The Legend Of Random

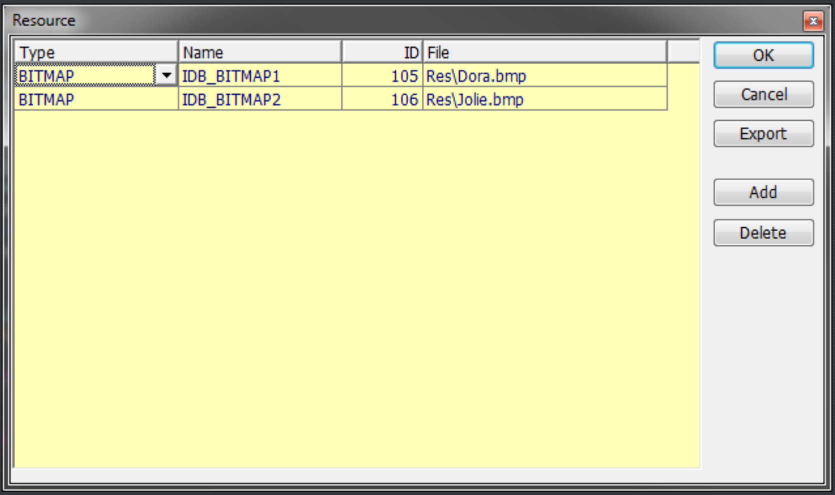
Programming and Reverse Engineering

R4ndom’s Guide to RadASM: Adding an Icon and Menu

by R4ndom on Sep.27, 2012, under Beginner, Tools, Tutorials

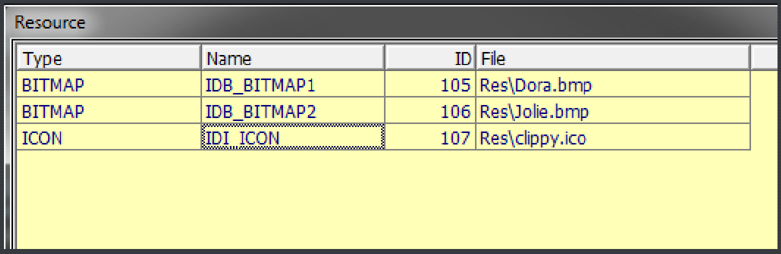
Adding an Icon

Let’s start by adding an icon. Load up our HelloWorld project from the last tutorial. Click on Project->Resources. This will bring up the Resources window:



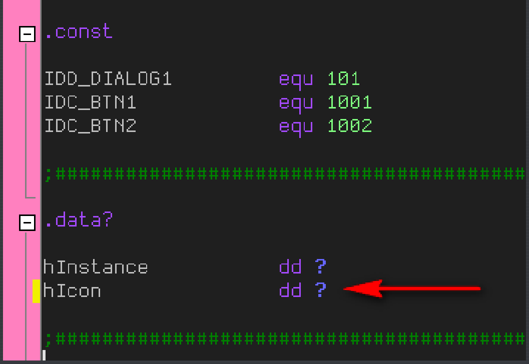
Type	Name	ID	File
BITMAP	IDB_BITMAP1	105	Res\Dora.bmp
BITMAP	IDB_BITMAP2	106	Res\Jolie.bmp

Click the Add button and enter in the following info for our icon. Make sure before you do this you copy the icon (supplied in the download) into the Res folder in your HelloWorld project directory. The name of the icon I included is called “clippy.ico”. If you wish to include your own, be my guest, just make sure it’s an .ico file:



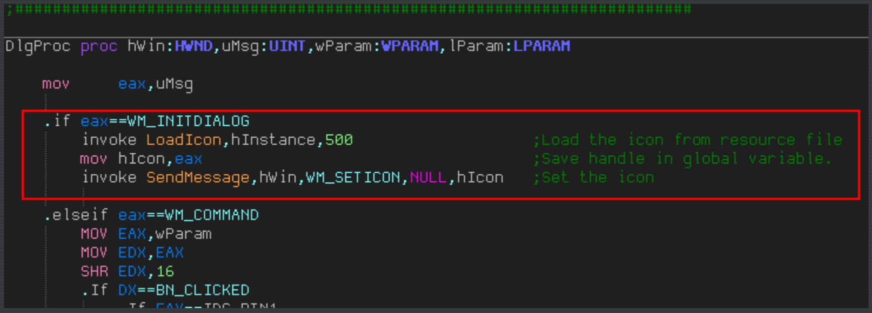
Type	Name	ID	File
BITMAP	IDB_BITMAP1	105	Res\Dora.bmp
BITMAP	IDB_BITMAP2	106	Res\Jolie.bmp
ICON	IDI_ICON	107	Res\clippy.ico

Now we’re going to register the icon with our program. The first thing we need is a global variable for the handle to the icon. Open up HelloWorld.inc and add this line:



```
.const

IDD_DIALOG1          equ 101
IDC_BTN1             equ 1001
IDC_BTN2             equ 1002

;#########################################

.data?

hInstance            dd ?
hIcon                dd ?

;#########################################
```

Now we need to actually load the icon to assign it to our program. We will need to add the appropriate code to the WM_INITDIALOG section:

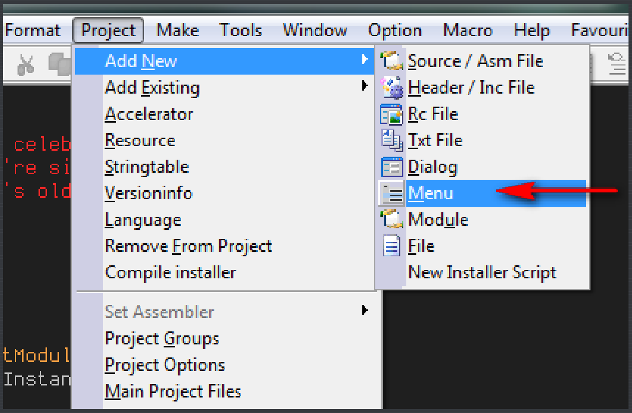


```
;#########################################################################

DlgProc proc hWin:HWND,uMsg:UINT,wParam:WPARAM,lParam:LPARAM

    mov      eax,uMsg

    .if eax==WM_INITDIALOG
        invoke LoadIcon,hInstance,500                   ;Load the icon from resource file
        mov hIcon,eax                                   ;Save handle in global variable.
        invoke SendMessage,hWin,WM_SETICON,NULL,hIcon   ;Set the icon

    .elseif eax==WM_COMMAND
        MOV EAX,wParam
        MOV EDX,EAX
        SHR EDX,16
        .If DX==BN_CLICKED
            .If EAX==IDC_BTN1
```

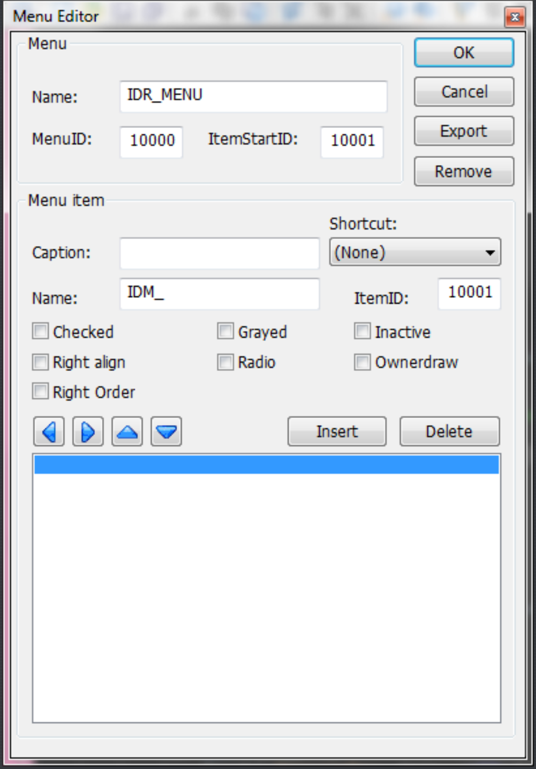
Now, after you build the project, you will see that the .exe file that RadASM has created has our new icon:

Adding a Menu

RadASM actually makes it quite easy to add a menu, as it has a menu resource editor built-in:



Clicking “Add new”->”Menu” under the Project menu will display the menu resource editor:



Name: IDR_MENU

MenuID: 10000 ItemStartID: 10001

Caption:

Shortcut: (None)

Name: IDM_

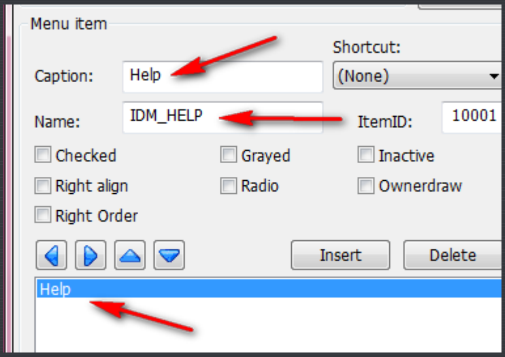
ItemID: 10001

☐ Checked ☐ Grayed ☐ Inactive

☐ Right align ☐ Radio ☐ Ownerdraw

☐ Right Order

This editor let's us create single button menus, drop down menus and multi-drop down menus (menus within menus). We'll create a simple menu bar with two options; "help" and "quit". These will not have drop-down capability. First, enter the information for the first:



Caption: Help

Shortcut: (None)

Name: IDM_HELP

ItemID: 10001

☐ Checked ☐ Grayed ☐ Inactive

☐ Right align ☐ Radio ☐ Ownerdraw

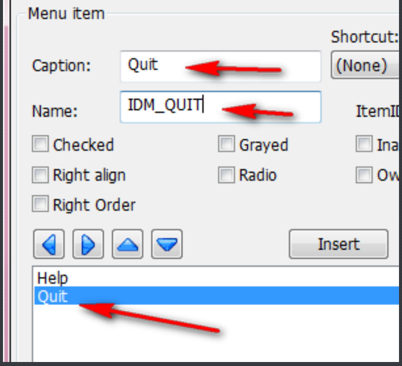
☐ Right Order

Help

Here, we enter the caption (the part that will be displayed) and the name. As soon as you enter these the item will display in the bottom box. At this point you can delete the item by clicking "Delete", or you can edit it by just highlighting it in the box.

You also have the options to set the attributes of the menu item (for example, greyed out) and changing the item ID. In our case, we'll leave it at 10001. You can also set up a hot-key for the menu item by clicking the "Shortcut" spinner. Like all other Windows IDEs, you can enter an ampersand (&) before any letter to make it an accelerator (for example, E&xit makes it so you just have to press the 'x' key to trigger the menu item.)

Let's add our second. Click the empty line below "Help" in the bottom box which gives us a new item to modify:


Menu item

Caption: Quit

Name: IDM_QUIT

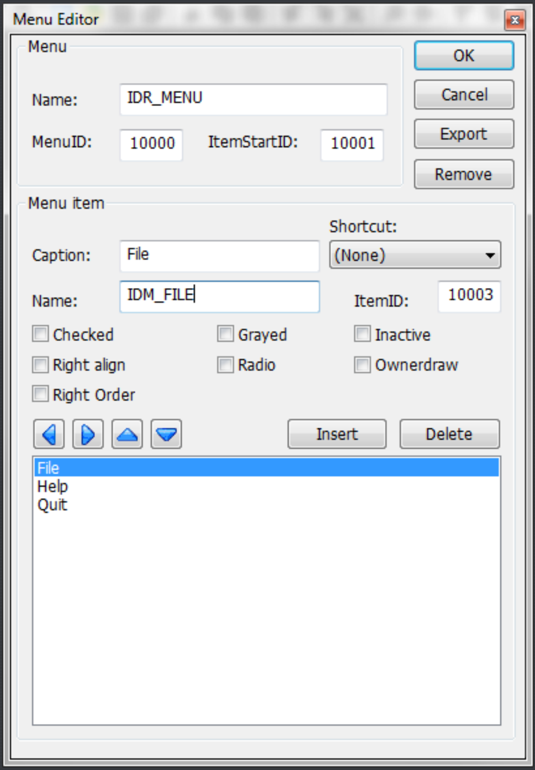
Shortcut: (None)

□ Checked □ Grayed □ Ina
□ Right align □ Radio □ Ow
□ Right Order

Help
Quit

At this point, if you wanted to insert an item between the two, you would simply highlight the first and click “Insert”. This will insert an empty item between the two.

Another thing you may have noticed are the four blue arrow icons. The up and down icons allow you to change the order of the menu items. The left and right one determine if the menu item is a header or a sub item. For example, let’s add a “File” menu, and when you click this, an additional menu drops from it with the option to “Save”. In this case, we would first create the File name (at the top by highlighting Help” and clicking the “Insert” button):


Menu Editor

Menu

Name: IDR_MENU

MenuID: 10000 ItemStartID: 10001

Menu item

Caption: File

Name: IDM_FILE

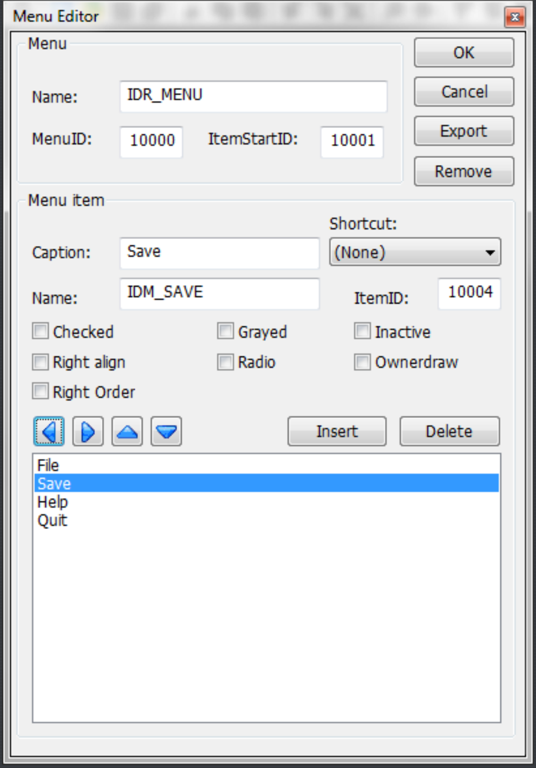
Shortcut: (None)

ItemID: 10003

□ Checked □ Grayed □ Inactive
□ Right align □ Radio □ Ownerdraw
□ Right Order

File
Help
Quit

We would then create the save item going in this menu by clicking the “Help” item and clicking “Insert”, then inserting the data for save:



Menu

Name: IDR_MENU

MenuID: 10000 ItemStartID: 10001

Menu item

Caption: Save Shortcut: (None)

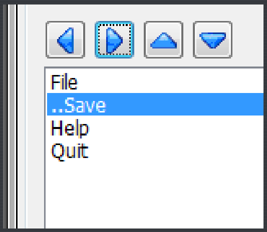
Name: IDM_SAVE ItemID: 10004

□ Checked □ Grayed □ Inactive

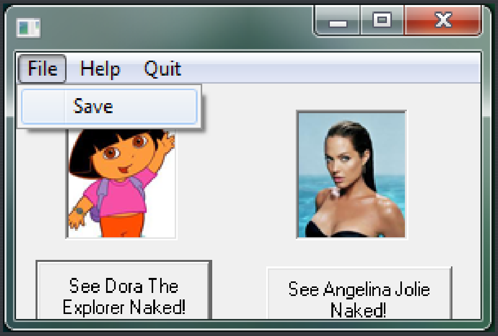
□ Right align □ Radio □ Ownerdraw

□ Right Order

Now, higlighting the “save” item and clicking the right arrow key makes this item a sub-menu item of the one on top of it, in this case the :File” header:

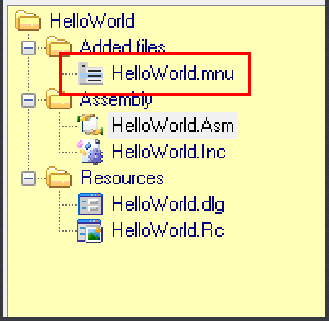


Now when we run the app, we see a new menu header with a Save option under it:

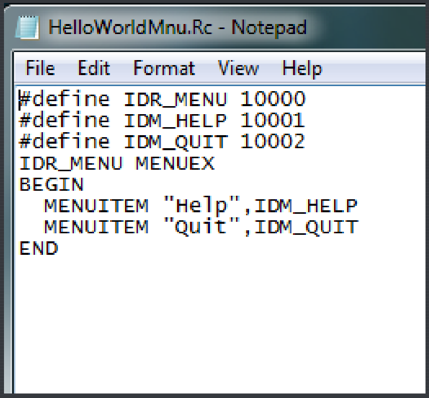


Note: For the sake of this tutorial, I deleted the “File” and “Save” menu items.

You will also notice, after clicking OK, RadASM has created our new menu in a .mnu file:

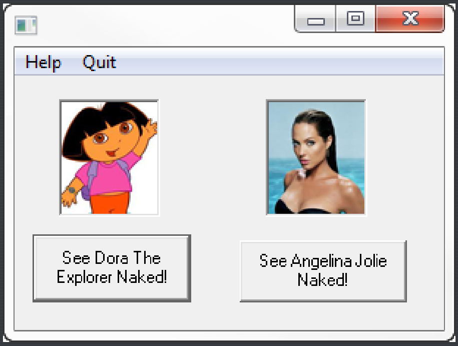


Any time you open this file, the menu editor will appear. If you open this file in Notepad, you can see what’s in it:

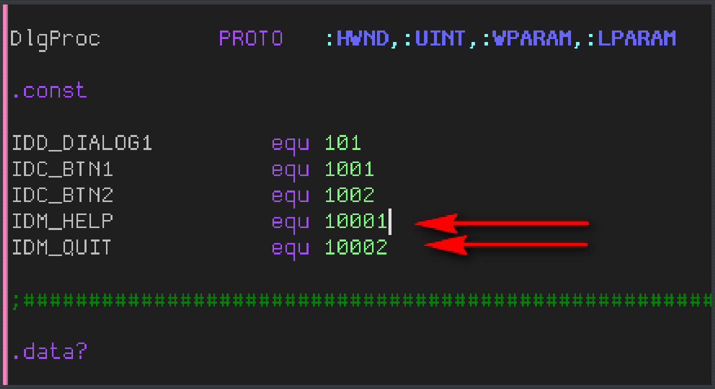


```
#define IDR_MENU 10000
#define IDM_HELP 10001
#define IDM_QUIT 10002
IDR_MENU MENUEX
BEGIN
  MENUITEM "Help",IDM_HELP
  MENUITEM "Quit",IDM_QUIT
END
```

Now, when we run our program, we see our menu items automatically appear, Of course, they don’t do anything yet:

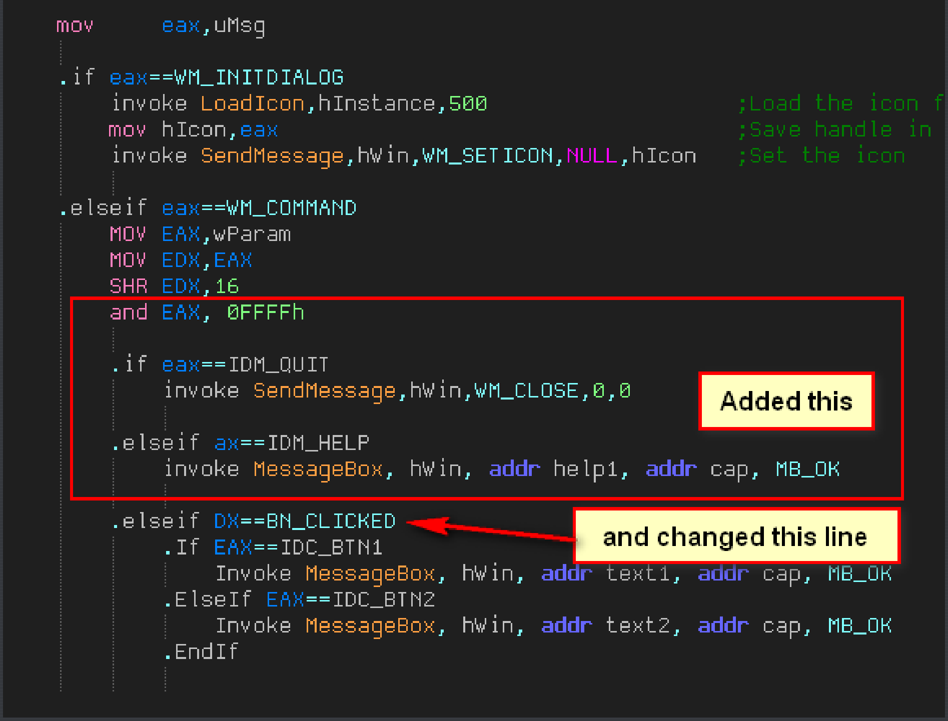


We need to add the code to handle he menu’s messages. First, the two lines shown need to be added to the HelloWorld.inc file:



```
DlgProc         PROTO   :HWND,:UINT,:WPARAM,:LPARAM

.const

IDD_DIALOG1         equ 101
IDC_BTN1            equ 1001
IDC_BTN2            equ 1002
IDM_HELP            equ 10001
IDM_QUIT            equ 10002

;#####################################################

.data?
```

Then we need to add the message handlers into the HelloWorld.asm file:



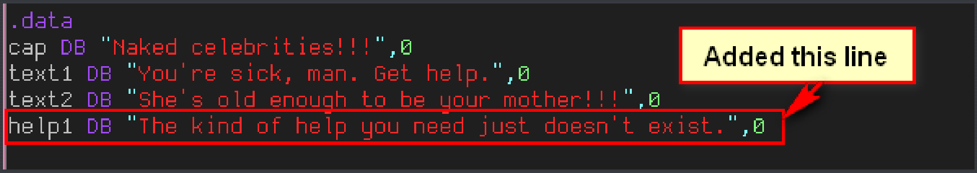
```
    mov     eax,uMsg

    .if eax==WM_INITDIALOG
        invoke LoadIcon,hInstance,500                       ;Load the icon f
        mov hIcon,eax                                       ;Save handle in
        invoke SendMessage,hWin,WM_SETICON,NULL,hIcon       ;Set the icon

    .elseif eax==WM_COMMAND
        MOV EAX,wParam
        MOV EDX,EAX
        SHR EDX,16
        and EAX, 0FFFFh

        .if eax==IDM_QUIT
            invoke SendMessage,hWin,WM_CLOSE,0,0

        .elseif ax==IDM_HELP
            invoke MessageBox, hWin, addr help1, addr cap, MB_OK

        .elseif DX==BN_CLICKED
            .If EAX==IDC_BTN1
                Invoke MessageBox, hWin, addr text1, addr cap, MB_OK
            .ElseIf EAX==IDC_BTN2
                Invoke MessageBox, hWin, addr text2, addr cap, MB_OK
            .EndIf
```

Added this

and changed this line

Notice that, because the menu handlers had to go first, the ".if DX==BN_CLICKED" line was changed to ".elseif DX==BN_CLICKED".

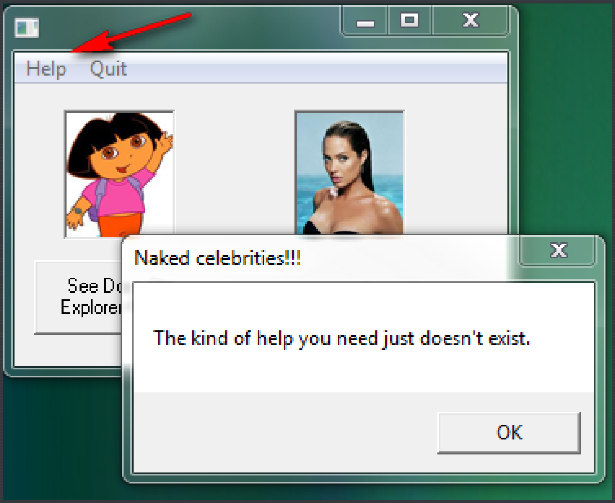
Finally, we need to add a new string so that our menu does something when you click "About":



```
.data
cap DB "Naked celebrities!!!",0
text1 DB "You're sick, man. Get help.",0
text2 DB "She's old enough to be your mother!!!",0
help1 DB "The kind of help you need just doesn't exist.",0
```

Added this line

Now when we run our program, we can select our new menu items:



Naked celebrities!!!

The kind of help you need just doesn't exist.