

（ダイジェスト版）

MT4（メタトレーダー）のMQL4言語による

アメンボ式EA開発法

基礎編

初版： 2015年10月

本ダイジェスト版について

- 「総合目次」と「第1章」は、正規品と全く同じ内容を掲載しました。
「2章～9章」は、一部のダイジェスト画面等のみを掲載しました。

<正規品について>

- ◎正規品（有料）では、全章の「印刷」が可能です、じっくりと読んでいただく事を前提に「印刷」可としました。

また正規品には、本文中で解説する「MQL4コード」「DLL」「ヒストリーデータ」の全てを添付していますので、読者自身による記載内容の再現・検証にお使い頂けます。

- 正規品は「ZIP形式」で提供いたしますので、解凍してからご利用ください。

[正規品；本文、添付資料の構成] (2015年10月現在)

正規品は「電子書籍」、および「電子データ」から構成されています。

媒体	内容	頁数	備考
電子書籍 (PDF)	総合目次、前書き、序章	18	章ごとに 個別の電子書籍 にしてあります (合計；418頁)
	第1章；EA開発手順概説	57	
	第2章；EAコード開発と準備概要	25	
	第3章；バックテスト	19	
	第4章；最適化	61	
	第5章；EAの環境耐性評価	36	
	第6章；参考資料1（バックテスト関係）	32	
	第7章；参考資料2（ヒストリー・データ関係）	62	
	第8章；MQL4コード内容一覧	53	
電子データ	第9章；コード解説	55	章分けした フォルダーに 収録しています
	MQL4コード (.mq4)	14本	
	DLL（ダイナミック・リンク・ライブラリ）(.dll)	1個	
	ヒストリー・データ (.csv)	7組	

※MQL4コードは「.mq4」形式ですので、読者自身で色々試すことができます。

- 「正規品」購入ご希望の場合は、下記にアクセスしてください。

<http://fx-on.com/ebooks/> (電子書籍・ツール・インジ・FX)



商品が見つかり難い場合は、[検索]窓でキーワードを入力してください。



「アメンボ式・・・」と入力して [検索] してみてください。

下記の画面が見られるはずですが、(fx-on 上には、ダイジェスト版もあります)



(注) fx-on.com で初めて購入される場合；

ご購入には、無料の「会員登録」が必要となります。

希望商品をクリックしたときなどに、WEB 表示の一番下に

「fx-on を初めて利用する方へ」が表示されていますので、

無料の「会員登録」を行ってからログインして、ご購入の手続きをしてください。

fx-onを初めてご利用する方へ

商品のご利用まではかんたん3ステップ！

fx-onでは、会員様向けに、あなたの投資生活を豊かにするサービスをご用意しております。
また、商品のご購入は通常会員様に限らせていただいております。
今すぐ [会員登録](#) をして、fx-onを活用してみませんか！

ユーザー情報を入力し会員登録 → 案内に従い購入 → マイページからダウンロード

[会員登録する](#) [購入する Buy](#)

その他；

アメンボの「連絡先、ホームページ」；

E-mail； amenbo.k@gmail.com

WEB； <http://www.green.dti.ne.jp/sdimension/mql/>

※上記 WEB では、アメンボが 2011 年 7 月から掲載し始めた、様々な MT4 の関数や機能についての調査結果の詳細を参照することができます。

以上

<次頁からは、「ダイジェスト版」の本体内容です>

総合目次

前書き

序章

第1章； EA開発手順概説

1. 1 サンプルEAでバックテストを行ってみる
 1. 1. 1 「サンプルEA」によるバックテストを行う準備をする
 1. 1. 2 現時点の為替ペア「USDJPY」データでバックテストを行う
 1. 1. 3 サンプル「[2012_05_10] ヒストリー・データ」でバックテストを行う
1. 2 2つのバックテスト結果を観る
 1. 2. 1 使用するヒストリー・データにより、損益に差が出る
 1. 2. 2 バックテスト結果の見方（概略）
1. 3 「サンプルEA」の中身を覗く
 1. 3. 1 「サンプルEA」の基本仕様を確認する
 1. 3. 2 Bollin_EA_08.mq4 の「extern 変数（最適化変数）」について
 1. 3. 3 最適化とは何か？を再考する
1. 4 「Bollin_EA_08.mq4」を「2012_05_10」以外のヒストリー・データでも最適化してみた
 1. 4. 1 最適化後の extern 値
 1. 4. 2 バックテスト結果1
 1. 4. 3 バックテスト結果2
1. 5 MT4を「オフライン」に設定する方法
 1. 5. 1 プロキシサーバーの設定を利用する
 1. 5. 2 「無効な口座」を利用する

第2章； EAコード開発と準備概要

2. 1 解説のまえに
 2. 1. 1 解説ポイント
 2. 1. 2 開発手順（経緯）概要
2. 2 EA開発のサンプル（実例）概要とツールについて
 2. 2. 1 どの時点でのチャートで開発をスタートするか
 2. 2. 2 まず、インディケータで確認する
 2. 2. 3 「売買ルール」の基本構造と着目ポイント
2. 3 インディケータを用いたEAの事前検証
 2. 3. 1 検証用ツールを確認する
 2. 3. 2 準備・DLL導入と最新チャートの固定を行う
 2. 3. 3 売買シグナル全てを表示する
 2. 3. 4 「フ型」フィルターを組込む
 2. 3. 5 「フ型」フィルターの目視判断で、売買シグナルを選別する
 2. 3. 6 「Cross_Bollin_DLL」に「フ型」フィルターを組込む前後での「売買シグナル」表示
2. 4 EA「Cross_Bollin_EA.mq4」の作成
 2. 4. 1 「Cross_Bollin_Filtered.mq4」⇒「Cross_Bollin_EA.mq4」へと変換する
 2. 4. 2 「Cross_Bollin_Filtered.mq4」と「Cross_Bollin_EA.mq4」コード構造を比較する
 2. 4. 3 サンプルEA「Cross_Bollin_EA.mq4」の最終仕様を再確認する
2. 5 開発ツール「共有メモリ用DLL（shared_memory.dll）」の使い方について

第3章； バックテスト

- 3. 1 バックテスト実施について
 - 3. 1. 1 解説ポイント
 - 3. 1. 2 開発手順（経緯）概要
- 3. 2 バックテスト環境を統一する
 - 3. 2. 1 通貨ペアを設定する
 - 3. 2. 2 スプレッド（spread）値の設定
 - 3. 2. 2 使用するヒストリー・データ
- 3. 3 バックテストを実施する
 - 3. 3. 1 「サンプルEA（Cross_Bollin_EA.mq4）」のバックテストを実施する
 - 3. 3. 2 buy_IN、sell_IN 専用のEAで、バックテストを再実施する
 - 3. 3. 3 総合結果と考察
- 3. 4 補足（チョット休憩、雑知識）
 - 3. 4. 1 バックテスト時の特記事項・制限・および利用できない機能
 - 3. 4. 2 期待値と勝率
 - 3. 4. 3 スプレッドの壁（損益「0；ゼロ」モデル）

第4章； 最適化

- 4. 1 最適化の実施
 - 4. 1. 1 解説ポイント
 - 4. 1. 2 開発手順（経緯）概要
- 4. 2 「売りIN（sell_IN）」専用EA「Cross_Bollin_EA_ShortOnly.mq4」を最適化する
 - 4. 2. 1 最適化実施条件を整理する
 - 4. 2. 2 最適値探索を実行する
 - 4. 2. 3 結果と解析
 - 4. 2. 4 収益が最大になる「extern 変数値の組合せ」でバックテストしてみる
 - 4. 2. 5 最適化結果の観察
- 4. 3 「売・買」で extern 変数を分離したEA「Cross_Bollin_EA_Long&Short.mq4」を最適化する
 - 4. 3. 1 最適化実施条件の確認
 - 4. 3. 2 最適値探索を実行する
 - 4. 3. 3 結果と解析
 - 4. 3. 4 損益が最大になる「extern 変数値の組合せ」でバックテストしてみる
 - 4. 3. 5 最適化結果から「extern 変数値の最適組合せ」を選定しなおす
 - 4. 3. 6 最適化後のEAによるバックテスト結果を確認する
 - 4. 3. 7 期間の異なる「ヒストリー・データ」での最適化結果を比較する
 - 4. 3. 8 「遺伝的アルゴリズム」による最適化結果は、Build バージョンにより異なる
- 4. 4 補足（チョット休憩、雑知識）
 - 4. 4. 1 遺伝的アルゴリズムとは
 - 4. 4. 2 最適化で利用できない機能
 - 4. 4. 3 設定ファイル（.set）とは

第5章； EAの環境耐性評価について

- 5. 1 EAを評価する
 - 5. 1. 1 解説ポイント
 - 5. 1. 2 開発手順（経緯）概要
- 5. 2 EAを評価するとは？
- 5. 3 評価試験を実施する
 - 5. 3. 1 extern 値による変動耐性を観る
 - 5. 3. 2 スプレッド変動耐性を観る
 - 5. 3. 3 トレードの約定時間遅れ（ディレー）耐性を観る
 - 5. 3. 4 為替ペア依存性を観る
 - 5. 3. 5 チャート期間依存性を観る
- 5. 4 評価結果が判明したら、次にすべきこと
 - 5. 4. 1 本書では未実施の重要評価項目について（概要）
 - 5. 4. 2 FX 業者の形態について（概要）

第6章； 参考資料1（バックテスト関係）

- 6. 1 本章の解説手順
 - 6. 1. 1 解説ポイント
 - 6. 1. 2 解説概要
- 6. 2 バックテスト関連
 - 6. 2. 1 「必要・最低限の知識」をおさらいする
 - 6. 2. 2 バックテスト時に生成されるファイル（データ）類の内容
 - 6. 2. 3 各モデルでの「FXTファイル（「.fxt」データ）とテスト・レポート」の関係
 - 6. 2. 4 FXTデータ（ファイル）と補間

第7章； 参考資料2（履歴・データ関係）

- 7. 1 本章の解説手順
 - 7. 1. 1 解説ポイント
 - 7. 1. 2 チャートのいくつかの側面
- 7. 2 為替データとチャートの関係概要
 - 7. 2. 1 チャートの種類と機能による分類
 - 7. 2. 2 チャート設定機能の概要
 - 7. 2. 3 補足
- 7. 3 MT4の「終了プロセス・起動プロセス」とチャートの関係
 - 7. 3. 1 MT4終了時のチャート・データの処理
 - 7. 3. 2 MT4「終了⇒開始」時のチャート処理
 - 7. 3. 3 補足
- 7. 4 履歴・センターの使い方（機能）
 - 7. 4. 1 「履歴・データ（過去データ）」の扱い概念図
 - 7. 4. 2 履歴・センターの操作
- 7. 5 履歴・データの準備
 - 7. 5. 1 証券会社のアカウント・サーバーから入手する
 - 7. 5. 2 他の証券会社（ブローカー）のデータを利用するには
- 7. 6 履歴・データの扱い例
 - 7. 6. 1 データ切り出し（例）
 - 7. 6. 2 データ・マージ（合成例）

第8章； MQL4コード内容一覧

8.1 使用（開発）全コード一覧

- 8.1.1 「第1章」で初めて登場するコード
- 8.1.2 「第2章」で初めて登場するコード
- 8.1.3 「第3章」で初めて登場するコード
- 8.1.4 「第4章」で初めて登場するコード
- 8.1.5 「第5章」で初めて登場するコード
- 8.1.6 「第6章」で初めて登場するコード

第9章； コード解説

9.1 コード解説の考え方

- 9.1.1 本書に登場するMQL4のコード一覧と集約解説
- 9.1.2 集約解説するMQL4コードの親子関係（系譜）
- 9.1.3 作成EA間の差異概要

9.2 EA・インディケータのコード解説

- 9.2.1 EA「Cross_Bollin_EA_fitted.mq4」の詳細解説
- 9.2.2 EA「Cross_Bollin_EA_fitted_delayed.mq4」の詳細解説
- 9.2.3 その他EAについて
- 9.2.4 「Cross_Bollin_DLL.mq4」と「Filter_fu_type_DLL.mq4」を組み合わせについて
- 9.2.5 「Cross_Bollin_DLL.mq4」の詳細解説
- 9.2.6 「Filter_fu_type_DLL.mq4」の詳細解説
- 9.2.7 「Cross_Bollin_Filtered.mq4」の詳細解説
- 9.2.8 「Cross_Bollin_Filtered.mq4」からEA「Cross_Bollin_EA.mq4」への変換概要

<特記>

※サンプルEAのコード（.mq4ファイル）は、MQL4コード集に収録しています。

また、その内容一覧は「第8章」、詳細な解説は「第9章」を参照ください。

※本書中の画像の一部は、WindowsXPとMT4;Build646の組合せ動作時にキャプチャーしたものです、
その他は、Windows 8.1とMT4;Build711以降の組合せ時のキャプチャー画像です。

※本書中の各EAコードは **2015.11.15 更新 Build 910** にても正常動作を確認済みです

第1章； EA開発手順概説

1. 1 サンプルEAでバックテストを行ってみる
 1. 1. 1 「サンプルEA」によるバックテストを行う準備をする
 - (1) まず「サンプルEA ; Bollin_EA_08.mq4」をコピーします
 - (2) 次に、MT4を立ち上げ、データフォルダを開いて「Bollin_EA_08.mq4」を設定します
 1. 1. 2 現時点の為替ペア「USDJPY」データでバックテストを行う
 - (1) 一覧表（実施条件）でバックテスト条件を確認します
 - (2) MT4で、「ストラテジー・テスター」を表示し、データをセットしていきます
 - (3) バックテスト結果を大雑把に観る
 1. 1. 3 サンプル「[2012_05_10] ヒストリー・データ」でバックテストを行う
 - (1) [2012_05_10] ヒストリー・データを準備する
 - (2) ヒストリー・センターから、現在の「5分足、1分足」データを退避・保存します
 - (3) バッファ・サイズを変更します
 - (4) MT4を「オフライン」にします
 - (5) ヒストリー・センターから、現在の「5分足、1分足」データ（退避済み）を削除します
 - (6) ヒストリー・センターに、[2012_05_10] のデータを下記手順でインポートします
 - (7) [2012_05_10] データでバックテストを行う
1. 2 2つのバックテスト結果を観る
 1. 2. 1 使用するヒストリー・データにより、損益に差が出る
 - (1) バックテスト結果の「資産残高」推移を比較してみます
 - (2) ?なぜ、大きな違いが発生するのでしょうか
 1. 2. 2 バックテスト結果の見方（概略）
 - (1) エクセルに「Strategy Tester Report」のデータを取り込む
 - (2) ポイントだけ、拾い出してみます
1. 3 「サンプルEA」の中身を覗く
 1. 3. 1 「サンプルEA」の基本仕様を確認する
 - (1) [売買シグナル・ジェネレータ] 部分の概要
 - (2) [フィルター] 部分の概要
 - (3) 「サンプルEA」コードの大きな構造を観る
 1. 3. 2 Bollin_EA_08.mq4の「extern 変数（最適化変数）」について
 - (1) 一覧表（実施条件）でバックテスト条件を再度確認します
 - (2) Bollin_EA_08.mq4のコード中にある「extern 変数」の働きを確認する
 - (3) Bollin_EA_08.mq4の「extern 変数」の最適値を見つけるには（まとめ）
 1. 3. 3 最適化とは何か？を再考する
 - (1) 準備
 - (2) 1次元の最適化を考えてみる
 - (3) 次に、2次元の最適化を考えてみます
 - (4) 結果のまとめ

1. 4 「Bollin_EA_08.mq4」を「2012_05_10」以外のヒストリー・データでも最適化してみた
 1. 4. 1 最適化後の extern 値
 1. 4. 2 バックテスト結果1
 - (1) 資産グラフ（最適化の前後比較）
 - (2) バックテスト・レポート（最適化後）
 - (3) ストラテジー・テスター・レポート（最適化後）
 1. 4. 3 バックテスト結果2
 - (1) 資産グラフ（最適化の前後比較）
 - (2) バックテスト・レポート（最適化後）
 - (3) ストラテジー・テスター・レポート（最適化後）
1. 5 MT4を「オフライン」に設定する方法
 1. 5. 1 プロキシサーバーの設定を利用する
 1. 5. 2 「無効な口座」を利用する

※本章は Build 830 の画像で解説しています。（気になる「誤訳」もそのまま表示）
（本章中の各 EA コードは [2015.11.15 更新 Build 910](#) にても正常動作を確認済みです）

※サンプルEAのコード（.mq14 ファイル）は、MQL 4 コード集に収録しています。
また、その内容一覧は「第8章」、詳細な解説は「第9章」を参照ください。

第1章では、アメンボ式EA開発手順（最適化まで）を大雑把に理解するための解説を行います、個々の手順詳細については、第2章以降で解説していますので参照ください。

1. 1 「サンプルEA」でバックテストを行ってみる

- ・EAとは、簡単に言うと「売買ルール」をMQL4言語に翻訳（変換）したものです。
EAの良し悪しは、その「売買ルール」により略決まりますが、本書では「売買ルール」の作成法や検討には触れず、開発手順の全体を俯瞰し解説することに焦点を絞ります。
- ・初心者が開発手順を一通り理解するには、手ごろな「サンプルEA」を使った「バックテスト」から始めるのが近道と考え、本書ではこのための「サンプルEA（USDJPY、5分足用）」を準備しました。
まずは、この「サンプルEA」のバックテストから初めてみて下さい。
（「売買ルール」は、それ自体が極めて大きなテーマであり、多くの書籍・文献が存在します）

1. 1. 1 「サンプルEA」によるバックテストを行う準備をする

電子データ「MQL4コード（*.mq4ファイル）、DLL」のダウンロードは完了しているものとします。

(1) まず「サンプルEA ; Bollin_EA_08.mq4」をコピーします

表 1 ; 電子データ一式は下記の構成です。（再掲）

<p>[(1)アメンボ式EA開発法 基礎編]</p>	<p>[(3)DLL(ダイナミック・リンク・ライブラリ)]</p>
<p>[総合目次] [前書き] [序章] [第1章];EA開発手順概説 [第2章];EAコード開発と準備概要 [第3章];バックテスト [第4章];最適化 [第5章];EAの環境耐性評価 [第6章];参考資料1(バックテスト関係) [第7章];参考資料2(歴史カル・データ関係) [第8章];MQL4コード内容一覧 [第9章];コード解説</p>	<p>[第2章] ・shared_memory.dll</p>
<p>[(2)MQL4コード集(Build 610以降_)]</p>	<p>[(4)ヒストリー・データ]</p>
<p>[第1章] ・Bollin_EA_08.mq4 [第2章] ・Cross_Bollin_DLL.mq4 ・Cross_Bollin_EA.mq4 ・Cross_Bollin_Filtered.mq4 ・Filter_fu_type_DLL.mq4 [第3章] ・Cross_Bollin_EA_Long&Short.mq4 ・Cross_Bollin_EA_LongOnly.mq4 ・Cross_Bollin_EA_ShortOnly.mq4 [第4章] ・Cross_Bollin_EA_fitted.mq4 ・Cross_Bollin_EA_fitted_2014.mq4 ・Cross_Bollin_EA_fitted_2015.mq4 [第5章] ・Cross_Bollin_EA_fitted_delayed.mq4 [第6章] ・Empty_EA_1.mq4 ・Tick_Print_FXT_01.mq4</p>	<p>[第1章] [2012_05_10] ・USDJPYFXF1;2012.05.10.csv ・USDJPYFXF5;2012_05_10.csv [2014_10_17] ・USDJPYFXF1;2014.10.17.csv ・USDJPYFXF5;2014.10.17.csv [2015_05_07] ・USDJPYFXF1;2015.05.07.csv ・USDJPYFXF5;2015.05.07.csv [第3~5章;共通] [2013_09_17max] ・USDJPYFXF1;2013_9_17.csv ・USDJPYFXF5;2013_9_17.csv [第5章] [2013_09_26_big_data_multi_currency] ・AUDJPYFXF1;2013_9_27_big_data.csv ・AUDJPYFXF5;2013_9_27_big_data.csv ・EURJPYFXF1;2013_9_27_big_data.csv ・EURJPYFXF5;2013_9_27_big_data.csv ・USDJPYFXF1;2013_9_27_big_data.csv ・USDJPYFXF5;2013_9_27_big_data.csv [期間統一データ(2012_10_26~2013_09_17)] ・USDJPYFXF1;2012_12_24_2013_9_17.csv ・USDJPYFXF5;2012_10_26_2013_9_17.csv ・USDJPYFXF15;2012_10_26_2013_9_17.csv ・USDJPYFXF30;2012_10_26_2013_9_17.csv ・USDJPYFXF60;2012_10_26_2013_9_17.csv [第7章] [データ・マージ確認用] ・USDJPYFXF1440;2009_10~2011_12(異常値入り).csv ・USDJPYFXF1440;2011_01~2013_03.csv</p>

1章の解説に使用

手順1；[(2) MQL4 コード集 (Build610 以降)] フォルダを開き、さらに、その中の [第1章] フォルダを開きます。

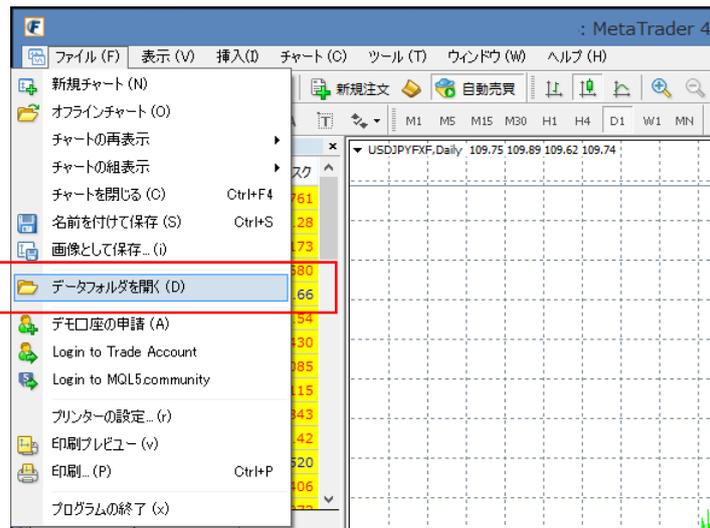
「Bollin_EA_08.mq4」が中に収録されています。

名前	更新日時	種類	サイズ
Bollin_EA_08.mq4	2014/06/05 18:54	MQL4 Source File	24 KB

手順2；この「サンプル EA」を、任意の場所にコピーしておいてください。

(2) 次に MT4 を立ち上げ後、データフォルダを開いて「Bollin_EA_08.mq4」を設定します

手順1；プルダウン・メニューの [ファイル] - [データフォルダを開く] を選択 (クリック)、



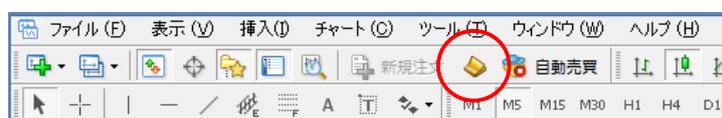
手順2；データフォルダが開いたら、さらに [MQL4] - [Experts] の順でフォルダを開いていきます

手順3；この [Experts] フォルダの中に、先ほど入手した「Bollin_EA_08.mq4」をコピーします。(いくつか、プレ・インストールされている EA が見えます)

名前	更新日時	種類	サイズ
Bollin_EA_08.mq4	2014/06/05 18:54	MQL4 Source File	24 KB
MACD Sample.ex4	2014/05/12 2:39	EX4 ファイル	8 KB
MACD Sample.mq4	2014/05/12 2:39	MQL4 Source File	7 KB
Moving Average.ex4	2014/05/12 2:39	EX4 ファイル	13 KB
Moving Average.mq4	2014/05/12 2:39	MQL4 Source File	6 KB
mqldata.dat	2014/06/03 1:28	DAT ファイル	4 KB

手順4；次に、下記の手順で MT4 のメタエディター (MetaEditor) を立ち上げます

- ・ [MetaEditor] アイコンを選択 (クリック)、
- またはプルダウン・メニューの [ツール] - [メタエディター] を選択 (クリック)

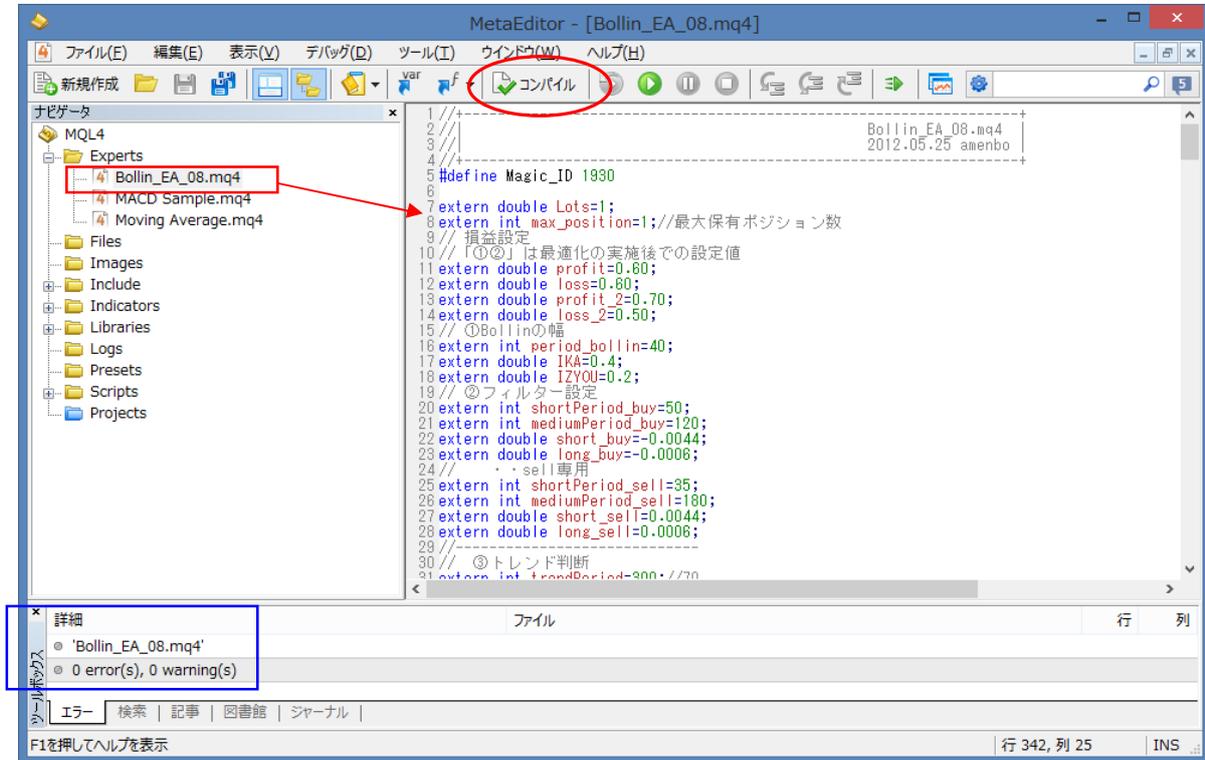


▼
「MetaEditor」ウィンドウが開きます

手順5；「Bollin_EA_08.mq4」を下記手順でコンパイルします

- ・メタエディター（MetaEditor）上で「Bollin_EA_08.mq4」を選択・表示させ、
[コンパイル] アイコンをクリックするか、
またはプルダウン・メニューの [ファイル] - [コンパイル] を選択（クリック）
します

（MT4 を再起動しても同様の結果が得られます）



エラーが発生していないことを確認します。

[Experts] フォルダの中身を確認すると、「Bollin_EA_08.ex4」が作成されています。

名前	更新日時	種類	サイズ
Bollin_EA_08.ex4	2014/07/01 23:36	EX4 ファイル	32 KB
Bollin_EA_08.mq4	2014/06/05 18:54	MQL4 Source File	24 KB
MACD Sample.ex4	2014/05/12 2:39	EX4 ファイル	8 KB
MACD Sample.mq4	2014/05/12 2:39	MQL4 Source File	7 KB
Moving Average.ex4	2014/05/12 2:39	EX4 ファイル	13 KB
Moving Average.mq4	2014/05/12 2:39	MQL4 Source File	6 KB
mqldata.dat	2014/07/01 23:36	DAT ファイル	4 KB

1. 1. 2 現時点の為替ペア「USDJPY」データでバックテストを行う

- サンプルEA ; 本書で準備済みの「Bollin_EA_08.mq4」を使います
- ヒストリー・データ ;

読者は、お使いのMT4に保存されている最新の「USDJPY」データを使用してください。

- ・ 本書では電子データ一式中の [(4) ヒストリー・データ] フォルダ中の、さらに下の [第1章] フォルダにある [2015_05_07]データ（1分足と5分足）を使って解説します。

(1) 一覧表（実施条件）でバックテスト条件を確認します

項目	内容	備考
エキスパート・アドバイザー	サンプルEA ; Bollin_EA_08.mq4	
通貨ペア	USDJPY	本書ではUSDJPYFXF
期間	M5（5分足）	
モデル	全ティック（Every tick）	
スプレッド	2	
ヒストリー・データ	※本書 ; [2015_05_07] フォルダ中のデータ ※読者 ; MT4に保存された現状データ	M1とM5データを使う

※読者が、何時の時点で本書を読むのかが判りませんので、本書の解説では事前に準備したヒストリー・データ（本書の原稿を書いている最中に収集したデータ）を使います。

(2) MT4で、「ストラテジー・テスター」を表示し、データをセットしていきます

手順1 ; [Strategy Tester] のアイコンを選択（クリック）、

またはプルダウン・メニューの [表示] - [ストラテジーテスター] を選択（クリック）

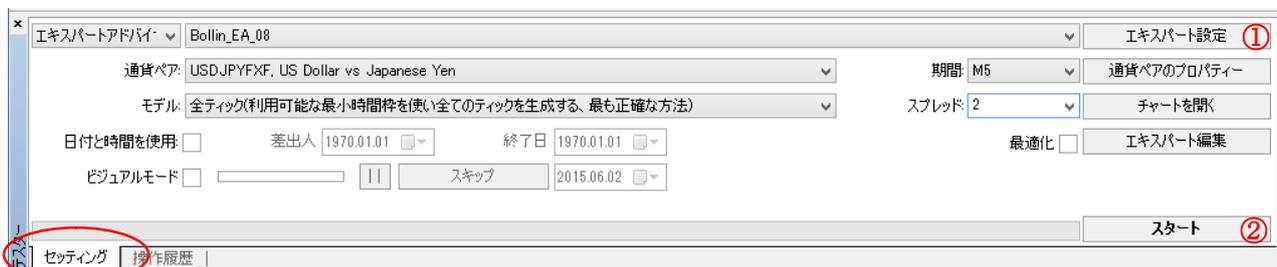


「ストラテジーテスター」の [セッティング] タブが表示されます

手順2 ; 「エキスパートアドバイザー」「通貨ペア」「モデル」「期間」「スプレッド」を設定します。

それぞれ右隅の [V] 部をクリックして、表示される候補から選んでいき、
下図の様に設定します。

(通貨ペアは [気配値表示] 表示中のものからのみ選択可能、詳細は3章「3.2.1」参照)



(拡大) [セッティング] タブ画面 ; 「←」印が設定する項目



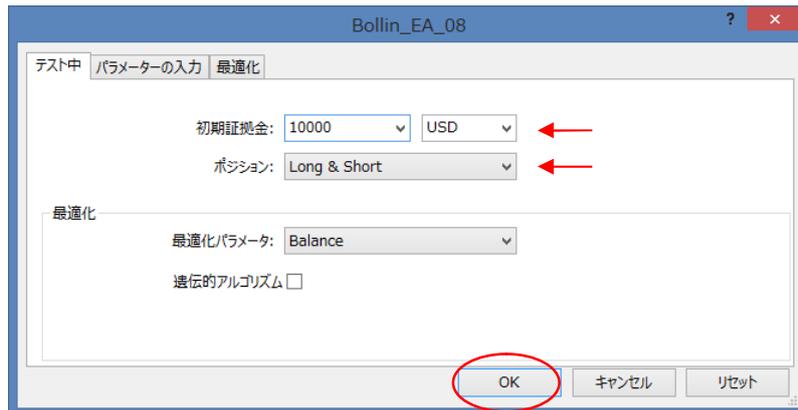
- ・本書では、条件を一定にするため USDJPY の [スプレッド] は、「2」としました。休日など、サーバーが未稼働のときは [スプレッド] で「現在値」を選択すると USDJPY の場合で「9」など、非常に大きな値になるので注意が必要です。

《注意》 Build 700、800 代（含 840）では 600 代には無かった下記「誤訳」が発生中です。

誤訳；「差出人」 ← 正解；「開始日」

本章では画面合成等の修正は行わずに、そのまま使用しました。

手順 3；[エキスパート設定] (①) を選択し、[テスト中] タブの設定内容が下図の様に設定されていることを確認します。



[OK] を選択（クリック）

（参考）バックテストの場合は、[最適化パラメータ] と [遺伝的アルゴリズム] の設定内容は結果や表示に影響しません。

手順 4；最後に [スタート] (②) を選択（クリック）します

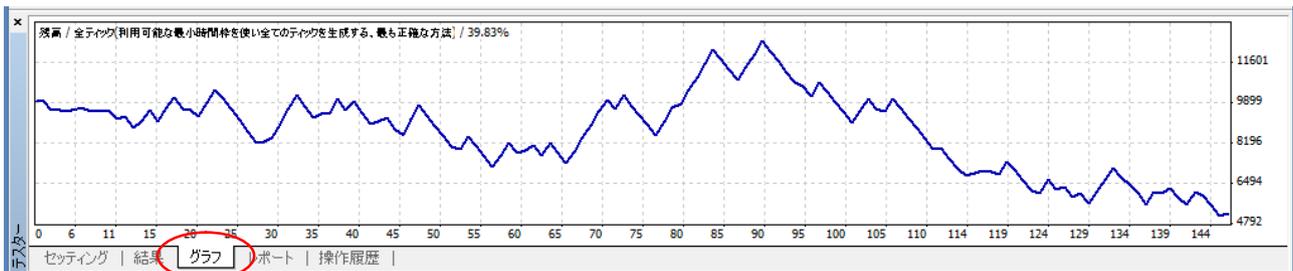
▼
バックテストが開始されます



▼
バックテストが終了しました。

(3) バックテスト結果を大雑把に観る

① [グラフ] タブ (資産残高) を開いてください



「なんだ！全然良く無いじゃないか」とガッカリしないでください。
 多分、読者の結果も同じようになったかもしれませんが、もし良い結果が出たら幸運でした。
 ○この作業は、EAの本質を理解するのに必要なステップです。

※上記結果は [2015_05_07] ヒストリー・データを使った場合です。

(読者の MT4 に保存されているヒストリー・データではどの様になりましたか?)

② [レポート] タブを開いてください

テストバー数	66330	モデルティック数	2420222	モデリング品質	39.83%
不整合チャートエラー	3				
初期証拠金	10000.00			スプレッド	2
総損益	-4735.06	総利益	25269.96	総損失	-30005.02
プロフィットファクター	0.84	期待利得	-32.43		
絶対ドローダウン	4931.95	最大ドローダウン	7744.75 (60.45%)	相対ドローダウン	60.45% (7744.75)
総取引数	146	ショートポジション(勝率%)	81 (34.57%)	ロングポジション(勝率%)	65 (53.85%)
		勝率(%)	63 (43.15%)	負率(%)	83 (56.85%)
	最大	勝トレード	701.14	負トレード	-480.68
	平均	勝トレード	401.11	負トレード	-361.51
	最大	連勝(金額)	7 (3618.30)	連敗(金額)	6 (-2325.31)
	最大化	連勝(トレード数)	3618.30 (7)	連敗(トレード数)	-2325.31 (6)
	平均	連勝	2	連敗	3

③次に、テストレポートを下記の手順で開いてください

・[レポート] タブ画面上で [右クリック] - [レポートの保存] を選択 (クリック) します

テストバー数	66330	モデルティック数	2420222	モデリング品質	39.83%
不整合チャートエラー	3				
初期証拠金	10000.00			スプレッド	2
総損益	-4735.06	総利益	25269.96	総損失	-30005.02
プロフィットファクター	0.84	期待利得	-32.43		
絶対ドローダウン	4931.95	最大ドローダウン	7744.75 (60.45%)	相対ドローダウン	60.45% (7744.75)
総取引数	146	ショートポジション(勝率%)	81 (34.57%)	ロングポジション(勝率%)	65 (53.85%)
		勝率(%)	63 (43.15%)	負率(%)	83 (56.85%)
	最大	勝トレード	701.14	負トレード	-480.68
	平均	勝トレード	401.11	負トレード	-361.51
	最大	連勝(金額)	7 (3618.30)	連敗(金額)	6 (-2325.31)
	最大化	連勝(トレード数)	3618.30 (7)	連敗(トレード数)	-2325.31 (6)
	平均	連勝	2	連敗	3



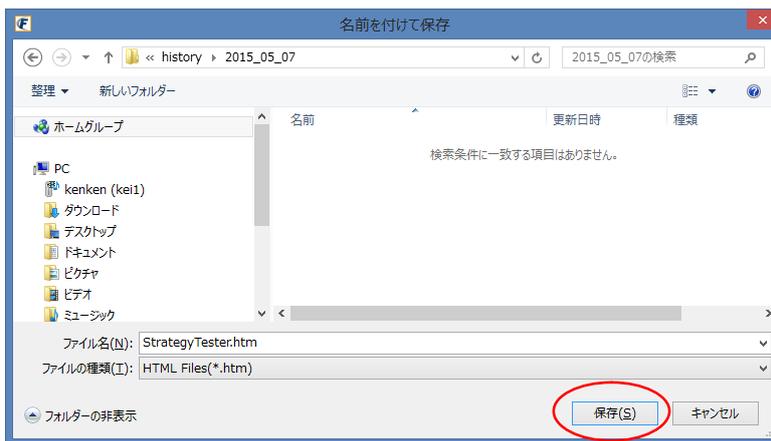
○保存先「フォルダー」等を選択する「名前を付けて保存」ウィンドウが表示されます

デフォルトでは、

[ファイル名] ; StrategyTester.htm

[ファイルの種類] ; HTML です。

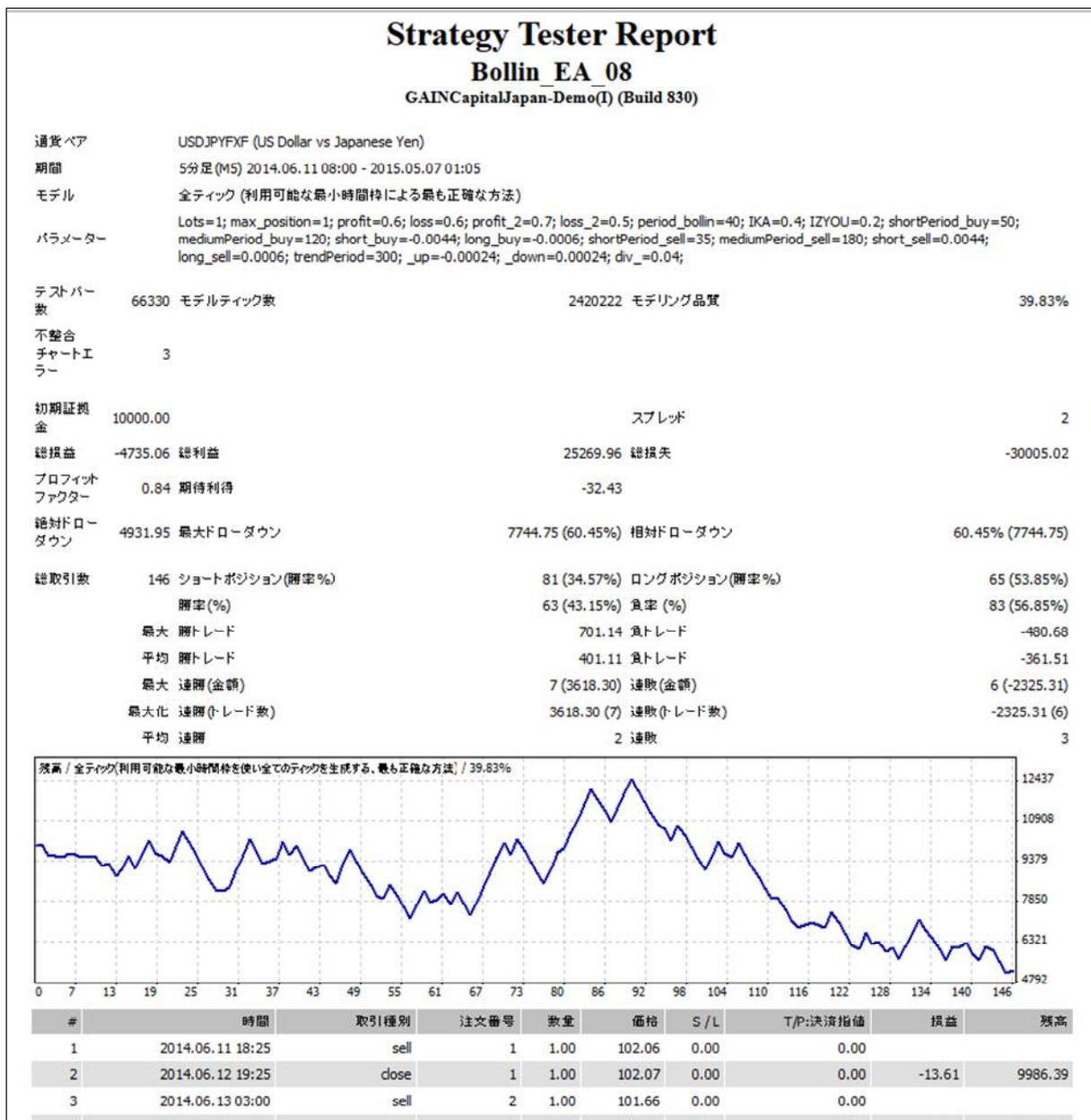
(保存先ホルダーやファイル名は任意に設定可能)



※本書ではデフォルト設定のまま、[保存] を選択（クリック）



テスト・レポートが保存と同時に表示されます（※内容の読み方については「1. 2」を参照）



1. 1. 3 サンプル「[2012_05_10] ヒストリー・データ」でバックテストを行う

- ・次に「1. 2. 1」と同一のEAと条件で、[2012_05_10] フォルダ内のヒストリー・データを使って、バックテストを実施してみます。
- ・資料 [(4) ヒストリー・データ] フォルダの中の [第1章] フォルダ中に、さらに [2012_05_10] フォルダがあり、その中に必要なデータが入っています。(1.1.1 (1) ; 表1を参照)

(1) [2012_05_10] ヒストリー・データを準備する

[2012_05_10] をフォルダーの中身；

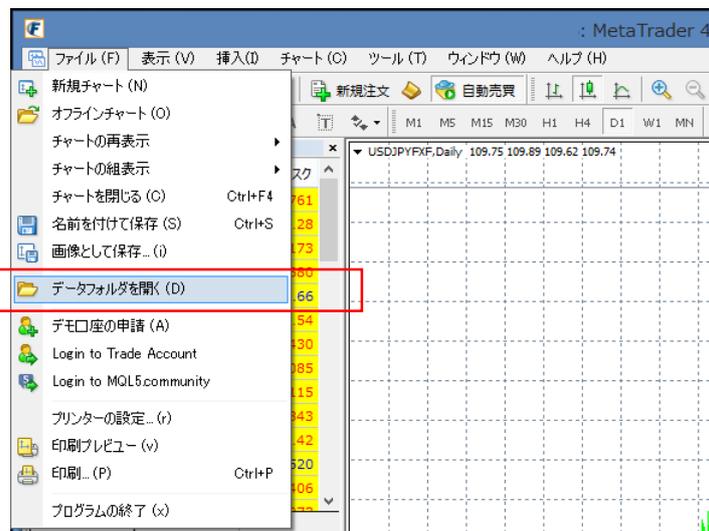
[2012_05_10] ヒストリー・データ		備考
USDJPYFXF5;2012.05.10.csv	5分足データ	バックテスト対象
USDJPYFXF1;2012.05.10.csv	1分足データ	期間が1つ下のデータ

※バックテストでは、直下の周期データ（期間）を参照出来るほうが精度が向上するので M5 でバックテストをすることを前提に、1分足データを添付しました。

- ・5分足データは、「2012.01～2012.04末」の範囲のデータを再現用に切り出したものです。

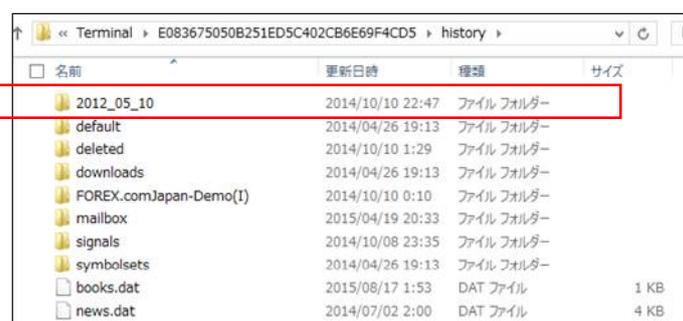
名前	更新日時	種類	サイズ
USDJPYFXF1;2012.05.10.csv	2012/05/10 2:09	Microsoft Excel ...	2,913 KB
USDJPYFXF5;2012.05.10.csv	2013/12/18 0:30	Microsoft Excel ...	1,142 KB

手順1 ; プルダウン・メニューの [ファイル] - [データフォルダを開く] を選択 (クリック)、



手順2 ; データフォルダが開いたら、さらに [history] フォルダを開きます

手順3 ; この [history] フォルダの中に、[2012_05_10] をフォルダごとコピーします

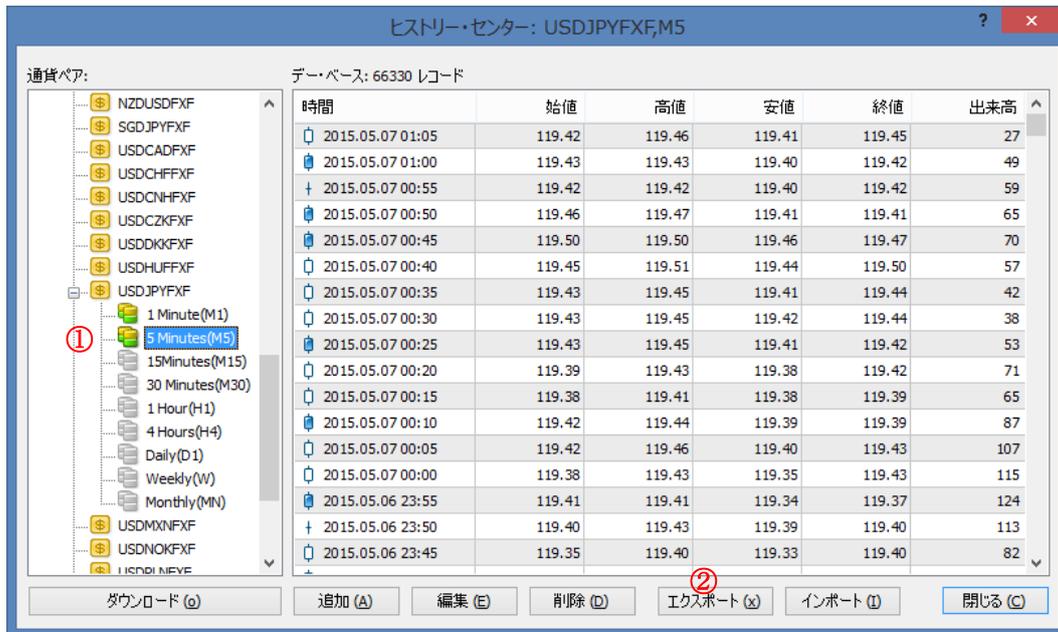


(2) ヒストリー・センターから、現在の「5分足、1分足」データを退避・保存します

※実際にトレードを行う予定の業者のヒストリー・データは、出来れば定期的に保存しておくことを推奨します。

手順1 ; 「history」ホルダー内に適当なホルダーを作る
(本書では [2015_05_07] とします)

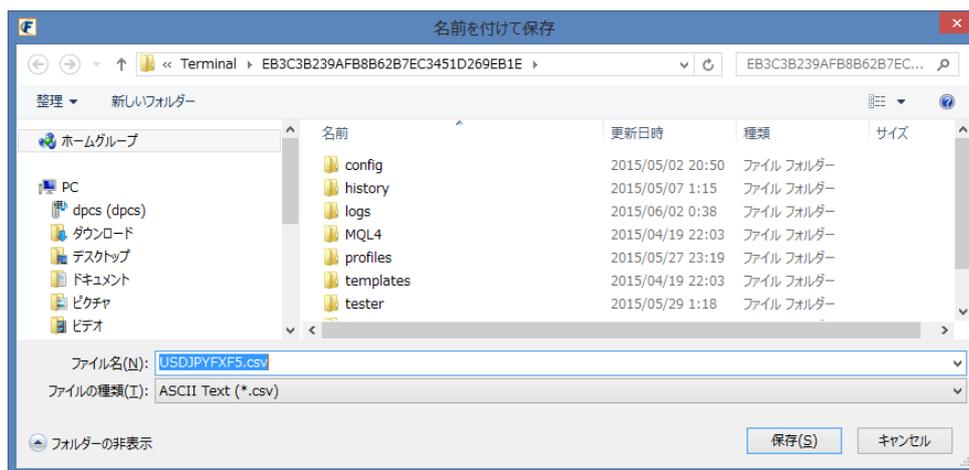
手順2 ; プルダウン・メニューの [ツール] - [ヒストリーセンター] を選択 (クリック)、



「ヒストリー・センター」ウインドウが開きます

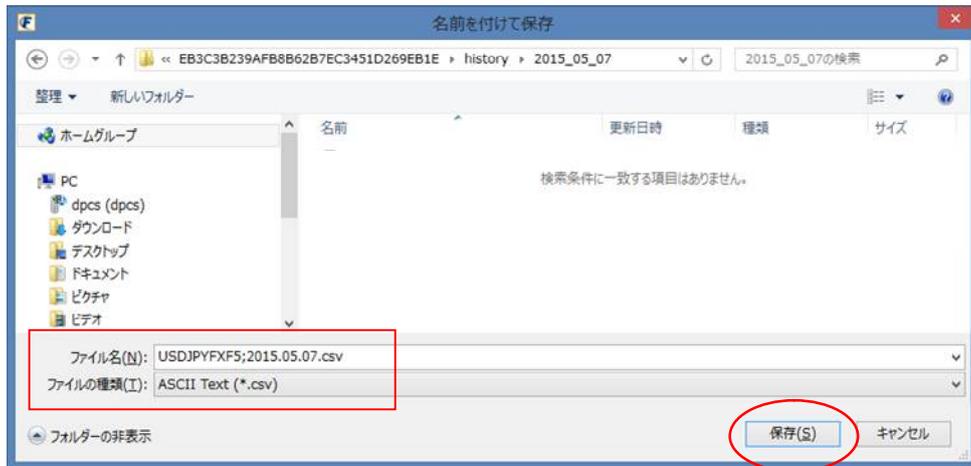
手順3 ; [5Minutes] (1) をダブル・クリックし、M5 データを表示します

手順4 ; [エクスポート] (2) を選択 (クリック)



保存先を決める [名前を付けて保存] ウインドウが表示されます

- 手順5 ; [ファイル名] を、例えば「USDJPYFXF5; 2015.05.07」（日付を付加）に変更し、
[ファイルの種類] は「ASCII Text(*.csv)」のままで、
保存先を「history¥2015_05_07」フォルダに変更して、[保存] を選択（クリック）



5分足データが「history¥2015_05_07」フォルダに保存されます。

- <1分足データも同様の手順で「history¥2015_05_07」フォルダにエクスポートします>



名前	更新日時	種類	サイズ
USDJPYFXF1;2015.05.07.csv	2015/05/07 1:08	Microsoft Excel ...	3,264 KB
USDJPYFXF5;2015.05.07.csv	2015/05/07 1:08	Microsoft Excel ...	3,174 KB

※現在のデータが「csv」形式で保存（退避）できました。

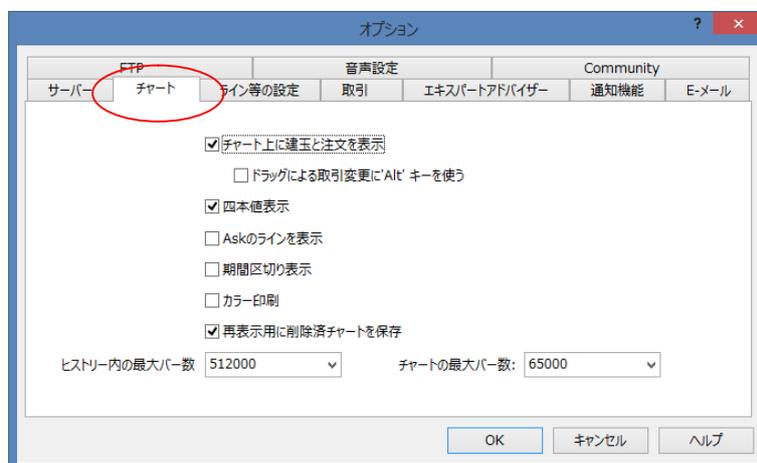
(3) バッファ・サイズを変更します

- ・ヒストリー・データを目一杯利用できるようにするため、以下の手順で、バッファサイズを最大に設定しておきます。

- 手順1 ; プルダウンメニューの [ツール] - [オプション] を選択（クリック）します

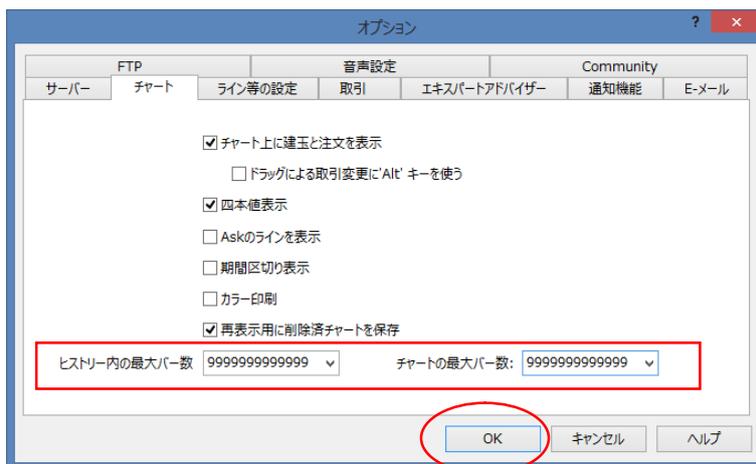


[オプション] ウィンドウが表示されたら、[チャート] タブを選択（クリック）



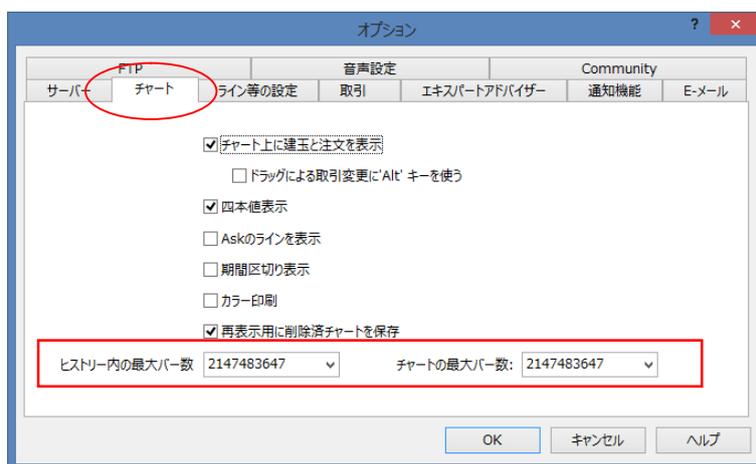
注 ; Build バージョンによってはタブが横一列に表示される場合もあります。

手順2；[履歴内の最大バー数] と [チャートの最大バー数] に、
「9999・・9999」と「9」を10個以上入れます。



[OK] を選択します

手順3；確認のため、もう一度 [オプション] ウィンドウを表示させ、
[チャート] タブを選択します



※ 2項目ともに、自動的に「最大バッファサイズ」に設定されています。

(4) MT4を「オフライン」にします

- ・バックテスト条件を固定する（一定にする）ためには、サーバーが止まっている休日等以外の日には、MT4はオフラインにしておく必要があります。

○バックテストや最適化を実行する場合は、常に「オフライン」にすると決めましょう。

※オンライン状態では、例えば10年前のデータでバックテストする場合にも、サーバーからの最新データが追加されていくため、古いデータと最新データの接続部が不連続となります。

- オフラインにする方法は何通りかありますが、Buildバージョン「例 880、890」では使えない方法があり注意が必要です。(Build No 900以降では840以前と同じ方法が可能です)

MT4のオフライン設定方法（例）；

方法	オフライン設定方法	表示	Build		
			419 ~ 840	880、890	900 ~
A	無効なプロキシサーバーを設定し、再起動	[回線不通!]	○	×	○
B	接続後、「無効な口座」に設定	[無効な口座]	(未確認)	○	○
C	接続後、ネット用コネクタを切断する	[回線不通!]	(未確認)	○	—

- 上記、「AとB」の具体的な「設定手順」は、

本章の「1. 5 MT4を「オフライン」に設定する方法」を参照ください。

◎本書での記述方法；

Build 880、890と同様のバグが今後も発生した場合を考えて、回線のオフライン状態を示す表示は、下記の様にします。

- ・メイン（A；1.5.1の方法）； [回線不通!]



- ・サブ（B；1.5.2の方法）； [無効な口座]



（推奨する方法は「A」（再起動を含む）です、880と890では「B」を使います）

Build 900からバグが修正されて、方法「A」がまた使えるようになりました。

※MT4では、バージョン・アップでデグレード「すなわち以前は問題なかった機能等に、新たにバグが作り込まれる」ことが結構あるので要注意です。

例；Build600バージョン・アップ時に、「開始日、終了日」⇒「差出人、終了日」へと、誤訳（デグレード）したまま、Build890まで続いていました。

このバグはBuild 900で一部修正されてストラテジーテスターの[セッティング]タブでは「開始日・終了日」になりましたが、一方[オフラインチャート一覧]は未修正のままです。

Build880、890におけるデグレード内容；

※Build「880、890」では、「A」の設定（含、再起動）を実施すると、バックテスト時に「error 131」エラーが発生します。

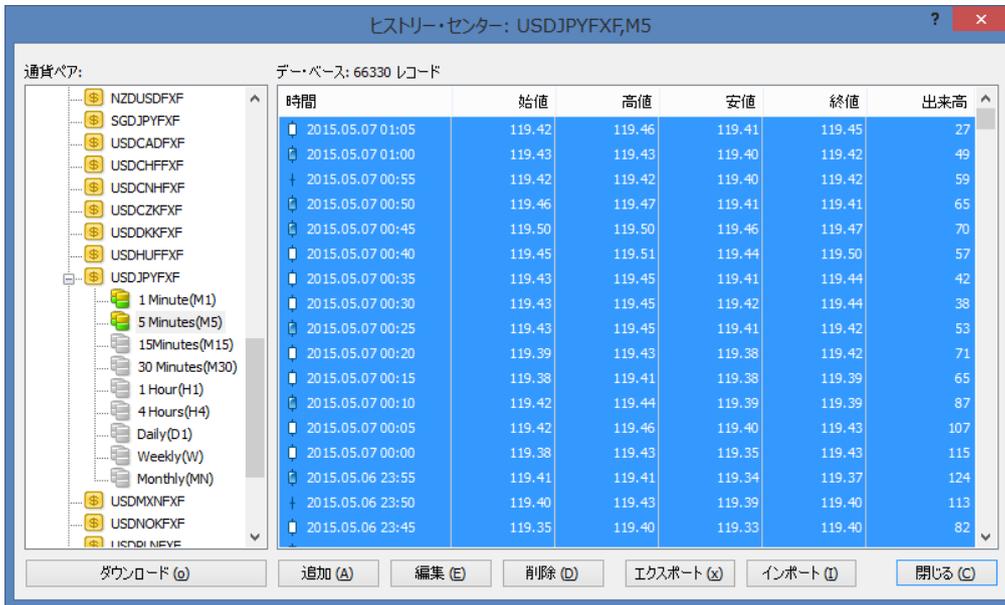
これは「最小ロット数よりも小さいロット数でオーダーした」という意味ですが、原因は「回線不通」状態にしてから再起動すると、Build880、890ではシステム上で設定されるべき「取引の最小ロットを認識できなくなるらしい」ことにあるようです。（このバグは「回線不通」状態に設定後、再起動しなければ回避できます）

(5) ヒストリー・センターから、現在の「5分足、1分足」データ（退避済み）を削除します

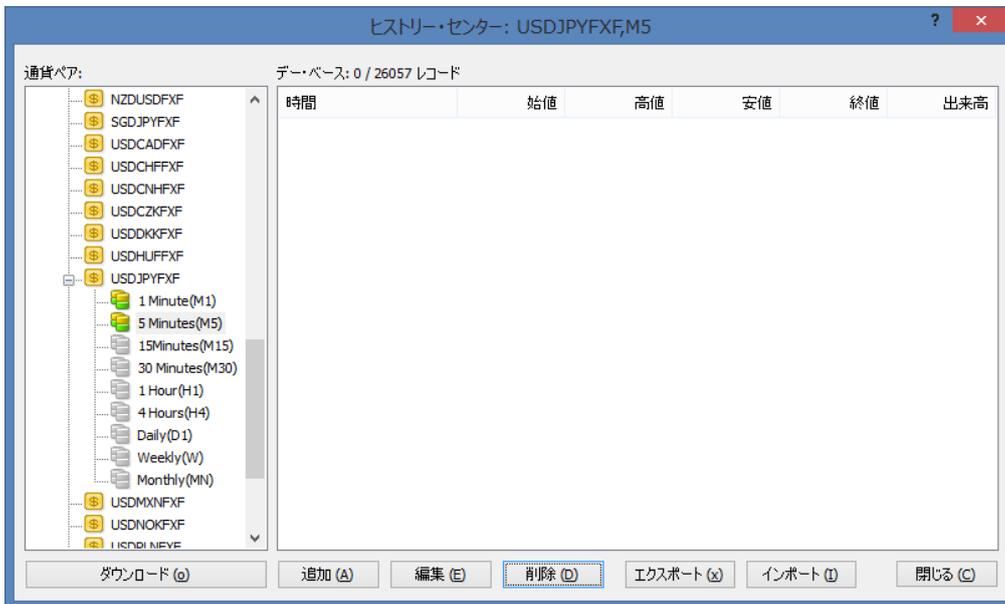
- ・ヒストリー・データを入れ替えるため、まず現状の「5分足」データを削除します。

手順1；プルダウン・メニューの [ツール] - [ヒストリーセンター] を選択し、

ヒストリーセンターを表示したら、「USDJPY；5分足」データを全て選択します



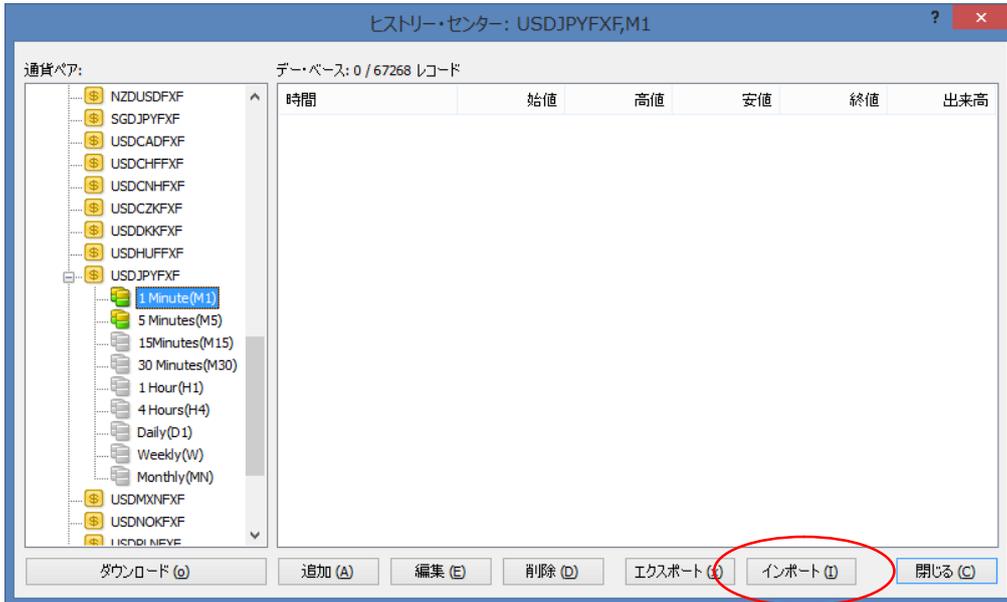
手順2；全てのデータを選択した状態で、[削除] を選択すると空っぽになります



- ・「USDJPY；5分足」の次に「USDJPY；1分足」も同様にして全て削除します・

(6) ヒストリー・センターに、[2012_05_10] のデータを下記手順でインポートします

手順1 ; 本稿では「1分足」からインポートするので、先ず [1Minute] をダブルクリックし、次に [インポート] を選択します

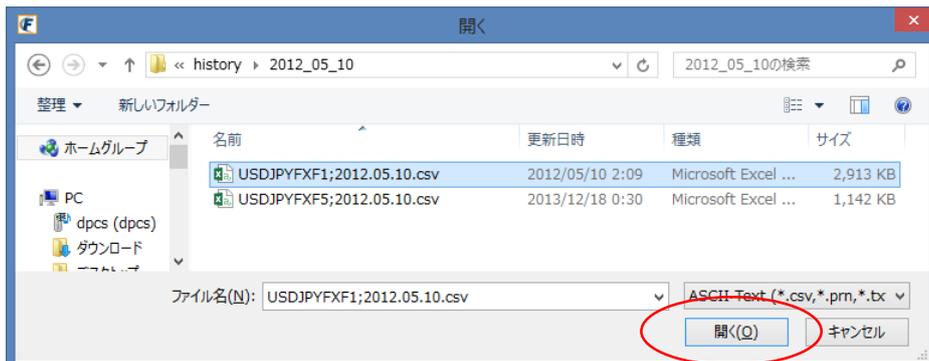


手順2 ; [インポート] ウィンドウが表示されたら、[ブラウズ...] を選択しインポート先を指定

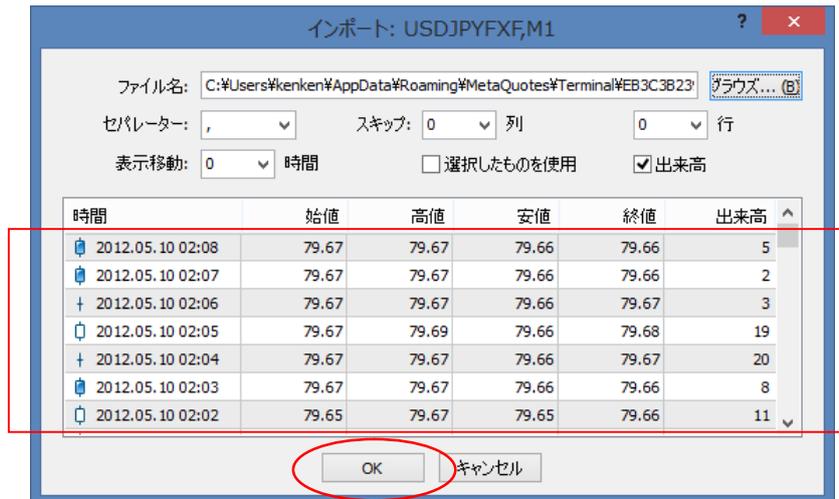


(注) このとき表示されているデータは前回操作したデータです (目的のデータではない)

手順3 ; フォルダを何回か移動し、history / [2012_05_10] フォルダとしてコピー済みの「1分足 (FXF1)」データを指定し、次に [開く] を選択します

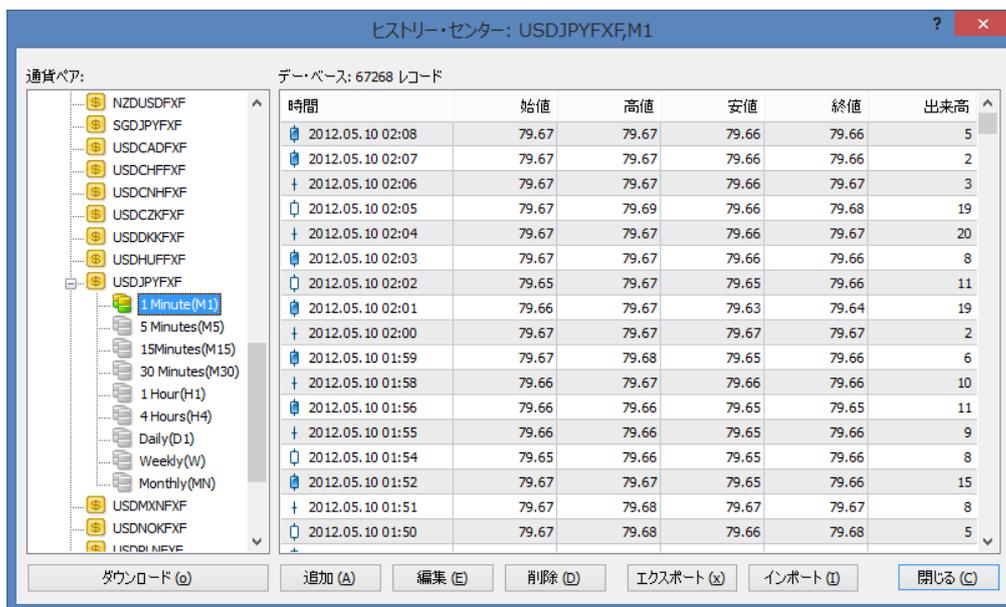


手順4；インポートする目的のデータ「2012.05.10 **:**」が表示されたかを確認します



目的のデータが表示されたら、[OK] を選択

手順5；[履歴・センター] の [USDJPY；1分足] に「2012.05.10 **:**」のデータが表示されたら、インポートの完了です



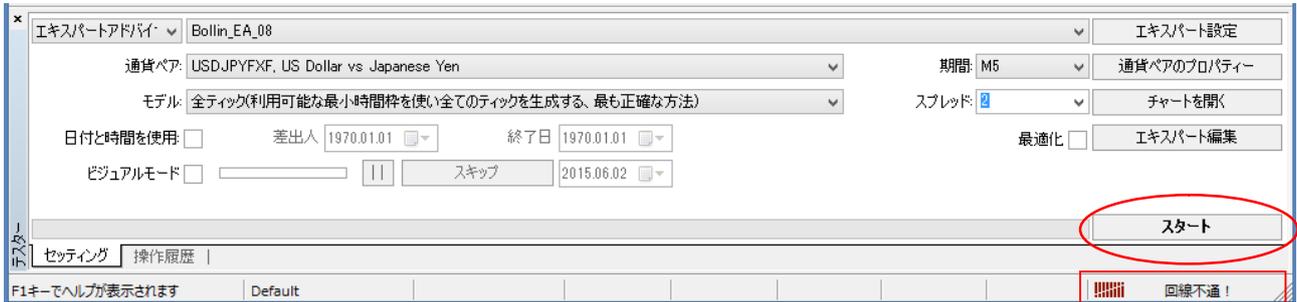
・ ・ 「USDJPY；1分足」に続き、「USDJPY；5分足」も同様にインポートします ・ ・

手順6；インポート終了後に [履歴・センター] の [閉じる] を選択し、
[2012_05_10] データの設定を終えます

(注) 入れ替えたはずの履歴・データがチャート上にまだ反映されていない場合は、一度「M1、M5」以外のチャートを表示させてから、再度 M1、M5 チャートを表示させてください。(変更が反映されます)

(7) [2012_05_10] データでバックテストを行う

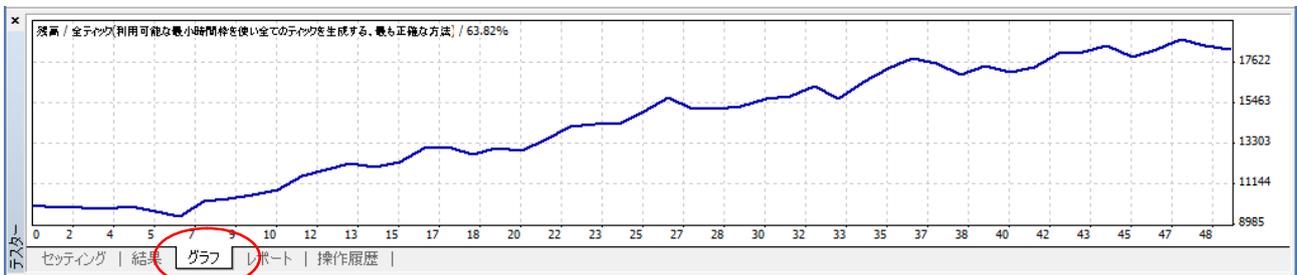
※「1. 1. 1」とデータ以外は「全く同一設定」でバックテストを行います
但し、MT4は必ずオフラインの状態にしておきます



[スタート] を選択

<バックテストの実施結果>

① [グラフ] タブ (資産残高)

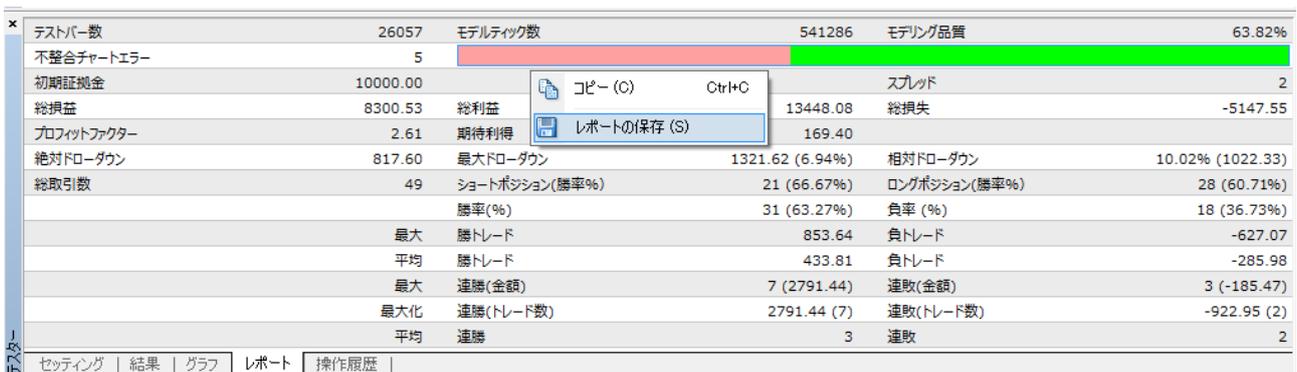


② [レポート] タブ

テストバー数	26057	モデルティック数	541286	モデリング品質	63.82%
不整合チャートエラー	5				
初期証拠金	10000.00			スプレッド	2
総損益	8300.53	総利益	13448.08	総損失	-5147.55
プロフィットファクター	2.61	期待利得	169.40		
絶対ドローダウン	817.60	最大ドローダウン	1321.62 (6.94%)	相対ドローダウン	10.02% (1022.33)
総取引数	49	ショートポジション(勝率%)	21 (66.67%)	ロングポジション(勝率%)	28 (60.71%)
		勝率(%)	31 (63.27%)	負率(%)	18 (36.73%)
	最大	勝トレード	853.64	負トレード	-627.07
	平均	勝トレード	433.81	負トレード	-285.98
	最大	連勝(金額)	7 (2791.44)	連敗(金額)	3 (-185.47)
	最大化	連勝(トレード数)	2791.44 (7)	連敗(トレード数)	-922.95 (2)
	平均	連勝	3	連敗	2

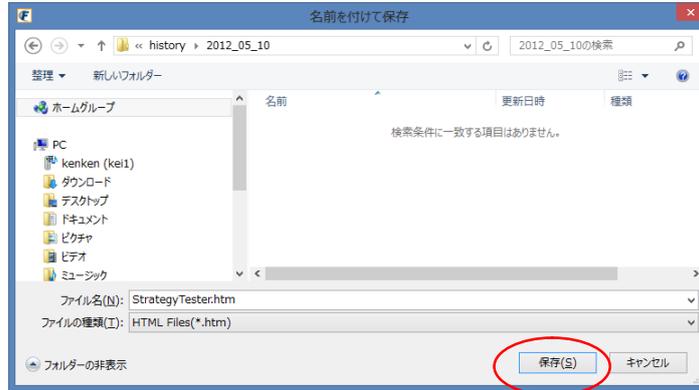
③テストレポートを下記の手順で開いてください

・[レポート] タブ画面上で [右クリック] - [レポートの保存] を選択します



○保存先「フォルダー」等を選択する「名前を付けて保存」ウインドウが表示されます、保存先ホルダーやファイル名は任意に設定可能です。

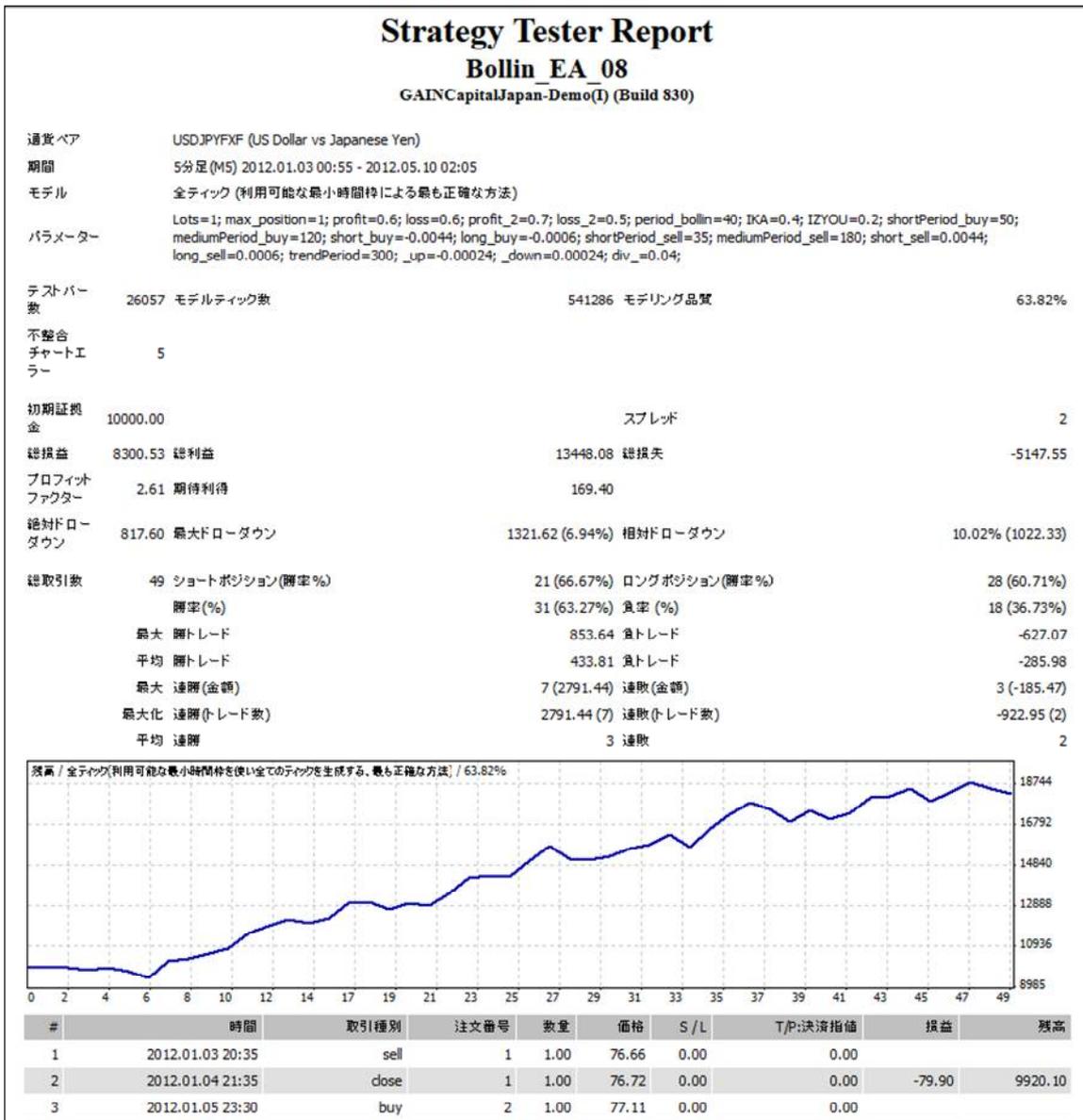
デフォルト； [ファイル名] StrategyTester.htm 、[ファイルの種類] HTML



※本書ではデフォルト設定のまま、[保存] を選択



テスト・レポートが保存と同時に表示されます

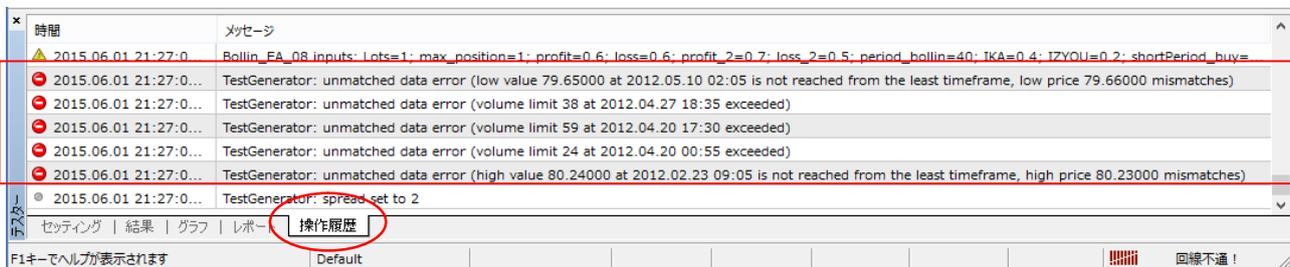


④ [操作履歴] タブ・・重大なエラー等が発生していないか、必ず確認しましょう

- ・データ先端部では、「Volume」値が不整合と言うエラーが良く出ます。

これは切り出した「M5足」1個のVolume値に相当する「M1足」5個分のデータを切り出し損ねた場合に、Volume値の不整合表示として発生するものです。

(このレベルであれば、バックテスト結果への影響は殆どありませんので、先へ進みます)



- 「バックテスト」や「最適化」を行う場合は、結果を観ると共に、必ず「操作履歴」内容を確認し、処理中に重大なエラー等が発生していないかを確認することを習慣にしてください。エラー内容によっては原因の対策後に再度「バックテストや最適化」を行う必要があります。

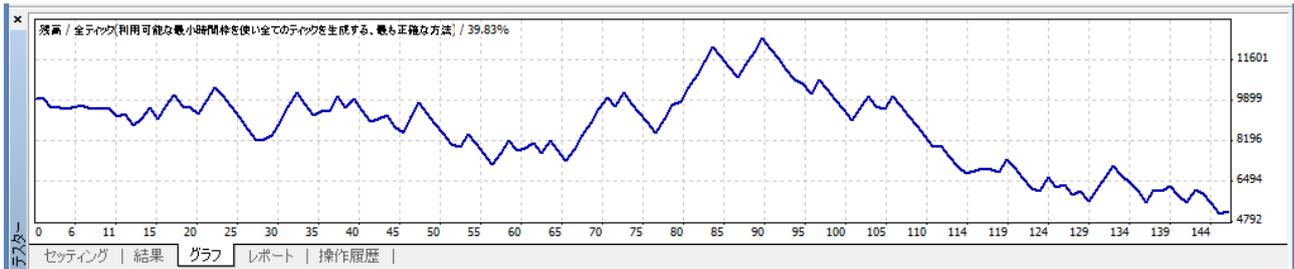
1. 2 2つのバックテスト結果を観る

- ・EA を入手あるいは作成したら、先ずバックテストして、その結果を観ることが鉄則です。
「サンプル EA」についても、バックテスト評価が第1歩ですので内容を検討します。

1. 2. 1 使用するヒストリー・データにより、損益に差が出る

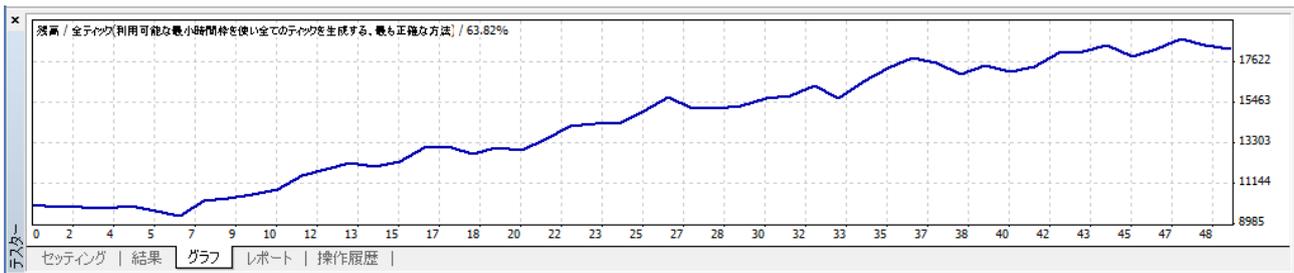
(1) バックテスト結果の「資産残高」推移を比較してみます

- ①読者の最新（為替）データでバックテストした結果で置き換えて、お読み下さい；
- ・ ・ 本書では例として「2015_05_07」（～2015年05月）時点のデータで示しました
※読者の結果は、上記より「良くなったり、悪くなったり」だと思います。



⇕ この違いは何故か？

- ②添付「2012_05_10」（2012年1月～2012年4月）の為替データでバックテストした結果；



(2) ?なぜ、大きな違いが発生するのでしょうか

理由は、添付の「サンプル EA」が何時でもどの様な市場でも「利益を叩き出せる実力」が有るわけでは無く、市場の状況（チャート）に依存するからです。

- 違いの発生原因1； 市場（チャート）トレンドの違い
- ①2012年5月近辺は「円高進行」局面であった、一方、
 - ②2015年5月近辺は「円安進行」局面でした
- 違いの発生原因2； EAの仕様
- ①「サンプル EA ; Bollin_EA_08.mq4」はトレンド相場対応、ですが
 - ②レンジ相場や急変（ブレイク発生）相場には対応していません

言い換えると、優れた EA であるためには、下記のアルゴリズム組込が必要と考えます。

- ①「レンジ、トレンド」両相場に対応
- ②市場の状況に合わせて「EAのパラメータ」を2～3段階に自動変更
- ③「市場の急変、時間での管理、ニュース内容」への対応 など

残る疑問；

それにしても、なぜ「2012_05_10」（2012年1月～2012年4月）の為替データでは、資産カーブは綺麗な「右上がり曲線」を描くのでしょうか。

大まかな種明かしは本章の「1. 3と1. 4」で行います、詳細内容は第4章を参照ください。

1. 2. 2 バックテスト結果の見方（概略）

- ・ 詳細な読み方は「第3章；バックテスト、第6章；参考資料1」に譲り、本節ではテストレポートを、ザッと眺めることにします。
- ・ ここでは、添付フォルダ [2012_05_10] の為替ペアデータでバックテストした結果をエクセルに取り込み、眺めることにします。（[2015_05_07] による結果も同様に処理します）

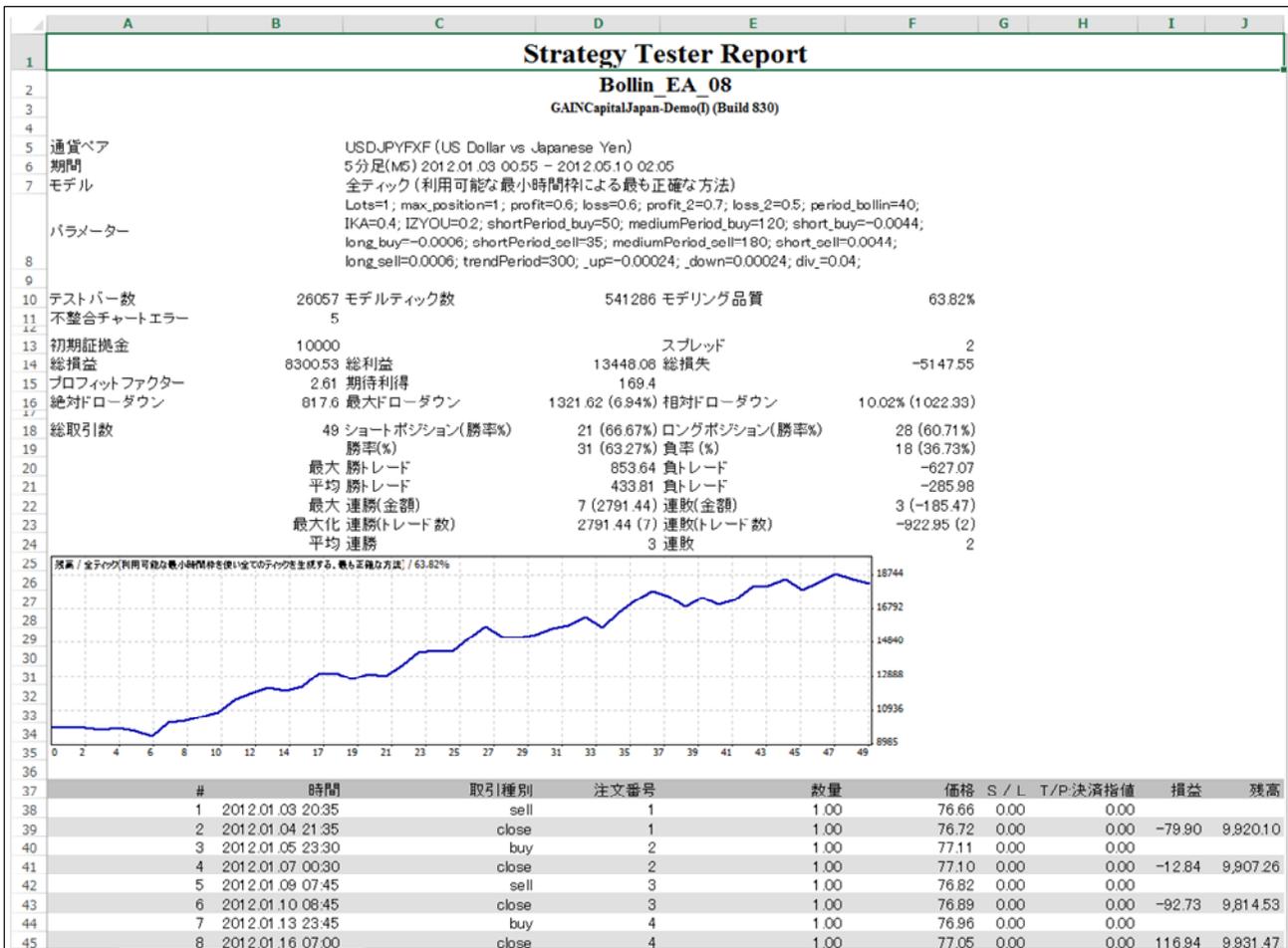
(1) エクセルに「Strategy Tester Report」のデータを取り込む

手順1；エクセルを立上げ、次に [ファイル] - [開く] で、

「StrategyTester.htm ファイル (Strategy Tester Report)」を指定したら、
[開く] を選択します



手順2；テストレポートがエクセルに取り込まれ、表示されます



※EAのトレード結果を詳細に検討する場合は、このデータが役に立ちます。

(実際の検討例は次節「(2)」を参照)

(2) ポイントだけ、拾い出してみます

・「Bollin_EA_08.mq4」を [2012_05_10] と [2015_05_07] のヒストリー・データでバックテストした結果を比較します。

ヒストリー・データ⇒	2012年	2015年
テストバー数	26057	66330
不整合チャートエラー	5	3
初期証拠金	10000	10000
総損益	8300.53	-4735.06
プロフィットファクター	2.61	0.84
絶対ドローダウン	817.6	4931.95
総取引数	49	146

※「不整合チャートエラー=5」が気になります。
 ・2012年5月時点でバックテストしたときは、「0」でした。
 当時の為替データを保存したのですが、切り出し時に一部データを取りこぼしたようです。（2015年05月データも同様）
 ・過去に保存したデータでバックテストすると、当時とは微妙に結果が異なることがあります。（原因は解析中）

ヒストリー・データ⇒	2012年	2015年	ヒストリー・データ⇒	2012年	2015年
総利益	13448.08	25269.96	総損失	-5147.55	-30005.02
期待利得	169.40	-32.43			
最大ドローダウン (6.94%)	1321.62	7744.75 (60.45%)	相対ドローダウン (1022.33)	10.02% (1022.33)	60.45% (7744.75)
ショートポジション (勝率%)	21 (66.67%)	81 (34.57%)	ロングポジション (勝率%)	28 (60.71%)	65 (53.85%)
勝率 (%)	31 (63.27%)	63 (43.15%)	負率 (%)	18 (36.73%)	83 (56.85%)

※本章では、特に筆者が重要と考える「期待値」を、「2012_05_10」の場合で計算してみます。

総取引数 49
 勝率 (%) 31 (63.27%) = 0.63
 負率 (%) 18 (36.73%) = 0.37

「Bollin_EA_08.mq4」では、USDJPYで「profit=0.6、profit_2=0.7、loss=0.6、loss_2=0.5」としたので、大雑把に「profit=0.65、loss=0.55」として計算すると、

$$\begin{aligned} \text{期待値} &= (\text{勝ったときの利益}) \times \text{勝率} + (\text{負けたときの損失}) \times \text{負率} \\ &= 0.65 \times 0.63 + (-0.55 \times 0.37) = 0.206 \end{aligned}$$

(期待値の詳細は第3章「3.4.2; 参考資料2」を参照ください)

◎大雑把に言うと通貨ペアはUSDJPYなので、このEAは

「1回のトレード(売買)で、平均0.206(約21 pips)の獲得を期待できる」と成ります。

(利益の総額ではありません、総額はその時点の売買枚数に依存します)

※「2015_05_07」の場合は「期待値」は計算するまでもなく、マイナスになります。

1. 3 「サンプルEA」の中身を覗く

※本節では、「Bollin_EA_08.mq4（サンプルEA）」の中身を覗き、次にEA玉成の肝となる「最適化」との関係を概説します。

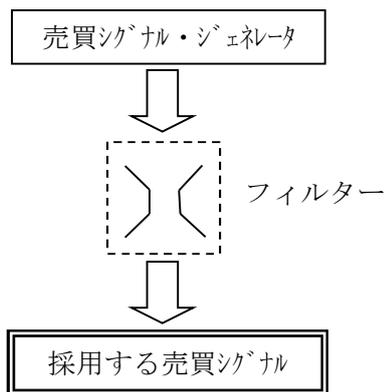
2012年5月10日のヒストリー・データ（[2012_05_10] データ）でバックテストをすると資産カーブが綺麗な「右上がり曲線」を描くように出来たのは、この「最適化」によるものです。

1. 3. 1 「サンプルEA」の基本仕様を確認する

・EAの売買ルールには色々ありますが、サンプルEAは下記の構成をしています。

<基本構造>

・まず、「サンプルEA」は、基本的に「トレンド・フォロー型」です



(1) [売買シグナル・ジェネレータ] 部分の概要

- ・多少、売買シグナルとしては信頼性が低くても、とにかく、一定の売買ルールに従ってシグナルを次々と発生する部分です。
- ・サンプルでは「Bollin Band（ボリンジャー・バンド）」の「 2σ 」バンドを「 $1/4(\text{Open}+\text{High}+\text{Low}+\text{Close})$ 」が
 - ①上抜けしたら → 買いシグナル
 - ②下抜けしたら → 売りシグナル
 と、しました。

(2) [フィルター] 部分の概要

- ・追加の売買ルールを適用することで、「売買シグナル」の信頼性を向上させる部分です。
- ・サンプルでは、筆者オリジナルの「フ型（松葉型）」フィルターを採用しています。ポイントはチャート上の「売買シグナル」を起点とした2本の「回帰直線の勾配」を比較して、売買シグナルの採用可否を判断していることです。（詳細は「第2章」と「第9章」を参照）

売買シグナル・ジェネレータ	フィルター（以下、目視によるイメージ）		採用可否
買いシグナル	「短・中」周期の回帰直線の勾配が共にプラス（右上り）で、 且つ（and） 絶対値（中周期の勾配） < 絶対値（短周期の勾配）	⇒	採用
	上記以外	⇒	廃棄
売りシグナル	「短・中」周期の回帰直線の勾配が共にマイナス（右下り）で、 且つ（and） 絶対値（中周期の勾配） < 絶対値（短周期の勾配）	⇒	採用
	上記以外	⇒	廃棄

※上表は、以下の様に言い直すことができます。

売買シグナルを起点とする「回帰直線の勾配」に着目し、目視したときに、 絶対値（中周期の勾配） < 絶対値（短周期の勾配） で、 且つ（and） 勾配が共にプラス（+）ならば ⇒ 買い 勾配が共にマイナス（-）ならば ⇒ 売り
--

(3) 「サンプルEA」コードの大きな構造を覗く

※Bollin_EA_08.mq4 のコードは、おおよそ下記の構造をしています。

```
int start()
{
// NewBar かチェック・・・「ティック (Tick) が Open 足か、それ以外の足かを判断」
if(IsNewBar() && (Bars>barsTotal))
{
// <Open 足であるときの処理>
① 「Open 値」を判断して売買に I N し、または手仕舞い O U T する。
}
else
{
// <Open 足では無いときの処理>
// [5 分足] 内で起こる急変に対応する処理を記載する
// バックテストでは確認が困難な部分
② 「擬似ティック値」を判断して、必要なら手仕舞い O U T のみを実行する。
}
return(0);
}
```

※上記の構造にした最大の理由は「シンプル イズ ベスト」です。(作り手側の要請)

すなわち、

- ① 「Open 値」で「マーケット (売買) I N」と「手仕舞い O U T」の判断・実行を行い、
- ② 「1 足内 (例えば 5 分足内)」でのティック (Tick) 値の急変に対しては、
「損切り・利確レベル」との比較で「手仕舞い」のみを行い、マーケット I N は行わない、
ような構造にすることで、EA の作りを単純化しました。

・実は、EA を上記のような構造にしたことで、「全ティック・モード」でバックテストするとき、その結果と「実際のトレード」結果の対応が、とても良くとれると考えています。

※詳細は「第 2 章」、「第 8 章 (コード)」と「第 9 章」を参照ください

1. 3. 2 Bollin_EA_08.mq4 の「extern 変数（最適化変数）」について

※本節では、「extern 変数」の機能に焦点を当てて考察します。

「Bollin_EA_08.mq4」コードから、「extern 変数」設定部分のみを抜粋；

```

// 損益設定
// 「①②」は最適化の実施後での設定値
extern double profit=0.60;
extern double loss=0.60;
extern double profit_2=0.70;
extern double loss_2=0.50;
// ①Bollinの幅
extern int period_bollin=40;
extern double IKA=0.4;
extern double IZYOU=0.2;
// ②フィルター設定
extern int shortPeriod_buy=50;
extern int mediumPeriod_buy=120;
extern double short_buy=-0.0044;
extern double long_buy=-0.0006;
//    ・ ・ sell 専用
extern int shortPeriod_sell=35;
extern int mediumPeriod_sell=180;
extern double short_sell=0.0044;
extern double long_sell=0.0006;
//
// ③トレンド判断
extern int trendPeriod=300;//70
extern double _up=-0.00024;//-0.0004
extern double _down=0.00024;
extern double div_=0.04;
//=====
    
```

この部分の「extern 変数」を最適化に利用しています。

※1；フィルターを構成する2本の回帰線の勾配
 ・ 上が「買い」用
 ・ 下が「売り」用

※2；この部分の「extern 変数」は使っていない。

※1；履歴データは Series 型として扱うので、データ上は買いの場合が「マイナス」符号で、売りの場合が「プラス」符号になります。（目視のイメージとは逆になります）

※2；実は、「Bollin_EA_08.mq4」では「③トレンド判断」用の extern 変数を設定していたのですが、使っていません。（将来の拡張用と理解してください）

(1) 一覧表（実施条件）でバックテスト条件を再度確認します

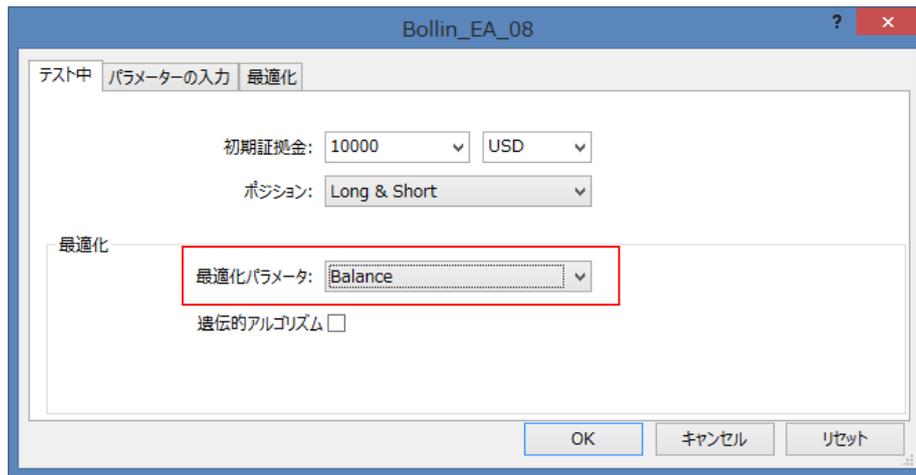
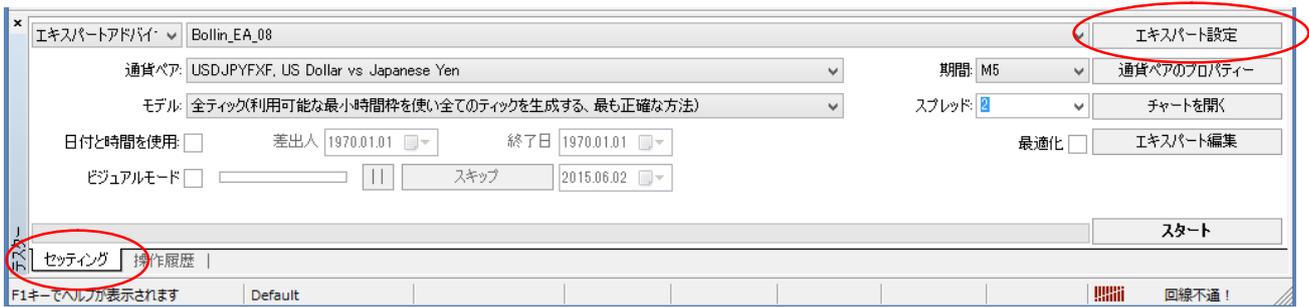
- ・履歴データとして [2012_05_10] データを用いている今回のバックテスト条件を再度整理しておきます。

項目	内容	備考
エキスパート・アドバイザー	Bollin_EA_08.mq4	
通貨ペア	USDJPY	本書では USDJPYFXF
期間	M5 (5分足)	
モデル	全ティック (Every tick)	
スプレッド	2	
履歴データ	※ [2012_05_10] フォルダ中のデータ	M1 と M5 データを使う

(「(2)」以降の解説の前提)

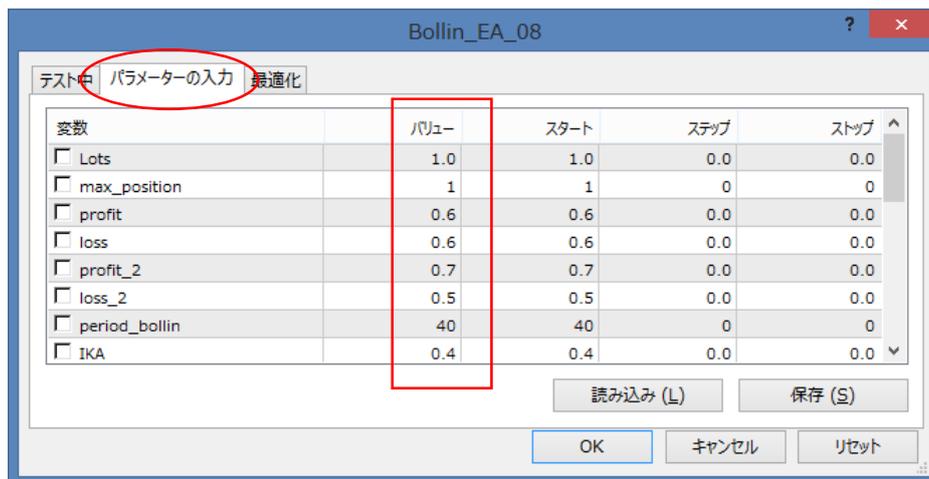
(2) Bollin_EA_08.mq4 のコード中にある「extern 変数」の働きを確認する

ストラテジー・テスターの [セッティング] タブで [エキスパート設定] を選択



※ [最適化パラメータ] は「Balance」である場合で解説します
 (実は [遺伝的アルゴリズム] を使わない場合は、特に何を指定しても結果は同じです)

表示されるウインドウの [パラメータの入力] タブを観てみます。



※この表示の中で、[バリュー] はコードの中で「extern 変数」として設定した値です。

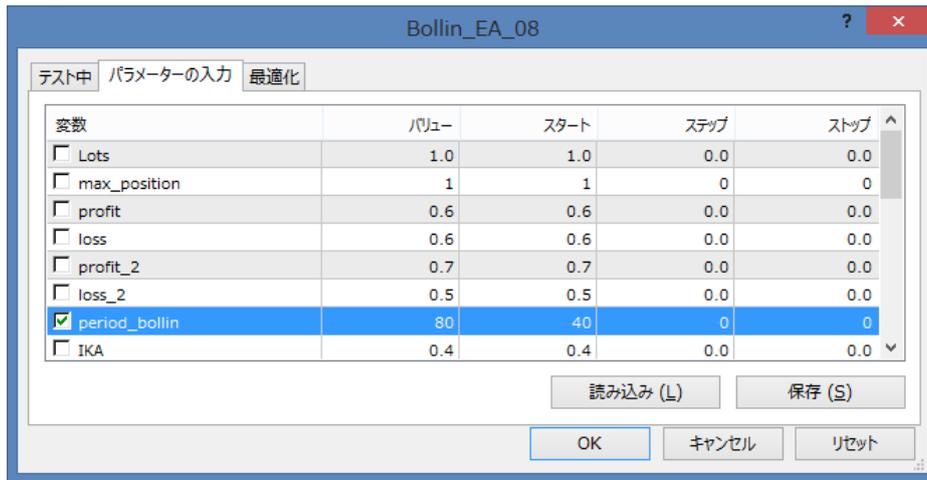
(例) コード中 ; extern int period_bollin=40;

バリュー ; 40

コード中の「extern 変数」は全て [パラメータの入力] タブの [変数] として表示されます。

この [バリュー] 値は、選択・変更し、その値でバックテストすることが可能です。

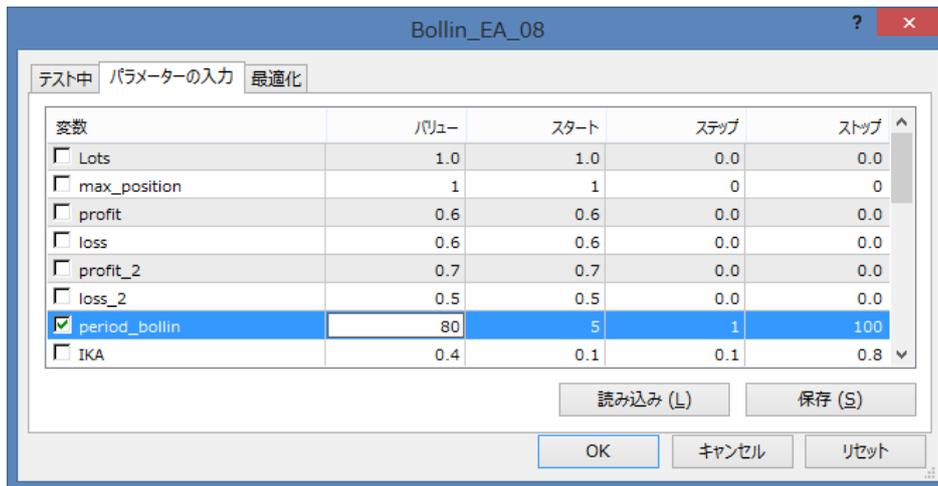
(例) period_bollin を「40」→「80」へ変更



※それでは、[スタート、ステップ、ストップ] は、何に使えるのでしょうか？

⇒ これは「変数」を順次変えてのバックテストに使えるのです。（これが最適化です）

(例) [スタート=5、ステップ=1、ストップ=100]



上記の状態、以下の様にストラテジー・テスターの [セッティング] タブ上の [最適化] にチェックを入れると、



ストラテジー・テスターは、profit 変数を「5」から初めて、「01」おきに「5→6→7→・・・→99→100」のように「100」になるまで設定を変えながら、それぞれの場合でバックテストを連続して行います。

※そして、その結果を整理したデータとして表示します。

[最適化結果] タブでは、デフォルトでは下記の様に [損益] が降順で表示され、最大損益（収益）パスが一番上に表示されます。

⇒ これを「全探索」による「最適化」と呼んでいます、
言い換えると、変数の組合せに対して連続でバックテストした結果の表示順を変えているだけとも言えます。
([遺伝的アルゴリズム] を使う場合は全く異なります)

×	パス	損益	総取引数	プロフィット
	◎ 63	8458.73	53	
	◎ 38	8423.33	51	
	◎ 37	8423.11	51	
	◎ 62	8384.67	53	
	◎ 36	8300.53	49	
	◎ 61	8239.90	53	
	◎ 60	8239.90	53	
	◎ 59	8237.17	52	

※通常、「損益」を降順で表示する

(しかし、連続してバックテストを実行すること自体が凄い機能だと思います)

※ところで、[最適化パラメータ] で「Balance」を指定する意味ですが、
[最適化グラフ] で縦軸の表示を Balance（資産残高）に設定することになります。
（「1. 3. 3 (2)」を参照）

ただし [遺伝的アルゴリズム] を使う場合は、単に「縦軸に何を表示するか」だけではなく、何をベースにして「間引き探索」をするのかも指定することになるので注意が必要です。
（通常は [最適化パラメータ : Balance] で使う事が殆どだと思います)

(3) Bollin_EA_08.mq4 の「extern 変数」の最適値を見つけるには（まとめ）

まとめ 1 ;

読者は前節「(2)」から「資産残高 ; Balance」を最大化する「extern 変数値 ; 最適化変数値」を見つけ出すには、各「extern 変数」の全てで「値」を色々組み合わせてバックテストすれば、すなわち「最適化」すれば良い、と言う事が容易に推測出来たと思います。

・それではここで、Bollin_EA_08.mq4 中で使用している「最適化変数 ; extern 変数」が、「資産残高 ; Balance」を最大化するために実施した最適化で、どの様に変化したかのみを確認しておきます。(本節では最適化の手順については触れません)

- ① 「開発時当初の値 (データと目視確認で決めた) ; 最適化前」 と
- ② 「右肩上がりの収益を生み出す値 (現状設定値) ; 最適化後」 の

2つの組合せを記載します。

<最適化変数 (extern 変数) と、その最適化前後での値の変化>

大項目		extern 変数 (パラメータ)	Bollin_EA_08.mq4	
			①開発時当初の値 (最適化前)	②右肩上がり収益用 (最適化後)
損益パラメータ		Open 時 profit	0.20	0.6
		Open 時 loss	0.20	0.6
		足内 profit_2	0.20	0.7
		足内 loss_2	0.20	0.5
Bollin パラメータ		周期 period_bollin	40	40
		幅制限上限 IKA	0.4	0.4
		幅制限下限 IZYU	0.3	0.2
Filter パラメータ	買い パラメータ	短周期 shortPeriod_buy	20	50
		中周期 mediumPeriod_buy	100	120
		UP 短勾配 short_buy	- 0.0008	-0.0044
		UP 長勾配 long_buy	- 0.0004	-0.0006
	売り パラメータ	短周期 shortPeriod_sell	20	35
		中周期 mediumPeriod_sell	100	180
		DOWN 短勾配 short_sell	+ 0.0008	+0.0044
		DOWN 長勾配 long_sell	+ 0.0004	+0.0006

(注) 一部 2012 年当時の変数名を使っているため、「long」は本来「medium」とすべきですが、そのまま変更せずに使っています、「長勾配 ; long ⇒ 中周期勾配 ; medium」と読み替えてください。

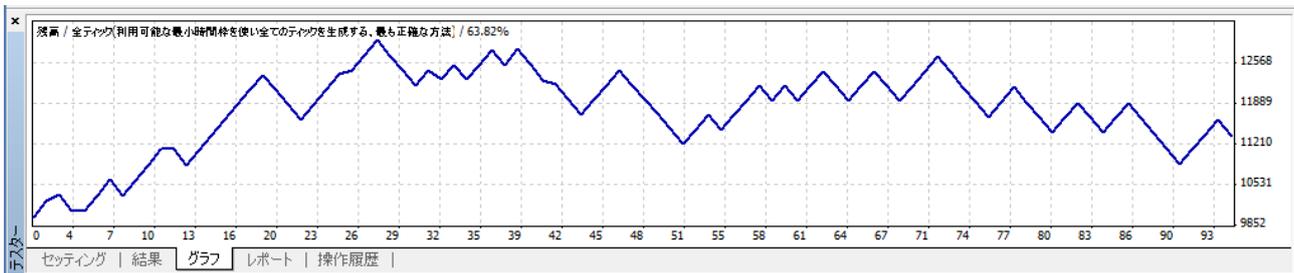
※開発時当初の値 (最適化前) ;

「第 2 章」で詳述するツールを使った方法等で、一応「適正值」ではないかと、ある程度自信を持って設定した値だったのですが、実際に最適化を実施してみると、大きく値が変わってしまいました。(これが、最適化を行う大きな理由です。)

※最適化前後での「資産カーブ；[グラフ] タブ」を比較する

下記は両方共に [2012.05.10] ヒストリー・データに対してバックテストした結果です

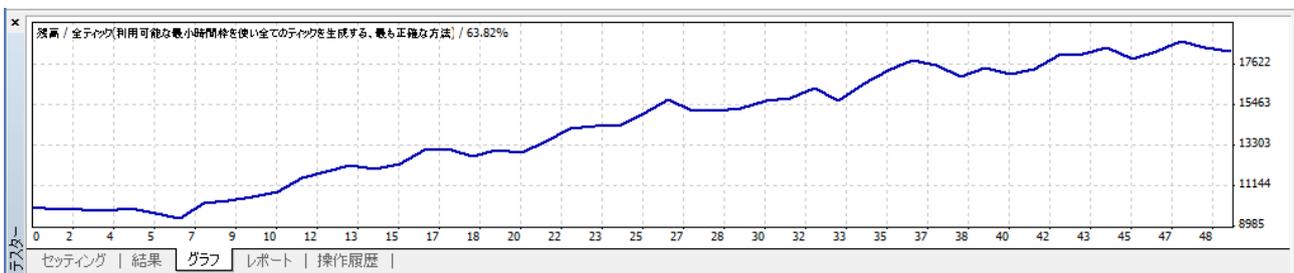
<extern 変数=① 開発当初の値（最適化前）>



・一応「適正值」ではないか、と設定した値ですので、そこそこの結果と思います。

最適化 ▼

<extern 変数=② 最適化後の値>



・最適化前とは比較にならないくらい改善されています！

まとめ2；

- 上記の例で示した様に、MT4には非常に強力な「最適化」機能が備わっています、正直、この最適化機能なしでは独自にEAを開発することは略不可能であると考えます。
- ただし、導き出された最適「extern 変数値」は、テストするチャート・データ、すなわちヒストリー・データに大きく依存することに注意が必要です。

※MT4の「最適化」機能とは！

誤解を恐れずに言えば、条件を変えてバックテストを繰り返し行い、その結果を判り易い形式で表現してくれる機能のことです。

補足；

それでは、「最適化」さえすれば、どのようなEAでも利益が出るようになるのでしょうか。結論から言うと、次の通りです。

- 「最適化しても利益が出ないEA」は存在するか？
⇒確実にあります。(何度も経験済み！)

一方で、

- 「最適化しなくても利益が出るようなEA」は存在するか？
⇒あります。(ごく稀に経験)

1. 3. 3 最適化とは何か？を再考する

- ・前節「1. 3. 2」では、「最適化」の大雑把な概念と、その効果のみを解説しましたが、イマイチ釈然としない部分があったのではと思います。そこで、イメージを掴むため本節ではもう少し具体的に「最適化」機能と手順について解説することにします。
- ・そもそも最適化機能とはどう言う内容かを、実際に単純化した例で動作させながら解説いたします。

(1) 準備； 最適化機能を動作させるための設定概要

※例示する最適化の対象と条件は、以下の内容とします

対象EA	Bollin_EA_08.mq4
通貨ペア (履歴・データ)	USDJPY 「2012_05_10」ホルダー中のM5とM1データを使用
モデル：期間：スプレッド	全ティック：M5：2
最適化モード	「遺伝的アルゴリズム」は使用しない
MT4バージョン	Build610以降(本章ではBuild830を使用)

「2012_05_10」フォルダ中のデータを履歴・センターに設定済とします

※MT4はオフラインに設定済みとします

「回線不通！」か「無効な口座」状態（オフライン）にします。

(1. 1. 3 (4) を参照)

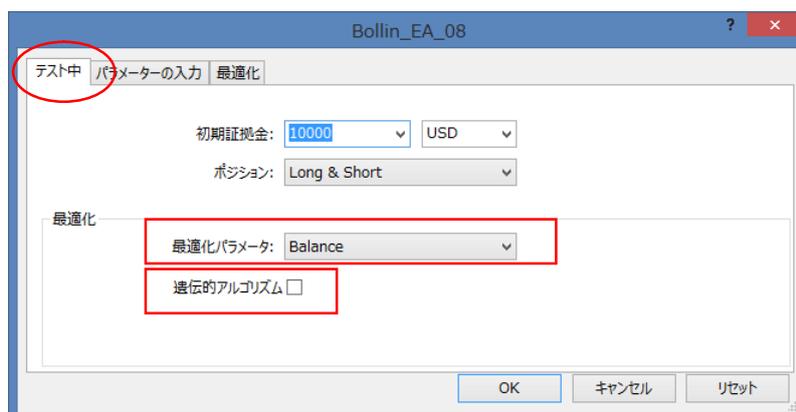


(2) 1次元の最適化を考えてみる

- ・1次元と表現したのは、最適化変数（extern 変数）の内「1つ」だけ変化させて、最適化を行う、という意味です。

(注) ここでは「遺伝的アルゴリズム」は使いません

手順1；ストラテジーテスター（Strategy Tester）を表示させてから、
[エキスパート設定] を選択、[テスト中] タブを表示させます、
[遺伝的アルゴリズム] は使わないので、チェックはしないでください。



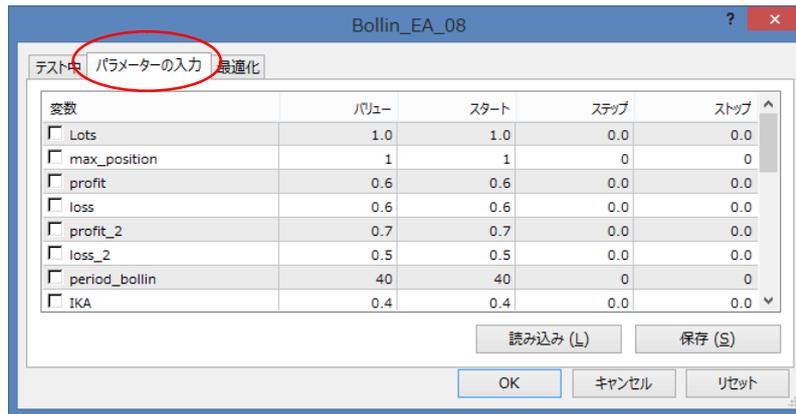
[最適化パラメータ] は「Balance」にしておきます

⇒ ・[最適化グラフ]；「資産残高（Balance）」がY軸（縦軸）として表示

※ [最適化パラメータ] は連続して実施したバックテスト結果（最適化）のまとめ方（何に焦点を当てて表示するか）を決めます。

但し、[最適化結果] の [損益] はデフォルトは常に降順（大きい値順）で表示。

手順2；[パラメータの入力（extern 変数のことです）] タブを表示させます

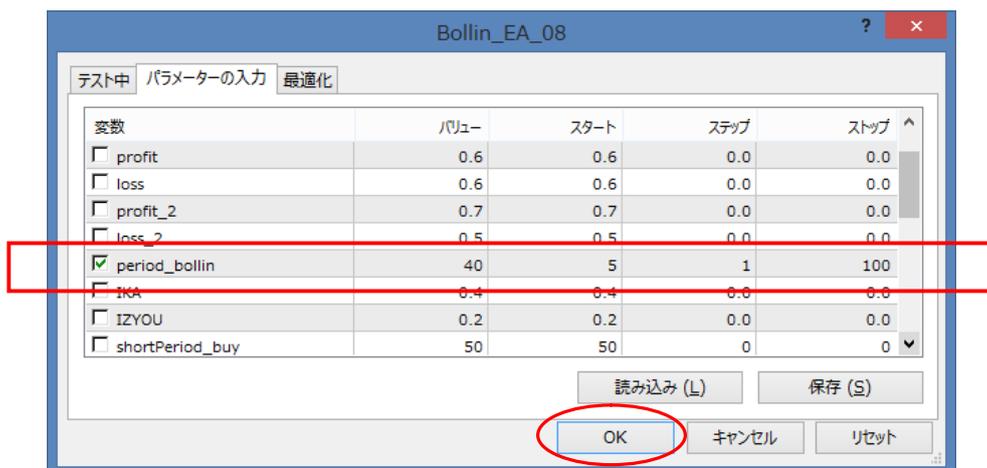


手順3；[パラメータの入力] タブで、変数を1つだけ設定する

※ここでは Bollin バンドを決めるパラメータの「周期;period_bollin」を取り上げます。

本節では、探索範囲の設定 [スタート]・[ステップ]・[ストップ] に、「5」・「1」・「100」と入力します。

◎変数 [period_bollin] にチェックを入れ、スタート「5」、ステップ「1」、ストップ「100」に設定したら、[[OK] を選択。

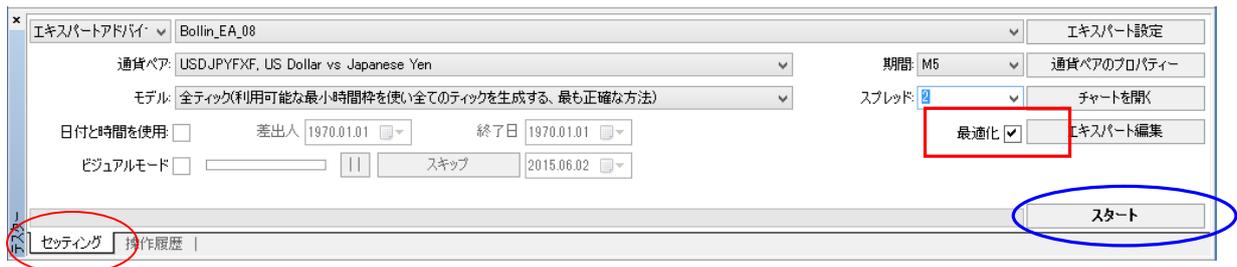


period_bollin は「5」から「100」までの、「96」通り（ステップ）の値を試すこととなります。（96回バックテストを繰り返します）

上記処理の意味；

- ・ period_bollin 変数を「5」から初めて「1」ステップずつ増加していき、各 period_bollin 値でバックテストを行い、資産残高（Balance）値を計算し、period_bollin 値が「100」になったら止める、という意味です。
- そして、その結果の資産残高（Balance）の変化が [最適化グラフ] として表示されます。

手順4；ストラテジーテスター（Strategy Tester）の「[セッティング]」タブで「最適化」にチェック後、「スタート」を選択



手順5；最適化が始まり、同時に「[スタート]」が「[ストップ]」に変わります



※左端「3/96」；

下（96）が実行する extern 変数の総ステップ数（変化させる数）を示し、
上（3）が現在までに実行が完了したステップ数です。

右端「0:00:14/0:07:14」；

下（0:07:14）が実行完了までの予測時間 [時分秒]、
上（0:00:14）が現在までに実行済みの経過時間を示します。

・「96 通り」の組合せを実行するのに「7分以上かかる」と言うわけです。

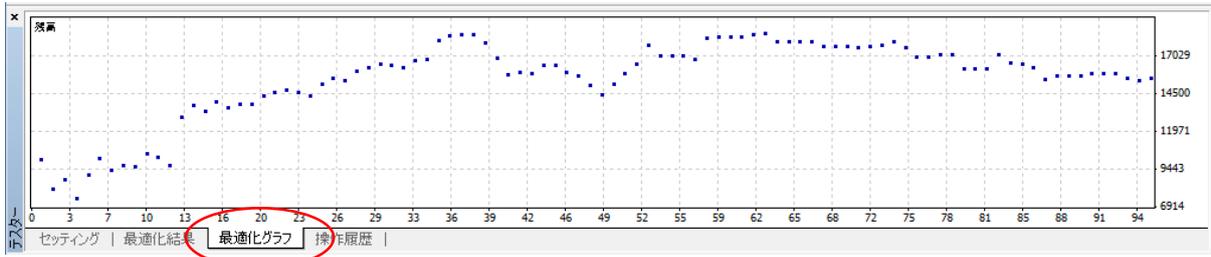
最適化に要する所要時間は使用するパソコン性能と MT4 のバージョンに大きく依存します。
終了するまで、しばらく休憩を取りましょう



※本例では最適化の探索が「5分32秒」で終了、「ストップ」が「スタート」に変わりました。

手順5；最適化結果を確認します（使用する環境で結果が微妙に変わる場合があります）

① [最適化グラフ] タブを観る； X軸は「パス No」、Y軸が資産残高「Balance」



② [最適化結果] タブを観る

パス	損益	総取引数	プロフィットファクター	期待利得	ドローダウン \$	ドローダウン %	パラメーターの入力
63	8458.73	53	2.37	159.60	2274.81	12.36	period_bollin=67; Lots=1; max_position=1; profit=...
38	8423.33	51	2.45	165.16	1321.62	8.60	period_bollin=42; Lots=1; max_position=1; profit=...
37	8423.11	51	2.45	165.16	1321.62	8.60	period_bollin=41; Lots=1; max_position=1; profit=...
62	8384.67	53	2.36	158.20	2274.81	12.36	period_bollin=66; Lots=1; max_position=1; profit=...
36	8300.53	49	2.61	169.40	1321.62	10.02	period_bollin=40; Lots=1; max_position=1; profit=...
61	8239.90	53	2.31	155.47	2274.81	12.46	period_bollin=65; Lots=1; max_position=1; profit=...
60	8239.90	53	2.31	155.47	2274.81	12.46	period_bollin=64; Lots=1; max_position=1; profit=...
59	8237.17	52	2.29	158.41	2274.81	12.53	period_bollin=63; Lots=1; max_position=1; profit=...

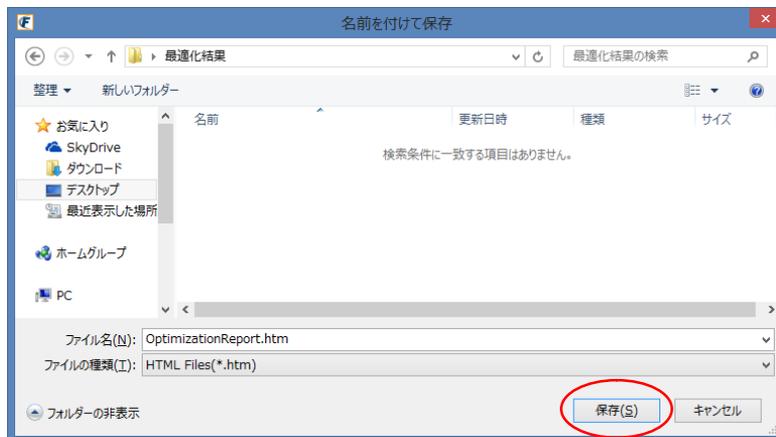
③最適化レポートを観る

※最適化結果の上で、[右クリック] - [レポートの保存] を選択



○保存先「フォルダー」等を選択する「名前を付けて保存」ウィンドウが表示されます、保存先ホルダーやファイル名は任意に設定可能です。

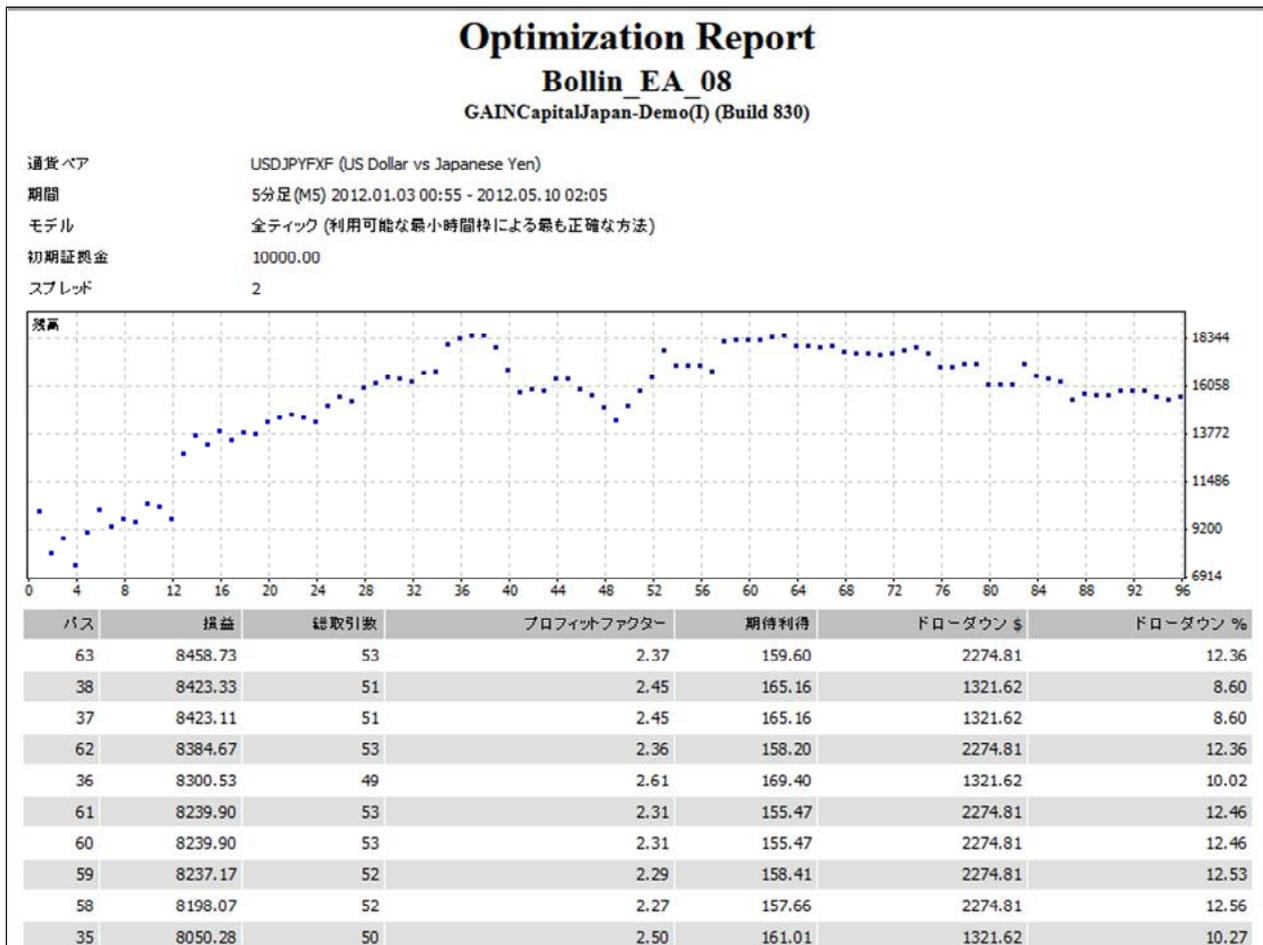
デフォルト； [ファイル名] StrategyTester.htm 、[ファイルの種類] HTML



※本書ではデフォルト設定のまま、[保存] を選択



テスト・レポートが保存と同時に表示されます



手順6 ; 詳細データを確認する・・・再度、[最適化結果] タブを表示します

パス	損益	総取引数	プロフィットファクター	期待利得	ドロウダウン \$	ドロウダウン %	パラメータの入力
63	8458.73	53	2.37	159.60	2274.81	12.36	period_bollin=67; Lots=1; max_position=1; profit=...
38	8423.33	51	2.45	165.16	1321.62	8.60	period_bollin=42; Lots=1; max_position=1; profit=...
37	8423.11	51	2.45	165.16	1321.62	8.60	period_bollin=41; Lots=1; max_position=1; profit=...
62	8384.67	53	2.36	158.20	2274.81	12.36	period_bollin=66; Lots=1; max_position=1; profit=...
36	8300.53	49	2.61	169.40	1321.62	10.02	period_bollin=40; Lots=1; max_position=1; profit=...
61	8239.90	53	2.31	155.47	2274.81	12.46	period_bollin=65; Lots=1; max_position=1; profit=...
60	8239.90	53	2.31	155.47	2274.81	12.46	period_bollin=64; Lots=1; max_position=1; profit=...
59	8237.17	52	2.29	158.41	2274.81	12.53	period_bollin=63; Lots=1; max_position=1; profit=...

※デフォルト設定では [損益] が大きい順に並んでいます。

([パス] [損益] などのタブをクリックすると、昇順・降順に並び替えることができます)
 複数行を選択して、[コピー] とすると、指定部分のデータをコピーできます。

・上から3番目までをコピーしてみます ; [最適化結果] の上で [右クリック] - [コピー] 選択

パス	損益	総取引数	プロフィットファクター	ドロウダウン %	パラメータの入力
63	8458.73	53	2.37	12.36	period_bollin=67; Lots=1; max_position=1; profit=...
38	8423.33	51	2.45	8.60	period_bollin=42; Lots=1; max_position=1; profit=...
37	8423.11	51	2.45	8.60	period_bollin=41; Lots=1; max_position=1; profit=...
62	8384.67	53	2.36	12.36	period_bollin=66; Lots=1; max_position=1; profit=...
36	8300.53	49	2.61	10.02	period_bollin=40; Lots=1; max_position=1; profit=...
61	8239.90	53	2.31	12.46	period_bollin=65; Lots=1; max_position=1; profit=...
60	8239.90	53	2.31	12.46	period_bollin=64; Lots=1; max_position=1; profit=...
59	8237.17	52	2.29	12.53	period_bollin=63; Lots=1; max_position=1; profit=...

パラメータの設定 (I)
 コピー (C) Ctrl+C
 全てコピー (y) Alt+A
 レポートの保存 (S)
 マイナスの結果を表示しない (U)
 パラメータの表示 (P)
 自動整理 (A) A
 グリッド (G) G

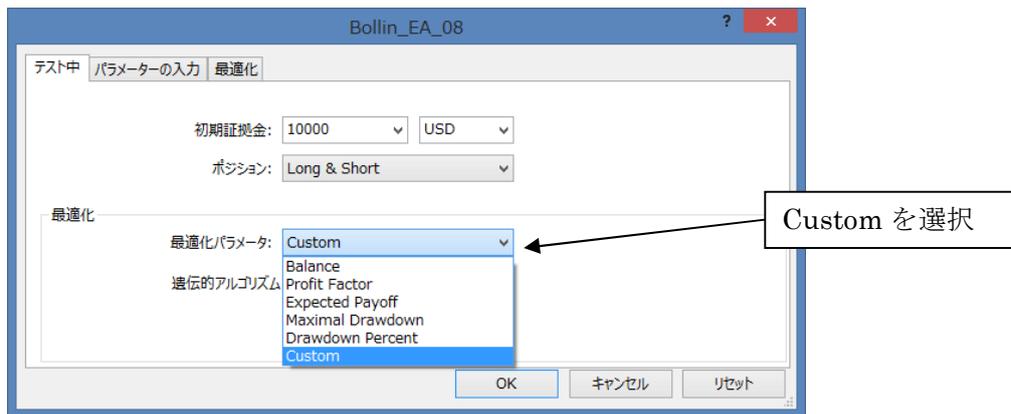
上から3番目までのコピー結果；

63	8458.73 53	2.37	159.60	2274.81	12.36%	0.0000000	period_bollin=67
							Lots=1 max_position=1 profit=0.6 loss=0.6 profit_2=0.7 loss_2=0.5
							IKA=0.4 IZYOU=0.2 shortPeriod_buy=50 mediumPeriod_buy=120
							short_buy=-0.0044 long_buy=-0.0006 shortPeriod_sell=35
							mediumPeriod_sell=180 short_sell=0.0044 long_sell=0.0006
							trendPeriod=300 _up=-0.00024 _down=0.00024 div_=0.04
38	8423.33 51	2.45	165.16	1321.62	8.60%	0.0000000	period_bollin=42
							Lots=1 max_position=1 profit=0.6 loss=0.6 profit_2=0.7 loss_2=0.5
							IKA=0.4 IZYOU=0.2 shortPeriod_buy=50 mediumPeriod_buy=120
							short_buy=-0.0044 long_buy=-0.0006 shortPeriod_sell=35
							mediumPeriod_sell=180 short_sell=0.0044 long_sell=0.0006
							trendPeriod=300 _up=-0.00024 _down=0.00024 div_=0.04
37	8423.11 51	2.45	165.16	1321.62	8.60%	0.0000000	period_bollin=41
							Lots=1 max_position=1 profit=0.6 loss=0.6 profit_2=0.7 loss_2=0.5
							IKA=0.4 IZYOU=0.2 shortPeriod_buy=50 mediumPeriod_buy=120
							short_buy=-0.0044 long_buy=-0.0006 shortPeriod_sell=35
							mediumPeriod_sell=180 short_sell=0.0044 long_sell=0.0006
							trendPeriod=300 _up=-0.00024 _down=0.00024 div_=0.04

※Build 600代以降のバージョンでは、「ドロダウン%」と「パラメータの入力データ」の間に [Cutom] 用の欄が確保されており、「Custum」を使用していないときに [最適化結果] のデータをコピーすると「0.0000000」が挿入されるようになりました。

(参考) 「Custom」の設定は例えば「OnTester0」を使うときに必要となります。

mql4 コード内で、OnTester0を使うとき



最適化結果

パス /	損益	総取引数	プロフィットファクター	期待利得	ドロダウン \$	ドロダウン %	OnTester 結果	パラメータの入力
①	0.00	0	0.00	0.00	0.00	0.00	-0.75098725	x=-3; y=-3;
②	0.00	0	0.00	0.00	0.00	0.00	-0.91155737	x=-2.95; y=-3;
③	0.00	0	0.00	0.00	0.00	0.00	-0.99140033	x=-2.9; y=-3;
④	0.00	0	0.00	0.00	0.00	0.00	-0.98781629	x=-2.85; y=-3;
⑤	0.00	0	0.00	0.00	0.00	0.00	-0.90527930	x=-2.8; y=-3;
⑥	0.00	0	0.00	0.00	0.00	0.00	-0.75426683	x=-2.75; y=-3;
⑦	0.00	0	0.00	0.00	0.00	0.00	-0.54972645	x=-2.7; y=-3;
⑧	0.00	0	0.00	0.00	0.00	0.00	-0.30937596	x=-2.65; y=-3;

[最適化結果] タブに「OnTester 結果」欄が挿入されて、その値も表示されます。

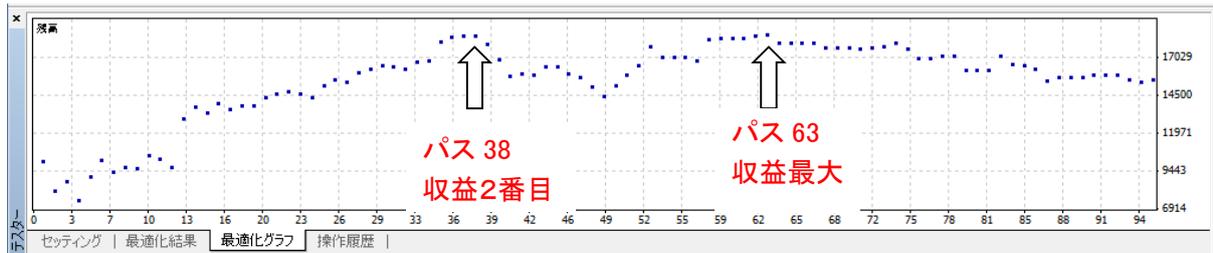
ところが、[Cutom] として何も設定していない場合；

[最適化結果] タブ上には表示されません。

パス	損益	総取引数	プロフィットファクター	期待利得	ドローダウン \$	ドローダウン %	パラメーターの入力
63	8458.73	53	2.37	159.60	2274.81	12.36	period_bollin=67; Lots=1; max_position=1; profit=...
38	8423.33	51	2.45	165.16	1321.62	8.60	period_bollin=42; Lots=1; max_position=1; profit=...
37	8423.11	51	2.45	165.16	1321.62	8.60	period_bollin=41; Lots=1; max_position=1; profit=...

一方、データとして「コピー」した内容には、「0.00000000」として表示されます。

手順7；改めて [最適化グラフ] タブで、最適化の意味を考察します

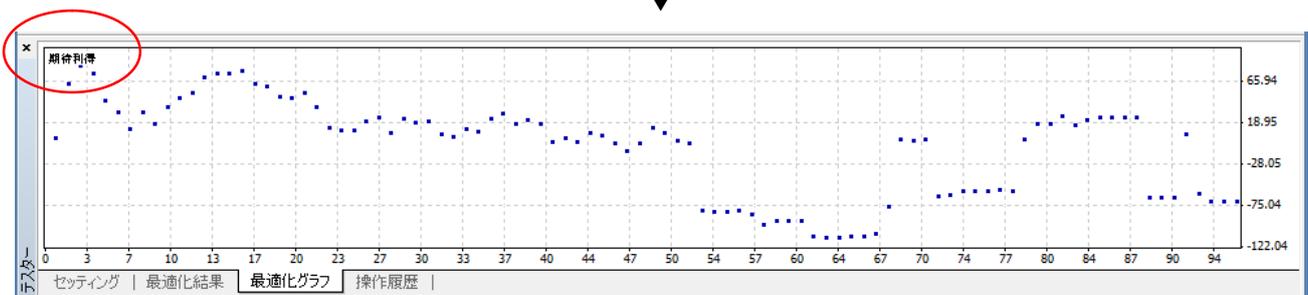
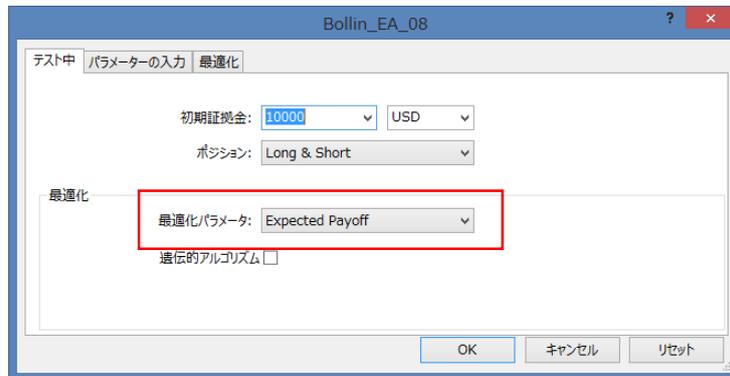


※パス=63 近辺が山の頂上になっているのが判ります、パス=38 近辺にも山があります。

ポイント；

- ・最適化とは、例えば他の条件は一定として「period_bollin のみ」をパラメータとして、変化させたときに、どこに資産残高「Balance」のピーク値があるかを探し出すものです。
- ・EA 設計の立場から言うと、「最適化変数値（exter 値）」が多少変化しても残高（Blance）の値の変化が少ない「安定点（山の傾斜が緩やかな点）」を採用することが多いです、必ずしも「収益が最大」になるピーク点を選択する訳ではありません。

（参考）[最適化パラメータ] を「Expected Payoff」とした場合で、同じ1次元の最適化を実施



[最適化グラフ] タブのY軸（縦軸）が、[期待利得（Expected Payoff）] に変わっています。

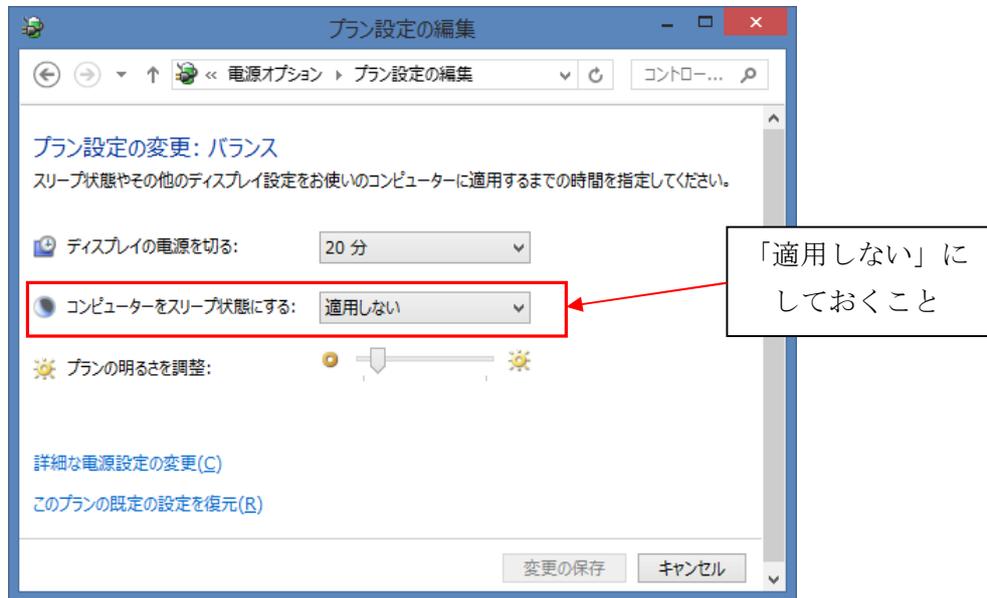
・但し、[最適化結果] の内容（データ）は「Balance」のときと同じであり、変わりません。

(3) 次に、2次元の最適化を考えてみます

◎時間のかかる最適化を実行する前に、ご注意！；

2次元の最適化では時間がかかります、

コンピュータの「電源オプション」で、「スリープ状態」にならないように設定してください。



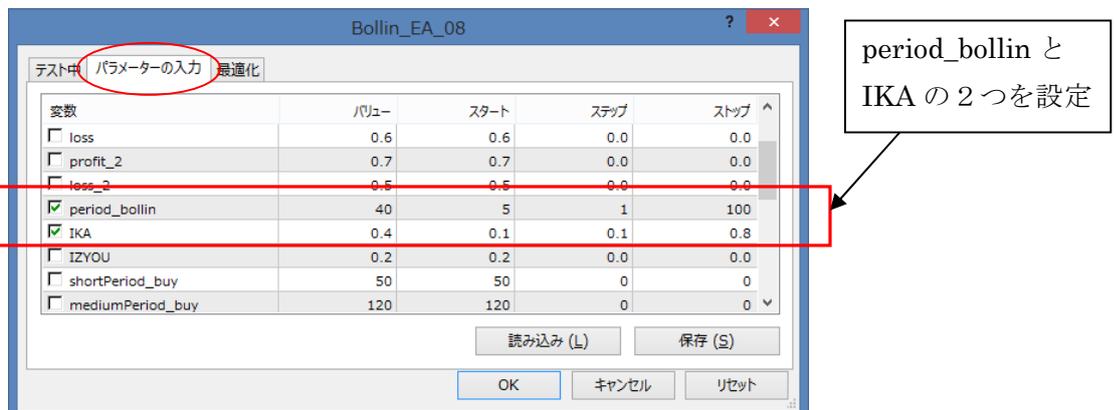
※筆者は何回か「スリープ」設定の解除を忘れたまま、長時間かかる最適化をスタートして、長時間席を離れていたらPCがスリープ状態になって「最適化が止まっていた」と言う苦い経験をしました。

(夜に最適化をスタートして、翌朝見たら止まっていた！などと言う経験が何度かありました)

2次元の最適化とは；

- 2次元の最適化とは最適化変数 (extern 変数) を「2つ」使って最適化を行う、と言う意味です、一つ一つの最適化変数 (extern 変数) を設定する方法は、1次元の場合と同様です。

手順1；「period_bollin」に続いて、2つ目の最適化変数として「IKA」を設定してみます。

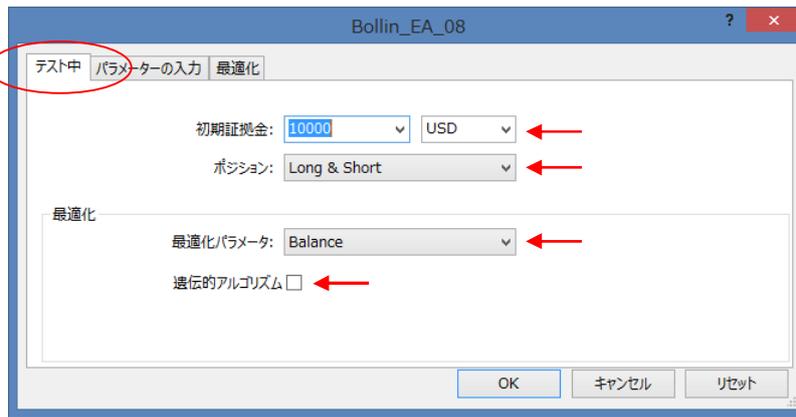


※period_bollin；「5～100」間を「1」ステップずつ実行するので、ステップ数は「96」

IKA；「0.1～0.8」間を「0.1」ステップずつ実行するので、ステップ数は「8」

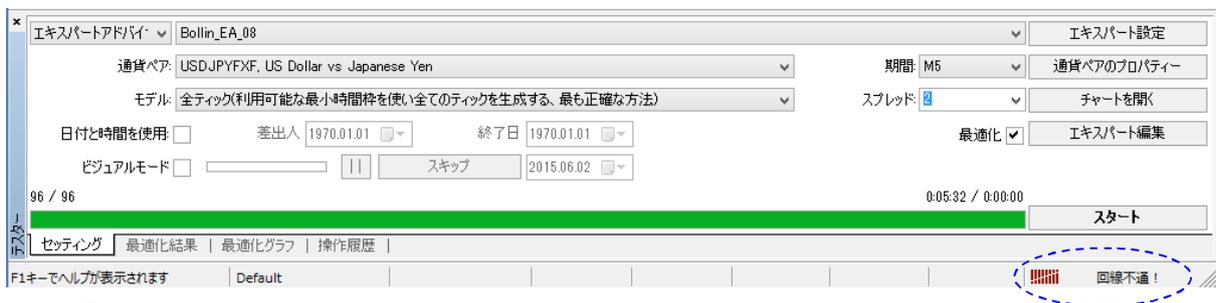
従って2次元での総ステップ数は「96×8=768」となります。

(なんと、最適化に要する時間は「1次元（前節）」の場合の8倍かかります！)



[OK] を選択

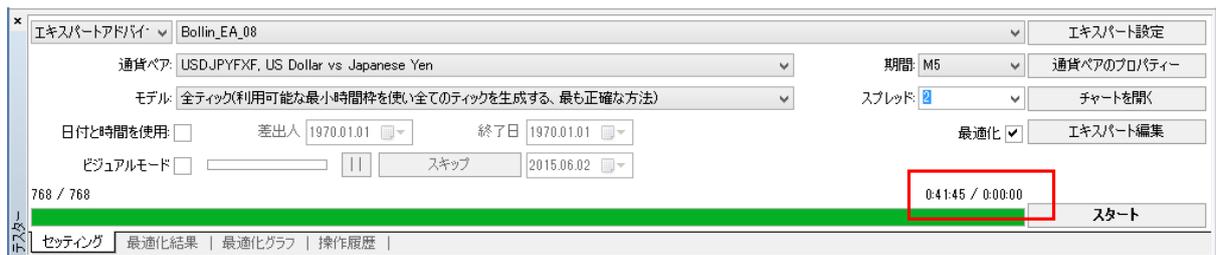
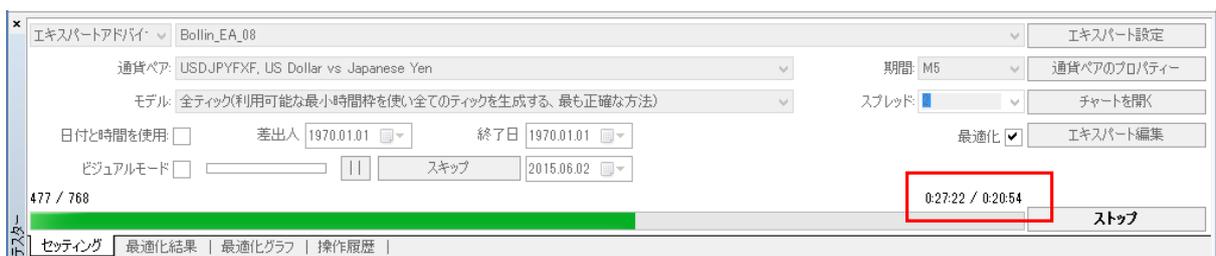
手順2 ; 他は「1次元」の時と同様に設定し、[スタート] を選択（1次元最適化から継続）



最適化がスタート

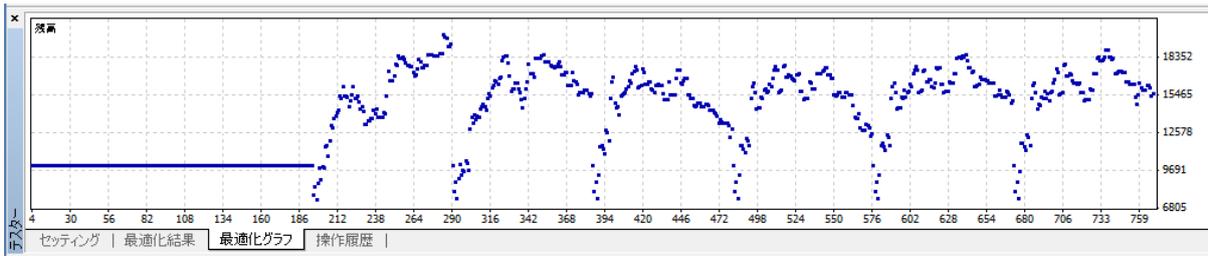


※あと、1時間ぐらひはかかる！ ・しばらく、パソコンの前から離れましょ！



※どうやら、約40分ほどかかったようです。（使用環境によって大きく変わります）

手順3 ; [最適化グラフ] を観ます ・不思議な波形です！（初期のパスでは売買不成立の様）

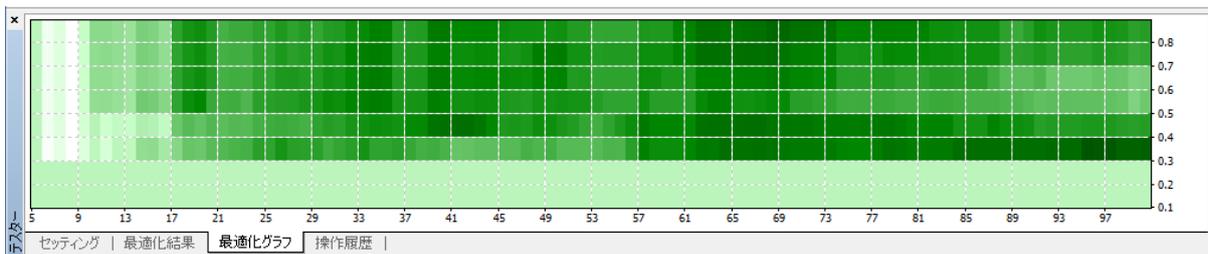
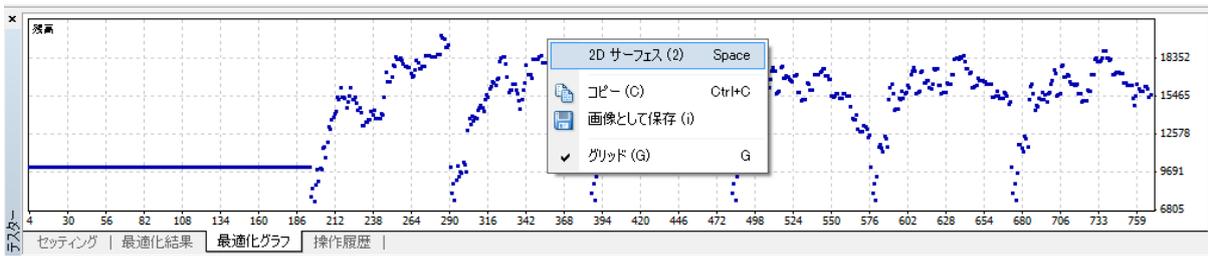


※X軸は「パス No」、Y軸が資産残高（Balance）です。

[エキスパート] - [テスト中] タブの、[最適化パラメータ：Balance] としているので、Y軸がBalanceになります。（使用する環境と設定内容で結果が若干変わる場合があります）

(注) このままでは解読するのが大変なので、2次元表示にしてみます。

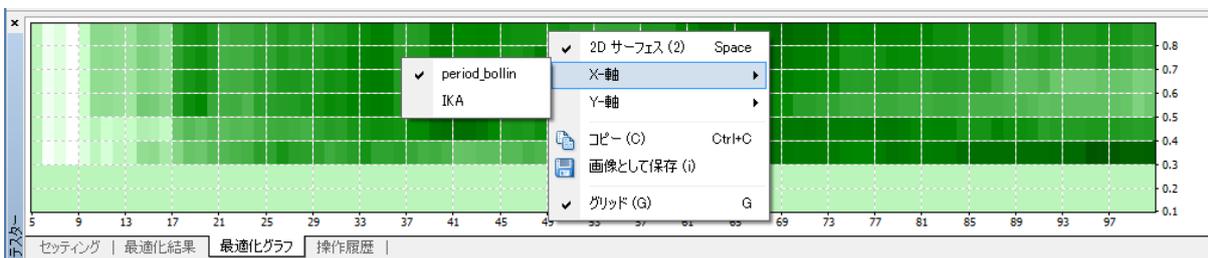
・グラフ上で、[右クリック] - 「2Dサーフェス」を選択、



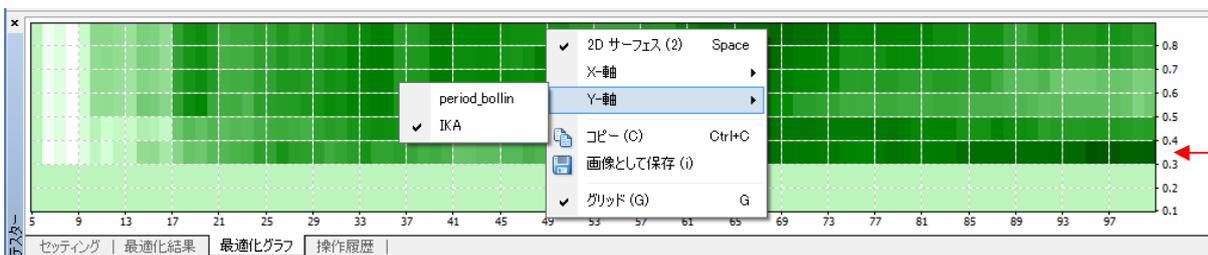
※緑色の濃い部分が、資産残高（Balance）の多いことを示しています。

[最適化パラメータ：Balance] と設定しているののでZ軸方向（緑色）がBalanceです。

・X軸とY軸を確認するために、[最適化グラフ] 上で [右クリック] - [X-軸] を選択

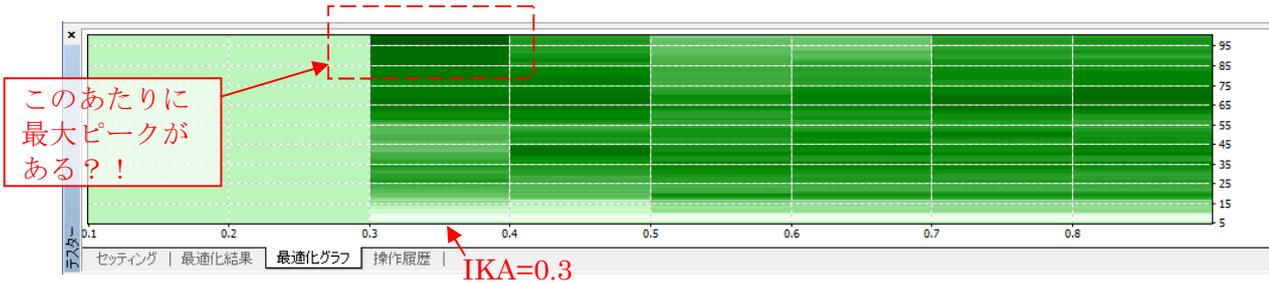


・さらにグラフ上で、[右クリック] - [Y-軸] で観ると、以下の様に確認できます



※X 軸と Y 軸の指定を変更するには、上記の手順で「X 軸」か「Y 軸」の片方のみの指定を変えれば実行できます、以下に結果だけ表示しておきます。

以下は、「X 軸」を「IKA」に、「Y 軸」を「period_bollin」に変更した場合のグラフです。



※今回の最適化では、最適化変数の最適組合せが探索した範囲の端っこの方に来ました。

手順 4 ; [最適化結果] を観ます

デフォルトでは、上から [収益] が大きいものから順番に並びます。(降順)

パス	損益	総取引数	プロフィットファクター	期待利得	ドローダウン \$	ドローダウン %	パラメーターの入力
283	9983.23	33	6.38	302.52	1299.25	8.34	period_bollin=95; IKA=0.3; Lots=1; max_position=...
284	9796.65	33	6.28	296.87	1299.25	8.34	period_bollin=96; IKA=0.3; Lots=1; max_position=...
285	9749.43	33	6.12	295.44	1299.25	8.34	period_bollin=97; IKA=0.3; Lots=1; max_position=...
288	9304.18	32	6.16	290.76	1298.69	8.34	period_bollin=100; IKA=0.3; Lots=1; max_position=...
287	9141.03	32	5.65	285.66	1298.69	8.34	period_bollin=99; IKA=0.3; Lots=1; max_position=...
286	9141.03	32	5.65	285.66	1298.69	8.34	period_bollin=98; IKA=0.3; Lots=1; max_position=...
736	8823.71	62	2.14	142.32	2423.15	13.20	period_bollin=68; IKA=0.8; Lots=1; max_position=...
737	8823.65	62	2.14	142.32	2423.15	13.20	period_bollin=69; IKA=0.8; Lots=1; max_position=...

・ [損益] 部をクリックしてみます。→ 収益が昇順に並びます

パス	損益	総取引数	プロフィットファクター	期待利得	ドローダウン \$	ドローダウン %	パラメーターの入力
196	-2567.89	18	0.51	-142.66	2892.72	28.02	period_bollin=8; IKA=0.3; Lots=1; max_position=1...
292	-2536.65	19	0.54	-133.51	2975.72	28.51	period_bollin=8; IKA=0.4; Lots=1; max_position=1...
388	-2536.65	19	0.54	-133.51	2975.72	28.51	period_bollin=8; IKA=0.5; Lots=1; max_position=1...
484	-2536.65	19	0.54	-133.51	2975.72	28.51	period_bollin=8; IKA=0.6; Lots=1; max_position=1...
580	-2536.65	19	0.54	-133.51	2975.72	28.51	period_bollin=8; IKA=0.7; Lots=1; max_position=1...
676	-2536.65	19	0.54	-133.51	2975.72	28.51	period_bollin=8; IKA=0.8; Lots=1; max_position=1...
194	-2264.23	7	0.25	-323.46	2589.06	25.08	period_bollin=6; IKA=0.3; Lots=1; max_position=1...
386	-1964.38	8	0.35	-245.55	2403.45	23.02	period_bollin=6; IKA=0.5; Lots=1; max_position=1...

・ [パス] をクリックしてみます。→ パス No が昇順・降順に切り替わります

パス	損益	総取引数	プロフィットファクター	期待利得	ドローダウン \$	ドローダウン %	パラメーターの入力
1	0.00	0	0.00	0.00	0.00	0.00	period_bollin=5; IKA=0.1; Lots=1; max_position=1...
2	0.00	0	0.00	0.00	0.00	0.00	period_bollin=6; IKA=0.1; Lots=1; max_position=1...
3	0.00	0	0.00	0.00	0.00	0.00	period_bollin=7; IKA=0.1; Lots=1; max_position=1...
4	0.00	0	0.00	0.00	0.00	0.00	period_bollin=8; IKA=0.1; Lots=1; max_position=1...
5	0.00	0	0.00	0.00	0.00	0.00	period_bollin=9; IKA=0.1; Lots=1; max_position=1...
6	0.00	0	0.00	0.00	0.00	0.00	period_bollin=10; IKA=0.1; Lots=1; max_position=...
7	0.00	0	0.00	0.00	0.00	0.00	period_bollin=11; IKA=0.1; Lots=1; max_position=...
8	0.00	0	0.00	0.00	0.00	0.00	period_bollin=12; IKA=0.1; Lots=1; max_position=...



パス	損益	総取引数	プロフィットファクター	期待利得	ドローダウン \$	ドローダウン %	パラメーターの入力
768	5478.59	71	1.48	77.16	2607.91	16.52	period_bollin=100; IKA=0.8; Lots=1; max_position=...
767	5315.44	71	1.46	74.87	2607.91	16.52	period_bollin=99; IKA=0.8; Lots=1; max_position=...
766	5937.74	70	1.54	84.82	2608.21	15.91	period_bollin=98; IKA=0.8; Lots=1; max_position=...
765	6100.89	70	1.56	87.16	2608.21	15.91	period_bollin=97; IKA=0.8; Lots=1; max_position=...
764	6172.55	70	1.57	88.18	2585.47	15.73	period_bollin=96; IKA=0.8; Lots=1; max_position=...
763	5774.35	71	1.51	81.33	2585.47	16.12	period_bollin=95; IKA=0.8; Lots=1; max_position=...
762	5774.35	71	1.51	81.33	2585.47	16.12	period_bollin=94; IKA=0.8; Lots=1; max_position=...
761	5774.35	71	1.51	81.33	2585.47	16.12	period_bollin=93; IKA=0.8; Lots=1; max_position=...

スクロール

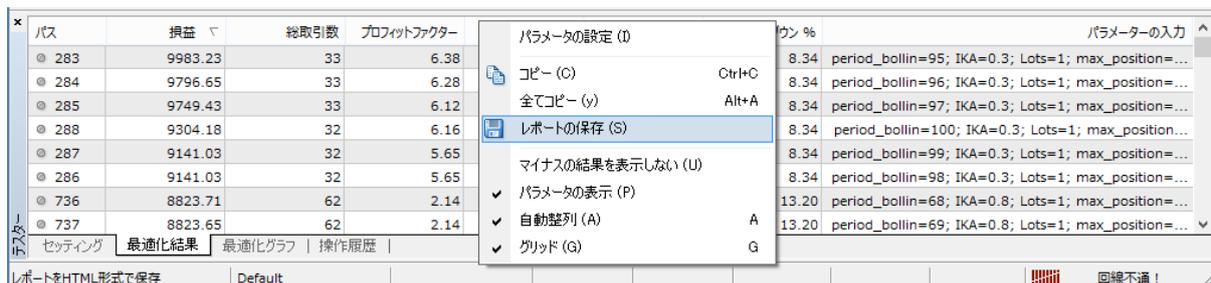


バス	損益	総取引数	プロフィットファクター	期待利得	ドローダウン \$	ドローダウン %	パラメーターの入力
584	1488.98	28	1.28	53.18	2417.66	20.58	period_bollin=12; IKA=0.7; Lots=1; max_position=...
583	1742.86	29	1.33	60.10	2540.58	21.63	period_bollin=11; IKA=0.7; Lots=1; max_position=...
582	1543.93	27	1.31	57.18	1877.05	17.04	period_bollin=10; IKA=0.7; Lots=1; max_position=...
581	-679.53	22	0.86	-30.89	2491.65	21.60	period_bollin=9; IKA=0.7; Lots=1; max_position=1...
580	-2536.65	19	0.54	-133.51	2975.72	28.51	period_bollin=8; IKA=0.7; Lots=1; max_position=1...
579	-1273.24	14	0.70	-90.95	2349.15	21.21	period_bollin=7; IKA=0.7; Lots=1; max_position=1...
578	-1964.38	8	0.35	-245.55	2403.45	23.02	period_bollin=6; IKA=0.7; Lots=1; max_position=1...
577	0.00	0	0.00	0.00	0.00	0.00	period_bollin=5; IKA=0.7; Lots=1; max_position=1...

手順5；最適化レポートを保存してみます

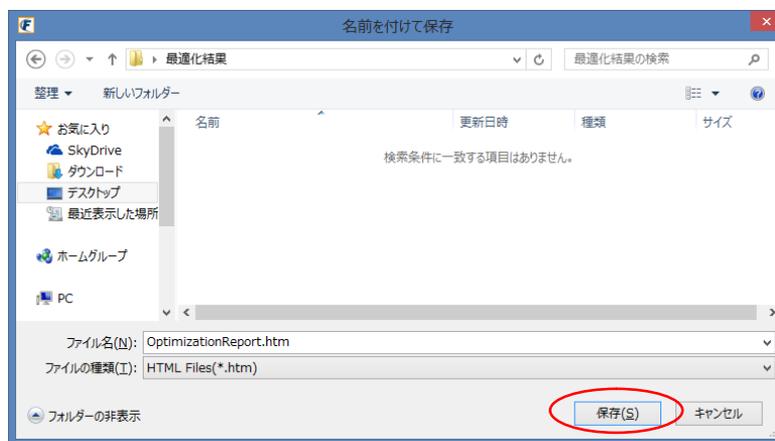
まず「損益」が降順になっていることを確認し、異なる場合は「損益」部をクリックして降順に揃えます、

次に「最適化結果」の上で、「右クリック」－「レポートの保存」を選択、



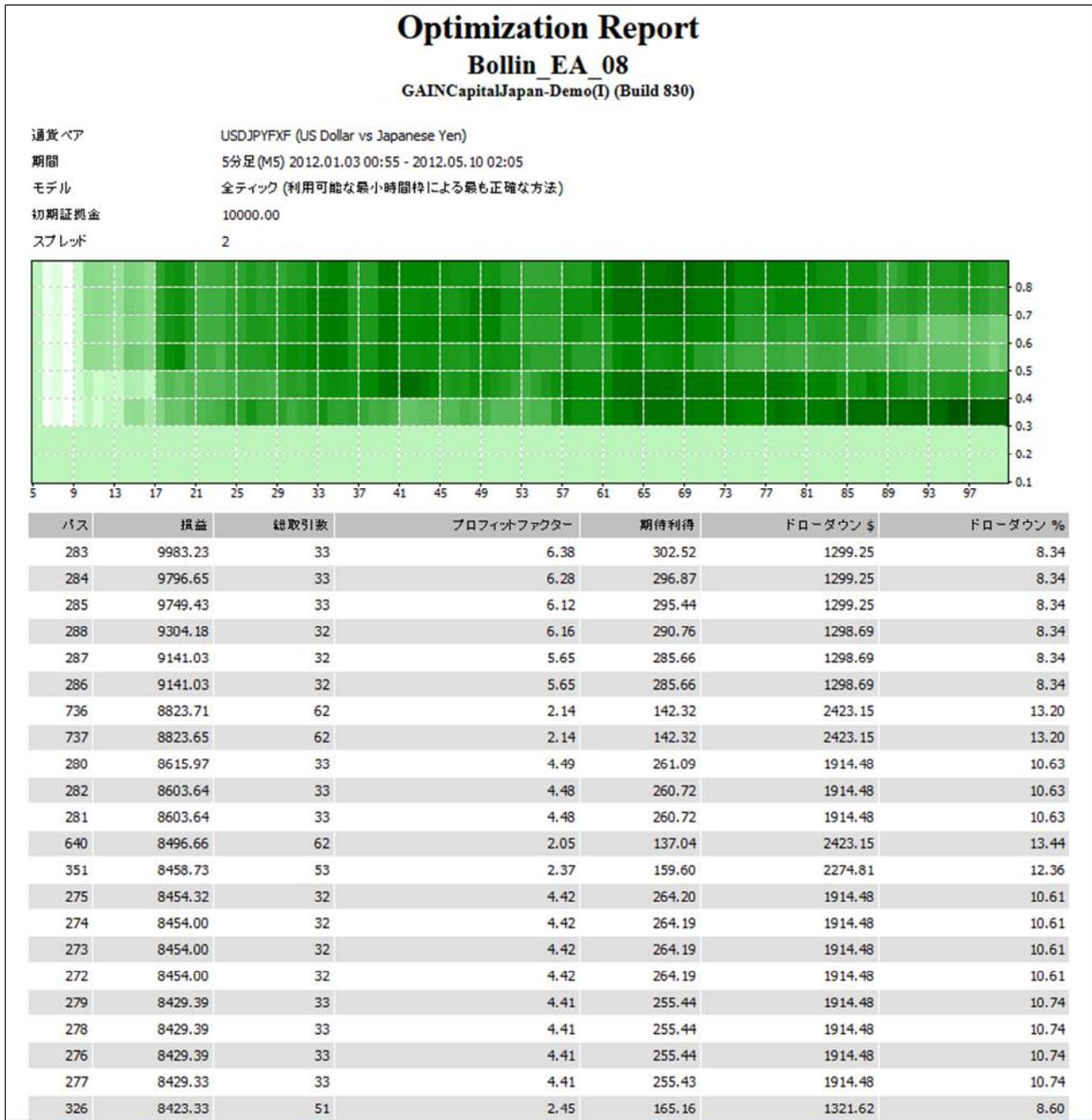
○保存先「フォルダー」等を選択する「名前を付けて保存」ウィンドウが表示されます、保存先ホルダーやファイル名は任意に設定可能です。

デフォルト； [ファイル名] StrategyTester.htm 、[ファイルの種類] HTML



※本書ではデフォルト設定のまま、「保存」を選択

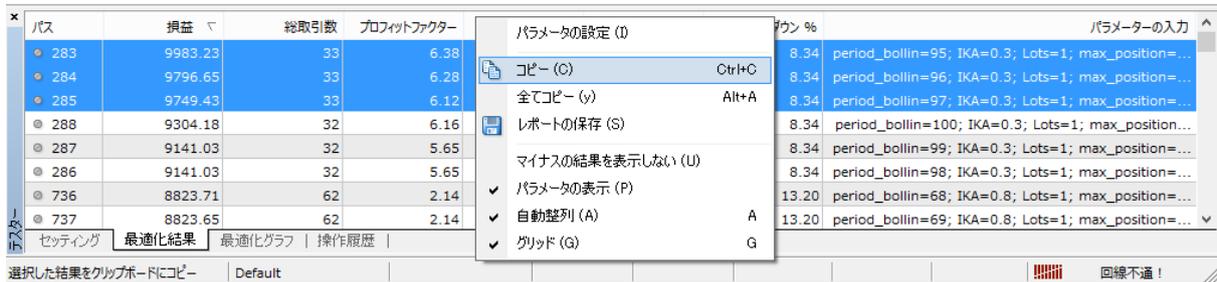
テスト・レポートが保存と同時に表示されます



手順6；最適化結果をコピーしてみます

（[損益] は降順（損益の大きい順）に表示されているとします）

収益の大きいもの3パスを選択し、[右クリック] - [コピー] を選択、



以下に、コピー内容を例示します

283	9983.23 33	6.38	302.52	1299.25	8.34%	0.00000000	period_bollin=95
	IKA=0.3Lots=1	max_position=1	profit=0.6	loss=0.6	profit_2=0.7		
	loss_2=0.5	IZYOU=0.2	shortPeriod_buy=50	mediumPeriod_buy=120			
	short_buy=-0.0044	long_buy=-0.0006	shortPeriod_sell=35				
	mediumPeriod_sell=180	short_sell=0.0044	long_sell=0.0006				
	trendPeriod=300	_up=-0.00024	_down=0.00024	div_=0.04			
284	9796.65 33	6.28	296.87	1299.25	8.34%	0.00000000	period_bollin=96
	IKA=0.3Lots=1	max_position=1	profit=0.6	loss=0.6	profit_2=0.7		
	loss_2=0.5	IZYOU=0.2	shortPeriod_buy=50	mediumPeriod_buy=120			
	short_buy=-0.0044	long_buy=-0.0006	shortPeriod_sell=35				
	mediumPeriod_sell=180	short_sell=0.0044	long_sell=0.0006				
	trendPeriod=300	_up=-0.00024	_down=0.00024	div_=0.04			
285	9749.43 33	6.12	295.44	1299.25	8.34%	0.00000000	period_bollin=97
	IKA=0.3Lots=1	max_position=1	profit=0.6	loss=0.6	profit_2=0.7		
	loss_2=0.5	IZYOU=0.2	shortPeriod_buy=50	mediumPeriod_buy=120			
	short_buy=-0.0044	long_buy=-0.0006	shortPeriod_sell=35				
	mediumPeriod_sell=180	short_sell=0.0044	long_sell=0.0006				
	trendPeriod=300	_up=-0.00024	_down=0.00024	div_=0.04			

※Build 600 代以降のバージョンでは、「ドロダウン%」と「パラメータの入力データ」の間に

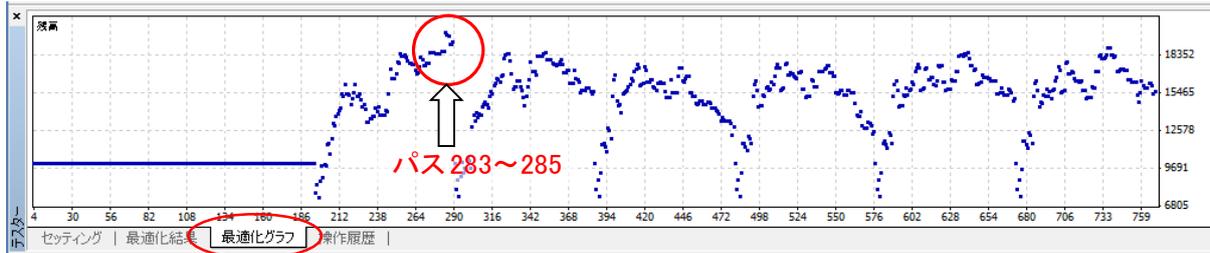
[Cutom] 用の欄が確保されており、「Custum」を使用していないときには、[最適化結果] タブ上には表示されませんが、[最適化結果] のデータをコピーすると「0.00000000」が挿入される様になりました

パス	損益	総取引数	プロフィットファクター	期待利得	ドロダウン \$	ドロダウン %	パラメータの入力
741	16260.06	145	2.11	112.14	2070.23	8.99	period_bollin=73; IKA=0.8; Lots=1; max_position=...
742	16131.23	145	2.10	111.25	2070.23	8.97	period_bollin=74; IKA=0.8; Lots=1; max_position=...
743	15703.55	146	2.04	107.56	2070.23	8.97	period_bollin=75; IKA=0.8; Lots=1; max_position=...

(参考) [収益] が最大のデータ (行) の上で [右クリック] - [パラメータの設定] 選択で、この値をバックテスト用の [パラメータの入力] タブに設定することが出来ます。

(詳細は4章「4. 2. 4」等を参照)

手順 7 ; 改めて [最適化グラフ] タブで最適化の意味を考察します



※X 軸は「パス No」、Y 軸が資産残高（Balance）です。

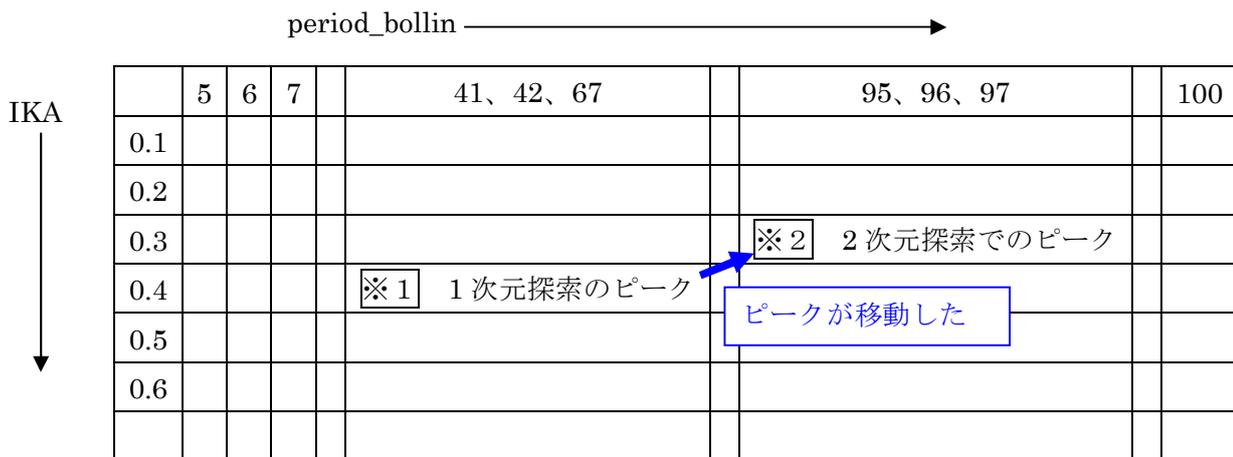
(4) 結果のまとめ

※最適化してみると「1次元」と「2次元」では period_bollin の最適値が変化していることに注意！

実は、色々と試すと直ぐに判るのですが、「パラメータの最適組合せ値」は、「組み合わせる数」を増やしていくと、変わっていく事が多いです。

1次元と2次元での収益が大きい順に「3パス」をそれぞれ選択；

	候補パス	period_bollin	IKA	IZYOU	備考
1次元	37、38、63	41、42、67	(0.4)	—	IKA=0.4に固定
2次元	283、284、285	95、96、97	0.3	—	探索範囲の端っこにピークあり



※1 ; IKA=0.4（固定）で、period_bollin を「5~100」間を1次元スキャンしたときの収益が最大の領域。

※2 ; 2次元 [period_bollin 「5~100」 × IKA 「0.1~0.8」] でスキャンしたときの収益が最大の領域。

結論；

- ・収益が最大になる「最適化変数の組合せ」は、全ての変数について組み合わせるまでは、判りません。
- ・ならば『全ての「最適化変数」を組み合わせる「最適化」を行えば、いいじゃないか』との結論になりそうですが、「最適化実行時間」が途方も無く長時間になり、とても実現できません。
⇒ そこで出番となるのが、「遺伝的アルゴリズム」です。
(詳細は「第4章と第6章」を参照ください)

1. 4 「Bollin_EA_08.mq4」を「2012_05_10」以外の履歴・データでも最適化してみた

※「Bollin_EA_08.mq4」の各種パラメータを、「第4章」で解説する方法で「2012_05_10」以外のデータに対して最適化後、そのextern値を用いてバックテストした結果を以下に示します。

○比較する履歴・データとしては、「2012_05_10」、「2014_10_17」、「2015_05_07」を採用しました。（「2014_10_17」は本稿のメイン部を執筆中に保存したものです）

1. 4. 1 最適化後のextern値

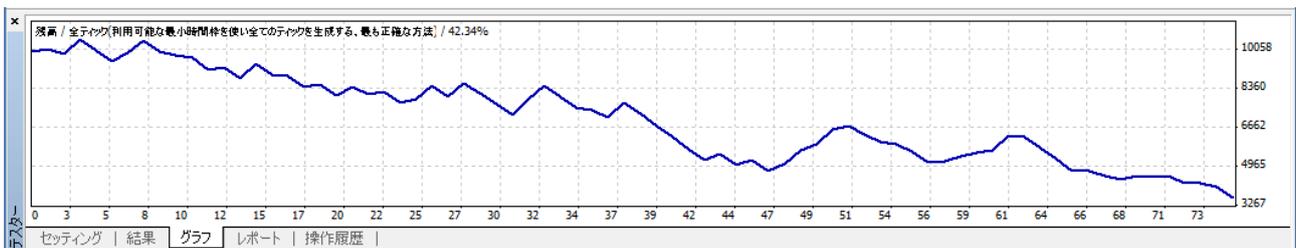
大項目		最適化実施履歴・データ⇒	2012_05_10	2014_10_17	2015_05_07
		extern項目（パラメータ）	Bollin_EA_08での設定値		
損益パラメータ	Open時 profit		0.6	0.8	0.8
	Open時 loss		0.6	0.7	0.8
	足内 profit_2		0.7	0.8	1.2
	足内 loss_2		0.5	1.1	0.6
Bollinパラメータ	周期 period_bollin		40	80	80
	幅制限上限 IKA		0.4	1.2	1.3
	幅制限下限 IZYU		0.2	0.2	0.0
Filter パラメータ	買い パラメータ	短周期 shortPeriod_buy	50	55	5
		中周期 mediumPeriod_buy	120	100	140
		UP短勾配 short_buy	-0.0044	-0.001	-0.006
		UP長勾配 long_buy	-0.0006	-0.0003	-0.0005
	売り パラメータ	短周期 shortPeriod_sell	35	75	75
		中周期 mediumPeriod_sell	180	140	100
		DOWN短勾配 short_sell	+0.0044	0.006	+0.008
		DOWN長勾配 long_sell	+0.0006	0.0013	+0.0011

・「2012_05_10」と「2014_10_17」「2015_05_07」の履歴・データでは、最適化後のパラメータ値がかなり異なることを確認できます。

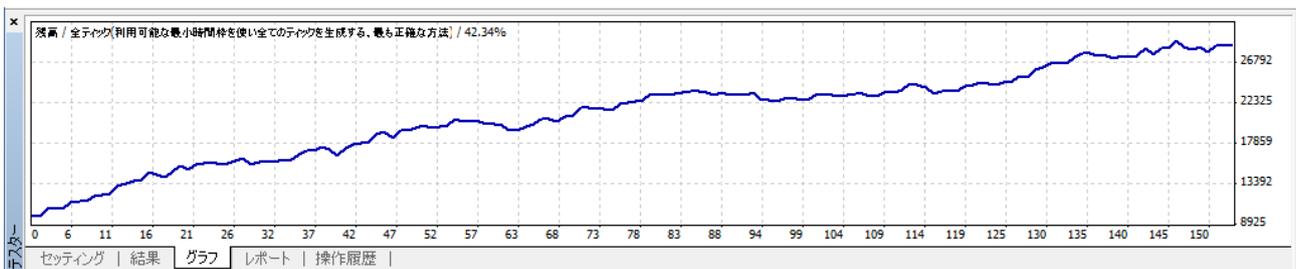
1. 4. 2 バックテスト結果1； [2014_10_17]履歴・データの場合

(1) 資産グラフ（最適化の前後）

<最適化前>・・・[「2012_05_10」データに最適化されたextern値をそのまま使用した場合



<最適化後>



(2) バックテスト・レポート（最適化後）

テストバー数	66830	モデルティック数	1623803	モデリング品質	42.34%
不整合チャートエラー	2				
初期証拠金	10000.00			スプレッド	2
総損益	18679.45	総利益	32382.85	総損失	-13703.40
プロフィットファクター	2.36	期待利得	122.09		
絶対ドローダウン	632.78	最大ドローダウン	1571.03 (7.56%)	相対ドローダウン	7.65% (969.08)
総取引数	153	ショートポジション(勝率%)	27 (62.96%)	ロングポジション(勝率%)	126 (63.49%)
		勝率(%)	97 (63.40%)	負率(%)	56 (36.60%)
	最大	勝トレード	792.79	負トレード	-720.03
	平均	勝トレード	333.84	負トレード	-244.70
	最大	連勝(金額)	11 (4011.48)	連敗(金額)	6 (-984.28)
	最大化	連勝(トレード数)	4011.48 (11)	連敗(トレード数)	-1030.77 (3)
	平均	連勝	3	連敗	2

(3) ストラテジー・テスター・レポート（最適化後）

Strategy Tester Report

Bollin_EA_08

GAINCapitalJapan-Demo(I) (Build 830)

通貨ペア: USDJPYFXF (US Dollar vs Japanese Yen)
 期間: 5分足(M5) 2013.11.15 14:15 - 2014.10.17 17:55
 モデル: 全ティック (利用可能な最小時間枠による最も正確な方法)

パラメーター: Lots=1; max_position=1; profit=0.8; loss=0.7; profit_2=0.8; loss_2=1.1; period_bollin=80; IKA=1.2; IZYOU=0.2; shortPeriod_buy=55; mediumPeriod_buy=100; short_buy=-0.001; long_buy=-0.0003; shortPeriod_sell=75; mediumPeriod_sell=140; short_sell=0.006; long_sell=0.0013; trendPeriod=300; _up=-0.00024; _down=0.00024; div_=-0.04;

テストバー数	66830	モデルティック数	1623803	モデリング品質	42.34%
不整合チャートエラー	2				
初期証拠金	10000.00			スプレッド	2
総損益	18679.45	総利益	32382.85	総損失	-13703.40
プロフィットファクター	2.36	期待利得	122.09		
絶対ドローダウン	632.78	最大ドローダウン	1571.03 (7.56%)	相対ドローダウン	7.65% (969.08)
総取引数	153	ショートポジション(勝率%)	27 (62.96%)	ロングポジション(勝率%)	126 (63.49%)
		勝率(%)	97 (63.40%)	負率(%)	56 (36.60%)
	最大	勝トレード	792.79	負トレード	-720.03
	平均	勝トレード	333.84	負トレード	-244.70
	最大	連勝(金額)	11 (4011.48)	連敗(金額)	6 (-984.28)
	最大化	連勝(トレード数)	4011.48 (11)	連敗(トレード数)	-1030.77 (3)
	平均	連勝	3	連敗	2

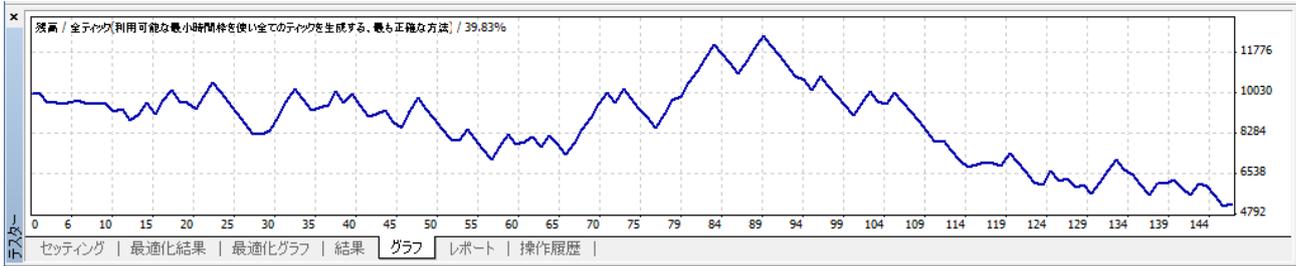
残高 / 全ティック(利用可能な最小時間枠を使い全てのティックを生成する。最も正確な方法) / 42.34%

#	時間	取引種別	注文番号	数量	価格	S/L	T/P:決済指値	損益	残高
1	2013.11.19 02:30	buy	1	1.00	100.19	0.00	0.00		
2	2013.11.20 03:30	close	1	1.00	100.08	0.00	0.00	-109.81	9890.19
3	2013.11.21 02:40	buy	2	1.00	100.11	0.00	0.00		

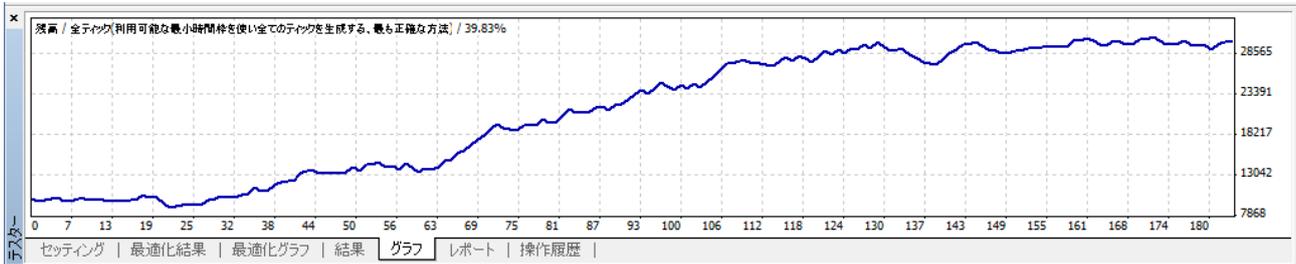
1. 4. 3 バックテスト結果2； [2015_05_07] ヒストリー・データの場合

(1) 資産グラフ（最適化の前後比較）

<最適化前>・・・[2012_05_10] データに最適化された extern 値をそのまま使用した場合



<最適化後>

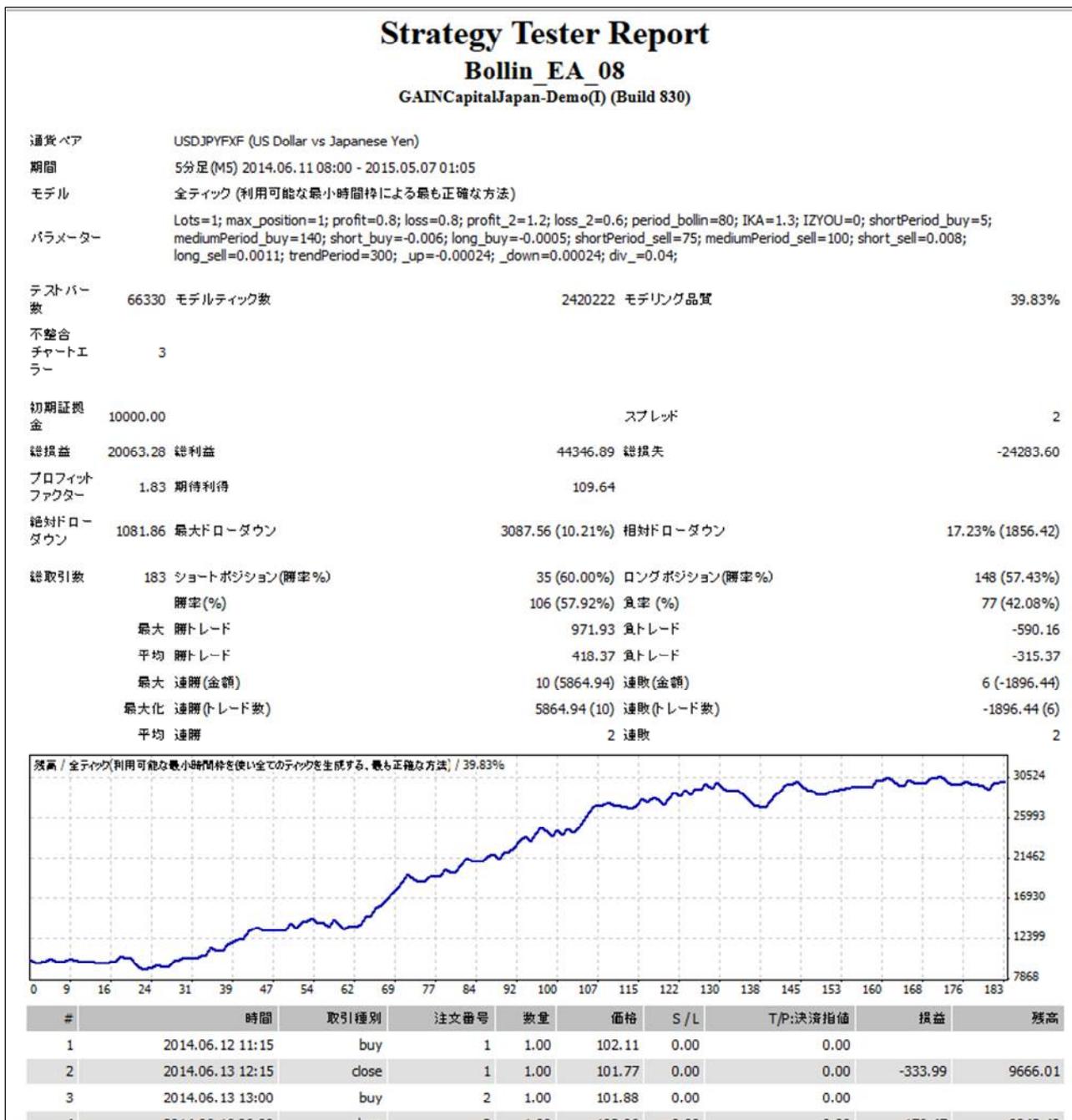


(2) バックテスト・レポート（最適化後）

テストバー数	66330	モデルティック数	2420222	モデリング品質	39.83%
不整合チャートエラー	3				
初期証拠金	10000.00	総利益	44346.89	スプレッド	2
総損益	20063.28	総損失	-24283.60		
プロフィットファクター	1.83	期待利得	109.64		
絶対ドローダウン	1081.86	最大ドローダウン	3087.56 (10.21%)	相対ドローダウン	17.23% (1856.42)
総取引数	183	ショートポジション(勝率%)	35 (60.00%)	ロングポジション(勝率%)	148 (57.43%)
		勝率(%)	106 (57.92%)	負率(%)	77 (42.08%)
	最大	勝トレード	971.93	負トレード	-590.16
	平均	勝トレード	418.37	負トレード	-315.37
	最大	連勝(金額)	10 (5864.94)	連敗(金額)	6 (-1896.44)
	最大化	連勝(トレード数)	5864.94 (10)	連敗(トレード数)	-1896.44 (6)
	平均	連勝	2	連敗	2

設定 | 結果 | **グラフ** | **レポート** | 操作履歴

(3) ストラテジー・テスター・レポート（最適化後）



※ [2014_10_17] と [2015_05_07] の何れのヒストリー・データに対しても最適化が有効であることが判ります。
 しかし、言い方を変えるなら、時期の異なるヒストリー・データごとに、最適化が必要となるわけです。

1. 5 MT4を「オフライン」に設定する方法

「1. 1. 3（4）」で解説しなかった、具体的な手順を示します。

MT4のオフライン設定方法（例）；（再掲）

方法	オフライン設定方法	表示	Build		
			419 ~ 840	880, 890	900 ~
A	無効なプロキシサーバーを設定し、再起動	[回線不通!]	○	×	○
B	接続後、「無効な口座」に設定	[無効な口座]	(未確認)	○	○
C	接続後、ネット用コネクタを切断する	[回線不通!]	(未確認)	○	—

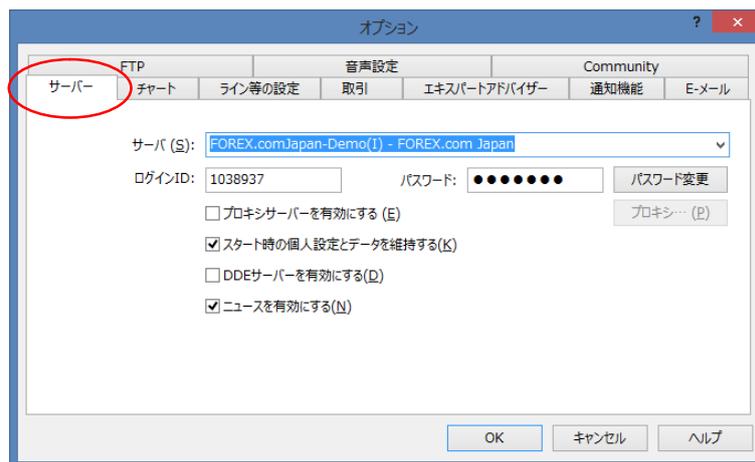
本書では、設定と復旧が簡単な方法「AとB」の方法について順次解説します。

（「C」は解説を省略）

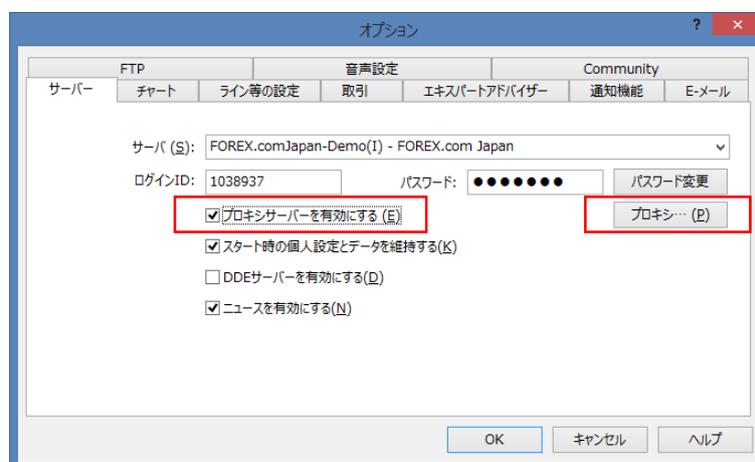
1. 5. 1 プロキシサーバーの設定を利用する

※無効な「プロキシサーバー」を設定することで、オフライン状態にします。

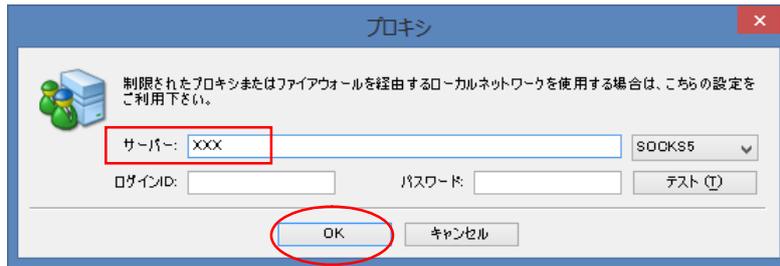
手順1；プルダウン・メニューの [ツール] - [オプション] を選択、
[オプション] ウィンドウが表示されたら、次に [サーバー] タブを開きます



手順2；[プロキシサーバーを有効にする] にチェックを入れ、[プロキシ...] を選択



手順3；[サーバー] に「XXX」など（適当な、あり得ない名称）を入力し、[OK] を選択



※ [オプション] ウィンドウに戻ったら、更に [OK] を選択

手順4；最後にMT 4を終了し、再起動すると下記の様に「オフライン」になります

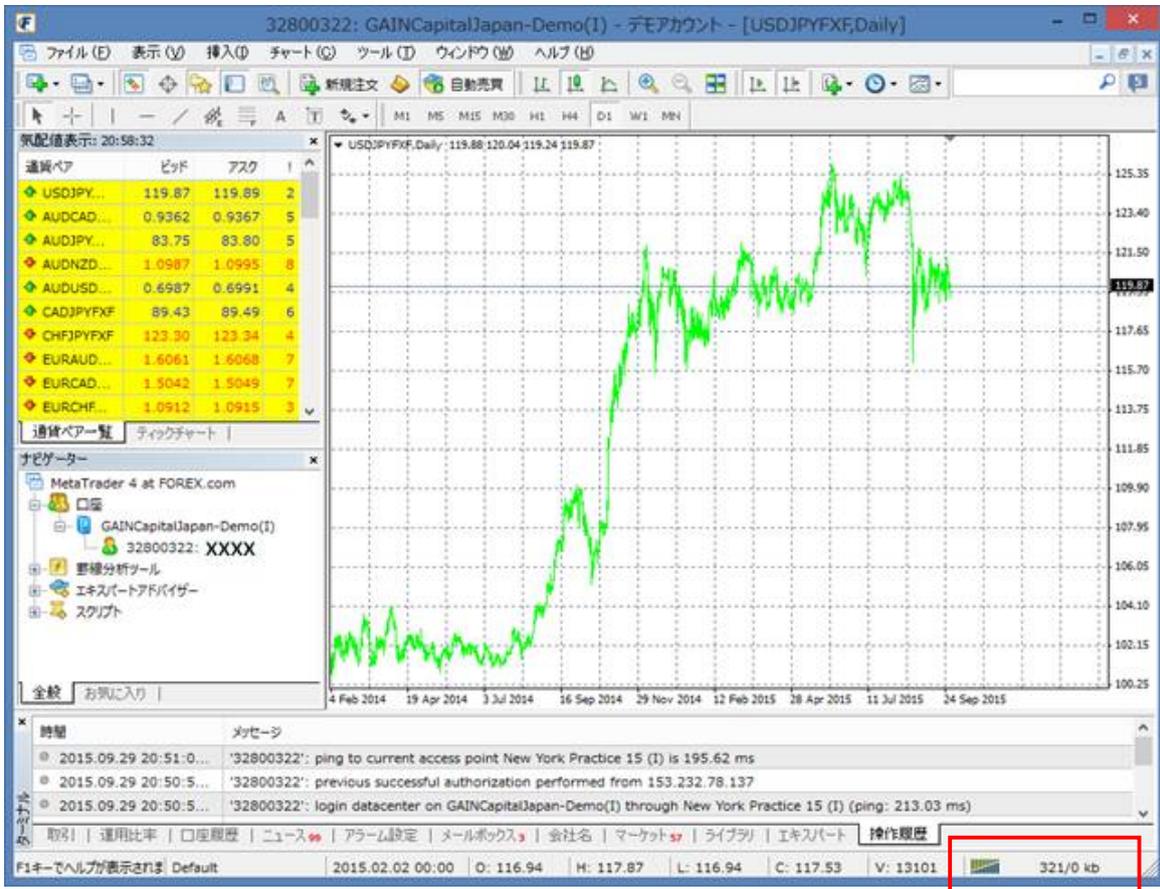


※右下に [回線不通!] が表示されていることを確認します。

1. 5. 2 「無効な口座」を利用する

一旦、正常にログインしてから、無効な口座でログインし直して、
 [回線不通!] ⇒ [無効な口座] へと、オフライン状態にする

手順1；一旦、正規の口座でMT4を立ち上げて回線を接続します



手順2；先ず、メールボックス中の「Regisration」を保存しておきます（2回目からは不要）

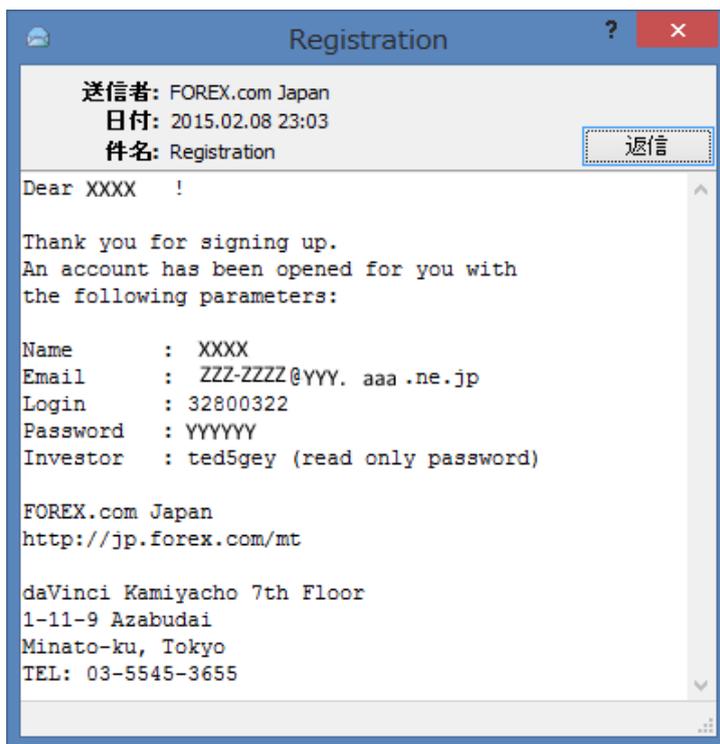
理由；以下の操作で解説する、一旦 [無効な口座] にした状態のまま、誤って MT4 を終了すると、再起動したときに「Regisration」が失われるためです。
 [無効な口座] にした状態から、MT4 を終了せずに再度「正規の口座」でログインし直せば「Regisration」は保持し続けることができます。

「Regisration」を選択（ダブルクリック）します

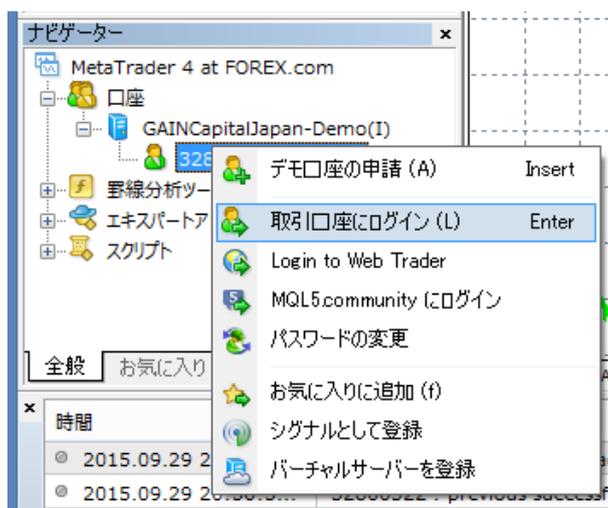


[Regisration] ウィンドウが開きます。

例；下記の内容、特に「Login」と「Password」はコピー&ペーストなどで保存しておきます。（テキストで保存すると使い易い）



手順3 ; 現在使っている口座 No の上で、[右クリック] – [取引口座にログイン] を選択



[ログイン] ウィンドウが開きます



手順4；[ログインID:] はそのまま、[パスワード:] を不正なものに入れ替えます
同時に、[ログイン情報を保持] のチェックは外しておきます



不正なパスワード（例；1234567）を入力し、[ログイン] を選択



手順4；ターミナル表示に戻りますので、[回線不通！] が表示されていることを確認します



数秒後に、[回線不通！] ⇒ [無効な口座] へと表示が変わります



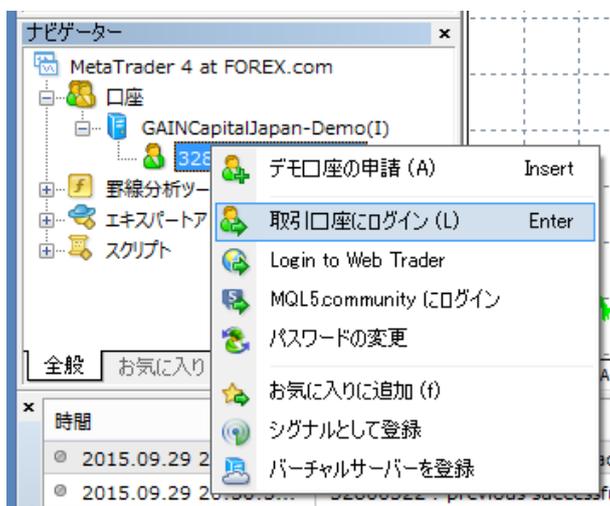
この [無効な口座] の状態でも、[回線不通！] の場合と同じように、回線が切り離された状態ですので、オフライン状態として「バックテスト」や「最適化」を実行することが出来ます。（Build880、890 の場合に利用可能であることを確認済みです）

MT4 を再起動することなしに「オフライン状態」になりましたので、この状態で「バックテスト」や「最適化」等を実施します

手順5；必要な作業が終わったら、オンライン状態に戻しておくために、正規の [ログインID:] と [パスワード:] でログインします、これには「手順2」で保存した [Registration] の「Login」と「Password」を使います。

Name	: XXXX
Email	: ZZZ-ZZZZ@YYY. aaa .ne.jp
Login	: 32800322
Password	: YYYYYY

口座 No の上で、[右クリック] - [取引口座にログイン] を選択



[パスワード] が空白の [ログイン] ウィンドウが開きます

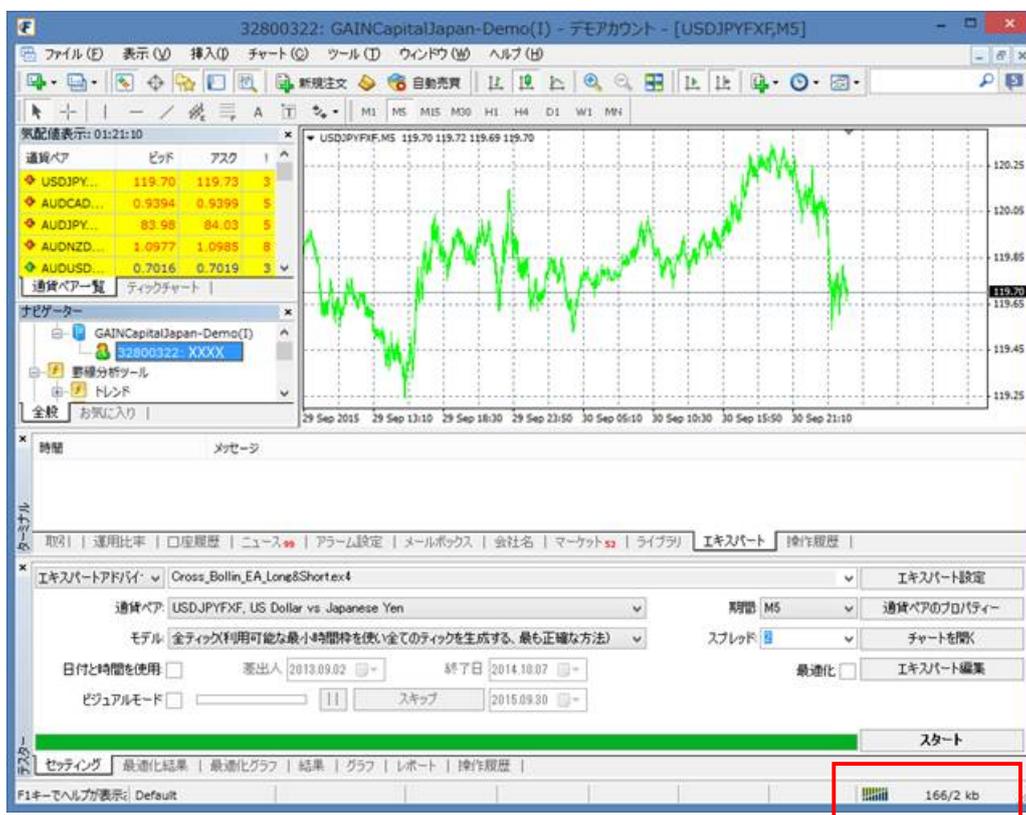
保存していた正規の「パスワード」を入力します



更に、[ログイン情報を保持] にチェックを入れて、[ログイン] を選択



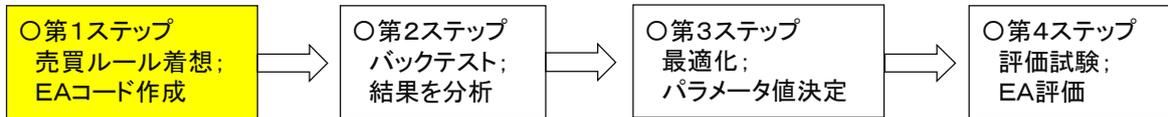
「バックテスト」や「最適化」等を終了したままの状態で、
回線が再度接続されます



2章以降のダイジェスト内容；

「2章以降」は、正規品から一部「記述、画像や表」を切り取りダイジェストとしました、記載内容のイメージが判って頂ければ幸いです。

第2章；EAコード開発と準備概要 一部抜粋



解説手順（経緯）概要

※各章では以下の様に、先ず「サンプルEA」の開発ステップをフロー図で示し、続く個別の節で各ステップの詳細を順に解説していきます。

● 試行錯誤の中から、一つのアイデア（売買ルール）を着想する
⇒ 本書例；「フ型フィルター」は有効だろうか？

① 作成EAの基本「仕様と構造」を決める

<インディケータを用いた「売買ルール」の検証>

② 売買ルール検証用「インディケータ1」を作成；
⇒ 本書例；「Cross_Bollin_DLL.mq4」
ボリンジャーバンドの一定要件を満たす売買ポイント全てに
「縦線」を表示する

③ 売買ルール検証用「インディケータ2」を作成；
⇒ 本書例；「Filter_fu_type_DLL.mq4」
指定売買ポイントに「フ型フィルター（2本線）」を表示させます
・DLL「shared_memory.dll」を使用することで、表示するポイントを指定

④ 売買ルール検証用「インディケータ3」を作成；
⇒ 本書例；「Cross_Bollin_Filtered.mq4」
目視上OKの「フ型フィルター（2本線）勾配値」を
「Cross_Bollin_DLL.mq4」にセットしたものを
「Cross_Bollin_Filtered.mq4」とする

⑤ EAを作成；
⇒ 本書例；「Cross_Bollin_EA.mq4」
・インディケータ「Cross_Bollin_Filtered.mq4」から、変換して
EA「Cross_Bollin_EA.mq4」を作成する。
（同時に、NewBar 判定や1足内の処理等も追加する）

● EA作成；第1ステップを完了 ⇒ 次ステップ（第3章；バックテスト）へ

<概要；「サンプル EA」の設計原理を「検証用のインディケータ」で解説しています>

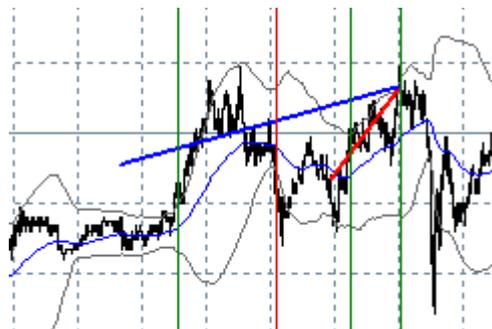
検証用インディケータでは；

※売買ルール検証用インディケータでは、上記のルールで生成した
 買いシグナル部には「**緑色**の縦線」を、売りシグナル部には「**赤色**の縦線」を
 表示することにしました。

売り 買い

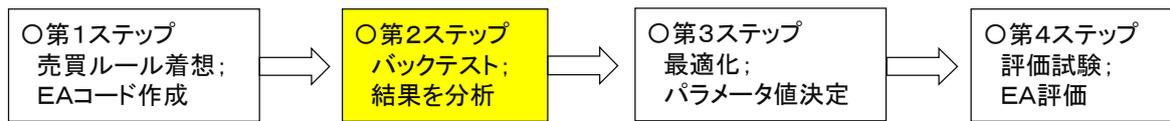
上記の「売買シグナル」に「フ型」フィルターを乗せて、シグナルの妥当性を判定します。

	売買シグナル「起点」部での「フ型」フィルター組合せ例	目視判定 トレンド・フォロー狙い	(参考) ブレイクアウト狙いなら
Bollinger バンドが生成する 「買いシグナル」	中期回帰直線 短期回帰直線	採用 OK	
		不採用 NG	(採用か?)
Bollinger バンドが生成する 「売りシグナル」		採用 OK	
		不採用 NG	(採用か?)



「フ型」フィルターによる判定は、あくまで「売買シグナル」の起点から先の「未来チャート」が未だ存在していないものとして行います。

第3章；バックテスト 一部抜粋



解説手順（経緯）概要

※以下に、再現のために実施したバックテストの手順（要点）をフロー図で示します、
続く個別の節で、各ステップの詳細内容を解説します。



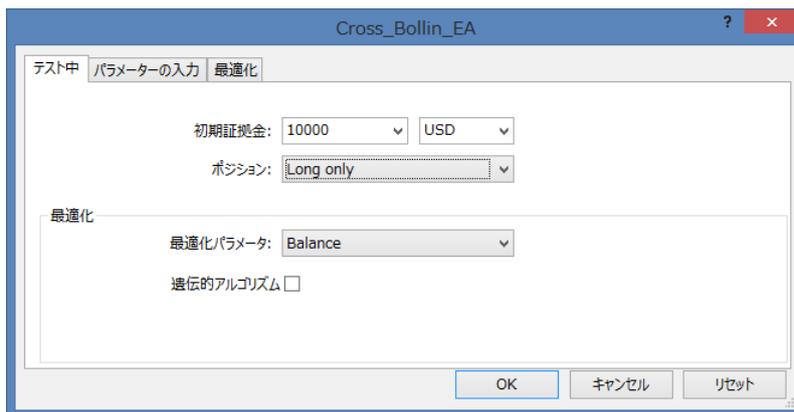
<概要；「サンプルEA」を色々な側面からバックテストした結果を解説しています>

「Long & Short」での結果を観る；

資産残高 [グラフ]

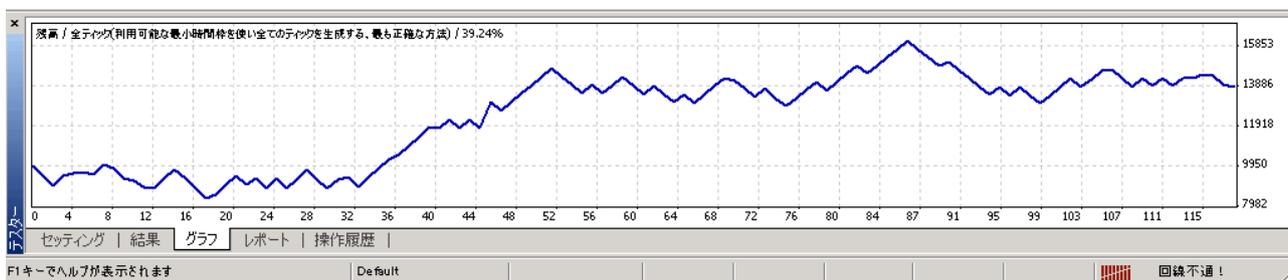


次に [ポジション : Long only] で実行してみる



資産残高 [グラフ]

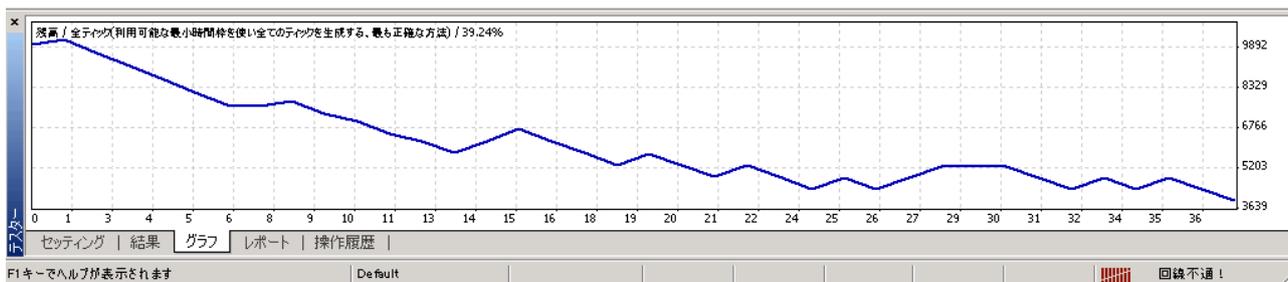
▼ !!何と、結構いけそうな結果がでた (右肩上がり)



更に [ポジション : Short only] で実行してみる

資産残高 [グラフ]

▼ こちらの場合は、思わしくない! (右肩下がり)



バックテストの結果が悪かったと言って、即座にEAを捨ててはいけません、

「バックテスト」は開発のためのツールなのです。

様々な側面から「バックテスト」を行うと、突破口が開ける場合があります。

次ステップ（最適化）への準備

※上記の内容から判断して、「extern 変数（最適化変数）」を分離することにしました、
分離前と分離後のコード名称は下記です。

分離前；「Cross_Bollin_EA.mq4」 ⇒ 分離後；「Cross_Bollin_EA_Long&Short.mq4」

[建玉（buy_IN と sell_IN）用の exetrn 変数]

・「買い IN（buy_IN）、売り IN（sell_IN）」の exetrn 変数（最適化変数）を分離して、
それぞれ個別に「最適化」を行える形にした。

Cross_Bollin_EA.mq4； （exetrn 変数部のみ抜き出して記載）

```
// ③フィルター設定
extern int shortPeriod=35;
extern int mediumPeriod=140;
extern double short_slope=0.004;
extern double long_slope=0.0009;
```

（分離した） ▼

Cross_Bollin_EA_Long&Short.mq4；

```
// ③フィルター設定-----
extern int shortPeriod_buy=35;
extern int mediumPeriod_buy=140;
extern double short_slope_buy=-0.004;
extern double long_slope_buy=-0.0009;
//
extern int shortPeriod_sell=35;
extern int mediumPeriod_sell=140;
extern double short_slope_sell=0.004;
extern double long_slope_sell=0.0009;
```

[5分足内で発生する手仕舞いのレベル用 exetrn 変数]

・「5分足内での Tick 値による手仕舞いレベル（利確・損切り）」を、
「Open 値で判断する手仕舞いレベル（利確・損切り）」から切り離し（分離し）て、
それぞれ個別に「最適化」を行える形にした。

Cross_Bollin_EA.mq4； （extern 変数部のみ抜き出して記載）

```
// ①損益設定
extern double profit=0.40;
extern double loss=0.40;
```

（分離した） ▼

Cross_Bollin_EA_Long&Short.mq4；

```
// ①損益設定-----
extern double profit=0.40;
extern double loss=0.40;
// 5分足内での手仕舞いレベル
extern double profit_2=0.40;
extern double loss_2=0.40;
```

分離前後の extern 変数（最適化変数）部分をまとめて比較する

Cross_Bollin_EA.mq4 ; (extern 変数部のみ抜き出して記載)

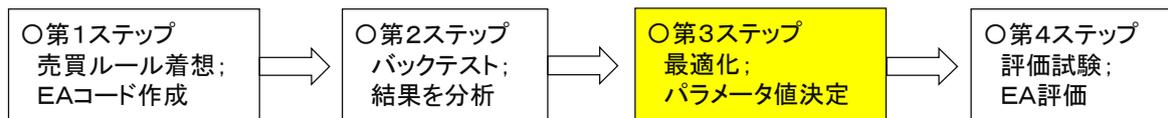
```
// ①損益設定
extern double profit=0.40;
extern double loss=0.40;
// ②Bollinの幅
extern int period_bollin=40;
extern double IZYOU=0.10;
extern double IKA=0.30;
// ③フィルター設定
extern int shortPeriod=35;
extern int mediumPeriod=140;
extern double short_slope=0.004;
extern double long_slope=0.0009;
//=====
```



Cross_Bollin_EA_Long&Short.mq4 (最適化前品) ;

```
// ①損益設定-----
extern double profit=0.40;
extern double loss=0.40;
// 5分足内での手仕舞いレベル
extern double profit_2=0.40;
extern double loss_2=0.40;
// ②Bollinの幅-----
extern int period_bollin=40;
extern double IZYOU=0.10;
extern double IKA=0.30;
// ③フィルター設定-----
extern int shortPeriod_buy=35;
extern int mediumPeriod_buy=140;
extern double short_slope_buy=-0.004;
extern double long_slope_buy=-0.0009;
//
extern int shortPeriod_sell=35;
extern int mediumPeriod_sell=140;
extern double short_slope_sell=0.004;
extern double long_slope_sell=0.0009;
//=====
```

第4章；最適化 一部抜粋



<概要；2章、3章で検討したEAの最適化方法と、その結果を解説しています>

[遺伝的アルゴリズム] を使うと、全探索に比較して「最適値」探索時間を大幅に短縮できます、強力である一方、とても癖のあるアルゴリズムなので、使い方に慣れる必要があります。

解説手順（経緯）概要

- EA作成；[第2ステップ；第3章]の結果から最適化対象EAを決める
本書； 下記の2つを選定しました
 - ・「Cross_Bollin_EA_ShortOnly.mq4」 ; 収益が右肩下がりEA
 - ・「Cross_Bollin_EA_Long&Short.mq4」 ; 「売り・買い」パラメータ分離EA



①選定したEAの最適化を実行する； 遺伝的アルゴリズムによる



②最適化結果を解析して、「最適 extern 変数値」を決定（選定）する
選定時には「収益」項目だけではなく、「安定性やドロウダウン」など他の項目も考慮に入れる



③「最適 extern 変数値」を設定したEAを準備する
本書；
最適化前の「extern 値」設定EA；Cross_Bollin_EA_Long&Short.mq4
⇒ 最適化後の「extern 値」設定EA；Cross_Bollin_EA_fitted.mq4

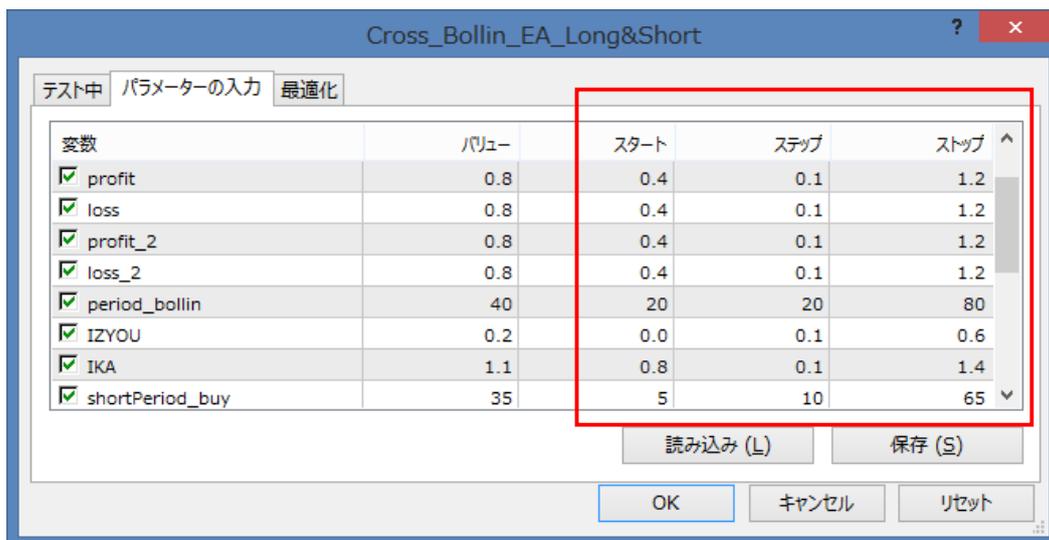
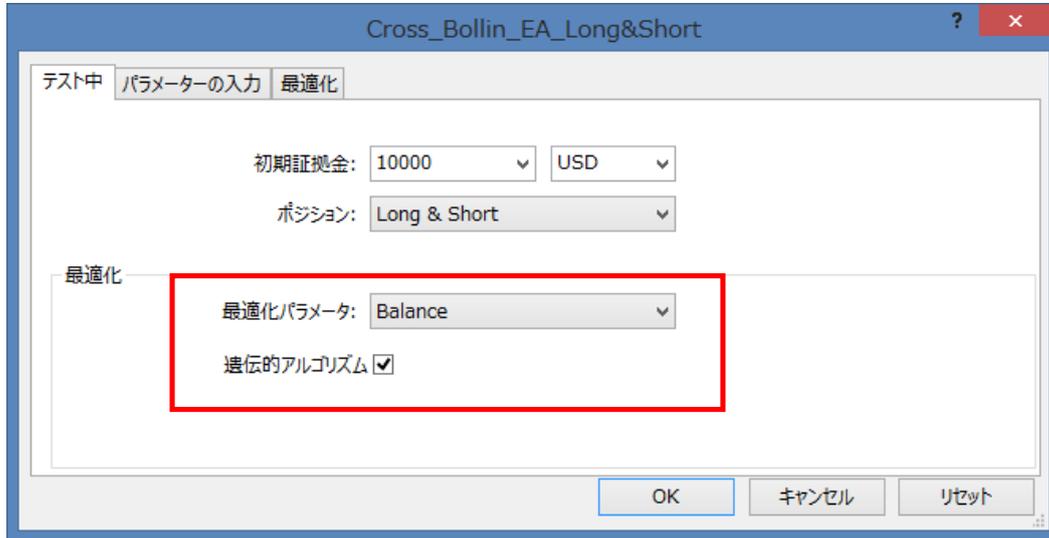


● EA作成；第3ステップを完了 ⇒ 次ステップ（第5章；EA評価）へ

「売・買」で extern 変数を分離したEA「Cross_Bollin_EA_Long&Short.mq4」を最適化する

※実施目的；

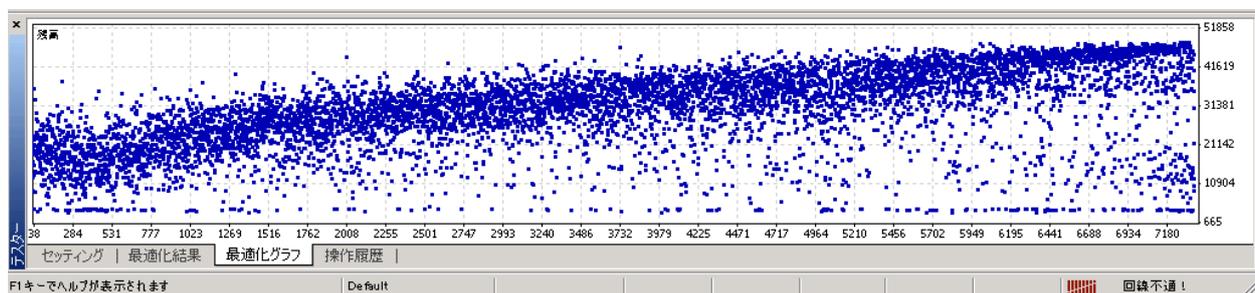
- ・「売りIN (sell_IN)」と「買いIN (buy_IN)」で extern 変数を分離したEAで、各 extern 変数（パラメータ）を一括で最適化した場合の損益等を確認します。（※ [遺伝的アルゴリズム] を使います）

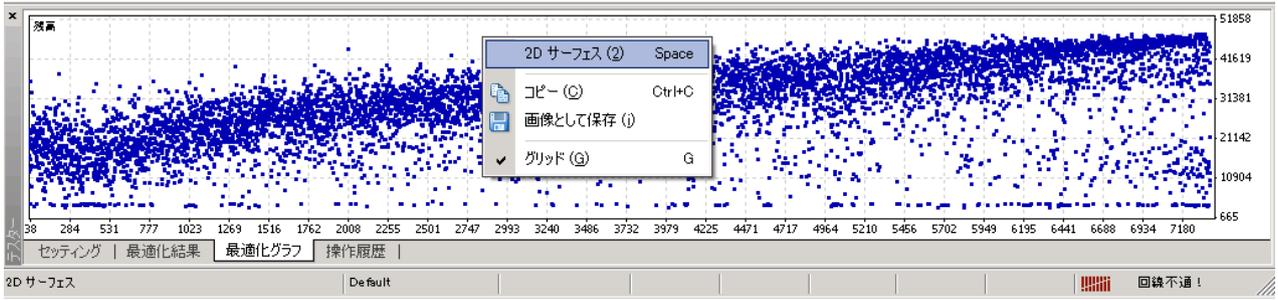


[最適化グラフ] を観る ・ ・ 遺伝的アルゴリズムを使った場合の表示です

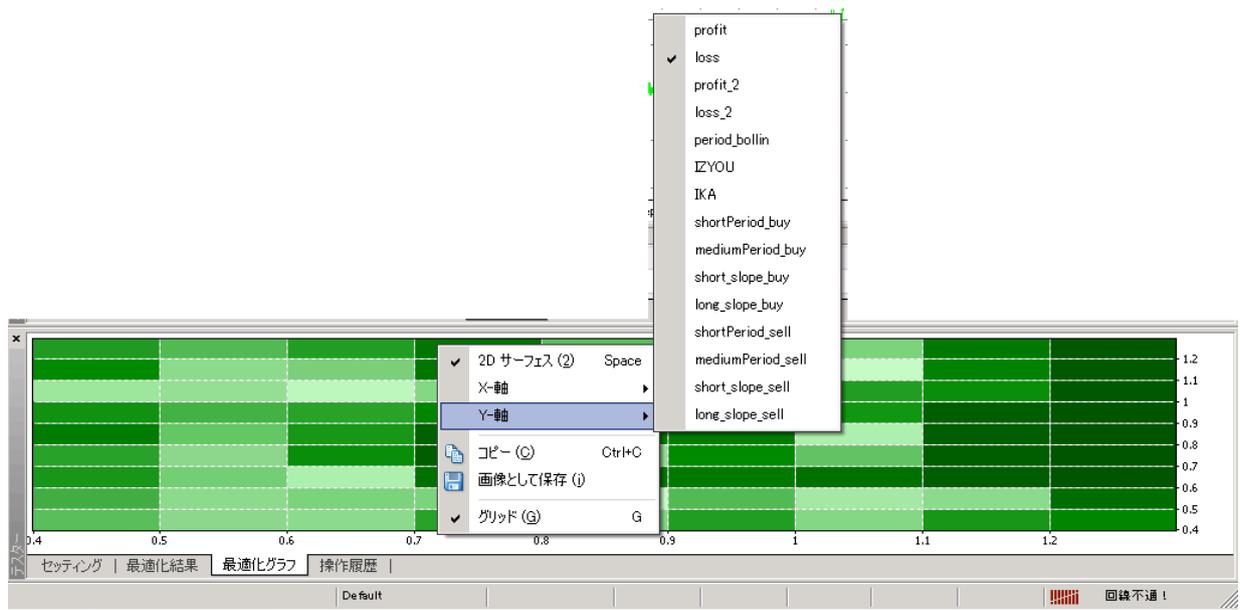
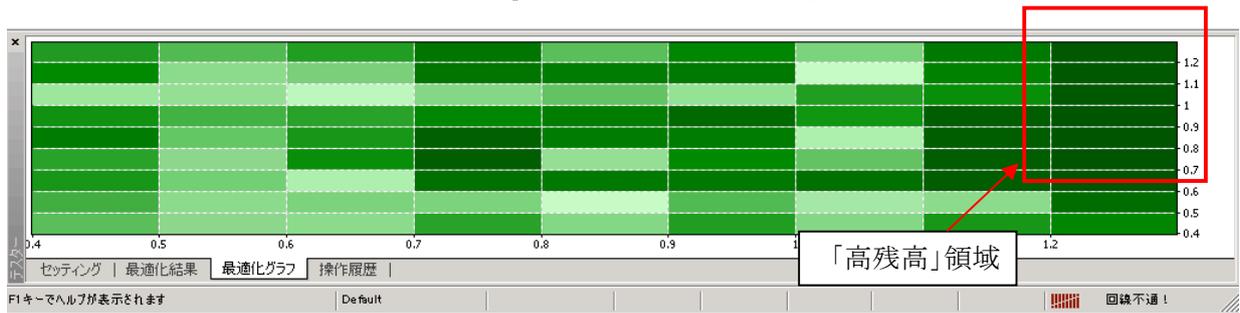
最適化の進行と残高（収益）の関係を観ます

- ・ [最適化グラフ] タブを開くと、デフォルトは、X軸がパス No、Y軸が収益になっています。グラフは左から右へと「最適化プロセスが進行」するのに伴う「最適化の状況」を表示。





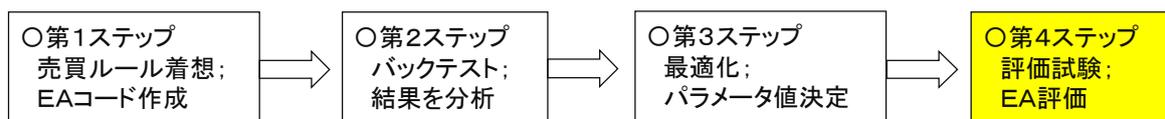
※ 2次元表示「X軸・Y軸」のデフォルトは、[パラメータの入力] タブで最適化の変数として使う様に指定した「最適化変数 (exetrn 変数)」の1個目と2個目になっています。
従って、本2D表示では「X軸=「profit」、Y軸=「loss」となります



最大収益が得られた組合せデータを「パラメータの入力」タブに設定し、バックテストした結果
[損益] タブをクリックして、降順表示にする
⇒ 最大収益のパスNoが、一番上に表示される



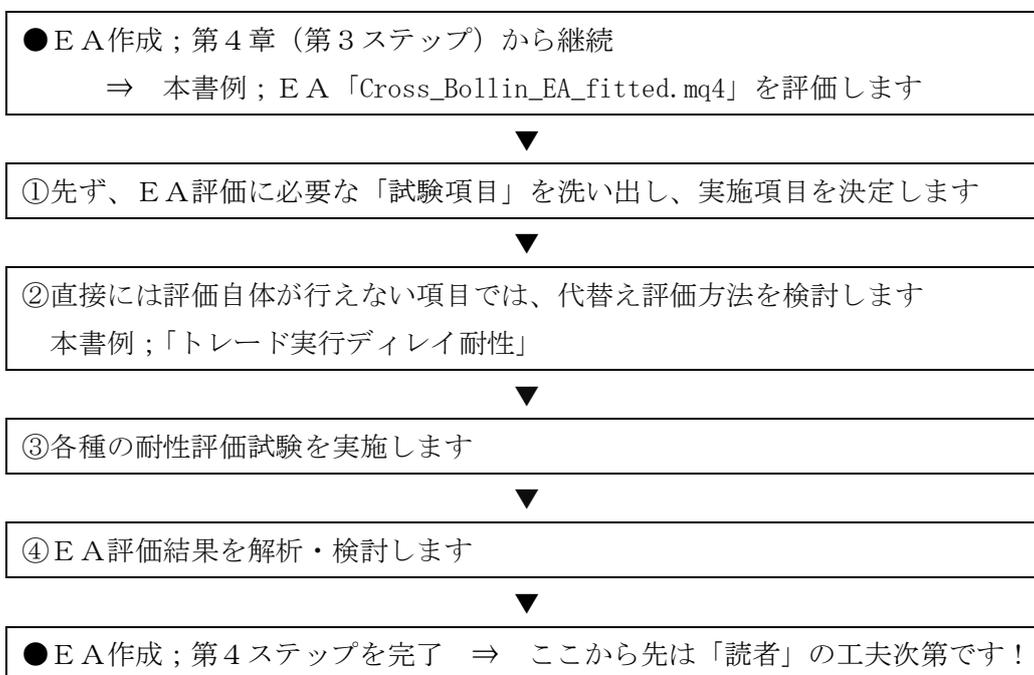
第5章；EAの環境耐性評価について 一部抜粋



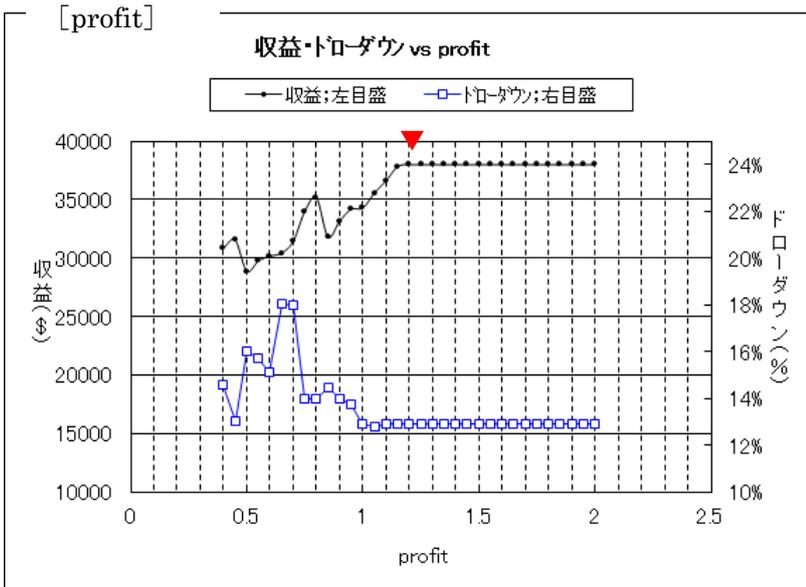
<概要；4章で作成したEAの「評価方法の解説」と、幾つかの評価結果を解説しています>

EAの評価と言うと、究極は「リアル」でのランニング試験に尽きますが、それ以前にやるべきことが実はたくさんあります、その一端を理解頂ければ幸いです。
(評価結果をどの様に活かすか、読者の腕次第)

開発手順（経緯）概要



extern 変数のスキャン結果グラフ表示一覧； 本書での例を示します



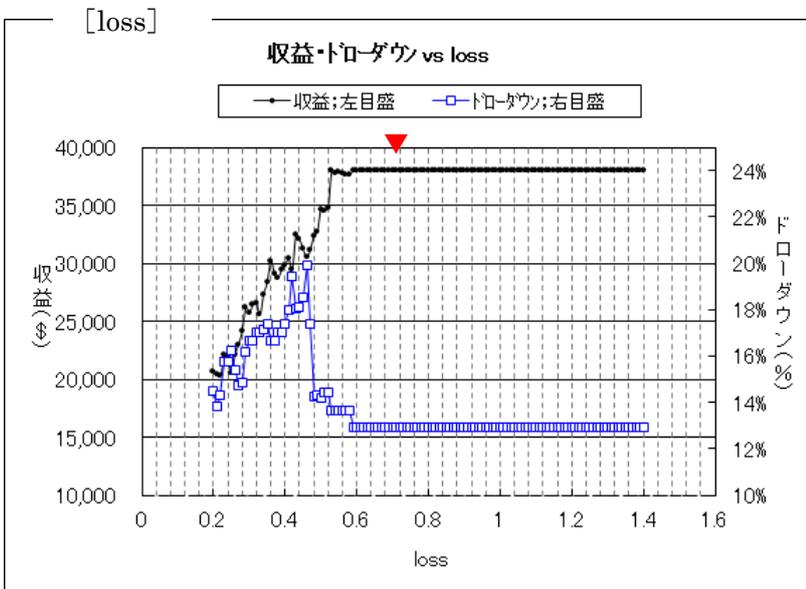
▼ = 「現状の設定値」(表 5-2) を示します

※以下、各 **exetren** 変数値を「1 ステップ」づつスキャンした結果を図示します。

なお、スキャン対象外の **extern** 変数は「表 5-2; 現状の設定値」に固定しています。

コメント；

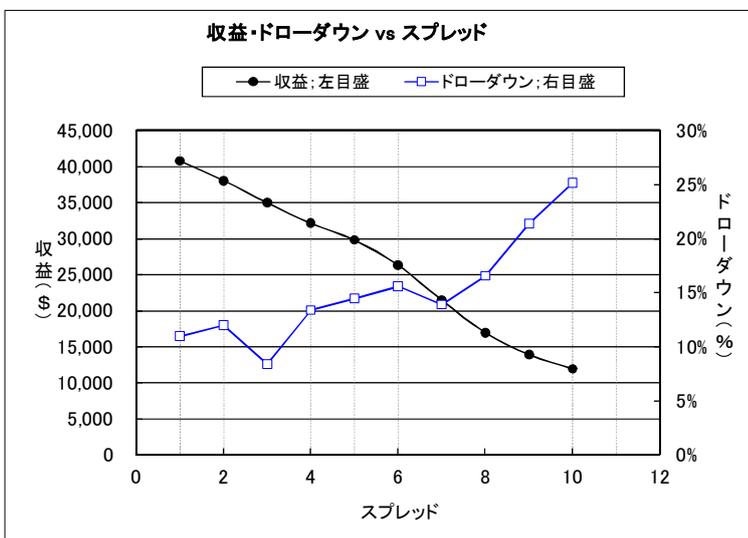
- **profit** が「1.2」以上で収益が最大かつ一定値になると言う、当初は想定していなかった意外な結果です。
(この設定値は再検討に値する)



コメント；

- **loss** が「0.6」以上で収益が最大かつ一定値になる、**profit** と同じ様な結果となった。
- 感覚的には、これも意外な結果。
(安定な場所に設定値がありました)

「収益・ドローダウン」のスプレッド依存性グラフ



- スプレッドを「1~10」の範囲で変化させたときの結果；
(含、前節 (2) のデータ)
[初期証拠金：10000、USD]
ドローダウンの単位；%

為替ペア依存性を観る；

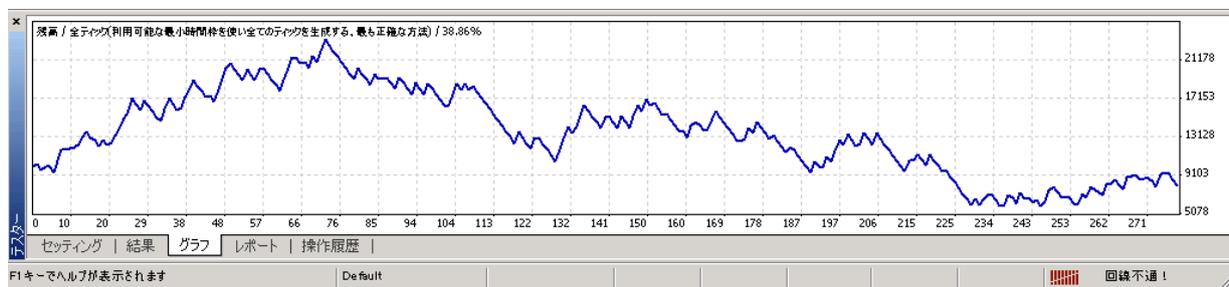
結果を一覧で観る

<資産残高 [グラフ] > 判断基準・・・「右肩上がり」であること

[USDJPY]



[EURJPY]



[AUDJPY]



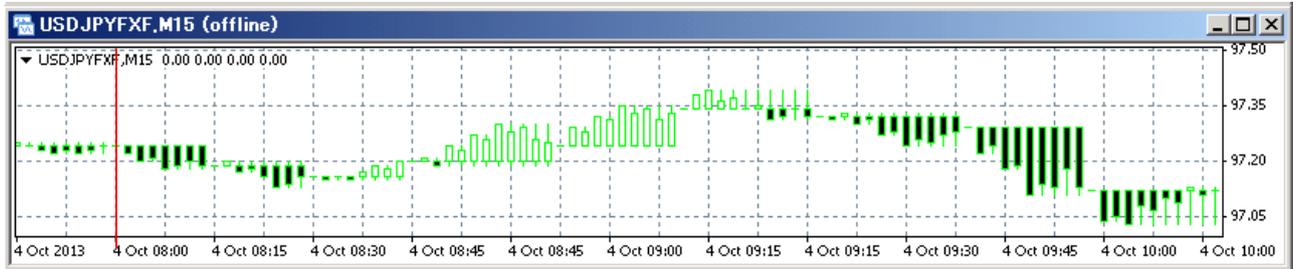
第6章 ; 参考資料1 (バックテスト関係) 一部抜粋

<概要 ; バックテストに係る、少しマイナーですが、かなり深掘した内容を解説しています>

[モデル] = 「コントロールポイント」に設定時 ; (「Empty_EA_1.mq4」バックテスト結果)

ひとつ下の時間枠 (期間) データを使い補完する方法、ティック・データ再現性は低い。

バックテストで生成されるオフライン・チャート (.FXT) ; ,M15(Control points)



バックテスト [レポート] タブ ;

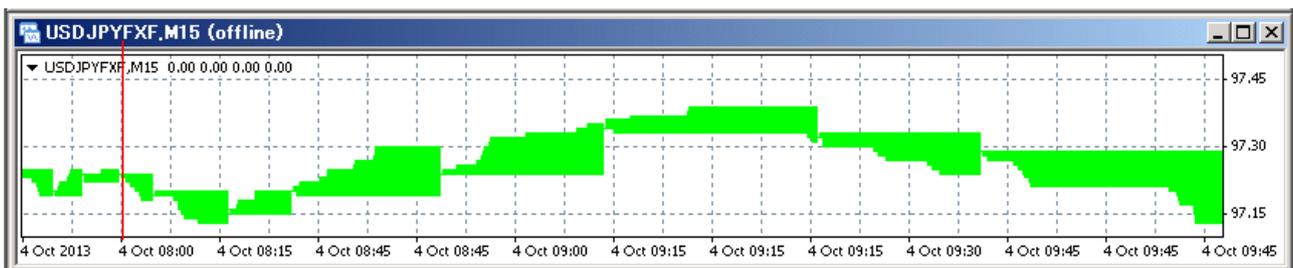
テストバー数	4361	モデルティック数	58792	モデリング品質	n/a
不整合チャートエラー	0	[Progress Bar]			
初期証拠金	10000.00			スプレッド	2
総損益	0.00	総利益	0.00	総損失	-0.00
プロフィットファクター		期待利得	0.00		
絶対ドローダウン	0.00	最大ドローダウン	0.00 (0.00%)	相対ドローダウン	0.00% (0.00)
総取引数	0	ショートポジション(勝率%)	0 (0.00%)	ロングポジション(勝率%)	0 (0.00%)
		勝率(%)	0 (0.00%)	負率(%)	0 (0.00%)
		最大 勝トレード	0.00	負トレード	-0.00
		平均 勝トレード	0.00	負トレード	-0.00
		最大 連勝(金額)	0 (0.00)	連敗(金額)	0 (-0.00)
		最大化 連勝(トレード数)	0.00 (0)	連敗(トレード数)	-0.00 (0)
		平均 連勝	0	連敗	0

[モデル] = 「全ティック」に設定時 ; (「Empty_EA_1.mq4」バックテスト結果)

利用可能な最小時間枠 (期間) を使い疑似ティックを生成・補間する方法、

ティック・データ再現性は高い。

バックテストで生成されるオフライン・チャート (.FXT) ; ,M15(Every tick)



バックテスト [レポート] タブ ;

テストバー数	4361	モデルティック数	351355	モデリング品質	49.36%
不整合チャートエラー	0	[Progress Bar]			
初期証拠金	10000.00			スプレッド	2
総損益	0.00	総利益	0.00	総損失	-0.00
プロフィットファクター		期待利得	0.00		
絶対ドローダウン	0.00	最大ドローダウン	0.00 (0.00%)	相対ドローダウン	0.00% (0.00)
総取引数	0	ショートポジション(勝率%)	0 (0.00%)	ロングポジション(勝率%)	0 (0.00%)
		勝率(%)	0 (0.00%)	負率(%)	0 (0.00%)
		最大 勝トレード	0.00	負トレード	-0.00
		平均 勝トレード	0.00	負トレード	-0.00
		最大 連勝(金額)	0 (0.00)	連敗(金額)	0 (-0.00)
		最大化 連勝(トレード数)	0.00 (0)	連敗(トレード数)	-0.00 (0)
		平均 連勝	0	連敗	0

ここを境に
色が変化
しています

「H4足」； 同一期間内（時間範囲）にM1データが無い領域で、もう一度見直す
念のため、追加でもう一つ陰線のローソク足部分も観てみました。

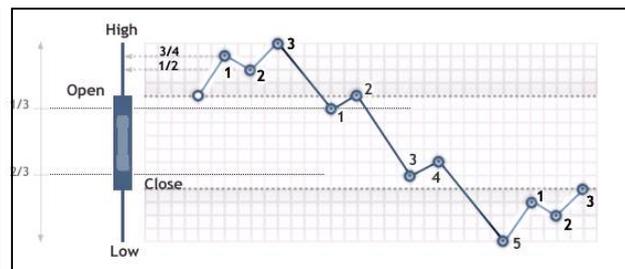
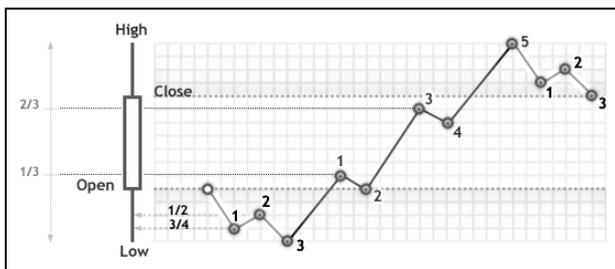
・上から、H4足.hst→H4コントロール・ポイント.fxt（H1データの無い領域）



拡大して切り出す； （筆者は、この波形に思い当たる節がありました！）



「3-5-3」モデルと比較する； 何と、ポイント数が「1個」違うだけで、あとはソックリ！



※補問法「3-5-3」モデルは、本書では解説しません。WEB上でMT4、5フォーラムか、MetaQuotes社の資料を参照ください。

第7章；参考資料2（ヒストリー・データ関係） 一部抜粋

<概要；ヒストリー・データに係る、少しマイナーですが、かなり深掘した内容を解説しています>

解説ポイント

本章の位置づけ；

※「第6章」と同様にかかなりマニアックな内容ですが、EA開発に欠かせないと筆者が勝手に考える知識を解説しています。

各節； 各節は、独立した「読み切り」構成ですので、必要な時に必要な節のみを読めば済みます、辞書代わりに活用してください。

構成； ①「ヒストリー・データ関連の知識」および
②「その他、基礎知識（関連知識）」を記載しています。

チャートのいくつかの側面

・チャートの説明は意外と厄介です、一言では表現できず、幾つかの側面から解説する必要があります。

筆者はMT4に触りだしてから、かなり長い間、頭の中の「チャートの全体像」は混乱していました。

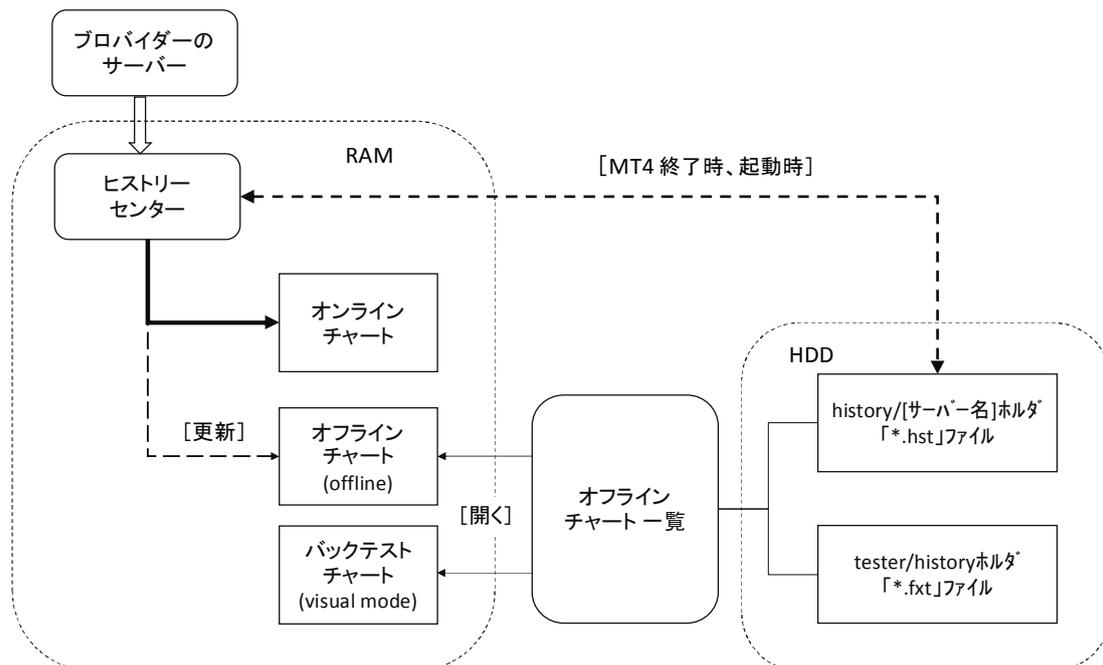
・そこで本章では、読者の頭の整理に役立つよう、「幾つかの側面から観たチャート」を念頭に解説することにしました。

※一側面からのみでは解説しきれない部分が多々あるので、同じ内容を視点を変えて、重複解説しています。

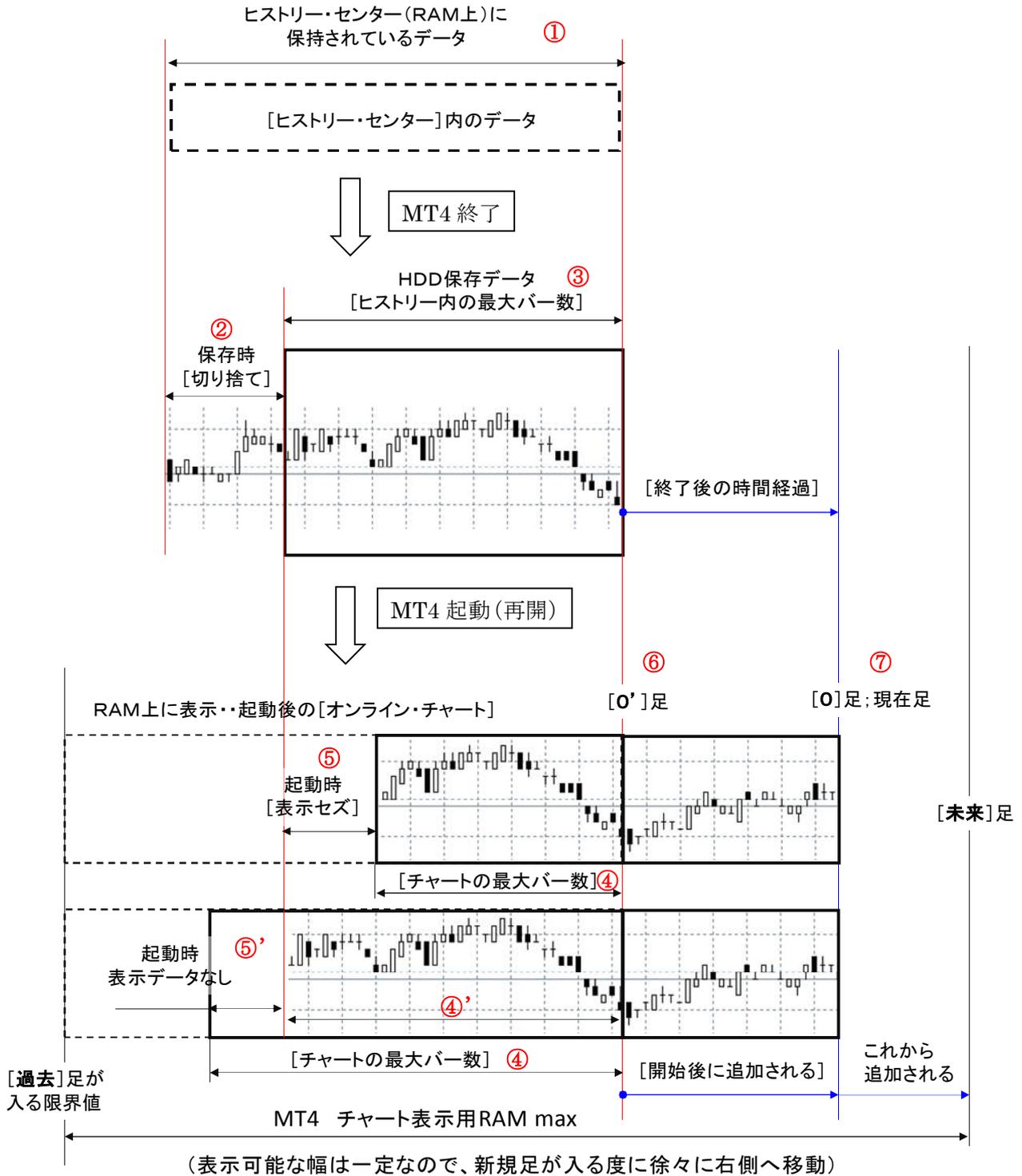
チャートの全体像について

チャートは表示の種類で分類すると、3種類に分けることができます。（以下参照）

本節では、主に「オンライン、オフライン」チャートについて解説します、3つ目の「バックテスト・チャート（.fxt）」については、第6章を参照ください。



[図7. 3] MT4「終了・開始」プロセスと、2つの「最大バー数」の関係図；



(注)

- ⑥ [0'] ; MT4 を終了した時点での最新足
- ⑦ [0] ; MT4 を開始 (再起動) した時点での最新足 (現在足)
- 「①～⑤」は、次節以降で解説に使用

第8章 ; MQL4 コード内容一覧 一部抜粋

使用（開発）全コード一覧

※「1章～6章」で、開発・使用したEA・インディケータ・スクリプトのコードを記載しています。

※なるべく、大きな文字（ポイント数）で記述したいので、1行に入りきらない部分は、

「サイズ」を小さくしています。（また、同時に書出し位置（字下げ）の変更部分があります）

※添付するコードには「エラー処理、例外処理」は入れていません。

※本コードは Build 600 以降で使われる場合（全ての読者がそうであるはずですが）、

必ず「#property strict」 **記述無し**でコンパイルしてください。

（Build 509 以前の時期に開発したコードのため、「#property strict」仕様には未対応です）

本書に登場する MQL4 のコード一覧；

初登場章	名称	使用目的・内容
第1章	Bollin_EA_08.mq4	過去（2012年）に開発したEA
第2章	Cross_Bollin_DLL.mq4	インディケータ；売買シグナル・ジェネレータ
	Filter_fu_type_DLL.mq4	インディケータ；「フ型」フィルター
	Cross_Bollin_Filtered.mq4	インディケータ；フィルター挿入後の売買シグナル・ジェネレータ
	Cross_Bollin_EA.mq4	試作EA（2013年；開発経緯の再現途中品）
第3章	Cross_Bollin_EA_LongOnly.mq4	「買いIN」専用EA
	Cross_Bollin_EA_ShortOnly.mq4	「売りIN」専用EA
	Cross_Bollin_EA_Long&Short.mq4	最適化前の「売り・買い」分離EA
第4章	Cross_Bollin_EA_fitted.mq4	最適化後のEA・・・(A)
	参考；Cross_Bollin_EA_fitted_2014.mq4	[2014年10月]データで再度(A)を「最適化」
	参考；Cross_Bollin_EA_fitted_2015.mq4	[2015年5月]データで再度(A)「最適化」
第5章	Cross_Bollin_EA_fitted_delayed.mq4	ティックのカウントでディレーを発生するEA
第6章	Empty_EA_1.mq4	空のEA、FXTファイル生成用に使用
	Tick_Print_FXT_01.mq4	FXTファイル中の補間ティック・データの確認EA

※上表中の「MQL4コード」は、全て「.mq4」形式ファイルで利用頂けます。

.....

「 Cross_Bollin_DLL.mq4 」； 売買シグナル・ジェネレータ ・・・コードの一部のみ掲載

```
//+-----+
//|                                     Cross_Bollin_DLL.mq4 |
//|                                     amenbo |
//|                                     泉の森の弁財天池 |
//+-----+
#property copyright "amenbo"
#property link      "泉の森の弁財天池"
//-----
#import "shared_memory.dll"
double set_c();
double write_c(double, int);
double read_c(int);
double close_c();
#import
//-----
```

第9章；コード解説 一部抜粋

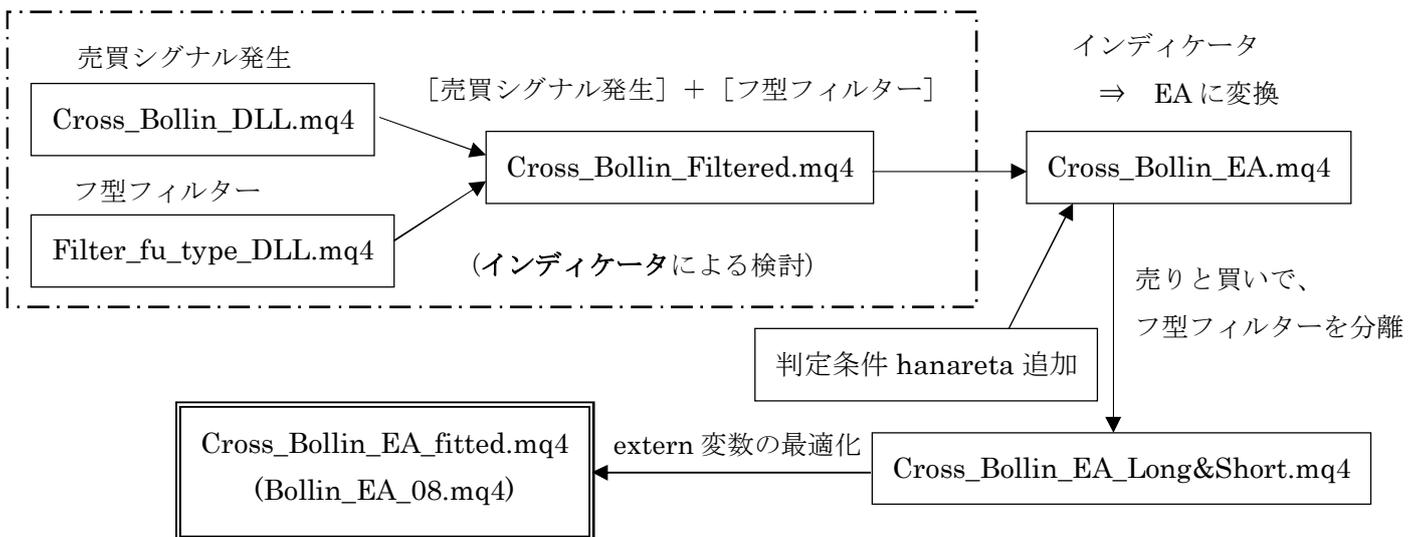
<概要；本書で使用している MQL4 コードを読者が理解できる様に、かなり詳細に解説しています>

本書に登場する MQL4 のコード一覧と解説について

初登場章	名称	解説の概要	解説
第1章	Bollin_EA_08.mq4	「Cross_Bollin_EA_fitted」との比較解説	9.2.3
第2章	Cross_Bollin_DLL.mq4	詳細な解説を行います	9.2.5
	Filter_fu_type_DLL.mq4	詳細な解説を行います	9.2.6
	Cross_Bollin_Filtered.mq4	ポイント部分のみ解説	9.2.7
	Cross_Bollin_EA.mq4	「Cross_Bollin_EA_fitted」との比較解説	9.2.3
第3章	Cross_Bollin_EA_LongOnly.mq4	特に解説はしません	
	Cross_Bollin_EA_ShortOnly.mq4	特に解説はしません	
	Cross_Bollin_EA_Long&Short.mq4	「Cross_Bollin_EA_fitted」の最適化前品	9.2.3
第4章	Cross_Bollin_EA_fitted.mq4	○詳細な解説を行います	9.2.1
第5章	Cross_Bollin_EA_fitted_delayed.mq4	「Cross_Bollin_EA_fitted」との比較解説	9.2.2
第6章	Empty_EA_1.mq4	特に解説はしません	
	Tick_Print_FXT_01.mq4	特に解説はしません	

解説する MQL4 コードの親子関係（系譜）

※「Cross_Bollin_EA_fitted.mq4（Bollin_EA_08.mq4）」開発に至るまでの、途中作成コード間の親子関係（系譜）を下図に示します。



作成 EA・インディケータ間の差異概要

	コード名称	A との相違
	Bollin_EA_08.mq4	余分な指標も定義（レンジ相場対応など拡張用）
	Cross_Bollin_EA.mq4	「フ型」フィルター値が、売りと買いで共通
	Cross_Bollin_EA_Long&Short.mq4	「extern 変数」最適化前のもの（A の原型）
A	Cross_Bollin_EA_fitted.mq4	「extern 変数」最適化後の EA
	Cross_Bollin_EA_fitted_delayed.mq4	発注・手仕舞いタイミングにディレーを発生

<コード解説例>

start();

※コード本体は、見通しを良くするために4つに分割して解説します。

ブロック構成（再掲）；

```
int start()
{
  ・（前処理）バーの数をチェック ⇒ 足数が 100 以上の場合に処理を続行、未満の場合は中止
  <ティックの判断> < 1 / 4 >
  1. ティックが、「新しい足の始まり（Open 値）」の場合の処理
  1) ポジション（建玉）が有る場合 < 2 / 4 >
    ⇒ <1> 「利確か損切」レベルに達したならば、手仕舞いする
    ⇒ <2> 「利確か損切」レベルに達していないなら、何もしない
  2) ポジション（建玉）が無い場合 < 3 / 4 >
    ⇒ <1> 「買い IN」か「売り IN」の条件が満たされていれば、ポジションを設定する
    ⇒ <2> 「買い IN」か「売り IN」の条件未達ならば、何もしない
  2. ティックが、Open 値以外の場合の処理（1 足内でのティックである場合の処理） < 4 / 4 >
  1) ポジション（建玉）が有る場合
    ⇒ <1> 「利確か損切」レベルに達したならば、手仕舞いする
    ⇒ <2> 「利確か損切」レベルに達していないなら、何もしない
  2) ポジション（建玉）が無い場合 ⇒ 何もしない
}
```

start()—< 1 / 4 > ;

```
int start()
{
  if(Bars<100 || IsTradeAllowed()==false) return(0);           ①
  // NewBar かチェック
  if(IsNewBar() && (Bars>barsTotal))                             ②
  {
    barsTotal=Bars;                                             ③
  }
}
```

機能と解説； Open 値を検出する・・・最初の1個目と、2個目以降では動作が異なります

- 1. 動作開始後に1個目の「Open 値」を検出するまで；
 - ①；チャート上の足数が「100 足未満か、トレードが許可されていない」ときは
 - ⇒ 以降のコードは実行しません
 - ②；EA スタートの init{・・・barsTotal=Bars・・・}が実行された時よりも、
 - 「足数（Bars）が増加していて、且つ新しい足（Open）のティック」が来たら、
 - ⇒ 以降のコードに進む
 - ③；先ず、現在の足数を barsTotal に再設定する
- 2. 2個目以降の「Open 値」を検出する場合の処理；
 - ①；同上（通常は常にスキップ）
 - ②；前回（直近）のティック到達時よりも「足数（Bars）が増加していて、
 - 且つ新しい足（Open）のティック」が来たら、
 - ⇒ 以降のコードに進む
 - ③；先ず、現在の足数を barsTotal に再設定する

start0-< 2 / 4 > ;

```

barsTotal=Bars;
//<ポジションが在る場合の処理>
if(OrdersTotal()>=1) ①
{
    for(int i=0;i<OrdersTotal();i++) ②
    {
        if(OrderSelect(i,SELECT_BY_POS,MODE_TRADES)==false) break; ③
        if(OrderMagicNumber()!=Magic_ID || OrderSymbol()!=Symbol()) continue; ④

        if(OrderMagicNumber()==Magic_ID && OrderSymbol()==Symbol()) ⑤
        {
            bool ikisugi=(Time[0]-in_time)>=(300*Period()*60); ⑥
            -----
            if(OrderType()==OP_BUY) ⑦
            {
                double kachi_buy_price=OrderOpenPrice()+profit; ⑧
                double make_buy_price=OrderOpenPrice()-loss;
                //
                bool kachi_buy=(Bid>=kachi_buy_price); ⑨
                bool make_buy=(Bid<=make_buy_price);
                //
                if(kachi_buy || make_buy || ikisugi) ⑩
                {
                    bool c1=OrderClose(OrderTicket(),Lots,Bid,0,Blue);
                    continue;
                }
            }
            -----
            if(OrderType()==OP_SELL) ⑪
            {
                double kachi_sell_price=OrderOpenPrice()-profit; ⑫
                double make_sell_price=OrderOpenPrice()+loss;
                //
                bool kachi_sell=(Ask<=kachi_sell_price); ⑬
                bool make_sell=(Ask>=make_sell_price);
                //
                if(kachi_sell || make_sell || ikisugi) ⑭
                {
                    bool c2=OrderClose(OrderTicket(),Lots,Ask,0,Blue);
                    continue;
                }
            }
            -----
        }
    }
}
} //end_of_if(OrdersTotal()>=1)

//---ポジションが無い場合の処理---

```

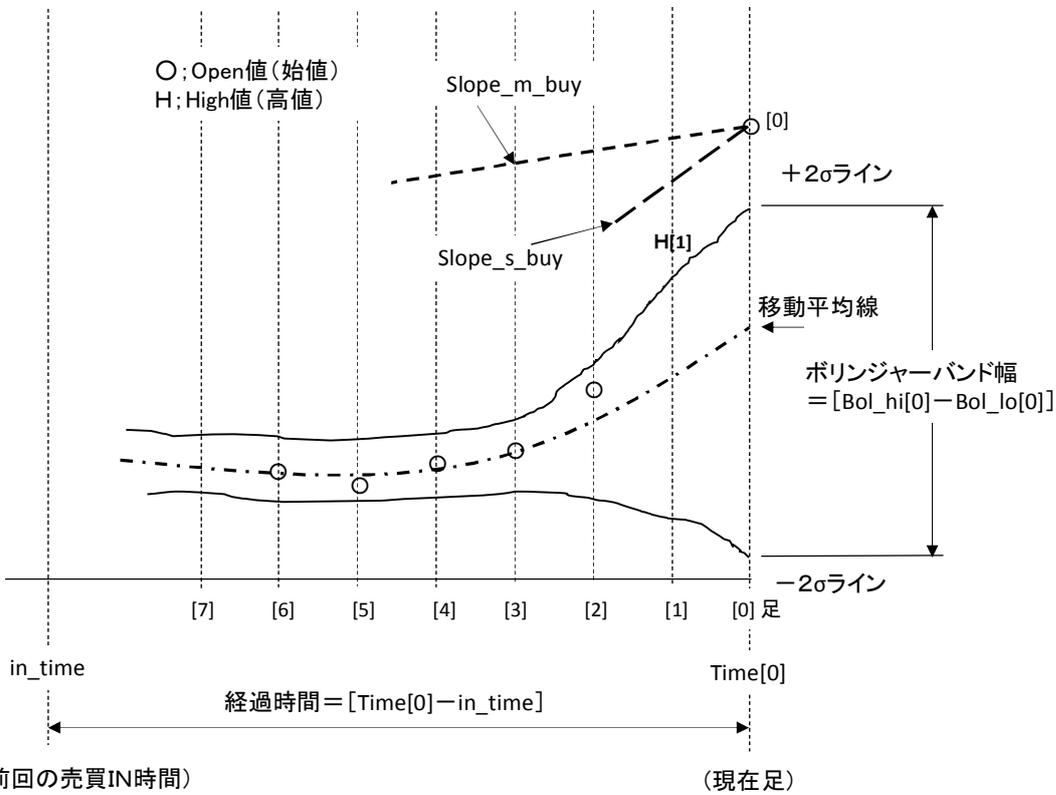
.....

.....

```

.....
.....
//買い条件は整ったか ⑤
bool cross_up_1=((High[1]>Bol_hi[1]) && (Open[0]>Bol_hi[0]));
bool cross_up_2=((Open[3]<Bol_hi[3]) && (Open[2]<Bol_hi[2]));
bool cross_up_3=((Open[4]<Bol_hi[4]) && (Open[3]<Bol_hi[3]));
bool cross_up_4=((Open[5]<Bol_hi[5]) && (Open[4]<Bol_hi[4]));
bool cross_up_5=((Open[6]<Bol_hi[6]) && (Open[5]<Bol_hi[5]));
bool cross_up=(cross_up_1 && (cross_up_2 || cross_up_3 || cross_up_4 ||
                    cross_up_5));

//条件の補足
bool cross_up_rapid=((High[2]>Bol_hi[2]) && (High[1]>Bol_hi[1]) &&
                    (Open[0]>Bol_hi[0]));
    
```



//買い条件は整ったか

⑤ ; 7足 ([6]~[0]) の「+2σ、Open値、High[1]値」間の位置関係(配置)から、
買い条件が整ったかを判定する (<買いIN判定の概念図>を参照)
条件成立 ⇒ 「cross_up=true」

.....

以上