



AutoX

Setup Guide

August 19, 2015

TradeStation™ Interface via email notification

The AutoX plugin supports capturing four types of strategy notifications, order type mapping (buy to cover to buy or sell short to sell), manual order confirmation, symbol mapping, advanced account assignment, and redirection of email notifications to user's email client. In order for signals to be captured from TradeStation™, the local SMTP port cannot be used by another process. Moreover, only one OEC Trader can capture signals per computer.

Installation Instructions:

AutoX may be found on our website at: <http://www.gainfutures.com/education-center/downloads.cfm>. Please make sure OEC Trader is closed when installing the plugin. Download the file from link provided and run the installer and follow the screen prompts.

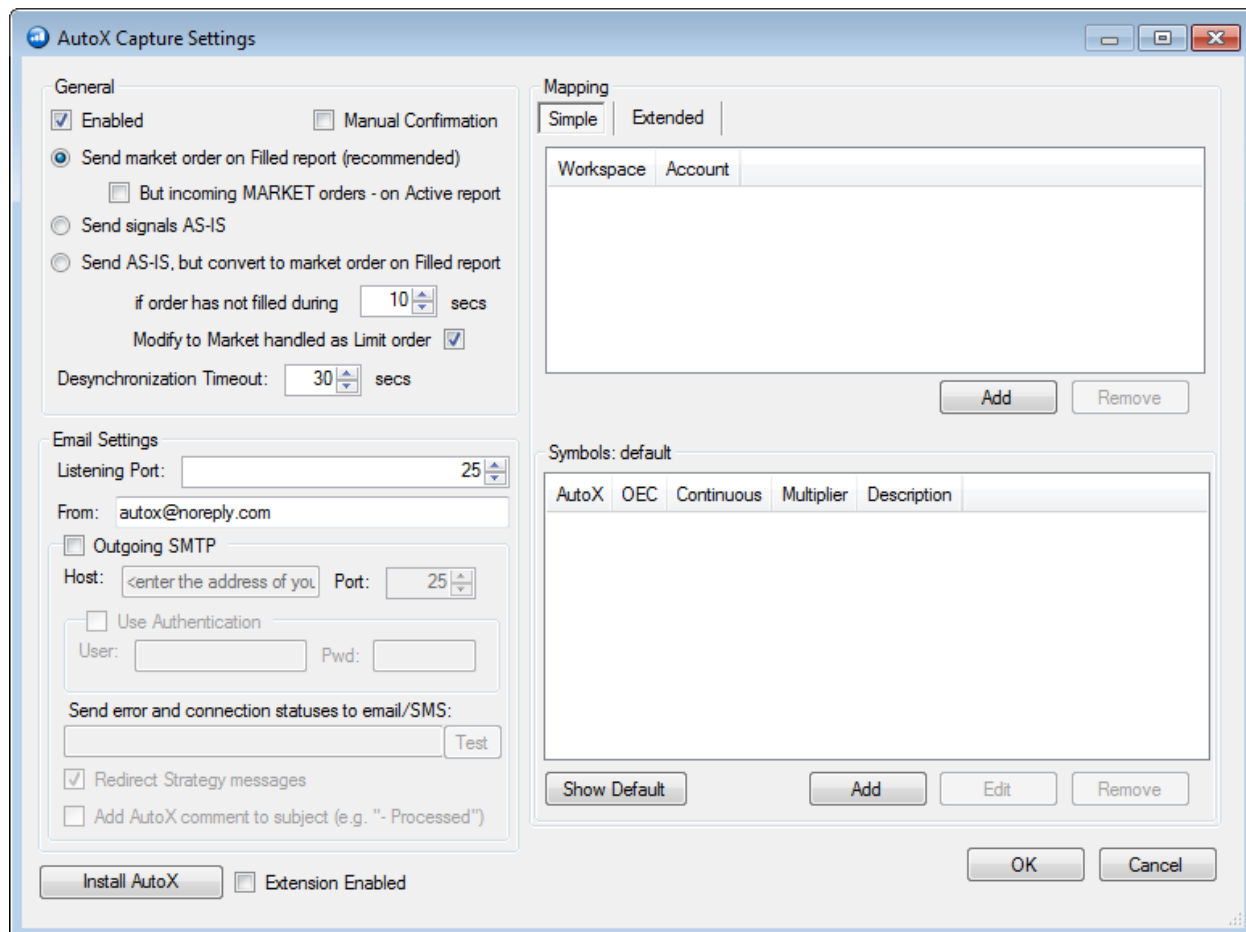
TradeStation™ and OEC Trader Settings

System requirements

- If using AutoX on a server, please be sure that Windows SMTP service is disabled. To check, open the Windows Control Panel and go to the Administrative Tools, then go to Services area. Disable the Windows SMTP service, if it is installed and started: use a right click to open "Properties." Change "Startup type" to "Disabled" and press "Stop," then OK.
- If you use a SMTP service, please stop this service or re-configure it to not use port 25. Be sure to close all other TradeStation™ capture applications which use port 25.
- Check your firewall settings and make sure the SMTP port is not blocked for local use.
- Only one instance of OEC Trader can capture TradeStation™ signals (eg. Demo or Live but not both platforms running AutoX at once).

OEC Trader

1. From the menu bar of OEC Trader, go to **File > Plugins > AutoX > Capture Settings:**
2. Fill out the **"From:"** field in the **Email settings** section with the same email address used in TradeStation™ Messaging, eg. **autox@noreply.com**.
3. Check **"Outgoing SMTP"** box, if you want to redirect the TradeStation™ messages to your email client. This is not necessary unless you require notification be redirected to your email.
4. Check **"Enabled"** and adjust **"Manual Confirmation"** setting, if necessary.
5. Set the appropriate mapping settings, explained in the next section of user guide on page 8.



AutoX Capture Settings

General

☒ Enabled ☐ Manual Confirmation

☒ Send market order on Filled report (recommended)
☐ But incoming MARKET orders - on Active report

☐ Send signals AS-IS

☐ Send AS-IS, but convert to market order on Filled report
if order has not filled during secs

Modify to Market handled as Limit order ☒

Desynchronization Timeout: secs

Mapping

Simple Extended

Workspace	Account

Add Remove

Email Settings

Listening Port:

From:

☐ Outgoing SMTP

Host: Port:

☐ Use Authentication

User: Pwd:

Send error and connection statuses to email/SMS:
 Test

☒ Redirect Strategy messages

☐ Add AutoX comment to subject (e.g. "- Processed")

Install AutoX ☐ Extension Enabled

Symbols: default

AutoX	OEC	Continuous	Multiplier	Description

Show Default Add Edit Remove

OK Cancel

Options Explained:

Feature	Description
Enabled	Start capturing TradeStation™ signals
Manual Confirmation	Asks for user a confirmation on all TradeStation™ reports. User can review corresponding order and send, modify or cancel, decline report and corresponding action, or allow it to execute corresponding action immediately
Send market order on Filled report	A market order will be sent, when a TradeStation™ Strategy order is filled. All other types from TradeStation™ reports are ignored and skipped. If the order is cancelled, no actions are performed at all. This option ensures the most accurate synchronization of positions (during this process, if positions are synchronized initially)
Send signals AS-IS	OEC Trader sends an order that is identical to the

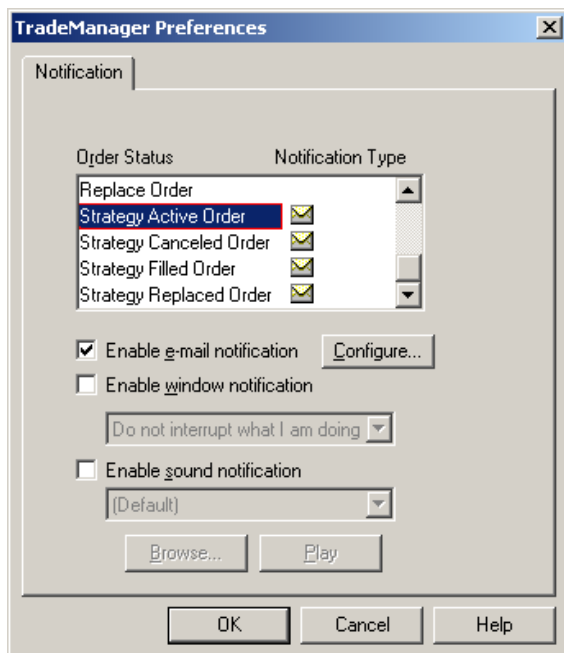
Feature	Description
	TradeStation™ Strategy order description, with exception to short order type (“sell short” is treated as “sell”, “buy to cover” is treated as “buy”). Because TradeStation™ order types and OEC Trader order types are filled in different environments, the TradeStation™ Strategy and OEC Trader positions could become unsynchronized.
Send AS-IS, but convert to market on Filled report	<p>The same as previous option, but a conditional order will be modified to a market order on a Filled report from TradeStation™, if an OEC order is not filled in selected timeout.</p> <p>Modify to Market handled as limit. Since our order gateway does not support modify requests to a Market order, this command would get rejected. As a workaround, this option will adjust the limit price to simulate a market order.</p>
Desynchronization Timeout	Displays an alert, if the OEC order has been filled, but the TradeStation™ Strategy order is still working (no incoming Filled report for this order).

Note: To decrease desynchronization of positions, a market order will be sent on a Filled report from TradeStation™ when the option “Send signals AS-IS” is selected. If the original order is rejected and the reason is related to trigger price, the rejection message will state “trigger price.”

TradeStation™

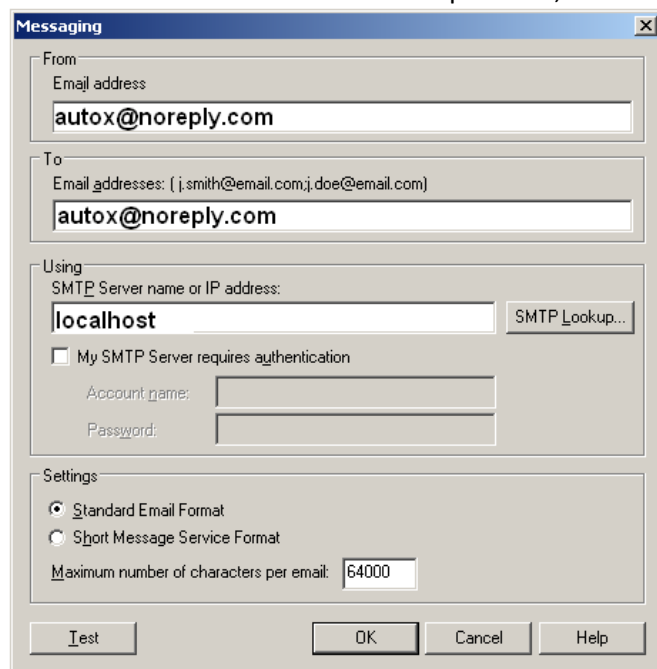
1. Open the TraderManager window in TradeStation™. Go to **File > New Window > Trade Manager** or use the navigational pane on the right side.
2. Open the Trader Manager Preferences via View in main menu of TradeStation™.
3. Set email notification for Strategy Active, Canceled, Filled and Replaced Order

4. Check "Enable e-mail notification" and press "Configure..."

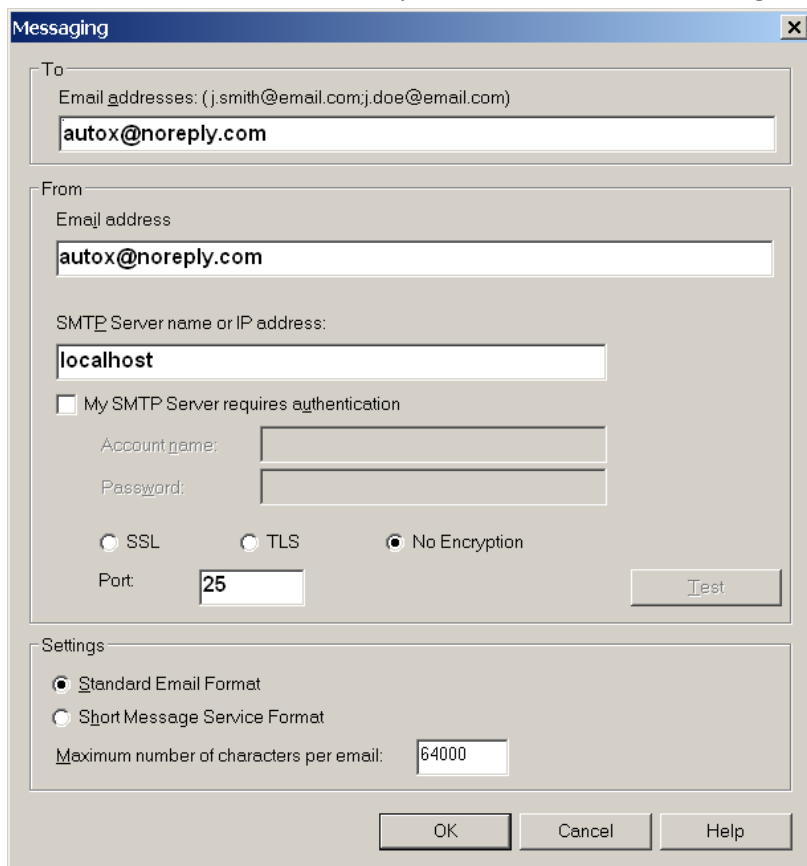


5. In the message dialog box enter "autox@noreply.com" for the "From" and "to" email address fields, and "LOCALHOST" or "127.0.0.1" in "SMTP Server name" field, see image on next page.

(For TradeStation™ versions before 9.1 Update 12, enter settings like below)



(For TradeStation™ version 9.1 Update 12 or later, enter settings like below)



Messaging

To
Email addresses: (j.smith@email.com;j.doe@email.com)
autox@noreply.com

From
Email address
autox@noreply.com

SMTP Server name or IP address:
localhost

☐ My SMTP Server requires authentication

Account name:
Password:

☐ SSL ☐ TLS ☒ No Encryption

Port: 25 Test

Settings
☒ Standard Email Format
☐ Short Message Service Format
Maximum number of characters per email: 64000

OK Cancel Help

Press OK and a prompt will appear to send a test message. You will see dialog two windows: one from TradeStation and one from OEC Trader. The message will confirm if the test has been sent and received. If the message is not received, please check Firewall and Anti-virus settings.

Go back into OEC Trader and go to **File > Plugins > AutoX > Capture Settings** to configure additional settings.

Messages and Notifications (optional)

It is possible to redirect all incoming messages to a 3rd party SMTP-enabled software or email accounts. Please check with your email provider for the host or server name, port, and if authentication is required. AutoX can send the connection status and internal error notifications to the selected email addresses. Additionally, it is possible to send SMS messages to cellular or mobile phones. The user can simply type their cell phone number in the "Send error and connection notifications to:" field. Use the following guide for assistance of what email address to use:

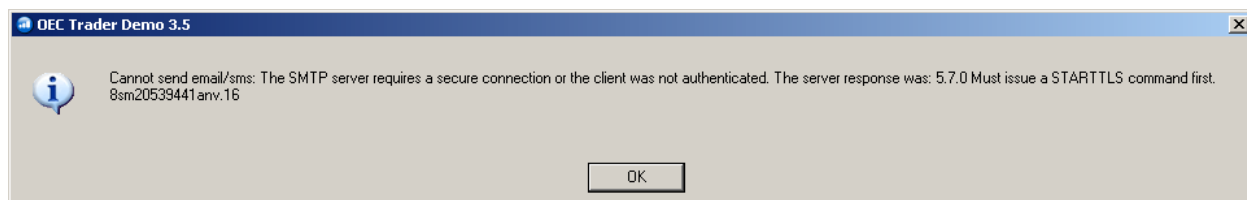
If your carrier is not listed, please check:

http://en.wikipedia.org/wiki/List_of_SMS_gateways

Carrier	Address
AT&T	10digitphonenumber@txt.att.net
Boost Mobile	10digitphonenumber@myboostmobile.com
CenturyLink (formerly CenturyTel and Qwest)	Many exist, check http://qwest.centurylink.com/internethelp/email-server-settings.html
Metro PCS	10digitphonenumber@mymetropcs.com
Nextel	10digitphonenumber@messaging.nextel.com
Sprint	10digitphonenumber@messaging.sprintpcs.com
T-Mobile	10digitphonenumber@tmomail.net
Tracfone	10digitphonenumber@txt.att.net
US Cellular	10digitphonenumber@email.uscc.net
Verizon	10digitphonenumber@vtext.com
Virgin Mobile	10digitphonenumber@vmobl.com

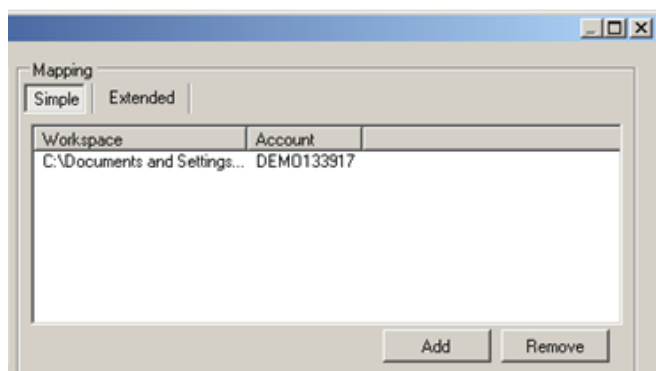
Message and Notification Limitations

The AutoX plugin currently only supports regular, non-secured SMTP connections. Secured connections like TLS (Transport Layer Security) and STARTTLS (as required by Gmail, for example) are not supported. If the mail server requires a SSL connection, you will get the following error:



Mapping (required)

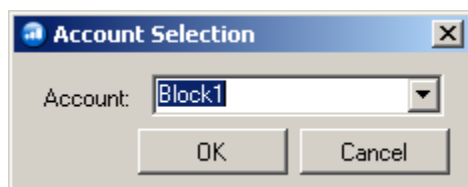
There are two modes of mapping: Simple and Extended. Simple mapping has two parts: one maps the workspace name to account and the other maps the symbols. Each TradeStation™ workspace is mapped separately. The default symbol map is used for extended mapping only.



Simple Mapping

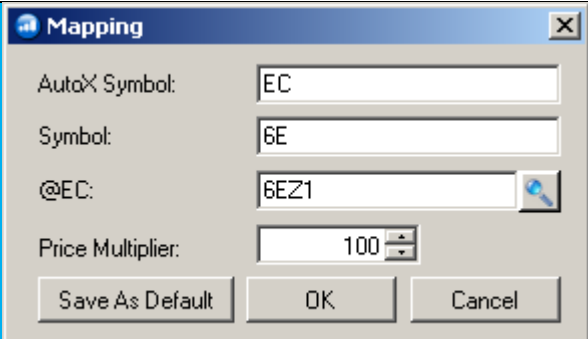
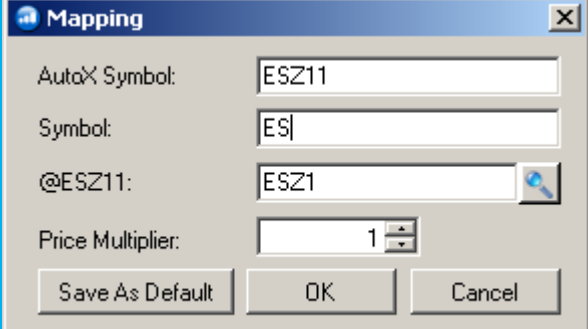
Account Selection

This window maps the workspace name of the incoming signal(s) to an account number. The signals from non-defined workspaces are ignored. Each workspace can only be assigned to one account or one allocation block name.



Symbol Mapping:

This window allows the mapping of the AutoX (TradeStation™) symbol to the OEC Trader symbol. Specifically, it maps corresponding TradeStation™ continuous symbols to specific OEC Trader symbols. In the event of quotation differences, you may also adjust the sent price by using a Price Multiplier to reflect the price quoted by OEC Trader. Each workspace has its own symbol mapping. If the symbol is not defined in the mapping, its signals will be ignored.

Price multiplier used	
Non-continuous mapping	

Continuous mapping

Mapping

AutoX Symbol: ES
Symbol: ES
@ES: ESZ1
Price Multiplier: 1

Save As Default OK Cancel

Extended Mapping

This mode uses the incoming workspace name, AutoX symbol, and the interval to map to the OEC Trader account and contract. To get started, click on the **Extended** tab and then click on **Add**.

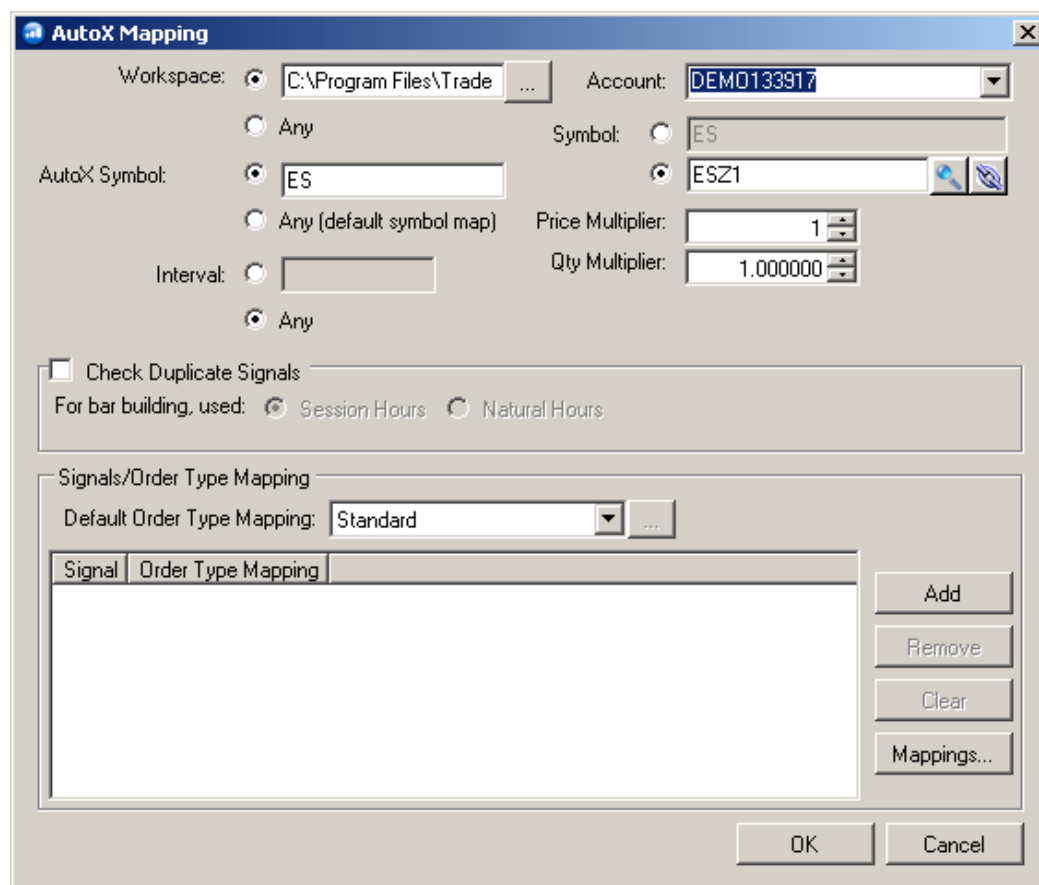
Mapping

Simple Extended

Workspace	Symbol	Interval	Account	Symbol	Price Mult.	Qty Mult.
C:\Program Files\Tr...	ESZ11	<Any>	DEM0133917	ESZ1	1	1

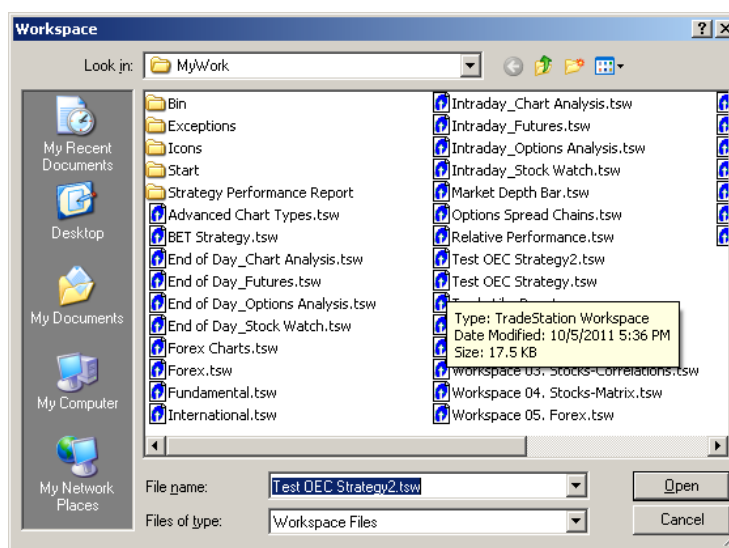
Add Edit Remove

There are several areas within extended mapping that can be configured.



Workspace

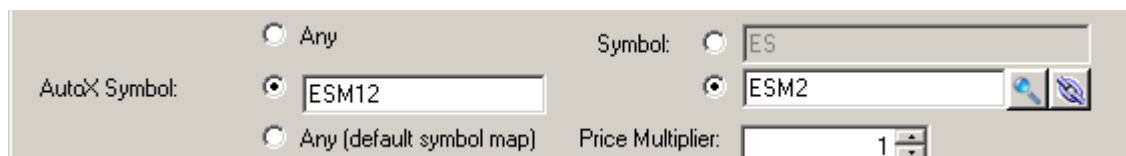
Pick the Workspace file used in TradeStation™. They are usually located in MyWork folder (eg. C:\Program Files\TradeStation 9.1\MyWork).



AutoX to OEC Symbol Mapping

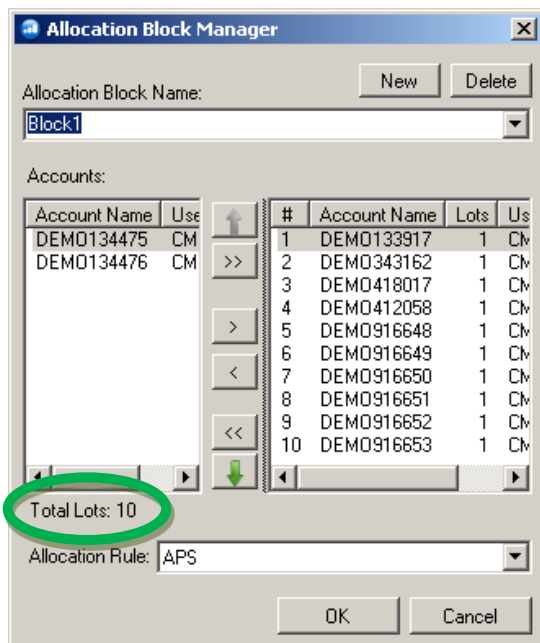
Enter the symbol used in the TradeStation™ platform as the **AutoX** symbol, eg. ESM12 and to the immediate right the OEC symbol, eg. ESM2.

- If you wish to map the AutoX symbol to the base OEC symbol, select the radio box next to **Symbol:**. Please be aware that this will map the AutoX symbol to the first month offered in the OEC data feed. This may not necessarily be the active month traded.
- If using the continuous symbol, it is usually mapped as ES and not @ES, even though your chart may display this formatting of symbol. The reason for this is the TradeStation™ email notification sends us ES instead of @ES. Be sure to use AutoX log window to see if adjustments will be necessary.
- If you wish to use the default symbol map, from prior screen, select **Any (default symbol map)**.

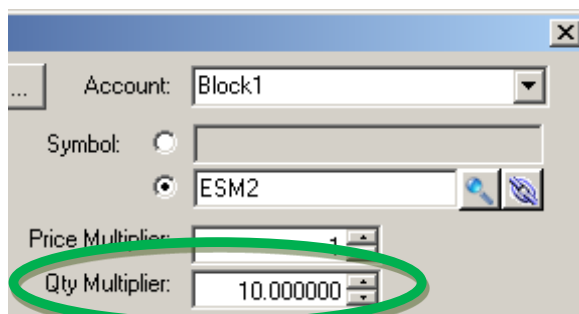


Quantity Multiplier (typically used for Allocation Block Trading)

This field allows you to increase the number of lots traded without having to adjust in TradeStation™. This is typically used for Allocation Blocks. When trading into an Allocation Block, you should match the **Total Lots** with **Qty Multiplier** and designate in account drop down the **Allocation Block** name. If you make changes to the block, you must change the AutoX setting as well.

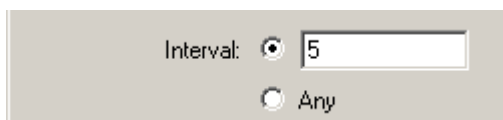


Account Name	Use	#	Account Name	Lots	Us
DEM0134475	CM	1	DEM0133917	1	CM
DEM0134476	CM	2	DEM0343162	1	CM
		3	DEM0418017	1	CM
		4	DEM0412058	1	CM
		5	DEM0916648	1	CM
		6	DEM0916649	1	CM
		7	DEM0916650	1	CM
		8	DEM0916651	1	CM
		9	DEM0916652	1	CM
		10	DEM0916653	1	CM



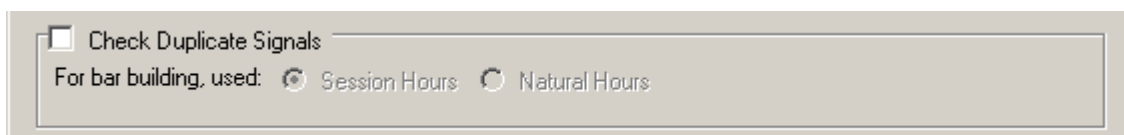
Interval:

This field is used to map the extended AutoX mapping to a specific minute interval used in the TradeStation™ chart. This could be helpful if you use the same workspace file with same symbol mapping but have a different strategy for a different interval. Most users will keep this setting as **Any**.



Interval: ☒ 5 ☐ Any

Check Duplicate Signals

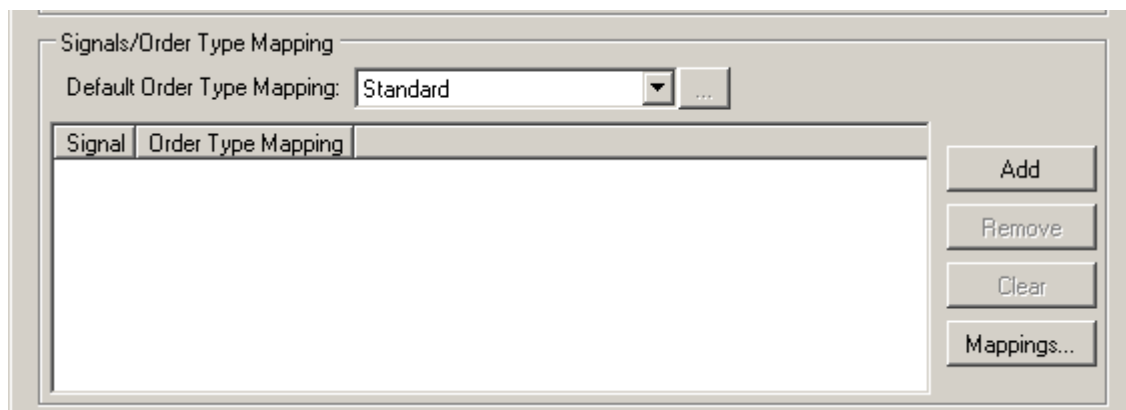


☐ Check Duplicate Signals
For bar building, used: ☒ Session Hours ☐ Natural Hours

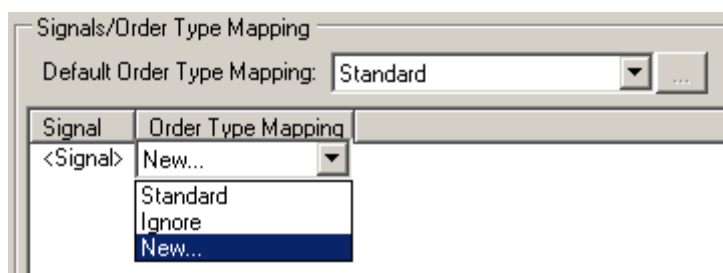
Signals/Order Type Mapping

This section allows mapping order types that are not recognized by OEC but are in TradeStation™. An example of this would Buy to Cover or Sell Short.

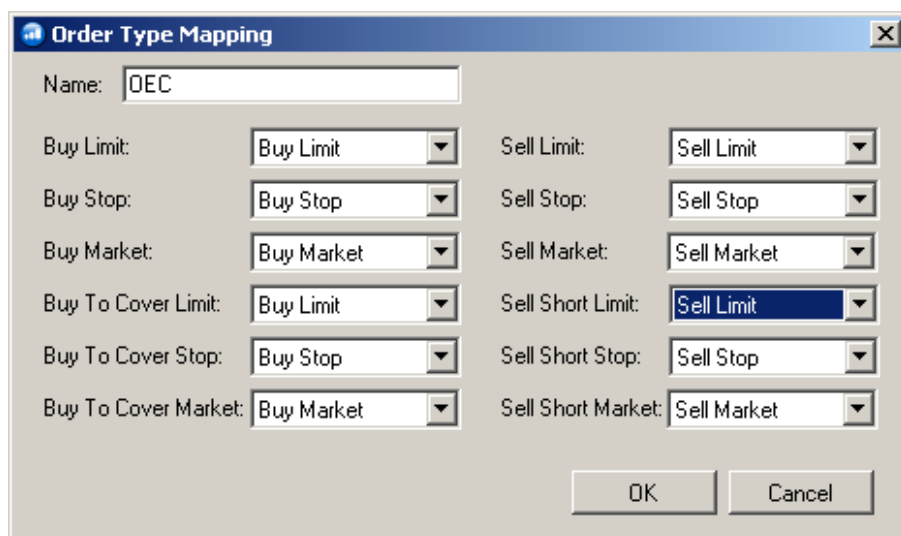
To add a new order type mapping for all or particular signal names, click on **Add**. To view current mappings, click on **Mappings**.



When you add a new order type mapping, a new entry will appear. Pick the signal name from the drop down list and select **New...** under **Order Type Mapping** in the drop down list.



Name the order type mapping and choose how OEC Trader should send the order for execution. (Eg. Send Buy-to-Cover Limit as Buy Limit).



Below is an example mapping **BUY** and **SELL** signal using the custom OEC mapping and the **EXIT** signal using the Standard mapping.

AutoX Mapping

Workspace: ☒ C:\Program Files\Trade ... Account: Block1

AutoX Symbol: ☒ ESM12 ☐ Any ☐ Any (default symbol map) Symbol: ☐ ☒ ESM2

Interval: ☒ 5 ☐ Any Price Multiplier: 1 Qty Multiplier: 10.000000

☐ Check Duplicate Signals

For bar building, used: ☒ Session Hours ☐ Natural Hours

Signals/Order Type Mapping

Default Order Type Mapping: OEC

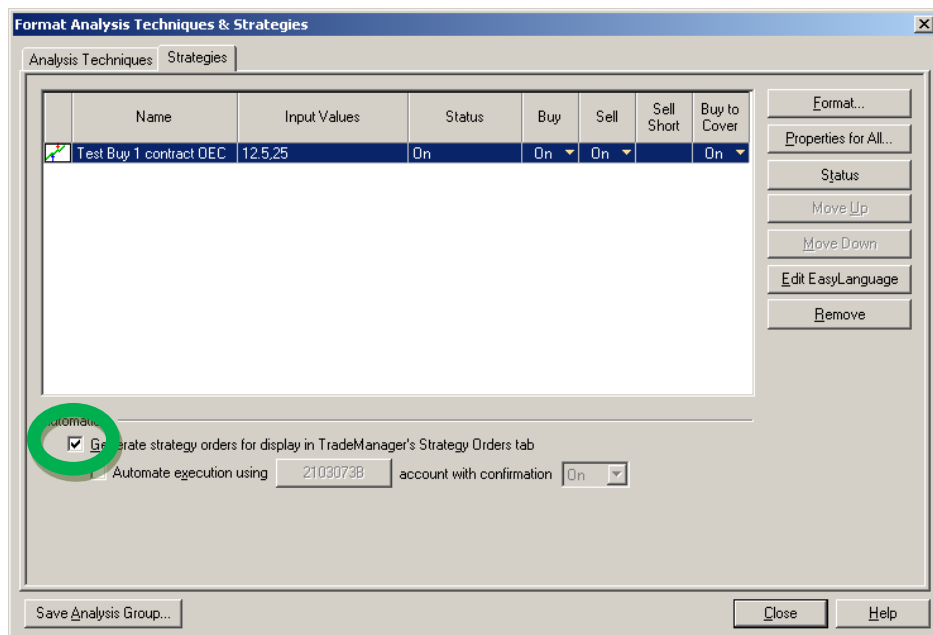
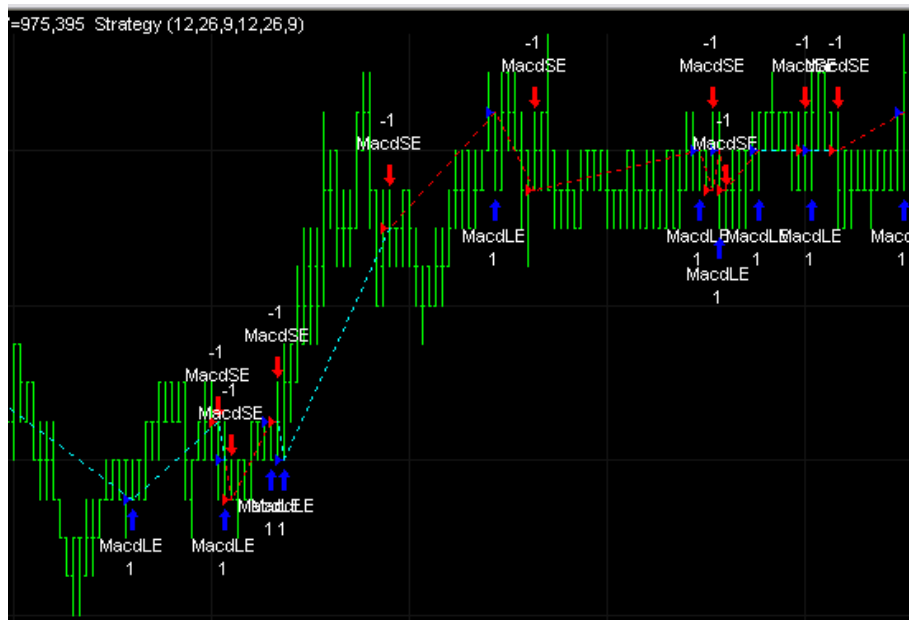
Signal	Order Type Mapping
BUY	OEC
SELL	OEC
EXIT	Standard

Add Remove Clear Mappings...

OK Cancel

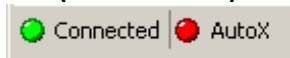
Process

1. Insert the EasyLanguage strategy in the "Chart Analysis" window in TradeStation™. To enable the Strategy in TradeStation™, right click in chart and go to **Format Strategies**. Then, activate Generate strategy orders...

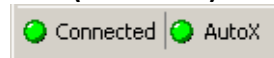


2. Login to OEC Trader and ensure AutoX is enabled. Go to **File > Plugins > AutoX > Capture Activated (checked)**

(disconnected)



(connected)



- Catch and confirm (if "Manual Confirmation" is checked) the signals from TradeStation™ in OEC Trader. The signal name and interval of the base chart is located in comments of the OEC Trader order.

Account	Order #	State	Side	Qty	Symbol	Price	Avg. Price	Comments	Time
VIC01	1337674	Completed	Buy	1/1	YMM7	MKT FOK	12973	MA3CrSE, 1 min.	09:53:12.103
VIC01	1337675	Completed	Sell	1/1	YMM7	MKT FOK	12972	CustomLX, 1 min.	09:53:13.009
VIC01	1337676	Completed	Sell	1/1	ER2M7	MKT	831.8	Profit Target, 1 min.	09:53:17.025
VIC01	1337677	Completed	Buy	1/1	ER2M7	MKT	831.8	KeyRevLE, 1 min.	09:54:09.681
VIC01	1337678	Completed	Sell	1/1	YMM7	MKT FOK	12969	CustomSE (short), 1 min.	09:54:10.916
VIC01	1337679	Completed	Buy	1/1	YMM7	MKT FOK	12971	EbDirSX (to cover), 1 min.	09:54:14.103
VIC01	1337680	Completed	Sell	1/1	ER2M7	MKT	831.9	Profit Target, 1 min.	09:54:15.634
VIC01	1337681	Completed	Buy	1/1	ER2M7	MKT	832.0	MomLE, 1 min.	09:54:16.572
VIC01	1337682	Completed	Sell	1/1	ER2M7	MKT	831.8	Stop Loss, 1 min.	09:54:17.134
VIC01	1337683	Completed	Buy	1/1	ER2M7	MKT	831.6	NewHi, 1 min.	09:54:34.447
VIC01	1337684	Completed	Sell	1/1	ER2M7	MKT	831.7	Profit Target, 1 min.	09:54:36.056
VIC01	1337685	Completed	Buy	1/1	ER2M7	MKT	831.8	PChLE, 1 min.	09:54:36.931
VIC01	1337686	Completed	Sell	1/1	ER2M7	MKT	831.7	Stop Loss, 1 min.	09:54:37.291
VIC01	1337687	Completed	Buy	1/1	YMM7	MKT FOK	12969	NewHi, 1 min.	09:54:47.447
VIC01	1337688	Completed	Sell	1/1	YMM7	MKT FOK	12969	Stop Loss, 1 min.	09:54:47.759

Log

A log of TradeStation™ interaction is stored to file and can be viewed inside the OEC Trader using the AutoX Log window. To view this window go to: **File > Plugins > AutoX > Log**

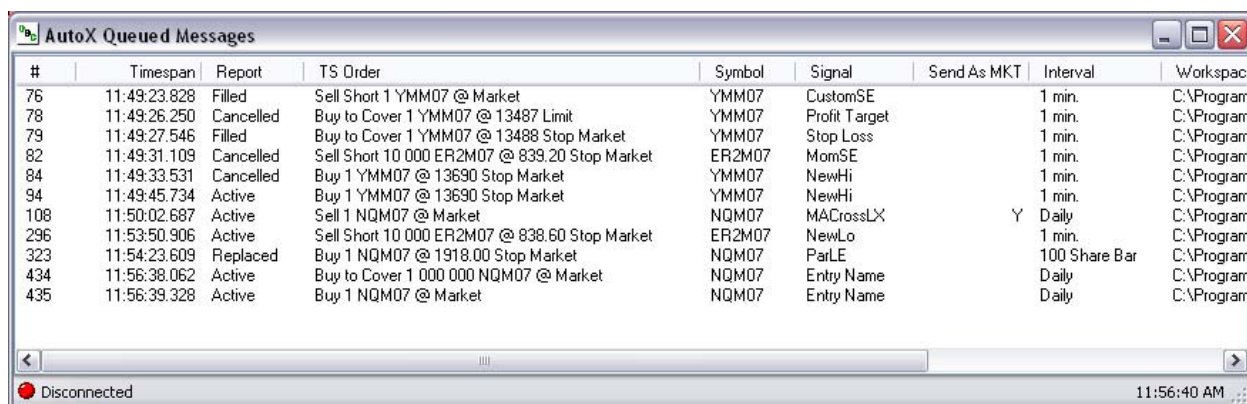
Timespan	Level	Report	Message	TS Order	Order #	OEC Order	Contract	Signal	Interval	Wo
11:47:52.593	Info		Starting							
11:47:53.093	Info	Cancelled	Cancel sent	Sell Short 1 YMM07 @ 13483 Stop Market	1359059	Sell 1 YMM7 STP 13483	YMM7	NewLo	1 min.	C/V
11:47:54.343	Info	Active	Sent	Sell Short 1 YMM07 @ Market	1359079	Sell 1 YMM7 MKT	YMM7	CustomSE	1 min.	C/V
11:47:55.546	Info	Active	Sent	Buy 1 YMM07 @ 13690 Stop Market	1359080	Buy 1 YMM7 STP 13690	YMM7	NewHi	1 min.	C/V
11:47:56.765	Info	Active	Sent	Buy to Cover 1 YMM07 @ 13493 Stop Market	1359081	Buy 1 YMM7 STP 13493	YMM7	Stop Loss	1 min.	C/V
11:47:57.968	Info	Active	Sent	Buy to Cover 1 YMM07 @ 13492 Limit	1359082	Buy 1 YMM7 LMT 13492	YMM7	Profit Target	1 min.	C/V
11:47:58.593	Info		Possible desynchronization		1359082	Buy 1 YMM7 LMT 13492	YMM7			
11:47:59.203	Info	Filled	Filled is skipped because of market order	Sell Short 1 YMM07 @ Market			YMM07	CustomSE	1 min.	C/V
11:48:00.437	Info	Active	Sent	Sell Short 1 YMM07 @ Market	1359083	Sell 1 YMM7 MKT	YMM7	MACrossSE	1 min.	C/V
11:48:01.540	Error	Cancelled	Non-working order: Completed	Buy to Cover 1 YMM07 @ 13492 Limit	1359082	Buy 1 YMM7 LMT 13492	YMM7	Profit Target	1 min.	C/V
11:48:02.859	Info	Filled	Convert to market-on-filled in 10 secs	Buy to Cover 1 YMM07 @ 13493 Stop Market	1359081	Buy 1 YMM7 STP 13493	YMM7	Stop Loss	1 min.	C/V
11:48:04.078	Info	Active	Sent	Buy to Cover 1 YMM07 @ 13493 Stop Market	1359084	Buy 1 YMM7 STP 13493	YMM7	Stop Loss	1 min.	C/V
11:48:05.312	Info	Active	Sent	Buy to Cover 1 YMM07 @ 13492 Limit	1359085	Buy 1 YMM7 LMT 13492	YMM7	Profit Target	1 min.	C/V
11:48:06.703	Info		Possible desynchronization		1359085	Buy 1 YMM7 LMT 13492	YMM7			
11:48:06.500	Info	Filled	Filled is skipped because of market order	Sell Short 1 YMM07 @ Market			YMM07	MACrossSE	1 min.	C/V
11:48:07.750	Info	Active	Sent	Sell Short 1 YMM07 @ Market	1359086	Sell 1 YMM7 MKT	YMM7	MA3CrSE	1 min.	C/V
11:48:08.953	Error	Cancelled	Non-working order: Completed	Buy to Cover 1 YMM07 @ 13492 Limit	1359085	Buy 1 YMM7 LMT 13492	YMM7	Profit Target	1 min.	C/V
11:48:10.156	Info	Filled	Convert to market-on-filled in 10 secs	Buy to Cover 1 YMM07 @ 13493 Stop Market	1359081	Buy 1 YMM7 STP 13493	YMM7	Stop Loss	1 min.	C/V
11:48:11.296	Info	Active	Sent	Buy to Cover 1 YMM07 @ 13493 Stop Market	1359087	Buy 1 YMM7 STP 13493	YMM7	Stop Loss	1 min.	C/V
11:48:12.500	Info	Active	Sent	Buy to Cover 1 YMM07 @ 13492 Limit	1359088	Buy 1 YMM7 LMT 13492	YMM7	Profit Target	1 min.	C/V
11:48:12.859	Info		Market-on-Filled modify is sent		1359081	Buy 1 YMM7 STP 13493	YMM7			
11:48:12.875	Info		Possible desynchronization		1359088	Buy 1 YMM7 LMT 13492	YMM7			
11:48:13.718	Info	Filled	Filled is skipped because of market order	Sell Short 1 YMM07 @ Market			YMM07	MA3CrSE	1 min.	C/V
11:48:14.937	Info	Active	Sent	Sell Short 1 YMM07 @ 13491 Stop Market	1359090	Sell 1 YMM7 STP 13491	YMM7	KIChSE	1 min.	C/V
11:48:16.140	Error	Cancelled	Non-working order: Completed	Buy to Cover 1 YMM07 @ 13492 Limit	1359088	Buy 1 YMM7 LMT 13492	YMM7	Profit Target	1 min.	C/V
11:48:17.265	Info	Filled	Convert to market-on-filled in 10 secs	Buy to Cover 1 YMM07 @ 13493 Stop Market	1359084	Buy 1 YMM7 STP 13493	YMM7	Stop Loss	1 min.	C/V
11:48:19.518	Info	Active	Sent	Buy to Cover 1 YMM07 @ 13492 Limit	1359091	Buy 1 YMM7 LMT 13492	YMM7	Profit Target	1 min.	C/V

Queued Messages

All TradeStation™ messages are queued in the case of a disconnection to OEC Servers and pushed to execution after a successful reconnection. In addition, OEC Trader pre-processes the queued messages with following rules:

1. If queue contains Active message and corresponding Cancel is incoming, **no order will be sent on reconnection.**
2. If queue contains Active message and corresponding Replace is incoming, **modified version of order will be issued instead of original one**
3. If queue contains Active or Replace message and corresponding Filled is incoming, **market order will be issued.**
4. If queue contains two Active messages with equal quantity, workspace, interval, symbol and both marked as "Send as MKT", **these orders will be removed from queue ("closed in order received")**

To view the current queued messages, go to **File > Plugins > AutoX > Queued Orders**.



#	Timespan	Report	TS Order	Symbol	Signal	Send As MKT	Interval	Workspace
76	11:49:23.828	Filled	Sell Short 1 YMM07 @ Market	YMM07	CustomSE		1 min.	C:\Program
78	11:49:26.250	Cancelled	Buy to Cover 1 YMM07 @ 13487 Limit	YMM07	Profit Target		1 min.	C:\Program
79	11:49:27.546	Filled	Buy to Cover 1 YMM07 @ 13488 Stop Market	YMM07	Stop Loss		1 min.	C:\Program
82	11:49:31.109	Cancelled	Sell Short 10 000 ER2M07 @ 839.20 Stop Market	ER2M07	MomSE		1 min.	C:\Program
84	11:49:33.531	Cancelled	Buy 1 YMM07 @ 13690 Stop Market	YMM07	NewHi		1 min.	C:\Program
94	11:49:45.734	Active	Buy 1 YMM07 @ 13690 Stop Market	YMM07	NewHi		1 min.	C:\Program
108	11:50:02.687	Active	Sell 1 NQM07 @ Market	NQM07	MACrossLX	Y	Daily	C:\Program
296	11:53:50.906	Active	Sell Short 10 000 ER2M07 @ 838.60 Stop Market	ER2M07	NewLo		1 min.	C:\Program
323	11:54:23.609	Replaced	Buy 1 NQM07 @ 1918.00 Stop Market	NQM07	ParLE		100 Share Bar	C:\Program
434	11:56:38.062	Active	Buy to Cover 1 000 000 NQM07 @ Market	NQM07	Entry Name		Daily	C:\Program
435	11:56:39.328	Active	Buy 1 NQM07 @ Market	NQM07	Entry Name		Daily	C:\Program

Disconnected 11:56:40 AM

TradeStation™ Email format:

Example of a new order email notification:

TradeStation - Strategy Active Order for NQM07

Order: Buy 1 NQM07 @ 1834.00 Stop Market

Account:

Occurred: 4/13/2007 4:24:06 PM

Signal: NewHi

Interval: 1 min.

Workspace: C:\Program Files\TradeStation 8.2 (Build 3894)\MyWork\OECTEST

Example of a cancel request notification:

TradeStation - Strategy Canceled Order for NQM07

Order: Buy to Cover 1 NQM07 @ 1834.00 Stop Market

Account:

Occurred: 4/13/2007 4:27:06 PM

Signal: NewHi

Interval: 1 min.

Workspace: C:\Program Files\TradeStation 8.2 (Build 3894)\MyWork\OECTEST

Optional Reading

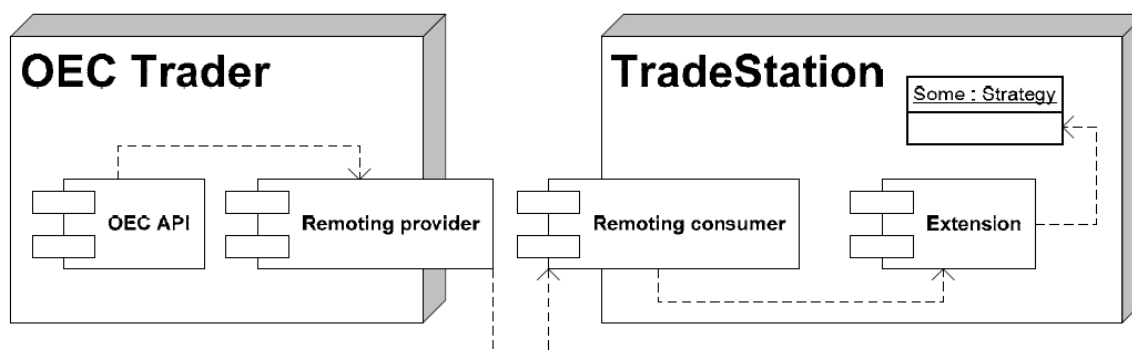
OEC API functionality inside EasyLanguage

General

Implemented in OEC Trader is a TradeStation™ extension that provides the full functionality of OEC API inside TradeStation™ strategy scripts. The extension has 457 functions that reflect all objects, properties and methods of OEC API. OEC API code is called from the extension indirectly. Actually the OEC API is hosted by OEC Trader and exposes OEC API functionality to TradeStation™. The extension allows a strategy to avoid the limitations of TradeStation™. The TradeStation™ strategy will have access to OEC Trader average positions, account balances, basic and advanced orders, handle failure situations, avoid synchronization troubles, and etc. In other words, TradeStation™ + extension provide one more programming environment that can be used by TradeStation™-related programmers instead of .Net environment. This extension can be used independently of the email notification capture feature (send orders through extension instead of TradeStation™ constructions) as well as together. A strategy could get extension functions to obtain positions and balances, but use TradeStation™ construction to send orders.

Infrastructure

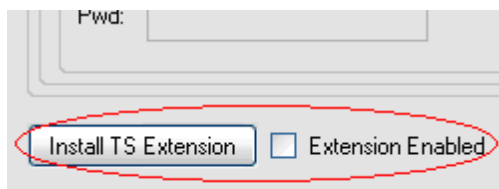
The scheme below shows the process boundary, inter-process communication via .NET remoting, and data flow.



So by this scheme, OEC Trader shows all order manipulations real-time from the TradeStation's strategy and vice versa. The strategy can take into account user operations, real OEC Trader positions and prices.

Installation and Settings

Open AutoX Capture Settings dialog to find "Install TS Extension" and "Extension Enabled" at the bottom left of the window.



Installation

TradeStation™ Extension consists from of files (OecFuncs.dll and TSGateway.dll) and is installed to OEC Trader folder at the time of the OEC Trader installation. At this time these files are placed on the users computer, but not integrated into TradeStation™. A user can install and integrate this extension to TradeStation™ environment via the “Install TS Extension” button. The required files will be copied to TradeStation™ folder. Now “Extension” and “Remoting consumer” components from [scheme](#) above are installed and a user can export OEC functions from the extension and verify strategies.

Enabling

In spite of the components installed, the link between “Remoting provider” and “Remoting consumer” is not established yet (see [scheme](#) above). “Extension enabled” option responds only when the Extensions Enabled checkbox is selected.

Error Processing

The TradeStation extension supports native TradeStation events and error processing mechanisms that protects TradeStation from crashing. All exceptions are captured and shown in TradeStation Event Manager.

OEC Extension Functions

All functions have the same pattern and use OEC API name convention.

1. Selection of objects for further operations

```
oecSelectXXX(name of object);
oecSelectXXXByID(id of object);
oecSelectXXXByIndex(index in collection)
where XXX is OEC API object.
```

Sample:

```
* oecSelectAccount("TEST001")    → select and use TEST001 in further operations
* oecSelectContractGroupByIndex(0) → select and use first contract group from collection of groups
```

2. Access to OEC API data and methods

After object selection, strategy can read its data and call its methods. The pattern of getter functions and methods is:

oecXXXXYYYZZZ(parameters)
 where XXX – name of OEC API object (optionally);
 YYY – name of nested OEC API object (optionally)
 ZZZ – name of OEC API method or property.

Sample:

* oecIsCompleteConnected() → OEC.API.OECClient.Global. CompleteConnected
 * oecAccountName() → OEC.API.Account.Name. Account should be selected before by means of any oecSelectAccount***() function.

Example of strategy

This example code shows the use of different functions from different areas of the OEC Trader TradeStation™ Extension (contract listing, average positions, sending orders ((simple, OCO, brackets)), allocation block construction/usage, canceling and modifying).

```
{ Import external OEC API functions }
External: "OecFuncs.dll", bool, "oecIsCompleteConnected";
External: "OecFuncs.dll", int, "oecAvgPositionNetVolume";
External: "OecFuncs.dll", double, "oecAvgPositionNetPrice";
External: "OecFuncs.dll", bool, "oecSelectAccountByIndex", int{{AccountIndex}};
External: "OecFuncs.dll", bool, "oecSelectContract", LPSTR{{symbol}};
External: "OecFuncs.dll", LPSTR, "oecAvgPositionContract";
External: "OecFuncs.dll", double, "oecAvgPositionGain";
External: "OecFuncs.dll", LPSTR, "oecAccount";
External: "OecFuncs.dll", void, "oecFreeze";
External: "OecFuncs.dll", void, "oecUnfreeze";
External: "OecFuncs.dll", bool, "oecSelectContractByIndex", int{{ContractIndex}};
External: "OecFuncs.dll", int, "oecContractsCount";
External: "OecFuncs.dll", LPSTR, "oecContractPositionSymbol";
External: "OecFuncs.dll", LPSTR, "oecContractName";
External: "OecFuncs.dll", LPSTR, "oecContract";
External: "OecFuncs.dll", LPSTR, "oecCreateDraft", LPSTR{{Side}}, int{{Qty}}, LPSTR{{Type}},
double{{Price}}, double{{Price2}}, LPSTR{{Flags}}, LPSTR{{Comments}};
External: "OecFuncs.dll", LPSTR, "oecSetDraftTime", LPSTR{{Draft}}, double{{Start}},
double{{End}};
External: "OecFuncs.dll", LPSTR, "oecCheckDraft", LPSTR{{Draft}};
External: "OecFuncs.dll", bool, "oecSendOrder", LPSTR{{Draft}};
External: "OecFuncs.dll", LPSTR, "oecSetDraftIceberg", LPSTR{{Draft}}, int{{Volume}};
External: "OecFuncs.dll", double, "oecCurrentPriceLastPrice";
External: "OecFuncs.dll", int, "oecOrdersCount";
External: "OecFuncs.dll", bool, "oecIsOrderIsFinalState";
External: "OecFuncs.dll", bool, "oecCancelOrder";
External: "OecFuncs.dll", int, "oecOrder";
External: "OecFuncs.dll", bool, "oecSelectOrderByIndex", int{{OrderIndex}};
External: "OecFuncs.dll", bool, "oecSendOCO", LPSTR{{Draft1}}, LPSTR{{Draft2}};
External: "OecFuncs.dll", bool, "oecSendLinked2", LPSTR{{MainDraft}}, LPSTR{{LinkedDraft1}},
LPSTR{{LinkedDraft2}};
External: "OecFuncs.dll", bool, "oecSelectAvgPositionByIndex", int{{AvgPositionIndex}};
External: "OecFuncs.dll", int, "oecAvgPositionsCount";
External: "OecFuncs.dll", LPSTR, "oecAvgPositionPositionContract";
External: "OecFuncs.dll", double, "oecTotalBalanceNetLiquidatingValue";
External: "OecFuncs.dll", LPSTR, "oecContractType";
External: "OecFuncs.dll", void, "oecInitializeAllocationBlock", LPSTR{Name}, LPSTR{Rule},
LPSTR{AccountLotsList};
External: "OecFuncs.dll", int, "oecAccountsCount";
External: "OecFuncs.dll", LPSTR, "oecAccountCH";
```

```

External: "OecFuncs.dll", LPSTR, "oecAccountType";
External: "OecFuncs.dll", LPSTR, "oecAccountTrader";
External: "OecFuncs.dll", bool, "oecModify", int{Quantity}, LPSTR{Type}, double{Price},
double{Price2};
External: "OecFuncs.dll", int, "oecOrderQuantity";
External: "OecFuncs.dll", LPSTR, "oecOrderType";
External: "OecFuncs.dll", int, "OnCreateProc", IEasyLanguageObject;
External: "OecFuncs.dll", int, "OnDestroyProc", IEasyLanguageObject;
#Events
OnCreate = OnCreateProc;
OnDestroy = OnDestroyProc;
#End;

variables: counter(0), qty(0), price(0.0), draft(""), draft2(""), draft3(""), AB(""), ABSize(0);
if LastBarOnChart then
begin

    { Freeze OEC Trader and OEC API data during strategy body calculations }
    oecFreeze();

    { Check connection status of OEC Trader }
    if oecIsCompleteConnected() then
    begin
        AB = "";
        ABSize = 0;
        qty = 0;

        { Select and use ESM7 hereinafter }
        oecSelectContract("ESM7");
        for counter = 0 to oecAccountsCount() - 1
        begin
            oecSelectAccountByIndex(counter);
            Print(oecAccount(), ", ", oecAccountCH(), ", ", oecAccountType(), ", ",
oecAccountTrader());
            if oecAccountCH() = "RJO" and oecAccountType() = "Customer" and
oecAccountTrader() = "vic" then begin
                if AB <> "" then
                    AB = AB + ",";
                AB = AB + oecAccount() + "=1";
                ABSize = ABSize + 1;
                qty = qty + oecAvgPositionNetVolume();
            end;
        end;

        Print("AB: ", AB, ", Size: ", ABSize, "; sum pos: ", qty);
        if qty < 0 then
            ABSize = - ABSize;

        { Use first account }
        oecSelectAccountByIndex(0);

        { list and prints all available contracts }
        For counter = 0 To oecContractsCount()-1 Begin
            oecSelectContractByIndex(counter);
            Print(oecContract(), ": ", oecContractName(), " ", oecContractType());
        end;

        for counter = 0 to oecAvgPositionsCount()-1
        begin
            oecSelectAvgPositionByIndex(counter);
            Print("Position: ", oecAvgPositionPositionContract(),
                "; Volume:", oecAvgPositionNetVolume():10,
                "; Avg.Price:", oecAvgPositionNetPrice():10,
                "; Gain: ", oecAvgPositionGain());
        end;
    end;
end;

```

```

Print("Net Liq. Value: ", oecTotalBalanceNetLiquidatingValue():10:2);

{ Select and use ESM7 hereinafter }
oecSelectContract("ESM7");

{ Print ESM7 average position }
qty = oecAvgPositionNetVolume();
price = oecAvgPositionNetPrice();
Print("Selected ESM7");
Print("Net Volume:", qty,
      "; Avg.Price:", price,
      "; Contract: ", oecAvgPositionContract(),
      ", Account: ", oecAccount(),
      "; Gain: ", oecAvgPositionGain());

{ Cancel all working orders }
for counter=0 to oecOrdersCount()-1 Begin
  oecSelectOrderByIndex(counter);
  if oecIsOrderIsFinalState() = false then
  begin
    if oecOrderType() = "Limit" then
      oecModify(oecOrderQuantity() * 2, "", 0, 0);
    if(oecCancelOrder()) then
      Print("Canceling order #", oecOrder());
    end;
  end;
end;

{ Unfreeze OEC Trader and API and use local variables hereinafter }
oecUnfreeze();

if oecIsCompleteConnected() then
begin
  if qty > 0 then
    draft = "Sell"
  else
    draft = "Buy";

  { Create order draft. draft is a string variable with description of order }
  draft = oecCreateDraft(draft, 2 * AbsValue(qty), "Iceberg", (oecCurrentPriceLastPrice() +
price)/2, 0, "None", "Reverse By TS");

  { Set Start time. End time stays default }
  draft = oecSetDraftTime(draft, ComputerDateTime() + 1, 0);

  { Set Iceberg data. Volume = 1 }
  draft = oecSetDraftIceberg(draft, 1);
  Print(draft);

  { Check draft data. oecCheckDraft should return empty string or invalid parts of order }
  Print("Invalid parts: ", oecCheckDraft(draft));

  { Send order }
  if(oecSendOrder(draft)) then
    Print("Sent successfully")
  else
    Print("Did not send");

  { Prepare other two drafts of OCO }
  draft = oecCreateDraft("Buy", 2, "Limit", oecCurrentPriceLastPrice() - 1, 0, "", "");
  draft2 = oecCreateDraft("Buy", 2, "Stop", oecCurrentPriceLastPrice() + 1, 0, "", "");

  { And send OCO }
  if(oecSendOCO(draft, draft2)) then
    Print ("OCO is sent");

```

```

    draft3 = oecCreateDraft("Sell", 2, "Limit", oecCurrentPriceLastPrice() + 0.5, 0,"",
"Main");

    { And bracket }
    if oecSendLinked2(draft3, draft, draft2) then
        Print ("Bracket is sent");
    if AB <> "" then begin
        if ABSIZE > 0 then
            draft = "Sell"
        else begin
            draft = "Buy";
            ABSIZE = - ABSIZE;
        end;
    draft = oecCreateDraft(draft, ABSIZE, "Market", 0, 0, "None", "Test of AB from TS");

    { Initialize Allocation Block structure before sending }
    oecInitializeAllocationBlock("TS AB", "LowAcctHighPrice", AB);
    if oecCheckDraft(draft) <> "" then
        Print("Invalid AB order: ", oecCheckDraft(draft))
    else
        begin
            if oecSendOrder(draft) then
                Print("AB order is sent")
            else
                Print("AB order is not sent");
            end;
            { Reset the using of allocation block }
            oecInitializeAllocationBlock("", "", "");
        end;
    end;
else
    Print("OEC is not connected");
end;

```

Help

There is a generated HTML file with short documentation on using the OEC API TradeStation™ function located at <http://www.openecry.com/trading/software-highlights/index.cfm>.

Format of declaration:

1. Original OEC API hierarchy
2. Name of function without "oec" prefix
3. Line for copy-pasting to strategy code to import function
4. Implemented .NET code in OEC API terms.

Sample:

Client.Accounts.SelectAccount

SelectAccount

```
External: "OecFuncs.dll", void, "oecSelectAccount", LPSTR{{Account}};  
_Account = Client.Accounts[Account];
```

Client.Accounts.SelectAccountByID

SelectAccountByID

```
External: "OecFuncs.dll", void, "oecSelectAccountByID", int{{AccountID}};  
_Account = Client.Accounts[AccountID];
```

Client.Accounts.SelectAccountByIndex

SelectAccountByIndex

```
External: "OecFuncs.dll", void, "oecSelectAccountByIndex",  
int{{AccountIndex}};  
_Account = Client.Accounts.Values[AccountIndex];
```