### Development of software for automation of EDS

Alexander Koroban



### TABLE OF CONTENTS

### ALGORITHM OF THE

The full algorithm shown in two slides

**AUTO IT** A programming language

04

05

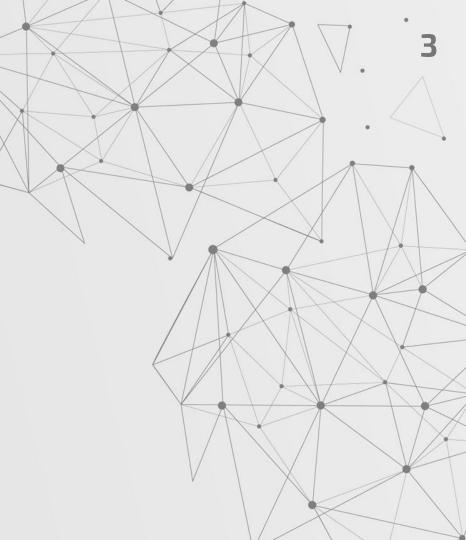
06

#### SOFTWARE DEVELOPMENT

Developing and testing all modules of the program

# **O1** INTRODUCTION

Functional units





### INTRODUCTION



### Main functional units of electronic reporting

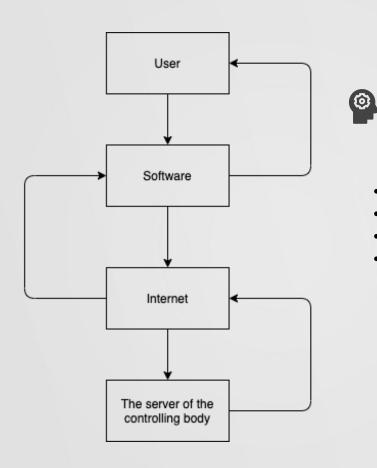
- Internet Connection
  - Software
    - Report
  - Digital Key



# STRUCTURAL SCHEME

Main Components

### **STRUCTURAL SCHEME**



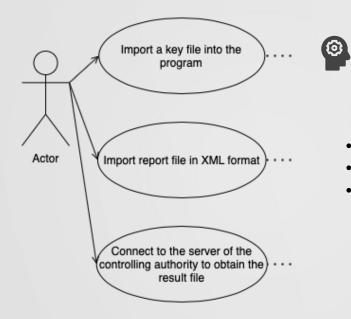
Main components of the structural scheme of the electronic reporting system

- User
- Software
- Internet
- The server of the controlling body

# PRECEDENT DIAGRAM

Main modules of the diagram

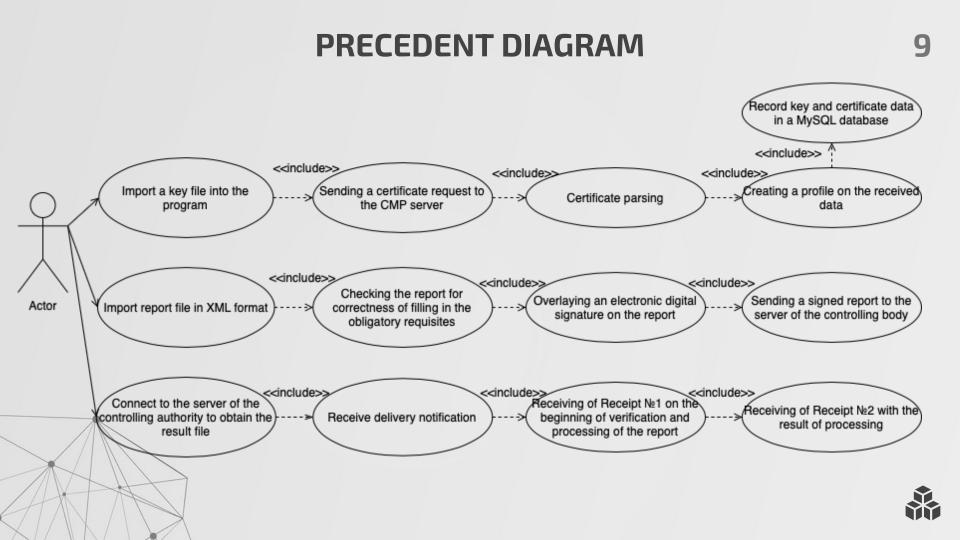
### **PRECEDENT DIAGRAM**



## Precedent diagram consists of three main modules

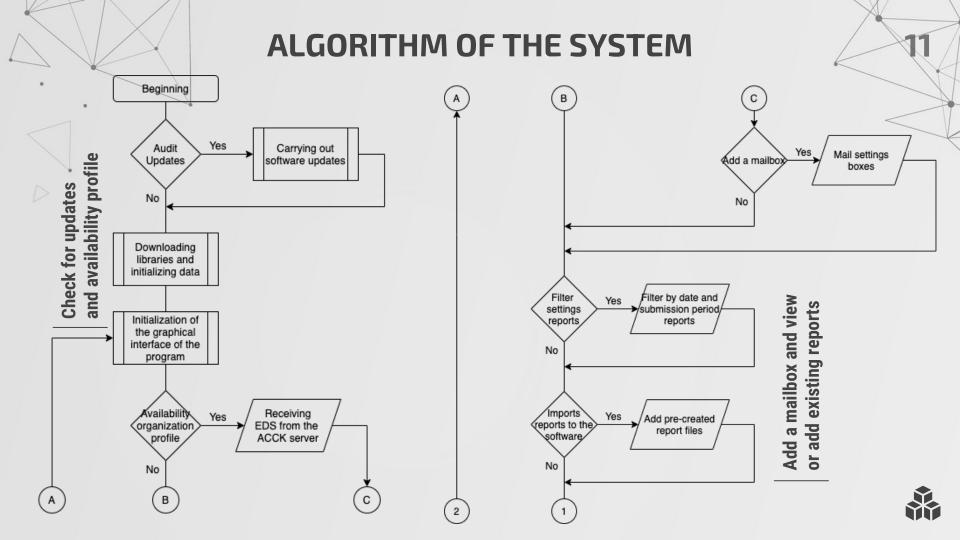
- Import a key file into the program
- Import report file in XML format
- Connect to the server of the controlling body

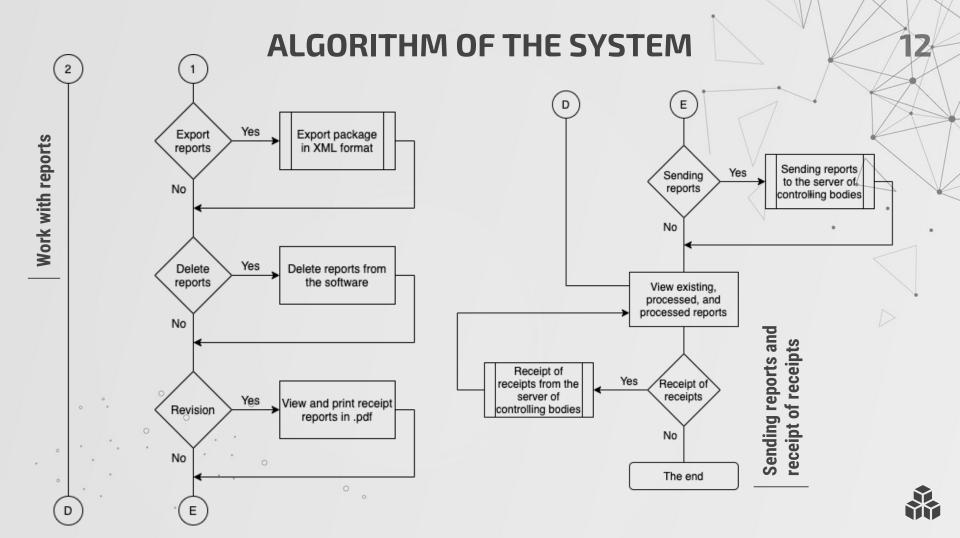
The full diagram is shown on the next slide.



# **O**4 ALGORITHM OF THE SYSTEM

The full algorithm shown in two slides







13

A programming language

### **AUTO IT**

#### C:\Program Files (x86)\Autolt3\Examples\GU|\Advanced\msgboxwizard.au3 - SciTE-Lite File Edit Search View Tools Options Language Buffers Help

**SciTE Editor** 

01     Select       02     Case \$iMSG = \$GUI_EVENT_CLOSE Or \$iMSG = \$idBTNEXIT       03     Exit       04     -       05     Case \$iMSG = \$idOptOK       06     \$idButton = \$idOptOK       07     GUICtrlSetState(\$idOptFirst, \$GUI_CHECKED)       08     GUICtrlSetState(\$idOptFirst, \$GUI_DISABLE)       09     GUICtrlSetState(\$q_idOptSecond, \$GUI_DISABLE)       10     GUICtrlSetState(\$q_idOptOkCancel       11     -       12     Case \$iMSG = \$q_idOptOkCancel       13     \$idButton = \$q_idOptOkCancel       14     GUICtrlSetState(\$dioptFirst, \$GUI_ENABLE)       15     GUICtrlSetState(\$dioptFirst, \$GUI_ENABLE)	🗅 💣		$   \underline{\mathscr{G}}    \& \boxtimes \underline{\mathfrak{m}} \boxtimes X   \bowtie \simeq   \mathbf{Q}, \mathbf{q} $	
<pre>81 82 GUICtrlCreateGroup("Timeout", 230, 450, 200, 50) 83 \$idTimeout = GUICtrlCreateInput(", 240, 470, 100, 20, 5ES_NUMBER) 84 85 GUICtrlSetIp(-1, "Optional Timeout in seconds. After the timeout has elapsed the message box will be automatically closed.") 85 GUICtrlSetIp(-1, "Show the MessageBox") 86 \$idBINPREVIEW = GUICtrlCreateButton("&amp;Preview", 10, 510, 100) 89 GUICtrlSetIp(-1, "Show the MessageBox") 80 GUICtrlSetIp(-1, "Goy the generated AutoIt code to the Clipboard") 91 \$idBINENT = GUICtrlCreateButton("&amp;Preview", 10, 510, 100) 92 GUICtrlSetIp(-1, "Goy the generated AutoIt code to the Clipboard") 93 \$idBUtton = \$idOptoK 94 GUICtrlSetIp(-1, "Quit the program") 94 \$idButton = \$idOptoK 95 GUISetState() ; will display an empty dialog box 97 ; Run the GUI until the dialog is closed 98 While 1 99 GuiCtrlSetSetGer() 90 GUICtrlSetState(); will display an empty dialog box 97 ; Run the GUI until the dialog is closed 99 While 1 90 GuiCtrlSetState(); furth_close Or \$iMSG = \$idBINEXIT 93 GUICtrlSetState(); furth_close Or \$iMSG = \$idBINEXIT 94 GUICtrlSetState(); furth_close Or \$iMSG = \$idBINEXIT 95 GUICtrlSetState(); furth_close Or \$iMSG = \$idBINEXIT 95 GUICtrlSetState(); furth_close(); furth_</pre>	79		GUICtrlSetState(-1, \$GUI_DISABLE)	^
<pre>GUICETICERETEGROUP "Timeout", 230, 450, 200, 501 \$idTimeout = GUICETICERETENPEUT", 240, 470, 100, 200, 565_NUMBER) GUICETISETIP(-1, "Optional Timeout in seconds. After the timeout has elapsed the message box will be automatically closed.") GUICETISETIP(-1, "Soft the MessageBox") fidBINERVIEW = GUICETICERETENDENC("&amp;Copy", 130, 510, 100) GUICETISETIP(-1, "Soft the MessageBox") fidBINERVIEW = GUICETICERETENDENC("&amp;Copy", 130, 510, 100) GUICETISETIP(-1, "Quit the MessageBox") fidBINEXII = GUICETICERETENDENC("&amp;Exit", 230, 510, 100) GUICETISETIP(-1, "Quit the program") fidBUTON = \$idOptOK GUISETState() ; will display an empty dialog box r Run the GUI until the dialog is closed While 1 GuiCeTISETIP(-1, "Copy the fit of the fidBUTON = \$idBUTON = \$idBUTON = \$idOptOK GUICETISETIP(-1, "Copy the fit of the fit</pre>	80		GUICtrlCreateGroup("", -99, -99, 1, 1) ;close group	
<pre>sidilmeout = GUICTLICreateInput(", 240, 470, 100, 20, 625 NUMBER) GUICtLISeTIp(-1, "Optional Timeout in seconds. After the timeout has elapsed the message box will be automatically closed.") GUICtLISeTIp(-1, "Optional Timeout ("&amp;Preview", 10, 510, 100) GUICtLISeTIp(-1, "Show the MessageBox") sidBINNOTF = GUICtLICreateButton("&amp;Preview", 10, 510, 100) GUICtLISeTIp(-1, "Copy the generated AutoLic code to the Clipboard") sidBINNIT = GUICtLICreateButton("&amp;Preview", 10, 510, 100) GUICtLISeTIp(-1, "Copy the generated AutoLic code to the Clipboard") sidBINNIT = GUICtLICreateButton("&amp;Lit", 230, 510, 100) GUICtLISeTIp(-1, "Quit the program") sidButton = \$idOptOK GUIStLISeTIG(-1, "Quit the program") sidButton = \$idOptOK scare \$iHSG = GUIGtHMag() Select GUIStLISeTIG(-) = \$iHSG = GUI_EVENT_CLOSE or \$iHSG = \$idBINEXIT Exit Case \$iHSG = GidOptOK GUICtLISetState(\$idOptFirst, \$GUI_CHECKED) GUICtLISetState(\$idOptFirst, \$GUI_CHE</pre>	81			
<pre>84 GUTCr1SetTip(-1, "Optional Timeout in seconds. After the Timeout has elapsed the message box will be automatically closed.") 65 GUTCr1SetTip(-1, "Spt he MessageBox") 65 SidBINPREVIEW = GUTCr1CreateButton("&amp;Preview", 10, 510, 100) 66 GUTCr1SetTip(-1, "Spt he MessageBox") 87 SidBINORY = GUTCr1CreateButton("&amp;Copy", 120, 510, 100) 98 GUTCr1SetTip(-1, "Quit the MessageBox") 91 SidBINT = GUTCr1CreateButton("&amp;Exit", 230, 510, 100) 67 GUTCr1SetTip(-1, "Quit the program") 93 SidButton = SidOptOK 96 GUTSetState() ; will display an empty dialog box 97 // **********************************</pre>	82		GUICtrlCreateGroup("Timeout", 230, 450, 200, 50)	
<pre>85 GUICtrlCreateGroup(", -99, -99, 1, 1) ;close group 86 SidBINPREVIEW = GUICtrlCreateButton("aPreview", 10, 510, 100) 87 GUICtrlSetTip(-1, "Show the MessageBox") 89 SidBINCOYF = GUICtrlCreateButton("sCoy", 120, 510, 100) 90 GUICtrlSetTip(-1, "Copy the generated AutoIc code to the Clipboard") 91 SidBINEXIF = GUICtrlCreateButton("sChirt,", 230, 510, 100) 92 GUICtrlSetTip(-1, "Quit the program") 93 SidButton = \$idOptOK 95 SidButton = \$idOptOK 96 GUISetState() ; vill display an empty dialog box 97 ; Run the GUI until the dialog is closed 98 While 1 90 SiHSG = GUIGetHsg() 91 Select 92 Case \$iHSG = \$GUI_EVENT_CLOSE Or \$iHSG = \$idBINEXIT 93 SidButton = \$idOptOK 94 SidButton = \$idOptOK 95 Case \$iHSG = \$GUI_EVENT_CLOSE Or \$iHSG = \$idBINEXIT 95 Exit 96 Case \$iHSG = \$idOptOK 97 GUICtrlSetState(\$idOptFirst, \$GUI_CHECKED) 98 GUICtrlSetState(\$idOptFirst, \$GUI_DISABLE) 99 GUICtrlSetState(\$idOptFirst, \$GUI_DISABLE) 99 GUICtrlSetState(\$idOptFirst, \$GUI_DISABLE) 90 GUICtrlSetState(\$idOptFirst, \$GUI_CHECKED) 91 GUICtrlSetState(\$idOptFirst, \$GUI_CHECKED) 92 GUICtrlSetState(\$idOptFirst, \$GUI_CHECKED) 93 GUICtrlSetState(\$idOptFirst, \$GUI_DISABLE) 94 GUICtrlSetState(\$idOptFirst, \$GUI_DISABLE) 95 GUICtrlSetState(\$idOptFirst, \$GUI_DISABLE) 96 GUICtrlSetState(\$idOptFirst, \$GUI_CHECKED) 97 GUICtrlSetState(\$idOptFirst, \$GUI_CHECKED) 98 GUICtrlSetState(\$idOptFirst, \$GUI_CHECKED) 99 GUICtrlSetState(\$idOptFirst, \$GUI_CHECKED) 90 GUICtrlSetState(\$idOptFirst, \$GUI_CHECKED) 91 GUICtrlSetState(\$idOptFirst, \$GUI_CHECKED) 91 GUICtrlSetState(\$idOptFirst, \$GUI_CHECKED) 92 GUICtrlSetState(\$idOptFirst, \$GUI_CHECKED) 93 GUICtrlSetState(\$idOptFirst, \$GUI_CHECKED) 94 GUICtrlSetState(\$idOptFirst, \$GUI_CHECKED) 95 GUICtrlSetState(\$idO</pre>	83		<pre>\$idTimeout = GUICtrlCreateInput("", 240, 470, 100, 20, \$ES_NUMBER)</pre>	
<pre>% % idBINPREVIEW = GUICtrlCreateButton("&amp;Preview", 10, \$10, 100) % GUICtrlSetTip(-1, "Show the MessageBox") % idBINCOPY = GUICtrlCreateButton("&amp;Copy", 120, 510, 100) GUICtrlSetTip(-1, "Quit the generated AutoIt code to the Clipboard") % idBINEXIT = GUICtrlCreateButton("&amp;Exit", 230, 510, 100) GUICtrlSetTip(-1, "Quit the program") % % idButton = % idOptOK % GUISetState() ; vill display an empty dialog box % ; Run the GUI until the dialog is closed % % while 1 % idBings = GUIGetMsg() % Select % Case % iMSG = % GUI_EVENT_CLOSE Or % iMSG = % idBINEXIT % Kit %  % GUICtrlSetTig(-1, "Guit the your close of % idBings = % idBINEXIT %  % % % % % % % % % % % % % % % % % %</pre>	84		GUICtrlSetTip(-1, "Optional Timeout in seconds. After the timeout has elapsed the message box will be automatically closed.")	
<pre>sidBINDEEVIEW = GUICERICReateButton("&amp;Preview", 10, 510, 100) GUICERISECTIP(-1, "Show the MessageBox") GUICERISECTIP(-1, "Gopy the generated AutoIt code to the Clipboard") GUICERISETTIP(-1, "Gopy the generated AutoIt code to the Clipboard") GUICERISETTIP(-1, "Quit the program") GUICERISETTIP(-1, "Quit the program") GUICERISETTIP(-1, "Quit the program") GUICERISETTIP(-1, "Quit the program") GUICERISETTIP(-1, "Quit the dialog is closed GUISESTate() ; vill display an empty dialog box GUISESTate() ; vill display an empty display and empty</pre>	85		GUICtrlCreateGroup("", -99, -99, 1, 1) ;close group	
<pre>88 GUICT:ISetTip(-1, "Show the MessageBox") 93 \$idBINCOFY = GUICT:ICreateButton("#Copy", 120, 510, 100) 94 \$idBINEXIT = GUICT:ICreateButton("#Exti", 230, 510, 100) 95 GUICT:ISetTip(-1, "Quit the program") 94 \$idButton = \$idOptOK 95 GUISetState() ; vill display an empty dialog box 96 GUISetState() ; vill display an empty dialog box 97  98 ; Run the GUI until the dialog is closed 99 While 1 90 \$iMSG = GUIGetMag() 91 Select 92 Case \$iMSG = \$GUI_EVENT_CLOSE Or \$iMSG = \$idBINEXIT 93 Exit 94 Case \$iMSG = \$GUI_EVENT_CLOSE Or \$iMSG = \$idBINEXIT 95 Case \$iMSG = \$idOptOK 96 GUICt:ISetState(\$idOptFirst, \$GUI_CHECKED) 97 GUICt:ISetState(\$idOptFirst, \$GUI_CHECKED) 98 GUICt:ISetState(\$idOptFirst, \$GUI_DISABLE) 99 GUICt:ISetState(\$idOptFirst, \$GUI_DISABLE) 10 GUICt:ISetState(\$idOptFirst, \$GUI_CHECKED) 11 - 12 Case \$iMSG = \$g_idOptOkCancel 13 \$idButton = \$g_idOptOkCancel 14 GUICt:ISetState(\$idOptFirst, \$GUI_CHECKED) 15 GUICt:ISetState(\$idOptFirst, \$GUI_CHECKED) 16 GUICt:ISetState(\$idOptFirst, \$GUI_CHECKED) 16 GUICt:ISetState(\$idOptFirst, \$GUI_CHECKED) 16 GUICt:ISetState(\$idOptFirst, \$GUI_CHECKED) 16 GUICt:ISetState(\$idOptFirst, \$GUI_CHECKED) 17 Case \$iMSG = \$g_idOptOkCancel 18 GUICt:ISetState(\$idOptFirst, \$GUI_CHECKED) 19 GUICt:ISetState(\$idOptFirst, \$GUI_CHECKED) 19 GUICt:ISetState(\$idOptFirst, \$GUI_CHECKED) 10 GUICt:ISetState(\$idOptFirst, \$GUI_CHECKED) 13 SidButton = \$g_idOptOkCancel 14 GUICt:ISetState(\$idOptFirst, \$GUI_CHECKED) 15 GUICt:ISetState(\$idOptFirst, \$GUI_CHECKED) 16 GUICt:ISetState(\$idOptFirst, \$GUI_CHECKED) 17 Case \$iMSG = \$g_idOptOkCancel 18 SidButton = \$g_idOptOkCancel 19 SidButton = \$g_idOptOkCanc</pre>	86			
<pre>89 \$ idBTROOPY = GUICCTLGreateButton("sCopy", 120, 510, 100) 90 GUICTLSetTlp(-1, "Copy the generated Autolt code to the Clipboard") 91 \$ idBTRENT = GUICCTLGreateButton("sKx1t", 230, 510, 100) 92 GUICTLSetTlp(-1, "Quit the program") 94 \$ idButton = \$ idOptOK 95 GUISetState() ; vill display an empty dialog box 97 ; Run the GUI until the dialog is closed 99 While 1 90 \$ iMSG = GUIGetMag() 91 Select 92 Case \$ iMSG = \$GUI_EVENT_CLOSE Or \$ iMSG = \$ idBTNEXIT 93 \$ 94 \$ idButton = \$ idOptOK 95 GUICTLSetState(\$ dioptFirst, \$GUI_CHECKED) 96 GUICTLSetState(\$ dioptFirst, \$GUI_CHECKED) 97 \$ GUICTLSetState(\$ dioptFirst, \$GUI_DISABLE) 98 \$ idButton = \$ idOptOK 99 \$ GUICTLSetState(\$ dioptFirst, \$GUI_DISABLE) 99 \$ GUICTLSetState(\$ dioptFirst, \$GUI_CHECKED) 99 \$ GUICTLSetState(\$ dioptFirst, \$GUI_DISABLE) 99 \$ GUICTLSetState(\$ dioptFirst, \$GUI_CHECKED) 10 \$ GUICTLSetState(\$ dioptFirst, \$GUI_CHECKED) 11 \$ Case \$ iMSG = \$ gidOptOKCancel 13 \$ idButton = \$ gidOptOKCancel 14 \$ GUICTLSetState(\$ dioptFirst, \$GUI_CHECKED) 15 \$ GUICTLSetState(\$ dioptFirst, \$GUI_CHECKED) 16 \$ GUICTLSetState(\$ dioptFirst, \$GUI_CHECKED) 17 \$ Case \$ iMSG = \$ gidOptOKCancel 18 \$ idButton = \$ gidOptOKCancel 19 \$ idButton = \$ gidOptOKCancel 19 \$ idButton = \$ gidOptOKCancel 10 \$ GUICTLSetState(\$ dioptFirst, \$GUI_CHECKED) 11 \$ Case \$ iMSG = \$ gidOptOKCancel 13 \$ idButton = \$ gidOptOKCancel 14 \$ GUICTLSetState(\$ dioptFirst, \$GUI_CHECKED) 15 \$ GUICTLSetState(\$ dioptFirst, \$GUI_CHECKED) 16 \$ GUICTLSetState(\$ dioptFirst, \$GUI_CHECKED) 16 \$ GUICTLSetState(\$ dioptFirst, \$GUI_CHECKED) 17 \$ \$ GUICTLSetState(\$ dioptFirst, \$GUI_CHECKED) 18 \$ GUICTLSetState(\$ dioptFirst, \$GUI_CHECKED) 19 \$ \$ \$ GUICTLSetState(\$ dioptFirst, \$GUI_CHECKED) 19 \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$</pre>	87		<pre>\$idBTNPREVIEW = GUICtrlCreateButton("&amp;Preview", 10, 510, 100)</pre>	10
<pre>90 GUICTISETIP(-1, "Copy the generated AutoIt code to the Clipboard") 91 \$idBitNENIT = GUICtrICertateButton("#Extr", 230, 510, 100) 92 GUICTISETIP(-1, "Quit the program") 93 \$idButton = \$idOptOK 95 GUISetState() ; vill display an empty dialog box 96 GUISetState() ; vill display an empty dialog box 97  98 ; Run the GUI until the dialog is closed 99 While 1 90 \$iHSG = GUIGetHag() 91 Select 92 Case \$iHSG = \$GUI_EVENT_CLOSE Or \$iHSG = \$idBINEXIT 93 Exit 94 \$idButton = \$idOptOK 95 GUICtrISetState(\$doptFirst, \$GUI_CHECKED) 96 GUICtrISetState(\$doptFirst, \$GUI_DISABLE) 97 GUICtrISetState(\$doptFirst, \$GUI_DISABLE) 98 GUICtrISetState(\$doptFirst, \$GUI_DISABLE) 99 GUICtrISetState(\$doptFirst, \$GUI_DISABLE) 10 GUICtrISetState(\$doptFirst, \$GUI_DISABLE) 11 F 12 Case \$iHSG = \$g_idOptOkCancel 13 \$idButton = \$g_idOptOkCancel 14 GUICtrISetState(\$doptFirst, \$GUI_CHECKED) 15 GUICtrISetState(\$doptFirst, \$GUI_CHECKED) 16 GUICtrISetState(\$doptFirst, \$GUI_CHECKED) 16 GUICtrISetState(\$doptFirst, \$GUI_DISABLE) 16 GUICtrISetState(\$doptFirst, \$GUI_DISABLE) 16 GUICtrISetState(\$doptFirst, \$GUI_CHECKED) 16 GUICtrISetState(\$doptFirst, \$GUI_CHECKED) 17 Case \$iHSG = \$g_idOptOkCancel 18 SidButton = \$g_idOptOkCancel 19 GUICtrISetState(\$doptFirst, \$GUI_CHECKED) 19 GUICtrISetState(\$GUICHECKED) 19 GUICtrISetState(\$GUICHECKED) 19 GUICtrISetState(\$GUICHECKED) 19 GUICtrISetState(\$GUICHECKED) 19 GUICtrISetState(\$GUICHECKED) 19 GUICtrISetState(\$GUICHECKED) 19 GUICtrISetState(\$G</pre>	88		GUICtrlSetTip(-1, "Show the MessageBox")	
<pre>91 \$\ddfTEXIT = GUICtrlCreateButton("&amp;Exit", 230, 510, 100) 92 GUICtrlSetTip(-1, "Quit the program") 93 94 \$\ddfTeXIT = GUICtrlCreateButton("&amp;Exit", 230, 510, 100) 95 96 GUISetState() ; vill display an empty dialog box 97 97 ; Run the GUI until the dialog is closed 99 \$\thile 1 90 \$\thild Select 99 \$\thild L 90 \$\thild Select 90 \$\thild Select 91 \$\thild Select 92 \$\thild Case \$\thild Select \$\thild GUICtrlSetState(\display LEVENT_CLOSE Or \$\thild Select 93 \$\thild Select 94 \$\thild Select 95 \$\thild Case \$\thild Select \$\thild GUICtrlSetState(\display LEVENT_CLOSE Or \$\thild Select 95 \$\thild Select 96 \$\thild Case \$\thild Select \$\thild GUICtrlSetState(\display LEVENT_CLOSE Or \$\thild Select 97 \$\thild Case \$\thild Select \$\thild GUICtrlSetState(\display LEVENT_CLOSE Or \$\thild Select 96 \$\thild GUICtrlSetState(\display LEVENT_CLOSE Or \$\thild Select 97 \$\thild Case \$\thild Select 98 \$\thild Select 99 \$\thild Select 90 \$\thild Case \$\thild Gelt GUICtrlSetState(\display LEVENT_CLOSE Or \$\thild Select 91 \$\thild Case \$\thild Select \$\thild GUICtrlSetState(\display LEVENT_CLOSE Or \$\thild Select 95 \$\thild GUICtrlSetState(\display LEVENT_CLOSE Or \$\thild Select 96 \$\thild GUICtrlSetState(\display LEVENT_CLOSE Or \$\thild Select 97 \$\thild Case \$\thild Select \$\thild GUICtrlSetState(\display LEVENT_CLOSE Or \$\thild Select 95 \$\thild GUICtrlSetState(\display LEVENT_CLOSE OR \$\thild GUICTLSES Select 95 \$\thild GUICtrlSetState(\display LEVENT_CLOSE OR \$\thild Select 95 \$\thild GUICtrlSe</pre>	89			
<pre>92 GUICtrlSetTip(-1, "Quit the program") 93 \$idButton = \$idOptOK 95 GUISetState() ; vill display an empty dialog box 97 ; Run the GUI until the dialog is closed 99 Hhile 1 00 \$iLMSG = GUIGetMag() 01 G \$Select 02 G Case \$iMSG = \$GUI_EVENT_CLOSE Or \$iMSG = \$idBINEXIT 03 Exit 04 Exit 05 Case \$iMSG = \$GUI_EVENT_CLOSE Or \$iMSG = \$idBINEXIT 05 G Case \$iMSG = \$GUI_EVENT_CLOSE Or \$iMSG = \$idBINEXIT 06 \$idButton = \$idOptOK 07 GUICtrlSetState(\$idOptFirst, \$GUI_CHECKED) 08 GUICtrlSetState(\$idOptFirst, \$GUI_DISABLE) 10 GUICtrlSetState(\$g_idOptSecond, \$GUI_DISABLE) 11 Case \$iMSG = \$g_idOptOKCancel 13 \$idButton = \$g_idOptOKCancel 14 GUICtrlSetState(\$dIoptFirst, \$GUI_CHECKED) 15 GUICtrlSetState(\$dIoptFirst, \$GUI_CHECKED) 16 GUICtrlSetState(\$dIoptFirst, \$GUI_CHECKED) 17 Case \$iMSG = \$g_idOptOKCancel 18 GUICtrlSetState(\$dIoptFirst, \$GUI_CHECKED) 19 GUICtrlSetState(\$dIoptFirst, \$GUI_CHECKED) 10 GUICtrlSetState(\$dIoptFirst, \$GUI_CHECKED) 13 Case \$iMSG = \$g_idOptOKCancel 14 GUICtrlSetState(\$dIoptFirst, \$GUI_CHECKED) 15 GUICtrlSetState(\$dIoptFirst, \$GUI_CHECKED) 16 GUICtrlSetState(\$dIoptFirst, \$GUI_CHECKED) 17 Case \$iMSG = \$g_idOptOKCancel 18 GUICtrlSetState(\$dIoptFirst, \$GUI_CHECKED) 19 GUICtrlSetState(\$dIoptFirst, \$GUI_CHECKED) 19 GUICtrlSetState(\$dIoptFirst, \$GUI_CHECKED) 19 GUICtrlSetState(\$dIoptFirst, \$GUI_CHECKED) 10 GUICtrlSetState(\$dIoptFirst, \$GUI_CHECKED) 13 Case \$iMSG = \$g_idOptOKCancel 14 GUICtrlSetState(\$dIoptFirst, \$GUI_CHECKED) 15 GUICtrlSetState(\$dIoptFirst, \$GUI_CHECKED) 16 GUICtrlSetState(\$dIoptSecond, \$GUI_ENABLE) 17 Case \$GUICtrlSetState(\$dIoptSecond, \$GUI_ENABLE) 18 Case \$GUICtrlSetState(\$dIoptSecond, \$GUI_ENABLE) 19 Case \$GUICtrlSetState(\$dIoptSecond, \$GUI_ENABLE) 19 Case \$GUICtrlSetState(\$dIoptSecond, \$GUI_ENABLE) 19 Case \$GUICtrlSetState(\$dIoptSecond, \$GUI_ENABLE) 19 Case \$GUICtrlSetState(\$GUICTRISE\$GUICTRI</pre>	90		GUICtrlSetTip(-1, "Copy the generated AutoIt code to the Clipboard")	
<pre>93 94 \$idButton = \$idOptOK 95 96 GUISetState() ; vill display an empty dialog box 97 98 ; Run the GUI until the dialog is closed 99 While 1 90 \$iHSG = GUIGetMag() 91 \$Select 92 Case \$iHSG = \$GUI_EVENT_CLOSE Or \$iHSG = \$idBTNEXIT 93 Exit 94 \$ 15 Case \$iHSG = \$idOptOK 94 GUICtrlSetState(\$idOptFirst, \$GUI_CHECKED) 95 GUICtrlSetState(\$idOptFirst, \$GUI_DISABLE) 96 GUICtrlSetState(\$idOptFirst, \$GUI_DISABLE) 97 GUICtrlSetState(\$idOptOKnet] 18 \$ 18 GUICtrlSetState(\$idOptFirst, \$GUI_CHECKED) 19 GUICtrlSetState(\$idOptFirst, \$GUI_CHECKED) 10 GUICtrlSetState(\$idOptFirst, \$GUI_DISABLE) 11 - 12 Case \$iHSG = \$g_idOptOKCancel 13 \$ 14 GUICtrlSetState(\$idOptFirst, \$GUI_CHECKED) 15 GUICtrlSetState(\$idOptFirst, \$GUI_CHECKED) 16 GUICtrlSetState(\$idOptFirst, \$GUI_CHECKED) 17 Case \$iHSG = \$g_idOptOKCancel 18 GUICtrlSetState(\$idOptFirst, \$GUI_CHECKED) 19 GUICtrlSetState(\$idOptFirst, \$GUI_CHECKED) 10 GUICtrlSetState(\$idOptFirst, \$GUI_CHECKED) 11 - 12 Case \$iHSG = \$g_idOptOKCancel 13 \$ 14 GUICtrlSetState(\$idOptFirst, \$GUI_CHECKED) 15 GUICtrlSetState(\$idOptFirst, \$GUI_CHECKED) 16 GUICtrlSetState(\$idOptSecond, \$GUI_ENABLE) 17 \$ 18 GUICtrlSetState(\$idOptSecond, \$GUI_ENABLE) 19 \$ 19 \$ 10 \$ 10 \$ 10 \$ 10 \$ 10 \$ 10 \$ 10 \$ 10</pre>	91			
<pre>94 \$idButton = \$idOptOK 95 GUISetState() ; vill display an empty dialog box 97 ; Run the GUI until the dialog is closed 99 While 1 00 \$iMSG = GUIGetMag() 01 GuiSetState() 99 Case 5iMSG = \$GUI_EVENT_CLOSE Or \$iMSG = \$idBINEXIT 02 Case 5iMSG = \$GUI_EVENT_CLOSE Or \$iMSG = \$idBINEXIT 03 Case 5iMSG = \$GUI_EVENT_CLOSE Or \$iMSG = \$idBINEXIT 04 - Case 5iMSG = \$idOptOK 05 Case 5iMSG = \$idOptOK 06 \$idButton = \$idOptOK 07 GUICtrlSetState(\$idOptFirst, \$GUI_CHECKED) 08 GUICtrlSetState(\$idOptFirst, \$GUI_DISABLE) 09 GUICtrlSetState(\$g_idOptFirst, \$GUI_DISABLE) 10 GUICtrlSetState(\$idOptFirst, \$GUI_DISABLE) 11 - Case 5iMSG = \$g_idOptOkCancel 13 \$idButton = \$g_idOptOkCancel 14 GUICtrlSetState(\$idOptFirst, \$GUI_CHECKED) 16 GUICtrlSetState(\$idOptSecond, \$GUI_ENABLE) 16 GUICtrlSetState(\$idOptSecond, \$GUI_ENABLE) 16 GUICtrlSetState(\$idOptSecond, \$GUI_ENABLE) 16 GUICtrlSetState(\$idOptSecond, \$GUI_ENABLE) 16 GUICtrlSetState(\$idOptSecond, \$GUI_ENABLE) 16 GUICtrlSetState(\$idOptSecond, \$GUI_ENABLE) 16 GUICtrlSetState(\$idOptSecond, \$GUI_ENABLE) 17 GUICtrlSetState(\$idOptSecond, \$GUI_ENABLE) 18 GUICtrlSetState(\$idOptSecond, \$GUI_ENABLE) 19 GUICtrlSetState(\$idOptSecond, \$GUI_ENABLE) 10 GUICtrlSetState(\$idOptSecond, \$GUI_ENABLE) 11 - Case \$iMSG = \$g_idOptOkCancel 12 - Case \$iMSG = \$g_idOptOkCancel 13 - \$idButton = \$g_idOptSecond, \$GUI_ENABLE} 14 - \$GUICtrlSetState(\$idOptSecond, \$GUI_ENABLE) 15 - \$GUICtrlSetState(\$idOptSecond, \$GUI_ENABLE) 16 - \$GUICtrlSetState(\$idOptSecond, \$GUI_ENABLE) 17 - \$GUICtrlSetState(\$idOptSecond, \$GUI_ENABLE) 18 - \$GUICtrlSetState(\$idOptSecond, \$GUI_ENABLE) 19 - \$GUICtrlSetState(\$GU</pre>	92		GUICtrlSetTip(-1, "Quit the program")	
<pre>95 96 97 96 96 97 97 98 97 97 98 97 98 96 96 96 96 96 97 98 97 97 98 97 98 97 98 97 98 97 98 97 98 97 98 97 98 97 98 97 98 97 98 97 98 98 98 98 98 99 98 98 99 9 98 98 98 9</pre>	93			
96       GUISetState() ; vill display an empty dialog box         97       ; Run the GUI until the dialog is closed         99       ; Run the GUI until the dialog is closed         99       While 1         00       \$LMSG = GUIGetMag()         01       Select         02       Case \$HMSG = \$GUI_EVENT_CLOSE Or \$HMSG = \$idBTNEXIT         03       Exit         04       -         05       Case \$HMSG = \$idOptOK         06       \$idButton = \$idOptOK         07       GUICtrlSetState(\$idOptFirst, \$GUI_CHECKED)         08       GUICtrlSetState(\$idOptFirst, \$GUI_DISABLE)         09       GUICtrlSetState(\$idOptFirst, \$GUI_DISABLE)         09       GUICtrlSetState(\$idOptFirst, \$GUI_DISABLE)         10       GUICtrlSetState(\$idOptFirst, \$GUI_DISABLE)         11       -         12       Case \$HMSG = \$g_idOptOkCancel         13       \$idButton = \$g_idOptOkCancel         14       GUICtrlSetState(\$idOptFirst, \$GUI_CHECKED)         15       GUICtrlSetState(\$dioptFirst, \$GUI_ENABLE)         16       GUICtrlSetState(\$dioptSecond, \$GUI_ENABLE)	94		<pre>\$idButton = \$idOptOK</pre>	
<pre>97</pre>	95			
<pre>98  ; Run the GUI until the dialog is closed 99  \$\thild 1 54HSG = GUIGetHsg() 301  \$\text{Select} 302  Case 54HSG = \$GUI_EVENT_CLOSE Or \$\text{iHSG} = \$\text{idBTNEXIT} 303  Exit 304  Exit 305  Case 54HSG = \$\text{idOptOK} 406  \$\text{idOptChist} = \$\text{idOptFirst}, \$GUI_CHECKED} 306  GUICtrlSetState(\$\text{idOptFirst}, \$GUI_CHECKED} 307  GUICtrlSetState(\$\text{idOptFirst}, \$GUI_DISABLE}) 309  GUICtrlSetState(\$\text{idOptFirst}, \$GUI_DISABLE}) 309  GUICtrlSetState(\$\text{idOptChird}, \$GUI_DISABLE}) 310  GUICtrlSetState(\$\text{idOptOKCancel} 311  \$\text{idButton} = \$\text{idOptOKCancel} 313  \$\text{idButton} = \$\text{idOptOKCancel} 314  GUICtrlSetState(\$\text{idOptDirst}, \$GUI_CHECKED}) 315  GUICtrlSetState(\$\text{idOptDirst}, \$GUI_ENABLE}) 326  GUICtrlSetState(\$\text{idOptDirst}, \$GUI_ENABLE}) 337  \$\text{idButton} = \$\text{idOptDirst}, \$GUI_CHECKED}) 348  GUICtrlSetState(\$\text{idOptDirst}, \$GUI_ENABLE}) 358  GUICtrlSetState(\$\text{idOptDirst}, \$GUI_ENABLE}) 369  GUICtrlSetState(\$\text{idOptDirst}, \$GUI_ENABLE}) 360  GUICtrlSetState(\$\text{idOptDirst}, \$GUI_ENABLE}) 360  GUICtrlSetState(\$\text{idOptDirst}, \$GUI_ENABLE}) 360  GUICtrlSetState(\$\text{idOptDirst}, \$GUI_ENABLE}) 360  GUICtrlSetState(\$\text{idOptDirst}, \$GUI_ENABLE}) 360  GUICtrlSetState(\$\text{idOptDirst}, \$GUI_ENABLE}) 360  GUICtrlSetState(\$\text{idOptSecond}, \$GUI_ENABLE}) 360  GUICtrlSetState(\$\text{idOptSecond}, \$GUI_ENABLE}) 360  GUICtrlSetState(\$\text{idOptSecond}, \$GUI_ENABLE}) 360  GUICtrlSetState(\$\text{idOptSecond}, \$GUI_ENABLE}) 360  GUICtrlSetState(\$\text{idOptSecond}, \$GUI_ENABLE}) 370  GUICtrlSetState(\$\text{idOptSecond}, \$GUI_ENABLE}) 370  GUICtrlSetState(\$\text{idOptSecond}, \$GUI_ENABLE}) 370  GUICtrlSetState(\$\text{idOptSecond}, \$GUI_ENABLE}) 370  GUICtrlSetState(\$\text{idOptSecond}, \$\text{idOptSecond}, \$\text{idOptSecond}, \$\text{idOptSecond}, \$\text{idOptSecond}, \$\text{idOptSecond}, \$\text{idOptSecond}, \$\text{idOptSecond}, \$\text{idOptSecond}, \$\text{idOptSecond}, \$\text{idOptSecond}</pre>	96		<pre>GUISetState() ; will display an empty dialog box</pre>	
<pre>99</pre>	97			
<pre>61MSG = GUIGetHag() 61</pre>	98		; Run the GUI until the dialog is closed	
01 Select Case \$iMSG = \$GUI_EVENT_CLOSE Or \$iMSG = \$idBINEXIT Case \$iMSG = \$idOptOK 64		白	While 1	
02       Case \$iMSG = \$GUI_EVENT_CLOSE Or \$iMSG = \$idBINEXIT         03       Exit         04       Exit         05       Case \$iMSG = \$idOptOK         06       \$idButton = \$idOptOK         07       GUICtrl3etState(\$idOptFirst, \$GUI_CHECKED)         08       GUICtrl3etState(\$idOptFirst, \$GUI_DISABLE)         09       GUICtrl3etState(\$idOptThird, \$GUI_DISABLE)         10       GUICtrl3etState(\$idOptChecond, \$GUI_DISABLE)         11       -         12       Case \$iMSG = \$g_idOptOkCancel         13       \$idButton = \$g_idOptOkCancel         14       GUICtrl3etState(\$idOptFirst, \$GUI_CHECKED)         15       GUICtrl3etState(\$idOptFirst, \$GUI_CHECKED)         16       GUICtrl3etState(\$idOptFirst, \$GUI_ENABLE)	100			
03     Exit       04     Exit       05     Case \$iMSG = \$idOptOK       \$idDutton = \$idOptOK       06     \$idDutton = \$idOptEx       07     GUICtrlSetState(\$idOptFirst, \$GUI_CHECKED)       08     GUICtrlSetState(\$idOptFirst, \$GUI_CHECKED)       09     GUICtrlSetState(\$idOptFirst, \$GUI_DISABLE)       10     GUICtrlSetState(\$idOptCancel       11     -       12     Case \$iMSG = \$g_idOptOkCancel       13     \$idButton = \$g_idOptOkCancel       14     GUICtrlSetState(\$idOptFirst, \$GUI_ENABLE)       15     GUICtrlSetState(\$idOptSecond, \$GUI_ENABLE)       16     GUICtrlSetState(\$idOptSecond, \$GUI_ENABLE)	101	白		
<pre>04 - 05 - Case \$iMSG = \$idOptOK 64 \$idOutton = \$idOptOK 07 \$GUICt13etState(\$idOptFirst, \$GUI_CHECKED) 08 \$GUICt13etState(\$id_ptFirst, \$GUI_DISABLE) 09 \$GUICt13etState(\$g_idOptOkSecond, \$GUI_DISABLE) 10 \$GUICt13etState(\$g_idOptThird, \$GUI_DISABLE) 11 - 12 Case \$iMSG = \$g_idOptOkCancel 13 \$\$idOutton = \$g_idOptOkCancel 14 \$GUICt13etState(\$idOptFirst, \$GUI_CHECKED) 15 \$GUICt13etState(\$idOptFirst, \$GUI_CHECKED) 16 \$GUICt13etState(\$g_idOptSecond, \$GUI_ENABLE) 16 \$GUICt13etState(\$g_idOptSecond, \$GUI_ENABLE) 16 \$GUICt13etState(\$g_idOptSecond, \$GUI_ENABLE) 17 \$\$GUICt13etState(\$g_idOptSecond, \$GUI_ENABLE)\$</pre>	102	白	Case \$iMSG = \$GUI_EVENT_CLOSE Or \$iMSG = \$idBTNEXIT	
05       Case \$iMSG = \$id0pt0K         06       \$id0utton = \$id0pt0K         07       GUICt19etState(\$id0ptFirst, \$GUI_CHECKED)         08       GUICt19etState(\$id0ptFirst, \$GUI_ENABLE)         09       GUICt19etState(\$id0ptFirst, \$GUI_DISABLE)         10       GUICt19etState(\$id0ptFirst, \$GUI_DISABLE)         11       -         12       Case \$iMSG = \$g_id0pt0kCancel         13       \$id0utton = \$g_id0pt0kCancel         14       GUICt19etState(\$id0ptFirst, \$GUI_ENABLE)         15       GUICt19etState(\$id0ptFirst, \$GUI_ENABLE)         16       GUICt19etState(\$g_id0ptSecond, \$GUI_ENABLE)	103		Exit	
06       \$idButton = \$idOpt0K         07       GUICtrlSetState (\$idOptFirst, \$GUI_CHECKED)         08       GUICtrlSetState (\$idOptFirst, \$GUI_ENABLE)         09       GUICtrlSetState (\$idOptFirst, \$GUI_DISABLE)         10       GUICtrlSetState (\$idOptFirst, \$GUI_DISABLE)         11       -         12       Case \$iMSG = \$g_idOpt0kCancel         13       \$idButton = \$g_idOpt0kCancel         14       GUICtrlSetState (\$idOptFirst, \$GUI_ENABLE)         15       GUICtrlSetState (\$idOptFirst, \$GUI_ENABLE)         16       GUICtrlSetState (\$idOptSecond, \$GUI_ENABLE)	104	-		
07     GUICtrlSetState(\$idOptFirst, \$GUI_CHECKED)       08     GUICtrlSetState(\$idOptFirst, \$GUI_INABLE)       09     GUICtrlSetState(\$gidOptDiscend, \$GUI_DISABLE)       10     GUICtrlSetState(\$gidOptThird, \$GUI_DISABLE)       11     GUICtrlSetState(\$gidOptChancel       12     Case \$iMSG = \$g_idOptOkCancel       13     \$idButton = \$g_idOptOkCancel       14     GUICtrlSetState(\$idOptFirst, \$GUI_ENABLE)       15     GUICtrlSetState(\$idOptFirst, \$GUI_ENABLE)       16     GUICtrlSetState(\$idOptFirst, \$GUI_ENABLE)	105	Þ		
08     GUICtrlSetState(\$id0ptFirst, \$GUI_ENABLE)       09     GUICtrlSetState(\$g_id0ptSecond, \$GUI_DISABLE)       10     GUICtrlSetState(\$g_id0ptSecond, \$GUI_DISABLE)       11     -       12     Case \$iMSG = \$g_id0pt0kCancel       13     \$idButton = \$g_id0pt0kCancel       14     GUICtrlSetState(\$di0ptFirst, \$GUI_ENABLE)       15     GUICtrlSetState(\$di0ptFirst, \$GUI_ENABLE)       16     GUICtrlSetState(\$dj0ptSecond, \$GUI_ENABLE)	106		<pre>\$idButton = \$idOptOK</pre>	
09 6UICtrlSetState(\$g_id0ptSecond, \$GUI_DISABLE) 6UICtrlSetState(\$g_id0ptThird, \$GUI_DISABLE) 12 12 13 14 14 14 15 15 15 16 16 17 17 17 17 17 17 17 17 17 17	107			
10     GUICtrlSetState(\$g_idOptThird, \$GUI_DISABLE)       11     -       12     Case \$iMSG = \$g_idOptOkCancel       13     \$idButton = \$g_idOptOkCancel       14     GUICtrlSetState(\$idOptFirst, \$GUI_CHECKED)       15     GUICtrlSetState(\$idOptFirst, \$GUI_ENABLE)       16     GUICtrlSetState(\$idOptFirst, \$GUI_ENABLE)	108			
11     -       12     Case \$iMSG = \$g_id0pt0kCance1       13     \$iddutton = \$g_id0pt0kCance1       14     GUICt13etState(id0ptFirst, \$GUI_CHECKED)       15     GUICt13etState(\$id0ptFirst, \$GUI_ENABLE)       16     GUICt13etState(\$g_id0ptSecond, \$GUI_ENABLE)	109			
12       Case \$iMSG = \$g_idOptOkCancel         13       \$idButton = \$g_idOptOkCancel         14       GUICtrl3etState(\$idOptFirst, \$GUI_CHECKED)         15       GUICtrl3etState(\$idOptFirst, \$GUI_ENABLE)         16       GUICtrl3etState(\$idOptSecond, \$GUI_ENABLE)	110		GUICtrlSetState(\$g_id0ptThird, \$GUI_DISABLE)	
13     \$idButton - \$g_idOptOkCancel       14     GUICtrlSetState (\$idOptFirst, \$GUI_CHECKED)       15     GUICtrlSetState (\$idOptFirst, \$GUI_ENABLE)       16     GUICtrlSetState (\$g_idOptSecond, \$GUI_ENABLE)	111	-		
14     GUICtrlSetState(\$id0ptFirst, \$GUI_CHECKED)       15     GUICtrlSetState(\$id0ptFirst, \$GUI_ENABLE)       16     GUICtrlSetState(\$g_id0ptSecond, \$GUI_ENABLE)	112	P		
15     GUICtrlSetState(\$idOptFirst, \$GUI_INABLE)       16     GUICtrlSetState(\$g_idOptSecond, \$GUI_ENABLE)	113			
16 GUICtr1SetState(\$g_idOptSecond, \$GUI_ENABLE)	114			
	115			
	116		GUICtrlSetState(\$g_idOptSecond, \$GUI_ENABLE)	~
				6
			-	-

0

#### Autolt and SciTE

- Autolt is a language for automating tasks in Windows. Includes a GUI framework that allows to use simple scripts with typical graphical components in scripts.
- Autolt is distributed with a SciTEbased IDE. The compiler, helper utilities, and reference materials are fully integrated, making the editor the default environment for developers using Autolt.

# **O6** Software Development

Developing and testing all modules of the program



### **CRYPTOGRAPHIC MODULE**

Signed Data Format



Alice's data Alice signs data and sends it to Bob

### ASN.1 AND PKCS # 7

ASN.1 is a standard that describes the structure of messages received using cryptography. The standard p7s file has an ASN.1 structure. 16

Standard PKCS # 7 / p7b file structure:

- version
- digestAlgorithms
- contentInfo
- certificates
- signerInfos

Alice's message (in CMS)

### **GRAPHICAL INTERFACE**

(ତ)

3594310738 ФО КО	особа): РОБАНЬ ОЛЕКС	АНДР СЕРГІЙС	)BI - 🗓 🖏 💐
Рільтр для звітів: портовані/надіслан	i pozrarou Mio	яця 🔻 до	15.09.2020 -
віти та квитанції:		нцн • до	15.03.2020
9			<u>a</u>
Код форми	Період	Стан № зві	ту Дата/час

Main parts of interface

- Work with the profile of the organization (individual).
- Window filter with imported and transmitted reports.
- Functionality for working with reports.

### **SOFTWARE TESTING**

### "Notification of receipt of reports"

 $\odot$ 

After the successful transfer of reports to the server of regulatory authorities, the attached report changed color from white to yellow, and an arrow appeared when you click on it to open a tree of messages and received receipts for viewing and saving.

	АНЬ ОЛЕКС	АНДР С	ENINOBI	
ільтр для звітів:				
портовані/надіслані про	Міс:	яця 🔻	до 1	5.09.2020 -
віти та квитанції:				
9 🖪 🍳				
бод форми	Період	Стан	№ звіту	Дата/час
F1301802	09.2020	1	1	15.09.2020 15:55:52
≫ <mark>Квитанція №1</mark>	09.2020			15.09.2020 15:49:32
Повідомлення	09.2020			15.09.2020 15:49:32
F1401802	09.2020	1	1	15.09.2020 15:49:32

**Testing receipt of reports** 

## THANKS

Does anyone have any questions?

CREDITS: This presentation was created by Alexander Koroban.