definition of audit mode on TechNet (http://technet.microsoft.com/en-us/library/cc722413(v=WS.10).aspx), when writing this text, is as follows:

> *"Audit Mode. Audit mode is used by OEMs and corporations to add customizations to their Windows images. Audit mode does not require settings in Windows Welcome to be applied. By bypassing Windows*

Welcome, you can get to the desktop quicker and perform your customizations. You can add additional device drivers, install applications, and test the validity of the installation. OEMs and corps should use audit mode to complete their manual customizations before shipping the computer to an end user."

In audit mode, settings in an unattended answer file in the `auditSystem` and `auditUser` configuration passes are processed. For more information about these configuration passes, see `auditSystem` and `auditUser`. If you are running in audit mode, to configure the installation to boot to Windows Welcome, run the `sysprep/oobe` command. For more information, see Sysprep Technical Reference. OEMs are required to run `sysprep/oobe` before shipping a computer to an end user.

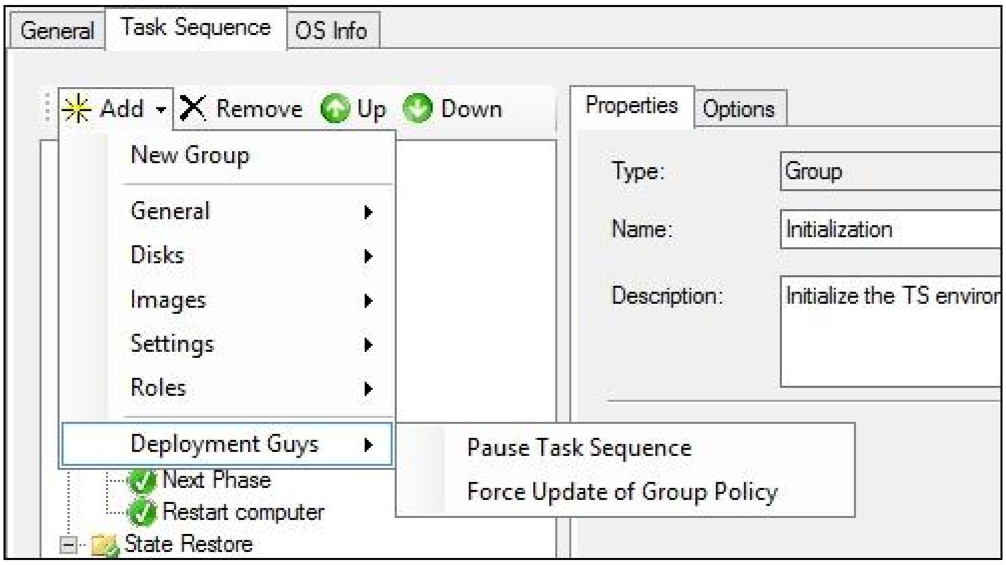
In the audit mode, all the changes to the **base administrator account** can become the default settings for other users, provided the `CopyProfile=true` switch is flipped in `Unattend.xml` (as we observed previously).

Now, MDT will enter the audit mode for us. However, if we want to customize the image further than just what WSIM provides, we can pause the Task Sequence and modify this logon for all sorts of customizations that WSIM does not expose to us.

There are a few methods of pausing the task sequence. My personal favorite is to modify the MDT Task Sequence Editor as documented at http://blogs.technet.com/b/deploymentguys/archive/2010/08/26/customising-the-mdt-task-sequence-editor.aspx .

This method adds the following lines to the `actions.xml` XML file, located in the workbench machine:

```
<action divider="true" />
<action>
  <Category>Deployment Guys</Category>
  <Name>Pause Task Sequence</Name>

<Type>SMS_TaskSequence_RunCustomSuspendCommandLineAction</Type
>   <Assembly>Microsoft.BDD.Actions</Assembly>
  <Class>Microsoft.BDD.Actions.ActionRunCommandLine</Class>
  <Property type="string" name="CommandLine"
    default="cscript.exe %SCRIPTROOT%\LTISuspend.wsf" />
  <Property type="string" name="WorkingDirectory" />
  <Property type="string" name="SuccessCodes" default="0 3010"
/>
  <Property type="string" name="PackageID" />
  <Property type="string" name="RunAsUser" default="false" />
  <Property type="string" name="SMSTSRunCommandLineUserName"
/>
  <Property type="string"
name="SMSTSRunCommandLineUserPassword" />     <Property
type="boolean" name="LoadProfile" default="false" />
  <Property type="string" name="SupportedEnvironment"
default="WinPEandFullOS" />
</action>
<action>
  <Category>Deployment Guys</Category>
  <Name>Force Update of Group Policy</Name>

<Type>SMS_TaskSequence_RunCustomGPUpdateCommandLineAction</Typ
e>     <Assembly>Microsoft.BDD.Actions</Assembly>
<Class>Microsoft.BDD.Actions.ActionRunCommandLine</Class>
  <Property type="string" name="CommandLine"
    default="gpupdate.exe /force" />
  <Property type="string" name="WorkingDirectory" />
  <Property type="string" name="SuccessCodes" default="0 3010"
/>
  <Property type="string" name="PackageID" />
  <Property type="string" name="RunAsUser" default="false" />
  <Property type="string" name="SMSTSRunCommandLineUserName"
/>
  <Property type="string"
name="SMSTSRunCommandLineUserPassword" />
  <Property type="boolean" name="LoadProfile" default="false"
/>
  <Property type="string" name="SupportedEnvironment"
default="WinPEandFullOS" />
</action>
```
Now, my wizard in the MDT console appears as shown in the following image:

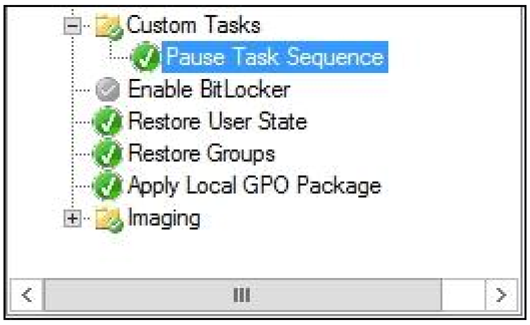


Type: Group

Name: Initialization

Description: Initialize the TS enviror

So what I would do here is place the **Pause Task Sequence** step in my **Custom Tasks** folder, as follows:

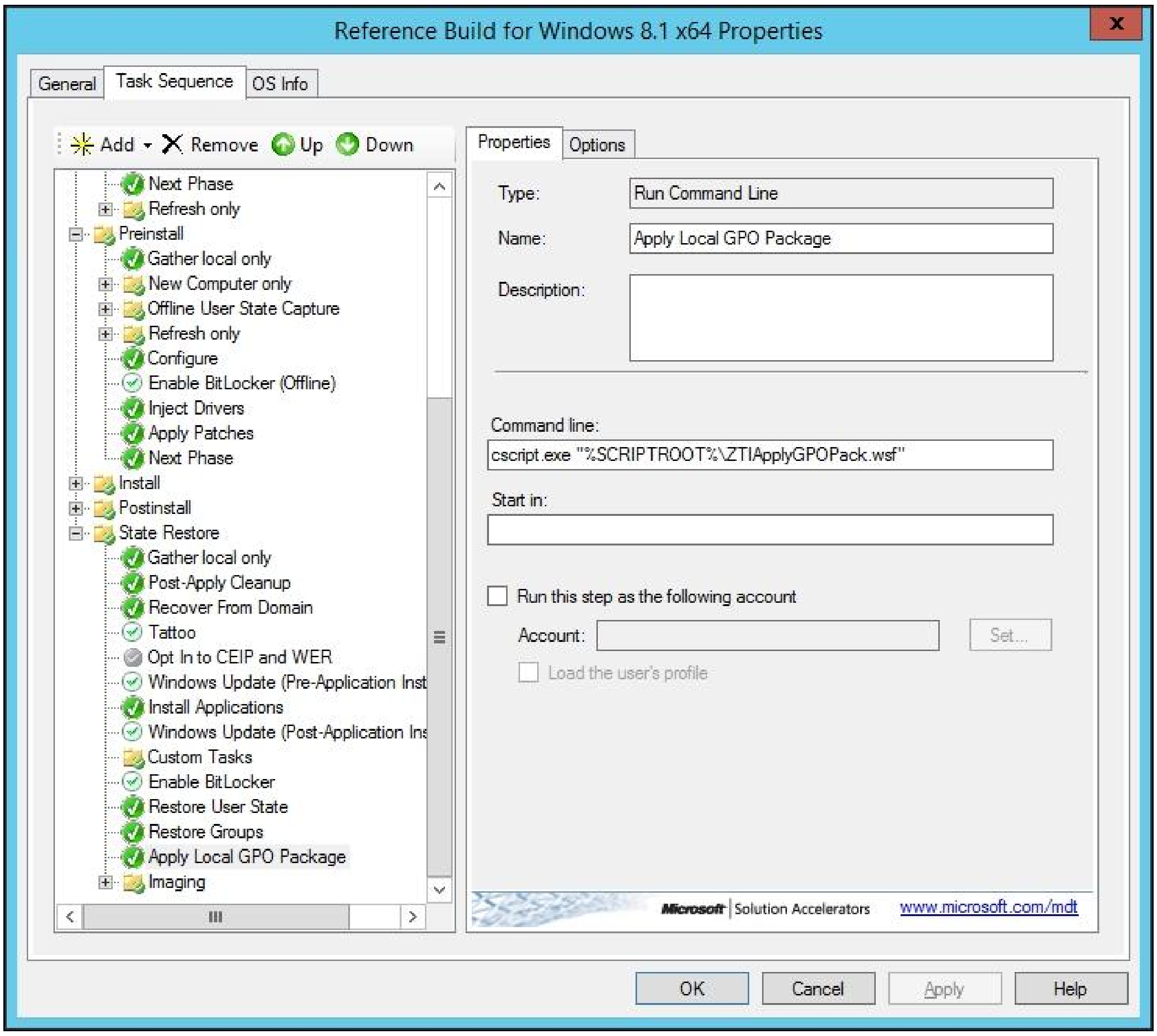


This will put a friendly link on the desktop to double-click when you choose to resume the task sequence. Thanks **Deployment Guys**!

Local Policy Object Customizations and SCM

In 2008, Aaron Margosis published the **Local Group Policy Object** (**LGPO**) toolset to manage and deploy local group policy objects. With these tools, it became easy to manage settings in local policy. You could insert these policies into your image with the scripts and tools that Aaron published and it worked fairly well.

Fast forward to today, **Security Compliance Manager** (**SCM**) (what LGPO became in essence) is now the tool for doing this. With SCM, you can manage policies, create and compare them, and then back them up. They are then (from a deployment perspective) GPO Packs, and then MDT can import them as part of the task sequence, as shown in the following image:


Reference Build for Windows 8.1 x64 Properties

General | Task Sequence | OS Info

Add | Remove | Up | Down

- Next Phase
- Refresh only
- Preinstall
  - Gather local only
  - New Computer only
  - Offline User State Capture
  - Refresh only
  - Configure
  - Enable BitLocker (Offline)
  - Inject Drivers
  - Apply Patches
  - Next Phase
- Install
- Postinstall
- State Restore
  - Gather local only
  - Post-Apply Cleanup
  - Recover From Domain
  - Tattoo
  - Opt In to CEIP and WER
  - Windows Update (Pre-Application Inst
  - Install Applications
  - Windows Update (Post-Application Ins
  - Custom Tasks
  - Enable BitLocker
  - Restore User State
  - Restore Groups
  - Apply Local GPO Package
  - Imaging

Properties | Options

Type: Run Command Line

Name: Apply Local GPO Package

Description:

Command line:

```
cscript.exe "%SCRIPTROOT%\ZTIApplyGPOPack.wsf"
```

Start in:

☐ Run this step as the following account

Account: Set...

☐ Load the user's profile

Microsoft | Solution Accelerators www.microsoft.com/mdt

OK | Cancel | Apply | Help

Using this suite of tools, we can then bake into the image customizations that we want to be the default for Windows. Many security-conscious organizations will implement this as a practice so that the machines that are built immediately have a default set of GPO settings applied. These can then be modified remotely via SCM and Active Directory Group Policy processing.

One must, however, consider the policy processing order when making these image modifications. According to http://technet.microsoft.com/en-us/library/cc785665(v=WS.10).aspx , the ordering of policy processing and precedence is as follows:

"Local Group Policy object—Each computer has exactly one Group Policy object that is stored locally. This processes for both computer and user Group Policy processing."

"Site—Any GPOs that have been linked to the site that the computer belongs to are processed next. Processing is in the order that is specified by the administrator, on the Linked Group Policy Objects tab for the site in Group Policy Management Console (GPMC). The GPO with the lowest link order is processed last, and therefore has the highest precedence."

"Domain-Processing of multiple domain—linked GPOs is in the order specified by the administrator, on the Linked Group Policy Objects tab for the domain in GPMC. The GPO with the lowest link order is processed last, and therefore has the highest precedence."

"Organizational units—GPOs that are linked to the organizational unit that is highest in the Active Directory hierarchy are processed first, then GPOs that are linked to its child organizational unit, and so on. Finally, the GPOs that are linked to the organizational unit that contains the user or computer are processed."

Therefore, changes we make in our LGPO will not be overridden by future updates of Active Directory policies, without modifying the local policy again, that is.

TIP

Another reason to utilize a tool such as SCM for configuration templates is that the work put into the templates by Microsoft Consulting Services can be leveraged as a baseline standard for your organization as well. Literally hundreds of man-hours of experience and case work have resulted in these templates for use in a security and management perspective.

Shell customizations

In Windows XP, in the old days, one meticulously set up the desktop and Start menu layout, and then copied the profile over the default profile by hand. This was never a supported method for user customization, and over the years, some problems occurred due to its use as a standard practice.

With the Windows Vista model of user profile customization, this has changed somewhat. Questions on Windows 7, 8, 8.1, and beyond, generally boil down to topics such as how to pin an internal application to the Start menu, how to pin it to the taskbar, and how to customize the Start screen in Windows 8.

Windows 7 Start menu and taskbar

Supported methods of Windows 7 customization are well documented in several blogs and support articles by Microsoft. In this text, we'll cover these modifications and the pros and cons of the mentioned methods.

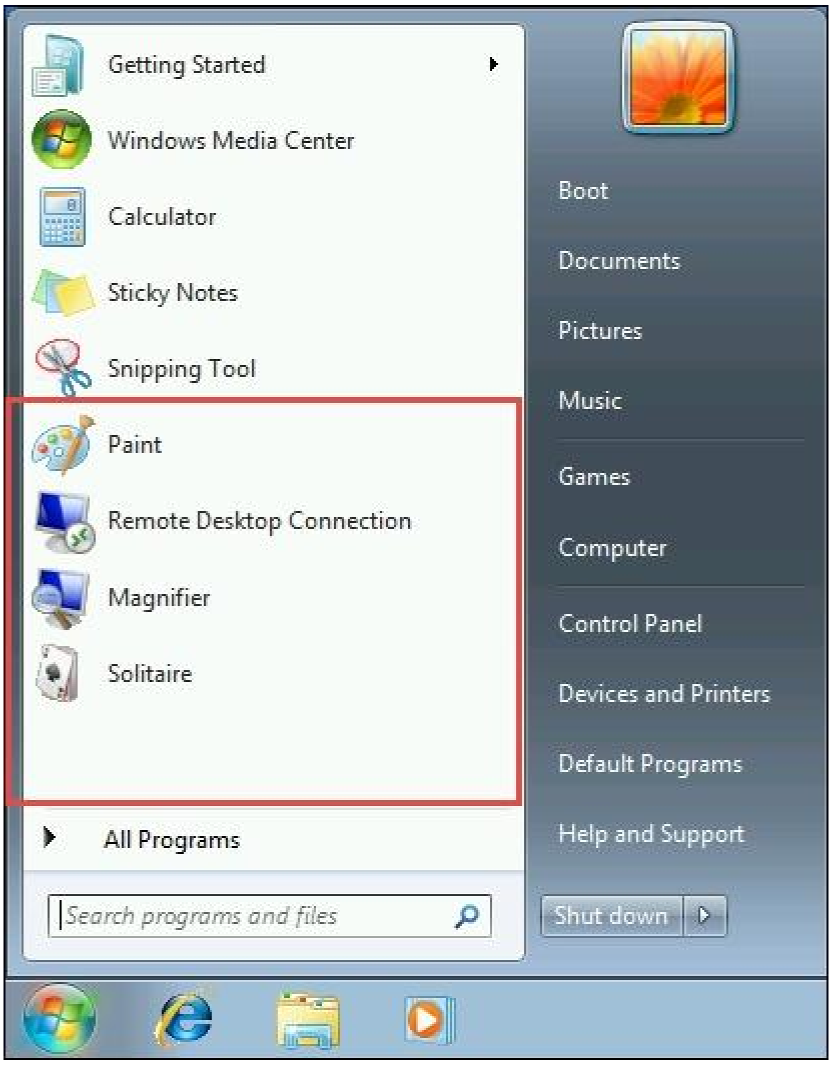
TIP

Note that the licensing of Windows 7 in some cases precludes image customization. The TechNet article at http://technet.microsoft.com/en-us/library/ff730914.aspx , in particular, discusses Windows 7 Professional and the customization of this image based on your particular licensing scenario. It is not supported in any case to take an OEM image that you get on a factory-built PC and capture and customize it. This image has already been syspreped, and therefore, could cause unpredictability in image creation and deployment, as well as violate the OEM license.

First, Scott McArthur reviewed ways of customizing the Start menu and taskbar in Windows 7 by way of `Unattend.xml`. This is a pretty straightforward method of image customization and is fully supported. The article was posted to AskCore on March 16th 2010, and is located at http://blogs.technet.com/b/askcore/archive/2010/03/16/how-to-customize-the-windows-7-start-menu-and-taskbar-using-unattend-xml.aspx .

In this blog, Scott goes over two items in `Unattend.xml`, `Microsoft-Windows-Shell-Setup\StartPanelLinks` and

Microsoft-Windows-Shell-Setup\TaskBarLinks. These settings should be placed in the oobeSystem part of the Unattend.xml file. The StartPanelLinks area is fairly straightforward; it's modifying the part of the Start menu shown in the following red box (the spot above is system-reserved):



To modify these, simply define them as links, as shown in the following example:

```
<StartPanelLinks>
<Link0>%ALLUSERSPROFILE%\Microsoft\Windows\Start
Menu\Programs\accessories\Notepad.lnk</Link0>
<Link1>%ALLUSERSPROFILE%\Microsoft\Windows\Start
Menu\Programs\accessories\Windows Explorer.lnk</Link1>
</StartPanelLinks>
```

This example will modify the menu to display Link0 as Notepad and Link1 as Windows Explorer.

To modify the TaskbarLinks area, the logic is the same:

```
<TaskbarLinks>
<Link0>%ALLUSERSPROFILE%\Microsoft\Windows\Start
Menu\Programs\accessories\Notepad.lnk</Link0>
<Link1>%ALLUSERSPROFILE%\Microsoft\Windows\Start
Menu\Programs\accessories\Windows Explorer.lnk</Link1>
</TaskbarLinks>
```

There are some caveats to this setup that one must consider:

- We need `CopyProfile=true` to be enabled in our `Unattend.xml`. This should be expected for the default profile configuration at this point.
- You cannot remove the icons in the Start menu above the red box that I detailed. If you do, they are simply recreated when a new user logs in. Microsoft does not support removing these icons in any manner.

Windows 7 background, logon screen, and user tiles

Modifying the Windows 7 logon screen is a fairly straightforward process. One simply modifies a `regkey` value, drops in a graphic, and it's done.

The `regkey` is located at `HKLM\Software\Microsoft\Windows\CurrentVersion\Authentication\LogonUI\Background`, and the value of `OEMBackground` is set to `1`.

Simply place your custom image in `%windir%\system32\oobe\info\backgrounds`. The desired logon wallpaper should have the `backgroundDefault.jpg` filename.

TIP

Images must be less than 256 KB in size. With regards to resolution of the image, set the resolution as you desire. If the monitor is set to an alternate resolution, the graphic will stretched to fit.

The Windows 7 wallpaper is pretty easy too, you can set it in the image prior to `sysprep/capture` through the normal UI; and as long as `CopyProfile=true`, it will keep it as the default setting. Another method might be to set the `HKEY_CURRENT_USER\Control Panel\Desktop` registry key value of wallpaper to the `C:\Windows\Web\Wallpaper\wallpaper.jpg` path, or even use the GPO documented at http://gpsearch.azurewebsites.net/#141 (Desktop Wallpaper User Policy).

To modify user tile to the company branding, the images are located in the following path:

- `%ProgramData%\Microsoft\User Account Pictures\Guest.bmp`
- `%ProgramData%\Microsoft\User Account Pictures\User.bmp`

However, I think that utilizing the picture of the user for the user tile is even more interesting. On a domain-joined Windows 7 machine, the user tiles are stored for the users in the `C:\ProgramData\Microsoft\User Account Pictures` path. There is an API to pull the tiles from Active Directory. So get to work!

Just kidding. Jacob Steenhagen was kind enough to create a `setUserTile.exe` utility that does the work for you. You can check it out at http://jacob.steenhagen.us/blog/2012/02/loading-a-windows-7-user-tile-using-the-picture-in-active-directory/.

You can use this tool (source documented in the blog post) to set this automatically from your Active Directory User Tile attribute that you have already populated.

Windows 8 customizations

Windows 8 brought us a new Start screen experience that changes the game somewhat with regards to image customization.

With regards to customizing the Start screen, Ben Hunter wrote (http://blogs.technet.com/b/deploymentguys/archive/2012/10/26/start-screen-customization-with-mdt.aspx):

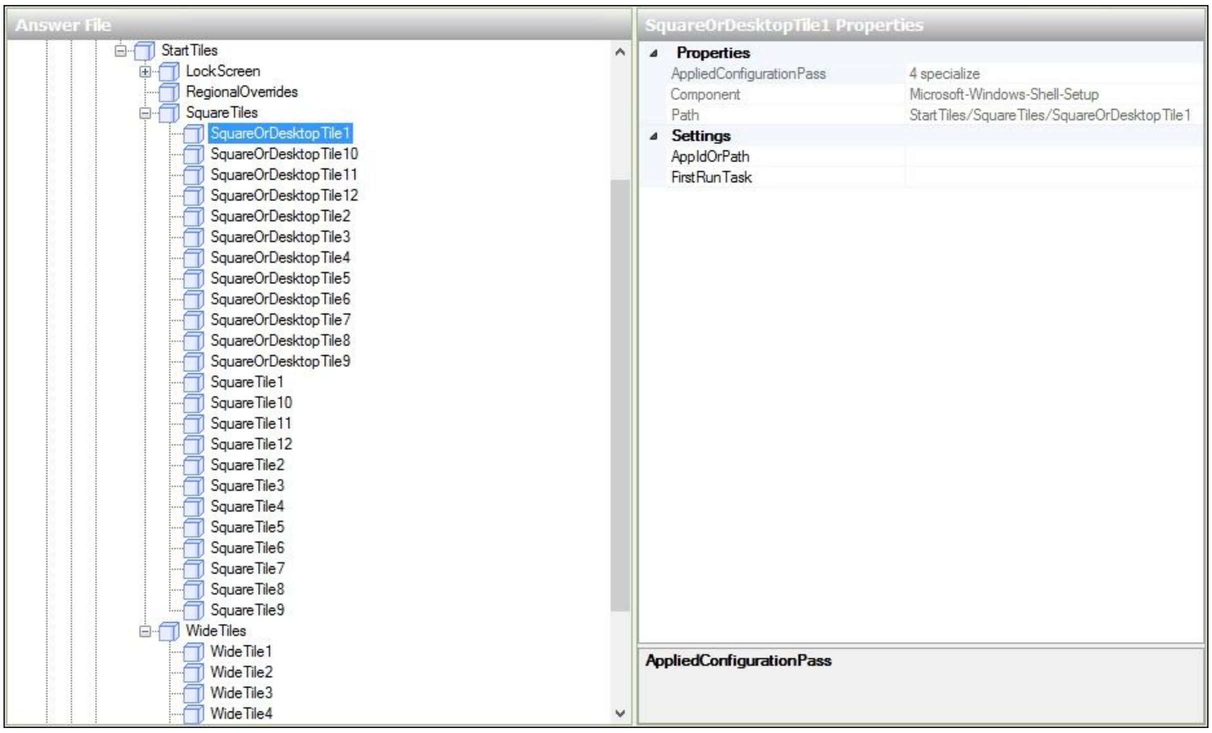
"There are three approaches that you can use:

1. *Use the Unattend.xml file to define which applications will appear in each "slot" on the Start Screen as detailed here.*
2. *Manually customize the Start Screen, then use CopyProfile to apply the customizations to the default user profile.*
3. *Manually customize the Start Screen and capture the layout file created, then copy this file to the default user during image deployment."*

We've already discussed `CopyProfile=true` to some extent in this text; the other two options are interesting, where new customization capabilities are present.

So, let's discuss using the `Unattend.xml` file to define apps and the slotting model in the `Unattend.xml` file. This will require the `AppID` key of the applications needed to be pinned by the way, and TechNet covers the how-to at http://technet.microsoft.com/en-us/library/jj134269.aspx#BKMK_StartTiles , but in essence, `Get-AppxPackage` will reveal this. Once you have your AppxPackage IDs, you can add the `StartTiles` setting to the `Unattend.xml` file with WSIM or Notepad. You will construct the layout using the Wide Tile##, Square Tile## and DesktopOrSquare Tile##.

In WSIM, these appear as follows:



```
StartTiles
  LockScreen
  RegionalOverrides
  SquareTiles
    SquareOrDesktopTile1
    SquareOrDesktopTile10
    SquareOrDesktopTile11
    SquareOrDesktopTile12
    SquareOrDesktopTile2
    SquareOrDesktopTile3
    SquareOrDesktopTile4
    SquareOrDesktopTile5
    SquareOrDesktopTile6
    SquareOrDesktopTile7
    SquareOrDesktopTile8
    SquareOrDesktopTile9
    SquareTile1
    SquareTile10
    SquareTile11
    SquareTile12
    SquareTile2
    SquareTile3
    SquareTile4
    SquareTile5
    SquareTile6
    SquareTile7
    SquareTile8
    SquareTile9
  WideTiles
    WideTile1
    WideTile2
    WideTile3
    WideTile4
```

Properties	
AppliedConfigurationPass	4 specialize
Component	Microsoft-Windows-Shell-Setup
Path	StartTiles/SquareTiles/SquareOrDesktopTile1
Settings	
AppIdOrPath	
FirstRunTask	

AppliedConfigurationPass

`AppsFolderLayout.bin` is actually discussed in the same TechNet article at http://technet.microsoft.com/en-us/library/jj134269.aspx#BKMK_AppFolder. To do this, create a user profile, set up the Start screen, and then copy this design back into your master image using **xcopy**. This works similar to the `CopyProfile=true` method, except you overwrite `AppsFolderLayout.bin` from the customized profile to the default user profile while in the audit mode. Then, all future profiles will load this customized layout.

NOTE

Windows 10

All the concepts shown in this chapter are still valid for Windows 10 image creation. In addition the following information is relevant to Windows 10 specifically:

- Windows 10 comes with a new **Windows Imaging and Configuration Designer** (**Windows ICD**). This Windows ICD is used for configuring features and policies by provisioning packages. This tool is designed specifically for Windows 10 customization and is discussed in Appendix, *Additional Enterprise Configuration Items*, of this book.
- MDT still needs modification of `Unattend.xml`. `Unattend.xml` can only be modified with **Windows System Image Manager** (**WSIM**), so you will still need WSIM.
- Currently (at the time of writing) `CopyProfile` on Windows 10 can create unexpected results. Therefore the recommendation is that you do not use `CopyProfile` with Windows 10 in your task sequences. This is subject to change with future updates to Windows 10.
- Start menu layout customization has been extended for Windows 10. It can be partially locked down. More information can be found at https://blogs.technet.microsoft.com/deploymentguys/2016/03/07/windows-10-start-layout-customization/.