

DiskCleaner - A Windows service

Clément Nicolas

July 25, 2014

Part I

Project

1 Aim

DiskCleaner is a project for ETNA school. The aim is to install a Windows service that cleans a directory, which path is stored in a registry key.

2 Technology used

For this, we use the Windows API.

Part II

Functions

3 Main function

The main function.

```
int    main(int argc, char* argv[])
{
    BOOL logfileready = TRUE;
    SERVICE_TABLE_ENTRY table[] = { { SERVICE_NAME, ServiceMain }, {
        NULL, NULL } };

    g_LogFile = CreateFile(SERVICE_LOG_FILE, GENERIC_WRITE, 0, NULL,
        CREATE_ALWAYS, FILE_ATTRIBUTE_NORMAL, NULL);
    if (g_LogFile == INVALID_HANDLE_VALUE)
    {
        logfileready = FALSE;
        fprintf(stderr, "Main => CreateFile() failed with :
            %d\nThe service won't log correctly...\n",
            GetLastError());
    }

    if (argc > 1)
    {
        if (strcmp(argv[1], "-i") == 0)
            return (launchInstall(logfileready));
        else if ((strcmp(argv[1], "-p") == 0) && (argc > 2))
            return (launchSetPath(argv[2], logfileready));
        else if (strcmp(argv[1], "-d") == 0)
            return (launchDelete(logfileready));
        else
        {
            fprintf(stderr, "Bad argument. Expected :\n\t[-i]
                to install\n\t[-d] to delete\n\t[-p path] to
                set a path to clean");
            return (EXIT_FAILURE);
        }
    }
    else
        StartServiceCtrlDispatcher(table);

    if (!CloseHandle(g_LogFile))
        fprintf(stderr, "Main => CloseHandle failed with : %d\n",
            GetLastError());

    return (EXIT_SUCCESS);
}
```

4 Main function

InstallMyService. This function sets up the service.

```
BOOL          InstallMyService()
{
    char        strDir[MAX_PATH + 1];
    SC_HANDLE    schSCManager;
    SC_HANDLE    schService;

    if (!GetCurrentDirectory(MAX_PATH, strDir))
    {
        writeInLogFile("InstallMyService => GetCurrentDirectory()
            failed with : ", GetLastError());
        writeInLogFile("\r\n", ERROR_SUCCESS);
        return (FALSE);
    }
    lstrcat(strDir, "\\\"SERVICE_BIN_NAME);

    if ((schSCManager = OpenSCManager(NULL, NULL,
        SC_MANAGER_ALL_ACCESS)) == NULL)
    {
        writeInLogFile("InstallMyService => OpenSCManager()
            failed with : ", GetLastError());
        writeInLogFile("\r\n", ERROR_SUCCESS);
        return (FALSE);
    }

    schService = CreateService(schSCManager, SERVICE_NAME,
        SERVICE_DESCRIPTOR,
        SERVICE_ALL_ACCESS, SERVICE_WIN32_OWN_PROCESS,
        SERVICE_DEMAND_START, SERVICE_ERROR_NORMAL,
        (LPCSTR)strDir, NULL, NULL, NULL, NULL, NULL);

    if (schService == NULL)
    {
        writeInLogFile("InstallMyService => CreateService()
            failed with : ", GetLastError());
        writeInLogFile("\r\n", ERROR_SUCCESS);
        return (FALSE);
    }

    if (!CloseServiceHandle(schService))
    {
        writeInLogFile("InstallMyService => CloseServiceHandle()
            failed with : ", GetLastError());
    }
}
```

```
        writeInLogFile("\r\n", ERROR_SUCCESS);  
        return (FALSE);  
    }  
    return (TRUE);  
}
```
