

VISTA AND WINDOWS 2008 DEPLOYMENT WITH WDS-WAIK-MDT

By Jacques

I created this post to document the major steps I went through during my Vista deployment project. So it is intended as a reference for my own usage, but you might find it useful if you want to deploy Vista SP1 and/or Windows 2008 server in a fully automated way without third party tools, nor any MS-System Center Config Manager. It's all in the box or freely available for download on microsoft.com.

Credits: Sincere thanks to Guillaume Ducroix, Windows Core Support Specialist, Microsoft EMEA for his professional help on MDT.

Here are the ingredients of this recipe:

- Windows Deployment Services (WDS) - included as a role in Windows
- Windows Automated Installation Kit 1.1 (WAIK 1.1)
- Very nice to have: MS-SQL Server 2005 Express Edition with Advanced Services Service Pack 2
- Microsoft Deployment Toolkit 2008 (MDT 2008)

Do you have everything? OK, let's start cookin:

Note that this post is provided "as is" with no warranty nor supportability engagement. It does not engage my responsibility at all. Nevertheless if you consider this not clear, incomplete or inadequate, please let me know !

I. WDS: WINDOWS DEPLOYMENT SERVICES

Since we will use the MDT LiteTouch to build a hardware unspecific, but customized Vista SP1 image, to sysprep and capture it, and to multicast-deploy it to all kind of target computers, with all kind of specific drivers and applications, we will use WDS just as a PXE server to deliver the first LiteTouch WinPE boot image, plus the multicast transport of WDS to deploy the install images. So we will only use a small part of WDS. We do not tell WDS what install image to deploy, what unattended.xml file to use, nor how to name and join the target computers to the domain. This will be done by the LiteTouch client and the scripts contained in the LiteTouch PE boot image and the Task Sequence of the MDT Deployment Point.

1. Installation

On Windows 2008, just install the WDS role, including **Deployment Server** and **Transport Server**. On Windows 2003, you have to download WDS and install it.

Prerequisites include:

- The server must be a **domain member server** or a **domain controller**. The level of the domain/forest is not important.
- A **NTFS partition** must be available.

VISTA AND WINDOWS 2008 DEPLOYMENT WITH WDS-WAIK-MDT

By Jacques

- **DNS** and **DHCP** services must be available. If the DHCP service is located on the WDS server, additional configuration is necessary (see below). If the PXE client are located in a different subnet as the DHCP and WDS servers, the router must be configured with IP Helpers for both the WDS and the DHCP server. The WDS server is also a kind of DHCP server.
- To install WDS, you must be member of the **local admin group**. To run the WDS console, you must be member of the **domain users group**.

At the first launch of the WDS console, you will have to configure the server. This will create and populate a folder called **RemotInstall**. This folder is shared as **reminst**. Yes, these names come from the old RIS. It is recommended not to store this folder on the system partition, but it is OK to ignore this recommendation.

Now you will find a new MMC 3.0 console in your Administrative Tools called "Windows Deployment Services". Remember that you must be a domain user to be able to run this console. In the console, right click your new WDS server, and configure it by answering the few questions. You can change all your answers and define additional settings later in the properties of your WDS server.

If your WDS server also act as a DHCP server, you will have to edit the DHCP tab as follow:

- Check "Do not listen on port 67"
- Check "Configure DHCP option 60 to 'PXE Client'"

Do not forget to start WDS. That's it, you have now a multicast capable (only on Windows 2008), PXE-TFTP server to provide boot images to PXE clients.

2. Initial configuration and tests

Now you will review the properties of your WDS server and play with them depending on your needs and your environment. The Help (F1) should answer all your questions.

Let's shortly discuss the folders under your WDS server:

- **Install Images:** typically, the install.wim file of your Vista or Windows 2008 DVD (source folder), or your customized captured images, organized in image groups. Location on the file system: \RemotInstall\Images\ImageGroup\...When using MDT LiteTouch, you will not use this folder at all, because the images will be stored in the MDT Deployment Point.
- **Boot Images:** contains the WinPE images. This is the most important folder, because all installations of Vista start from a WinPE OS. The original WinPE (boot.wim) file is found under the source folder of the OS DVD. You can import this file by right clicking the Boot Image folder (Add Boot Image). Once imported in WDS (the file system location is \RemotInstall\Boot\architecture\Images), you can produce other type of WinPE images from this initial WinPE image by right-clicking it: you can create a **Capture Boot Image**, a customized WinPE with which you can PXE-boot a PC containing a customized, syspreped

VISTA AND WINDOWS 2008 DEPLOYMENT WITH WDS-WAIK-MDT

By Jacques

image and automatically capture it to the WDS Server. Or you can create a **Discover Boot Image**, a WinPE image containing a PXE client to put on a CD or USB Stick for old, really old PCs without PXE-capable hardware. When using MDT LiteTouch, the WinPE boot images will be the **Lite Touch Windows PE** images created by MDT and manually added here. These WinPE images contain the LiteTouch client and scripts, and they will redirect the target computers to the MDT Deployment point(s) containing the your installation and customized images.

- **Legacy Images:** Non WIM Images. Let's forget them.
- **Pending Devices:** Depending on your PXE Response Settings, (WDS server properties), your PXE-booting clients will appear here for approval and/or naming.
- **Multicast Transmission:** If you create a multicast session here, the WDMCast.exe on the WinPE client (present in the boot.wim of Vista SP1 and Windows 2008) will get the installation image using multicast. When you activate Multicast in MDT (see part iV and V), the multicast session will be created automatically by MDT! I know it seems to easy to be true, but it is! In a multicast session, a client will join and "dock" to the multicast stream at any time and get the bits being streamed. The first client start the stream, the other client can dock at any time during the stream. At the end of the stream, the WIM is retransmitted, so the later clients can get the bits they missed because they docked "underway". So it's pretty different than GhostCast, if you were used to, and pretty like AppleCast (I was told...). With multicast, the entire WIM file is transmitted to the clients, and then expanded locally. Without multicast, the WIM file is expanded on the server and then the individual files are transmitted to the clients, at least, that what I observed. Multicast is not only incredibly fast, it lets you choose the bandwidth you want to allow for deployment. Remember that multicast is not available in WDS for Windows 2003!

I created a WinRE.wim file, a WinPE containing all the recovery tools for Windows (do you remember BartPE or Winternals recovery CD?). I added my WinRE.wim to the Boot Images of my WDS Server. I also put it on a bootable USB key. Now, I can PXE boot from WinRE computers on my LAN and from my WinRE USB key on out-of-band computers. This works to fix and recover XP, Vista, Windows 2003 and Windows 2008. Pretty cool, don't you think?

You are now ready to press **F12** on y target PC and test your WDS Server.

A last tip: If you do not want to press F12 twice, go to the Boot tab of your WDS Server properties and configure the "Default boot program (optional)" with **Boot\x86\pxeboot.n12**. Put your preferred boot PE image as the Default boot image, so it will be pre-selected on the target computer.

Yet another last tip: If your target computer is x64-capable also configure both architecture x86 and x64 with **Boot\x86\pxeboot.n12**.

VISTA AND WINDOWS 2008 DEPLOYMENT WITH WDS-WAIK-MDT

By Jacques

The very very last tip: The default selection timeout value for the PXE menu is set to 30 seconds. To reduce this timeout, you can configure this value by setting the appropriate option in the default.bcd file for your client's architecture. The steps are:

- **bcdedit /store c:\RemoteInstall\Boot\architecture\default.bcd /set {bootmgr} timeout**
- Check with: **bcdedit /enum all /store c:\RemoteInstall\Boot\architecture\default.bcd**
- Force regeneration of the BCD files in the \Tmp directory by sending a control signal to the server service: **sc control wdsserver 129**

II. WAIK 1.1: WINDOWS AUTOMATED INSTALLATION KIT

The WAIK contains a collection of tools such as ImageX, System Image Manager, Package Manager, CopyPE, PEImg, help and doc files, SDK, ... Just download the .iso file, mount it and install the WAIK. I installed it on my WDS server. The WAIK is a compound of MDT, so you need to have it on your MDT computer. Thanks to MDT, you will not have to use the WAIK tools intensively, most of the work is done by the Task Sequence in MDT. But still, some configurations might be needed in the unattended.xml file or you might need to mount a wim image in order to inject drivers, to service it with packages or just to manually edit or modify files. For now, just install it.

III. MS-SQL SERVER 2005 EXPRESS SP2

If you already have an MS-SQL Server, just use it. If not, download MS-SQL Express 2005 Express SP2 and install it with the Management Studio Express feature, so you will be able to backup and restore your MDT database. Here how I installed MS-SQL Server Express:

- I created a **named instance** called "DEPLOYMENT"
- Default configuration for the Service Account (built-in system account : **Network Service**)
- **Windows Authentication Mode**
- Collation settings defaults
- Configuration Options Enable User Instances (Default)
- No Error and Usage Reporting
- With SQL Server Configuration Manager, enable the **Named Pipes** or the **TCP/IP** (or both) protocol for your DEPLOYMENT instance and **restart the SQL service**.

IV. MDT 2008: MICROSOFT DEPLOYMENT TOOLKIT 2008

VISTA AND WINDOWS 2008 DEPLOYMENT WITH WDS-WAIK-MDT

By Jacques

The Components

First you'll add all the ingredients you need for your recipes. You will add new ingredients for new recipes later, but you need to start with those needed for the first none.

1. After installation of the bits, launch the Deployment Workbench MMC3 console and review the Information Center. Under Components, in the Action pane, **Check for Updates**.
2. **Prepare a Distribution Share.** Click Distribution Share. The content pane tells you that the WAIK is already installed, and it shows how to create your Distribution Share. With MDT LiteTouch, we will use this Distribution Share (the ingredients), the Task Sequence (the recipe), the Deployment Point and the Database (the variable options, what will make your Deployment Infrastructure really flexible). So we will not use the RemoteInstall folder of WDS for all this stuff, as I explained above. Under the Actions pane, click Create Distribution Share Directory. For my small environment (200-250 clients), I installed it on C:\Distribution, but it would be a great idea to put it on a DFS share if you need a scalable solution.
3. **Import an original OS** (Vista SP1 or Windows 2008). Mount your OS DVD or iso image. Right click Operating Systems and click New. Select Full set of source files. Give the drive letter of your OS DVD or mount iso file, and Finish. The bits are copied to the Operating Systems folder of your Distribution Share. You'll have time for a coffee... When it's done, review the properties of the OS you just imported in MDT.
4. **Define your applications.** This is the container for all kind of application you wish to install during the build of your master image, or even later during specific deployments. If you have msi packages and silent install files, it's nice, but not necessary. Any interactive setup.exe will work as well. Interactive installation is not really a problem when building your master image, you do it just once. When adding specific applications during deployment, you will prefer silent installation of course.

I avoid to copy the bits of my applications to the distribution folder, so I put the path to my DFS share in the setup command line. To define an application, click New in the context menu of Applications, Select Application without source files or elsewhere on the network, (you can create bundles of interdependent applications at this stage). Fill in the details, type the install command line and the working directory. Here are some examples:

- \\FQDN\dfs\apps\msft\offent2007\setup.exe /config
\\FQDN\dfs\apps\msft\offent2007\Enterprise.WW\config.xml
- \\FQDN\dfs\apps\msft\offent2007\addins\SaveAsPDFandXPS.exe
- msixec /i \\FQDN\dfs\apps\msft\offent2007\addins\o2007pia.msi
- msixec /i \\FQDN\dfs\apps\msft\offent2007\addins\Word2007GetStartedTabSetup.msi
- \\FQDN\dfs\apps\nai\Viruscan\Enterprise\8.5\Setup.exe
- \\FQDN\dfs\apps\msft\mdop\SGAV\installers\softgrid-wd-setup.exe
- \\FQDN\dfs\apps__dell\755\R152685\setup.exe /s

VISTA AND WINDOWS 2008 DEPLOYMENT WITH WDS-WAIK-MDT

By Jacques

Note that I must use the FQDN instead of NetBIOS name for accessing my DFS root because the target computer is not yet joined to my domain when building my master image. Microsoft does not support syspreping of domain-joined computers and the LiteTouch scripts will ignore the sysprep task in this case.

Identify the applications which force a reboot. For those applications, check the option **Reboot the computer after installing this application** in the **Details** tab of the application properties. In LiteTouch, it is crucial that the scripts are aware of every reboot, or the LiteTouch sequence will be interrupted. By checking this option, you make the scripts aware of the reboot, so it can be controlled by LiteTouch. If other reboots are needed, you can easily control them with the Task Sequence (see below).

My strategy for applications: I decided to build thin images, containing just the applications I easily can maintain with my WSUS server such as Office, or which my Protection Pilot server such as McAfee VirusScan. This is for the master build image. During deployment, I also install hardware specific applications containing kernel mode drivers. That's all. All other application will be streamed as SoftGrid packages, the coolest thing in IT in the last 10 years!. The third tier of my application strategy is provided by the great seamless applications of the Terminal Server 2008, this for heavy applications like MatLab.

5. **Add OS Packages.** This is the OS online servicing part. In this container, you'll add security updates, service packs, language packs... you can browse your depositories and the .CAB and .MSU files will be added to this container. Pretty cool. If you do not want to download the security packs, you can let the task sequence perform automatic update during the installation of the target computer. Notice that online service of Vista RTM with SP1 was not possible. For SP2, servicing will be OK.
6. **Add Plug & Play in the Out-of-Box Drivers container:** This is the last cool container. As soon as you get new hardware you will download its latest drivers and add them to this container. Just expand all the drivers you need for a specific client in \\server\share\drivers\\, for example. Then click **New** in the context menu of **Out-of-Box Drivers** and browse to your driver depository for this model, in my case \\mydomain\dfs\apps\dell\755. Add a **new driver group**, e.g. Dell Optiplex 755 and select it. I also would select **Import drivers even they are duplicates of an existing driver**, so you can replace old bits by new ones. Click **Finish**. It's so easy. The MDT workbench will scan the folders and add all valid drivers in the container. If the driver packs are correctly done, the newest and most appropriate driver will be picked up during deployment. It's like if you would update your OS DVD with the newest OEM drivers. Repeat this step for every new hardware in your fleet. Short before deployment you can check for the availability of new driver versions and simply add them to this container. Unfortunately, it is not possible to keep the bits on your file server, this would be too perfect.

In order to automatically inject critical drivers such as network or mass storage drivers when generating the LiteTouchPE WIM file, I also created the drivers group NIC_PE and

VISTA AND WINDOWS 2008 DEPLOYMENT WITH WDS-WAIK-MDT

By Jacques

MassStorage_PE and I added the corresponding drivers to these groups. This is only necessary for very new drivers not included in the original WinPE of the WAIK. Without MDT you would have to offline-inject these critical drivers in the PE WIM file. MDT will do it for you! You will see how soon.

The Task Sequence to build a master image

You will now compose the recipe to automatically build a generic (hardware-independent) master vista image containing common applications. Only critical NIC or mass storage drivers will be added to this master image. All other drivers will be generic drivers provided by the Microsoft distribution. Definitive, specific drivers as well as specific applications will be provided later, during the definitive deployment. At the end of the build, the computer is syspreped and captured to the MDT Deployment share in an unjoined state. The master image can be built on any target computer. In my case, it was done on a Dell Optiplex 755.

1. Click **New** in the **Task Sequences** context menu and fill in the **General Settings**. In my case:
 - **Task sequence ID:** MAS-VistaSP1x86
 - **Task sequence name:** Vista SP x86 Master
 - **Task sequence comments:** I put the paragraph just above
2. On the **Select Template** screen, select the **Standard Client Task Sequence**
3. On the **Select OS** screen, select the OS you added in the Operating Systems container: Windows Vista ENTREPRISE in "Windows Vista\install.wim" in my case
4. On the **Specify Product Key** screen, type in your Product Key or Do not specify a product key at this time if your has a Select Distribution and if your are using a KMS to activate your OS.
5. On the **OS Settings** screen, fill in your organization name and a home page for IE.
6. On the **Admin Password** screen, type the local admin password of the target computer. You can type another password in the task sequenced used for deployment later, if you want. You also can leave it blank and give it manually during the build. Finish.

The famous **Unattended.xml** file is generated. You will find it under C:\Distribution\Control\MAS_VISTASP1X86 together with your Task Sequence **TS.xml**. Each Task sequence get its own folder under **C:\Distribution\Control**. You can have a look to the content of the other folders (Operating Systems, Packages and Out-of-Box Drivers).

You can edit the **General** properties of your first Task Sequence to specify on what platform it is allowed to run and to hide and/or disable it as until it is productive. Under **OS Info**, you can **edit the Unattended.xml** file to add features not covered by MDT; Unattended.xml will open in the **System Image Manager** tool of the WAIK. Under Task Sequence, you will add, delete, modify

VISTA AND WINDOWS 2008 DEPLOYMENT WITH WDS-WAIK-MDT

By Jacques

steps in the Task Sequence template to fulfill your needs and caprices... See what I changed or added in Unattended.xml below.

But let's first do some customizations on this Task Sequence:

- 1. Perform Windows Update.** When building your master you'd like capture an up-to-date OS, so you will update it before capturing it. There are 2 tasks that invoke the ZTIWindowsUpdate.wsf script, but the task options prevents them to run in a capture scenario. So just remove the condition **Task sequence variable DoCapture not equals YES** in the options of the **Windows Update** tasks. To be sure that forced reboots will not interrupt LiteTouch, you might add a **Restart computer** task (Add/General/Restart computer) after each Windows Update task.
- 2. Connect to DFS Share.** Remember that I install my applications from a domain-based DFS share, what is pretty difficult from a workgroup computer. I will add a step in the Task Sequence to instruct the client to connect to this share. On the Task Sequence Tab of the Task Sequence properties, I add a custom task in the State Restore section before the task "Install Applications":
 - Select the task Windows Update (Pre-Application Installation) and click Add/General/Run Command Line
 - Name your task: JL-Map DFS (JL are my initials) and give a description.
 - Command Line: `cscript.exe "%SCRIPTROOT%\zticonnect.wsf" /uncpath:\\FQDN\dfs`
 - Go on the Options tab and check the Continue on Error checkbox
- 3. Pause LiteTouch.** During the build of the master image, you might want to manually configure parts of your image before the task sequence sysprep and capture it. Here is a usefull trick to let you do that:
 - I created a JL-Pause.vbs script containing this code (Thanks to Guillaume Ducroix):

```
Set WshShell = CreateObject("Wscript.Shell")
On Error Resume Next
WshShell.Run "notepad.exe " & wscript.arguments(0) , 3 , True
Wscript.Quit(0)
```
 - I created a JLPause.txt file containing these 2 lines:

```
JLPause.txt
SysPrep will execute as soon as you close NotePad!
```
 - I saved these 2 files under the C:\Distribution\Scripts folder.
 - In the Task Sequence, under the State Restore/Capture Image section, I added a Run Command Line custom task named JL-Pause LiteTouch after the Apply Windows PE(XP)

VISTA AND WINDOWS 2008 DEPLOYMENT WITH WDS-WAIK-MDT

By Jacques

task. The command line reads:

```
cscript.exe "%SCRIPTROOT%\JLPause.vbs" "%SCRIPTROOT%\JLPause.txt"
```

- Immediately after this pause task, I force LiteTouch to restart the computer by adding a General/Restart computer custom task named JL-Restart computer.
- Immediately after this Restart computer task, I add once again a JL-Pause LiteTouch custom Command Line task.
With this trick, you have a chance to customize your master image manually as long as you want, to restart the computer for checking your customization and to correct them if necessary.

Things you cannot do within the task sequence can be done directly in the answer file **Unattended.xml** generated by the task sequence. Here are my modifications:

1. Force Swiss-German keyboard during PE

```
Components|1 windowsPE| x86_Microsoft-Windows-International-Core-WinPE__neutral:  
InputLocale = 0409:00000807
```

2. Get rid of the Network Location prompt

```
Components|specialize| x86_Microsoft-Windows-Deployment__neutral|RunSynchronous:  
Insert New RunSynchronousCommand:  
Action = AddListItem  
Description = Config Network Location  
Order = 4  
Path = cmd /c reg add "HKLM\Software\Policies\Microsoft\Windows  
NT\CurrentVersion\NetworkList\Signatures\FirstNetwork" /v Category /t REG_DWORD /d  
00000001 /f  
WillReboot = Never  
Delete the credential subsection.
```

3. Disable Automatic Update. We don't want Windows Setup to make updates and reboots in an uncontrolled way, we are using Task Sequences instead.

```
Components|oobeSystem| x86_Microsoft-Windows-Shell-Setup__neutral|OOBE:  
ProtectYourPC = 3
```

These are just some example. As you can see, you have near to unlimited possibility to configure your master image automatically.

Create a Deployment Point

The really funny part is coming now. With the deployment point and the database, you will combine the recipes (task sequences) with the ingredients (components you put in the Distribution Share) in any specific combination.

1. Click **New** in the context menu of Deployment Points

VISTA AND WINDOWS 2008 DEPLOYMENT WITH WDS-WAIK-MDT

By Jacques

2. **Choose Type:** Lab or single-server deployment
3. **Specify Deployment Point Name:**BUILDandDEPLOY
4. **Application List:** Do not allow users to select additional applications on Upgrade (you specify everything in the database)
5. **Allow Image Capture:** Do not ask if an image should be captured
6. **Allow Admin Password:** Do not ask to set the local Administrator Password
7. **Allow Product Key:** Do not ask user for a product key
8. **Network Share:** Leave the suggested names
9. **Configure User State:** Do not save data and settings.
10. **Finish**

Now review the **Properties** of your new Deployment Point:

Under the **General** tab, you can enable **Multicast**

Under the **WindowsPE** tab:

Uncheck both **ISO Images** to generate, since you will use your WDS server.

Leave the **ADO Components** checked, since you will connect to your database.

In the **Driver Injection** section, check only **Include all network drivers** in the selected group, leave the other driver types unchecked. In the **Driver group** list of values, pick the PE-NIC driver group you've created above. (If you need to inject missing mass storage drivers, check them and create a group containing them.). You also can personalize the **background** of the Windows PE screen by providing your BMP file here. Unfortunately, the LiteTouch background, which shows on the target computer during the whole task sequence cannot (yet) be personalized.

It's now time to **create your database**:

1. Click **New** in the context menu of Database
2. **Server Name:** your SQL Server
3. **Instance:** your deployment instance, in my Case: DEPLOYMENT
4. **Port:** leave empty
5. **Network Library:** Named Pipes (you can choose TCP/IP Sockets if you enabled it)
6. **Create a new database:** DeployDB
7. **SQL Share:** an existing share on your server: reminst or deployment
8. **Finish**

If you have problems to connect to the database, make sure that the network protocol you want to use is enabled and the SQL Browser service is running.

VISTA AND WINDOWS 2008 DEPLOYMENT WITH WDS-WAIK-MDT

By Jacques

Now you will define your master computer and give the rules it will receive to process the task sequence. Click **New** in the context menu of Computers.

1. Under the **Identity** tab, give a **Description** such as Dell Optiplex 755 (MASTER) and at least one of the 4 possible identifiers. I like to use the **Serial number**, in my case, the Dell tag (not the Asset tag!), XXXX63J. The LiteTouch script will use WMI to query the serial number of the target computer and query the database for getting the rules that we will define now for this specific computer.
2. Under the **Details** tab, you can define the following rules:

[Display Settings]

BitsPerPixel = 32

Vrefresh = 60

Xresolution = 1280

Yresolution = 1024

[Domain and Workgroup]

Domain Admin = *yourdomainadminaccount*

DomainAdminDomain = *yourdomainname*

DomainAdminPassword = *yourdomainadminpassword* (encrypted)

JoinDomain = *leaveempty*

JoinWorkgroup = Workgroup

MachineObjectOU = *leaveempty*

[Identification]

ComputerName = Master

FullName = *yourfullname*

OrgName = *yourorg*

[Miscellaneous]

AdminPassword = *localadminpassword* (encrypted)

BuildID = *TaskSequenceIDtoUse* (in my case: MAS_VISTASP1X86)

DoCapture = YES

DriverGroup = PE-NIC

OSInstall = YES

[Regional and Locale Settings]

KeyboardLocale = 0409:00000807 (for Swiss-German)

TimeZone = 110

TimeZoneName = W. Europe Standard Time (this value is mandatory if you want to specify TimeZone!)

UILanguage = us-EN

UserLocale=us-EN

[User Data]

ComputerBackupLocation = *yourserver*\Distribution\$\Captures

[Wizard Control]

All value = YES

VISTA AND WINDOWS 2008 DEPLOYMENT WITH WDS-WAIK-MDT

By Jacques

- Under the **Applications** tab, add all the applications bundles or the applications you want to be installed on your master, in the desired sequence
To instruct your deployment point to use the rules you defined in the database, right click it and click **Configure DB**. Only select **Query for computer-specific settings** and **Query for applications to be installed on this computer** under the **Computer Options**. Unselect all other queries in all options. When it's done, you can review your rules on the Rules tab of your deployment point properties. It should read like this:

```
[Settings]
```

```
Priority=CSettings, CApps, Default  
Properties=MyCustomProperty
```

```
[Default]
```

```
OSInstall=Y
```

```
[CSettings]
```

```
SQLServer=chase  
Instance=DEPLOYMENT  
Database=DeployDB  
Netlib=DBNMPNTW  
SQLShare=reminst  
Table=ComputerSettings  
Parameters=UUID, AssetTag, SerialNumber, MacAddress  
ParameterCondition=OR
```

```
[CApps]
```

```
SQLServer=chase  
Instance=DEPLOYMENT  
Database=DeployDB  
Netlib=DBNMPNTW  
SQLShare=reminst  
Table=ComputerApplications  
Parameters=UUID, AssetTag, SerialNumber, MacAddress  
ParameterCondition=OR  
Order=Sequence
```

These rules are stored in the **CustomsSettings.ini** file, under the **Control** folder of your Distribution share.

To fully make your build unattended, you will now edit the Bootstrap.ini file. You can do it by clicking **Edit Bootstrap.ini** on the Rules tab of your deployment point properties.

My Bootstrap.ini reads:

VISTA AND WINDOWS 2008 DEPLOYMENT WITH WDS-WAIK-MDT

By Jacques

```
[Settings]
Priority=Default
[Default]
DeployRoot=\\yourserver\Distribution$
UserDomain=yourdomainname
UserID= An domain account with RW rights on \\yourserver\Distribution$UserPassword=
userpassword
KeyboardLocale=0409:00000807 (for Swiss-German keyboard)
SkipBDDWelcome=YES
```

Note: *if you omit the KeyboardLocale, the task sequence will stop at its first task: "Gather local only"!* Both the CustomsSettings.ini and the Bootstrap.ini file are stored under the Control folder of your Distribution share. I suggest you to make a backup copy of these 2 INI files now.

Note: *You can have multiple Task Sequences per Distribution share, but only one set of CustomsSettings.ini and Bootstrap.ini files. So a Distribution Share can contain multiple original OS, multiple captured images, all applications you'd like to install during the master build or during deployment, all needed out-of-box drivers, the LiteTouch scripts and tools, different set of task sequence control files (TS.xml and Unattended.xml in their respective SequenceID subfolder). BUT: only one set of control files (CustomsSettings.ini and Bootstrap.ini) for the Deployment Point. But you do not need to create multiple deployment points. You will use the database to point the reference computer to the "Master Build" Task Sequence (MAS_VISTASP1X86), and later to point target computers to the "Deployment" Task Sequence (see part V)..*

Another Note:

I had problems accessing the deployment share from at LiteTouch initialization in PE. The issue seems to be related to the fact that the network is not ready when wpeinit run the LiteTouch process (LiteTouch.wsf). Guillaume Ducroix provided a workaround to let the script run only when the connectivity to the server is ok.

As a workaround, I modified the script C:\Distribution\Scripts\LiteTouch.wsf:

- 1. First, make a backup copy of the script*
- 2. Before the line 354 add the following lines (in the 'Find the "real" DeployRoot', just before 'Make a connection to it '):*

```
Dim sMachine
Dim bPingOK
Dim iPingDelay
Dim objPing
Dim objStatus
sMachine = "yourserver"
bPingOK = false
iPingDelay = 0
```

VISTA AND WINDOWS 2008 DEPLOYMENT WITH WDS-WAIK-MDT

By Jacques

```
Do While ((bPingOK = false) and (iPingDelay < 10))
Set objPing = GetObject("winmgmts:{impersonationLevel=impersonate}").ExecQuery("select *
from Win32_PingStatus where address = " & sMachine & """)
For Each objStatus in objPing
If IsNull(objStatus.StatusCode) or objStatus.StatusCode0 Then
oLogging.CreateEntry "Computer " & sMachine & " is not reachable, " & iPingDelay,
LogTypeInfo
bPingOK = false
iPingDelay = iPingDelay + 1
wscript.sleep(1000)
Else
bPingOK = true
End If
Next
Loop
oLogging.CreateEntry "Ping status : " & bPingOK & " " & iPingDelay, LogTypeInfo
```

3. Update your Deployment Point in order to generate a new LiteTouchPE_x86.wim

The last step will be to update your deployment point: Click Update in the context menu of your deployment point. This will generate the LiteTouchPE WIM file, and inject the PE-NIC drivers you specified on the fly. The **LiteTouchPE_x86.wim** file is located in the newly created Boot folder in the Distribution share. **C:\Distribution** should now be shared as **Distribution\$**

Open now the **WDS console**, right click **Boot Image**, click **Add Boot Image** and browse to **C:\Distribution\Boot\LiteTouchPE_x86.wim** to import the image in WDS. Now Edit the Boot properties of your WDS Server and select the **Boot\X86\images\LiteTouchPE_x86.wim** as your default boot image for your x86 and x64 architecture (as you wish to use it on x64 capable target computer as well).

You are now ready to build your first master image. So the building of your master is fully programmed, documented and can be updated and modified any time. If you want to rebuild a new master in one year, you just update your drivers and applications, modify some tasks and rules if needed, update your deployment point and rebuild a new master and re-capture it. This is much cleaner than modifying an old master and re-sysprep it.

Build and capture your master

Take a reference PC (or a virtual machine) and PXE-Boot (F12) to your WDS Server. The LiteTouchPE.wim file will be downloaded and mounted to an X RAM disk, WinPE will boot and connect to your distribution share using the information of the Bootstrap.ini file (which has been copied to the LiteTouchPE.wim). The rules specified in the CustomSettings.ini file and the database (matched against WMI query results) will be used to execute your task sequence (TS.xml and Unattended.XML). The original OS will install and all the steps of your task sequence will execute, including Windows Update, reboots, installation of applications (unattended or attended, depending on the setup commands you specified), and so on. At the end, the pauses you placed in the sequence

VISTA AND WINDOWS 2008 DEPLOYMENT WITH WDS-WAIK-MDT

By Jacques

with the Notepad trick will let you make all the customizations you want. When you close the last Notepad pause, LiteTouch will prepare for Sysprep, sysprep your machine (processing the unattended file can take quite a while). Then your machine will restart in LiteTouch PE, create a WIM file of your custom image and transfer it to your distribution share on the server in the Captures folder, under the name MINIT-NNNNNNN.wim.

You are now ready to deploy this hardware-unspecific, but highly customized Vista image to any make and model of the same architecture (x86, x64). The next part will show you how to do these last steps.

V. MDT - DEPLOYING MASTER IMAGES IMAGES

Add your captured image

Your custom image has now been captured on your deployment server under C:\Distribution\Captures. You can rename the wim file to MasterVistaSP1x86.wim if you want. Now return to the Deployment Workbench and add it to ingredients:

1. Click **New** in the context menu of Operating Systems
2. Choose **Custom image file**
3. Browse your Captures folder and select your image. You can move the wim file instead of copying it so you save space
4. Setup and Sysprep files are **not needed**
5. Chose a **Directory name** for this image under the C:\Distribution\Operating Systems, e.g. MasterVistaSP1x86
6. **Finish**

Create a deployment Task Sequence

1. Click **New** in the context menu of Task Sequences
2. Give a **Task Sequence ID**, e.g. DEPL_VISTASP1X86, a **Task Sequence Name**, e.g. Vista SP1 x86 Deployment, and a **Comment** describing your deployment scenario.
3. Choose the **Standard Client Task Sequence** template
4. **Select your captured master image**
5. **Do not specify a product key** if you will use a KMS server
6. Fill the **OS Settings** fields as usual

VISTA AND WINDOWS 2008 DEPLOYMENT WITH WDS-WAIK-MDT

By Jacques

7. Type the local **admin password** of the target computer. You can type another password in the task sequenced used for deployment later, if you want. You also can leave it blank and give it manually during the build.

8. Finish

Unattended.xml and TS.xml are generated under C:\Distribution\Control\DEPL_VISTASP1X86. You can edit the **General properties** of your new Task Sequence to specify on what platform it is allowed to run and to hide and/or disable it as until it is productive. Under **OS Info**, you can edit the Unattended.xml file.

In this new Task Sequence **DEPL_VISTASP1X86**, I added the following:

KMS configuration: Since our Key Management Server for Volume License Activation is not in my Domain, and since I did not have the SRV records of the KMS in my DNS zones, I created a task that configure the address of the KMS during deployment:

- Under State Restore, after the Apply Network Settings task, I added un General/Run Command Line task called JL-KMS Configuration
- The Command Line reads: **cscript.exe "%WINDIR%\system32\slmgr.vbs" -skms *kmsserver.xxx.ch:portnumber***
- Under the Option tab of this task, do not forget to select **Continue on error**

Connect to DFS Share: ...as in Part IV (JL-Map DFS)

Windows Update: I enabled the Windows Update (Post-Application Installation) task, as in Part IV

In **Unattended.xml** , I did the similar modification as in Part IV:

- Force Swiss-German keyboard during PE:
Components|1 windowsPE| x86_Microsoft-Windows-International-Core-WinPE__neutral_
InputLocale = 0409:00000807
- Get rid of the Network Location prompt:
Components|specialize| x86_Microsoft-Windows-Deployment__neutral_|RunSynchronous:
Insert New RunSynchronousCommand:
Action = AddListItem
Description = Config Network Location
Order = 4
Path = cmd /c reg add "HKLM\Software\Policies\Microsoft\Windows
NT\CurrentVersion\NetworkList\Signatures\FirstNetwork" /v Category /t REG_DWORD /d
00000001 /f
WillReboot = Never
Delete the credential subsection.

VISTA AND WINDOWS 2008 DEPLOYMENT WITH WDS-WAIK-MDT

By Jacques

- Get rid of disturbing autorun values:
Components|specialize|x86_Microsoft-Windows-Deployment__neutral_|RunSynchronous:
Insert New RunSynchronousCommand:
Action = AddListItem
Description = Delete AutoRun values
Order = 5
Path = cmd /c reg.exe delete "HKLM\Software\Microsoft\Windows\CurrentVersion\Run" /v
SoundMAXPnP /f
WillReboot = Never
Delete the credential subsection.

Make and Model-specific drivers and applications

You will have now to find, download, and extract the specific drivers for your makes and models. You add them to the **Out-of-Box Drivers** container as described in Part IV. Do not forget to create one driver group per make and model and to associate the drivers to the corresponding group. If you need to deploy specific applications, add them to the **Applications** container as described in Part IV.

Populate the database

In the **Make and Model** table of the Database, create a new row (New in the context menu). Here an example:

1. Identity

Make = Dell Inc.
Model = OptiPlex 755

2. Details

[Display Settings]
BitsPerPixel = 32
Vrefresh = 60
Xresolution = 1280
Yresolution = 1024

[Domain and Workgroup]
Domain Admin = yourdomainadminaccount
DomainAdminDomain = yourdomainname
DomainAdminPassword = yourdomainadminpassword (encrypted)
JoinDomain = yourdomain
JoinWorkgroup = leave empty
MachineObjectOU =
OU=OrgUnit,OU=ParentOrgUnit,OU=ParentOrgUnit,DC=yourdomain,DC=yourorg,DC=com

[Identification]
ComputerName = Opti755-%%SERIALNUMBER%

VISTA AND WINDOWS 2008 DEPLOYMENT WITH WDS-WAIK-MDT

By Jacques

FullName = yourfullname
OrgName = yourorg

[Miscellaneous]

AdminPassword = localadminpassword (encrypted)
BuildID = TaskSequenceIDtoUse(in my case: DEPL_VISTASP1X86)
DoCapture = NO
DriverGroup = Dell OptiPlex 755
Home_Page =
http://support.dell.com/support/downloads/driverslist.aspx?c=us&l=en&s=gen&ServiceTag=&SystemID=PLX_PNT_P4_755&os=WLH&osl=en&catid=&impid=
OSInstall = YES

[Regional and Locale Settings]

KeyboardLocale = 0409:00000807 or de-CH(for Swiss-German)
TimeZoneName = W. Europe Standard Time (this value is mandatory if you want to specify TimeZone!)
UILanguage = us-EN
UserLocale=us-EN

[User Data]

ComputerBackupLocation = leave empty

[Wizard Control]

All value = YES

3. Applications

Add the specific application you want to install during deployment to this make and model

This only one example on how you can play with the database. You can also distribute language packages, use the Locations or define roles. The flexibility is near unlimited.

Update the Deployment Point

Right click your BUILDandDEPLOY Deployment Point and click **ConfigureDB**. Select the ake/Model Options you've just defined in your database. The CustomSettings.ini file will now look like this one:

[Settings]

Priority=MMSetsings, MMApps, CSettings, CApps, Default
Properties=MyCustomProperty

[Default]

OSInstall=Y

[CSettings]

SQLServer=Chase

Revised March 21, 2008

VISTA AND WINDOWS 2008 DEPLOYMENT WITH WDS-WAIK-MDT

By Jacques

```
Instance=DEPLOYMENT
Database=DeployDB
Netlib=DBNMPNTW
SQLShare=SQLShare
Table=ComputerSettings
Parameters=UUID, AssetTag, SerialNumber, MacAddress
ParameterCondition=OR
```

```
[CApps]
SQLServer=Chase
Instance=DEPLOYMENT
Database=DeployDB
Netlib=DBNMPNTW
SQLShare=SQLShare
Table=ComputerApplications
Parameters=UUID, AssetTag, SerialNumber, MacAddress
ParameterCondition=OR
Order=Sequence
```

```
[MMSettings]
SQLServer=Chase
Instance=DEPLOYMENT
Database=DeployDB
Netlib=DBNMPNTW
SQLShare=SQLShare
Table=MakeModelSettings
Parameters=Make, Model
```

```
[MMApps]
SQLServer=Chase
Instance=DEPLOYMENT
Database=DeployDB
Netlib=DBNMPNTW
SQLShare=SQLShare
Table=MakeModelApplications
Parameters=Make, Model
Order=Sequence
```

You do not have to update the deployment point, unless you made change in your LiteTouchPE configuration or in the BootStrap.ini file. Both the CustomSettings.ini file and the database, as well as the Task Sequence (TS.xml and Unattended.xml) are accessed by the client directly on from the distribution share of the server.

You are ready to multicast-deploy a unique master image from a single deployment point to all the makes and models, or the locations, or the roles you have defined.

Revised March 21, 2008

Page 1 of 20

VISTA AND WINDOWS 2008 DEPLOYMENT WITH WDS-WAIK-MDT

By Jacques

Toubleshooting Tips

- **LiteTouchPE stops at "Gather local only":** Make sure that your bootstrap.ini file has all the values defined, including KeyboardLocale
- **LiteTouchPE stops at "Format C:":** If you started with a PC without any partition, but with a USB flash card reader, the first card reader will take the drive letter C: and format C: will abort. Disable USB Flexbay in your BIOS to avoid mounting a removable volume to be mounted on C:
- **Looking at the logs:** Here <http://blogs.technet.com/benhunter/archive/2007/06/16/bdd-2007-understanding-bdd-logs.aspx> an extensive list of the logs to look at...
- **Database connection problems:** KB938701 <http://support.microsoft.com/kb/938701>

Note that this post is provided "as is" with no warranty nor supportability engagement. It does not engage my responsibility at all. Nevertheless if you consider this not clear, incomplete or inadequate, please let me know !