Lite Touch Toolkit

By

Chris Miller and Thomas Moore

Submitted to
the Faculty of the Information Technology Program
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

University of Cincinnati
College of Applied Science

May 2009
Lite Touch Toolkit

by

Chris Miller and Thomas Moore

Submitted to
the Faculty of the Information Technology Program
in Partial Fulfillment of the Requirements
for
the Degree of Bachelor of Science
in Information Technology

© Copyright 2009 Chris Miller and Thomas Moore

The author grants to the Information Technology Program permission
to reproduce and distribute copies of this document in whole or in part.

Chris Miller                                        Date

Thomas Moore                                       Date

Professor Mark Stockman                           Date

Hazem Said, Ph.D. Department Head                   Date
Acknowledgements

First off, we would like to send out a huge thanks to William “Bill” Weed. Bill was instrumental in the origination of this idea, and was a great asset to our project team. We would also like to thank Professor Mark Stockman. His guidance was crucial to the completion of this project. Our regular meetings with him kept us on pace. Lastly, we would like to thank Professor Laura Wilson in the Humanities Department. Given the large amount of documentation in this project, it was crucial to have the guidance of Professor Wilson with us. We are very thankful to all that helped us to succeed. Thank you.
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acknowledgements</td>
<td>iii</td>
</tr>
<tr>
<td>Table of Contents</td>
<td>iv</td>
</tr>
<tr>
<td>List of Illustrations</td>
<td>v</td>
</tr>
<tr>
<td>Abstract</td>
<td>vi</td>
</tr>
<tr>
<td>1. Statement of the Problem</td>
<td>1</td>
</tr>
<tr>
<td>2. Solution Description</td>
<td>2</td>
</tr>
<tr>
<td>2.1 User Profiles</td>
<td>4</td>
</tr>
<tr>
<td>2.1.1 Lab Consultants</td>
<td>6</td>
</tr>
<tr>
<td>2.1.2 System Administrator</td>
<td>6</td>
</tr>
<tr>
<td>2.2 Design Protocol</td>
<td>7</td>
</tr>
<tr>
<td>2.2.1 Design protocol Discussion</td>
<td>8</td>
</tr>
<tr>
<td>3. Proof of Design</td>
<td>9</td>
</tr>
<tr>
<td>4. Objectives</td>
<td>11</td>
</tr>
<tr>
<td>5. Design and Development</td>
<td>12</td>
</tr>
<tr>
<td>5.1 Timeline</td>
<td>12</td>
</tr>
<tr>
<td>5.2 Budget</td>
<td>13</td>
</tr>
<tr>
<td>5.3 Hardware</td>
<td>15</td>
</tr>
<tr>
<td>5.4 Software</td>
<td>15</td>
</tr>
<tr>
<td>5.5 Testing</td>
<td>15</td>
</tr>
<tr>
<td>6. Risk Management</td>
<td>16</td>
</tr>
<tr>
<td>7. Conclusions and Recommendations</td>
<td>18</td>
</tr>
<tr>
<td>8. Appendix A1: Switches</td>
<td>19</td>
</tr>
<tr>
<td>9. Appendix A2: Instructions</td>
<td>24</td>
</tr>
<tr>
<td>References</td>
<td>61</td>
</tr>
</tbody>
</table>
# List of Illustrations

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>Lite Touch Toolkit Comparison to Other Brands</td>
<td>9</td>
</tr>
<tr>
<td>Figure 2</td>
<td>Use Case Diagram</td>
<td>11</td>
</tr>
<tr>
<td>Figure 3</td>
<td>System Flowchart</td>
<td>13</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Selection Screen</td>
<td>15</td>
</tr>
<tr>
<td>Figure 5</td>
<td>Application and Scripts List</td>
<td>16</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Answer File Screen</td>
<td>17</td>
</tr>
<tr>
<td>Figure 7</td>
<td>Gantt Chart</td>
<td>19</td>
</tr>
<tr>
<td>Table 1</td>
<td>Budget</td>
<td>20</td>
</tr>
<tr>
<td>Table 2</td>
<td>Risk Management</td>
<td>23</td>
</tr>
</tbody>
</table>
Abstract

The Lite Touch Toolkit (LTT) establishes a more efficient workflow in creating and deploying desktop images. This toolkit was specifically configured to be used by the IT department at The University of Cincinnati, College of Applied Science. By combining four Microsoft programs, this toolkit provided a high level of functionality at a very low cost. The Lite Touch Toolkit provides the ability for the administrator to automate the installation of an operating system, a variety of programs, drivers, and system configurations. These images that get created can be stored and managed through the LTT. Furthermore, the administrator can then create a capture of the computer and deploy a full desktop image throughout an entire lab. By creating scripts and answer files, it is possible to automate a large portion of the lengthy process. The Lite Touch Toolkit will provide a reliable alternative to other desktop deployment software and also, reduce the amount of work that is required to re-image the lab computers.
Lite Touch Toolkit

1. Statement of problem

The Department of IT budget currently licenses Symantec Ghost software. Using this software to make new images in the labs is cumbersome. This is a long process that can take weeks to accomplish. In the fall of 2008, the IT department at the College of Applied Science was looking to upgrade the software that was used to deploy images to the labs. An update of the current software would cost the IT department $5,910 (9). The project that Tom Moore and Chris Miller completed provides the same services at no cost. This project will be more efficient for the department’s five student workers. This was completed by utilizing software that was currently licensed by the IT Department and other software programs that are available from Microsoft at no cost.
2. Solution Description

The solution is a toolkit of software products that are available from Microsoft for no additional cost. These products are available to use with a licensed version of Windows operating system (1,2,3,4). Since the IT Department has already paid for the operating system (9), the toolkit was no extra charge. The proposed toolkit is comprised of four separate pieces of software. The four pieces of software are Microsoft Deployment Toolkit (MDT), Windows Automated Installation Kit (WAIK), Windows System Image Manager (WSIM), Windows Deployment Services (WDS) (1,2,3,4). Each piece of software serves a specific purpose. By utilizing this toolkit, the IT Department is able to:

a. Build images that contain an operating system
b. Deploy pre-scripted applications that install with minimal user interaction
c. Deploy system drivers
d. Deploy system updates
e. Deploy other files to further configure the operating system
f. Capture the master image and deploy it again

By providing the IT Department with this toolkit, they have saved a large amount of money, became more efficient in building images, have the ability to edit images, and will improve efficiency in the amount of time and work it takes to deploy images.

Figure 1 compares the Lite Touch Toolkit (LTT) against three other popular software companies. As shown, Lite Touch is capable of performing the same operations as the other software packages. The one area in which Lite Touch is incapable of performing is deploying heterogeneous systems. This means that Lite Touch can only deploy Microsoft operating
systems, while the other software packages are capable of deploying Windows, Linux, and Macintosh operating systems. However, Clonezilla SE (Server Edition) has shown support for Macintosh’s HFS file system.

<table>
<thead>
<tr>
<th></th>
<th>Lite Touch Toolkit</th>
<th>Norton Ghost 14</th>
<th>Novell Zenworks Configuration Management 10</th>
<th>Clonezilla SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Type</td>
<td>GUI</td>
<td>GUI</td>
<td>GUI</td>
<td>Text-Based</td>
</tr>
<tr>
<td>Supports Heterogeneous Systems</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Supports Both 32-bit and 64-bit</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Hardware Verification Check</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Multicast Deployment</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Autocast Deployment</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Silent Installation of Applications</td>
<td>√</td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Capture Images</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Total Cost</td>
<td>$0</td>
<td>$5,910</td>
<td>$9,000</td>
<td>$0</td>
</tr>
</tbody>
</table>

**Figure 1: Lite Touch Comparison to other Brands**

The main advantage of the Lite Touch Toolkit is the price. While Lite Touch is provided from Microsoft at no cost, Norton’s Ghost costs $5,910 (6,9) and Novell Zenworks Configuration Management is more expensive, costing $9,000 (9). Clonezilla SE is an open source product that can be used at no cost but contains no graphical user interface (GUI). Clonezilla is text based and the user must know commands to operate the software. This means if the user does not have an extensive technical background, that person would not be able to run Clonezilla. Figure 1 shows
that Lite Touch meets the necessary requirements and has additional features that are unavailable from the competitors.

2.1 User Profiles

There will be two users of the Lite Touch Toolkit: the lab consultants and the system administrator. Figure 2 shows a use case diagram of how the system will interact with the users.
Lab Consultant

- Upload operating system or application install files.
- Create scripts or installation files for silent installations.
- Create or modify a task sequence.
- Deploy new unattended image to a computer.
- Perform tasks which cannot be automated.
- Sysprep the computer.
- Capture final image.
- Deploy final image throughout a specific lab.
- Complete final configurations to lab computers.

System Administrator

Figure 2: Use Case Diagram
2.1.1 Lab Consultants

The user of the Lite Touch Toolkit should have some IT background.

The lab consultant should know:

   a. Basic system administration
   b. Knowledge of PC images
   c. Basic networking skills
   d. Knowledge of Windows operating systems

With this projects documentation, writing a silent install script should be relatively easy for a lab consultant to complete. The Lab consultants employed by the Department of IT have the ability of using this toolkit.

2.1.2 System Administrator

The administrator of this system will be the Information Technology Analyst for the Department of IT at the College of Applied Science. The administrator will only have to update this toolkit when new applications need to be installed. This is also a task that can be delegated to the lab consultants. The administrator of this system will have to have a larger knowledge base of system administration. The administrator should also have extensive knowledge of the software applications and operating systems used in the various labs.

Since the System Administrator is the boss, he/she will have to delegate the actions taken with the system. This persons major job function is managing the Lab Consultants use of the LTT.
2.2 Design Protocol

In order for this system to operate effectively, a specific sequence must be followed. This sequence is showcased in Figure 3 below.
2.2.1 Design Protocol Discussion

The light blue boxes in Figure 3 represent steps that are pre-configured by the System Administrator. The grey boxes are controlled by the system itself. These five steps are fully automated and do not need user interaction. The orange-colored steps represent actions that the user will have to take to complete the imaging process.

Gathering the source files and placing them into the system is time consuming. This step is only required to be completed one time. The next step is to configure the build, which compiles the operating system installation files, the application installation files, hardware drivers, and operating system packages. After configuring the build, the deployment point should be updated with a current build image. The custom boot image is sent to the deployment server, where it can be deployed to a technician’s workstation. The computer will be PXE (Pre Executable Environment) booted from the network. The PXE boot does not require the computer to have a hard drive or the use of diskettes to boot.

After the build image loads, the lab consultant will select which image should be installed. The operating systems and drivers will change depending on the lab being imaged. Selecting applications is another user input step. Finally, the lab consultant must provide a computer name and administrator password. If the password credentials are met, then Windows will begin to install. The chosen Windows operating system will be installed over the current operating system on the computer. After Windows is installed, an automated script runs, which launches software application installs. The automated script begins to silently install all the selected applications and it executes the task sequence of those applications. The task sequence is important because some software is dependent on others to run properly. Those dependent applications will be installed first.
3. Proof of Design

Desktop imaging is made simple using the Lite Touch Toolkit. Creating the original image in Microsoft Deployment Workbench will already be completed by the time this project goes to production in May, 2009. The only changes to the original image will be if new applications have to be added. Once the upload process is started, it will transfer to the target computer. Once the upload has finished, the user will be asked to select the build and operating system that is needed. There will be a selection screen that is presented to the user. This is where the user can specify which operating system to deploy. Figure 4 below shows what this selection screen looks like.

![Selection Screen](image)

Figure 4: Selection Screen

When selecting applications for the image, there is a long list. The ability to silently install applications requires each program to have a script to run. Figure 5 depicts the software
applications and the associated scripts in a list.

Figure 5: Application and Scripts List

The answer files are used to help minimize user interaction and can be created using Microsoft Windows System Image Manager (WSIM). The answer files provide the necessary information that is required by Windows, which is typically input by the user. An answer file will be created for each lab. If the same naming convention is kept for these lab images, one answer file can be kept for each lab and deployment image.

The answer file is another part that is working behind the scenes (4). Figure 6 shows the WSIM server screen as the user selects answer files to be used on each image deployment. The answer file is a file that contains the settings and configuration that is applied to a Windows image during an installation.
Figure 6: Answer File Screen

4. Objectives

The Lite Touch ToolKit includes the following deliverables:

1. Lite Touch ToolKit Documentation
   a. Instructions
   b. Explanation of Switches

2. Video Tutorial

3. Working Prototype
   a. Images for all Dept. of IT Labs
   b. Answer Files
   c. Scripts
   d. Batch Files
4. **DVD with all Installation Files**

This project delivers a new way to image lab computers for the Department of Information Technology (9). There is a working base with all currently used applications loaded up, and with unattended installation scripts. These scripts allow those applications that can be, to be fully installed without user interaction. This project team set up all necessary servers in the Department of IT server closet. This project team provided training in means of a video tutorial and proper documentation on the use and maintenance of the project. Other than a Windows Server 2008 License, this project has no cost to the department. The Windows Server 2008 License will be purchased for other uses in the lab. This Windows Server 2008 license will only have to be installed on one server (1, 9, and 10).

5. **Design and Development**

5.1 **Timeline**

The project expanded over the timeline given through Senior Design. Figure 7 displays a Gantt chart that shows tasks and dates.
<table>
<thead>
<tr>
<th>ID</th>
<th>Task Name</th>
<th>Start</th>
<th>Finish</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Research</td>
<td>1/10/2008</td>
<td>1/16/2008</td>
<td>55d</td>
</tr>
<tr>
<td>2</td>
<td>Gather Resources</td>
<td>1/10/2008</td>
<td>1/15/2008</td>
<td>51d</td>
</tr>
<tr>
<td>3</td>
<td>Obtain Requirements</td>
<td>1/10/2008</td>
<td>1/15/2008</td>
<td>47d</td>
</tr>
<tr>
<td>4</td>
<td>Write Proposals</td>
<td>1/11/2008</td>
<td>1/28/2008</td>
<td>14d</td>
</tr>
<tr>
<td>5</td>
<td>Proposal For Senior Design</td>
<td>1/12/2008</td>
<td>1/24/2008</td>
<td>0d</td>
</tr>
<tr>
<td>6</td>
<td>Setup Test Environment</td>
<td>1/12/2008</td>
<td>1/19/2008</td>
<td>6d</td>
</tr>
<tr>
<td>7</td>
<td>Internal Testing</td>
<td>1/12/2008</td>
<td>2/4/2008</td>
<td>53d</td>
</tr>
<tr>
<td>8</td>
<td>Report Problems</td>
<td>1/19/2008</td>
<td>2/13/2008</td>
<td>20d</td>
</tr>
<tr>
<td>16</td>
<td>Tech Expo Presentation</td>
<td>5/7/2009</td>
<td>5/7/2009</td>
<td>0d</td>
</tr>
</tbody>
</table>

**Figure 7: Gantt chart**

### 5.2 Budget

This budget allocated zero costs. At the start of the project, the team was informed the IT Department was going to upgrade one of the current servers to Windows Server 2008. This is the only piece of software that is not free that the project requires. Given the fact this will already be purchased by time of implementation, May 2009, this project carries no additional cost to the department. Table 1 below shows the estimated cost for this project if completed in the private sector.
<table>
<thead>
<tr>
<th>Item</th>
<th>Provided By</th>
<th>Retail Cost</th>
<th>Cost Incurred</th>
</tr>
</thead>
<tbody>
<tr>
<td>HP xw4400 Workstation x4</td>
<td>IT Department</td>
<td>$6476.00</td>
<td>$0</td>
</tr>
<tr>
<td>Linksys EG008W Switch</td>
<td>IT Department</td>
<td>$69.99</td>
<td>$0</td>
</tr>
<tr>
<td>Cat-5 Cable x 4</td>
<td>IT Department</td>
<td>$15.00</td>
<td>$0</td>
</tr>
<tr>
<td>Windows Server 2008</td>
<td>IT Department</td>
<td>$999.00</td>
<td>$0</td>
</tr>
<tr>
<td>Microsoft Deployment Toolkit</td>
<td>Microsoft</td>
<td>$0</td>
<td>$0</td>
</tr>
<tr>
<td>Microsoft Windows System Image Manager</td>
<td>Microsoft</td>
<td>$0</td>
<td>$0</td>
</tr>
<tr>
<td>Microsoft Windows Deployment Service</td>
<td>Microsoft</td>
<td>$0</td>
<td>$0</td>
</tr>
<tr>
<td>Microsoft Windows Automated Installation Kit</td>
<td>Microsoft</td>
<td>$0</td>
<td>$0</td>
</tr>
<tr>
<td>Render Soft Cam Studio</td>
<td>IT Department</td>
<td>$0</td>
<td>$0</td>
</tr>
<tr>
<td>Microsoft Office 2007</td>
<td>IT Department</td>
<td>$150.00</td>
<td>$0</td>
</tr>
<tr>
<td><strong>Retail Cost</strong></td>
<td></td>
<td><strong>$7,710.99</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Team Cost</strong></td>
<td></td>
<td><strong>$0</strong></td>
<td></td>
</tr>
</tbody>
</table>
5.3 Hardware

The hardware used was from the IT Department. There were four PC workstations used to run certain aspects of the project. Three of these PC’s function as servers. These servers run the Domain Controller, DNS, and DHCP. The last PC workstation was used as a clean workstation with no operating system. This box was PXE booted and loaded with an operating system during tests.

5.4 Software

The software was the initial Microsoft tools that were put together. The three servers ran the Microsoft Deployment Services (a part of Windows Server 2008), Windows Automated Installation Kit, Microsoft Deployment Workbench, and Windows System Image Manager. There are over 100 applications that are used between the four labs that the IT department services. All of these applications were copied into WDS for the deployment.

5.5 Testing

The testing environment that was established allowed the project team to test in both virtual machines and physical boxes. At first there were three virtual machines that were used as the hardware for the toolkit. One virtual machine was used as the domain controller. The second virtual machine ran Microsoft Deployment Workbench. The final virtual machine ran Windows Deployment Services, Windows Systems Image Manager, and Windows Automated Installation
Kit. The second and third virtual machines were condensed down to one server. While testing deployments, this project team utilized both physical hardware and virtual machines. The physical hardware is used to test drivers, x64 bit operating systems, and provides more realistic results. Moving forward, the plan is using a virtualized rollout to determine benchmarks based on the number of workstations that are being imaged. This allowed the project to show how efficient the deployment is during the design freeze presentation. For lab testing, the project team used the Networking Lab, Science 302, as a location for a real-time rollout. The lab is equipped with 25 workstations and requires an x64 bit operating system. If the test would have been unsuccessful, there was time set aside to be able to troubleshoot possible problems. However, when the test was successful, the next step involved documentation and training that will be used by the lab consultants. Imaging the network lab was a big milestone for the project, and any testing which is required afterwards will have to be completed in a virtual environment. This will ensure a stable build for the Lite Touch Toolkit. The tests gave the team good quality benchmarks for the time to upload images and it also gave the team knowledge of the applications used in the Science 302 lab.

6. Risk Management Plan

Table 2 below shows the risks the project team had to prepare for.
There are six scenarios that could occur during the course of the project which could be considered risks. For each scenario, the team described the risk which could occur, the probability of it occurring, and the actions that would be done to mitigate this risk.

<table>
<thead>
<tr>
<th>Risk</th>
<th>Level</th>
<th>Mitigation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware failure</td>
<td>High</td>
<td>Create software backups and have access to additional hardware.</td>
</tr>
<tr>
<td>Project is cancelled by UC, Dept of IT</td>
<td>Low</td>
<td>Continue to work on the project and show that the original project was able to be completed.</td>
</tr>
<tr>
<td>Failure to receive necessary equipment from Department of IT</td>
<td>Medium</td>
<td>Create a full virtualized environment for the server and testing area.</td>
</tr>
<tr>
<td>Test rollout failure</td>
<td>High</td>
<td>Troubleshoot the failure, attempt to correct the problem and determine if we are able to still have enough time or resources available to complete the test rollout.</td>
</tr>
<tr>
<td>Loss of time to complete project</td>
<td>Low</td>
<td>Scale back on the deliverables and update what will be delivered.</td>
</tr>
<tr>
<td>Hacker attempting to steal software images</td>
<td>Low</td>
<td>Using group policy and strict authentication, only the authorized users will have access to images.</td>
</tr>
</tbody>
</table>
7. Conclusions and Recommendations

The Lite Touch Toolkit will revolutionize the way lab images are created, stored, and deployed for the Department of Information Technology. The LTT will drastically shorten the time it takes to image a lab. The LTT is cost effective for the Department of IT and it will save the lab consultants employed by the department time. Once Windows Server 2008 is purchased for the lab, the team recommends a server be placed into the Server closet area and loaded up with the LTT. Given the current situation of the ghost software, it is imperative that this great toolkit be added for the Lab Consultants and the Network Administrator.
8. Appendix A1: Switches

To successfully install a program with no user interaction required, the technician must create a script, which passes the necessary parameters to the installer. By using switches, the technician is able to do tasks such as install software silently, create logs related to the installation and force a reboot of the operating system if necessary. In the section below, the switches, which were used in the Lite Touch Toolkit, are defined.

Software installers may come in different formats. Common formats are executable files (.exe) and Microsoft Installers (.msi). The format of the installer is important because it will determine what switches are required for installation. Furthermore, if an installer is InstallShield Installer, more switches are available to the lab technician to help pass parameters along. Software manufactures may also have their own set of switches to use on their applications.

Many software manufactures will provide instructions or help inside of the installer setup. This information can help a user learn what switches can be used to install the software. Silent installation help can be found by locating the install file and typing a simple command:

Setup.exe /?
Setup.exe /help

InstallerName.msi /?
InstallerName.msi /help
The help box that appears for the .MSI file is a standard help box that is included with all .MSI installers. A similar type of box would appear for an executable file, if silent installations are supported. Many of the installers that are .exe seem to be more varied on what parameters can be input. The .exe installers are typically created by the manufacture and are typically specific to that software application. Unlike .MSI installers, the .exe does not follow a standard set of scripts.

In the sections below, we cover each installer and the switches that can be applied to it. Since software manufactures can create their own switches, the list doesn’t cover all possible switches. Rather this list provides a list of common switches that tend to appear in most installers.

**Executables (.EXE)**

/Silent

Unattended mode with a progress bar displayed to user.

/VerySilent

Silent installation with no user interaction or interface displayed.
/SuppressMsgBoxes
   Instructs the installer to suppress all message boxes during installation. This switch must be combined with /Silent or /VerySilent to work properly.

/Log
   Creates log files detailing information that occurred during installation. The log files are placed in the TEMP directory.

/Log="FileName"
   Same functions are /Log, but allows the user to name the log file.

/NoCancel
   Prevents a user from cancelling an installation.

/NoRestart
   Cancels any restarts which may be necessary during an installation of a product. The reboot will be suppressed during installation.

/SaveINF= “FileName”
   Instructs installer to save installation settings to a specified INF file.

/LoadINF= “FileName”
   Instructs installer to load settings from the specified INF file, which contains installation parameters for the installer to follow.

/Dir=”C:\Folder”
   Allows the technician to specify a location for the installation to occur at

/Group= “Folder Name”
   Allows the technician to change the default folder name that is shown in the start menu.

/NoIcons
   Instructs installer to not create any icons on the desktop.

Microsoft Installation Setup (.MSI)

/i
   Instructs installer to install the specified application.

/quiet
   Quiet mode, with no user interaction or interface displayed.

/passive
   Unattended mode with a progress bar displayed to the user.
/q[parameter]  
b- basic interface  
n- no interface  
/help  
Provides tips or help to user.  
/norestart  
Cancels any restarts which may be necessary during an installation of a product. The  
reboot will be suppressed during installation.  
/promptrestart  
Prompts the user to restart if necessary.  
/forcereboot  
Forces a reboot after installation is complete.  
/l[parameter]<LogFileName>  
i- Status messages  
w- Nonfatal warnings  
e- All error messages  
a- Startup actions  
r- Action specific records  
u- Requests provided by user  
c- Initial UI parameters  
m- Fatal exit information or out of memory information  
o- Out of disk space messages  
p- Terminal properties  
v- Verbose output  
x- Extra debugging information  
+- Append information to existing log file  
!- Flush each line to the log  
*- Log all information except for v and x parameters  
/log <<LogFileName>  
Creates a general log to TEMP folder.  
/Update <Update1.msp>  
Applies updates or patches to applications.  
/AllUsers = [parameter]  
Instructs installer to apply application to all users or no users besides current.  
1- All Users  
2- No Users
InstallShield is an installer that supports the ability to record installations to a file. To do this, a user would type in a script, run the installer, and a recording of the installer is generated to a file. During this process, the installation is only recorded, the actual software isn’t installed. A new script is then written, which instructs the installer to reference the script during installation. The switches below are set to be used during an installation that includes an InstallShield installer.

-A
   Runs installation as an administrator.

-S
   Runs an InstallShield setup in silent mode. No user interaction is required or shown during this process

-R
   Instructs the installer to generate a silent installation setup file. This file contains a recording of information and parameters that are input during installation.

-SMS
   Prevents a network connection and setup.exe from closing before the installation has been completed. This is a case-sensitive switch

-Z
   Prevents installer from checking the amount of available memory during the initialization period, before the program begins to install. If the computer does not have sufficient memory, the installer will exit and quit.

-Verbose
   Provides information to the user, which details what errors occurred during an installation, if any errors did occur.

-D
   Tells the installer to run in debug mode.

-M[File]
   Instructs installer to create a Management Information Format (.MIF) file at the end of setup. A .MIF is used to describe a software application or component

-M1[Serial#]
   Instructs installer to write the specified serial number into the .MIF file.

-F[.INS Script]
   Informs the installer to follow to record an alternate .INS script during installation. If script is located in a different location than setup.exe, please provide the fully qualified network path to the script.
-F1[.ISS Script]
Informs the installer to follow or record an alternate .ISS script during installation. If script is located in a different location than setup.exe, please provide the fully qualified network path to the script.

-F2[.LOG]
Informs the installer to create a log of actions that occurred during the installation. If the user wants the log placed in a specified location, the full network place must be provided in the script.

Software manufacturers may allow the installer the ability to follow an XML file. The XML file would contain configurations, information, serial numbers, and other specifications made by the user. Much like .ISS scripts, an XML script should be specified alongside of the setup.exe inside of a script. An XML script can either be recorded from the installer or manually written by the user.

If the installer does support the ability to reference an XML file, the script below will record an installation and generate an XML based file:
Setup.exe –record “\Path\InstallationSetup.xml”

Once the file is recorded, manual changes can be completed to coding. To inform the installer to reference the XML during installation, create a new script:
Setup.exe -- silent --state “\Path\InstallationSetup.xml”

The switches or silent installation method that is used is heavily based on the installer. If help information or tips are not available from the installer, contact the manufacture to find out if a silent installation is possible with the software.

9. Appendix A2: Instructions

This part of the documentation will provide and explain detailed steps about how The Lite Touch Toolkit was configured. These directions can also be used a tutorial for the Lab Technicians to use once the toolkit is put into place. The directions will explain all necessary configurations, installations, and deployment steps that are needed to complete a successful deployment.

The first step is to install the Microsoft Deployment Workbench (MDW). If this software has not been downloaded, please click on the link below and download the software before going any further.

After MDW has been installed, launch the application by going to Start>All Programs>Microsoft Deployment>Deployment Workbench. Once the application has loaded, open the Information Center node and go to Components. The components section allows you to download a variety of additional applications, which can be used during a deployment.

At this point, the Windows Automated Installed Kit (WAIK) should be downloaded. Please download both the x86 and x64 versions. The x86 version will be utilized in the 4th floor labs, while the x64 version will be utilized in the networking lab. An Internet connection must be present in order to download software. Once the WAIK installation files have been downloaded, install the software.

Additional information can be located in the Information Center. A basic guide to getting started, documentation related to the Microsoft Deployment Workbench, and current news is all located in this section. Once WAIK is installed, go down and click on Distribution Share.

The Distribution Share is a shared folder that contains all setup files, drivers, application source files, deployment scripts, and updates. Furthermore, the distribution share screen tells the technicians which steps have been configured and which steps still need to be configured. To create a new distribution share, browse to the right side, and select “Create distribution share directory.” At this point, the lab technician will create and configure the distribution share.
Once the wizard has launched, the technician can either choose to create a new distribution share or upgrade an existing share. Since this is a new configuration, choose to create a new distribution share. The default location is placed on the C drive. Please make any necessary changes, then click finish.

Once the distribution share has been created, the next step is to create the deployment point. The deployment point is where the deployment settings are configured. This also contains the boot images that are used to begin the deployment process. To create a new deployment point, right click on the distribution share node and select “Create New Deployment Point”.

The type of distribution share should now be selected. Four options are provided to choose from, but only the first two can be used for the Lite Touch Toolkit. The first option is a Lab or single-server deployment. This option will create a new distribution share on the computer that MDW is installed on. The other option is to create a separate deployment share, on a secondary computer that the LTT can reference. The Lite Touch Toolkit uses a single-server deployment in the testing phases.
After choosing the type of deployment point, the technician should provide a name that will be assigned.

The rest of the steps in the configuration wizard should be completed. Under the Application list node, the technician should deselect the option that allows users to upgrade applications. Furthermore, deselect the option to create an image capture at the end of the deployment. The technician should provide the proper credentials and network share locations. The configure user state node can be chosen if the technician chooses to backup current files on a machine before completing the imaging process. Once all of the steps have been completed, click finish. The technician is now able to add operating systems and applications to the workbench.
Adding An Operating System:

Microsoft Deployment Workbench is compatible with XP, Vista, Server 2003 and Server 2008. To add an operating system, right click on the operating systems node and select the option to add a new operating system. Once the wizard has loaded, select option to add a full set of source files. This option will extract all files from the source directory and load the files into the distribution share. On the source option, browse the wizard to the source directory that contains the installation files.

![Operating System Wizard](image)

The additional steps will require the technician to input a name and select setup options. After all the steps have been completed, MDW will upload the operating system. Please note, this process may take a couple of minutes. Follow the same steps to add additional operating systems.

After the operating systems have been uploaded, the technician can begin to load applications. The Lite Touch Toolkit contains scripts and instructions of how to install over one hundred and thirty applications, please reference page 19 to find specific scripts. This guide will provide three separate installation examples. The three examples will cover the configuration of an executable (.exe), a Microsoft Windows installer file (.msi), and Microsoft Office products.

Adding Applications:

A large majority of the applications will contain .exe installation files. We will use Microsoft Office 2007 as a general example of how to upload and configure an application. The uploading of Microsoft Office products follows the same steps that any other product that has an
.exe installer. The big difference is the post configuration, which can be completed after the software has been uploaded.

To add a new application, right click on the applications node under the distribution share and select the option to add new applications. The application wizard will launch and prompt the options to choose the type of application to be added. If the application installation files aren’t located in any shared folder on the network select “Application with source files.” Otherwise, if the installation files are located in a share on the network select “Application without source files or elsewhere on the network.” Please note, during testing we selected “Application with source files”, but if this were being implemented into a pre-existing network, the option of “Application without source files or elsewhere on the network” would be the better option to choose. “Application with source files”, if chosen, will copy files from the source location into the distribution share. In this example, we have chosen “Application with source files”.

The details section will provide naming conventions to the application. This includes a publisher name, application name, version, and language. In this example, we are uploading Microsoft Office 2007 with English compatibility.
The source section will have the technician browse to the location where the installation files are located. If the other option was chosen a few steps back to link to the installation files, this step may not be necessary.

The command details section is where the silent installation script will be written and the working directory will be specified. If the silent installation script is unknown at this time, simply type in the file name of the installer to upload the software. Simply typing in setup.exe would allow the image to be uploaded. Once the command has been written and the working
directory is correct, click finish. The application will begin to be loaded into MDW; this may take a few minutes to upload.

At this point, if the silent installation script has been written, the technician can move on to the next application. If the script is not written, please read the part of the manual which covers switches and silent installation scripts.

We just walked through uploading Microsoft Office 2007. To further configure this software, right click on Microsoft Office 2007 and select properties. An Office Customization Tool can be used from here to further configure Microsoft Office products. This includes Office 2003, Office 2007, Visio 2007 Professional, and Project 2007 Professional. Please note, this option is not available for other Microsoft products such as Visual Studio or SQL Server. Inside of the Office Customization Tool, the technician can choose to install or select applications, input product keys, and determine what the user is able to see during the installation process. These configurations are saved to a Microsoft Installer Patch (.msp) file that is referenced by the installer during the installation process.
To upload an application that contains an .exe installer, follow the same steps that were used to upload Microsoft Office. At the “command details” page, write the silent installation script for the application. In this example, ‘iTunesSetup.exe /Quiet’ is the silent installation script that will install iTunes 7.5 with no user interaction. The script calls the executable and provides a switch that tells the installer to be quiet during the installation.

To upload an application that contains an .msi installer, once again, follow the same steps that were used to upload Microsoft Office. At the “command details” page, write the silent installation script for the application. In this example,
‘msiexec /qb /I ASPNETFutures.msi’ is the silent installation script that will install ASP.Net Futures with a simple basic user interface to show the user that the application is installing.

OS Packages is the next section. During testing, we only worked with one OS package, the Windows Service Pack 3 update. We had chosen to slipstream this update instead of having it be included in the OS Packages section.

Adding Drivers:
The Lite Touch Toolkit allows for the ability to upload and inject drivers during deployment. To upload a driver, right click on the “Out of Box Drivers” node and select the option to upload new drivers. Once the wizard has loaded, browse to the folder that contains the drivers. The driver wizard allows for groups to create categories for drivers. This is particularly useful when uploading drivers for several different images. Once the source has been located, select finish and the drivers will now be loaded.
All of the necessary files have now been uploaded and the next step is to create and configure the task sequence. The task sequence is the driving force behind the deployment. The task sequence is a series of steps that are ran on the client computer to install the operating system, applications, drivers, and other configurations. The technician can change the order of the steps inside the task sequence or add custom scripts.

To create a new task sequence, right click on the task sequence node and select the option to add a new task sequence. The task sequence wizard will prompt the technician to enter in a task sequence ID, name and comments. In this example, we are creating a task sequence for the Networking Lab. Here are the inputs the team provided to the task sequence wizard:

- Task Sequence ID: S302
- Task Sequence Name: Science 302 Networking Lab
- Task Sequence Comments: Vista x64 version for the networking lab
After providing proper identification to the task sequence, select a template for the sequence to follow. This step is important because it provides the necessary scripts, which are needed during the deployment. Five separate templates are provided:

1. Standard Client Task Sequence
2. Standard Client Replace Task Sequence
3. Custom Task Sequence
4. Lite Touch OEM Task Sequence
5. Standard Server Task Sequence

The standard client task sequence and the standard client replace task sequence are the two templates used during the Lite Touch deployment. The standard client task sequence should be used when deploying to a computer that doesn’t currently contain an operating system. If the computer does contain an operating system, the standard client replace task sequence provides scripts to remove the current operating system and install a new image. In this example, a standard client task sequence was used.

After selecting a template, the wizard will have the technician specify the operating system. This is important when installing an operating system that has multiple versions. The technician should then fill the remaining pages in with the proper credentials, and then click finish. The task sequence has been created and can now be further configured by the technician. To change the task sequence steps, right click on the task sequence that was just created and choose properties. A window with information related to the task sequence will appear. The general tab displays the name and comments related to the task sequence. The technician has the ability to specify a client platform for the task sequence to run on. The task sequence can be enabled and disabled in this section as well.
After reviewing the information and completing configurations on the general page, click on the “task sequence tab.” The task sequence tab brings up the ordered list in which the scripts and commands will run. The commands are broken down into seven separate sections

1. Initialization
2. Validation
3. State Capture
4. Pre-install
5. Install
6. Post-install
7. State Restore

The technician has the ability to add scripts or commands into any of these seven sections, if desired. Inside of each section, multiple steps are placed in order to pass information of what task to do to the computer. In the example below, a script to disable the user access control function, in Windows Vista, has been placed in the state restore section.
After the technician is finished, the next task to complete is to create the image through the deployment point. Earlier in this tutorial, the technician created a deployment point. The deployment point is the shared folder, which contains all necessary operating system source files, application source files, drivers, scripts, and boot images.

By right clicking on the deployment point and selecting properties, the deployment point properties will appear. Inside the properties window, the technician will find the name of the deployment point, the network path, and local path to the deployment point, and what platforms are supported. In this example, both x86 and x64 platforms are being supported.
The rules tab is where the technician can specify which screens need to be seen during the imaging process. By setting rules, the technician can cut down on what information needs to be inputted manually. In the example below, the user can specify which screens should be shown during the deployment.

After specifying the rules, the technician will move to the Windows PE tab. This tab allows for additional fonts to be added, allows the technician to select which driver group should be incorporated with the image, and allows the technician to specify whether an .ISO file should be generated alongside the .WIM bootable image.
When the technician is finished configuring the properties for the deployment point, the bootable image is now ready to be created. The technician can create the bootable image by selecting the deployment point and selecting update. The deployment point will begin to update and a progress bar will appear.

After the progress bar has completed, the .WIM image can be found in ‘C:\Distribution\Boot’. Please note, the technician will need the network path to this image for the purpose of adding the image into the Windows Deployment Services. At this point, the technician is finished using the Deployment Workbench and can now open Microsoft Deployment Services.

Microsoft Deployment Services (WDS) allows for the image to be deployed, captured and re-deployed to client computers. The image, which was created back in MDW, now has to be added to WDS. To add an image, right click on ‘Boot Images’ and select ‘Add Boot Image’. As noted before, please input the network path back to the .WIM image, which was just created a few moments ago.
After verifying the network path and selecting the image, the technician can apply an image name and image description into WDS.
The image is then verified by a summary screen and begins to upload to the WDS server. During this process, WDS uploads the image and checks the data integrity of the image to ensure the image is not corrupt in any way. After the image has been added, the technician should configure WDS, if not already done inside the WDS properties, the setting for PXE Response Settings is shown. The PXE Response Policy is set to respond to all known and unknown client computers. This is configured this way, since the user not joining the computers to the domain as part of the automated process. Computers should be able to join the transmission, even if they are not in Active Directory.
The next few steps will not be used right away, but are important to have ready to help capture an image of the client machine after it has been deployed and configured. The Install Images will need to be populated with groups. The captured image of the client machine will be placed in this group at a later point. To create an image group, simply right click on Install Images and select ‘Add Image Group’.

As a part of WDS, the image groups will hold the groups of images, which are multicasted to computers. The nice feature of WDS, in this area, is that the images are stored in differences. This means that if the technician has one image that runs Windows Vista and five applications and a second image that also has Windows Vista, the same five applications plus two other applications can be used. The second image only contains the two applications. This is useful as it helps save a large amount of hard drive space.

The next task to complete is to create a capture boot image. The client machine will be booted off of this special boot image, which is used to capture images of client computers. To create this image, the technician will add the boot.wim file from the Windows Server 2008 installation files. After the boot.wim file has been added to WDS, right click on the file and select ‘Create Capture Boot Image’.
The capture image wizard will appear. The technician should give the capture image a WDS name, description of the image, a file name, and a location to save the file to. The technician should do this with both x86 and x64 boot.wim files.
Once the image has been created, the technician should add this image into WDS. At this point, the client computer is ready to be booted and imaged. The client machine will connect to the WDS server by being booted from a Windows PE image. As the computer begins to boot, the technician should press F12 to instruct the computer to perform a network service boot.

Once the computer performs a network service boot, the client computer will begin to boot off the image that was just added into WDS a few steps ago. The client computer will obtain an IP address from the DHCP server and will begin to load.
Once Windows fully loads the files, the technician will select which task sequence should be ran on this computer. Once again, if an operating system is already installed on the computer, the technician should create a Standard Client Replace Task Sequence. In this example, the technician would choose the ‘Windows XP SP2 New Computer’ task sequence.
After choosing the task sequence, the technician will input the computer name that the client machine will be given. The technician can either leave the default name or input a different name.
At this point, the technician can start choosing which applications should be silently installed on this system. The applications can be selected by clicking the box next to each program name.
After choosing the applications to be installed, the administrator password is required to launch the deployment. The password is required to help protect the images from unauthorized access or deployment. If the password is correct, the deployment will begin to launch and the installation scripts will start to run.

![Image of a computer screen showing the installation of Windows Installer 4.15.](image)

This process will run all the scripts that install the operating system, applications, and other configurations. The technician should give this process a long period of time, as it may take several hours, depending on the number of applications selected to silently install.

The operating system is the first software to install. The operating system will install, the client machine will reboot, and the applications will begin to install. In the example below, Windows Installer 4.15 is installing. Depending on how the application’s script is written, some applications may show an interface while installing, while others will not show any interfaces.
After the scripts run, the client machine reboots, and all tasks are finished, the deployment is now finished. The technician can now complete manual configurations or installations that cannot be automated. After the manual tasks are completed, the technician should now sysprep the computer. Sysprep is short for system preparation. This will prepare the computer to be captured by generating a new security identifier (SID) and clearing the cache.
After the sysprep has successfully prepared the computer for capture, the machine will reboot and the technician should, once again, network service boot the machine. Once the client machine has network booted, the technician must select which image to boot the client machine off of. As seen below, there are three bootable .wim files, from which to choose. The technician should select the image which was created to be used as a capture boot image. This was titled ‘DeployImageBoot.wim’. Once the technician selects the proper image, the computer will now
The technician is greeted with an image capture wizard. By following the steps through the wizard, the image of the client computer will be captured, saved to the client computer, and uploaded to the WDS server.

![Windows Boot Manager](image.png)
The wizard now directs the technician to select a source to capture and provide an image name and description. If a source doesn’t show up in the ‘Volume To Capture’ drop down menu, please reboot the computer and run sysprep again. Otherwise, select the C drive, since it contains an operating system and applications.
After the volume to capture has been selected and an image name and description has been applied, the image will start to be created. The size of the source image will determine how much time is required to create the capture image.
Once the image has been created and uploaded to the WDS server, the technician should go back and verify that image is available in WDS. This image will be used during a multicast transmission. The image below verifies the image has been added and is available in WDS.
Now is the time to create a multicast transmission and deploy the image across a large number of computers. To create a multicast transmission, the technician should right click on the image that will be included in the multicast.
The technician should give the multicast transmission a name. The next step is to choose which type of multicast should be run. There are two choices available: scheduled cast or autocast. The scheduled multicast can be set to launch based off the number of clients joined to the multicast or the technician can set a time to launch the multicast. Autocast allows for client machines to run at a synchronous pace. The autocast will begin when one client joins the transmission. As other clients join, those clients will start off at the same point that the first client is at. Once the clients finish, the client computers who joined the transmission late will wrap around and complete the missing process.

The example below shows that a scheduled multicast transmission will be ran once 25 client computers have joined the multicast.

At this point, the technician should network boot the computers to be imaged. At this point, the client computers should now be booted off the Boot.wim file that was pulled from the Windows Server 2008 installation files. Once the computer is booted, the technician can select the image to download and install from the WDS server.
The following screen allows the technician to specify where Windows will be installed. The technician can choose to format, extend, or create a disk.
After selecting a destination for Windows, the technician is now finished and the MAC address will appear in WDS and the client computer will be waiting for the server.
Once the multicast has launched, the deployment process is at the end. The technician will complete manual configurations to each machine and have successfully deployed images using The Lite Touch Toolkit.

There are more options that can be added to the WDS server. By using Windows System Image Manager (WSIM), more processes can be automated. WSIM can tell the client machine which image to download from the WDS server, how the disk should be partitioned and many more configurations. During testing, WSIM was not used because the client machines contained different size hard drives, which ended up partitioning the disk and leaving space that has become unusable. If the technician desires to create these XML based answer files, then follow the steps in the Windows System Image Manager.

To launch Windows System Image Manager:
- go to Start>
- All Programs>
- Microsoft Windows AIK>
- Windows System Image Manager.

The illustration below is what the desktop should appear like.
References


5) Stockman, Mark. Professor. Personal Interview. 10 October 2008


