

R4ndom's Guide to RadASM: Installing and Set Up

by R4ndom on Sep.21, 2012, under Beginner, Reverse Engineering, Tutorials

RadASM is a very good IDE (Integrated Development Environment: think Visual Studio) for developing in most programming languages, though assembly language is where it shines. Not only is it one of the few IDEs out there that will work with assembly language, but the author has also put a significant amount of time and effort into it, making it extremely feature rich and robust. Many people use inline assembly in Visual Studio (using `__asm` directives) but you really can't get the level of control in these big-name IDEs as you can in this product. I used WinASM, a very good product, for years, but when I finally tried RadASM, it felt like someone had created it that really knew assembly language, and the hardships that can bring.

In this tutorial I will go over installing RadASM, running it, and creating programs in it, as well as some additional plugins and features. In the download for this tutorial (available on the [tutorials](#) page) I have included all of the installation files for RadASM, as well as the Masm SDK, providing everything you need to program in assembly language. So let's get started.

Please keep in mind that this is not an assembly language nor a Windows programming tutorial. It is assumed that both of these have already been learned. If you are rusty (or a beginner) I highly recommend the Windows assembly tutorials by lczelion. Fortunately, they are also based on RadASM, so the transition should be very easy.

<http://www.woodmann.com/RCE-CD-SITES/lczelion/index.html>

Installation

The first thing we need to do is install Masm onto our computer. Masm is the actual compiler (the same one Visual Studio uses) and has been around a very long time. There is an `install.exe` file in the `Masm32v11r.zip` file included in this download. Unzip it and run it- it will automatically install Masm onto your computer. I suggest, in order to eliminate problems in the future, you simply install it into it's own folder in your root folder on your drive, for example "C:/MASM".

Now we need to install RadASM itself.

The first file we need is `RadASM.zip`. This is the main IDE files. Copy these into a new folder. This will now be your root install directory, so you may want to copy this folder in to your Program Files folder, make a shortcut etc.) Next we need to add support for Masm (or TASM, GoASM or nASM). As RadASM is a generic IDE, and works with pretty much any language you can throw at it, we need to give RadASM the specific details about the assembly language we'll be using.

Open the `Assembly.zip` file and copy the appropriate folder to the root folder of the RadASM installation. I copied the Masm folder, as that's the type of assembly I work in (and most others do as well). If you wish to work in one of the other languages, you can copy those as well. We also need to copy the associated `.ini` file in the root folder as well, in my case `Masm.ini`. Finally, copy the appropriate `.ct` and `.kwl` files into the plugins folder in the root of the RadASM folder.

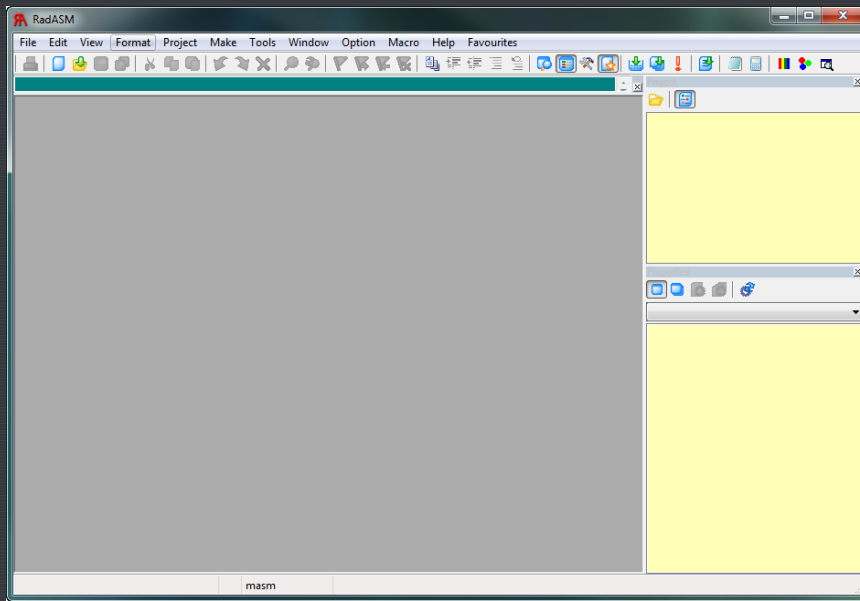
Next, copy the `RadASM.chm` help file into the Help folder in the RadASM root installation folder. Now, when you click "Help"->"RadASM" in the IDE, this help file will be displayed.

"RadLanguages.zip" contains additional languages (besides English) if you wish to have the IDE in a different language. It defaults to English.

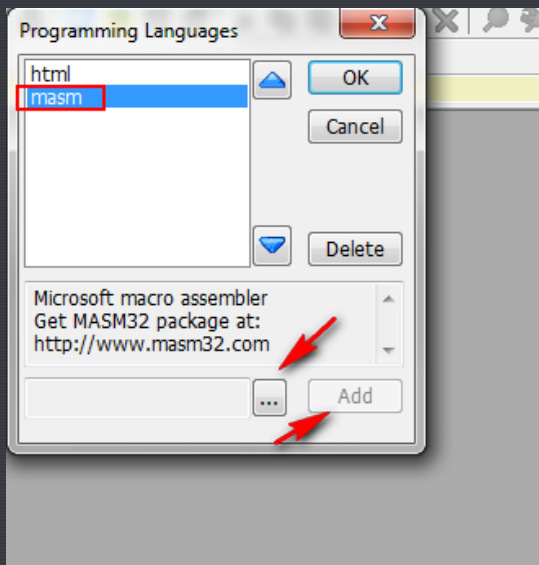
Finally, as you are probably just getting started with assembly language, you can copy the "Demos.zip" and "Games.zip" contents into a Demo folder in the root installation folder. This will give you plenty of examples to look at.

Setting Up the Environment

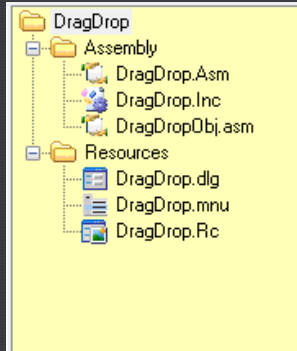
Now run RadASM and you will see the basic layout:



First things first, select Options->Programming Languages. Click the "..." button, select the Masm.ini file and hit the "add" button. Masm should now appear in the list of languages. If you have a different assembly type (or want to add additional types) like Tasm or GoASM, repeat the process for each one. Then click OK. You now have support for that particular language:



Now, you may want to open up any project just to see the color scheme. Here, I loaded the "DragNDrop" project from RadASM root/Masm/Projects/DragDrop. The project file ends with a .rap. You will see the project in our Project Explorer on the right:



Now double-click on the DragDrop.asm file to open it in our code window:

```
InitCommonControls

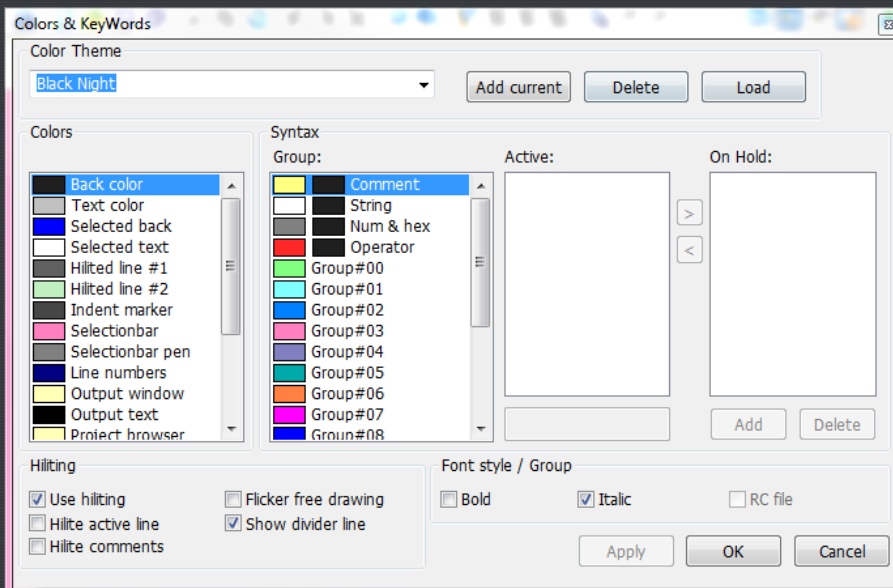
mov     hInstance, eax
invoke  GetCommandLine
invoke  InitCommonControls
invoke  WinMain, hInstance, NULL, CommandLine, SW_SHOWDEFAULT
invoke  ExitProcess, eax

WinMain proc hInst:HINSTANCE, hPrevInst:HINSTANCE, CmdLine:LPSTR, CmdShow:DWORD
LOCAL   wc:WNDCLASSEX
LOCAL   msg:MSG

mov     wc.cbSize, SIZEOF WNDCLASSEX
mov     wc.style, CS_HREDRAW or CS_VREDRAW
mov     wc.lpfnWndProc, OFFSET WndProc
mov     wc.cbClsExtra, NULL
mov     wc.cbWndExtra, DLGWINDOWEXTRA
push    hInst
pop     wc.hInstance
mov     wc.hbrBackground, COLOR_BTNFACE+1
mov     wc.lpszMenuName, 10000
mov     wc.lpszClassName, OFFSET ClassName
invoke  LoadIcon, NULL, IDI_APPLICATION
mov     wc.hIcon, eax
mov     wc.hIconSm, eax
invoke  LoadCursor, NULL, IDC_ARROW
mov     wc.hCursor, eax
invoke  RegisterClassEx, addr wc

invoke  CreateDialogParam, hInstance, 1000, NULL, addr WndProc, NULL
invoke  ShowWindow, hWnd, SW_SHOWNORMAL
invoke  UpdateWindow, hWnd
while TRUE
```

as you can see, the default color scheme leaves a little to be desired. I personally can't even see the dark blue text, though I am getting old, so what can I say. Let's change the color scheme to something a little softer on the eyes. Select "Option"->"Color & Keywords". This brings up the main color selector:



I don't know about you, but I like to personalize my coding environment. My window allows you to do just that. The method to change colors starts with a theme. There are a couple that come with RadASM and if you look really hard, you can find some additional ones online. But since EVERY option (and I mean EVERY) is changeable, you really won't need one. I started with the default theme, Dark Night, and modified it from there. If you prefer a lighter background, try one of the other themes. Just select it and click "Load", then "Apply" and you will see it pop up. Start with whatever theme is closest to what you like, then you can fine-tune it from there.

To fine-tune it, we go to the Colors and Syntax box below. Here, we choose any type of instruction and select a color for it. You can either select a default color under the Colors group, or you can double-click the type in the Syntax group and select any color you want. For example, if I write a quick comment in the code:

```
;This is a comment
```

and decide I would instead like it green, I would open the color picker, double-click "Comment" under the Syntax group, and pick a nice green. Clicking Apply will then show me the results:

```
;This is a comment
```

Because there are so many options that can have their colors changed, a lot of times I will simply look for a color I don't like in the Syntax group and then change it. Eventually, you will learn what each group type is, or through trial and error will figure it out. After selecting the color I prefer, I now have my own color scheme:

```
.386
.model flat,stdcall
option casemap:none

include DragDrop.inc
include DragDropObj.asm

.code

start:

    invoke GetModuleHandle,NULL
    mov     hInstance,eax
    invoke GetCommandLine
    invoke InitCommonControls
    invoke WinMain,hInstance,NULL,CommandLine,SW_SHOWDEFAULT
    invoke ExitProcess,eax

WinMain proc hInst:HINSTANCE,hPrevInst:HINSTANCE,CmdLine:LPSTR,CmdShow:DWORD
LOCAL wc:WNDCLASSEX
LOCAL msg:MSG

    mov     wc.cbSize,SIZEOF WNDCLASSEX
    mov     wc.style,CS_HREDRAW or CS_VREDRAW
    mov     wc.lpfnWndProc,OFFSET WndProc
    mov     wc.cbClsExtra,NULL
    mov     wc.cbWndExtra,DLGWINDOWEXTRA
    push    hInst
    pop     wc.hInstance
    mov     wc.hbrBackground,COLOR_BTNFACE+1
    mov     wc.lpszMenuName,10000
```

If one of you reading this tutorial would like my exact colors (or want to trade with someone else), you can open the RadASM.ini file provided in this download and copy everything under the "[Colors]" section into the RadASM.ini file in your root folder, replacing what you have with what you copied. Your colors will then match mine exactly.

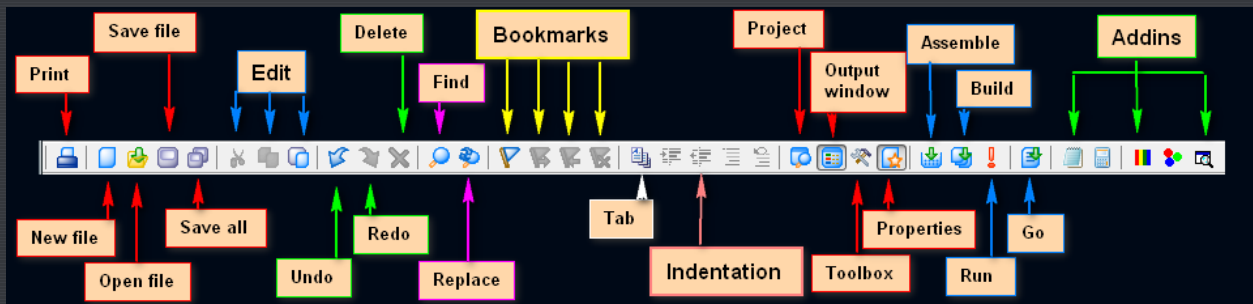
The last thing I like to do is change the default font. I find the default one a little blocky, so I have included a font designed for assembly language users in mind (I don't know if this is true, but it sure is a lot better than the default.) It is called Dina, and is in the download with this tutorial. Just unzip the contents and copy them in to the Windows font folder, re-run RadASM, and select "Option"->"Font options". Now, select the installed Dina font for the first three:



Some other configuration in the Option menu you may want to look at are the following:

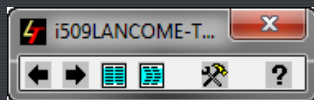
- Languages: Changes the IDE language (if you prefer German, for example)
- Programming Languages : Adds new templates for additional programming languages.
- Code Editor Options: Change tab stops, backup policies etc.
- Dialog Editor Options: For changing the look of dialogs in the Dialog Editor.
- Menu Accelerators: For adding your own hot-keys.
- Addin Manager: For adding and deleting addins.

The Toolbar



Depending on which addins you have installed, the addins icons at the right-hand side may be different, though Notepad and Windows Calc will always be the first two icons.

One interesting feature that should be pointed out is the "Tabs" button. This opens the Tabs toolbar:



These options allow you to auto-indent code, normalize code (make it the same indentation), as well as set tab lengths and other tab related options.