

SCREEN SAVER IN VISUAL BASIC 6



- Somdutt Ganguly

Namaste, friends

Somdutt Ganguly here just wanted to present you a tutorial on making screen saver in visual basic. Making screen savers in visual basic is easy. It's very very easy. But there are not many people who are aware of such stuff. But don't worry! If you don't know to make screen savers then go for this tutorial. So, let's start...and I am sorry for any grammatical/typing/technical errors. If any?

SO, WHAT THE HELL ARE SCREEN SAVERS AND WHAT FOR AND WHY YOU MUST KNOW ABOUT IT?

For any one who is new to computers..he is introduced to screen savers as if they are mend for entertainment or to show the computer more lively. The screen savers generally come into play when the computer is untouched for some time. There is a technical reason for this. They are here to stop phosphor burn-in of graphics or text on a screen. Actually image over the screen is formed by bombardment of phosphor over the Screen (Actually over Cathode ray tube) continuously. Now if at a particular area of the screen the phosphor continuously falls then it makes the screen burn i.e. it may permanently burn an image over the screen. Don't think that there will be fire out!!!...haha!. I haven't seen such a burn but you may notice it at ATMs sometimes when u can see shadow like thing at the main window (or menus). This is due to screen burning. It is due to displaying the same screen for hours and hence burning the image permanently into the screen. Hence, programmers have come out with the concept of screen saver to avoid such a screen burns. Now with the passage of time screen savers are basically used for fun purpose. In short it is now there for business. There are companies/individuals who sell screen savers. If you are not able to earn any amount by selling your software..then..making a screen saver is a good option..hah!. Why not..If you are living in a good location then why not take photographs of beautiful scenes and embed them into a screen saver. Also, screen savers are very good for advertisement purpose. If you have a dazzling screen saver for your company / bio-data whatever?...that would really work... why not embed a flash movie or video or sound into your Visual basic form and make a screen saver. Anyway!!....let's do this stuff...by programming!!

BIGGEST MISTRY FOR SCREEN SAVER'S NON-PROGRAMMER?

what's the difference between an exe (executable) file and scr (screen saver)
file...!!!!..think!!

NO difference.....only the extension is different...haha!.....hence, there won't be any problem to make setups!!.....just make sure that you place it on the windows subdirectory where other window screen savers are kept the screen saver property window will automatically load it.

NOW LET's MAKE A SCREEN SAVER

WHAT YOU MUST KNOW IN ORDER TO MAKE A SCREEN SAVER?

- HOW WINDOWS CALLS THE SCREENSAVER?**
- HOW YOUR VISUAL BASIC APPLICATION UNDERSTANDS THE WINDOWS CALL FOR SCREENSAVER?**
- WHEN THE SCREEN SAVER MUST UNLOAD.**
- HOW TO HIDE/UNHIDE THE CURSOR.**
- HOW TO LINK THE SCREEN SAVER WITH WINDOWS FOR MAKING THE SAVER REALLY WORK.**

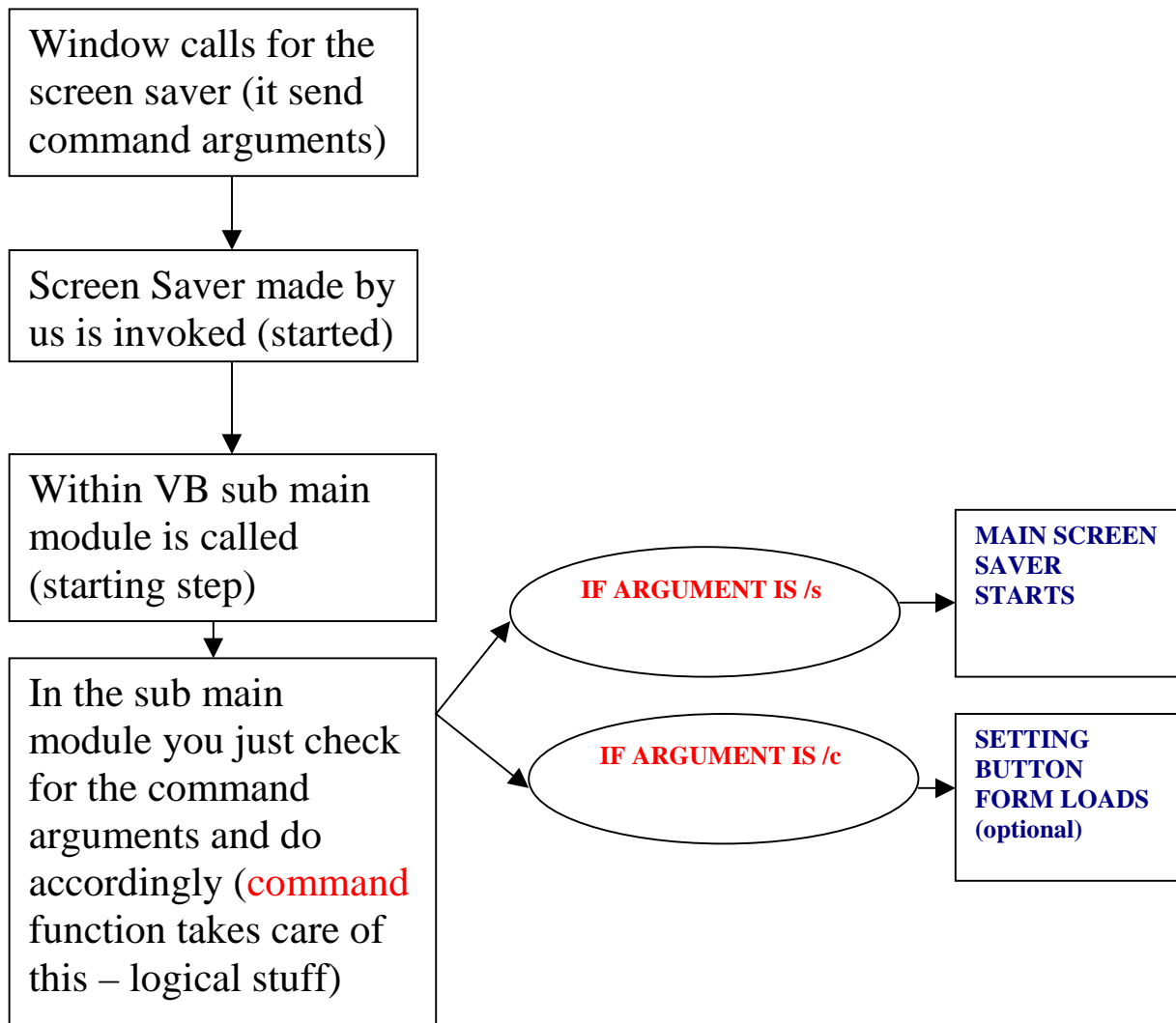
Let's go step by step to understand the above steps:

Windows calls the screen saver by passing command line argument. There are basically two arguments via which windows calls the screen saver.

- /s - when windows calls the screen saver
- /c - when the windows screen saver setting button is pressed in the display properties.

Now you know that windows passes /s or /c to call the screen saver. Now just think how your visual basic application will know about this call. There is a function named command available with visual basic which returns the argument portion of the command line. Hence, this is it.

From above we can draw the following picture:



Now, the next step is to stop the screen saver when mouse or keyboard is moved or pressed respectively. So in the following Visual basic form events you must unload the form.

- keypress
- keydown
- keyup
- mousedown
- mousemove
- mouseup

I hope you are understanding my point. That's why screen savers stop running when you move the mouse or strike the keyboard or whatever!!!

Also another step is to hide the cursor. This is the most important step since, if the cursor is on it will distract the user. So there is a api call for it. You must use the below api call. Showcursor api call is used to achieve this.

Declare Function ShowCursor Lib "user32" (ByVal bShow As Long) As Long

Now if false is passed to showcursor then cursor hides otherwise if it is true then cursor becomes visible.

**TO MAKE THE
CURSOR HIDE USE
ShowCursor FALSE**

**TO MAKE THE CURSOR
VISIBLE USE
ShowCursor TRUE**

Also now when the screen saver starts you must use showcursor false to make the cursor invisible. Now when the screen saver stop showcursor true is used to make the cursor visible. I hope you get this logic.

After you make the screen saver what you must do to make it really work. First of all from file menu click the make filename.exe option. When the dialog box comes to give a name and specify a location, give any name and also give .scr extension. No proceed. Now, you will see that filename.scr is formed. Hence, hurray!!..your hard work is done. Now when you run this filename.scr file it runs just like a screensaver.

Now the next small milestone is how to set it in the Display property. This is very easy. Copy the filename.scr file and place it in the windows subdirectory where the other .scr files are kept. In windows 2000 you must place it in C:\WINNT\system32 folder. In case of windows 98 you must place filename.scr file in d:\windows\system folder. In short what you must do is search for files with .scr files. Now whatever result you will get just get the path and place it there. Got it!!!

If you use Package and Deployment Wizard to distribute y our item then please don't forget to specify the .scr file's installation path as (WinPath)\. Also in the setup.LST modify the default director to (WinPath)\.

Now the windows will automatically take care of loading it in the display properties. You can select the screen saver which you have made by selecting from the combo box.

So let's directly go on to coding (let's first make just a basic screen saver)

Open VB and select .exe type project. Now a VB IDE will open with form module open. Now first of all add one module into your project and give any name you like. Yes, obviously you must know the basic knowledge of VB in order to go for this code...right!!.

In the module make a sub main module. Sub module must be the startup object. Now place the below code exactly the same order as I am writing.

```
Sub main()
```

```
  If App.PrevInstance Then End
```

```
  If InStr(Command, "/s") > 0 Then
```

```
    Call CursorOff
```

```
    frmmain.Show
```

```
  ElseIf InStr(Command, "/c") > 0 Then
```

```
    ' MsgBox "There is no setup", _
```

```
    ' vbInformation, " Screen Saver"
```

```
    frmsettings.Show
```

```
  Else
```

```
    Call CursorOff
```

```
    frmmain.Show
```

```
  End If
```

```
End Sub
```

As shown above the following things are done:

App.previnstance checks if there is any previous instance of the same program. Also command function returns the arguments of the command line. Here, we have logically adjusted with instr function to see whether it is "/s" or "/c". This I have talked about previously. If it is /s then the screen saver starts by making the screensaver window opens. Here it is indicated by frmmain.show. Also cursor gets off.

This is it!!

Now whatever you want to display or show you do it in frmmain window. Just use timers and make any animation.

Also in frmmain form keypress ,Keydown, keyup, mousedown, mousemove, mouseup events just unload the form and also make the cursor visible.

`example

```
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
    CursorOn
End
End Sub
```

Also here are the cursoron and cursoroff functions.

Declare in the module the following :

```
Declare Function ShowCursor Lib "user32" (ByVal bShow As Long) As Long
```

And make the following procedures (for making the cursor off and on)

```
Public Sub CursorOff()
    ShowCursor False
End Sub
```

```
Public Sub CursorOn()
    ShowCursor True
End Sub
```

There is now one problem mousemove event is automatically triggered when you load the screen saver. And in mousemove event you have unloaded the screensaver. Hence, there is a problem. But it is necessary to unload the screensaver when you move your mouse. So, what we want to do is by pass the mousemove event when the form once get's loaded. This is very easy. This is when the **boolean** value becomes handy.

Just declare a variable

```
Dim boolvalue As Boolean
```

Now when the form loads set

```
Boolvalue=True
```

And in mousemove event do this

```
Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If boolvalue = False Then
        CursorOn
    End If
End Sub
```

I think you caught my logic. What I am doing is avoiding the **form_mousemove** event when the form loads for the first time by using the boolean value. Since, on start up **boolvalue=true** and since, **frmmain** gets unloaded only when **boolvalue=false** so it avoids it for the first time. Now the question is how the **boolvalue** becomes **false**. HAHAAH!! That's easy just make a timer of half a second and set it to false.

Eg:

```
`1000 millisecond timer
```

```
Private Sub timerman_Timer()
```

```

If lblLabel.ForeColor = vbWhite Then
lblLabel.ForeColor = vbRed
Else
lblLabel.ForeColor = vbWhite
End If
boolvalue = False
End Sub

```

'5 millisecond timer. Also place a shape named shapte2. It must be a square with white borders. This is for giving screen saver effects.

```

Private Sub Timer1_Timer()
Shape2.Left = Rnd(Me.Height) * Me.Height
Shape2.Top = Rnd(Me.Height) * Me.Height
End Sub

```

As above I have set the `boolvalue` to `false` using a timer. Also, in the timer I have embedded a simple logic for screen saver. This is just a basic example for your learning purpose..ok. The logic shows how the screen saver blinks red and white at once, it is loaded. Here, `lblLabel` is the lable set in the form. Also, in the screen saver window i.e. `frmmain` in our case you must set the `controlbox` property to `false` so that it doesn't display the title. Also, remove the caption (Title) whatever. Set the `windowstate` to `maximized`. Also `border style` must be `fixed single`. Don't forget to make all these changes in the property otherwise it won't be a complete screen saver.

One thing I have forgotten to tell you, we have already accomplished the logic of screen saver main window i.e. what the user sees. As we know window passes two arguments i.e. `/s` and `/c` , this we have discussed before. But, we have made the logic for `/s` but not for `/c`. By `/c` argument windows meant to tell us to open the customization window. This is optional, if we don't want to give any customization window just display a message box otherwise open a window which can change the setting of the screen saver. Hence, we are required to open another window i.e. customization window for `/c` command. So, what's next let's dig in.

Yes, one thing whatever customized changes we will do we have to store the setting forever right. For storing the changes we have numerous options such as storing the setting details in a initialization file or a registry. I think the best option is to store it in the registry.

So, we will study to store the settings in a registry. Obviously you can use the file handling technique using file system objects to store in a file.

Here, we have made a form named `frmsetting`, where we have designed the logic for changing the setting of the screen saver. Just make a textbox and a button in the form and do the following:

```

Private Sub cmdclose_Click()
SaveSetting App.Title, "startup", "message", Text1.Text
On Error Resume Next
Unload frmmain
On Error Resume Next
Unload Me
End Sub

```



```
Private Sub Form_Unload(Cancel As Integer)
On Error Resume Next
Unload frmmain
On Error Resume Next
Unload Me
End Sub
```

As shown above, savesetting is used to save the settings within the registry. Just check the syntax. The value in the text1 is saved in the registry.

Also, the following logic will be added to the form load event of frmmain form so that lbllabel can easily retrieve the value from the registry.

```
Private Sub Form_Load()
Dim setting As String
setting = GetSetting(App.Title, "startup", "message", "hello")
If Len(Trim(setting)) = 0 Then
lbllabel.Caption = "I LOVE YOU!!"
Else
lbllabel.Caption = setting
End If
boolvalue = True
End Sub
```

I hope you understand. Getsetting is used to retrieve the value from the registry key where we have previously saved the data.

Now if there is no value then the default "I LOVE YOU!!" will be displayed otherwise registry data will be displayed. OK!

Now you can test your screen saver. Click make filename.exe and while saving the location just give .scr extension. Place it in C:\windows\system folder in win 98 or C:\windows\system32 in win 2000 whatever!!!..just check out where the other screen saver files are kept and place it there..ok!!

Now you can choose to run your screen saver from display setting. Just choose it from the combo box. Also when you click setting that frmsetting window will open. This is just there for the configuration. Whatever!

That's it!!!... You have just made your first screen saver in your life

Now some of the extra things we have to take care of. Let's do it! As we have talked about previously that /s argument is passed by windows in order to indicate the starting of screen saver and /c for showing settings. Apart from this there is another argument i.e. /p to show the preview for screen saver. As we have seen in various screen savers that the screen saver can be seen in the preview mode in a small window embedded within the display property. Computer does not automatically do this, for this we have to set the logic. So, therefore this is an important stuff and we have to deal with it.

Now the logical thing depends on numerous APIs. Just go through them most of them are self-explanatory. Don't harass me for this.

```

Declare Function GetWindowLong Lib "user32" Alias "GetWindowLongA"
(ByVal hwnd As Long, ByVal nIndex As Long) As Long
Declare Function GetClientRect Lib "user32" (ByVal hwnd As Long, lpRect
As RECT) As Long
Declare Function SetWindowLong Lib "user32" Alias "SetWindowLongA"
(ByVal hwnd As Long, ByVal nIndex As Long, ByVal dwNewLong As Long)
As Long
Declare Function SetParent Lib "user32" (ByVal hWndChild As Long, ByVal
hWndNewParent As Long) As Long
Declare Function SetWindowPos Lib "user32" (ByVal hwnd As Long, ByVal
hWndInsertAfter As Long, ByVal X As Long, ByVal Y As Long, ByVal cx As
Long, ByVal cy As Long, ByVal wFlags As Long) As Long

```

Also, declare the above in the module and also declare some the variables as follows.

```

Type RECT
    Left As Long
    Top As Long
    Right As Long
    Bottom As Long
End Type

```

`just don't confuse these are just address to memory locations – you have to by
`default mug this up

```

Public Const SWP_NOACTIVATE = &H10
Public Const SWP_NOZORDER = &H4
Public Const SWP_SHOWWINDOW = &H40

```

```

Public Const HWND_TOP = 0

```

```

Public Const WS_CHILD = &H40000000
Public Const GWL_HWNDPARENT = (-8)
Public Const GWL_STYLE = (-16)

```

And also modify the sub main module a little. Just comment the app.previousinstance since, this is not required now otherwise it will create problems while running it from the screen saver display property. Since, it will make the screen saver child window to coincide with the screen saver main screens during their loading.

This is making the stuff more complicated for you to understand and for me to explain. Anyway just read out the stuff numerous times.

APIs	USE
GetClientRect	Get the dimensions of the preview area
GetWindowLong	Get the current window style
SetWindowLong	Set the window's new style – preferably WS_CHILD
SetParent	Set the window's parent so it appears Inside the preview area.
SetWindowPos	Show the preview

Also you have to get hwnd from command line using a function which we will make. This is passed since window doesn't pass /p alone it also passes /p with some number, which can be used to get the hwnd. Ok

```
' Get the hWnd for the preview window from the
' command line arguments.
Private Function GetHwndno(ByVal args As String) As Long
Dim argslen As Integer
Dim i As Integer
Dim ch As String

' Take the rightmost numeric characters.
args = Trim$(args)
argslen = Len(args)
For i = argslen To 1 Step -1
    ch = Mid$(args, i, 1)
    If ch < "0" Or ch > "9" Then Exit For
Next i

GetHwndFromCommand = CLng(Mid$(args, i + 1))
End Function
```

Also your sub main module should now look like below. Also don't forget to declare the api's and constants as discussed before.

```
Sub main()
Dim preview_hwnd As Long
Dim preview_rect As RECT
Dim window_style As Long
Dim args As String
```

```
'If App.PrevInstance Then End
```

```

If InStr(Command, "/s") > 0 Then
    MsgBox "showing screensaver", vbInformation
    Call CursorOff
    frmmain.Show

ElseIf InStr(Command, "/c") > 0 Then
    MsgBox "showing configuration", vbInformation
    'MsgBox "There is no setup", _
    'vbInformation, " Screen Saver"
    frmsettings.Show
'preview logic
ElseIf InStr(Command, "/p") > 0 Then
    'Get argument especially for /p with some hwnd numbers..ok – don't panic
    args = UCase$(Trim$(Command$))
    ' Get the preview area hWnd.
        preview_hwnd = GetHwndno(args)
        ' Get the dimensions of the preview area.
        GetClientRect preview_hwnd, preview_rect
        Load frmmain
        ' Get the current window style.
        window_style = GetWindowLong(frmmain.hwnd, GWL_STYLE)
        ' Add WS_CHILD to make this a child window.
        window_style = (window_style Or WS_CHILD)
        ' Set the window's new style.
        SetWindowLong frmmain.hwnd, _
            GWL_STYLE, window_style
        ' Set the window's parent so it appears
        ' inside the preview area.
        SetParent frmmain.hwnd, preview_hwnd
        ' Save the preview area's hWnd in
        ' the form's window structure.
        SetWindowLong frmmain.hwnd, _
            GWL_HWNDPARENT, preview_hwnd

    ' Show the preview.
        SetWindowPos frmmain.hwnd, _
            HWND_TOP, 0&, 0&, _
            preview_rect.Right, _
            preview_rect.Bottom, _
            SWP_NOZORDER Or SWP_NOACTIVATE Or _
            SWP_SHOWWINDOW
Else
    Call CursorOff
    frmmain.Show
End If
End sub

```

Now The project is over. You can make the .scr file and hence, test the screen saver. Also, the preview may not be centered. This is because we haven't implemented the project with scalewidth and scaleheight. This is because otherwise the project would be more complicated to explain.

Anyway have a nice screen saver. Just try to embed more complicated logic. Why not let's embed macromedia flash into our screen saver to show the best animations. For this purpose you need to know flash. Now the first part of you is that you must make a movie clip with .swf extension. Make a dazzling movie clip. Make the size according to your requirement. Now open visual basic and from components select shockwave flash component. Draw the component over the screen. Just adjust it's height and width as per your choice. Remember this component is generally there if you have some flash player / flash installed. The component may be present in the installation file of another other third party software.

Now, what you have to do is just do the following:

'just place the movie clip into the same folder where the screen saver file is kept and
'make the code accordingly.

'write the following in load event of form. That's it!

```
sf.Movie = App.Path & "\" & "dancefun.swf"
```

'here sf is the name of the shockwave flash control object

Thus!!... just play with animation. Hereby you can even insert a avi file anything which you can ever imagine. You can even embed a game or so.

Caution: Don't attempt to run the virus script using the screen saver. You can very well create a virus.

Anyway! Now! With the power of screen saver you can reach the world. Make your resume, do publicity for your product, software, company, photographs, social issue, wild life etc.

A bit about me:

I hope you like this article/tutorial. I really appreciate feedback or threats! Whatever? My name is Somdutt Ganguly. I am a programmer cum student right now doing my MCA – 2nd semester from S.V. Institute of computer studies, kadi, North Gujarat University, Gujarat, INDIA. Previously got my BCA degree (bac of computer application) from CPICA College, Gujarat University. Also, just trying to develop and sell some software's related stuff! Let's hope everything goes well. Anyway you all enjoy this article and try to develop some really good screen savers. Also, obvious acknowledgement to the NET resources from where I was able to study information.

Email: gangulysomdutt@yahoo.com

Address: No 6, chandrodaya apt, bhaikaka nagar, thaltej, ahmedabad, Gujarat, INDIA – 380059

Date: Feb - 2003

(You have full right to copy this stuff for your learning purpose – you can publish this – just make sure you let me know – this is a freeware)

