

Encrypting an Autolt 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
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]==$dummyspw 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     Case Else
290         _AES_Startup()
291     EndSwitch
292
293 - EndFunc
294
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)
```

#include: MCFinclude.au3

Encrypting an Autolt script with *CodeCrypter*

Step 2 of 5

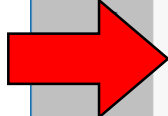
In your target script, add the line:

#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.



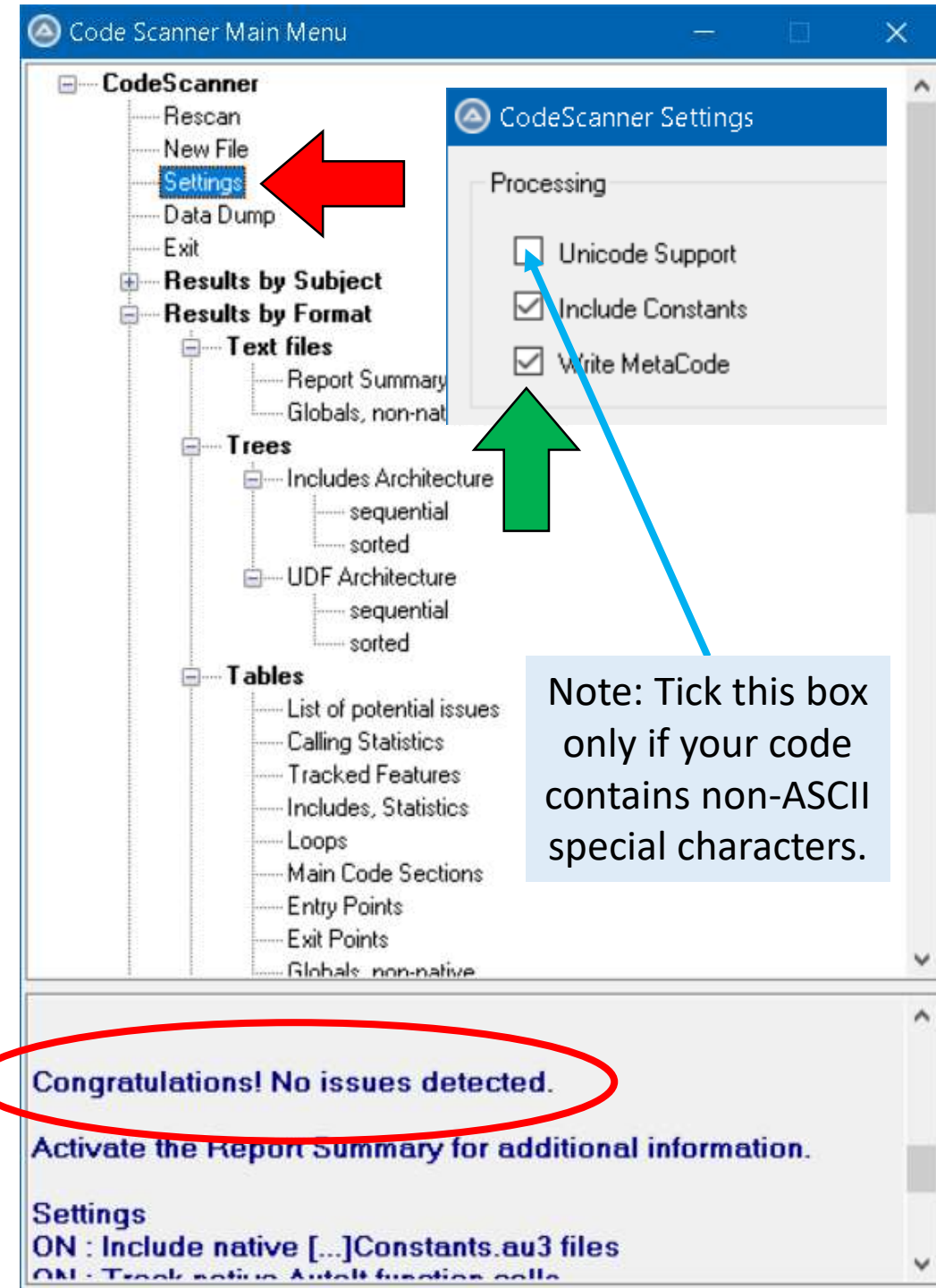
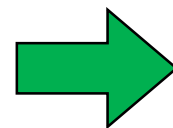
```
1 MCFinclude.au3 2 helloworld.au3
1 #NoTrayIcon
2 #include <GUIConstantsEx.au3>
3 #include <MsgBoxConstants.au3>
4
5 #include ".\\MCFinclude.au3" ; <<< ADD THIS #INCLUDE, BELOW ALL OTHER INCLUDES
6
7 ; Show a simple message box.
8 MsgBox ( $MB_TOPMOST+$MB_ICONINFORMATION, "Test", "Hello, World!" )
9
10 ; call function to create a GUI
11 ExampleGUI()
12
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
```

Target script: HelloWorld.au3

Encrypting an Autolt script with *CodeCrypter*

Step 3 of 5

- Run **CodeScanner** 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.



Note: Tick this box only if your code contains non-ASCII special characters.

Congratulations! No issues detected.

Activate the Report Summary for additional information.

Settings

ON : Include native [...]Constants.au3 files

ON : Track native Autolt function calls

Encrypting an Autolt script with *CodeCrypter*

Step 4 of 5

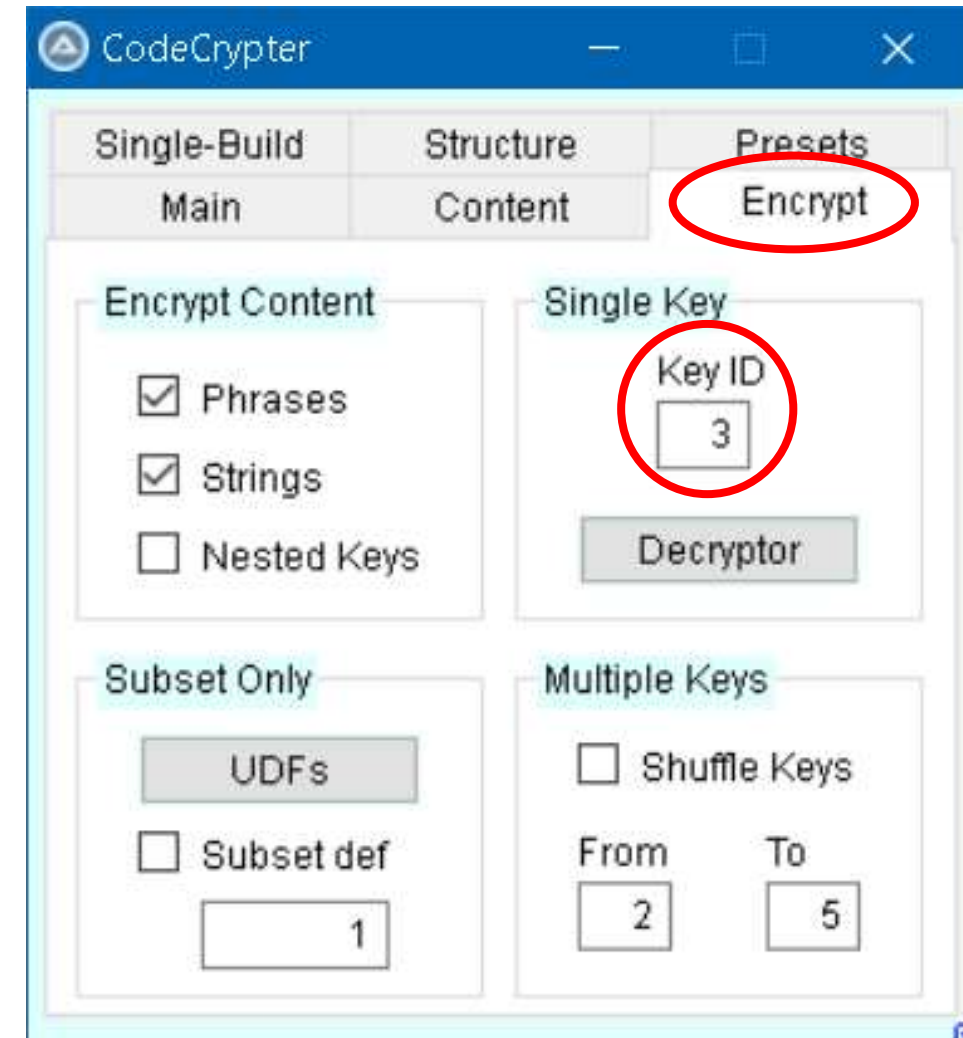
- Start **CodeCrypter**.
- Press **[Source]** to load your script.
(the CS_DATA path is filled automatically).
- Tick Options **Create MCF0** and **BackTranslate**
- Press **[Run]** (this will take a long time), then Exit.
- A new test file **MCF0test.au3** has been created in your target file's home directory. Open it in Scite, check for errors, and do test-runs.
Ensure that it works exactly as the original.



Encrypting an Autolt script with *CodeCrypter*

Step 5 of 5

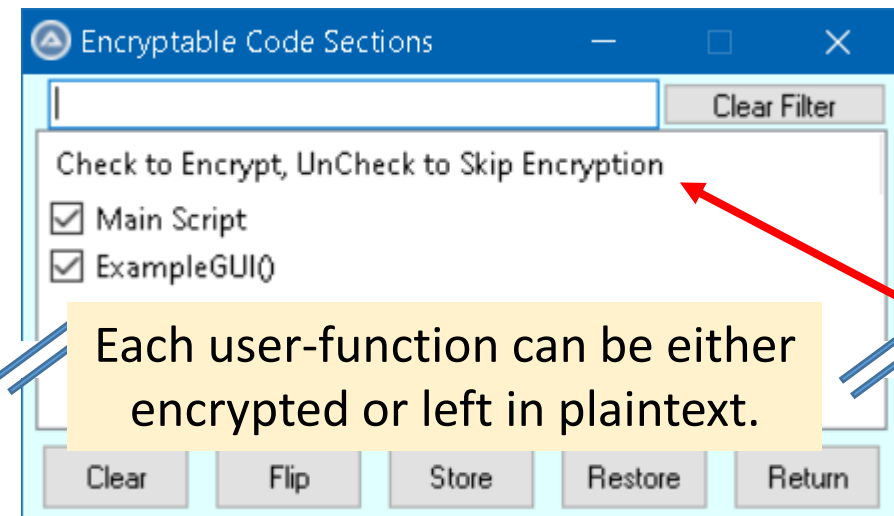
- Restart **CodeCrypter**.
- Under Tab Encrypt, specify the **Key ID**; this is Step 1's index in array **\$CCkey** in *MCFinclude.au3*
- Under Tab Main, disable option **[Create MCF0]** and enable option **Encrypt**; then Press **[Run]**.
- A new, **encrypted** file **MCF0test.au3** is created in your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



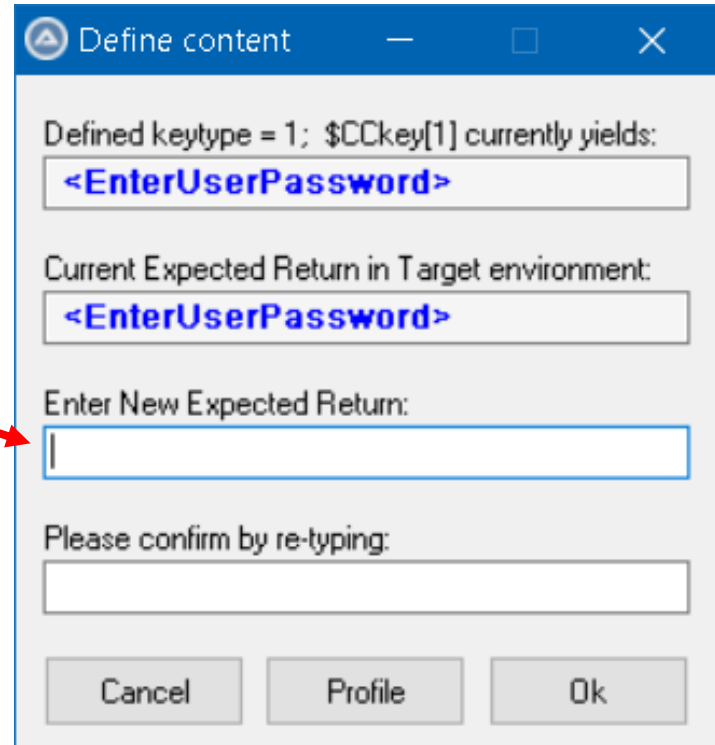
Encryption Options

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

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



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



Define a **Subset** 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.

```
1 MCFinclude.au3 2 helloworld.au3 3 MCF0test.au3
1575
1576 Func _MCFCC_Init($type=0,$query=True,$useCNG=False)
1577     $CCkey[3]=Execute(_MCFCNG("0x5B455582482BDA740E25541BB82C03DD"))
1578     $CCkey[4]=Execute(_MCFCNG("0x5766D"))
1579     $CCkey[5]=Execute(_MCFCNG("0xE6A87"))
1580     $CCkey[6]=Execute(_MCFCNG("0x7C73B"))
1581     $CCkey[7]=Execute(_MCFCNG("0x6D08096"))
1582     If Execute(_MCFCNG("0x0A8FBAA2646DD3C1A2D91C1310DBFEAAE33735E52EA6CFC063EE6DC0EC0B6EC78"))
1583     $type=Execute(_MCFCNG("0xD06055E70F97B4788DC748E9BB8B6481E6D6086530F65EE95278BCD7EE001B"))
1584     If Execute(_MCFCNG("0xB481D364196DE"))
1585     If Execute(_MCFCNG("0xC4924A7D31901"))
1586     If Execute(_MCFCNG("0x037261753500E929E53E34101EC17BF422E7DE7842631013AA4A1508D36AAC038"))
1587     $CCkeytype=$type
1588     Switch $useCNG
1589     Case True
1590     For $cc= 1 to Execute(_MCFCNG("0x0343059751FC6A3CD2A548A464648FEFF5A9B29F44C5EAB17E1A63"))
1591     If Execute(_MCFCNG("0x7CC342AB67B439"))
1592     If Execute(_MCFCNG("0x87DD7B43BD9D04"))
1593     Next
1594     Case Else
1595     Execute(_MCFCNG("0x989DA7E15A948E05A410BA72A696C5CAE4ED5C099E4F61444480FCD154CF704F"))
1596     EndSwitch
1597 EndFunc
1598
1599 Execute(_MCFCNG("0x9E0647B2A325A15B968C71B73C0F20B736AC99191BAFA5A0593D84B7FFBAB932F66321F5"))
1600 Execute(_MCFCNG("0xC58E6731B492E6859F9D4311AE8"))
1601
1602 Func ExampleGUI()
1603     Local $hGUI=Execute(_MCFCNG("0x8D3806CCED7"))
1604     Local $myButton=Execute(_MCFCNG("0x340F8F8"))
1605     Execute(_MCFCNG("0xDFAD8783444D5ABC8B847FE"))
1606     While Execute(_MCFCNG("0xF89398FD8B6929018"))
1607     Switch Execute(_MCFCNG("0xF10448FE7E1D40C1"))
1608     Case $GUI_EVENT_CLOSE,$myButton
1609     ExitLoop
1610     EndSwitch
1611     WEnd
1612     Execute(_MCFCNG("0x88B0D3FF426F6CA19"))
1613 EndFunc
1614
```

Encryption key definitions are themselves fixed-key encrypted.

Password query at start-up is here.

Key generation is here.

Original script starts from here.

Protected Application

Please Enter Password:

OK Cancel

Functions and variable names can additionally be obfuscated.

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(), ...

CodeCryper

Single-BuildMainStructureContentPresetsEncrypt

Encrypt Content

☒ Phrases☒ Strings☐ Nested Keys

Subset Only

UDFs

☐ Subset def

1

Single Key

Key ID

3

Decryptor

Multiple Keys

☐ Shuffle Keys

From

2

To

5

Define content

Defined keytype = 1; \$CCkey[1] currently yields:

<EnterUserPassword>

Current Expected Return in Target environment:

<EnterUserPassword>

Enter New Expected Return:

Please confirm by re-typing:

CancelProfileOk

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 = 1 st 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