

CQG, Inc

# XLS Trader

# ユーザーガイド

**Feb 15th, 2021 | Based on Software version 2020**

<変更履歴>

バージョン	章	変更内容

## Contents

XLS Trader .....	5
要件 .....	5
インストール .....	6
Trade System の作成 .....	7
関数の閲覧と入力 .....	9
アカウントの設定 .....	10
Entries (新規注文) .....	12
Boolean パラメーター .....	15
Exits (仕切注文) .....	16
キャンセル注文関数 .....	18
OCO の設定 .....	20
XLS TRADER で取引 .....	25
Execution Instructions (スマートオーダータイプ) .....	27
All or Nothing 関数 .....	27
Cancel If Price Is Invalid 関数 .....	28
Execution Instructions 関数 .....	28
Funari (不成) 関数 .....	29
Iceberg 関数 .....	29
Market if Touched 関数 .....	30
Open Close Instruction 関数 .....	30
Quantity Triggered Stop 関数 .....	31
Trailing 関数 .....	31
Trade System 関連の関数 .....	32
Default Duration 関数 (デフォルトの有効期限を設定) .....	32
Default Symbol 関数 (デフォルトのシンボルを設定) .....	33
Disable New Entries 関数 (新規発注を停止) .....	34
Is Reversal System 関数 (ドテン設定) .....	35
Maximum Position 関数 (最大建玉枚数の設定) .....	36
Maximum Pyramid 関数 (ピラミッド注文の上限設定) .....	36
Order 関数 .....	37
注文追跡、新規取引 and 取引情報関数 .....	38
約定情報取得関数グループ .....	38
FillPrice 関数 (約定値段取得) .....	38
FillSize 関数 (約定枚数取得) .....	39
FillTime 関数 (約定時間) .....	40
新規約定、ワーキング注文、合算の注文件数 .....	40
NumberOfOpenEntries 関数 (新規約定件数) .....	40
NumberOfOrders 関数 (新規約定およびワーキング注文件数) .....	41
NumberOfWorking 関数 (ワーキング注文件数) .....	41

新規約定情報取得関数グループ .....	42
OpenEntryPrice 関数（新規約定値段） .....	42
OpenEntrySize 関数（新規約定枚数） .....	42
OpenEntryTime 関数（新規約定時間） .....	43
注文詳細取得関数グループ .....	43
Order Average Filled Price 関数（約定平均値） .....	43
Order Filled Size 関数（約定枚数） .....	44
Order Limit Price 関数（注文の指値） .....	44
Order Place Time 関数（発注時間） .....	45
Order Size 関数（注文枚数） .....	45
Order Status 関数（注文のステータス） .....	46
Order Stop Price 関数（ストップ注文の値段を取得） .....	47
Working 注文関数グループ .....	48
WorkingLimitPrice 関数 .....	48
WorkingPlaceTime 関数 .....	48
WorkingSize 関数 .....	49
WorkingStopPrice 関数 .....	49
取引情報関数グループ .....	50
Trade Profit 関数 .....	50
Trade Size 関数 .....	50
Trade Time 関数 .....	51
注文詳細を保存する関数 .....	52
Open Trade Equity 関数 .....	52
Closed Net Profit 関数 .....	53
CSV Fills 関数 .....	54
CsvTrades 関数 .....	55
オーダーログ関数 .....	57
XLS Trader ログ関数 .....	62
ユーティリティ関数 .....	64
ApiTimeout 関数 .....	64
Instrument 関数 .....	65
Connectivity アイコン .....	65
LineTime 関数 .....	66
NAPlaceholder 関数 .....	66
Order Transaction Per Second Limit 関数 .....	67
XLS Trader Version 関数 .....	67
XLS Trader コマンドバー ボタン .....	68
XLS TRADER を閉じる際の注意 .....	70

# XLS Trader

XLS Trader は、エクセルにて自動売買のストラテジーを構築するツールです。

このツールは 70 を超える関数を実装し、それらの関数で計算した様々な数値を用いて、直接エクセルから取引の執行および制御を行います。

下図は、XLS Trader のアドインをインストールすると追加される関数式の一覧です。

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1		<b>Trade System</b>		<b>Logs</b>			<b>Orders (Monitoring)</b>			<b>Execution Instructions</b>			<b>Back Office</b>		<b>Utility</b>						<b>Orders</b>
2		=DefaultAccount()		=AllOrdersLog()			=FillPrice()			=AllOrNothing()			=AverageEntryPrice()		=ApiTimeout()						=AddFill()
3		=DefaultDuration()		=CancelledOrdersLog()			=FillSize()			=CancelPricelsInvalid()			=ClosedNetProfit()		=DefaultMIFIDAlgold()						=Buy()
4		=DefaultSymbol()		=FailedOrdersLog()			=FillTime()			=ExecInstr()			=CsvFills()		=Instrument()						=CancelAll()
5		=DisableNewEntries()		=FilledOrdersLog()			=NumberOfOpenEntries()			=Funari()			=CsvTrades()		=IsMarketDataConnected()						=Exit()
6		=IsReversalSystem()		=WorkingOrdersLog()			=NumberOfOrders()			=Iceberg()			=OpenPosition()		=IsOrderRouterConnected()						=Limit()
7		=MaxPosition()		=XlsTraderLog()			=NumberOfWorking()			=MarketIfTouch()			=OpenTradeEquity		=LineTime()						=Market()
8		=MaxPyramid()					=OpenEntryPrice()			=OpenCloseInstruction()					=LineTimeLag()						=Sell()
9		=TradeSystem()					=OpenEntrySize()			=QuantityTriggeredStop()					=NAPlaceholder()						=Stop()
10							=OpenEntryTime()			=Trailing()					=OrderTransactionPerSecondLimit()						=StopLimit()
11							=OrderAverageFilledPrice()								=XlsTraderVersion()						
12							=OrderFilledSize()														
13							=OrderLimitPrice()														
14							=OrderPlaceTime()														
15							=OrderSize()														
16							=OrderStatus														
17							=OrderStopPrice()														
18							=TradeProfit()														
19							=TradeSize()														
20							=TradeTime()														
21							=WorkingLimitPrice()														
22							=WorkingPlaceTime()														
23							=WorkingSize()														
24							=WorkingStopPrice()														
25																					
26																					
27																					

## 要件

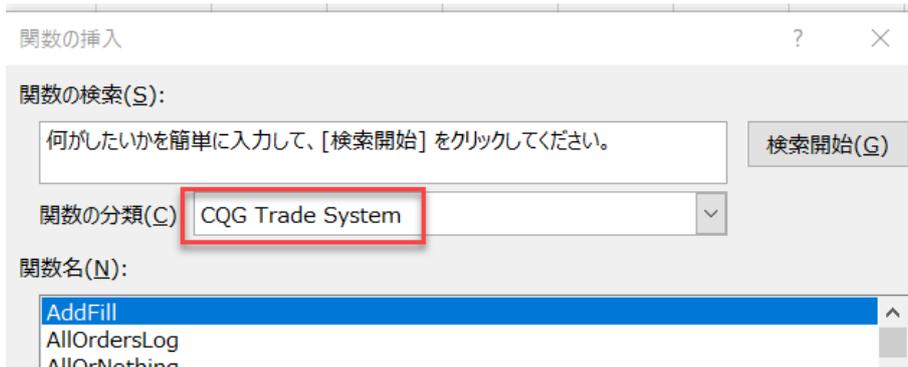
- お使いの CQG IC に “XLS Trader”、“Back Testing” および “CQG Trading” の使用権限 (Enablement) が付与されている
- CQG IC のバージョンが CQG Integrated Client 2020 (20.12.XXXX) 以上
- Microsoft Office Professional Excel 2016, 2019, 32 or 64-bits がインストールされている (Cloud 上は推奨しません)
- CPU および RAM は [システム仕様](#) 参照
- Microsoft Windows 10 以上

## インストール

下記リンクより XLS Trader をダウンロードし、インストールしてください。

<https://downloads.cqg.com/XLSTrader/>

インストール後エクセルを開き、関数の挿入ダイアログ（Shift+F3）を表示してください。XLS Trader のインストールが完了していれば、CQG Trade System が関数の分類ドロップダウンリストに表示されます。

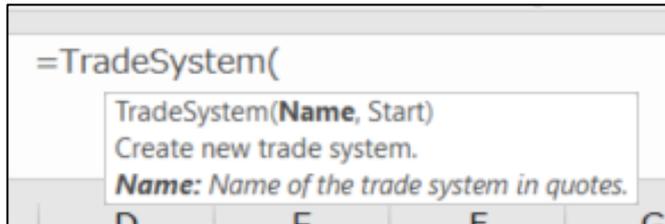


## Trade System の作成

**注** : Trade System を実行するためには、まず CQG IC より Order Routing にログインしてください。

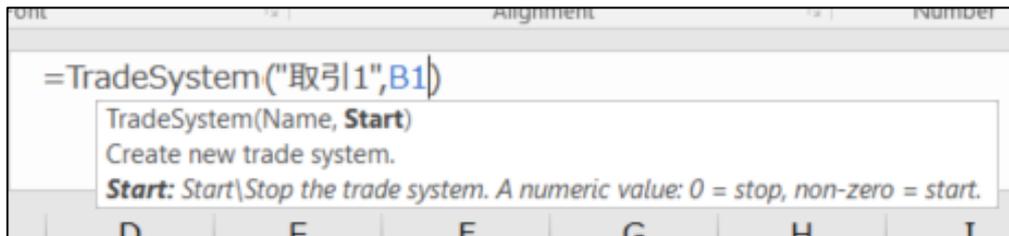
Trade System を構築するためには、まず Trade System 関数にて売買ストラテジーの作成を宣言する必要があります。

任意のセル、例えばセル A1 を選択し、“=TradeSystem(” と入力すると、下図のとおりツールチップが表示されます。



TradeSystem 関数内の、Name パラメーターで Trade System に名前を付けます。そして、Start パラメーターで Trade System の稼働/停止するためのセルを参照します。

下図の場合、“取引 1” が Trade System 名となり、続いてカンマ ‘,’ を入力するとその次のパラメーターツールティップが表示されます。



上図の関数式の場合、稼働/停止スイッチとしてセル B1 が参照されています。最後に ‘)’ にて TradeSystem 関数を終了させます。そして下図のとおり Start パラメーターのセル B1 に “0” を入力し、Trade System を停止状態にしておきます。

	A	B
1	取引1	0
2		

セル A1 には TradeSystem 関数式にて設定した “取引 1” が表示されています。

こちらのセルには Trade System 名が表示されるだけでなく、Trade System の稼働/停止も確認できます。セル B1 を 0 から 1 に変更すると、下図のとおり “- Running” と表示され、Trade System の稼働が確認できます。

	A	B
1	取引1 - Running	1

Trade System 関数の稼働／停止はその配下の全関数に影響を与えます。つまり、Trade System 関数が停止している場合は、その配下にあるいずれの関数も停止します。

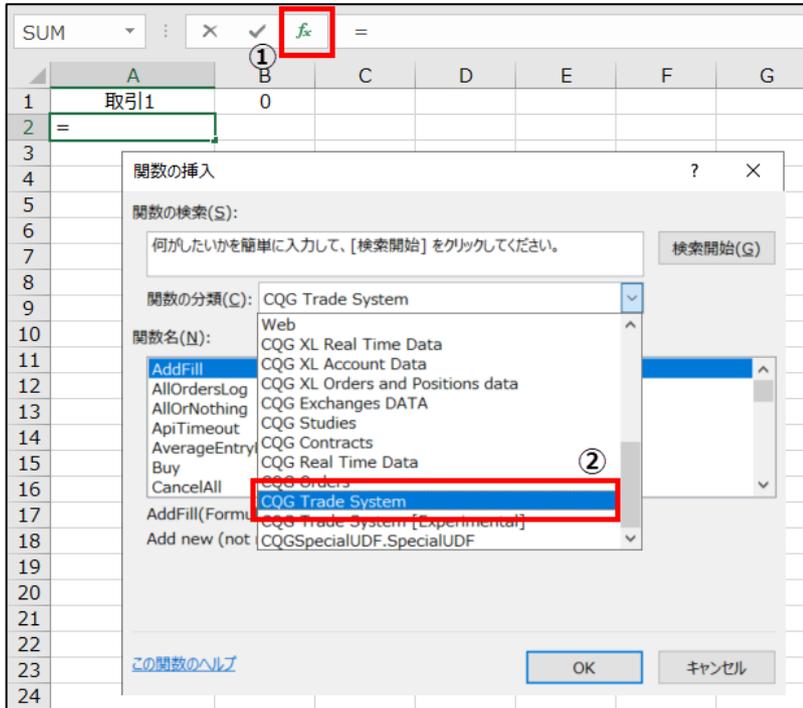
Trade System 関数にて Trade System を宣言した後、新規取引や仕切り取引を作成したり、ワーキングオーダー情報やアクティビティログを表示する関数を設定するなどして売買ストラテジーを構築していきます。

**注：** XLS Trader は CQG IC の Order Routing にログインすることによって使用可能になります。まず Order Routing にログインしてから、エクセルファイルを開いてください。ログインしていない場合は XLS Trader が正しく動作しません。その場合はエクセルを一旦終了させ、Order Routing にログインし、再度エクセルファイルを立ち上げてください。

また、Trade System が稼働している状態だと、パラメーターの変更が反映されません。変更を反映させるためには、一旦 Trade System を停止し、再度起動させてください。

## 関数の閲覧と入力

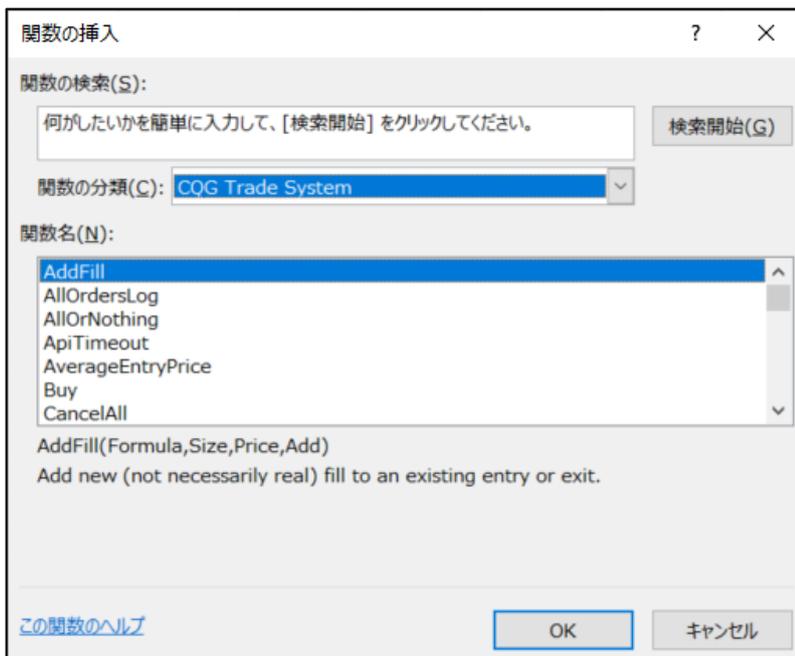
XLS Trader の関数一覧を閲覧するためには、①関数挿入ボタンをクリックして関数の挿入ウィザードを表示してください。その後、ドロップダウンより ② “CQG Trade System” を選択してください。



こちらの画面に XLS Trader に実装されている全ての関数が表示されます。

任意の関数を選択し “OK” をクリックすると、各パラメーターの入力画面が開きます。

関数の入力セルに直接打ち込むこともできますが、こちらのウィザードを利用すればより簡単に設定できます。



## アカウントの設定

CQG ICにて単一のアカウントのみ使用している場合は、Trade System内でのアカウントの設定は不要です。しかし複数のアカウントを使用している場合は、どのアカウントを使用するか、DefaultAccount 関数で宣言する必要があります。

下図の例では System パラメーターにセル A1 を入力し、Account パラメーターにアカウント番号の“PSYukiMogi”を設定しました。

	A	B	C	D	E	F	G
1	取引1	0					
2	=DefaultAccount(A1,"PSYukiMogi")						

関数の引数

DefaultAccount

System A1 = "取引1"

Account "PSYukiMogi" = "PSYukiMogi"

数式の結果 = PSYukiMogi

この関数のヘルプ(H)

OK キャンセル

アカウント名またはアカウント番号は Orders and Positions ウィンドウのアカウントフィルタードロップダウンにて確認できます。

FCM: Spreader Trade System:

Account: -all- Symbol:

Account	Number	Name
<input checked="" type="checkbox"/>	-all-	
<input checked="" type="checkbox"/>	PSYukiMogi	SIMYukiMogi
<input checked="" type="checkbox"/>		
<input checked="" type="checkbox"/>		
<input checked="" type="checkbox"/>		

Margin Exces

Margin Value

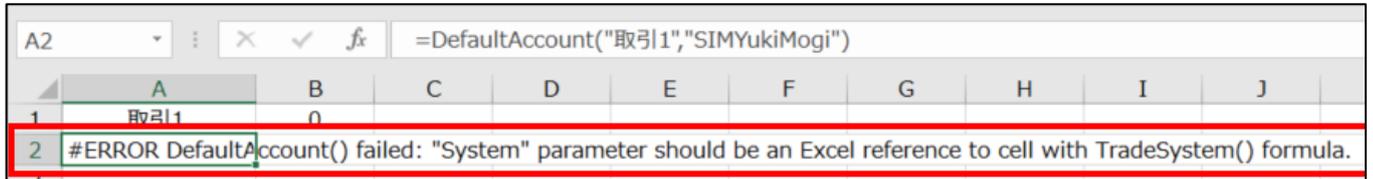
Purchasing P

Margin Credit

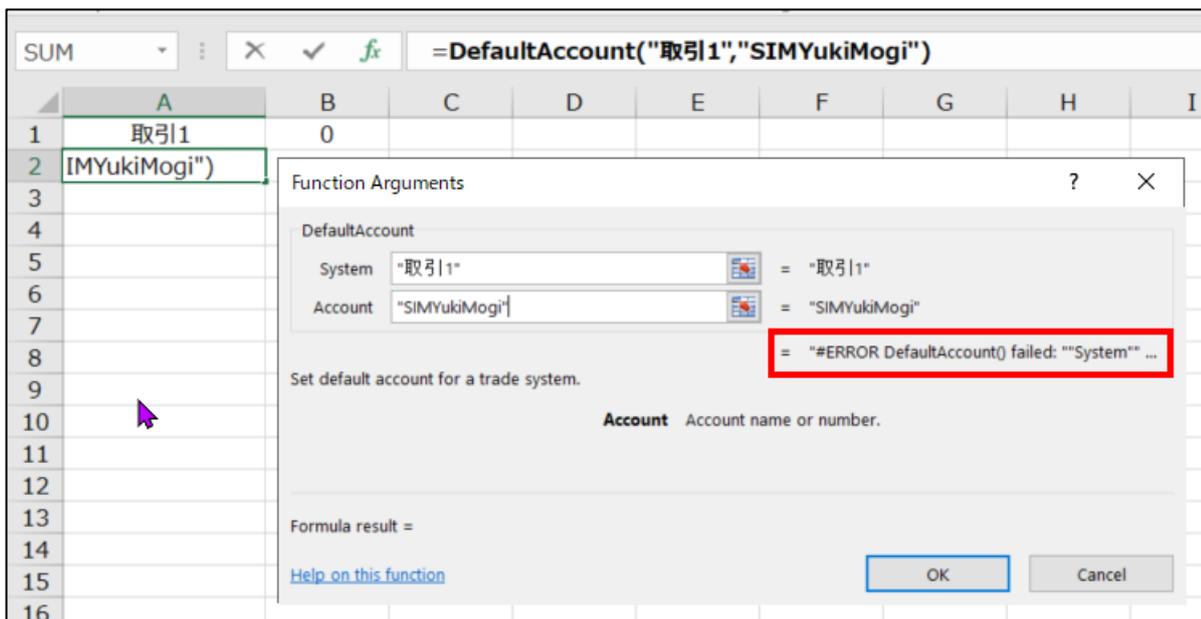
Account Balan

Apply Cancel

Account パラメーターには Account Name、Account Number のどちらも使用できます。DefaultAccount 関数の入力完了すると、そのセルに Account Name/Number が表示されます。入力に誤りがある場合、例えば System パラメーターには “A1” のように Trade System 関数のセル参照を入力する必要がありますが、下図のように “取引 1” と Trade System 名を入力した場合は、エラーメッセージが表示されます。

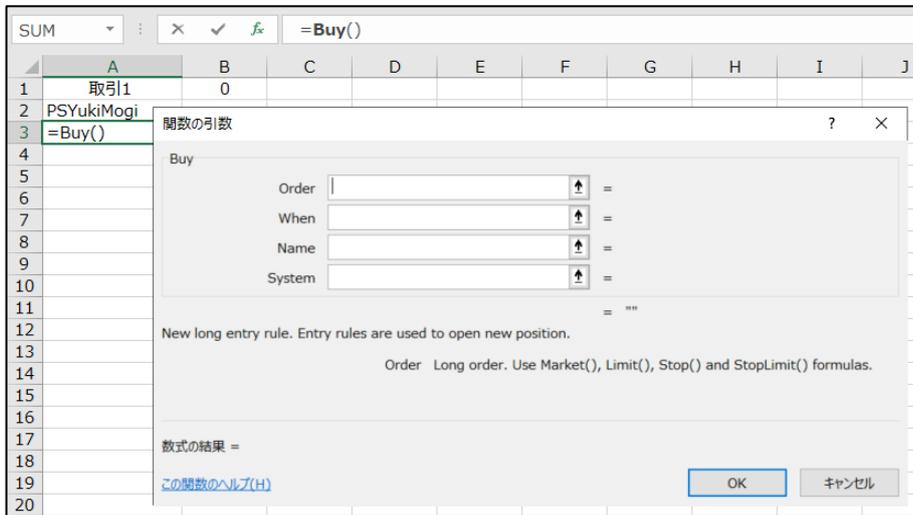


また、関数ウィザード内にて、関数式がエラーであることをあらかじめ確認できます。



## Entries（新規注文）

Trade System とアカウントの設定が完了したので、次は **Entry**（新規発注）の構築に進みます。Entry には **Buy** および **Sell** 関数があります。Buy 関数のウィザードを開くと下図のパラメーターが表示されます。



Buy 関数のパラメーター及び入力例は以下のとおりです。

パラメーター	入力例	“” ダブルクォーテーション
Order	Limit("EP",5,2665.25,"Day") または A4（Limit 関数などの参照セル）	不要
When	B8（稼働／停止の参照セル）	不要
Name	“S&P 買い” など任意の名前	必要
System	A1（Trade System 関数の参照セル）	不要

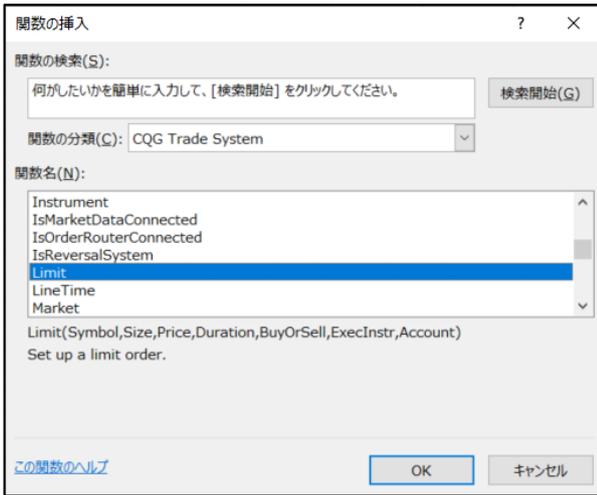
Buy 関数の Order パラメーターには、Entry の注文タイプに応じて、Market、Limit などの関数式を入力します。

上図のウィザード内には Order パラメーターに入力可能な関数が表示されています。

Entry で指値注文を発注する場合は、Order パラメーターに Limit 関数を直接入力するか、もしくは Limit 関数が入力されたセルを参照します。

Order パラメーターに設定する Limit 関数について説明します。

セル A4 に Limit 関数を入力し、Buy 関数（セル A5）の Order パラメーターにてセル A4 をセル参照する場合、セル A4 に “=Limit(” と入力、もしくは **Shift+F3** を押下し関数の挿入画面を開き、プルダウンから **Limit** を選択してください。

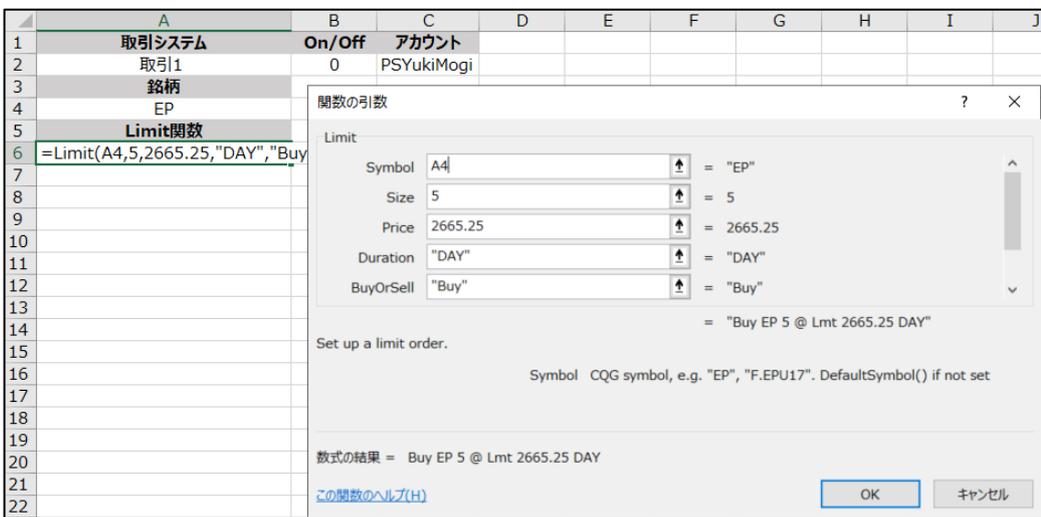


Limit 関数のパラメーター及び入力例は以下のとおりです。

パラメーター	入力例	“”ダブルクォーテーション
Symbol (銘柄コード)	"EP"	要
Size (枚数)	5	不要
Price (値段)	2665.25	不要
Duration (有効期限)	"DAY" or "GTC"	要
BuyorSell (買い or 売り)	"Buy" or "Sell"	要
ExecInstr (※1)	CancelIfPriceIsValid 関数などを使用する場合は入力してください。通常の指値注文はブランクです ( <a href="#">Execution Instructions</a> 参照)。	
Account (※1)	デフォルト以外を使用する場合はアカウント番号/アカウント名を入力してください。	

※1 下図のイメージには表示されていません。表示するにはスクロールバーで下げてください。

上表内の各パラメーターは全て参照セルで対応することもできます。下図のようにセル A4 に “EP” と入力した上で、Symbol パラメーターはセル A4 を参照させます。ウィザード内の参照結果に “EP” が表示されます。また画面下部には計算結果が表示されます。



OK をクリックすると、下図のとおりウィザードにて設定した計算式が表示されます。

	A	B	C
1	取引システム	On/Off	アカウント
2	取引1	0	PSYukiMogi
3	銘柄		
4	EP		
5	Limit関数		
6	Buy EP 5 @ Lmt 2665.25 DAY		
7			

上記例では、ウィザードに指値“2665.25”を直接入力しました。これ以外に、RTD 関数をセル参照させて現在値、前日終値などの数値を設定することもできます。

注意点として、RTD 関数の入力に不備があり、参照セルの値段が#Value! エラー もしくは #N/A エラーとなっていた場合でも、トリガーとなるイベントが発生した場合には注文は発注されてしまいます。もし無効な値の場合には発注を止めたい場合は、下図のとおり ExecInstr パラメーターに CancelIfPriceIsValid 関数を設定してください。

関数の引数

Limit

Size 5 = 5

Price 2665.25 = 2665.25

Duration "DAY" = "DAY"

BuyOrSell "Buy" = "Buy"

**ExecInstr CancelIfPriceIsValid() = "\$EI\_XIP()"**

= "Buy EP 5 @ Lmt 2665.25 XbadPx DAY"

Set up a limit order.

ExecInstr Optional execution instruction formula: ExecInstr(), AllOrNothing(), Funari(), Iceberg(), MarketIfTouch(), OpenCloseInstruction(), QuantityTriggeredStop(), Trailing()

数式の結果 = Buy EP 5 @ Lmt 2665.25 XbadPx DAY

[この関数のヘルプ\(H\)](#) OK キャンセル

Order パラメーターにセル A6 を参照させ、When、Name および System パラメーターを下表例のとおり入力すれば完成です。

	A	B	C	D	E	F	G	H	I	J	K
1	取引システム	On/Off	アカウント								
2	取引1	0	PSYukiMogi								
3	銘柄										
4	EP										
5	Limit関数										
6	Buy EP 5 @ Lmt 2665.25 DAY										
7	Buy関数	執行									
8	=Buy(A6,B8,"S&P買い",A2)	0									
9											
10											
11											
12											
13											
14											
15											
16											
17											
18											
19											
20											
21											

関数の引数

Buy

Order A6 = "Buy EP 5 @ Lmt 2665.25 DAY"

When B8 = 0

Name "S&P買い" = "S&P買い"

System A2 = "取引1"

= "S&P買い = Buy EP 5 @ Lmt 2665.25 ..."

New long entry rule. Entry rules are used to open new position.

System Reference to trading system formula: TradeSystem().

数式の結果 = SP買い = Buy EP 5 @ Lmt 2665.25 DAY when \$B\$8

[この関数のヘルプ\(H\)](#) OK キャンセル

## Boolean パラメーター

多くの関数、TradeSystem、Buy、CancelAll 関数などは Boolean または Binary 設定値をトリガーとして使用します。

例えば Trade System 関数については、バイナリ (Start) パラメーターは 0 (停止) と 1 (稼働) のみです。

しかし Buy/Sell 関数はトリガーとなる When パラメーターの稼働/停止に使用されるバイナリパラメータが 0 (停止) と 0 以外 (稼働) となり、Trade System 関数のトリガーとは若干異なります。

Buy/Sell 関数は When パラメーターが 0 から 1 に変わると注文を発注します。そして 1 から 2 に変わると追加で同じ注文を発注します。さらに 2 から 1 に変わってもまた同じ注文を発注します。つまりパラメーターが 0 以外の数値に変わり続ける限り発注し続けます。

この仕組みを利用して、イベント発生毎に When パラメーターをカウントアップし、発注を行うストラテジーを構築すれば、その数値によって注文が何件発注されているか監視することができます。

イベント発生時の挙動を確認するためには、When パラメーターを手作業で 0 から 1 に変更します。

ただし、ほとんどの場合イベント発生のトリガーは、条件式をセル参照しています。

例えば、IF 関数を使用し、「10 日移動平均が前日の終値よりも高かった場合に 0 から 1 に変更する」などの条件式が入力されたセルを参照させます。

## Exits（仕切注文）

Exit 関数は Trade System 内の Entry 関数で約定した建玉を仕切ります。Exit 関数は基本的に Entry 関数と類似したパラメーターで構成されていますが、若干の相違点があります。

例として Market 注文を用いた Exit 関数をセル A10 に、Limit 注文を用いた Exit 関数を A12 にそれぞれ作成します。Limit 注文を用いた Exit 関数のウィザード画面は以下のとおりです。

パラメーター	入力例	“” ダブルクォーテーション
Formula	A5	不要
Order	Limit(,,2638.75,\"GTC\")	不要
When	Market 関数を使用する場合は入力必須、それ以外は入力しなくても良い。入力する場合はセル参照でなければならない	
Name	“Limit 仕切り” など任意の名前	要

Exit 関数は基本的に Formula パラメーターにて Entry 関数をセル参照するので、Entry 関数の売り買いの別を自動的に認識します。このため Order パラメーターへの Sell/Buy の入力は省略可能です。

ただし Formula パラメーターで Trade System 関数をセル参照している場合は、Sell/Buy の入力が必要となります。

	A	B	C
1	<b>取引システム</b>	<b>On/Off</b>	<b>アカウント</b>
2	取引1	0	PSYukiMogi
3	<b>銘柄</b>		
4	EP		
5	<b>Limit関数</b>	<b>発注価格</b>	
6	Buy EP 5 @ Lmt 2665.25 DAY	2665.25	
7	<b>Buy関数</b>	<b>執行</b>	
8	S&P買い = Buy EP 5 @ Lmt 2665.25 DAY when \$B\$8	0	
9	<b>Exit関数@Market</b>	<b>執行</b>	
10	Market仕切り = Exit S&P買い = Sell 0 @ Mkt DAY when \$B\$10	0	
11	<b>Exit関数@Limit</b>		
12	Limit仕切り = Exit S&P買い = Sell 0 @ Lmt 2638.75 GTC		
13			

繰り返しとなりますが、Exit 関数は Entry 関数で建てた建玉を仕切るため、Formula パラメーターにて参照するセルは Entry 関数か Trade System 関数となります。上図ではセル A8 の Entry 関数“S&P 買い”を参照しています。

Exit 関数の Order パラメーターにおけるシンボル、枚数、売／買は Entry 関数での設定値と同様となります。したがって Exit 関数の Order パラメーターにて、値段、有効期限以外は原則入力不要となります。

Entry 関数の注文内容が Exit 関数に取り込まれていることは、上図ウィザード内、Formula パラメーターの右側に表示されている “= S&P 買い = Buy EP 5 @Lmt 2665.25 DAY When...” から確認できます。このため、たとえ Order パラメーターにて枚数 (Size) を入力せず、ウィザード内の数式の結果に 0 が表示されていても、セル参照している Entry 関数 (S&P 買い) が約定すれば、Exit 関数の注文枚数に数値が設定されます。

Exit 関数にて Limit、Stop など、Market 以外の注文タイプを使用している場合は、When パラメーターの入力は不要です。ただし、未入力の場合は常に True を返していることになるため、Entry 注文が約定後直ちに Exit の指値注文が発注されます。

Exit 関数が利食いを目的としている場合、Exit 注文の指値は “B6 (Entry 注文の指値) + X.XX (0.25)” と設定することもできます。上の例 セル A12 では、もし Entry 注文が 2638.5 で約定した場合は、約定後直ぐに 2638.75 に Exit 注文が入るストラテジーとなっています。

一方、セル A10 の Exit 関数のようにマーケット注文を使用している場合は、トリガーとなる条件を When パラメーターに設定する必要があります。例えばセル移動平均線や Volatility Spike などの RTD 関数を使った条件式を B10 に配置し、When パラメーターにてセル参照します。

セル A10 に配置された Exit 関数の挙動を解説すると、セル A8 の Entry 関数にポジションがある場合で、且つセル B10 が 1 (True) となったら、Exit 関数は Market 注文を発注します。

Exit 関数は常に Entry 注文の建玉を監視し、建玉があれば仕切注文を発注し、なくなったらキャンセルする仕組みとなっています。したがって、一つの Entry 注文に対し複数の Exit 注文が発注されていた場合、いずれかの Exit 注文が約定しポジションが 0 になるとその他の Exit 注文は自動的にキャンセルされます。

# キャンセル注文関数

CancelAll 関数は Trade System や Entry 関数のワーキング注文をキャンセルします。

CancelAll パラメーター	入力例
Formula	A8 (Entry 関数)
Cancel	B14 (条件式が入力されたセルを参照)

CancelAll 関数の Formula パラメーターは、Exit 関数の Formula パラメーターと同様に、対象となる Entry 関数や Trade System を参照します。

このパラメーターはセル参照の入力が必須となり、計算式の入力は不可となります。この例の中では、セル A8 は Entry 関数が入力されているセルです (Entry 関数式 : =Buy(A6,B8,"S&P 買い",A2))。

Cancel パラメーターへは条件式をセル参照する必要があります。条件式が 1 (=True) になると、Entry 関数から発注された注文が全てキャンセルされます。

また、CancelAll 関数を使わずとも、Entry 関数にて注文枚数 (Size) をセル参照で取得している場合は、参照しているセルの数値を 0 に変更すれば、全ての注文がキャンセルされます。

	A	B	C
1	<b>取引システム</b>	<b>On/Off</b>	<b>アカウント</b>
2	取引1	0	PSYukiMogi
3	<b>銘柄</b>		
4	EP		
5	<b>Limit関数</b>	<b>発注価格</b>	
6	Buy EP 5 @ Lmt 2665.25 DAY	2665.25	
7	<b>Buy関数</b>	<b>執行</b>	
8	S&P買い = Buy EP 5 @ Lmt 2665.25 DAY when \$B\$8	0	
9	<b>Exit関数@Market</b>	<b>執行</b>	
10	Market仕切り = Exit S&P買い = Sell 0 @ Mkt DAY when \$B\$10	0	
11	<b>Exit関数@Limit</b>		
12	Limit仕切り = Exit S&P買い = Sell 0 @ Lmt 2638.75 GTC		
13	<b>CancelAll関数</b>	<b>執行</b>	
14	CancelAll(S&P買い;False)	0	

上図 セル A14:CancelAll(Long1;False)のとおり CancellAll 関数のセパレーターは「;」となります。

全 Entry 関数の注文を一括キャンセルしたい場合は、Formula パラメーターのセル参照を Trade System 関数に設定してください。

下図の例ではセル C16 に配置されている Volatility が 0.25 を超えた場合、Trade System 下の全注文がキャンセルされます。

関数の引数

CancelAll

Formula A2 = "取引1"

Cancel B16 = 0

= "CancelAll(取引1;False)"

Cancel all working orders.

Formula Reference to trading system, entry or exit formula: TradeSystem(), Buy(), Sell() or Exit().

数式の結果 = CancelAll(取引1;False)

[この関数のヘルプ\(H\)](#) OK キャンセル

	A	B	C
1	取引システム	On/Off	アカウント
2	取引1	0	PSYukiMogi
3	銘柄		
4	EP		
5	Limit関数	発注価格	
6	Buy EP 5 @ Lmt 2665.25 DAY	2665.25	
7	Buy関数	執行	
8	S&P買い = Buy EP 5 @ Lmt 2665.25 DAY when \$B\$8	0	
9	Exit関数@Market	執行	
10	Market仕切り = Exit S&P買い = Sell 0 @ Mkt DAY when \$B\$10	0	
11	Exit関数@Limit		
12	Limit仕切り = Exit S&P買い = Sell 0 @ Lmt 2638.75 GTC		
13	CancelAll関数	執行	
14	CancelAll(S&P買い;False)	0	
15	CancelAll関数		Volatility
16	CancelAll(取引1;False)	FALSE	0.1247
17			

## OCO の設定

Exit のストラテジーとして、OCO 注文の構築も可能です。利食いの指値と、損切りの逆指値は、Trade System 下の建玉平均値から算出します。

OCO 注文なので、一方の注文が約定したらもう一方の注文は自動的にキャンセルされます。OCO 注文の構築方法を下記に説明します。

	A	B	C	D
1	システム名	On/Off	執行条件	アカウント名
2	OCO注文 - Running	1	DAY	SIMYukiMogi
3	銘柄	ティックサイズ	直近終値	発注数
4	EPH21	0.25	3687.00	5
5		建玉	約定平均値	
6		0	0	
7	新規発注@ロング	発注		
8	ロングエントリー = Buy EPH21 5 @ Lmt 3687 DAY when \$B\$8	0		
9	利食い	ターゲットティック	ターゲット価格	
10	利食い = Exit ロングエントリー = Sell EPH21 0 @ Lmt 0 DAY	10	0.00	
11	損切り	ストップロスティック	ストップロス価格	
12	損切り = Exit ロングエントリー = Sell 0 @ Stp 0 DAY	10	0.00	

まず、フィールドをいくつか追加します。上図のとおりセル A4 にはシンボルを、そしてセル B4 にはティックサイズを追加します。ティックサイズの取得には下記 RTD 関数を用います。

```
=RTD("cqg.rtd",,"ContractData",$A$4,"TickSize",,"T")
```

ターゲット価格とストップロス価格は “新規約定値段 +/- ティック数 × ティックサイズ” で計算するため、ティックサイズのフィールドが必要となります。

また、セル C4 は一分足チャートの直近終値を RTD 関数にて取り込んでおり、この値が Entry 注文の指値となります。もし現在値などリアルタイムで更新される値段を参照すると、指値もそれに応じて変更されます。セル C4 への RTD 関数式は以下のとおりです。

```
=RTD("cqg.rtd",,"StudyData",$A$4,"Bar",,"Close","1",-1,"all",,"False","T")
```

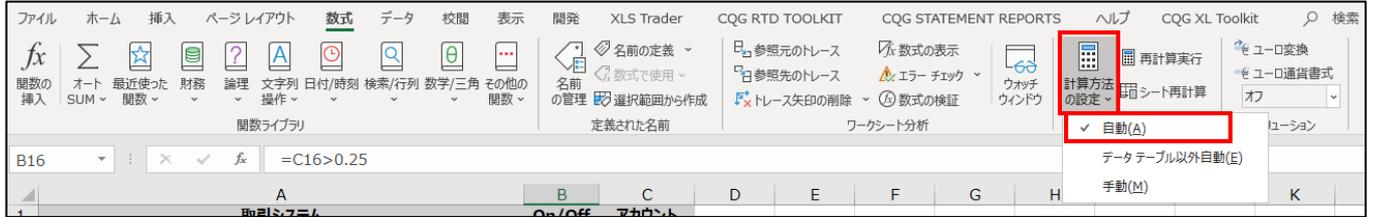
補足となりますが、下記リンクからダウンロードするエクセルファイル内に、RTD 関数で取得できるチャートタイプとスタディが掲載されています。

<http://news.cqg.com/workspaces/main/2012/11/cqg-to-rtd-excel.html>

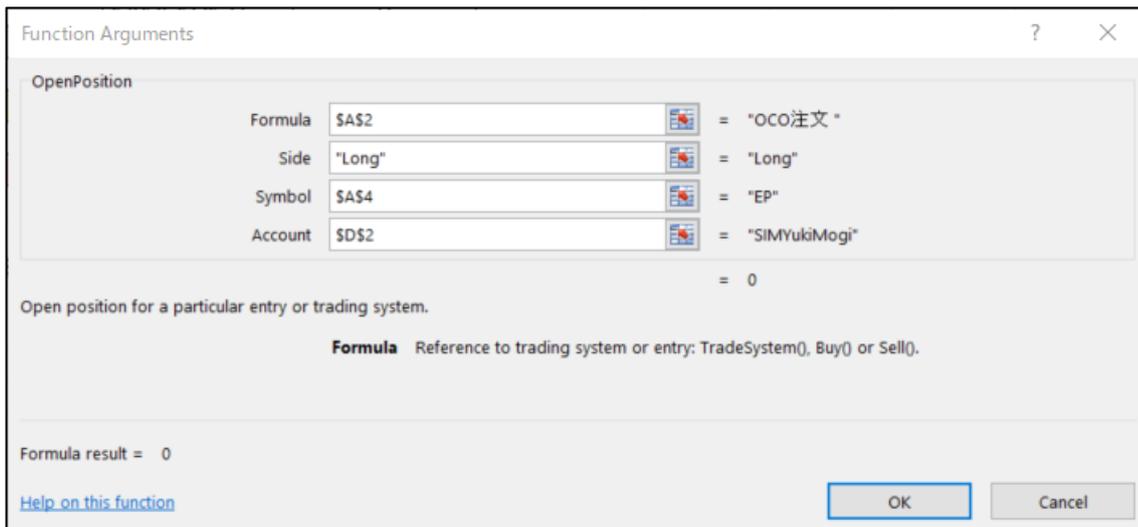
加えて、RTD 関数式の使用に際し、Excel RTD Throttle の調整を推奨します。詳細は下記リンクの 3 ページ目を確認してください。

[https://jp.cqg.com/sites/default/files/files/TroubleShooting\\_RTD.pdf](https://jp.cqg.com/sites/default/files/files/TroubleShooting_RTD.pdf)

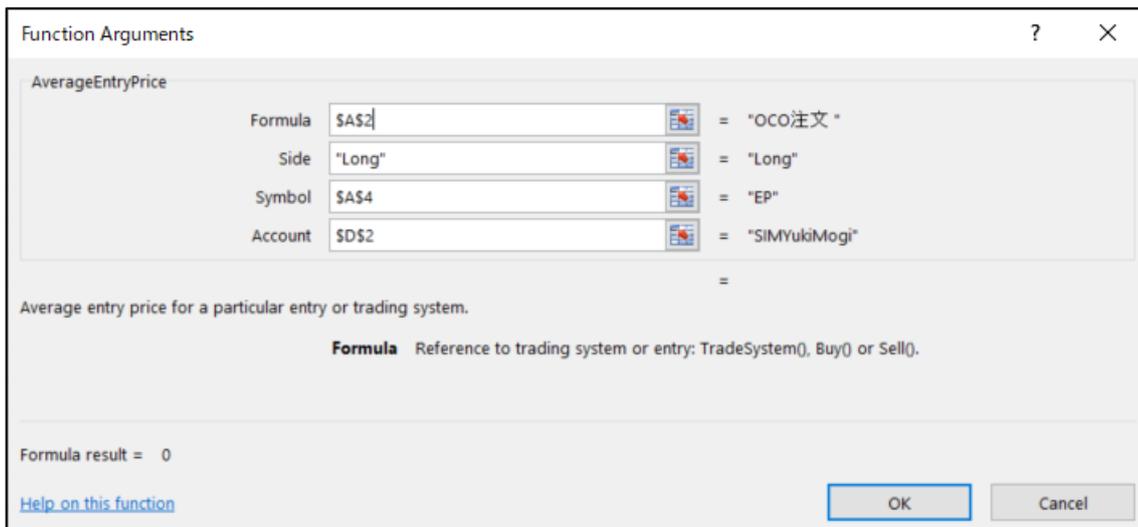
また、エクセルが自動的に再計算を実行する設定となっているか確認してください。



Target および Stop 注文の発注枚数を取得するために、OpenPosition 関数をセル B6 に設定します。OpenPosition 関数は Trade System 下の建玉枚数を返します。下図の例にて返す数値は、セル A8 の Entry (Buy) 関数にて約定した買い玉の枚数です。



Target と Stop 注文の値段を取得するために約定平均値が必要です。このため、セル C6 に AverageEntryPrice 関数を配置します。ウィザードは下図のとおりです。



AverageEntryPrice 関数で使用するパラメーターは、OpenPosition 関数と全く同じです。この平均値は Target および Stop 値段を計算するうえでの基準値段として使われます。

セル C10 に配置するターゲット価格の計算式は下記のとおりです。

$$=C6+(B4*B10)$$

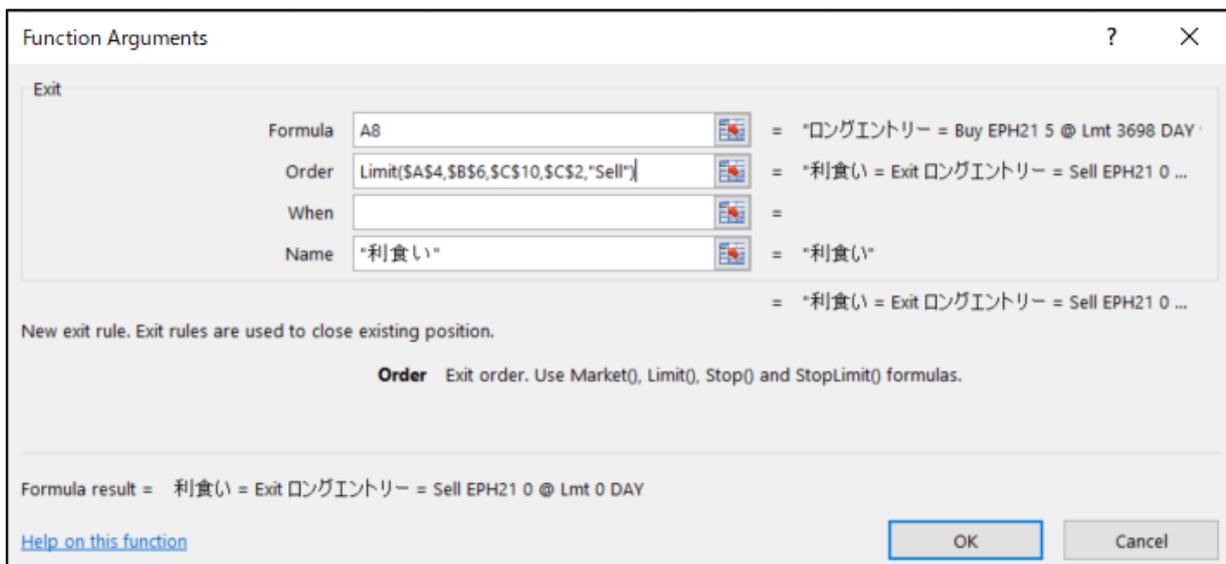
=平均値 + (ティック数×ティックサイズ)

また、セル C12 に配置するストップロス価格の計算式は下記のとおりです。

$$=C6-(B4*B12)$$

=平均値 - (ティック数×ティックサイズ)

Target 値段と Stop 値段の配置が完了したので、利食い注文用の Exit 関数をセル A10 に入力します。Order パラメーター内の入力値はセル参照ではなく、Limit 関数が直接入力されていますが、その中の symbol (\$A\$4)、size (\$B\$6)、price(\$C\$10)、duration(\$C\$2) はセル参照となっています。また、When パラメーターはブランクです。



Function Arguments

Exit

Formula	A8	= "ロングエントリー = Buy EPH21 5 @ Lmt 3698 DAY"
Order	Limit(\$A\$4,\$B\$6,\$C\$10,\$C\$2,\"Sell\")	= "利食い = Exit ロングエントリー = Sell EPH21 0 ..."
When		=
Name	"利食い"	= "利食い"

New exit rule. Exit rules are used to close existing position.

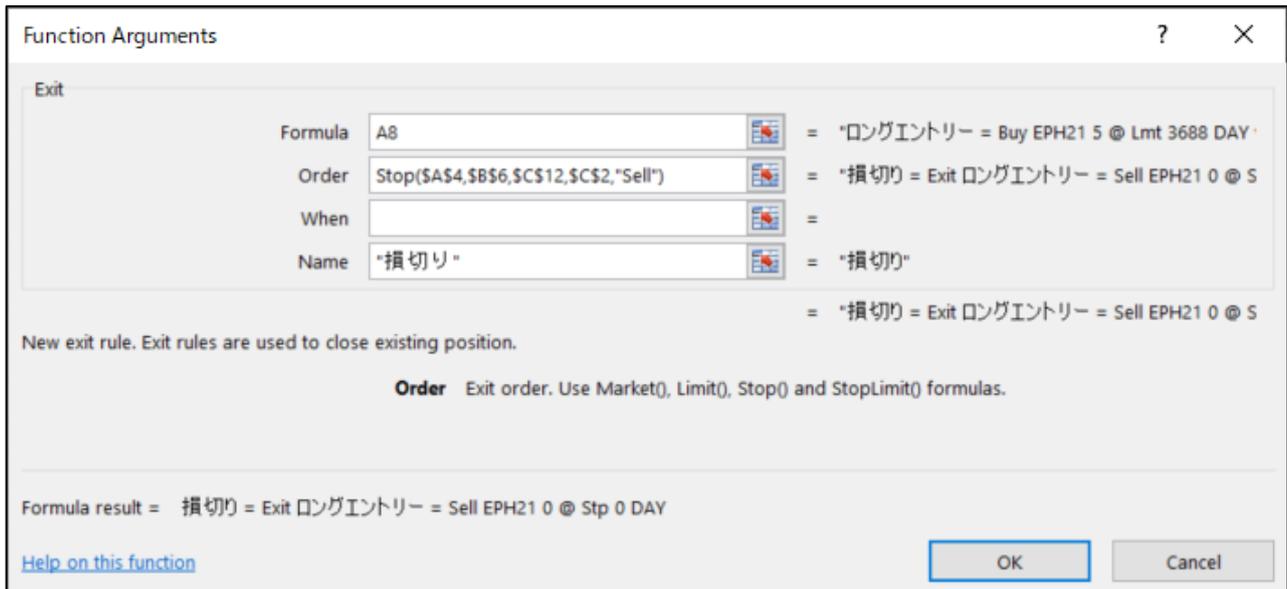
**Order** Exit order. Use Market(), Limit(), Stop() and StopLimit() formulas.

Formula result = 利食い = Exit ロングエントリー = Sell EPH21 0 @ Lmt 0 DAY

[Help on this function](#) [OK] [Cancel]

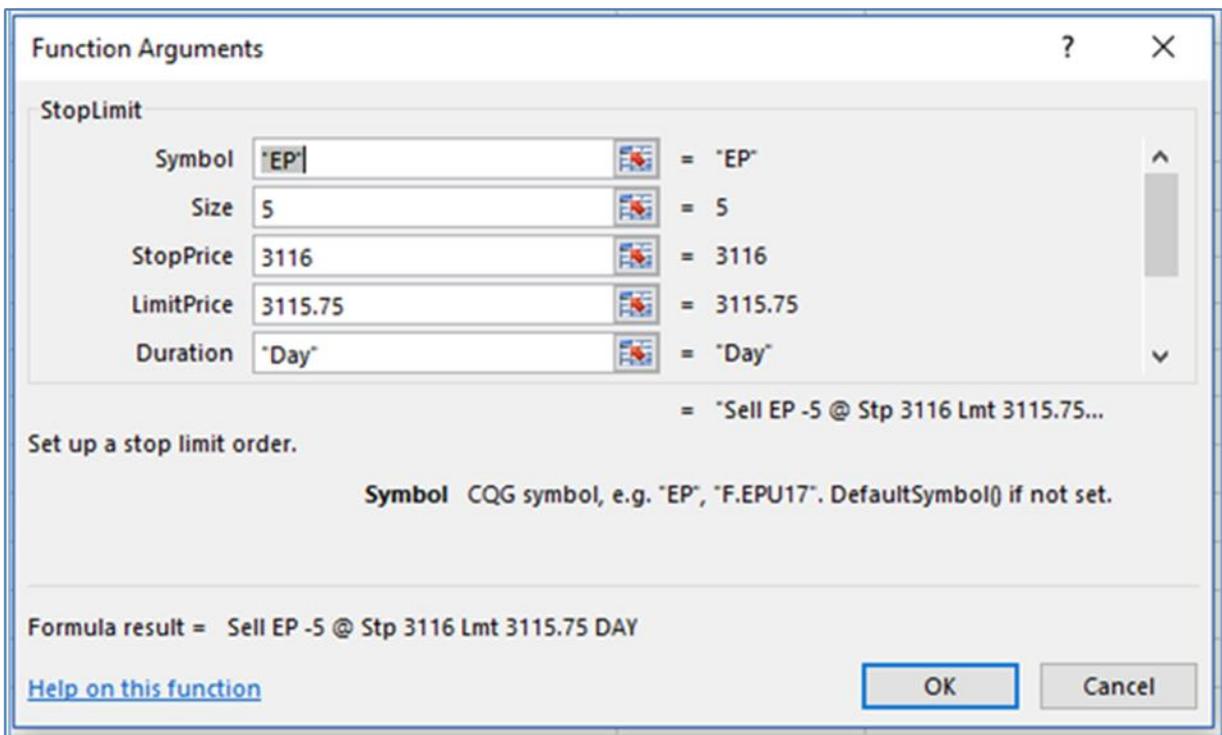
セル A12 には Stop 注文用の Exit 関数を入力します。

Target 注文と同様に、Stop 注文の Order パラメーターもセル参照を使わず Stop 関数を入力しています。ただし Stop 関数式にはセル参照を使っています。また、When パラメーターはブランクとなります。



上の例では Stop 注文を使っていますが、Stop Limit 注文も下図のとおり利用可能です。

StopLimit 関数は、LimitPrice パラメーターにトリガー後の指値も入力する必要があります。



下図の例では、検証のために Entry 関数（ロングエントリー）の When パラメーターであるセル B8 に手動で 1 を入れたことにより、注文 : Buy EP 5 @ 3687.75 が約定し、買い玉を 5 枚保有しているため、Exit 注文である Target 注文と Stop 注文が発注されています。

指値および逆指値は、約定値の 10 ティック上(利食い注文)と 10 ティック下(損切り注文)となります。

	A	B	C	D
1	システム名	On/Off	執行条件	アカウント名
2	OCO注文 - Running	1	DAY	SIMYukiMogi
3	銘柄	ティックサイズ	直近終値	発注数
4	EP	0.25	3687.75	5
5		建玉	建玉平均値	
6		5	3687.75	
7	新規発注@ロング	発注		
8	ロングエントリー = Buy EP 5 @ Lmt 3687.75 DAY when \$B\$8	1		
9	利食い	ターゲットティック	ターゲット価格	
10	利食い = Exit ロングエントリー = Sell -5 @ Lmt 3690.25 DAY	10	3690.25	
11	損切り	ストップロスティック	ストップロス価格	
12	損切り = Exit ロングエントリー = Sell -5 @ Stp 3685.25 DAY	10	3685.25	

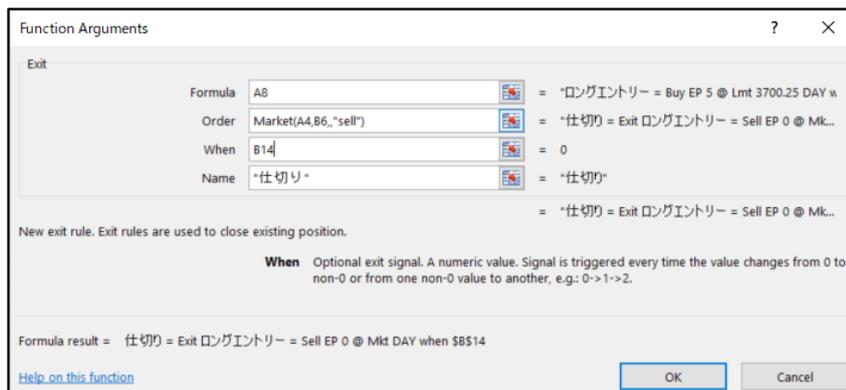
約定値が複数となったことにより、約定平均値が呼び値の単位を割り込み、取引可能な値段ではなくなった場合、C10 のターゲット価格と C12 のストップロス価格には約定平均値より計算された取引不可値段がそのまま表示されます。しかし Trade System は自動的にターゲットおよびストップロス価格を取引可能な値段に調整します。

追加で、成行注文の Exit 関数をセル A14 に構築します。

	A	B	C	D
1	システム名	On/Off	執行条件	アカウント名
2	OCO注文 - Running	1	DAY	SIMYukiMogi
3	銘柄	チックサイズ	直近終値	発注数
4	EP	0.25	3704.75	5
5		建玉	約定平均値	
6		0	0	
7	新規発注@ロング	発注		
8	ロングエントリー = Buy EP 5 @ Lmt 3704.75 DAY when \$B\$8	0		
9	利食い	ターゲットチック	ターゲット価格	
10	利食い = Exit ロングエントリー = Sell 0 @ Lmt 0 DAY	10	0.00	
11	損切り	ストップロスチック	ストップロス価格	
12	損切り = Exit ロングエントリー = Sell 0 @ Stp 0 DAY	10	0.00	
13				
14	仕切り = Exit ロングエントリー = Sell EP 0 @ Mkt DAY when \$B\$14	0		

上図では、When パラメーターがセル参照しているセル B14 は現在 0 を返しています。もしセル B14 が 1 を返したら成行注文が発注され建玉を仕切ります。

仕切注文の約定により、セル B6（OpenPosition 関数）が 0 になると、利食いと損切り注文はキャンセルされます。



# XLS TRADER で取引

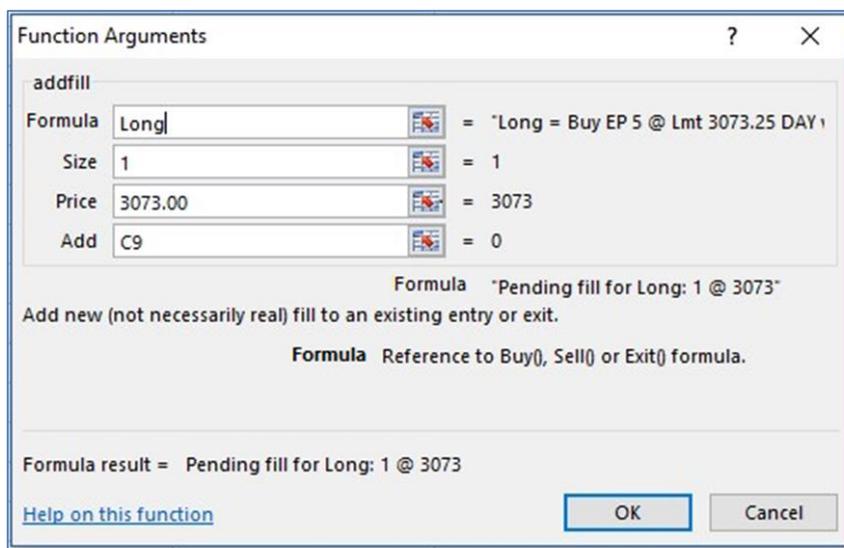
XLS TRADER より発注された約定注文は CQG IC 上に表示されます。しかしその逆、つまり CQG IC より発注された約定注文は XLS TRADER に認識されません。

CQG IC より発注された約定注文を XLS TRADER に追加する場合など、XLS TRADER で取引していない約定を追加するには AddFill 関数を使用します。

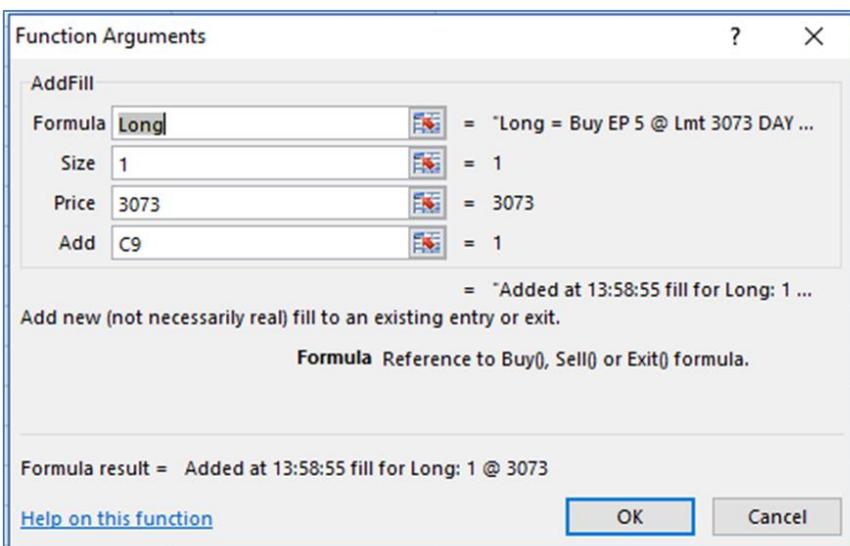
下図のとおり、まず Formula パラメーターに買い／売り、および新規／落ちを設定するため、対象となる Entry または Exit 関数をセル参照します。

次に、Size（売りは負数、買いは正数）および Price パラメーターを入力します。

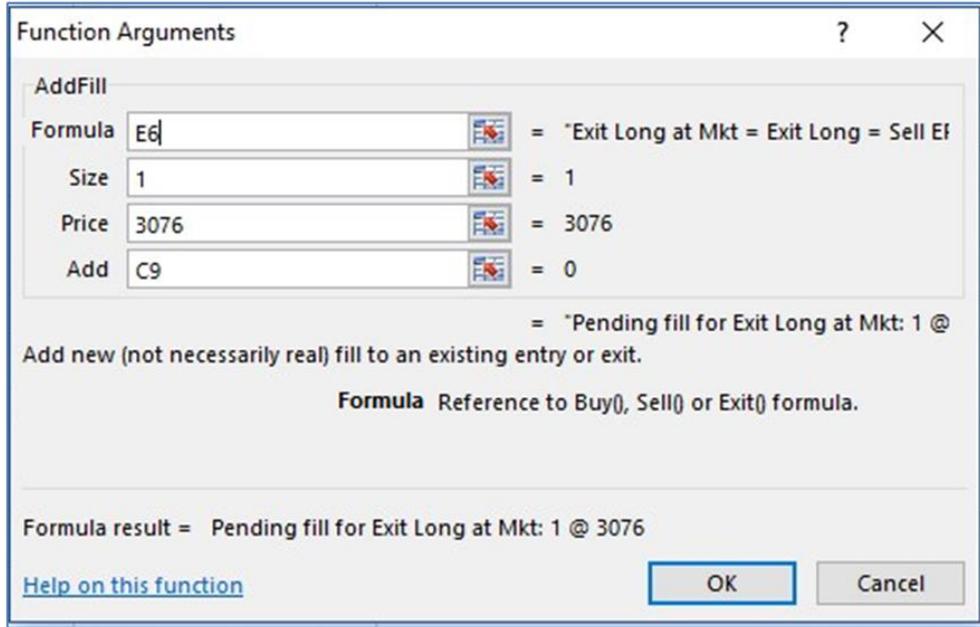
Add パラメーターのセル参照先 C9 は 0 を返しているため、“Pending fill for long: 1 @ 3073.”と、約定追加を待機する旨のメッセージが表示されています。



セル C9 に 1 を入力すると、待機状態が解除され約定が追加されます。セルには“Added at 13:58:55 fill for Long: 1 @ 3073.”と表示されます。これにより OpenPosition 関数や Trade System がこの約定を認識し、ポジションを加減算します。



ポジションを減らすためには、Formula パラメーターに Exit 関数をセル参照します。



## Execution Instructions (スマートオーダータイプ)

Limit、Stop、Market 関数を説明した際に触れた、ExecInstr パラメーター／関数について詳しく説明します。Limit 関数などの ExecInstr パラメーターに下記実行タイプをオプションで加えることができます。ただし、これらの注文タイプが取引所にて許可されている場合のみ、発注可能となります。

- =AllOrNothing()
- =CancelIfPriceIsInvalid()
- =ExecInstr()
- =Funari()
- =Iceberg()
- =MarketIfTouch()
- =OpenCloseInstruction()
- =QuantityTriggeredStop()
- =Trailing()

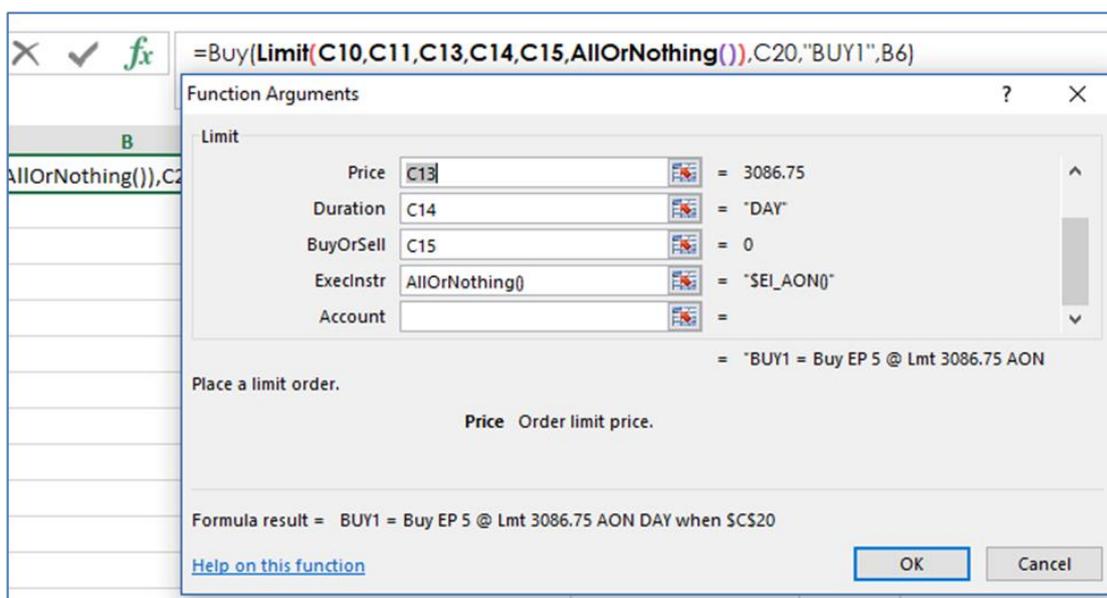
上記関数は、ExecInstr パラメーターに直接入力するか、セル参照します。

### All or Nothing 関数

=AllOrNothing()

全量が一度に約定される数量がマーケットになれば、全量キャンセルされます。この実行タイプを許可する CME の商品については下記リンクを参照してください。

<https://www.cmegroup.com/clearing/trading-practices/all-or-none.html>



## Cancel If Price Is Invalid 関数

= CancelIfPriceIsInvalid()

この関数を追加すると、発注時に注文値段を参照するセルがブランクや#N/A など、有効な値段ではなかった場合に注文をキャンセルします。

Function Arguments

Limit

Price	B5	=	3115.5
Duration	E4	=	"DAY"
BuyOrSell		=	
ExecInstr	CancelIfPriceIsInvalid()	=	"SEI_XIP0"
Account		=	

Set up a limit order.

Price Limit price.

Formula result =

[Help on this function](#) OK Cancel

## Execution Instructions 関数

=ExecInstr()

この関数は複数の ExecInstr 関数を設定したい場合に使用します。パラメーターに下図のとおり関数を直接入力するか、もしくはセル参照も可能です。

Function Arguments

ExecInstr

Instr1	Trailing(5)	=	"SEI_TRAIL(5;trade)"
Instr2	CancelIfPriceIsInvalid()	=	"SEI_XIP0"
Instr3		=	
Instr4		=	
Instr5		=	

Combine execution instructions: AllOrNothing(), Funari(), Iceberg(), MarketIfTouch(), OpenCloseInstruction(), QuantityTriggeredStop(), Trailing().

Instr1 Instruction 1, any of AllOrNothing(), Funari(), Iceberg(), MarketIfTouch(), OpenCloseInstruction(), QuantityTriggeredStop(), Trailing().

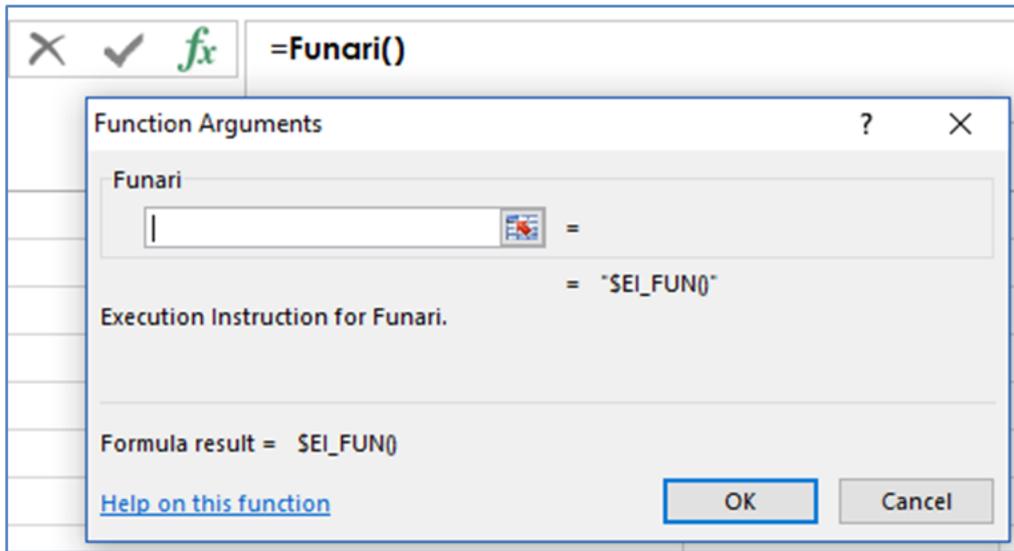
Formula result =

[Help on this function](#) OK Cancel

## Funari (不成) 関数

=Funari()

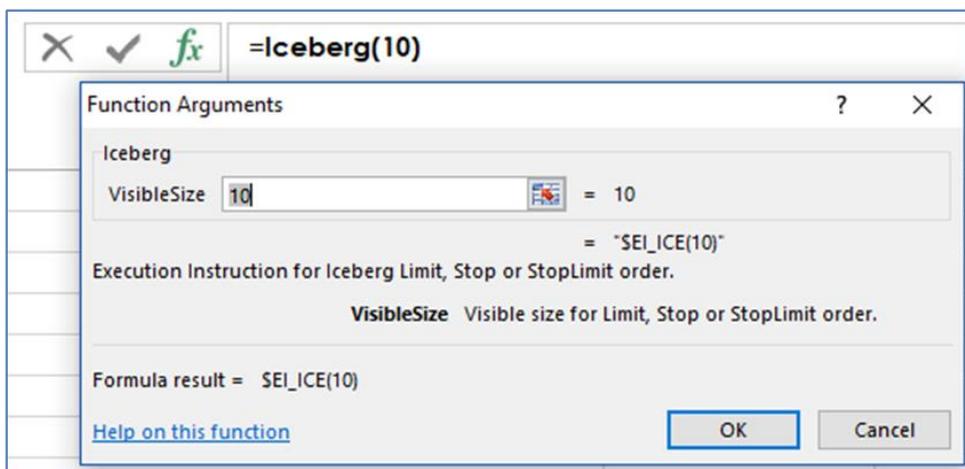
セッション終了直前に注文が未約定、または部分約定だった場合に、残数を **Market** 注文で発注し全量成立させます。この実行タイプが許可されている取引所及び商品はごく一部となります。この実行タイプを発注するには、CQG IC の Smart Order Preferences で Funari を有効にしてください。



## Iceberg 関数

=Iceberg(10)

合計枚数に達するまで小口枚数での発注を繰り返す注文タイプです。小口注文枚数を下図のとおり設定する必要があります。この実行タイプを発注するには、CQG IC の Smart Order Preferences で Iceberg を有効にしてください。



## Market if Touched 関数

=MarketIfTouch()

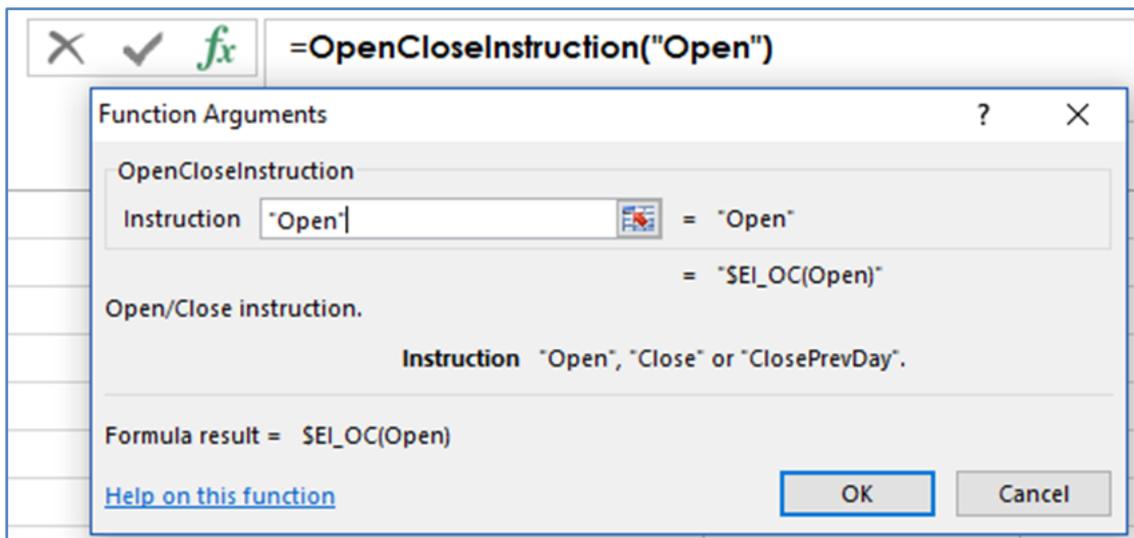
指値注文としてワークしますが、約定圏内に達した段階で **Market** 注文となります。現在この注文タイプを許可している取引所はありません。



## Open Close Instruction 関数

=OpenCloseInstruction("Open")

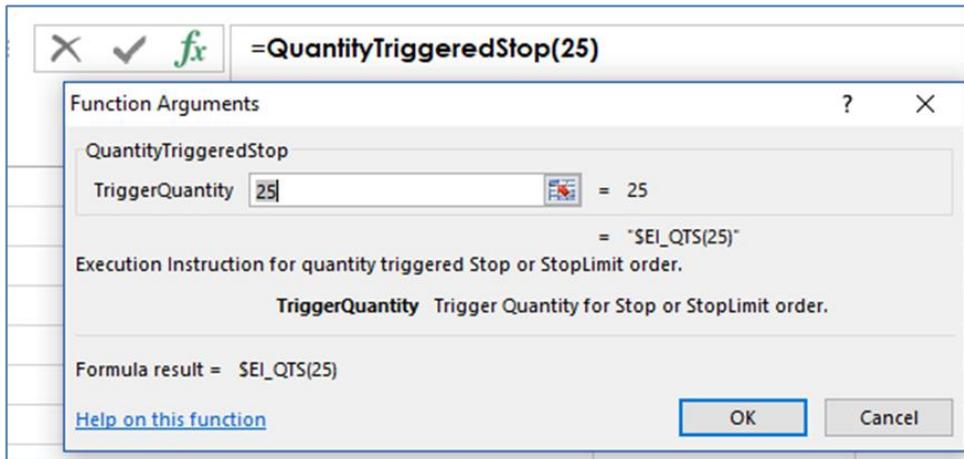
中国の取引所への注文は、新規／落ちを発注時に宣言する必要があります。この関数で **Open**, **Close Today** or **Close Yesterday** を設定することができます。



## Quantity Triggered Stop 関数

=QuantityTriggeredStop(25)

反対側（売り注文なら買い気配／買い注文なら売り気配）の枚数が設定数より下回った場合にストップ注文が執行されます。基準となる気配枚数を下図のとおり設定する必要があります。この執行タイプを発注するには、CQG IC の Smart Order Preferences で Quantity Triggered Stop を有効にしてください。



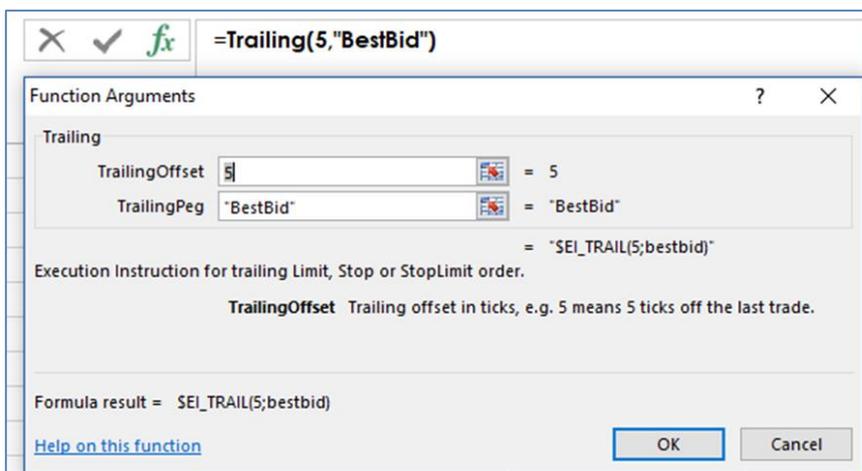
## Trailing 関数

=Trailing(5,"BestBid")

ベストビッド／アスクまたは直近値を自動的に追跡する注文タイプです。

**Trailing offset** パラメーターには、何ティック離れて追跡するのか、その設定値を入力する必要があります。

**Trailing Peg** パラメーターはベストビッド／アスクまたは直近値のいずれかを入力します。デフォルトは直近値が設定されています。この執行タイプを発注するには、CQG IC の Smart Order Preferences で Trailing を有効にしてください。



## Trade System 関連の関数

ここでは、下記 Trade System 関連の関数について説明します。

なお、DefaultAccount 関数については[アカウントの設定](#)にて既に触れているため、ここでは割愛します。

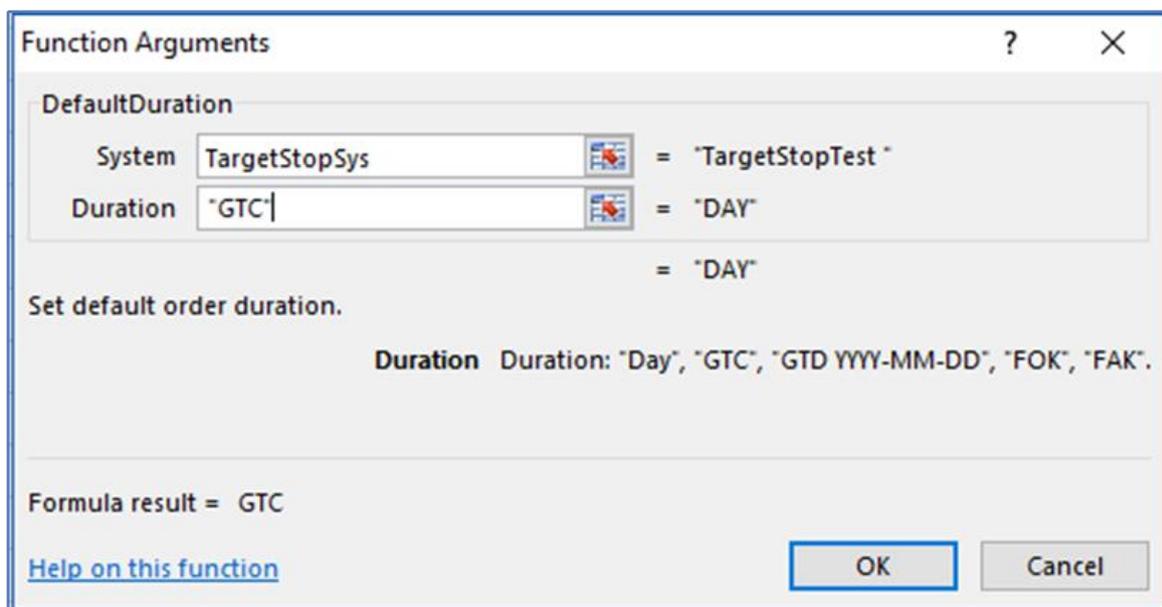
<Trade System 関連の関数>

- =DefaultAccount()
- =DefaultDuration()
- =DefaultSymbol()
- =DisableNewEntries()
- =IsReversalSystem()
- =MaxPosition()
- =MaxPyramid()
- =TradeSystem()

### Default Duration 関数（デフォルトの有効期限を設定）

Trade System 下の全オーダー関数に一括設定される、デフォルトの有効期限を設定できます。この関数を使用しない場合、デフォルトの有効期限は Day となります。

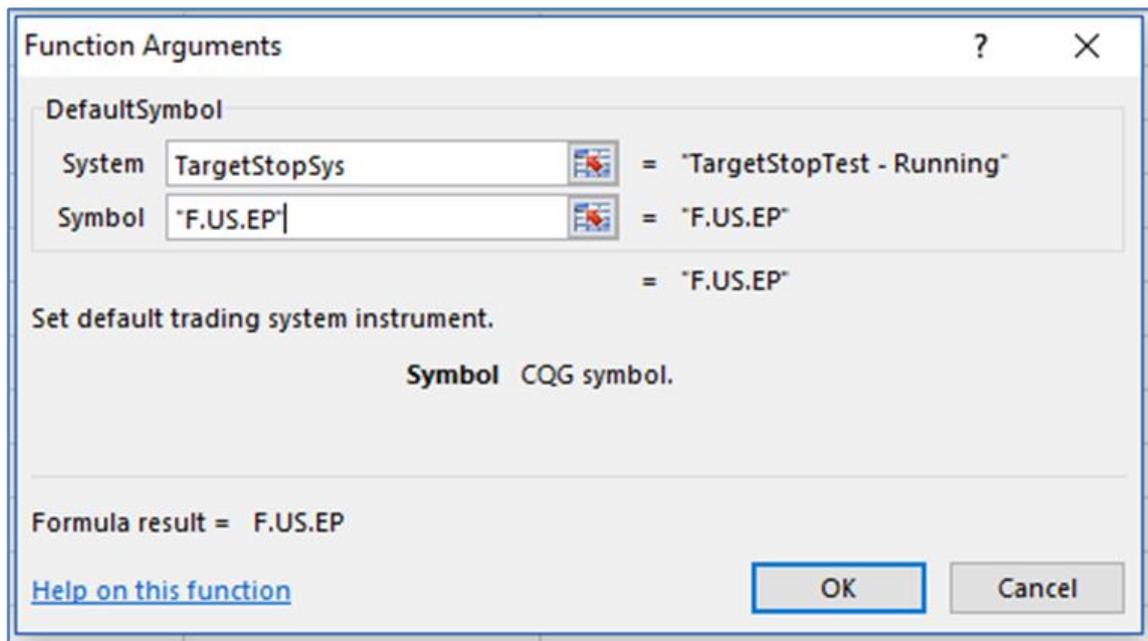
=DefaultDuration(TargetStopSys,"GTC")



## Default Symbol 関数（デフォルトのシンボルを設定）

Trade System 下の全関数に一括設定される、デフォルトのシンボルを設定できます。この設定後、各々の関数式において Symbol パラメーターをブランクにすると、デフォルトのシンボルが適用されます。

=DefaultSymbol(TargetStopSys,"F.US.EP")



## Disable New Entries 関数（新規発注を停止）

Trading System の新規発注を止めたい場合はこの関数を使用します。

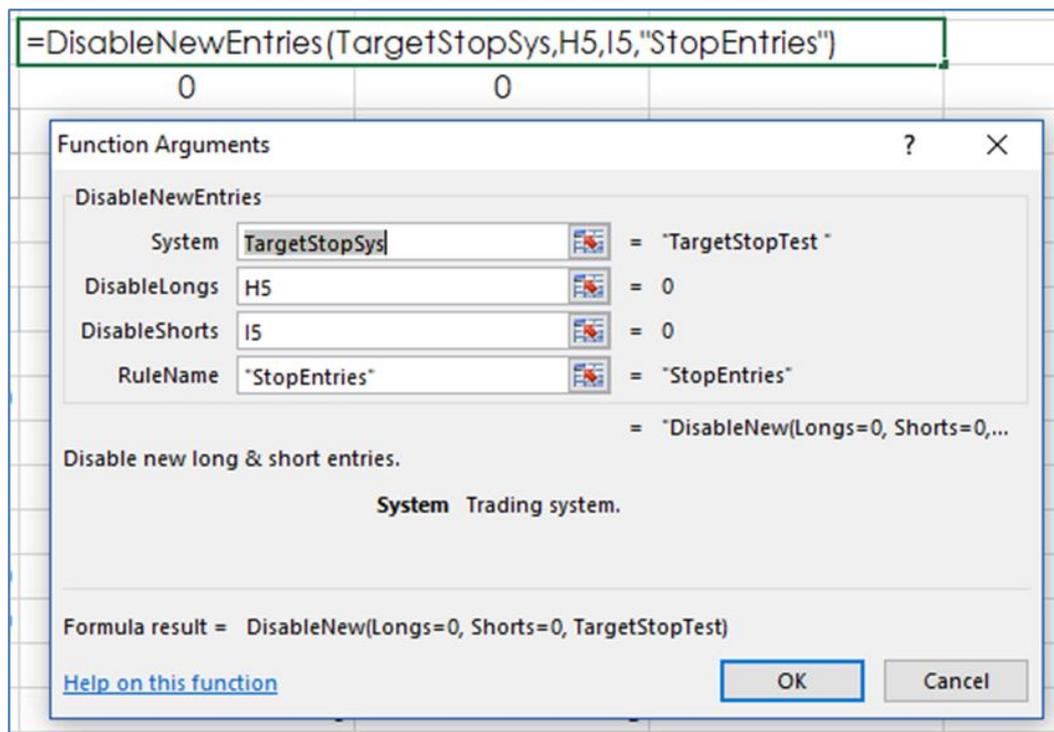
下図では、DisableLongs パラメーターにセル H5 が、DisableShorts パラメーターにセル I5 がそれぞれセル参照されています。そして計算結果で 0 を返していることから分かりますが、どちらのセルにも 0 が入力されています。

セル H5 に 1 を入力すると、Trade System 下の全ての新規 Buy 関数から発注されている注文がキャンセルされ、新規発注が止まります。

同じく I5 に 1 を入力すると、Trade System 下の全ての新規 Sell 関数から発注されている注文がキャンセルされ、新規発注を止めます。

Rule name パラメーターは任意の名前を設定します。Rule name をブランクに設定した場合には自動的に名前が生成されます。

```
=DisableNewEntries(TargetStopSys,H5,I5,"StopEntries")
```



## Is Reversal System 関数（ドテン設定）

ポジションを仕切ると同時にドテンしたい場合に、この関数を使用します。なお、この関数は下表のとおり Trade System が Entry 関数のみで構成されていた場合に機能します。Exit 関数にて建玉を仕切ってもドテンしません。

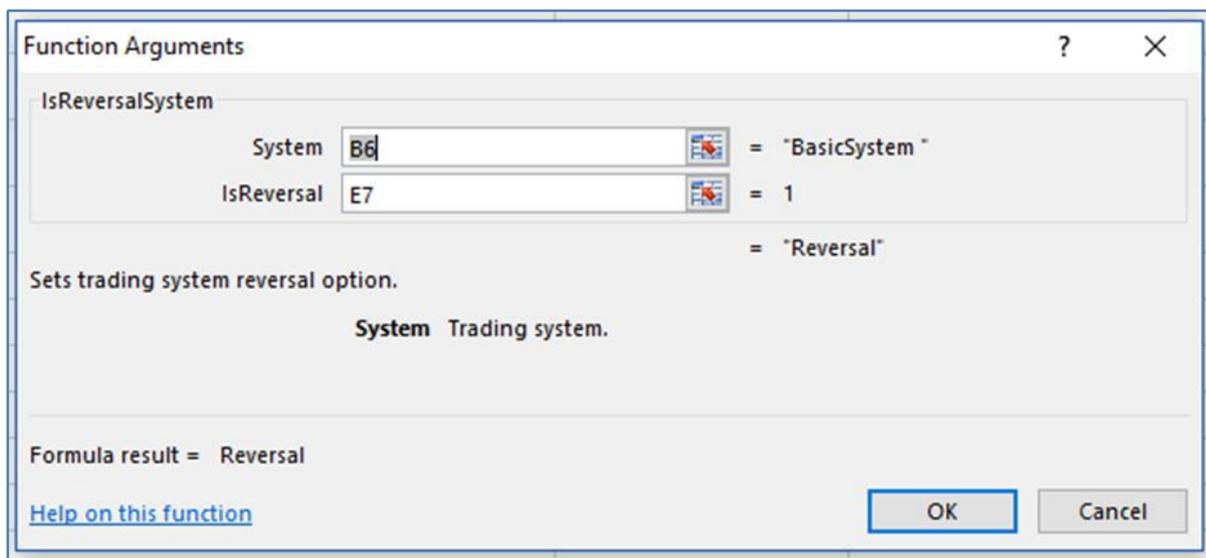
Name	On/Off
Reversal	0
BUY1 = Buy EPZ20 1 @ Mkt GTC when \$B\$6	0
Sell1 = Sell EPZ20 -1 @ Mkt GTC when \$B\$9	0

下図のウィザードでは System パラメーターには B6 がセル参照されており、Trade System 名 “BasicSystem” を返しています。

IsReversal パラメーターには E7 がセル参照されており、1 を返しています。この状態は Reversal が有効になっているため、ポジションを仕切ると自動的にドテンします。

ドテンを止めるためには E7 を 0 にする必要があります。また、この関数式のパラメーターを変更するためには一度 Trade System を止めてパラメーターを変更し、再度 Trade System を稼働させる必要があります。

=IsReversalSystem(B6,E7)

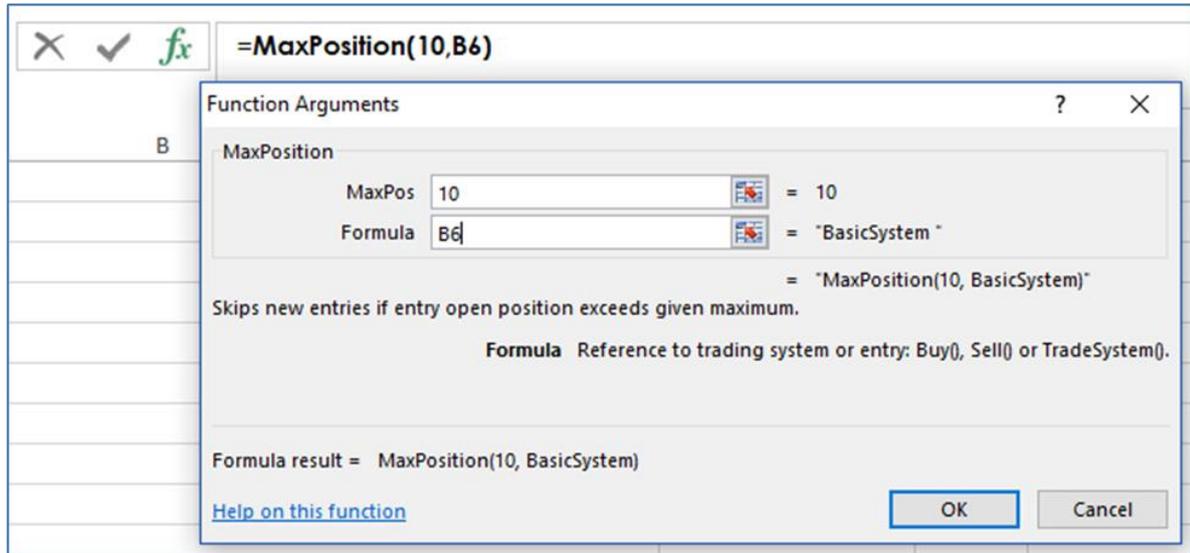


## Maximum Position 関数（最大建玉枚数の設定）

Trade System もしくは Entry 関数における最大建玉枚数を設定できます。

下図では、Trade System “BasicSystem” において、建玉枚数が 10 枚に達した時点で、それ以降の新規注文は止まります。

=MaxPosition(10,B6)

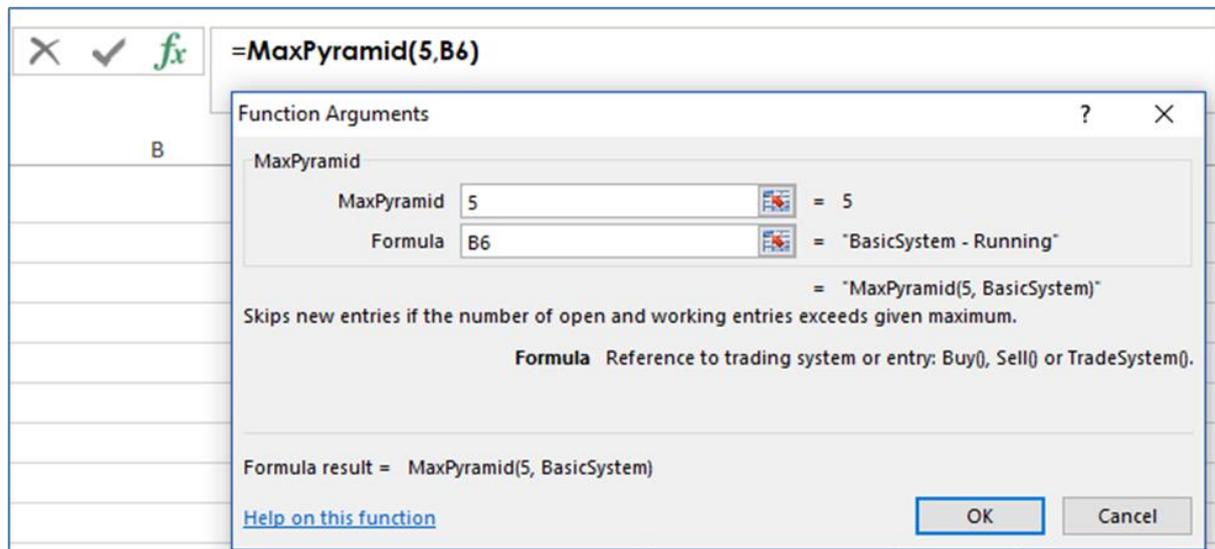


## Maximum Pyramid 関数（ピラミッド注文の上限設定）

ピラミッド注文の数量上限を設定することができます。

下図では、Trade System “BasicSystem” において、ピラミッド注文（建玉+ワーキング注文枚数）が 10 枚に達した時点で、それ以降の新規注文は止まります。

=MaxPyramid(5,B6)



## Order 関数

下記が XLS TRADER でカバーしている Order 関数グループです。

- =AddFill()
- =Buy()
- =CancelAll()
- =Exit()
- =Limit()
- =Market()
- =Sell()
- =Stop()
- =StopLimit()

上記関数内の下記パラメーターを変更するためには、**Trade System** を止める必要があります。  
**Trade System** の稼働中に変更しても、再稼働後まで反映されません。

- Symbol
- Account
- Duration
- BuyOrSell
- ExecInstr

一方、下記パラメーターは **Trade System** が稼働中であっても変更が反映されます。

- Size
- Limit および StopLimit 関数の Limit price パラメーター
- Stop および StopLimit 関数の Stop price パラメーター

## 注文追跡、新規取引および取引情報関数

ワーキング注文と約定注文を表示する Order Colletion 関数グループを紹介します。これらの関数では、何番目の注文を表示するかを指定することができます。表示する注文を選択するためには Offset パラメーターを用います。このパラメーターがブランクの場合は直近の注文情報を取得します。下記が Offset パラメーターの仕様になります。

- 1, 2, 3, ... を設定すると、古い順から 1、2、3 番目の注文を取得します。
- 0 を設定すると、直近の注文を取得します。
- -1, -2, -3, ... を設定すると、直近より 1、2、3 個前の注文を取得します。

もし Offset パラメーターが無効な場合、もしくはそのオフセット値に注文がない場合は、Time と Price 関数については #N/A を返し、Size 関数については 0 を返します。Offset パラメーターにはセル参照も使用できます。

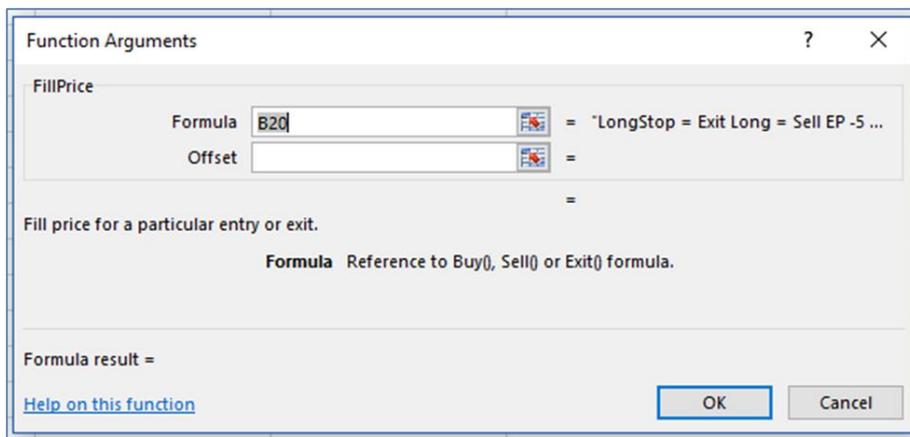
### 約定情報取得関数グループ

約定情報を取得する関数は 3 種類あります。ここで紹介する FillPrice、FillSize、FillTime 関数は、下記の Exit 関数式（セル B20 に配置）の約定情報取得を想定した説明となります。また、表示する約定を選択する場合は Offset パラメーターに数値を入力します。

LongStop = Exit Long = Sell EP -5 @ Stp 3084.75 DAY

#### FillPrice 関数（約定値段取得）

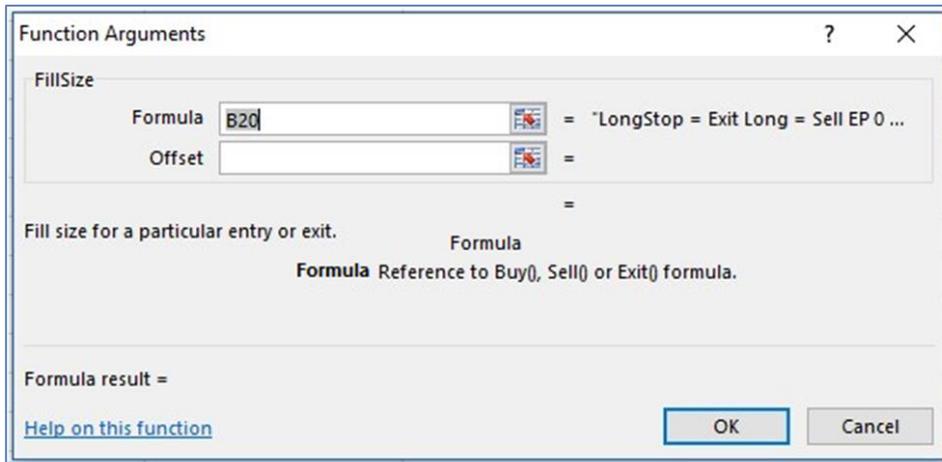
=FillPrice(B20)



## FillSize 関数（約定枚数取得）

=FillSize(B20)

上記式のように Offset がブランクの場合は直近約定情報を返します。FillSize(B20,-1) の場合はそのもう一つ前を、FillSize(B20,-2) はさらにもう一つ前の約定を返します。



単一の注文が複数の約定に分かれている場合は、部分約定内の最終約定のみ取得します。例えば下記約定の場合は一番上の約定だけ取得します。Offset パラメーターで表示する約定を選択している場合はこの限りではありません。

```
11/5/2019 12:46:51 PM 3 @ 3075.25
11/5/2019 12:46:51 PM 1 @ 3075.25
11/5/2019 12:46:51 PM 1 @ 3075.25
```

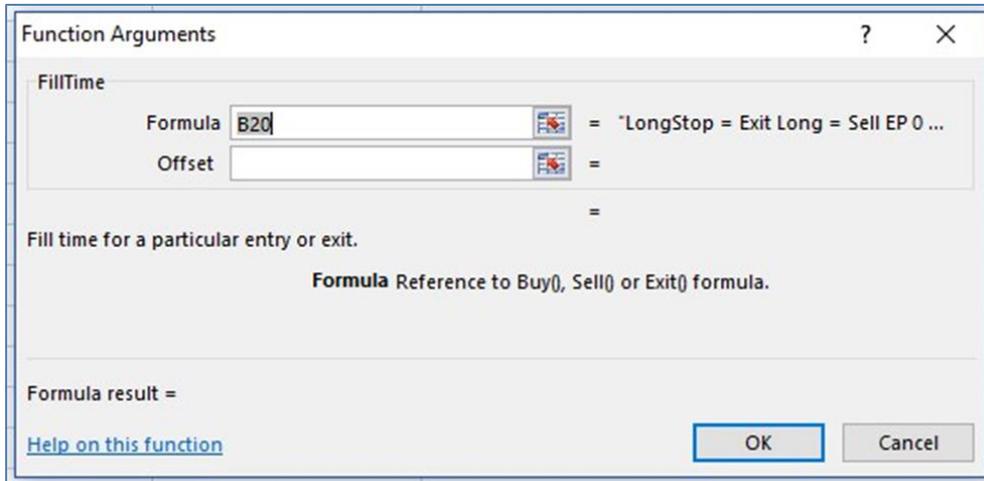
下表のように Fill Price、Fill Size を別々のコラムで分けて、それぞれの関数に Offset 値をセル参照させると、より簡単に表を作ることができます。

=FillSize(Long,I16)			
F	H	I	
Fill Price	Fill Size		
#N/A	0	0	
#N/A	0	-1	
	0	-2	
	0	-3	
	0	-4	
#N/A	=FillSize(Long,I16)	-5	
#N/A	0	-6	
	0	-7	
	0	-8	
#N/A	0	-9	
#N/A	0	-10	

## FillTime 関数（約定時間）

=FillTime(B20)

この関数は約定時間を取得します。正しく表示されない場合は、セルの書式設定にて表示形式が時刻に設定されているか確認してください。



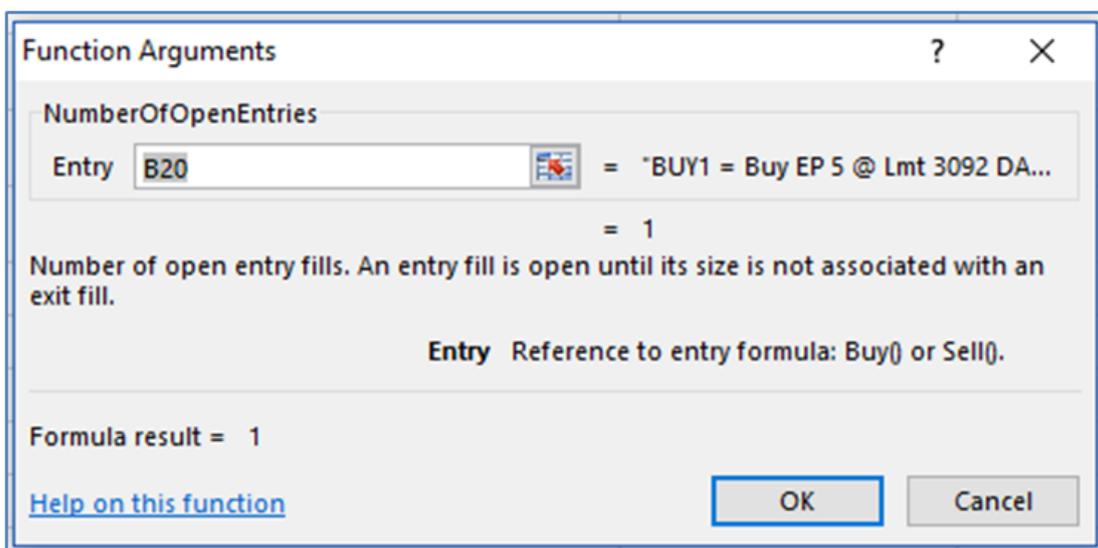
## 新規約定、ワーキング注文、合算の注文件数

新規約定、ワーキング注文、新規とワーキング合算の注文件数を取得する三つの関数式を紹介いたします。下記の Entry 関数式（セル B20 に配置）を例に挙げ、注文件数の取得方法を説明します。

BUY1 = Buy EP 5 @ Lmt 3087.25 DAY when \$C\$20

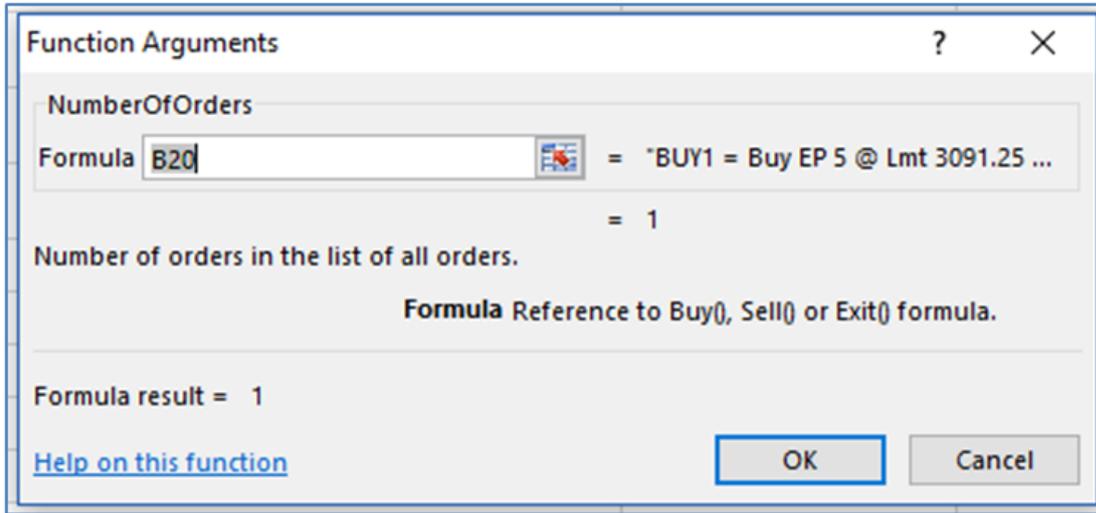
## NumberOfOpenEntries 関数（新規約定件数）

=NumberOfOpenEntries(B20)



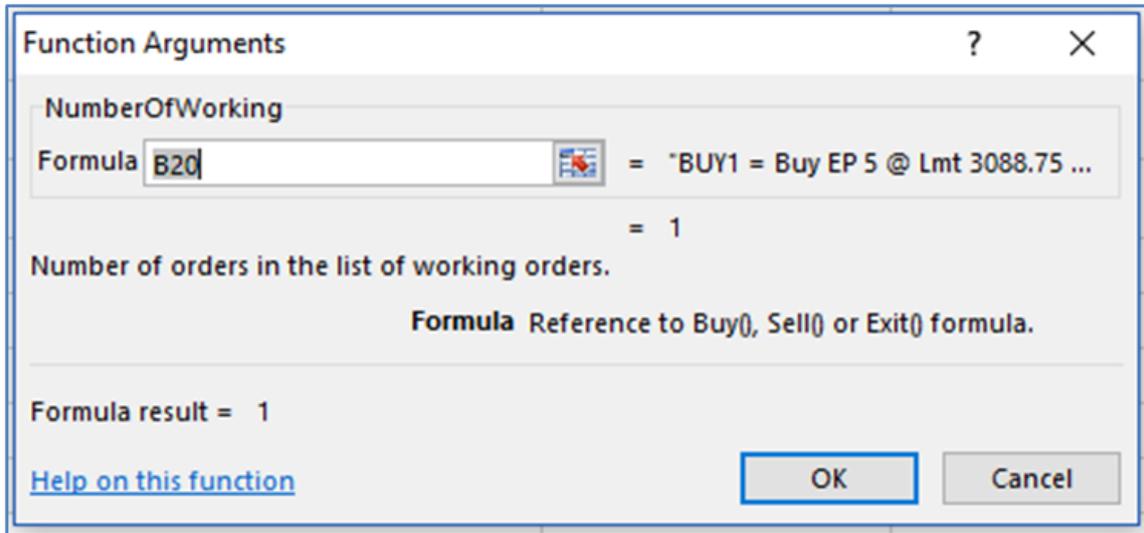
## NumberOfOrders 関数（新規約定およびワーキング注文件数）

=NumberOfOrders(B20)



## NumberOfWorking 関数（ワーキング注文件数）

=NumberOfWorking(B20)



## 新規約定情報取得関数グループ

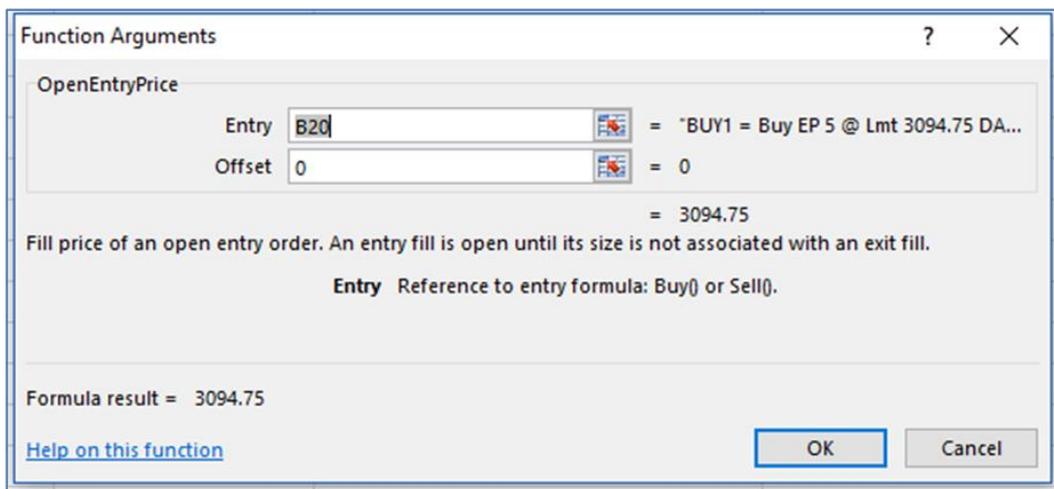
下記 3 つの関数にて新規約定情報（値段、枚数、時間）が取得できます。また **Offset** パラメーターに数値を設定することにより、表示する新規約定情報を選択することができます。

下記の **Entry** 関数式がセル B20 に配置されているという想定の下、新規約定情報の取得方法を説明します。

Long = Buy EP 5 @ Lmt 3094.75 DAY when \$B\$6

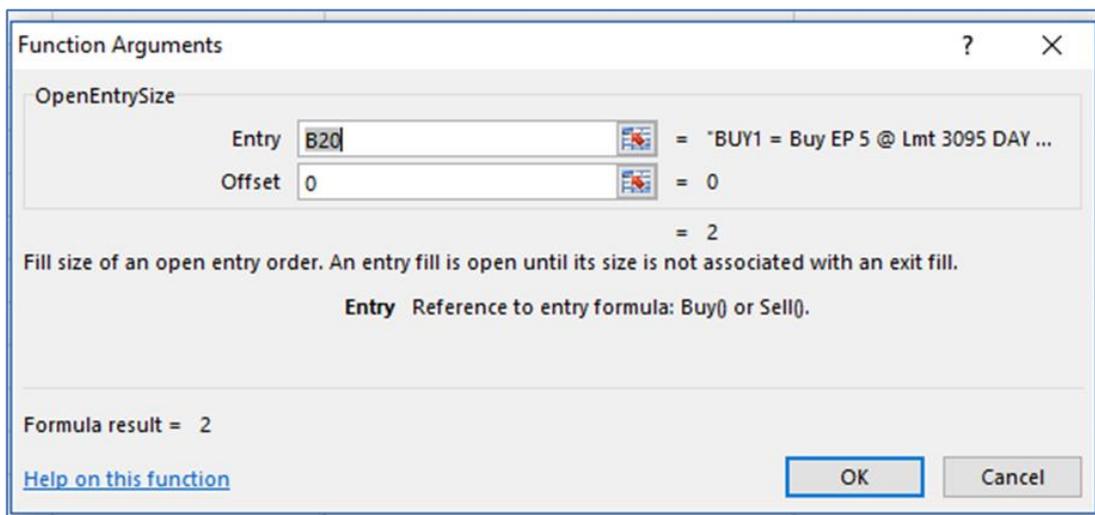
### OpenEntryPrice 関数（新規約定値段）

=OpenEntryPrice()



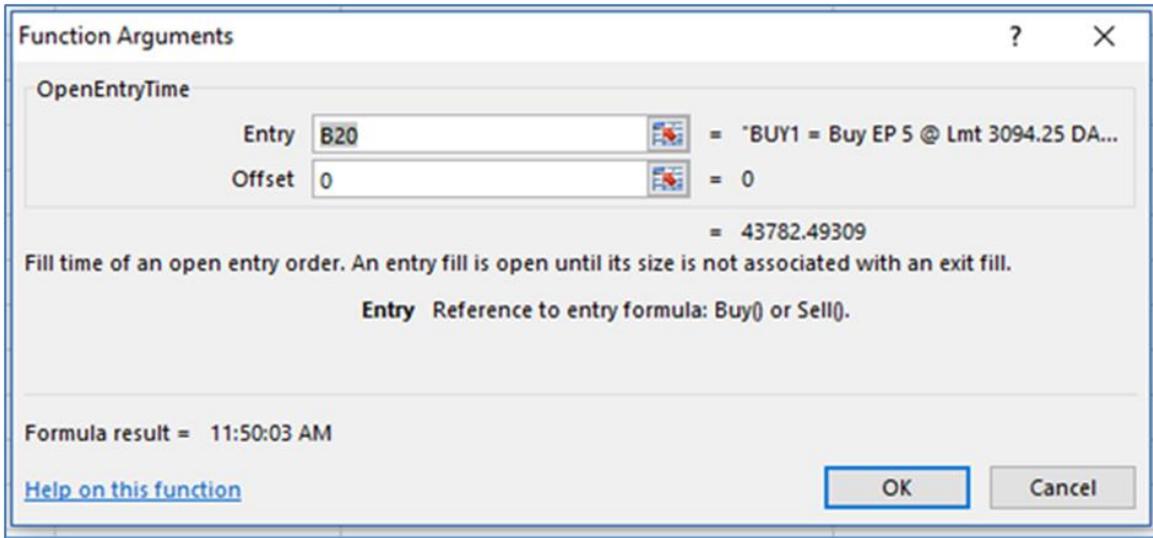
### OpenEntrySize 関数（新規約定枚数）

=OpenEntrySize()



## OpenEntryTime 関数（新規約定時間）

=OpenEntryTime()



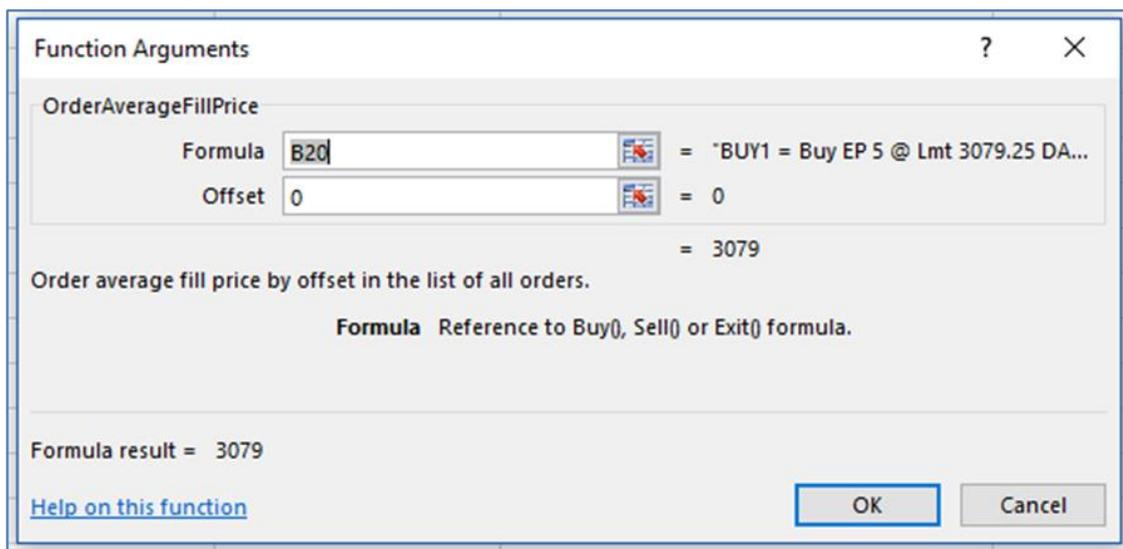
## 注文詳細取得関数グループ

下記の関数は注文の詳細を取得します。これらの関数は Entry（Buy、Sell）および Exit 関数の両方に対応しているため、どちらの情報も取得できます。

## Order Average Filled Price 関数（約定平均値）

Order Average Filled Price 関数は指定した Entry/Exit 関数の平均約定値を取得します。

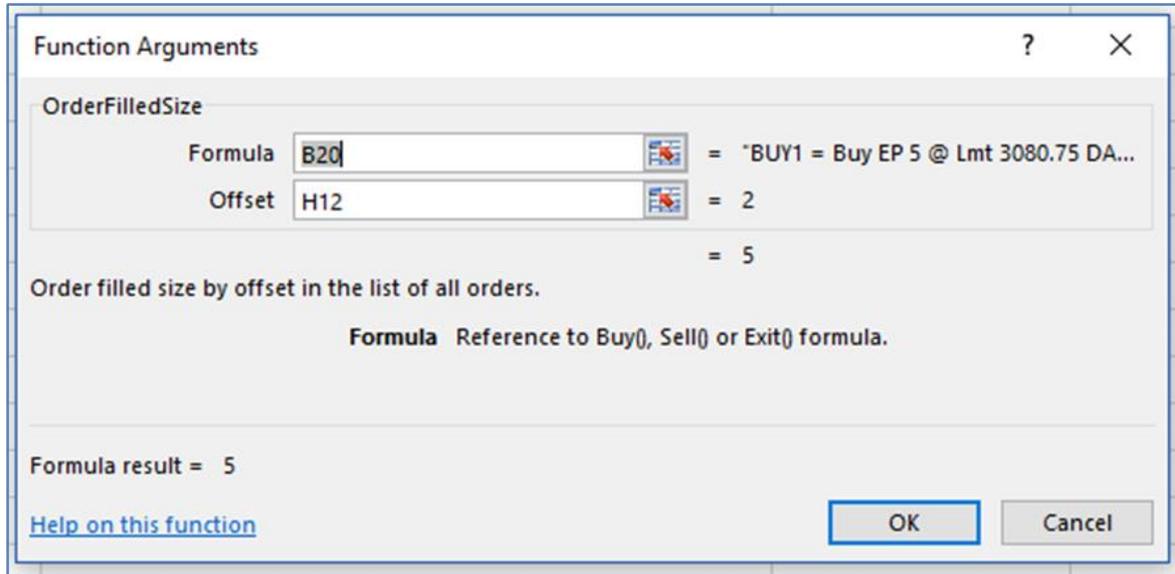
=OrderAverageFilledPrice()



## Order Filled Size 関数（約定枚数）

Order Filled Size 関数は約定枚数を取得します。

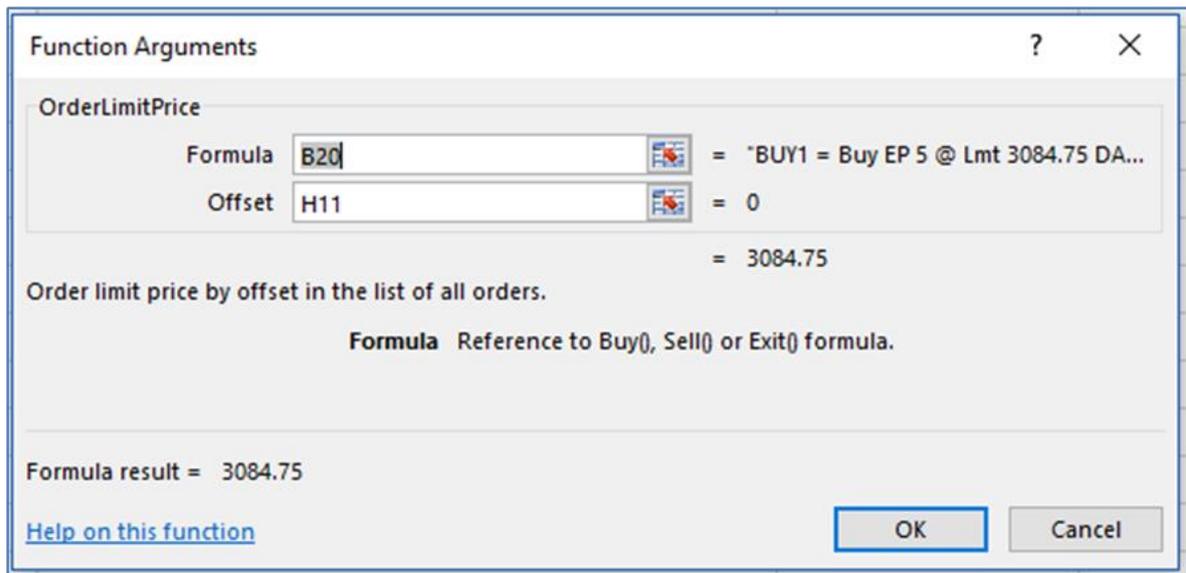
=OrderFilledSize()



## Order Limit Price 関数（注文の指値）

Order Limit Price 関数は指値を取得します。

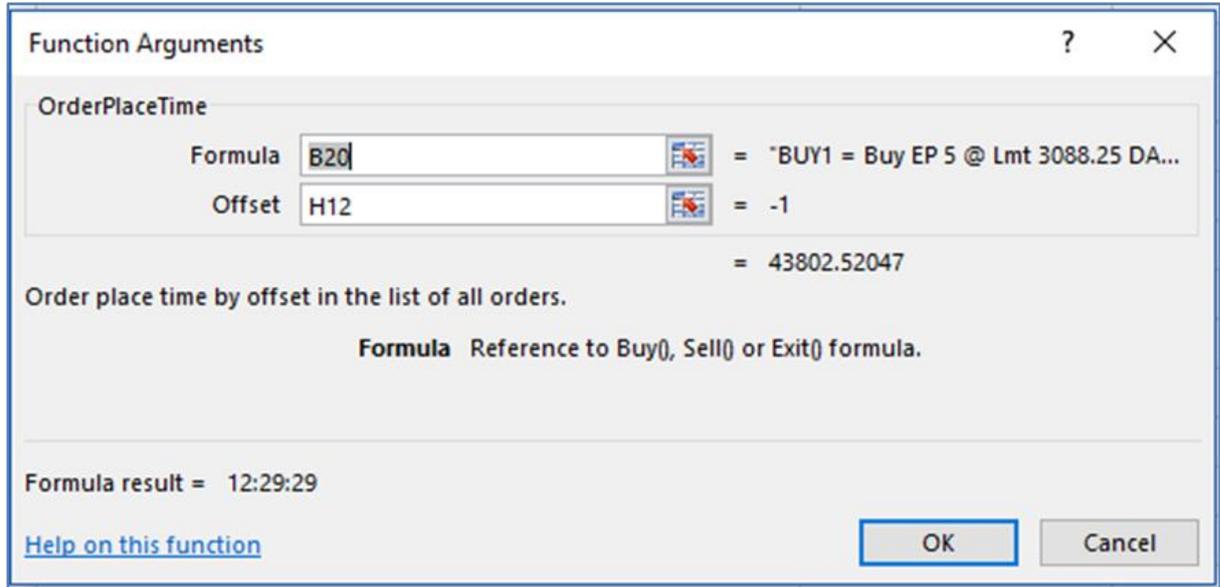
=OrderLimitPrice()



## Order Place Time 関数（発注時間）

Order Place Time 関数は発注時間を取得します。

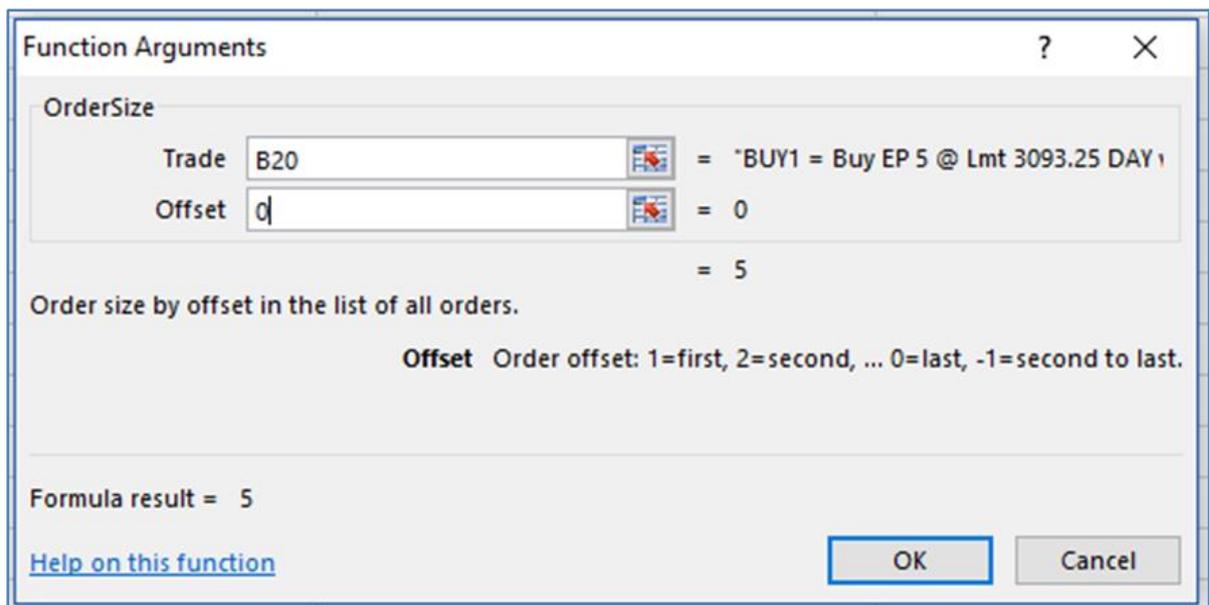
=OrderPlaceTime()



## Order Size 関数（注文枚数）

Order Size 関数は注文枚数を取得します。

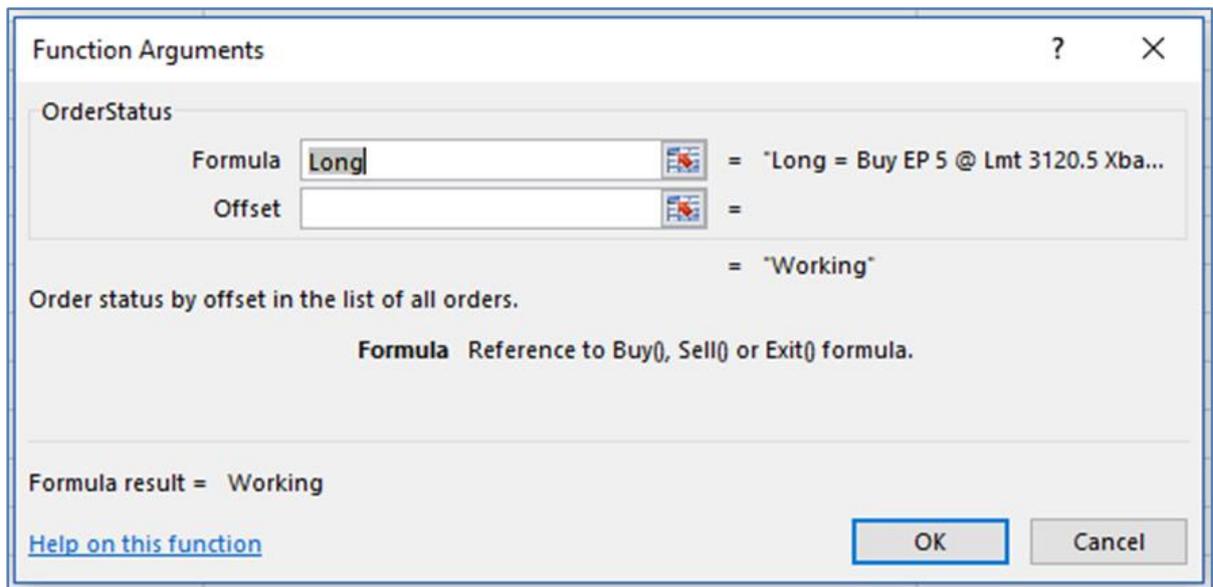
=OrderSize()



## Order Status 関数（注文のステータス）

Order Status は注文のステータスを取得します。

=OrderStatus()



下記が取得可能なステータス一覧です。

**New:** 注文を生成しましたが発注されていません。これはゲートウェイサーバー滞留時のステータスであるため、XLS Trader 上ではほとんど表示されません。

**Working:** 注文は取引所に到達し、取引所からも返り電文を受領している、正常な状態です。

**PendingPlace:** 注文は取引所に送られましたが、取引所からの返りがありません。

**PendingPlaceAtSessionStart:** 取引開始時間前のため、注文はゲートウェイサーバー内に留まります、取引開始後に取引所に送られます

**PendingReplace:** 指値／枚数変更指示を送りましたが、取引所から返りがありません。

**PendingCancel:** キャンセル指示を送りましたが、取引所から返りがありません。

**Failed:** 注文はリジェクト、タイムアウト、または失効されました。

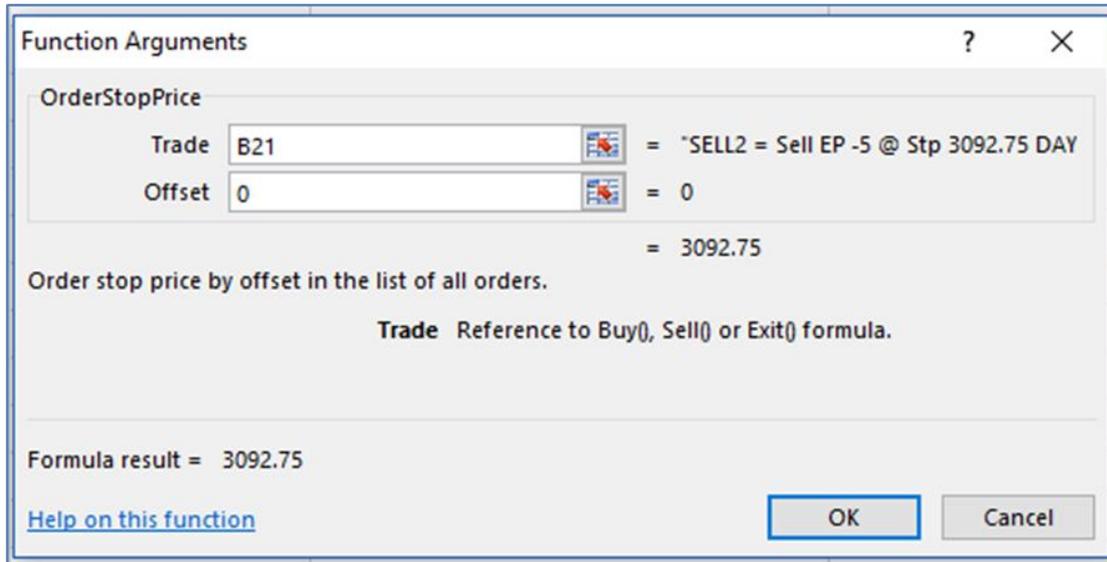
**Canceled:** 注文はユーザーもしくは取引所によってキャンセルされました。

**Filled:** 全量約定しました。

## Order Stop Price 関数（ストップ注文の値段を取得）

ストップ注文の値段を取得します。

=OrderStopPrice()



## Working 注文関数グループ

下記 4 つの関数にて、ワーキング中の指値注文情報が取得できます。

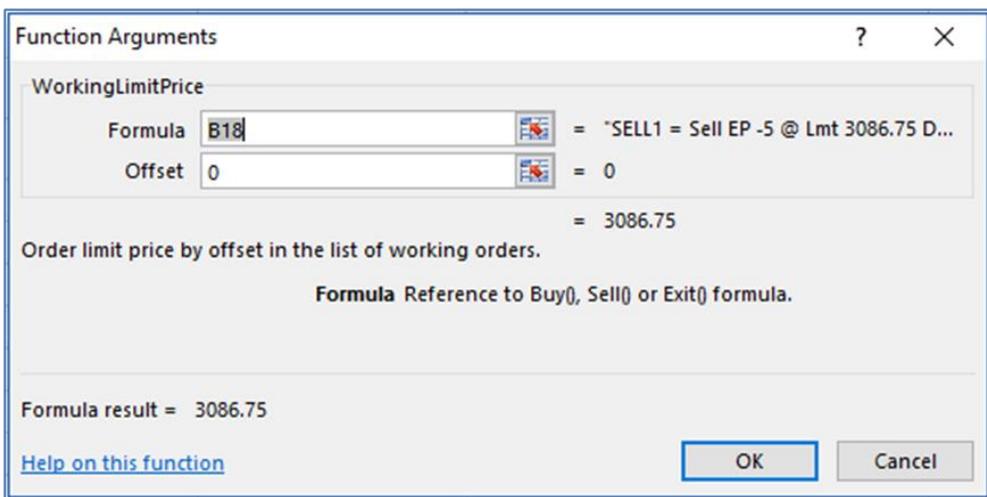
下記の Entry 関数式がセル B18 に配置されているという想定の下、ワーキング注文情報の取得方法を説明します。

SELL1 = Sell EP -5 @ Lmt 3086.75 DAY when \$C\$18

### WorkingLimitPrice 関数

ワーキング注文の指値を取得します。

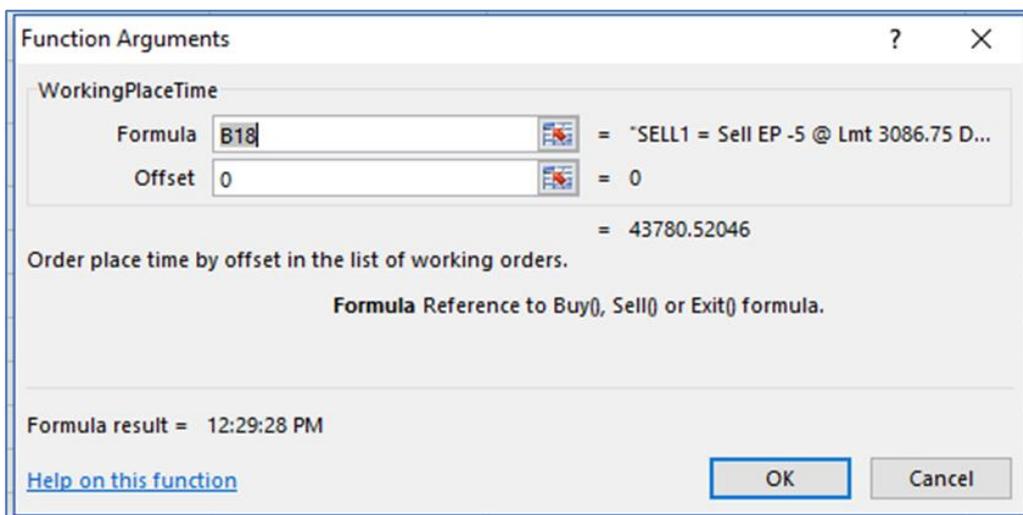
=WorkingLimitPrice(B18,0)



### WorkingPlaceTime 関数

ワーキング注文の発注時間を取得します。

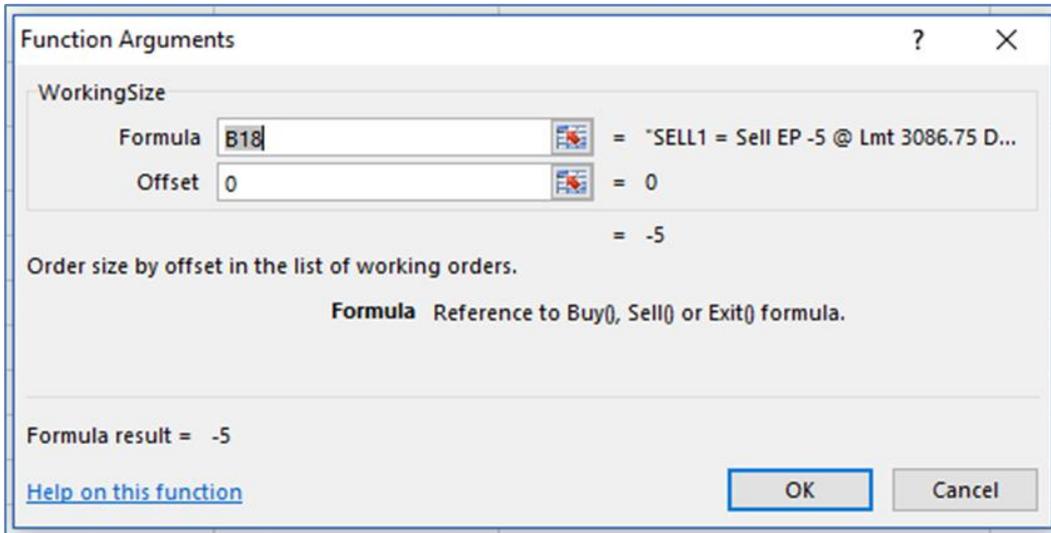
=WorkingPlaceTime(B18,0)



## WorkingSize 関数

ワーキング注文の枚数を取得します。

=WorkingSize(B18,0)

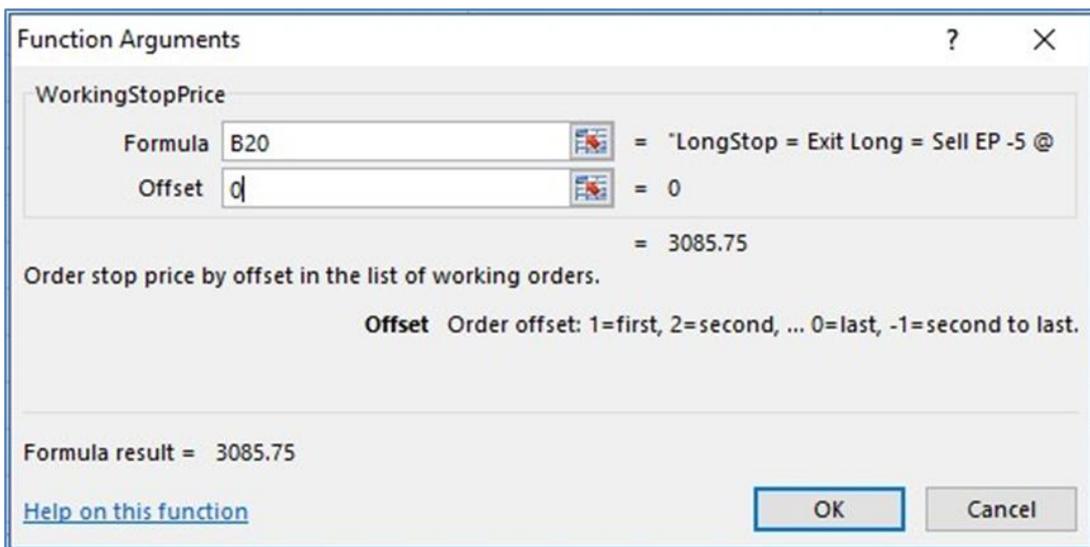


## WorkingStopPrice 関数

この関数はワーキング中のストップ注文の値段を取得します。

セル B20 に配置されている、Exit 関数式 (LongStop = Exit Long = Sell EP -5 @ Stp 3085.75 DAY) のストップ注文情報を取得する場合の、ウィザードの作成例は下図のとおりです。

=WorkingStopPrice



## 取引情報関数グループ

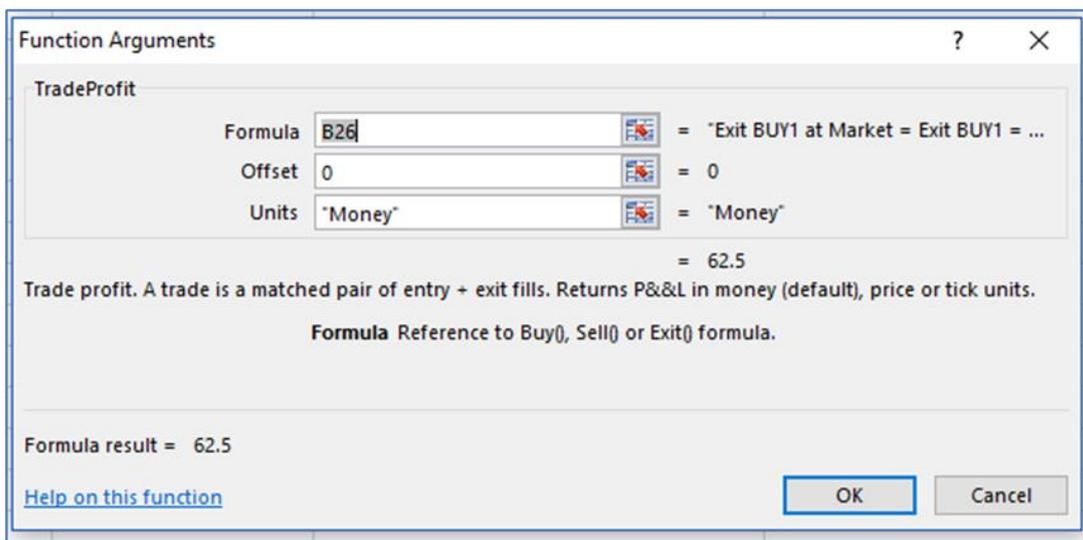
ここでは、損益などの情報を取得するための3つの関数を紹介します。

いずれの関数も、Formula パラメーターには Exit 関数を設定します。また、Offset 値の設定により、表示する落ち商いを選択することも可能です。

### Trade Profit 関数

金額 (Money)、値段 (Price) もしくはティックを取得します。デフォルト値は Money となります。

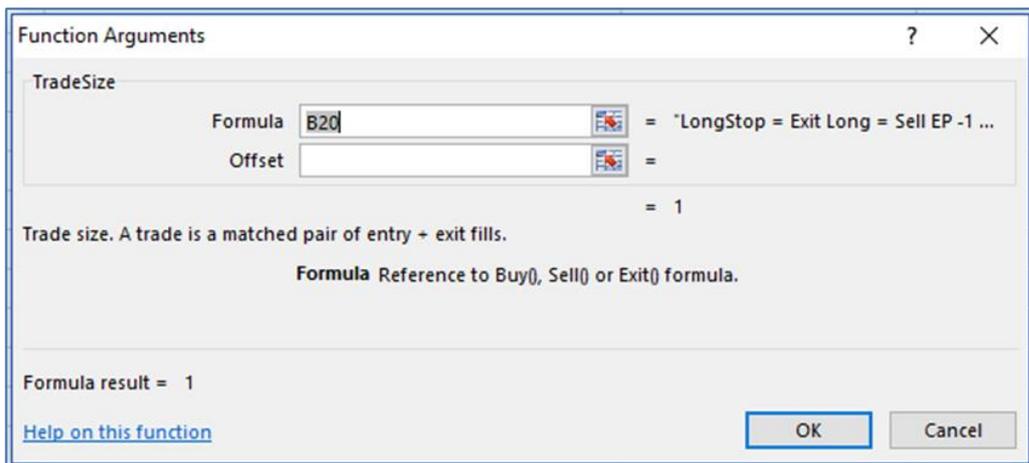
=TradeProfit(B26,0,"Money")



### Trade Size 関数

Formula パラメーターで設定した Exit 関数における、約定枚数を取得します。

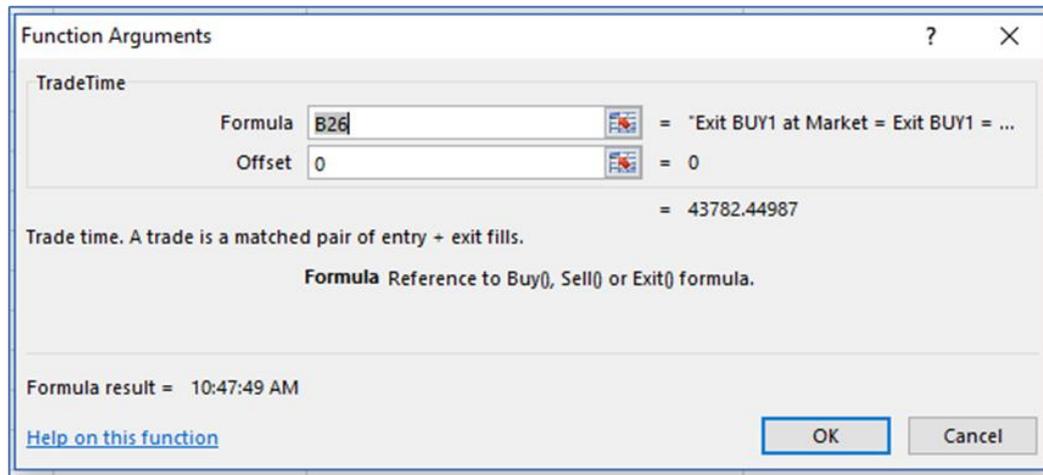
=TradeSize(B20)



## Trade Time 関数

Formula パラメーターで設定した Exit 関数における、約定時間を取得します。

=TradeTime(B26,0)



## 注文詳細を保存する関数

注文詳細を保存する関数グループを以下のとおり紹介します。

なお AverageEntryPrice 関数と OpenPosition 関数は [OCO の設定](#) で既に紹介したので、ここでは割愛します。

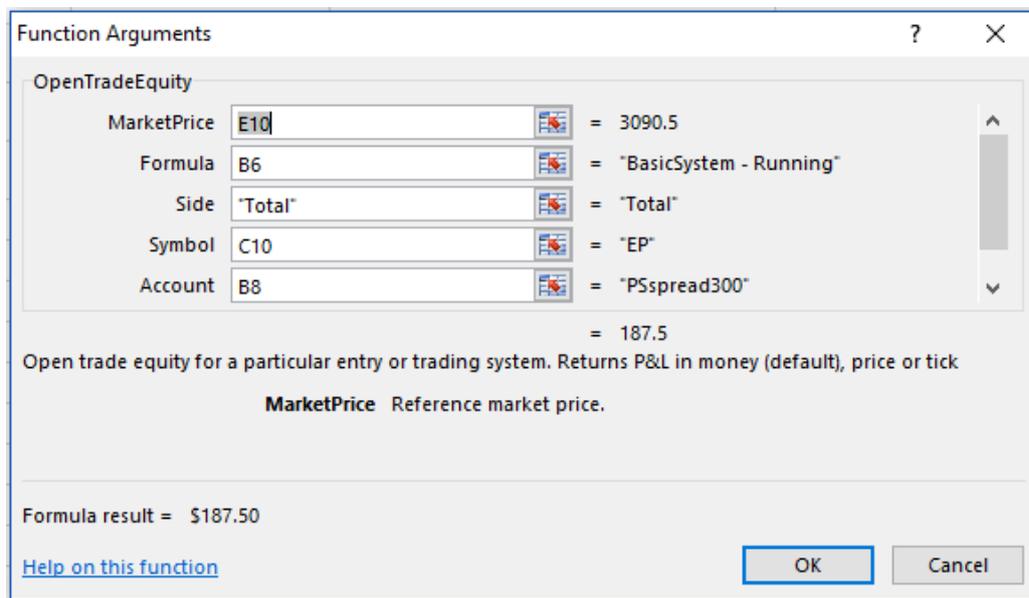
- =AverageEntryPrice()
- =OpenTradeEquity
- =ClosedNetProfit()
- =CsvFills()
- =CsvTrades()
- =OpenPosition()

### Open Trade Equity 関数

Open Trade Equity 関数は Entry 関数もしくは Trading System、Side、Symbol および Account を設定し、対象の値洗い情報を取り出すことができます。

MarketPrice パラメーターは RTD 関数式で直近値を配置しているセルを参照します。

=OpenTradeEquity(E10,B6,"Total",C10,B8,"Money"]]



## Closed Net Profit 関数

Closed Net Profit 関数は TradeSystem 関数、Entry 関数、Exit 関数における直近の損益を取得します。取得可能な単位は金額、値段、及びティックスとなります。

= ClosedNetProfit(B6,C10,C8)

**Function Arguments**

**ClosedNetProfit**

Formula	B6	= "BasicSystem - Running"
Symbol	C10	= "EP"
Account	C8	= "PSspread300"
Units	"Money"	= "Money"

= 187.5

Closed net profit for a particular entry, exit, or trading system. Returns P&L in money (default), price or tick units.

**Formula** Reference to trading system, entry or exit formula: TradeSystem(), Buy(), Sell() or Exit().

Formula result = \$187.50

[Help on this function](#) OK Cancel

## CSV Fills 関数

CsvFiles 関数を使って、約定表を csv 形式で作ることができます。

**System** パラメーターは対象の Trade System をセル参照します。

**OnOff** パラメーターは 0 が待機、1 が出力です。

**DirPath** (Directory Path) パラメーターには CSV ファイルの保存場所を指定します。ブランクにすると使用中のファイル内にワークシートを生成します。

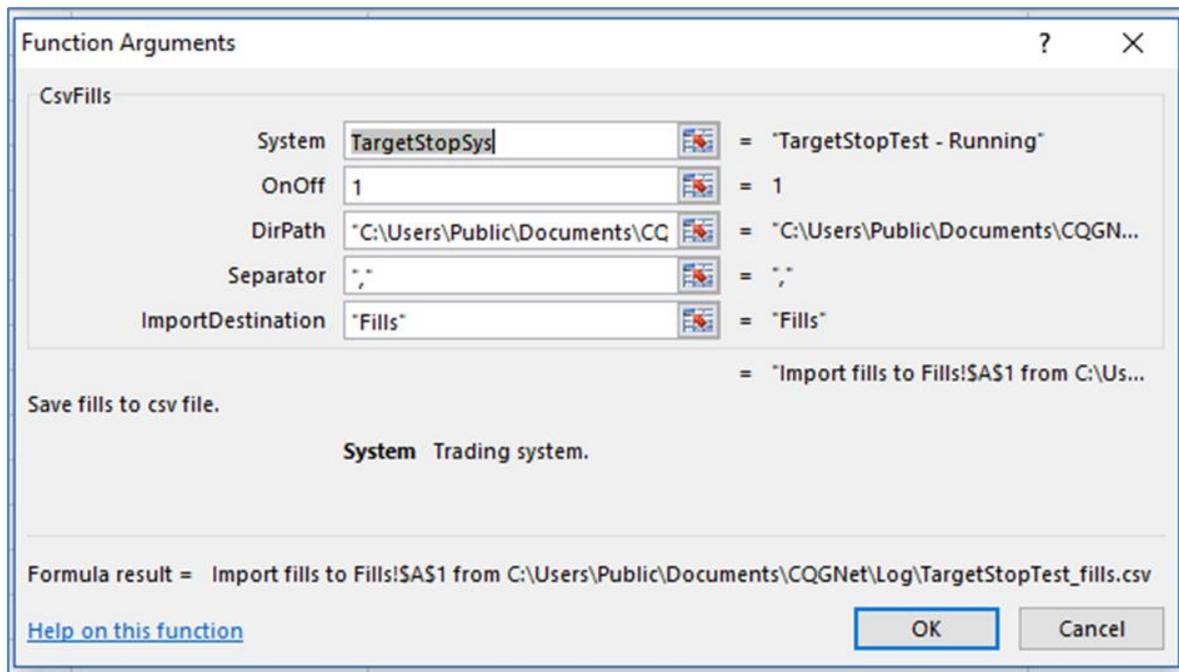
**Separator** パラメーターは必要に応じて設定してください。区切り位置に使う記号を変更することができます。デフォルトは"tab"(大文字、小文字可) タブレーションが設定されており、下記記号に変更可能です。

- "," カンマ
- ";" セミコロン
- "|" パイプ キャラクター

**ImportDestination** パラメーターはワークシートの名前を設定します。

以下が設定例となります。

```
=CsvFills(TargetStopSys,1,"C:¥Users¥Public¥Documents¥CQGN¥Log",",","Fills")
```



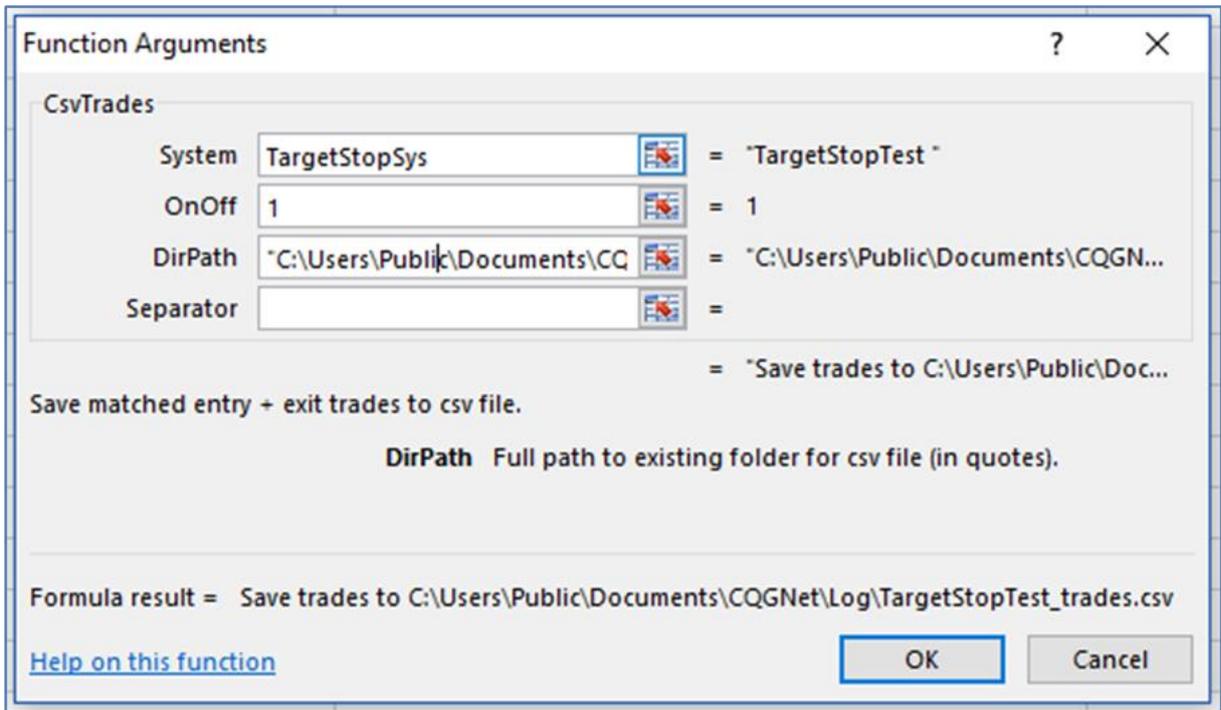
下表が上記設定で出力した約定表のサンプルです。

	A	B	C	D	E	F	G	H
1	Time	Trade	Size	Price	OrderId	Symbol	Account	StartTime
2	10:31:19	Long	1	3114.75	1049062976	F.US.EPZ19	PSspread300	8:30:27
3	10:31:20	Long	1	3114.75	1049062976	F.US.EPZ19	PSspread300	8:30:27
4	10:31:50	Long	1	3114.75	1049062976	F.US.EPZ19	PSspread300	8:30:27
5	10:31:54	Long	1	3114.75	1049062976	F.US.EPZ19	PSspread300	8:30:27
6	10:31:54	Long	1	3114.75	1049062976	F.US.EPZ19	PSspread300	8:30:27
7	10:44:42	LongStop	-1	3113.5	1049052928	F.US.EPZ19	PSspread300	8:30:27
8	10:44:42	LongStop	-1	3113.5	1049052928	F.US.EPZ19	PSspread300	8:30:27
9	10:44:42	LongStop	-2	3113.5	1049052928	F.US.EPZ19	PSspread300	8:30:27
10	10:44:42	LongStop	-1	3113.5	1049052928	F.US.EPZ19	PSspread300	8:30:27
11	10:44:42	LongTarget	-1	3113.5	1049042842	F.US.EPZ19	PSspread300	8:30:27
12	11:00:54	Long	1	1686.3	1049033724	F.US.PAEZ19	PSspread300	8:52:34
13	11:03:15	Long	1	1685.7	1049063503	F.US.PAEZ19	PSspread300	8:52:34

## CsvTrades 関数

CsvTrades 関数は全ての新規と仕切り約定をエクセル csv ファイルで指定のディレクトリに保存することができます。

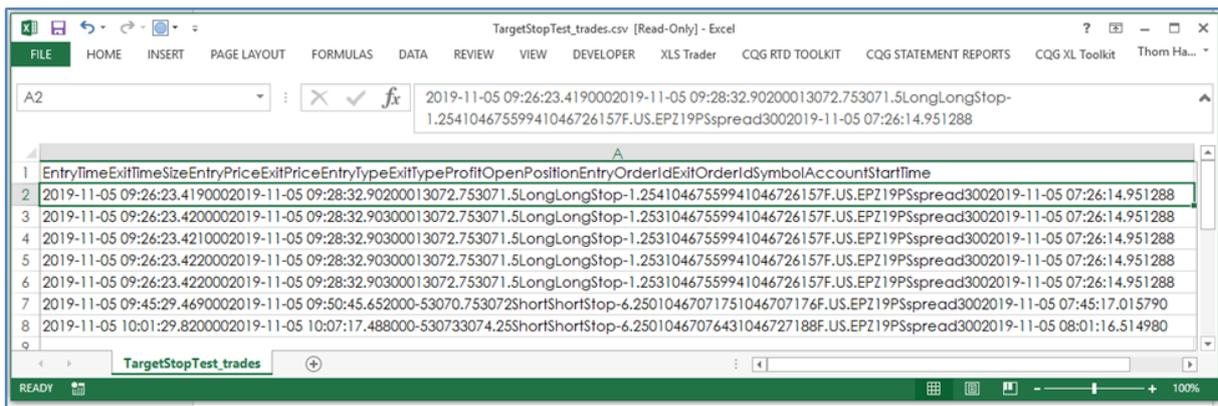
=CsvTrades(TargetStopSys,1,"C:\Users\Public\Documents\CQNet\Log")



下記が CSV ファイルで出力できる項目です。

- EntryTime
- ExitTime
- Size
- EntryPrice
- ExitPrice
- EntryType
- ExitType
- Profit
- OpenPosition
- ExitOrderId
- Symbol
- Account
- StartTime

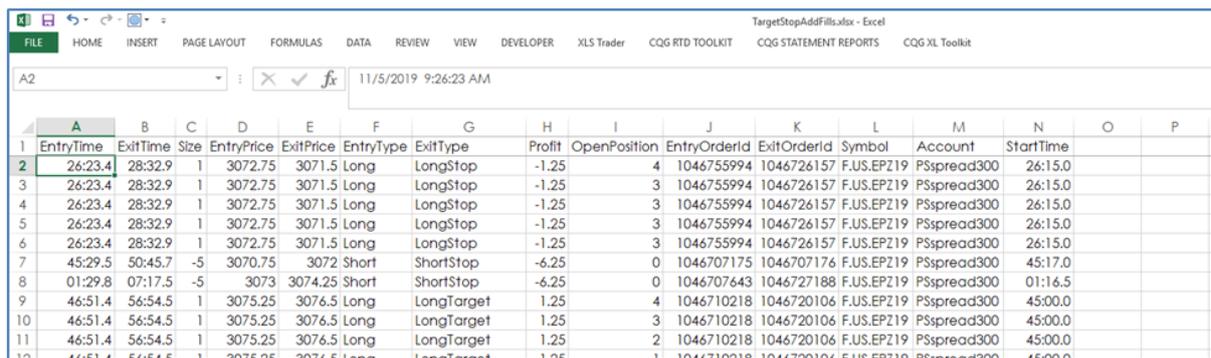
サンプルの CSV ファイルは以下のとおりです。



ImportDestination パラメーターにて設定した文字列はワークシートの名前となります。

例えば、下記関数にて生成されるワークシートの名前は“Trades.”です。全ての仕切商いはこのワークシートに更新されます。

=CsvTrades(TargetStopSys,1,"C:¥Users¥Public¥Documents¥CQGNet¥Log",",","Trades")



## オーダーログ関数

次に、Trade System の取引詳細のログを呼び出す、下記 4 つの関数について説明します。

- =AllOrdersLog()
- =FilledOrdersLog()
- =WorkingOrdersLog()
- =FailedOrdersLog()

AllOrdersLog 関数は全てのワーキング、約定、キャンセルおよびリジェクト注文を表示します。

FilledOrdersLog 関数は約定注文のみ、WorkingOrdersLog 関数はワーキング注文のみを、そして FailedOrderLog 関数はリジェクトされた注文を表示します。

Order Log 関数について、2つのポイントを説明します。一つ目は、これらの関数はエクセル内の配列数式として入力しなければならないこと。二つ目は、この関数はデータを保存しないということです。Trade System を停止したら、それ以降のログ更新はストップし、次にログインした時点でログはリセットされます。また、停止する以前のログを再び呼び出すことはできません。

取引ログは下へ下へ更新されるので、十分なスペースを確保する必要があります。

この関数では、Formula パラメーターに Trade System 関数をセル参照する必要があります。ここでは、セル C2 に配置されている Trade System 関数を参照することを想定します。

AllOrdersLog 関数の入力、対象のセルに=AllOrdersLog(\$C\$2) と入力し、Shift+Ctrl+Enter を押下してウィザードを開いてください。

ウィザード内の計算結果として{=AllOrdersLog(\$C\$2)}と表示されます。{} はこのデータが配列であることを表しています。

その後、発注がなされるまでは、セル上には#N/A error が表示されます。発注がなされた段階で、下図のような情報が表示されます。より新しい情報が上のセルに表示され、古い情報は下へ下へ降りていきます。下図の例ではセル B30 に Long 注文の約定時刻、枚数及び値段が、セル B29 に関連する注文詳細が表示されます。

最終的にこの建玉は Market 注文にて仕切られ、その日付、時刻、枚数、値段などの約定詳細がログに表示されています。また、Target 注文及び Stop 注文がキャンセルされていることが分かります。

B29		: X ✓ f {=AllOrdersLog(\$C\$2)}	
A	B		
22			
23	<b>All Orders Logs</b>		
24	4 orders:		
25	LongStop - Cancelled - Sell F.US.EPM19 -1 @ Stp 2888.5 DAY (account="SIMThom Spreader (PSspread300)"; gw_id=986416328(986416328) guid=3497944e-ba1b-430d-8d06- eee189abcef9 (id=X1_E2_1))		
26	LongTarget - Cancelled - Sell F.US.EPM19 -1 @ Lmt 2893.5 DAY (account="SIMThom Spreader (PSspread300)"; gw_id=986416329(986416329) guid=d2100dd5-e3c1-4fee-8233- ebdfb71b21c0 (id=X2_E2_1))		
27	Exit Long at Mkt - Filled - Sell F.US.EPM19 -1[-1] @ Mkt[2890.5] DAY (account="SIMThom Spreader (PSspread300)"; gw_id=986436311(986436311) guid=620baf20-7b0d-4c39-9179- b986e1a618a1 (id=X3_E2_1))		
28	5/8/2019 1:52:12 PM -1 @ 2890.5		
29	Long - Filled - Buy F.US.EPM19 1[1] @ Lmt 2891[2891] DAY (account="SIMThom Spreader (PSspread300)"; gw_id=986396893(986396893) guid=6b6253f4-0f69-4e7c-af5c- d06377ca9370 (id=E2_1))		
30	5/8/2019 1:44:53 PM 1 @ 2891		
31	#N/A		

参考 1 : Order Logs 関数は全注文詳細を表示するので、中には表示不要な情報も含まれているかもしれません。

下記の関数式は Model ワークシートのセル B25 を参照しています。そして注文詳細内の、最初の“(”を見つけ出し、その後のデータを全て非表示にします。下表がサンプルとなります。

こちらは XLS TRADER 独自の関数ではないため、使用する場合は下記をコピーしてエクセルファイルに貼り付けてください。

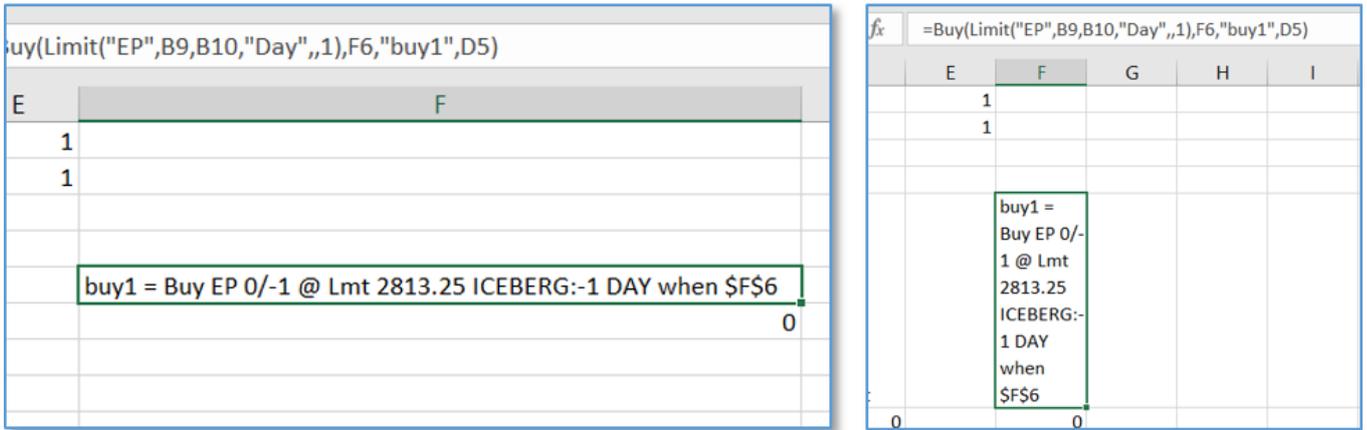
=IFERROR(LEFT(Model!B25,FIND("~",SUBSTITUTE(Model!B25,"(", "~",1))-1),Model!B25)

4 orders:					
LongStop - Cancelled - Sell F.US.EPM19 -1 @ Stp 2888.5 DAY					
LongTarget - Cancelled - Sell F.US.EPM19 -1 @ Lmt 2893.5 DAY					
Exit Long at Mkt - Filled - Sell F.US.EPM19 -1[-1] @ Mkt[2890.5] DAY					
5/8/2019 1:52:12 PM -1 @ 2890.5					
Long - Filled - Buy F.US.EPM19 1[1] @ Lmt 2891 [2891] DAY					
5/8/2019 1:44:53 PM 1 @ 2891					
#N/A					
#N/A					

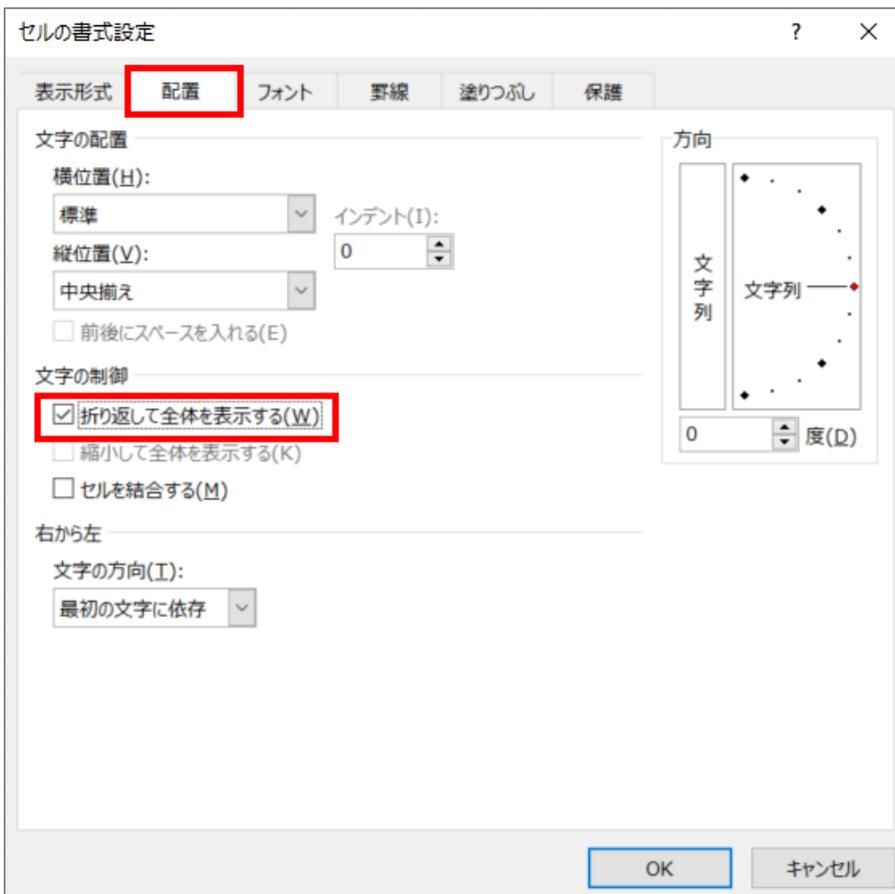
FilledOrdersLog、WorkingOrdersLog 及び FailedOrdersLog は、AllOrdersLogs と同様の表示形式になります。



もし全ての文字列を表示させたい場合は、二つの選択肢があります。一つ目はセルの幅を広げる、もう一つはテキストをセルの幅内に収まるように折りたたむ、この二つです。



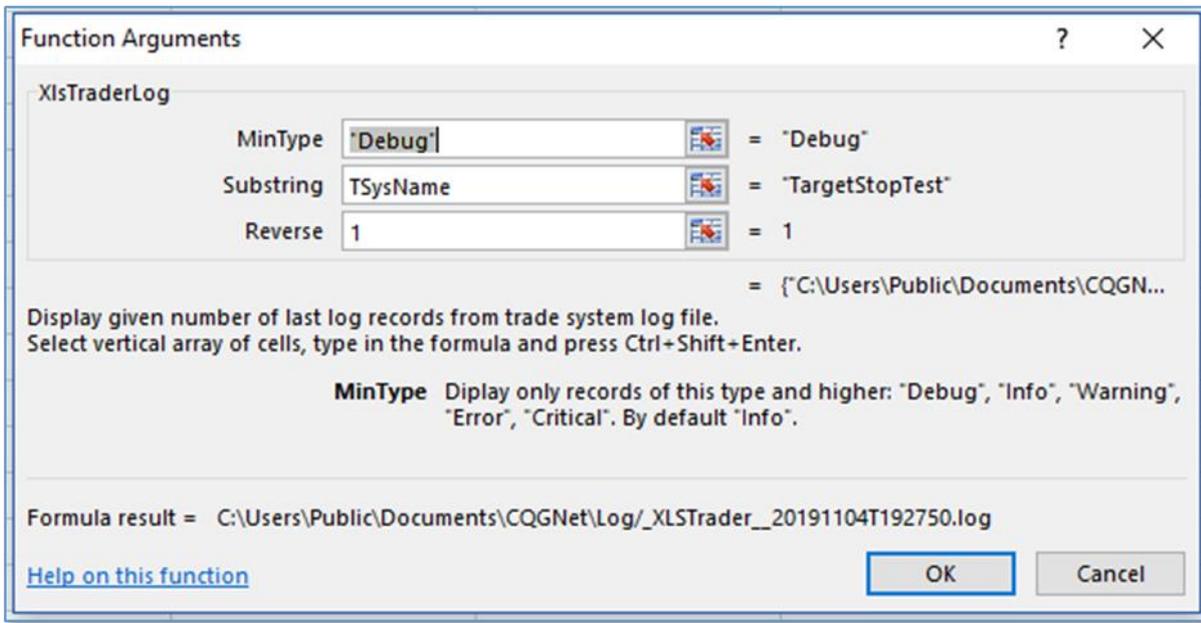
折りたたみ設定を有効にするためには、まず対象のセルを右クリックし、セルの書式設定を選択すると設定画面が表示されるので、配置タブを選択してください。そしてタブ内の“折り返して全体を表示する”にチェックを入れてください。



## XLS Trader ログ関数

XlsTrader ログ関数はエクセルにログを表示します。このログはその他の注文ログとは表示内容が異なります。

この関数は配列数式として入力されます。表示させたい列とカラムを選択し、=XlsTraderLog 関数を追加してください。



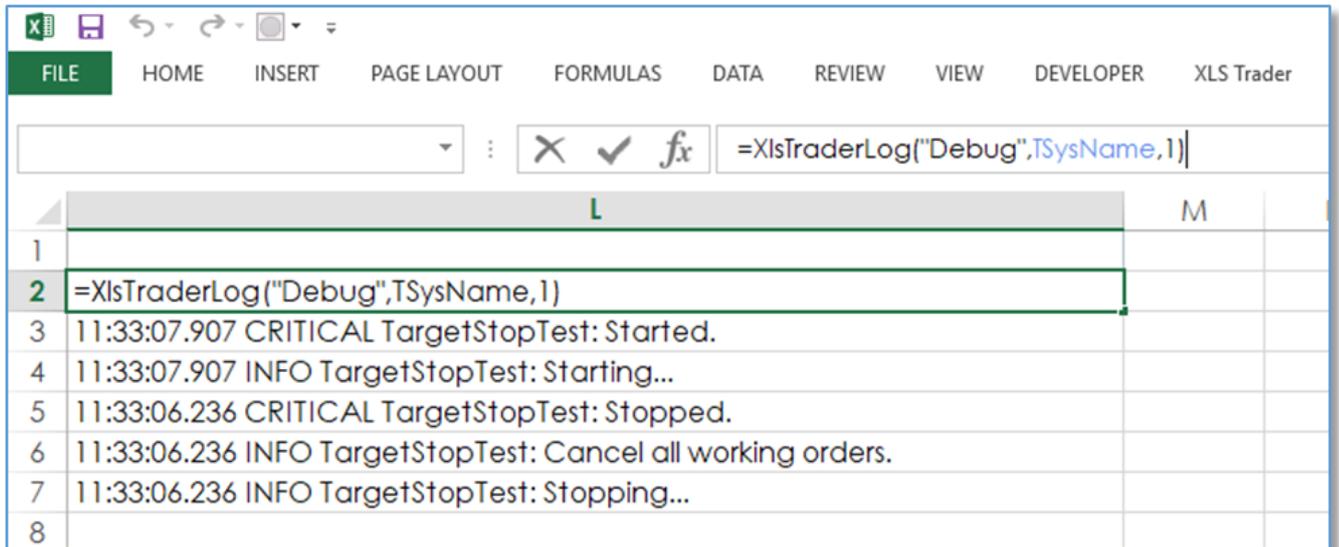
MinType には出力するログタイプを設定します。各々のレコードタイプは Trade system 自身の動作、もしくは取引に影響する何らかの外部イベント（注文が取引所によって認識済など）を詳細に記述します。デフォルトは Info となります。

- Debug:** テクニカル情報を書き出します。このログはソフトウェアのデバッグ（不具合修正）には極めて重要なため、このログタイプを “Debug” と呼んでいます。
- Info:** これは最もよく使われるログです。このログは Trade System の動作や外部イベントを書き出します。
- Warning:** Warning（警告）とは異常動作ではあるが、不具合ではない情報を書き出します。Warning は不具合発生の先行シグナルである可能性もありますが、結局不具合には至らない場合もあります。
- Error:** Trade System の不具合発生、注文リジェクトなどを報告
- Critical:** たとえば、システムが停止し稼働したが、Order routing が切断されていた場合、など動作上の問題を書き出します。このログはその他のログ履歴より強調して表示されます。

**Substring** パラメーターはシステム名によってログにフィルターをかけます。

**Reverse** パラメーターはログの並び順の昇順と降順（時系列）を設定します。

この配列を修正した場合は、更新するために **Shift+Ctrl+Enter** を押下する必要があります。



The screenshot shows an Excel spreadsheet with the following data:

	L	M
1		
2	=XlsTraderLog("Debug",TSysName,1)	
3	11:33:07.907 CRITICAL TargetStopTest: Started.	
4	11:33:07.907 INFO TargetStopTest: Starting...	
5	11:33:06.236 CRITICAL TargetStopTest: Stopped.	
6	11:33:06.236 INFO TargetStopTest: Cancel all working orders.	
7	11:33:06.236 INFO TargetStopTest: Stopping...	
8		

## ユーティリティ関数

これらの関数グループは接続、バージョン、時刻などのユーティリティ情報を表示します。

- =ApiTimeout()
- =Instrument()
- =IsMarketDataConnected()
- =IsOrderRouterConnected()
- =LineTime()
- =NAPlaceholder()
- =OrderTransactionPerSecondLimit()
- =XlsTraderVersion()

### ApiTimeout 関数

スペックの低い PC やネット接続が遅いコンピューターは、Order Routing サーバーへの接続確立に想定以上の時間を要することがあります。この関数はタイムアウトの時間を調整することができます。

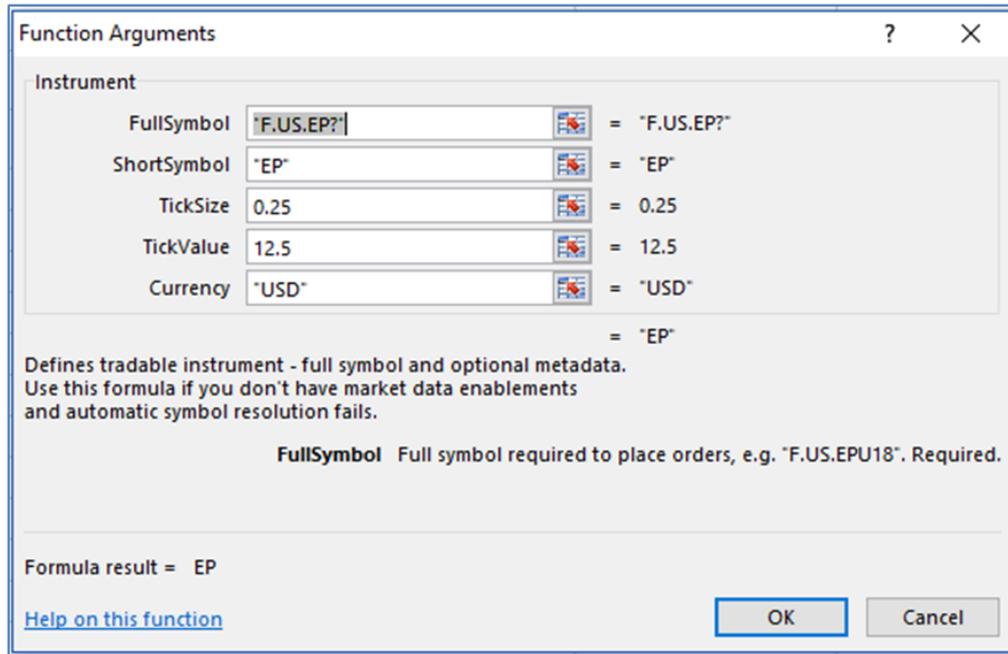
=ApiTimeout()

	F	G	H	I
1	IsReversal	Version		
2	Not reversal	0.28.0.12_2019-11-27_x64		
3	Market Data	Order Routing	Time-out is 15 seconds	
4	1	1		
5				

## Instrument 関数

取引所の Market Data 閲覧権限がない銘柄を XLS TRADER で取引したい場合は、この Instrument 関数にて対象の銘柄を追加し、メタデータとして書き出すことにより、その銘柄を取引可能にすることができます。下記は EP (E-mini S&P500) を取引すべく追加する場合の計算式です。

```
=Instrument("F.US.EP?","EP",0.25,12.5,"USD")
```



## Connectivity アイコン

Order Routing と Market Data への接続を確認する関数を紹介します。エクセルリボン内の XLS Trader をクリックすると、“Connectivity” アイコンがあります。もし True、つまり接続が確立されていれば、IsMarketDataConnected 関数および IsOrderRouterConnected 関数は共に 1 を返します。

したがって「=@IsMarketDataConnected()+@IsOrderRouterConnected()」が True、つまりこのセルの値が 2 の場合は、下図のとおり “Connected” を表示され、緑にハイライトされます。

	F	G	H	I
1	IsReversal	Version		
2	Not reversal	0.28.0.12_2019-11-27_x64		
3	Market Data	Order Routing	Time-out is 15 seconds	
4	1	1	Connected	
5				

## LineTime 関数

ラインタイム、いわゆるアメリカ中央部の時間帯を取得する関数です。

=LineTime()

	F	G	H	I
1	IsReversal	Version		Line Time
2	Not reversal	0.28.0.12_2019-11-27_x64		4:09:28 PM
3	Market Data	Order Routing	Time-out is 15 seconds	
4	1	1	Connected	

## NAPLaceholder 関数

AverageEntryPrice 関数などは、データが何もない場合は#N/A を返しますが、エクセルには、#N/A エラーの代わりにブランクなどを表示する関数があります。

例えば、=IFERROR(AverageEntryPrice(Long),"")は取得した約定平均値がエラー値であった場合はブランクを返します。

XLS TRADER には#N/A エラーに対応する独自の変換関数があり、これはワークシート内の全てのセルを対象とし、代替え表示が可能です。これは数字ではなく String で設定する必要があります。

=NAPLaceholder()

下図はセル D1 に=NAPLaceholder("...")が配置されており、セル D2 に=AverageEntryPrice(A2)が配置されています。セル A2 には買い指値注文が配置されていますが、まだ約定はありません。したがって約定平均値は N/A エラーとなり、D2 には"..."が代替え表示されます。

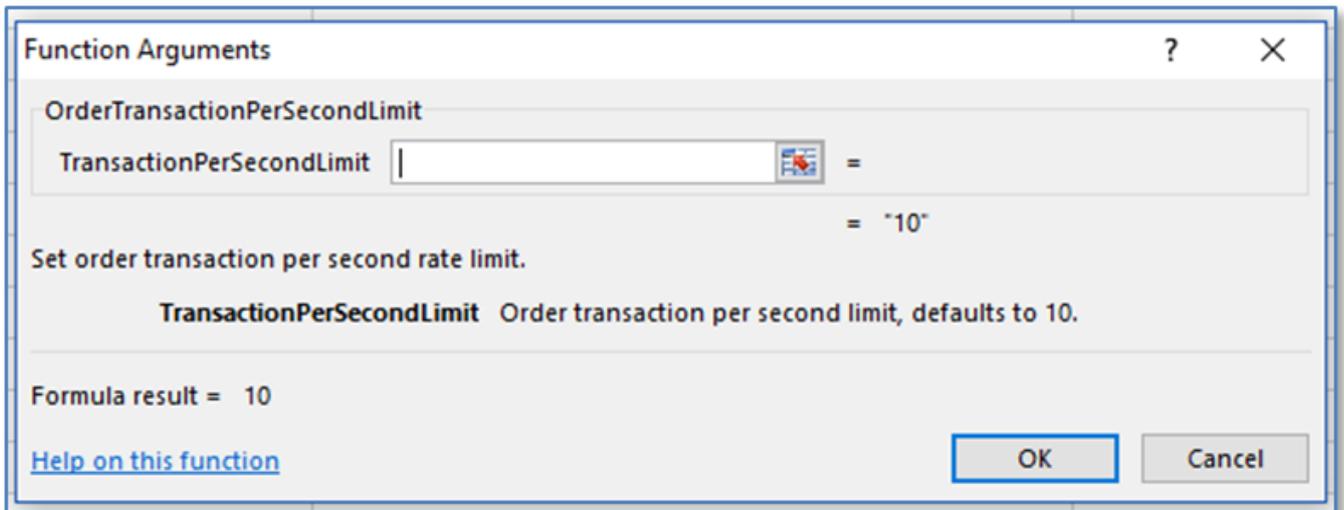
	F	G	H	I
1	IsReversal	Version		Line Time
2	Not reversal	0.28.0.12_2019-11-27_x64		4:17:17 PM
3	Market Data	Order Routing	Time-out is 15 seconds	Show #N/A as "..."
4	1	1	Connected	

## Order Transaction Per Second Limit 関数

Order Transaction Per Second Limit 関数は Order RoutingAPI に直ちに送信可能な秒間の最大送信件数を設定します。このリミットを超えたトランザクションは遅延します。デフォルト値は 10 に設定されています。

一方、CQG Gateway 側でも秒間のトランザクション件数は設定されています。上記関数の場合は遅延するだけですが、Gateway 側の場合は、設定値を超えたトランザクションはリジェクトされます。

=OrderTransactionPerSecondLimit()



## XLS Trader Version 関数

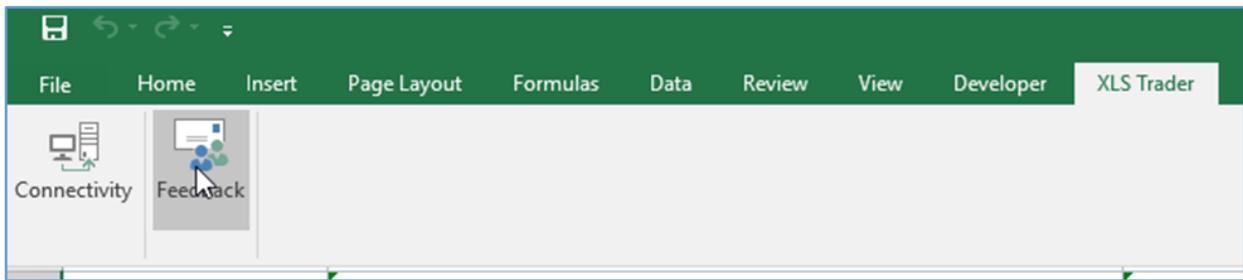
この関数は現在使用しているバージョンを返します。このバージョン情報は何らかの不具合が発生した場合に、CQG にて調査するうえでの重要な情報です。

=XlsTraderVersion()

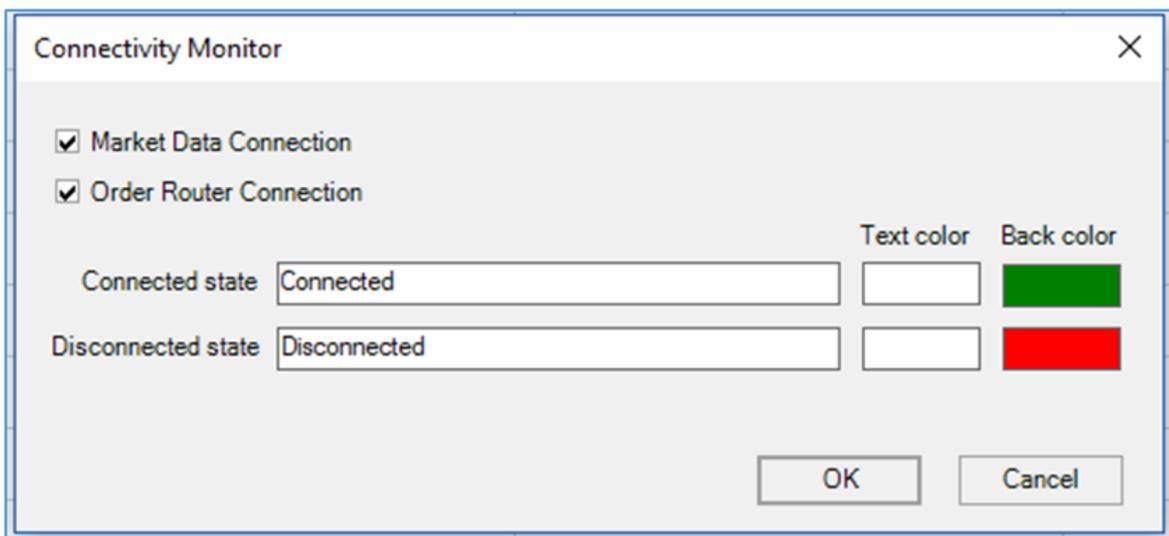
File Home Insert Page Layout Formulas Data Review View Developer XLS Trader			
G2 : X ✓ fx =XlsTraderVersion()			
	F	G	H
1	<b>IsReversal</b>	<b>Version</b>	
2	Not reversal	0.28.0.12_2019-11-27_x64	
3	<b>Market Data</b>	<b>Order Routing</b>	Time-out is 15 seconds
4	1	1	
5			

## XLS Trader コマンドバー ボタン

XLS TRADER をインストールすると、コマンドバーに XLS Trader タブが表示されます。タブ内には Connectivity と Feed Back アイコンがあります。



ある任意のセルを選択し “Connectivity” をクリックすると、対象のセルに `=IsMarketDataConnected()+IsOrderRouterConnected()` の関数式が配置されます。Market Data、Order Routing 共に接続が確立されている場合は緑のハイライト、どちらか一方のみもしくはどちらにも接続がなされていない場合は赤のハイライトでセルが色付けされます。こちらの設定はダイアログボックスから変更可能です。



Feedback アイコンは不具合を CQG に報告するためのものです。これはサポートやトレーニングを依頼するためのものではありません。サポートやトレーニングの依頼は CQG Product Specialist に連絡してください。

Feedback アイコンをクリックすると、下図のダイアログが表示されます。“Your Message” ボックスに不具合の内容を入力し、ログやエクセルファイルを送付する場合は“Attach CQG IC logs” もしくは“Attach active Excel Workbook” のいずれかもしくは両方にチェックを入れ、ファイルを添付してください。また、Attach File ボタンよりその他のファイル（スクリーンショットなど）の添付が可能です。CQG はこれらの情報をユーザー占有のものとし、他社と共有することはありません。

Contact CQG Customer Support

**Your message**

**Select Files to Report**

**Your CQG IC Serial Number: 181330**

We appreciate your taking time to send us your files. Please send us as much as you can. The more information we have, the better we're able to troubleshoot.

**Attach CQG IC logs**  
Limit logs to the last  days (3 min).

**Attach active Excel Worksheet**

Attach File

Submit Cancel

## XLS TRADER を閉じる際の注意

もし Trade System が稼働し、且つワーキング注文がある状態で XLS TRADER を閉じた場合、注文はキャンセルされず、約定するか、もしくは期限切れで失効となるまでワーキング状態のまま残ります。

XLS TRADER を閉じる際にワーキング注文の存在に気付くために、セルがアラート表示される設定をお勧めします。

下図はセル C6 の Trade System が 0 以外の場合、セルは赤くハイライトされます。他の条件式を伴う関数にも、XLS Trader を閉じる前に 0 以外の数値が入力されていないか確認するために、条件アラートの設定が有効です。

A	B	C
<b>XLS Trade System Demo</b>		
<b>Trading System Name</b>		<b>On (1) / Off (0)</b>
BasicSystem - Running		1

**Conditional Formatting Rules Manager**

Show formatting rules for: Current Selection

New Rule... Edit Rule... Delete Rule

Rule (applied in order shown)	Format	Applies to	Stop If True
Formula: =SC56<>0	AaBbCcYyZz	=SC56	<input type="checkbox"/>

OK Close Apply