**Capstone Project Cover Sheet**



|  |  |
| --- | --- |
| **Capstone Project Name:** | **Upgrading Classroom Messaging System** |
| **Student Name:** | **John Deardurff** |
| **Degree Program:** | **Bachelor of Science: IT Administration** |
| **Student Mentor Name:** | **Dale Akita** |

**Table of Contents**

[Capstone Project Introduction 1](#_Toc434622930)

[Problem Statements 1](#_Toc434622931)

[Goals and Objectives 3](#_Toc434622932)

[Review of Other Work 4](#_Toc434622933)

[Project Rationale 5](#_Toc434622934)

[Systems Analysis and Methodology 6](#_Toc434622935)

[Project Deliverables 10](#_Toc434622936)

[Project Plan and Timelines 11](#_Toc434622937)

[Project Implementation 12](#_Toc434622938)

[References 34](#_Toc434622939)

[Appendix 1: 35](#_Toc434622940)

# Capstone Project Introduction

The purpose of this project was to upgrade the existing messaging infrastructure for the Microsoft Outlook email software courses that the application instructors teach at New Horizons Computer Learning Centers of Indianapolis (NHCLC). The Exchange administrator for NHCLC (John Deardurff) upgraded the existing messaging system from an old server with Windows 2003 Server operating system and the Microsoft Exchange Server 2003 messaging software installed to a new separate server with the Windows 2012 Server operating system and the Microsoft Exchange Server 2010 messaging software installed. In addition, the Exchange administrator created a PowerShell script to automate the process of mailbox management. Project approval has been obtained from the NHCLC Facilities Manager (Breanna Ridgway). The project achieved the following objectives.

1. Added 8 GB of RAM to a computer that had a 64-bit processor.
2. Prepared environment for messaging infrastructure upgrade.
3. Installed the Windows Server 2012 operating system on the new server.
4. Installed DNS Services and Active Directory Domain Services on the new server.
5. Installed the Exchange Server 2010 messaging software on the new server.
6. Created PowerShell scripts to automate classroom setup.
7. Tested classroom setup.

# Problem Statements

In deciding to undertake this project, we analyzed and evaluated several factors and determined the need for this project and the need to perform the project within the current timeframe. The objectives of the project listed above are intended to correct the problem factors listed below.

1. Current infrastructure is aging - The current server named APU maintains the mailboxes for the Outlook email software classes. When we built the messaging server in June of 2003, the Windows Server 2003 operating system and Exchange Server 2003 messaging software was the appropriate infrastructure to teach the Outlook 2003 courses. This is no longer the case as the Outlook 2003 courses are no longer part of the curriculum offered by NHCLC. In addition, currently we are unable to teach the newer Outlook 2013 courses, as that software product is not compatible with Exchange 2003. To teach the email courses beyond Outlook 2010 an Exchange 2010 server is the minimum requirement.
2. End of Microsoft product support - Exchange Server 2003 reached the end of the Microsoft Support lifecycle on November 15, 2013. Windows Server 2003 reached the end of the Microsoft support lifecycle on July 14, 2015. This poses security and management risks as these products do not have security patches and product updates provided by Microsoft.
3. Current server is overworked - Overtime it became convenient to add additional services to the APU server including the web filtering services that control internet traffic for all of NHCLC. This not only overworks the server, but when setting up the Outlook classes it is sometimes necessary to restart this server. Internet service disruption occurs for both the classroom environment and NHCLC in general whenever restarting the APU server. Therefore, classroom setup is required outside of normal working and classroom hours.
4. Time required for student mailbox setup – The process for building the classroom environment for Outlook 2010 is a time exhaustive process. The facilities manager creates mailboxes to hold student emails, calendars, and other related course items before the beginning of class. At the end of each class the facilities manager deletes each student mailbox. This ensures that each course begins fresh without lingering items from a previous class. Exchange 2010 allows the ability to create PowerShell scripts to automate this process saving the facilities manager time to perform other tasks.
5. Time required connecting Outlook clients - Exchange 2003 server requires manual configuration to connect the Outlook software client to each students’ mailbox. This consumes time by the facilities manager or instructor when setting up an Outlook class. However, Exchange 2010 includes the AutoDiscover service that allows each Outlook software client to self-configure a connection to the appropriate student mailbox thereby eliminating the need for the facilities manager from doing this process manually.
6. Project window timeframe – Project completion is required before the next course delivery of Outlook 2010 on December 4, 2015. Pre-planning and preparation will occur the week of November 2, 2015. Planned downtime occurs during non-work hours on November 3, 2015. Installation and testing is planned for November 4, 2015.

# Goals and Objectives

The goals of this project was to implement a new messaging infrastructure that upgraded the existing aging infrastructure, updated the existing operating system and messaging software to allow for security and software updates from Microsoft product support, lessened the burden and workload of the current server hosting the messaging system, reduced the time required for student mailbox setup, reduced the time required connecting Outlook clients, and provided the solution within the projected window timeframe. These goals were met by accomplishing the objectives listed below.

1. Located an unused server that had a 64-bit processor and 8 GB of RAM.
2. Prepared environment for messaging infrastructure upgrade.
3. Installed the Windows Server 2012 operating system on the new server.
4. Installed DNS Services and Active Directory Domain Services on the new server.
5. Installed the Exchange Server 2010 messaging software on the new server.
6. Created PowerShell scripts to automate classroom setup.
7. Tested classroom setup.

# Review of Other Work

To begin working on this project, it was necessary to review the requirements and procedures of upgrading to an Exchange 2010 messaging infrastructure. As a Microsoft Certified Trainer, I have taught the Microsoft Exchange 2010 courses at least 20 times over the last five years. In addition, I have previously performed at least two dozen Exchange upgrades over the last 15 years. It was with this in mind that to begin the review of work, I began by talking to myself. (Deardurff, 2015). However, to provide proper documentation, I looked online to see if an upgrade was possible from our current environment. My first stop was a Microsoft TechNet article that stated,

“You can’t perform an in-place upgrade from Exchange 2003 to Exchange 2010. However, you can install an Exchange 2010 server into the existing Exchange organization, and then move the Exchange resources, such as mailboxes, public folders, and connect to Exchange 2010.” (Microsoft TechNet, 2010)

This information reinforced that the existing server could not be used for the upgrade and a new server would need to be built and joined to the existing organization. Next, I wanted to discover if there were any prerequisites needed for the Exchange environment. I discovered that “Exchange 2003 must be in Native mode.” (Michael B Smith, 2010). In addition, I documented the following Active Directory prerequisites:

“The Domain Controller holding the Schema Master role and the Global Catalog must be running Windows Server 2003 SP1 or higher and Active Directory must be in Server 2003 Forest Functional Level and Domain Functional Level.” (Jaap Weeselius, 2009)

For the individual systematic instructions on performing the upgrade, I downloaded a copy of the “Exchange Server 2003 to 2010 Migration Guide” by Paul Cunningham. (Paul Cunningham, 2010). Finally, to assist with writing the PowerShell scripts I just went straight to Exchange Online help to find the cmdlets needed. (Microsoft.com, 2009)

# Project Rationale

As previously mentioned in the Introduction section, six factors played into the role of undertaking this project. The current infrastructure is aging, end of Microsoft product support, current server is overworked, time required for student mailbox setup, time required connecting Outlook clients, and project window timeframe. The rationale in this project is that by providing a server that supports a 64-bit capable processor and contains 8GB of RAM, we replaced the existing, aging infrastructure. By upgrading to the Windows 2012 operating system and Exchange 2010 messaging software, the server will be able to receive the most recent updates and service packs from Microsoft product support. By removing the classroom messaging system from the original server, the resources of that server will not be as overworked. A future project will use the current server strictly as the Internet web filter server. Since we went to a Windows 2012 environment, I was able to create PowerShell scripts to automate the task of student mailbox setup. This will save valuable time of both the facilities manager and application instructors. By installing Exchange Server 2010, the AutoDiscover feature now allows Outlook clients to locate and self-configure mailboxes. This will also save classroom setup time. Finally, the reason for the current project timeframe is to take advantage of a gap in our classroom schedule from when our instructors are teaching the Outlook course.

# Systems Analysis and Methodology

The Systems Analysis and Methodology approach taken for this project was the ADDIE model. During the Analysis phase it was discovered that a new Exchange messaging system was needed for NHCLC Outlook courses for the following six reasons. The current infrastructure is aging, end of Microsoft product support, current server is overworked, time required for student mailbox setup, time required connecting Outlook clients, and project window timeframe.

During the Design phase of the project it was determined that a new physical server would be needed for this project. This step was accomplished by locating a server that had a 64-bit capable processor and to upgrade it to 8GB RAM. In addition, the current environment did not meet all the pre-requisites of being able to perform the upgrade. It was originally thought that downtime would be needed. However, after starting the project it was discovered that a reboot of the existing server would not be needed. Therefore, downtime during off hours was no longer needed to perform the upgrade. The current Exchange Server 2003 environment that was running on Windows Server 2003 Service Pack 2 had not been elevated to the correct Exchange Mode, Domain Functional Level, and Forest Functional Level. The first few steps of the project included these three steps.

1. Raise the Exchange Organization to at least Exchange 2003 Native Mode.
2. Raise the Active Directory Domain Functional Level to 2003 Native Mode.
3. Raise the Active Directory Forest Function Level to 2003 Native Mode.

In addition, to prepare the messaging environment for the migration, two Exchange commands was needed prior to installation of Exchange Server 2010. Originally these commands were going to be run during the downtime period, however, it was decided to wait until after Windows 2012 was installed to run the commands on the new server.

1. Run Exchange Setup.exe /PrepareSchema
2. Run Exchange Setup.exe /PrepareLegacyExchangePermissions.

This lead to the Development phase of planning for the project. The software requirements for the new classroom messaging system is Microsoft Windows Server 2012 Standard Edition and Microsoft Exchange Server 2010 with Service Pack 1. The Active Directory Domain Service (ADDS) and the Domain Naming Service (DNS) were also installed on the new server. Here are the high level steps that was taken during the Implementation phase of the project.

**Setup of Windows Server 2012 Standard Edition**

1. Insert the Windows Server 2012 DVD.
2. On the press any key to continue screen, press any key.
3. On the Language to Install, Time and currency format, and Keyboard input select English and select Next.
4. On the Install Windows screen, select Install Now.
5. Select the GUI install of Windows Server 2012 and then click Next.
6. Check the box to Accept the End User License Agreement and click Next.
7. Select Custom: Install Windows only.
8. Select to create an NTFS partition using the entire 127gb drive and click Next.
9. Wait for Windows Server to install and when it reboots set password to the password used for NHCLC classroom servers and click OK.
10. Download and Install any suggested updates. Restart and login with Administrative credentials for the NHCLC classroom servers.

**Install the DNS and Active Directory Domain Services Roles to the server.**

1. Configure a static IP Address with the following settings.
   1. IP Address: 192.168.19.200
   2. Subnet Mask: 255.255.255.0
   3. Default Gateway 192.168.19.250
   4. DNS Server 192.168.19.253
2. Change the computer name to CECIL and reboot the server.
3. Join the server to the Classrooms.Realm domain and reboot the server.
4. Start Server Manager – Add Roles - Add Domain Name System (DNS) Server Role.
5. In Server Manager – Add Roles – Active Directory Domain Services.
   1. In Server Manager – Promote this server to a Domain Controller.
   2. Add a domain controller to an existing domain named Classrooms.Realm
   3. Specify that this server will be a DNS Server and Global Catalog Server place in the Default Site. Enter the NHCLC administrative password as the Directory Services Restore password.
   4. Select to replicate from existing Domain Controller and Place database and log files in the default location.
   5. Select to Finish Installation of Active Directory Domain Services and reboot.

**Pre-Exchange Server 2010 Installation Steps**

1. Run Exchange Setup.exe /PrepareSchema.
2. Run Exchange Setup.exe /PrepareLegacyExchangePermissions.
3. Install .NET Framework 3.5 Feature
4. Install Web Services
5. Install RPC over HTTP
6. Install Microsoft Office 2010 Filter Packs
7. Start Net.TCP Port Sharing Service and Set it to Automatic.

**Install and Configure Microsoft Exchange Server 2010**

1. Insert the Microsoft Exchange Server 2010 DVD.
2. When setup starts select Install or run program from media.
3. Choose English Language to install and then Next.
4. Select Install Microsoft Exchange and wait until files are copied to server.
5. Accept End User License Agreement and select Next.
6. On Installation Type stay with Typical Install and select Next.
7. On the Exchange Organization page, type CLASSROOMS and select Next.
8. Select Finish to Install Exchange Server 2010.
9. Test Connectivity

**Create PowerShell Script to Delete existing mailboxes and Create new mailbox.**

1. Ensure that student accounts are in Active Directory under Students OU.
2. Run PowerShell Script and Test PowerShell script.

This finishes the Analysis, Design, and Development phases of the ADDIE model of the Systems Analysis and Methodology section of this project. We will now focus on the Implementation and Evaluation phases of this project.

# Project Deliverables

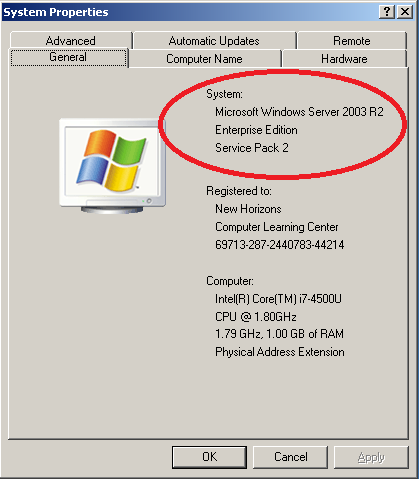
The following is the step-by-step implementation of the project that was completed on November 3, 2015. As was previously mentioned, there were a few design changes to the project that happened the day the project started. The main change was that I had originally thought a reboot was needed after Raising the Domain and Forest Functional Levels on the existing server. However, after further reading this was not the case and we were able to start the project 5 hours earlier than expected since we did not have to wait until after work hours. Another design change was the Active Directory Schema Prep and Exchange Legacy Permission commands needed to be run on the new server instead of the existing server. These steps were moved to later in the project but did not affect the timeline. Two other steps were added to the project to install the pre-requisites service features of the .NET framework 3.5 and the Web Services. This added about 10 minutes to the total project. Finally, as I started working on the PowerShell script I wanted to add a few features to the original code to make it flow a little smoother. The project was scheduled for 12 hours over a two-day time period. Due to being able to perform the project during normal working hours and because of pre-planning, the project was completed in 6 hours during a single day.

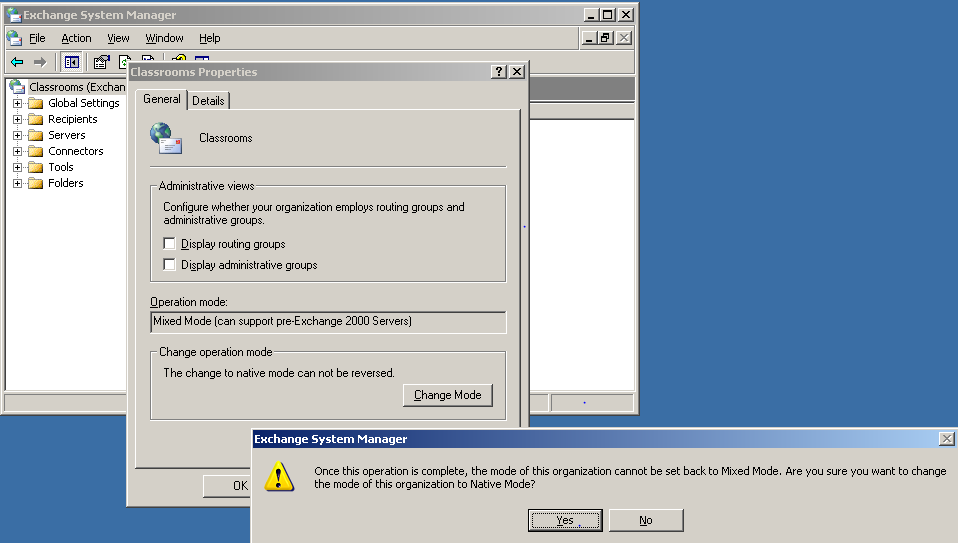
# Project Plan and Timelines

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Deliverable or Milestone** | **Planned Duration** | **Planned Start Date** | **Actual Duration** | **Actual Start Date** |
| Locating and Building of a 64bit server with 8GB RAM | 4 hours | 11/3/2015 | 1 hour | 11/3/2015 |
| Pre-planning and PowerShell script creation. | 2 hours | 11/3/2015 | 1 hour | 11/3/2015 |
| Scheduled downtime for implementing environment pre-requisites. | 1 hour | 11/3/2015 | Moved | Moved |
| Install Windows Server 2012 R2 | 1 hour | 11/4/2015 | 45 minutes | 11/3/2015 |
| Install DNS and Active Directory Services | 1 hour | 11/4/2015 | 15 minutes | 11/3/2015 |
| Install Exchange Server 2010 messaging software. | 1 hour | 11/4/2015 | 1 hour | 11/3/2015 |
| Test PowerShell scripts and classroom environment setup | 2 hours | 11/4/2015 | 2 hours | 11/3/2015 |

# Project Implementation

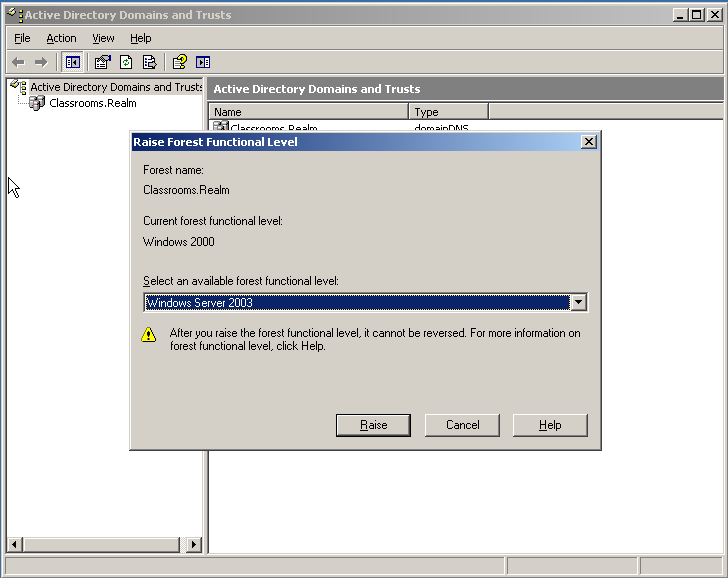
**Preparing the Exchange Messaging Environment**

Server is running Windows Server 2003 R2 and Service Pack 2

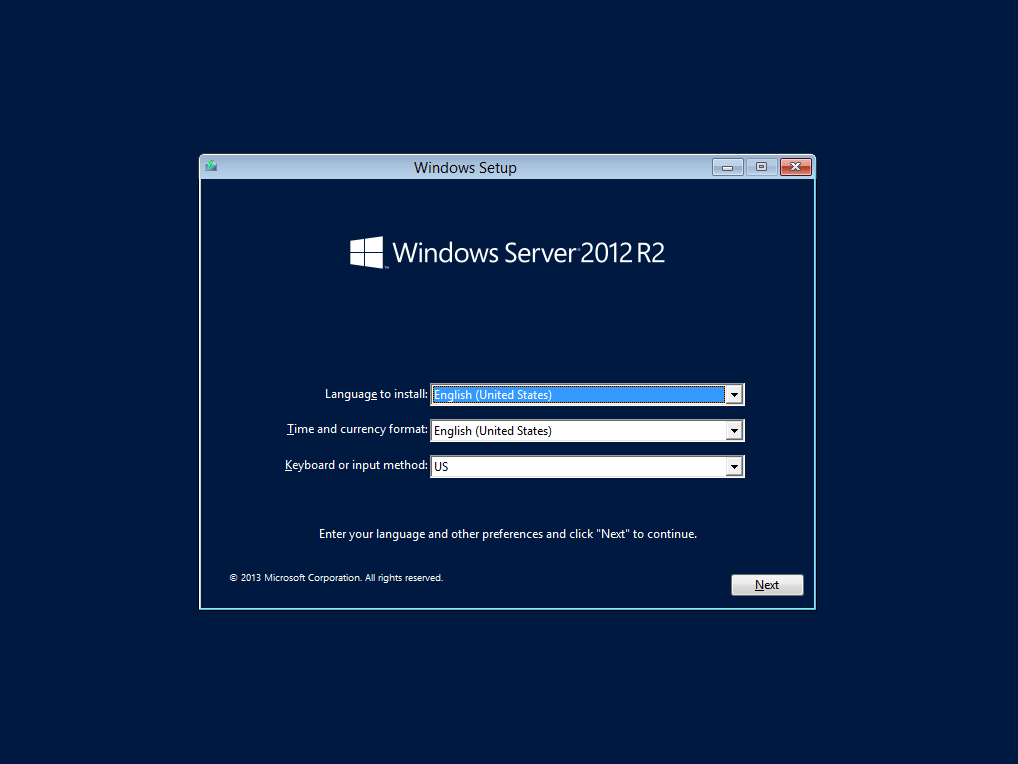
Raised the Exchange Organization to at least Exchange 2003 Native Mode.

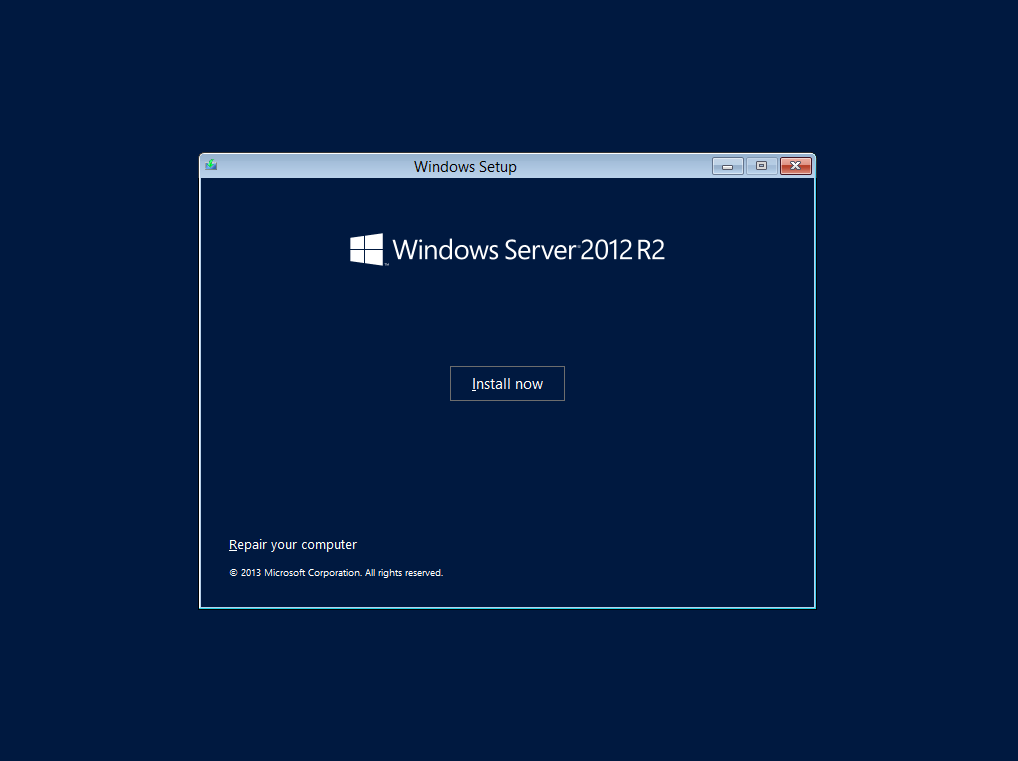
Raised the Active Directory Domain Functional Level to 2003 Native Mode.

Raised the Active Directory Forest Function Level to 2003 Native Mode.

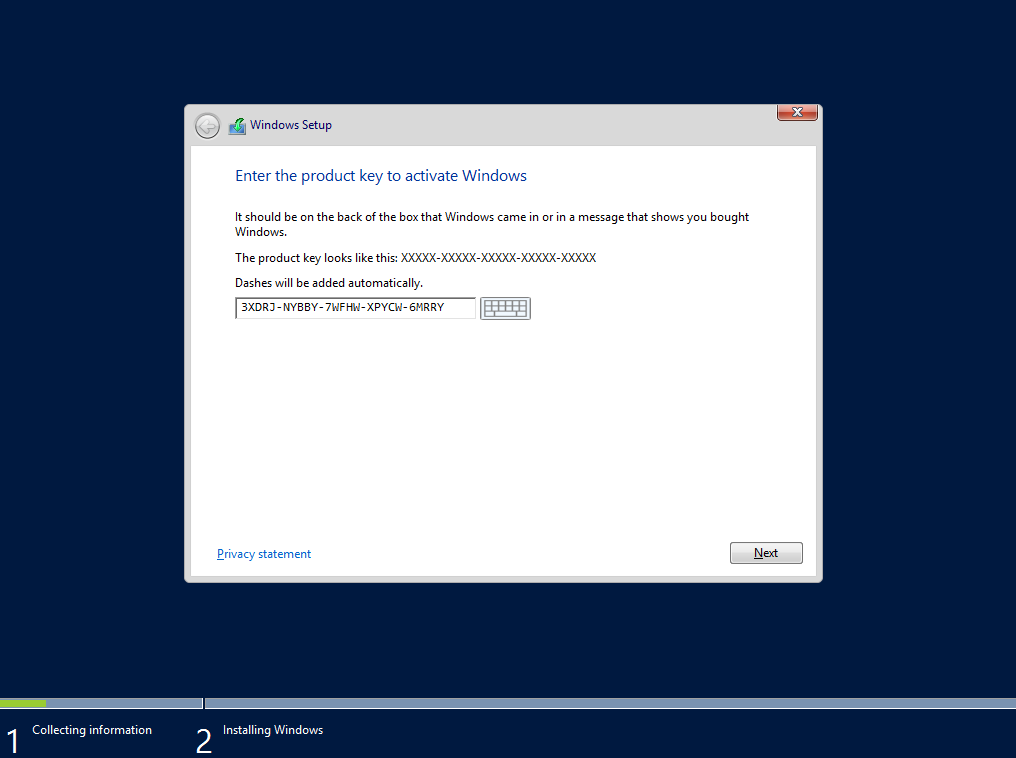


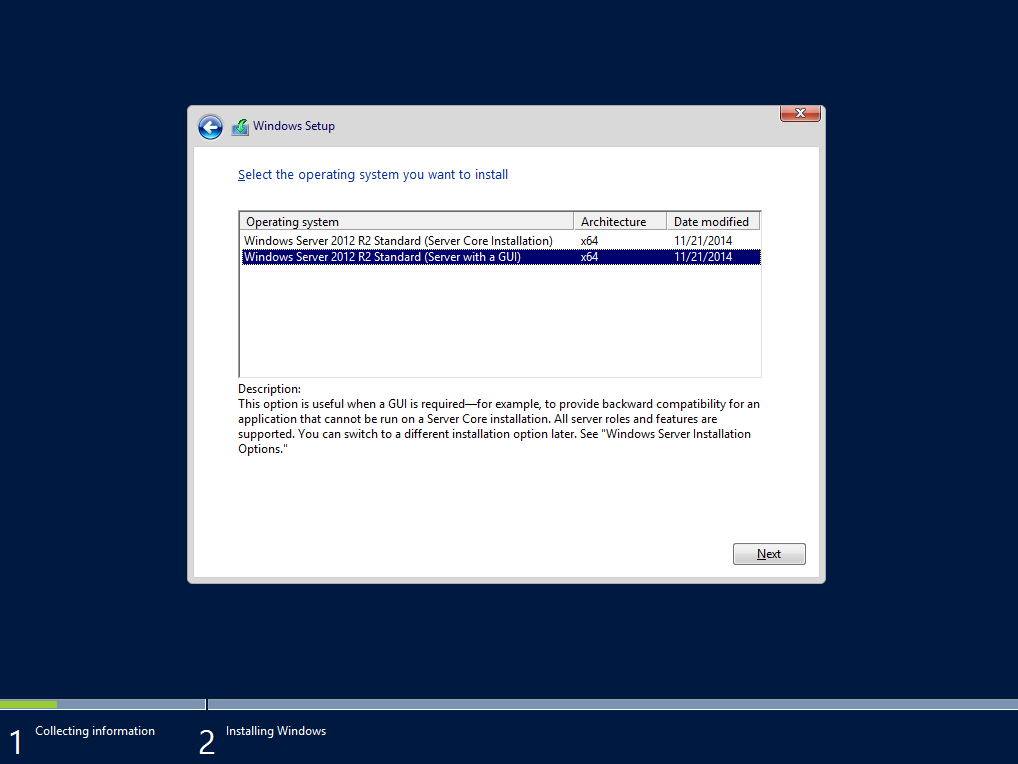
**Install Windows Server 2012 Standard Edition**

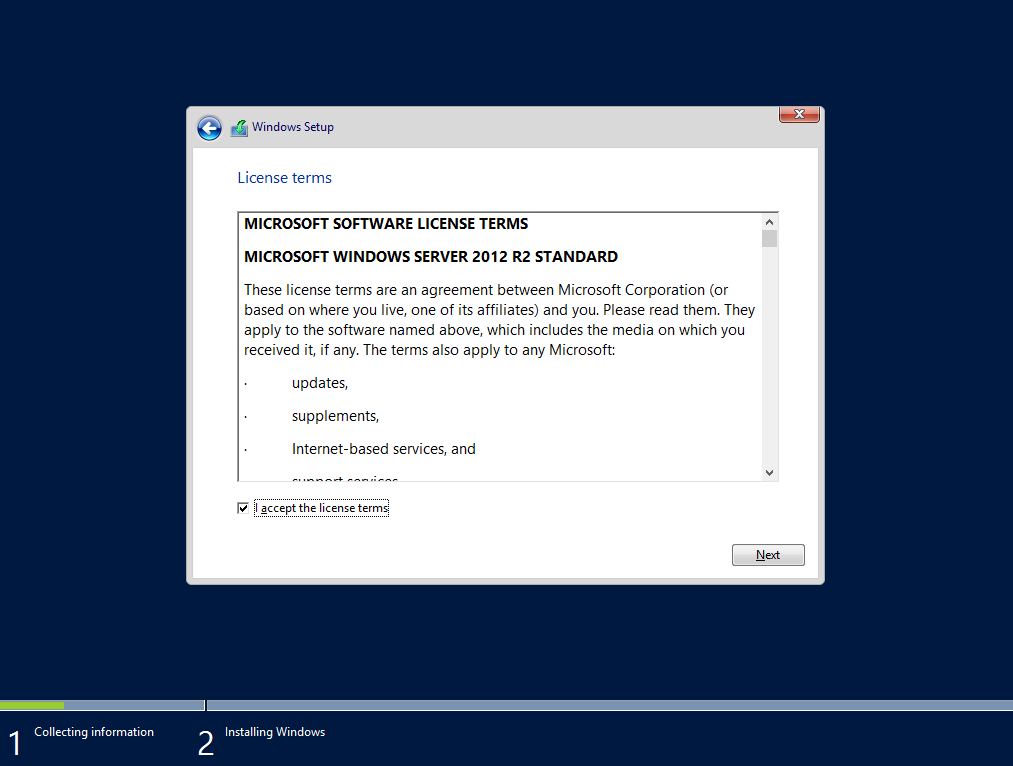
Insert the Windows Server 2012 DVD. Select English for Language, Time and Currency.****

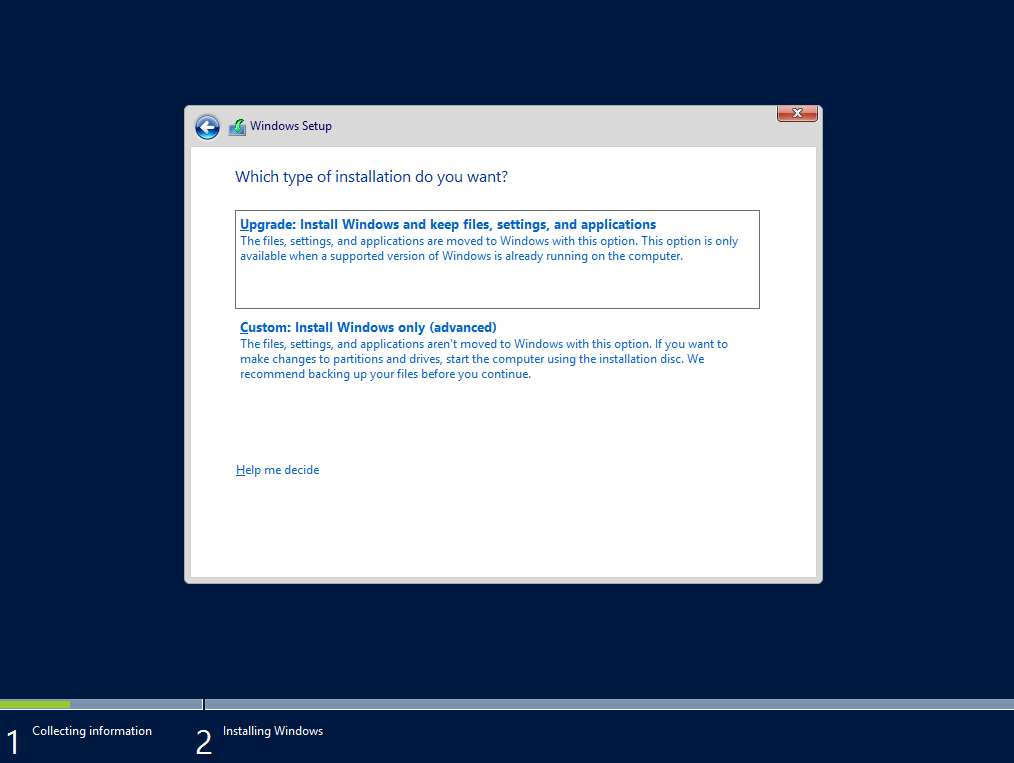
On the Install Windows screen, select Install Now.****

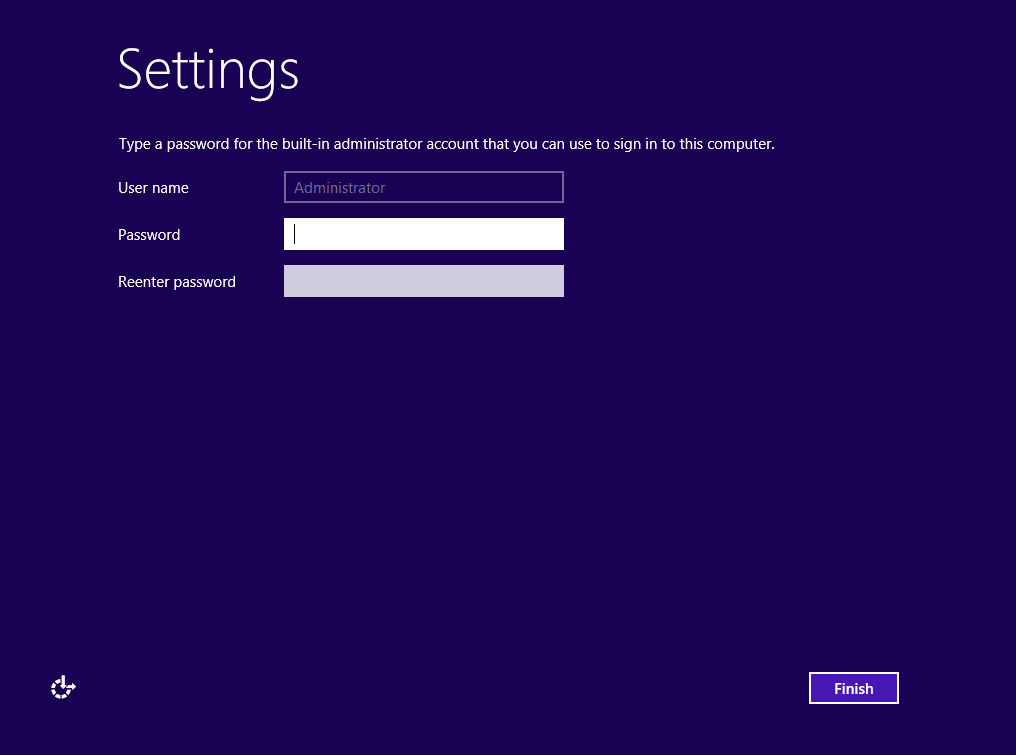
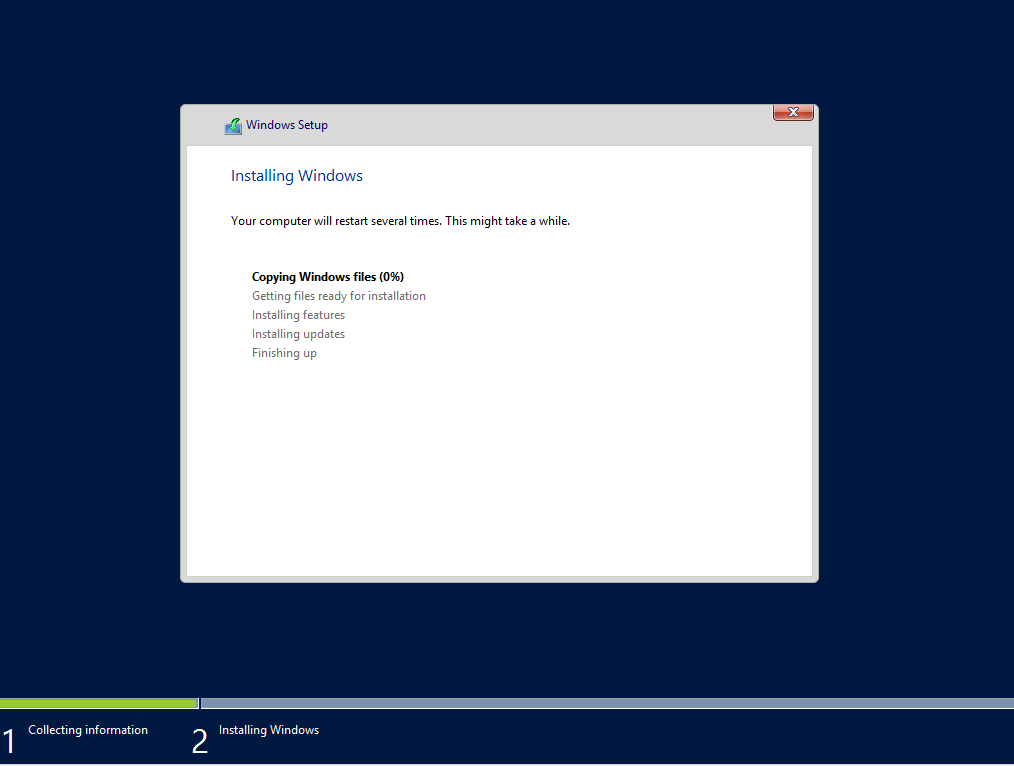
Add the Product Key to Activate

****

Select the GUI install of Windows Server 2012 and then click Next.****

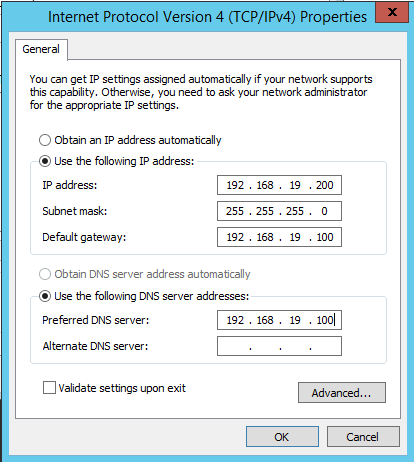
Check the box to Accept the End User License Agreement and click Next.****

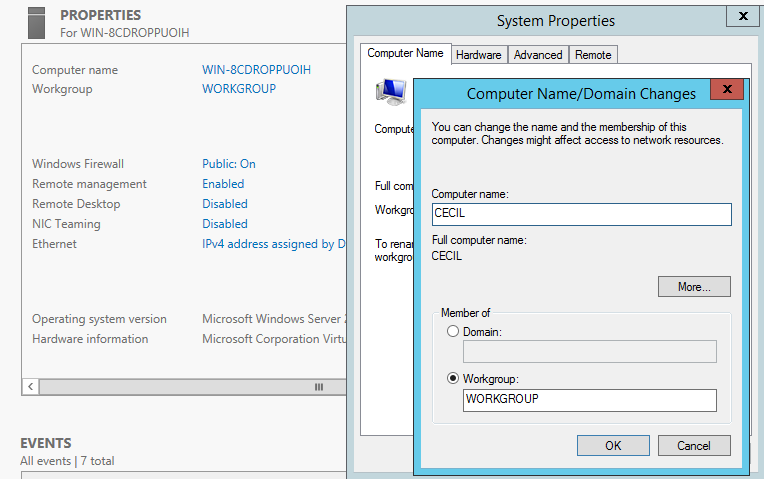
Select Custom: Install Windows only.****

Wait for Windows Server to install and when it reboots set password to the password used for NHCLC classroom servers and click OK.****

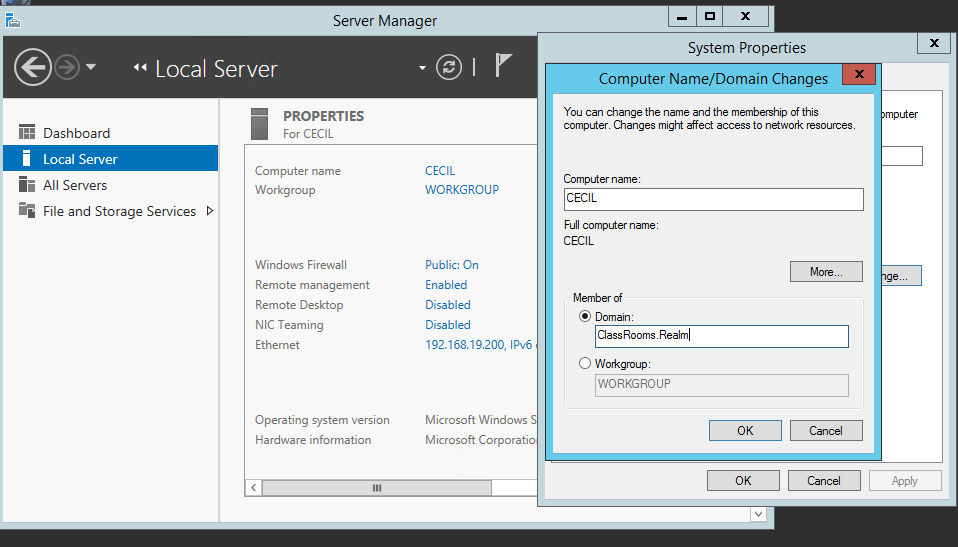
**Install the DNS and Active Directory Domain Services Roles to the server.**

Configure a static IP Address

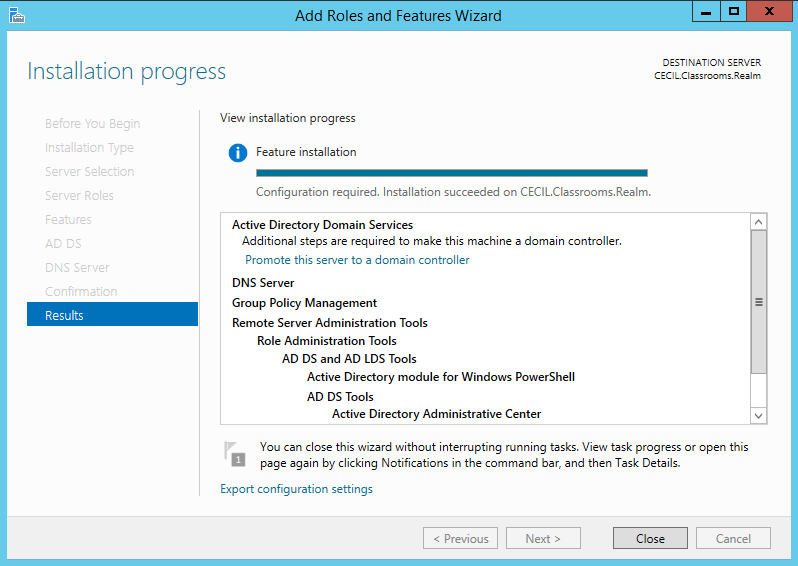


Change the computer name to CECIL and reboot the server.

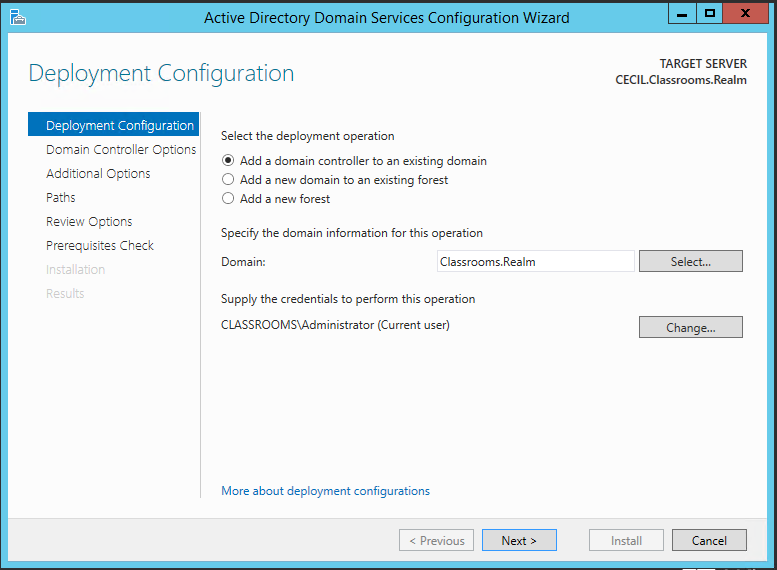
Join the server to the Classrooms.Realm domain and reboot the server.

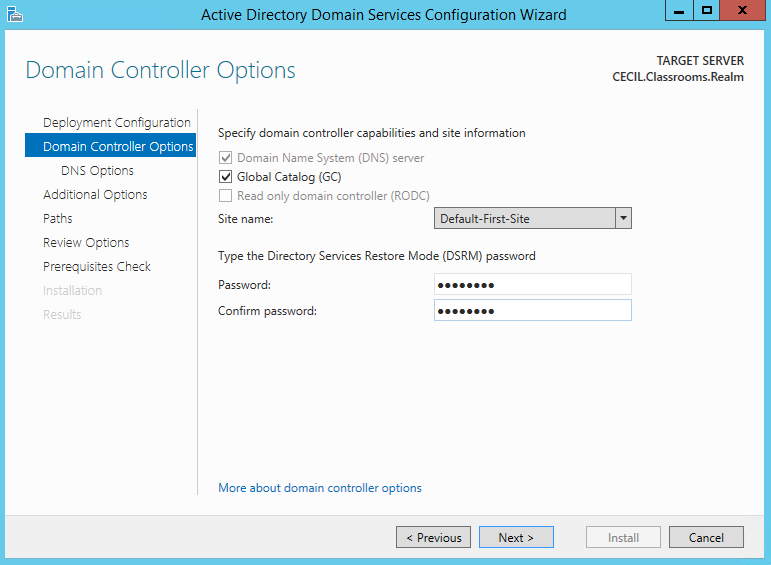


Start Server Manager – Add Roles - Add Domain Name System (DNS) and Active Directory Domain Services (ADDS) Server Roles

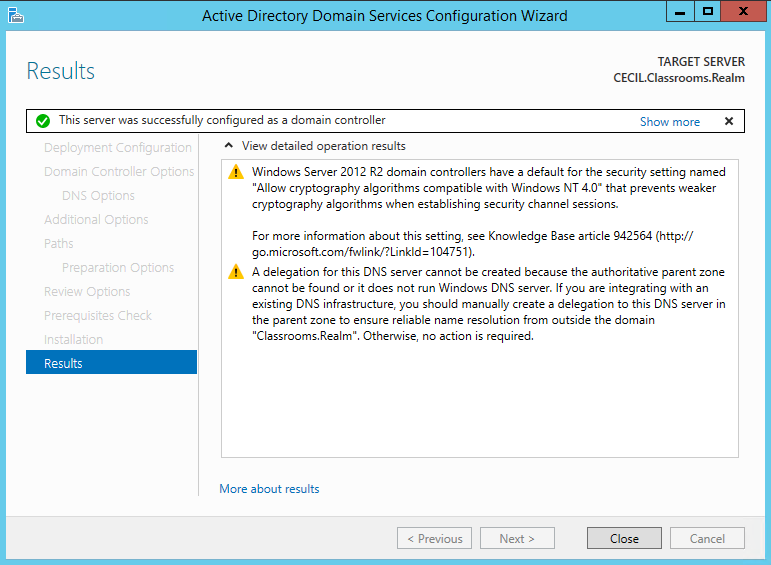


Promote this server to a Domain Controller.

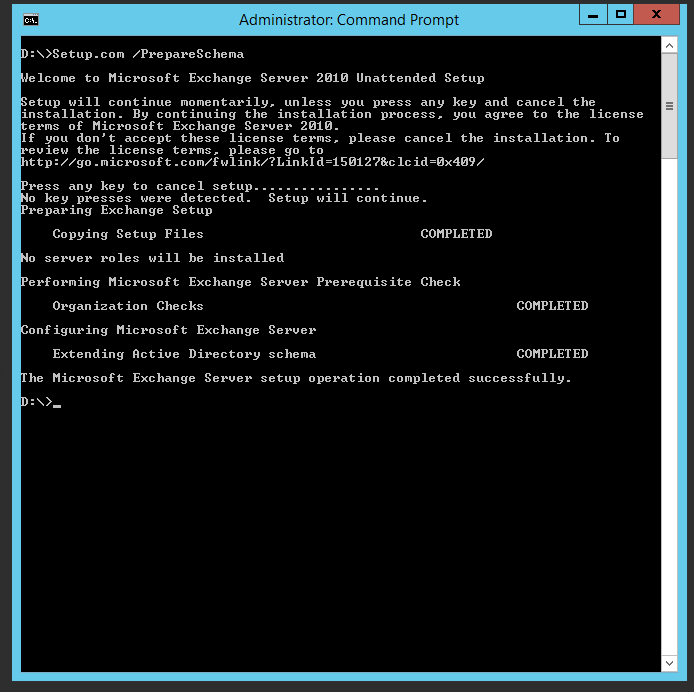


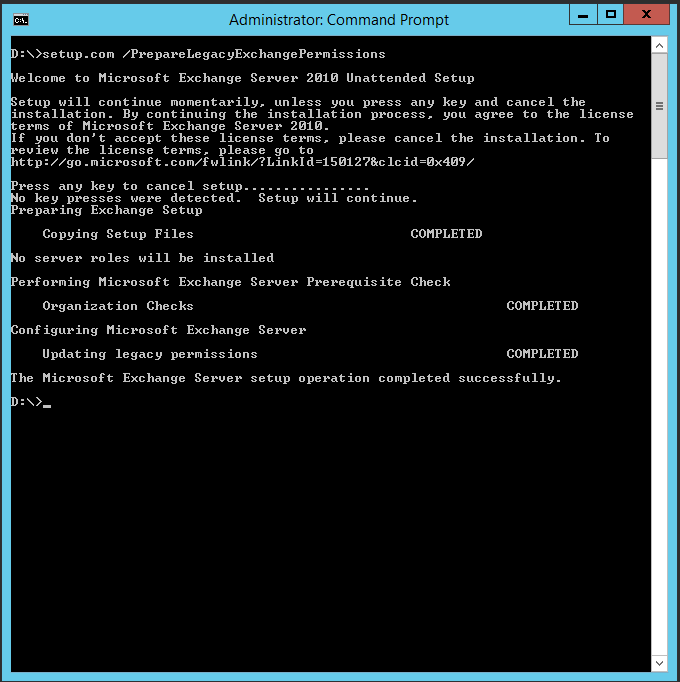
Specify that this server will be a DNS Server and Global Catalog Server place in the Default Site. Enter the NHCLC administrative password as the Directory Services Restore password.

Select to Finish Installation of Active Directory Domain Services and reboot.

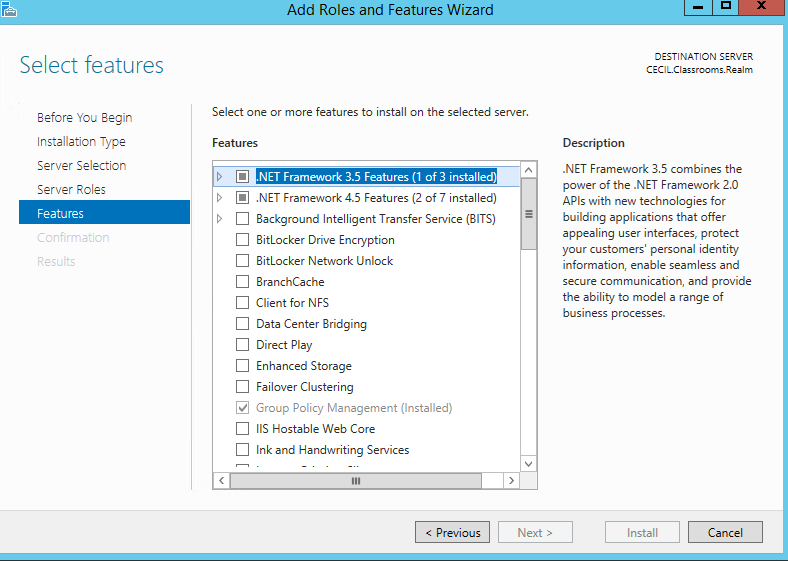


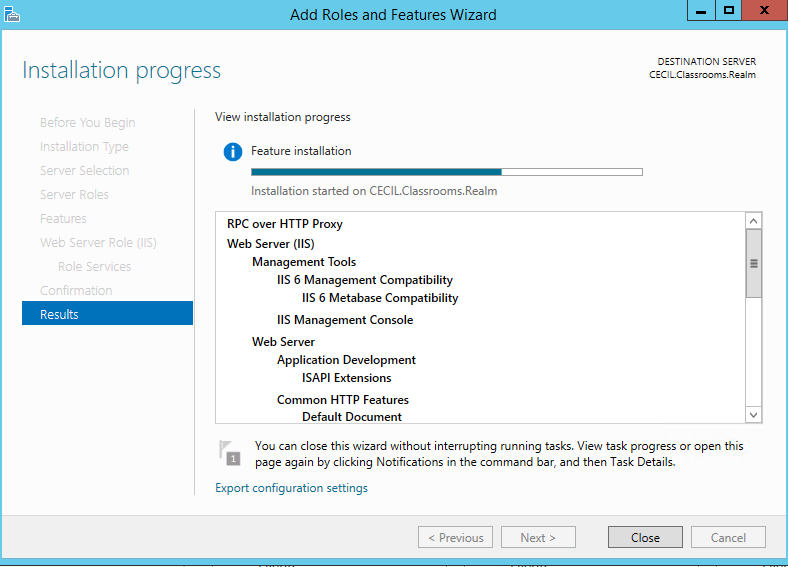
**Pre-Exchange Server 2010 Installation Steps**

Run Exchange Setup.exe /PrepareSchema.

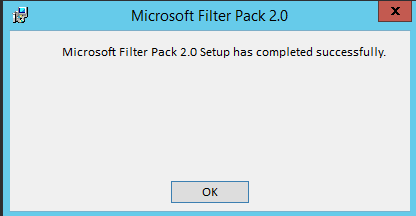
Run Exchange Setup.exe /PrepareLegacyExchangePermissions.

Install .NET Framework 3.5 Feature

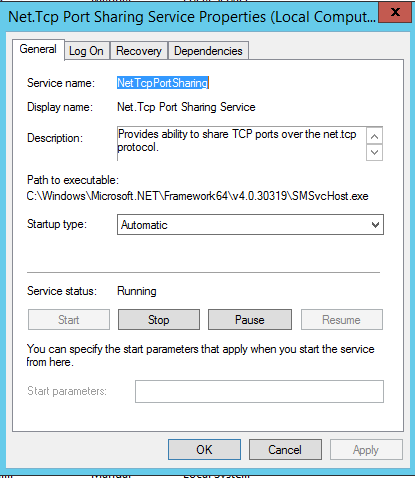


Install Web Services and RPC over HTTP Services

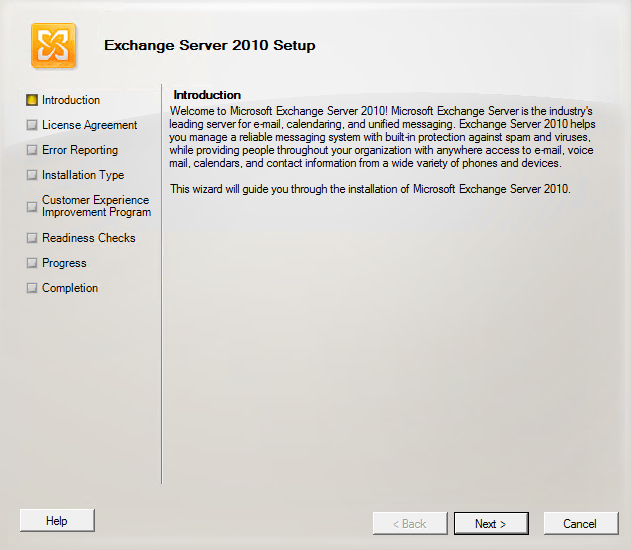
Install Microsoft Office 2010 Filter Packs

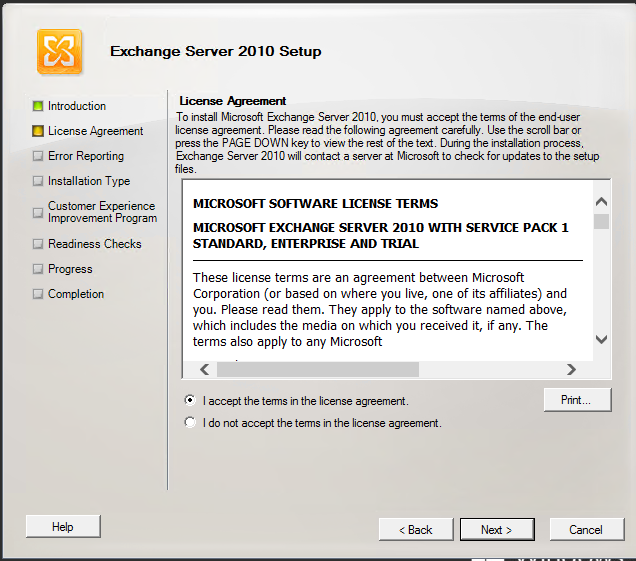


Start Net.TCP Port Sharing Service and Set it to Automatic

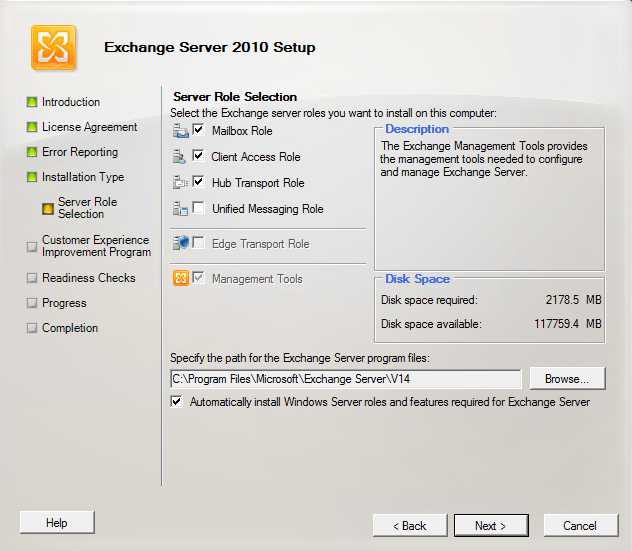


**Install and Configure Microsoft Exchange Server 2010**

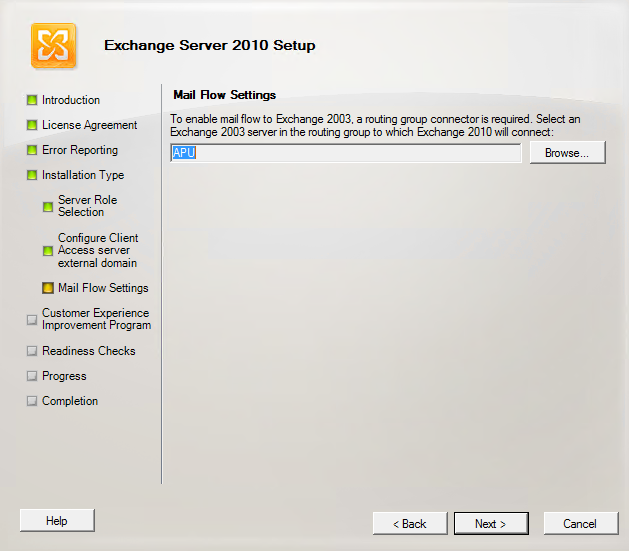
Insert Exchange Server 2010 DVD ****

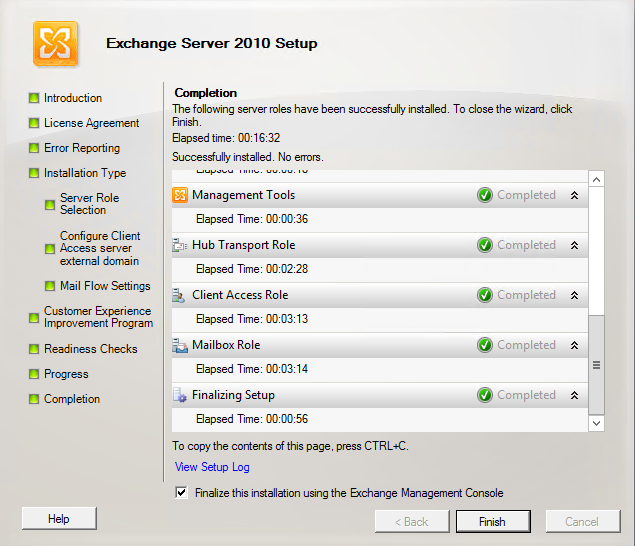
Check the box to Accept the End User License Agreement and click Next.

Select Custom Exchange Server Installation and click Next.

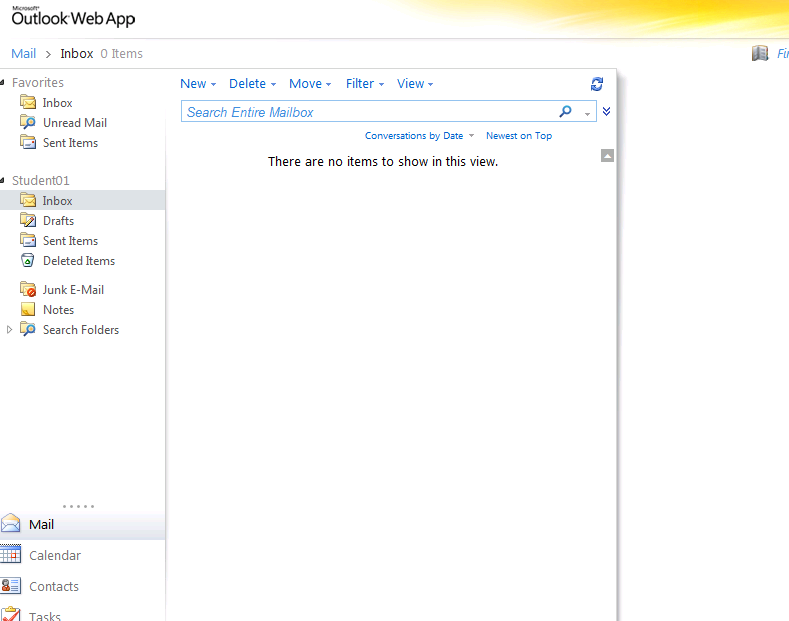
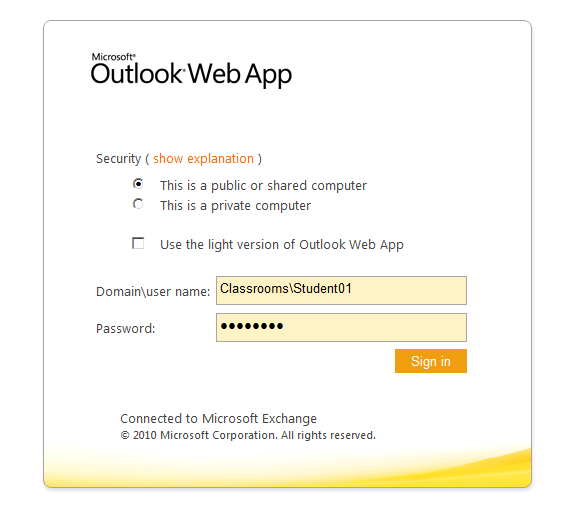
Select Mailbox, Client Access, and Hub Transport Roles and click Next.

Select Configure Client Access external domain as webmail.classrooms.realm and click Next.

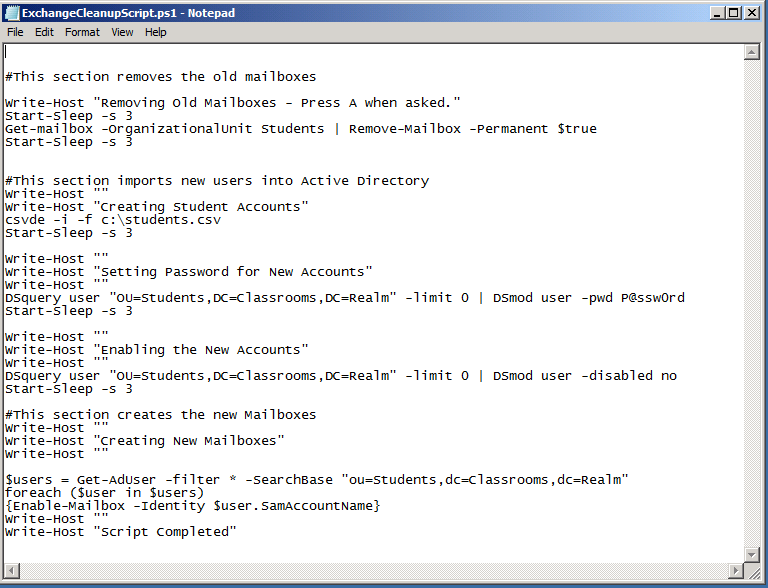
Join Exchange 2010 to the Exchange 2003 Routing Group and click Next.

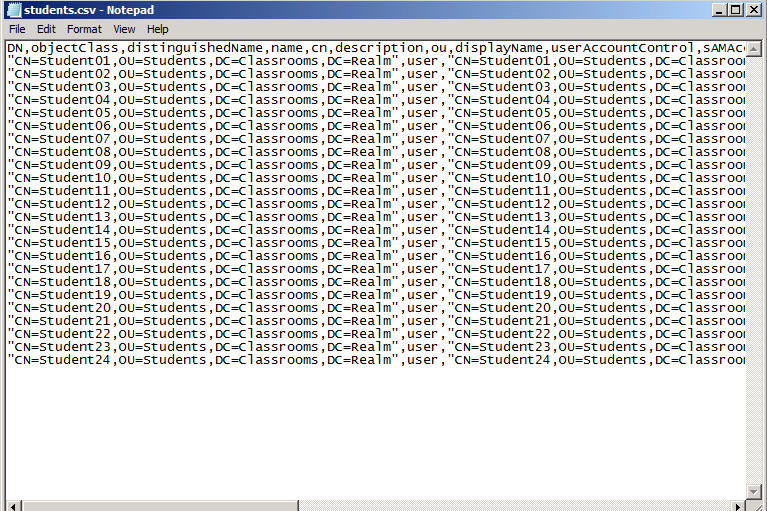
Select Wait for Exchange Installation to complete and the click Finish 

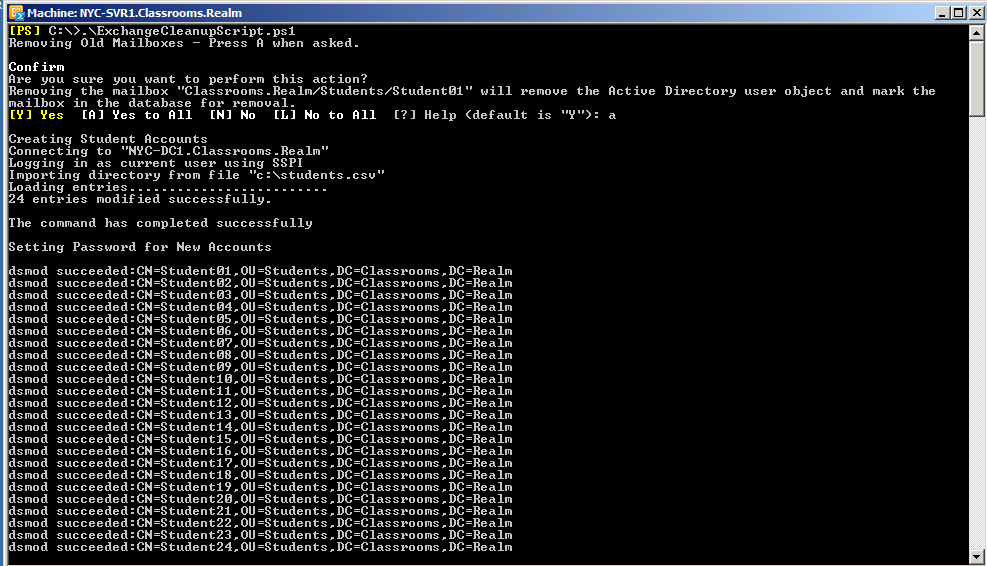
**Test Outlook Connectivity**

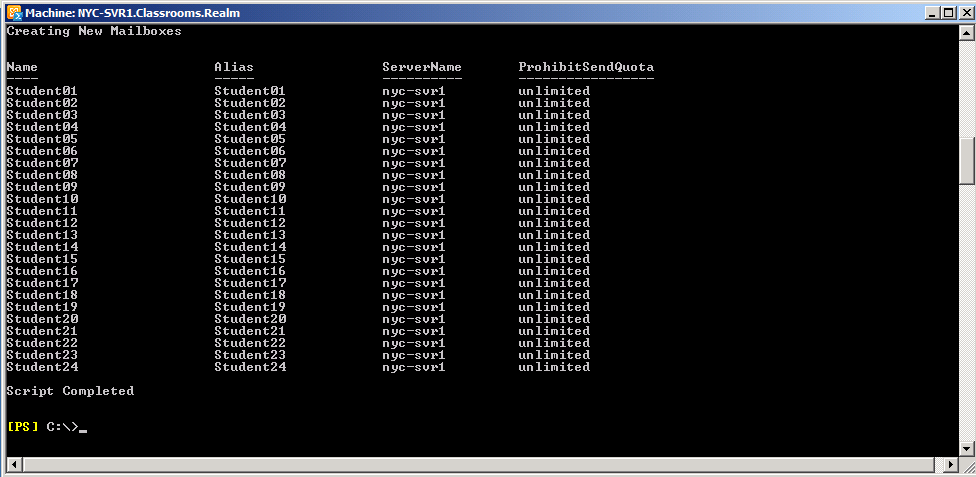
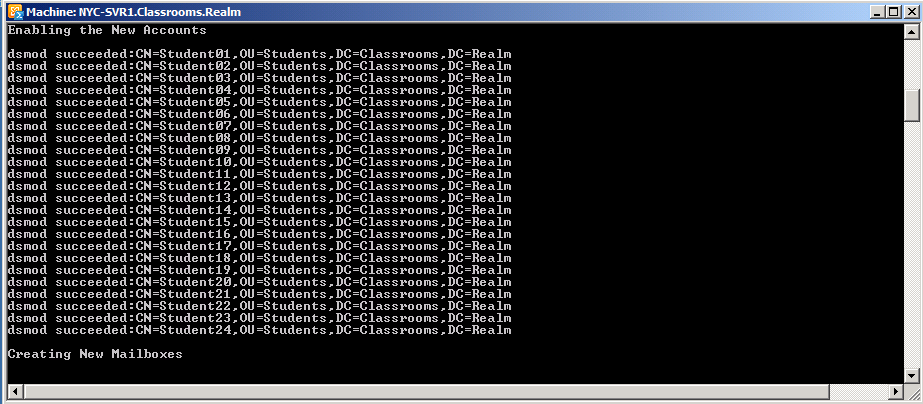
****

**Create PowerShell Script to Delete existing mailboxes and Create new mailboxes.**



CSV Script that the PowerShell script uses to creates new Student Accounts.

**Test PowerShell Script Page 1**

**Test PowerShell Script Page 2**

# References

John Deardurff (2015). Retrieved 11 October 2015 from personal experience.

Microsoft TechNet (2010). *Install Exchange 2010 in an Existing Exchange 2003 Organization*. Retrieved 11 October, 2015, from <https://technet.microsoft.com/en-us/library/dd638130(v=exchg.141).aspx>

Michael B Smith, M.B.S. (2015, 04 April 2010). Migrating from Exchange Server 2003 to Exchange Server 2010: A Small Organization Perspective. [Weblog]. Retrieved 11 October 2015, from <http://windowsitpro.com/exchange-server-2010/migrating-exchange-server-2003-exchange-server-2010-small-organization-perspect>

Jaap Weeselius, J.W. (2009, 11 December 2009). Upgrade Exchange 2003 to Exchange 2010. [Weblog]. Retrieved 11 October 2015, from <https://www.simple-talk.com/sysadmin/exchange/upgrade-exchange-2003-to-exchange-2010/>

Paul Cunningham, P.C. (2010, no-date). Exchange Server 2003 to 2010 Migration Guide. [Weblog]. Retrieved 11 October 2015, from <http://l.exchangeserverpro.com/exchange-2003-2010-migration-guide/>

Microsoft.com. (2009). *Microsoft.com*. Retrieved 11 October, 2015, from <https://technet.microsoft.com/en-us/library/aa995948(v=exchg.141).aspx>

# Appendix 1:

