Vladislav Sharapov

IMPLEMENTING A HYBRID NETWORK DEPLOYMENT SERVER FOR WINDOWS AND LINUX

Bachelor's Thesis
Information Technology

2016

MAMK
University of Applied Sciences

DESCRIPTION

MAMK
University of Applied Sciences

Date of the bachelor's thesis

06 May 2016

Author(s)

Vladislav Sharapov

Degree program and option

Information Technology

Name of the bachelor's thesis

Implementing a hybrid network deployment server for Windows and Linux

Abstract

The main objective of the thesis was selecting, testing and building hybrid network deployment server that should be capable of installing Windows and Linux-based operating systems. The server built to help system administrators at the Mikkeli University of Applied Sciences.

The thesis focused on the Windows operating system, because it was the main operating system used at the university. At first, different methods were sorted out by technical specifications and then tested to prove their usability.

Windows deployments are carried out using Microsoft Deployment Toolkit and Windows Assessment and Deployment Kit. The chosen strategies are High-Touch, Low-Volume and Lite-Touch, High-Volume. Linux-based operating systems are deployed with Syslinux package installed in Windows Deployment Services.

Results of the study showed that the network deployment of operating systems saves an incredible amount of time in a long run and is not incredibly difficult to setup. Additionally, the thesis shows points of attention which should be taken care of during practical implementation.

Subject headings, (keywords)

Windows, Linux, Deployment, Network

Pages	Language	URN
36	English	

Remarks, notes on appendices

Tutor	Bachelor's thesis assigned by
Matti Juutilainen	Mikkeli University of Applied Sciences

CONTENTS

TERMINOLOGY

TFTP – Trivial File Transfer Protocol - This protocol is used for transferring files over the network. Its primary advantage is a simple design which helps to maintain tiny code footprint and ease of use.

DHCP – Dynamic Host Configuration Protocol – This is protocol of automatically acquiring network information such as IP address, subnet mask, default gateway and DNS server.

DNS – Domain Name System – This is a system that associates various information with a domain name and is used to translate a human-readable domain name to a numerical IP address. Additionally, it simplifies server migration since the system administrator can change where the domain name points to.

BIOS – Basic Input Output System – This is firmware built into the memory of motherboards and used to perform hardware initialization. Hence it is the first software the computer runs on boot up.

UEFI – Unified Extensible Firmware Interface – According to Microsoft (2012), it is a newer firmware interface, designed to replace BIOS.

Image (ISO) – This is a digital copy of an optical disk. However, nowadays one can consider the image as an archive that can be written onto an optical disk or used directly for installation purposes over the network.

NFS – Network File System – This a file system standard for access to network locations.

PXE – Preboot Execution Environment – This is an industry standard client-server low-level software that allows computers to be booted and configured via the network.

WDS – Windows Deployment Services – This is a Windows Server role that allows to boot PXE clients.

MDT – Microsoft Deployment Toolkit – This is a software used for creation of automated installation sequences.

ADK – Assessment and Deployment Kit – This is a collection of different software packages that are used by WDS and MDT.

1 INTRODUCTION

Computers became part of our lives. They can help with almost any imaginable task. Their number increases every day. Computers are especially important in companies and universities; it is tough to imagine a business without the use of computers in a modern world.

Computers are pieces of hardware and they require an operating system to function. Installing it on a single computer is not a difficult task, but as the number increases, it becomes a demanding and tedious process. Operating systems can be quickly and easily installed on multiple computers simultaneously with a process called deployment.

That is why selected topic for this study is the deployment of Windows and Linux operating systems using Windows Server. The aim of this study is to implement a system that will help system administrators at the university by giving them the ability to install an operating system from the network with a high degree of automation. For the study, I will use qualitative methods aiming at better understanding of the network boot technique.

All computers at the university have Windows installed on them and because of that, I will concentrate on the advanced topics of this operating system. IT students sometimes need to use the Linux-based operating system, and to make it easier to install, I will, therefore, provide a solution for that as well.

During the theoretical part I am planning to find all possible implementation options for both operating systems. After that I will filter them out by a variety of factors such as compatibility with selected systems, difficulty of configuration and the amount of benefit in prolonged use.

During the practical part, I will test all solutions that were filtered out during the theoretical part. I will try to use them as close to a real life case as possible. Additionally, during this part, I will use virtual and physical machines with different configurations to find out, if there are some limitations on the server or the client side.

After the solutions are selected and tested, I will introduce additional topics of security and availability. This thesis will only concentrate on the deployment of operating systems and will not deal with the management of the installed operating systems.

2 OPERATING SYSTEMS AND DEPLOYMENT

An operating system is low-level software that runs on a computer and does basic functions such as managing underlying hardware and scheduling tasks. Without an operating system the computer is no more than a blinking machine making noise. With an operating system it can do almost anything. Operating systems on modern computers with the graphical user interface can be as small as 21 megabyte or industry giants containing several gigabytes of data (KolibriOS Team 2016).

The operating system consists of multiple smaller pieces of software, such as the kernel that handles all communications between hardware and software. Alongside with it work kernel-mode device drivers that interact directly with the hardware devices. Drivers can be of three types:

- Low-level – They control hardware directly with very few well-defined instructions.
- Intermediate level – They rely on the low-level drivers to control hardware with more advanced sequences of low-level instructions.
- Highest level – They rely on lower level drivers. An example of this level would be file system drivers that tell how device should be used, but do not provide exact low-level instructions.

Additionally, the operating system has a lot of small programs such as file, device, and network managers, schedulers, loggers. And on top of that, a graphical interface is built as a final highest-level piece of software.

Apart from operating systems, some basic drivers are included in the BIOS, which makes it possible to perform basic wired network connection, read and write operations on hard drives and to output graphics to the displaying device. They are not highly

sophisticated and have much lower performance than advanced drivers, but they exist to provide a starting point for computer operations.

2.1 Operating system installation

The installation is a process of making the operating system ready for the execution from the hard drive. The first step to installation is to partition the hard drive – specify which parts of the disk to use. Different operating systems take various amounts of space and require specific partitions; this will be described in more details in next paragraphs. The second step is to copy files into the partitioned drive and perform necessary configurations, such as specifying language and the time zone.

During the installation, Windows can be installed in two ways: BIOS or UEFI. BIOS is older but more compatible while UEFI is more advanced and secure. BIOS method requires two partitions: standard partition where the system is installed which is usually called disc C, and the System Reserved Partition where Boot Manager and Boot Configuration Data reside, as well as the encryption files. When Windows starts, it reads that special partition first and boots up using its configurations. If the BitLocker is activated, then Windows also reads encryption files to decrypt the main partition.

The UEFI method requires three partitions. The boot partition contains the same files as aforementioned System Reserved Partition. Microsoft Reserved Partition stores various metadata files that during BIOS partitioning were scattered across the hard drive using the hidden sectors. For example, Logical Disk Manager stores 1 MB metadata file when the disk is converted from basic to logical. Storing these files in one place at the beginning of the disk is a much better and more reliable practice than scattering. And the last partition is the standard partition where the system is installed.

Linux, on the other hand, requires at least two partitions. The swap partition is used when there is not enough RAM and the less actively used parts of it are transferred to the hard drive. The root partition is the main partition where the operating system is installed. These two partitions are enough to function, but it is recommended to divide the main partition. Boot partition may contain files that are used at the boot stage. Home partition is where user files are stored. Boot and home division is used to provide stability to the system: if the user fills up the whole drive with his files, then the

operating system will not be able to write configuration changes, and it will be impossible to boot. If the home partition is separated, then the user will not be able to fill the whole drive and the system will continue to operate normally.

2.2 Operating system deployment

Deployment is a process where system administrators install an operating system on one or more computers. It consists of a few preparation steps such as preparing the installation image and installation medium which can be of two types: network or physical storage device based. While both are used, physical installation is used only in small computer fleets. It takes the least amount of time to carry out a single deployment. However, as the number of computers increases, walking to every PC in need of the operating system becomes a huge time spender. A system administrator needs to connect a device or insert the compact disc, switch the boot method, complete necessary installation steps manually and then just wait until the installation finishes. Only after that he can go to the next computer. It will take days to deploy an operating system on the hundreds of computers in the office.

Network deployment of operating systems is a great relief for system administrators managing any number of computers. The main advantages are scalability and efficiency ratio of time spent to the number of successful deployments. An additional benefit is that a company can get rid of single-use installation media. It has one obvious requirement – an existing managed network. Network deployments are done through PXE, which is described in the next section.

2.3 PXE

Since the beginning of computer networks, there has been a search for a way to boot software images from a server located in the network. The first attempt at doing so was the Bootstrap Loading which included TFTP and Bootstrap Protocol (BOOTP) around the year 1986. BOOTP allowed the computer to find out its own and server IP address as well as the name of Network Bootstrap Program (NBP) which a computer needed to execute in order to proceed with the network boot. The problem with this method was that BOOTP only allowed limited number of connections, as it did not have any way of

redeeming used IP addresses. Additionally, it did not define standardized client side of the provisioning environment. (Intel Corporation 1999).

PXE 2.0, which is used everywhere nowadays and can be seen in Figure 1, was released in 1998, right after DHCP was developed in 1997 (Intel Corporation 1998). PXE uses the standardized combination of IP + DHCP + TFTP. DHCP allowed for better network scaling and it is more practical: servers no longer needed to have multiple NBPs for each possible device. TFTP is a fast and lightweight transport protocol that allows quick transfer of files over the network.

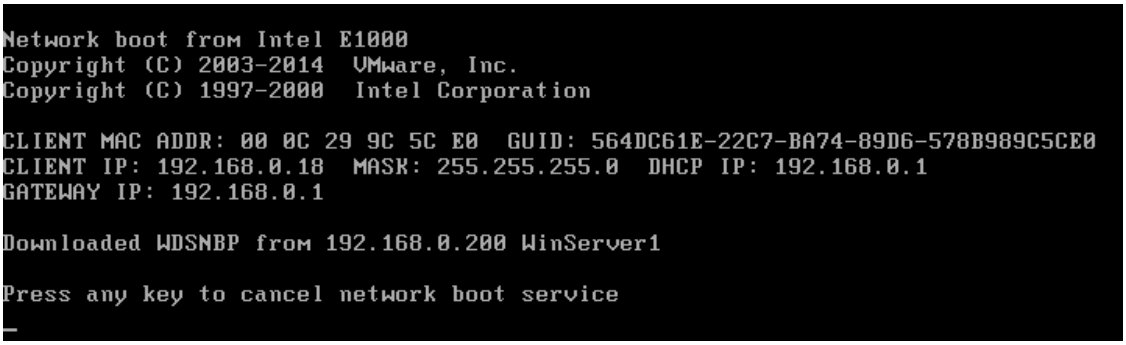

FIGURE 1. PXE 2.0

Even though PXE was published 18 years ago, it has not changed much for compatibility reasons. Moreover, because of that, any computer in that time frame is capable of booting from the network. The most recent change was made to TFTP in 2015 and is called Windowsize Option which allowed increasing TFTP throughput over high-latency network links (Internet Engineering Task Force 2015).

A considerable disadvantage of PXE is its inability to work over wireless connections. This is due to wireless communications being much more complicated than simple Ethernet connections. Wireless standards are changing a lot quicker, and so are devices and implementation methods that make the wireless connection possible. It is impossible to store every possible wireless standard protocol and wireless network card driver in a BIOS image.

Advances in computer technologies allowed to create sophisticated ultrabook and tablet computers that are so thin that they do not even have an Ethernet jack anymore. It may seem that PXE nowadays became limited to only desktop computers, servers, and full-

sized corporative notebooks. However, this is not true since all computers have at least one USB connector, which will always be installed (at least in foreseeable future). USB allows connecting vast number of peripheral devices and it even can connect an USB-Ethernet converter which enables us to execute a network boot on tablets or ultrabooks.

At this point, it is possible to advance into the world of Windows Deployment which is the most used case for today's client machines.

3 THEORY OF WINDOWS DEPLOYMENT

Microsoft provided a basic network deployment solution in Windows Server 2000 called Remote Installation Services (RIS). It was slow and difficult to manage. Many organizations decided not to use it, preferring third-party solutions, such as Symantec Ghost which is a disk cloning tool. (Finn et al. 2011, 174).

Situation changed with the release of Windows Server 2003 SP2, which introduced Windows Deployment Services (WDS). It is a full-featured network based operating system deployment solution, which makes use of boot and installation images. The WDS became very mature in Windows Server 2008 R2. It can be used as a standalone solution to allow administrators and users to install an operating system, applications and their configurations.

3.1 Deployment methods overview for Windows operating system

According to Microsoft (2009) system administrators have a choice of 4 strategies during the Windows deployment:

- High-Touch with Retail Media
- High-Touch, Low-Volume
- Lite-Touch, High-Volume
- Zero-Touch, High-Volume

High-Touch with Retail Media is a manual, hands-on deployment without the use of the network. It is a method which takes the least amount of time to prepare, but the highest

amount of time per computer. The administrator can start using it immediately with retail media, but he will need to answer all questions during installation and perform all configurations after the installation has finished. The latest step would be installing additional software.

High-Touch, Low-Volume is the first network-based deployment. During this deployment, the administrator creates a golden image of the operating system, optimized by including necessary configurations and additional software. This step takes a considerable amount of time at the beginning, but it reduces the time needed to configure each and every computer during the deployment. The administrator sets up one computer as he needs it to be and then converts it to a deployment image so that all computers will have the same configuration as the original one. However, during installation, he still needs to provide answers to all questions, although there are almost zero configuration changes required in the post-install period.

Lite-Touch, High-Volume takes it a step further than High-Touch, allowing to create task sequences for the installation. The administrator needs to spend even more time to configure the sequence, but during installation it is just a two-step process: Boot the installation image and select which sequence needs to be executed. During this deployment it is possible to use golden images, but this is not recommended. The administrator can configure sequences to install the base operating system with specific configurations and then install additional applications afterwards. It allows having only one operating system installation image, while additional software is sideloaded afterwards. The benefit is much better management of software between different user groups.

Zero-Touch, High-Volume is the highest end of automation where software manages everything. During deployment it reboots the computer into PXE, selects the needed sequence, installs the operating system using a correct answer file, does all the configurations and installs all the programs and drivers. Using this method, the administrator does not need to walk to the target computer since he can control everything remotely. This method does require an extremely sophisticated infrastructure and it is not a jack-of-all-trades. If a company receives an entirely clean computer without an operating system, it will not be possible to use the Zero-Touch

method, since the management software cannot connect to it. In this case, the administrator needs to fall back to the Lite-Touch method and boot the computer into PXE manually.

Zero-Touch will be impossible to implement at a university environment because it requires System Center Configuration Manager (SCCM) which is very expensive software. Windows Server is already running in the environment, and it is possible to use free software to carry out Lite-Touch client deployments with high-degree of automation. An overall comparison of all four deployments methods can be found in Table 1.

TABLE 1. Deployment Comparison

	Speed of multiple deployments	Preferred amount of computers	Cost
High-Touch with Retail Media	Low	<10	Low
High-Touch, Low Volume	Medium	10-50	Medium (Windows Server License)
Lite-Touch, High Volume	High	50-500	Medium (Windows Server License)
Zero-Touch, High Volume	Very High	>500	High (Windows Server + SCCM license)

3.2 Imaging the Windows operating system

Imaging is a process of installing the operating system, adding necessary configurations and software and planning how the ready image is going to be installed. It is one of the first steps in deployment – considering which type of imaging system administrator is going to use.

Thick imaging is an old approach of building images before Windows Vista, and Windows Server 2008 were developed (Microsoft 2010). This method means installing and pre-configuring operating systems with following installation of everything that end users might use and even more. It also includes all driver packages for all hardware that can be found in the environment. The resulting images are huge and even a cleanly installed system can be slow. Capturing an image and its further installation takes a very long time.

Thin imaging is at the other extreme where almost nothing is installed on the computer, and only necessary configurations are made. The administrator needs to customize each computer during deployment time. This includes the installation of applications using unattended installation packages. In the end, the image is very light, takes the lowest amount of time to capture and install, but everything other than the operating system should be installed and configured post-factum.

Hybrid is the combination of thin and thick imaging. In this method the administrator installs the operating system with applications and configurations that are needed everywhere. Additional software is supplied in separate unattended installation packages and installed afterwards, if needed. Applications that are usually supplied might include an office package, antivirus, Virtual Private Network (VPN) and voice communication software. Image size, capturing and installation time depends on the number of applications included.

3.3 Capturing the Windows operating system

Capturing is a sequence of steps to turn the installed operating system into an installation image, implemented after the administrator has built the perfect operating system following the chosen imaging strategy. System Preparation Tool – Sysprep – is a built-in Windows tool that prepares the Windows operating system to be captured and transformed into an installation image. During this process, all user-related information is removed, the whole system is cleaned up by removing temporary files and caches. According to Microsoft (2010), Sysprep can be launched up to three times due to hardcoded activation limitations.

Sysprep will reset all settings to default, so that should be carefully considered. If some settings should persist, then Unattended.xml should be created. During installation the wizard looks for that file and configures the operating system accordingly. Unattended.xml can be easily created by using Windows System Image Manager that is included in Windows Assessment and Deployment Toolkit (ADK).

ADK is a collection of tools that administrators can use to customize, assess, and deploy Windows operating systems on computers. Each Windows release has its own ADK; if the administrator is planning to maintain many operating systems, then he will need to

download ADK for each of them (Microsoft 2015). The most important tools to mention are:

- Deployment tools that include Deployment Imaging Servicing and Management (DISM) which is used for image management, and aforementioned Windows System Image Manager for the creation of Unattended.xml.
- User State Migration Tool that is used to migrate data from a previous Windows installation to a new one.
- Volume Activation Management Tool for the activation of multiple computers over the network.
- Windows Preinstallation Environment which is used to install the operating system.

After Sysprep has finished, it will shut down the computer. The next step will be booting into PXE and loading an image that is created for capturing images from the hard drive. The image will be created and can be saved on the local hard drive or transferred to Deployment Server, with the second being preferred.

Windows deployment has a lot more to it, and it heavily depends on the use case. However, the same can be said about other operating systems such as Linux, which will be discussed in the next chapter.

4 THEORY OF LINUX DEPLOYMENT

First of all, Linux is not an operating system. It is a kernel – the low-level part of the operating system. Moreover, Linux-based operating systems are called distributions. Many distributions exist, and they all are maintained by different people, companies and organizations. One can say: “How many people, so many opinions”. This phrase best describes the world of Linux distributions. However, because of a multitude of available distributions, it is impossible to test all of them in this study.

New distributions can be built by starting completely anew. This option has some uses, but it is not practical, because it will take years even for a big organization to create a

more or less working operating system. The second option is to base the distribution on someone elses work. This way the organization does not need to spend time making basic functions from scratch, but instead, concentrate on things that it wants to implement. This way the Ubuntu Studio distribution was created: the base of Ubuntu was combined with many applications for sound, photo and video editing (Canonical Ltd. 2016).

However, as someone can add functions, he can also remove them if they are not needed in the distribution. Moreover, this becomes a problem since no one knows if a specific feature is supported out of the box or not. The Smoothwall operating system is an example of the latter. It is designed to make a computer perform firewall tasks. Smoothwall was not designed to be deployed on many machines simultaneously. Hence, it only has one way of being installed – from a compact disk drive. It is hardcoded, meaning that the administrator cannot change the way it behaves. During initial research, it was possible to boot the installation program from the network or the USB flash drive, but the setup wizard would still ask to insert installation CD, because it could not find setup files.

That is why the practical part of this study will concentrate on one of the most popular, user-friendly and feature-full distribution family – Ubuntu, which is well tested and suitable for all kinds of environments. In addition to being able to install Ubuntu, it is possible to use it in the Live mode. It gives users a taste of a fully installed operating system, but without actually installing it. Additionally, administrators can check the system for driver compatibility.

4.1 Linux-based operating system deployment

The deployment of the Linux-based operating system is similar to that of Windows. The administrator needs to prepare the image and the installation sequence, but he does not need to capture the installed image. Unlike Windows Server, Ubuntu Server does not provide a deployment service of any kind as an inbuilt software.

However, because of Linux-esque modularity and openness, the administrator can use open software called FAI – Fully Automatic Installation. It is a free full-featured package, that makes possible attended and unattended network deployments and is

relatively easy to use (Lange 2016). It is completely automated with no interaction possible during the installation.

Another way is to use the Landscape, which is released by Canonical and as an alternative to Windows SCCM. It is an expensive, but extremely comprehensive tool for deployment and management of Ubuntu desktop, server and cloud. It allows to carry out all sorts of configurations and remote updates.

Another way is to combine several packages, that together will perform as a deployment service. They should include TFTP, DHCP, NFS servers, and Syslinux package which consists different bootloaders and supplementary programs. This way allows to boot standard and modified images of different operating systems.

Unfortunately, it is not possible to combine FAI or Landscape with the Windows Deployment Server. Therefore, to add ability to deploy Ubuntu, part of Syslinux package will be used and embedded into Windows, which will allow implementing the hybrid deployment server. Only that part needs to be added, since Windows Server already includes TFTP, DHCP and NFS roles. When computer will enter PXE boot and connect to a deployment server, the user will see a list of possible operating system family options.

At this point theory is known, but it is nothing without practice. Therefore, in the next chapter it will be put to good use starting with Windows deployment, following by a Linux which requires a bit of tinkering.

5 WINDOWS DEPLOYMENT

The starting point to the deployment of the Windows operating system is having the Windows Server with all the latest updates. At the time of writing Windows Server 2012 R2 is the newest version available.

5.1 Preparing for Windows Deployment

PXE makes use of TFTP and DHCP, meaning that to deploy operating system, network should be managed by the available DHCP server. During the initial stage of PXE, the computer tries to find out its own IP address. It sends a request for it to the network and receives two offers: one from the DHCP server and other from the WDS server. Then it completes the process with the DHCP server and gets an IP address. After that, it asks for information from the WDS server, which provides a boot program to download through TFTP. The deployment server can be configured with the DHCP role, but since we are adding the server to an already managed university network with an existing router, we do not need to care about it at this step – although it might need to be kept in mind for troubleshooting.

The next step is to install the WDS role and configure it. Role installation is done through Server Manager, in the Manage – Add Roles and Features Wizard, as can be seen in Figure 2. After installation succeeded, WDS should be opened for the configuration. The configuration is straightforward with only a few questions. The wizard asks if it should install WDS as part of Active Directory. The answer to this question depends on whether the network has Active Directory or not. The installation folder will contain boot and installation images, PXE boot files and management tools. It can be placed anywhere, but it is recommended not to put it in the system drive for performance reasons. And finally to which clients server should respond, which has a few options:

- Do not respond – This option is useful during initial installation and testing period when the administrator does not want anyone to be able to use the server.
- Respond only to known clients – Basically, this is a whitelist of computers that are allowed to use PXE.
- Respond to all clients – This does what it says, with a possible option for the administrator to manually approve unknown clients, which is more secure.

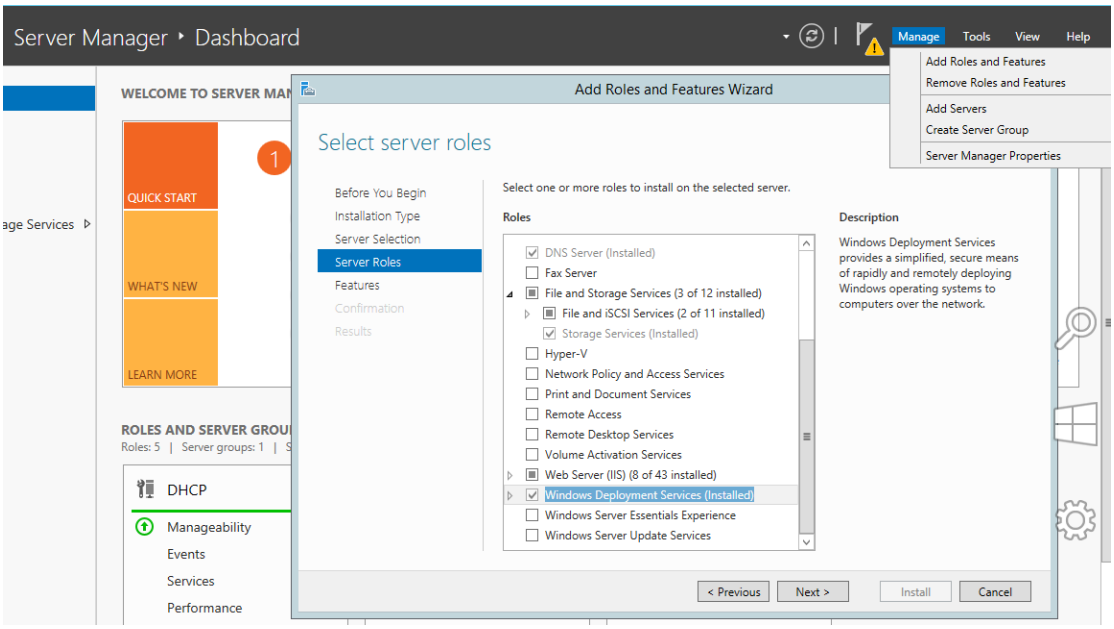

FIGURE 2. Adding roles to the server

The basic configuration is done, but there are a few things that are worth mentioning. In the properties of the WDS server it is possible to configure how known or unknown clients behave when they find a PXE server:

- Require the user to press the F12 key to continue the PXE boot – This is preferred in most cases since the computer will not end up in PXE unless a button is pressed.
- Always continue to PXE boot.
- Continue the PXE boot unless ESC key is pressed – This one preferred for the mass deployment time period.
- Never continue the PXE boot – This option can be used as a switch to disable the PXE boot temporarily.

At this point the WDS server is ready, and it should be possible to discover it on the client PC if we enter PXE boot, although it will be useless since operating systems were not added. The next step is to install Windows Assessment and Deployment Kit (ADK) and Microsoft Deployment Toolkit (MDT) from the Microsoft website to have an ability to do the Lite-Touch deployment. MDT is a comprehensive tool that uses ADK to create boot images and task sequences. ADK does not require any configuration in the scope of this study. On the other hand, in MDT a new Deployment Share should be created which will contain operating systems, drivers, and applications.

At this point the server should have all the components needed for successful deployment. Therefore, the next parts will describe the configuration of High-Touch and Lite-Touch deployment.

5.2 High-Touch, Low Volume

The first step to start High-Touch deployment is to create a capture and a boot image. At this point, the administrator should have an original image of the operating system that he wants to install. For the installation of Windows 10 Home and Pro editions can be used either the retail DVD or Media Creation Tool that can be downloaded from Microsoft website. The Enterprise edition can be acquired from Volume Licensing Service Center or downloaded from TechNet Evaluation Center. The image should then be extracted to a folder in the deployment drive.

Two main files on the original image are *boot.wim* and *install.wim* or *install.esd*. The install file contains the whole Windows system compressed to a small size and boot file contains Windows Preinstall Environment (PE). WIM is the older but more compatible format, while ESD is newer and allows for higher compression. At the time of writing, MDT and WDS are not able to read the *.esd* format. Thus it needs to be converted to *.wim*. A utility called ESD2WIM can be used for conversion, which is just a nice tool to interact with the official command line Deployment Image Servicing and Management (DISM) (Microsoft 2016).

At first, the boot WIM file should be added to the Boot Images folder in WDS. During PXE boot it allows initiating High-Touch deployment and the administrator can install the operating system located in “Install Images” folder. In the context menu of the added image a couple of options are available; the most important one is Create Capture Image. After the process is completed, the image for capturing should appear in the Boot Images folder.

The second step is to create a generalized image of the operating system. In this study a hybrid imaging will be used. The installation of the operating system on a virtual machine is preferred since it speeds up the whole process of installation, capturing and transferring. After the installation is done and the first startup configuration is

completed, additional drivers and software can be installed. For the purpose of this study, additional software will include Adobe Reader DC.

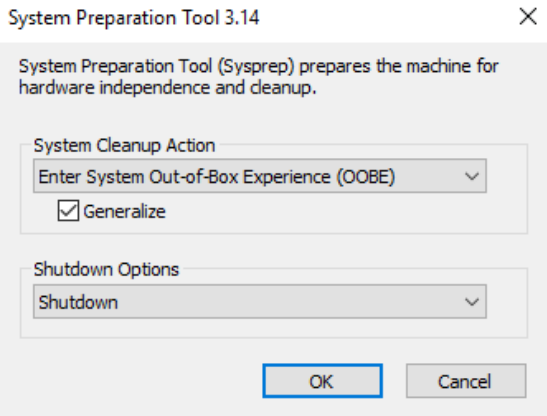

FIGURE 3. Sysprep configuration

The next step is to run Sysprep and to generalize the image. Generalization is a process of removing all hardware records, so that the image can be installed on any hardware. Without generalization the image can be only reinstalled on the machine it came from. Sysprep should shut down the machine rather than reboot to give the administrator more time to enter the PXE boot. This configuration can be seen in Figure 3. Tool is located in:

- C:/Windows/system32/Sysprep/sysprep.exe

After Sysprep has finished and the computer is turned off, the administrator needs to enter PXE and boot into the capture image which was created earlier. During configuration a few parameters should be specified:

- Image location on the host computer: The disk drive obviously should have enough space to accommodate captured image which can range from 3.5 to 50+ GB, depending on the amount of additional software.
- Which drive should be captured: If the administrator is using a virtual machine with a single drive, this parameter should have only one option.
- Should the captured image be transferred to WDS: This option should be turned on for the best results, because it transfers the image and adds it in WDS automatically speeding up the whole process. Alternatively, the target computer should be turned on, the image moved to the USB drive and brought

to the deployment server and then manually added to WDS Install Images folder.

After the whole process has succeeded, the High-Touch deployment can be executed by entering PXE, selecting the boot image and selecting an available operating system. During installation, the administrator will be asked usual install questions, but in the end all the additional software will be already installed.

High-Touch deployment has some useful optimizations compared to standard retail installation. But it can be speeded up and automated to a much higher level, which will be described in next section.

5.3 Lite-Touch, High-Volume

Lite-Touch consists of two parts: adding all possible operating systems, drivers, applications and creating new task sequences in Microsoft Deployment Toolkit, which requires Assessment and Deployment Kit.

5.3.1 Adding software

First of all, the administrator needs to import operating systems that he is going to deploy. It can be done in following three ways:

- full set of source files that can be acquired from the retail DVD or Media Creation Tool
- custom captured image file (WIM that was used for High-Touch)
- WDS images, in case the administrator is moving from High-Touch deployment to a Lite-Touch

The first option is the most advisable because of not carrying over old packages. As mentioned earlier, MDT does not support importing ESD files, and therefore it needs to be converted to WIM and placed in the same folder for further import.

Adding applications is quite a broad topic because of the number of applications available. It is possible to deploy most of them, although some might not work properly.

That is why, it is important for the administrator to do the research on his own and find out, if the application can be installed during the Lite-Touch deployment. If it is impossible, it should be installed in the operating system and captured as in the High-Touch method. However, even this method does not guarantee 100% success rate. Applications added to MDT will be displayed during installation as optional, that the administrator can select to install. Also, they can be made mandatory and hidden.

Most of the famous applications include some instructions on how it is possible to deploy their product. Some applications just require the command line parameter for silent installation. Silent means that users will not see any dialog or popups and application is installed with default parameters. More advanced applications may supply a configuration manager to customize the installation or create an answer file that will direct the installation during deployment.

Adobe Reader DC will be used as an example of application. It comes in two flavors: offline and online installer. Online installation is useful because it gathers information about the operating system and fetches the correct installation files. However, it is not possible to deploy it. Thankfully, Adobe supplies offline files for deployment on their website where the administrator needs to specify which operating system, language and version he needs. Source files should be placed in a separate folder, and the name of the executable file should be put in the command line with an additional */sAll* parameter that makes the installation silent. (Adobe 2016).

Adding drivers is relatively easy, but slightly tedious, if an enormous number of different workstations is in use. Companies and universities usually have pre-built desktop and laptop computers rather than ones built piece by piece. In this case, driver packages are easy to find on the maker's website. If a company has ten different laptops, they require ten different driver packages. MDT allows creating subfolders for drivers. Therefore, in the end, it might be a hierarchical folder structure starting from the operating system name on top, then architecture, computer make and model. Drivers can be added to a Selection Profile which groups them together and allows installing them at a later stage.

If operating systems do not work without specific updates, it is possible to download them from the Microsoft website and to add to MDT into the Packages folder. After all

software is added, it is time to start combining it into something that can be deployed, and for that a task sequence will be created.

5.3.2 Creating Task Sequence

Task sequence which can be seen in Figure 4 as a list of steps that are executed one by one during Lite-Touch deployment. The step is a specific action that runs at a specific time. Additionally, they can be grouped for better organization and versatility. Steps can have the following three states:

- Disabled
- Enabled and will stop, if an error occurs
- Enabled and will continue, if an error occurs. This is useful for non-critical steps.

When creating a task sequence, the administrator can choose between several different templates that are varying from Sysprep and capture image to install and upgrade. The standard client task sequence is used for normal client deployments. However, it also includes many steps that are not necessarily executed during every deployment, but rather when a specific condition is true.

Every step and a group of them can have set conditions, when they will run. The condition can include a preset variable that is checked at runtime, or it can be a custom command. An example of a preset variable is “IsUEFI” which determines in which mode the hard drive will be formatted – in UEFI, if the condition is true, or in BIOS, if the condition is false. The custom condition can determine the make and model of the computer and run a step that will install correct drivers.

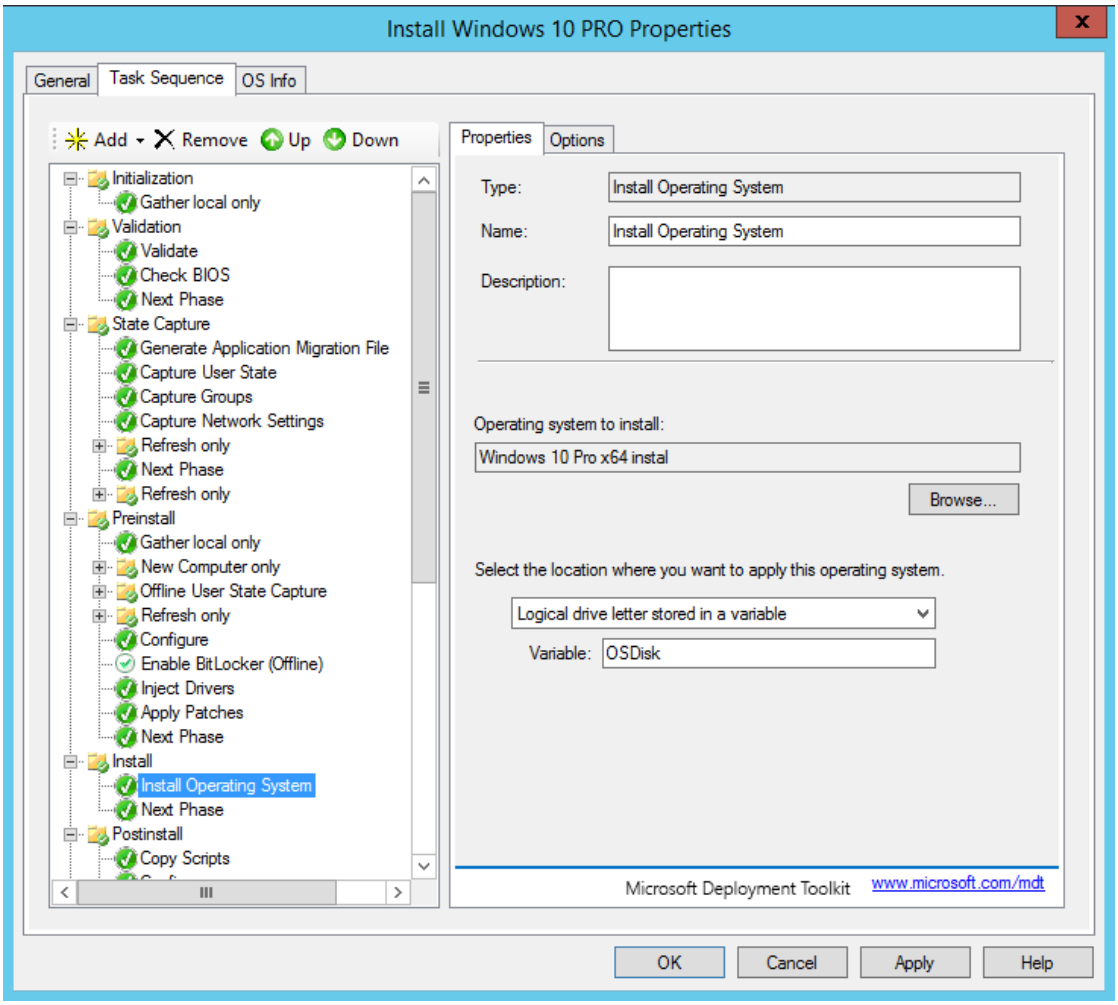

FIGURE 4. Task sequence manager

Client deployment consists of seven main groups of steps:

- Initialization – Consist of the one Gather step that collects all the necessary information and sets variables required for the deployment process.
- Validation – Consists of the steps that are used to validate, if the operating system can be installed on this hardware. It checks for the minimum RAM and processor speed.
- State Capture – Consists of the steps that capture the network and groups settings of the target computer. It runs only if operating system was previously installed.
- Preinstall – Backups the system, formats the disk, injects the drivers, applies patches or encrypts the drive.
- Install – One step that installs the selected operating system.

- Postinstall – Consists of the steps that configures the operating system, injects drivers and restarts.
- State Restore - Consists of the steps that restore previously captured settings, clean up after installation, install applications, enable encryption and even capture the whole system and send the image to WDS.

Every step and the group can be changed by the administrator, as he wants them to be. The standard task sequence is a good starting point, with only a couple of changes needed. The first is the drive formatting. By default, 500MB are reserved for the system, 99% are given to the first primary partition and the rest for system recovery. Depending on the needs this can be adjusted to create more but smaller partitions.

The second change is selecting a correct operating system that was added previously. After that the system administrator should add applications, or rather choose between displaying the application selection menu during installation or installing the specific ones that follow set conditions. By default, the step which runs Windows Update at the end of installation is disabled, but it is recommended to enable it.

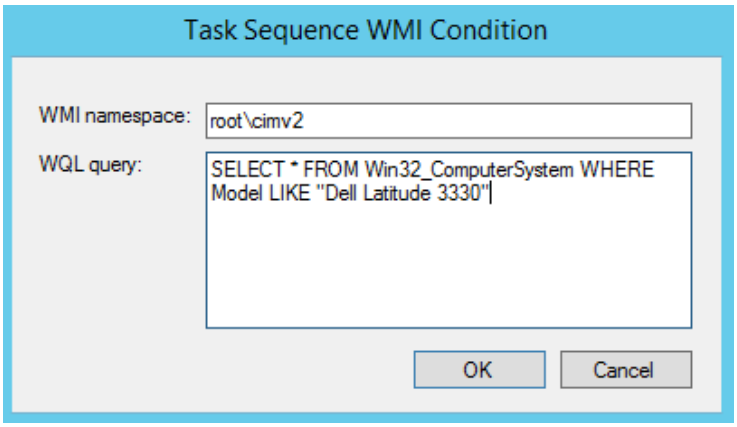

FIGURE 5. WMI query for computer model

Drivers are added at the new Inject Drivers step. For every computer model, there should be a separate step with correct Selection Profile. The computer model is checked by WMI query, as can be seen in Figure 5. Query compares the model name extracted from the system database with the name provided in quotes which should be replaced in every Inject Driver step.

After all the changes are implemented, the deployment share should be updated. It will create the Lite-Touch boot image WIM file which should be added to WDS. After booting the client computer in PXE, the menu will appear as can be seen in Figure 6, allowing to select either High-Touch or Lite-Touch Installation, as well as the image capture option. Booting into Lite-Touch will present an install wizard that will ask a few questions about backup, copying user documents and settings, and which applications to install. After the installer is finished, the client computer will reboot into the operating system and proceed with post install configurations and installing applications.

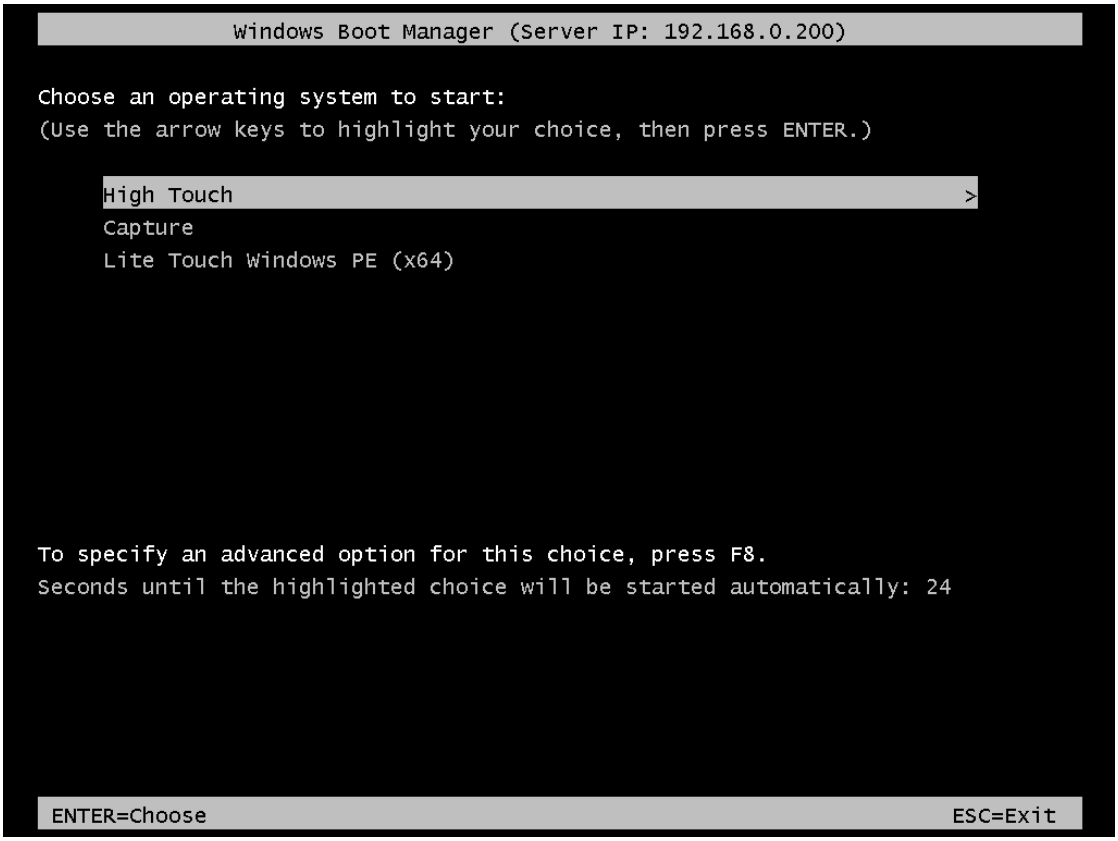

FIGURE 6. Windows Boot Manager

At this point, Lite-Touch is working nicely, but something else can be done which will speed up the deployment process even more. This will be described in the next section.

5.4 Automating Lite-Touch

During Lite-Touch installation, the wizard asks for administrator credentials for access to a deployment share and other questions that can be skipped by additional fine tuning

and pre-configuring. In properties of deployment share two files can be edited: *Bootstrap.ini* and *CustomSettings.ini*. By default, these files have a bare minimum configuration, but they can be heavily customized.

The Bootstrap file contains configurations that are applied after selecting the Lite-Touch boot image and before connection to a deployment share. Initially, it has the only parameter where share is located on the network, but it can be configured to include the domain name, administrator username, and the password. Filling everything allows connecting immediately to a deployment share without entering credentials. Any change in this file should follow with updating deployment share, and replacing WIM file in WDS.

CustomSettings file includes configurations for the Lite-Touch wizard. Default settings include options to skip some steps such as capture, backup, BitLocker, product key and local administrator password. Activating all of them reduces the number of steps to five: the configuration of the computer name, domain credentials, copying user data, time, and application selection. TechNet has an excellent article about all possible properties that can reduce the number of additional steps to zero (Microsoft 2006).

To configure the time zone the administrator should use the Microsoft Time Zone Index Values table, in the case of Finland the time zone name is FLE Standard Time. The keyboard, user locale and interface language can be configured using the standard language codes (en-us, fi-fi). Multiple keyboard layouts can be added using a semicolon. The computer name can be generated dynamically by using variables such as the computer model, unique hardware ID, if it is a laptop or a desktop, current time and many others. (Microsoft 2006).

Critical entry to the CustomSettings file is HideShell. It will not allow user interaction with the installed operating system until the wizard finishes configurations and application installations. Otherwise, the user might change some settings that will cause errors during the last steps of installation.

After all the MDT configurations are finished, it is important to remember to update the deployment share and to replace the image in WDS. Booting the target computer in PXE and selecting the Lite-Touch image should immediately display a selection of task

sequences with additional steps that the administrator decided not to skip. However, if all the steps are pre-configured and skipped, the installation will start right after the task sequence is selected. This it makes a two-touch deployment which is extremely fast compared to a standard installation. An example of before and after skipping steps can be found in Figure 7.

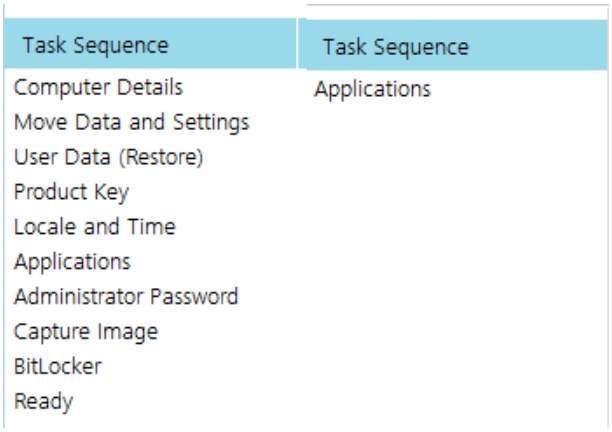

FIGURE 7. Before and after skipping steps in installation sequence

Windows is used in almost all corporations, and the number of computers that require operating system grows every hour. It is certainly worth for system administrators to spend a few hours to set the whole deployment system up, which can be used to speed up tedious installations and make them easy. The Windows deployment is supported almost out-of-the-box, with a streamlined process and a lot of official instructions. But the addition of Linux-based operating systems for special purposes will require a bit more tinkering and finesse in finding the correct solution.

6 ADDING LINUX

As mentioned during the theoretical part, it is impossible to combine two complete solutions, such as WDS and FAI, into one. Therefore, the necessary base functionality will be added into WDS to make it able of providing Linux-based operating system images.

6.1 Preparing for Linux deployment

A rerouting method will be used to add Linux into Windows Deployment Services. PXElinux will be provided for the target computer instead of Windows Boot Manager. Syslinux is a package which includes several lightweight bootloaders and one of them – PXElinux, allows booting Linux installers as well as Windows Boot Manager over the network (The Syslinux Project 2015).

The system administrator should perform following steps to add PXElinux into WDS:

- Download the latest Syslinux package (at the time of writing 6.03 is the latest).
- Unpack following files: *pxelinux.0*, *veamenu.c32*, *chain.c32.*
- Place files in the WDS Boot folder, under both x86 and x64 architectures, in case both of them are used.
- Rename *pxelinux.0* to *pxelinux.com*, *pxeboot.n12* to *pxeboot.0*, and *abortpxe.com* to *abortpxe.0*. It will make make possible interaction between PXElinux menu and Windows Boot Manager.
- Create Linux and pxeconfig.cfg folders under one or both architectures. The first folder is used to store future images and the second folder will contain configurations.

The next step is to tell WDS which files to provide to the clients when they connect through PXE. In Windows Server 2008 R2 and the later graphical menu has changed, and now it is impossible to do necessary configurations through the graphical interface. However, there is a solution to use the *wdsutil* through the command line using following commands:

- `wdsutil /set-server /bootprogram:boot\x64\pxelinux.com /architecture:x64`
- `wdsutil /set-server /N12bootprogram:boot\x64\pxelinux.com /architecture:x64`

The first command will set the boot program to PXElinux for BIOS computers and the second one will set it for UEFI computers. The next step is to make a boot menu. For that the text file named *default,* without an extension, should be created inside of the

pxelinux.cfg folder and opened with a text editor. The finished menu can be found in Figure 8. The file consists of multiple lines, where each line is a single statement. The first part contains of parameters and the second part includes all possible boot menu entries. The most important lines are:

- DEFAULT – This specifies which file to open during the load of PXElinux.
- PROMPT – If value of this parameter is 0, then menu appears immediately. If value equals to 1, then command prompt will show up for manual control.
- NOESCAPE – If this value is 1, then it is impossible to exit from this menu (to select different boot option rather than PXE).
- ALLOWOPTIONS – If this value is 1, then user can press TAB to edit boot menu, which is not recommended.
- MENU INCLUDE – This command allows to include additional files. In this case, the graphics configuration was placed in a separate file for the easier management.
- TIMEOUT – This parameter is specified in ⅒ of a second. After time runs out, default entry is automatically selected.
- MENU TITLE – This is title of the menu, which is useful to separate different boot menus.
- MENU DEFAULT – This parameter sets the entry as the first selected on boot.
- LABEL – This is internal identifier (ID) of the menu entry.
- MENU LABEL – This is title of the entry.
- KERNEL – This parameter specifies a file to load when the entry is selected.
- APPEND – This parameter specifies which additional files to load after the kernel.

The first entry in the menu points to Windows Boot Manager, which establishes a connection to the Windows Deployment Services. The second entry opens a menu with Linux distributions. The third aborts PXE and computer boots from different media if present. The fourth is the default, and it tries to boot operating system from the local hard drive.

```
DEFAULT      vesamenu.c32
PROMPT       0
NOESCAPE     0
ALLOWOPTIONS 0

MENU INCLUDE pxelinux.cfg/graphics.conf

# Timeout in units of 1/10 s
TIMEOUT 0

MENU TITLE MAMK PXE BOOT MENU

#---
LABEL wds
   MENU LABEL Windows Deployment Services
   KERNEL pxeboot.0
#---
LABEL linuxmenu
   MENU LABEL Linux Menu
   KERNEL vesamenu.c32
   APPEND pxelinux.cfg/graphics.conf pxelinux.cfg/linux.menu
#---
LABEL abort
   MENU LABEL Abort PXE
   KERNEL  abortpxe.0
#---
LABEL local
   MENU DEFAULT
   MENU LABEL Boot from Harddisk
   LOCALBOOT 0
   Type 0x80
```

FIGURE 8. Finished default menu

The menu can be configured with a lot of graphic options, but the most important parameter to mention is the screen size. By default, the resolution of the menu is 640x480 pixels, and it is the most compatible resolution. It can be increased, even up to 1920x1080. However, the problem is that it will not scale if the computer screen is smaller and the menu will not fit the screen. And on native resolution fonts will look too small. The font type can be changed, but font size cannot, thus limiting menus to low resolutions. The comparison between the default and customized graphics configuration can be found in Figure 9.

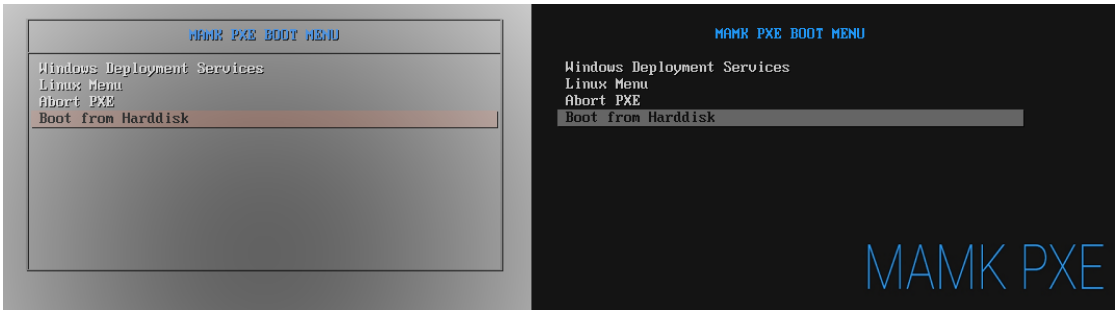

FIGURE 9. Before and after graphics configuration

At this point, the first page of the menu should be working, and it should be possible to load WDS, abort PXE or boot from hard drive. In the next part will be described the process of adding Linux distributions to the separate menu.

6.2 Adding images

First of all, the distribution image should be downloaded to the server and unpacked into the Linux folder with name that will describe which distribution it is. After that, in the *pxelinux.cfg* folder the new text file called *linux.menu* should be created. The Ubuntu entry will look like this:

- LABEL Ubuntu_15_10_x64
- MENU LABEL Ubuntu_15_10_x64
- KERNEL /Linux/Ubuntu_15_10_x64/casper/vmlinuz.efi
- APPEND boot=casper netboot=nfs
  nfsroot=192.168.0.200:/Linux/Ubuntu_15_10_x64
  initrd=/Linux/Ubuntu_15_10_x64/casper/initrd.lz quiet splash

The KERNEL statement points to unpacked distribution folder, where Linux kernel called *vmlinuz.efi* is located. In the APPEND statement specified that Ubuntu should load in Casper mode, which is a code name for the live mode. Installer will find boot files in the distribution folder in the NFS server share. And finally, it will load alternative of Windows PE called *initrd.lz*. *Quiet* and *splash* parameters are used to display Ubuntu logo during loading and to not display boot text on the screen.

After new menu and fresh entry are created, it is time to create the NFS share. To do so, a new role in the Windows Server should be added called NFS Server. After it is added, it will become possible to navigate to Linux folder, enter its properties and configure access to this folder through NFS as can be seen in Figure 10.

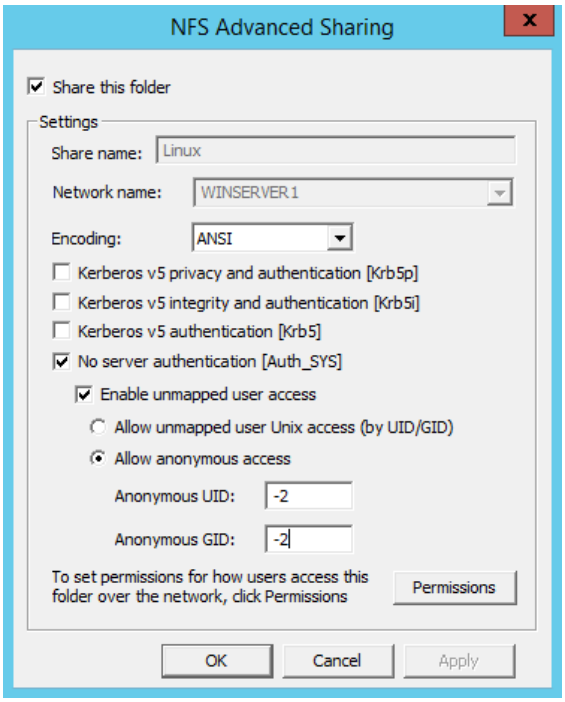

FIGURE 10. NFS Sharing

The main point of the configuration is to allow anonymous access, since it is impossible to authenticate computer during early boot. The first set of permissions is given for computers under the Permissions button. For security, only read-only access without root should be granted. The second set of permissions is provided in Share Properties in Server Manager. NTFS Permissions read and execute should be granted for Anonymous Logon.

After the NFS share is created and properly configured, it should be possible to PXE boot into new Ubuntu entry. The same method of adding new entries will work for all Ubuntu-family distributions such as Lubuntu, and also for most of the other distributions that provide bootable kernel and initrd. Unfortunately, it is not possible to install Ubuntu Server from the local network, because it only includes boot files for installation from CD and from network mirror.

In addition to the various Ubuntu distributions, Canonical provides Minimal ISO, which is a very special image. It allows users to create their own distribution on the fly. It is small (less than 40MB), and boots very quickly. It asks the user to enter basic settings such as the computer name, password, and locale and then it connects to Ubuntu Internet mirror. Next, it downloads small package of utilities to map and format the hard drive.

And then it downloads and installs the base of the operating system. At the final step it shows a list of all additional software that is possible to install such as:

- Basic Ubuntu server.
- Different roles such as DNS, mail, print, and file server.
- Full version of different Ubuntu flavors with graphical interface.
- Software for photo/video/sound editing.
- Manual selector for granular control with more than 100 000 available packages.

If the user wants some specific graphical interface or features for Ubuntu that are not provided as a separate distribution, then Minimal ISO is a perfect solution. The administrator will not need to store the whole multitude of different distributions, but just a bunch of most used ones and a Minimal ISO that will be utilized by those, who know what they need.

After all needed distributions were added to the menu and verified that everything works, it is time to take a look at the security and availability.

7 SECURITY AND TROUBLESHOOTING

Deployment server should be made available for everyone, yet only those who have access to it should be able to use it. This is where security with passwords comes in. Additionally, when deploying on multiple computers simultaneously there might come up a network bottleneck or some problems, which will be resolved in the following sections.

7.1 Securing PXE

PXE can be secured by enabling access to the WDS server only for specific computers that need an operating system at a specific moment. But it is not very useful, because it increases the number of steps during deployment: enable access, install the operating system and then disable access. And it also affects students who will not be able to use PXE to boot the operating system for laboratory work.

A better way is to protect the PXElinux menu with passwords that will be asked before allowing to boot into the specific entry. The password can be written in the menu file in MENU PASSWD line, right after MENU LABEL. It is possible to use insecure plain text passwords, which is not recommended, or secure passwords encrypted with MD5 or SHA1 algorithms. The Syslinux package includes two scripts *md5pass* and *sha1pass* that encrypt passwords with the respective algorithm. After the password is added, the user will be asked to authorize as can be seen in Figure 11 when selecting a menu entry.

FIGURE 11. Password entry

Additionally, it makes sense to protect the WDS menu. Bootstrap.ini uses unencrypted credentials for establishing a connection to a deployment share which is unsafe. Specifying an encrypted password in PXElinux menu increases overall security.

7.2 Multicast during Windows deployments

The main bottleneck of network deployments is the network throughput. Usually, local area networks have speeds of 1000Mbit/s, which is excellent for one or two computers at the same time. But if the operating system is deployed on 20 computers simultaneously, each of them receives only 50Mbit/s, and instead of ten minutes to install the operating system, it can take an hour or even more.

According to Cisco (2004) multicast is a type of network communication that allows a single node to send the same information to many nodes. Instead of sending multiple copies of the same information to each computer, the server waits until all clients have connected and then it initiates the multicast transmission, where only one copy of the information is transmitted to all computers. This drastically speeds up the deployment for multiple computers.

To activate multicast, in the properties of the deployment share the administrator needs to tick the checkbox Enable Multicast and to update the share. The multicast transmission will appear in WDS, which will become active as soon as clients start to install the operating system by using Lite-Touch method. For the High-Touch method,

the administrator needs to manually create multicast transmission by selecting which installation image to use for it.

7.3 Troubleshooting

When something does not work as it should, it needs to be troubleshooted. During this study, many hours were spent on solving various problems and below are the most important things to check.

If computers do not find the WDS server on the network, it means that the DHCP server is not configured correctly, or WDS has the wrong status of the checkbox “Do not listen to DHCP ports”. If WDS has the DHCP role, then checkbox should be enabled. If the DHCP server already exists on the network, checkbox should be disabled. Sometimes it happens, so that the WDS service does not start automatically and it should be started manually by using the following command: *wdsutil* /*start-server*.

Some computers which are running hard drives in the IDE mode can fail to boot Linux distribution images from the PXE server. During this study, it was found out that the Linux kernel progresses onto the more advanced and modern AHCI mode while becoming less compatible with IDE. This may be the reason, why computers with the IDE mode are crashing during boot.

At the beginning of the practical part Hyper-V was used as the main virtualization technology. But soon it became apparent that it was causing multiple errors when booting Linux distributions over the network. At the same time, the products of VMware were working fine.

8 CONCLUSIONS

During this study I studied the network boot technique and PXE. I have got an in-depth knowledge of Windows Deployment Server as well as Microsoft Deployment Toolkit and PXElinux. Based on all the information I gathered, selected, compiled and implemented a hybrid deployment server, which is relatively easy to setup and configure, but gives a great number of possibilities in operating systems installation.

It is impossible to have a combination of two advanced full-featured Windows and Linux deployment servers; one will be less advanced than the other. Since most of the computers at the university are running Windows, I decided to focus on it. Linux based operating systems were used much less often and only in specific circumstances. Therefore, it got a runner-up priority.

Windows can be deployed in four ways, and it appeared during the study that one almost useless in terms of increasing usability, and one was very expensive and difficult to configure because of its immense requirements. While High-Touch and Lite-Touch are both great, it certainly is much more cost-effective to implement the Lite-Touch method in a long run. It will take more time to do initial configurations, but it saves a lot of time just after a single medium-scale deployment.

TABLE 2. Efficiency comparison

	Time to install	Amount of effort during installation
Lite-Touch	20-40 minutes	Low
High-Touch	30-50 minutes	Medium
Disk Cloning	5-8 hours	High
Clean Reinstall	1-2 days	Extreme

To demonstrate time savings, I will take a simple use case at the university and the methods of how it is solved now and how it can be solved using this completed study. Currently, the system administrator at the university installs operating system and all the needed applications, and then clones the hard drive. It takes extremely long time, because cloning is done by copying every single sector on the drive. Then, if someone breaks an operating system, the master hard drive will be cloned to the target hard drive, which usually takes 5-8 hours to complete. The High-Touch method will be able to

restore the whole system in a less than an hour, and Lite-Touch is even faster. The comparison between methods can be seen in Table 2.

Linux is not that easy to deploy using Windows Deployment Server, but this was expected, since both operating systems are completely different in how they are built and supported. However, it is definitely possible, if some research is completed beforehand, and all the needed functions exist in the selected operating system.

In conclusion, I would like to say that network deployment is vital for system administrators in a modern world. It is incredibly useful, and it saves an unbelievable amount of time. There are absolutely no reasons for not using it in organizations with a large number of computers.

9 BIBLIOGRAPHY

Adobe 2016. Adobe Reader DC Enterprise. The company's WWW-pages. https://get.adobe.com/reader/enterprise/. Referred 10.02.2016.

Canonical Ltd. 2016. Ubuntu Studio. The company's WWW-pages. https://ubuntustudio.org/. Referred 20.03.2016.

Cisco 2004. IP Multicast. WWW-document. http://www.cisco.com/c/en/us/products/ios-nx-os-software/ip-multicast/index.html. Referred 02.04.2016.

Finn Aidan, Gibson Darril, Kenneth van Surksum 2011. Mastering Windows 7 Deployment. Wiley Publishing Inc.

Intel Corporation 1998. Wired for Management Baseline. PDF-document. http://download.intel.com/design/archives/wfm/downloads/base20.pdf. Updated 18.12.1998. Referred 15.03.2016.

Intel Corporation 1999. Preboot Execution Environment (PXE) Specification. PDF-document. ftp://download.intel.com/design/archives/wfm/downloads/pxespec.pdf. Referred 16.01.2016.

Internet Engineering Task Force (IETF) 2015. RFC 7440. WWW-document. https://tools.ietf.org/html/rfc7440. Updated 02.2015. Referred 03.02.2016.

KolibriOS Team 2016. KolibriOS. The company's WWW-pages. http://kolibrios.org/. Referred 15.03.2016.

Lange, T., 2016. FAI - Fully Automatic Installation. The company's WWW-pages. http://fai-project.org. Updated 31.01.2016. Referred 10.04.2016.

Microsoft 2006. Microsoft Time Zone Index Values. WWW-document. https://msdn.microsoft.com/en-us/library/ms912391%28v=winembedded.11%29.aspx. Referred 20.01.2016.

Microsoft 2006. Property Reference. WWW-document. https://technet.microsoft.com/en-us/library/bb490304.aspx. Updated 30.11.2006. Referred 20.01.2016.

Microsoft 2009. Choosing A Deployment Strategy. WWW-document. https://technet.microsoft.com/en-us/library/dd919185%28v=ws.10%29.aspx. Updated 16.09.2009. Referred 15.08.2015.

Microsoft 2010. Choosing an Image Strategy and Building Windows 7 System Images. WWW-document. https://technet.microsoft.com/en-us/library/ee956904%28v=ws.10%29.aspx. Referred 17.08.2015.

Microsoft 2010. How Sysprep Works. WWW-document. https://technet.microsoft.com/en-us/library/cc766514%28v=ws.10%29.aspx. Referred 17.08.2015.

Microsoft 2012. What is UEFI?. WWW-document. http://windows.microsoft.com/en-US/windows-8/what-uefi. Referred 19.08.2015.

Microsoft, 2015. Windows ADK. The company's WWW-pages. https://msdn.microsoft.com/en-us/windows/hardware/dn913721.aspx. Referred 26.08.2015.

Microsoft 2016. ESD2WIM. WWW-document. https://static.spiceworks.com/attachments/post/0016/8794/ESD2WIM-WIM2ESD.cmd. Referred 20.01.2016.

The Syslinux Project 2015. The Syslinux Project. The company's WWW-pages. http://www.syslinux.org/wiki/index.php?title=The_Syslinux_Project. Referred 10.08.2015.