



TRADESTATION INTERFACE VIA EMAIL NOTIFICATION	2
TRADESTATION AND OEC TRADER SETTINGS	2
<i>System requirements</i>	2
<i>TradeStation</i>	2
<i>OEC Trader</i>	4
Messages and Notifications.....	6
Mapping	7
Extension.....	8
PROCESS.....	8
LOG	10
QUEUED MESSAGES	10
TRADESTATION EMAIL FORMAT	11
OEC API FUNCTIONALITY INSIDE EASLANGUAGE.....	11
GENERAL	11
INFRASTRUCTURE	12
INSTALLATION AND SETTINGS.....	12
<i>Installation</i>	12
<i>Enabling</i>	12
ERROR PROCESSING	13
OEC EXTENSION FUNCTIONS.....	13
EXAMPLE OF STRATEGY	14
HELP.....	17

TradeStation Interface via email notification

OEC Trader captures TradeStation signals and parses the email notifications coming from TradeStation. OEC Trader supports all type of email strategy notification; all types of orders (market, limit and stop); optional manual confirmation; symbol mapping; advanced account assignment; redirection email notification to user's email client. In order for signals to be captured from TradeStation the local SMTP port cannot be busy. Moreover, only one OEC Trader on the computer can capture signals.

TradeStation and OEC Trader Settings

System requirements

1. If using Windows Professional or above, please be sure that Windows SMTP service is disabled.
2. Open the Control Panel from Windows Start menu and go to the Administrative Tools > Services application. Disable the Windows SMTP service, if it is installed and started: use a right click to open "Properties" of this service. Change "Startup type" to "Disabled" and press "Stop" and OK.
3. If you use an SMTP service of another provider, please stop this service or re-configure it to not use 25 port.
4. Be sure to close all other TradeStation capture applications. These applications use port 25 too.
5. Check your firewall settings. The SMTP port should not be prohibited for local use.
6. Only one instance of OEC Trader can capture TradeStation signals at one.

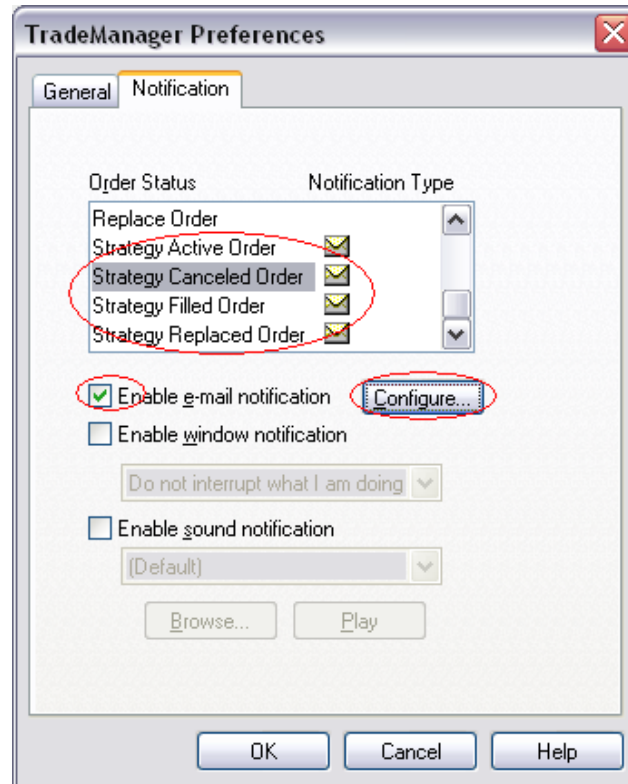
TradeStation

1. Open the TraderManager Tools in TradeStation

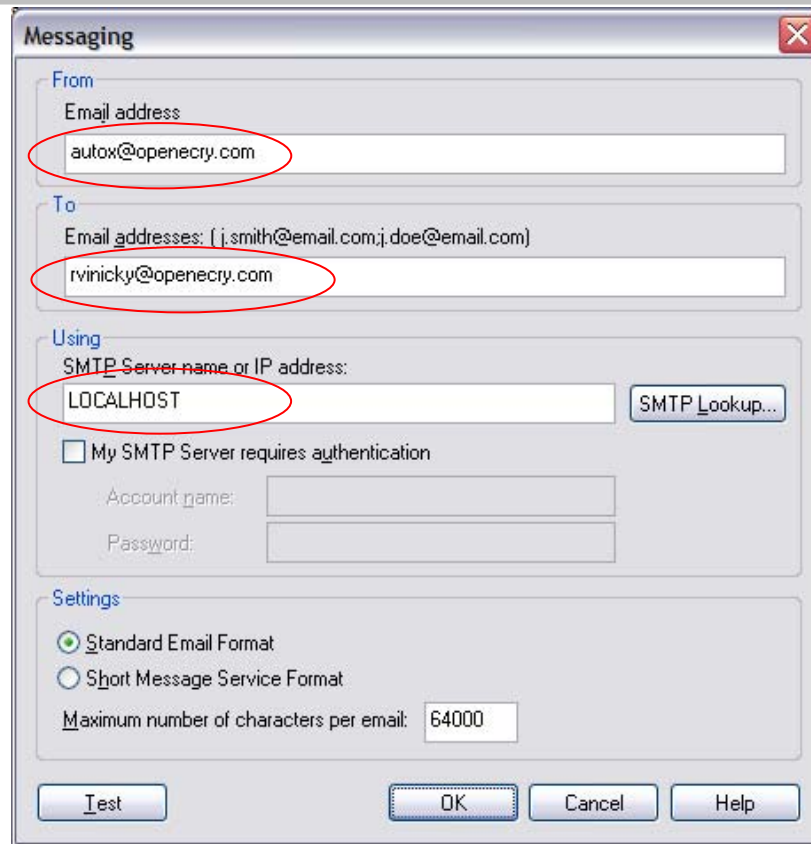


2. Open the TraderManager Preferences via View in main menu of TradeStation.
3. Set email notification for Strategy Active, Canceled, Filled and Replaced Order

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

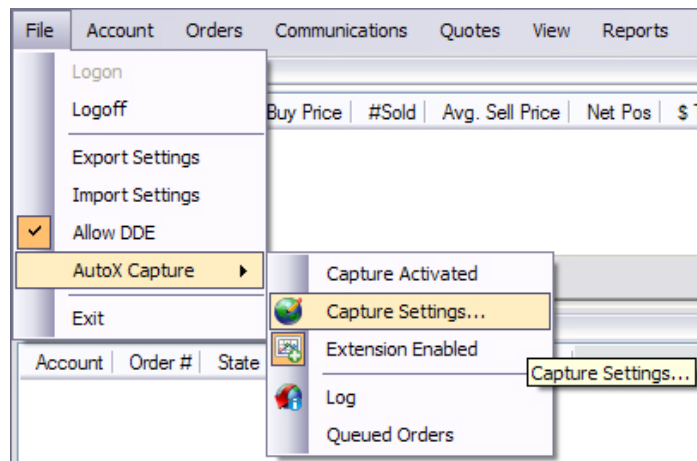


5. In the message dialog box enter any address for the "From email address" field, your email address to "To" field and "LOCALHOST" or "127.0.01" in "SMTP Server name" field.

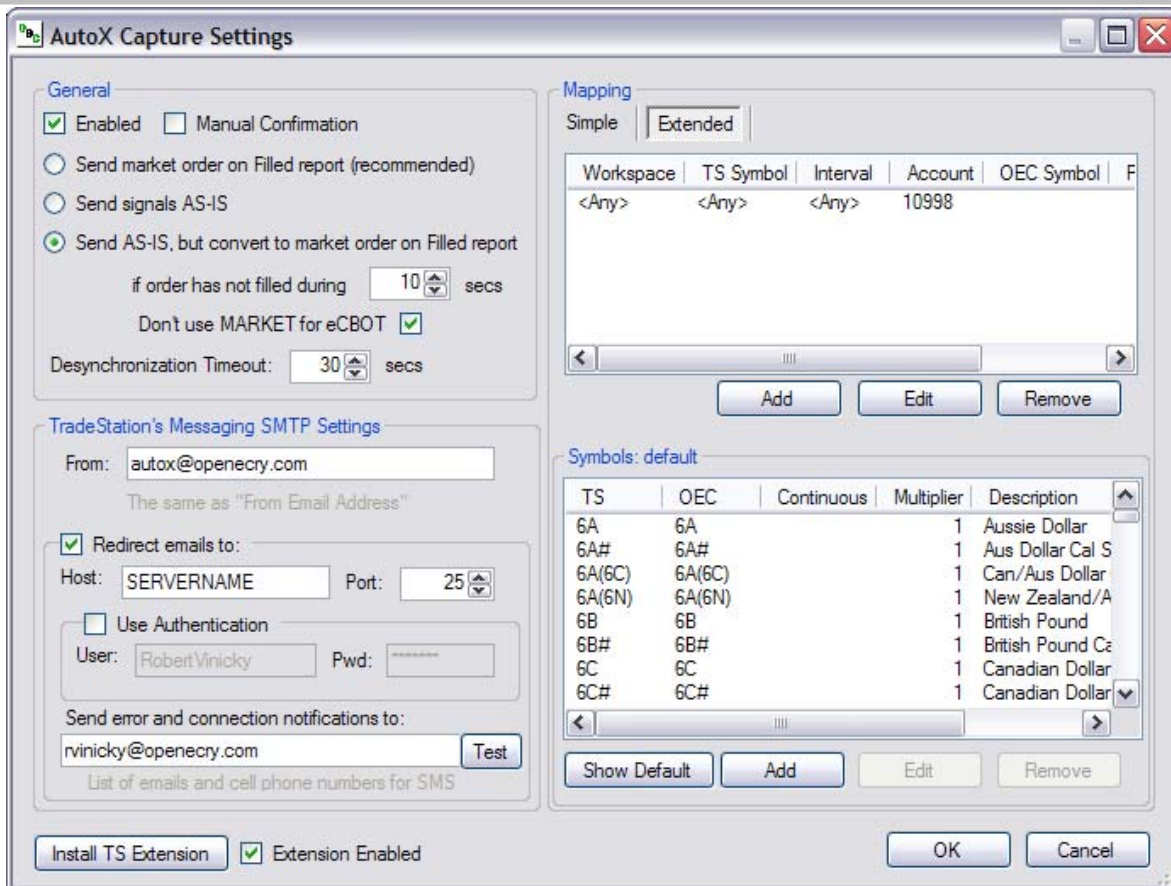


OEC Trader

1. From the main menu of OEC Trader open File > AutoX Capture > Capture Settings



2. Fill out the "From" field with the same email address in TradeStation Messaging.
3. Check "Redirect emails to:" box, if you want to redirect the TradeStation messages to your email client.
4. Check "Enabled" and press OK



Other options

- 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 mode options manage how and when an OEC order will be sent depending on state of related TradeStation Strategy Order.
 - **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 TradeStation Strategy order description, except short order side ("sell short" treats as "sell", "buy to

cover” treats as “buy”). Because TradeStation Strategy orders and OEC Trader orders 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.** 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 Trader order is not filled in selected timeout.
 - **Don't use Market for eCBOT.** Currently OEC Trader does not support modify requests from Stop/Limit/Stop-Limit to a Market order for eCBOT contracts. So, all modify requests on Filled Report for eCBOT contracts will be rejected. As a workaround, this option allows us to simulate market orders via opposite a order type. Example: A Stop order will be modified to Limit order with the same price and vice versa.
 - Desynchronization Timeout displays an alert, if the OEC Trader 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 in the case of a “Send signals AS-IS”. If the original order is rejected and the reason is related to trigger price the reason will contain “trigger price”.

Messages and Notifications

It is possible to redirect all incoming messages to 3rd party SMTP-enabled software, including 3rd party email services. Note: The 3rd party email service may require authentication. The user will need to find out this information from their provider and fill out the host, port and authentication fields of Redirect group box to enable this feature. OEC Trader can send connection status and internal error notifications through the selected redirect point and deliver to the selected email addresses. Moreover, it is possible to send SMS to cell phones. The user can simply type his cell phone number in the “Send error and connection notifications to:” field.

SMS Examples

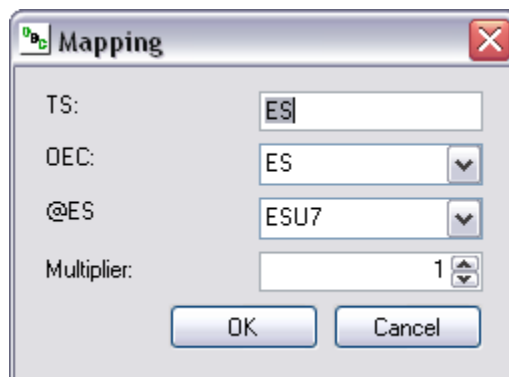
Alltel	6505551111@alltelmessage.com
AT&T, formerly Cingular	16505551111@mobile.mycingular.com or 6505551111@mobile.att.net (Not sure which format to use? Contact AT&T for more information)
Rogers Wireless	6505551111@pcs.rogers.com
Sprint PCS	6505551111@messaging.sprintpcs.com
T-Mobile	6505551111@t-mobile.net
Telus Mobility	6505551111@msg.telus.com
Verizon Wireless	6505551111@vtext.com

Messages and Notifications Limitations - OEC Trader supports only regular SMTP connections. Secured connections like TLS (Transport Layer Security) and STARTTLS (as required by GMail, for example) are not supported.

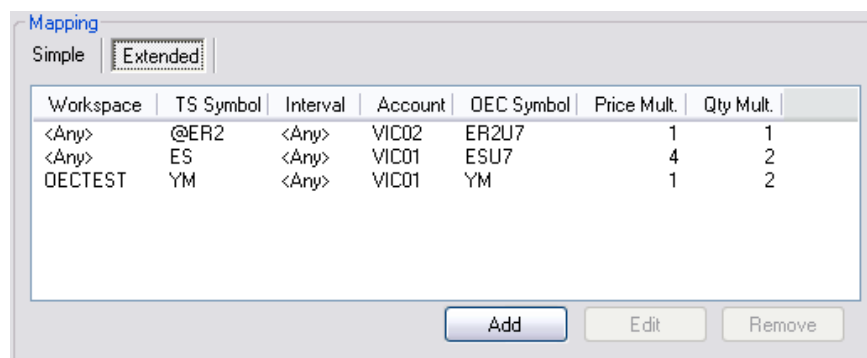
Mapping

There are two modes of mapping: Simple and Extended. Simple mapping has two maps: one maps the workspace name to account and the other maps the symbols. Symbol map is assigned to workspace. Default symbol map is used for references only.

- Account Assignment group box maps the workspace name of the incoming signals to an OEC Trader account. The signals from non-defined workspaces are ignored. One workspace can map to one account.
- Symbol Mapping: Converts TradeStation symbol names and price scales to OEC Trader. Additionally, it maps corresponding TradeStation continuous contract to specific OEC Trader contract. Each workspace has its own mapping. If the symbol is not defined in the mapping, its signals will be ignored. If there are no selected workspaces the Symbol Mapping will show default mapping. The settings from this mapping are used at the time of workspace-related mapping editing.



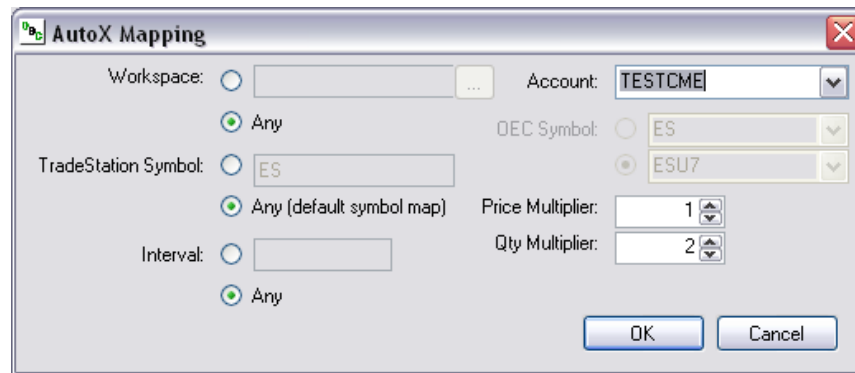
Extended mode uses the incoming workspace name, TradeStation symbol and the interval to map to the OEC Trader account and contract.



The above mapping says:

1. “@ER2 TradeStation symbol from any workspace and any interval chart should be treated as ER2U7 contract and processed through VIC02 account”
2. “any ES TradeStation symbol from any workspace and any interval chart should be treated as ESU7 contract, processed through VIC01 account with price multiplier = 4 and quantity multiplier = 2”
3. “all YM TradeStation symbols from workspace OECTEST and any interval chart should be treated as corresponding YM contract by default symbol mapping rule (e.g., YMM07 => YMM7), processed through VIC01 account with quantity multiplier = 2”

User can add and edit these mapping via the AutoX Mapping dialog box (as seen below):



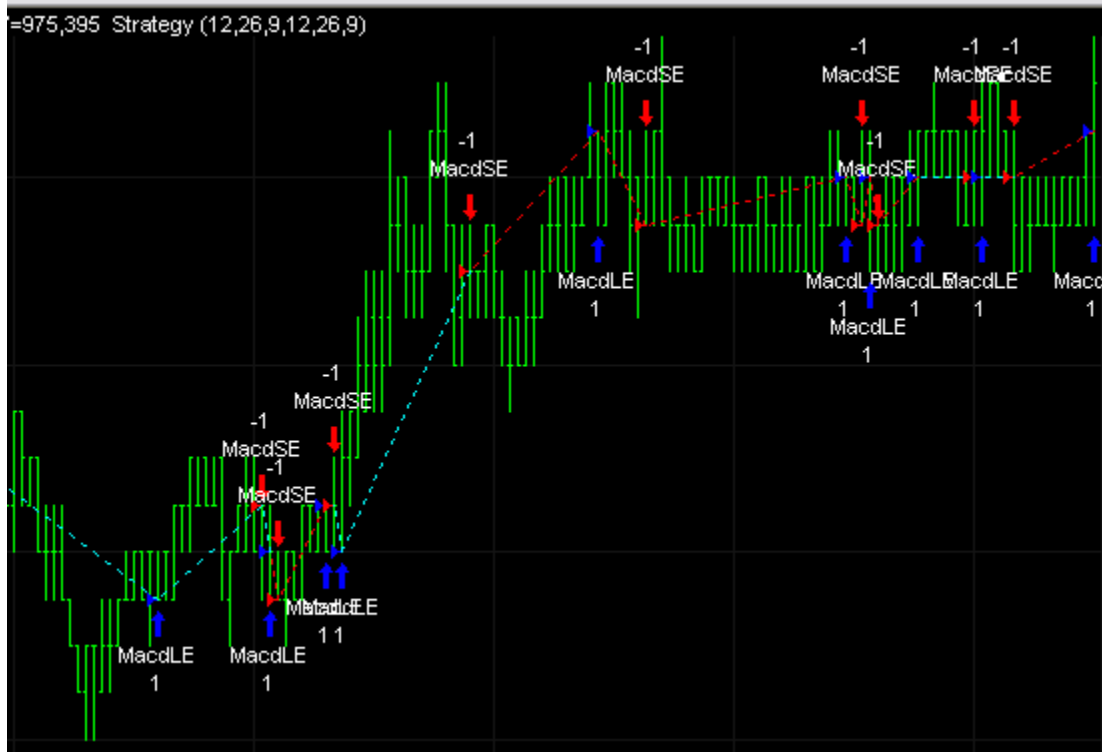
This dialog show the rule that accepts all signals and routes to TESTCME account with quantity multiplier = 2

Extension

- Install TS Extension: This checks the installed version of the TradeStation extensions and recommends installing the latest version if it is obsolete or missing. Please see [OEC API functionality inside EasyLanguage](#) topic for details.
- Extension Enabled: Starts revealing OEC API functionality to TradeStation. If this check box is unchecked TradeStation Extension has no access to the OEC API. Please see [OEC API functionality inside EasyLanguage](#) topic for details.

Process

1. Insert any strategy in “Chart Analysis” in TradeStation. Be sure that the simulated mode is turned on.



2. Login to OEC Trader

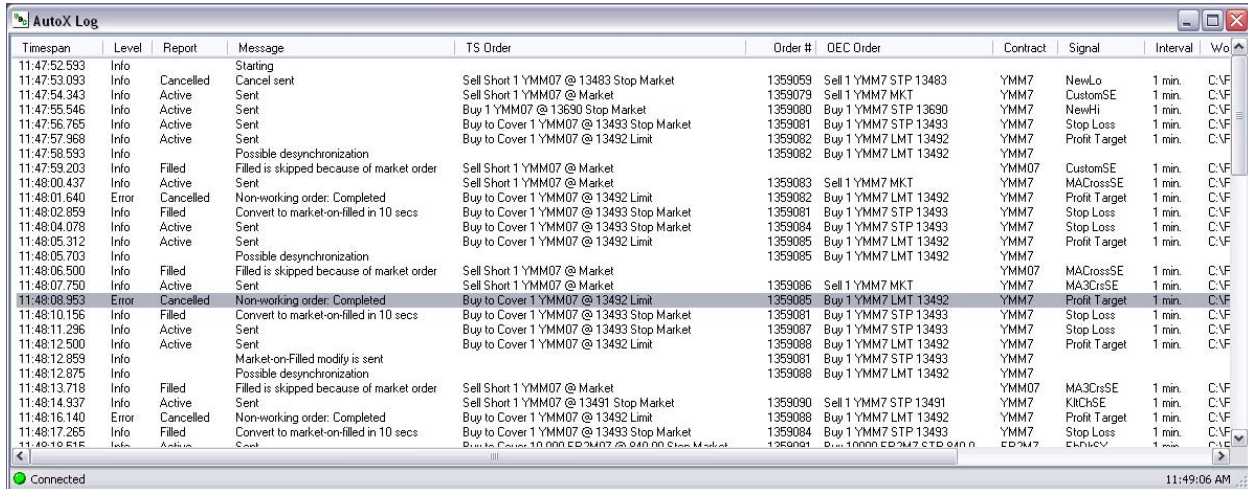
3. Catch and confirm (if "Manual Confirmation" is checked) the signals from TradeStation in OEC Trader. The signal name and interval of the base chart is located in comments of the OEC Trader order.

Completed Orders									
Account	Order #	State	Side	Qty	Symbol	Price	Avg.Price	Comments	Time
VIC01	1337674	Completed	Buy	1/1	YMM7	MKT FOK	12973	MA3CrLE, 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
09:54:36.056 #1337684 Create Sell 1 ER2M7 MKT									
09:54:36.056 Sent									
09:54:36.056 #1337684									

Log

A log of TradeStation interaction is stored to file and can be viewed inside the OEC Trader using the AutoX Log window.

File -> AutoX Capture -> 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							
11:47:59.203	Info	Filled	Filled is skipped because of market order	Sell Short 1 YMM07 @ Market			YMM7	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
11:48:01.640	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
11:48:02.959	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
11:48:05.703	Info		Possible desynchronization							
11:48:06.500	Info	Filled	Filled is skipped because of market order	Sell Short 1 YMM07 @ Market			YMM7	MACrossSE	1 min.	C:V
11:48:07.750	Info	Active	Sent	Sell Short 1 YMM07 @ Market	1359086	Sell 1 YMM7 MKT	YMM7	MACrsSE	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
11:48:12.959	Info		Market-on-Filled modify is sent							
11:48:12.975	Info		Possible desynchronization							
11:48:13.718	Info	Filled	Filled is skipped because of market order	Sell Short 1 YMM07 @ Market			YMM7	MACrsSE	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	KlChSE	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
11:48:18.515	Info	Active	Sent	Buy to Cover 10,000 EP2M07 @ 240.00 Stop Market	1359091	Buy 10000 EP2M7 STP 240.00	EP2M7	EP2M7	1 min.	C:V

Queued Messages

All TradeStation messages are queued in the case of a lost of connection 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 counter Active messages with equal quantity, workspace, interval and symbol and both marked as "Send as MKT", these orders will be removed from queue ("closed by counter order")

A user can view the current queued messages via the File -> AutoX Capture -> Queued Orders dialog box.

#	Timespan	Report	TS Order	Symbol	Signal	Send As MKT	Interval	Workspac
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
```

OEC API functionality inside EasyLanguage

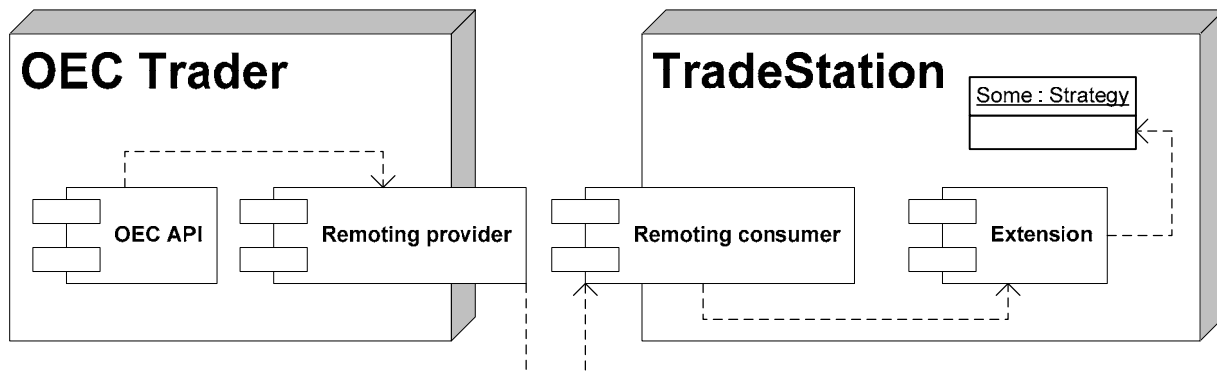
General

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

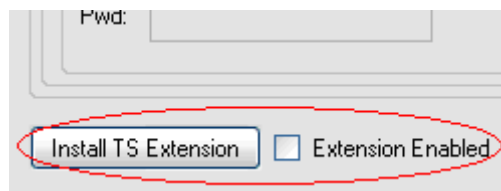
The scheme below shows the process boundary, inter-process communication via .NET remoting, and data flow.



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:

```
oecXXXYYYZZZ(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 = "";
    ASize = 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";
            ASize = ASize + 1;
            qty = qty + oecAvgPositionNetVolume();
        end;
    end;

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

    { 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;

    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/traderstoolbox/autox.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];
```