

Forum:[Hack Forums](#)

Thread: [\[TUT\] What tha Fuck is An API??\[VB.NET\]](#)

GoldStar - May 18, 2010, 12:26 pm

Intro:

for more advanced coding, we do some real close coding to the OS. that means we create some functions to work with, but the system has some similar method to work with it self. now Microsoft gave us a way to make our job easier and simpler, by offering us to use the functions and procedures that windows it self uses. all those methods are stored in files called libraries, with the extension .DLL. those are called APIs, Literaly Application Programming Interface. so far sounds cool right?

Declaring:

in VB.Net, there is a defined method of declaring APIs. here how it works:

Declare Function FuncName Lib "Library" Alias "Alias" ([arglist]) As Type

now lets explain it:

- the word Declare means we are calling a method from the outside.
- the word Function defines the type of the method we are calling. in some cases it can be a SUB instead.
- the word FuncName is what we chose for name of the function.
- now Lib "Library" means that we will specify from where we are gonna call the method. e.g : Lib "User32.Dll". here our function is stored in the User32.dll file located in the system directory.
- Alias "Alias" will specify the name of the function inside that dll. it is well defined, not something we can choose.
- the ([arglist]) is the arguments that we will pass to the method,same as regular methods. e.g: (Byval i As Integer, Byval j As String). those parameters are well defined for each api.
- and as any function, its type needs to be specified. As type will define the type of the returned Value.

Extras: this is not it! Options are more than that!

- Before Declare, it is optionnal to add visiblity properties, such as Private,Public and Friend.
- After Declare, it is optionnal to add the encoding type of the method, such as Ansi, Unicode and Auto. the default in vb.net is unicode.
- it can Be a Function or a Sub. but remember that a Sub does not return a value, so we remove the "As Type" when using SUBs.
- we can declare an API without an "Alias", but in this case, the Name of the method (FuncName) must be the same as the alias, means the same name of the function in it's dll. and it's logic, cuz the computer gonna look for the dll, and search the function with the alias inside. when an alias is not specified, he will use the Name of the method to search. if we choose w.e, it will not be found.
- note that some APIs have no Arguments.

here is an exemple of an api:

```
Private Declare Ansi Function GetPrivateProfileStringA Lib "kernel32.dll" (ByVal lpApplicationName As String, ByVal lpKeyName As String, ByVal lpDefault As String, ByVal lpReturnedString As String, ByVal nSize As Integer, ByVal lpFileName As String) As Integer
```

New Declaration !!

Now this looks a lot like vb6 declarations, but some options are added.
in VB.NET, we got the possibility to declare those in a new way:

After Adding This Import :

Code Imports System.Runtime.InteropServices

We can declare those APIs this way:

```
<DllImport("Library", Parameters)> Function FuncName ([arglist]) As Type  
End Function
```

Explaining Again: (tired of typing lol):

-DllImport : this word means we are calling a method from the outside. if you forget the System.Runtime.InteropServices import, this will give you an error, as DllImport is not defined.

-"Library" : with the same meaning, the Dll File name. e.g : "User32.dll"

-Parameters : this part is a set of parameters for the api. in the earlier declaration, we can add ansi for example to set the encoding. now in the new declaration, all parameters go here. they are in total 8 parameters, all optionals. here is an e.g:

```
<DllImport("user32.dll", CharSet:=CharSet.Unicode, ExactSpelling:=True)>
```

-after that, we have the word Function, that will define the type of method (Function/Sub).

-([arglist]): almost the same as the other declaration, just arguments to pass to your method. the only difference, is that we can do some more specification about one or more variables, with the key word MarshalAs, to set some properties to some variable, when it is unmanaged. here is an example:

```
(ByVal I As Integer,<MarshalAs(UnmanagedType.LPArray,SizeParamIndex:=6)>ByVal R() As Byte)
```

MarshalAs Offers 10 Parameters To set, 1 necessary and 9 optionals.

-we come to the As Type, which is the same, removed when declaring a Sub.

-our Declaration ends with The Statement 'End Function' Or 'End Sub' according to what you declared.

Extras!! yes!!

-this declaration supports also being Private,Public,Friend,Shared...ect...

-well nothing to add much, i wanted to talk more about the parameters you can set and options, also about MarshalAs and unmanaged types, but options are a lot, and i don't want to spend the night typing lol, so just do some more research about those, you will learn a lot.

here is an example of an api declaration:

```
<DllImport("user32.dll", SetLastError:=True, CharSet:=CharSet.Auto)>
```

```
Private Shared Function FindWindow(  
    ByVal lpClassName As String, _
```

```
    ByVal lpWindowName As String) As IntPtr
```

```
End Function
```

I'm About Done, Conclusion

Windows API is Just a big Chapter, this is the very basic of it.

now, for the APIs, the way to find them is MSDN of microsoft. you can find all informations you need about the function of a method, its arguments and parameters, and more.

you can also use google, another way to find APIs.usually when you search the name of an api with google, the MSDN page is first result anyway. Also there are a lot of sites made specially to store API declarations for a lot of programming languages.

one more thing, after declaring an api, you can use it like you wrote it in your class, and call it when ever you like. good thing about api's, they can replace big methods with one line declaration, which

saves space in your app.so have fun searching what apis you need to make your code smaller, and easier to use.

Well i think this is it. tut fully written by Miharbi-Dono.
don't be shy to reply.

Deathader - May 18, 2010, 12:26 pm

Nice title, and nice tutorial :)

DeltaNox - May 18, 2010, 12:27 pm

Great tutorial, very useful and well explained.

Vorfin - May 18, 2010, 12:28 pm

YAY ? High quality posts in this section. <3

TehShrink - May 18, 2010, 12:28 pm

Thanks for the another great tutorial, you must'a spent much time with this.

Montage - May 18, 2010, 12:32 pm

(May 18, 2010 12:26 PM)Miharbi-DoNo Wrote: [\[CITE\]](#) Private Declare Ansi Function
GetPrivateProfileStringA Lib "kernel32.dll" (ByVal lpApplicationName As String, ByVal lpKeyName
As String, ByVal lpDefault As String, ByVal lpReturnedString As String, ByVal nSize As Integer,
ByVal lpFileName As String) As Integer

Ouch... reminds me why I hate using VB .NET when using a lot of external DLLs!

VB .NET's sub/function declarations are a pisstake... ?

Tbh, I usually end up using C++/CLI in situations like this. Then you get .NET, and you simply include windows.h (or whatever), and it's all done - just call the functions with no function/consts/structure declarations.

Anyway, great tutorial. Would like to see more low-level OS usage in .NET around ? It's very readable to n00bs...

StRaXxD - May 18, 2010, 12:36 pm

In first i didnt see [TUT] i see just "What the fuck is an API?" and posted by: "Miharbi-DoNo"...I was thinking: " He asking that lol"

Now i see its great TUT and well expaling....Great job !

GoldStar - May 18, 2010, 12:53 pm

thx guys appreciate it.

(May 18, 2010 12:36 PM)StRaXxD Wrote: [\[CITE\]](#) In first i didnt see [TUT] i see just "What the fuck is an API?" and posted by: "Miharbi-DoNo"...I was thinking: " He asking that lol"

Now i see its great TUT and well expaling....Great job !
loooooool thx ^^

(May 18, 2010 12:32 PM)Montage Wrote: [\[CITE\]](#) Tbh, I usually end up using C++/CLI in situations like this. Then you get .NET, and you simply include windows.h (or whatever), and it's all done - just call the functions with no function/consts/structure declarations.

.

meeh... it's true, unlike alot others (c++/delphi...), .Net doesn't have a class for already defined apis, but that's because it has the most popular apis in .Net Classes anyway, that will call the apis. for example, the method GetValue from a RegistryKey class to read a key from registry, will call the api RegQueryValueEx, which is declared in the Class Win32Native. so .Net has an advantage in this one, it's not only using apis, also getting the direct reformatted result (directly as string in your variable with one line) without even the need to declare the api or to include anything.

adware - May 18, 2010, 1:15 pm

Thanks for this Miharbi.
Pretty good explained :)

Bitwin - May 18, 2010, 1:16 pm

Great tutorial, well written.

Zer[o] - May 18, 2010, 1:22 pm

great tutorial, thanks for the share. Should double check grammar though

rawi16 - May 18, 2010, 1:22 pm

WHAT a great TUT , thanks miharbi ! u are the best.

GoldStar - May 18, 2010, 1:26 pm

(May 18, 2010 01:22 PM)Zer[o] Wrote: [\[CITE\]](#) great tutorial, thanks for the share. Should double check grammar though
nah. probably not. English is my 3rd Language. so i think it's good enough :p

Cunt - May 18, 2010, 1:31 pm

Nice tut mate :) Very good like usall :) well done :bow:

The_KGB - May 18, 2010, 1:38 pm

nice tutorial, very informative. Hope to see more and more tutorials like this in this section.

Deathader - May 18, 2010, 1:40 pm

(May 18, 2010 01:26 PM)Miharbi-DoNo Wrote: [\[CITE\]](#) nah. probably not. English is my 3rd Language. so i think it's good enough :p
I've been wondering how many you knew. What is your primary language?

GoldStar - May 18, 2010, 1:48 pm

(May 18, 2010 01:40 PM)Deathader Wrote: [\[CITE\]](#) I've been wondering how many you knew. What is your primary language?Arabic^^

rapidguide - May 19, 2010, 7:12 am

Very Very Good Tututorial SUPER

[EPIC] LOL [/EPIC] - May 19, 2010, 7:16 am

The '??' isn't needed in the title.
Thanks for the tut tho :D!

Kotei - May 19, 2010, 8:16 am

Very nice Miharbi. I like how in detail it is.

GoldStar - August 13, 2010, 8:39 pm

bumping a tut i find good.

wyterzzz - August 14, 2010, 12:44 pm

Very good explained tutorial, thanks for it.

pr3fix - August 14, 2010, 1:00 pm

Great tutorial. You forgot to mention meaning of API, it means Application Programming Interface.

Btw rating added!

GoldStar - August 14, 2010, 1:17 pm

(Aug 14, 2010 01:00 PM)pr3fix Wrote: [CITE](#) Great tutorial. You forgot to mention meaning of API, it means Application Programming Interface.

Btw rating added!

no i didn't, it's in the first paragraph. and thx.

H4?0R - August 14, 2010, 2:32 pm

That's pretty cool tut bro. thanks for sharing

nickcm4891 - August 31, 2010, 8:13 pm

I have been looking for someone to explain this!!! I was asking other people how they know what to use and all I kept hearing was "Go away noob!!!" I've known how to program in basic since I was 7, and used vb6 and .net all through high school. I'm not a noob, I've just never had to use a windows api for anything until I was trying to understand how a keylogger worked and trying to code one. A million thanks to you and others who help out like this!!

xartermate - December 21, 2010, 5:18 am

awsome share dude!!!

good tut to get in to api's, now i can start understand this.

i think this is very important and good for those who want a nice fud runpe.

Ghostenigma - December 21, 2010, 6:49 am

wow nice tut and very helpfull. I have to read 5 times to understand it well:)

bob42 - September 15, 2011, 12:14 am

Thank you for sharing, everyone should bookmark this thread. ^_-

wertyl11 - September 15, 2011, 12:33 am

Yay! A great tutorial! :D

It's good to see HQ threads.

Penthos - September 15, 2011, 1:24 am

Very HQ tut,

I have actually saved it for future reference :)

Thanks for taking the time out for the community and us noobs!

+1 coming your way!