This is a Simple Watcher program for managing Restricted PC's.

This program creates and monitors a directory called Watch under the public desktop (C:\Users\Public\Desktop\Watch).

When a new file is added to the watched directory:

- If the file is an .exe this program will execute it.
- If the file is a .txt this program will treat it as a shell script file and run its contents line by line.
- If the file is a .dll this program will load it and call an export function called “runCommand” of type (void (__cdecl *RUNCMD)(DWORD)) with the current PID as an input argument.
- Any other file type will not be treated.
- After any of the above the program will attempt to delete the new file.

Command line Params:

- -install will install the service.
- -delete will delete the service.
- no cmd line will start the service.

Installation and deletion should be done as Admin!

Execution routine:

- The “ServiceWorkerThread” creates the watched directory (C:\Users\Public\Desktop\Watch). and opens a handle to it.
- It then enters a calling “WatchDir” in each iteration
- “WatchDir” uses “ReadDirectoryChangesW” to monitor changes in this directory and then calls “HandleNotification” for each notification.
- “HandleNotification” determines whether this notification is of type FILE_ACTION_ADDED and if so it parses the notifications file name to extract its extension.
- It then runs “RunExe”, ”RunScript” or “LoadDll” accordingly afterwards deleting the file.
- “RunExe” simply creates a new process with the given exe path.
- “RunScript” reads the given text file line by line, executing each line in the shell using “system” command
- “LoadDll” loads the given dll using “LoadLibrary”, then uses “GetProcAddress” to find the address of the export “runCommand”, finally it executes runCommand with the current process PID obtained by “GetCurrentProcessId” and frees the library.

Possible Problems:

- The RunScript Command dose not parse each line not check its content type before handing it to system command. This might cause bugs given incorrect input.
- The notifications are received before the file has finished copying to the watcher directory thus the file handling might fail with the ERROR_SHARING_VIOLATION code. Therefore, I added a sleep time of 2000ms at the beginning of each handling method.
- The Watched Directory is in the Public Desktop thus anyone that has access to it might use it. It is Recommended to change the watched directory to a different directory to which only the admin is aware/has access rights to.
- We might want to use scheduled tasks to run the service on a specific user logon/command and stop it upon logout.