

Hiding program using STARTUPINFO flags

Hiding program using STARTUPINFO flags

```
BOOL CreateProcess(  
    lpApplicationName,    // Path to the executable  
    lpCommandLine,        // Command-line arguments  
    lpProcessAttributes,  // Security attributes for the process  
    lpThreadAttributes,   // Security attributes for the primary thread  
    bInheritHandles,      // Inherit handles from parent process  
    dwCreationFlags,       // Process creation flags  
    lpEnvironment,        // Pointer to environment block  
    lpCurrentDirectory,   // Working directory of the new process  
    lpStartupInfo,        // Pointer to STARTUPINFO structure  
    lpProcessInformation  // Pointer to PROCESS_INFORMATION structure  
);
```

It controls how the new process starts (window size, position, appearance, etc.).

FLAGS

1. STARTF_USESHOWWINDOW	(0x00000001)
2. STARTF_USESIZE	(0x00000002)
3. STARTF_USEPOSITION	(0x00000004)
4. STARTF_USECOUNTCHARS	(0x00000008)
5. STARTF_USEFILLATTRIBUTE	(0x00000010)
6. STARTF_RUNFULLSCREEN	(0x00000020)
7. STARTF_FORCEONFEEDBACK	(0x00000040)
8. STARTF_FORCEOFFFEEDBACK	(0x00000080)
9. STARTF_USESTDHANDLES	(0x00000100)
10. STARTF_USEHOTKEY	(0x00000200)

```
typedef struct _STARTUPINFO {  
    DWORD cb;                // Size of the structure  
    LPSTR lpReserved;        // Reserved (always NULL)  
    LPSTR lpDesktop;          // Desktop name (NULL = default)  
    LPSTR lpTitle;           // Console title (NULL = default)  
    DWORD dwX, dwY;          // Window position (ignored if not set)  
    DWORD dwXSize, dwYSize;  // Window size  
    DWORD dwFlags;           // specify which flag is used (e.g., SW_HIDE)  
    WORD wShowWindow;        // Controls the window state (SW_SHOW, SW_HIDE, etc.)  
    ... (Other fields not commonly used)  
} STARTUPINFO;
```

Code :

```
#include <windows.h>
#include <stdio.h>

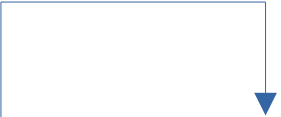
int main() {
    STARTUPINFO si = { sizeof(si) }; // setting zero
    PROCESS_INFORMATION pi = {0};    // setting zero

    si.dwFlags = STARTF_USESHOWWINDOW; // Use wShowWindow
    si.wShowWindow = SW_HIDE; // Start Notepad in hidden mode

    // Create Notepad process
    if (CreateProcess(
        "C:\\Windows\\System32\\notepad.exe", // Application name
        NULL, // No command-line arguments
        NULL, NULL, FALSE, // Default security
        0, // No special creation flags
        NULL, NULL, // Use parent's environment and directory
        &si, &pi // Pass structures
    ))
    {
        printf("[+] Process created successfully!\n");
        printf("[+] Process ID: %lu\n", pi.dwProcessId);
        printf("[+] Thread ID: %lu\n", pi.dwThreadId);

        // Close process handles
        CloseHandle(pi.hProcess);
        CloseHandle(pi.hThread);
    }
    else {
        printf("[-] Failed to create process. Error: %lu\n", GetLastError());
    }
    return 0;
}
```

Command Flag



SW_SHOW
SW_HIDE
SW_MINIMIZE
SW_MAXIMIZE