TRADING TECHNOLOGIES®

XTAPI RTD Server Feature Guide

Version 7.17.0

Document Version 7.17.0 DV1 2/29/16

Legal Notices

This document and all related computer programs, example programs, and all TT source code are the exclusive property of Trading Technologies International, Inc. (“TT”), and are protected by licensing agreements, copyright law and international treaties. Unauthorized possession, reproduction, duplication, or dissemination of this document, or any portion of it, is illegal and may result in severe civil and criminal penalties.
Unauthorized reproduction of any TT software or proprietary information may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under the law.
Information in this document is subject to change without notice. Companies, names, and data used in examples herein are fictitious unless otherwise noted. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of TT.
All trademarks displayed in this document are subject to the trademark rights of TT, or are used under agreement by TT. These trademarks include, but are not limited to, service brand names, slogans and logos and emblems including but not limited to: Trading Technologies®, the Trading Technologies Logo, TT™, X_TRADER®, X_RISK®, MD Trader®, Autospreader®, TT_TRADER®, X_STUDY®, Autotrader™, ADL™, TT Trainer™, TTNET™, TT CVD®. All other referenced companies, individuals and trademarks retain their rights. All trademarks are the property of their respective owners. The unauthorized use of any trademark displayed in this document is strictly prohibited.

Copyright © 2011-2012 Trading Technologies International, Inc.
All trademarks are the property of their respective owners.

Table of Contents

Table of Contents 3

Chapter 1: Introducing the XTAPI RTD Server 5

Network and software overview 5
Excel notes 6
Formatting XTAPI values 8

Chapter 2: Creating Instrument IDs 9

About instruments and formulas 9
Defining an instrument equation 13
Examples of using the X_TRADER RTD generator for additional types of contracts 17
Specifying additional instrument options 19

Chapter 3: Creating Order Set IDs 23

About order sets and formulas 23
Defining an order set equation 25
Creating order set filters 27

Chapter 4: Requesting Values 29

Overview 29
Requesting an attribute that returns a single value 30
Requesting an attribute that returns an array of values 32
Requesting multiple attributes in a single equation 34
Creating a global order book 36
Creating a global fill book 38

Appendix A: Instrument Attributes 39

TTInstrObj object attributes 39

Appendix B: Order Set Attributes 57

TTOrderSet object attributes 57

Appendix C: Order Attributes 61

TTOrderObj object attributes 61

Appendix D: Fill Attributes 71

TTFillObj object attributes 71

1 Introducing the XTAPI RTD Server

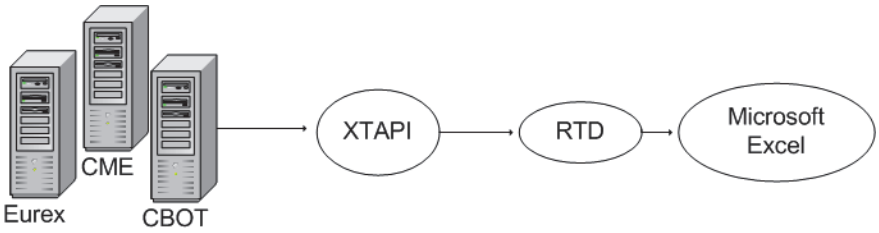
Network and software overview

XTAPI RTD Server network

XTAPI allows customers of Trading Technologies to leverage the functionality of TT's core network technology by building their own custom applications. Software applications that use the XTAPI can access live prices from every Exchange supported by TT, including:

- Access live prices from every exchange supported by TT Gateways
- Enter and modify orders
- Receive order acknowledgments and fills
- Calculate accurate P&L

The TT XTAPI RTD Server also uses the XTAPI to allow traders to display live data in an Excel spreadsheet. However, the XTAPI RTD Server does not support entering or modifying orders.



Software prerequisites

To use the TT XTAPI RTD Server, you must meet the following prerequisites:

- An X_TRADER®, X_TRADER® Pro, or TT_Trader™ application must be installed on your workstation.
- You must be able to access at least one TT Gateway with valid credentials.
- Microsoft Excel 2002 or newer must be installed on your workstation. This guide uses Microsoft Excel 2007 for its examples.

Additional resources

TT provides the following resources to help you develop XTAPI RTD Server applications:

- Sample Excel spreadsheets, located in **<root_drive>:\tt\x_trader\xtapi\samples**
- *XTAPI RTD Tutorial*, available from the TT DevNet website and the TT Customer Portal

Trading Technologies International, Inc. Proprietary and Confidential Information

Excel notes

Adjusting the Excel update throttling rate

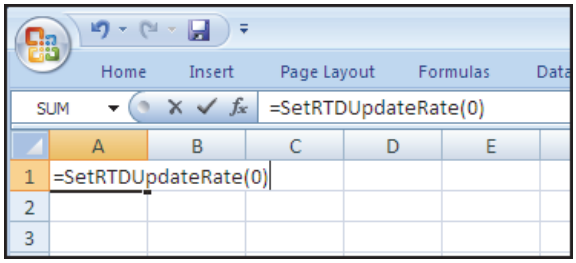
Excel automatically limits the rate at which it checks for updates. By default, it sets the throttling rate to 2000 milliseconds (2 seconds). In a fast-paced trading environment, such an update rate could, and likely would, result in stale data. TT recommends that you disable the default Excel throttling rate by setting its value to 0. With this value, Excel continuously checks for updates, ensuring that you have accurate and up-to-date values.

TT provides a function called `SetRTDUpdateRate()` that allows you to change the Excel throttling rate from within a spreadsheet.

Note: Changing the throttling rate with this function in any spreadsheet affects the default throttling rate for the current and all future Excel sessions.

To adjust the Excel throttling rate:

1. In your spreadsheet, place the cursor in a cell A1.
2. Enter the following formula to set the rate to 0 milliseconds, which disables throttling:



Some common error messages

The following table lists common Excel or RTD error conditions you might encounter.

Displayed text	Description
#NAME?	The formula in the cell contains an error, such as an invalid property name or an improperly formatted formula.
#N/A	The cell contains an invalid reference, or your formula is incorrectly formatted. For example: **=RTD("xtapi.rtd",,A1,A2,A3,A5,"Last")**, when A5 should be A4.
empty	For price data, the XTAPI RTD Server lost its connection to the Price Server. Because the existing data would no longer be valid, XTAPI RTD Server changes all prices to null strings and sets the Depth iterate to {}. An empty cell does not indicate a problem with a formula, just that the requested attribute has no value. For example, if there is no quantity exists on the Ask side, a request for AskQty$ returns an empty string. You can verify values with X_TRADER when developing formulas.

Stopping Excel from interpreting your values

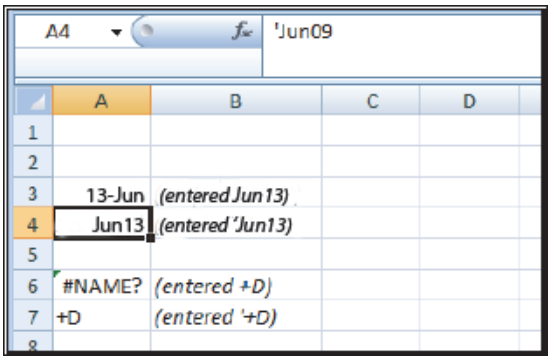
Both Excel and the XTAPI RTD Server allow you to use the values of cells in formulas, which gives you the ability to create dynamic spreadsheets. However, this functionality can create problems if you allow Excel to interpret the cell values for formatting purposes, which is Excel's default behavior.

Trading Technologies International, Inc. Proprietary and Confidential Information

If you use default cell formatting in a spreadsheet, Excel automatically formats the data based on its value. For example, if you enter 15 into a cell with default formatting, Excel interprets and formats it as a number instead of a string. When requesting certain types of information from the XTAPI RTD Server, you must specify the values with specific formats. The name of a contract, for example, might look like a date, such as Jun13 for the ES Jun13 contract on the TT CME Gateway. If you enter the string **Jun13** in a cell with default formatting, Excel interprets the text as a date and displays the value as **13-Jun**, which would not match the name of the contract.

Likewise, some of the optional arguments for instruments and order sets begin with a plus sign (**+**). When Excel sees the plus sign, it assumes you intend to perform some calculation. If you enter something like **+D** in a cell that you use in a formula, Excel interprets it as an invalid calculation and displays **#NAME?** in the cell.

To stop Excel from interpreting and formatting your data, you precede your text with a single quote ('). The following demonstrates the effect of using the single quote.



Tip: You can also set the Excel cell format to **Text** to stop Excel from reformatting the date.

Trading Technologies International, Inc. Proprietary and Confidential Information

Formatting XTAPI values

Specifying value formats for requested data

Some of the attributes allow you to specify the format for price and quantity data. The XTAPI provides certain prices and quantities in several formats. When specifying an attribute that deals with prices or quantities, you can append a special character to the end of the attribute name to specify the format for the data. For example, price data can be represented in ticks or points.

The following table describes the different formats you can use.

Value type	Character	Description
Price	&	Ticks (integer)
	#	Points (decimal)
	$	X_TRADER® display format (string)
	~	Differential for inverted tick products (string). Used for trading BrokerTec yield products, because the "$" format cannot differentiate between an actual price (LTP) or a price delta (net change).
	.^	Native currency (decimal)
	@	Primary currency (decimal)
Quantity	&	Number (integer)
	#	Number (decimal)
	$	Number (string)

Note: Some market data, such as implied prices, might not always be available when you request it. In such cases, the XTAPI RTD Server returns a null value instead of the normal data type. TT recommends that your application checks for null values before using an attribute value.

Some examples

The following table illustrates how you can use the attribute modifiers to format the return values.

Specifying...	Returns...
Price	Current price in the X_TRADER display format
Price#	Current price represented as points
Limit.^	Limit price represented in the native currency, as defined in the Guardian Price Table
BidQty	Number of bid offers at a price, expressed as an integer
AskQty$	Number of ask offers at a price, converted to a string value

Tip: When using different values in combination, such as performing calculations, you should use the format modifiers to normalize the data.

Trading Technologies International, Inc. Proprietary and Confidential Information

2 Creating Instrument IDs

About instruments and formulas

XTAPI instrument objects

The XTAPI RTD Server works in conjunction with the X_TRADER API (XTAPI) to access information about instruments from TT Gateways. The XTAPI stores information about instruments in its **`TTInstrObj`** objects, which you can access through Excel `RTD()` instrument formulas.

RTD() formulas for instruments

When requesting updates for instrument-related information from the XTAPI RTD Server, you typically call the `RTD()` function in one of the following forms.

To...	Use the following formula
Return an Instrument ID	`=RTD(`*`ServerID`*`,`*`ServerName`*`,`*`Topic`*`,`*`Exchange`*`,`*`Product`*`,`*`ProductType`*`,`*`Contract`*`,`*`InstrumentOptions`*`)`
Request an attribute value using contract qualifiers	`=RTD(`*`ServerID`*`,`*`ServerName`*`,`*`Topic`*`,`*`Exchange`*`,`*`Product`*`,`*`ProductType`*`,`*`Contract`*`,`*`InstrumentOptions`*`,`*`AttributeName`*`)`
Request a single instrument attribute value using an Instrument ID	`=RTD(`*`ServerID`*`,`*`ServerName`*`,`*`InstrumentID`*`,`*`AttributeName`*`)`
Request multiple values for an Instrument ID	`=RTD(`*`ServerID`*`,`*`ServerName`*`,`*`InstrumentID`*`,`*`AttributeName`*`,`*`AttributeName`*`,`*`AttributeName`*`,`*`DataDisplayOptions`*`)`

The remainder of this chapter shows how to use these different types of formulas to create instrument IDs and retrieve basic instrument data.

Parameters for instrument RTD() formulas

The following table describes these formula parameters.

Parameter	Description
`ServerID`	Program ID of the local RTD Server. For the XTAPI RTD Server; you must use the value, **"xtapi.rtd"**
`ServerName`	Name of the external server running the RTD Sever. As the XTAPI RTD Server always runs locally, you must omit a value for this parameter or supply an empty string (“”). However, you must account for the parameter in the formula, as shown: Valid: `RTD(“xtapi.rtd”,””,”Instr”...)` Valid: `RTD(“xtapi.rtd”,,”Instr”...)` Invalid: `RTD(“xtapi.rtd”,”Instr”...)`
`Topic`	Type of object to query. When querying instruments, use **Instr**.
`Exchange`	TT Gateway name, including the flavor, such as **CME** or **CME-A**.
`Product`	Product symbol, such as **ES**.
`ProductType`	Product type, as it appears in an X_TRADER Market Explorer window, such as **FUTURE**.

Trading Technologies International, Inc. Proprietary and Confidential Information

Parameter	Description
Contract	Contract name, as it appears in the X_TRADER Market Grid.
InstrumentOptions	Options that provide access to additional information related to an instrument. Possible values include: **"+D"**: Provides access to market depth updates for an instrument. **"+DD"**: Provides access to detailed depth updates for an instrument, if the market supports detailed depth. For more information, refer to the section called More About Detailed Depth. **"+OS"**: Provides access to instrument updates contained in an order set (such as orders, positions, and P&L). **"+OS.LIFO"** or **"+OS.FIFO"**: Specifies which algorithm to use when you request average price updates for an instrument. **"+OS.*Attribute*=*Value*"** or **"+OS.*Attribute*<>*Value*"**: Allows you to filter instruments based on attribute values for the order set containing the instrument. **"+OS.FILLS"**: Provide access to fill updates associated with the instrument. **"+OS.FILLS"**: Provide access to fill updates associated with the instrument. **"+TS=*Value*"**: Provides access to the specified number of Time and Sales updates that involve this instrument (range=1-500; default=100). Each update includes the following fields: Time, Quantity, Price, Market Side, OTC, and, optionally, the Bid Member and Bid Ask counterparty member fields. When using this option, you must specify **TIMEANDSALES** for the *AttributeName* in the formula. For X_TRADER 7.11.3 and earlier, the RTD returns integer values for the Bid Member and Bid Ask fields. In X_TRADER 7.11.4 and later, you can choose to return these fields as integers, as the full name of the counterparty, or as the alias defined for the counterparty in X_TRADER. **"-M"**: Disables merging of implied prices and quantities with the direct prices and quantities when requesting updates for the **Bid**, **Ask**, **BidQty**, and **AskQty** attributes. By default, the XTAPI RTD Server merges implied and direct values. **"-I"**: Disables using the TT Implieds Engine to calculate implied prices and quantities. By default, the XTAPI RTD Server uses the TT Implieds Engine to calculate implied values.
	Note: The **+DD** (detailed depth) and **+TS** (time and sales) options are available only in XTAPTI RTD 7.7.7 or later. Additionally: • You cannot use the Instrument keys returned by formulas using these options as Instrument key inputs to VBA objects. • You cannot access the data directly through the XTAPI; to access it, you must use RTD.
InstrumentID	Unique internal TT instrument identifier returned from another `RTD()` formula that specifies the contract qualifiers. **Note**: The XTAPI RTD Server generates this number each time it loads the spreadsheet. Consequently, do not store and reuse this ID for later use.

Trading Technologies International, Inc. Proprietary and Confidential Information

Parameter	Description
AttributeName	Name of any instrument attribute returned by the **TTInstrObj** `Get` property. For a comprehensive list of possible attribute values, refer to Appendix A, Instrument Attributes or to the **TTInstrObj Property: Get** section in the *XTAPI Class Reference*.
DataDisplayOptions	Optional cell addresses to use for displaying array data. The XTAPI RTD Server supports Excel-style cell address, such as: "A1","B1","C1" CELL("Address",A1)

Trading Technologies International, Inc. Proprietary and Confidential Information

More About Detailed Depth

Note: Detailed depth is supported only for XTAPI RTD 7.7.7 or later.

You can access detailed depth for an instrument through RTD, but not through the XTAPI. You can enable either market depth (+D) or detailed depth (+DD) for an instrument reference, but not both.

When you specify detailed depth for an instrument that supports it, RTD returns values for the following Instrument attributes. Each row returned represents a distinct order resting in the market. For descriptions of these Instrument attributes, refer to Appendix A, Instrument Attributes.

- AskDepth(*levels*)
- AskDepth90(*levels*)
- BidDepth(*levels*)
- BidDepth90(*levels*)
- PriceDepth(*levels*)
- PriceDepth90(*levels*)
- XTDepth(*levels*)
- XTDepth90(*levels*)

If you request detailed depth for an instrument that does not support it, RTD returns blank values for all cells referencing detailed depth fields. RTD does not support detailed depth for implied, indicative, or theoretical prices.

Trading Technologies International, Inc. Proprietary and Confidential Information

Defining an instrument equation

Calling the RTD() formula

To define an instrument equation, use the following Excel `RTD()` formula syntax:

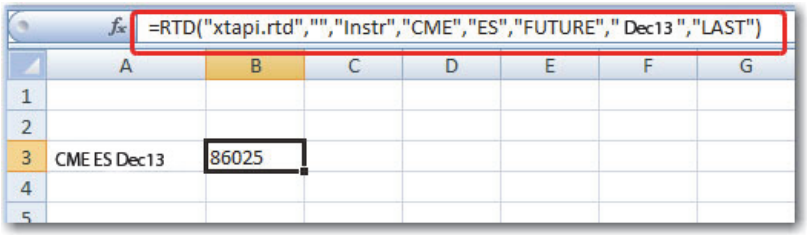
```
=RTD("xtapi.rtd",,"Instr", Exchange, Product, ProductType, Contract,
AdditionalInstrParams)
```

Specifying the instrument identifiers and an attribute in a single equation

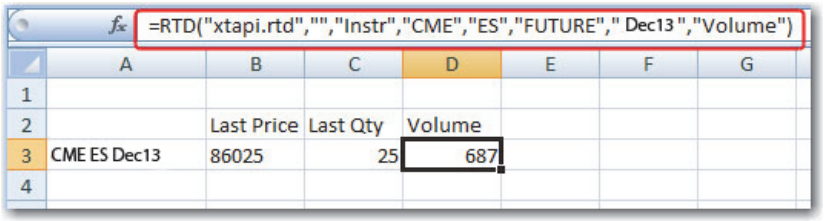
If you know the instrument identifiers for a contract, you can enter them directly in the `RTD()` formula and specify which instrument attribute you want to retrieve. For example, suppose you want to know the last price for the ES Dec13 future trading on the TT CME Gateway. You could enter the following formula in a cell, as follows.

```
=RTD("xtapi.rtd","","Instr","CME","ES","FUTURE","Dec13","Last")
```

After you enter the formula, the XTAPI RTD Server retrieves the value of the instrument's **Last** attribute and displays its value in the cell (**B3**, in this example).



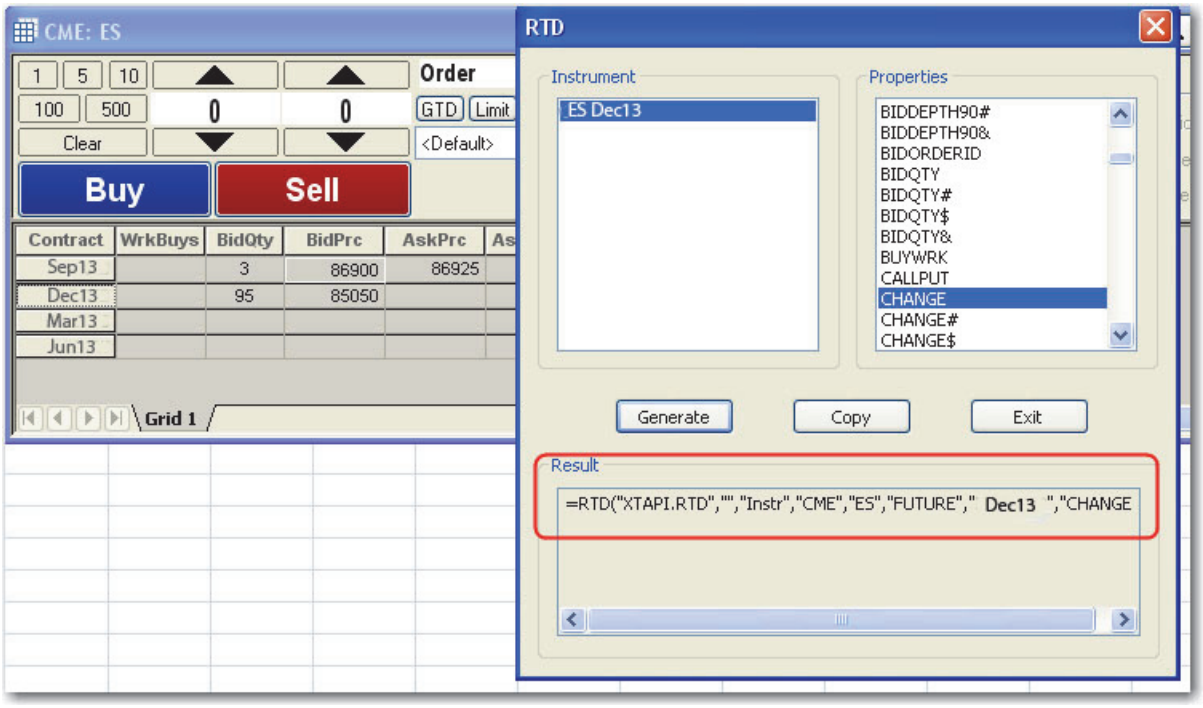
You can enter multiple formulas in multiple cells to get a broader picture of the market. For example, you could display additional information about the contract in adjacent cells. The following adds the quantity traded at the last price and the total traded volume for the contract for the trading day.



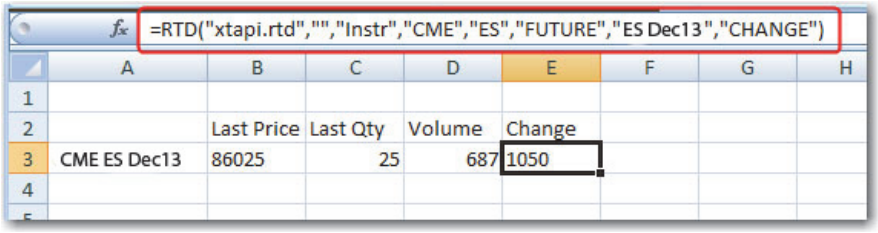
Generating an formula from X_TRADER

If you are unsure of the instrument identifiers or the name of the instrument you want, you can use the RTD Generator within X_TRADER to create the formula you need to enter in a cell. The following example illustrates the RTD formula it generates for the VOLUME property of the CME ES Dec09 contract.

Trading Technologies International, Inc. Proprietary and Confidential Information



You can access the RTD Generator from the context menu of any contract shown in the **Market Grid**, then copy the formula and paste it into a cell in your spreadsheet.



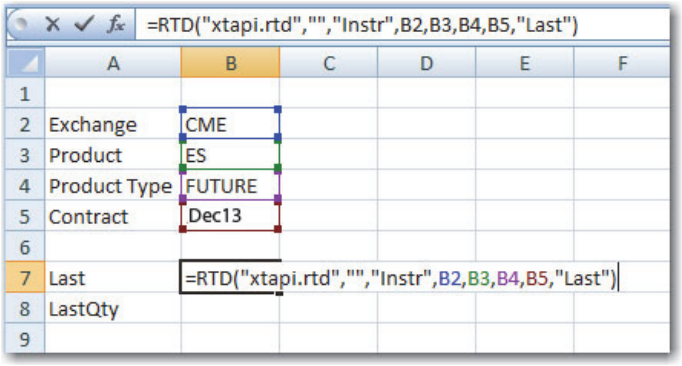
For more information about using the X_TRADER RTD Generator, refer to the X_TRADER help system.

Referencing instrument identifiers stored in other cells

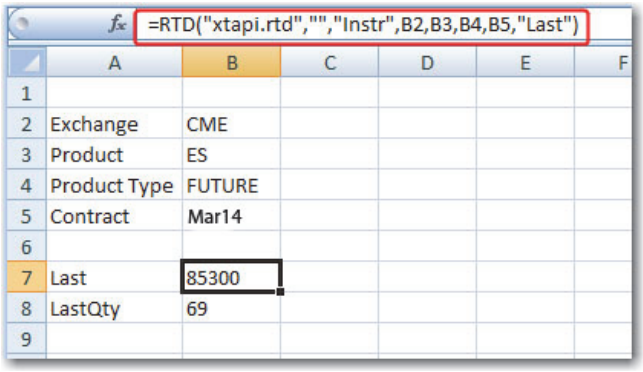
Including an attribute in the formula provides the desired results, but does not provide any flexibility. Because you specify all information in every formula, you must update every formula if you want to monitor a different contract or add additional parameters, such as requesting Market Depth or Fill information. To create a spreadsheet that you can update dynamically, you can leverage Excel's support for referencing the values from other cells in a formula. Instead of specifying the actual value in the `RTD()` formula, you can specify other cell addresses.

The following example shows one way you could enter the instrument identifiers in individual cells. Cell **B7** contains the Last price of the whatever contract is defined in cells **B2** through **B5**.

Trading Technologies International, Inc. Proprietary and Confidential Information



If you want to check another contract, you need only enter new values in any of these cells. The **Last** price automatically updates, without needing to create a new formula. For example, if you want to check the information for the Mar14 contract, you would just change the value in cell **B5**, as shown.



Notes:

The values you specify must exactly match those used in X_TRADER.

When entering **Mar14** in a cell, Excel interprets it as a date and displays it to 10-Mar, which does not match the name of the contract. To prevent Excel from interpreting the value, you can either:

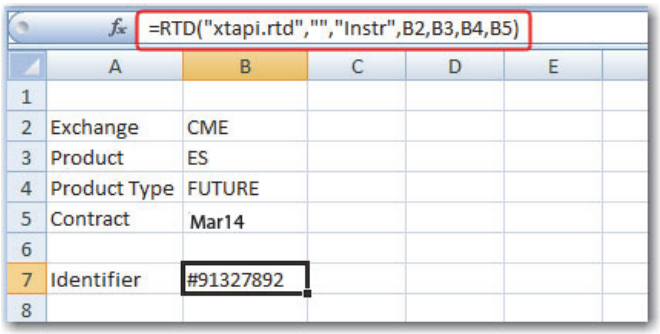
Precede the string with a single quote, as in **'Mar14**.

Change the format of the cell to type **Text**.

If you mis-type a value, Excel displays **#N/A** to indicate your formula contains an error.

Storing and reusing an instrument ID

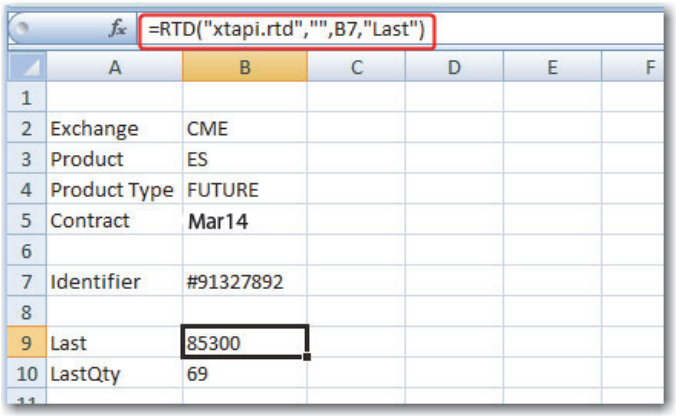
You can make your spreadsheet even more flexible by storing and reusing the ID's that the XTAPI RTD Server returns for each contract. If you omit the attribute from the RTD() formula, as follows, the XTAPI RTD Server returns a unique ID for the instrument.



Trading Technologies International, Inc. Proprietary and Confidential Information

When you have an instrument ID, you can use a simplified form of the `RTD()` formula, where you reference the cell containing the ID:

```
=RTD("xtapi.rtd","",B7,"Last")
```

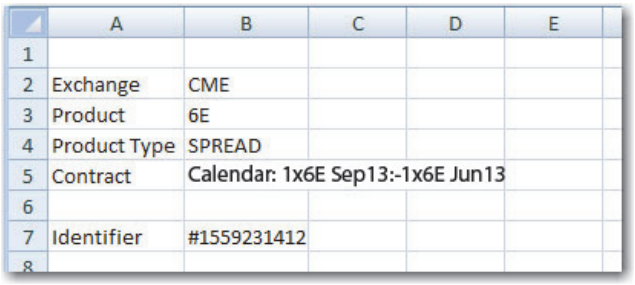


Trading Technologies International, Inc. Proprietary and Confidential Information

Examples of using the X_TRADER RTD generator for additional types of contracts

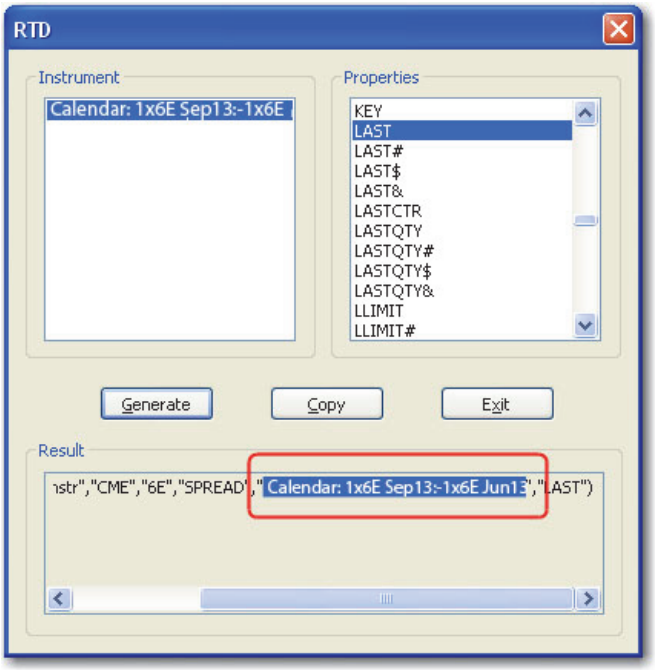
Specifying a spread contract

The following example shows how to specify a spread contract used by the instrument ID generation formula, specifically the 6E-Sep09/Jun09 calendar spread on the TT CME Gateway.



	A	B	C	D	E
1					
2	Exchange	CME			
3	Product	6E			
4	Product Type	SPREAD			
5	Contract	Calendar: 1x6E Sep13:-1x6E Jun13			
6					
7	Identifier	#1559231412			
8					

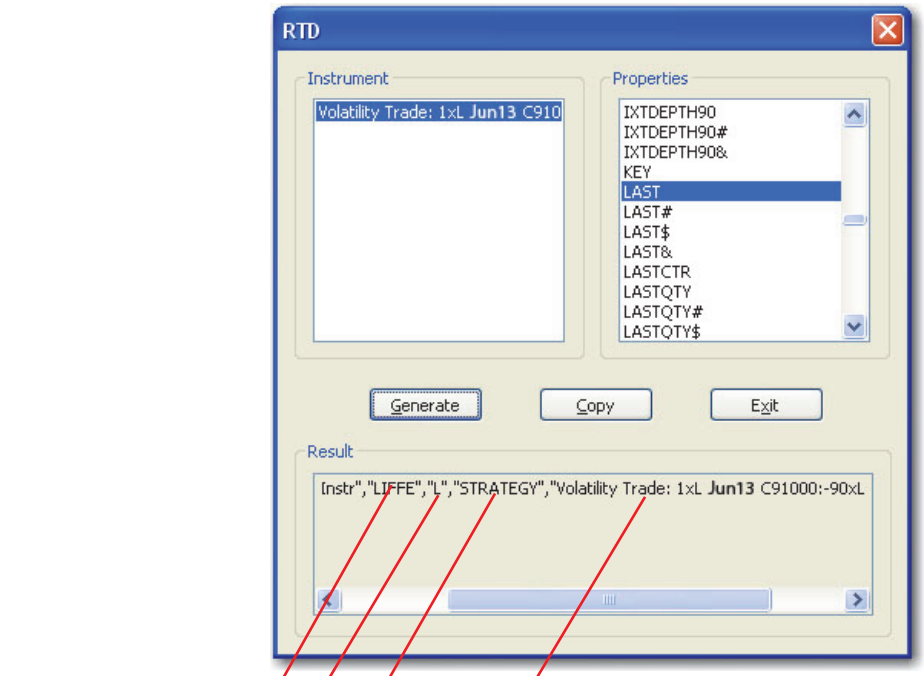
Unlike the names used for futures contracts, spread contracts typically use lengthy names with special syntax. Because contract names must exactly match the name used by X_TRADER (including spaces and special characters), you can take advantage of the X_TRADER RTD generator to get the correct contract name. You can generate a formula for the instrument and a property, and copy the string shown for the contract name. Then you can simply paste the name into the appropriate cell in the spreadsheet.



Specifying a strategy contract

The following example shows how you can use the RTD generator to get the values to use for the instrument identifier fields in your spreadsheet. in this case, you use the values generated for a Sterling (L) strategy contract trading on the TT LIFFE Gateway to populate the corresponding cells in your spreadsheet.

Trading Technologies International, Inc. Proprietary and Confidential Information



	A	B	C	D	E	F
1						
2	Exchange	LIFFE				
3	Product	L				
4	Product Type	STRATEGY				
5	Contract	Volatility Trade: 1xL Jun13 C91000:--90xL Jun13				
6						
7	Identifier	#1559231412				

Trading Technologies International, Inc. Proprietary and Confidential Information

Specifying additional instrument options

Available options

When you create instrument ID formulas, you can specify the following options to provide a broader view of the market for an instrument:

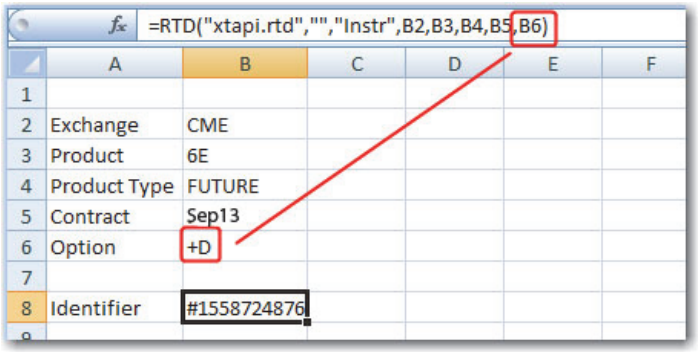
- **+D**, which provides access to market depth information for an instrument
- **+OS**, which allows you to retrieve information about orders associated with an instrument

Market depth

The XTAPI RTD Server does not access market depth unless you specifically request it. To enable market depth access, you include the **+D** instrument option in the `RTD()` formula, as follows:

```
=RTD("xtapi.rtd",,"Instr","CME","6E","FUTURE","Sep13","+D")
```

You can also expand the spreadsheet to allow you enable market depth, as follows:



Notes:

When entering **+D** in a cell, Excel interprets it as a mathematical expression instead of a string. To prevent Excel from interpreting the value, you can:

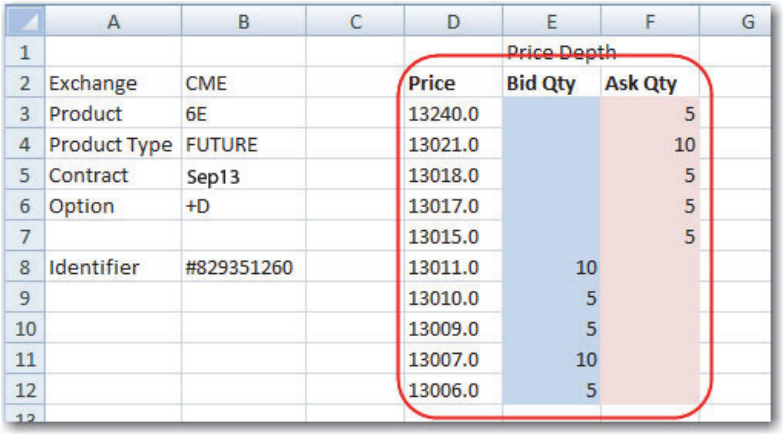
Precede the string with a single quote, as in **'+D**.

Change the format of the cell to type **Text**.

If you do not override the default formatting, Excel displays **#NAME?** because the value is not valid for an instrument option.

When you enable market depth for an instrument, you can access additional information about an instrument, such as the Bid and Ask quantities at different price levels. With market depth enabled, you can create a grid that displays the information similar to the following. This example uses the **PriceDepth** attribute to return an array of data for five levels of depth on each side of the market.

Trading Technologies International, Inc. Proprietary and Confidential Information



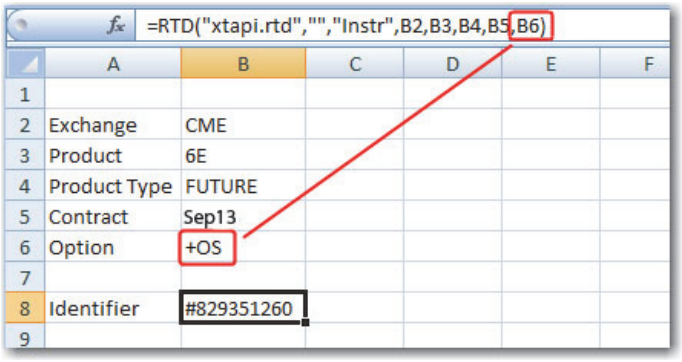
	A	B	C	D	E	F	G
1					Price Depth		
2	Exchange	CME		Price	Bid Qty	Ask Qty	
3	Product	6E		13240.0		5	
4	Product Type	FUTURE		13021.0		10	
5	Contract	Sep13		13018.0		5	
6	Option	+D		13017.0		5	
7				13015.0		5	
8	Identifier	#829351260		13011.0	10		
9				13010.0	5		
10				13009.0	5		
11				13007.0	10		
12				13006.0	5		

For more information about retrieving array data, refer to Requesting an attribute that returns an array of values, on page 32 or to the *XTAPI RTD Tutorial*.

Accessing order-related information for an instrument

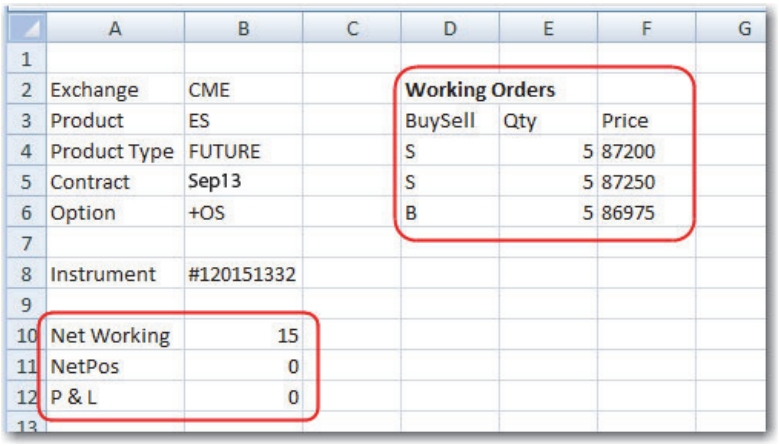
Some instrument data, such as net position or P&L, is valid only when an instrument is part of a working order. To access this type of information, the XTAPI RTD Server needs to access the orders that reference an instrument. Using the **+OS** option when you create an instrument ID, as follows, allows you to retrieve information from order sets.

```
=RTD("xtapi.rtd",,"Instr","CME","6E","FUTURE","Sep13","+OS")
```



When you enable order set access, you can use the XTAPI RTD Server to get a broader picture of the instrument in the market, and you can access information about working orders and fills. The following example shows some of additional information you can access when you use the **+OS** option.

Trading Technologies International, Inc. Proprietary and Confidential Information



	A	B	C	D	E	F	G
1							
2	Exchange	CME		**Working Orders**			
3	Product	ES		BuySell	Qty	Price	
4	Product Type	FUTURE		S	5	87200	
5	Contract	Sep13		S	5	87250	
6	Option	+OS		B	5	86975	
7							
8	Instrument	#120151332					
9							
10	Net Working	15					
11	NetPos	0					
12	P & L	0					
13							

For a list of values you can retrieve from an order set, refer to Appendix B, Order Set Attributes, on page 57. Also, Chapter 4, Requesting Values provides more information and examples for requesting information from order sets.

Trading Technologies International, Inc. Proprietary and Confidential Information

Trading Technologies International, Inc. Proprietary and Confidential Information

3 Creating Order Set IDs

About order sets and formulas

XTAPI order set objects

The XTAPI RTD Server accesses information about order sets through the XTAPI **`TTOrderSet`** objects, which you can access through Excel `RTD()` order set formulas.

RTD() formulas for order sets

When requesting updates for order set information from the XTAPI RTD Server, you typically call the `RTD()` function in one of the following forms.

To...	Use the following formula
Return an Order Set ID	`=RTD(`*`ServerID`*`,`*`ServerName`*`,`*`Topic`*`,`*`OrderSetOptions`*`)`
Request an attribute value using order set options	`=RTD(`*`ServerID`*`,`*`ServerName`*`,`*`Topic`*`,`*`OrderSetOptions`*`,`*`AttributeName`*`,`*`DataDisplayOptions`*`)`
Request a single instrument attribute value using an Instrument ID	`=RTD(`*`ServerID`*`,`*`ServerName`*`,`*`OrderSetID`*`,`*`AttributeName`*`)`
Request multiple values for an OrderSet ID	`=RTD(`*`ServerID`*`,`*`ServerName`*`,`*`OrderSetID`*`,`*`AttributeName`*`,`*`AttributeName`*`,`*`AttributeName`*`,...,`*`DataDisplayOptions`*`)`

The remainder of this chapter shows how to use these different types of formulas to create order set IDs and retrieve basic data from the order set.

Parameters for order set RTD() formulas

The following table describes these function parameters.

Parameter	Description
`ServerID`	Program ID of the local RTD Server. For the XTAPI RTD Server; you must use the value, **"xtapi.rtd"**
`ServerName`	Name of the external server running the RTD Sever. As the XTAPI RTD Server always runs locally, you should omit this parameter or supply and empty string (“”).
`Topic`	Type of object to query. When querying order sets, use **OS**.
`OrderSetOptions`	Optional parameters, including: **+OS.*Attribute*=*Value*** and **+OS.*Attribute*<>*Value***, which allow you to create filters based on order attributes. For more information about using these filters, refer to Creating order set filters, on page 27. **"+OS.Fills**, which causes the XTAPI RTD Server to retain fill records.
`OrderSetID`	Unique internal TT order set identifier returned from another `RTD()` formula that defines an order set. **Note**: The XTAPI RTD Server generates this number each time it loads the spreadsheet. Consequently, do not store and reuse this ID for later use.

Trading Technologies International, Inc. Proprietary and Confidential Information

Parameter	Description
AttributeName	Name of any order set attribute returned by the **TTOrderSet** Get property. For a comprehensive list of possible attribute values, refer to Appendix B, Order Set Attributes or to the **TTOrderSet Get** section in the *XTAPI Class Reference*.
DataDisplayOptions	Optional cell addresses to use for displaying array data. The XTAPI RTD Server supports Excel-style cell address, such as: "A1","B1","C1" CELL("Address",A1)

Trading Technologies International, Inc. Proprietary and Confidential Information

Defining an order set equation

Identifying an order set

When requesting information from order sets, you identify an order set as shown in the `RTD()` function call:

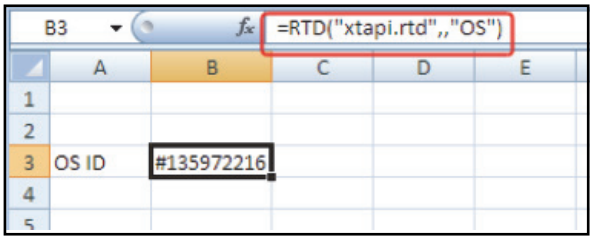
```
=RTD("xtapi.rtd",,"OS",OrderSetOptions)
```

For example, if you want to generate an order set ID that can access all information in the order set, you would specify the formula as follows.

```
=RTD("xtapi.rtd",,"OS")
```

Storing an order set ID

When you use an `RTD()` formula that contains only an OS topic and order set options, the XTAPI RTD Server generates a unique ID for the order set and returns it as a result in the cell. The following illustration shows how to return an order set ID that contains all information in the order set.



Cell **B3** now contains an ID that uniquely identifies the order set. You can use this value in place of all the contract qualifiers in other formulas, as described later.

Note: The XTAPI RTD Server generates this ID each time it opens the spreadsheet and ensures that it uniquely identifies the order set through the current session. However, it does not ensure that the actual identifier remains the same each time you open the spreadsheet. Consequently, you should not use the ID directly in formulas; but you can reference the cell that contains the ID.

Accessing order set data through the XTAPI TTOrderSet object

After you define an order set, you can specify order set attribute names as parameters in the `RTD()` formula, using either of the following methods:

```
=RTD("xtapi.rtd",,"OS",OrderSetOptions,AttributeName)
```

```
=RTD("xtapi.rtd",,OrderSetID,AttributeName)
```

For example, if you want to determine the net position for an order set, you would specify the function parameters as follows. For the second case, assume cell **B3** contains the Order Set ID.

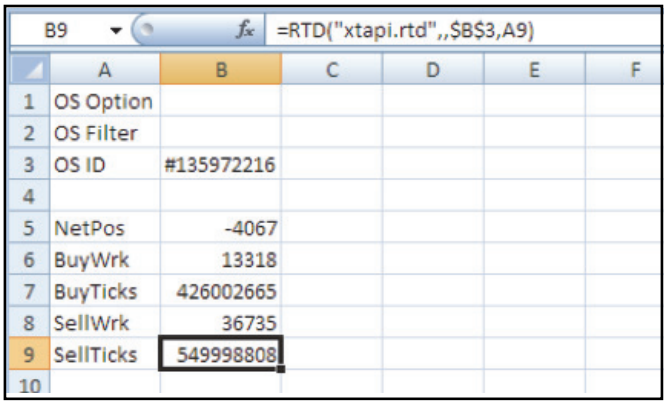
```
=RTD("xtapi.rtd",,"OS","NetPos")
```

```
=RTD("xtapi.rtd",,B3,"NetPos")
```


Trading Technologies International, Inc. Proprietary and Confidential Information

Retrieving data for the whole order set

When an order set formula contains no options or filters, the XTAPI RTD Server automatically defines an order set that contains all orders visible to the current trader login. The following example returns the order set attributes specified in columns **A5**-**A9**, calculated for all visible orders.



	A	B	C	D	E	F
1	OS Option					
2	OS Filter					
3	OS ID	#135972216				
4						
5	NetPos	-4067				
6	BuyWrk	13318				
7	BuyTicks	426002665				
8	SellWrk	36735				
9	SellTicks	549998808				
10						

Trading Technologies International, Inc. Proprietary and Confidential Information

Creating order set filters

Creating filters based on order attributes

The *OrderSetOptions* parameter enables you to filter information within an order set, based on the values of order attributes. When a trader places an order for an instrument, the XTAPI uses an order set as the mechanism for submitting the order to the Exchange. Because an order set can contain many instruments or many orders involving the same instrument, you can use attributes of the order to limit the instruments for which the XTAPI RTD Server returns values. For example, you might want the last price (**Last** instrument attribute) of an instrument only if it was submitted through a specific account (**Acct** order attribute).

Combining multiple filters

When you use multiple name/value pairs in a formula, the XTAPI RTD Server applies Boolean logic to the values based on whether you specify the same or different attributes and whether you test for equalities or inequalities. The following table explains how the XTAPI RTD Server determines which logic to use when finding matching records.

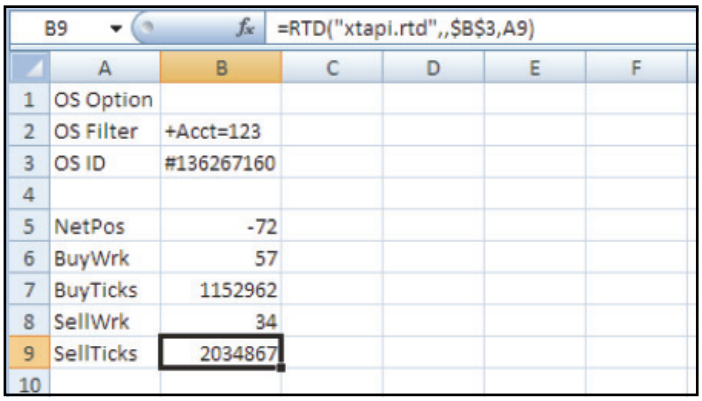
Attributes	Operators	Logic	Examples
Same	All **=**	Boolean **OR**	`"'+OS.Acct=JDOE","'+OS.Acct=BSMITH"` Returns records for both JDOE and BSMITH accounts
Same	All **<>**	Boolean **AND**	`"'+OS.Acct<>JDOE","'+OS.Acct<>BSMITH"` Returns records for all accounts except JDOE and BSMITH
Same	Mixed	Boolean **AND**	`"'+OS.Acct=JDOE","'+OS.Acct<>BSMITH"` Returns records for both JDOE
Different	All **=**	Boolean **AND**	`"'+OS.Acct=JDOE","'+OS.ClrMember<>ABC"` Returns records for both JDOE
Different	All **<>**	Boolean **OR**	`"'+OS.Acct<>JDOE","'+OS.ClrMember<>ABC"` Returns records for all accounts except JDOE and all clearing members except ABC
Different	Mixed	Boolean **OR**	`"'+OS.Acct<>JDOE","'+OS.ClrMember=ABC"` Returns records for all accounts except JDOE and all records with clearing member ABC

Trading Technologies International, Inc. Proprietary and Confidential Information

Requesting data for a filtered order set

You can use order set filters to create smaller, more focused order sets. For example, you could define an order set that contains orders only for a specific account or a single contract. As a reminder, you can use any of the `TTOrderObj` attributes listed in Appendix C, Order Attributes in the **OS Filter** field.

The following example shows the same data, but limits it only to the orders submitted with the **Acct** attribute set to **123**. The order set ID formula (cell **B3**) in this spreadsheet uses the values in cells **B1** and **B2**, if specified, when defining the order set.



Trading Technologies International, Inc. Proprietary and Confidential Information

4 Requesting Values

Overview

Types of data you can request

The XTAPI RTD Server allows you to request values from the XTAPI by accessing attributes associated with certain objects. You can get values for the following:

- Instrument information through the XTAPI **`TTInstrObj`** attributes
- Order set information through the XTAPI **`TTOrderSet`** attributes
- Order information through the XTAPI **`TTOrderObj`** attributes
- Fill information through the XTAPI **`TTFillObj`** attributes

Ways to request values

When you request information from the XTAPI RTD Server, you need to take into account not only the object, but also the type of data you want. This chapter shows you how to access different types of information in the following ways:

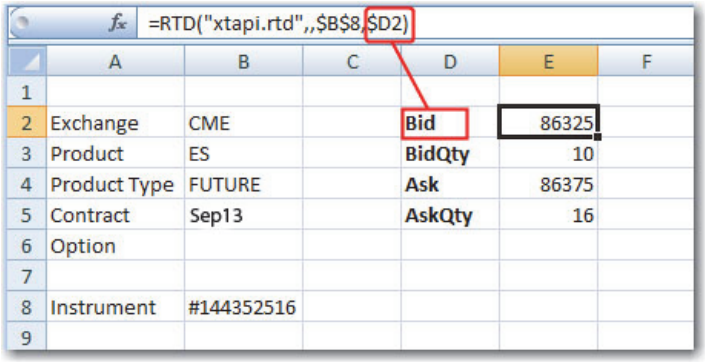
- Requesting an attribute that returns a single value
- Requesting an attribute that returns an array of values
- Requesting multiple attributes in a single equation
- Creating a global order book
- Creating a global fill book

Trading Technologies International, Inc. Proprietary and Confidential Information

Requesting an attribute that returns a single value

Requesting an instrument's bid price

The following example shows how to request the Bid price for the ES Sep13 contract from the TT CME Gateway. The formula for cell **E2** uses the text entered in cell **D2** to specify the data to retrieve (the **Bid** attribute, in this example).

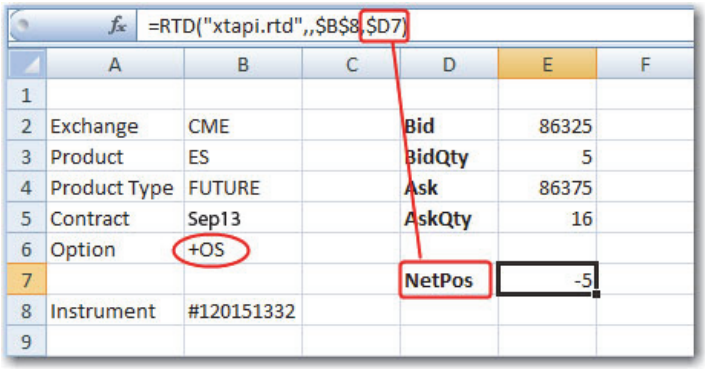


Note: Because this example does not use the **+D** or **+OS** options when generating the Instrument ID, it can request only inside market updates.

For a complete list of instrument attributes you can request, refer to Appendix A, Instrument Attributes.

Requesting the net position for an instrument

The following example shows how to request the net position for the ES Sep13 contract on the TT CME Gateway. Because the net position for an instrument requires access to working orders, the Instrument ID formula must include the **+OS** option so the XTAPI RTD Server can access information contained in order sets.



In this example, the formula in cell **B8** used the following formula to include the **+OS** option specified in cell **B6**.

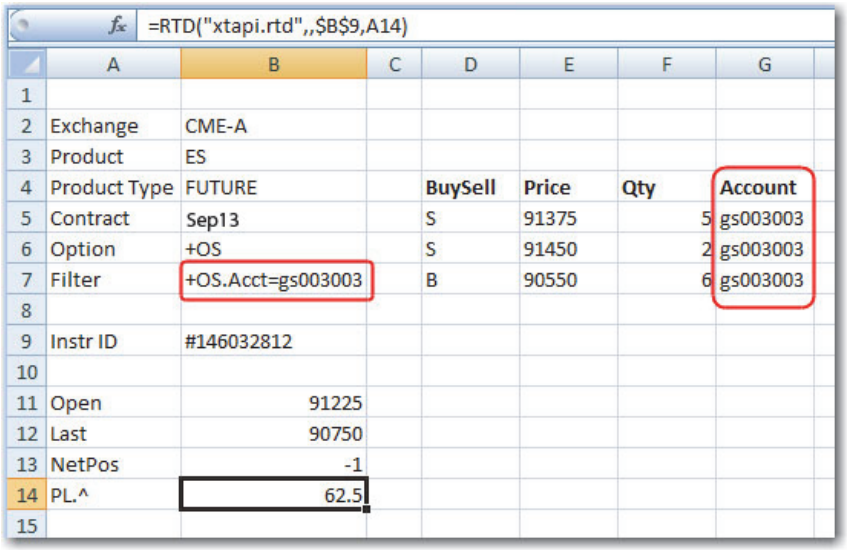
```
=RTD("xtapi.rtd",,"Instr",B2,B3,B4,B5,B6)
```

Requesting filtered P&L

When you enable the **+OS** option, you gain access to attributes of working orders in the order set. You can use the order attributes to define filters that limit which orders from which the XTAPI RTD Server retrieves values.

The following example displays the net position and P&L for the ES Sep09 contracting trading on the TT CME-A Gateway for only those orders submitted through the gs003003 account. The **PL.^** attribute displays the P&L, calculated with the last traded price (**Last** attribute), in the native currency.

Trading Technologies International, Inc. Proprietary and Confidential Information



Tip: You can specify multiple optional arguments to create more detailed filters.

For a list of order attributes you can use as filters, refer to Appendix C, Order Attributes.

Trading Technologies International, Inc. Proprietary and Confidential Information

Requesting an attribute that returns an array of values

How Excel handles array data

The `RTD()` function call you use to request market depth looks much like the one you use to request single values, except that in addition to the instrument attribute, you specify a place in memory to store the data. The following example shows how to request five levels of market depth and store it in an in-memory array:

```
=RTD("xtapi.rtd",,$C$8,"PriceDepth(5)",CELL("Address",A1))
```

You determine the amount of data to request from the XTAPI by the number you specify for the **PriceDepth** attribute. In this example, you specify **PriceDepth(5)**; then the X_TRADER API RTD Server returns five levels of depth on each side of the market, where column **A** contains the prices, column **B** contains the Bid quantities, and column **C** contains the Ask quantities.

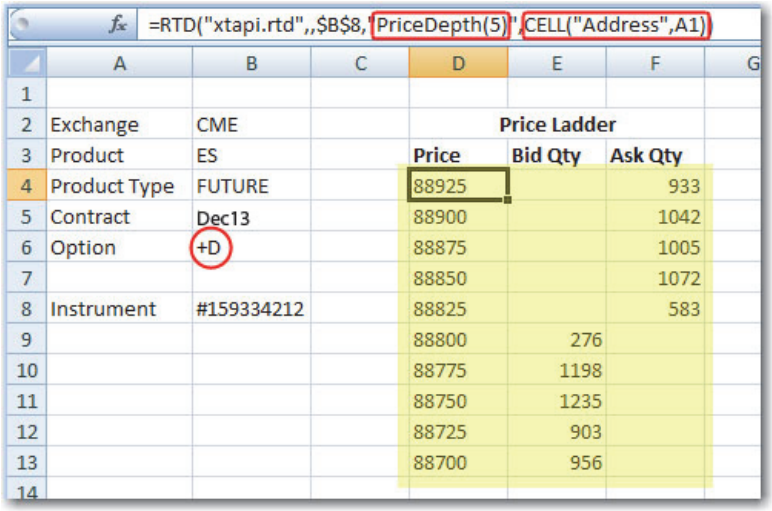
	A	B	C
1	90600		20
2	90575		20
3	90550		30
4	90525		22
5	90500		23
6	90450	13	
7	90425	17	
8	90400	96	
9	90375	5	
10	90350		

By specifying the **CELL("Address",A1)** parameter in the formula, you instruct Excel to use cell **A1** of the hidden spreadsheet as the starting location for the data. When it receives the data, Excel populates the hidden cells just like pasting an array of values into a regular spreadsheet, where each value occupies its own cell. You can think of the returned data as a three-column, ten-row spreadsheet and can reference the different values just like you would in a regular spreadsheet. For example, if you specify **CELL("Address",C2)**, Excel returns the value in the third column of the second row (in this case, the number of Asks four ticks away from the inside market, or 90575).

Requesting market depth for an instrument

The following example shows how to request five levels of market depth for ES Sep09 contract from the TT CME Gateway. Because the XTAPI RTD Server retrieves only inside market data by default, you must use the **+D** option to enable market depth access. The example uses the **PriceDepth** attribute to request five levels of depth.

Trading Technologies International, Inc. Proprietary and Confidential Information



The **BidQty** and **AskQty** columns both contain empty cells because the Ask side of the market contains no bid offers and the Bid side of the market contains no ask offers.

Note: If you do not specify the number of levels, the XTAPI RTD Server returns five levels of depth on each side of the market.

Identifying attributes that return array data

You can identify the attributes that return arrays of data by checking their data type information. For example, by looking at the **Data Type** column for the instrument attributes in Appendix A, Instrument Attributes, you can see that the following attributes return array values:

- AskDepth
- AskDepth90
- BidDepth
- BidDepth90
- IAskDepth
- IAskDepth90
- IBidDepth
- IBidDepth90
- IPriceDepth
- IPriceDepth90
- IXTDepth
- IXTDepth90
- PriceDepth
- PriceDepth90
- XTDepth
- XTDepth90

You can use the same approach to determine the fill, order, and order set attributes that return arrays of values. For lists of attributes, refer to the appropriate appendixes.

Trading Technologies International, Inc. Proprietary and Confidential Information

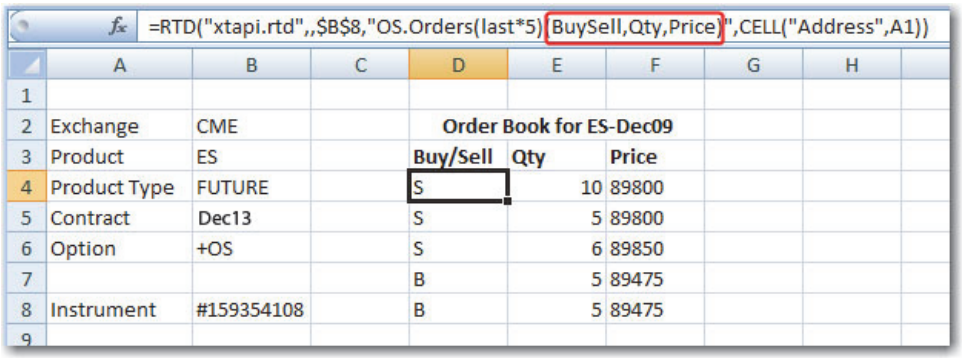
Requesting multiple attributes in a single equation

About arrays with multiple values

Some attributes, such as **OS.Orders** and **OS.Fills** for instruments, return variable-sized arrays. You can specify the number to retrieve (rows) and any number of corresponding attributes (columns). For example, you could request three attributes for 10 orders, eight attributes for 15 fills, and so on.

Requesting an order book update for an instrument

The following example requests updates for three attributes (**BuySell**, **Qty**, and **Price**) for the last five orders submitted for the ES Dec13 contract trading on TT CME Gateway. The **OS.Orders** attribute returns an array of values retrieved from an order object. You can specify any number of order attributes. In this example, the XTAPI RTD Server returns a 5x3 array.



Note: To access information from working orders, you must use the **+OS** option (cell **B6**) when generating the instrument ID.

When using the **OS.Orders** attribute, you can use the following expressions to specify how many orders to request:

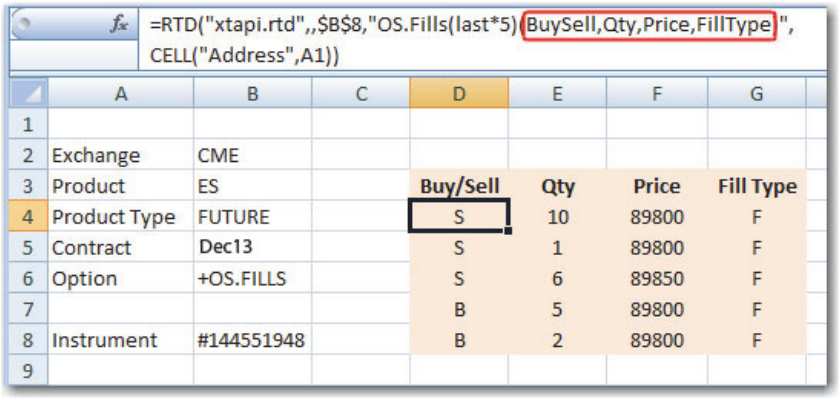
- **OS.Orders(*n*)** to return the first *n* orders
- **OS.Orders(first*n*)** to return the first *n* orders
- **OS.Orders(last*n*)** to return the last *n* orders
- **OS.Orders(*n*:*m*)** to return orders *n* through *m*, inclusive

For a complete list of all order attributes you can request through the **OS.Orders** attribute, refer to Appendix C, Order Attributes.

Requesting fill updates for an instrument

The following example requests updates for four attributes (**BuySell**, **Qty**, **Price**, and **FillType**) for the last five fills received for the ES Dec13 contract trading on TT CME Gateway. The **OS.Fills** attribute returns an array of values retrieved from an order object. You can specify any number of order attributes. In this example, the XTAPI RTD Server returns a 5x4 array.

Trading Technologies International, Inc. Proprietary and Confidential Information



	A	B	C	D	E	F	G
1							
2	Exchange	CME					
3	Product	ES		Buy/Sell	Qty	Price	Fill Type
4	Product Type	FUTURE		S	10	89800	F
5	Contract	Dec13		S	1	89800	F
6	Option	+OS.FILLS		S	6	89850	F
7				B	5	89800	F
8	Instrument	#144551948		B	2	89800	F
9							

Note: To access information from fills, you must use the **+OS.FILLS** option (cell **B6**) when generating the instrument ID.

When using the **OS.Fills** attribute, you can use the following expressions to specify how many fills to request:

- **OS.Fills(*n*)** to return the first *n* fills
- **OS.Fills(first*n*)** to return the first *n* fills
- **OS.Fills(last*n*)** to return the last *n* fills
- **OS.Fills(*n*:*m*)** to return fills *n* through *m*, inclusive

For a complete list of fill attributes you can request through the **OS.Fills** attribute, refer to Appendix D, Fill Attributes.

Recommendations when specifying array sizes

When you specify the number of rows for the array in the formula, TT recommends that you specify a number slightly larger than the number of rows you expect to return. If you do not specify a value for the number of array rows, Excel retrieves the data and updates only enough rows to contain the data it received. However, when you specify a number of rows, Excel inserts empty strings into the rows not filled by the data. Allowing Excel to determine the number of rows to update dynamically can leave expired orders still displayed in the spreadsheet.

In the previous example, suppose you specified **Orders()** instead of **Orders(5)** in the formula. The first update fills the spreadsheet as shown, with five rows of data. Suppose on the next update that the order in row four had been filled, so the RTD server sent updates only for the first three orders. Because you did not specify a number of rows to fill, Excel updates the array with the only the new data. As a result, the fourth and fifth rows would still contain the data for orders that are no longer valid, and you would incorrectly think that the orders are still valid. If you had specified a number of rows, five in this example, Excel would have updated the first three rows with the new data, and then filled the remaining two rows with empty strings. The filled orders would not still appear in the spreadsheet.

Trading Technologies International, Inc. Proprietary and Confidential Information

Creating a global order book

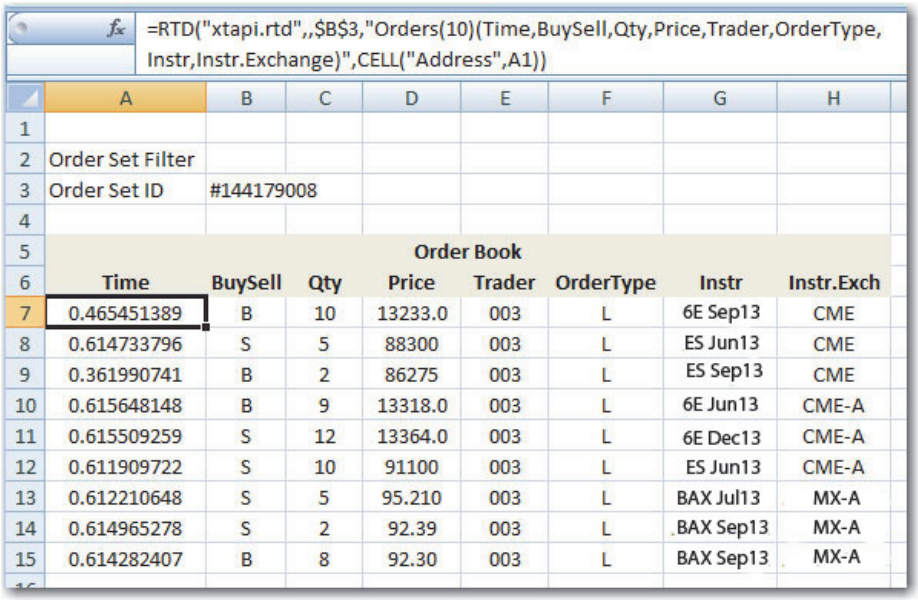
About order object attributes

The **Orders** attribute retrieves order information through an order set ID, in the same way that the **OS.Orders** attribute returns order information through an instrument ID. From an order set, the **Orders** attribute requests information from all visible orders, not just those orders associated with some instrument.

For a complete list of order attributes you can request, refer to Appendix C, Order Attributes.

Sample order book

The following example requests updates for several order attributes for the first ten working orders visible in the order set.



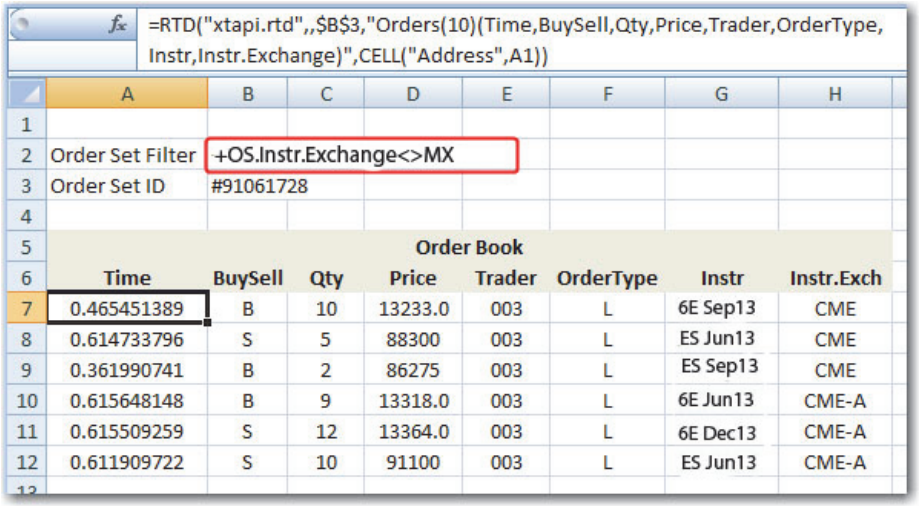
	A	B	C	D	E	F	G	H
1								
2	Order Set Filter							
3	Order Set ID	#144179008						
4								
5				Order Book				
6	Time	BuySell	Qty	Price	Trader	OrderType	Instr	Instr.Exch
7	0.465451389	B	10	13233.0	003	L	6E Sep13	CME
8	0.614733796	S	5	88300	003	L	ES Jun13	CME
9	0.361990741	B	2	86275	003	L	ES Sep13	CME
10	0.615648148	B	9	13318.0	003	L	6E Jun13	CME-A
11	0.615509259	S	12	13364.0	003	L	6E Dec13	CME-A
12	0.611909722	S	10	91100	003	L	ES Jun13	CME-A
13	0.612210648	S	5	95.210	003	L	BAX Jul13	MX-A
14	0.614965278	S	2	92.39	003	L	BAX Sep13	MX-A
15	0.614282407	B	8	92.30	003	L	BAX Sep13	MX-A

Filtering the order book by TT Gateway

By default, the XTAPI RTD Server requests information from all orders visible in the order set. You might want to limit the results to some particular subset of visible orders. If you specify an order set filter when generating the order set ID, the XTAPI RTD Server returns data only for those orders which match the filter criteria.

The following example uses a filter option (cell **B2**) to limit the query to orders not trading on the TT MX Gateway. It requests the last ten orders. However, as the order set does not contain ten orders that match the filter criteria, the XTAPI RTD Server returns the six orders that match and fills the remaining four rows with empty strings.

Trading Technologies International, Inc. Proprietary and Confidential Information



=RTD("xtapi.rtd",,B3,"Orders(10)(Time,BuySell,Qty,Price,Trader,OrderType, Instr,Instr.Exchange)",CELL("Address",A1))

	A	B	C	D	E	F	G	H
1								
2	Order Set Filter	+OS.Instr.Exchange<>MX						
3	Order Set ID	#91061728						
4								
5				Order Book				
6	Time	BuySell	Qty	Price	Trader	OrderType	Instr	Instr.Exch
7	0.465451389	B	10	13233.0	003	L	6E Sep13	CME
8	0.614733796	S	5	88300	003	L	ES Jun13	CME
9	0.361990741	B	2	86275	003	L	ES Sep13	CME
10	0.615648148	B	9	13318.0	003	L	6E Jun13	CME-A
11	0.615509259	S	12	13364.0	003	L	6E Dec13	CME-A
12	0.611909722	S	10	91100	003	L	ES Jun13	CME-A

Trading Technologies International, Inc. Proprietary and Confidential Information

Creating a global fill book

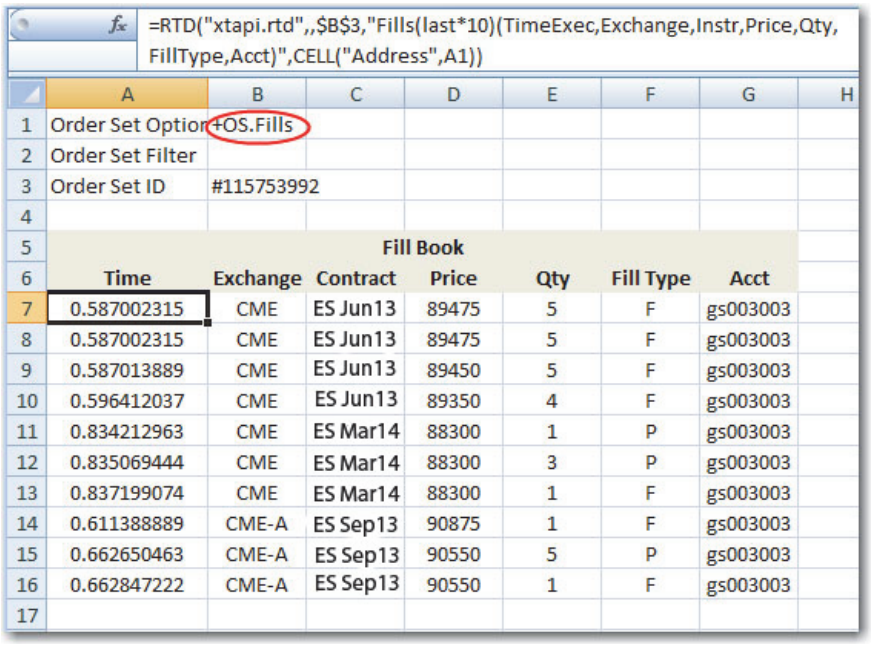
About fill object attributes

The **Fills** attribute retrieves fill information through an order set ID, in the same way that the **OS.Fills** attribute returns fill information through an instrument ID. From an order set, the **Fills** attribute requests information from all fills in the order set, not just those fills associated with a particular instrument.

For a complete list of fill attributes you can request, refer to Appendix D, Fill Attributes.

Sample fill book

The following example requests information about the last ten fills. To use the **Fills** attribute, you must include the **+OS.Fills** option (cell **B2**) when generating the order set ID.



	A	B	C	D	E	F	G	H
1	Order Set Option	+OS.Fills						
2	Order Set Filter							
3	Order Set ID	#115753992						
4								
5				Fill Book				
6	Time	Exchange	Contract	Price	Qty	Fill Type	Acct	
7	0.587002315	CME	ES Jun13	89475	5	F	gs003003	
8	0.587002315	CME	ES Jun13	89475	5	F	gs003003	
9	0.587013889	CME	ES Jun13	89450	5	F	gs003003	
10	0.596412037	CME	ES Jun13	89350	4	F	gs003003	
11	0.834212963	CME	ES Mar14	88300	1	P	gs003003	
12	0.835069444	CME	ES Mar14	88300	3	P	gs003003	
13	0.837199074	CME	ES Mar14	88300	1	F	gs003003	
14	0.611388889	CME-A	ES Sep13	90875	1	F	gs003003	
15	0.662650463	CME-A	ES Sep13	90550	5	P	gs003003	
16	0.662847222	CME-A	ES Sep13	90550	1	F	gs003003	
17								

Trading Technologies International, Inc. Proprietary and Confidential Information

A Instrument Attributes

TTInstrObj object attributes

About the TTInstrObj object

The XTAPI stores information about instruments in the **`TTInstrObj`** object. When the XTAPI RTD Server requests information about an instrument, the XTAPI looks for the attributes defined for the object. You can request values for any of the **`TTInstrObj`** object attributes.

Available attributes

You can access information about an instrument with the following XTAPI **`TTInstrObj`** attributes.

Attribute	Description	DataType
Ask (~Ask) Ask$ (~Ask$) Ask# (~Ask#) Ask& (~Ask&)	Best ask price, in X_TRADER display format. You can also use one of the following modifiers to display the value in a specific format: **$** : X_TRADER display format **#** : Decimal value **&** : Ticks The tilde (~) prefix returns the last change (delta) in the attribute. **Note:** When the `MergeImpliedsIntoDirects` property is enabled, the Ask price contains the better of the direct and implied prices.	As specified
AskDepth(*levels*) AskDepth#(*levels*) AskDepth&(*levels*)	Two-dimensional array of prices and the quantity associated with each (each row is price, quantity) for the specified number of depth levels.	Array of specified data types
AskDepth90(*levels*) AskDepth90#(*levels*) AskDepth90&(*levels*)	Column-based, two-dimensional array of prices and corresponding quantities for the specified number of depth levels, where each column contains a price and a quantity. You can also use one of the following modifiers to display the value in a specific format: **#** : Decimal value **&** : Integer (Ticks)	Array of specified data types
AskMktQty (~Ask) AskMktQty$ (~Ask$) AskMktQty# (~Ask#) AskMktQty& (~Ask&)	Market order quantity during the pre-open phase, as an integer. You can also use one of the following modifiers to display the value in a specific format: **$** : String **#** : Decimal value **&** : Integer The tilde (~) prefix returns the last change (delta) in the attribute.	As specified
AskOrderId	Order ID for the best ask. (Currently supported only for BrokerTec)	String

Table 1. TTInstrObj Get property attributes

Trading Technologies International, Inc. Proprietary and Confidential Information

Attribute	Description	DataType
AskQty (~AskQty) AskQty$ (~AskQty$) AskQty# (~AskQty#) AskQty& (~AskQty&)	Best ask quantity, as an integer. You can also use one of the following modifiers to display the value in a specific format: **$** : String **#** : Decimal value **&** : Integer The tilde (~) prefix returns the last change (delta) in the attribute. **Note:** When the `MergeImpliedsIntoDirects` property is enabled, **AskQty** contains the quantity for the better of the direct and implied Ask prices.	As specified
AvgOpenPriceInDecimal	Average price of open position in decimal format. Requires the **+OS.FILLS** option in the **InstrumentOptions** parameter for the RTD formula.	Decimal
AvgOpenPriceInTicks	Average price of open position in ticks and fractions of a tick. Requires the **+OS.FILLS** option in the **InstrumentOptions** parameter for the RTD formula.	Decimal
AvgOpenPriceRnd AvgOpenPriceRnd# AvgOpenPriceRnd$ AvgOpenPriceRnd&	Average price of open position rounded to the nearest tick. You can also use one of the following modifiers to display the value in a specific format: **$** : X_TRADER display format **#** : Decimal value **&** : Ticks Requires the **+OS.FILLS** option in the **InstrumentOptions** parameter for the RTD formula.	As specified
Bid (~Bid) Bid$ (~Bid) Bid# (~Bid#) Bid& (~Bid&)	Best bid price, in X_TRADER display format. You can also use one of the following modifiers to display the value in a specific format: **$** : X_TRADER display format **#** : Decimal value **&** : Ticks The tilde (~) prefix returns the last change (delta) in the attribute. **Note:** When the `MergeImpliedsIntoDirects` property is enabled, the Bid price contains the better of the direct and implied prices.	As specified
BidAskMid (~BidAskMid) BidAskMid$ (~BidAskMid$) BidAskMid# (~BidAskMid#) BidAskMid& (~BidAskMid&)	Average of the best bid price and the best ask price (rounded down to the nearest tick), in X_TRADER display format. You can also use one of the following modifiers to display the value in a specific format: **$** : X_TRADER display format **#** : Decimal value **&** : Ticks The tilde (~) prefix returns the last change (delta) in the attribute.	As specified

Table 1. TTInstrObj Get property attributes (Continued)

Trading Technologies International, Inc. Proprietary and Confidential Information

Attribute	Description	DataType
BidDepth(*levels*) BidDepth#(*levels*) BidDepth&(*levels*)	Row-based, two-dimensional array of prices and corresponding quantities for the specified number of depth levels, where each row contains a price and a quantity. You can also use one of the following modifiers to display the value in a specific format: **#** : Decimal value **&** : Integer (Ticks)	As specified
BidDepth90(*levels*) BidDepth90#(*levels*) BidDepth90&(*levels*)	Column-based, two-dimensional array of prices and corresponding quantities for the specified number of depth levels, where each column contains a price and a quantity. You can also use one of the following modifiers to display the value in a specific format: **#** : Decimal value **&** : Integer (Ticks)	As specified
BidMktQty (~Ask) BidMktQty$ (~Ask$) BidMktQty# (~Ask#) BidMktQty& (~Ask&)	Market order quantity during the pre-open phase, as an integer. You can also use one of the following modifiers to display the value in a specific format: **$** : String **#** : Decimal value **&** : Integer The tilde (~) prefix returns the last change (delta) in the attribute.	As specified
BidOrderId	Order ID for the best bid. (Currently supported only for BrokerTec)	String
BidQty (~BidQty) BidQty$ (~BidQty$) BidQty# (~BidQty#) BidQty& (~BidQty&)	Best bid quantity as an integer. You can also use one of the following modifiers to display the value in a specific format: **$** : String **#** : Decimal value **&** : Integer The tilde (~) prefix returns the last change (delta) in the attribute. **Note:** When the `MergeImpliedsIntoDirects` property is enabled, **BidQty** contains the quantity for the better of the direct and implied Bid prices.	As specified
BuyWrk	Quantity of working buy orders. Valid only when you attach the instrument to a **TTOrderSet** object.	Integer
CallPut	Whether the option on the instrument is a call or put: **C**: Call **P**: Put	String
Change (~Change) Change$ (~Change$) Change# (~Change#) Change& (~Change&)	Net price change in X_TRADER display format. You can also use one of the following modifiers to display the value in a specific format: **$** : X_TRADER display format **#** : Decimal value **&** : Ticks The tilde (~) prefix returns the last change (delta) in the attribute.	As specified

Table 1. TTInstrObj Get property attributes (Continued)

Trading Technologies International, Inc. Proprietary and Confidential Information

Attribute	Description	DataType
Close (~Close) Close$ (~Close$) Close# (~Close#) Close& (~Close&)	Prior session's closing price for the instrument in X_TRADER display format. You can also use one of the following modifiers to display the value in a specific format: **$** : X_TRADER display format **#** : Decimal value **&** : Ticks The tilde (~) prefix returns the last change (delta) in the attribute.	As specified
Contract	Default name of the instrument as displayed in the X_TRADER® **Market Grid Contract** field (example: ES Sep13).	String
ContractAlias	Alias name associated with the instrument (as assigned in Guardian), along with the contract's expiry information. If the instrument has no alias, the property returns the default instrument name and expiry information.	String
ContractShort	Short name associated with the instrument (example: ES M5)	String
Currency	ISO currency code associated with the instrument. ISO symbol for the source and target currencies.	String
Decimals	Number of implied decimals used in prices.	Integer
DeliveryUnit	Exchange-provided delivery unit for instruments in flow contracts; 1 for all others.	Integer
Exchange	Name of the exchange on which the instrument trades.	String
ExpirationDate	Instrument expiration date in the form, *MM/DD/YY*. For contracts that have no expiration date, this attribute returns NULL.	COleDateTime
Expiry	Instrument expiration, in month and year (example: Sep13).	String
FillsDownloaded	Whether all fills have been downloaded from the exchange.	Boolean
Fractional Fractional$	Ticking scheme for the instrument. Possible values include: **D**: Decimal **F**: Fractional	String
High (~High) High$ (~High$) High# (~High#) High& (~High&)	High price during the current trading session, in X_TRADER display format. You can also use one of the following modifiers to display the value in a specific format: **$** : X_TRADER display format **#** : Decimal value **&** : Ticks The tilde (~) prefix returns the last change (delta) in the attribute.	As specified
IAsk (~IAsk) IAsk$ (~IAsk$) IAsk# (~IAsk#) IAsk& (~IAsk&)	Implied ask price for the instrument, in X_TRADER display format. You can also use one of the following modifiers to display the value in a specific format: **$** : X_TRADER display format **#** : Decimal value **&** : Ticks The tilde (~) prefix returns the last change (delta) in the attribute.	As specified

Table 1. TTInstrObj Get property attributes (Continued)

Trading Technologies International, Inc. Proprietary and Confidential Information

Attribute	Description	DataType
IAskDepth(*levels*) IAskDepth#(*levels*) IAskDepth&(*levels*)	Row-based, two-dimensional array of implied ask prices and corresponding quantities for the specified number of depth levels, where each row contains a price and a quantity, in X_TRADER display format. You can also use one of the following modifiers to display the value in a specific format: **$** : X_TRADER display format **#** : Decimal value **&** : Ticks	Array of specified data types
IAskDepth90(*levels*) IAskDepth90#(*levels*) IAskDepth90&(*levels*)	Column-based, two-dimensional array of implied ask prices and corresponding quantities for the specified number of depth levels, where each column contains a price and a quantity, in X_TRADER display format. You can also use one of the following modifiers to display the value in a specific format: **$** : X_TRADER display format **#** : Decimal value **&** : Ticks	Array of specified data types
IAskQty (~IAskQty) IAskQty$ (~IAskQty$) IAskQty# (~IAskQty#) IAskQty& (~IAskQty&)	Implied ask quantity, as a integer. You can also use one of the following modifiers to display the value in a specific format: **$** : String **#** : Decimal value **&** : Integer The tilde (~) prefix returns the last change (delta) in the attribute.	As specified
IBid (~IBid) IBid$ (~IBid$) IBid# (~IBid#) IBid& (~IBid&)	Implied bid price, in X_TRADER display format. You can also use one of the following modifiers to display the value in a specific format: **$** : X_TRADER display format **#** : Decimal value **&** : Ticks The tilde (~) prefix returns the last change (delta) in the attribute.	As specified
IBidDepth(*levels*) IBidDepth#(*levels*) IBidDepth&(*levels*)	Row-based, two-dimensional array of implied prices and corresponding quantities for the specified number of depth levels, where each row contains a price and a quantity. You can also use one of the following modifiers to display the value in a specific format: **#** : Decimal value **&** : Integer (Ticks)	Array of specified data types
IBidDepth90(*levels*) IBidDepth90#(*levels*) IBidDepth90&(*levels*)	Column-based, two-dimensional array of implied prices and corresponding quantities for the specified number of depth levels, where each column contains a price and a quantity. You can also use one of the following modifiers to display the value in a specific format: **#** : Decimal value **&** : Integer (Ticks)	Array of specified data types

Table 1. TTInstrObj Get property attributes (Continued)

Trading Technologies International, Inc. Proprietary and Confidential Information

Attribute	Description	DataType
IBidQty (~IBidQty) IBidQty$ (~IBidQty$) IBidQty# (~IBidQty#) IBidQty& (~IBidQty&)	Implied bid quantity, as an integer. You can also use one of the following modifiers to display the value in a specific format: **$** : String **#** : Decimal value **&** : Integer The tilde (~) prefix returns the last change (delta) in the attribute.	As specified
IndAsk (~IndAsk) IndAsk$ (~IndAsk$) IndAsk# (~IndAsk#) IndAsk& (~IndAsk&)	Indicative ask price, in X_TRADER display format. You can also use one of the following modifiers to display the value in a specific format: **$** : X_TRADER display format **#** : Decimal value **&** : Ticks The tilde (~) prefix returns the last change (delta) in the attribute.	As specified
IndAskQty (~IndAskQty) IndAskQty$ (~IndAskQty$) IndAskQty# (~IndAskQty#) IndAskQty& (~IndAskQty&)	Indicative ask quantity, as an integer. You can also use one of the following modifiers to display the value in a specific format: **$** : String **#** : Decimal value **&** : Integer The tilde (~) prefix returns the last change (delta) in the attribute.	As specified
IndBid (~IndBid) IndBid$ (~IndBid$) IndBid# (~IndBid#) IndBid& (~IndBid&)	Indicative bid price, in X_TRADER display format. You can also use one of the following modifiers to display the value in a specific format: **$** : X_TRADER display format **#** : Decimal value **&** : Ticks The tilde (~) prefix returns the last change (delta) in the attribute.	As specified
IndBidQty (~IndBidQty) IndBidQty$ (~IndBidQty$) IndBidQty# (~IndBidQty#) IndBidQty& (~IndBidQty&)	Indicative bid quantity, as an integer. You can also use one of the following modifiers to display the value in a specific format: **$** : String **#** : Decimal value **&** : Integer The tilde (~) prefix returns the last change (delta) in the attribute.	As specified
IndClosePrice (~IndClosePrice) IndClosePrice$ (~IndClosePrice)$) IndClosePrice# (~IndClosePrice)#) IndClosePrice& (~IndClosePrice)&)	Indicative closing price indicating the settlement price for the session, as an integer. You can also use one of the following modifiers to display the value in a specific format: **$** : String **#** : Decimal value **&** : Integer The tilde (~) prefix returns the last change (delta) in the attribute.	As specified

Table 1. TTInstrObj Get property attributes (Continued)

Trading Technologies International, Inc. Proprietary and Confidential Information

Attribute	Description	DataType
IndCloseQty(~IndCloseQty) IndCloseQty$ (~IndCloseQty$) IndCloseQty# (~IndCloseQty#) IndCloseQty& (~IndCloseQty&)	Indicative close quantity, as an integer. You can also use one of the following modifiers to display the value in a specific format: **$** : String **#** : Decimal value **&** : Integer The tilde (~) prefix returns the last change (delta) in the attribute.	As specified
IndOpenPrice (~IndOpenPrice) IndOpenPrice$ (~IndOpenPrice$) IndOpenPrice# (~IndOpenPrice#) IndOpenPrice& (~IndOpenPrice&)	Indicative opening price indicating the settlement price for the session, as an integer. You can also use one of the following modifiers to display the value in a specific format: **$** : String **#** : Decimal value **&** : Integer The tilde (~) prefix returns the last change (delta) in the attribute.	As specified
IndOpenQty(~IndOpenQty) IndOpenQty$ (~IndOpenQty$) IndOpenQty# (~IndOpenQty#) IndOpenQty& (~IndOpenQty&)	Indicative open quantity, as an integer. You can also use one of the following modifiers to display the value in a specific format: **$** : String **#** : Decimal value **&** : Integer The tilde (~) prefix returns the last change (delta) in the attribute.	As specified
IndPrice (~IndPrice) IndPrice$ (~IndPrice$) IndPrice# (~IndPrice#) IndPrice& (~IndPrice&)	Indicative, or equalibrium, pre-open phase price indicating the price a market would match if it were open, as an integer. You can also use one of the following modifiers to display the value in a specific format: **$** : String **#** : Decimal value **&** : Integer The tilde (~) prefix returns the last change (delta) in the attribute.	As specified
IndSettle (~IndSettle) IndSettle$ (~IndSettle$) IndSettle# (~IndSettle#) IndSettle& (~IndSettle&)	Indicative closing price indicating the intra-day settlement price, as an integer. You can also use one of the following modifiers to display the value in a specific format: **$** : String **#** : Decimal value **&** : Integer The tilde (~) prefix returns the last change (delta) in the attribute.	As specified
Instr	Default instrument name as displayed in the X_TRADER® **Market Grid Contract** field.	String

Table 1. TTInstrObj Get property attributes (Continued)

Trading Technologies International, Inc. Proprietary and Confidential Information

Attribute	Description	DataType
InstrStatus	Status of the connection for underlying series. Possible values include: Connected Found Nonexistent Valid Undefined	String
IPriceDepth(*levels*) IPriceDepth#(*levels*) IPriceDepth&(*levels*)	Row-based, two-dimensional array of implied prices and corresponding quantities for the specified number of depth levels, where each row contains a price, bid quantity, and ask quantity, in X_TRADER display format. You can also use one of the following modifiers to display the value in a specific format: **$** : X_TRADER display format **#** : Decimal value **&** : Ticks	As specified
IPriceDepth90(*levels*) IPriceDepth90#(*levels*) IPriceDepth90&(*levels*)	Column-based, two-dimensional array of implied prices and corresponding quantities for the specified number of depth levels, where each column contains a price, bid quantity, and ask quantity, in X_TRADER display format. You can also use one of the following modifiers to display the value in a specific format: **$** : X_TRADER display format **#** : Decimal value **&** : Ticks	As specified
IsAskOwner	Whether an order belonging to the current trader is the best ask. (Currently supported only for BrokerTec)	Boolean
IsBidOwner	Whether an order belonging to the current trader is the best bid. (Currently supported only for BrokerTec)	Boolean
IsFlowContract	Whether the instrument is an energy product and can trade with quantities in flow.	Boolean
IsHit	Whether the aggressor in the workup was a seller. Valid only for BrokerTec.	Boolean
IsLTQAccum	Whether the GAL setting for LTQ accumulation is enabled.	Boolean
IsOTC	Whether the trade was an OTC trade.	Boolean
IsPrivateWorkup	Whether the market is in a private workup state. (Currently supported only for BrokerTec)	Boolean
IsPublicWorkup	Whether the market is in a public workup state. (Currently supported only for BrokerTec)	Boolean
IsServerDown	Whether the price server associated with the instrument is down.	Boolean
IsServerUp	Whether the price serve associated with the instrument is up.	Boolean
IsSubscribed	Whether the instrument receives live price updates.	Boolean
IsTake	Whether the aggressor in the workup was a buyer. Valid only for BrokerTec.	Boolean

Table 1. TTInstrObj Get property attributes (Continued)

Trading Technologies International, Inc. Proprietary and Confidential Information

Attribute	Description	DataType
IsTradedInFlow	Whether the contract is a flow contracted delivering quantities in flow. (Valid only for energy products)	Boolean
IsWorkup	Whether the market is in a public or private workup state. (Currently supported only for BrokerTec)	Boolean
IXTDepth(*levels*) IXTDepth#(*levels*) IXTDepth&(*levels*)	Row-based, two-dimensional array of implied prices and corresponding quantities for the specified number of depth levels, where each row contains a bid price, bid quantity, ask price, and ask quantity, in X_TRADER display format. You can also use one of the following modifiers to display the value in a specific format: **$** : X_TRADER display format **#** : Decimal value **&** : Ticks	Array of specified data types
IXTDepth90(*levels*) IXTDepth90#(*levels*) IXTDepth90&(*levels*)	Column-based, two-dimensional array of implied prices and corresponding quantities for the specified number of depth levels, where each column contains a bid price, bid quantity, ask price, and ask quantity. You can also use one of the following modifiers to display the value in a specific format: **#** : Decimal value **&** : Integer (Ticks)	Array of specified data types
Key	Unique, internal ID that TT associates with the instrument. Same as **Series** and **SeriesKey** attributes.	String
Last (~Last) Last$ (~Last$) Last# (~Last#) Last& (~Last&)	Price at which the instrument last traded, in X_TRADER display format. You can also use one of the following modifiers to display the value in a specific format: **$** : X_TRADER display format **#** : Decimal value **&** : Ticks The tilde (~) prefix returns the last change (delta) in the attribute.	As specified
LastQty (~LastQty) LastQty$ (~LastQty$) LastQty# (~LastQty#) LastQty& (~LastQty&)	Quantity of all trades at the last traded price, as in integer. You can also use one of the following modifiers to display the value in a specific format: **$** : String **#** : Decimal value **&** : Integer The tilde (~) prefix returns the last change (delta) in the attribute.	As specified
LLimit (~LLimit) LLimit$ (~LLimit$) LLimit# (~LLimit#) LLimit& (~LLimit&)	Lower limit used during price reasonability checks, as an integer. You can also use one of the following modifiers to display the value in a specific format: **$** : String **#** : Decimal value **&** : Integer The tilde (~) prefix returns the last change (delta) in the attribute.	As specified

Table 1. TTInstrObj Get property attributes (Continued)

Trading Technologies International, Inc. Proprietary and Confidential Information

Attribute	Description	DataType
Login.field	Information about the current user login. Possible field values include: CLRMEMBER DEFAULTACCT GATEWAYIP GROUP HANDOFFGROUP HANDOFFMEMBER HANDOFFTRADER MEMBER TRADER	String
LotSize	Exchange-provided lot size for instruments that are flow contracts; 1 for all others.	Integer
Low (~Low) Low$ (~Low$) Low# (~Low#) Low& (~Low&)	Lowest trading price for the current trading session, in X_TRADER display format. You can also use one of the following modifiers to display the value in a specific format: **$** : String **#** : Decimal value **&** : Integer The tilde (~) prefix returns the last change (delta) in the attribute.	As specified
LTPDirection	Direction of the most recently traded price for the instrument. Possible values include: **1** : Up **0** : Unchanged **-1** : Down If retrieved before price updates are received, the value returns TT_INVALID_QTY.	Integer
LTPDirection$	Direction of the most recently traded price for the instrument. Possible values include: '**+**' : Up ' ' : Unchanged '**-**' : Down If retrieved before price updates are received, the value returns an empty string.	String
Market	Name of the exchange associated with the TT Gateway; equivalent to the **Exchange** attribute.	String
MaxDecimals	Maximum number of significant digits to the right of the decimal point of a price.	Integer
MIAsk (~MIAsk) MIAsk$ (~MIAsk$) MIAsk# (~MIAsk#) MIAsk& (~MIAsk&)	Better of the direct best ask price and the implied ask price, in X_TRADER display format. You can also use one of the following modifiers to display the value in a specific format: **$** : String **#** : Decimal value **&** : Integer The tilde (~) prefix returns the last change (delta) in the attribute.	As specified

Table 1. TTInstrObj Get property attributes (Continued)

Trading Technologies International, Inc. Proprietary and Confidential Information

Attribute	Description	DataType
MIAskQty (~MIAskQty) MIAskQty$ (~MIAskQty$) MIAskQty# (~MIAskQty#) MIAskQty& (~MIAskQty&)	One of the following: Direct best ask quantity if the direct best ask price is better than the implied ask price Implied ask quantity if the implied ask price is better than the direct best ask price Direct and implied ask quantities added together if the direct and implied ask prices are the same. The attribute returns an integer value by default. You can also use one of the following modifiers to display the value in a specific format: **$** : String **#** : Decimal value **&** : Integer The tilde (~) prefix returns the last change (delta) in the attribute.	As specified
MIBid (~MIBid) MIBid$ (~MIBid$) MIBid# (~MIBid#) MIBid& (~MIBid&)	Better of the direct best bid price and the implied bid price, in X_TRADER display format. You can also use one of the following modifiers to display the value in a specific format: **$** : X_TRADER display format **#** : Decimal value **&** : Ticks The tilde (~) prefix returns the last change (delta) in the attribute.	As specified
MIBidQty (~MIBidQty) MIBidQty$ (~MIBidQty$) MIBidQty# (~MIBidQty#) MIBidQty& (~MIBidQty&)	One of the following: Direct best bid quantity if the direct best bid price is better than the implied bid price Implied bid quantity if the implied bid price is better than the direct best bid price Direct and implied bid quantities added together if the direct and implied bid prices are the same. The attribute returns an integer value by default. You can also use one of the following modifiers to display the value in a specific format: **$** : String **#** : Decimal value **&** : Integer The tilde (~) prefix returns the last change (delta) in the attribute.	As specified
MktImbalanceQty	Degree of imbalance between the number of bids and asks for the full depth of the Instrument. Positive values indicate more bids than asks; negative values indicate more asks that bids.	Integer
MMMYY	Contract name, using the 3-character month identifier and 2-digit year (example: Jun13). For products of type STOCK, this value returns an empty string.	String
Name	Default instrument name as displayed in the X_TRADER® **Market Grid Contract** field.	String

Table 1. TTInstrObj Get property attributes (Continued)

Trading Technologies International, Inc. Proprietary and Confidential Information

Attribute	Description	DataType
NetCnt	Value of the **NetCnt** attribute in the attached **`TTOrderSet`**, which is the total number of working orders (Number of Buy Orders + Number of Sell Orders). Valid only when you attach the instrument to a **TTOrderSet** object.	Integer
NetPos	Value of the **NetPos** attribute in the attached **`TTOrderSet`**, which is the current net position (Quantity Purchased - Quantity Sold), based on fill information. Valid only when you attach the instrument to a **TTOrderSet** object.	Integer
NetTicks	Value of the **NetTicks** attribute in the attached TTOrderSet, which is the net cost in ticks of the current position (Revenue from all Sells - Cost of all Buys). Valid only when you attach the instrument to a **TTOrderSet** object.	Integer
NetWrk	Value of the **NetWrk** attribute in the attached TTOrderSet, which is the total quantity of working orders (Quantity of Working Buys + Working Sells). Valid only when the instrument has an attached **`TTOrderSet`** object.	Integer
NumberOfAskOrders(*priceLevel)*	Number of resting ask orders at the specified priceLevel (0 for the inside market).	Integer
NumberOfBidOrders(*priceLevel)*	Number of resting bid orders at the specified priceLevel (0 for the inside market).	Integer
Open (~Open) Open# (~Open#) Open$ (~Open$) Open& (~Open&)	Opening price of the instrument for the current trading session, in X_TRADER display format. You can also use one of the following modifiers to display the value in a specific format: **$** : X_TRADER display format **#** : Decimal value **&** : Ticks	As specified
OpenPL OpenPL& OpenPL# OpenPL$	Current value of open positions in ticks, as an integer. You can also use one of the following modifiers to display the value in a specific format: **$** : X_TRADER display format **#** : Decimal value **&** : Ticks The **OpenPL$** value is expressed as USD currency. Requires the **+OS.FILLS** option in the **InstrumentOptions** parameter for the RTD formula.	As specified
OpenPL~ OpenPL^ OpenPL.^ OpenPL@	Current market value of open positions as: **~** : String delta format **^** : Native currency **.^** : Native currency **@** : Primary currency Requires the **+OS.FILLS** option in the **InstrumentOptions** parameter for the RTD formula.	As specified

Table 1. TTInstrObj Get property attributes (Continued)

Trading Technologies International, Inc. Proprietary and Confidential Information

Attribute	Description	DataType
OpenPL.[Currency]	Current market value of open positions in the currency whose code you specify (Currency). Possible values include: • ARS • CHF • HKD • PTE • ATS • DEM • INR • RUR • AUD • ESP • ITL • SGD • BEF • EUR • JPY • USD • BRL • FRF • KRW • XBA • CAD • GBP • NLG • XEU Requires the **+OS.FILLS** option in the **InstrumentOptions** parameter for the RTD formula.	String
OS.objAttribute	Value of the specified object attribute in TTOrderSet. Valid only when the instrument has an attached **`TTOrderSet`**.	As specified
OTCPrc (~OTCPrc) OTCPrc$ (~OTCPrc$) OTCPrc# (~OTCPrc#) OTCPrc& (~OTCPrc&)	Price of the last wholesale trade for the instrument in X_TRADER display format. You can also use one of the following modifiers to display the value in a specific format: **$** : X_TRADER display format **#** : Decimal value **&** : Ticks The tilde (~) prefix returns the last change (delta) in the attribute.	As specified
OTCQty (~OTCQty) OTCQty$ (~OTCQty) OTCQty# (~OTCQty) OTCQty& (~OTCQty&)	Quantity of the last wholesale trade, as an integer. You can also use one of the following modifiers to display the value in a specific format: **$** : String **#** : Decimal value **&** : Integer The tilde (~) prefix returns the last change (delta) in the attribute.	As specified
OwnAskOrderKey	Order key of the best ask if the order belongs to the current trader. Valid only when you attach the instrument to a **TTOrderSet** object. (Currently supported only for BrokerTec)	String
OwnBidOrderKey	Order key of the best bid if the order belongs to the current trader. Valid only when you attach the instrument to a **TTOrderSet** object. (Currently supported only for BrokerTec)	String
PctChange PctChange$	Percentage change from the previous close price to the current price, as a decimal value. You can use the $ modifier to return the value as a string.	As specified

Table 1. TTInstrObj Get property attributes (Continued)

Trading Technologies International, Inc. Proprietary and Confidential Information

Attribute	Description	DataType
PL PL$ PL# PL&	P & L (NetPos * LTP + NetTicks), as an integer. You can also use one of the following modifiers to display the value in a specific format: **$** : String **#** : Decimal value **&** : Integer The PL$ value is expressed as USD currency. Valid only when you attach the instrument to a **TTOrderSet** object.	As specified
PL~ PL^ PL.^ PL@	P & L in string delta format, in one of the following formats: **~** : String delta format **^** : Native currency **.^** : Native currency **@** : Primary currency Valid only when you attach the instrument to a **TTOrderSet** object.	As specified
PL.[Currency]	P & L in the currency whose code you specify (Currency). Possible values include: • ARS • CHF • HKD • PTE • ATS • DEM • INR • RUR • AUD • ESP • ITL • SGD • BEF • EUR • JPY • USD • BRL • FRF • KRW • XBA • CAD • GBP • NLG • XEU Valid only when you attach the instrument to a **TTOrderSet** object.	String
PL.[LOTS]	P & L for instruments that trade in flow in lots; for instruments that do not trade in flow, returns in contracts. Valid only when you attach the instrument to a **TTOrderSet** object.	Integer
PL.Z PL.Z$ PL.Z# PL.Z&	P & L since the last ClearQuotePosition call, as an integer. You can also use one of the following modifiers to display the value in a specific format: **$** : String **#** : Decimal value **&** : Integer Valid only when a quoting **`TTOrderSet`** is attached to the instrument.	As specified
PointValue	Value of one price point. ProductName Variable extracts this value from the TT Guardian product table for the instrument.	Double
PriceDepth(*levels*) PriceDepth#(*levels*) PriceDepth&(*levels*)	Row-based, two-dimensional array of prices and corresponding quantities for the specified number of depth levels, where each row contains a price, bid quantity, and ask quantity. The value returns in X_TRADER display format. You can also use one of the following modifiers to display the value in a specific format: **#** : Decimal value **&** : Integer (Ticks)	Array of specified data types

Table 1. TTInstrObj Get property attributes (Continued)

Trading Technologies International, Inc. Proprietary and Confidential Information

Attribute	Description	DataType
PriceDepth90(*levels*) PriceDepth90#(*levels*) PriceDepth90&(*levels*)	Column-based, two-dimensional array of prices and corresponding quantities for the specified number of depth levels, where each column contains a price, bid quantity, and ask quantity. The value returns in X_TRADER display format. You can also use one of the following modifiers to display the value in a specific format: **#** : Decimal value **&** : Integer (Ticks)	Array of specified data types
PriceState PriceState$	Current workup state and type as string or bitmask according to the following: 0x00 : "None" 0x01 : "Workup Private" 0x02 : "Workup Public" 0x0F : "Other States" 0x10 : "Aggressor is bid" 0x20 : "Aggressor is ask" 0x40 : "Workup correction" "Other States" serves as a mask for unused values. Common values for PriceState include: 0 : "No workup" 17 : "Private workup aggressor is bid" 18 : "Public workup aggressor is bid" 33 : "Private workup aggressor is ask" 34 : "Private workup aggressor is ask" (Currently supported only for BrokerTec)	As specified
PriceType	ProductName Variable price display type. For a description of possible values, refer to "Price types" on page 557.	Integer
ProdType ProdType&	Name of (**ProdType**) or integer (**ProdType&**) representing the instrument's product type. For a list of possible product types, see "Product types" on page 559.	As specified
Product	Name of the product.	String
ProductAlias	Name of a product alias (as it is assigned in Guardian). If the product does not have an alias, **ProductAlias** returns the default name of the product.	String
ProductTitle	Product name followed by the product type in parenthesis (for example, ES (FUTURE)).	String
RoundLotQty	Size of trading quantity multiplier. Some contracts must be traded in fixed mulitples of the lot quantity. For example, a contract might require you to trade it only in mulitples of 5.	Integer
SellWrk	Quantity of working sells. Valid only when you attach the instrument to a **TTOrderSet** object.	Integer
Series	Unique, internal ID that TT associates with the instrument. Same as **Key** and **SeriesKey** attributes.	String
SeriesKey	Unique, internal ID that TT associates with the instrument. Same as **Key** and **Series** attributes.	String
SeriesName	Exchange-provided name for the instrument, if it exits; "Contract" otherwise.	String

Table 1. TTInstrObj Get property attributes (Continued)

Trading Technologies International, Inc. Proprietary and Confidential Information

Attribute	Description	DataType
Settle (~Settle) Settle$ (~Settle$) Settle# (~Settle#) Settle& (~Settle&)	Instrument's settle price, in X_TRADER display format. You can also use one of the following modifiers to display the value in a specific format: **$** : X_TRADER display format **#** : Decimal value **&** : Ticks The tilde (~) prefix returns the last change (delta) in the attribute.	As specified
Status Status& Status$	Status of the instrument on the exchange, such as Trading, Pre-trade, etc., as an integer. You can use the following modifiers to return the value in a specific format: & : Integer $: String For more information, refer to Product status types, on page 639.	As specified
Strike	Strike price as an integer (not ticks). Valid only for options instruments.	Integer
StrikeDecimals	Number of significant digits to the right of the decimal used for the strike price. Valid only for options instruments.	Integer
StrikeFlags	Display format for the instrument's strike price. Possible values include: D: Decimal points (default) N: Fractional tick Valid only for options instruments.	As specified
StrikePrecision	Default market precision for the instrument's strike price.	Integer
TAsk (~TAsk) TAsk$ (~TAsk$) TAsk# (~TAsk#) TAsk& (~TAsk&)	Theoretical ask price for a mass quote, in X_TRADER display format. You can also use one of the following modifiers to display the value in a specific format: **$** : X_TRADER display format **#** : Decimal value **&** : Ticks The tilde (~) prefix returns the last change (delta) in the attribute.	As specified
TAskQty (~TAskQty) TAskQty$ (~TAskQty$) TAskQty# (~TAskQty#) TAskQty& (~TAskQty&)	Theoretical ask quantity, as an integer. You can also use one of the following modifiers to display the value in a specific format: **$** : String **#** : Decimal value **&** : Integer The tilde (~) prefix returns the last change (delta) in the attribute.	As specified

Table 1. TTInstrObj Get property attributes (Continued)

Trading Technologies International, Inc. Proprietary and Confidential Information

Attribute	Description	DataType
TBid (~TBid) TBid$ (~TBid$) TBid# (~TBid#) TBid& (~TBid&)	Theoretical bid price for a mass quote, in X_TRADER display format. You can also use one of the following modifiers to display the value in a specific format: **$** : X_TRADER display format **#** : Decimal value **&** : Ticks The tilde (~) prefix returns the last change (delta) in the attribute.	As specified
TBidQty (~TBidQty) TBidQty$ (~TBidQty$) TBidQty# (~TBidQty#) TBidQty& (~TBidQty&)	Theoretical ask quantity, as an integer. You can also use one of the following modifiers to display the value in a specific format: **$** : String **#** : Decimal value **&** : Integer The tilde (~) prefix returns the last change (delta) in the attribute.	As specified
Text	Same as the Name and Instr attributes.	String
TickIncrement	Smallest valid tick increment as an integer. The default value is 1. For reduced tick spreads, the value of underlying legs is either 2 or 4.	Integer
TPrc (~TPrc) TPrc$ (~TPrc$) TPrc# (~TPrc#) TPrc& (~TPrc&)	Theoretical mid price, as a decimal. You can also use one of the following modifiers to display the value in a specific format: **$** : X_TRADER display format **#** : Decimal value **&** : Ticks The tilde (~) prefix returns the last change (delta) in the attribute.	As specified
ULimit (~ULimit) ULimit$ (~ULimit$) ULimit# (~ULimit#) ULimit& (~ULimit&)	Upper limit used during price reasonability check, as an integer. You can also use one of the following modifiers to display the value in a specific format: **$** : String **#** : Decimal value **&** : Integer The tilde (~) prefix returns the last change (delta) in the attribute.	As specified
Version	Version of the option.	String
Vola (~Vola) Vola$ (~Vola$) Vola# (~Vola#) Vola& (~Vola&)	Theoretical volatility, as an integer. You can also use one of the following modifiers to display the value in a specific format: **$** : String **#** : Decimal value **&** : Integer The tilde (~) prefix returns the last change (delta) in the attribute.	As specified

Table 1. TTInstrObj Get property attributes (Continued)

Trading Technologies International, Inc. Proprietary and Confidential Information

Attribute	Description	DataType
Volume (~Volume) Volume$ (~Volume$) Volume# (~Volume#) Volume& (~Volume&)	Total quantity traded for the day, as a integer. You can also use one of the following modifiers to display the value in a specific format: **$** : String **#** : Decimal value **&** : Integer The tilde (~) prefix returns the last change (delta) in the attribute.	As specified
WorkupState	Current workup state as a string. Possible values include: "None" "Public" "Private" (Currently supported only for BrokerTec)	String
WorkupState&	Current workup state as an enumeration. Possible values include: WORKUP_TYPE_NONE WORKUP_TYPE_PRIVATE WORKUP_TYPE_PUBLIC (Currently supported only for BrokerTec)	Enum
WorkupType	Current workup type as a string. Possible values include: "None" "Hit" "Take" (Currently supported only for BrokerTec)	Enum
WorkupType&	Current workup type as an enumeration. Possible values include: WORKUP_TYPE_NONE WORKUP_TYPE_HIT WORKUP_TYPE_TAKE (Currently supported only for BrokerTec)	Enum
XTDepth(*levels*) XTDepth#(*levels*) XTDepth&(*levels*)	Row-based, two-dimensional array of prices and corresponding quantities for the specified number of depth levels, where each row contains a bid price, bid quantity, ask price, and ask quantity. Returns prices in X_TRADER display format. You can also use one of the following modifiers to display the value in a specific format: **#** : Decimal value **&** : Integer (Ticks)	Array of specified data types
XTDepth90(*levels*) XTDepth90#(*levels*) XTDepth90&(*levels*)	Column-based, two-dimensional array of prices and corresponding quantities for the specified number of depth levels, where each column contains a bid price, bid quantity, ask price, and ask quantity. Returns prices in X_TRADER display format. You can also use one of the following modifiers to display the value in a specific format: **#** : Decimal value **&** : Integer (Ticks)	Array of specified data types

Table 1. TTInstrObj Get property attributes (Continued)

Trading Technologies International, Inc. Proprietary and Confidential Information

B Order Set Attributes

TTOrderSet object attributes

About the TTOrderSet object

The XTAPI stores information about order sets in the `TTOrderSet` object. When the XTAPI RTD Server requests information about an order set, the XTAPI looks for the attributes defined for the object. You can request values for any of the `TTOrderSet` object attributes.

Available attributes

You can access information about an order set with the following XTAPI `TTOrderSet` attributes.

Attribute	Description	DataType
BuyCnt	Number of buy orders in the order set	Integer
BuyFillCnt	Number of fills for buy orders. Requires the **+OS.FILLS** option in the **InstrumentOptions** parameter for the RTD formula.	Integer
BuyFills	Row-based, two-dimensional array of values from all filled buy orders, where each row contains an fill and each column contains an attribute value.	Variant
BuyFills90	Column-based, two-dimensional array of values from all filled buy orders, where each column contains an fill and each row contains an attribute value.	Variant
BuyLimits	Whether the order set checks self-imposed limits (**MaxBuy** attributes) when submitting buy orders	Boolean
BuyOPos	Quantity of contracts purchased based on successfully executed buy orders	Integer
BuyOrders	Row-based, two-dimensional array of values from all buy orders, where each row contains an order and each column contains an attribute value.	Variant
BuyOrders90	Column-based, two-dimensional array of values from all buy orders, where each column contains an order and each row contains an attribute value.	Variant
BuyPos	Quantity purchased based on fill information	Integer
BuyTicks	Cost in ticks of all purchased contracts	Integer
BuyWrk	Quantity of working buys	Integer
Fills	Row-based, two-dimensional array of values from all fills, where each row contains an fill and each column contains an attribute value.	Variant
Fills90	Column-based, two-dimensional array of values from all fills, where each column contains an fill and each row contains an attribute value.	Variant
MaxBuyOrderQty	Maximum quantity allowed in a single buy order	Integer
MaxBuyOrders	Maximum number of buy orders allowed in the order set	Integer
MaxBuyWorking	Maximum number of working orders allowed in the order set	Integer
MaxFills	Maximum number of fill records that can be included in one fill block	Integer

Table 1. TTOrderSet Get attributes

Trading Technologies International, Inc. Proprietary and Confidential Information

Attribute	Description	DataType
MaxOrderQty	Maximum quantity allowed in a single buy or sell order	Integer
MaxOrders	Maximum number of orders allowed in the order set	Integer
MaxOrderStatusUpdates	Maximum number of consecutive order updates allowed before allowing other events to fire	Integer
MaxPosition	Maximum long or short position (in ticks) allowed in the order set	Integer
MaxRejects	Maximum number of order rejection events sent in succession	Integer
MaxSellOrderQty	Maximum quantity allowed in a single sell order	Integer
MaxSellOrders	Maximum number of sell orders allowed in the order set	Integer
MaxSellWorking	Maximum number of working sell orders allowed in the order set	Integer
MaxWorking	Maximum number of working orders allowed in the order set	Integer
NetCnt	Total number of working orders (Buys + Sells)	Integer
NetFillCnt	Total number of filled orders	Integer
NetLimits	Whether the order set checks self-imposed position limits when submitting an order	Boolean
NetLongPosMax	Maximum net long position since order set was opened (quantity purchased - quantity sold), based on fill information	Integer
NetOPos	Current net position based on orders	Integer
NetPos	Current net position based on fill information	Integer
NetShortPosMax	Maximum net short position since order set was opened (quantity sold - quantity purchased), based on fill information	Integer
NetTicks	Value of the **NetTicks** attribute, which is the net cost in ticks of the current position. Valid only when you attach the instrument to a **TTOrderSet** object.	Decimal
NetWrk	Total quantity of working orders (working buys + working sells)	Integer
Orders	Row-based, two-dimensional array of values from all orders, where each row contains an order and each column contains an attribute value.	Array
Orders90	Column-based, two-dimensional array of values from all orders, where each column contains an order and each row contains an attribute value.	Array
OrderStateCorrection	TT internal use only.	Boolean
RealizedPL	Current realized P&L in ticks	Integer
SellCnt	Number of sell orders in the order set	Integer
SellFillCnt	Number of fills for sell orders. Requires the **+OS.FILLS** option in the **InstrumentOptions** parameter for the RTD formula.	Integer
SellFills	Row-based, two-dimensional array of values from all filled sell orders, where each row contains an fill and each column contains an attribute value.	Variant
SellFills90	Column-based, two-dimensional array of values from all filled sell orders, where each column contains an fill and each row contains an attribute value.	Variant
SellLimits	Whether the order set checks self-imposed limits (**MaxSell*** attributes) when submitting sell orders	Integer

Table 1. TTOrderSet Get attributes

Trading Technologies International, Inc. Proprietary and Confidential Information

Attribute	Description	DataType
SellOPos	Quantity of contracts sold based on successfully executed sell orders	Integer
SellOrders	Row-based, two-dimensional array of values from all sell orders, where each row contains an order and each column contains an attribute value.	Array
SellOrders90	Column-based, two-dimensional array of values from all sell orders, where each column contains an order and each row contains an attribute value.	Array
SellPos	Quantity sold based on fill information	Integer
SellTicks	Cost in ticks of all sold contracts	Integer
SellWrk	Quantity of working sells	Integer
ThrowLimitErrors	How the XTAPI responds when a self-imposed limit is violated. Possible values include: **FALSE**, reduces order to self-imposed limit **TRUE**, throws an error	Integer

Table 1. TTOrderSet Get attributes

Trading Technologies International, Inc. Proprietary and Confidential Information

Trading Technologies International, Inc. Proprietary and Confidential Information

C Order Attributes

TTOrderObj object attributes

About the TTOrderObj object

The XTAPI stores information about orders in the **`TTOrderObj`** object. When the XTAPI RTD Server requests information about an order, the XTAPI looks for the attributes defined for the object. You can request values for any of the **`TTOrderObj`** object attributes.

Available attributes

You can access information about an order set with the following XTAPI **`TTOrderObj`** attributes.

Attribute	Description	DataType
Acct	Customer's or trader's account number	String
AcctType AccountType	Code indicating the desired account type	String
Broker	Broker's name (multi-broker only)	String
Broker#	Broker's ID (multi-broker only)	String
BuySell BuySell$	Type of order. Possible values include: **B**: Buy **S**: Sell Specifying BuySell$ returns Buy or Sell.	String
Change Chg	Whether the order changed	Boolean
ClrMember	Clearing Member ID	String
CntrGroup	Counter Side Clearing Group ID	String
CntrMember	Counter Side Clearing Member ID	String
CntrTrader	Counter Side Clearing Trader ID	String
ColorPri ColorSec	Color to display in the MD Trader® working orders grid, where: **ColorPri** represents the order's main color (left-side of the grid) **ColorSec** represents the secondary color (right-side of the grid) The value is an integer value representing the RGB value for the color. The value corresponds to a Microsoft `ColorRef` data type.	Integer
Company	Broker's company name (multi-broker only)	String
Company#	Broker's company ID (multi-broker only)	String
ConnectionName	Broker's connection name (multi-broker only)	String
Contract Contract$	Instrument's name with expiration (e.g. ES Sep13)	String
ContractAlias	Instrument's alias name with expiration (e.g. E-Mini Sep13)	String

Table 1. TTDropHandler Get attributes

Trading Technologies International, Inc. Proprietary and Confidential Information

Attribute	Description	DataType
Contract.getAttribute	Get attribute for the attached instrument (TTInstrObj)	Various
Date	Date of the order in *MM*/*DD*/*YY* (e.g. 05/18/12) The value contains valid data only in the date portion of the `COleDateTime` object.	COleDateTime
DisclosedQty DiscQty	Quantity of the order visible in the market. Valid only for disclosed quantity orders.	Integer
Ex:OrderNo	Order number for the specified exchange (e.g. CME: 107578464)	String
ExchCred	Exchange credentials (multi-broker only)	String
ExecutionType	Current state of the order. Possible values include: **-1**: Order is in an invalid state **0**: Order is in pre-execution state (not yet sent to a TT Gateway) **1**: Order is in post-execution state and the XTAPI has received a response from the TT Gateway	Integer
ExGroup	Exchange Group ID for this order	String
ExMember	Exchange Member ID for this order	String
ExOrderID	Exchange Order ID assigned to this order. This attribute corresponds to the **OrderNo** attribute for exchanges that do not return strictly numeric order numbers. **Note:** If the exchange does not provide a value, the XTAPI populates it with the OrderNo attribute value.	String
Expiry	Expiration date for the order. Possible values include: Null: GTC order Today's date: GIS or GTD order Future date: Expiration date The value contains valid data only in the date portion of the `COleDateTime` object.	COleDateTime
ExTrader	Exchange Trader ID for this order	String
ExTransNo	Unique identifier for the order transaction.	String
FFT1 FFT2 FFT3	Free-form text fields. **FFT1** typically contains the account number.	String
Fill FillQty	Order quantity already filled	Integer
GiveUp	Give-up member's clearing ID, when the fill is assigned to another member **Note:** The TT ICE Gateway does not support the **GiveUp** attribute; instead, it uses the **ClrMember** attribute for Give Up orders.	String
Group	Group name of the trader from the Member/Group/Trader ID	String
Instr Instr$	Default instrument name as displayed in the X_TRADER® **Market Grid Contract** field (e.g., ES Sep13)	String
InstrAlias	Instrument's alias name with expiration (e.g. E-Mini Sep13)	String

Table 1. TTDropHandler Get attributes

Trading Technologies International, Inc. Proprietary and Confidential Information

Attribute	Description	DataType
Instr.getAttribute	Value of the specified object attribute in the TTInstrObj object	Various
IsAON	Whether the order is an All-Or-None order	Boolean
IsAuction	Whether the order is only valid for auction	Boolean
IsAuto-Aggress	Whether the order is auto-aggressive (Currently supported only for BrokerTec)	Boolean
IsAutomated	Whether the order orginates from an automated program.	Boolean
IsAwayMarket	Whether an away market maker should match BBO of an away market	Boolean
IsBasis	Whether the order is a basis order	Boolean
IsBestLimit	Whether the order is a best limit order (Currently supported only for the TT TOCOM Gateway)	Boolean
IsBlock	Whether the order is a block order	Boolean
IsBuy	Whether the order is a buy order	Boolean
IsFAS	Whether the order is a fill and save order	Boolean
IsFlex	Whether the order is a flex option order	Boolean
IsFOK	Whether the order is a fill or kill order	Boolean
IsGIS	Whether the order is a good-in-session order	Boolean
IsGTC	Whether the order is a good-tll-canceled order	Boolean
IsGTD	Whether the order is a good-till-date order	Boolean
IsIceberg	Whether the order is an Iceberg order	Boolean
IsIfTouched	Whether the order requires additional action when a price is touched	Boolean
IsIOC	Whether the order is an immediate-or-cancel order	Boolean
IsLOC	Whether the order is a limit-on-close order to be filled or killed when market closes	Boolean
IsLOO	Whether the order is a limit-on-open order to be filled or killed when the market opens	Boolean
IsLSM	Whether the order is a limit-stop-market order	Boolean
IsLTM	Whether the order is a limit to market order to be converted to a market order on close	Boolean
IsMarketToLimit	Whether the order is a market-to-limit order	Boolean
IsMLM	Whether the order is a market-limit-market order	Boolean
IsMOC	Whether the order is a market-on-close order	Boolean
IsMOO	Whether the order is a market-on-open order	Boolean
IsMTL	Whether the order is a market-to-limit order	Boolean
IsMV	Whether the order is a minimum volume order	Boolean
IsNon-Leave	Whether the order is a non-leave order (Currently supported only for BrokerTec)	Boolean

Table 1. TTDropHandler Get attributes

Trading Technologies International, Inc. Proprietary and Confidential Information

Attribute	Description	DataType
IsOrderColorSupported	Whether the version of X_TRADER® installed on a trader's system supports receiving order color information from an XTAPI application. If color is supported, X_TRADER® uses the values in the **ColorPri** and **ColorSec** object attributes for the **`TTOrderProfile`** and **`TTOrderObj`** objects in the MD Trader® widow. You can define the colors by setting the attributes with the Set method.	Boolean
IsPartial	Whether the order is partially filled	Boolean
IsPassive	Whether the order is passive (Currently supported only for BrokerTec)	Boolean
IsSell	Whether the order is a sell order	Boolean
IsShortExempt	Whether the order is a sell short order exempt from notification	Boolean
IsStop	Whether the order is a stop order	Boolean
IsTimed	Whether the order is a timed order	Boolean
IsTriggered	Whether the order has been triggered	Boolean
IsTriggeredTime	Whether the order was triggered after a time submission	Boolean
IsVolatility	Whether the order is a volatility order	Boolean
Key	Internal static key that represents the order throughout its lifetime	String
LinkType	Type of link identifying the relationship between orders. Values include: Algo AlgoC AlgoL AS ASQ ASH ASR OCO OMA SSE SSEC SSER Stage-BKR Stage-INT Stage-UNK UNM **Note**: In some instances an order can have more than one link type, separated by a comma (e.g. SSE, OCO).	String

Table 1. TTDropHandler Get attributes

Trading Technologies International, Inc. Proprietary and Confidential Information

Attribute	Description	DataType
Limit Limit# Limit$ Limit&	Limit used during price reasonability checks. You can also use one of the following modifiers to display the value in a specific format: **$** : X_TRADER display format **#** : Decimal value **&** : Ticks	As specified
Location	Order sender country and region code.	String
Member	Member ID of the trader from the Member/Group/Trader ID	String
MinQty	Minimum quantity needed to be immediately available at the limit price for this order to be submitted into the market	Integer
Msg	Exchange or gateway message describing the reason for an order reject or order delete	String
OldOrderNo	Previous number used for the order, if the exchange created a new order number	String
OnHold	Whether an order is on hold (out of the market)	Boolean
OpenClose OpenClose$	Whether the order opens or closes a position. Possible values include: **O**: Open **C**: Close **F**: FIFO Specifying **OpenClose$** returns the corresponding full string. **Note:** The TOCOM Exchange changes the **OpenClose** value of a fill corresponding to a FIFO order as either **C** (Close) or **O** (Open).	String

OrdAction	Action associated with the order. Possible values include:
	`ORD_ACTION_ADD`
	`ORD_ACTION_CHANGE`
	`ORD_ACTION_DELETE`
	`ORD_ACTION_SUBMIT`
	`ORD_ACTION_HOLD`
	`ORD_ACTION_REPLACE`
	`ORD_ACTION_UNKNOWN`
		String

OrderNo	Order number assigned by the exchange. **Note:** For CME products, this property contains a different representation of the order number that matches the format used by other exchanges. To see the actual order number received from the CME, check the **ExOrderID** attribute.	String
OrderQty	Number of contracts contained in the order	Integer
OrderRestr	Restriction on the order. For a list of order restrictions, see the section called Order restrictions, on page 555.	String

Table 1. TTDropHandler Get attributes

Trading Technologies International, Inc. Proprietary and Confidential Information

Attribute	Description	DataType
OrderSource OrderSource$	Where the order originated: **0**: X_TRADER **1**: Autospreader Desktop **2**: Autotrader **3**: XTAPI **4**: Ext Source (Exchange) **5**: FA Licensed (Licensed FIX Adapter) **6**: FA Unlicensed (Unlicensed FIX Adapter) **7**: XT Lite (X_TRADER Lite) **8**: TT_TRADER **9**: Autospreader Engine **10**: Synthetic SE Engine **11**: TT API in X_TRADER Mode **12**: Algo SE **21**: TT API in Universal Login Mode **22**: ADL Specifying **OrderSource$** returns the corresponding string.	As specified
OrdStatus OrdStatus&	Indication of the response to the most recent action performed on the order. Possible values include:	As specified
OrderTag	User-definable text (15 char max) containing user-defined information included in the corresponding order profile.	String
OrderTraits	Traits associated with the order as a concatenated string containing the order status history, order flag, and order restriction.	String

Integer	String	Description
0	Void	Default value
8200	New	New order released by client
8201	Accepted	Order accepted by the server
8202	OK	Order active in the market
8203	XT Filled	Order filled
8204	Hold	Order is on hold
8205	Deleted	Order has been deleted
8206	Expired	Order has expired
8207	Rejected	Order was rejected
8208	Resubmitted	Hold order was resubmitted
8299	Unknown	Order status could not be retrieved

Table 1. TTDropHandler Get attributes

Trading Technologies International, Inc. Proprietary and Confidential Information

Attribute	Description	DataType
OrderType OrderType$	Type of order. Possible values include: **B** (OCO): OCO order **C** (CMO): CMO order **L** (Limit): Limit order **M** (Mkt): Market order **O** (BatchOrder): Batch order **Q** (Quote): Quote order **S** (BL): Best Limit order (TOCOM only) **T** (MTL): Market To Limit order **X** (Cross): Cross order The text in parenthesis indicates the value associated with the $ suffix.	String

OrigOrdAction	Action specified in the original order submission, which might differ from the current order action. For example, a Cancel/Replace order comprises two separate order actions: Delete and Add. This attribute stores the original action (Replace, in this example). Possible values include:
	`ORD_ACTION_ADD`
	`ORD_ACTION_CHANGE`
	`ORD_ACTION_DELETE`
	`ORD_ACTION_SUBMIT`
	`ORD_ACTION_HOLD`
	`ORD_ACTION_REPLACE`
	`ORD_ACTION_UNKNOWN`
		String

ParentKey	Site order key identifying the parent order.	String
Price Price# Price$ Price&	Order price. You can also use one of the following modifiers to display the value in a specific format: **$** : X_TRADER display format **#** : Decimal value **&** : Ticks	As specified
PriceCheck	Whether price reasonability checking is enabled	String
Qty	Number of contracts contained in the order	Integer
RecoverableFailure	Whether an order was rejected temporarily, but might succeed in the future	Boolean
Sender Sender$	IP address of the machine that submitted the order, as an integer or string	As specified
SenderSubID	Unique trader (or operator) ID used to distinguish between different traders when the current MGT routes orders for multiple traders. **Note:** Some exchanges, such as CME, require clients to provide this information when routing orders on behalf of multiple traders.	String (32 character limit)
Series.getAttribute	Value of the specified attribute in the **`TTInstrObj`** object	Various
SeriesKey	Internal ID that uniquely identifies the instrument contained in the order	String
SiteKey SiteOrderKey	TT-generated static key that represents the order throughout its lifetime	String

Table 1. TTDropHandler Get attributes

Trading Technologies International, Inc. Proprietary and Confidential Information

Attribute	Description	DataType
Source	Where the order originated: **0**: X_TRADER **1**: Autospreader Desktop **2**: Autotrader **3**: XTAPI **4**: Ext Source (Exchange) **5**: FA Licensed (Licensed FIX Adapter) **6**: FA Unlicensed (Unlicensed FIX Adapter) **7**: XT Lite (X_TRADER Lite) **8**: TT_TRADER **9**: Autospreader Engine **10**: Synthetic SE Engine **11**: TT API in X_TRADER Mode **12**: Algo SE **21**: TT API in Universal Login Mode **22**: ADL	Integer
Stop Stop# Stop$ Stop&	Stop price. You can also use one of the following modifiers to display the value in a specific format: **$** : X_TRADER display format **#** : Decimal value **&** : Ticks	As specified
Tag(x)	User-defined tag, where x is any integer	String
Time Time$	Time the CoreAPI sent the order to the exchange, in the forms: Variant containing a `COleDateTime` structure *HH*:*MM*:*SS*:*mmm* The value contains valid data only in the time portion of the `COleDateTime` object.	As specified
TimeExchange TimeExchange$	Time the exchange received the order. The time represents the local time at the exchange, in the forms: Variant containing a `COleDateTime` structure *HH*:*MM*:*SS*:*mmm* The value contains valid data only in the time portion of the `COleDateTime` object.	As specified
TimeLocalUpd TimeLocalUpd$	Time the XTAPI received the order, in the forms: Variant containing a `COleDateTime` structure *HH*:*MM*:*SS*:*mmm* The value contains valid data only in the time portion of the `COleDateTime` object.	As specified
TimeProcessed TimeProcessed$	Time the TT Gateway processed the order and sent it to the exchange, in the forms: Variant containing a `COleDateTime` structure *HH*:*MM*:*SS*:*mmm* The value contains valid data only in the time portion of the `COleDateTime` object.	As specified

Table 1. TTDropHandler Get attributes

Trading Technologies International, Inc. Proprietary and Confidential Information

Attribute	Description	DataType
TimeReceived TimeReceived$	Time the TT Gateway received the order from the XTAPI, in the forms: Variant containing a `COleDateTime` structure *HH*:*MM*:*SS*:*mmm* The value contains valid data only in the time portion of the `COleDateTime` object.	As specified
TimeRouted TimeRouted$	Time the XTAPI sent the order to X_TRADER®, in the forms: Variant containing a `COleDateTime` structure *HH*:*MM*:*SS*:*mmm* The value contains valid data only in the time portion of the `COleDateTime` object. NOTE: These values exist only for the lifetime of your application and are not visible to other applications.	As specified
Trader	Trader ID of the trader from the Member/Group/Trader ID	String
TriggerQty	Minimum traded quantity required for a stop to trigger at the specified stop price	Integer
UserData(x)	User-defined tag, where x is any integer Valid values: 0-1023	Integer
UserTag	User-definable text (15 char max) containing user-defined information included in the corresponding order profile.	String
Username	Trader login. The value differs based on the type of credentials used: For MGT-based logins, the value contains the concatenation of the Member, Group, and Trader attributes. For universal logins, the value contains the trader's login ID	String
Wrk WrkQty	Order quantity that is working and has not been filled	Integer

Table 1. TTDropHandler Get attributes

Trading Technologies International, Inc. Proprietary and Confidential Information

Trading Technologies International, Inc. Proprietary and Confidential Information

D Fill Attributes

TTFillObj object attributes

About the TTFillObj object

The XTAPI stores information about fills in the `TTFillObj` object. When the XTAPI RTD Server requests information about a fill, the XTAPI looks for the attributes defined for the object. You can request values for any of the `TTFillObj` object attributes.

Available attributes

You can access information about a fill with the following XTAPI `TTFillObj` attributes.

Attribute	Description	DataType
AccountType	Code indicating the desired account type	String
Acct	Customer's or trader's account number	String
AcctType	Code indicating the desired account type	String
Broker	Broker's name (multi-broker only)	String
Broker#	Broker's ID (multi-broker only)	String
Buy	Number of contracts bought (same as Long)	Integer
BuySell BuySell$	Type of order. Possible values include: **B**: Buy **S**: Sell For a buy order, BuySell returns B, while BuySell$ returns Buy.	String
ClearingDate	Clearing date for the fill, as specified by an Exchange. Some Exchanges, such as the TT SGX Gateway, use this attribute to support different dates for when a trade executes (**DateExec** attribute) and when the clearing firm clears the trade (**ClearingDate** attribute).	COleDateTime
ClrMember	Clearing Member ID	String
CntrClearing	Counter party trader's Firm ID	String
CntrMember	Counter party trader's Member ID	String
ColorPri ColorSec	Color to display in the MD TRADER® working orders grid, where: **ColorPri** represents the order's main color (right-side of the grid) **ColorSec** represents the order's secondary color (left-side of the grid) The value is an integer representing the RGB value of the color. The value corresponds to the Microsoft `ColorRef` data type.	Integer
ComboCode	Combination code for the fill. Possible values include: "" (empty string): Non-spread fill **C**: Calendar spread **F**: All other spread fills	String
Company	Broker's company name (multi-broker only)	String

Table 1. TTDropHandler Get attributes

Trading Technologies International, Inc. Proprietary and Confidential Information

Attribute	Description	DataType
Company#	Broker's company ID (multi-broker only)	String
ConnectionName	Broker's connection name (multi-broker only)	String
Contract Contract$	Instrument's name with expiration (e.g. ES Sep13)	String
Contract.objAttribute	Object attribute for the attached instrument (**`TTInstrObj`**)	Various
ContractAlias	Instrument's alias name with expiration (eg. E-Mini Sep13)	String
Date	Date of the order in *MM*/*DD*/*YY* (e.g. 03/18/13) The value contains valid data only in the date portion of the `COleDateTime` object.	COleDateTime
DateExec	Date the fill occurred in *MM*/*DD*/*YY* (e.g. 03/18/13) The value contains valid data only in the date portion of the COleDateTime object.	COleDateTime
Ex:OrderNo	Order number for the specified exchange (eg. CME-S: 107578464)	String
Exchange	Name of the TT Gateway that sent the fill	String
ExchCred	Exchange credentials (multi-broker only)	String
ExGroup	Exchange Group ID for the fill	String
ExMember	Exchange Member ID for the fill	String
ExOrderID	Exchange Order ID assigned to the fill	String
	Note: If the exchange does not provide a value, the X_TRADER API populates it with the OrderNo attribute value.	
ExRecNo	Transaction ID provided by the exchange	String
ExTrader	Exchange Trader ID for the fill	String
ExTransNo	Transaction ID provided by the exchange	String
FFT1 FFT2 FFT3	Free-form text fields. FFT1 typically contains the account.	String
Fee	Fee code for the fill. The fee codes are exchange-dependent.	String
FillKey	Fill ID supplied by the exchange	String
FillTicks	Total value of the fill in ticks	Integer
FillType	Type of fill. Possible values include: **P**: Partial **F**: Full	String
GiveUp	Give-up member's clearing ID, when the fill is assigned to another member	String
	Note: The TT ICE Gateway does not support the **GiveUp** attribute; instead, it uses the **ClrMember** attribute for Give Up orders.	
Group	Group name of the trader from the Member/Group/Trader ID	String
Instr Instr$	Default instrument name, as displayed in the XTAPI **Market Grid Contract** field (e.g., ES Sep13)	String

Table 1. TTDropHandler Get attributes

Trading Technologies International, Inc. Proprietary and Confidential Information

Attribute	Description	DataType
InstrAlias	Instrument's alias name with expiration (e.g. E-Mini Sep13)	String
Instr.objAttribute	Value of the specified object attribute for the related **`TTInstrObj`** object **Note:** You can access only static, contract-related attributes, such as **Contract**, **Expiry**, and **Exchange**.	Various
IsAON	Whether the order is an All-Or-None order	Boolean
IsAuction	Whether the order is only valid for auction	Boolean
IsAuto-Agress IsAuto-Aggress	Whether the order is auto-aggressive (Currently supported only for BrokerTec)	Boolean
IsAutomated	Whether the matching order orginated from an automated program.	Boolean
IsAwayMarket	Whether an away market maker should match BBO of an away market (Currently supported only for BrokerTec)	Boolean
IsBasis	Whether the fill represents a basis order	Boolean
IsBestLimit	Whether the fill represents is a best limit order	Boolean
IsBlock	Whether the fill represents a block order.	Boolean
IsBuy	Whether the fill represents a buy order	Boolean
IsConfirmed	Whether the fill is a confirmed fill	Boolean
IsDetailedSOD	Whether the fill represents a detailed SOD record	Boolean
IsFAS	Whether the fill represents a fill and save order	Boolean
IsFlex	Whether the fill represents a flex option order	Boolean
IsFOK	Whether the fill represents a fill or kill order	Boolean
IsIceberg	Whether the fill represents an iceberg order	Boolean
IsIfTouched	Whether the order requires additional action when a price is touched	Boolean
IsInverted	Whether the fill is canceled and that the trader’s position has been adjusted to reflect the cancelled fill	Boolean
IsIOC	Whether the fill represents an immediate-or-cancel order	Boolean
IsLOC	Whether the fill represents a market to limit order to be filled or killed on market close	Boolean
IsLOO	Whether the fill represents a market to limit order to be filled or killed on market open	Boolean
IsLSM	Whether the fill represents a limit-stop-market order	Boolean
IsLTM	Whether the fill represents a limit order to be converted to a market order on close	Boolean
IsManualFill	Whether the fill was entered as a manual fill	Boolean
IsMarketToLimit	Whether the fill represents a market-to-limit order	Boolean
IsMLM	Whether the fill represents a market-limit-market order	Boolean
IsMOC	Whether the fill represents a market-on-close order	Boolean

Table 1. TTDropHandler Get attributes

Trading Technologies International, Inc. Proprietary and Confidential Information

Attribute	Description	DataType
IsMOO	Whether the fill represents a market-on-open order	Boolean
IsMTL	Whether the fill represents a market-to-limit order	Boolean
IsMV	Whether the fill represents a minimum volume order	Boolean
IsNon-leave	Whether the fill represents a non-leave order (Currently supported only for BrokerTec)	Boolean
IsPassive	Whether the fill represents a passive (Currently supported only for BrokerTec)	Boolean
IsSell	Whether the fill represents a sell order	Boolean
IsShortExempt	Whether the fill represents a sell short order exempt from notification	Boolean
IsSOD	Whether the fill is a start-of-day fill	Boolean
IsStop	Whether the fill represents a stop order	Boolean
IsTimed	Whether the fill represents a timed order	Boolean
IsTriggered	Whether the order has been triggered	Boolean
IsTriggeredTime	Whether the order was triggered after a time submission	Boolean
IsVolatility	Whether the fill represents a volatility order	Boolean
Key	Internal static key that represents the order throughout its lifetime	SiteOrderKey
LinkType	Type of link identifying the relationship between orders. Values include: Algo AlgoC AlgoL AS ASQ ASH ASR OCO OMA SSE SSEC SSER Stage-BKR Stage-INT Stage-UNK UNM **Note**: In some instances an order can have more than one link type, separated by a comma (e.g. SSE, OCO).	String
Location	Order sender country and region code.	String
Long	Number of contracts bought (same as Buy)	Integer
Member	Member ID of the trader from the Member/Group/Trader ID	String
NetQty	Total quantity of contracts or shares executed for this fill	Integer

Table 1. TTDropHandler Get attributes

Trading Technologies International, Inc. Proprietary and Confidential Information

Attribute	Description	DataType
NetTicks	Total value of the fill in ticks	Integer
OpenClose OpenClose$	For market fills, whether the fill opened or closed a position; for injected fills, the source of an injected fill. Possible values include: **A**: XRisk Admin **C**: Close **D**: Detailed SOD **F**: FIFO **M**: Manual Fill **O**: Open **R**: Rollover **S**: SOD **Note:** The TOCOM Exchange changes the **OpenClose** value of a fill corresponding to a FIFO order as either **C** (Close) or **O** (Open).	String
OrderNo	Order number assigned by the exchange. For CME products, this property contains a different representation of the order number that matches the format used by other exchanges. To see the actual order number received from the CME, check the **ExOrderID** attribute.	String
OrderRestr	Order restriction on the order. For a list of possible values, see Order restrictions, on page 553.	String
OrderSource OrderSource$	Where the order originated: **0**: X_TRADER **1**: Autospreader Desktop **2**: Autotrader **3**: XTAPI **4**: Ext Source (Exchange) **5**: FA Licensed (Licensed FIX Adapter) **6**: FA Unlicensed (Unlicensed FIX Adapter) **7**: XT Lite (X_TRADER Lite) **8**: TT_TRADER **9**: Autospreader Engine **10**: Synthetic SE Engine **11**: TT API in X_TRADER Mode **12**: Algo SE **21**: TT API in Universal Login Mode **22**: ADL	Integer or String
OrderTag	User-definable text (15 char max) containing user-defined information included in the corresponding order profile.	String
OrderTraits	Traits associated with the order as a concatenated string containing the order status history, order flag, and order restriction.	String

Table 1. TTDropHandler Get attributes

Trading Technologies International, Inc. Proprietary and Confidential Information

Attribute	Description	DataType
OrderType	Type of order corresponding to the fill. Possible values include: **B**: OCO order **C**: CMO order **L**: Limit order **M**: Market order **O**: Batch order **Q**: Quote order **S**: Best Limit order (TOCOM only) **T**: Market To Limit order **X**: Cross order	String
ParentKey	Site order key identifying the parent order	String
Partial	Type of fill executed. Possible values include: **F**: Full fill **P**: Partial fill **Note:** A TT Eurex Gateway that enables Fast Fills does not set this attribute to "F" for partial Fast Fills. When your application later reconciles its fills with the Eurex Fill Server, one of the partial fills marked as "P" will be changed to "F" to indicate which fill completely filled the order.	String
Price Price$ Price# Price&	Fill price. You can also use one of the following modifiers to display the value in a specific format: **$** : X_TRADER display format **#** : Decimal value **&** : Ticks	As specified
Price@	Fill price converted to the primary currency (as set in the Guardian currency tables). If you access this attribute before the currency table finishes loading, the attribute returns TT_INVALID_PRICE.	Decimal
Price[Currency]	Current market value of open positions in the currency whose code you specify (Currency). Valid only when the instrument has an attached **`TTOrderSet`**.	Decimal
Price^	Fill price converted to the native currency (as set in the Guardian currency tables)	String
Qty	Number of contracts contained in the fill	Integer
RiskAccount	ID for the risk account	String
Sell	Number of contracts sold (same as Short)	Integer
SenderSubID	Unique trader (or operator) ID used to distinguish between different traders when the current MGT routes orders for multiple traders. **Note:** Some exchanges, such as CME, require clients to provide this information when routing orders on behalf of multiple traders.	String (32 character limit)
Series Series$	Unique ID for the instrument, provided by the exchange	String
Series.objAttribute	Value of the specified object attribute for the instrument (**`TTInstrObj`**)	Varied

Table 1. TTDropHandler Get attributes

Trading Technologies International, Inc. Proprietary and Confidential Information

Attribute	Description	DataType
SeriesKey	Internal ID that uniquely identifies the instrument contained in the order	String
SessionId	Session ID for the fill	Integer
Short	Number of contracts sold (same as Sell)	Integer
SiteKey SiteOrderKey	Internal static key that represents the order throughout its lifetime	String
SOD	Whether the fill represents a start-of-day record	Boolean
SODType	Type of start-of-day message. Possible values include: "Start of Day" "Current" "Correction"	String
Source	Source of the fill. For normal fills, the value contains a unique ID for the Fill server. For manual fills generated by an XTAPI application, the value contains an ID based on the system running the application.	String
Tag(x)	User-defined tag, where x is any integer.	String
Time Time$	Time the X_TRADER API sent the order to the exchange, in the form HH:MM:SS:mmm.	Time String
TimeExec TimeExec$	Time the exchanged matched the trade, in the form *HH*:*MM*:*SS*:*mmm*.	Time String
TimeLocalUpd TimeLocalUpd$	Time the X_TRADER API received the fill, in the form HH:MM:SS:mmm.	String
Trader	Trader ID of the trader from the Member/Group/Trader ID.	String
TTFillCode	Fill code associated with the fill. Possible values include: **1000**: Normal fill **1001**: Fast Fill **1002**: Reverse Trade **1003**: Adjustment **1004**: Clearing Error **1005**: Rollover	Integer
UserData(*x*)	User-defined tag, where *x* is any integer. Valid values: 0-1023	Integer
UserTag	User-definable text (15 char max) containing user-defined information included in the corresponding order profile.	String
Username	Trader login. The value differs based on the type of credentials used: For MGT-based logins, the value contains the concatenation of the Member, Group, and Trader attributes. For universal logins, the value contains the trader's login ID	String

Table 1. TTDropHandler Get attributes

Trading Technologies International, Inc. Proprietary and Confidential Information

Trading Technologies International, Inc. Proprietary and Confidential Information

Send Us Your Comments

X_TRADER API RTD Server Feature Guide Feature Guide 7.X

Trading Technologies® welcomes your comments and suggestions on the accuracy and usefulness of this publication. Your input is important and valuable in revising our documentation and helps ensure a constantly improving level of quality.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- Which features did you find particularly useful?
- What did you like most about this manual or document?

If you encounter any errors in this document or would like to share other suggestions you might have for improving this document, send comments to: documentation.dept@tradingtechnologies.com.

If possible, please indicate the chapter, section, and page number relevant to your feedback.