# 09.Explaining the Visual Basic concept, introduction to SmartCheck and configuration

2012 년 1 월 29 일 일요일
오전 11:51

Hello everybody.
모두들 안녕.
Welcome to this Part 9 in my series about reversing for newbies/beginners.
나의 초보자 reversing series Part 9 에 온 것을 환영해.
This "saga" is intended for complete starters in reversing, also for those without any programming experience at all.
이 "saga"는 완벽히 reversing 초보자를 맞춰서 만들어졌다. 또한 어떠한 programming 경험이 없어도 된다.
Lena151 (2006)

Set your screen resolution to 1152*864 and press F11 to see the movie full screen !!!
Again, I have made this movie interactive.
You screen 해상도를 1152*864 로 설정해 그리고 full screen 으로  movie 를 보기 위해 F11 를 눌러
So, if you are a fast reader and you want to continue to the next screen, just click here on this invisible hotspot. You don't see it, but it IS there on text screens.
그래서, 네가 이것을 빨리 읽고 다음 screen 을 보고 싶다면, 보이는 hotspot 여기를 눌러. 보고 싶지 않을 때는 여기에 두지마.
Then the movie will skip the text and continue with the next screen.
Movie 는 text 와 다음 screen 을 skip 할 수 있다.
If something is not clear or goes too fast, you can always use the control buttons and the slider below on this screen.
무언가 명확하지 않거나 빨리 넘기고자 할 때, 항상 control button 과 이 screen 밑에 있는 slider 바를 사용해.
He, try it out and click on the hotspot to skip this text and to go to the next screen now!!!
도전해봐. 그리고 이 text 와 다음 screen 을 보기 위해 hotspot 을 click 해.
During the whole movie you can click this spot to leave immediately
이 movie 어디에서나 즉시 떠나기 위해 이 spot 을 click 할 수 있다.

## 1. Abstract

In this Part 9, we will reverse a "real" application to learn something about Visual Basic programs(VB), while studying the reversing of a registration scheme.

이 Part 9 에서, registration scheme 을 reversing 공부할 때, 우리는 "real" application 을 Visual Basic 에 대하여 배우기 위해 reverse 할 것이다.
That is because indeed, the best practice is found in real applications.
Real application 에서 가장 좋은 연습 방법을 찾았다.
This registration scheme will also differ from previous parts because it is rather unusual : the program installs a unique key specific for your PC at first startup.
Registration scheme 는 또한 이전 part 와 다르다. 왜냐하면 꽤 일반적이지 않다 : 너의 컴퓨터에 처음 시작할 때 program 은 명확하고 유일한 key 를 install 했다.
For better comprehension and if you are a newbie, I advise you to first see the previous parts in this series before seeing this movie.
좀 더 좋은 이해력을 위해 네가 초보자라면, 이 series 의 첫번째 part 부터 보고 이 movie 를 보라고 충고를 한다.
The goal of this tutorial is to teach you something about a program's behaviour.
이 tutorial 의 목표는 너에게 program's 의 behaviour 를 가르치는 것이다.
In my search not to harm authors, I found an old version of PC to Answering machine(v2.0.0.4).
나의 연구는 제작자에게 해가 되지 않는다. 나는 PC 의 old version 인 Answering machine(v2.0.0.4)을 찾았다.
Taking a look in the specialized media, I also found this application to be "cracked" already.
특별한 media 에 관심을 가져, 이미 "crack"된 application 을 발견했다.
Here, this application is only chosen because it is ideal for this tutorial in reversing and it is targeted for educational purposes only.
여기, 이 application 은 오직 선택됐다. 왜냐하면 이것은 reversing tutorial 에 이상적이다. 그리고 이것은 오직 교육적인 목적이다.
I hope you will exploit your newly acquired knowledge in a positive way.
너는 너의 새롭게 얻은 지식으로 긍정적인 방향으로 이용할 수 있다.
In this matter, I also want to refer to part 1.
이 문제는, Part 1 을 참고하길 바란다.

## 2. Tools and Target

**이것도 똑같음**
The tools for today are : Ollydebug and... your brain.
이 tools 은 ollydebug 와 너의 두뇌다.
The first can be obtained for free at
먼저 free 로 얻을 수 있다.
http://www.ollydbg.de

Numega Technologies' SmartCheck was a software company that was acquired by Compuware in 1997.
Numega Technologies' SmartCheck 는 Compuware 에서 1997 년에 얻은 software company 다.
Compuware only continued development of SmartCheck till late 2001. Then it was discontinued.
Compuware 는 오직 SmartCheck 를 2001 년까지 만들었다. 이것은 비연결적이다.
SmartCheck was old as shareware.
SmartCheck 는 shareware 로 오래됐다.
However now, it can still be found on internet as freeware. Just google for it.
I'm using version 6.20 here.
그러나 이제, 이것은 아직 freeware 로써 흥미를 끄는 점이 있다. Google 로 검색해 봐.
나는 6.20 version 을 사용했다.

As usual, the brain is your responsibility ;)
보통, 두뇌는 너의 책임감이다.

Todays target is the program called PC2Answering Machine Pro v2.0.0.4 For your research, I have included a DIFFERENT old version of this program(v2.0.8.2) in this package
오늘 target program 은 너의 연구에서 PC2Answering Machine Pro v.2.0.0.4 로 불린다. 이 package 에 다른 옛날 version program(v2.0.8.2)를 첨부했다.

## 3. Behaviour of the program

As you can see, I have already loaded the application in Olly and in PEiD.
네가 볼 때, 나는 이미 이 application 을 Olly 와 PEiD 에 load 했다.
Starters should always load a program in PEiD first(or similar like RDG Packer Detector, etc) to see what language it was made in or if/which packer/protector was used.
초보자는 항상 program 을 PEiD 에서 먼저 load 해라.(비슷한 program 인 RDG Packer Detector, etc) 어떤 언어로 만들어졌는지 또는 어떤 packer/protector 가 사용되었는지 보기 위해
More will be explained about this as soon as we deal with those (see later Parts).
We are here at the EP(entry point) of the program.
나는 좀 더 이것에 대해서 바로 설명할 것이다. 우리는 이것과 거래를 했다.(나중 part 에서 보자)
우리는 program 의 EP(entry point)에 있다.

Today, this is what I want you to nice
Because this part will deal with Visual Basic concepts
오늘, 이것은 내가 원하는 것이 무엇인지 보여준다.
왜냐하면 이 part 는 Visual Basic 개념을 거래한다.

INFO :

VISUAL BASIC is a high level programming language (HLL) evolved from the earlier DOS version called BASIC.
VISUAL BASIC 은 DOS version 에서는 BASIC 이라고 불리었던 진화 된 high level program language 다.
BASIC stands for "Beginners All-purpose Symbolic Instruction Code".
BASIC 은 "초보자용 상징 명령 코드" 을 나타냅니다.
VISUAL BASIC is a VISUAL and events driven Programming Language: the programming is done in a graphical environment.
VISUAL BASIC 은 시각적이고 events 중심 Programming Language : programming 은 graphic 환경에서 실행된다.
Programmers may click on a certain object randomly, hence each object has to be programmed independently to be able to response to those actions (events).
Programmer 는 정확한 object 를 랜덤으로 click 할 수 있다. 그리하여 각 object 는 그들의 action(events)에 반응할 수 있게 독립적으로 programmed 됐다.
Therefore, a VISUAL BASIC Program is made up of many "subprograms", each has its own program codes, and each can be executed independently and at the same time each can be linked together in one way or another.
그러므로, VISUAL BASIC program 은 많은 "subprograms"들로 만들어졌다. 각각은 그 program code 와 각각 독립적으로 실행될 수 있다. 같은 시간에 한 가지 방법이나 다른 방법으로 각 link 가 될 수 있다.

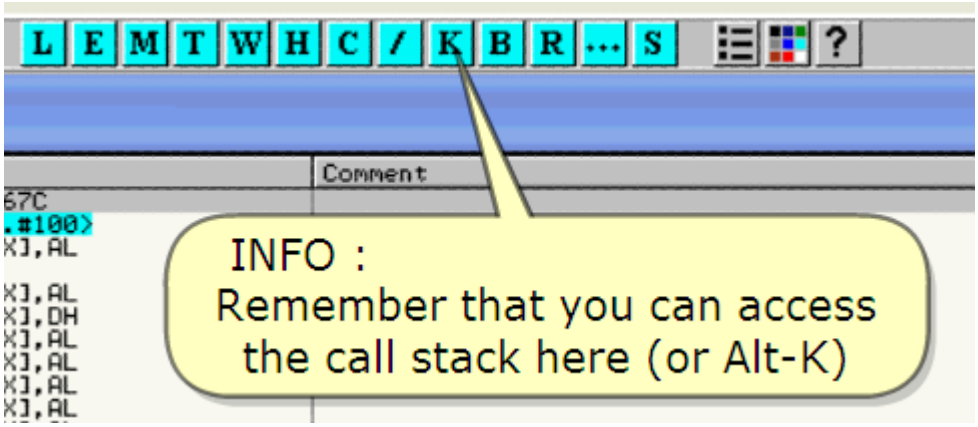
Keep your mouse pointer here and click whenever you are ready reading(on each textscreen)
너의 mouse pointer 를 여기에 유지하고 네가 다음 textscreen 을 읽기 위해 준비가 되었다면 click 해

INFO :
VB Application are fully compiled applications, but have a specific behaviour which complicates OllyDbg's job.
VB Application 은 완전히 compile 된 application 이다. 그러나 특정한 behavior 를 가지고 있다. 그것은 OllyDbg's 의 복잡한 일이다.
Olly is a debugger for compiled languages but is far from ideally constituted to handle VB and as such works far better with languages such as C/C++.
Olly 는 compiled 된 language debugger 다. VB 를 handle 하기 위해 C/C++로 짠 것보다 이상적으로 구성되어 있다.
The VB behaviour looks very efficient by the language and the programmer's point of view: VB programmers can focus on the specific applications' events and behaviors.
VB behaviour 는 language 와 programmer's 의 관점에서 충분히 볼 수 있다. : VB programmer 는 명확한 application events 와 behaviors 에 집중할 수 있다.

INFO :

All VB programs rely on an external dynamic link library (dll) (MSVBVM60.DLL for VB v6.0 and similar dll's for other versions). This dll implements all the APIs and the events.
모든 VB program 은 외부적인 dll 에 신뢰한다.(MSVBVM60.DLL 은 VB v6.0 이고 다른 version 도 비슷한 dll 이다.) 이 dll 은 모든 APIs 와 events 를 시행한다.
Because all the VB APIs are implemented into the dll, the executed code is almost all the time inside this dll or jumps in and out constantly.
왜냐하면 모든 VB APIs 는 dll 에서 시행됐다. 실행된 code 는 거의 모든 시간 dll 안에 있거나 jump 한다. 그리고 끊임없이 있다.
This results into a problem for OllyDbg, and as such, the call stack won't be able to help us further.
이 결과는 OllyDbg 를 위한 문제다. 그리고 그러한 call stack 은 우리를 도와주지 않는다.
This is very important when reversing: the call stack is seldom a help in Olly exactly because the application is almost continuously inside the VB specific dll.
이것은 reversing 할 때 매우 중요하다 : call stack 은 Olly 에서 좀처럼 도와주지 않는다. 왜냐하면 application 은 거의 끊임없이 VB 의 명확한 dll 에 있다.
BTW, the application is mostly the event handlers, used as callbacks from the Dll to answer to specific events/messages.
By the way, application 은 주로 event handler 다. Dll 에서 명확한 events/message 에 대해 대답하기위해 Callbacks 일 때 사용했다.
The rest of the VB application are the resources, the variables and the functions used to associate event-handlers.
다른 VB application 은 resource 가 있다. 다양하고 기능을 event-handlers 와 연관시켜 사용했다.

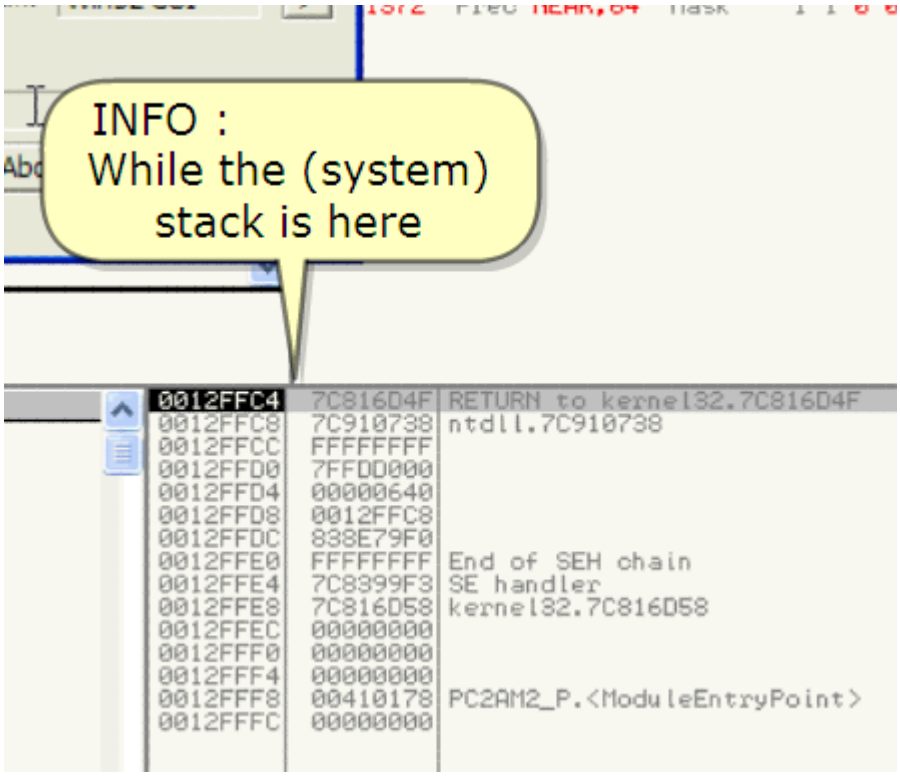
INFO :
VB is stack-based, meaning that it uses the system stack for virtually all its operations.
VB 는 stack base 다. 그것은 가상으로 그 명령을 실행하기 위해 system stack 을 사용한다.
This in contrast to most other languages who use registers for most of the operations and use the stack primarily to perform function call mechanics.
거의 다른 language 는 거의 모든 명령을 등록하여 사용한다. 그리고 주로 function call 을 실행하기 위해 stack 을 사용한다.
Application created in Visual Basic are compiled as interpreted or p-code executables. At run time, the instructions are translated or interpreted by a run-time dynamic-link library.
Visual Basic 으로 만들어진 Application 은 interpreted 나 p-code 가 실행되게 compile 됐다. 실행 시간에, 명령들은 번역되거나 실행 시간에, dynamic-link library 에 의해 해석 된다.
If used, the p-code engine is a simple machine that processes the opcodes.
사용할 때, p-code engine 은 간단한 machine 이다. 그것은 opcode 를 처리한다.
All operands used by p-code instructions are stored on the stack.
모든 operand 는 p-code 명령에 의해 stack 에 저장된다.




INFO :

Remember that you can access the call stack here (or Alt-K)

기억해. 너는 call stack 을 access 할 수 있다. (또는 Alt-K)




INFO :

While the (system) stack is here

Stack 은 여기 있다.

INFO :

In general, a dynamic link library (DLL) is a (collection of) small "program(s)" which can be called upon when needed by the executable program (exe) that is running.

보통, dynamic link library 는 작은 (집합체) "program" 이다. Executable program 에 의해서 필요할 때 불려질 수 있다. Exe 는 실행되고 있다.

Mostly, the DLL lets the executable communicate with a specific device such as a printer or may contain source code to do particular functions.
대부분, dll 은 executable 의 특정한 device 와 함께 전달된다. Printer 나 특정한 function 을 하기 위해 source code 가 들어있다.

INFO :
An example would be if the program (exe) needs to get the free space of your hard drive.
Program 이(exe) hard drive 의 자유로운 공간을 얻기 위해서 필요하다. Example 이 있다.
The program can call the DLL file that contains the function with parameters and a call function.
Program 은 dll file 을 call 한다. 그것은 parameter function 과 call function 을 포함하고 있다.
The DLL will then tell the executable the free space.
이 dll 은 executable free space 를 전할 수 있다.
This allows the executable to be small in size : no need to rewrite the function that was already written in the DLL.
이 executable 은 size 가 작아야 한다 : function 을 재 작성할 필요없다. 이미 DLL 에 작성되어져 있다.
This allows any program to obtain the information about the free space without the author having to rewrite all the source code.
모든 program 은 저작자가 source code 를 재 작성 할 필요 없이 여유공간에 대한 정보를 얻을 수 있다.
It also saves space on your hard drive.
너의 hard drive 공간에 저장할 수 있다.

INFO :
The advantage of DLL files is that, because they do not get loaded into random access memory (RAM) together with the main program, space is saved in RAM.
진화된 dll file 이 있다, RAM 의 공간을 절약하기 위해 main program 과 함께 RAM 에 load 되지 않는다.
When and if a DLL file is called, then it is loaded.
Dll file 은 그것들이 load 된 후에 불려졌다.
For example, you are editing a Microsoft Word document, the printer DLL file does not need to be loaded into RAM.
예를 들어, 우리가 Microsoft Word document 를 수정할 때, printer DLL file 은 RAM 에 load 될 필요가 없다.
If you decide to print the document, then the printer DLL file is loaded and run.
Document 를 print 할 **때** Printer dll file 이 load 되고 실행된다.

INFO :

All in all a DLL is an executable file that cannot run on its own, it can only run from inside an executable file.
모든 DLL 파일은 executable file 이다. 그것은 그들 스스로 실행할 수 없다. 그것은 오직 executable file 의 안에서 실행된다.
To do this, an executable needs to declare the DLL function, then when needed the call is made with required parameters.
이것을 하려면, Executable 은 DLL function 을 선언해야 한다. Call 이 필요할 때 필수 parameter 와 함께 만든다.
Now, for VB and its language specific DLL, all said on this page is also true, except that the DLL is loaded all the time of course.
이제, VB 와 명확한 DLL language 를 위해, 이 page 에서 모든 것을 말했다. 이것은 진실이다.
예외적으로 dll 은 모든 시간에 load 된다.
이제, VB 및 언어 특정 DLL language 를 위해 예외적으로 dll 은 모든 시간에 load 되는 것이 있다.
이것은 진실이다.

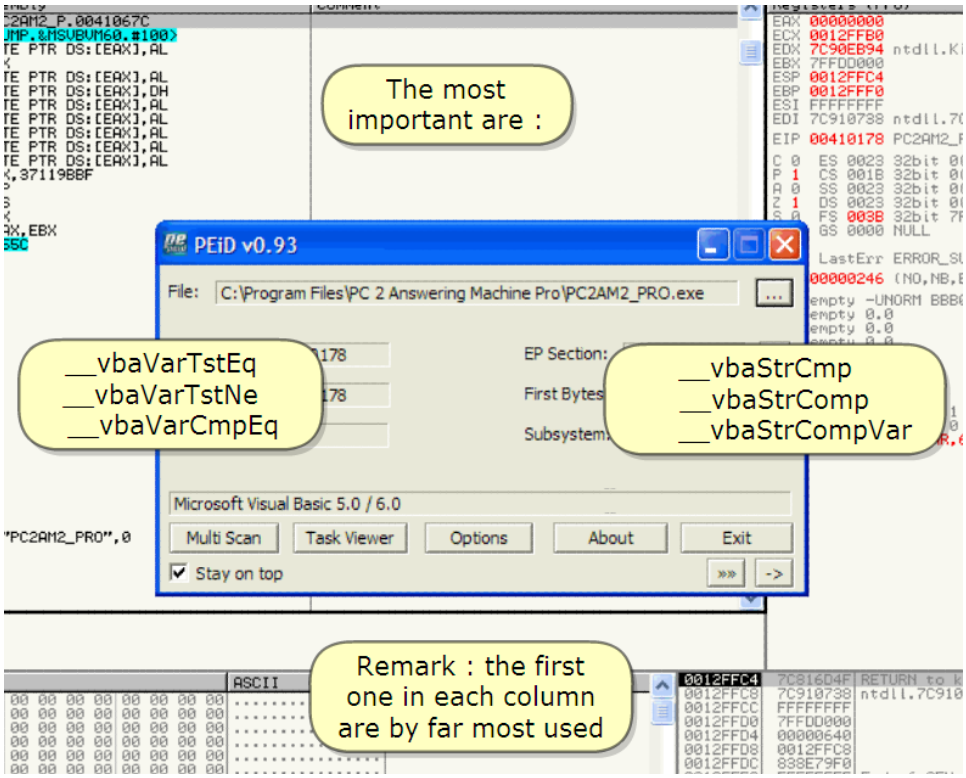
Perhaps you think now that VB will be a very difficult to handle language?
아마 VB 는 handle 하기 매우 어렵다.
Well, it's more the contrary because we have some very useful tools for this which I will explain in the next Parts in this tutorial series.
좋아, 꽤 정반대 된다. 왜냐하면 우리는 이것을 위한 매우 유용한 tool 이 있다. 그것을 이 series 의 다음 part 에서 설명할 것이다.
So, stay tuned ;)
계속 지켜봐 주시기 바랍니다.
However, don't think that Olly is completely useless in VB either : after all, each language is translated to assembly ...
그러나, 생각하지 말자. Olly 는 VB 에서 완벽히 유용하지 않다. : 결국, 각 language 를 assembly 로 번환 한다.

But let's talk about the behaviour of this program.
Program 의 behaviour 에 대하여 이야기 하자.
Well, I kept my eyes wide open and remarked this: at first startup after installing, the program asks for some time and states it is calculating and install "a unique key" for your computer.
좋아, 나의 눈을 넓게 유지하겠다. 그리고 이것을 발언했다 : install 후에 첫번째로 startup 한다.
Program 은 약간의 시간과 계산하는 상태와 install 할 때 너의 computer 의 "a unique key"에 대해 묻는다.
Now, this is rather unusual but it gives us already a good hint: it means the program has made a registration code from some ID(for example a hard disk ID).
이제, 이것은 꽤 일반적이지 않다. 우리에게 좋은 hint 를 준다. Program 은 약간의 ID(예를 들면 hard disk ID)에서 registration code 를 만들었다.

But ... this also states that the program needs to keep this code somewhere so it can verify at each startup if it is registered or not!
그러나 ... 이것은 program 이 code 를 유지할 때 필요하다는 걸 명시한다. 등록되었든지 아니든지 각 startup 에서 검증할 수 있다.
Let's exploit this as in fact, we know enough already.
이 사실을 이용하자. 우리는 이미 충분히 알았다.

## 4. Finding the serial

I said we know enough already, hence, let me explain what we will do.
우리는 이미 충분히 알았다고 말했다. 그리하여 우리가 무엇을 할 지 설명하겠다.
The program needs to verify at each startup if it is registered or not.
등록되었는지 아닌지 Program 은 각 startup 을 검증하기 위해 필요하다.
Verifying this naturally includes a compare "Am I registered ?" in the code somewhere.
자연스럽게 비교하는 "등록되었어?" 것을 code 의 어디엔가 첨부했다.
Well, in VB, this is done in the API's in the DLL.
좋아, VB, API 와 DLL 에서 됐다.




The most important are :
가장 중요한 것은
__vbaVarTstEq
__vbaVarTstNe

__vbaVarCmpEq
__vbaStrCmp
__vbaStrComp
__vbaStrCompVar

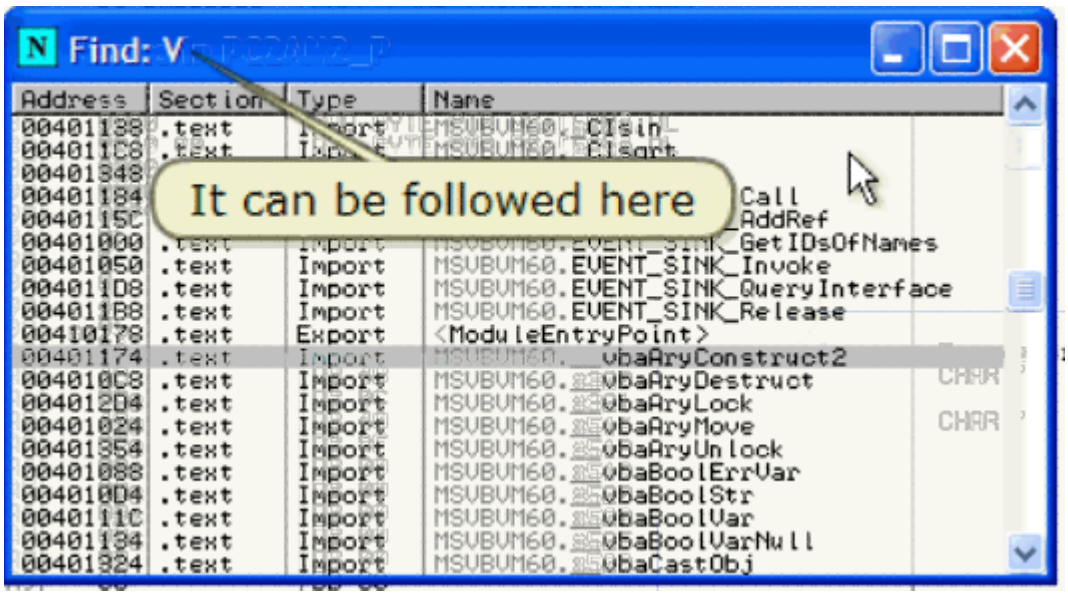
*번역 주)vba Variable Test Equal, Negative, Compare, String*

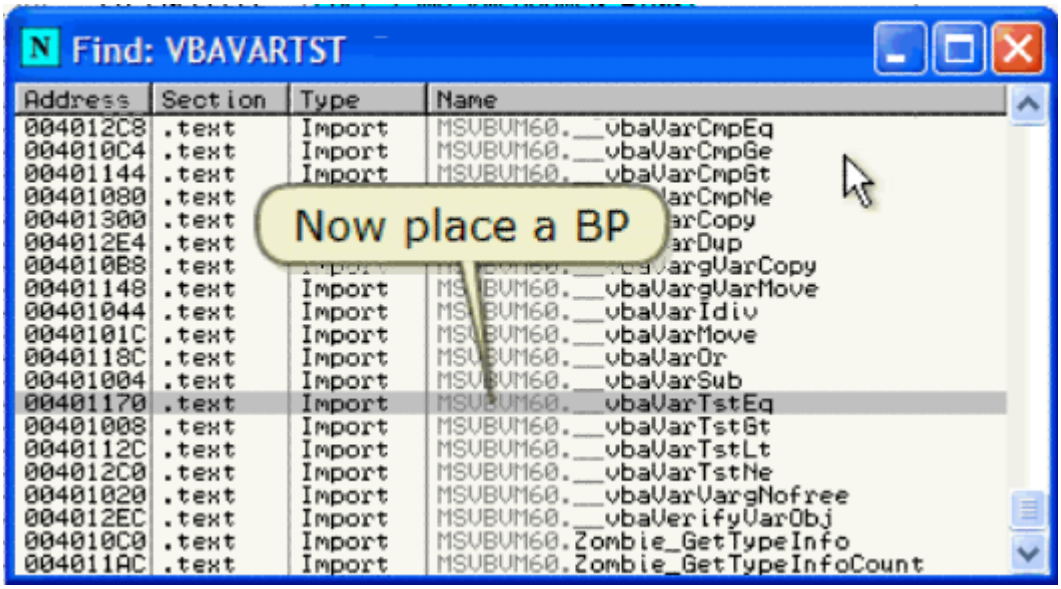
Remark : the first one in each column are by far most used
각 열의 첫번째는 가장 많이 사용되어졌다.

So, let's try the first API and place a breakpoint on it.
First close PEiD, then follow how this can be done ...
첫번째 API 다. Breakpoint 를 설정할 수 있다.
첫번째로 PEiD 를 닫자. 어떻게 끝낼 수 있는지 봐.

And we are at the OEP now
So, press Ctrl-N to show the names module
우리는 OEP 에 있다.
그래서, name module 을 보기 위해 Ctrl-N 을 눌러.

And press vbavartst on the keyboard
.... To easily find the right API Also notice that all AP's are in the DLL
Keyboard 의 Vbavartst 를 눌러.
쉽게 올바른 API 를 찾을 수 있다. DLL 에 모든 API 가 있다.



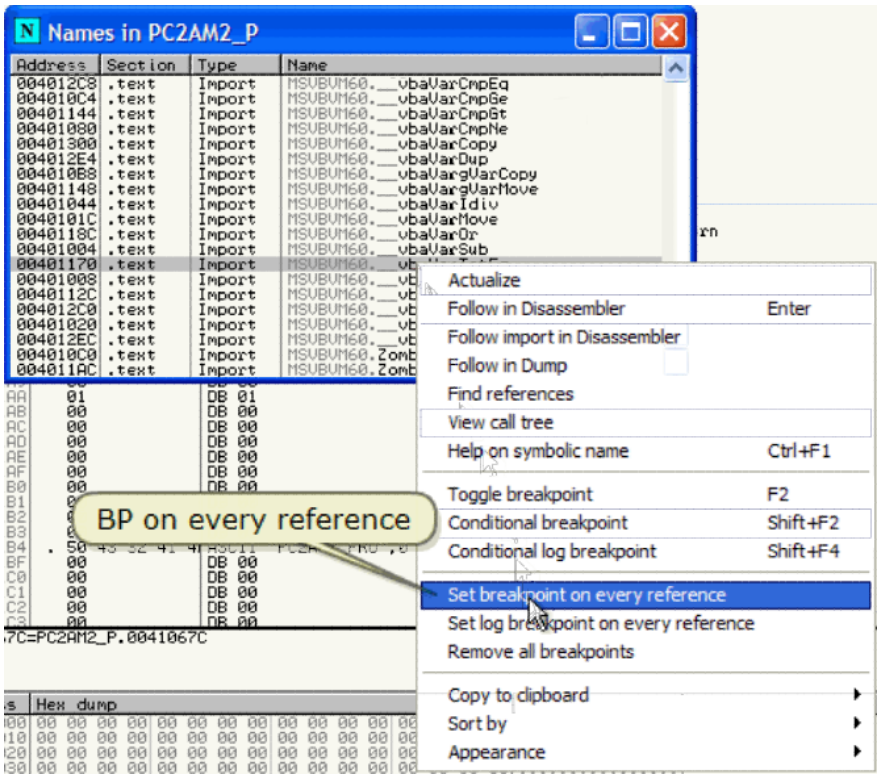


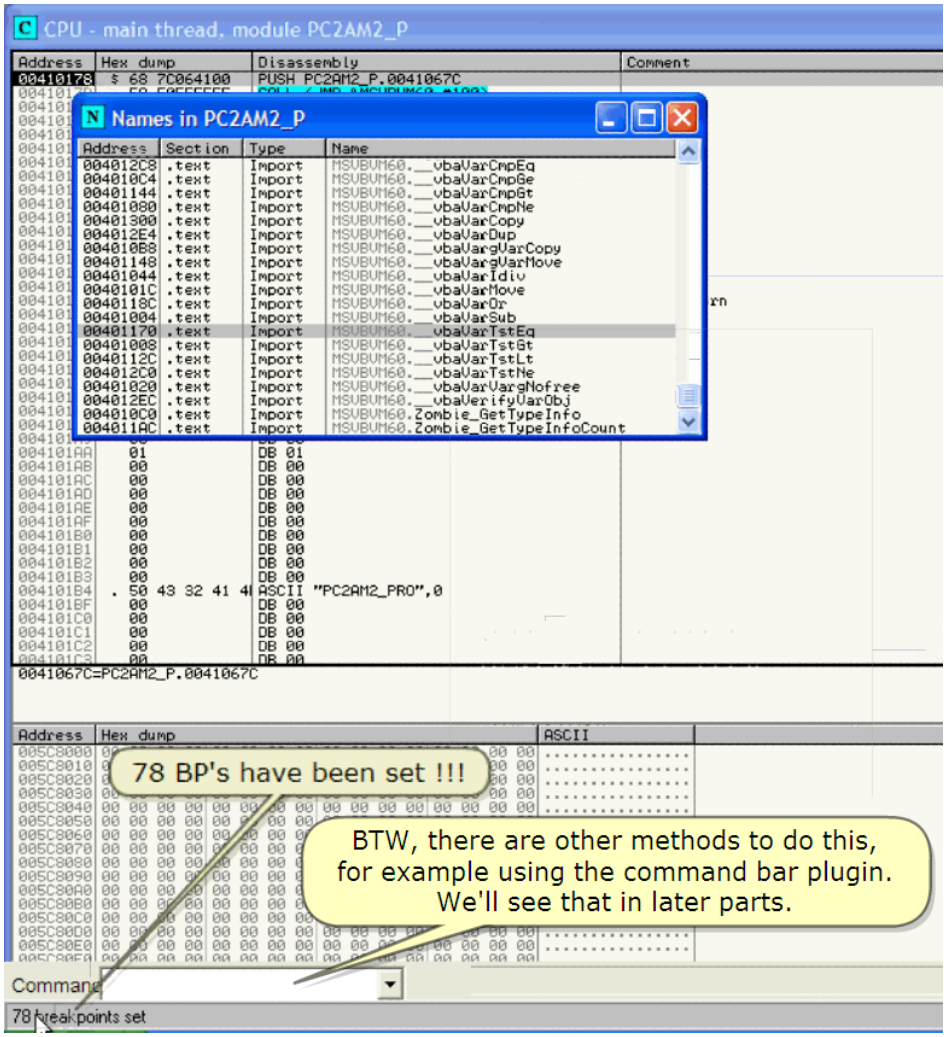

It can be followed here

Now place a BP

이것은 따라갈 수 있다.

이제 BP 를 설치해.






BP on every reference

78 BP's have been set !!!

모든 reference 에 BP 가 있다.

78 BP 가 set 됐다.

BTW, there are other methods to do this, for example using command bar plugin.

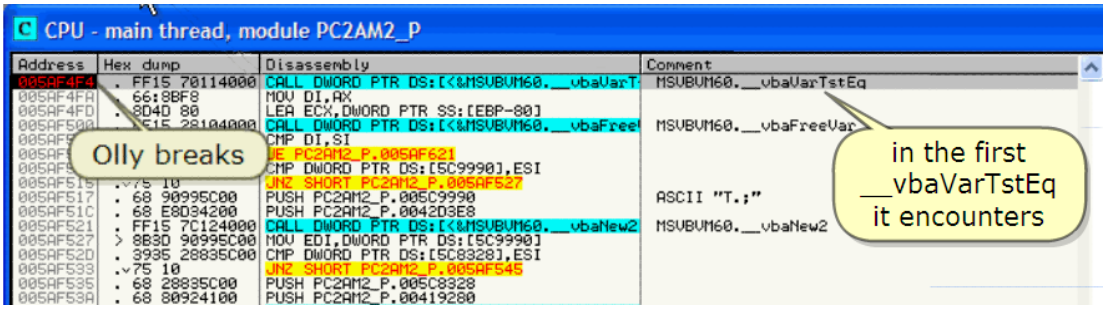
We'll see that in later parts.

By the way, 그곳에는 그 일을 할 다른 method 가 있다. 예를 들어 command bar plugin 을 사용할 수 있다.

좋아 이것은 다음 parts 에서 보자.

And run or F9

실행하거나 F9 를 눌러.




Olly breaks

In the first __vbaVarTstEq it encounters

Olly 는 멈췄다.

처음 __vbaVarTstEq 에 맞닥뜨렸다.

Mmmm, there is not much to see in the code.

음, 그곳은 code 를 보기 위해 충분하지 않다.

Nor above this screen : I scrolled up but removed it from movie

이 위의 screen : 나는 scroll up 했다. 그러나 이것을 movie 에서 삭제했다.

Now, step with F8 to get an overview of the program

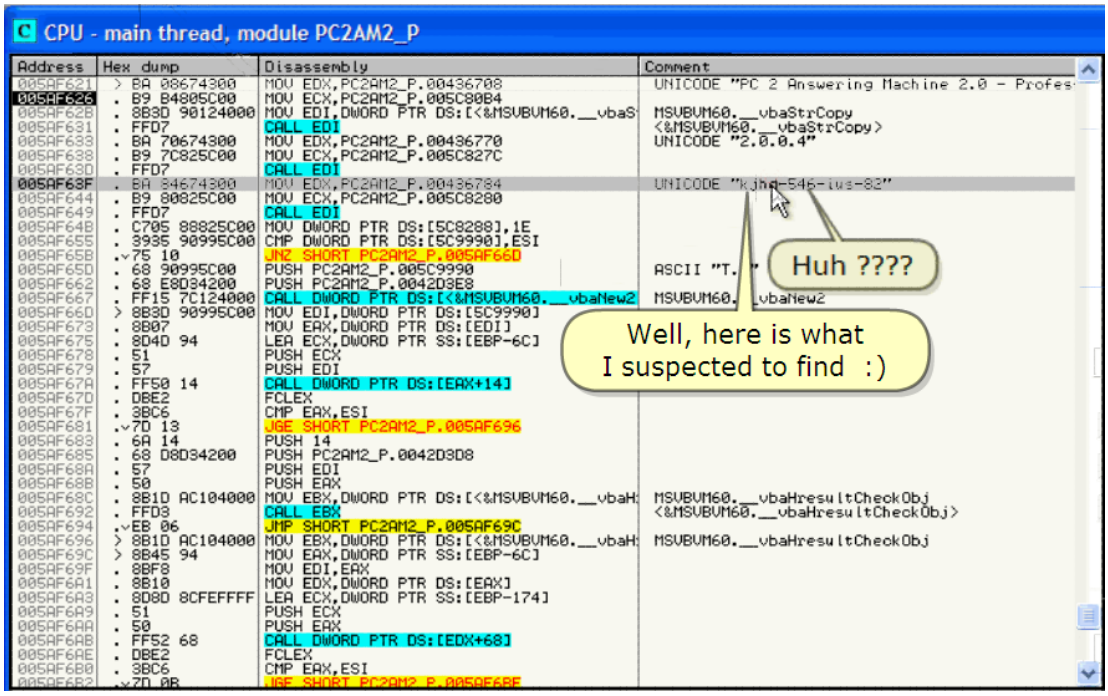
이제, F8 을 눌러 program 의 관점을 얻자.

Hehe, that's interesting but first let's jump and look further on

헤헤, 이것은 매우 흥미롭다. 그러나 먼저 jump 하고 더 멀리 봐라.

Now, take a look in this piece of code. Do you remark something ?

이제, code 조각을 봐라. 무언가를 주목했어?

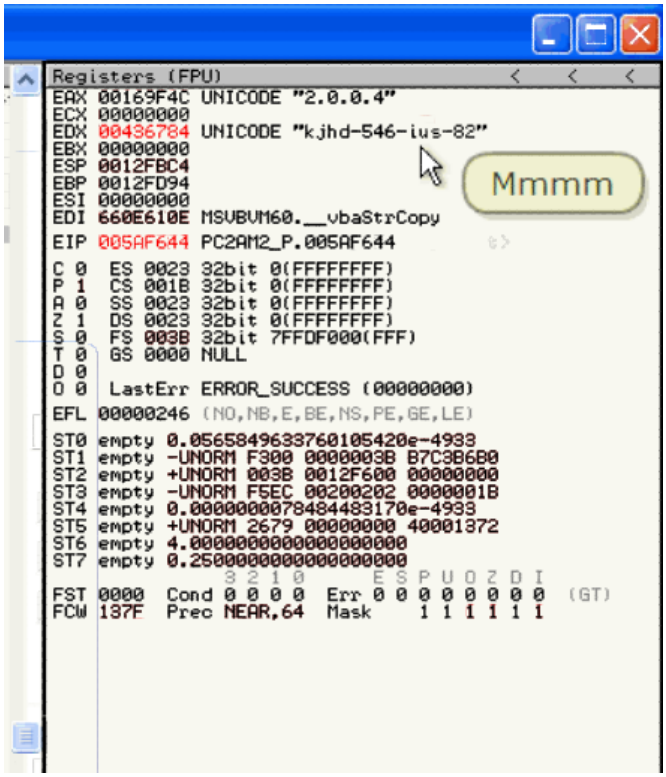



Well, here is what I suspected to find :)

좋아, 이것을 찾기 위해 의심했다.

Continue stepping to see a little further

조금 더 멀리 보기 위해 계속 가보자.




Mmmm

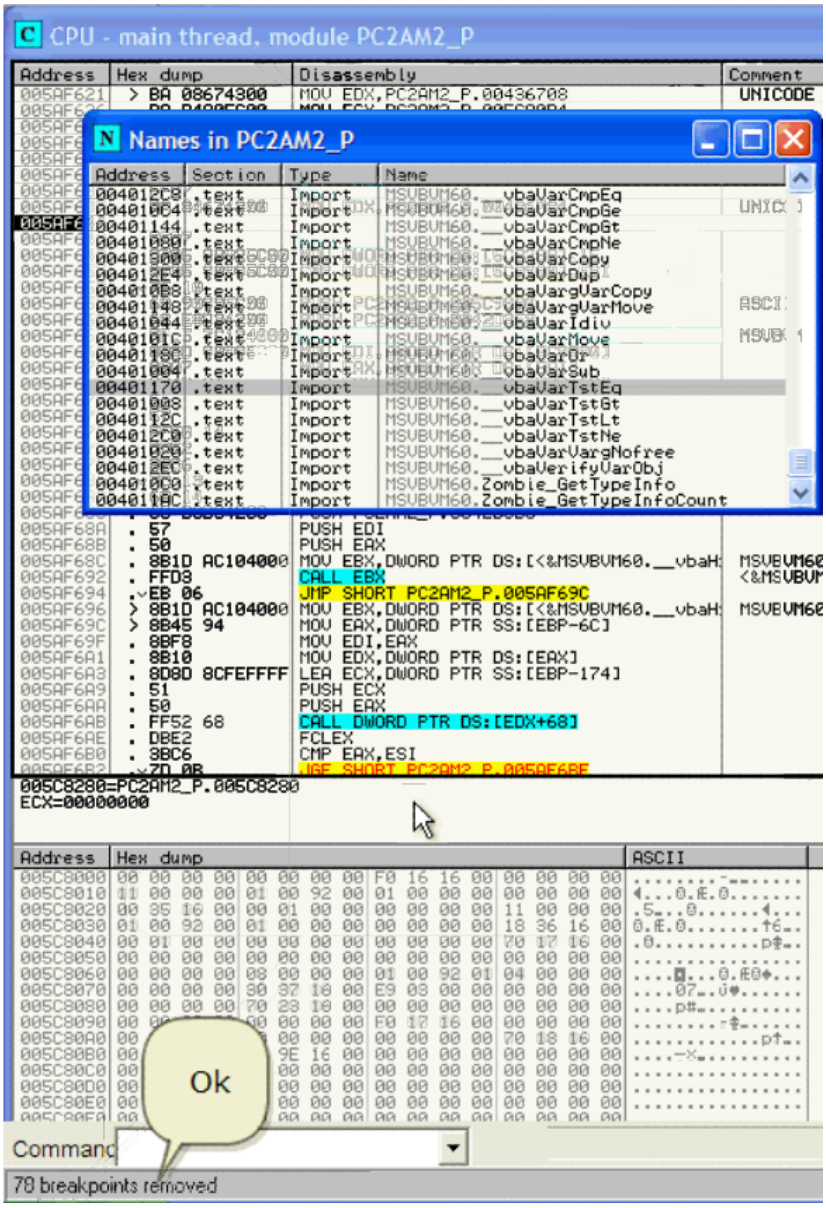
All right, try it out

First remove all BP's

음

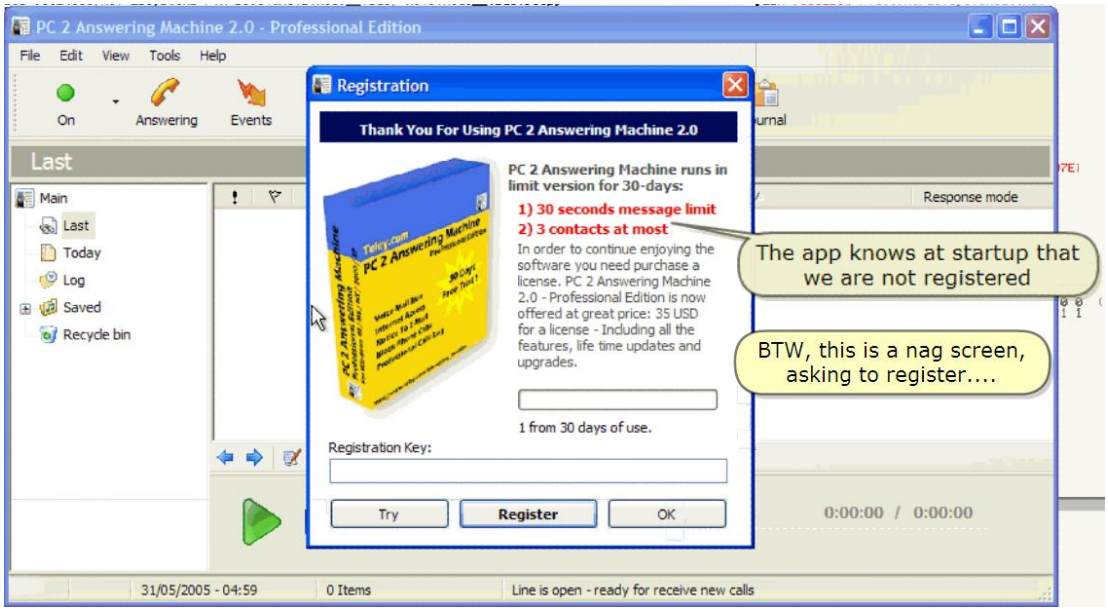
좋아, 해보자.

모든 BP's 를 삭제하자.




Ok

## 5. Registering the program




And run ...

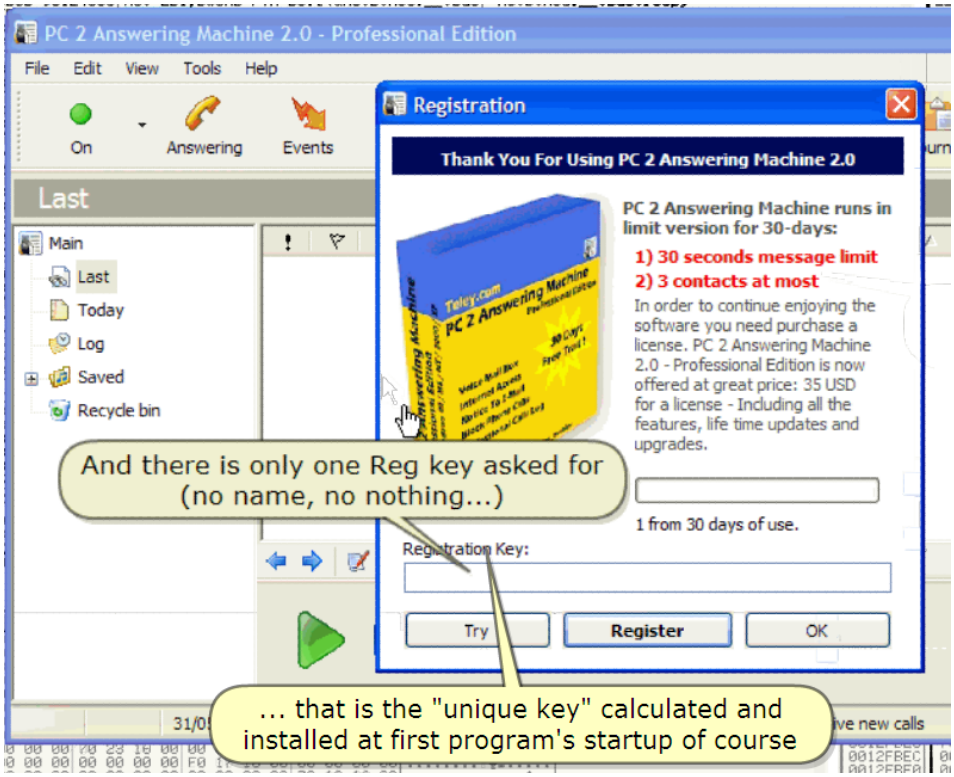
The app knows at startup that we are not registered

BTW, this is a nag scree, asking to register ....

실행하자...

처음 시작할 때 이 app 은 등록되지 않은걸 알고 있다.

By the way, 이 nag scree 은 등록됐는지 물어본다...

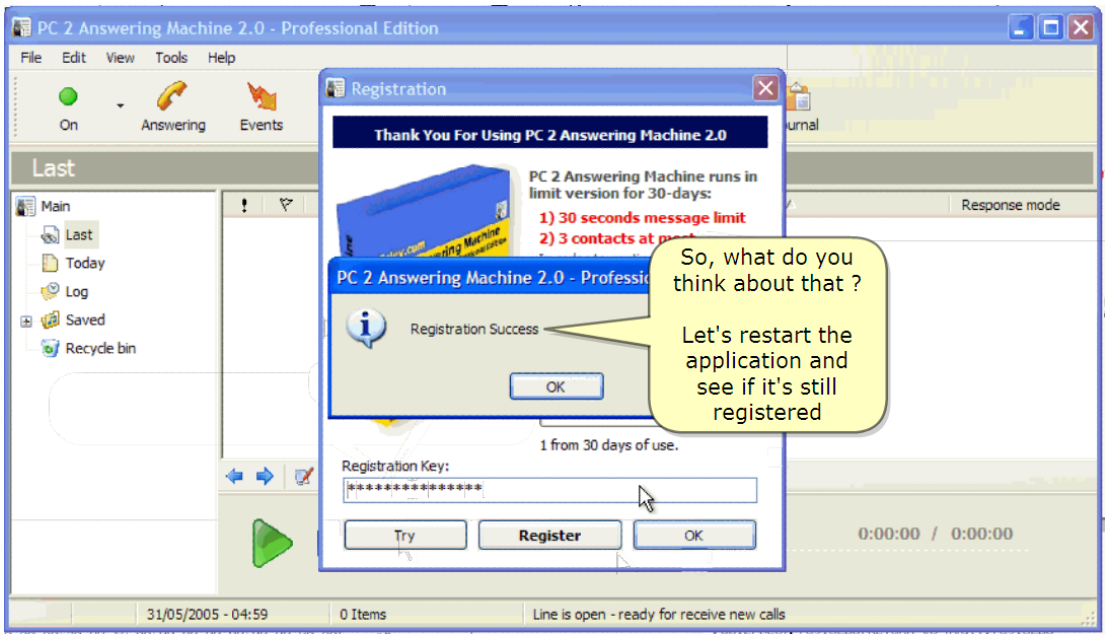



And there is only one Reg key asked for (no name, no nothing...)

...that is the "unique key" calculated and installed at first program's startup of course

그곳은 오직 하나의 reg key 가 묻는다.(no name, no nothing)
그것은 "unique key"로 계산됐다. 그리고 첫번째 program 이 startup 될 때 install 됐다.
*번역 주)즉, 딱 한 개의 key 만 있다는 뜻이다. 예를 들자면 사용자에게 이름과 주소 등을 변수에 넣어서 unique key 를 만든다면 이름이 영향을 주는 요인이 되겠지만 여기에서는 영향을 줄 수 없다. 왜냐하면 이름과 주소를 받는 변수가 없기 때문이다.*

Go for registration :)
Try the key!!!
등록하기 위해 가자 :)
Key 를 찾자.

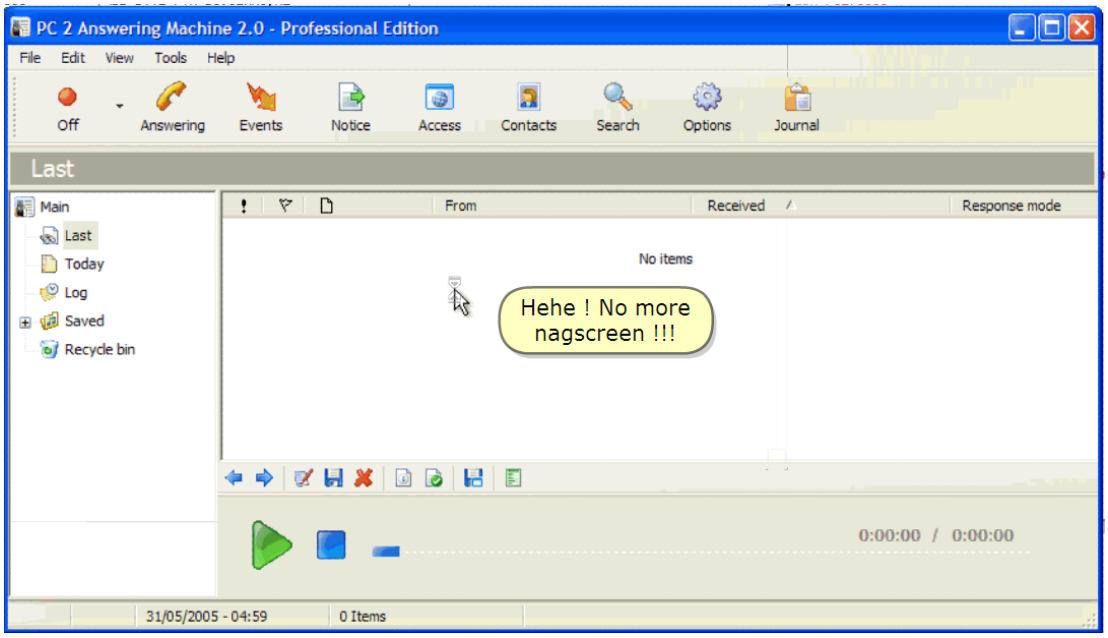



So, what do you think about that?
Let's restart the application and see if it's still registered
너는 이것에 어떻게 생각해?
Application 을 restart 해. 그리고 여전히 등록됐는지 봐.

## 6. Testing the registration

Ctrl-F2 to restart or click here...
Restart 하기 위해 Ctrl-F2 를 click 해.

And run or F9
Run 이나 F9 를 눌러.

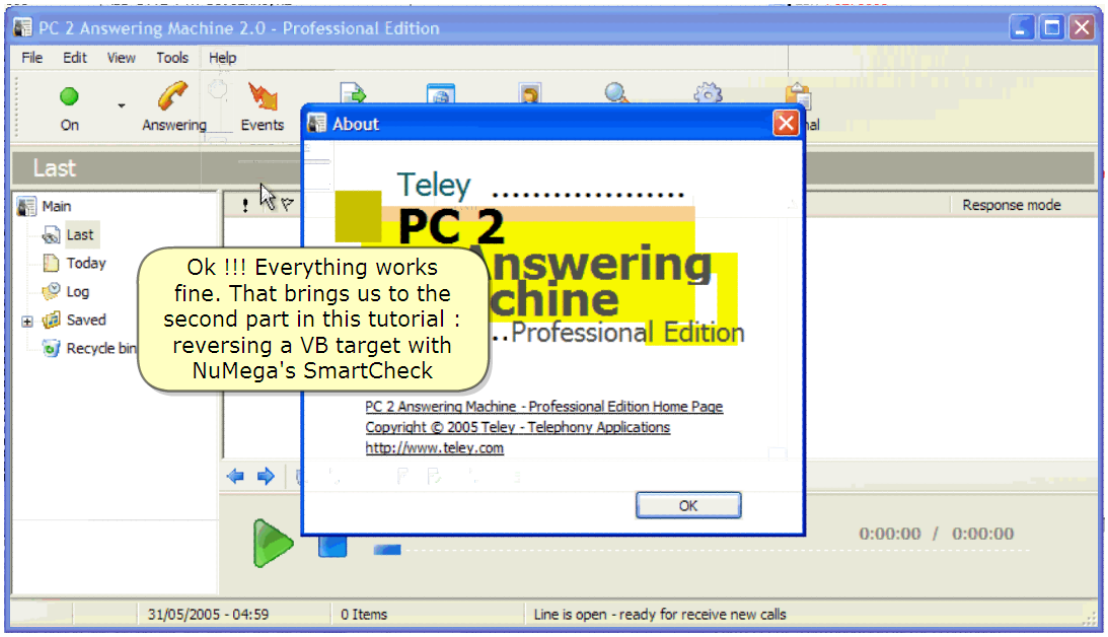



Hehe! No more nagscreen !!!

헤헤! 더 이상 nagscreen 이 없다 !!!

:)

:)




Ok!!! Everything works fine.

That brings us to the second part in this tutorial : reversing a VB target with NuMega's SmartCheck

Ok!!! 모든 게 잘 됐어.

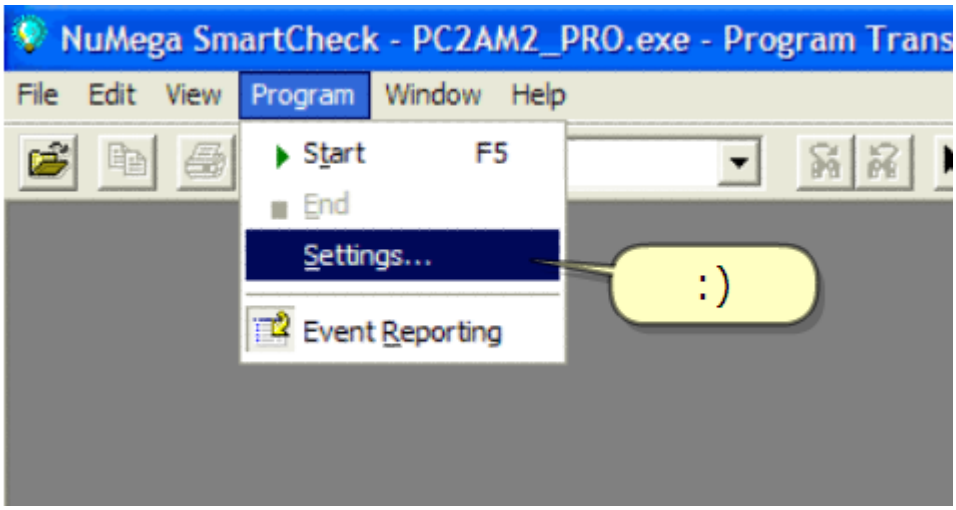
이 tutorial 의 2 번째 part 를 가져온다. : VB target 을 reversing 를 할 때 NuMega's SmartCheck 를 사용한다.

## 7. Configuration of SmartCheck

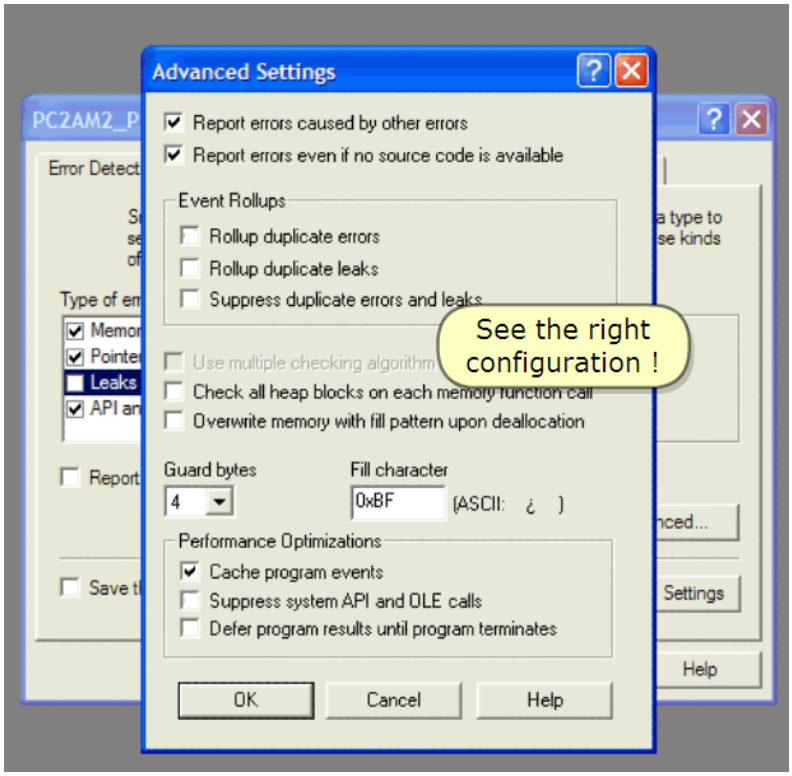
Hence, we come to this new "old" program: SmartCheck from NuMega.
그리하여, 우리는 새로운 "old" program 을 가져온다 : NuMega 에서 SmartCheck
This program was made specifically for VB debugging and reversing.
이 program 은 VB debugging 과 reversing 에 특화되어 만들어졌다.
After installing, SmartCheck is not well configured.
Install 후에, SmartCheck 는 잘 된 설정이 필요없다.
It needs some finetuning in the settings and the configuration, and that can be rather annoying.
Setting 과 configuration 에서 약간의 미세조정이 필요하다. 그리고 꽤 짜증스럽다.
That's why I will show you here how to do it. So, install SmartCheck first and then load a program in the tool.
SmartCheck install 후 이 tool 에서 program 을 load 했을 때 제일 먼저 어떻게 하는지 너에게 보여줄 것이다.
I have chosen the same target as in the first part in this tutorial.
이 tutorial 의 첫번째 part 에서 비슷한 target 을 선택했다.
Remark: I am doing this on a different Computer: on the first computer, PC 2 Answering Machine is registered :)
주목 : 나는 다른 computer 에서 하고 있다. : 첫번째 computer 에서, PC 2 Answering Machine 은 등록됐다 :)

Loading PC 2 Answering Machine Pro
PC 2 Answering Machine Pro 을 Loading 해.

Loaded but automatically minimized
Load 됐다. 그러나 자동으로 최소화 되어있다.

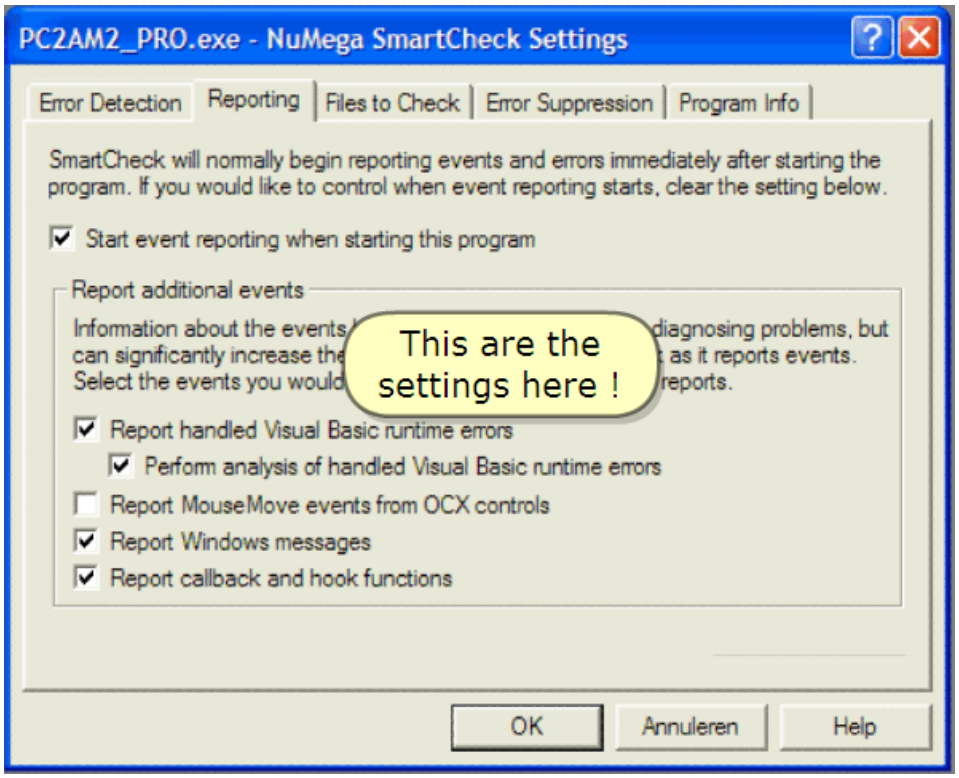


:)



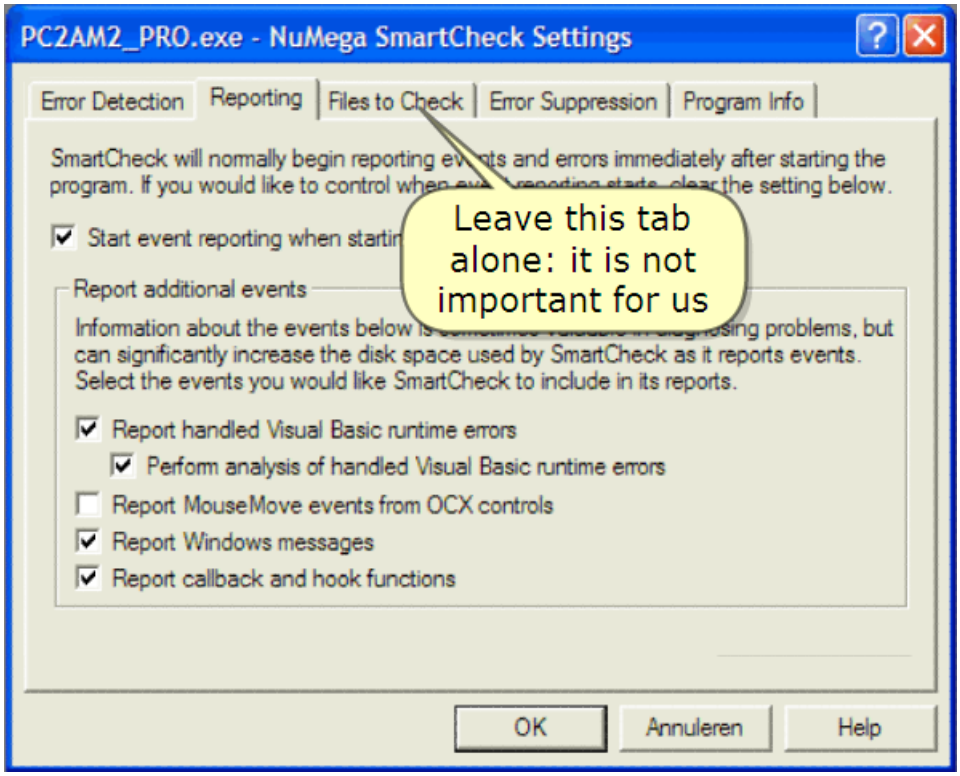

See the right configuration !

Configuration 을 봐.



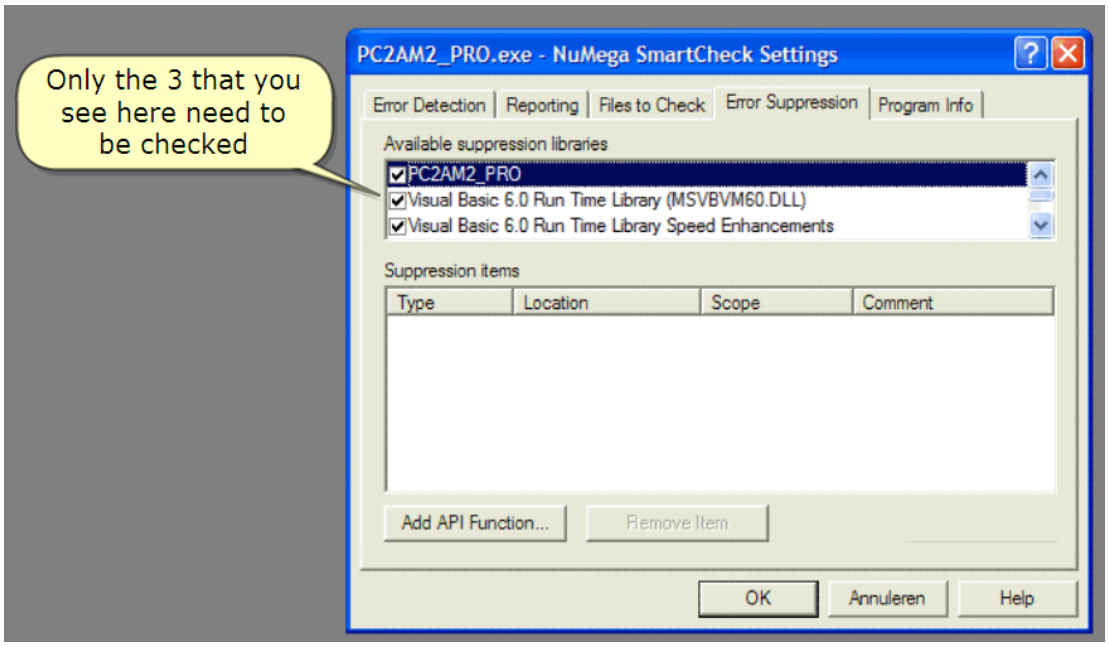

This are the settings here !

이 setting 처럼 해.



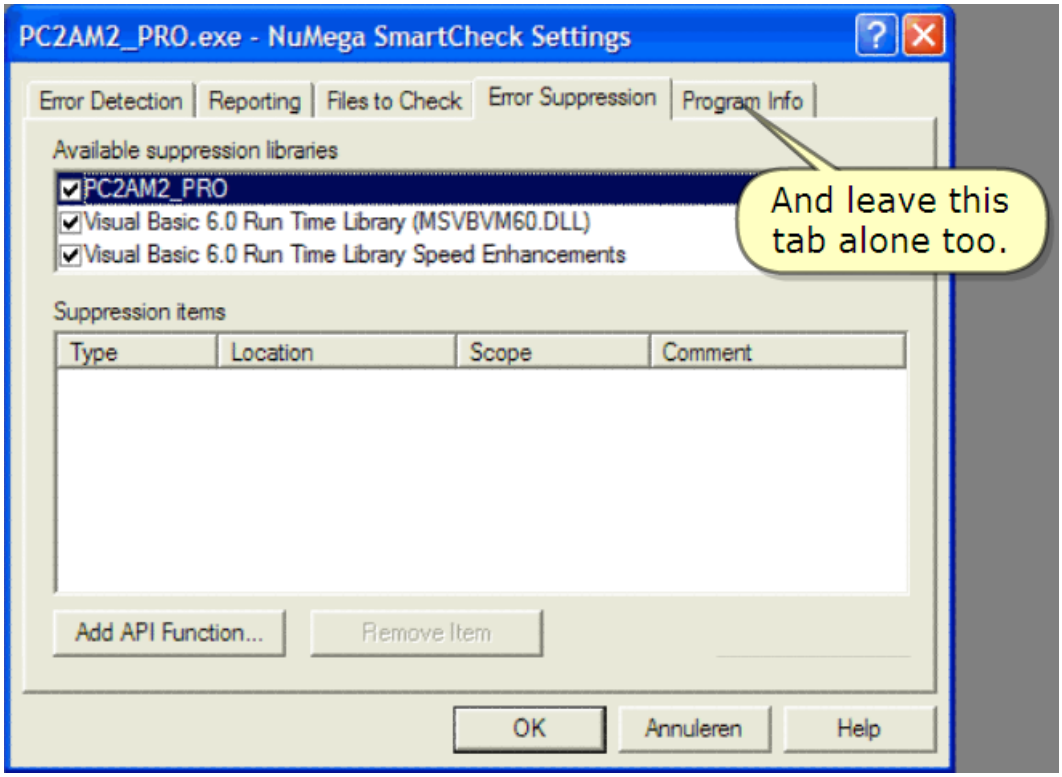

Leave this tab alone: it is not important for us

이 tab 은 건드리지마 : 이것은 우리에게 중요하지 않아.



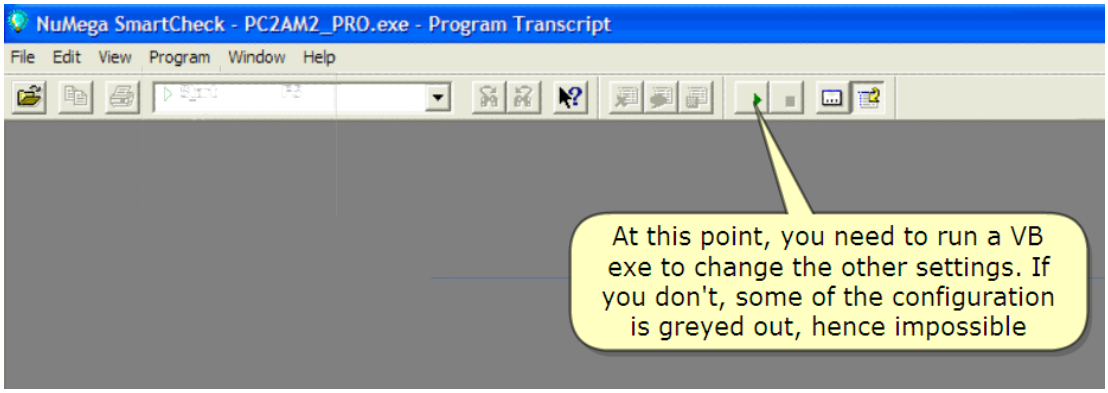

Only the 3 that you see here need to be checked

오직 보이는 3 개만 check 해.




And leave this tab alone too.

그리고 이 tab 은 건드리지마.



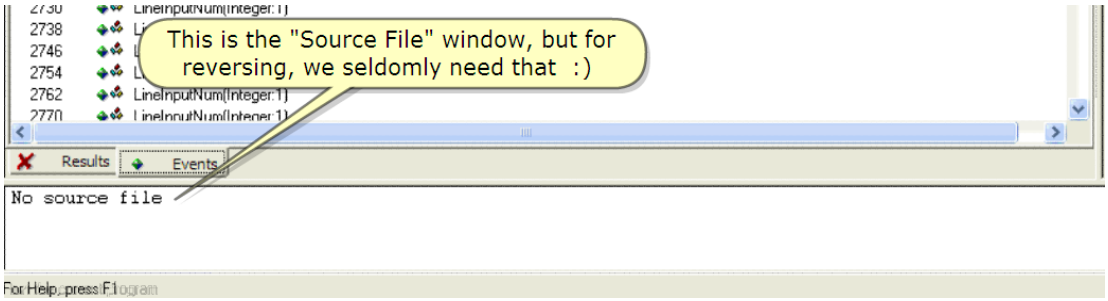

At this point, you need to run a VB exe to change the other settings.

이 point, 다른 설정을 바꾸기 위해서는 VB.exe 를 실행해야 됩니다.

If you don't some of the configuration is greyed out, hence impossible

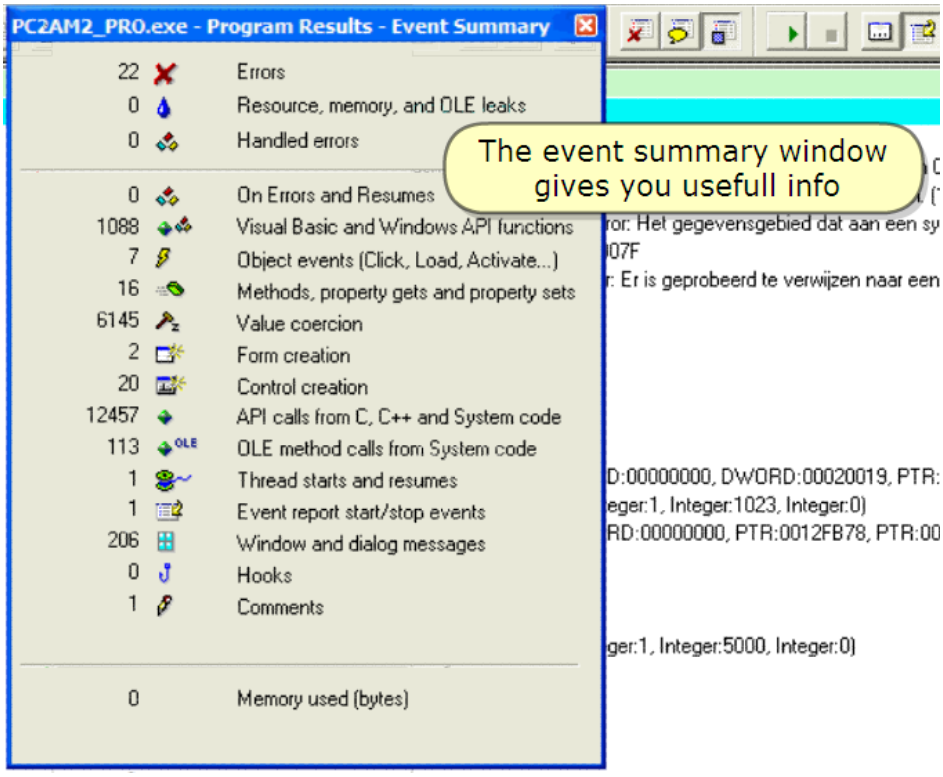
그렇지 않으면 구성의 일부가 회색으로 됩니다. 그러면 불가능하게 됩니다.


Right. PC 2 Answering Machine is running. First of all, make the "Details" window visible if hidden.
좋아. PC 2 Answering Machine 은 실행 중이다. 첫번째로, "Details" window 이 감춰져 있다면 보게 만들자.




This is the "Source File" window, but for reversing, we seldomly need that :)
이것은 "Source File" window 다. 그러나 reversing 을 위해 좀처럼 필요하지 않다.

The event summary window gives you useful info
Event 요약 window 는 우리에게 유용한 정보를 준다.




The goal for you is to see all these exactly like this.

너의 목표는 이것들을 정확히 보는 것이다.

BTW, I'm using SmartCheck v6.20 here.

By the way, 나는 SmartCheck v6.20 을 사용한다.

If your version is different, it doesn't matter: you will easily find your way too because since version 6, there are no big differences.

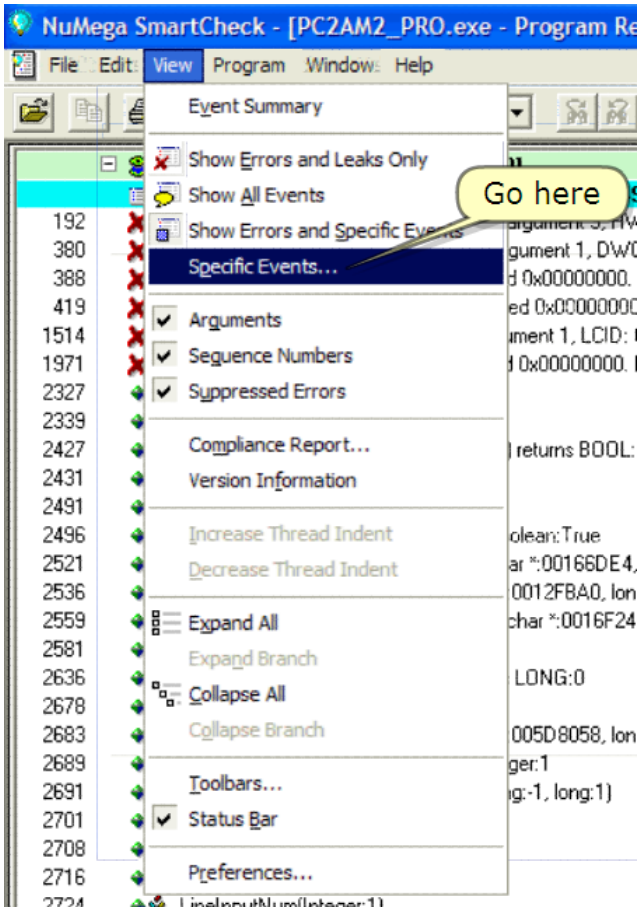
너의 version 이 다르다면, 문제없다. : 너는 쉽게 너의 방법을 찾을 수 있다. 왜냐하면 version 6 이후로, 크게 다른 점이 없다.

Notice also that I have the "Sequence Numbers" checked. I find this useful to find my way in the code later.

"Sequence Numbers"를 check 했다는 것을 알린다. 나중에 Code 에서 나만의 방법으로 이것의 유용한 점을 찾았다.

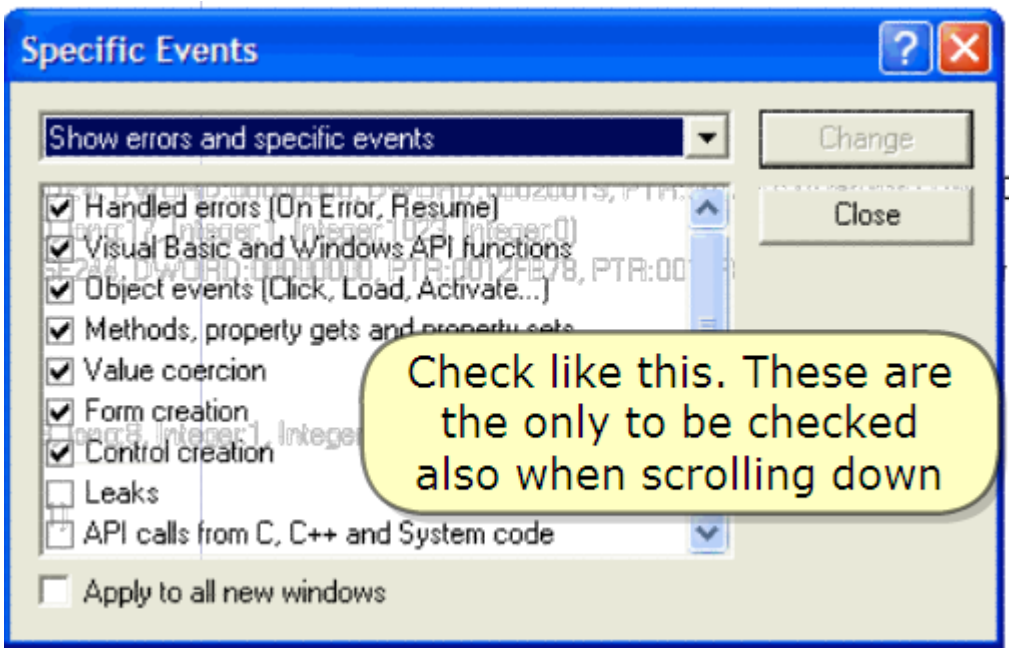
We will see further that it is really easy to drown in the tens of thousands of lines of code without it ....

수 많은 code 에서 쉽게 구별할 수 있다.



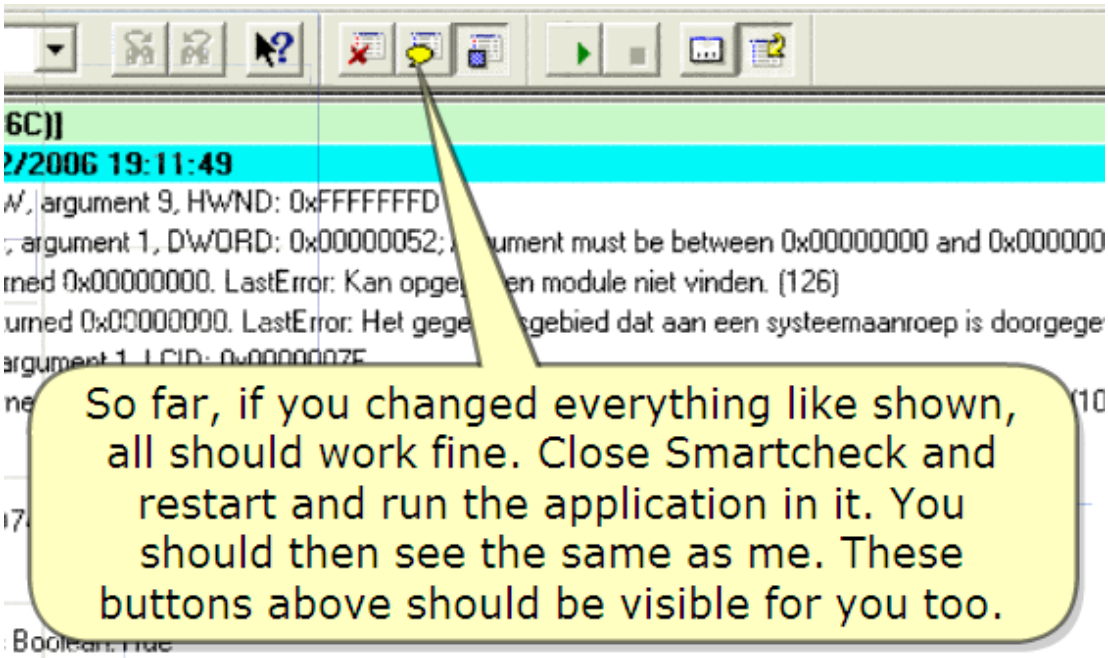

Go here

가자.




Check like this. These are the only to be checked also when scrolling down

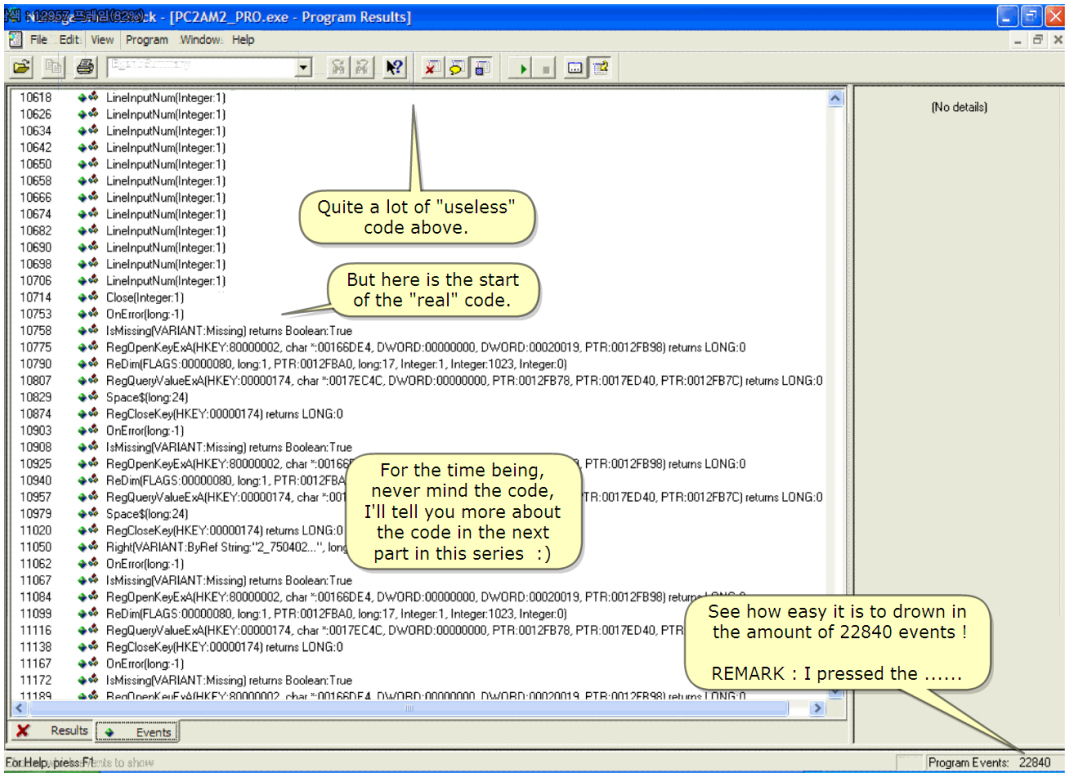
이렇게 check 해. 이것들은 오직 scrolling down 할 때 check 됐다.




So far, if you changed everything like shown, all should work fine. Close SmartCheck and restart and run the application in it.
멀리 보자, 만약에 네가 이것처럼 모든 것을 바꿨다면, 모든 것은 괜찮을 것이다. SmartCheck 를 닫고 restart 해. Application 을 실행하자.
You should then see the same as me. These buttons above should be visible for you too.
너는 나와 같은 걸 보게 될 것이다. 이 buttons 은 너를 위해 보여줄 수 있다.

Finally, SmartCheck is configured.
Let's reverse the application now.
Take an overview and scroll down.
마지막으로, SmartCheck 설정을 했다.
Application 을 reverse 하자.
관점을 가지고 scroll down 하자.

## 8. Fishing the serial in SmartCheck




Quite a lot of "useless" code above.

꽤 "유용하지 않은" code 가 있다.

But here is the start of the "real" code.

그러나 이곳은 "real" code 의 시작이다.

For the time being, never mind the code, I'll tell you more about the code in the next part in this series :)

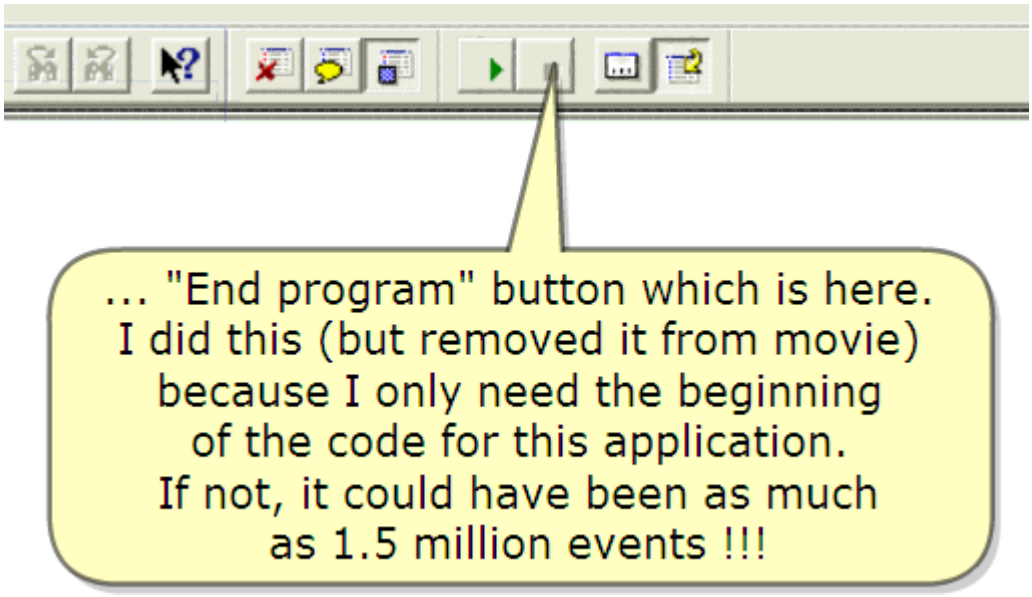
시작하는 사람이라면, code 에 대해서 걱정하지마. 너에게 이 series 의 다음 part 에서 code 에 대해 많은 것을 말할 것이다.

See how easy it is to drown in the amount of 22840 events !

얼마나 쉽게 22840 events 를 보여주는지 봐라.

REMARK : I pressed the .....

주목 : 나는 눌렀다.




... "End program" button which is here.

"End program" button 은 이곳에 있다.

I did this (but removed it from movie)

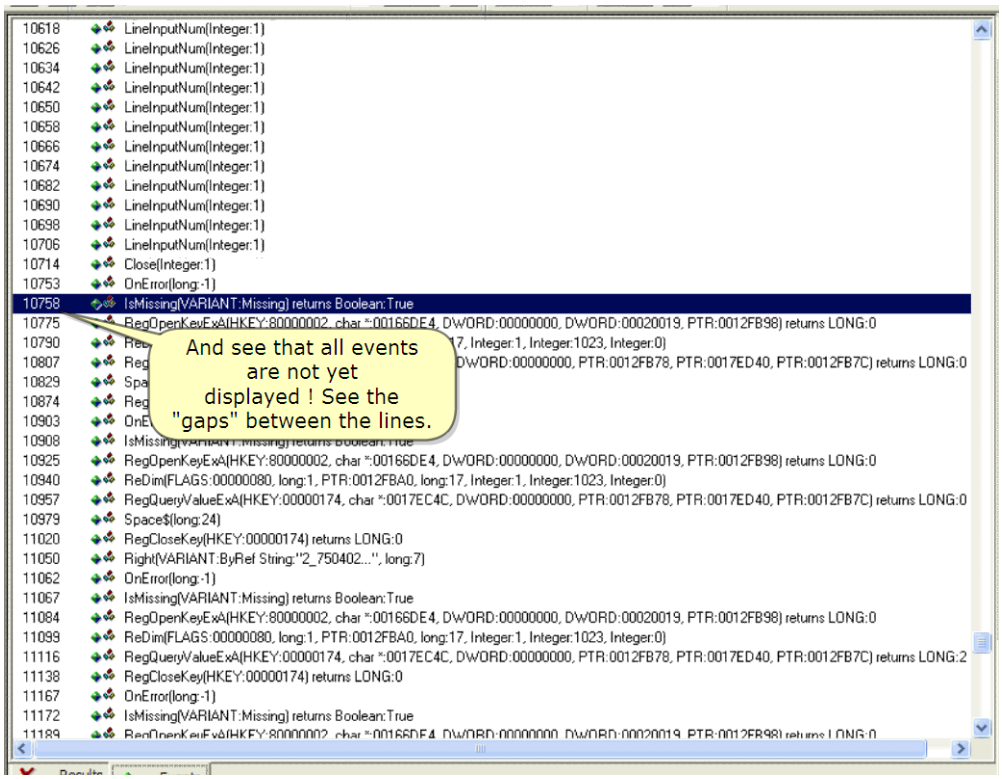
나는 했다.(그러나 movie 에서 삭제했다)

Because I only need the beginning of the code for this application.

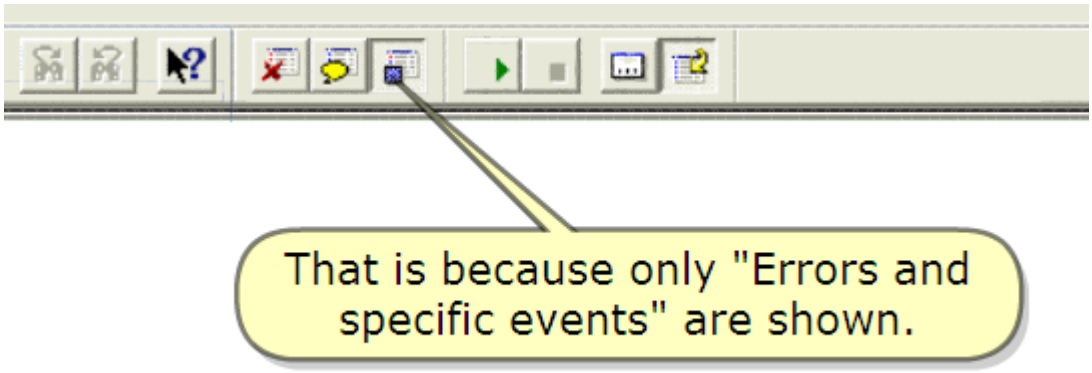
왜냐하면 오직 이 application 의 시작하는 code 가 필요하다.

If not, it could have been as much as 1.5 million events !!!

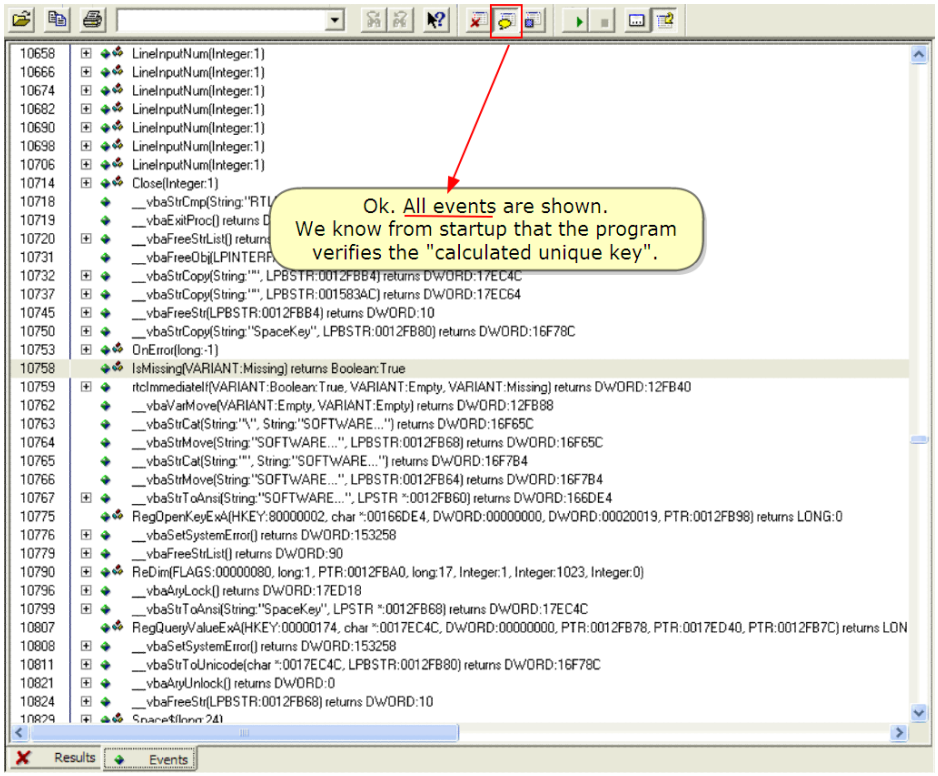
아니라면, 나는 1.5 million events 를 봐야 한다.



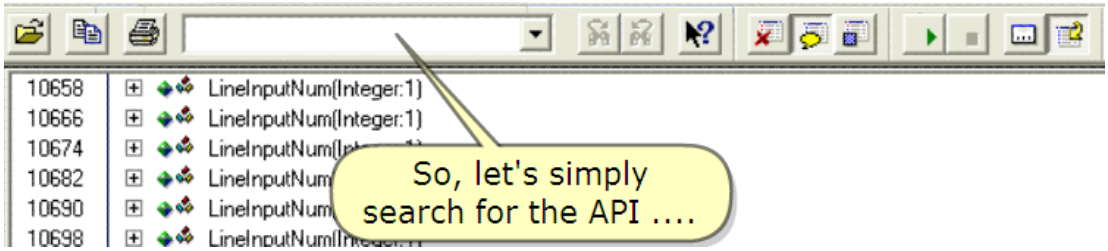
And see that all events are not yet displayed! See the "gaps" between the lines.
모든 events 는 아직 display 되지 않은 것을 봐. Line 사이에 있는 "gaps"을 봐.




That is because only "Errors and specific events" are shown.
오직 "Errors and specific events"가 있다.
*번역 주)10758 과 10775 사이에 있는 차이를 보세요. 그 안에 엄청 많은 event 와 error 가 있겠죠?*

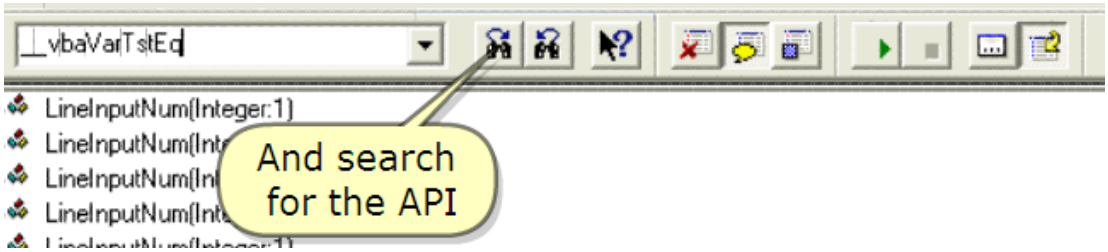



Ok. All events are shown.
We know from startup that the program verifies the "calculated unique key".
Ok. 모든 events 를 보여준다.
우리는 startup 을 알고 있다. Program 이 "계산된 unique key"를 검증한다.



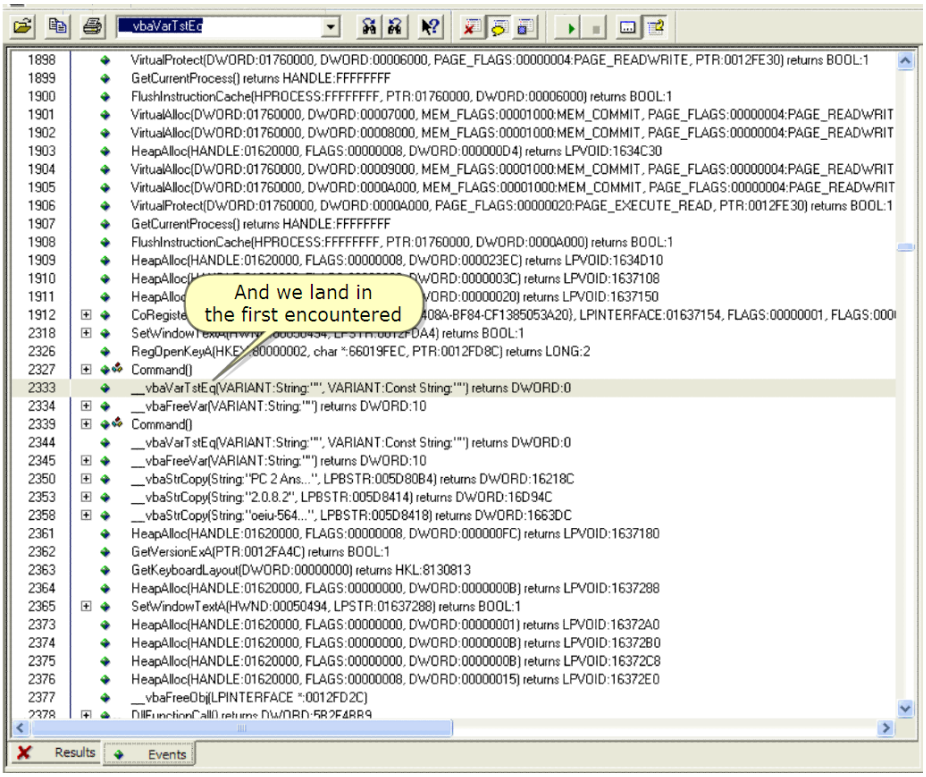

So, let's simply search for the API ....

그래서, API 를 간단히 검색하자.



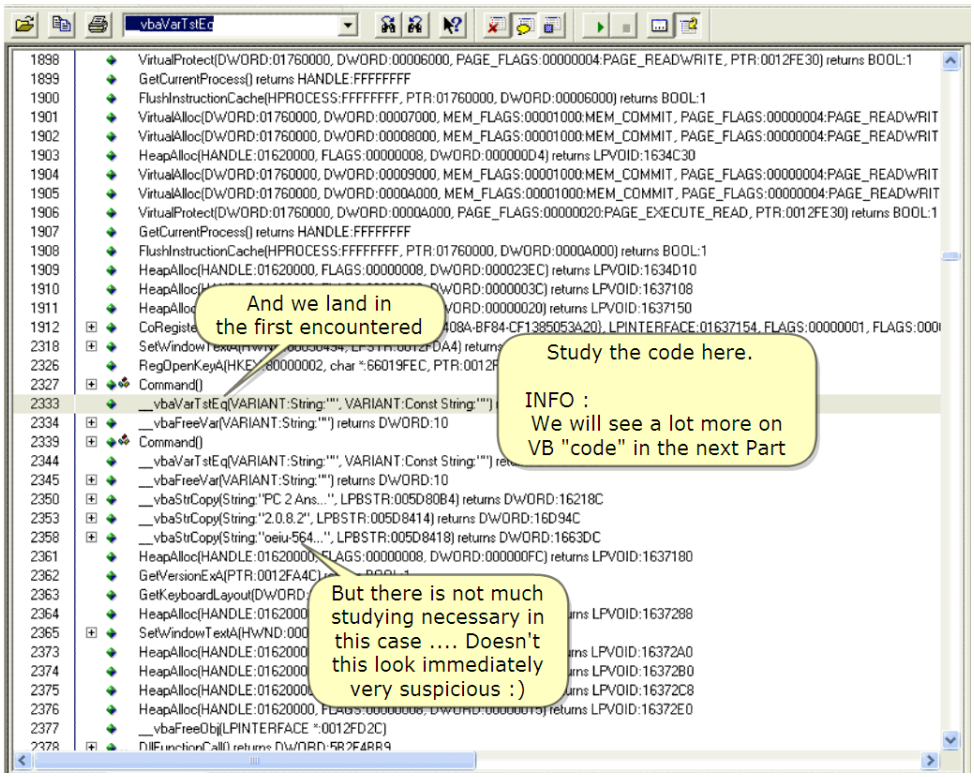

And search for the API

API 를 검색하자.




And we land in the first encountered

그리고 우리는 첫번째로 맞닥뜨렸다.




Study the code here.

여기 Code 를 공부하자.

INFO:

We will see a lot more on VB "code" in the next Part

우리는 다음 part 에서 VB "code" 에 대한 많은 정보를 볼 수 있다.

But there is not much studying necessary in this case....

그러나 이번 경우에는 더 이상 공부가 필요하지 않다.

Doesn't this look immediately very suspicious :)

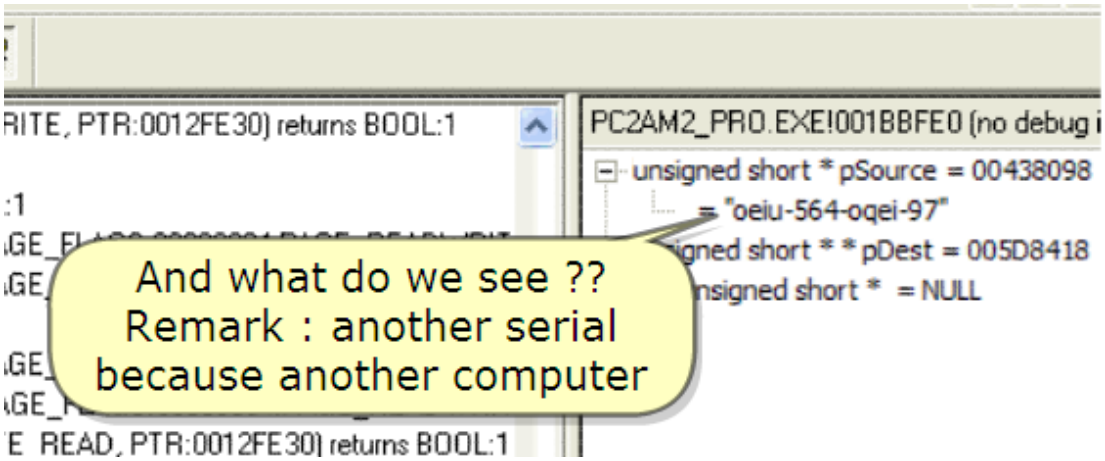
매우 의심스럽다.

Mmmm. Expanding the tree doesn't give us more info.

음. Tree 를 확장해도 정보를 더 이상 주지 않는다.

Let's highlight the line and look in the details window.

Highlight line 을 details window 에서 보자.




And what do we see??

Remark : another serial because another computer

우리는 무엇을 봤는가?

주목 : 다른 computer 이기 때문에 다른 serial 이다.

Because of movie size, I am not testing the serial in this movie. I suppose you believe me that it IS the serial :)

Movie size 를 줄이기 위해 이 movie 에서 serial 을 test 하지 않는다. 이 serial 과 나를 믿을 거라고 생각한다.

REMARK :

This was a rather unusual reversing for SmartCheck because of the comparing at startup.

SmartCheck is very convenient for fishing a serial in a "normal" registration scheme.

We will do that in next Part. See me there!

주목 :

이것은 SmartCheck 를 위한 꽤 일반적이지 않은 reversing 이다. 왜냐하면 startup 할 때 비교한다.

SmartCheck 는 "normal" registration scheme 에서 serial 을 얻을 때 매우 간편하다.

우리는 다음 Part 에서 할 수 있다. 그곳에서 보자.

REMARK :

There exist a number of possible anti-SmartCheck tricks. One of them consists in that the application looks if it finds the name "NuMega SmartCheck".

Anti-SmartCheck trick 의 많은 가능성이 존재한다. "NuMega SmartCheck"를 찾는 Application 으로 구성되어져 있다.

I have not tested this application for it because I had no problems, possibly because my SmartCheck had a "Re-pair0.6" cure :)

그것을 위해 이 application 을 test 하지 않는다. 나는 문제없다, 나의 SmartCheck 는 "Re-pair0.6" cure 이기 때문에 가능하다.

I have included this tool just in case you should have problems with this. Big thanks to the author(s) !
나는 이번에 이 tool 을 첨부했다. 너는 문제가 있을 것이다. 제작자들에게 정말 고맙다.

## 9. Conclusion

We have learned a little about a program made in Visual Basic. In this part 9 of this reversing series, the primary goal was to study the behaviour of a program compiled in VB.
우리는 Visual Basic 에서 program 을 어떻게 만드는지 배웠다. 이번 reversing series 의 Part 9 에서는, 주요한 목표는 VB 에서 compile 된 program 의 behaviour 를 공부하는 것이다.
We have also reversed the same program as well in Olly as in SmartCheck.
우리는 같은 program 을 Olly 와 SmartCheck 에서 reverse 했다.
In this Visual Basic program, due to the calculation of a registration key at startup, it was really easy to find the registration key.
이 Visual Basic program 은 startup 할 때 registration key 를 계산 덕분에 필요하다. 그것은 정말로 registration key 를 찾기 쉽다.
Next Parts in this series will bring you some more difficult VB reversing, also introducing another tool.
이 series 의 다음 part 에서 너에게 약간 어려운 VB reversing 을 가져올 것이다. 또한 다른 tool 도 소개할 것이다.

I hope you understood everything fine and I also hope someone somewhere learned something from this. See ma back in part 10 ;)
네가 모든 것을 좋게 이해했기를 희망한다. 또한 누구든지 어디서든지 이것에서 무언가를 배웠으면 좋겠다. Part 10 에서 보자.

The other parts are available at
다른 parts 는 사용 가능하다.
http://tinyurl.com/27dzdn (tuts4you)
http://tinyurl.com/r89zq (SnD Filez)
http://tinyurl.com/l6srv (fixdown)

Regards to all and especially to you for taking the time to look at this tutorial.
Lena151 (2006, updated 2007)
모두에게 안부를 전하고 특별히 이 tutorial 에 시간을 투자해준 너에게 감사한다.