

**ProRealTime.com**  
The leading web-based charting software



# 目次

## はじめに

---

トレーディングシステム・プログラミングへのアクセス.....	6
トレーディングシステム作成ウインドウ.....	7
キーボードショートカット.....	9

## トレーディングシステムのプログラミング

---

ProBuilder 言語でのプログラミング.....	10
市場エントリーとエグジット.....	10
ストップと目標値.....	14
プロテクティブストップ.....	14
目標利益の設定.....	15
トレーリングストップ.....	16
「IF」条件文を使用した「目標設定」と「ストップ設定」の使用.....	16
複数のストップと目標レベル.....	18
トレーディングシステムを QUIT でストップする.....	22
ポジショントラッキング.....	22
ポジションステータス 変数.....	22
ポジション変数のサイズ.....	23
TradeIndex(トレードインデックス).....	23
TradePrice(トレード価格).....	23
PositionPerf(ポジションパフォーマンス).....	25
PositionPrice(ポジション価格).....	25
StrategyProfit(戦略的利益).....	25
トレーディングシステム実行パラメータの定義.....	27
Cumulate Orders(累積注文).....	27
PreLoadBars(プレロードバー).....	29
FlatBefore(フラットビフォー) と FlatAfter(フラットアフター).....	29
NoCashUpdate(ノーキャッシュ更新・バックテストのみ).....	30
呼出インジケータ.....	30
ProRealTime インジケータ.....	30
パーソナルインジケータ.....	30
プログラミングに関するアドバイス.....	31
インジケータの呼出数を減少させる.....	31
パーソナルインジケータの呼出.....	31

## ProBacktest - トレーディングシステム・バックテスト

---

変数最適化.....	34
バックテスト実行期間の定義.....	39
バックテストトレーディング時間のカスタマイズ.....	39
ProBacktest のストップの理由.....	40

## PROORDER 自動トレーディング

---

トレーディングシステム自動実行のための準備.....	43
トレーディングシステムのスタートと結果確認の方法.....	45
自動トレーディングパラメータと実行条件.....	48
トレーディングシステムパラメータ.....	49
トレーディングシステム自動ストップ.....	49
ワークステーション内の手動・自動トレーディングの共存.....	51
複数のトレーディングシステムを同じ証券で実行する.....	51
インジケーター制限.....	53

## 付録 A :トレーディングシステム結果の表示

---

### 付録 B : 具体的なコード例

---

Heiken Ashi トレーディングシステム.....	60
ZigZag トレーディングシステム.....	60
トレーリングストップによる簡易ブレイクアウト範囲.....	61
スムーズドストキャスティックトレーディングシステム.....	62
スウィングトレーディング、ADX、移動平均.....	62
ポジションカウンター付トレーディングシステム.....	64
トレードインデックス付トレーディングシステム - ファインド・インサイドバー.....	65
資金管理戦略.....	66
プロテクティブストップと利益目標.....	66
インアクティビティストップ.....	66
累積注文 - ポジションカウンタを使用して既存ポジション を追加する.....	67
Classic Martingale (クラシック・マーチンゲール).....	68
Great Martingale (グレート・マーチンゲール).....	69
Piquemouche (ピケモツシュ).....	70
D'Alembert Pyramid (ダランベール・ピラミッド).....	71
Contre d'Alembert (コントレ・ダランベール).....	72

## 用語解説

---

## ProRealTime v10 を使用したプログラミング・トレーディングシステムの導入

本バージョンのプログラミングガイドは、ProRealTime v10 以降の製品に適用されます。

ProRealTime のトレーディングシステムツールを使用することで、プログラミングまたは作成アシスト(プログラミング不要)により個人に合わせた投資戦略を作成することができます。

トレーディングシステムは、以下の目的で実行されます：

- 証券の過去データへのパフォーマンステストを行うバックテストとして
- 自動トレーディングシステムとして：トレーディングアカウントまたはペーパーパートレーディングアカウントからリアルタイムで注文

トレーディングシステムプログラミングは、トレーディングシステムのプログラミングのみに適用される付加的指示とともに、ProRealTime のインジケーターの書き込みにも使用される ProBuilder プログラミング言語を使用しています。

トレーディングシステムプログラムには、以下のような個別の条件に基づいて、それぞれのトレーディングシステムのポジションを建てたり、ストップ設定やリスク管理設定を指示することができます：

- ワークステーションの定義済みインジケーターまたはプログラムされたインジケーター
- トレーディングシステムの過去のパフォーマンス
- ご利用のトレーディングシステムの最新の注文

トレーディングシステムの結果は、以下のフォーマットで表示されます：

- 損益曲線が特定の証券におけるシステムの利益と損失の状態を示す。
- ポジションヒストグラムは、システムのポジション(ポジション購入を示す緑のバーと空売りポジションを示す赤いバー)を表します。ポジションが特定の時間帯の中にある場合、バーは表示されません。
- このアプリケーション内にある詳細なレポートは、特定の証券についてバックテストまたは実行された時間帯のシステム統計を示します。

バックテストモードでは、トレーディングシステムの変数を最適化し、過去の特定期間においてどの値が最善の結果を示しているかを見ることができます。

自動トレーディングモードでは、トレーディングシステムによる注文がポートフォリオと注文リストに表示されます。ポートフォリオは、利益と損失によって更新されます。

このプログラミングガイドは以下のように構成されています：まず初めに、トレーディングシステム作成関数へのアクセス方法の解説があります。2 番目に、プログラムするのに使用されている ProBuilder の指示についての説明、3 番目に ProBacktest を使用してトレーディングシステムをバックテストする方法についての説明、そして 4 番目はトレーディングシステムを自動的に実行する方法についての説明です。巻末の付録には、トレーディングシステムの結果がどのように表示されるかを解説、またいくつかのプログラム例と、ProBuilder 言語の用語解説が掲載されています。

このプログラミングガイドに記載されている内容は、トレーディングシステムの書き方を学び作成した計画をテストすることをサポートするためのものです。いかなる場合においても投資のアドバイスではありませんのでご注意ください。

皆様のトレーディングの成功を祈りつつ、このプログラミングガイドがお役に立てることを願っています。

## はじめに

### トレーディングシステム・プログラミングへのアクセス

トレーディングシステム作成のためのゾーンは、ワークステーションのすべてのチャートの右上にある「インジケーターとトレーディングシステムウインドウ」からアクセスすることができます。



デフォルトでは、「指標タブ」が選択されています。2番目の「バックテスト&自動取引」タブを選択して下さい。ここでは以下の作業を行うことができます：

- 既存トレーディングシステム一覧へのアクセス(パーソナルまたは定義済み)
- 新規トレーディングシステムの作成または既存システムの証券への適用
- 既存トレーディングシステムの変更または削除
- トレーディングシステムのインポートまたはエクスポート

## トレーディングシステム作成ウインドウ

トレーディングシステム作成ウインドウは、2つの主要なゾーンで構成されています：

- 「バックテスト&自動取引」→「新規」の順でクリックするとストラテジーの作成ゾーンが開きます。
- 作成ゾーン(作成アシストまたはプログラミングによる作成)は、左側に表示されます。

- 戦略実行ゾーンは右側に表示されます。過去データを使ってトレーディングシステムをバックテストする ProBacktest タブと、自動的にトレーディングシステムを実行するオートトレーディングタブが含まれます。ProBacktest タブのオプションに関しては、本書の第3項をご覧ください。

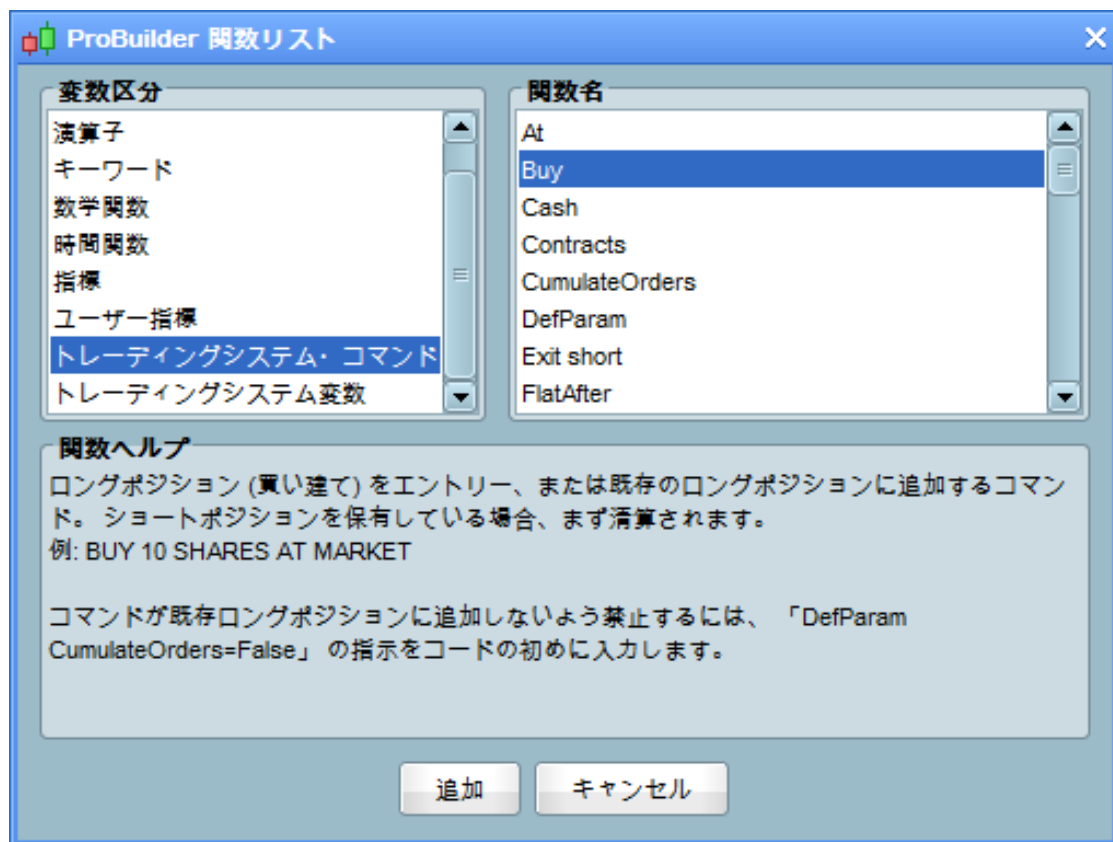
作成ゾーンでは、このようなことが可能です：

- テキストエディタを使用してトレーディングシステムをプログラムすることができます。
- 「関数挿入」ボタンを押すと新しいウインドウが開き、別々のカテゴリに分けられた ProBuilder のリストとトレーディングシステムコマンドが表示されます。ここでは、プログラミングの状況に応じたヘルプが参照できます。ウインドウの下の方に、選択したコマンドまたは関数に関連したヘルプテキストが表示されます。

例:

「関数挿入」をクリックし、関数ライブラリを使用してみましょう。

「トレーディングシステムコマンド」セクションを選択します。「Buy」を選択し、「追加」をクリックします。コマンドはプログラムに追加されます。



今度は、コードを一行丸ごと書いてみましょう。まず、市場で 10 株購入してみます。

関数「SHARES」、「AT」、「MARKET」(スペースで各々の語を切り離す)を選択し、「BUY」と「SHARES」の間に購入する数(10)を指定してください。

すると、市場で 10 株、10 ロットまたは 10 コントラクトを買うべく「BUY 10 SHARES AT MARKET」という指示が出されます。次に、プログラムトレーディングシステムに利用できるすべての指示が提示されます。

文末の付録 B では、トレーディングシステムの事例を完全な形でご参照いただけます。



## トレーディングシステムのプログラミング

### キーボードショートカット

ProRealTime バージョン 10 からは、トレーディングシステム作成ウィンドウに、キーボードショートカットからアクセスできるいくつかの便利な機能が新しく備わっています：

- すべて選択 (Ctrl + A) : プログラミングウィンドウ内のすべてのテキストを選択する
- コピー (Ctrl + C) : 選択したテキストをコピーする
- 貼り付け (Ctrl + V) : コピーしたテキストを貼り付ける
- 元に戻る (Ctrl + Z) : プログラミングウィンドウで行った最後のアクションの前に戻る
- やり直し (Ctrl + Y) : プログラミングウィンドウで行った最後のアクションをやり直す
- 検索と置換 (Ctrl + F) : プログラミングウィンドウ内のテキストを検索またはプログラミングウィンドウ内のテキストを置き換える (この機能は文字と小文字を区別します)
- コメント化と非コメント化 (Ctrl + R) : 選択したコードをコメント化または非コメント化する (コメント化されたコードは「 // 」または「REM」で始まり、グレー色の文字で表記されます。コードが実行される際には無視されます)。

Mac の場合、「Ctrl」キーを「Apple」キーで置き換えることで同じショートカットが使用できます。トレーディングシステム作成ウィンドウのプログラミングゾーンを右クリックすることで、ここにあるほとんどの機能にアクセスすることが可能です。

## トレーディングシステムのプログラミング

### ProBuilder 言語でのプログラミング

この項では、トレーディングシステム特有の ProBuilder コマンドについて詳しく説明します。

ProBuilder はワークステーションで使用されるプログラミング言語です。非常に使いやすく、これを使用して多くのことを行うことができます。以下、ProBuilder の重要な原則です：

- 各バーの終わりに計算が行われます。トレーディングシステムとすべての関数を含む ProBuilder プログラムは、最初から最後まで各バーの終わりに算出されます。
- 進行中のローソク足チャートの計算終了後、すべての注文指示が出されます(注文は最速で次のローソク足の始値で実行されます)。

### 市場エントリーとエグジット

ポジションのタイプによって、異なる指示が使用されます：

#### ロングポジション:

- BUY は、ロングポジションエントリー指示(買い)
- SELL はロングポジションエグジット指示(売り)

BUY では市場のロングポジションをオープン(保有)する、または既存オープンポジションに加えることができます。これは、ポジションをクローズまたは部分的にクローズするための指示である SELL と関連しています。オープンのロングポジションが存在しない場合、SELL の指示は何も意味を持ちません。

#### ショートポジション:

- SELLSHORT は、ショートポジションエントリー指示(空売り)
- EXITSHORT は、ショートポジションエグジット指示(買戻し)

これら2つの指示は、「BUY」と「SELL」と類似した働きをします。オープンのショートポジションが存在しない場合、EXITSHORT の指示は何も意味を持ちません。

1つの証券に対してロングとショートのポジションを同時に扱うことはできません。実際には、SELLSHORT コマンドでロングポジションをクローズしたり、または BUY コマンドでショートポジションをクローズしたりすることが可能であることも意味しています。

**注意：** ProBacktest の「リスク管理」または自動トレーディングシステムの ProOrder で定められる最大ポジションサイズを確認してください。許容最大ポジションサイズを超過して注文を出そうとすると注文は拒絶され、当初のポジションが維持されることになります。

これらの各コマンドは、以下に示す通り数量と指示の内容に従います：

＜注文＞＜数量＞ AT ＜モード＞

例：

BUY 1000 CASH AT MARKET または SELL 1 SHARE AT 1.56 LIMIT

## 数量

数量を定義する 2 つの方法があります：

- SHARES は証券を表します。「1 Share」は 1 株、1 先物コントラクト、または 1 外国為替コントラクトを意味します。SHARES は、どんな種類の銘柄にでも使用可能で、SHARE、CONTRACT、CONTRACTS、LOT または LOTS に置き換えることができます。外国為替の場合、購入数量は 1 ロットのサイズで乗算されます。数量が指定されない場合、以下のデフォルト値が使用されます：
  - ポジションエントリーの 1 単位（例：BUY AT MARKET、市場価格で「1」買う）
  - エグジットするポジションの全体数量（例：SELL AT MARKET、すべてのロングポジションを売る）
- CASH はキャッシュの量を表し(€や\$など)、この指示は株の売買にのみ使用できます。注文の数量はバーの終値で計算され、デフォルトで端数処理されます。キャッシュでの売買における数量を計算する場合、売買委託手数料は考慮されません。
  - 例：BUY 1000 CASH AT MARKET
  - ROUNDEDUP という指示は、数量の切り捨てではなく、切り上げに使用されます。
  - 例：BUY 1000 CASH ROUNDEDUP AT MARKET

## モード

この種類の注文には、3 つのモードが使用できます：

AT MARKET：次のバーの始値の市場価格で注文が出されます

例：BUY 1 SHARE AT MARKET

AT <price> LIMIT：示唆された価格で指値注文が出されます

AT <price> STOP：示唆された価格での逆指値注文が出されます

例：BUY 1 SHARE AT 10.5 LIMIT

**特定のレベルの指値及び逆指値注文は、デフォルトで 1 つのバーに適用され、次のバーの始値で始まります。実行されない場合はキャンセルされます。**

これらの注文はオープンポジションにリンクし、ポジションのクローズまで有効なプロテクティブストップや利食い注文（次項参照）とは異なります。

以下の条件において、注文を成り行き注文として処理することができます：

- 市場価格を上回る価格で買いの指値注文をする場合、注文は成り行き注文として処理。
- 市場価格を下回る価格で買いの逆指値注文をする場合、注文は成り行き注文として処理。
- 市場価格を下回る価格で空売りの指値注文をする場合、注文は成り行き注文として処理。

## トレーディングシステムのプログラミング

- 市場価格を上回る価格で空売りの逆指値注文をする場合、注文は成り行き注文として処理。

## トレーディングシステムのプログラミング

例：

以下のプログラムは、RSI が売られすぎで (RSI < 30)、且つ価格がロウワーボリンジャーバンドの下にあるならば、市価で 1 つの株を買います。また、RSI が買われすぎで (RSI > 70)、且つ価格がアッパーボリンジャーバンドより上にあるならば、売ります。

```
MyRSI = RSI[14](Close)
MyBollingerDown = BollingerDown[25](Close)
MyBollingerUp = BollingerUp[25](Close)

IF MyRSI < 30 AND Close < MyBollingerDown THEN BUY
    1 SHARE AT MARKET
ENDIF

IF MyRSI > 70 AND Close > MyBollingerUp THEN
    SELL AT MARKET
ENDIF
```

指値及び逆指値注文の有効期限を設定することができます。

以下の例は、変数を用いて特定の数のバーに設定された有効期間を持つ指値注文をどのように作成するかを示しています。コードは、移動平均線クロスオーバーが起こったバーの終値で買いの指値注文を行います。この制限は、クロスオーバーが起きたバーの後の 10 本のバーに対して有効です。もしこれら 10 本のバーで実行されなかった場合、キャンセルされます。

例：

```
// 注文の有効期間の定義づけ
ONCE NbBarLimit=10

MM20=Average[20](close)
MM50=Average[50](close)

// MM20 が MM50 にクロスオーバーする場合、その時点での終値とクロスバーインデックスを含む 2 つの変数「MyLimitBuy」
と「MyIndex」を定めます。
IF MM20 CROSSES OVER MM50 THEN
    MyLimitBuy = close
    MyIndex = Barindex
ENDIF

IF BarIndex >= MyIndex + NbBarLimit THEN
    MyLimitBuy = 0
ENDIF
```

## トレーディングシステムのプログラミング

```
// この変数が0を超え、且つロングポジションを持っていない場合、有効な価格 MyLimitBuy で注文。
```

```
// 注釈: MyLimitBuy は、交差のバーの後、10 のバーで 0 より大きくなります。
```

```
IF MyLimitBuy > 0 AND NOT LongOnMarket THEN
```

```
    BUY 1 SHARES AT MyLimitBuy LIMIT
```

```
ENDIF
```

注文が実行されない場合、失効した買い指値注文を市場価格での注文に置き換えることができます。これは、以下のコードを前のコードに追加することで実行できます。

```
IF MyIndex + NbBarLimit AND MyLimitBuy > 0 AND NOT LongOnMarket THEN BUY 1
```

```
    SHARES AT MARKET
```

```
ENDIF
```

## ストップと目標値

ProBuilderを使用して、利益目標とプロテクティブストップを定義づけることができます。構文は以下の通りです：

**SET STOP <種類> <値> または SET TARGET <種類> <値>**

例：

```
SET STOP %LOSS 2 // 2%のストップを設定してください。
```

各指示の詳細は、次の節の説明をご参照ください。

STOP コマンドの違いに注意してください：

**AT <価格> STOP** は、ポジションのエントリーに使用されます。この注文は、デフォルトで1つのバーに適用されます。

**SET STOP LOSS <価格>** はプロテクティブストップで、ポジションのエグジットに使用されます。この注文は、ポジションがクローズされるまで有効です。

## プロテクティブストップ

プロテクティブストップは、ポジションの損失を制限します。相対項または絶対項で定義づけることができます。

**SET STOP LOSS x**：ポジション x 単位をエントリー価格からクローズするためのストップロスの設定

**SET STOP pLOSS x**：ポジション x ポイントをエントリー価格からクローズするためのストップロスの設定

**SET STOP %LOSS x**：売買委託手数料を除く損失が x%に達した際の、ポジションをクローズするためのストップロスの設定

**SET STOP \$LOSS x**：売買委託手数料を除く X €, \$ (銘柄の通貨)のポジションをクローズするためのストップロスの設定

## トレーディングシステムのプログラミング

プロテクティブストップ注文の数量と指示（ロングポジションをエグジットまたはショートポジションをエグジット）は、自動的に現在オープンしているポジションに適用されます。プロテクティブストップは、ポジションとリンクしています。オープンしているポジションがない場合、ストップロスは無効にはなりません。

ストップロスをコードによってキャンセルする場合は、以下の指示で行うことができます：

**SET STOP LOSS 0, SET STOP pLOSS 0, SET STOP %LOSS 0, SET STOP \$LOSS 0**

### 目標利益の設定

これらの指示は、特定の価格に達したらポジションをエグジットするようにするものです。

**SET TARGET PROFIT x**：ポジション x 単位をエントリー価格からクローズするための目標利益の設定。

**SET TARGET pPROFIT x**：ポジション x ポイントをエントリー価格からクローズするための目標利益の設定。

**SET TARGET %PROFIT x**：売買委託手数料を除く利益が x%に達した際の、ポジションをクローズするための目標利益の設定。

**SET TARGET \$PROFIT x**：売買委託手数料を除く利益が x €, \$ (銘柄の通貨)に達した際の、ポジションをクローズするための利益目標の設定。

利益目標注文の数量と指示（ロングポジションをエグジットまたは ショートポジションをエグジット）は、自動的に現在オープンしているポジションに適用されます。すべての利益目標は、ポジションとリンクしています。オープンしているポジションがない場合は、利益目標注文はアクティブにはなりません。

利益目標をコードによってキャンセルする場合は、以下の指示で行うことができます。

**SET TARGET PROFIT 0, SET TARGET pPROFIT 0, SET TARGET %PROFIT 0, SET TARGET \$PROFIT 0**

### トレーリングストップ

トレーリングストップとは価格が価格の進化に従い変動する逆指値注文です。ロングポジションでは、価格が上昇するとトレーリングストップのレベルも上昇します。しかし、価格が下降してもトレーリングストップのレベルは維持されます。ショートポジションの場合、これとは逆になります。価格が下降すると、トレーリングストップのレベルも下降しますが、価格が上昇してもトレーリングストップのレベルは維持されます。

プロテクティブストップ同様、トレーリングストップは以下の様に相対項または絶対項で定義づけることができます：

**SET STOP TRAILING y**：ポジションエントリー価格からのトレーリングストップ y 単位の設定。

**SET STOP pTRAILING y**：ポジションエントリー価格からのトレーリングストップ y ポイントの設定。

**SET STOP %TRAILING y**：売買委託手数料を除くポジション エントリー価格からのトレーリングストップ y% の設定。

**SET STOP \$TRAILING y**：売買委託手数料を除くポジション エントリー価格からのトレーリングストップ y €, \$ (銘柄の通貨) の設定

トレーリングストップ注文の数量と指示（ロングポジションをエグジットまたはショートポジションをエグジット）は、自動的に現在オープンしているポジションに適用されます。すべてのトレーリングストップは、ポジションとリンクしています。オープンしているポジションがない場合、トレーリングストップはアクティブにはなりません。

ポジションの数量が変化すると、ストップのレベルは初期化されます。

トレーリングストップをコードによってキャンセルする場合は、以下の指示で行うことができます：

**SET STOP TRAILING 0, SET STOP pTRAILING 0, SET STOP %TRAILING 0, SET STOP \$TRAILING 0**

例：

DAX でロングポジションが 6000 ポイント、及びトレーリングストップが 50 ポイントの場合

**SET STOP pTRAILING 50**

ストップは初めの時点で 5950 とします。価格が 6010 にまで上昇した後 5980 に減少し、ストップは 5960 まで 10 ポイント増加し、価格が 6010 を超過するまでそこに留まります。価格が 5960 に達するとトリガーとなります。

### 「IF」条件文を使用した「目標設定」と「ストップ設定」の使用

IF 条件文を用いて、設定した条件に従いコードによって目標またはストップの種類を変えることが可能です。

例：

REM 前のトレードの利益が少なくとも 10%だった場合は 10%のストップロスを使用し、そうでない場合は 5%のストップロスを使用。



## トレーディングシステムのプログラミング

```
IF PositionPerf(1) > 0.1 THEN  
    SET STOP %LOSS 10  
Else  
    SET STOP %LOSS 5  
ENDIF
```

### 複数のストップと目標レベル

通常の状態では、「ストップ設定」と「目標設定」はコマンド 1 つのみをアクティブにすることができます。連続した「ストップ設定」または「目標設定」コマンドがコードに存在する場合、最後のコマンドが前のコマンドに取って代わります。

例：

SET STOP %LOSS 10 // 10%のストップロスを設定

SET TARGET PROFIT 50 // 50 単位の利益目標を設定

SET TARGET %PROFIT 5 // 50 単位の前の目標を削除し、5%の利益目標に置換

SET STOP %TRAILING 2 // 10%のストップロスを削除し、2%のトレーリングストップを設定

しかしながら、以下のように1つの指示で修正ストップとトレーリングストップ、または損失ストップとトレーリングストップを結合することは可能です。

SET STOP	<u>    &lt;モード&gt; &lt;値&gt;    </u>	<u>    &lt;トレーリングの種類&gt; &lt;値&gt;    </u>
	fixed(修正)	trailing(トレーリング)

モード：Loss, pLOSS, %LOSS, \$LOSS

トレーリングの種類：TRAILING, pTRAILING, %TRAILING, \$TRAILING

この指示は、以下のように表示されます：

SET STOP [LOSS/pLOSS/\$LOSS/%LOSS] <値> [TRAILING/pTRAILING/\$TRAILING/%TRAILING] <値>

### 使用例:

SET STOP LOSS x TRAILING y：ストップロスがポジションエントリー価格から x 単位のところに置かれ、トレーリングストップレベルがストップロスレベルよりも現在価格に近づくと、y 単位のトレーリングストップとなる（これは現在価格がポジションエントリー価格 + y - x を超えると起きる）。

SET STOP LOSS x pTRAILING y：ストップロスがポジションエントリー価格から x 単位のところで置かれ、トレーリングストップレベルがストップロスレベルよりも現在価格に近づくと、y ポイントのトレーリングストップとなる（これは現在価格がポジションエントリー価格 + y ポイント - x 単位を超えると起きる）。

SET STOP LOSS x \$TRAILING y：ストップロスがポジションエントリー価格から x 単位のところで置かれ、トレーリングストップレベルがストップロスレベルよりも現在価格に近づくと、y \$ または €（銘柄の通貨）トレーリングストップとなる。

SET STOP LOSS x %TRAILING y：ストップロスがポジションエントリー価格から x 単位のところで置かれ、トレーリングストップレベルがストップロスレベルよりも現在価格に近づくと、y%のトレーリングストップとなる。

**SET STOP pLOSS x TRAILING y** : ストップロスがポジションエントリー価格から  $x$  ポイントのところであられ、トレーリングストップレベルがストップロスレベルよりも現在価格に近づくと、 $y$  単位のトレーリングストップとなる(これは現在価格がポジションエントリー価格+  $y$  単位-  $x$  ポイントを超えると起きる)。

**SET STOP pLOSS x pTRAILING y** : ストップロスがポジションエントリー価格から  $x$  ポイントのところであられ、トレーリングストップレベルがストップロスレベルよりも現在価格に近づくと、 $y$  ポイントのトレーリングストップとなる(これは現在価格がポジションエントリー価格 +  $y$  ポイント -  $x$  ポイントを超えると起きる)。

**SET STOP pLOSS x \$TRAILING y** : ストップロスがポジションエントリー価格から  $x$  ポイントのところであられ、トレーリングストップレベルがストップロスレベルよりも現在価格に近づくと、 $y$  \$ または € (銘柄の通貨)のトレーリングストップとなる。

**SET STOP pLOSS x %TRAILING y** : ストップロスがポジションエントリー価格から  $x$  ポイントのところであられ、トレーリングストップレベルがストップロスレベルよりも現在価格に近づくと、 $y\%$  のトレーリングストップとなる。

**SET STOP \$LOSS x TRAILING y** :  $x$  \$ または € (銘柄の通貨) のストップロスが置かれ、トレーリングストップレベルがストップロスレベルよりも現在価格に近づくと、 $y$  単位のトレーリングストップとなる。

**SET STOP \$LOSS x pTRAILING y** :  $x$  \$ または € (銘柄の通貨) のストップロスが置かれ、トレーリングストップレベルがストップロスレベルよりも現在価格に近づくと、 $y$  ポイントのトレーリングストップとなる。

**SET STOP \$LOSS x \$TRAILING y** :  $x$  \$ または € (銘柄の通貨) のストップロスが置かれ、トレーリングストップレベルがストップロスレベルよりも現在価格に近づくと、 $y$  \$ または € (銘柄の通貨)のトレーリングストップとなる。

**SET STOP \$LOSS x %TRAILING y** :  $x$  \$ または € (銘柄の通貨) のストップロスが置かれ、トレーリングストップレベルがストップロスレベルよりも現在価格に近づくと、 $y\%$ のトレーリングストップとなる。

**SET STOP %LOSS x TRAILING y** :  $x\%$ のストップロスが置かれ、トレーリングストップレベルがストップロスレベルよりも現在価格に近づくと、 $y$  単位のトレーリングストップとなる。

**SET STOP %LOSS x pTRAILING y** :  $x\%$ のストップロスが置かれ、トレーリングストップレベルがストップロスレベルよりも現在価格に近づくと、 $y$  ポイントのトレーリングストップとなる。

**SET STOP %LOSS x \$TRAILING y** :  $x\%$ のストップロスが置かれ、トレーリングストップレベルがストップロスレベルよりも現在価格に近づくと、 $y$ \$ または € (銘柄の通貨)のトレーリングストップとなる。

**SET STOP %LOSS x %TRAILING y** :  $x\%$ のストップロスが置かれ、トレーリングストップレベルがストップロスレベルよりも現在価格に近づくと、 $y\%$ のトレーリングストップとなる。

例：

SET STOP LOSS x TRAILING y：

ストップがポジションエントリー価格から x 単位のところで置かれ、トレーリングストップレベルがストップロスレベルよりも現在価格に近づくと、y 単位のトレーリングストップとなる。

例えば、DAX 先物に 6500 でロングポジションをエントリーした場合、以下のコードによってポジションエントリー価格からストップロス 20 単位が置かれ、もし価格が 6530 を超えた場合、50 単位のトレーリングストップとなります。

SET STOP LOSS 20 TRAILING 50

以下の図でこの例を説明します：

初回ストップがポジションの始値(6480)の下に固定レベル 20 単位で置かれる



価格が 6530 (=6500 + (50-20)) に達したときにのみ、ストップは 50 単位のトレーリングストップとなります。

## トレーディングシステムのプログラミング



もし、上図のように価格が 6535 まで上昇した場合、トレーリングストップは 6485 に上昇します。

### トレーディングシステムを QUIT でストップする

「QUIT」の指示で、トレーディングシステムをストップすることができます。現在のバーの後に、ストップが実行されます。待機中の注文はキャンセルされ、すべてのオープン(保有)ポジションはクローズされます。高い損失の場合や以下の例にあるように特定の日付の後のストップが可能です。

例：

```
If date > 20130101 THEN // 2013 年 1 月 1 日以降、戦略をストップ  
    QUIT  
ENDIF
```

### ポジショントラッキング

#### ポジションステータス変数

これら3つの変数によって、トレーディングシステムのポジションの状態を確認することができます：

**ONMARKET:** オープンポジションを持っていたら 1、持っていなければ 0

**LONGONMARKET:** ロングポジションを持っていたら 1、持っていなければ 0

**SHORTONMARKET:** ショートポジションを持っていたら 1、持っていなければ 0

これらはカッコとともに使用します。例えば、オープンポジションを前のバーの終値で持っていたならば ONMARKET[1] は 1 で、持っていなければ 0 です。

こうした変数は、通常ポジションの入力前に IF コマンドで挿入されます：

例：

```
REM MACD を定義づけ  
Indicator1= MACD[12,26,9](Close)  
REM MACD ヒストグラムの交差を観察  
c1 = (Indicator1 Crosses Over 0)  
REM 買い: ロングポジションが存在せず、MACD > 0の場合、3 ロット購入  
IF NOT LONGONMARKET AND c1 THEN  
    BUY 3 SHARES AT MARKET  
ENDIF
```

### ポジション変数のサイズ

これら 3 つの変数によって、オープンポジションの数を知ることができます。

**COUNTOFPOSITION** : ポジションのサイズ(ロット、株、コントラクト)。オープンのロングポジションが存在するならば正の値、オープンのショートポジションが存在するならば負の値となります。

**COUNTOFLONGSHARES** : ロングポジションのサイズ(ロット、株、コントラクト)。オープンのロングポジションが存在するならば正の値、存在しなければ 0 です。

**COUNTOFSHORTSHARES** : ショートポジションのサイズ(ロット、株、コントラクト)。オープンのショートポジションが存在するならば正の値、存在しなければ 0 です。

こうした変数は、通常ポジションの入力前に IF コマンドで挿入されます。

ステータス変数使用のヒント:コードは各々のバーの終わりで評価され、注文は次のバーの始値で出されます。

例えば、以下のコードは最初のローソク足の終値において変数「long」は 1 とはならず、2 番目のローソク足の終値で 1 となっています。なぜなら、最初の買い注文は 2 番目のバーの始値で出されているからです。

```
BUY 1 SHARE AT MARKET
IF NOT LONGONMARKET THEN
    long=1
ENDIF
```

### TradeIndex(トレードインデックス)

TRADEINDEX(n) のコマンドで、n 番目の実行済み注文のバーインデックスにアクセスすることができます。

#### TRADEINDEX(過去の n 番目の注文)

**注意:** TradeIndex はカッコに数を入れずに使うことができます。この場合、プログラムは最後に実行された注文のバーを考慮します: TradeIndex=TradeIndex(1).

TradeIndex は通常 BarIndex(バーインデックス)と併用します。.

**例 :**

REM : ロングポジションが少なくとも 3 つのバーの間オープンなら、クローズする

```
IF LONGONMARKET AND (BarIndex - TradeIndex) >= 3 THEN SELL
    AT MARKET
ENDIF
```

### TradePrice (トレード価格)

TRADEPRICE(n)のコマンドは、実行済み取引の価格を表示します。構文は以下の通りです:

#### TRADEPRICE(過去の n 番目の注文)

n が指定されない場合、最後に実行された注文の価格が参照されます。 TradePrice=TradePrice(1)

## トレーディングシステムのプログラミング

**例：**

REM：価格が前の注文よりプラス 2%大きい場合、ロングポジションをクローズする

IF LONGONMARKET AND CLOSE > 1.02\*TRADEPRICE THEN

    SELL AT MARKET

ENDIF



## PositionPerf(ポジションパフォーマンス)

POSITIONPERF(n)の指示で、以下のことを確認することができます:

- $n > 0$  (売買委託手数料を除く) の場合、最後から  $n$  番目のクローズされたポジションのパフォーマンス%
- $n = 0$  (売買委託手数料を除く) の場合、現在のオープンポジションのパフォーマンス%

構文は以下の通りです:

### POSITIONPERF(過去の $n$ 番目のポジション)

$n$  が指定されない場合、 $n = 0$  とします。PositionPerf=PositionPerf(0)

例:

REM 前のトレードで少なくとも 20% の利益を得た場合、買い

IF NOTONMARKET AND PositionPerf(1) > 0.2 THEN

BUY 1000 CASH AT MARKET

ENDIF

## PositionPrice(ポジション価格)

PositionPrice のコマンドで、現在のオープンポジションの平均購入価格を見ることができます。

### POSITIONPRICE

各注文数量によって増加するすべてのエントリー価格の合計金額として計算されます。ポジションを追加することでのみ、ポジション価格の値は変更されます。

この指示は、オフセットを導入するためにカッコを使用します: POSITIONPRICE[1]は前のバーの終値のポジション価格が表示されます。

例:

5 € で1株買い、価格が 10€の時に再度同じ株を買い、価格が 15€の時に同じ株を買った場合、ポジション価格は、 $(5 + 10 + 15)/3 = 10$  €となります。

さらに 20 € で1株売った場合でも、ポジション価格は 10 € のままです(変化なし)。

## StrategyProfit(戦略的利益)

このコマンドは、トレーディングシステム使用開始からの損益の実績を表示します(売買委託手数料を除く絶対値、銘柄の通貨)。損失が大きくなってしまったトレーディングシステムを終了する「QUIT」とよく併用されます。

### STRATEGYPROFIT

この指示には、カッコが使用されます: StrategyProfit[1] は、前のバーの終値における利益を表示しています。

例:

IF STRATEGYPROFIT < -500 THEN

QUIT

ENDIF

**注意：**

トレーディングシステムは、バーの終値で評価されます。上記の例では、一つのローソク足内の大きな損失またはギャップの場合 500 €以上となります。

結果、500 € の損失の後にトレーディングシステムを終了したいユーザーは、事前に STOPLOSS を設定してまずは損失を制限し、その上でシステムを終了するコードを使用します。

## トレーディングシステム実行パラメータの定義

付加的パラメータは、DEFPARAM の指示で定義づけることができます。

### Cumulate Orders(累積注文)

変数「CumulateOrders」は、市場にエントリーしたりポジションに加えたりするためにユーザーが注文を累積することを許可または禁止します。このパラメータは、プログラミングによって作成されるコードではデフォルトで「真」に設定されます。つまりトレーディングシステムは、ポジションにエントリーする条件が「真」であるすべてのバーの既存ポジションに追加することができることを意味しています。同時に、市場にアクティブな状態でエントリーするために、指値/逆指値注文を複数にすることも可能です。

オープンポジションのサイズを増加させる戦略を止めるためには、以下の指示をコードの始めに置く必要があります：

```
DEFPARAM CumulateOrders = False
```

DefParam の指示は、トレーディングシステム実行中有効性を保ちます。注文を累積するためやトレーディングシステム実行中ではない時に、トレーディングシステムの設定を変更することはできません。

例:

```
// このコードは 1 つのバーから 1 株を、最大 3 つまで購入
```

```
DEFPARAM CumulateOrders = True
```

```
If CountOfPosition < 3 then
```

```
    Buy 1 shares at market
```

```
Endif
```

```
// このコードは 1 株を 2 株の価格で、追加の株は 3 株の価格で購入
```

```
DEFPARAM CumulateOrders = True
```

```
If CountOfPosition < 2 then
```

```
    Buy 1 shares at 2 Limit
```

```
    Buy 1 shares at 3 Limit
```

```
Endif
```

```
// このコードは 1 度だけ 5 株購入
```

```
DEFPARAM CumulateOrders = False Buy
```

```
5 shares at market
```

CumulateOrders のパラメータが「偽」にセットされていても、市場を同じ方向にエグジットするために複数注文することができます。

例:

```
// このコードは、5 株を購入。価格が 40 日移動平均の下で交差したら、最高 3 株まで売却。10%の損失の場合、すべての株を売却。
```

```
DEFPARAM CumulateOrders = False
```

```
Buy 5 shares at market
```

```
If price CROSSES UNDER Average[40] THEN
```

```
    SELL 3 SHARES AT MARKET
```

```
    Set stop %Trailing 10
```

```
Endif
```

**CumulateOrders がアクティブな時のストップと目標レベルに関する注意：**

累積注文がアクティブな状態でストップロス、トレーリングストップ、または利益目標を設定するためにこの指示を使用する場合、レベルはポジション平均エントリー価格に基づいて計算され、さらにポジションの数量変更の度に再計算されます。

例:

\$10.00 で1株買い、10%のストップロスと 150%の利益目標を設定する場合の初期レベルはストップが\$9.00、目標が\$25.00 となります。\$20 で 2 番目の株を買い、平均エントリー価格が \$15 となった場合の新規レベルはストップが\$13.50、目標が\$37.50 となります(ポジション全体で)。

**作成アシストモードで作成されたコードに関する注意:** 作成アシストモードで作成されたトレーディングシステムでは、CumulateOrders はデフォルトで「偽」に設定されています(「DefParam CumulateOrders= False」の指示は、これらのコードの頭に置きます)。

## PreLoadBars(プレロードバー)

「DefParam PreLoadBars」の指示は、システムのスタート前にシステムで使用されるインジケータの計算のために、トレーディングシステム開始前にプレロードされたバーの最高値の設定を可能にします(パーソナルまたは定義済みインジケータ)。デフォルトで、このパラメータは 200 となっています。0 以下 または 5000 以上にすることはできません。プレロードされたデータをキャンセルする場合は、PreLoadBars = 0 と設定します。

プレロードされることが可能なバーの値は、所定の証券と時間枠に利用できるデータ量に依存するので、選択される値は最大となります。

例 :

```
DEFPARAM PreLoadBars = 300 a =  
(close + open) / 2  
If price CROSSES OVER Average[250](a) THEN BUY  
    1 SHARE AT MARKET  
Endif
```

PreLoadBars を 300 に設定する場合、250 のバーの移動平均はトレーディングシステム開始後の一番最初のバーによって定義づけられることを意味しています。200 のバーがプレロードされている場合は、該当しません。

最大値は 300 です: トレーディングシステムの開始前に 300 以下のバーが使用可能な場合、この使用可能なバーのみがプレロードされます。

300 のバーがプレロードされている例の場合、トレーディングシステム開始後の BarIndex における最初のバーは 300 です。一方、プレロードされているバーが 0 の場合、戦略開始後の BarIndex における最初のバーは 0 となります。

## FlatBefore(フラットビフォー) と FlatAfter(フラットアフター)

```
DEFPARAM FlatBefore = HHMMSS DEFPARAM FlatAfter = HHMMSS
```

HHMMSS は、時、分、秒を表しています。

これらの指示で、FlatBefore すなわち特定の日時前、または FlatAfter すなわち特定の日時後に、指値/逆指値注文をキャンセルし、これ以上注文をしないようにすることができます。

FlatBefore パラメータは、市場始業時間(カスタマイズの有無に関わらず)より常に後でなければならず、FlatAfter は標準的な市場終了(カスタマイズの有無に関わらず)より前でなければ影響を与えません。選択した時間がトレーディングシステムの主要時間枠の倍数ではない場合(ローソク足の真ん中で起こる)、DEFPARAM FlatAfter の指示はそのローソク足の終了で実施され、DEFPARAM FlatBefore の指示は前のローソク足の終了まで適用されます。

## トレーディングシステムのプログラミング

注文はこの期間中制限され出されません。トレーディングシステムに注文を出す許可が与えられるときでも、注文が次の期間の開始では出されないことを意味します。よって、「OnMarket」タイプの変数は、これらの時間「偽」となります。

例：

```
DEFPARAM FlatBefore = 093000 // 市場タイムゾーン 9:30:00 前は、トレーディングシステムで指値/逆指値すべてをキャンセルし、すべてのポジションをクローズし、追加の注文を阻止  
DEFPARAM FlatAfter = 160000 //市場タイムゾーン 16:00:00 以降は、トレーディングシステムで指値/逆指値すべてをキャンセルし、すべてのポジションをクローズし、追加の注文を阻止
```

### NoCashUpdate(ノーキャッシュ更新・バックテストのみ)

```
DEFPARAM NoCashUpdate = True
```

このオプションがアクティブになると、利用できるキャッシュが利益、損失、及び売買委託手数料によって更新されなくなります。デフォルトで NoCashUpdate = False(偽) となります。

例：

初期投資 10 000 €における NoCashUpdate = True(真)の場合、最大投資額はバックテスト実施中の損益に関わらず 10000€に制限されます。

注意：

DEFPARAM の指示によって定義づけられたパラメータは、コードの最初のラインで定義づけされる必要があります(コメントの後)。

## 呼出インジケーター

### ProRealTime インジケーター

ユーザー自身のインジケーターのプログラミングに利用できる ProRealTime インジケーターを含むすべての機能は、トレーディングシステムをプログラムするためにも利用できます(リスト一覧は、本書末尾の用語解説を参照してください)。

これらの機能については、ProBuilder のマニュアルを読み詳細をご確認ください。

インジケーターの計算に必要な過去データの量は、インジケーターのタイプによって異なります。

例えば、n 期間(ExponentialAverage[N])の指数移動平均を計算する場合、通常正確な結果を得るには 10\*N バーが必要であると考えます。

バックテストの開始がチャートを開始するまさに寸前であるならば、トレーディングシステムの計算のためにさらに過去データをサーバから追加し、バーの始めに結果が出るようにすることができます。

### パーソナルインジケーター

ユーザーがトレーディングシステムで「CALL」指示を使ってプログラムした ProBuilder インジケーターを呼び出すことができます。

例：

```
a,b = CALL "HistoMACD[5,6]" // a と b は関数の出力で、5 と 6 は入力
```

CALL 機能の使用については、プログラムを最適化する方法について専門的に書かれた章を注意深くお読みください。

## プログラミングに関するアドバイス

トレーディングシステムを計算するために必要な時間は、使われるインジケーターの複雑さとそれらの呼出方法に大きく依存しています。次の節で、プログラミングの観点からコードを最適化するための簡単なアドバイスをいくつかご紹介します。

### インジケーターの呼出数を減少させる

プログラムで繰り返し同じインジケーターを使う場合、再度インジケーターを呼び出すよりも媒介変数（下記の例では avg40）でインジケーターをストックしてください。これを行うことにより、実行速度を効果的に上げることができます。

非最適化コード	最適化コード
<pre>IF NOT LONGONMARKET AND close &gt; Average[40] THEN BUY 1     SHARE AT MARKET ENDIF  IF NOT SHORTONMARKET AND close &lt; Average[40] THEN     SELLSHORT 1 SHARE AT MARKET ENDIF</pre>	<pre>avg40 = Average[40] IF NOT LONGONMARKET AND close &gt; avg40 THEN BUY 1     SHARE AT MARKET ENDIF  IF NOT SHORTONMARKET AND close &lt; avg40 THEN     SELLSHORT 1 SHARE AT MARKET ENDIF</pre>

異なるオフセットで同じインジケーターを数回使う場合にも有効です。

非最適化コード	最適化コード
<pre>a = ExponentialAverage[40](close) b = ExponentialAverage[40](close[1]) c = ExponentialAverage[40](close)  IF a &gt; b THEN     BUY 1 SHARE AT MARKET ENDIF  IF a &lt; c[1] THEN     SELLSHORT 1 SHARE AT MARKET ENDIF</pre>	<pre>a = ExponentialAverage[40](close)  IF a &gt; a[1] THEN     BUY 1 SHARE AT MARKET ENDIF  IF a &lt; a[1] THEN     SELLSHORT 1 SHARE AT MARKET ENDIF</pre>

### パーソナルインジケーターの呼出

CALL 指示でパーソナルインジケーターを呼び出すと、ProRealTime インジケーターを使った場合より計算時間が長くなります。ProRealTime インジケーターでは、どんな計算が必要でそれがどのように行われるかが事前にわかります。これにより、ユーザー自らがプログラムするパーソナルインジケーターでは不可能な計算速度向上を実現します。

CALL 指示でトレーディングシステムの実行速度を改善するために、プログラムでできるだけ効率的に CALL を使うことが重要です。

### 同一の呼出の回数を制限する：

ProRealTime インジケーターでは、プログラムでインジケーターが呼び出される回数を制限することができます。

非最適化コード	最適化コード
<pre>myindic1= CALL "My Function"  IF NOT LONGONMARKET AND close &gt; myindic1 THEN BUY 1     SHARE AT MARKET ENDIF  myindic2 = CALL "My Function" IF NOT SHORTONMARKET AND close &lt; myindic2 THEN     SELLSHORT 1 SHARE AT MARKET ENDIF</pre>	<pre>myindic = CALL "My Function"  IF NOT LONGONMARKET AND close &gt; myindic THEN BUY 1     SHARE AT MARKET ENDIF  IF NOT SHORTONMARKET AND close &lt; myindic THEN     SELLSHORT 1 SHARE AT MARKET ENDIF</pre>

### ネスト化した呼出の制限：

バックテストのコードでパーソナルインジケーターを使用する場合、このパーソナルインジケーター自体のコードに CALL 指示が存在しないことを確認してください。別のパーソナルインジケーターを呼び出すパーソナルインジケーターを呼び出すことは、計算時間が長くなることがあります。可能であれば CALL 関数を使用するのではなく、常にパーソナルインジケーターのコードをコピーして ProBacktest を直接呼び出し、バックテストコードが標準の ProRealTime インジケーターのみを使用するようにした方が賢明です。

非最適化コード	最適化コード
<p>トレーディングシステムのコード：</p> <pre>myindic = CALL "MyDCLOSEMix LinearReg"  IF NOT longonmarket AND close &gt; myindic THEN BUY     1 SHARE AT MARKET ENDIF  IF NOT shortonmarket AND close &lt; myindic THEN     SELLSHORT 1 SHARE AT MARKET ENDIF</pre> <p>「MyDCLOSEMix LinearReg」のコード：</p> <pre>dayclosemix = CALL "MyDCLOSEMix" RETURN  LinearRegression[5](dayclosemix)</pre> <p>「MyDCLOSEMix」のコード：</p> <pre>mix = ( DClose(0) + 3.5* DClose(1) + 4.5* DClose(2) + 3*DClose(3) + 0.5* DClose(4) - 0.5*DClose(5) - 1.5*DClose(6) ) / 10.5</pre>	<p>トレーディングシステムのコード：</p> <pre>dayclosemix = ( DClose(0) + 3.5*DClose(1) + 4.5*DClose(2) + 3*DClose(3) + 0.5*DClose(4) - 0.5*DClose(5) - 1.5*DClose(6) ) / 10.5 myindic = LinearRegression[5](dayclosemix)  IF NOT longonmarket AND close &gt; myindic THEN BUY     1 SHARE AT MARKET ENDIF  IF NOT shortonmarket AND close &lt; myindic THEN     SELLSHORT 1 SHARE AT MARKET ENDIF</pre>



### ネスト化ループの制限：

すべての条件指示 (IF...THEN...ENDIF)において、計算時間の観点から、以下に挙げられた n 指示を使用するのではなく常に n 条件を検証する1つの条件を使用する方が賢明です。

非最適化コード	最適化コード
<pre>IF CLOSE &gt;= 0.0014 THEN   IF CLOSE &lt;= 0.0047 THEN     IF INTRADAYBARINDEX &gt;= 5 THEN       IF INTRADAYBARINDEX &lt;= 20 THEN IF         NOT SHORTONMARKET THEN           BUY 1 SHARES AT MARKET         ENDIF       ENDIF     ENDIF   ENDIF ENDIF</pre>	<pre>IF CLOSE &gt;= 0.0014 AND CLOSE &lt;= 0.0047 AND INTRADAYBARINDEX &gt;= 5 AND INTRADAYBARINDEX &lt;= 20 THEN   IF NOT SHORTONMARKET THEN BUY     1 SHARES AT MARKET   ENDIF ENDIF</pre>

### FOR ループの使用：

FOR ループの使用が必要な場合がありますが、計算時間を増長させるのでできる限り使用を制限することを推奨します。

以下は、FOR ループの使用を避けたいいくつかの例です：

// 条件 C1 が少なくとも最後から n 本のローソク足で 1 度は真であったことを確認

```
IF HIGHEST[n](c1) = 1 THEN
```

...

// 条件 c1 が最後から n 本のローソク足で常に真であったことを確認

```
IF LOWEST[n](c1) = 0 THEN
```

...

// 条件 c1 が最後から n 本のローソク足で何回検証されたか確認

```
num = SUMMATION[n](c1)
```

// c1 が真であったところからのバーの数を決定

```
IF c1 THEN
```

```
  lastoccurence = barindex
```

```
ENDIF
```

```
timesince = barindex - lastoccurence
```

// 変数最高値の検索(a,b,c,d,e,f,g)：

```
top = MAX(a, MAX(b, MAX(c, MAX(d, MAX(e, MAX(f, g))))))
```

## ProBacktest – トレーディングシステム・バックテスト

トレーディングシステム作成ウインドウの「ProBacktest」タブから、システムのパラメータを設定することができます。

ProBacktest

オートトレーディング

EUR/USD (-) - 5 分

最初の資金: 10000 \$

コストパラメーター

☐ 売買手数料: 0 \$ / 注文

☐ 最小: 0 \$ / 注文

☐ 最大: 0 \$ / 注文

☒ 証拠金: 1 %

☐ スプレッド: 2 ピップ

リスク管理

☐ 最大ポジション数: 1 取引

変数の最適化

開始日: ☒ 最近の表示日

☐ 09月 02日 2014 11:12

末日: ☒ リアルタイム日付け

☐ 09月 02日 2014 11:12

☐ 検証後もウインドウを継続する

追加

訂正

削除

ProBacktest my system

### 変数最適化

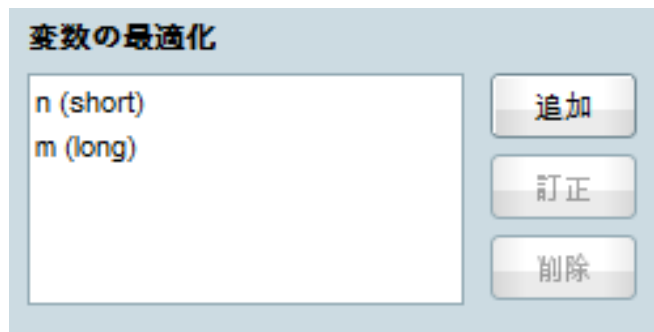
変数最適化では、バックテストで変数の異なる組合せのテストを行い、どの組合せが所定の証券とテスト対象の過去データ期間においてベストな結果をもたらすかを見ることができます。

最適化の結果は、「最適化レポート」で提示されます。トレーディングシステムでどの変数を使用するかを決定するために、変数のベストな組合せに関する統計結果を見て、実際にこの情報を使用することができます。

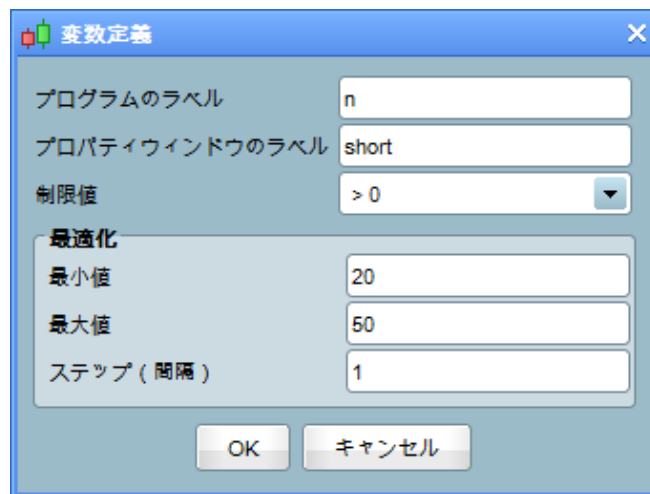
以下は、n と m の期間の 2 つの移動平均で最適化することができるプログラムの例です：

```
AVGm=ExponentialAverage[m](Close) AVGn=ExponentialAverage[n](Close) IF AVGm Crosses Over AVGn THEN  
    BUY 100 SHARES AT MARKET  
ENDIF  
  
IF AVGm Crosses Under AVGn THEN  
    SELL 100 SHARES AT MARKET  
ENDIF
```

最適化セクションの「追加」ボタンをクリックし、変数  $n$  と  $m$  を定義づけることができます：



以下のウィンドウが開き、ここから最適化のセットアップができます：



- 「プログラムのラベル」は、コードの変数名です（この場合  $n$ ）。変数名は大文字と小文字を区別します。
- 「プロパティウィンドウのラベル」は、より簡単に変数を認識するためのラベルです（例：  $n$  は「ショート」または「期間数」）。
- 「最小値」と「最大値」は、最適化テストのための変数の範囲です。
- 「ステップ(間隔)」は、最適化でテストされる変数の間隔です。

こちらは最適化レポートの例です:

レポートを最適化 - Ex Invert Position each bar						
利益	利益率	Nbrポジション	評価益ポジション比率(%)	ポジション毎の平均利益	short	long
103,100.0000	+1031.00%	36	+63.89%	2,863.8889	33	37
94,100.0000	+941.00%	36	+69.44%	2,613.8889	23	37
87,500.0000	+875.00%	27	+73.08%	3,240.7407	50	40
84,500.0000	+845.00%	41	+60.98%	2,060.9756	33	24
83,700.0000	+837.00%	36	+66.67%	2,325.0000	35	37
81,700.0000	+817.00%	44	+69.05%	1,856.8182	30	27
78,300.0000	+783.00%	41	+60.00%	1,909.7561	33	25
76,700.0000	+767.00%	35	+71.43%	2,191.4286	23	38
76,700.0000	+767.00%	27	+74.07%	2,840.7407	33	49
76,500.0000	+765.00%	27	+66.67%	2,833.3333	46	49
76,100.0000	+761.00%	27	+76.00%	2,818.5185	50	27
73,700.0000	+737.00%	27	+69.23%	2,729.6296	50	42
73,300.0000	+733.00%	35	+64.71%	2,094.2857	33	38
71,900.0000	+719.00%	30	+70.00%	2,396.6667	46	28
71,700.0000	+717.00%	27	+76.92%	2,655.5556	50	37
69,500.0000	+695.00%	41	+60.98%	1,695.1220	33	22
69,100.0000	+691.00%	39	+56.41%	1,771.7949	35	22
69,100.0000	+691.00%	58	+62.07%	1,191.3793	23	20
68,200.0000	+682.00%	27	+74.07%	2,525.9259	32	49
67,500.0000	+675.00%	30	+60.00%	2,250.0000	33	44
64,100.0000	+641.00%	39	+56.41%	1,643.5897	34	22
64,100.0000	+641.00%	41	+55.00%	1,563.4146	33	23
63,700.0000	+637.00%	30	+60.00%	2,123.3333	35	44
63,000.0000	+630.00%	27	+66.67%	2,333.3333	34	49
60,900.0000	+609.00%	40	+60.00%	1,522.5000	33	27

最適化レポートでは、1組の変数の組み合わせに関して5つの統計結果を見ることができます。これらの統計は、以下の様になります：

「利益」とは、トレーディングシステムで認識された利益と損失です。方程式は次の通りです：

$$\text{利益} = \text{最終資本} - \text{初期資本}$$

テストされた過去の期間と各変数の組合せの定義づけがなされたトレーディングシステムを使用し、この統計から絶対利益の可能性を評価することができます。

**注意：**「売買委託パラメータ」で定義づけされた売買委託手数料は、この計算では考慮されません。

「利益率(%)」は利益と損失のパーセンテージです。方程式は次の通りです：

$$\text{利益率}(\%) = 100 \times \text{利益} / \text{初期資本}$$

これは、対応する変数で構成されるバックテストの相対的パフォーマンスを表します。

「Nbr ポジション」はバックテスト中にオープンするポジション数を表します。

「評価益ポジションの比率(%)」は勝ちポジション率(%)を表します。計算は以下の通りです：

$$\text{評価益ポジションの比率}(\%) = (100 \times \text{勝ちポジション数}) / \text{ポジション数}$$

「評価益ポジションの平均利益額」はポジションの平均利益です。出される注文の効率を測定するのに便利です。以下のように定義づけできます：

$$\text{平均利益} = \text{利益} / \text{ポジション数}$$

**注意：**最適化レポートの結果は、証券、時間枠、または使用した過去データにより、所定のトレーディングシステムで異なることがあります。

## バックテスト実行期間の定義

このエリアでは、バックテストの開始と終了を定義づけします。バックテストで使用されるデータ量は、通常チャートで表示されるデータ量に限定されますのでご注意ください。各チャートの左側にあるプルダウンメニューからチャート内にロードされている過去データを増やすことが可能です。バックテスト実行前に、テストに必要なデータ量をロードする必要があります。

「リアルタイム」バックテストの場合、注文はシグナルがトリガーとなっていればいつでもチャート内に表示されます。また、「オプション」メニューにある「アラートとサウンド設定」を選択し、これらの注文をポップアップまたはサウンドと関連付けすることも可能です。

終了日が定義づけられている場合、終了日にすべてのオープンポジションをクローズします。

### 注意：

バックテストの実行が長時間に及ぶ場合、実行期間を短くすることができます。トレーディングシステムのバックテストにかかる時間は、トレーディングシステムがテストする過去データの量に比例します。

バックテスト実行後、以下の結果が表示されます：

- トレーディングシステムの利益と損失を表す資産カーブ
- ポジションのヒストグラム
- 詳細レポート

バックテストの結果表示に関する詳細は、文末の付録 A を参照してください。

## バックテストトレーディング時間のカスタマイズ

「オプション／タイムゾーンとトレーディング時間の設定」メニューから、市場のトレーディング時間をカスタマイズし定義づけることができます。

### カスタマイズされたトレーディング時間：

市場に短縮したトレーディング時間を定義づけた場合、それは短縮したトレーディング時間におけるデータのみがチャートに表示されることを意味します (ProBacktest で考慮される)。短縮したトレーディング時間の定義づけは、日をまたいで行うことができません。例えば、市場が 1 日 24 時間オープンの場合、10:00 から 14:00 の間のデータのみを考慮する選択はできますが、21:00 からその翌日の 9:00 の間のデータを考慮することはできません。注文が、カスタマイズされたトレーディング時間におけるトレーディング日の最後のバーのカスタマイズされた終値で出される場合、この注文は次のトレーディング日のカスタマイズされた始値で置かれるということになります。

### カスタマイズされたタイムゾーンと週末データに関する注意事項:

カスタマイズされたタイムゾーンはチャートでのみ考慮されます。バックテストでは考慮されません (バックテストでは常に市場の公式なタイムゾーンが使用されます)。

ある24時間市場には、「日々のローソク足チャート作成のために、日中足クオートを使う」というオプションがあります。カスタマイズされたタイムゾーン同様、このオプションは常に標準的なタイムゾーンに基づく日々の公式ローソク足チャートを使う ProBacktest では考慮されません。

ある市場（例えば外国為替）では、週末データも含まれます。こうした市場のために、ユーザーがチャートの週末データを隠すことができるチェックボックスが「オプション/タイムゾーン設定及びトレーディング時間の設定」にあります。週末データは、バックテストにおいて常に考慮されます。外国為替市場では、日曜日のデータはバックテスト用に月曜日のローソク足に含まれます（日曜日は日次のローソク足データはありません）。

## ProBacktest のストップの理由

ProBacktest は、以下のいずれかの理由でストップすることがあります：

- バックテストがプログラミングウィンドウで指定されている終了時間に到達。この場合、バックテストの終了はチャートに黒い垂直線のみで表示されます。
- 実行されるコードに「QUIT」指示が存在する。この場合、バックテストの終了は以下のアイコンによって表示されます：



- 利用できる資本が、もはや損失をカバーするのに十分ではなくなった（「予測」資本がマイナス）。この場合、バックテストの終了は以下のアイコンによって表示されます。



- 注文が不十分な現金のために拒絶された。この注文は詳細レポートの注文リストに記載されます。この場合、バックテストの終了は以下のアイコンによって表示されます。



以下はバックテストが不十分な資本のためにストップした例です：





## PROORDER 自動トレーディング

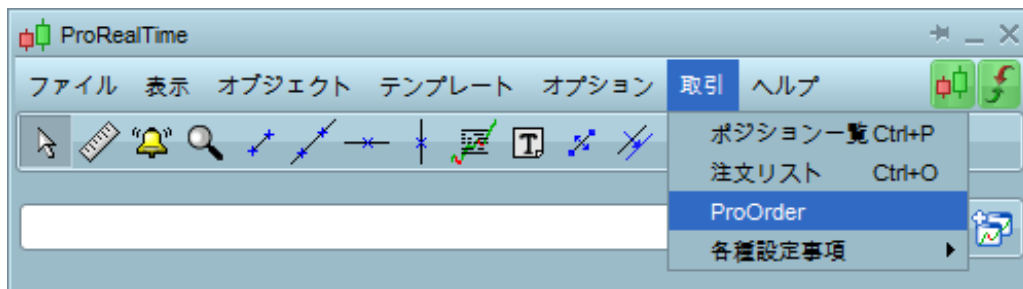
本章では、以前バックテストしたトレーディングシステムを自動トレーディングシステムとして実行する方法について説明します。

- 初めに、自動トレーディングシステム実行の準備を行うために ProOrder にトレーディングシステムを送信する方法を説明します。
- 2 番目に、トレーディングシステムを開始し、結果をチェックする方法を説明します。
- 3 番目に、トレーディングシステムのパラメータとそれらの実行条件を説明します。
- 4 番目に、説明書と自動トレーディングシステムがどのようにワークステーションで共存するかについて説明します。
- 5 番目に、同じ証券で複数の自動トレーディングシステムを実行することの意味を説明します。
- 最後は、計算方法が理由で自動トレーディングでは使用できないインジケータのリストがご覧いただけます。

トレーディングシステムを開始する前に、当プログラミングガイド全文を読みトレーディングシステムの実行方法について学習されることをお勧めします。

## トレーディングシステム自動実行のための準備


まず初めにメニューから ProOrder ウィンドウを開きます：

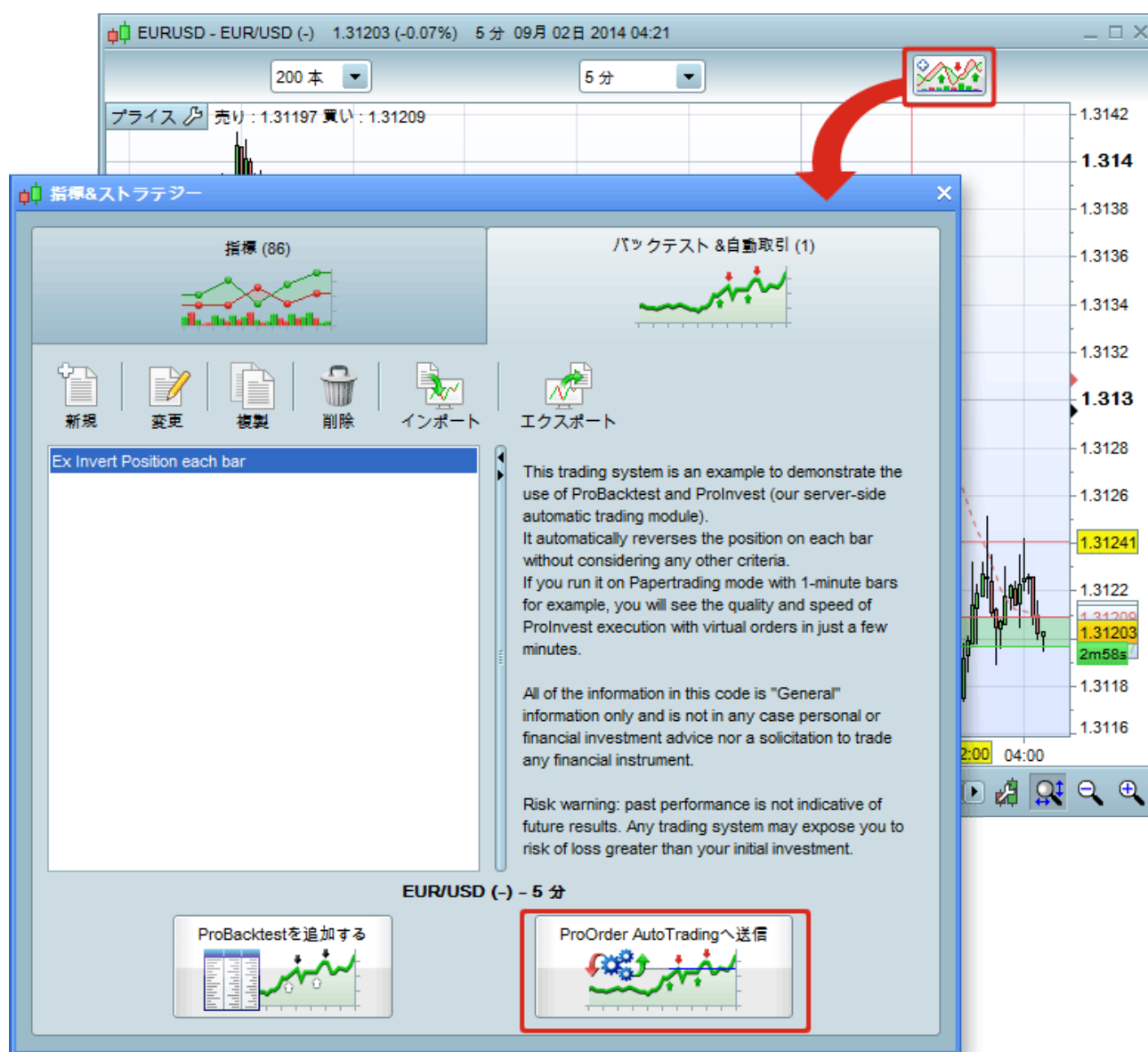


以下のように、自動トレーディングを行うトレーディングシステムを準備するための指示がウィンドウに表示されます：



## PROORDER 自動トレーディング

まず、実行したいチャートと時間枠を選択し、 ボタンをクリックします。すると、インジケータとトレーディングシステムウインドウが表示されます。トレーディングシステムのリストを見るために「バックテストと自動トレーディング」をクリックします。



自動的に実行するトレーディングシステムを選択し、「ProOrder Auto Trading へ送信」ボタンを押すと、トレーディングシステムが ProOrder に表示されます。

## トレーディングシステムのスタートと結果確認の方法

トレーディングシステムが ProOrder に追加されると、システムの最大ポジションサイズを定義づけることができます。その後「SRART」ボタンを押し、トレーディングを開始します：



ポップアップ・ウィンドウが表示され、システムを実行するかどうか尋ねてきますので、事前に注意深く説明を読んでください。

トレーディングシステムを開始する前に ProOrder ウィンドウで設定できる「最大ポジションサイズ」は、コードの数量よりも優先しますのでご注意ください。先物と外国為替の最大ポジションサイズは、ロット数またはコントラクト数に従って定義づけられます。例えば、コードが 3 ロット買うための指示を持っても、ユーザーが最大ポジションサイズを 1 に制限すると、3 ロットの買い注文は無視されます。同様に、コードが 1 ロット、次いで空売り 3 ロットを買うための指示を持っている場合、空売り注文は無視されます。結果 1 つのロングポジションを持ったままとなります。こうした理由で、コードを実行する前に最大ポジションサイズを常に確認する必要があります。

株の場合、最大ポジションサイズはキャッシュで定義づけられます(売買委託手数料を除く)。

「START」ボタンを押すと、システムは以下のように「実行中」セクションで表示されます。

## ProOrder 自動トレーディング



トレーディングシステム開始後、そのポジション、含み益及びすべての利益が ProOrder ウィンドウに表示されます。このシステムのコードのコピーを参照するには「バージョン」のリンクをクリックします。

システムの資産カーブとパフォーマンスに関する詳細レポートを参照するには、下記の赤色で示されたボタンをクリックします。

## PROORDER 自動トレーディング

以下は実行中のシステムの資産カーブとその詳細レポートの例です：



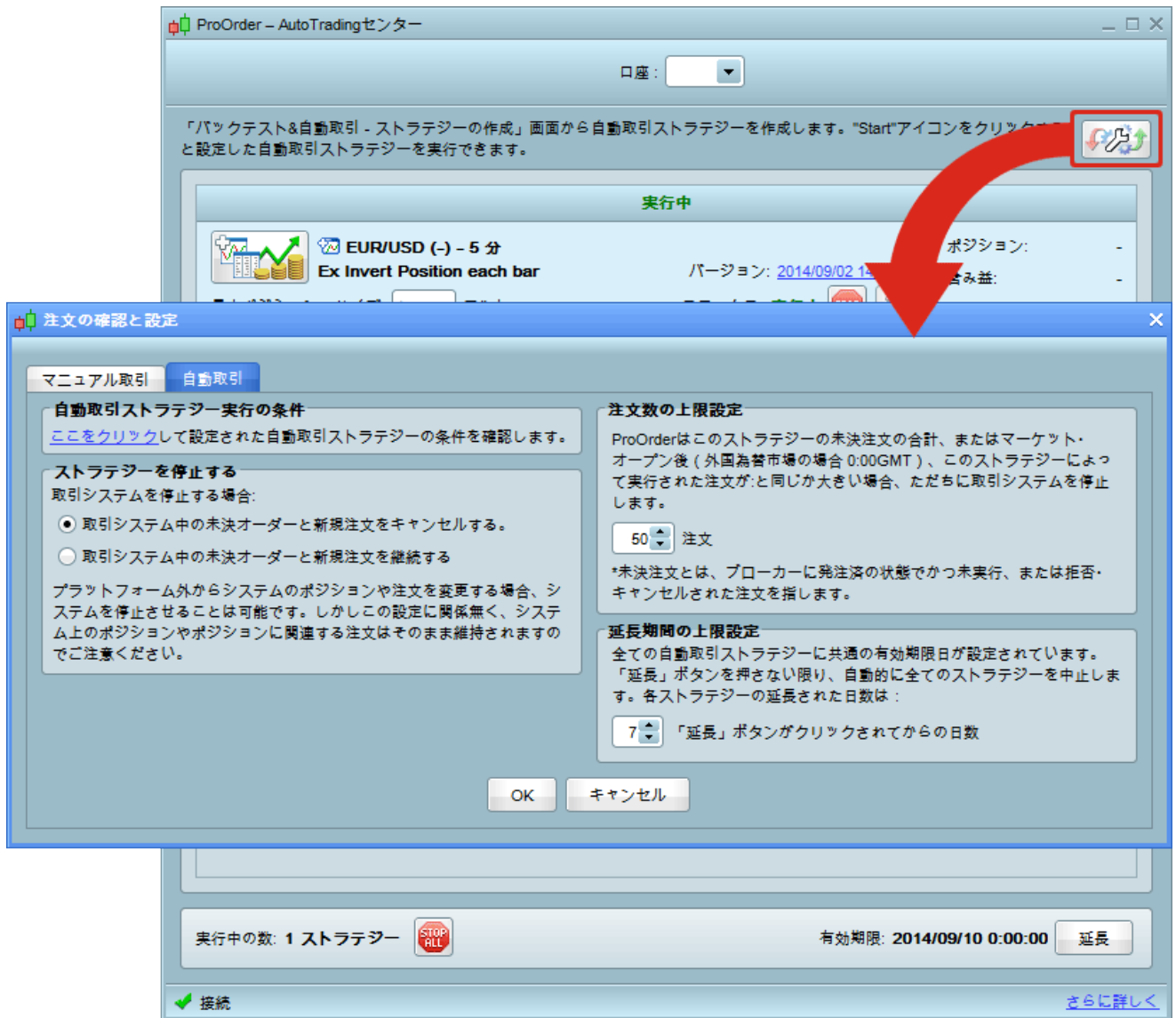


注意: システムが実行中で資産カーブがオープンなポジションを考慮するため、「クローズされたポジションの統計」セクションの利益は資産カーブの価値と異なることがあります。



## トレーディングシステムパラメータ

いずれのトレーディングシステムであっても、実行前にトレーディングプレファレンスを構成する必要があります。下記の赤/緑色で示されるレンチャアイコンをクリックしてください：



トレーディングシステム実行条件を参照するには、このウインドウの下にある「さらに詳しく」をクリックし、必ずこれらの条件をしっかりと読みください。

## トレーディングシステム自動ストップ

**有効期限**：実行中のすべてのトレーディングシステムには、共通の有効期限があります。有効期限の日までに「延長」ボタンを押さない場合、ProOrder は自動的にそれらをストップすることがあります。トレーディングシステム実行中、有効期限を ProOrder ウインドウで参照でき(コンピュータのタイムゾーンで表示されます)、ProOrder ウインドウの下にある「延長」ボタンをクリックしてトレーディングシステムの有効期間を延長することができます。

## PROORDER 自動トレーディング



各々の延長時間は、ウインドウの「自動トレーディング」タブから設定することができます。トレーディングシステムの実行中、このパラメータを増やすことができます。変更した内容は、次の延長に適用されます。

注意：プラットフォームがオープンの際にコンピュータのタイムゾーンを修正した場合、新規のタイムゾーンで「有効期限」の時間を表示するにはワークステーションを再起動する必要があります。

**注文数：** このシステムによって出される指値/逆指値注文合計、または市場オープン以来このシステムによって実行された合計注文数(外国為替市場では 0:00 GMT)が「トレーディングブリファレンス」ウインドウの「オートトレーディング」タブで選択した数量と同等または超過した場合、ProOrder は直ちに所定のトレーディングシステムをストップすることがあります。指値/逆指値注文は、仲買業者に送られたものの、実行されなかったり拒否またはキャンセルされた注文のことです。

例えば、当該の注文がキャンセルや拒否、または実行されない限り、「ストップ設定」「トレーリングストップ設定」「目標設定」の指示となります。

加えて、キャンセル、拒否、または実行されていない 3 つの異なる指値注文または逆指値注文は、3 つの待機注文として数えられます。3 つの注文が同じ価格レベルであっても異なる価格レベルであっても、適用されます。

例えば、ストップングレベルの 8 つの注文を選択し、市場がオープンしてから 5 つの注文を所定のトレーディングシステムで実行したとします。このシステムは 2 つの待機注文(「目標設定」1 つと「ストップ設定」1 つ)を持ち、システムは市場に追加注文を送る必要があります。この第 8 の注文は送られません(5+2+1 でストップレベルに到達)。このトレーディングシステムは最初にキャンセルされる待機注文で塞がれ、その後そのポジションはクローズすることになります。

トレーディングシステム実行中に、このパラメータを増やすことができます。

**注文拒否**：非常に多くの注文が拒否された場合、ProOrderは所定のトレーディングシステムをストップすることがあります。1回の注文拒否の場合、トレーディングシステムを止めたり再試行の数を選択したりすることができます。トレーディングシステム実行中、注文拒否パラメータを変更することはできません。

### ワークステーション内の手動・自動トレーディングの共存

トレーディングシステムがある証券で実行されている間、ワークステーションからその同じ証券を手動でトレードすることは不可能です。他の証券は引き続き手動でトレードすることができます。



このボタンをクリックし ProOrder ウィンドウを開き、実行中のトレーディングシステムを確認することができます。

### 複数のトレーディングシステムを同じ証券で実行する

1つの証券に対して複数のトレーディングシステムを実行している場合、ネットポジションはこれらすべてのトレーディングシステムによって定められます。例えば、2つのトレーディングシステムのうち1つが1ロットを買い、もう1つが1ロットを売ったら、ネットポジションは0となります。どの証券に対しても、すべてのトレーディングシステムを合わせたネットポジションのみが既定の時間に市場にオープンにされます。

トレーディングシステムの資産カーブを開くと、「ポジション」チャートを見ることができます。このチャートは、証券に関する個々のトレーディングシステムのポジションを示しており、その証券で実行されているすべてのシステムで定められるネットポジションとは異なることがあります。ネットポジションは、ポジションラインで表示されます。

例：

2つのトレーディングシステムを1つの証券で実行: 1つは各 100,000 を 6 ロット購入、もう1つは各 100,000 を 2 ロット購入 = ネットポジションは+800,000 の買い。

この例では、チャートに表示されるトレーディングシステムのポジションは、+600,000 です。ポジションラインで表示されるネットポジションは、+800,000 です。



+ 800,000 のネットポジションは、「取引」>「ポジション一覧」ウインドウで参照することができます。



Name	Portfolio value	Latent gain	Gain today	Exposure	Margin	Cash
My Portfolio	99,798.60\$	528.00\$	-230.00\$	1,007,200.00\$	978,103.48\$	99,238.60\$

Name	%Chg	Last	Qty	Unit cost	Exposure	Latent gain	Gain today
EUR/USD Spot	-0.08%	1.2590	800,000	1.2583	1,007,200.00\$	528.00\$	-230.00\$

1つの銘柄に対して複数のトレーディングシステムの実行を指示している場合、システム実行のそれぞれのインスタンスは現在のポジション、注文、トレード、及び利益を個別にトラッキングします。結果「LongOnMarket」及び「ShortOnMarket」の指示が、現在実行中のシステムがロングかショートかを知らせます。

証券のネットポジションは、所定のシステムのポジションと異なることがあります。同様に、「CountOfLongShares」、「CountOfShortShares」、「CountOfPosition」、「PositionPrice」、「StrategyProfit」、「TradeIndex」、「TradePrice」と「PositionPerf」のような他のステータス変数は、現在の戦略みに適用されます。

## インジケータ制限

計算方法がリアルタイムの利用を許可しないため、以下のインジケータは自動トレーディングには使用できません：

**ZigZag** : このインジケータに基づくシグナルは、事後再計算されます。その結果、リアルタイムに発信されるシグナルは、バックテストの間に発信されたシグナルと大きく異なることがあります。



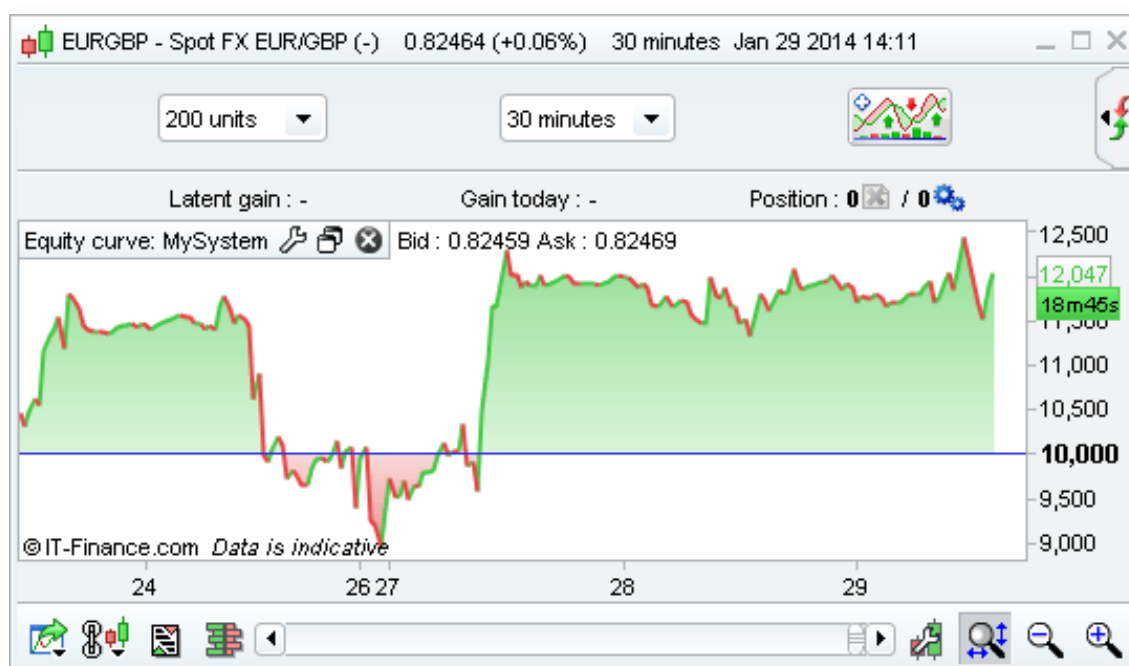
## 付録 A： トレーディングシステム結果の表示

トレーディングシステムの結果は、3 つの補完的な方法で表示されます。

### 1) 資産カーブ

バックテストの資産カーブは、トレーディングシステムまたはバックテストの損益を示します：

水平の青い線は、バックテストの場合初期資本、また自動トレーディングシステムの場合 0 を表します。10 000 の初期資本と 2705 の損失を持ったバックテストの場合、資産カーブの価格は下記の例で示すように 7295 となります。同等の損失をもつ自動トレーディングシステムの場合、初期値は 0、最終値は-2705 となります。



- パフォーマンスが正である場合（開始点からの利益）、**資産カーブ**は緑の濃淡で表されます。パフォーマンスが負であるならば、赤の濃淡で表されます。
- 前の点から増加した場合の**資産カーブのライン**は緑、減少した場合は赤です。

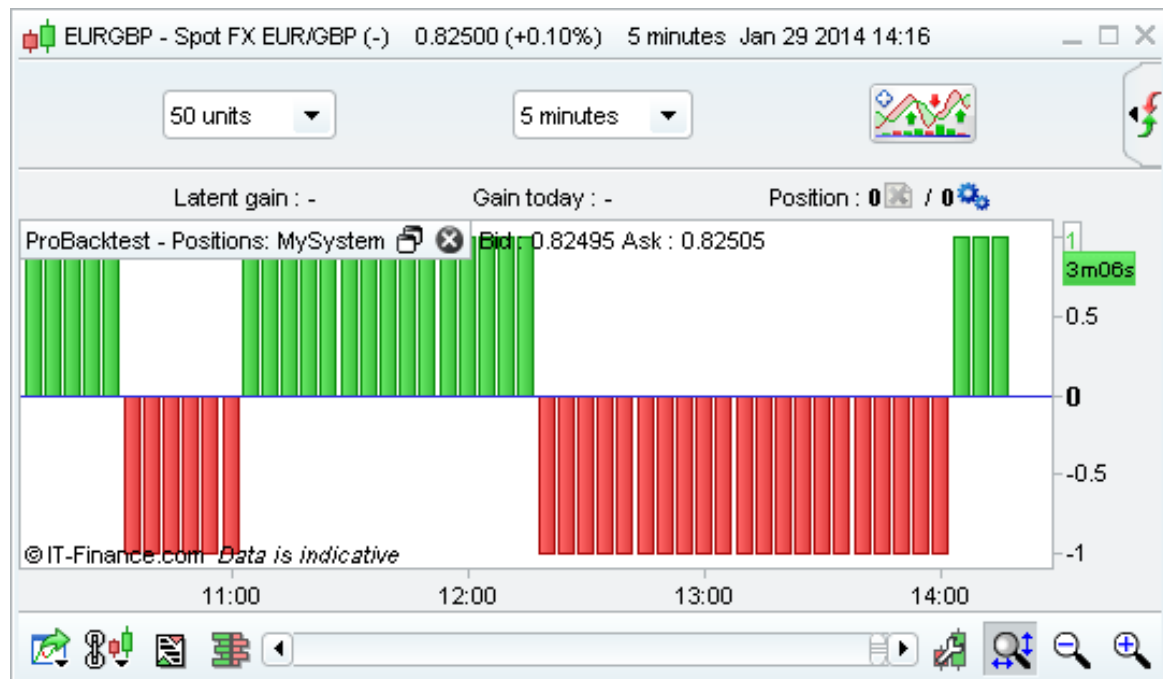
## 2) ポジションチャート

ポジションヒストグラムは、トレーディングシステムのシミュレーションの間のポジション推移をヒストグラムで表示します。

- 緑のバーは、オープンロングポジションを示します。
- 赤いバーは、オープンショートポジションを示します。
- バーが無い場合は、オープンポジションがないことを示します。

同じ色のいくつかのバーが連続する時は、ポジションがまだオープンであることを示します。

チャートの右側の垂直軸で、どれほどのポジションが現在オープンになっているかがわかります(ハイライトされています)。下記の例では、現在オープンロングポジションが、EUR/GBP 上に 100,000 が 1 ロットあることを示しています。



### 3) 詳細レポート

詳細レポートは、トレーディングシステムの統計と各ポジション及び注文の詳細を表示します：



詳細レポートは、3つのタブから成る独立したウィンドウに表示されます。



## 付録 A：トレーディングシステム結果の表示

「清算済みポジション統計」タブは、トレーディングシステムのパフォーマンス（利益または損失、勝ちトレードの数など）、及び最大ドローダウンのようなリスクのインジケータについて包括的な情報を提供します。レポートが生成される時点のオープンポジションの分析は含まれません（クローズポジションのみ考慮されます）。以下は統計のリストです：

「利益」は、トレーディングシステムが実現する利益または損失です。方程式は以下の通りです：

$$\text{利益} = \text{最終資本} - \text{初期資本}$$

この統計によってテストされた過去の期間とそれぞれの変数の組み合わせに対して定義付けされた絶対利益の可能性を評価します。

**注意：**「仲買業務パラメータ」で定められる売買委託手数料が、この計算において考慮されます。

「利益率(%)」は利益または損失のパーセンテージを表しています。方程式は以下の通りです：

$$\text{利益率}(\%) = 100 \times \text{利益} / \text{初期資本}$$

「Nbr ポジション数」はバックテスト中にオープンしたポジションの数を表します。

「評価益ポジションの比率(%)」は勝ちポジション率のパーセンテージを表します。以下のように計算します：

$$\text{評価益ポジションの比率}(\%) = (100 \times \text{勝ちポジション数}) / \text{ポジション数}$$

「評価益ポジションの平均利益」は、1つのポジションに対する平均利益を表します。出される注文の効率を評価するのに便利です。注文数の低いトレーディングシステムをつくる際、ポジションの平均利益は特に重要となります。以下のように定義されます：

$$\text{ポジションの平均利益} = \text{利益} / \text{ポジション数}$$

「最大利益確定ポジションの平均利益額」はトレーディングシステム開始以来オープンした所定のポジションの最大利益で、「評価発生ポジションの平均損失額」は所定のポジションの最大損失を表します。「利益と損失に関する標準偏差」は各ポジションの結果の標準偏差値です。

**最大ドローダウン**は、トレーディングシステムの最大損失の可能性を表します。ドローダウンは、資産カーブ中の所定のポイントとその前の最高ポイントの間の距離と定義されます：

$$\begin{aligned} \text{DD}(n) &= \text{Max } t \in [0; n] P(t) - P(n) \\ \text{ドローダウン}(n) &= \text{最大 } t \in [0; n] \text{ ポイント}(t) - \text{ポイント}(n) \end{aligned}$$

「最大ドローダウン」は、トレーディングシステム過去全体の最大ドローダウンとして計算されます。。

$$\begin{aligned} \text{MaxDD}(N) &= \text{Max } n \in [0; N] (\text{Max } t \in [0; n] P(t) - P(n)) \\ \text{最大ドローダウン}(N) &= \text{最大 } n \in [0; N] (\text{最大 } t \in [0; n] \text{ ポイント}(t) - \text{ポイント}(n)) \end{aligned}$$

**最大上昇** トレーディングシステムの最大利益の可能性を表します。上昇は、資産カーブの所定のポイントとその前の最高ポイントの差と定義されます：

$$\begin{aligned} \text{RU}(n) &= P(n) - \text{Min } t \in [0; n] P(t) \\ \text{上昇}(n) &= \text{ポイント}(n) - \text{最小 } t \in [0; n] \text{ ポイント}(t) \end{aligned}$$

「最大上昇」トレーディングシステムの過去全体の最高上昇値として計算されます。

付録 A：トレーディングシステム結果の表示

$$\text{MaxRU}(N) = \text{Max } n \in [0; N] \left( P(n) - \text{Min } t \in [0; n] P(t) \right)$$

$$\text{最大上昇}(N) = \text{最大 } n \in [0; N] \left( \text{ポイント}(n) - \text{最小 } t \in [0; n] \text{ポイント}(t) \right)$$

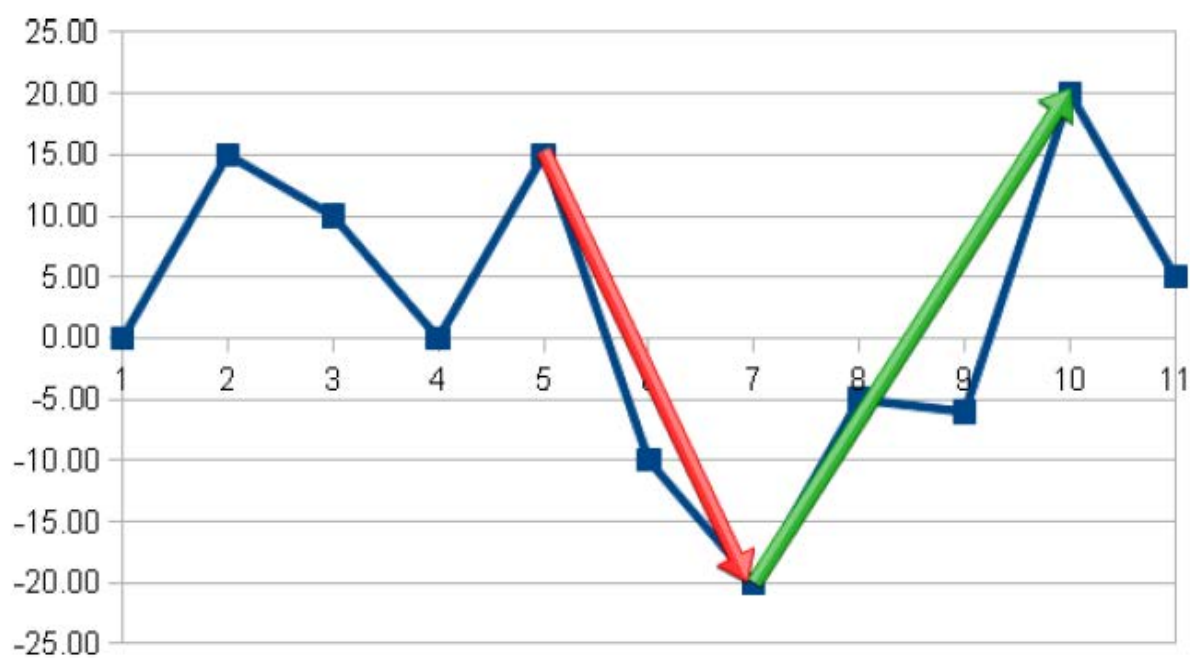
例：

バーインデックス	利益と損失	ドローダウン	上昇
1	0.00	0.00	0.00
2	15.00	0.00	-15.00
3	10.00	5.00	-10.00
4	0.00	15.00	0.00
5	15.00	0.00	-15.00
6	-10.00	25.00	0.00
7	-20.00	35.00	0.00
8	-5.00	20.00	-15.00
9	-6.00	21.00	-14.00
10	20.00	0.00	-40.00
11	5.00	15.00	-25.00

最大：

-35.00

40.00



「**最大リスクエクスポージャー率(%)**」: リスク指標は、ポジションの最大損失の可能性と資本の現在値の関係といえます。最大リスクエクスポージャー率(%)は、この最大値をパーセンテージで表したものです。それぞれ株、先物、外国為替の計算は以下の通りです:

- 株:

最大リスクエクスポージャー率(%) = 最大ポジション (数量 \* 平均価格 / capital) \* 100

- 先物:

最大リスクエクスポージャー率(%) = 最大ポジション (数量 \* デポジット / 資本) \* 100

- 外国為替:

最大リスクエクスポージャー率(%) = 最大ポジション (数量 \* 平均価格 \* レバレッジ / 資本) \* 100

同様に、「**平均リスクエクスポージャー率(%)**」は、平均リスクエクスポージャー率を表します。

「**売買委託手数料**」は、トレーディングシステム開始以来の各注文の総売買委託手数料です。これらの売買委託手数料は、バックテストの場合設定で定義づけすることができます。

市場時間率(%)は、オープンのポジションを持つバーの数をトレーディングシステムのバーの数で割って計算します。

以下の 2 つのタブは、自動トレーディングシステムまたはバックテストの実行中に出される注文や、オープン及びクローズされたポジションに関する情報を表示します。

- 「**注文リスト**」は、すべての注文の日付と時間、ディレクション、数量、及び価格を見ることができます。表示される時間は、銘柄のタイムゾーンです。
- 「**清算済みポジション一覧**」から、トレーディングシステムによって扱われたポジションに関する情報を見ることができます(ロングまたはショート、バーの数の期間、各ポジションのパフォーマンス、開始の日付と終了日)。レポート作成時にまだオープンのポジションが存在する場合、リストには含まれません。バックテストの場合、バックテストの終わりにすべてのポジションをクローズするのであればリアルタイムの日ではなく固定の終了日を設定してください。

## 付録 B : 具体的なコード例

### Heiken Ashi トレーディングシステム

このトレーディングシステムは、緑の Heiken Ashi ローソクが赤の後に現れた時に、買いシグナルを出します。

赤い Heiken Ashi ローソクが緑の後に現れた時に、売りシグナルを出します。

このバックテストは、通常のローソク足から Heiken Ashi 表示を再構築するものです。これは、Heiken Ashi ローソクではなく通常のローソク足を使用しているチャートに適用されなければなりません。

```
ONCE PreviousStatus = 0
IF BarIndex = 0 THEN
    XClose = TotalPrice
    XOpen = (Open + Close) / 2
ELSE
    XClose = TotalPrice
    XOpen = (XOpen[1] + Xclose[1]) / 2
ENDIF

IF XClose >= XOpen THEN
    IF PreviousStatus <> 1 THEN
        BUY 1 SHARES AT MARKET
        PreviousStatus = 1
    ENDIF
ELSE
    IF PreviousStatus <> -1 THEN
        SELLSHORT 1 SHARES AT MARKET
        PreviousStatus = -1
    ENDIF
ENDIF
```

### ZigZag トレーディングシステム

これはベストな売りと買いの機会を確定するための ZigZag に基づくバックテストです。このトレーディングシステムの株と先物における優れた結果は、ZigZag の非予測的特徴に関連しています。シグナルは事後再計算されるので、リアルタイムには有効なシグナルを出しません。

このトレーディングシステムの結果が興味深いのは、他のトレーディングシステムと比較してもかなり理想的な結果を出すことができるという点です。

```
// zigzag インジケーターの期間は、変数最適化を通じて最適化を行うことが可能
myZigZag = ZigZag[10]
c11 = (myZigZag > myZigZag[1])
c12 = (myZigZag < myZigZag[1])
IF c11 AND NOT LONGONMARKET THEN BUY
    1 SHARES AT MARKET
```

## 付録 B : コード詳細例

```
ENDIF
IF c12 AND NOT SHORTONMARKET THEN
    SELLSHORT 1 SHARES AT MARKET
ENDIF
```

### トレーリングストップによる簡易ブレイクアウト範囲

これは、ロングポジションのみを扱う基本的なブレイクアウト日中足トレーディングシステムです。初期範囲は、最初の 2 つのローソク足における1日の最高ポイントと最低ポイントで決定されます。支持線は最低ポイントで、抵抗線は最高ポイントで定められます。

価格が抵抗線と交差し 10 日移動平均が増加している場合、ロングポジションが取られます。

1%の利益目標が定義づけられます。

プロテクティブストップは支持線のレベルで設定され、価格がこのレベルに達した場合ポジションは逆指値注文でクローズされます。

このポジションはまた、夜を越さないように午後 5 時のローカル市場時間ですべてクローズされます。このトレーディングシステムのテストには、日中足データへのアクセスが必要です。

```
DEFPARAM CumulateOrders = False
```

```
MM = Average[10](close)
```

```
MyTarget = 1
```

```
EndTime = 170000
```

```
IF INTRADAYBARINDEX = 2 THEN
```

```
    MyResistance = HIGHEST[2](high)
```

```
    MySupport = LOWEST[2](low)
```

```
ENDIF
```

```
REM ロングにエントリー:
```

```
IF MM>MM[1] AND close CROSSES OVER MyResistance THEN BUY 1
    SHARES AT MARKET
```

```
ENDIF
```

```
REM ロングをエグジット:
```

```
IF time > EndTime THEN SELL
    AT MARKET
```

```
ENDIF
```

```
SELL AT MySupport STOP
```

```
SET TARGET %Profit MyTarget
```

### スムーズドストキャスティックトレーディングシステム

このトレーディングシステムは、中間価格に適用されたスムーズドストキャスティック(インジケーター1)、及びこの値の指数移動平均線(インジケーター2)に基づいています。トレーディングシステムは、インジケーター1がインジケーター2を上回る時に買い、またインジケーター1がインジケーター2の下にある時にショートポジションにエントリーします。

目標 (リミット)はエントリー価格の 1%上に定義づけられます。.

```
DEFPARAM CumulateOrders = False

REM 買い
Indicator1 = SmoothedStochastic[9,9](MedianPrice)
Indicator2 = ExponentialAverage[9](Indicator1)
// Init 変数
StopLimit = 10

REM ロング条件を入力
c1 = (Indicator1 >= Indicator2) IF
c1 THEN
    BUY 1 SHARES AT MARKET
ENDIF

REM ショート条件を入力
IF NOT c1 THEN
    SELLSHORT 1 SHARES AT MARKET
ENDIF

SET TARGET %PROFIT StopLimit
```

### スウィングトレーディング、ADX、移動平均

誤シグナルを減らしリスクを最小に抑えるため、このバックテスト ADX インジケーターと少なくとも 5 日間からのレベル 30 のポジションを使用します。これは日々実行される必要があります。

トレーディングシステムには、トレーディングの機会を制限する多くの条件があります。

```
DEFPARAM CumulateOrders = False

MyADX12 = ADX[12]
ADXperiods = 5
MyMM20 = Average[20](Close)
IsLow30 = 0

IF Lowest[ADXperiods+1](MyADX12)<30 THEN
    IsLow30 = 1
ENDIF

// ACHAT
// ADX 12 は、少なくとも 5 つのバーで 30 以上でなくてはならない
```

## 付録 B : コード詳細例

Condition1 = NOT IsLow30

// 現在の期間の 20 日移動平均が現在の期間の高値と安値の間にあり、前期間の移動平均が前期間の高値と安値の間にある場合

Condition2 = High > MyMM20 AND Low < MyMM20 AND High[1] < MyMM20[1] AND Low[1] < MyMM20[1]

// 現在の日付の高値が、前日の高値よりも高い場合

Condition3 = Dhigh(0) > Dhigh(1)

IF Condition1 AND Condition2 AND Condition3 THEN BUY

1 SHARES AT MARKET

ENDIF

// ショート

// ADX 12 は、少なくとも 5 つのバーで 30 以上でなくてはならない

Condition4 = NOT IsLow30

// 現在の期間の 20 日移動平均が現在の期間の高値と安値の間にあり、前期間の移動平均が前期間の高値と安値の間にある場合

Condition5 = High > MyMM20 AND Low < MyMM20 AND High[1] > MyMM20[1] AND Low[1] > MyMM20[1]

// 現在の日付の安値が、前日の安値よりも低い場合

Condition6 = Dlow(0) < Dlow(1)

IF Condition4 AND Condition5 AND Condition6 THEN

SELLSHORT 1 SHARES AT MARKET

ENDIF

## ポジションカウンター付トレーディングシステム

### RSI に適用されるインバース・フィッシャー(Inverse Fisher) 変換

このトレーディングシステムは、「インバース・フィッシャー変換 RSI(Inverse Fisher Transform RSI)」使い、売買注文を行います。

インバース・フィッシャー変換 RSI が 50 以上で交差するとロングポジションにエントリーし、80 以下で交差するとロングポジションをエグジットします。

インバース・フィッシャー変換 RSI が 50 以下で交差するとショートポジションにエントリーし、インバース・フィッシャー変換 RSI が 20 以上で交差するときショートポジションをエグジットします。

このトレーディングシステムは、1 時間表示の先物、または 1 日表示の株でバックテストされます。

REM RSI に適用するインバースフィッシャー変換

REM パラメータ : n = RSI の計算のためのバーの数

n = 10

Ind=RSI[n](Close)

x = 0.1 \* (Ind - 50)

y = (EXP (2 \* x) - 1) / (EXP (2 \* x) + 1) z

= 50 \* (y + 1)

myInverseFisherTransformsRSI = z

IF (myInverseFisherTransformsRSI Crosses Over 50) THEN

BUY 1 SHARES AT MARKET

ENDIF

IF (myInverseFisherTransformsRSI Crosses Under 80) THEN SELL

AT MARKET

ENDIF

IF (myInverseFisherTransformsRSI Crosses Under 50) THEN

SELLSHORT 1 SHARES AT MARKET

ENDIF

IF (myInverseFisherTransformsRSI Crosses Over 20) THEN

EXITSHORT AT MARKET

ENDIF



## トレードインデックス付トレーディングシステム - ファインド・インサイドバー

以下のトレーディングシステムの例は、「インサイドバー」と呼ばれる頻繁に使用される価格パターンと 2 つのローソク足の形に基づいています。

- 2 つ前のローソクの範囲が 1 つ前のローソクの範囲より大きいならば、第 1 の形成が起こります。1 つ前のローソクは白くなければなりません(クローズ > オープン)。この場合、ロングポジションとなります。
- 2 つ前のローソクの範囲が 1 つ前のローソクの範囲よりも小さく、且つ 1 つ前のローソクが黒(クローズ < オープン)ならば、第 2 の形成が起こります。この場合、ショートポジションとなります。

すべてのポジションは、オープンになった後自動的にその 3 バー後ろでクローズします。

```
DEFPARAM CumulateOrders = False

Condition1 = (High[2] >= High[1] AND Low[2] <= Low[1])
Condition2 = (High[2] <= High[1] AND Low[2] <= Low[1])
Condition3 = (Close[1] > Open[1])
Condition4 = (Close[1] < Open[1])
IF (Condition1 AND Condition3) THEN
    BUY 1 Share AT MARKET
ENDIF
IF LONGONMARKET AND (BarIndex - TRADEINDEX) = 3 THEN SELL
    1 share AT MARKET
ENDIF
IF (Condition2 AND Condition4) THEN
    SELLSHORT 1 share AT MARKET
ENDIF
IF SHORTONMARKET AND (BarIndex - TRADEINDEX) = 3 THEN
    EXITSHORT AT MARKET
ENDIF
```

## 資金管理戦略

バックテストの結果は、資金管理戦略で改善することができます。

これらの戦略は、「マーチンゲール(martingales)」で形式化されることがあります。それは、トレーディングシステムの数学期待値を最適化することを目的とします。多くの取引が行われる場合、期待値は各取引の平均的な勝ちまたは負けとなります。これは、勝ち取引の確率と利益または損失の推定額の予測が可能であることを意味します。

マーチンゲールを実施する上で、トレーディングシステムで完全カスタマイズが可能なストップロス、テイクプロフィット及びインアクティビティ注文を直接コーディングし、またポジションのサイズをダイナミックに管理するためのサブ戦略は非常に便利です。

### プロテクティブストップと利益目標

プロテクティブストップ、トレーリングストップ、及び利益目標に関する詳細は、本書の該当する項を参照してください。

### インアクティビティストップ

以下のコードは、トレーディングシステムでインアクティビティストップを使うことを可能にします。ここでは InactivityStopLong と InactivityStopShort と呼びますが、ストップの条件を必ず定義づけてください。以下の例では、ストップは 10 のバーの後で引き起こされます。

ONCE Count = 10

REM ポジションが自動的にクローズするバーの数を選択

IF ONMARKET AND (BARINDEX - TRADEINDEX + 1) > Count THEN IF

LONGONMARKET THEN

SELL AT MARKET

ENDIF

IF SHORTONMARKET THEN

EXITSHORT AT MARKET

ENDIF

ENDIF

## 累積注文 - ポジションカウンタを使用して既存ポジションを追加する

この項ではポジションサイズを増加させるための累積注文の例を取り上げます。

累積注文を有効にするには、「**DEFPARAM CumulateOrders=True**」のコマンドをプログラムの初めに入力します。

このトレーディングシステムは、初期ポジションのみのために、RSIに基づいた最初の条件を使用します。追加の株はオープンが前のクローズよりも大きな各バーに最多 3 つまで加えられます。

最大ポジションサイズを 3 に限定するため、「CountofPosition」がこのコードに使用されます。

```
DEFPARAM CumulateOrders = True
```

```
REM RSI < 30 で且つオープンのポジションが存在しない場合、1 購入
```

```
IF RSI[14](Close) < 30 AND NOTONMARKET THEN BUY
```

```
    1 SHARES AT MARKET
```

```
ENDIF
```

```
REM オープンロングポジションが存在し、オープン > 前の終値の場合、毎回最高 3 つまで追加購入
```

```
IF LONGONMARKET AND COUNTOFPOSITION < 3 AND Open > Close[1] THEN BUY 1
```

```
    SHARES AT MARKET
```

```
ENDIF
```

```
REM 価格が単純移動平均より下をクロスした場合、ポジションをクローズ
```

```
IF Close Crosses Under Average[14](Close) THEN
```

```
    SELL AT MARKET
```

```
ENDIF
```

これでツールが整い、マーチンゲールを見ることができるようになりました。ここで最も人気のあるものをいくつかご紹介いたします。これらのテクニックはどのようなトレーディングシステムにも追加することができます。

## The classic martingale (クラシックマーチンゲール)

クラシックマーチンゲールは、次のトレードが勝ちで、損失を補うために負けた時に、ポジションサイズを倍増します。このようなトレーディングシステムの欠点は、連続損失の場合、ポジションの倍増を非常に難しくする(または不可能にする)ということです。例えば、1000€から開始し、連続 5 回負けた場合、このトレーディングシステムを継続するためには  $1000 \times 32 = 32000\text{€}$ が必要となります。

結果、マーチンゲールを使用したトレーディングシステムは、先物や外国為替よりも株に適していると言えます。なぜなら、他の 2 つの市場タイプにおけるトレードに必要な初期資本は株よりも大きいからです。

このコードはユーザーのエントリーとエグジットの条件と統合する必要があります。

```
//*****トレーディングシステムの始めに挿入するコード*****//
ONCE OrderSize = 1
ONCE ExitIndex = -2
REM 初期ポジションサイズ 1
//*****
****//

//*****ポジションのクローズ直後に挿入するコード
*****// ExitIndex=BarIndex

//*****トレーディングシステムの終わりに挿入するコード
*****// IF Barindex=ExitIndex+1 THEN
    ExitIndex=0
    IF PositionPerf(1) < 0 THEN OrderSize
        = OrderSize * 2
        REM 最後のトレードが負けトレードの場合、OrderSize を 2 倍にする
    ELSIF PositionPerf(1) > 0 THEN
        OrderSize = 1
        REM 最後のトレードが勝ちトレードの場合、ポジションサイズを 1 にリセットする
    ENDIF
ENDIF
//*****
REM ポジションサイズはコード全体の変数 OrderSize によって決定されなければならない
```

## The great martingale(グレートマーチンゲール)

グレートマーチンゲールはクラシックマーチンゲールと類似していますが、負けの後ポジションサイズを倍増することに加え、さらに1つの単位を加えます。

これは連続損失の場合にはクラシックマーチンゲールよりもリスクが高くなりますが、それ以外の場合は利益を非常に大きくすることが可能になります。

このコードはユーザーのエントリーとエグジットの条件と統合する必要があります。

```
//*****トレーディングシステムの始めに挿入するコード*****//
```

```
ONCE OrderSize = 1
```

```
ONCE ExitIndex = -2
```

```
REM 初期ポジションサイズ 1
```

```
//*****//
```

```
//*****ポジションのクローズ直後に挿入するコード
```

```
*****// ExitIndex=BarIndex
```

```
//*****トレーディングシステムの終わりに挿入するコード
```

```
IF BarIndex=ExitIndex+1 THEN
```

```
ExitIndex=0
```

```
IF PositionPerf(1) < 0 THEN
```

```
OrderSize = OrderSize * 2 + 1 // 最後のトレードが負けの場合、OrderSize を2倍にし、且つ1
```

```
追加
```

```
ELSIF PositionPerf(1) >= 0 THEN
```

```
OrderSize = 1 //最後のトレードが勝ちの場合、OrderSize を1に設定
```

```
ENDIF
```

```
ENDIF
```

```
//*****//
```

```
REM ポジションサイズはコード全体の変数 OrderSize によって決定されなければならない
```

## Piquemouche (ピケモッシュュ)

ピケモッシュュ(Piquemouche)は、クラシックマーチンゲールのもう一つの変形です。3 つ未満の連続損失がある場合、ポジションサイズを 1 増やします。3 つ以上の連続損失の場合は、ポジションを 2 倍にします。利益の場合、ポジションサイズを 1 単位にリセットします。

このトレーディングシステムは、3 つの連続損失に達成するまでポジションサイズが指数的に増加されないで、前の 2 つよりもリスクは低いといえます。

このコードはユーザーのエントリーとエグジットの条件と統合する必要があります。

```
//*****トレーディングシステムの始めに挿入するコード*****//
ONCE OrderSize = 1
ONCE BadTrades = 0
ONCE ExitIndex = -2
REM 初期ポジションサイズ 1
//*****
****//
//*****ポジションのクローズ直後に挿入するコード
*****// ExitIndex=BarIndex

//*****トレーディングシステムの終わりに挿入するコード
*****// IF BarIndex=ExitIndex+1 THEN
    ExitIndex=0
    IF PositionPerf(1) < 0 THEN BadTrades
        = BadTrades + 1
        IF BadTrades < 3 THEN
            //最後のポジションが負けで、且つ 3 連続以上負けの場合、
            // OrderSize に1追加
            OrderSize = OrderSize + 1
        ELSEIF BadTrades MOD 3 = 0 THEN
            // 最後のポジションが負けで、且つ 3 連続以上負けの場合、
            // OrderSize を 2 倍にする
            OrderSize = OrderSize * 2
        ENDIF
    ELSEIF PositionPerf(1) >= 0 THEN
        // 前のトレードが勝ちの場合、OrderSize を 1 にリセットする OrderSize = 1
        BadTrades = 0
    ENDIF
ENDIF
//*****
REM ポジションサイズはコード全体の変数 OrderSize によって決定されなければならない
```

## D'Alembert Pyramid(ダランベール・ピラミッド)

このマーチンゲールは、フランスの 18 世紀の数学者ダランベールによって有名になりました。損失の場合ポジションサイズは 1 単位増やされ、利益の場合 1 単位減らされます。

ポジションサイズ管理のこのテクニックは、その連続した利益が再び勝つ可能性を低める場合と連続した損失が再び負ける可能性を減らす場合にのみ該当します。

このコードはユーザーのエントリーとエグジットの条件と統合する必要があります。

```
//*****トレーディングシステムの始めに挿入するコード*****//
ONCE OrderSize = 1
ONCE ExitIndex = -2
REM 初期ポジションサイズ 1
//*****
****//

//*****ポジションのクローズ直後に挿入するコード
*****// ExitIndex=BarIndex

//*****トレーディングシステムの終わりに挿入するコード
*****// IF BarIndex=ExitIndex+1 THEN
    ExitIndex=0
    IF PositionPerf(1) < 0 THEN OrderSize
        = OrderSize + 1
    ELSIF PositionPerf(1) >= 0 THEN
        OrderSize = MAX(OrderSize -1, 1)
    ENDIF
ENDIF
//*****//
REM ポジションサイズはコード全体の変数 OrderSize によって決定されなければならない
```

## Centre d'Alembert(コントレ・ダランベール)

これはダランベール・ピラミッドの逆的トレーディングシステムです。損失の場合にはポジションサイズを減少させ、利益の場合にはポジションサイズを増やします。

過去の損失が将来の損失の可能性を増やし、過去の利益が将来の利益の可能性を増やすという考えに関連したテクニックです。

このコードはユーザーのエントリーとエグジットの条件と統合する必要があります。

```
//*****トレーディングシステムの始めに挿入するコード*****//
ONCE OrderSize = 1
ONCE ExitIndex = -2
REM 初期ポジションサイズ 1
//*****//
//*****ポジションのクローズ直後に挿入するコード*****//
ExitIndex=BarIndex

//*****トレーディングシステムの終わりに挿入するコード
*****// IF Barindex=ExitIndex+1 THEN
    ExitIndex=0
    IF PositionPerf(1) < 0 THEN
        OrderSize = MAX(OrderSize -1, 1)
    ELSIF PositionPerf(1) >= 0 THEN
        OrderSize = OrderSize + 1
    ENDIF
ENDIF

//*****//
REM ポジションサイズはコード全体の変数 OrderSize によって決定されなければならない
```



## 用語解説

### A

コード	構文	機能
<b>Abs</b>	Abs(a)	「絶対値」関数 (Absolute Value) a
<b>AccumDistr</b>	AccumDistr(close)	典型的蓄積/分布インジケータ
<b>ADX</b>	ADX[N]	n期間の平均方向性指数または「ADX」
<b>ADXR</b>	ADXR[N]	n期間のインジケータ平均方向性指数率または「ADXR」
<b>AND</b>	a AND b	論理 AND 演算子
<b>AroonDown</b>	AroonDown[N]	n期間のアロンドアウン・インジケータ
<b>AroonUp</b>	AroonUp[N]	n期間のアロニアップインジケータ
<b>Atan</b>	Atan(a)	「逆正接」関数(Arctangent) a
<b>AS</b>	RETURN Result AS "ResultName"	チャートにあるラインまたはインジケータを命名するのに使用される指示。「RETURN」と併用される。
<b>AT</b>	AT (price)	コマンドを価格と関連させる
<b>Average</b>	Average[N](price)	n期間の単純移動平均
<b>AverageTrueRange</b>	AverageTrueRange[N](price)	「アベレージ・トゥルー・レンジ」 (Average True Range) -ワイルダー式で トゥルー・レンジ(True Range)を平均化した数値
<b>B</b>		
コード	構文	機能
<b>BarIndex</b>	BarIndex	データがロードされた当初からのバーの数 (ProBacktest または ProOrder の場合はトレーディングシステム、ProBuilder インジケータの場合はチャート内) PreLoadBars も参照
<b>BollingerBandWidth</b>	BollingerBandWidth[N](price)	ボリンジャーバンド幅 インジケータ
<b>BollingerDown</b>	BollingerDown[N](price)	ローボリンジャーバンド
<b>BollingerUp</b>	BollingerUp[N](price)	アッパーボリンジャーバンド
<b>BREAK</b>	(FOR...DO...BREAK...NEXT) または (WHILE...DO...BREAK...WEND)	FOR ループまたは WHILE ループからの強制エグジットの指示
<b>BUY</b>	BUY x SHARES	ロングポジションをオープンする指示
<b>C</b>		
コード	構文	機能
<b>CALL</b>	myResult = CALL myFunction	コーディングを行っているプログラムで使用するためにユーザーインジケータを呼び出し
<b>CAPITAL</b>	BUY x% CAPITAL	ポジションで使用されている資本

		の%
CASH	BUY x CASH	ポジションで使用されているキャッシュ
CCI	CCI[N](price) または CCI[N]	コモディティ・チャンネル・インデックスインジケーター
ChaikinOsc	ChaikinOsc[Ch1, Ch2](price)	チャイキンオシレーター
Chandle	Chandle[N](price)	シャンデモメンタム オシレーター
ChandeKrollStopUp	ChandeKrollStopUp[Pp, Qq, X]	ロングポジションに対するシャンデとクロールプロテクティブストップ
ChandeKrollStopDown	ChandeKrollStopDown[Pp, Qq, X]	ショートポジションに対するシャンデとクロールプロテクティブストップ
Close	Close[N]	現在のバーまたは最後から n 番目のバーの終値
COLOURED	RETURN Result COLOURED(R,G,B)	RGB で指定した色をカーブに着色する
COS	COS(a)	コサイン関数
COUNTOFLONGSHARES	COUNTOFLONGSHARES	オープンロング株またはロットの数を数える
COUNTOFPOSITION	COUNTOFPOSITION	オープン株またはロットの数を数える
COUNTOFSHORTSHARES	COUNTOFSHORTSHARES	オープンショート株またはロットの数を数える
CONTRACT	BUY 1 CONTRACT	買いコントラクトの数を指定。 「SHARES」に相当
Crosses Over	a Crosses Over b	カーブが別のカーブとクロスオーバーしたかどうかを確認するブール演算子
Crosses Under	a Crosses Under b	カーブが別のカーブとクロスアンダーしたかどうかを確認するブール演算子
CUMSUM	CUMSUM(price)	ロードされたすべてのデータに関する特定の価格の合計金額
CumulateOrders	DefParam Cumulateorders = false	偽に設定した際に同じ方向で市場にエントリーするために、コードがポジションを強化して複数の注文を設定するのを禁ずる
CurrentDayOfWeek	CurrentDayOfWeek	週の現在曜日を表示
CurrentHour	CurrentHour	現在時間を表示
CurrentMinute	CurrentMinute	現在分を表示
CurrentMonth	CurrentMonth	現在月を表示
CurrentSecond	CurrentSecond	現在秒を表示
CurrentTime	CurrentTime	現在時間を表示(時・分・秒)
CurrentYear	CurrentYear	現在年を表示
CustomClose	CustomClose[N]	チャートの設定ウインドウのカスタマイズ可能な条件(デフォルト:クローズ)
Cycle	Cycle(price)	サイクルインジケーター
D		
コード	構文	機能
Date	Date[N]	各バーがチャートにロードした日付を表示
Day	Day[N]	各バーがチャートにロードした曜日を表示
Days	Days[N]	1900 からの日々のカウンター
DayOfWeek	DayOfWeek[N]	各バーの曜日
Dclose	Dclose(N)	現在から数えて n 日前のクローズ

DEFPARAM	DEFPARAM	CumulateOrders(累積注文)と同様にパラメータの定義づけを可能にする
DEMA	DEMA[N](price)	2重指数移動平均(Double Exponential Moving Average)
Dhigh	Dhigh(N)	現在のバーから数えてn番目の高値
DI	DI[N](price)	(DI+) - (DI-)を表す
DIminus	Diminus[N](price)	DI- インジケーターを表す
Diplus	Diplus[N](price)	DI+インジケーターを表す
Dlow	Dlow(N)	現在のバーから数えてn番目の安値
DO	FOR 及び WHILE を参照	ループ・アクションを定義づけるためのFORループとWHILEループのオプションの指示
Dopen	Dopen(N)	現在のバーから数えてn番目のオープン
DOWNT0	FOR を参照	下降している注文でループ処理を行うためにFORループに使用される指示
DPO	DPO[N](price)	デトレンドッド・プライス・オシレーター(DPO:Detrended Price Oscillator)
E		
コード	構文	機能
EaseOfMovement	EaseOfMovement[I]	イーズオブムーブメント・インジケーター
ELSE	IF/THEN/ELSE/ENDIF を参照	If 条件文の第2の条件の呼出に用いられる指示
ELSEIF	IF/THEN/ELSIF/ELSE/ENDIF を参照	Else If(条件ループの中で使われる)を表す
EMV	EMV[N]	イーズオブムーブメント値インジケーター
ENDIF	IF/THEN/ELSE/ENDIF を参照	IF 条件文の終了指示
EndPointAverage	EndPointAverage[N](price)	a のエンドポイント移動平均
EXITSHORT	EXITSHORT x SHARES	ショートポジションをクローズする指示
Exp	Exp(a)	指数関数
ExponentialAverage	ExponentialAverage[N](price)	指数移動平均
F		
コード	構文	機能
FOR/TO/NEXT	FOR i=a TO b DO a NEXT	FOR ループ(上昇(TO)または下降(DOWNT0)している注文ですべての値を処理する)
FlatAfter	DefParam FlatAfter = HHMMSS	時刻、分、秒で指定した特定の日付けの後にポジションをクローズし、指値/逆指値注文をキャンセルし、追加注文を出さないようにする
FlatBefore	Defparam FlatBefore = HHMMSS	時刻、分、秒を指定した特定の日付けの前に、ポジションをクローズし、指値/逆指値注文をキャンセルし、追加注文を出さないようにする
ForceIndex	ForceIndex(price)	勢力指数インジケーター(誰が市場をコントロールするかを決定(バイヤーまたはセラー))
H		

コード	構文	機能
High	High[N]	現在のバーまたは最後からn番目のバーの高値
Highest	Highest[N](price)	定義づけされる複数のバーの最高値
HistoricVolatility	HistoricVolatility[N](price)	過去ボラティリティ（または統計ボラティリティ）
Hour	Hour[N]	チャートにロードされた各バーの時間を表す
I-J-K		
コード	構文	機能
IF/THEN/ENDIF	IF a THEN b ENDIF	2 番目の指示を除く条件指示のグループ
IF/THEN/ELSE/ENDIF	IF a THEN b ELSE c ENDIF	条件指示のグループ
IntradayBarIndex	IntradayBarIndex[N]	ロードされたすべてのデータのうち、1 日で表示されたバーを数える
L		
コード	構文	機能
LIMIT	BUY AT x LIMIT	指値注文を導入する指示
LinearRegression	LinearRegression[N](price)	リニアリグレーションインジケーター
LinearRegressionSlope	LinearRegressionSlope[N] (price)	リニアリグレーションインジケーターの勾配
Log	Log(a)	「ネイピアの対数」関数 (Neperian logarithm) a
LONGONMARKET	LONGONMARKET	ユーザーがオープンロングポジションを保有しているかどうかを表す
Low	Low[N]	現在のバーまたは最後からn番目のバーの安値
Lowest	Lowest[N](price)	定義づけされる複数のバーの最安値
Loss	Set STOP LOSS x	ポジションエントリー価格からの x 単位のストップロス設定
%LOSS	SET STOP %LOSS x	ポジションエントリー価格からの x% のストップロス設定
\$LOSS	SET STOP \$LOSS x	x 円または\$ (銘柄の通貨) のストップロス設定
Lot	BUY 1 LOT	買いまたは売りのロット数の定義づけ (「SHARE」に相当)
M		
コード	構文	機能
MACD	MACD[S,L,Si](price)	ヒストグラム中の移動平均収束拡散法 (MACD:Moving Average Convergence Divergence)
MACDline	MACDLine[S,L,Si](price)	MACD 線インジケーター
MARKET	BUY AT MARKET	市場価格での注文を指定
MassIndex	MassIndex[N]	N 本のバーに適用されたマスインデックスインジケーター
Max	Max(a,b)	「最大値」関数 (Maximum)
MedianPrice	MedianPrice	高値と安値の平均
Min	Min(a,b)	「最小値」関数 (Minimum)
Minute	Minute	チャートにロードされた各バーの分を表す

<b>Mod</b>	a Mod b	「剰余付除法」関数 (remainder of the division)
<b>Momentum</b>	Momentum[N]	モメンタムインジケーター (クローズ - 最後からn番目のバーのクローズ)
<b>MoneyFlow</b>	MoneyFlow[N](price)	マネーフロー インジケーター (-1 と 1 の間の結果)
<b>MoneyFlowIndex</b>	MoneyFlowIndex[N]	マネーフロー指数インジケーター
<b>Month</b>	Month[N]	チャートにロードされた各バーの月を表す
<b>N</b>		
<b>コード</b>	<b>構文</b>	<b>機能</b>
<b>NEXT</b>	FOR/TO/NEXT を参照	FOR ループの終了指示
<b>NextBarOpen</b>	AT MARKET NextBarOpen	次のバーのオープンで実行する注文の指定
<b>NoCashUpdate</b>	DEFPARAM NoCashUpdate=true/false	利益と損失によって初期資本を更新しないことをバックテストに許可 (バックテストのためのみの指示)
<b>NOT</b>	NOT A	理論演算子 NOT
<b>O</b>		
<b>コード</b>	<b>構文</b>	<b>機能</b>
<b>OBV</b>	OBV(price)	オンバランスボリューム (On-Balance-Volume) インジケーター
<b>ONCE</b>	ONCE VariableName = VariableValue	一度だけ行われる定義ステートメントの導入
<b>ONMARKET</b>	ONMARKET	ポジションがオープンかどうかを表す
<b>Open</b>	Open[N]	現在のバーまたは最後からn番目のバーをオープンにする
<b>OR</b>	a OR b	論理演算子 OR
<b>P-Q</b>		
<b>コード</b>	<b>構文</b>	<b>機能</b>
<b>PipValue</b>	PipValue	証券通貨€/ \$の1ピップ(または1ポイント) の値 ピップ値=ポイント値
<b>PipSize</b>	PipSize	ピップのサイズ (またはポイント) ピップサイズ=ポイントサイズ
<b>PointValue</b>	PointValue	銘柄の通貨€/ \$での1ピップ(または1ポイント) の値 ピップ値=ポイント値
<b>PointSize</b>	PointSize	ピップのサイズ (またはポイント) ピップサイズ=ポイントサイズ
<b>PositionPerf</b>	PositionPerf(n)	n番前のポジションの利益率(%)または損失率(%)を表す
<b>PositionPrice</b>	PositionPrice	現在の平均ポジション価格
<b>PreLoadBars</b>	DefParam PreLoadBars = 200	トレーディングシステムで使用するインジケーターの計算のためにプレロードされたバーの最大値の設定
<b>PriceOscillator</b>	PriceOscillator[S,L](price)	価格オシレーター率(%)
<b>PositiveVolumeIndex</b>	PriceVolumeIndex(price)	ポジティブボリューム指標インジケーター (PVI:Positive Volume Index)

		indicator)
PVT	PVT(price)	価格ボリュームトレンド指標
QUIT	QUIT	トレーディングシステムのストップに使用
R		
コード	構文	機能
R2	R2[N](price)	R 乗指標 (価格におけるリニアリグレーションのエラー率)
Range	Range[N]	範囲を計算 (高値と安値の差)
REM	REM comment	コメントの導入 (コードでは無視)
Repulse	Repulse[N](price)	リパルス指標 (各ローソク足のバイヤーとセラーの力関係を測る)
RETURN	RETURN Result	結果を返す指示
ROC	ROC[N](price)	価格変更率指標
RSI	RSI[N](price)	相対力指数 (RSI:Relative Strength Index) インジケーター
Round	Round(a)	「小数点大 1 位で四捨五入して正数にする」関数
ROUNDEDUP	ROUNDEDUP	買いまたは売りの数量の端数を切り上げる (キャッシュで買う指示の株取引に使用される)
ROUNDEDDOWN	ROUNDEDDOWN	買いまたは売りの数量の端数を切り下げる (キャッシュで買う指示の株取引に使用される)
S		
コード	構文	機能
SAR	SAR[At,St,Lim]	パラボリック SAR インジケーター
SARatdmf	SARatdmf[At,St,Lim](price)	スムーズドパラボリック SAR インジケーター
SELL	SELL x SHARES	ロングポジションをクローズする指示
SELLSHORT	SELLSHORT x SHARES	ショートポジションをオープンする指示
SET	SET	注文の種類を決定する。指値または逆指値。
SET TARGET	SET TARGET PROFIT x	利食い注文の設定を価格レベル x で指示
SHARES	BUY x SHARES	買いまたは売りの株数を決定
SHORTONMARKET	SHORTONMARKET	オープンショートポジションが存在するかどうかを表す
Sin	Sin(a)	「サイン」関数
Sgn	Sgn(a)	「サイン」a 関数を表す (正か負の数)
SMI	SMI[N,SS,DS](price)	ストキャスティックモメンタム指標インジケーター
SmoothedStochastic	SmoothedStochastic[N,K] (price)	スムーズドストキャスティック
Square	Square(a)	「a 乗」関数
Sqrt	Sqrt(a)	「平方根」a 関数
STD	STD[N](price)	「標準偏差」集計関数

STE	STE[N](price)	「標準誤差」集計関数
Stochastic	Stochastic[N,K](price)	ストキャスティック %K ライン インジケーター
STOP	AT price STOP	逆指値注文の作成指示
Summation	Summation[N](price)	最後から N 番目のローソク足上に特定の価格を合計する
SuperTrend	SuperTrend[STF,N]	スーパートレンドインジケーター
T		
コード	構文	機能
Tan	Tan(a)	「タンジェント」a 関数
TEMA	TEMA[N](price)	3 重指数移動平均
Target	SET TARGET PROFIT x	利食い注文の設定を価格レベル x で指示
THEN	IF/THEN/ELSE/ENDIF を参照	IF の初めの条件に従う指示
Ticksiz	Ticksiz	銘柄のティックサイズ (価格変動幅の最小単位)
Time	Time[N]	チャートにロードされた各バーの時間を表す
TimeSeriesAverage	TimeSeriesAverage[N](price)	暫定的一連の移動平均
TO	See FOR/TO/NEXT	「FOR」ループ内の方向性の指示
Today	Today[N]	現在のバーより前のn期間の日付
TomorrowOpen	AT MARKET TomorrowOpen	次の日の始値で実行する注文を指定
TotalPrice	TotalPrice[N]	(終値 + 始値 + 高値 + 安値)/4
TR	TR(price)	トゥルーレンジインジケーター
TradeIndex	TRADEINDEX(n)	最後に実行された注文からn番目のバーの指標
TradePrice	TradePrice(n)	最後に実行された注文からn番目の価格レベル
Trailing	SET STOP TRAILING x	ポジション エントリー価格から x 単位でトレーリングストップを設定
%TRAILING	SET STOP %TRAILING x	ポジション エントリー価格から x %でトレーリングストップを設定
\$TRAILING	SET STOP \$TRAILING x	x €または\$でトレーリングストップを設定 (銘柄の通貨)
TriangularAverage	TriangularAverage[N](price)	三角移動平均
TRIX	TRIX[N](price)	3 重スムーズド指数移動平均
TypicalPrice	TypicalPrice[N]	代表値 (高値、安値及び終値の平均)
U		
コード	構文	機能
Undefined	a = Undefined	未定義の変数値の設定



V		
コード	構文	機能
Variation	Variation(price)	最後のバーの終値と現在のバーの終値の差(%)
Volatility	Volatility[S, L]	チャイキンボラティリティ
Volume	Volume[N]	ボリュームインジケーター
VolumeOscillator	VolumeOscillator[S,L]	ボリュームオシレーター
VolumeROC	VolumeROC[N]	変更の価格率ボリューム
W		
コード	構文	機能
WeightedAverage	WeightedAverage[N](price)	加重移動平均
WeightedClose	WeightedClose[N]	(2*終値), (1*高値) 及び (1*安値) の平均
WEND	WHILE/DO/WEND を参照	WHILE ループの 終了指示
WHILE/DO/WEND	WHILE (condition) DO (action) WEND	WHILE ループ
WilderAverage	WilderAverage[N](price)	ワイルダーの移動平均
Williams	Williams[N](close)	ウィリアムズ%R インジケーター
WilliamsAccumDistr	WilliamsAccumDistr(price)	ウィリアムズインジケーターの蓄積/分布
X		
コード	構文	機能
XOR	a XOR b	理論演習子 eXclusive OR
Y		
コード	構文	機能
Year	Year[N]	現在のバーよりn期間前のバーの年
Yesterday	Yesterday[N]	現在のバーよりn期間前のバーの日付
Z		
コード	構文	機能
ZigZag	ZigZag[Zr](price)	エリオット波動理論 ( Elliott waves theory) で導入されている Zig-Zag インジケーター
ZigZagPoint	ZigZagPoint[Zp](price)	Zp ポイントを基に計算されている、エリオット波動理論 ( Elliott waves theory) の Zig-Zag インジケーター



# 演算子

## コード

+

-

\*

/

=

<>

<

>

<=

>=

//

## 機能

加算演算子

減算演算子

乗算演算子

除算演算子

等価演算子

差分演算子

厳密不等価演算子(右の値が左の値より大きい)

厳密不等価演算子(左の値が右の値より大きい)

不等価演算子(右の値が左の値より大きいか同等)

不等価演算子(左の値が右の値より大きいか同等)

コメントラインの導入