Encrypting an AutoIt script with *CodeCrypter*

Step 1 of 5

- Open **MCFinclude.au3** in Scite and find Func **MCFCC_Init()**.
- Select any existing @macro or Function call that defines run-time content of array $CCkey, or add your own key definitions.
- **$CCkey array index = key ID**
- Write down which key ID(s) to use for this encryption!
- Close MCFinclude.au3

1 MCFinclude.au3 | 2 helloworld.au3

#include: **MCFinclude.au3**

```
258  EndFunc
259
260
261  Func _MCFCC_Init($type=0,$query=True,$useCNG=False)
262  ; NOTE: edit/add your keytype definitions here for $CCkey array entries 3-N
263  ; this UDF will itself be fixed-key encrypted
264
265      $CCkey[3]=@UserName      ; case-sensitive!
266      $CCkey[4]=_WinAPI_UniqueHardwareID($UHID_MB)     ; motherboard specs
267      $CCkey[5]=_WinAPI_UniqueHardwareID($UHID_CPU)    ; CPU specs
268      $CCkey[6]=DriveGetSerial("C:")
269      $CCkey[7]=@IPaddress1    ; NB ensure your IP is fixed (no DHCP)
270      ; ...
271      ; Add your own definitions here
272      ; (e.g., calls to your own functions (to be placed ABOVE _MCFCC_Init()!))
273
274      If $type="" Or $type=Default Then $type=1
275      $type=Number($type)
276      If $type<0 Or $type>=UBound($CCkey) Then $type=1     ; pwd query
277      If $cmdline[0]>0 Then $CCkey[1]=$cmdline[1]
278      If ($CCkey[$type]==$dummypwd Or $CCkey[$type]="" Or _
279          $CCkey[$type]=Null) And $query=True Then $CCkey[$type]= _
280          InputBox("Protected Application","Please Enter Password: ","","*M",250
281      $CCkeytype=$type
282
283      Switch $useCNG
284          Case True
285              For $cc=1 to UBound($CCkey)-1    ; skip entry 0 (already done)
286                  if IsPtr($CCkeyhandle[$cc]) Then __CryptoNG_BcryptDestroyKey($
287                  If $CCkey[$cc]<>Null Then $CCkeyhandle[$cc]=_CryptoNGgetHandle
288              Next
289
290          Case Else
291              _AES_Startup()
292      EndSwitch
293
294  EndFunc
295  ; this func def should be the last one within this region,
296  ;   as it is used as its end marker (so do NOT rename it)
297  #endregion Encryption2 (to be fixed-key encrypted)
```

Encrypting an AutoIt script with *CodeCrypter*

Step 2 of 5

In your target script, add the line:
#include “**MCFinclude.au3**”
(with path, if located elsewhere)

- Below all other #includes
- Above your own code

Save the script, open it in Scite, check for errors, and ensure that it works exactly as the original.

Target script: **HelloWorld.au3**

1 MCFinclude.au3 2 helloworld.au3

```
1  #NoTrayIcon
2  #include <GUIConstantsEx.au3>
3  #include <MsgBoxConstants.au3>
4
5
6  #include ".\MCFinclude.au3" ; <<< ADD THIS #INCLUDE, BELOW ALL OTHER INCLUDES
7
8  ; Show a simple message box.
9  MsgBox ( $MB_TOPMOST+$MB_ICONINFORMATION, "Test", "Hello, World!" )
10
11 ; call function to create a GUI
12 ExampleGUI()
13
14
15 Func ExampleGUI()    ; adapted from Help file, first example for GuiCreate
16
17     ; Create a GUI with a button
18     Local $hGUI = GUICreate( "This is an Example GUI" )
19     Local $myButton = GUICtrlCreateButton( "Press to Quit", 80, 200, 120, 30 )
20
21     GUISetState( @SW_SHOW, $hGUI )  ; Display the GUI.
22
23     While 1     ; Loop until the user exits.
24         Switch GUIGetMsg()
25             Case $GUI_EVENT_CLOSE, $myButton
26                 ExitLoop
27         EndSwitch
28     WEnd
29
30     GUIDelete( $hGUI )  ; Delete the GUI and all controls.
31
32 EndFunc   ;==>Example
```

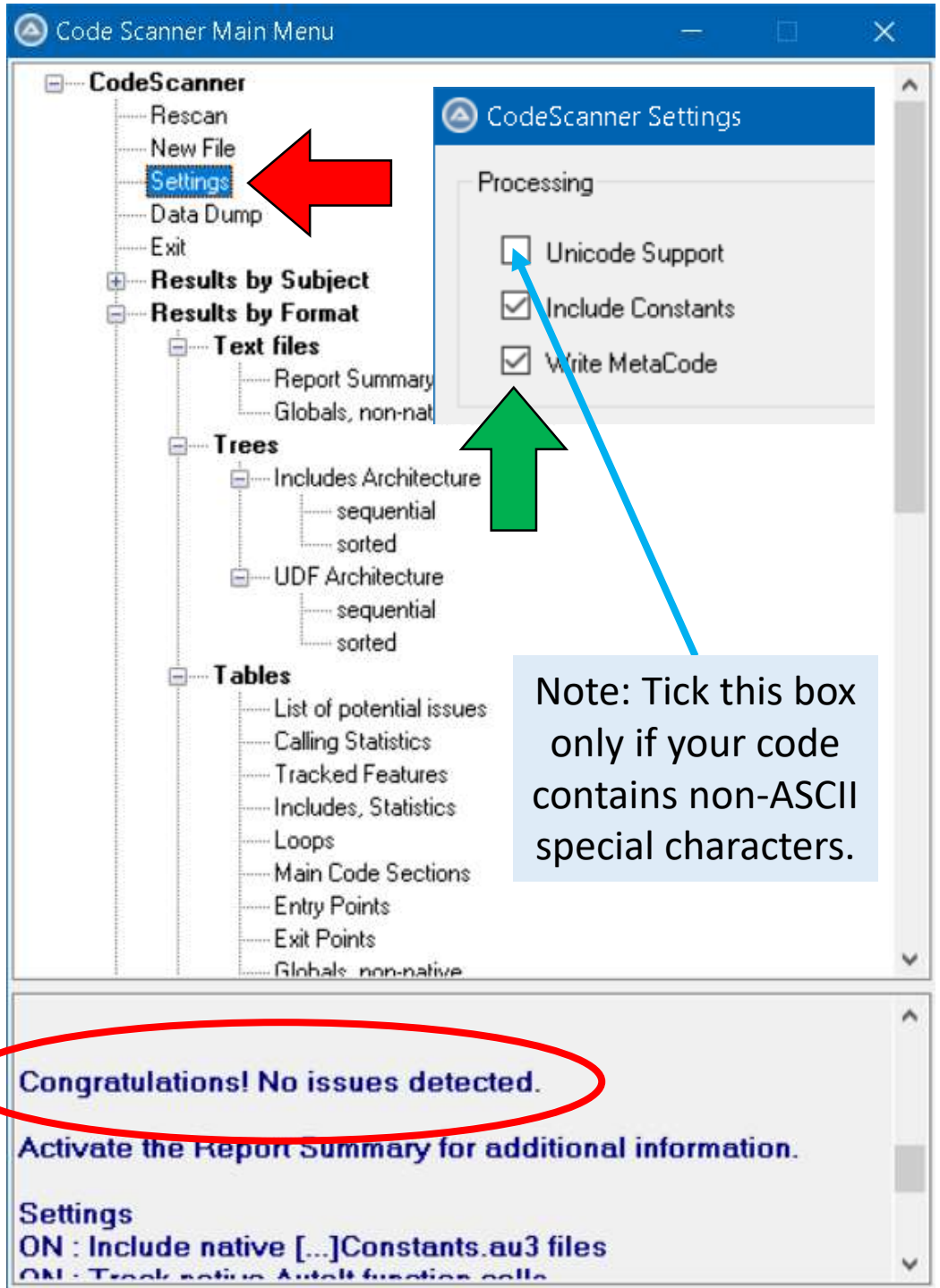
Encrypting an AutoIt script with *CodeCrypter*

Step 3 of 5

- Run **Code*Scanner*** on your script
  (this takes a long time).
- Check that no issues were detected
  (or fix these and re-run).
- **Subdirectory** :<name>.au3.CS_DATA is created

(if not, ensure that in CodeScanner's **Settings**, option **[Write MetaCode]** is enabled, and re-run.

- Close CodeScanner.

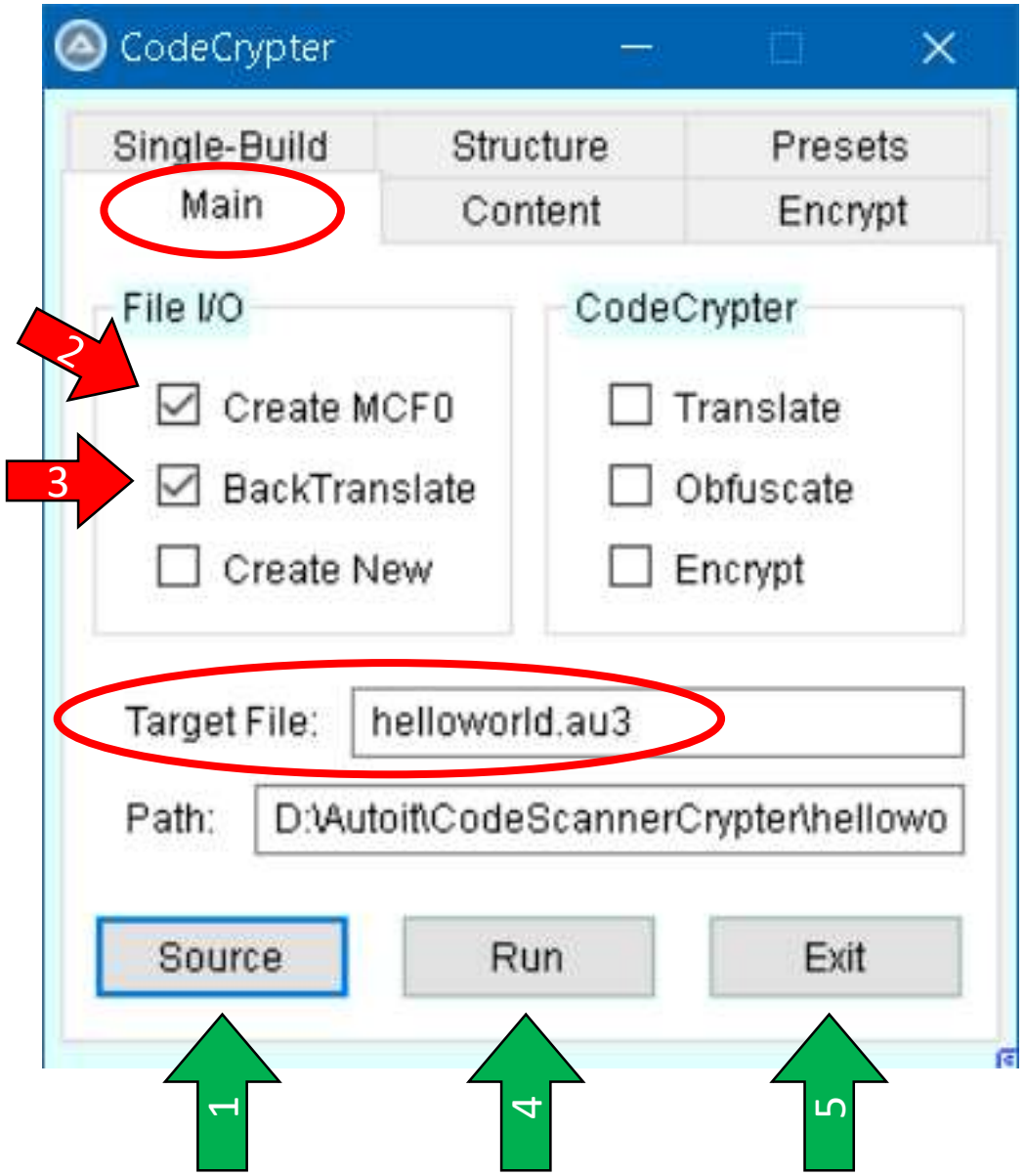


Encrypting an AutoIt script with *CodeCrypter*

Step 4 of 5

- Start **Code*Crypter***.
- Press ***[Source]*** to load your script.
  (the CS_DATA path is filled automatically).
- Tick Options **Create MCF0** and **BackTranslate**
- Press ***[Run]*** (this will take a long time), then Exit.

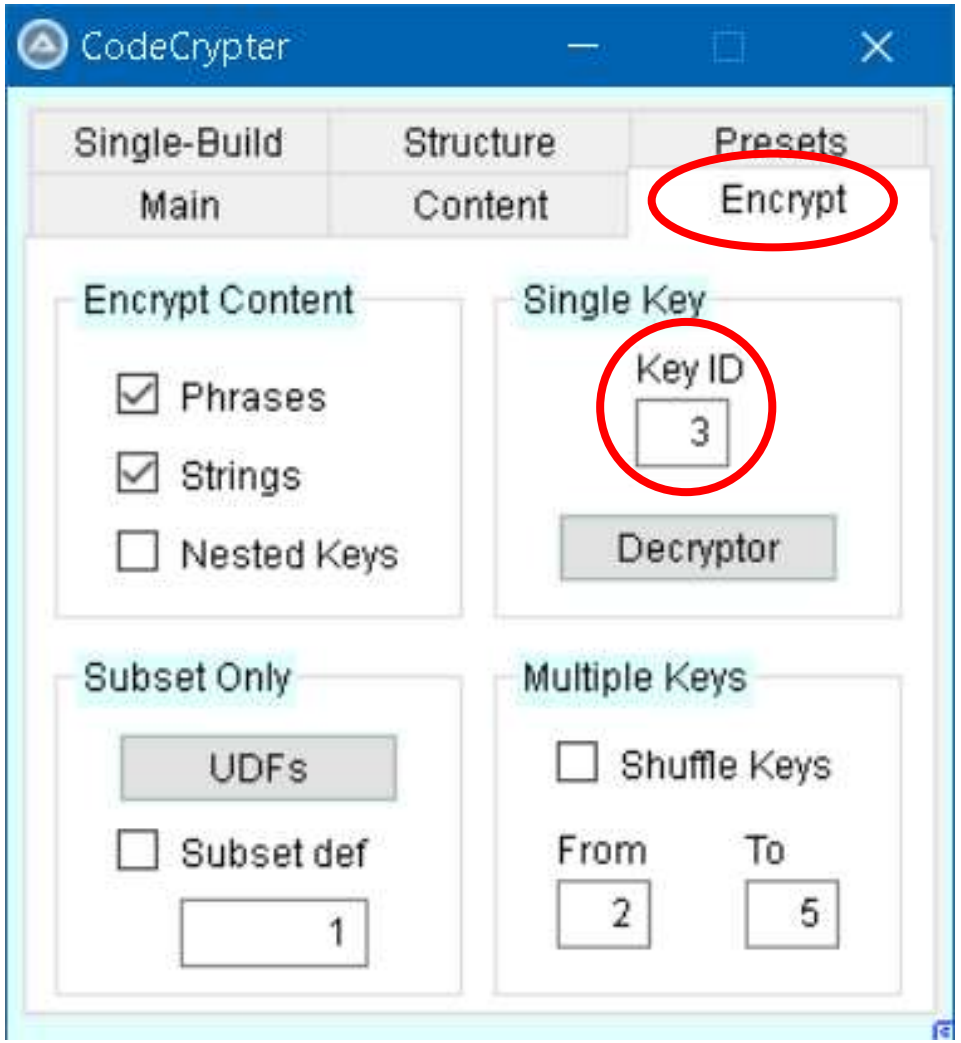
- A new test file **MCF0test.au3** has been created in your target file's home directory. Open it in Scite, check for errors, and do test-runs.
  Ensure that it works exactly as the original.



Encrypting an AutoIt script with *CodeCrypter*

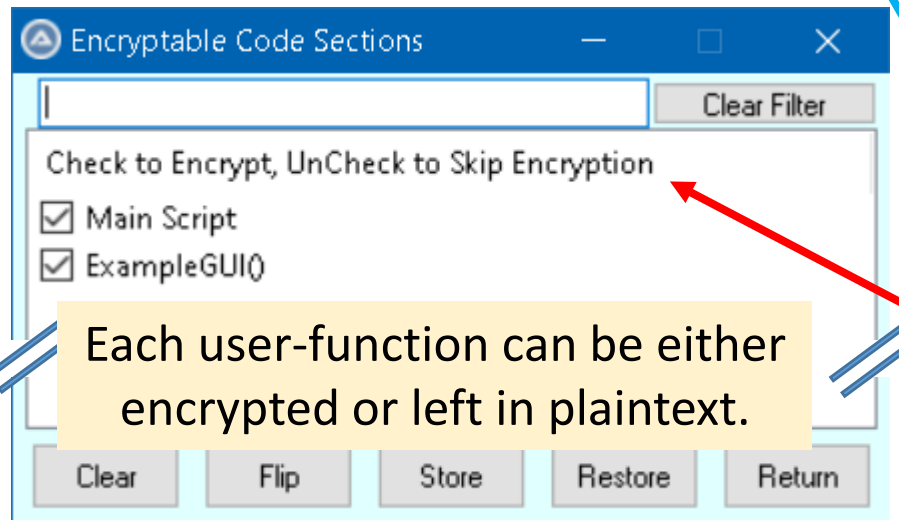
Step 5 of 5

- Restart **Code*Crypter***.
- Under Tab Encrypt, specify the **Key ID**; this is Step 1's index in array $CCkey in *MCFinclude.au3*
- Under Tab Main, disable option **[Create MCF0]** and enable option **Encrypt**; then Press ***[Run]***.
- A new, **encrypted** file **MCF0test.au3** is created in your target's home directory.
- Open it in Scite, check for errors, do test-runs.
  Ensure that it works exactly as the original.

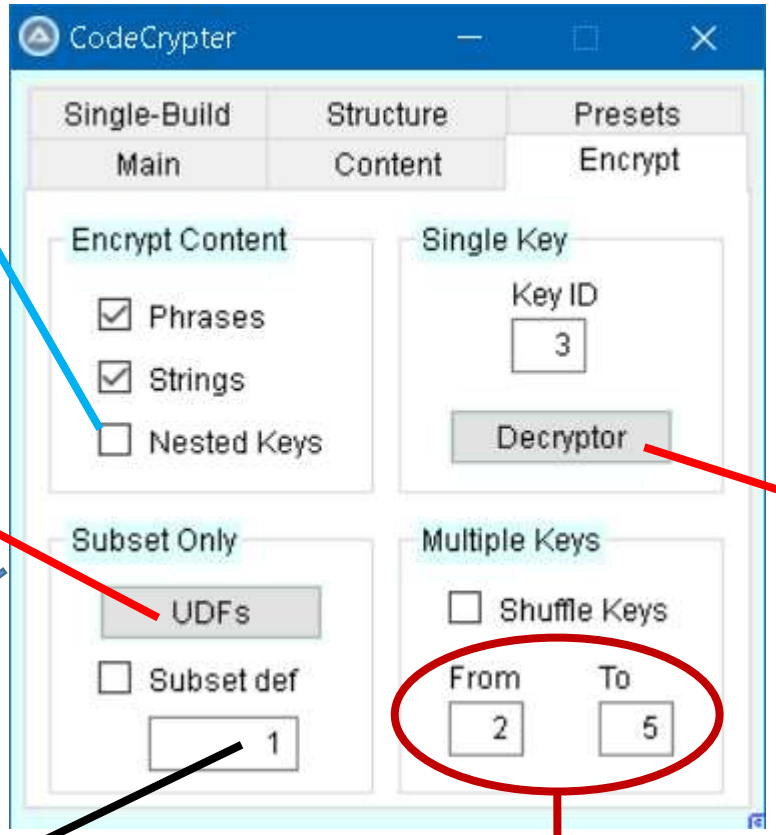


Encryption Options

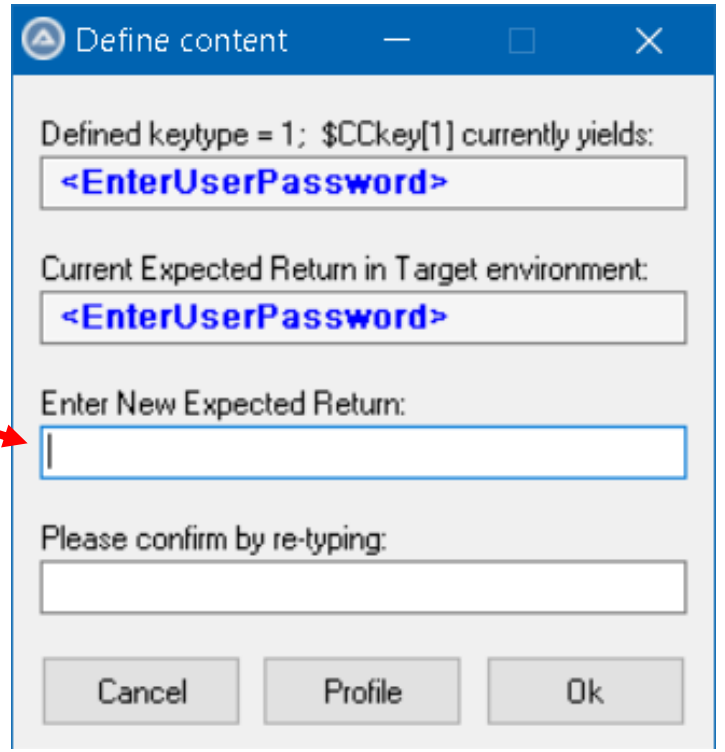
Two-pass encryption re-encrypts the encrypted code, to hide the key-ID (this causes massive slowdown.)



Each user-function can be either encrypted or left in plaintext.



Specify the expected return for any key ID in any environment.



Define a **S**ubset of lines to encrypt:

- [0 < **S** < 1] = random proportion
- [**S** > 1] = every **S**th line

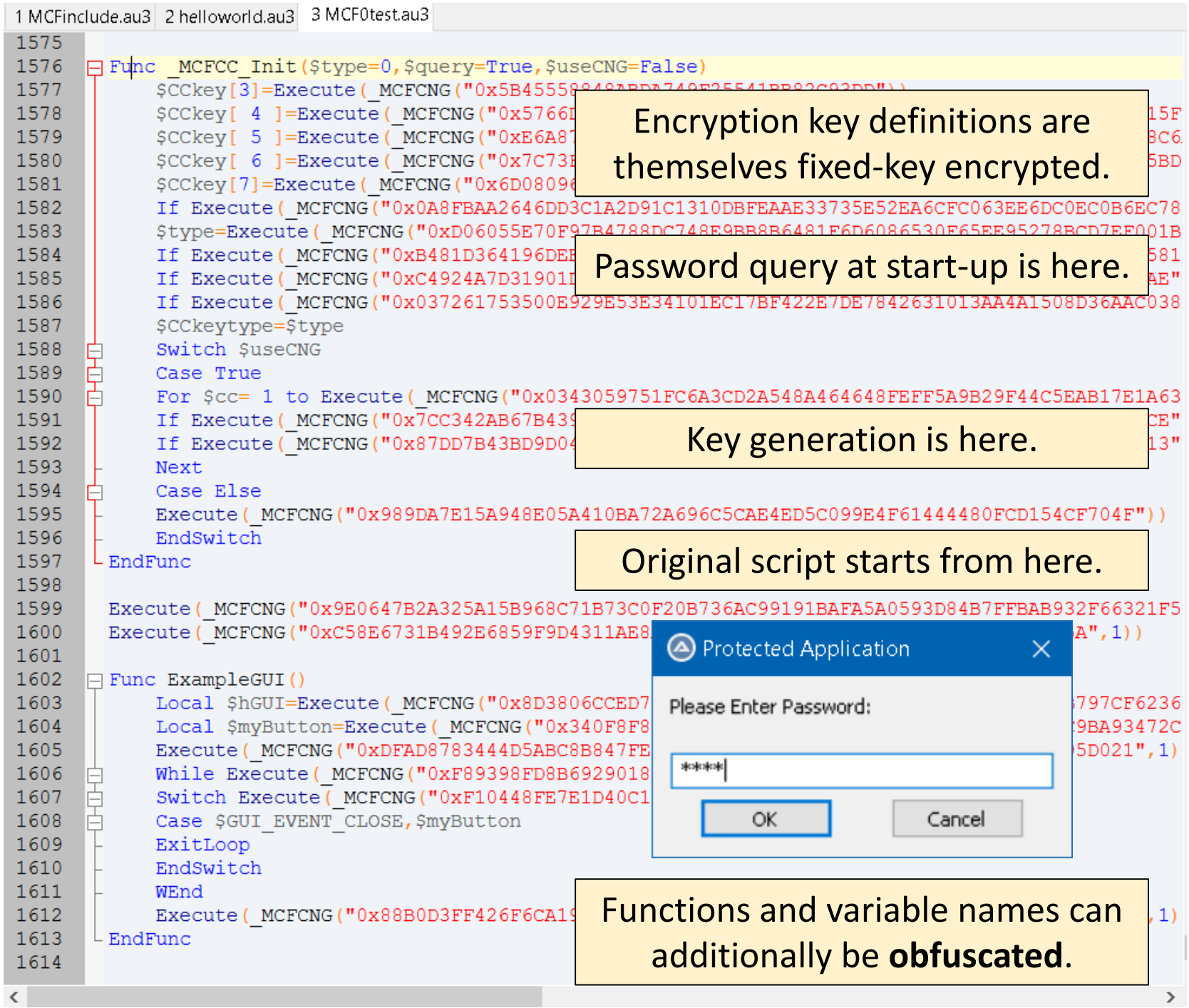
Randomly select from a range of encryption key IDs per line.

Final Result

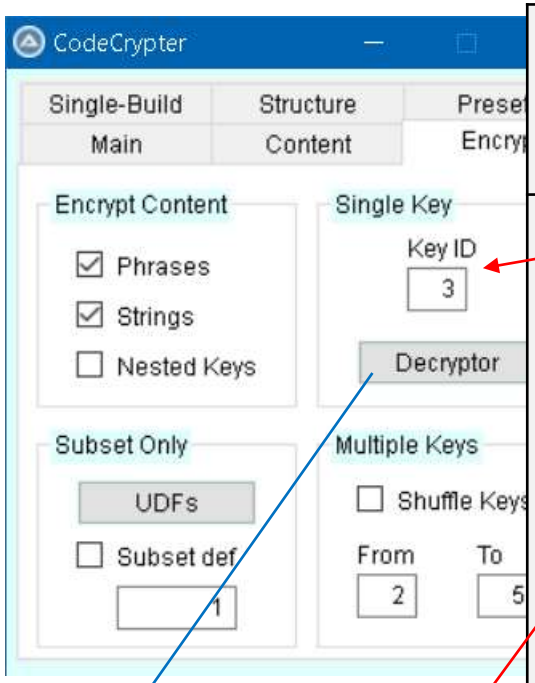
Script **HelloWorld.au3**, encrypted with user password: *test*.

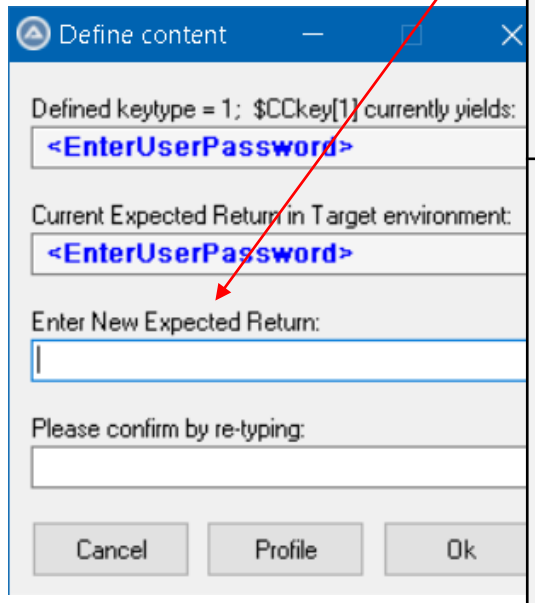
Features

- AES is practically unbreakable
- Each encryption pass is unique
- Fast decryption calls = minimal slow-down
- Optional targeted encryption of sensitive parts only
- **The decryption key is never stored** in the script, but **extracted from the run-time environment**
- The environment's expected ***response*** of key definitions can also be user-defined.



Suggested Key Types		User can be Trusted?	
		NO	YES
Environment controlled by Encrypter?	NO	CodeCrypter does **not** meet your needs	Key = Password query at start-up
	YES	1. #RequireAdmin 2. Key = admin-restricted environment specs returned at runtime	@username & _WinAPI_UniqueHardwareID(), DriveGetSerial(), …





Key		Key Definition	
		Empty	**Defined**
Decryption Key ("Expected Return")	**Empty**	***Fails***	ENcryption key = return from **key query** in current environment DEcryption key = return from **key query** at target start-up
	Defined	ENcryption key = typed **Expected Return** DEcryption key = 1st cmdline parameter or **user password**	ENcryption key = typed **Expected Return** DEcryption key = return from **key query** at target start-up

Target		Key Definition	
		Empty	**Defined**
Decryption Key ("Expected Return")	**Empty**	***Fails***	Target will run ***only*** in an environment that matches the encryption environment
	Defined	Target will run for whomever types the decryption key at target start-up	Target will run ***only*** when runtime key query response matches predefined **Expected Return**