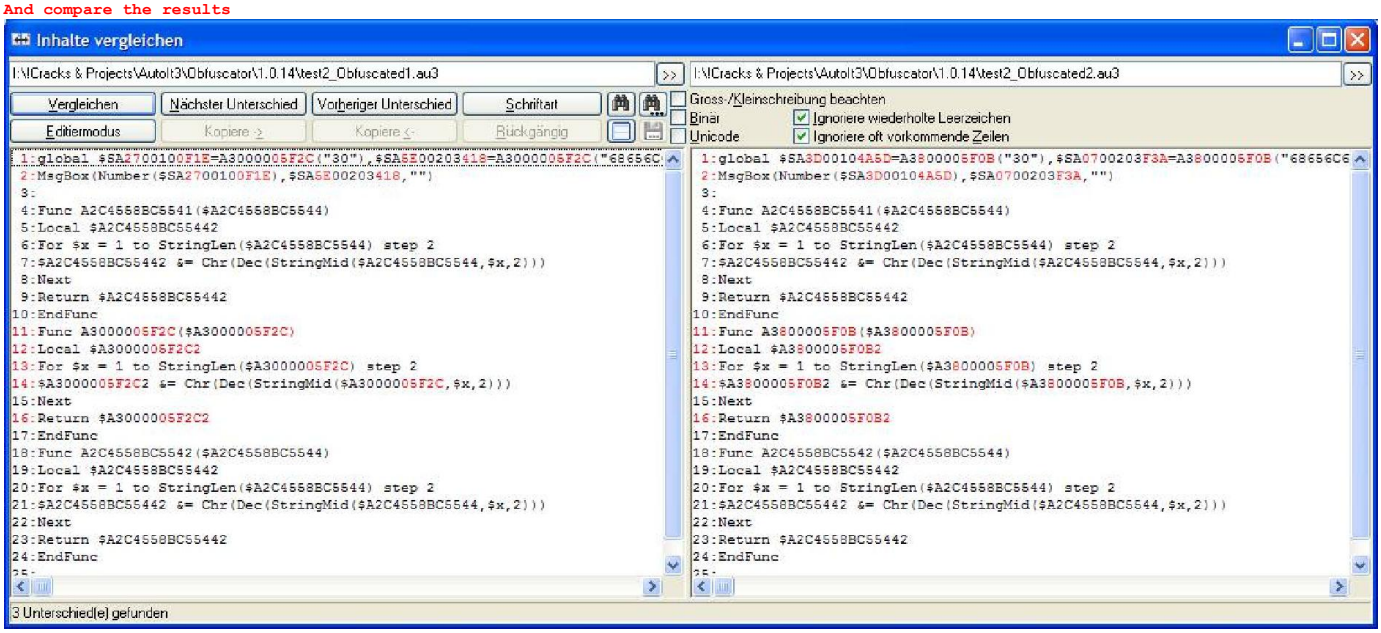
## So how to deobfucate?

*(For example Files and obfucator.exe I used here click >here< )*

Well first of all it about to
watch what the obfuscator does
So let's obfuscate simple this little poggie twice:

```
MsgBox(0,"hello world","")
```

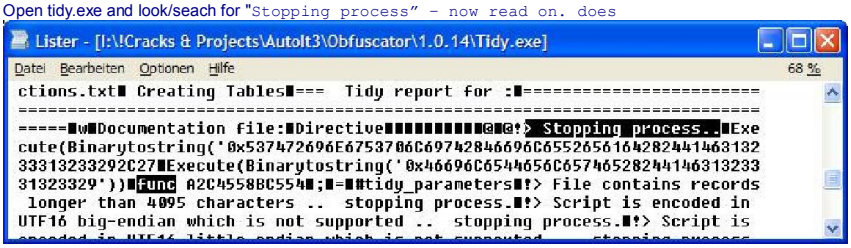
And compare the results




So much about randomisation.
But let's run tidy on this to make it look nicer:

```
->tidy test2_Obfuscated2.au3
Tidy AutoIt3 v2.0.23.24   Copyright (c) Jos van der Zande  September 30, 2008
!> Stopping process..
```

^-Whoops what is this!
Open tidy.exe and look/seach for "Stopping process" - now read on. does




And now think for ya self what Tidy.exe needs strings like
'Execute(Binarytostring('0x537472696E6753706C69742846696C6552656164282441463132333132332927'
'Execute(Binarytostring('0x46696C6544656C6574652824414631323331323329'))'
'func A2C4558BC554'
Yep rioght - regard you cute little example I some had the feeling I've seen it ' func A2C4558BC554' there before.
So now you have the choice to 'improve' tidy.exe using a hexeditor or do a search'n'replace like this 'A2C4558BC554' -> 'B2C4558BC554'in the au3 to tidy.
Now 'test2_Obfuscated1.au3' it looks a little nicer to work with:

```
global $SA2700100F1E = A3000005F2C("30")
global $SA5E00203418 = A3000005F2C("68656C6C6F20776F726C64")
MsgBox(Number($SA2700100F1E), $SA5E00203418, "")
Func A2C4558BC5541($A2C4558BC5544)
    Local $A2C4558BC55442
    For $x = 1 to StringLen($A2C4558BC5544) step 2
        $A2C4558BC55442 &= Chr(Dec(StringMid($A2C4558BC5544, $x, 2)))
    Next
    Return $A2C4558BC55442
EndFunc   ;==>A2C4558BC5541
Func A3000005F2C($A3000005F2C)
    Local $A3000005F2C2
    For $x = 1 to StringLen($A3000005F2C) step 2
        $A3000005F2C2 &= Chr(Dec(StringMid($A3000005F2C, $x, 2)))
    Next
    Return $A3000005F2C2
EndFunc   ;==>A3000005F2C
Func A2C4558BC5542($A2C4558BC5544)
    Local $A2C4558BC55442
    For $x = 1 to StringLen($A2C4558BC5544) step 2
        $A2C4558BC55442 &= Chr(Dec(StringMid($A2C4558BC5544, $x, 2)))
    Next
    Return $A2C4558BC55442
EndFunc   ;==>A2C4558BC5542
```

(^-please excuse my minor edits to original tided source here to make it better readable.)
You recognise our :

```
MsgBox(0,"hello world","")
```

from the beginning? Now it has changer to:

```
global $SA2700100F1E = A3000005F2C("30")
global $SA5E00203418 = A3000005F2C("68656C6C6F20776F726C64")
MsgBox(Number($SA2700100F1E), $SA5E00203418, "")
```

What happened to your "hello world" ?
Obviously it has changed it '68656C6C6F20776F726C64'. And as you also see to restore the original "hello world" string. The function is A3000005F2C() is used.
I bring in the only usefull reference 'Obfuscator.Log' gives us:

```
0.00 ############################## Starting ################################################################################
0.00 ### Processing file:test2.au3
0.02 Special Function: Functionname   Parameter:-1
…
0.02 RandomStringFuncName$: A3000005F2C
```

Let's rename ,A3000005F2C' to 'RandomStringFuncName'

```
Now let's look on the rest of the 3 functions the Obfucator added.
Is A3000005F2C -> 'RandomStringFuncName'
As you can see they are doing all the same but are not use. So you
Can delete A2C4558BC5541 and A2C4558BC5542
global $SA2700100F1E = A3000005F2C("30")
global $SA5E00203418 = A3000005F2C("68656C6C6F20776F726C64")
MsgBox(Number($SA2700100F1E), $SA5E00203418, "")
Func A3000005F2C($A3000005F2C)
   Local $A3000005F2C2
   For $x = 1 to StringLen($A3000005F2C) step 2
       $A3000005F2C2 &= Chr(Dec(StringMid($A3000005F2C, $x, 2)))
   Next
   Return $A3000005F2C2
EndFunc   ;==>A3000005F2C
…and try. Yes 'test2_Obfuscated1.au3' still runs as before.
Let's paste "68656C6C6F20776F726C64" into some hexeditor(I used Winhex here) to see what it is
Offset      0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000000   68 65 6C 6C 6F 20 77 6F  72 6C 64                  hello world
Hahar it's just as it is - no additional decryption or stuff is applied. Just convert the Hexstring to a BinString.
Replace all
$SA5E00203418 -> "hello world"
$SA2700100F1E -> "0"
The result will be
MsgBox(Number("0"),"hello world,"")

Or if you undo the last replace and use instead
Number($SA2700100F1E) -> 0
We get
MsgBox(0,"hello world","")
What is our unobfucated source!!!
```

## Writing a Deobfucator

Here I'm trying to give you the idea on how you break down the main task into several smaller task that can easy solved by a programming language.
Regard this as pseudo code. As code that is not meant to be direct executable but is there for to show the algorithm. And after understanding it makes you to 'easy' implement it in the programming language ya coding in.

### 1. detect if the script is obfuscated + detect the kind+version of obfucator that was used

Let's do it as tidy does it:
Scan for string "func A2C4558BC554" -> "AutoIt3 Source Obfuscator v1.0.14 dectected" else quit (or handle it differently)

### 2. Finding stringTranform functionname (RandomStringFuncName)

here this lines is really good for

```
global $SA2700100F1E = A3000005F2C("30")
```

```
I underlined the matchpatterns to mark the start and end of the string that should be cut out.
This will do the job
RandomStringFuncName = CropOutString("global $* =", "(")
RandomStringFuncName is now "A3000005F2C"
```

### 3. Getting search & replace data

```
SplitedScript = StringSplit( WholeScriptText,
    " = " & RandomStringFuncName&"("
This is more illustrative but not so general:
SplitedScript = StringSplit( MywholeScript,
    " = A3000005F2C ("
```

The result:

```
SplitedScript[0]="global $SA2700100F1E"
SplitedScript[1]=""30")"
SplitedScript[2]="global $SA5E00203418"
SplitedScript[3]=""68656C6C6F20776F726C64")"
```

… okay after some more cleaning (…I'm not gonna boring you this that / make your mind to lazy – Please code this yaself …)

```
SeachReplace[0].Search = "$SA2700100F1E"
SeachReplace[0].Replace = "30"
SeachReplace[1].Search = "$SA5E00203418"
SeachReplace[1].Replace = "68656C6C6F20776F726C64"
```

And after applied HexString to BinString

```
SeachReplace[0].Search = "$SA2700100F1E"
SeachReplace[0].Replace = "0"
SeachReplace[1].Search = "$SA5E00203418"
SeachReplace[1].Replace = "hello world"
```

### 4. Appling the search & replace data

```
Dim item as Collection
For each item in SeachReplace
Replace(WholeScriptText
"Number(" & Item.search & ")"
Item.replace)
Replace(WholeScriptText
Item.search
""" & Item.replace & """)
Next
```

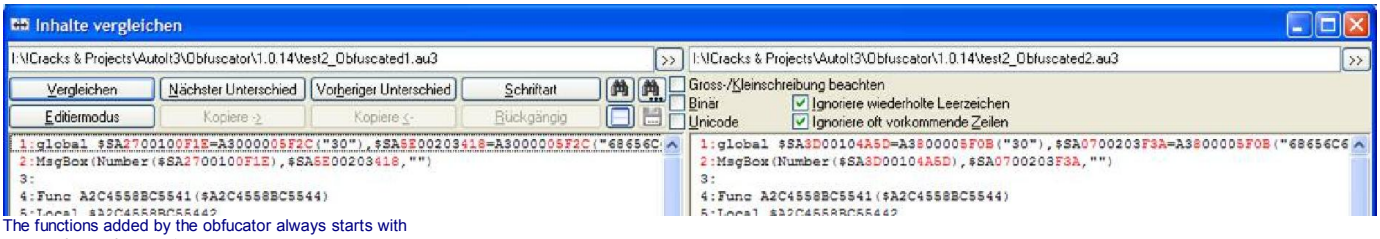
Okay why apply replace two times – well that will make the output nicer.
Hope this example will make you to understand for it works:

```
MsgBox(Number($SA2700100F1E), $SA5E00203418, "")
Number($SA2700100F1E)-> 0          MsgBox(0, $SA5E00203418, "")
$SA2700100F1E -> "0"               MsgBox(0, $SA5E00203418, "")

Number($SA5E00203418)-> hello worldMsgBox(0, $SA5E00203418, "")
$SA5E00203418->"hello world"       MsgBox(0, "hello world", "")
MsgBox(0, "hello world", "")
```

### 5. Cleaning up the script

Remember the compare?




The functions added by the obfucator always starts with
`Func A2C4558BC5541`
So let's simply cut the script there
`WholeScriptText= stringSplit(WholeScriptText," Func A2C4558BC5541")(0))`
…only uses the first part `stringSplit` returns and forget/delete the rest.
And what about those crappy
```
global $SA2700100F1E = A3000005F2C("30")
```
…
that are still there.
Here maybe a search & replace with regular expression can help out.
Search for
`".*global $.* = A3000005F2C(.*)\n"`
Replace with
`""`
But there are probably many other ways to - like to care of to remove them in the Step '3. Getting search & replace data'

Okay that it - hope you enjoy this little tut and got some inspiration from it.
;)

---

A preview:
New versions of the van Zande AutoIt3 Obfuscators store strings in a *.tbl file that regarding the example looks like that
**30o68656C6C6F20776F726C64o**
Instead of using **global**. You noticed the 'o' in the hex to seperate the Strings? Good. J And the now current version even uses randomise string separator for variable length.(so that it is not always 'o').
<Sorry I would be a good end for that tut – but I couldn't resist to continue>
Like
30!06O68656C6C6F20776F726C64!06O
30O428T68656C6C6F20776F726C64O428T

So you you may show the *.tbl to the user and hope he will enter the correct separator like 'O428T'.
Or you may use a heuristics like this on the tbl file.
Get last 3 chars (since separator length varies from 3 to 6 and may be more) check how many occurrences there are in the file and the add one char more to the possible separator string…
'28T' - 2 occurrence found (memorise this)
'428T' - 2 occurrence found
'O428T' - 2 occurrence found
'4O428T' - 1 occurrence found -> Stop here
Seperator string is 'O428T'
+ Validate if the separated HexString have an even length like: 2,4,6,8…

Or exploit the fact that first and last char of the separator string is NonNummeric
'O428T' or 't67|'
Okay heuristic are called like this because they might fail under certain circumstances. (Well regarding the last heuristic it is actually no heuristic anymore because it's safe and applied the fact of the last heuristic on the first makes also that safe. So this makes to a algorithm but will I hope you got the idea of it.)

The another way is to extract it as 'first-hand-information' directly from code in the au3-script.
And I confident enough that you can tell me your way to to it ;)

So finally here is: The EOT
The End Of da Tutorial.