Lab – File System Commands

Introduction

In this lab, you will use CLI commands to manage files and folders in Windows.

Recommended Equipment

- A computer running Windows

Instructions

Step 1: Access the Windows command prompt.

a. Log on to a Windows computer. The account **ITEUser** is used as the example user account throughout this lab.

b. To access the Windows command prompt, click **Start** and enter **cmd**. Select **Command Prompt** or **cmd.exe** to continue.

Step 2: Create and change directories.

In this step, you will use the change directory (**cd**), make directory (**md**), and directory (**dir**) commands.

Note: A directory is another word for folder. Directory and folder are used interchangeably throughout this lab.

a. Type **cd** at the command prompt.

Question:

What is the current directory?

C:\Users\Ralph

b. Type **dir** at the command prompt to list the files and folders that are in the current folder.

```
C:\Users\ITEUser> dir
 Volume in drive C has no label.
 Volume Serial Number is 9055-35E9

 Directory of C:\Users\ITEUser

04/27/2019  09:21 AM    <DIR>          .
04/27/2019  09:21 AM    <DIR>          ..
04/25/2019  11:39 AM    <DIR>          3D Objects
04/25/2019  11:39 AM    <DIR>          Contacts
04/26/2019  10:29 AM    <DIR>          Desktop
04/25/2019  11:39 AM    <DIR>          Documents
04/25/2019  11:39 AM    <DIR>          Downloads
<some output omitted>
               0 File(s)              0 bytes
              15 Dir(s)  32,671,969,280 bytes free
```

c. In the current directory, use the **md** command to create three new folders: **ITEfolder1**, **ITEfolder2**, and **ITEfolder3**. Type **md ITEfolder1** and press **Enter**. Create **ITEfolder2** and **ITEfolder3**. (**Note**: The command **mkdir** performs the same function as the command **md**.)

```
C:\Users\ITEUser> md ITEfolder1
C:\Users\ITEUser> md ITEfolder2 ITEfolder3
```

d. Type **dir** to verify that the folders have been created.

e. Type **cd ITEfolder3** at the command prompt and press **Enter**.

Question:

Which folder are you in now?

C:\Users\Ralph\ITEfolder3>

f. Create **ITEfolder4** in the current directory. Within the **ITEfolder4** folder, create a folder named **ITEfolder5**. Use the **dir** command to verify the folder decreation.

Question:

What command or commands did you use to create the nested folders?

I used command *md ITEfolder4* to create the ITEfolder4 folder then I used command *cd ITEfolder4* to access the folder to create the ITEfolder5 *md ITEfolder5.*

g. Change directory as necessary until you are in ITEfolder5.

h. Type **cd ..** to change the current directory. Each **..** is a shortcut to move up one level in the directory tree.

Question:

After issuing the **cd ..** command, what is your directory now?

C:\Users\Ralph\ITEfolder3\ITEfolder4>

What would be the current directory if you issue this command at C:\Users\ITEUser\ITEfolder3\ITEFolder4?
Well, I just copied and paste it (with ITEUser being Ralph) it would say "The system cannot find the path specified." Then it would still error because ITEFolder4 is supposed to be ITEfolder4 the capital "F" makes it error because it's referring to whole different file at that point.

Step 3: Create text files.

a. Navigate to the **C:\Users\ITEUser\ITEfolder1** directory. Type **cd ..\ITEfolder1** at the prompt.

b. Type **echo This is doc1.txt > doc1.txt** at the command prompt. The **echo** command is used to display a message at the command prompt. The **>** is used to redirect the message from the screen to a file. For example, in the first line, the message **This is doc1.txt** is redirected into a new file named **doc1.txt**. Use the **echo** command and **>** redirect to create these files: **doc2.txt**, **file1.txt**, and **file2.txt**.

```
C:\Users\ITEUser\ITEfolder1> echo This is doc1.txt > doc1.txt
C:\Users\ITEUser\ITEfolder1> echo This is doc2.txt > doc2.txt
C:\Users\ITEUser\ITEfolder1> echo This is file1.txt > file1.txt
C:\Users\ITEUser\ITEfolder1> echo This is file2.txt > file2.txt
```

c. Use the **dir** command to verify the files are in the **ITEfolder1** folder.

```
C:\Users\ITEUser\ITEFolder1> dir
<some output omitted>
04/29/2019  08:05 AM                19 doc1.txt
04/29/2019  08:06 AM                19 doc2.txt
04/29/2019  08:08 AM                20 file1.txt
04/29/2019  08:08 AM                20 file2.txt
```

```
               4 File(s)             78 bytes
               2 Dir(s)  32,625,397,760 bytes free
```

d. Use the **more** or **type** command to view the content of the newly created text files.

```
C:\Users\ITEUser\ITEfolder1> more doc1.txt
This is doc1.txt

C:\Users\ITEUser\ITEfolder1> type doc2.txt
This is doc2.txt
```

Step 4: Copy, delete, and move files.

a. At the command prompt, type **move doc2.txt C:\Users\ITEUser\ITEfolder2** to move the file **doc2.txt** to the **C:\Users\ITEUser\ITEfolder2** directory.

```
C:\Users\ITEUser\ITEfolder1> move doc2.txt C:\Users\ITEUser\ITEfolder2
        1 file(s) moved.
```

b. Type **dir** at the prompt to verify that **doc2.txt** is no longer in the current directory.

c. Navigate to **C:\Users\ITEUser\ITEfolder2** to change the directory to **ITEfolder2**. Type **dir** at the prompt to verify **doc2.txt** has been moved.

d. Type **copy doc2.txt doc2_copy.txt** to create a copy of **doc2.txt**. Type **dir** at the prompt to verify a copy of the file has been created.

```
C:\Users\ITEUser\ITEfolder2> dir
<some output omitted>
Directory of C:\Users\ITEUser\ITEfolder2

04/30/2019  09:07 AM                19 doc2.txt
04/30/2019  09:07 AM                19 doc2_copy.txt
               2 File(s)              38 bytes
               2 Dir(s)  31,753,068,544 bytes free
```

e. Use the **move** command to move **doc2_copy.txt** to **ITEfolder1**. Type **move doc2_copy.txt ..\ITEfolder1**.

```
C:\Users\ITEUser\ITEfolder2> move doc2_copy.txt ..\ITEfolder1
        1 file(s) moved.
```

f. A copy of the file **doc2.txt** can be created in a new location using the **copy** command. At the prompt, enter the **copy doc2.txt ..\ITEfolder1\doc2_new.txt**.

```
C:\Users\ITEUser\ITEfolder2> copy doc2.txt ..\ITEfolder1\doc2_new.txt
        1 file(s) copied.
```

g. The file **doc2.txt** can also be moved to a new location with a new filename using the **move** command. Type **move doc2.txt ..\ITEfolder1\doc2_new.txt** at the prompt.

```
C:\Users\ITEUser\ITEfolder2> move doc2.txt ..\ITEfolder1\doc2_move.txt
        1 file(s) moved.
```

h. Type **dir ..\ITEfolder1** to view the content in **ITEfolder1** without leaving the current directory.

```
C:\Users\ITEUser\ITEfolder2> dir ..\ITEfolder1
<some output omitted>
 Directory of C:\Users\ITEUser\ITEfolder1
```

```
04/29/2019  12:08 PM    <DIR>          .
04/29/2019  12:08 PM    <DIR>          ..
04/29/2019  08:05 AM                19 doc1.txt
04/29/2019  08:06 AM                19 doc2_copy.txt
04/29/2019  08:06 AM                19 doc2_move.txt
04/29/2019  08:06 AM                19 doc2_new.txt
04/29/2019  08:08 AM                20 file1.txt
04/29/2019  08:08 AM                20 file2.txt
               6 File(s)            116 bytes
               2 Dir(s)  31,467,700,224 bytes free
```

i. Change the current directory to **ITEfolder1**. Type **cd ..\ITEfolder1** at the prompt.

j. Move **file1.txt** and **file2.txt** into **ITEfolder3**. To move all the files that contain the word **file** into **ITEfolder3** with one command, use a **wildcard** (*) character to represent one or more characters. Type **move file* ..\ITEfolder3**.

```
C:\Users\ITEUser\ITEfolder1> move file* ..\ITEfolder3
C:\Users\ITEUser\ITEfolder1\file1.txt
C:\Users\ITEUser\ITEfolder1\file2.txt
        2 file(s) moved.
```

k. To view the content of the file doc2_new.txt, use the **type** or **more** command at the prompt.

```
C:\Users\ITEUser\ITEfolder1> type doc2_new.txt
This is doc2.txt
```

l. Use the **ren** command to rename doc2_new.txt to doc3.txt. Use the **dir** command to display the content in the directory.

```
c:\Users\ITEUser\ITEfolder1> ren doc2_new.txt file.log
```

m. Use the **type** or **more** command to view the content of the newly rename file **file.log.**

```
C:\Users\ITEUser\ITEfolder1> more file.log
This is doc2.txt
```

n. To delete a file or multiple files, use the **del** command with the file names at the prompt.

Question:

What single command would you use to delete all the files with doc2 in the filename? Use the command to remove the files with doc2 in the filename.
del doc2_copy.txt doc2_move.txt

What command would you use to delete all the files in the directory? Use the command to remove the files.
del doc1.txt file.log

Step 5: Use the xcopy and robocopy commands.

In this step, the **xcopy** and **robocopy** commands are used to copy the content in a directory.

a. View the content of ITEfolder3.

```
C:\Users\ITEUser\ITEfolder1> dir ..\ITEfolder3
<some output omitted>
 Directory of c:\Users\ITEUser\ITEfolder3

04/29/2019  03:11 PM    <DIR>          .
```

```
04/29/2019  03:11 PM    <DIR>          ..
04/29/2019  03:05 PM                20 file1.txt
04/29/2019  03:05 PM                20 file2.txt
04/29/2019  03:01 PM    <DIR>          ITEfolder4
               2 File(s)             40 bytes
               3 Dir(s)  31,492,157,440 bytes free
```

b. Type **xcopy ..\ITEfolder3 .** at the prompt to copy the content of **ITEfolder3** to **ITEfolder1**. Note the **.** at the end of the command. It is a shortcut for the current directory.

```
C:\Users\ITEUser\ITEfolder1> xcopy ..\ITEfolder3 .
..\ITEfolder3\file1.txt
..\ITEfolder3\file2.txt
2 File(s) copied
```

c. At the prompt, type **dir** to display the content of **ITEfolder1**. Only the files in the **ITEfolder3** were copied into **ITEfolder1**. The directory **ITEfolder4** was not copied into **ITEfolder3**.

```
C:\Users\ITEUser\ITEfolder1> dir
<some output omitted>
 Directory of C:\Users\ITEUser\ITEfolder1
04/29/2019  03:16 PM    <DIR>          .
04/29/2019  03:16 PM    <DIR>          ..
04/29/2019  03:05 PM                20 file1.txt
04/29/2019  03:05 PM                20 file2.txt
               2 File(s)             40 bytes
               2 Dir(s)  31,491,321,856 bytes free
```

d. Use **help xcopy** or **xcopy /?** to determine which switch would allow the **xcopy** command to copy **all** the files and directories.

Question:

What option allows you to copy all the files and directories, including the empty directories?
xcopy /E

e. Because **ITEfolder4** is a subfolder and **ITEfolder5** is both a subfolder and an empty folder, **/E** is needed to copy all the contents of **ITEfolder3** and the empty subfolder.

Type **xcopy /E ..\ITEfolder3 .** at the prompt to copy the files. When prompted, type **a** to allow overwriting the existing files.

```
c:\Users\ITEUser\ITEfolder1> xcopy /E ..\ITEfolder3
Overwrite C:\Users\ITEUser\ITEfolder1\file1.txt (Yes/No/All)? a
..\ITEfolder3\file1.txt
..\ITEfolder3\file2.txt
2 File(s) copied
```

f. Verify that the **ITEfolder4** and **ITEfolder5** were also copied in **ITEfolder1**.

```
c:\Users\ITEUser\ITEfolder1> dir
<some output omitted>
 Directory of c:\Users\ITEUser\ITEfolder1

04/29/2019  04:41 PM    <DIR>          .
04/29/2019  04:41 PM    <DIR>          ..
04/29/2019  03:05 PM                20 file1.txt
```

```
04/29/2019  03:05 PM                20 file2.txt
04/29/2019  04:41 PM    <DIR>          ITEfolder4
               2 File(s)             40 bytes
               3 Dir(s)  31,493,193,728 bytes free

c:\Users\ITEUser\ITEfolder1> dir ITEfolder4
<some output omitted>
 Directory of c:\Users\ITEUser\ITEfolder1\ITEfolder4

04/29/2019  04:41 PM    <DIR>          .
04/29/2019  04:41 PM    <DIR>          ..
04/29/2019  03:00 PM    <DIR>          ITEfolder5
               0 File(s)              0 bytes
               3 Dir(s)  31,493,193,728 bytes free
```

g. The robocopy command can also be used to copy the content of a directory to a new destination. The robocopy command has more capabilities than the built-in Windows copy and xcopy command, such as resume copying after a network interruption, skip files that appear to be identical to the files in the destination folders, and mirror a directory by keeping the destination directory in sync with the source directory.

   Copy the ITEfolder4 content to ITEUser folder using the robocopy command.

   ```
   C:\Users\ITEUser\ITEfolder1> robocopy /E
   C:\Users\ITEUser\ITEfolder3\ITEfolder4\ C:\Users\ITEUser
   ```

   Note the information provided by the command during the copying process.

h. Navigate to **C:\Users\ITEUser** to verify that the folder ITEfolder5 is copied.

Step 6: Delete directories.

In this step, you will delete an empty and a non-empty directory using the **rd** command.

a. Use the **rd ITEfolder2** to delete the empty directory and verify that the directory was deleted.

   ```
   C:\Users\ITEUser> rd ITEfolder2
   ```

b. Navigate to the **C:\Users\ITEUser\ITEfolder3** directory.

c. Use the **rd ITEfolder1** to delete the directory. Verify the directory removal using the **dir** command.

   Were you able to delete the directory? Explain.
   I couldn't delete ITEfolder1 because I was in a different directory C:\Users\Ralph\ITEfolder3 compared to the directory where ITEfolder1 was located which is C:\Users\Ralph.

d. Use **rd /?** command to determine the switch that allows the deletion of a non-empty directory.

   ```
   c:\Users\ITEUser\ITEfolder3> rd /S ITEfolder1
   ITEfolder4, Are you sure (Y/N)? y
   ```

e. Use the appropriate commands to delete all the text files and folders that you have created in this lab. Type **exit** to close the command prompt window.

Reflection Question

What are the advantages of using CLI vs. GUI?

The advantage of CLI (Command-Line Interface) is that once commands are known, tasks can be completed way faster without navigating through menus or windows. CLI also consumes significantly less CPU and memory compared to GUI-based applications because of its very simple interface, making it ideal for low-

resource environments. CLI also offers greater control and flexibility; it has access to more advanced options that aren't always shown in a GUI and can chain commands using pipes, redirection, and many more for a more powerful workflow.

The advantage of GUI, however, is its ease of use; it is user-friendly, especially for beginners, with its visual elements (buttons, menus, and icons), which reduces the learning curve a lot for some people. People are also able to have visual feedback because it is easier to see what's happening (file transfers, image editing) and immediate visual confirmation of actions (drag-and-drop, progress bars to indicate completion). Users can also explore options through menus rather than remembering commands. Tooltips, help pop-ups, and visual cues can also guide the user easier, but that sometimes depends on the person, whether they are more visual or a written person.

Conclusion:

I've learned how to make, copy, and move folders using CLI. I never realized I was able to do any of that in the CLI in the first place, to be honest, so it amazed me. The downside of it, of course, is memorizing/familiarizing yourself with all the basic commands to be able to operate efficiently and effectively to have a steady workflow. This hands-on activity has also taught me that there is more potential in CLI than GUI. Of course, GUI is easier to understand visually, but for us to be able to work faster and use fewer resources of our PC, using CLI dramatically increases efficiency for the user. The downside is that CLI has a lot of more advanced options and controls, which could overwhelm a lot of users if they are beginners, but the upside, however, is that CLI has a lot of more advanced options and controls, which could drastically help with more advanced workloads and complicated tasks once learned through experience of using CLI. This experience has shown me that while the CLI may have a steeper learning curve, the payoff in speed, control, and flexibility is well worth the effort and time. With continued practice, I can see myself becoming much more proficient and confident in handling complex tasks directly through the command line.