



Z SIGNALS

Managed Strategies

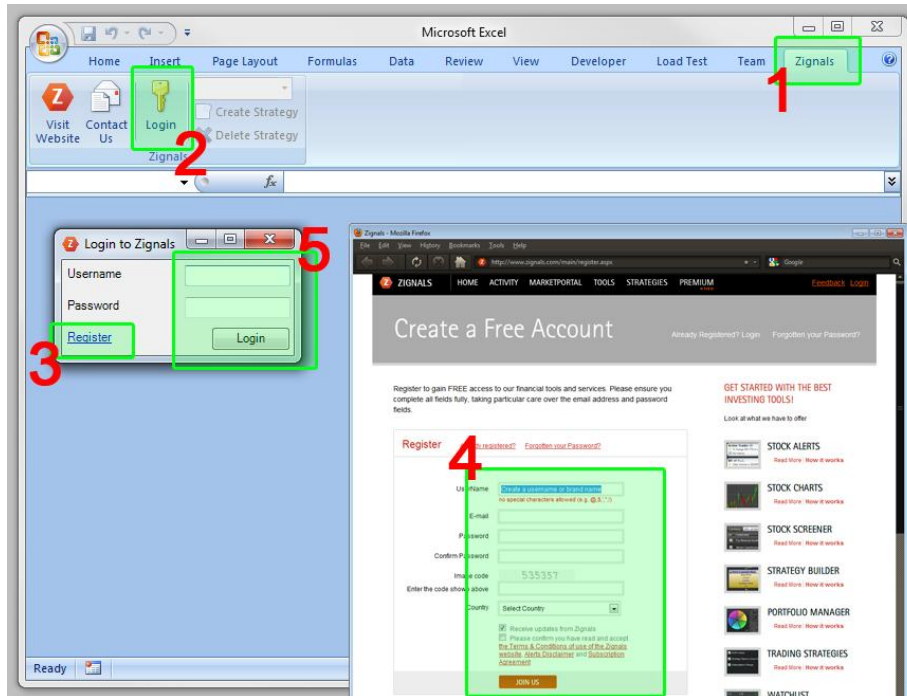
External Signals

Excel API

1. [How to setup Signals ribbon and Tab](#)
2. [How to update the workbook/VBA macro code](#)
3. [How to use the API to send Trading Signals and Updates to Signals.](#)
4. [Possible errors](#)
5. [Troubleshooting](#)

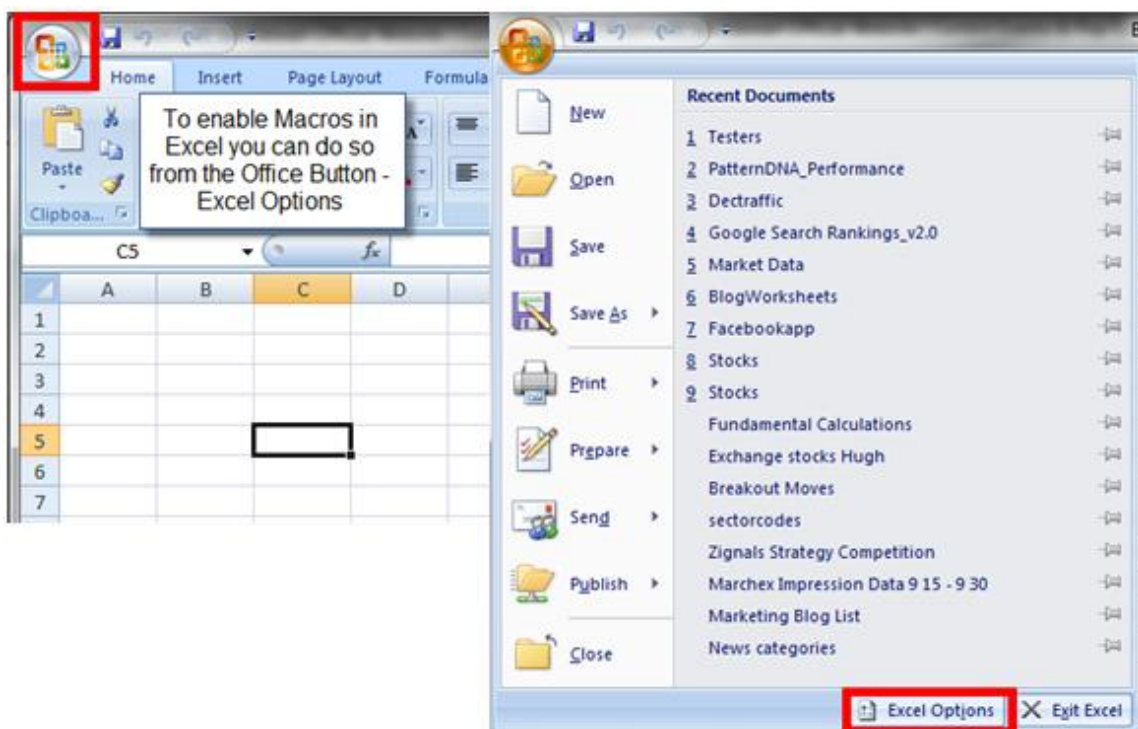
1. How to setup Signals ribbon and Tab:

1. Download the Signals Add-In file: from [here](#). Extract the file using [Quick Zip 5.1](#) or other file extractor.
2. Ensure Excel is closed (or close it)
3. Install the Signals Add-in by running **Setup.exe** from the directory you extracted the ZIP files too - typically this is yourname > My Documents > Downloads > XLSx.x.x.x
4. Open Excel
5. A new 'Signals' ribbon will appear as the last tab in the Excel menu
 - If the Signals tab doesn't appear in the excel install **o2007pia**. This file is available in the *Office2007PIARedist* directory of your extracted files - double click the file name to install it.
6. In the Signals tab of Excel, log in to your Signals account. This will open a new window where you can enter your Username and Password. A link is provided to Register with Signals if required.
7. Try to:
 - Log in
 - Create strategy
 - Strategy appears in the dropdown box
 - Delete strategy
 - Log out



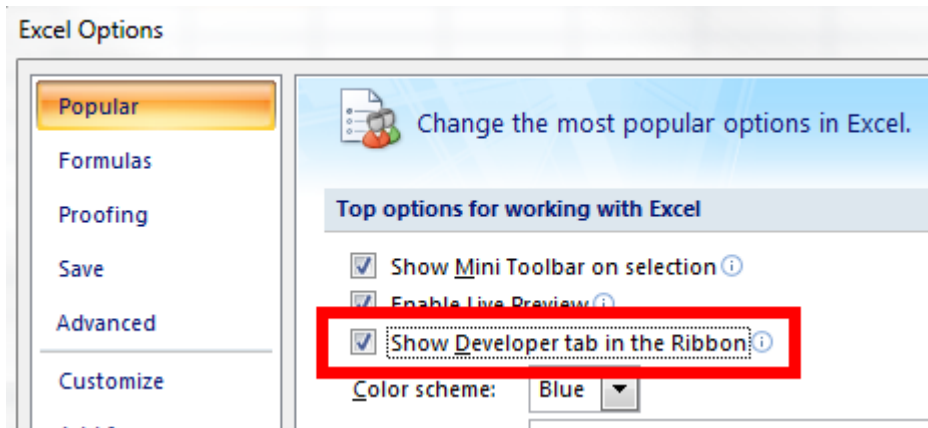
2. How to update the workbook/VBA macro code:

1. Open Excel (default spreadsheet)
2. From the Office Button in the top-left of the page, select Excel Options



[Excel1.png 580 x 373; lb: 348 x 223][Alt: Configuring Excel Options]

3. To enable Macros you need to check the box to "Show Developer tab in the Ribbon"

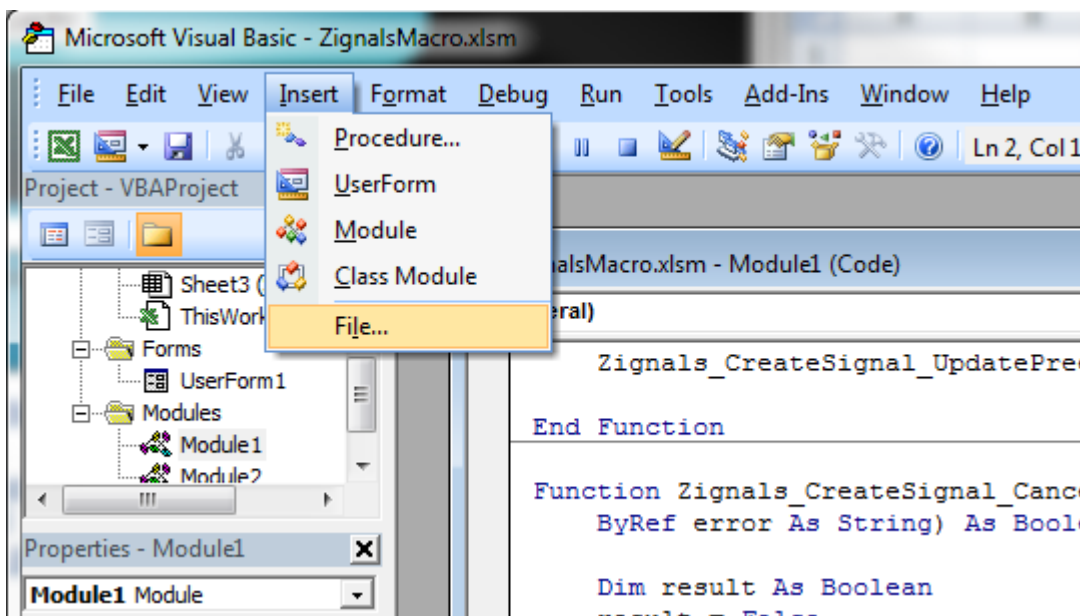


[Excel2.png 466 x 212; lb: 348 x 223][Alt: Select the Developer Tab for Excel; this will allow you create Macros]

4. From the Developer Tab; Select "Visual Basic"

5. From the Insert menu, insert a Module

6. Again, from the Insert menu in the Visual Basic Editor, select "File"

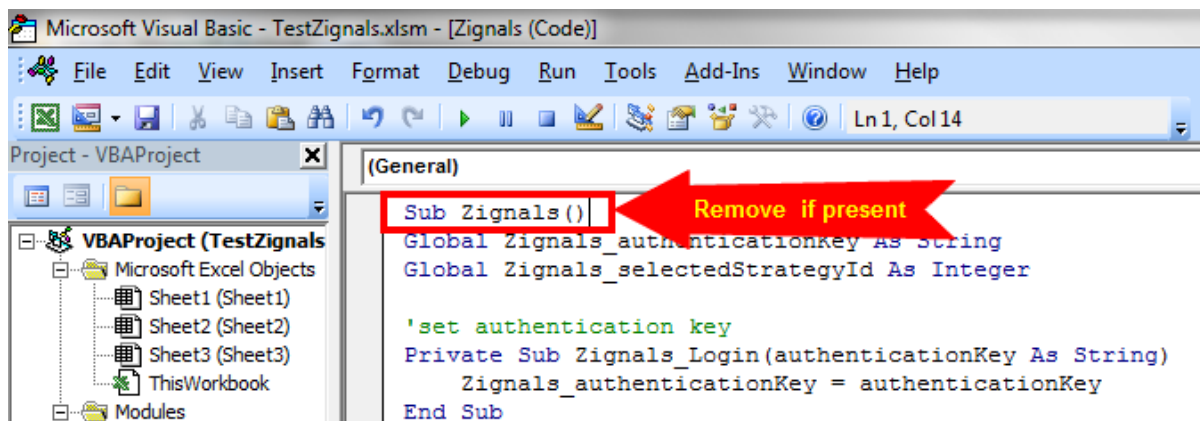


[Excel5.png: 539 x 305; lb: 431 x 244][Alt: Inserting the Signals visual basic code into your newly created Signals Macro]

7. Source the Signals VBA code, **signalsVbaModule.vba**, from the ZIP-downloaded directory - you may need to select to view All Files as the default view is *.txt.

This should add the VBA code to your open module.

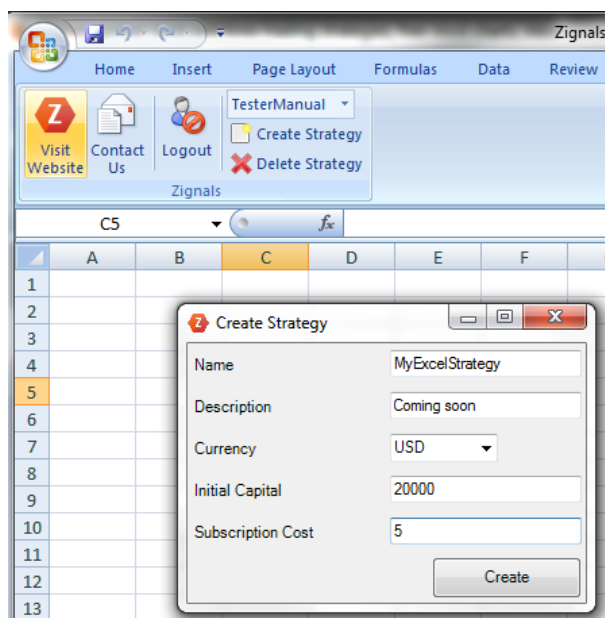
WATCH FOR! If you see "Sub Signals" as the first line, remove this line



[Excel7.png: 676 x 234; lb: 473 x 163][Alt: Sometimes this extra Sub command appears as the first entry; this will return an error if the Macro is run and this line is present]

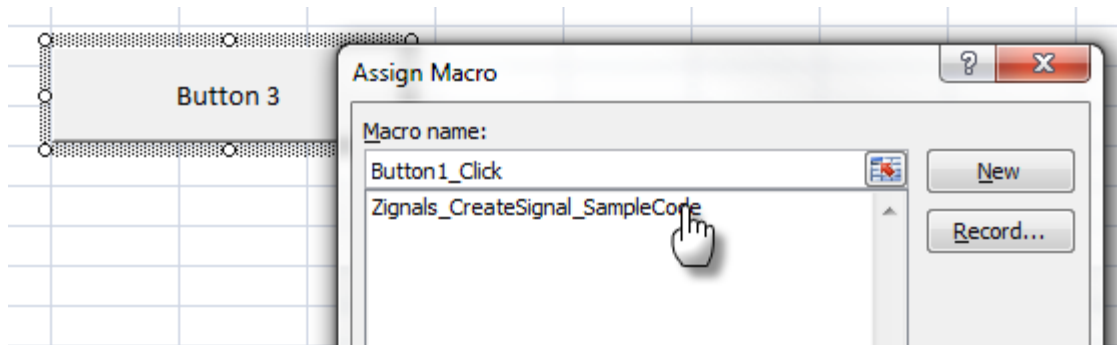
Close the Visual Basic dialog - you don't need to Save at this time, but if you do, remember to save as a Macro-Enabled Workbook.

8. In the Signals ribbon menu, create a Strategy if you haven't already; Subscription Costs rise in order of \$5 from \$5 up to \$100.



[Excel6.png 443 x 456.png; lb: 265 x 275][Alt: Create a Trading Strategy where you offer the trading ideas. Set your starting capital and subscription cost for potential subscribers]

9. From the Developer menu, insert a button. When a button is created you need to select "Signals_CreateSignal_SampleCode" and assign this Macro to your button.



[Excel8.png 443 x 456.png; lb: 335 x 102][Alt: Assigning a Macro to a button; Select the Zignals_CreateSignals_Samplecode and this will associate with your button]

9. The current Macro requires signals to be generated directly from the VBA code; the newly created Button is used to send the signals to your Zignals Portfolio.

	A	B	C	D
1				
2				
3		Run Signals Macro		
4		Macro reference		
5	Symbol	C5	MSFT.NMS	CSCO.NMS
6	Shares	C6	500	100
7	isBuying	C7	TRUE	TRUE
8	ExtraInfo	C8	Buy	Buy
9	TargetPrice	C9	40.20	99.00
10	StopPrice	C10	-1.00	36.99
11	TriggerPrice	C11	28.30	40.25

[Excel10.png 367 x 246.png; lb: 256 x 172][Alt: The Macro Button above an example format for accepting trades from a spreadsheet]

More advanced users can look to build out support so that signals can be taken directly from the parent spreadsheet.

10. module contains set of functions(operations) that communicate with Zignals External Signals platform:

- CreateEntry
- UpdateTarget
- UpdateStop
- UpdateTargetAndStop
- CreatePremarketEntry
- CancelPremarketEntry
- CreateExit
- CreatePartialExit
- CreatePreemptiveEntry
- UpdatePreemptiveEntry
- CancelPreemptiveEntry

- each function returns whether the operation was successful
- if the operation was not successful, the error variable(last parameter in every function) tells why

A New Buy Signal

Function `Signals_CreateSignal_CreateEntry(strategyName As String, symbol As String, shares As Double, isBuying As Boolean, extraInfo As String, targetPrice As Double, stopPrice As Double, ByRef error As String) As Boolean`

Update Target for an Existing Position

Function `Signals_CreateSignal_UpdateTarget(strategyName As String, symbol As String, extraInfo As String, targetPrice As Double, ByRef error As String) As Boolean`

Update Stop for an Existing Position

Function `Signals_CreateSignal_UpdateStop(strategyName As String, symbol As String, extraInfo As String, stopPrice As Double, ByRef error As String) As Boolean`

Update Target and Stop for an Existing Position

Function `Signals_CreateSignal_UpdateTargetAndStop(strategyName As String, symbol As String, extraInfo As String, targetPrice As Double, stopPrice As Double, ByRef error As String) As Boolean`

Create Next Day Entry Signal

Function `Signals_CreateSignal_CreatePremarketEntry(strategyName As String, symbol As String, shares As Double, isBuying As Boolean, extraInfo As String, targetPrice As Double, stopPrice As Double, ByRef error As String) As Boolean`

Cancel Next Day Entry Signal

Function `Signals_CreateSignal_CancelPremarketEntry(strategyName As String, symbol As String, extraInfo As String, ByRef error As String) As Boolean`

Create Next Day Exit Signal

Function `Signals_CreateSignal_CreatePremarketExit(strategyName As String, symbol As String, extraInfo As String, ByRef error As String) As Boolean`

Cancel Next Day Exit Signal

Function `Signals_CreateSignal_CancelPremarketExit(strategyName As String, symbol As String, extraInfo As String, ByRef error As String) As Boolean`

Exit a Current Position

Function `Signals_CreateSignal_CreateExit(strategyName As String, symbol As String, extraInfo As String, ByRef error As String) As Boolean`

Exit Part of a Current Position

Function `Signals_CreateSignal_CreatePartialExit(strategyName As String, symbol As String, shares As Double, extraInfo As String, ByRef error As String) As Boolean`

Create Pre-Emptive Signal

Function `Signals_CreateSignal_CreatePreemptiveEntry(strategyName As String, symbol As String, shares As Double, isBuying As Boolean, extraInfo As`

String, triggerPrice As Double, targetPrice As Double, stopPrice As Double, ByRef error As String) As Boolean

Update an Existing Pre-Emptive Signal

Function Signals_CreateSignal_UpdatePreemptiveEntry(strategyName As String, symbol As String, shares As Double, isBuying As Boolean, extraInfo As String, triggerPrice As Double, targetPrice As Double, stopPrice As Double, ByRef error As String) As Boolean

Cancel a Pre-Emptive Signal

Function Signals_CreateSignal_CancelPreemptiveEntry(strategyName As String, symbol As String, extraInfo As String, ByRef error As String) As Boolean

Commands

How do I enter a Short Position	Set "isBuying" to FALSE
What are the codes for each exchange? Eg. MSFT => MSFT.NMS	Nasdaq - NMS NYSE - NYS UK - LSE Ireland - ISE Toronto - TOR Canadian Venture Exchange - CVE Australian Index - ASX Netherlands - AMS Bombay - BSE National Stock Exchange India - NSI France - PAR Germany - FRA Alternext - ALT Portugal - LIS Belgium - BRU

3. How to use the API to send Trading Signals and Updates to Signals:

Open the workbook & Enable macros (Remember, the Signals module has to be in the module directory of the VBA Editor).

Login in to Signals (via ribbon) - logging in has to be done **after you** have enabled the macros

To create a signal follow these steps

1. get data from cells which will be used as parameters for
Signals_CreateSignal_CreateEntry function

```
Dim strategyName As String
Dim symbol As String
Dim shares As Double
Dim isBuying As Boolean
Dim extraInfo As String
Dim targetPrice As Double
Dim stopPrice As Double

strategyName = "MyFirstStrategy"
symbol = "MSFT.NMS"
shares = 100
isBuying = True
extraInfo = "the price is incredibly low"
targetPrice = 30.2
stopPrice = -1 'negative number means that stop is not specified
(the same applies to targetPrice)
```

2. Set the function call

```
Dim error As String
Dim result As Boolean

result = Signals_CreateSignal_CreateEntry(strategyName, symbol,
shares, isBuying, extraInfo, targetPrice, stopPrice, error)
```

3. Process response (error handling, output to user, ..)

```
MsgBox "Result: " & result & " Error: " & error
```

```

Sub Signals_CreateSignal_SampleCode()
Dim error As String
Dim result As Boolean

Dim symbol As String
Dim shares As Double
Dim isBuying As Boolean
Dim extraInfo As String
Dim targetPrice As Double
Dim stopPrice As Double
Dim triggerPrice As Double

'symbol = Range("C5").Value
'shares = 100
'isBuying = True
'extraInfo = "the price is incredibly low"
'targetPrice = 30.2
'stopPrice = -1 'negative number means that stop is not specified (the same applies to targetPrice)
'triggerPrice = 28.3

symbol = Range("C5").Value
shares = Range("C6").Value
isBuying = True
extraInfo = Range("C8").Value
targetPrice = Range("C9").Value
stopPrice = Range("C10").Value 'negative number means that stop is not specified (the same applies to targetPrice)
triggerPrice = Range("C11").Value

'SAMPLE 0: unrecognized symbol => this will return false
'symbol = "MSFT"
'result = Signals_CreateSignal_CreateEntry(symbol, shares, isBuying, extraInfo, targetPrice, stopPrice, error)
'MsgBox "Result: " & result & " Error: " & error

'SAMPLE 1: market has to be open for this to work
'symbol = "MSFT.NMS"
result = Signals_CreateSignal_CreateEntry(symbol, shares, isBuying, extraInfo, targetPrice, stopPrice, error)
MsgBox "Result: " & result & " Error: " & error

```

	A	B	C	D
1				
2				
3		Run Signals Macro		
4		Macro reference		
5	Symbol	C5	MSFT.NMS	CSCO.NMS
6	Shares	C6	500	100
7	isBuying	C7	TRUE	TRUE
8	ExtraInfo	C8	Buy	Buy
9	TargetPrice	C9	40.20	99.00
10	StopPrice	C10	-1.00	36.99
11	TriggerPrice	C11	28.30	40.25

[Excel 9: 981 x 629; lb: 588 x 377][Alt: An example Buy order for MSFT.NMS referencing a spreadsheet detailing the required information; note the requirement to uncomment the Macro code to accept the signal]

VBA Script

```
Sub Zignals_CreateSignal_SampleCode()  
    Dim error As String  
    Dim result As Boolean  
  
    Dim strategyName As String  
    Dim symbol As String  
    Dim shares As Double  
    Dim isBuying As Boolean  
    Dim extraInfo As String  
    Dim targetPrice As Double  
    Dim stopPrice As Double  
    Dim triggerPrice As Double  
  
    strategyName = "my_first_strategy"  
    symbol = "MSFT.NMS"  
    shares = 100  
    isBuying = True  
    extraInfo = "the price is incredibly low"  
    targetPrice = 130.2  
    stopPrice = -1 'negative number means that stop is not specified (the  
same applies to targetPrice)  
    triggerPrice = 128.3  
  
    'SAMPLE 0: unrecognized symbol => this will return false  
    symbol = "MSFT"  
    result = Zignals_CreateSignal_CreateEntry(strategyName, symbol,  
shares, isBuying, extraInfo, targetPrice, stopPrice, error)  
  
    'SAMPLE 1: market has to be open for this to work  
    'symbol = "AIB.ISE"  
    'result = Zignals_CreateSignal_CreateEntry(strategyName, symbol,  
shares, isBuying, extraInfo, targetPrice, stopPrice, error)  
  
    'SAMPLE 2: market has to be open for this to work  
    'symbol = "MSFT.NMS"  
    'result = Zignals_CreateSignal_CreateEntry(strategyName, symbol,  
shares, isBuying, extraInfo, targetPrice, stopPrice, error)  
  
    'SAMPLE 3: updates target, entry has to exist, setting the target to  
negative value removes the target  
    'result = Zignals_CreateSignal_UpdateTarget(strategyName, symbol,  
extraInfo, targetPrice, error)  
  
    'SAMPLE 4: updates stop, entry has to exist, setting the stop to  
negative value removes the stop  
    'result = Zignals_CreateSignal_UpdateStop(strategyName, symbol,  
extraInfo, stopPrice, error)  
  
    'SAMPLE 5: updates target & stop, entry has to exist
```

```

    'result = Signals_CreateSignal_UpdateTargetAndStop(strategyName,
symbol, extraInfo, targetPrice, stopPrice, error)

    'SAMPLE 6: entry is not allowed when the market is open
    'symbol = "AIB.NYS"
    'result = Signals_CreateSignal_CreatePremarketEntry(strategyName,
symbol, shares, isBuying, extraInfo, targetPrice, stopPrice, error)

    'SAMPLE 7: cancel premarket entry
    'result = Signals_CreateSignal_CancelPremarketEntry(strategyName,
symbol, extraInfo, error)

    'symbol = "MSFT.NMS"
    'result = Signals_CreateSignal_CreateExit(strategyName, symbol,
extraInfo, error)
    'result = Signals_CreateSignal_CreatePartialExit(strategyName, symbol,
shares, extraInfo, error)

    'result = Signals_CreateSignal_CreatePreemptiveEntry(strategyName,
symbol, shares, isBuying, extraInfo, triggerPrice, targetPrice, stopPrice,
error)
    'result = Signals_CreateSignal_UpdatePreemptiveEntry(strategyName,
symbol, shares, isBuying, extraInfo, triggerPrice, targetPrice, stopPrice,
error)
    'result = Signals_CreateSignal_CancelPreemptiveEntry(strategyName,
symbol, extraInfo, error)

    MsgBox "Result: " & result & " Error: " & error

    'Range("A1").Select
    'ActiveCell.FormulaR1C1 = result
    'Range("A2").Select
    'ActiveCell.FormulaR1C1 = error
End Sub

Function Signals_GetSignalCreator() As Object

    Dim addIn As COMAddIn
    Set addIn = Application.COMAddIns("SignalsExternalSignalsExcelAddIn")
    Set Signals_GetSignalCreator = addIn.Object

End Function

Private Function Signals_CreateSignal_CreateEntry(strategyName As String,
_
symbol As String, shares As Double, isBuying As Boolean, extraInfo As
String, _
targetPrice As Double, stopPrice As Double, ByRef error As String) As
Boolean

    'COM Call (returns true if the operation was successful, error
variable contains info if something went wrong)

```

```
    Signals_CreateSignal_CreateEntry =  
    Signals_GetSignalCreator().CreateEntry(strategyName, symbol, shares,  
    isBuying, extraInfo, targetPrice, stopPrice, error)
```

```
End Function
```

```
Function Signals_CreateSignal_UpdateTarget(strategyName As String, _  
    symbol As String, extraInfo As String, _  
    targetPrice As Double, ByRef error As String) As Boolean
```

```
'COM Call (returns true if the operation was successful, error  
variable contains info if something went wrong)
```

```
    Signals_CreateSignal_UpdateTarget =  
    Signals_GetSignalCreator().UpdateTarget(strategyName, symbol, extraInfo,  
    targetPrice, error)
```

```
End Function
```

```
Function Signals_CreateSignal_UpdateStop(strategyName As String, _  
    symbol As String, extraInfo As String, _  
    stopPrice As Double, ByRef error As String) As Boolean
```

```
'COM Call (returns true if the operation was successful, error  
variable contains info if something went wrong)
```

```
    Signals_CreateSignal_UpdateStop =  
    Signals_GetSignalCreator().UpdateStop(strategyName, symbol, extraInfo,  
    stopPrice, error)
```

```
End Function
```

```
Function Signals_CreateSignal_UpdateTargetAndStop(strategyName As String,  
    _  
    symbol As String, extraInfo As String, _  
    targetPrice As Double, stopPrice As Double, ByRef error As String) As  
Boolean
```

```
'COM Call (returns true if the operation was successful, error  
variable contains info if something went wrong)
```

```
    Signals_CreateSignal_UpdateTargetAndStop =  
    Signals_GetSignalCreator().UpdateTargetAndStop(strategyName, symbol,  
    extraInfo, targetPrice, stopPrice, error)
```

```
End Function
```

```
Function Signals_CreateSignal_CreatePremarketEntry(strategyName As String,  
    _  
    symbol As String, shares As Double, isBuying As Boolean, extraInfo As  
String, _  
    targetPrice As Double, stopPrice As Double, ByRef error As String) As  
Boolean
```

```
'COM Call (returns true if the operation was successful, error  
variable contains info if something went wrong)
```

```
    Signals_CreateSignal_CreatePremarketEntry =  
    Signals_GetSignalCreator().CreatePremarketEntry(strategyName, symbol,  
    shares, isBuying, extraInfo, targetPrice, stopPrice, error)
```

```
End Function
```

```
Function Signals_CreateSignal_CancelPremarketEntry(strategyName As String,
```

```
    symbol As String, extraInfo As String, _  
    ByRef error As String) As Boolean
```

```
    'COM Call (returns true if the operation was successful, error  
    variable contains info if something went wrong)
```

```
    Signals_CreateSignal_CancelPremarketEntry =  
    Signals_GetSignalCreator().CancelPremarketEntry(strategyName, symbol,  
    extraInfo, error)
```

```
End Function
```

```
Function Signals_CreateSignal_CreateExit(strategyName As String, _  
    symbol As String, extraInfo As String, _  
    ByRef error As String) As Boolean
```

```
    'COM Call (returns true if the operation was successful, error  
    variable contains info if something went wrong)
```

```
    Signals_CreateSignal_CreateExit =  
    Signals_GetSignalCreator().CreateExit(strategyName, symbol, extraInfo,  
    error)
```

```
End Function
```

```
Function Signals_CreateSignal_CreatePartialExit(strategyName As String, _  
    symbol As String, shares As Double, extraInfo As String, _  
    ByRef error As String) As Boolean
```

```
    'COM Call (returns true if the operation was successful, error  
    variable contains info if something went wrong)
```

```
    Signals_CreateSignal_CreatePartialExit =  
    Signals_GetSignalCreator().CreatePartialExit(strategyName, symbol, shares,  
    extraInfo, error)
```

```
End Function
```

```
Function Signals_CreateSignal_CreatePreemptiveEntry(strategyName As  
String, _  
    symbol As String, shares As Double, isBuying As Boolean, extraInfo As  
String, _  
    triggerPrice As Double, targetPrice As Double, stopPrice As Double,  
ByRef error As String) As Boolean
```

```
    'COM Call (returns true if the operation was successful, error  
    variable contains info if something went wrong)
```

```
    Signals_CreateSignal_CreatePreemptiveEntry =  
Signals_GetSignalCreator().CreatePreemptiveEntry(strategyName, symbol,  
shares, isBuying, extraInfo, triggerPrice, targetPrice, stopPrice, error)
```

```
End Function
```

```
Function Signals_CreateSignal_UpdatePreemptiveEntry(strategyName As  
String, _  
    symbol As String, shares As Double, isBuying As Boolean, extraInfo As  
String, _  
    triggerPrice As Double, targetPrice As Double, stopPrice As Double,  
ByRef error As String) As Boolean
```

```
    'COM Call (returns true if the operation was successful, error  
variable contains info if something went wrong)  
    Signals_CreateSignal_UpdatePreemptiveEntry =  
Signals_GetSignalCreator().UpdatePreemptiveEntry(strategyName, symbol,  
shares, isBuying, extraInfo, triggerPrice, targetPrice, stopPrice, error)
```

```
End Function
```

```
Function Signals_CreateSignal_CancelPreemptiveEntry(strategyName As  
String, _  
    symbol As String, extraInfo As String, _  
ByRef error As String) As Boolean
```

```
    'COM Call (returns true if the operation was successful, error  
variable contains info if something went wrong)  
    Signals_CreateSignal_CancelPreemptiveEntry =  
Signals_GetSignalCreator().CancelPreemptiveEntry(strategyName, symbol,  
extraInfo, error)
```

```
End Function
```

```
Function Signals_CreateSignal_CreatePremarketExit(strategyName As String,  
-  
    symbol As String, extraInfo As String, _  
ByRef error As String) As Boolean
```

```
    'COM Call (returns true if the operation was successful, error  
variable contains info if something went wrong)  
    Signals_CreateSignal_CreateExit =  
Signals_GetSignalCreator().CreatePremarketExit(strategyName, symbol,  
extraInfo, error)
```

```
End Function
```

```
Function Signals_CreateSignal_CancelPremarketExit(strategyName As String,  
-  
    symbol As String, extraInfo As String, _  
ByRef error As String) As Boolean
```

```
    'COM Call (returns true if the operation was successful, error  
variable contains info if something went wrong)
```

```
    Signals_CreateSignal_CreateExit =  
Signals_GetSignalCreator().CancelPremarketExit(strategyName, symbol,  
extraInfo, error)
```

```
End Function
```

4. Possible errors (list is incomplete, subject to change):

Error	Reason / FIX
Symbol is not recognized	MSFT is traded on NMS & AMS / it's required to use "MSFT.NMS" instead of "MSFT"
Market is not open at the moment	Market is closed / use "create premarket entry" instead of "create entry"
Positive integer is required for 'numbers of shares' - parameter.	
Market is open - price is available. Please use entry	
Target price is not set properly	For long position the target price has to be greater than actual price
Stop price is not set properly	For long position the stop price has to be less than actual price
Target & Stop are not properly set.	
Symbol is in short position.	If you own 1000 shares, you can't short 1000 shares / use exit to sell all 1000 shares
Symbol is in long position.	
Symbol is not in open position.	You can't sell 1000 shares if you don't own them / buy shares first with "create entry"