-------- **Steps to add Bitmap image resource in project (Splash Screen)** --------

1. Start "Visual Studio 201 7" IDE.
2. Select File> New > Project.
3. From "New Project" dialog box,
4. In left pane, select: Installed > Installed > Visual C++ > General.

In right pane, select: Empty Project.

1. In bottom area, in 'Name' text box, type desired project name, say "MyWindow”
2. In 'Location' text box, press 'Browse' button which will display 'Project Location' Dialog Box.

In it provide location for your Project Directory.

1. Keep "Create directory for solution" and "Add Source Control' check boxes unchecked.
2. Finally press "OK" button.
3. The IDE now shows “Solution Explorer” with our projects name at the top.
4. Copy a bitmap image of your choice to your project directory, say “MyImage.bmp"

(The Image should be high resolution 24-bit bitmap with “.bmp” extension)

1. Click on Project Menu > Add Existing Item > MyImage.bmp
2. Click on Project Menu > Add New Item,
3. In left pane, select: Installed > Installed > Visual C++ > code,
4. In right pane, select: Header File(.h)
5. In bottom area, in 'Name' text box, type desired header name, say "MyResource.h”
6. Click on Add Button. The header file should be visible in Solution Explorer.
7. In header file, declare the resource as:

**#define MY\_BITMAP 101**

(Do not forget to hit enter)

1. Click on Project Menu > Add New Item,
2. In left pane, select: Installed > Installed > Visual C++ > Resource.
3. In right pane, select: Resource File(.rc)
4. In bottom area, in 'Name' text box, type desired resource name, say "MyResource.rc”
5. Click on Add Button.
6. The IDE by default opens resource file in resource editor. Right click on .rc file > Open With > C++ Source Code Editor
7. IDE will ask to close already opened resource file. Click ‘Yes’ on the dialog Box.
8. Resource file will open in C++ source Code Editor, delete the existing contents of file if any

(If IDE by default creates a header file with resource file, delete the new header file from solution explorer as we have already added our own header file)

1. In resource file (MyResource.rc), include our header file:

**#include “MyResource.h”**

and declare resource statement for bitmap image:

**MY\_BITMAP BITMAP MyImage.bmp**

(Do not forget to hit enter)

1. Similarly, add main Source code file (.cpp) file to the project say “MyWindow.cpp”

**Assuming you have typed first window code in main source code file: -**

1. Include our header in .cpp file below windows.h:

**#include <windows.h>**

**#include “MyResource.h”**

1. Now for Displaying Bitmap image:
2. **Load** the bitmap first in **WM\_CREATE** message, using **LoadBitmap()** function**.**

(Because we load bitmap only once, WM\_CREATE will be appropriate place to load our image)

1. For drawing we use **WM\_PAINT** message.
2. As usual get device context **HDC** (specialist) using **BeginPaint()** function for our window.
3. For bitmap image, we have to create the “Compatible Device Context”, which is used for bitmap painting.

Therefore, we have to declare second **HDC** variable and get Compatible DC in it using **CreateCompatibleDC()** function.

1. Now select the new compatible DC for drawing using **SelectObject()** function.
2. Get the dimensions of our bitmap image using **GetObject()** function.
3. Copy the image data from our memory DC (compatible DC) to window DC [returned by BeginPaint()] using one of the following functions:
4. **BitBlt()**
5. **StretchBlt()**
6. **PatBlt()**

(Explore all parameters of this functions using MSDN as according to window size and image dimensions, parameters will vary)

1. Now for clean-up (As bitmap image is displayed on screen), we restore the device context to the original state.

So, select previously created Compatible DC using **SelectObject()** function and use **DeleteDC()** function to destroy it from memory.

1. Then end the paint using **EndPaint()** function.
2. Finally, before termination of our program, we first “delete the object/ free the bitmap” using **DeleteObject()** in **WM\_DESTROY** message.
3. Your bitmap image should be visible on your window when you execute the program using ALL above function correctly.
4. Finish