

EXCEL INTEGRATION GUIDE



Version 1.1.4



Contents

[Contents](#)

[Introduction](#)

[Installation](#)

[Retrieving data from ATP](#)

[Introducing RTD function](#)

[Quote Monitor](#)

[Account Summary. General info](#)

[Account Summary. Currencies and instruments](#)

[Trading using Excel](#)

[General information](#)

[ATPOrderType enumeration](#)

[ATPDurationType enumeration](#)

[ATPOrderStatus enumeration](#)

[ATPSession interface](#)

[ATPOrder IDispatch interface](#)

[ATPOrder IDispatch interface retrieving from EXANTE trading platform](#)

[Troubleshooting](#)

[FAQ](#)



Introduction

EXANTE Excel Integration brings data from EXANTE trading platform to Microsoft Excel and allows you to request quotes from Quote Monitor for multiple securities simultaneously as well as general information about your account and positions. With Excel integration users can automate manual activities normally done in ATP, execute algorithms and trading strategies which require automation, access to account and portfolio information, open orders, order statuses and executions.

There are two main reasons to use Excel for this task. First, Excel is familiar for most of the users. Second, learning curve of Excel VBA is small and it requires less time for development.

Installation

All you need to do is download and run an installation file appropriate to your OS version. The software is available in two editions:

- Version for 32-bit Windows: <https://updates.exante.eu/atp-excel/EXANTE-ATP-Excel-32.msi>
- Version for 64-bit Windows: <https://updates.exante.eu/atp-excel/EXANTE-ATP-Excel-all.msi>

If you are not sure which version of Windows you're using, please refer to this article:

<https://support.microsoft.com/en-us/kb/827218>

We also recommend you installing Microsoft Visual C++ 2017 Redistributable Package, version 14.16.27027 or newer. You can download it here:

- Version for 32-bit Windows: https://aka.ms/vs/15/release/vc_redist.x86.exe
- Version for 64-bit Windows: https://aka.ms/vs/15/release/vc_redist.x64.exe



Retrieving data from ATP

Introducing RTD function

RTD function is the main instrument for accessing data from EXANTE trading platform. This function retrieves real-time data from a program that supports automation. Please refer to <https://support.microsoft.com/en-us/kb/289150> for more details. Generic syntax for ATP data retrieval with this function is:

```
1 =RTD("atp.rtd";"atp";"type";"param1";"param2"["param3"["param4"...]])
```

Examples for specific modules can be found in following sections of this guide. Depending on localization, different separators could be used. If Excel raises validation error on this function, check troubleshooting section below.

Plugin and Connection Info

```
1 =RTD("atp.rtd";"atp";"info";"version")
2 =RTD("atp.rtd";"atp";"info";"heartbeat")
```

Quote Monitor

Quote Monitor allows you to request data for any instrument, that is available to you in EXANTE trading platform. All you need to do is to paste function shown in RTD function and replace type with "quote", param1 with EXANTE ID of desired instrument and param2 with data you would like to retrieve.

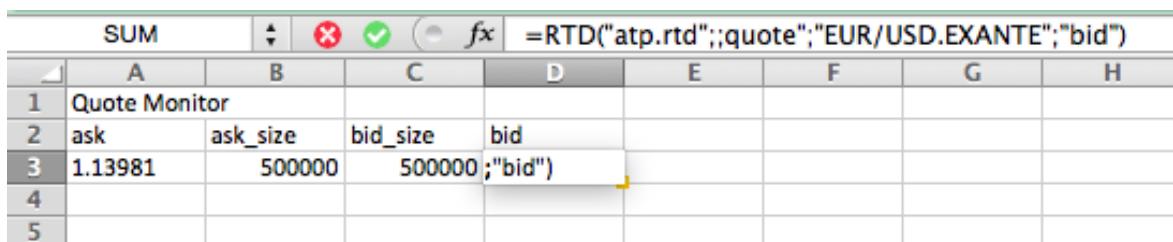
	A	B	C	D	E
1	Quote Monitor				
2		bid size	bid	ask	ask size
3	EUR/USD.EXANTE	500000	1.13828	1.13835	500000
4	EUR/GBP.EXANTE	500000	0.74065	0.74079	500000
5	GBP/USD.EXANTE	500000	1.53673	1.53684	500000
6	AUD/USD.EXANTE	500000	0.77949	0.77961	500000
7	EUR/AUD.EXANTE	500000	1.46013	1.46036	500000
8	GAZP.MICEX	80	157.47	157.5	166
9	GMKN.MICEX	2	11858	11868	66
10	ROSN.MICEX	82	278.9	278.95	62

Examples:

```

1 =RTD("atp.rtd";"atp";"quote";"EUR/USD.EXANTE";"ask")
2 =RTD("atp.rtd";"atp";"quote";"EUR/USD.EXANTE";"ask_size")
3 =RTD("atp.rtd";"atp";"quote";"EUR/USD.EXANTE";"bid")
4 =RTD("atp.rtd";"atp";"quote";"EUR/USD.EXANTE";"bid_size")
5 =RTD("atp.rtd";"atp";"quote";"EUR/USD.EXANTE";"last_trade_price")
6 =RTD("atp.rtd";"atp";"quote";"EUR/USD.EXANTE";"open_interest")
7 =RTD("atp.rtd";"atp";"quote";"EUR/USD.EXANTE";"daily_volume")

```



The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F	G	H
1	Quote Monitor							
2	ask	ask_size	bid_size	bid				
3	1.13981	500000	500000	"bid")				
4								
5								

The formula bar shows: `=RTD("atp.rtd";;"quote";"EUR/USD.EXANTE";"bid")`

For options, you can also request the Greeks and implied volatility.

```

1 =RTD("atp.rtd";"atp";"quote";"LO.EXANTE.X2016.C1000";"delta")
2 =RTD("atp.rtd";"atp";"quote";"LO.EXANTE.X2016.C1000";"gamma")
3 =RTD("atp.rtd";"atp";"quote";"LO.EXANTE.X2016.C1000";"vega")
4 =RTD("atp.rtd";"atp";"quote";"LO.EXANTE.X2016.C1000";"theta")
5 =RTD("atp.rtd";"atp";"quote";"LO.EXANTE.X2016.C1000";"implied_volatility")

```

Account Summary. General info

acc.summary module allows you to request all data related to your account, such as NAV or Margin Utilization. In order to request than, use "acc.summary" as type, your account name in param1 and desired information in param2 of RTD.

Examples:

```

1 =RTD("atp.rtd";"atp";"acc.summary";"ABK411.001";"account_id")
2 =RTD("atp.rtd";"atp";"acc.summary";"ABK411.001";"currency")
3 =RTD("atp.rtd";"atp";"acc.summary";"ABK411.001";"timestamp")
4 =RTD("atp.rtd";"atp";"acc.summary";"ABK411.001";"net_asset_value")
5 =RTD("atp.rtd";"atp";"acc.summary";"ABK411.001";"free_money")
6 =RTD("atp.rtd";"atp";"acc.summary";"ABK411.001";"money_used_for_margin")
7 =RTD("atp.rtd";"atp";"acc.summary";"ABK411.001";"margin_utilization")
    
```

	A	B	C	D	E	F	G
1							
2	ABK411.001				Cash		
3							
4						Asset	Value
5	account_id	ABK411.001			CHF	CHF	=RTD("atp.rtd";"atp";"acc.summary";"ABK411.001";"value")
6	currency	USD			EUR	EUR	
7	net_asset_value	585148.49			GBP	GBP	
8	free_money	356719.13			USD	USD	
9	money_used_for_margin	22842.36			RUB	RUB	
10	margin_utilization	39%					
11							

Account Summary. Currencies and instruments

In order to receive data about specific cash or instrument positions, use "acc.curr" for currencies and "acc.asset" for other instruments as type for the RTD function. Param1 should be name of the instrument (or currency) and param2 the is desired information. Also you can receive data about the number of different currencies and

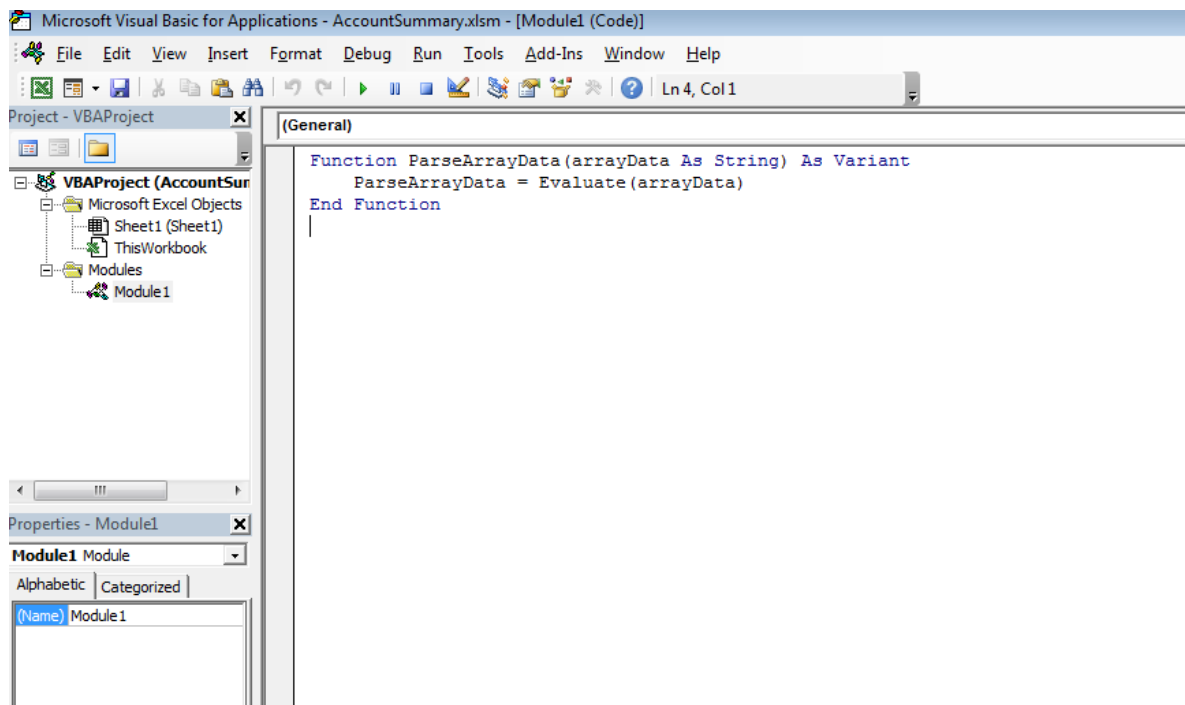


positions, and list of all currencies and positions that you have (examples 17-22). Examples 18-19 and 21-22 will return a pseudo-array, to evaluate this pseudo-array to Excel ones this macro required:

```
1 Function ParseArrayData(ArrayData as String) as Variant
2     ParseArrayData = Evaluate(ArrayData)
3 End Function
```

After the macro has been created you please use an according formula to count the number of cells you will need, select exact number of cells, then paste to them this formula and press Ctrl+Shift+Enter:

```
1 =ParseArrayData(RTD("atp.rtd", "atp", "acc.curr", "ABC1234.001",
"currencyes", "to_column"))
```



Examples:

```

1 =RTD("atp.rtd";"atp";"acc.curr";"ABK411.001";"USD";"asset")
2 =RTD("atp.rtd";"atp";"acc.curr";"ABK411.001";"USD";"value")
3 =RTD("atp.rtd";"atp";"acc.curr";"ABK411.001";"USD";"converted_value")
4
5 =RTD("atp.rtd";"atp";"acc.asset";"ABK411.001";"STOCK1.FIX";"id")
6 =RTD("atp.rtd";"atp";"acc.asset";"ABK411.001";"STOCK1.FIX";"instrument")
7 =RTD("atp.rtd";"atp";"acc.asset";"ABK411.001";"STOCK1.FIX";"exchange")
8 =RTD("atp.rtd";"atp";"acc.asset";"ABK411.001";"STOCK1.FIX";"cusip")
9 =RTD("atp.rtd";"atp";"acc.asset";"ABK411.001";"STOCK1.FIX";"quantity")
10 =RTD("atp.rtd";"atp";"acc.asset";"ABK411.001";"STOCK1.FIX";"currency")
11 =RTD("atp.rtd";"atp";"acc.asset";"ABK411.001";"STOCK1.FIX";"pnl")
12 =RTD("atp.rtd";"atp";"acc.asset";"ABK411.001";"STOCK1.FIX";"converted_pnl")
13 =RTD("atp.rtd";"atp";"acc.asset";"ABK411.001";"STOCK1.FIX";"price")
14 =RTD("atp.rtd";"atp";"acc.asset";"ABK411.001";"STOCK1.FIX";"average_price")
15 =RTD("atp.rtd";"atp";"acc.asset";"ABK411.001";"STOCK1.FIX";"value")
16 =RTD("atp.rtd";"atp";"acc.asset";"ABK411.001";"STOCK1.FIX";"converted_value")
17 =RTD("atp.rtd";"atp";"acc.curr";"ABC1234.001";"currencies";"count")
18 =RTD("atp.rtd";"atp";"acc.curr";"ABC1234.001";"currencies";"to_row")
19 =RTD("atp.rtd";"atp";"acc.curr";"ABC1234.001";"currencies";"to_column")
20 =RTD("atp.rtd";"atp";"acc.asset";"ABC1234.001";"assets";"count")
21 =RTD("atp.rtd";"atp";"acc.asset";"ABC1234.001";"assets";"to_row")
22 =RTD("atp.rtd";"atp";"acc.asset";"ABC1234.001";"assets";"to_column")

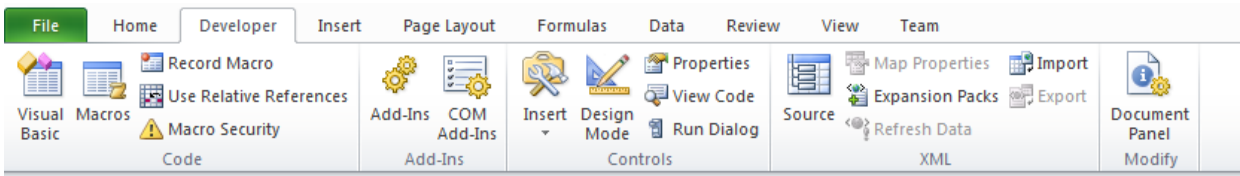
```

25													
26	Forex												
27		id	instrument	exchange	cusip	quantity	currency	pnl	converted_pnl	price	average_price	value	converted_value
28	AUD/CAD.E.FX	AUD/CAD.E.FX	AUD/CAD	E	0	12	CAD	-0.13	-0.11	0.967055	0.97749	11.6	9.36
29	EUR/GBP.E.FX	EUR/GBP.E.FX	EUR/GBP	E	0	1	GBP	-0.05	-0.08	0.741435	0.78866	0.74	1.14
30	EUR/JPY.E.FX	EUR/JPY.E.FX	EUR/JPY	E	0	28	JPY	-51.85	-0.44	135.27	137.1217	3787.56	31.89
31	EUR/USD.E.FX	EUR/USD.E.FX	EUR/USD	E	0	63	USD	-10.96	-10.96	1.1389	1.312820476	71.75	71.75
32	USD/CHF.E.FX	USD/CHF.E.FX	USD/CHF	E	0	1	CHF	0.07	0.07	0.93491	0.86968	0.93	1
33	USD/RUB.E.FX	USD/RUB.E.FX	USD/RUB	E	0	-999000	RUB	-26650772.55	-410455.32	62.66125	35.9838	-62598588.8	-964096.76
34													
35													
36													
37	Futures												
38		id	instrument	exchange	cusip	quantity	currency	pnl	converted_pnl	price	average_price	value	converted_value
39	6E.CME.M2015	AUD/CAD.E.FX	AUD/CAD	E	0	12	CAD	-0.13	-0.11	0.967055	1.3651	285037.5	285037.5
40	CL.NYMEX.J2015	CL.NYMEX.J2015	Light Sweet Crude	NYMEX	0	5	USD	-113400	-113400	54.12	76.8	270600	270600
41	HG.COMEX.Z2014	HG.COMEX.Z2014	Copper Dec 2014	COMEX	0	2	USD	65778.63	65778.63	2.84525	1.5296775	142262.5	142262.5
42	SI.COMEX.Z2015	SI.COMEX.Z2015	Silver Dec 2015	COMEX	0	1	USD	-15800.01	-15800.01	16.98	20.14	84900	84900
43	SI.FORTS.H2015	SI.FORTS.H2015	USD/RUB Mar 2015	FORTS	0	1	RUB	-3010.5	-46.37	63621.5	66632	63621.5	979.85

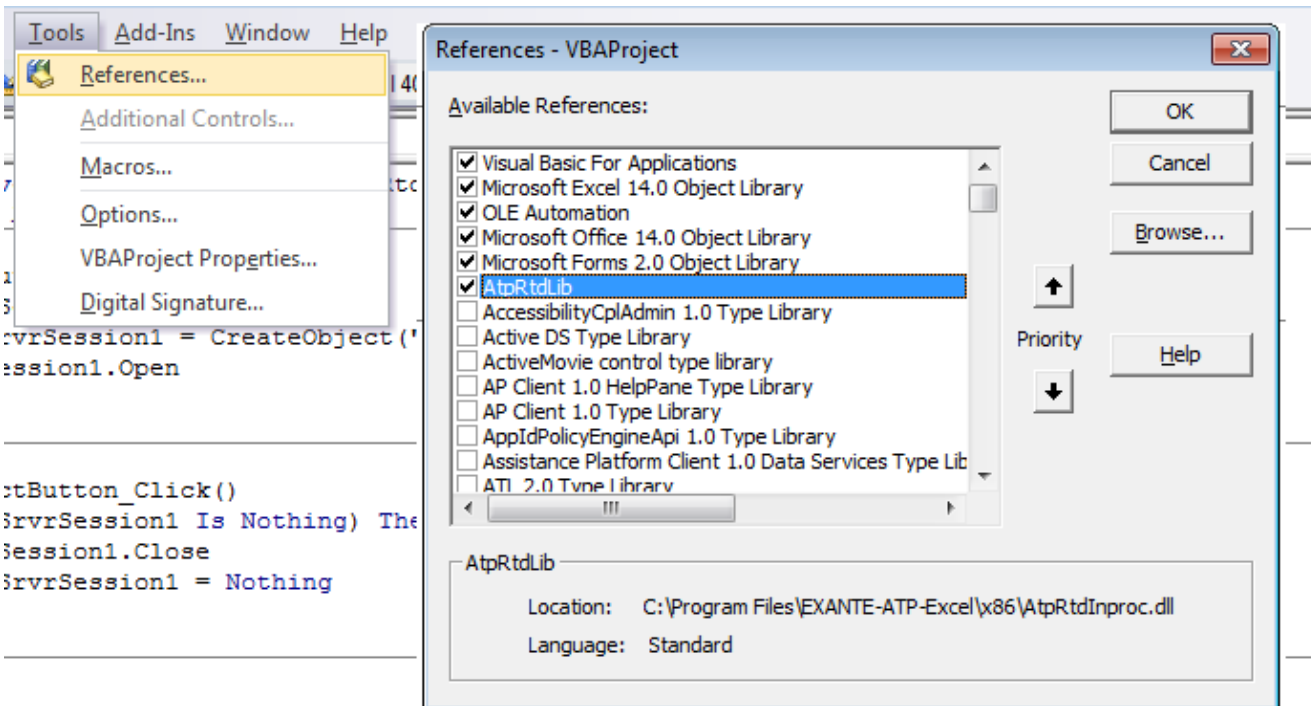
Trading using Excel

General information

In order to control orders from Excel you need to use any COM-compatible programming language. We highly recommend to use Visual Basic for Applications (VBA) to implement trading to Excel. In order to create or edit already created macro please go to Developer Ribbon and select Visual Basic. If Developer ribbon is missing, check out our FAQ.



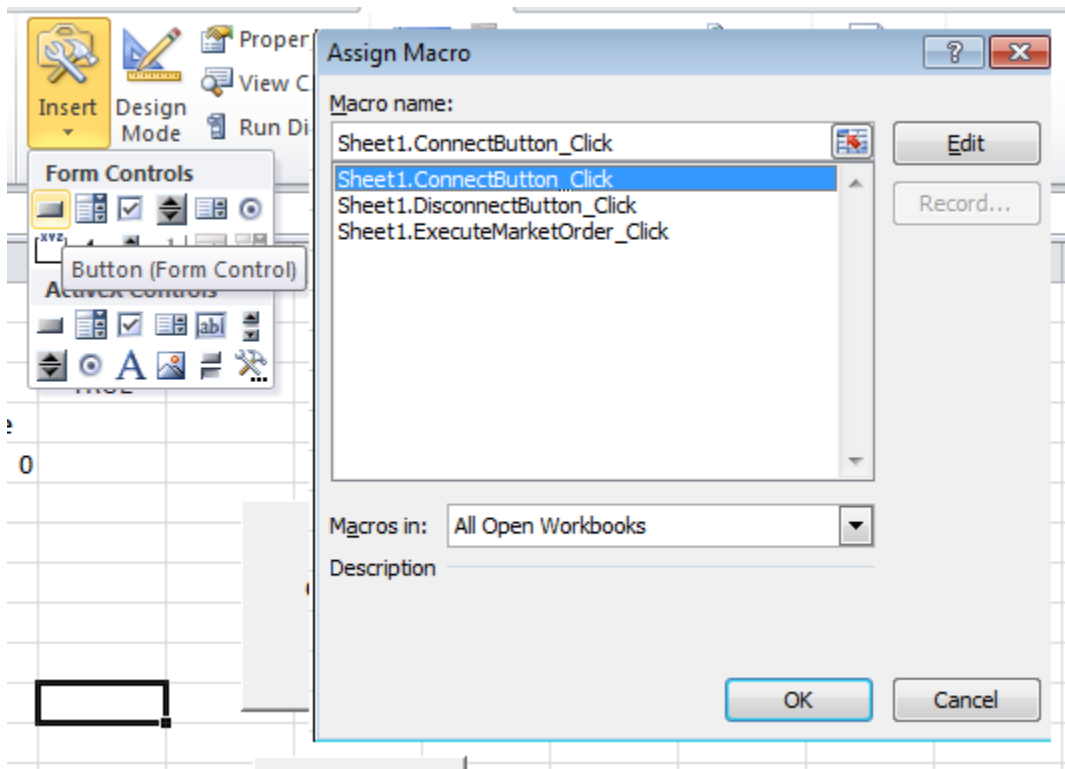
In order to make this work, AtpRtdLib should be added to reference via Tools -> References... menu





Every programmed action should be attached to a specific Form Control element, such as a button. Here's an example on how to create a connected button:

1. Click insert;
2. Choose Button;
3. Choose an area on your spreadsheet for that button;
4. In Assign Macro menu select a desired action



The methods that are recognizable by our parser may be found below.

Usually your macro should have the following structure:

1. Establish connection to EXANTE platform;
2. Check that connection is active;
3. Create a new order with required parameters;
4. Push order to EXANTE platform;
5. Pull active orders from EXANTE platform and read their parameters;
6. Close the connection to EXANTE platform.

Examples:

Here's a simple example for connect, disconnect and place order button. In order for it to work, account should be changed to one your trading platform have access to.

```
1 Public WithEvents SrvrSession1 As AtpRtdLib.ATPSession
2 Public Order As AtpRtdLib.ATPOrder
3
4 Sub ConnectButton_Click()
5     If (SrvrSession1 Is Nothing) Then
6         Set SrvrSession1 = CreateObject("atp.apisession")
7         SrvrSession1.Open
8     End If
9 End Sub
10
11 Sub DisconnectButton_Click()
12     If Not (SrvrSession1 Is Nothing) Then
13         SrvrSession1.Close
14         Set SrvrSession1 = Nothing
15     End If
16 End Sub
17
18 Sub ExecuteMarketOrder_Click()
19     If Not (SrvrSession1 Is Nothing) Then
20         Set Order = CreateObject("atp.apiorder")
21         Order.Init
22         Order.SetClientAccId ("AAN1328.001")
23         Order.SetInstrumentId ("ES.CME.Z2015")
24         Order.SetOrderType (1)
25         Order.SetQuantity (1)
26         Order.SetDuration (1)
27         If (SrvrSession1.IsActive) Then
28             Dim failPlaceOrigin As Long
29             failPlaceOrigin = SrvrSession1.PlaceOrder(Order)
30         End If
31         Set Order = Nothing
32     End If
33 End Sub
34
35 Private Sub SrvrSession1_OnATPConnectionStatus(ByRef status As Variant)
36     Cells(3, 5) = status
37 End Sub
38
39 Private Sub SrvrSession1_OnConnectEvent()
```

```

40     Cells(2, 5).Clear
41     Cells(2, 5) = "CONNECTED"
42     OrderCount = 0
43 End Sub
44
45 Private Sub SrvrSession1_OnDisconnectEvent ()
46     Cells(2, 5).Clear
47     Cells(2, 5) = "DISCONNECTED"
48 End Sub

```

ATPOrderType enumeration

Name	Value
Market	1
Limit	2
Stop	3
Stop Limit	4

ATPDurationType enumeration

Name	Value
Day	1
Good Till Cancel	2
Immediate Or Cancel	3
Fill Or Kill	4
At The Opening	5
At The Close	6

ATPOrderStatus enumeration

Name	Value
Placing	1
Working	2
Filled	3
Cancelled	4
Rejected	5
Pending	6



ATPSession interface

- ❑ `Open()`
Creates a connection to EXANTE trading platform.
- ❑ `Close()`
Closes the earlier created connection to EXANTE trading platform.
- ❑ `IsActive([out, retval] VARIANT_BOOL *value)`
Returns `True` if there is an active session, otherwise returns `False`.
- ❑ `OnConnectEvent()`
A method that is called when the connection to ATP is established.
- ❑ `OnDisconnectEvent()`
A method that is called when the connection to ATP is broken.
- ❑ `OnSnapshotBeginEvent()`
This method indicates a start of orders list enumeration.
NOTE: all orders declared before this method is called will be ignored.
- ❑ `OnSnapshotEndEvent()`
This method indicates the end of orders list enumeration.
- ❑ `OnOrderUpdateEvent([in] IDispatch *order_info)`
This method will be called on order creation or on order status update. Takes pointer to the order information as argument.
- ❑ `OnRequestFail([in] long request_id, [in] BSTR message)`
This method is called when a request (to place, cancel or replace an order) produces an error. Takes request ID and error string message as arguments.
- ❑ `OnATPConnectionStatus([in] VARIANT *connected)`

A method that is called in connection status between EXANTE trading platform and server change. Takes pointer to the current connection status (`boolean`).



ATPOrder IDispatch interface

- ❑ `Init()`
Initialize a new order.
- ❑ `PlaceOrder([in] IDispatch *order, [out, retval] long *request_id)`
Places an order. Takes an ATPOrder object as argument and returns the generated request ID `request_id`.
- ❑ `CancelOrder([in] BSTR order_id, [out, retval] long *request_id)`
Cancels an order with ID `order_id`. Takes order ID string as argument and returns the generated request ID `request_id`.
- ❑ `ReplaceOrder([in] BSTR order_id, IDispatch *order, [out, retval] long *request_id)`
Replaces an order with ID `order_id` to a new order `order`. Takes order ID string and new ATPOrder object as arguments and returns the generated request ID `request_id`.
- ❑ `QueryOrders([out, retval] long *request_id)`
Returns the generated request ID `request_id`.
- ❑ `GetOrderCount([out, retval] long *value)`

Returns orders count `value` from EXANTE trading platform.
- ❑ `GetOrderInfo([in] long idx, [out, retval] IDispatch **value)`
Gets order information by order index `idx`. Takes order index as an argument and returns pointer to ATPOrderInfo object.
- ❑ `GetId([out, retval] BSTR *value)`
Returns assigned order ID string `value`.
- ❑ `SetOrderType([in] ATPOrderType ordt)`
Sets order type to `ordt`. Takes ATPOrderType object as argument.
- ❑ `GetOrderType([out, retval] ATPOrderType *value)`
Returns current order type `value` as ATPOrderType object.
- ❑ `SetInstrumentId([in] BSTR iin)`
Sets EXANTE instrument ID as `iin` to the order. Takes EXANTE instrument ID as an argument.
- ❑ `GetInstrumentId([out, retval] BSTR *value)`
Returns current order EXANTE Instrument ID `value` (as string).
- ❑ `SetClientAccId([in] BSTR caid)`
Sets account as `caid` to the order. Takes account ID as argument.



- ❑ `GetClientAccId([out,retval] BSTR *value)`
Returns current account `value` of the order as string.
- ❑ `SetPrice([in] VARIANT prc)`
Sets price `prc` to the order. Takes new order price as argument.
- ❑ `GetPrice([out,retval] VARIANT *value)`
Returns current order price `value`.
- ❑ `SetStopPrice([in] VARIANT prc)`
Sets stop price `prc` to the order. Takes new order stop price as argument.
- ❑ `GetStopPrice([out,retval] VARIANT *value)`
Returns current order stop price `value`.
- ❑ `SetQuantity([in] VARIANT value)`
Sets current order quantity as `value`. Takes new order quantity as an argument.
- ❑ `GetQuantity([out,retval] VARIANT *value)`
Returns current order quantity `value`. >0 if you buy <0 if you sell.
- ❑ `SetDuration([in] ATPDurationType value)`
Sets current order time-in-force `value`. Takes `ATPDurationType` object as an argument.
- ❑ `GetDuration([out,retval] ATPDurationType *value)`
Returns current order duration `value` as `ATPDurationType` object.
- ❑ `SetIfDoneParentId([in] BSTR iid)`
Sets parent order ID `iid` for Stop Loss/Take Profit orders. Takes parent order ID string as an argument.
- ❑ `GetIfDoneParentId([out,retval] BSTR *value)`
Returns parent order ID `value` for Stop Loss/Take Profit orders as string.
- ❑ `SetOCOGroupId([in] BSTR iid)`
Sets ID `iid` to One-Cancels-the-Other Order (OCO) group. Takes OCO group ID as string.
- ❑ `GetOCOGroupId([out,retval] BSTR *value)`
Returns ID `value` of One-Cancels-the-Other Order (OCO) group as string.

ATPOrder IDispatch interface retrieving from EXANTE trading platform

- ❑ `GetId([out,retval] BSTR *value)`
Returns order ID `value` as string.
- ❑ `GetUser([out,retval] BSTR *value)`
Returns name of user `value` who owns the order as string.



- ❑ `GetAccountId([out,retval] BSTR *value)`
Returns EXANTE account ID `value` who owns the order as string.
- ❑ `GetStatus([out,retval] ATPOrderStatus *value)`
Returns the current order status `value` as `ATPOrderStatus` object.
- ❑ `GetDuration([out,retval] ATPDurationType *value)`
Returns the order duration `value` as `ATPDurationType` object.
- ❑ `GetType([out,retval] ATPOrderType *value)`
Returns order type `value` as `ATPOrderType` Object.
- ❑ `GetInstrumentId([out,retval] BSTR *value)`
Returns EXANTE instrument ID `value` of the order as string.
- ❑ `GetQuantity([out,retval] VARIANT *value)`
Returns the current order quantity `value`. >0 if you buy <0 if you sell
- ❑ `GetPrice([out,retval] VARIANT *value)`
Returns the current order price `value`.
- ❑ `GetStopPrice([out,retval] VARIANT *value)`
Returns the current order stop price `value`.
- ❑ `GetCreateTime([out,retval] VARIANT *value)`
Returns the order creation time `value`
- ❑ `GetLastUpdate([out,retval] VARIANT *value)`
Returns the order last update time `value`.
- ❑ `GetFilled([out,retval] VARIANT *value)`
Returns the order filled quantity `value`.
- ❑ `GetAvgPrice([out,retval] VARIANT *value)`
Returns average fill price `value` of the order.
- ❑ `GetIfDoneParentId([out,retval] BSTR *value)`
Returns parent order ID `value` for Stop Loss/Take Profit orders as string.
- ❑ `GetOCOGroupId([out,retval] BSTR *value)`
Returns ID `value` of One-Cancels-the-Other Order (OCO) group as string.

Robot example

On our website you can find a trading robot example: <https://exante.eu/clientsarea/help/excel-integration-guide/> — Robot example (moving averages).



It calculates two indicators by using moving average algorithm with averaging steps specified in cells B5 and B6 (indicators 1 and 2 respectively.) In case of indicator interception robot places an order with the quantity mentioned in B7; if indicator 1 becomes higher than indicator 2 it places BUY order, otherwise SELL order. The used instrument and account may be configured in cells B1 and B2 respectively. Cell B8 controls the number of dots on the chart. If you want to learn how it works, press Alt+F11 and navigate over the defined macros.

Troubleshooting

- ❖ If any issues with the Excel Integration software arise, please go through the following check list:
 1. Make sure that your Microsoft Office (Microsoft Excel) version is 2010 or newer. We do not support older versions or third party spreadsheet editing apps.
 2. Check whether you are running a 32-bit or a 64-bit Windows version. Here's the how-to: <https://support.microsoft.com/en-us/kb/827218>
 3. Download the corresponding versions of the Plugin **and** the Microsoft Visual C++ 2017 Redistributable Package, version 14.16.27027 or newer. Use the following links:
 - 32-bit [plugin](#) and [package](#)
 - 64-bit [plugin](#) and [package](#)
 4. Turn off your antivirus software (including Windows Defender) temporarily. Run an installer and follow the instructions you see on the screen, e.g. confirm that you allow the installation. Then install the Microsoft Visual C++ 2017 Redistributable Package, version 14.16.27027 or newer, in the same manner. Reboot your computer and turn on the antivirus protection.
- ❖ If Excel returns validation errors on your formula, check if your system locale uses a different separator or function name for RTD. An official page describing it is located here (switch the site language to see another locale settings): <https://support.office.com/en-US/article/RTD-function-E0CC001A-56F0-470A-9B19-9455DC0EB593>
- ❖ In case of any other issues, EXANTE Support Team will be ready to assist with your queries 24/7. In order to reach EXANTE Support Team, please send an email to support@exante.eu describing your issue.



FAQ

- ❖ Is there an open API available for other applications besides Microsoft Excel?
 - No. The API is closed at the moment.
- ❖ Is it possible to run several copies of Excel accessing ATP?
 - Yes, it is possible.
- ❖ Can I use your Excel Integration Plugin with Microsoft Office 2007 / 2003 / LibreOffice / Apache Open Office / Apple Numbers?
 - No. Excel Integration plugin works with Microsoft Office 2011 or newer versions only.
- ❖ Can I use Excel trading robots without the EXANTE trading terminal?
 - No. You have to run the trading terminal so the plugin can function.
- ❖ There's no development section in my Excel. How do I turn it on?
 - You'll need to enable it in Excel Options. For more details please check this article:
<https://support.office.com/en-us/article/Add-a-button-and-assign-a-macro-to-it-in-a-worksheet-d58edd7d-cb04-4964-bead-9c72c843a283>
- ❖ Macros and Visual Basic buttons are grayed out. How do I fix that?
 - This means Microsoft Office cannot access Visual Basic for Applications installation. It could happen due to the following reasons:
 - It wasn't installed. In this case you'll need to go to the Programs and Features menu, right click on Microsoft Office, click Change, select Add or Remove Features and check Visual Basic for Applications box.
 - If Visual Basic for Applications is already installed, Repair option should help to resolve this issue.
 - If Repair option does not help either, try removing VBA component and installing it again.
 - For more details, please check <https://support.microsoft.com/en-us/kb/282847>
- ❖ I have other questions.
 - Please contact support@exante.eu with your question. Our Support Team will be happy to assist you.