

```

//Only finds and patch the NT installation directory to Program
BOOL WINAPI ExecPgm(HWND hWnd,LPTSTR szProgramString, LPTSTR Args)
{
    TCHAR szProgramPath[512];
    PTCHAR pszWindir;
    pszWindir = _wgetenv(TEXT("windir") );
    _tcscpy(szProgramPath, pszWindir);
    _tcscat(szProgramPath, TEXT("\\system32\\") );
    _tcscat(szProgramPath, szProgramString );
    ExecAndWait(hWnd, szProgramPath, Args);
    return TRUE ;
}
// Run Program and Wait for it to end
BOOL WINAPI ExecAndWait(HWND hWnd, LPTSTR szProgramString, LPTSTR
Args)
{
    BOOL bFlag;
    DWORD dwRetVal;
    PROCESS_INFORMATION procInfo;
    STARTUPINFO sui;
    ShowWindow(hWnd, SW_HIDE);
    setStartupInfo(&sui);
    bFlag = CreateProcess(szProgramString,
        Args, //no command line
        NULL, //no process security
        NULL, //no primary thread security
        FALSE, //doesn't inherit handles
        0, //no creation flag
        NULL, //no new environment
        NULL, //no startup directory
        &sui, //STARTUPINFO struct
        &procInfo // PROCESS_INFORMATION struct
    );
    //Test Error
    //Wait for the spawned process to complete.
    dwRetVal = WaitForSingleObject(procInfo.hProcess,INFINITE);
    switch(dwRetVal) {
    case WAIT_ABANDONED:
    case WAIT_TIMEOUT:
        {ShowErrorMsg(hWnd, TEXT("WaitForSingleObject failed
        WAIT_ABANDONED|WAIT_TIMEOUT"));
        ShowWindow(hWnd, SW_SHOW);
        SetFocus(hWnd);
        return FALSE;
        }
    case WAIT_OBJECT_0:
        {
            ShowWindow(hWnd, SW_SHOW);
            SetFocus(hWnd);
            return (TRUE);
        }
    }
}

```

Programmers Society Inc

Previous