sterling trading tech

ActiveX API Guide

Version 7.0.2

Welcome

Sterling Trading Tech offers advanced API development integration as part of its Sterling Trader Pro platform. In order to work with the developers, we have provided an API guide, examples and technical support. We are continuously working to develop and advance our API functionality along with our front-end platform. In this guide you will find some basic examples of code to work with our API along with the fundamental library of components. This guide is primarily geared toward development in VB. Development in other languages is possible but is not supported as well by the API interface. Within the guide we will refer to functions as methods and members as properties. More properties and objects exist in the type library than you do in this guide mostly due to backward compatibility issues.

In addition to this guide developers have access to our online examples and our support team. Questions concerning the API can be directed to support@sterlingtradingtech.com or 312-346-9600 x 290. If you wish to be added to our developers email list please send a request to support@sterlingtradingtech.com with your email address and the subject, "Add to STI API Developers List".

Referencing the Type Library

ActiveX components can contain numerous classes, each with one or more programming interfaces. These interfaces can have methods and properties. The components can also have many enumeration constants which are symbolic names for constants that are passed or returned over the interface. For name scoping and management, each application defines its own group of these definitions into a *type library*. The type library is used by programming languages to check the correctness of calls to the component and it is used by COM when it creates the data packet which conveys a method call from one Windows process to another. (This data packet creation is called *marshalling*.)

You must reference the Sterling Type Library to communicate with the Sterling Trader Pro System. This type library gets installed when you install the Sterling Trader Pro. Before you can write Visual Basic code to communicate with the Sterling System you must reference the type library from within your Visual Basic project.

To reference these type libraries from Visual Basic, do the following:

1. From the Visual Basic Menu Bar, find the *References* menu item. Depending on the version of Visual Basic that you are running, this is either under the *Project* (VB6) or under the *Tools* (VBA) menu.
2. Select this menu item to bring up the references dialog. This dialog lists every registered type library on the System. This list is divided into two groups. The type libraries which are already referenced by the project are listed first. All of the remaining registered system type libraries are listed after that in alphabetical order.
3. Find the type library labeled "Sterling 1.0 Type Library" and click the check box to add the reference to your project. (Simply selecting the line is not sufficient; you have to check the box.)
4. Click OK to activate these changes.

After referencing this library, you can use it immediately. Also, if you reopen the references dialog, you will see that this library has been moved up to the top of the list with the other referenced libraries.

Each type library has a name, known as the *Library Name*, that is used in programming to qualify (or scope) all names used within it (such as components, methods, constants, and so on). Take note, however, that this name is not necessarily identical to the name of the file containing the type library. Visual Basic will look up the names in your program by going through the referenced type libraries in the order that they are listed in the references dialog. If two type libraries contain the same name, then Visual Basic will get its definition from the first library in the list, unless the name is qualified with a library name.

The Sterling library name is "SterlingLib". You can use this to qualify any identifier defined in that library.

The Visual Basic 6.0 Development Environment

After the type library is referenced, its definitions become available to the Visual Basic Object Browser. From the Visual Basic development environment, the Object Browser is typically available via a toolbar icon, a menu selection or the F2 key.

The Object Browser can show all referenced type libraries at one time or can focus on any one of them. To focus on the base Sterling Type Library, select "SterlingLib" from the drop-down box in the upper left. The left pane of the Object Browser will then show only the component classes, interfaces and enumerations for SterlingLib. If you select a component (such as *STIOrder*) on the left, its methods and properties will be shown on the right. When you select an item in the right pane, more complete information (such as the parameters to pass to a method and a brief description of the method) are shown at the bottom.

Referencing a type library also activates Visual Basic Intellisense™ for all programming language names in the library. For instance, as you begin to type in a name like *SterlingLib.STIOrder*, you will see the Visual Basic editor display a pop-up list of possible names to complete the typing. As you code a method call, Visual Basic will show you each parameter that you need to provide.

Development in Alternative Environments

Sterling allows for the development in languages other than VB 6.0. This however does come with some setbacks. For example, development in .net based languages will cause a delay in data transmission due to the inefficiencies in communicating between .net and the ActiveX events. We have dealt with this inefficiency by adding the XML functions to our API. They allow you to receive events more quickly. In addition to .net problems may users will have problems while trying to integrate with C++ or JAVA. We recommend steering clear from C++ and JAVA if at all possible. We have found that in VS 2010 you must open VS by right clicking the program and then selecting Run as administrator, once this is done you will be able to open your project. You also need to make sure that your project is compiled using .net 3.5 or earlier. We are not currently compatible with 4.0. We have implemented a solution for the XP 64-bit crashes while using API.

Using XML:

In order to receive the XML events, you will need to first set the mode to XML. To do this you will go under STIAPP on page 16 in the guide and set SetModeXML to true. Then you will need to choose the proper event. One such XML event is OnSTIQuoteUpdateXML vs OnSTIQuoteUpdate. You will also need to check for the XMLSerializer file with IO.File.Exists. Finally, you will need to decode the XML.

Decoding the XML:

In C#:

```
private void OnSTIQuoteUpdateXML(ref string strQuote)
    XmlSerializer xs = new XmlSerializer(typeof(SterlingLib.structSTIQuoteUpdate));
    SterlingLib.structSTIQuoteUpdate structQuote =
        (SterlingLib.structSTIQuoteUpdate)xs.Deserialize(new StringReader(strQuote));
```

In VB.NET:

```
Private Sub OnSTIQuoteUpdateXML(ByRef bstrQuote As String)
        Dim xs As New XmlSerializer(GetType(SterlingLib.structSTIQuoteUpdate))
        Dim sr As New StringReader(bstrQuote)
```

2

```
Dim structQuote As SterlingLib.structSTIQuoteUpdate = DirectCast(xs.Deserialize(sr),
SterlingLib.structSTIQuoteUpdate)

sr.Close()
```

The members of structQuote can now be accessed.

Sterling ActiveX API Performance Considerations

The ActiveX API delay has been reduced to produce a maximum of 20 orders per second.

Cancel requests can now be sent for any open orders which have not had any activity within the past ten seconds regardless of their current state; i.e. – no further order confirm messages are being received from the exchange. Therefore, the order state says “Pending” – the order appears to be “frozen”. Such an order can now be canceled after 10 seconds, either from within the ActiveX API itself or from the Trading Monitor screen.

Visual Basic Coding Examples

Create a reference to the Sterling ActiveX Library

-Select Project • References… from the menu. -Select Sterling 1.0 Type Library from the Available References. -If the Sterling 1.0 Type Library is not found, select Browse and find the Sterling.tlb file.

Enable Events

-Place the following line in the general section of your code to declare the object: Dim WithEvents m_STIEvents As STIEvents

-Place the following line in an initializing procedure, such as the Form_Load procedure:
Set m_STIEvents = New STIEvents

-Now select m_STIEvents in the Object drop down control in your code window (top left drop down control). You will see the available events in the Procedure drop down control (top right drop down control). Select the event that you would like to catch, and it will be inserted into your code. It should look something like this:

```
Private Sub m_STIEvents_OnSTIOrderUpdateMsg(ByVal oSTIOrderUpdateMsg As ISTIOrderUpdateMsg)
```

-Use the oSTIOrderUpdateMsg object to gather the information from the message.

Sending an Order

-Create the order object with the following code:

```
Dim order As STIOrder Set order = New STIOrder Dim storder As structSTIOrder
```

-Fill the order properties with order information:

3

storder.Account = “ACCT7” storder.Side = “B” storder.Symbol = “CSCO” storder.Quantity = 500 storder.PriceType = ptSTIMkt storder.Tif = “D” storder.Destination = “NYSE”

-Create CLOrderId: Dim theTime As SYSTEMTIME GetLocalTime theTime storder.bstrClOrderId = storder.Account & ostheTime.wYear & theTime.wMonth & theTime.wDay & theTime.wHour & theTime.wMinute & theTime.wSecond & theTime.wMilliseconds

-Check for Errors Text1.Text = order.SubmitOrderStruct(os)

Canceling an Order

-Place the following line in the general section of your code to declare the object: Dim m_STIOrderMaint As STIOrderMaint

-Place the following line in an initializing procedure, such as the Form_Load procedure: Set m_STIOrderMaint = New STIOrderMaint

-Call the CancelOrder Method of the STIOrderMaint object for passing the required order information.

You can use either the OrderRecordID or the ClOrdID to cancel an order.

-OrderRecordID is the value that you get back in the OrderUpdateMsg. This is a unique order ID generated by the Sterling Trader® System to track an order. -OldClOrdID is the client-generated order ID passed-in when the order is first sent. -ClOrdID is an optional field. It is the ActiveX API client-generated order ID used for canceling the order record. It must be a unique ID, and it must remain unique over multiple trading days.

-To cancel with the Client Order ID:
m_STIOrderMaint.CancelOrder “<Account>”, 0, “<Client Order Id of Order to cancel>”, “<Client Order Id of new cancel order record>”

-To cancel with the Record ID:
m_STIOrderMaint.CancelOrder “<Account>”, <Record ID>, “”, “”

Tracking Orders Using a Client Order ID

The Client Order ID is used for the purpose of assigning an ActiveX API client-generated ID to an order *before* that order is initially sent. The Client Order ID is an optional field – one that functions as a tool for helping ActiveX API clients track orders. It is one of a number of order IDs used by the Sterling Trader® System; each of the following fields listed below is available for the purpose of tracking orders by the system:

1 ExchClOrderID = is generated by the Sterling Trader® System; then sent to the exchange.
2 ExchOrderID = is exchange generated.
3 OrderRecordID = is an internal record ID generated by the Sterling Trader® System; it is guaranteed to be unique, relative to other OrderRecordIDs, for multiple trading days.

1. ClOrdId is a recommended field. It is the ActiveX API client-generated Order ID used for tracking the order record. It must be designated as a unique ID, and must remain unique over multiple trading days.
2. Please remember NOT to use commas (,), equal signs (=), or tilde (~) as part of your ClientOrderID.

The Client Order ID must be assigned to the ClOrderID property of the STIOrder object before you call the STIOrder SubmitOrder function. This ID must be unique over multiple days. For instance, you could use the combination of Account + TimeStamp(to the millisecond) + Counter. This would provide a unique ID that would allow for orders to be sent within the same millisecond and over multiple days.

STIEvents

OnSTILinkSymChange This event occurs when the symbol of a link group is changed on the front end or in the API.

OnSTIDrop This event is fired when a symbol is dragged out of a Sterling window and dropped.

OnSTIOrderConfirm This event is fired when an order submitted into Sterling Trader Pro is received by the destination.

OnSTIOrderReject This event is fired when an order is rejected by the Sterling Server level.

OnSTIOrderUpdate This event corresponds to any change on an order. The values in this update will correspond to the aggregate of the order, i.e. it will show the cumulative executed quantity rather than the single executions quantity.

OnSTITradeUpdate This event is fired for each execution the order received and contains the data for that execution.

Handling Rejections in the API

Orders will sometimes be rejected. In the API environment you will want to be able to handle these rejections so that they can be corrected. Within the API environment four different levels of rejection are possible. The first is the return on the SubmitOrder. Anything other than zero will be an error code. The second is the OnSTIOrderReject event. This event is triggered when the order is rejected in Sterling. The third is a backend rejection at Sterling which will change the status of the order to Rejected (STIOrderStatus = 12). This is also the case with an exchange reject, the fourth and final level of rejection.

The STIOrder SubmitOrder function will send back a return code indicating the success or failure of an order's admittance into the Sterling Trader® System. If the return code is a negative number, then the order failed and was not sent. The negative number of the return code will correspond to one of the defined error code values that are found in the "SubmitOrder Error Codes" section on page 18. If the return code is not a negative number, then the order was sent from the Sterling Trader® System to the exchange successfully. It is absolutely necessary to include this function in your program if it is going to be self-contained and not rely on the front end for the messages.

The second level of rejections will come as the event, OnSTIOrderReject. Within this event the nRejectReason will define the cause of the rejection. This will be a positive integer. This value corresponds to those listed on page 17 and 17 under STIRejectReason.

The next level of rejections is the Sterling backend. This level and the fourth level, exchange rejections, operate the same. Both will be seen as part of the OnSTIOrderUpdate event. This will show as part of the nOrderStatus the STIOrderStatus of 12 is the indication that the order was rejected. To get more information on the cause of this

5

rejection you will need to also pull The STIOrderUpdate property for bstrLogMessage. This message will be the best source for an explanation on the reject, but it may not always be clear.

Backward Compatibility and Algos

In order to make this guide as easy to understand and concise as possible some rarely used and obsolete functions were removed. These are potentially useful to those working on a special program or updating an existing project. In order to keep this data available a new backward compatibility API guide has been created.

6

Sterling ActiveX Object Library

STIOrder

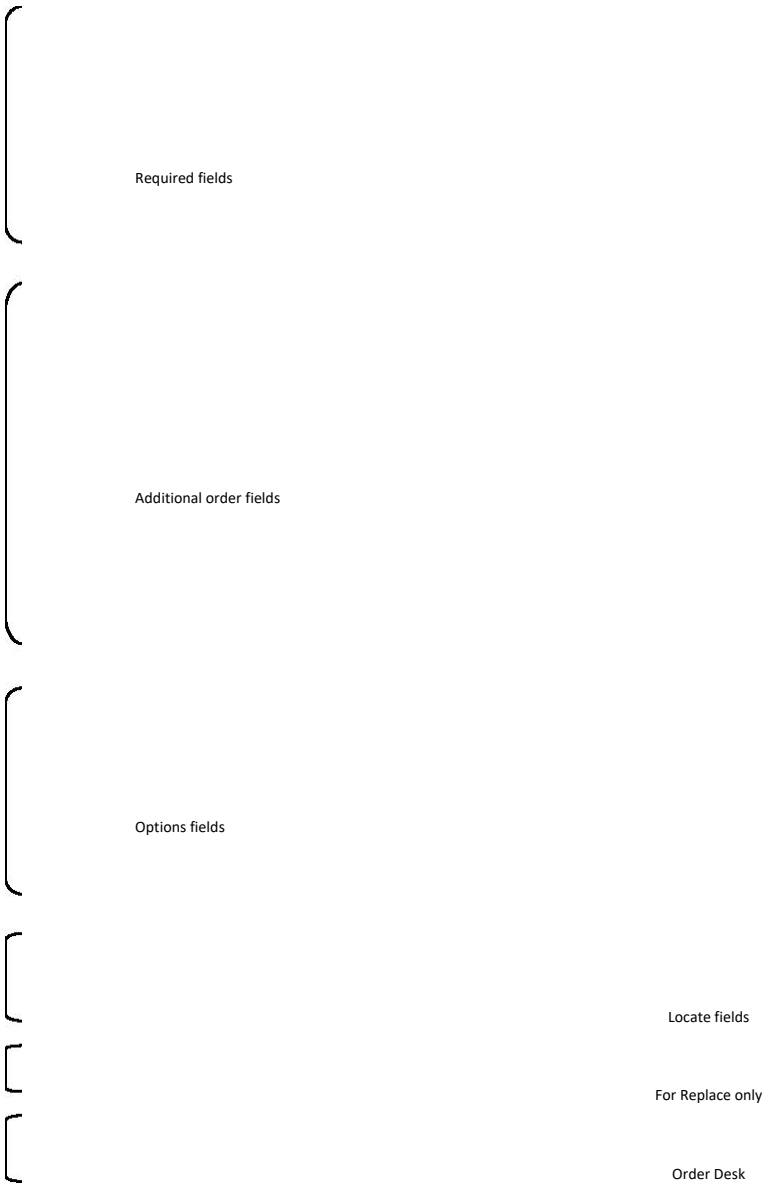
Methods

long GetQueueCount()

long SubmitOrderStruct(structSTIOrder *pOrder)

long ReplaceOrderStruct(structSTIOrder *pOrder, long nOldOrderRecId, BSTR bstroldClOrderId) [1]While both RecId and ClOrderId can be used ClOrderId is the recommended option. [2]The only fields that can be changed are Price (LmtPrice, StpPrice, and PriceType) and Quantity. Other fields must match the original order, with the exception of the ClOrderId which should be unique.

HRESULT ClearOrderStruct(structSTIOrder *pOrder)



StructSTIOrder

Struct (UDT)

Properties (Read Only)	Type	Description
bstrSide	BSTR	See section Side for values (page 20)
bstrSymbol	BSTR	The options or equities symbol.
bstrAccount	BSTR	The account exactly as it appears in Sterling.
nPriceType	long	See Price Types for values (page 20)
bstrTif	BSTR	See TIF for values (page 20).
nQuantity	long	The number of shares (or contracts)
bstrDestination	BSTR	The destination as it appears on the Sterling system ex. ARCA
bstrClOrderId	BSTR	Recommended field. Use this field to track orders (page 4)
fLmtPrice	double	If price type is set to ptSTILmt this will be the limit value.
nDisplay	long	Send Hidden (0), Reserve (qty displayed) and visible if blank
fDiscretion	double	Price discretion on a limit order.
bstrExecInst	BSTR	Field for specifying Special Order Designations (see page 23)
fPegDiff	double	PEG-If price type is set to ptSTIPegged this will be the discretion
fTrailAmt	double	T-STP-Amount by which you want to trail the last.
fTrailInc	double	T-STP-The value change needed to trigger an update on the trigger price.
fStpPrice	double	STP-The trigger price on a STP or S-STP order.
nMinQuantity	long	Minimum Fill quantity
bstrExecBroker	BSTR	Preference
bstrUser	BSTR	(Client-defined field)
bstrCurrency	BSTR	
bstrOpenClose	BSTR	O=Open, C=Close
bstrMaturity	BSTR	YYYYMMDD
bstrPutCall	BSTR	P=Put, C=Call
bstrUnderlying	BSTR	Underlying equities Symbol
bstrCoverUncover	BSTR	C=Covered, U=Uncovered
bstrInstrument	BSTR	For instrument values (see page 21)
fStrikePrice	double	Options strike price
bstrLocateBroker	BSTR	Text field for Broker information.
bstrLocateTime	BSTR	Text field for locate time information.
nLocateQty	long	Text field for locate quantity information.
bstrListingExchange	BSTR	The listing exchange.
nParentRecordId	long	RecordId of the order to be replaced.
bstrBatchID	BSTR	Used when splitting orders in the order desk manager

Sterling ActiveX Object Library

STIPosition

Events
void OnSTIPositionUpdate (structSTIPositionUpdate* structPositionUpdate) void OnSTIPositionUpdateXML(BSTR* bstrPosition)

void OnSTIShutdown()

Methods
HRESULT RegisterForPositions() HRESULT DeRegisterPositions() HRESULT GetCurrentPositions()

structSTIPositionUpdate GetPositionInfoStruct (BSTR bstrSymbol, BSTR bstrExch, BSTR bstrAccount)

long GetQueueCount() long GetPositionList(structSTIPosUpdate() arrayPos) long GetOptionsPosList(BSTR bstrUnderlyingSym, structSTIPosUpdate() arrayPos) long GetPosListBySym(BSTR bstrSymbol, structSTIPosUpdate() arrayPos

If a symbol is not specified, GetOptionsPosList() returns all options positions; GetPosListBySym returns all positions.

structSTIPositionUpdate

Properties (Read Only)	Type	Description
bstrSym	BSTR	Symbol
bstrAcct	BSTR	Trading account
bstrInstrument	BSTR	Position instrument
nOpeningPosition	long	The position in the account to start the day.
nSharesBot	long	Number of Share purchased
nSharesSld	long	Number of Shares Sold. (nSharesSldLong+ nSharesSldShort)
nSharesSldLong	long	Number of Shares sold long this is a component of nSharesSld.
nSharesSldShort	long	Number of Shares sold short this is a component of nSharesSld.
nTicketsBot	long	Buy side executions
nTicketsSld	long	Sell and Sell Short executions
nTicketsSldLong	long	Sell executions
nTicketsSldShort	long	Sell Short executions
fClosePrice	double	Yesterday's close
fDollarsBot	double	The total cost of BUY orders.
fDollarsSld	double	The total of fDollarsSldLong+ fDollarsSldShort
fDollarsSldLong	double	The total cost of SELL orders.
fDollarsSldShort	double	The total cost of SHRT orders.
fPositionCost	double	The total of fDollarsBot+ fDollarsSld
fReal	double	This is the Realized Profit/Loss value.
nSharesPerContract	long	Options field Number of underling shares per options contract
nPremiumMultiplier	long	This is the multiplier for options positions.
bLast	VARIANT_BOOL	Indicates this is the last of a list of events because of a request (GetCurrentPositions())
bMsgSnapShot	VARIANT_BOOL	Indicates that event is due to a request(GetCurrentPositions()) not a position change.

Not all position fields are provided some need to be calculated. Example: Position = nOpeningPosition + (nSharesBot -nSharesSld)

Sterling ActiveX Object Library

STIQuote

Properties Type Description
nUpdateID Long Optional quoteupdate ID of the STIQuote object.

Events
void OnSTIQuoteSnap(structSTIQuoteSnap* structQuoteSnap) void OnSTIQuoteUpdate(structSTIQuoteUpdate* structQuoteUpdate) void OnSTIGreeksUpdate(structSTIGreeksUpdate* structGreeksUpdate) void OnSTINewsUpdate(structSTINewsUpdate* structNewsUpdate) void OnSTIQuoteUpdateXML(BSTR* bstrQuote) void OnSTIQuoteSnapXML(BSTR* bstrQuote) void OnSTIQuoteRqstXML(BSTR* bstrQuote) void OnSTIGreeksUpdateXML(BSTR* bstrQuote) void OnSTINewsUpdateXML(BSTR* bstrQuote) void OnSTIShutdown()

Methods
HRESULT DeRegisterQuote(BSTR bstrSymbol, bstrExch) HRESULT DeRegisterAllQuotes() HRESULT RegisterQuote(BSTR bstrSymbol, bstrExch) HRESULT RegisterForMdx(bool bMdx) HRESULT RegisterForAllMdx(bool bMdx)

Use QuoteUpdate to receive MDX messages. HRESULT RegisterForNewMdx(bool bMdx, bool bAllSyms)

Registers for all MDX messages unless bAllSyms is false in which case only registered symbols will receive messages. HRESULT RegisterForAllNews(bool bNews) HRESULT RegisterQuoteEx(structSTIQuoteRegEx *pQuote)

Registers a symbol with the option of registering a different STIQuote object for only QuoteUpdate events via the nUpdateID member of structSTIQuoteRegEx. HRESULT RegisterMdxEx(structSTIMdxRegEx *pMdx) HRESULT SetTradesOnly(bool bTrades)

If bTrades is 'True', then only trade messages will be sent.

long GetQueueCount() [1]A setting allows the trader to select if they want to combine queued quotes. This setting can be found in the Quotes section of the Global Settings. [2]Each symbol will only be registered once. Registering again will only fire a OnSTIQuoteSnap. Deregistering the symbol will completely deregister the symbol. [3]If you register for a Quote for an options symbol you will also receive the Greeks for that symbol. [4]Exchange Alerts will be available for both REG and Info Imbalances be sure to recognize the message type.

structSTIQuoteSnap Struct (UDT)

Properties (Read Only) Type Description
bstrAskExch BSTR Exchange associated with the ask quote bstrBbo BSTR 'B'= best bid, 'A'=best ask, 'X'=best bid and ask bstrBidExch BSTR Exchange associated with the bid quote bstrCompanyName BSTR Company associated to

the equity symbol bstrCusip BSTR Identifier associated to equities bstrExch BSTR Returns the exchange from the request. bstrExDivDate BSTR Ex-dividend date bstrLastExch BSTR Exchange with the last trade bstrListingExch BSTR The exchange where the symbol is listed.

bstrSymbol	BSTR	The symbol.
bstrTradeCondition	BSTR	Trade Condition on the last trade.
bstrUnderlying	BSTR	Underlying symbol of the options symbol
bstrUpdateTime	BSTR	Time of the last quote update
f52WeekHigh	double	The 52 week high.
f52WeekLow	double	The 52 week low.
fAskPrice	double	The Ask Price.
fBeta	double	
fBidPrice	double	The bid price.
fClosePrice	double	The previous day close.
fDivAmt	double	The dividend amount.
fEps	double	
fHighPrice	double	Intraday High.
fLastPrice	double	The Last price.
fLowPrice	double	Intraday Low.
fOpenPrice	double	The opening price.
fVwap	double	Vwap calculation.
nAskSize	long	Number of shares on the Ask quote.
nAvgVolume	long	The average volume.
nBidSize	long	Number of shares on the Bid quote.
nCumVolume	long	The cumulative volume.
nDivFreq	long	0, 3=Not Available, 1= Annual, 2= Semi-Annual, 4=Quarterly
nLastSize	long	The number of shares in the last trade.
nOpenInterest	long	Shows the open interest in options quotes
nPremiumMultiplier	long	This is the multiplier for options positions.
nQuoteCondition	long	Denotes irregular trade conditions
nSharesOutstanding	long	
nSharesPerContract	long	Shares per contract in an options symbol.
nSpCode	long	
bAskPrice	VARIANT_BOOL	Informs if Ask is valid.
bBidPrice	VARIANT_BOOL	Informs if Bid is valid.
bHighPrice	VARIANT_BOOL	Informs if High is valid.
bLastPrice	VARIANT_BOOL	Informs if Last is valid.
bLowPrice	VARIANT_BOOL	Informs if Low is valid.
bOpenPrice	VARIANT_BOOL	Informs is Open is valid.

Sterling ActiveX Object Library

structSTIQuoteUpdate

Properties (Read Only) Type Description

bstrAskExch BSTR Exchange associated with the ask quote bstrBbo BSTR bstrBidExch BSTR Exchange associated with the bid quote bstrExch BSTR bstrHaltResumeReason BSTR See bstrHaltResumeReason on page 22 bstrHaltResumeStatus BSTR See bstrHaltResumeStatus on page 22 bstrLastExch BSTR Exchange with the last trade bstrSymbol BSTR The symbol. bstrTradeCondition BSTR Trade Condition on the last trade. bstrUpdateTime BSTR Time of the last update fAskPrice double The ask price. (will be 0 if the update is a bid or last) fBidPrice double The bid price. (will be 0 if the update is an ask or last) fHighPrice double Intraday high fIndicatorHigh double

fIndicatorLow	double	
fLastPrice	double	The last trade price
fLowPrice	double	Intraday low
fOpenPrice	double	The opening price
fVwap	double	
nAskSize	long	The number of shares on the Ask
nBidSize	long	The number of shares on the Bid
nCumVolume	long	The volume traded today
nImbalance	long	
nIntradayMktImbalance	long	
nLastSize	long	Size of the last trade
nMdxMsgType	long	
nMktImbalance	long	
nOpenInterest	long	Shows the open interest in options quotes
nQuoteCondition	long	
bAskPrice	VARIANT_BOOL	Informs if Ask is valid.
bBidPrice	VARIANT_BOOL	Informs if Bid is valid.
bHighPrice	VARIANT_BOOL	Informs if High is valid.
bLastPrice	VARIANT_BOOL	Informs if Last is valid.
bLowPrice	VARIANT_BOOL	Informs if Low is valid.
bMdxSnapShot	VARIANT_BOOL	
bOpenPrice	VARIANT_BOOL	Informs if Open is valid.
bValidMktImb	VARIANT_BOOL	Must be set to "True" to use nMktImbalance
bValidIntradayMktImb	VARIANT_BOOL	Must be set to "True" to use nIntradayMktImbalance
bValidIndicators	VARIANT_BOOL	Set to "True" to make fIndicatorHigh & fIndicatorLow available.

Sterling ActiveX Object Library

structSTIGreeksUpdate Struct (UDT)

Properties (Read Only) Type Description
bstrSymbol BSTR fDelta double fGamma double fTheta double fVega double fRho double fTheoPrice double fImpVol double

structSTINewsUpdate Struct (UDT)

Properties (Read Only) Type Description
bstrService BSTR bstrArticleDate BSTR bstrSeq BSTR bstrDisplayTime BSTR bstrKeys BSTR bstrHeadline BSTR bHot VARIANT_BOOL

structSTIQuoteRegEx Struct (UDT)

Sterling ActiveX Object Library

Properties (Read Only) Type Description
bstrSymbol BSTR bstrExch BSTR bReg VARIANT_BOOL nUpdateID long

structSTIMdxRegEx Struct (UDT)

Properties (Read Only)	Type	Description
bstrExchanges	BSTR	Comma-delimited string of Exchanges or empty string means all exchanges.
bstrMsgTypes	BSTR	Comma-delimited string of message types or empty means all messages.
bReg	VARIANT_BOOL	Registering or de-registering.
bAllSyms	VARIANT_BOOL	All symbols or quote-registered symbols.

STIBook

Events
void OnSTIBookUpdateMsg(structSTIBookUpdate* structBookUpdate) void OnSTIShutdown()

Methods
HRESULT DeRegisterForAllTopOfBookMsgs() HRESULT DeRegisterForTopOfBookMsgs(BSTR bstrSymbol, BSTR bstrBook) BSTR GetBookInfo(BSTR bstrSymbol, BSTR bstrBook) structSTIBookUpdate GetBookInfoStruct(bool bBid, BSTR bstrSymbol, BSTR bstrBook) HRESULT RegisterForTopOfBookMsgs(BSTR bstrSymbol, BSTR bstrBook) long GetQueueCount()

Available Books
ARCA (ARB) BATS (BAB) EDGA (ATB); 'EAB' is the front-end book-label. EDGX (NOB); 'EXB' is the front-end book-label.

Not Available
NY Open (NYB); NYB does not have "TopOfBook"

structSTIBookUpdate

Properties (Read Only) Type Description
bstrOrigin BSTR bstrSide BSTR bstrSymbol BSTR bstrTime BSTR fPrice double nQty long bMsgSnapShot VARIANT_BOOL

STIOrderMaint

Methods

Sterling ActiveX Object Library

HRESULT CancelOrder(BSTR bstrAccount, long OrderRecordId, BSTR bstrOldClOrderId, BSTR bstrClOrderId) HRESULT GetOrderInfo(BSTR bstrClOrderId, structSTIOrderUpdate* structOrder); HRESULT GetOrderList(VARIANT_BOOL bOpenOnly, SAFEARRAY(structSTIOrderUpdate) *arrayOrder, long *lCount); HRESULT CancelFuturesOrder(BSTR bstrAccount, long OrderRecordId, BSTR bstrOldClOrderId, BSTR bstrClOrderId); HRESULT CancelOptionsOrder(BSTR bstrAccount, long OrderRecordId, BSTR bstrOldClOrderId, BSTR bstrClOrderId); HRESULT CancelForexOrder(BSTR bstrAccount, long OrderRecordId, BSTR bstrOldClOrderId, BSTR bstrClOrderId); HRESULT GetEquityTradeList(SAFEARRAY(structSTITradeUpdate) *arrayTrade, long *lCount); HRESULT GetFuturesTradeList(SAFEARRAY(structSTITradeUpdate) *arrayTrade, long *lCount); HRESULT GetOptionsTradeList(SAFEARRAY(structSTITradeUpdate) *arrayTrade, long *lCount); HRESULT GetForexTradeList(SAFEARRAY(structSTITradeUpdate) *arrayTrade, long *lCount); HRESULT GetFuturesOrderList(VARIANT_BOOL bOpenOnly, SAFEARRAY(structSTIOrderUpdate) *arrayOrder, long *lCount); HRESULT GetOptionsOrderList(VARIANT_BOOL bOpenOnly, SAFEARRAY(structSTIOrderUpdate) *arrayOrder, long *lCount); HRESULT GetForexOrderList(VARIANT_BOOL bOpenOnly, SAFEARRAY(structSTIOrderUpdate) *arrayOrder, long *lCount); HRESULT CancelOrderEx(BSTR bstrAccount, long OrderRecordId, BSTR bstrOldClOrderId, BSTR bstrClOrderId, BSTR bstrInst, long *lRetVal); HRESULT GetOrderListEx(structSTIOrderFilter* pFilter, SAFEARRAY(structSTIOrderUpdate) *arrayOrder, long *lCount); HRESULT GetTradeListEx(structSTITradeFilter* pFilter, SAFEARRAY(structSTITradeUpdate) *arrayTrade, long *lCount);

[1] Returns include -16 (Pro is offline) and -37(Multiple sub-second replace and/or cancel attempts)

STIEvents

Events

void OnSTILinkSymChange(structSTILink* structLink) void OnSTIDrop(structSTIDrop* structDrop) void OnSTIOrderConfirm(structSTIOrderConfirm* structOrderConfirm) void OnSTIOrderReject(structSTIOrderReject* structOrderReject) void OnSTIOrderUpdate(structSTIOrderUpdate* structOrderUpdate) void OnSTITradeUpdate(structSTITradeUpdate* structTradeUpdate) void OnSTITradeUpdateXML(BSTR* bstrTrade) void OnSTIOrderUpdateXML(BSTR* bstrOrder) void OnSTIOrderRejectXML(BSTR* bstrOrder) void OnSTIOrderConfirmXML(BSTR* bstrOrder) void OnSTIShutdown()

Methods

MeHRESULT SetOrderEventsAsStructs(bool bStruct)

structSTIDrop Struct (UDT)

structSTILink Struct (UDT)

Properties (Read Only)	Type	Description
bstrSymbol	BSTR	Symbol you wish to link
bstrUnderlying	BSTR	Underlying equity symbol for Options
nGroup	long	Link group on Sterling you wish to link the symbol into.

Properties (Read Only) Type Description

bstrSymbol BSTR Symbol you wish to link

Sterling ActiveX Object Library

structSTIOrderConfirm Structs (UDT)

Properties (Read Only)	Type	Description
bstrAccount	BSTR	The name of the sterling account the order was placed in.
bstrClOrderId	BSTR	Trader generated order ID
bstrExchClOrderId	BSTR	
bstrExchOrderId	BSTR	
bstrExchOrderId2	BSTR	
bstrInstrument	BSTR	The trade type (equity or option)
bstrMsgConfirm	BSTR	Text message for the confirm

bstrUnderlying BSTR Underlying equity symbol for Options nGroup long Link group on Sterling you wish to link the symbol into.

structSTIOrderReject Struct (UDT)

Properties (Read Only) Type Description
bstrAccount BSTR bstrBatchId BSTR bstrClOrderId BSTR bstrCoverUncover BSTR (C=Covered, U=Uncovered) bstrDestination BSTR bstrExecBroker BSTR bstrExecInst BSTR bstrInstrument BSTR bstrListingExchange BSTR bstrMaturity BSTR (YYYYMMDD) bstrOpenClose BSTR (O=Open, C=Close) bstrPriceType BSTR bstrPutCall BSTR (P=Put, C=Call) bstrSide BSTR bstrSymbol BSTR bstrTif BSTR bstrText BSTR Contains text information on the Rejection. bstrUnderlying BSTR bstrUser BSTR (Client-defined field) fDiscretion double fLmtPrice double fPegDiff double fStpPrice double fStrikePrice double fTrailAmt double fTrailInc double nDisplay long nMinQuantity long nPriceType long nQuantity long nRejectReason long Contains the numerical reject code

StructSTIOrderUpdate Structs (UDT)

Sterling ActiveX Object Library

Properties (Read Only)	Type	Description
bstrAccount	BSTR	
bstrAction	BSTR	
bstrBatchId	BSTR	
bstrClOrderId	BSTR	
bstrCoverUncover	BSTR	(C=Covered, U=Uncovered)
bstrDestination	BSTR	
bstrExchClOrderId	BSTR	
bstrExchClOrderId2	BSTR	
bstrExchOrderId	BSTR	
bstrExecBroker	BSTR	
bstrExecInst	BSTR	
bstrInstrument	BSTR	
bstrLogMessage	BSTR	Provides the log messages on an order.
bstrMaturity	BSTR	(YYYYMMDD)
bstrOpenClose	BSTR	(O=Open, C=Close)
bstrOrderTime	BSTR	
bstrPriceType	BSTR	
bstrPutCall	BSTR	(P=Put, C=Call)
bstrSide	BSTR	
bstrSymbol	BSTR	
bstrTif	BSTR	
bstrUnderlying	BSTR	
bstrUpdateTime	BSTR	
bstrUser	BSTR	(Client-defined field)
bstrUserId	BSTR	(Trader/Login ID)
fAvgExecPrice	double	
fDiscretion	double	
fLmtPrice	double	
fPegDiff	double	
fStpPrice	double	
fStrikePrice	double	
fTrailAmt	double	
fTrailInc	double	
fUrStpdPrice	double	
nCumExecQuantity	long	
nDbsNo	long	
nDisplay	long	
nLvsQuantity	long	
nMinQuantity	long	
nOrderRecordId	long	
nOrderStatus	long	
nPriceType	long	
nQuantity	long	
nSeqNo	long	

StructSTITradeUpdate

Properties (Read Only)	Type	Description
bstrAccount	BSTR	

Sterling ActiveX Object Library

STIApp

Methods

bstrBatchId	BSTR	Used by Order Dest to associate a group of orders
bstrClOrderId	BSTR	
bstrContra	BSTR	Contra broker
bstrCoverUncover	BSTR	
bstrDestination	BSTR	
bstrExchClOrderId	BSTR	
bstrExchExecId	BSTR	
bstrExchOrderId	BSTR	
bstrExchOrderId2	BSTR	
bstrExecBroker	BSTR	
bstrExecInst	BSTR	
bstrInstrument	BSTR	
bstrLiquidity	BSTR	
bstrLogMessage	BSTR	
bstrMaturity	BSTR	
bstrOpenClose	BSTR	
bstrOrderTime	BSTR	
bstrPutCall	BSTR	
bstrSide	BSTR	
bstrSpecialist	BSTR	
bstrSymbol	BSTR	
bstrTif	BSTR	
bstrTradeTime	BSTR	
bstrUnderlying	BSTR	
bstrUpdateTime	BSTR	
bstrUserId	BSTR	
fDiscretion	double	
fExecPrice	double	
fLmtPrice	double	
fPegDiff	double	
fStpPrice	double	
fStrikePrice	double	
nDbsNo	long	Database number
nLvsQuantity	long	
nOrderRecordId	long	
nPriceType	long	
nQuantity	long	
nSeqNo	long	
nTradeRecordId	long	

HRESULT SwitchLinkGroupSymbol(long nLinkGroup, BSTR bstrSym, BSTR bstrExch) -Sends the symbol into Sterling long GetDestinationList(BSTR() arrayDests) -Pull a list of available destinations

BSTR GetTraderName() [1]See STIAcctMaint (on page 17) for GetAccountList() -Provides the login name of the user,

often the same as the Account.

BSTR GetServerTime()

Sterling ActiveX Object Library

Pulls the Time off Sterling DB in this format: CCYYMMDDhhmms

SetModeXML(bool bXML) Enables the use of XML events. VARIANT_BOOL
IsApiEnabled() Confirm with API that the trader is entitled to use API.

STIAcctMaint

Events
void OnSTIAcctUpdate(structSTIAcctUpdate* structAcctUpdate) void OnSTIAcctUpdateXML(BSTR* bstrAcct)

void OnSTIShutdown()

Methods
long GetQueueCount() long GetAccountList(BSTR() arrayAccts) STIAcctHRESULT
ClearAccountUpdateStruct(structSTIAcctUpdate* pAcctUpdate) HRESULT Destroy()

structSTIOrderFilter Struct (UDT)

Properties	Type	Description
bstrInstrument	BSTR	
bstrSymbol	BSTR	
bstrAccount	BSTR	
[illegible]	VARIANT_BOOL	

bstrSymbol BSTR bstrAccount BSTR

structSTITradeFilter Struct (UDT)

Properties Type Description
bstrInstrument BSTR

STIRejectReason Enums

Value Reject Description
0 rrSTIUnknown 1 rrSTIUnknownPid 2 rrSTIInvalidPassword 3 rrSTIAccessDenied 4 rrSTINotFound 5 rrSTICannotRoute 6 rrSTIPendingCancel 7 rrSTIPendingReplace 8 rrSTIOrderClosed 9 rrSTICannotCreate 10 rrSTIDupeClOrdId 11 rrSTINoSeqNoAvailable 12 rrSTIInvalidAcct

13	rrSTIInvalidDest	Sending a destination that the trader is not enabled for will trigger this.
14	rrSTIError	
15	rrSTIDupeSeqNo	
16	rrSTINoChange	
17	rrSTIInvalidSeqNo	
18	rrSTIInvalidQty	
19	rrSTITltc	Too late to cancel
20	rrSTIShareLimit	
21	rrSTIDollarLimit	
22	rrSTIBuyingPower	
23	rrSTITenSecRule	
24	rrSTINotSupported	
25	rrSTIDupeAcct	
26	rrSTIInvalidGroupId	
27	rrSTIDupeStation	
28	rrSTIPosTradingLmt	
29	rrSTITltcMoc	Too late to cancel MOC
30	rrSTIHardToBorrow	
31	rrSTIVersion	
32	rrSTIDupeLogin	
33	rrSTIInvalidSym	
34	rrSTINxRules	
35	rrSTIBulletNotRequired	
36	rrSTIMocMktImb	
37	rrSTINx30SecRule	
38	rrSTIEasyToBorrowOnly	
39	rrSTIStaleOrder	

Sterling ActiveX Object Library

SubmitOrder Error Codes Values

Value Error Description

0 No Errors
-1 Invalid Account
-2 Invalid Side
-3 Invalid Qty
-4 Invalid Symbol
-5 Invalid PriceType
-6 Invalid Tif
-7 Invalid Destination Not including a destination in the order will trigger this.
-8 Exposure Limit Violation
-9 NYSE+ Rules Violation
-10 NYSE+ 30-Second Violation
-11 Disable SelectNet Short Sales

-12 Long Sale Position Rules Violation Orders will not be split but a side change will occur
-13 Short Sale Position Rules Violation Orders will not be split but a side change will occur
-14 GTC Orders Not Enabled
-15 ActiveX API Not Enabled
-16 Sterling Trader® Pro is Offline
-17 Security Not Marked as Located
-18 Order Size Violation
-19 Position Limit Violation
-20 Buying Power /Margin Control Violation

-21 P/L Control Violation
-22 Account Not Enabled for this Product
-23 Trader Not Enabled for Futures
-24 Minimum Balance Violation
-25 Trader Not Enabled for odd lots open or cover transactions
-26 Order dollar limit exceeded
-27 Trader Not Enabled for Options
-28 Soft share limit exceeded

Sterling ActiveX Object Library

-29 Loss from max profit control violation (Title builds only) -30 Desk quantity enforcement violation -31 Account not enabled for Sell to Open (Options) -32 Account allowed to 'Close/Cxl' only -33 Trader not enabled for security locating -34 Order not able to be replaced (ReplaceOrder only) -35 Trader not enabled for 'Buy to Cover' -36 Invalid maturity date -37 Only one cancel and/or replace allowed per order per second -38 Account's maximum position value for this symbol exceeded -39 Symbol violates the account's min/max price settings
If a trader is subject to this condition they must first register the quote before an order can be sent. -40* Quote Unavailable-Order dollar limit -41* Quote Unavailable-Maximum Position Cost -42* Quote Unavailable-Buying Power -43* Quote Unavailable-Margin Control -44 Floating BP Violation -45 Market order would remove liquidity (Front end setting) -46 Not enabled for Server Stop orders -47 Not enabled for Trail Stop orders -48 Order would exceed the Max Open orders per side on this symbol -49 Compliance threshold exceeded or quote unavailable

*Note: Quote Unavailable will occur on MKT orders. A quote is needed to calculate the values for dollar limit, max position cost, buying power, margin control and the compliance threshold. In order to prevent this rejection simply register for the quotes on the symbol before you send the MKT order.

STIOrderStatus Enums

Value Order Status Description
0 osSTIUnknown 1 osSTIPendingCancel 2 osSTIPendingReplace 3 osSTIDoneForDay 4 osSTICalculated 5 osSTIFilled 6 osSTIStopped 7 osSTISuspended 8 osSTICanceled 9 osSTIExpired 10 osSTIPartiallyFilled 11 osSTIReplaced 12 osSTIRejected

Sterling ActiveX Object Library

13 osSTINew 14 osSTIPendingNew 15 osSTIAcceptedForBidding 16 osSTIAdjusted 17 osSTIStatused

STIPriceTypes Enums

Value Price Type Description
1 ptSTIMkt Market order 2 ptSTIMktClo Market on close order 3 ptSTIMktOb Market or better 4 ptSTIMktWow Market without waiting 5 ptSTILmt Limit 6 ptSTILmtClo Limit on close 7 ptSTILmtStp Stop order 8 ptSTILmtStpLmt Stop limit order 9 ptSTILmtOb Limit or better 10 ptSTIWow Without waiting 11 ptSTILmtWow Limit without waiting 12 ptSTIBas NYSE basis order 13 ptSTIClo Close order 14 ptSTIPegged Peg order 100 ptSTISvrStp Server side stop order 101 ptSTISvrStpLmt Server side stop limit order 102 ptSTITrailStp Trailing stop order

Side Values

Value Meaning
'B' = BUY 'C' = BUY TO COVER 'S' = SELL 'T' = SSHRT 'M' = BUY'P' = SELL+ 'E' = SSHRTEX

Tifs Values

Value Meaning
'D' = DAY 'G' = GTC 'X' = GTX 'F' = FOK 'I' = IOC 'O' = OPG 'E' = EXT '1' = OS 'A' = AEX (auto-x) 'N' = NOW

Sterling ActiveX Object Library

Action

Value Meaning Description
'A' = Add 'C' = Change 'D' = Delete 'S' =
Status

Exchange Definitions (Quotes & MDX) Values

Value Meaning
'A' American Stock Exchange 'B' Boston Stock Exchange 'C' National Stock Exchange 'D' ADFN (FINRA) 'I' ISE (Alpha Quote Feed) 'J' EDGA Exchange, Inc 'K' EDGX Exchange, Inc 'M' Chicago Stock Exchange (Midwest) 'N' New York Stock Exchange 'P' Pacific Stock Exchange (includes ARCA) 'T' NASDAQ (non-NASDAQ listed symbols) 'Q' NASDAQ (NASDAQ listed symbols) 'W' CBOE 'X' Philadelphia Stock Exchange 'Y' BATS Y-Exchange BYX '*' Composite (Equities) 'O' Composite (Options)

MaintainAccount Error Codes Values

Value Invalid Field
0 No Errors -1 Pro is offline -2 Traders are not allowed to maintain accounts -3 Invalid account -4 Manager is not entitled to change field

BBO (Best Bid or Offer) Values

Value Meaning
'B' = Best Bid 'A' = Best Ask 'X' = Both – Best Bid and Best Ask

Instrument Values

Value Meaning
"B" Bullet Order Bullet Trade "Non-B" Equity Order Equity
Trade "E" Equity

Sterling ActiveX Object Library

"O" Options "F" Futures "X" Forex

bstrHaltResumeStatus (Quotes) Values

Value Meaning
'1' Open Delay '2' Trading Halt '3' Resume '4' No Open /No Resume 'H' Halted 'Q' Quotation Only Period 'T' Trading On NASDAQ

bstrHaltResumeReason (Quotes) Values

Value	Meaning	
'D'	News Dissemination	
'E'	Order Influx	
'I'	Order Imbalance	
'J'	News Dissemination (due to related security)	
'K'	News Pending	
'M'	Additional Info	
'P'	News Pending	
'Q'	Due to Related Security	
'T'	Resume	
'V'	In View of Common	
'X'	Equipment Changeover	
'Y'	Sub-Penny Trading	
'Z'	No open/No resume	
'T1'	News Dissemination	
'T2'	Order Influx	
'T6'	Order Imbalance	
'T8'	News Dissemination	
'T12'	News Pending	
'H4'	Additional Info	
'H9'	News Pending	
'H10'	Additional Info	
'H11'	Resume	
'O1'	In View of Common	
'IPO1'	Equipment Changeover	
'M1'	Sub-Penny Trading	
'M2'	No open/No resume	
'T3'	News and Resumption Times	
'R4'	Qualifications Issues Reviewed/Resolved	Quotes/Trading to Resume
'R9'	Filing Requirements Satisfied/Resolved	Quotes/Trading To Resume
'C3'	Issuer News Not Forthcoming	Quotes/Trading To Resume
'C4'	Qualifications Halt ended	Maintenance Requirements Met Resume
'C9'	Qualifications Halt Concluded	Filings Met Quotes/Trades To Resume
'C11'	Trade Halt Concluded By Other Regulatory	Auth. Quotes/Trades Resume
'R1'	New Issue Available	
'R2'	Issue Available	

Sterling ActiveX Object Library

'IPOQ' IPO Security Released for Quotation (NASDAQ Securities Only) 'IPOE' IPO Security -Positioning Window Extension (NASDAQ Securities Only)

NxRules (NYSE+ Rules Enforcement) Values

Value Meaning
"0" Use Sterling Trader® Pro default settings (NYSE+ rules violation settings). "1" Convert the destination to NYSE if there is a NYSE+ rules violation. "2" Reject the order if there is a NYSE+ rules violation.

Special Order Designations Values

ExecInst Meaning
'E' DNI 'F' DNR 'G' AON 'M' Pegged Mid-Market 'P' Pegged Market 'R' Pegged Primary 'T' Pegged Best

ExecInst (ARCA only) Meaning

'2' Sweep Reserve '6' Post
No Preference

> **Note 1:** If you want more than one at a time, use them together separated by a single space between each. Order does
> not matter. Example: ExecInst = 'E F G' (for DNI, DNR, AON).

MDX Message Type Values

Value Meaning
1 Regulatory Imbalance 2
Informational Imbalance 3
Delay/Halt 4 Indication