



## Open & Save Dialog Help Guide

### About:

Open & Save Dialog Version 1.0.0

Copyright (c) 2018-2023 by Mark L. Alberi. All rights reserved.

The Open & Save Dialog Executable API (OpenSaveDialog) is a common dialog external executable binary application used as an independent non embedded systems application to reduce or illuminate file systems disruptions caused by the common dialog object having to pause your application during running time to open a file you need to process by your application. It uses a two way file system thread communication through OS 3.11 ini header communication. The [Header Name] is a secure way to communicate for some types of solutions saving registry kernel disruptions and unnecessary registry bloating. The ini file is created used and deleted within application run time milliseconds and is completely binded preventing tampering during its usage by the OpenSaveDialog API and then by your application after it reads the return contents and then deletes it, preventing any run time security obstruction. You define in your program as an open or save dialog with your own defined file filter parameters, the OpenSaveDialog API sets the filter parameters and opens the dialog you called. The OpenSaveDialog application returns its file query to a header formatted ini file for you to read once it closes and process the file path it returned. File system disruption is greatly reduced or even eliminated because it's your application that pauses itself monitoring the OpenSaveDialog running process & waiting for the OpenSaveDialog object API to finish and exit before continuing. The OpenSaveDialog API pauses only itself causing no file system disruptions to the kernel. It was primary written for server system script administration by VB script wscript, command line batch or powershell script interpreter compilers, giving them a safe way to utilize the windows common dialog object as a replacement solution for the obsoleted HTA script solutions allowing programmers to call windows functions such as the common dialog through embedded HTA script. SoftGlue has a working VB6 & VB.Net model of OpenSaveDialog Object API. VB6 model offers the vb script API as a CMDFileInfo.ini [Header Name] option. VB.Net uses the new Open and new Save dialog objects.

### Common Dialog Thread Demo:

To demonstrate the common dialog executable API, oscmdalg an independent thread application was written that calls oscmdalg.exe. The two programs communicate by the file system threading as described in the About above. oscmdalg.exe will not run on its own. It can only run through any application threaded to it. As evidence of this, visit Task Manager Details, and observe the running thread call, CMDThread.exe and the oscmdalg.exe it calls before you open a file path and read that file path in the final message displayed. By running both oscmdalg.exe & CMDThread.exe through binary analysis you will clearly observe CMDThread.exe calls a common dialog object only program and threads to it through the file system. SoftGlue encourages the licensed use to do so, leave any commits on the Commits page on SoftGlue.net. To help get the programmer started, see the free shareware VB Script example below. The purpose of it is to demonstrate, how to save the CMDFileInfo.ini file and how to call it with VB Script as object.exec shell command. This is so an application paus can be done using the status parameter. See CMDThreadInterface.vbs Demo.

As VB.net, the shell command is used to call oscmdalg and a wait process function to wait until the oscmdalg common dialog application has ended, before continuing and displaying the file path that was called.



**Startup:**

Selecting "Yes" in startup window, Image 1 will load the open common dialog  
Selecting "No" in startup window, Image 1 will load the save common dialog  
Selecting "Cancel" in startup window, Image 1 will open common dialog demo thread

**Note:**

When selecting the Save Common Dialog, to thread the save file path to the final message box, must select replace as "Yes" for the Save Common Dialog to transfer the file path. Nothing is replaced because all the Save Common Dialog object does is read the file path and return it. This is a Save Common Dialog run time return property.

**CMDFileInfo.ini file Format:**

Header	Value
API Reference Switch - [CMDSWITCH]  Visual Basic 6.0 Only	True for API CommonDialog script False for Microsoft CommonDialog object
File Access Filter - [CMDFILTER]  Save each parameter as new line entry after the CMDFILTER header by defining a file type title followed by a file extension when saving the file access filter parameters to CMDFileInfo.ini. The Common Dialog API executable oscmdalg will format for the run time common dialog object.	Executable *.exe VB Script *.vbs PowerShell *.ps1 Java Script *.js Batch Files *.bat Command Files *.cmd All Files *.*
Common Dialog Title - [CMDTITLE]	Example: Open Thread
Open & Save Dialog Switch - [SAVECMD]	True = Load Save Dialog False = Load Open Dialog



### Sample Script:

The sample script below is free to use and is only licensed to be used with the Open & Save Dialog application. It is a sample starter visual basic script that can be used to run the “Open & Save Dialog Demo”, tailored to your own needs.

**Note:** The CMDThread.exe in the licensed shareware sample script is a user input VBA program to the Open & Save Dialog application, oscmdalg.exe on the command line. It was originally written in both vb script and VBA6 & Vb.net. The command line console shows both command line formats for developer convince. The developer needs make their own command line thread script or copy the script below as a starter script.

### Visual Basic .Net

```
Dim AppPath = "C:\users\" & System.Environment.UserName & "\Root App Path\AppConfig.dat" As String
```

```
'Show or hide command console.
```

```
Dim cHide = AppWinStyle.NormalFocus / AppWinStyle.Hide As AppWinStyle
```

```
'Code double quotes " as Chr(34) ASCII code is optional.
```

```
Shell(Chr(34) & AppPath & Chr(34) & CMD1, cHide)
```

```
Call MonitorProcess("oscmdalg")
```

```
.
```

```
Public Sub MonitorProcess(ByVal GetProcessName As String)
```

```
    'Monitor the active common dialog running process  
    'and exit when it closes.
```

```
    Dim GetProcess As Process  
    Dim ProcessCnt As Integer = 1
```

```
    Do Until ProcessCnt = 0
```

```
        .                               'Do something
```

```
        .
```

```
        For Each GetProcess In Process.GetProcesses
```

```
            If (GetProcess.ProcessName = GetProcessName) Then
```

```
                'Do something
```

```
            End If
```

```
        Next
```

```
        .                               'Do something
```

```
        .
```

```
    Loop
```

```
End Sub
```

**Sample Script (Continued):**

**Note:** The vbs script full example below has to be translated into VB.net to work as VB.net script. The above is fundamental code not a part of vbs script translation, but unique to .net coding and WScript run command is not supported by .net.

**Sample Script CMDThreadInterface.vbs:**

'Copyright (c) by Mark L. Alberi 2018-2022  
 'The CMDThreadInterface-Rev{N} is a OpenSave common dialog tread demo shareware script which  
 'can be imported Into the programmers actual vb script for intgerfacing the the OpenSaveCMDThread-Rev{N}.exe  
 'thread script interface to the vb script common dialog oscmdalg.exe application.  
 'The CMDThreadInterface-Rev{N} is written with vbsedit

**Option Explicit**

```
Public AppPath
Public AppName
Public fso, MyFile
Public ObjShell
Public OpenFileDialogInfo
PublicRetVal
```

```
Const ForReading = 1
Const ForWriting = 2
```

**On Error Resume Next**

```
AppPath = WScript.ScriptFullName
AppName = WScript.ScriptName
AppPath = Replace(AppPath,AppName,"OpenSaveCMDThread-Rev1.exe",1)
```

```
Set ObjShell = CreateObject("WScript.Shell")
SetRetVal = ObjShell.Exec (Chr(34) & AppPath & Chr(34) & chr(32) & "false" & chr(32) &
""""Open Thread"""" & chr(32) & "false" & chr(32) & _
""""Executable (*.exe);*.exe;VB Script (*.vbs);*.vbs;PowerShell (*.ps1);*.ps1;Java Script (*.js);*.js;Batch Files (*.bat);"""" & _
""""*.bat;Command Files (*.cmd);*.cmd;All Files (*.*)*.*****"""" ' Run General Common Dialog.
```

'Input arguments must be placed in the above order.

```
Do Until (RetVal.Status = 1)
  If(RetVal.Status = 1) Then: Exit Do
Loop
```

```
Function GetCMDInfo()
```



```
'Get user queried common dialog full file name.  
'and delete the information file. Use script  
'in other support applications
```

```
AppPath = WScript.ScriptFullName  
AppName = WScript.ScriptName  
AppPath = Replace(AppPath,AppName,"CMDFileInfo.ini",1)  
Set fso = CreateObject("Scripting.FileSystemObject")  
Set MyFile = fso.OpenTextFile(AppPath, ForReading)
```

```
GetCMDInfo = MyFile.ReadLine
```

```
Set MyFile = Nothing
```

```
'Delay next command for 0.5 seconds to give chance to kill object  
Dim EndTime  
EndTime = DateAdd("s", 0.5, Now)
```

```
Do until now >= EndTime  
loop
```

```
'Delete Open Common dialog information file once done using it  
Set MyFile = fso.GetFile(AppPath)  
MyFile.Delete()
```

```
End Function
```

```
OpenFileDialogInfo = GetCMDInfo
```

```
'Trap any errors and clear them  
If(err.number = 0) And (OpenFileDialogInfo <> "[cancelcmd]") Then  
    MsgBox OpenFileDialogInfo,vbOKOnly + 64, "Common Dialog Retrieved File"  
Else  
    'Cant find CMDFileInfo.ini  
    err.Clear  
End If
```



**Image 1:**

**Open & Save Dialog Startup Window:**

