



本格的なExcelマクロの差分分析ツールです！
Excel棚卸【単独 マクロ差分】ツール
取り扱い説明書

2012. 7. 9

Ver 1.084

Excel 工房 Yuzu

目次

1. [はじめ](#)に
 - 1.1 [本システムの環境条件](#)
 - 1.2 [ユーザライセンスキー](#)
 - 1.3 [仕様の変更](#)
2. [概要と特徴](#)
 - 3.1 [動作環境とインストールアカウント](#)
 - 3.2 [分析対象](#)
 - 3.3 [インストール媒体](#)
 - 3.4 [インストールとアンインストール作業](#)
4. [単独 マクロ差分分析の実行](#)
 - 4.1 [制約条件](#)
 - 4.2 [分析の実行](#)
 - 4.3 [差分分析機能の詳細と効果](#)
 - 4.3.1 [基本機能](#)
 - 4.3.2 [差分分析条件のチューニングと効果](#)
 - 4.3.3 [表示制御](#)
 - 4.4 [印刷設定](#)
5. [お問い合わせ](#)

1. はじめに

本Excel棚卸【**単独** マクロ差分】ツール(以下、本ツール)は、新旧のExcelブックのVBAコードの差分の分析を行うものです。

プログラムコードの新旧の差分分析ツールは、開発者にとって生産性の向上や品質確保の観点からなくてはならないものです。しかし、Excel VBA(Visual Basic For Application)には、発売以来20年も経ちますが、これといった差分分析ツールはありませんでした。

今や、仕事にパソコンを使用している人で表計算ソフトを使っていない人はほとんどないといっても過言ではありません。その代表格がマイクロソフト社の Excel で、既に初期版の Excel 3.1 の発売以来20年以上も経過しています。

Excel を使えば、名簿の作成、簡単な表計算、集計データのグラフ表示などお手のものです。表計算にしても使い始めの頃は簡単な四則演算ですが、使いなれるに連れ徐々に Excel 関数等を利用して高度なものができるようになります。しかし、間違っ隣セルの計算式を壊したりすることも表計算機能だけでは自動化や品質の確保において限界があります。また、シートのセルをコピーすれば、シート保護が設定されていても、簡単に改造を加えることが可能でセキュリティ面で心配です。

この限界を超えられるのが VBA (Visual Basic For Application)です。VBA は一般のプログラミング言語とは異なり、Excel の言葉で処理を記述できる強力なプログラミング環境です。しかも Visual でインタプリタ言語ということもあり動作確認も簡単で大変に便利なものです。また、Excel ワークシート上での操作をそのままマクロとして自動記録機能があることや、解説本や Web 上でのサポートなども充実しており、初心者でも簡単に始められるといった特徴があります。

VBA を使った開発のスタイルに特徴があります。おぼろげな仕様状態からスタートし、動作を確認しながら徐々に機能を膨らませていきます。このような開発スタイルをスパイラル(渦巻き状に大きくしていく)開発方法と呼びます。この場合、追加・修正が頻繁に発生します。このため、昨日まで動作していたロジックがある日突然に動かなくなったということに度々遭遇します。何らかの修正や機能追加が影響を与えていることは誰でも解りますが、適当なツールなしではその修正箇所を確認する作業に多くの時間を使うことになります。

変更箇所を簡単に検索・表示したいのですが、残念ながら、VBA ではこの可視化が簡単ではありませんでした。およそ発売以来、このようなツールにはお目にかかっていません。C 言語のような本格的な開発環境に比べ、初心者の多い Excel VBA においてはその脆弱さが目立ちます。

私自身も Excel の VBA を使用している立場でもあり、我慢は限界ということで、重い腰を上げ、マクロ差分分析ツール(商品名: Excel 棚卸【**単独** マクロ差分】ツール)の開発に着手しました。そして、何とか使えるレベルのものが完成しました。是非、この機会をご利用いただければ幸いです。

また、部門に配布された Excel ブックの VBA コードが原本と同一か否かを一括して分析するツール(商品名:

Excel 棚卸【一括 マクロ差分】ツールも準備しましたので、併せてご利用をいただきたく。

《本ツールの設計思想》

本ツールの最大の特徴は、開発ステージに沿った差分分析機能を提供していることです。

《初期の開発ステージでは、一般論ですが》

- まず最終形態を想像しながらコアとなる部品相当の機能の実現と、単体部品としての動作確認が行われます。徐々に複数の部品の結合などへと進みます。この段階では、壊し(放棄)や追加・修正が頻繁に発生します。この為、デグレーション(変更等によって、動作していた機能が動作しなくなった)などが発生します。この為、一気に組み立てるのではなく徐々に組み立てていくことが重要となります。
- 機能の追加に伴って、繋ぎとなるプロセスの追加や、プロセスの引数やモジュールの共通宣言分(主にPublic)の追加・変更が発生します。同種の宣言分は一行に記述しておくことが行われます。この場合、差分のカテゴリとして、旧定義を「削除」して新たに定義「追加」したとみるのか、旧仕様を継承した「変更」とするのかといった二つの見方があります。本ツールでは、後者の立場を採用しています。
- また、プロセスの記述場所や共通宣言文などは、発生の都度、あまり記述位置は意識せずに行われます。

従って、このステージで差分分析に求められる機能要件としては

デグレーションの原因となる、変更・修正箇所を素早く可視化することが求められます。特にTab、やコメントや空行の影響を排除して、ロジックの変更部だけにフォーカスした分析ができれば便利でしょう。また、同時にコード部の文字列の相違も分ることが望まれます。

《開発の中期ステージでは、ほぼ全体機能として動作する段階となると》

- 当初はExcelのコンテキストで動作させていたものを、最終形態として一部のロジックのアドイン化やDLL化が行われることになります。
- このようなステージでも、機能分担の見直しや、作品のブラシアップが頻繁に行われます。このステージからはバグ・改善に伴う変更管理表を使用することも重要です。変更理由や変更箇所を記載しておくことが、差分ツールとともにデグレーションからの脱出に大いに役立つものです。

従って、このステージで差分分析に求められる機能要件としては

アドインの拡張子としての".xla" やVB(VisualBasic) 6.0のコードの拡張子".bas"や".cls"が差分分析において区別無く利用できることが求められます。本連携ができないと、その都度、".xls"の形式への変更が余儀なくされます。

《開発の最終ステージ》

- ExcelのVBA開発に限ったことではありませんが、最近ではスパイラル型開発が主流になりつつあります。スパイラル

型開発では、プロトタイプから初めて製品へと徐々にレベルアップしていきます。従って、ウォーターフォール型開発と違って、仕様も同時並行的に決まっていきます。従って、仕様書は試行錯誤の末に最終ステージにできあがる感じです。このため、コードの体裁や、保守に向けたリーダビリティは後付けになっています。しかし、保守ができるコードにしておくことは、仕様書に基づく最終テストともに極めて重要な作業です。

従って、このステージで差分分析に求められる機能要件としては

コメントの追加や、空行の挿入、長いコードの折り返し、プロセスの記述順、共通宣言文の整理などが行われます。これは、基本的にロジックには直接影響をあたえませんが、編集ミス等によりコードを壊してしまうこともあります。このような手入れの後でも、手入れ前のコードとロジック部分に関して差分がないことを確認できれば、変更ミスによるデグレードのリスクを回避できます。同時にコードとコメントの区別なども必要な要件となります。

本Excel棚卸【単独 マクロ差分】ツールは、このような設計思想に基づいて開発したもので、十分に使っていただけるものと確信しています。しかし、差分分析のロジックにはアナログ的な要素を含むところも多くあります。この為、差分の位置など判断が分かれるケースもあるかと思われます。

お断り:

1. 本ツールにおいては、テキスト差分処理エンジンとしてNonSoftの[NonDiffVb6.dll](http://homepage2.nifty.com/nonnon/Download/TextDiffDsp/index.html)を使用しております。また、暗号化処理の[CryptMin.dll](http://homepage2.nifty.com/nonnon/Download/CryptMin/index.html)を使用しております。何れのDllもNonSoftの著作物であり、再配布の許諾を得ております。

<http://homepage2.nifty.com/nonnon/Download/TextDiffDsp/index.html>

<http://homepage2.nifty.com/nonnon/Download/CryptMin/index.html>

2. また本ツールにおいては、VBA保護操作のライブラリとして[K窓](#)のkcommand.dllを使用しております。

1.1 本システムの環境条件

本Excel棚卸【単独 マクロ差分】ツール(以下、本ツール)は、以下を条件としています

- (1).適切なクライアントの利用環境であること(OS、Microsoft Officeの版数、CPUの性能)
- 、
- (2).適切なユーザライセンスを所有していること

1.2 ユーザライセンスキー

(1).正規版ライセンス

ユーザライセンスは、利用者PCハードウェア数のライセンスが必要となります。

本ツールのインストール時に、ライセンスキーがチェックされます。

本ライセンスキーは[サポート](#)を受ける場合のユーザ識別子となります。

(2).評価版ライセンス

評価版のライセンスコードは、“1”を入力することにより、一定の期間(30日)使用することが可能です。

機能的な制約はありませんが、仮想OS環境は制限条件としています。

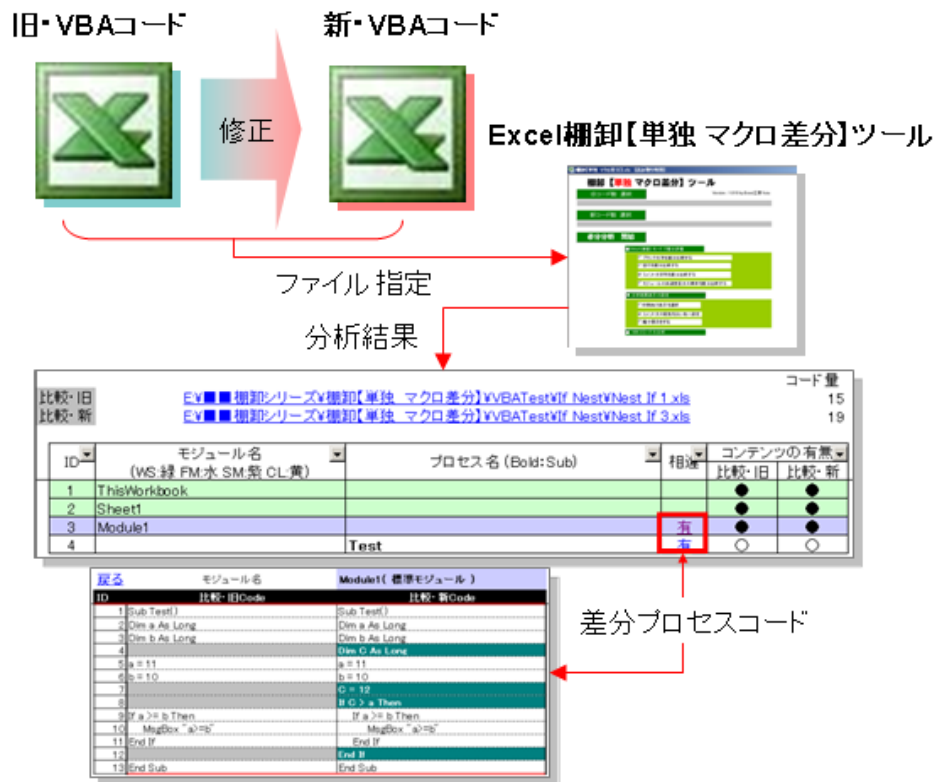
評価版のライセンスではサポートを受けることができません。

1.3 仕様の変更

本ツールは改善もしくは品質上の問題から、断りなく仕様を変更する場合があります。

2.概要と特徴

本ツールは下図に示すように、新・旧のVBAコードの差分を分析して、分析結果をレポートします。



本ツールの主な特徴を以下に示す。

- 開発ステージにあった差分分析機能を提供します:はじめに参照
- モジュール、プロセス単位の差分が取得で、差分詳細までドリルダウンができます。
- 分析条件の設定で、変更部分にフォーカスした差分分析が可能です

- ・共通宣言分の並び替え、プロセスの順序の整理などの影響を排除できます
- ・引数の追加などは、削除・追加ではなく変更とすることが可能です

Sub Test1(a As String) '当初、引数はひとつ | Sub Test1(wb As Workbook, a As String) '引数を追加

- ・コード部分とコメント部分の識別ができます。

Exit Function 'Endを単独で用いると、この段階でプロシージャが終了します

- 削除・追加・変更の表示が可能です。変更時は文字列の差分も表示されます

変更:薄橙色

削除:黄色

追加:青緑

変更無:白

If InStr(Dcl_Key, "Debugmode") > 0 Then

If InStr(Dcl_Key, "Public b") > 0 Then

- アドインおよび、VB 6.0の差分分析も可能です

3. インストールとアンインストール

3.1 動作環境とインストールアカウント

OS \ Excel	Excel 版数					インストールアカウント	
	2000	2002	2003	2007	2010	Admin ※4	Admin権限
Windows 2000	●	●	●	—	—	●	●
Windows XP	●	●	●	× ※2	● ※3	●	●
Windows Vista	●	●	●	× ※2	● ※3	●	
Windows 7	●	●	●	× ※2	● ※3	●	
備考	※2: VBAの互換性に問題 ※3: 97-2003互換モードで実行					※4: 「Administrator」のアカウント	

注意1: Windows Vista、Windows 7の環境では、Administorアカウント※でインストールしてください。

Windows Vista,7にてAdministorアカウントを有効にする方法は、以下の手順で実行してください。

「コントロールパネル」－「システムとセキュリティ」－「管理ツール」－「コンピュータの管理」と開き、
「システムツール」－「ローカルユーザーとグループ」－「ユーザー」とたどっていき、
「Administrator」ユーザをダブルクリックし、そして、「アカウントを無効にする」のチェックをはずす。

注意2: Excel版数

- Excel2007はVBAの互換上の問題でサポートしていません。
- Excel2010の場合は、Excel97-2003互換モードでご利用ください

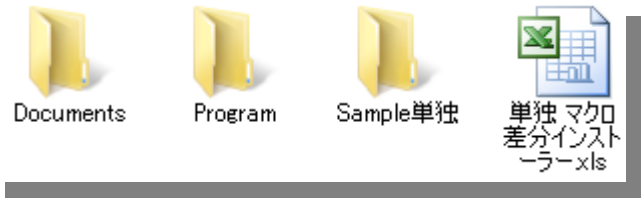
3.2 分析対象

	拡張子	備考
Excel VBA	.xls, .xla (Excel 2000-2003) .xlsm, .xlam (Excel 2010)	・プロテクトは解除済みであること
VB 6.0	.bas, .cls, .frm	・VB 6.0 フォームUIには対応できていません ・Exportした特定のExcel VBAのモジュール

3.3 インストール媒体

解凍されたフォルダとコンテンツ

ダウンロード後のファイルを解凍すると、以下のコンテンツがあります。ローカル環境に保存してください。



	解凍後のコンテンツ	摘要
1	単独 マクロ差分インストーラ	本ツールの専用インストーラです
2	Program フォルダ	<ul style="list-style-type: none">本ツール自信DLL 群: CryptMin、NonDiffVb6kcommand、kcommand YuzuCommand の4dll
2	Documents フォルダ	<ul style="list-style-type: none">Excel 棚卸【単独 マクロ差分】クイック評価マニュアルExcel 棚卸【単独 マクロ差分】ツール取扱説明書
3	Sample 単独フォルダ	クイック評価のためのサンプル Excel ブックです

3.4 インストールとアンインストール作業

以下の手順に沿ってインストールしてください。

3.4.1 Excel セキュリティレベルの変更

本品は Excel の VBA(マクロ)で動作するため、使用に当たってはマクロのセキュリティレベルの変更が必要です

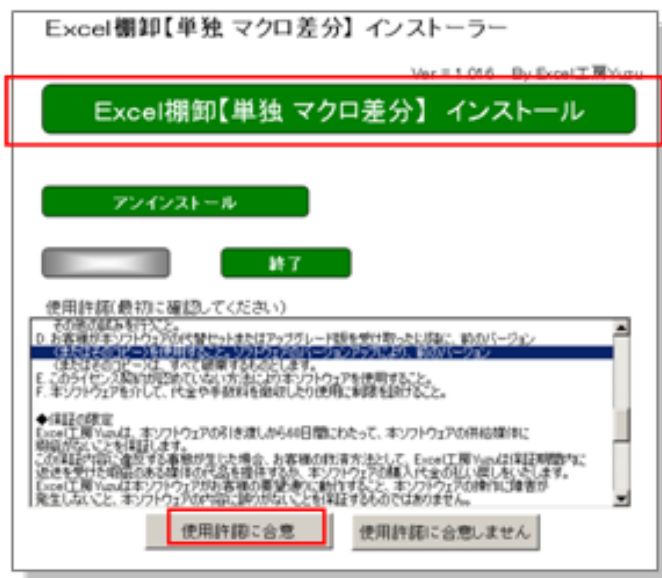
Excel版数	Excel 2000	Excel 2002,2003	Excel 2010
起動	[ツール]→[マクロ]→[セキュリティ]で設定後に再起動		[Excelオプション]→[セキュリティセンター]で設定後に再起動
設定内容			

3.4.2 ツールのインストール

① インストーラの起動

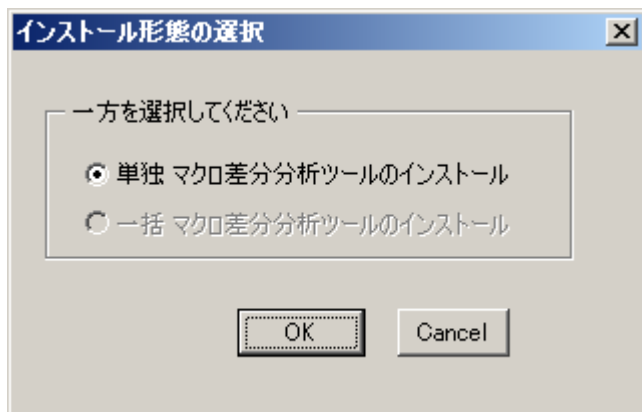
注: Windows Vista、Windows 7 の環境では、Administrator アカウント※でインストールしてください。
 なお、Windows 2000、Windows XP では Administrator 権限を持つユーザアカウントで行ってください。

② 使用許諾とインストールの開始

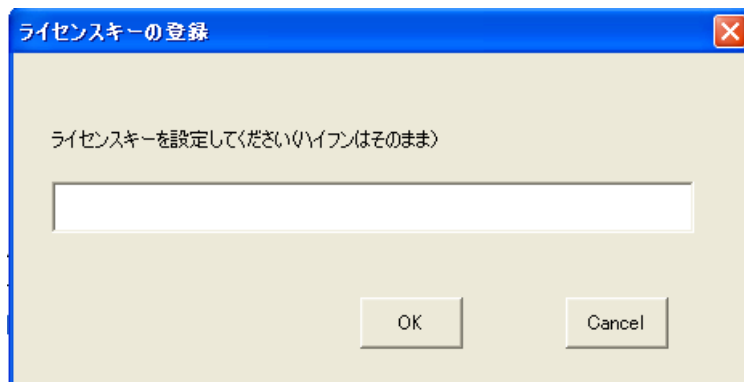


注意: 最初に使用許諾に同意が必要です

③ 単独 マクロ差分分析ツールのインストールを選択



④ライセンスキーの入力



・正規版のライセンスキーはそのまま貼り付けてください

・不正なライセンスキーの場合、エラーとなります。

評価版のライセンスキーは、“1”です。

30 日の評価が可能となります。ただし、評価版は仮想 OS 環境では利用できません。

⑤分析結果等の保存場所の指定



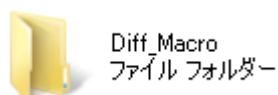
: 未登録の場合は、設定します。

※ ログは追記されますので、必要に応じてクリアしてください。

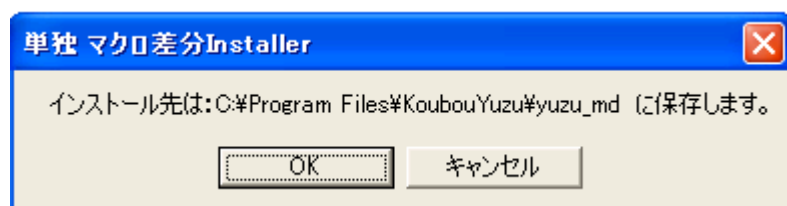
※ Windows Vista,7でインストール後にユーザアカウントで利用する場合、同様な設定画面が表示されます。

本フォルダ下には、分析結果を保存する『Diff_Macro』というフォルダが作成さ

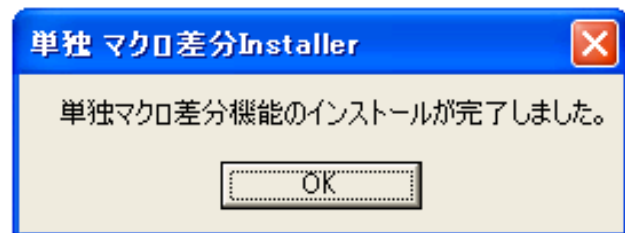
れます



⑥インストール先のフォルダパスの確認



⑦インストールの完了



注意: インストールの失敗のケース

- ・ 正しくないライセンスキーの入力
- ・ インストール先のアクセス権が無い
- ・ 評価期間の満了
- ・ インストールの中止

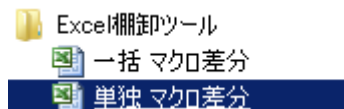
インストーラを終了してください。

⑧スタートメニューの確認

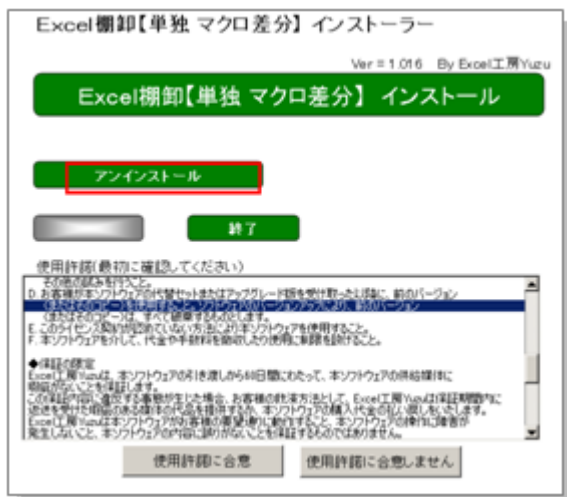
《Windows XP の場合のスタートメニュー》



《Windows 7 の場合のスタートメニュー》



3.4.3 ツールのアンインストール



注意:Windows Vista,7 にてご利用の場合は、Administrator アカウントでしてください。Administrator アカウントを表示する方法については、本マニュアルの末尾の参考を参照して下さい。

4. 単独 マクロ差分分析の実行

4.1 制約条件

本ツールで分析を行う場合の諸条件を以下に示す。

	制限項目	説明
1	分析対象のパスワード保護の解除	Read/Write保護、およびVBA保護は事前に解除済みであること。また共有保護はVBAコードへのアクセスエラーとなりますので、未設定のこと
2	プリンタの未搭載環境	分析結果レポートの印刷設定は行っておりません
3	最大分析コード数	最大10万行
4	最大モジュール数	最大250モジュール
5	最大プロセス数	最大1万プロセス
6	最大差分コード量／モジュール	最大1万行
7	シンタックスエラー	構文エラーは除去済みであること。誤動作の原因となります。

4.2 分析の実行

① ツールの起動と分析対象の指定

スタートメニューから<Excel 棚卸ツール>・<単独 マクロ分析>を起動し、<新・旧コード側 選択>ボタンから分析対象を選択します。

Excel棚卸【単独 マクロ差分】ツール
Version : 1.060 by Excel工房Yuzu

旧コード側 選択
E¥■■■棚卸シリーズ¥棚卸【単独 マクロ差分】¥VBATest¥BetweenProcessComment¥Old.xls

新コード側 選択
E¥■■■棚卸シリーズ¥棚卸【単独 マクロ差分】¥VBATest¥BetweenProcessComment¥New.xls

差分分析 開始

■ Strict(厳密)モードで差分評価

☐ ブランク文字を差分比較する
☒ 空行を差分比較する
☒ コメント文字列を差分比較する
☒ プロセス及び共通宣言文の順序を差分比較する

■ 分析結果表示の設定

☐ 印刷向け表示を選択
☒ コメント文の配色を淡い色へ設定
☒ 縮小表示をする

■ VB6.0またはExportされたVBAコードの分析

●旧側のコードで、差分詳細においては左側に表示されます

●新側のコードで、差分詳細においては右左側に表示されます

●分析の開始を指定

●差分ロジックをチューニングします

●分析結果の表示方法をチューニングします

●VB6.0等のコードを分析指定

《Excel またはアドインの分析の場合》

単独 マクロ差分分析

旧コード側のExcelまたはアドインのファイルまたはは選択してください。

OK キャンセル

注意: 同じパスを指定できません。

② 分析条件の設定

本設定は、明らかに差分があると想定される場合に設定する情報です。

☐ Strict(厳密)モードで差分評価

- ☐ ブランク文字を差分比較する

☒ 空行を差分比較する

☒ コメント文字列を差分比較する

☒ プロセス及び共通宣言文の順序を差分比較する

《分析条件の設定の機能説明》

条件選択	説明
Strict(厳密)モードで差分評価	本選択を行うと、以下の選択が全て有効(オン)になります。厳密な差分をみることができます
ブランク文字を差分比較する	コードを読みやすくするためTAB等によるインデントをおこないますが、オフにすると、インデントの影響を除外することで、実行コードだけの差分分析を可能とします。特に機能追加によるIf文のネストが深くなるときに便利です。
空行を差分比較する	コードを読みやすくするため、空行をいれますが、オフにすると、空行の影響を除外することで、実行コードだけの差分分析を可能とします。
コメント文字列を差分比較する	コメントもコードの保守のためには必須です。コードの完成に近づくにつれ、コメントも強化されます。オフにすると、コメントの影響を除外することで、実行コードだけの差分分析を可能とします。デフォルト値はオンです。
プロセス及び共通宣言文の順序を差分比較する	プロセス及び共通宣言文は順序性がないため、開発途上においては、その都度適当な場所に追加しますが、コードの完成に近づくにつれ記述順を再整理をおこないます。オフにすると、記述場所の違いを差分と認識しません。

③ 分析結果表示の設定




。

本設定は表示方法に関するものです。個別に設定することもかかのですが、本ツールではデフォルト設定を推奨します。

■ 分析結果表示の設定

- ☐ 印刷向け表示を選択
- ☒ コメント文の配色を淡い色へ設定
- ☒ 縮小表示をする

《分析結果表示の分析条件の設定の機能説明》

条件選択	説明
印刷向け表示を選択	 <p>オフの状態では、差分のある箇所の背景色が設定され、差分箇所を見つけ易くなります。オンの状態では、背景色は抑止され、トナーの消費を抑制できます。</p>
コメント文の配色を淡い色へ設定	 <p>のように、選択するとコメント部分は淡い青になり、コメントとコードを明確に区別できます</p>
縮小表示をする	 <p>のように、差分がなかった場合には、縮小表示されます。差分があった場合には縮小は行われません。</p>

④ VB 6.0 または Export した VBA モジュールの分析

以下の設定にチェックを入れて下さい

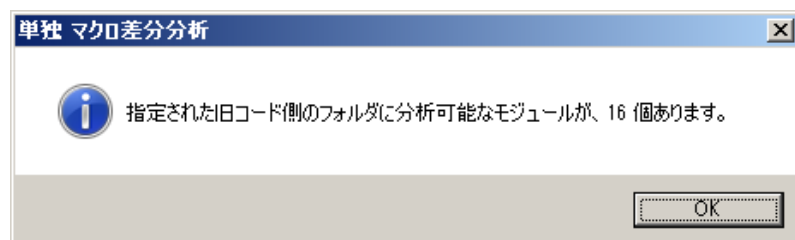
☒ VB6.0 または Export された VBA コード の 分析

すると、新・旧コード 選択において以下の確認メッセージが出力されます。
同じパスを指定できません。



●ファイルではなくフォルダであることに注意して下さい。
続いて、

≪分析可能なファイルが存在した場合≫



- 当該フォルダ内に分析可能なモジュール数の確認メッセージが出力されます。

≪分析可能なファイルが存在しない場合≫

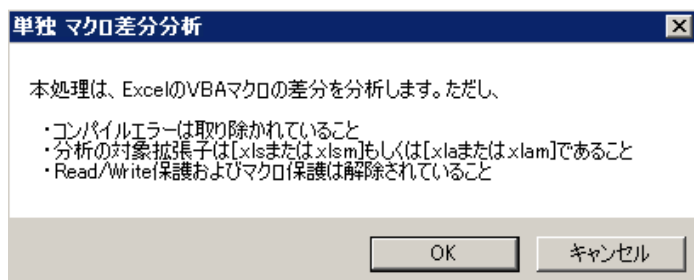


⑤ 分析の開始

ツール初期画面の<差分分析 開始>ボタンを押下します。

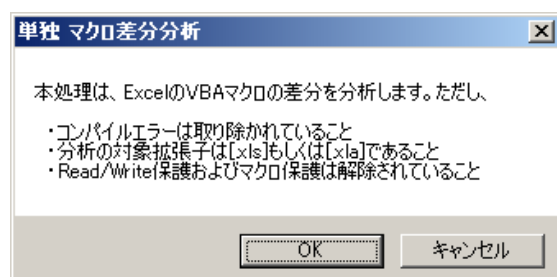
以下の初期起動画面が表示されます。

≪Excel 2010の場合≫



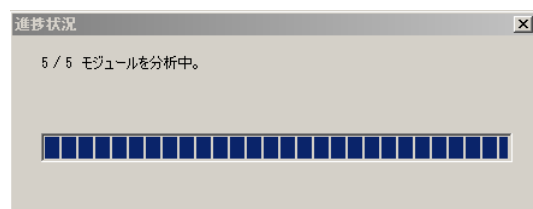
本注意事項は既に本マニュアルで解説済みの事項です。

≪Excel 2000～2003 の場合≫



⑥分析状態

≪進捗状況≫



⑦ 分析結果

《差分があった場合》



インストール時に指定した分析結果等の保存フォルダ下にレポートは保存されます
[Diff_VBA_timestamp.xls]名です。



《差分がなかった場合》



《分析結果レポート 差分有りの場合》

比較・旧 比較・新	E:\■■■棚卸シリーズ\棚卸【単独 マクロ差分】\VBATest\Declare文\Old.xls				コード量
	E:\■■■棚卸シリーズ\棚卸【単独 マクロ差分】\VBATest\Declare文\New.xls				18
					19
ID	モジュール名	プロセス名 (Bold:Sub)	相違	コンテンツの有無	
5	Module1		有	●	●

レポートは、オートフィルタで相違＝「有」のモジュール、プロセスのみを表示する仕様としています。全体のモジュール、プロセスを確認するにはオートフィルタで相違＝「すべて」を選択して下さい。

比較・旧 比較・新	E:\■■■棚卸シリーズ\棚卸【単独 マクロ差分】\VBATest\Declare文\Old.xls				コード量
	E:\■■■棚卸シリーズ\棚卸【単独 マクロ差分】\VBATest\Declare文\New.xls				18
					19
ID	モジュール名	プロセス名 (Bold:Sub)	相違	コンテンツの有無	
5	Module1		昇順で並べ替え 降順で並べ替え (すべて) (トップテン) (オプション) 有	●	●

比較・旧	E:\■■■棚卸シリーズ\棚卸【単独 マクロ差分】\VBATest\Declare文\Old.xls	コード量	18
比較・新	E:\■■■棚卸シリーズ\棚卸【単独 マクロ差分】\VBATest\Declare文\New.xls		19

ID	モジュール名 (WS:緑 FM:水 SM:紫 CL:黄)	プロセス名 (Bold:Sub)	相違	コンテンツの有無	
				比較・旧	比較・新
1	ThisWorkbook			●	●
2	Sheet1			●	●
3	Sheet2			●	●
4	Sheet3			●	●
5	Module1		有	●	●
6		test		○	○

《モジュール、プロセスに関する差分情報レポートの見方》

比較・旧	E:\■■■棚卸シリーズ\棚卸【単独 マクロ差分】\VBATest\Declare文\Old.xls	コード量	18
比較・新	E:\■■■棚卸シリーズ\棚卸【単独 マクロ差分】\VBATest\Declare文\New.xls		19

ID	モジュール名 (WS:緑 FM:水 SM:紫 CL:黄)	プロセス名 (Bold:Sub)	相違	コンテンツの有無	
				比較・旧	比較・新
1	ThisWorkbook			●	●
2	Sheet1			●	●
3	Sheet2			●	●
4	Sheet3			●	●
5	Module1		有	●	●
6		test		○	○

分析対象へのリンク

コード量

オートフィルタ

●モジュール
○プロセス

モジュール名
(色でタイプを区分)

太字:Sub
細字:Function
無:共通宣言コード

差分の有無と詳細へのリンク

注:本例では“Module1”の共通宣言コード部に差分があることを示しています。

《コードに関する差分情報レポートの見方》

戻る	モジュール名	Module1(標準モジュール)
	比較・旧Code	比較・新Code
1	Option Explicit	Option Explicit
2	無造作に宣言文を配置	宣言文を整理しました
3	Public y As Variant 'Old	Public y As Variant, yy As String 'add yy
4	Public a As Variant '	Public a As Variant '
5	Public b As Variant 'Old	Public b As Variant, b1 As Variant 'New
6	Public c As Variant, c1 As String	Public c As Variant
7		Public d As Variant 'Add
8	Public Const x = "xxx"	Public Const x = "xxx" 'New

戻りボタン

親モジュール名と(タイプ)

コード位置

青/黄:コメント
赤:文字列の差分
黒/白:コード

差分区分表示: 変更:薄橙色 削除:黄色 追加:青緑 変更無:白

4.3 差分分析機能の詳細と効果

4.3.1 基本機能

(1) 継続行の扱い

本ツールにおいては、継続行(”_”)は全て単一の行として扱われます。従って、差分が分かり易くなります。

。
<<継続行を使った例 : 新側>>

```
Option Explicit
Dim aaaa As String, bbbb As String, cccc As String, dddd As String, _
    eeee As String, fff As String

Sub Test()

    If (aaaa = "xxxxx" And bbbb = "yyyy" And _
        cccc = "zzzzz" And dddd = "wwwww" And eeee = "vvvv" _
        And fff = "vvv") = True Then
    End If

End Sub
```

<<差分結果>>

ID	比較・旧Code	比較・新Code
1	Option Explicit	Option Explicit
2	Dim aaaa As String, bbbb As String, cccc As String, dddd As String	Dim aaaa As String, bbbb As String, cccc As String, dddd As String, eeee As String, fff As String
3	Sub Test()	Sub Test()
4		
5		
6	If (aaaa = "xxxxx" And bbbb = "yyyy" And cccc = "zzzzz" And dddd = "wwwww") = True Then	If (aaaa = "xxxxx" And bbbb = "yyyy" And cccc = "zzzzz" And dddd = "wwwww" And eeee = "vvvv" And fff = "vvv") = True Then
7	End If	End If
8		
9	End Sub	End Sub

・3行にまたがった「変更」表示とするよりずっと見易くなります。更に、「縮小表示」を併用すると、差分が無い場合に限り、縮小表示されます。縮小表示された内容の確認は、「式」の表示欄にて行って下さい。

(2) プロセス間のコメント行の扱い

プロセスの開始説明のために、プロセス間にコメント行や空行が配置されます。
本ツールでは、以下のルールにて当該プロセスの付属コメントとして差分処理しています。

```
Option Explicit
'Module1の宣言文
Dim c As Variant 'c:
Public b As Long
Public a As String
'X1の説明

Sub X1(x As String, wb As Workbook) '新しいX1仕様
Dim xp1 As String

End Sub

'X2の説明
Function X2() 'X2///
Dim Xp2 As String

End Function

'X3の説明
Sub X3() 'X3

End Sub
'tail comment
```

共通宣言文の範囲

Sub X1の範囲

Fun X2の範囲

Sub X3の範囲

- 共通宣言文コード以下のコメントは、継続するプロセスに帰属させています。
- プロセス間のコメントは継続するプロセスのコメントに帰属させています
- 末尾のコメントは最後のプロセスのコメントに帰属させています

《コメントの扱い》

ID	比較・旧Code	比較・新Code
1	Option Explicit	Option Explicit
2	'Module1宣言	'Module1の宣言文
3		
4	Dim c As Variant 'cの用途	Dim c As Variant 'c:
6	Public b As Long	Public b As Long
7	Public a As String	Public a As String
7	'X1の説明	'X1の説明
8	Sub X1(x As String, wb As Workbook) '新しいX1	Sub X1(x As String, wb As Workbook) '新しいX1仕様
9	Dim xp1 As String	Dim xp1 As String
10	End Sub	End Sub
11	'X2の説明	'X2の説明
12	Function X2() 'X2	Function X2() 'X2///
13	Dim Xp2 As String	Dim Xp2 As String
14	End Function	End Function
15	'X3の説明	'X3の説明
16	Sub X3() 'X3	Sub X3() 'X3
17		
18	End Sub	End Sub
19		'tail comment

赤の二重線は共通宣言文およびプロセスのセパレータを示しています

(3) プロセスの引数の変更に関して

プロセスの機能が変更する場合、内部ロジックの修正とともに、引数の追加や変更が行われる場合があります。本ツールでは、同一モジュール内であれば、同一名のプロセスは引数に変更になっても、「変更」の区分として扱うことができます。

ID	モジュール名 (WS:緑 FM:水 SM:紫 CL:黄)	プロセス名 (Bold:Sub)	相違	コンテンツの有無	
				比較・旧	比較・新
1	ThisWorkbook			●	●
2	Sheet1			●	●
3	Module1		有	●	●
4		Test1	有	○	○
5		test2		○	○

●”Test1”に差分があり
その差分詳細は引数の
追加であることが分りま
す

ID	比較・旧Code	比較・新Code
1	Sub Test1(a As String)	Sub Test1(wb As Workbook, a As String)
2	Dim x As Long	Dim x As Long
3	End Sub	End Sub

●引数の”wb”が追加さ
れています

(3) Tab の影響を排除してコードの差分にフォーカスした分析を行う

本機能は、旧コードを残したままIf文等のロジックを追加した場合に、追加したロジック部分だけを確認するのに適しています。Ifのため旧コードはTab等の影響を受けますので、一般には別コードと認識しがちです。

《空白文字も差分比較の設定》

Before

☒ 空白文字を差分比較する

ID	比較・旧Code	比較・新Code
1	Sub Test()	Sub Test()
2	Dim a As Long	Dim a As Long
3	Dim b As Long	Dim b As Long
4		Dim c As Long
5	a = 11	a = 11
6	b = 10	b = 10
7	If a >= b Then	c = 12
8	MsgBox "a>=b"	If c > a Then
9		If a >= b Then
10		MsgBox "a>=b"
11		End If
12	End If	End If
13	End Sub	End Sub

● If ネストを追加したため、If 文に先行する Tab/スペースコードを差分として分析したため適切な比較分析となっていない

After

☐ 空白文字を差分比較する

ID	比較・旧Code	比較・新Code
1	Sub Test()	Sub Test()
2	Dim a As Long	Dim a As Long
3	Dim b As Long	Dim b As Long
4		Dim c As Long
5	a = 11	a = 11
6	b = 10	b = 10
7		c = 12
8		If c > a Then
9	If a >= b Then	If a >= b Then
10	MsgBox "a>=b"	MsgBox "a>=b"
11	End If	End If
12		End If
13	End Sub	End Sub

● コードに先行する Tab/スペースの影響を排除したため、追加されたコード部分がより鮮明になる。

一般に、コメントの評価もオフに設定するが必要である。

(4) モジュールの共通宣言文の順序分析に関して

本機能は、開発の最終ステージにおいて以降の保守に備えて宣言文等を整理を行います。誤った変更をしていないかを確認する場合に効果を発揮します。

☒ プロセス及び共通宣言文の順序を差分比較する

《宣言文の分析例》

ID	比較・旧Code	比較・新Code
1	Option Explicit	Option Explicit
2	無造作に宣言文を配置	宣言文を整理しました
3		Public y As Variant, yy As String 'add yy
4		Public a As Variant '
5		'blに関する宣言文
6		Public b As Variant, b1 As Variant
7		Public c As Variant
8		Public d As Variant 'Add
9	'xlに関する宣言文	'xlに関する宣言文
10	Public Const x = "xxx"	Public Const x = "xxx"
11	Public a As Variant '	
12	Public c As Variant	
13	Public y As Variant, yy As String 'add yy	
14	Public d As Variant 'Add	
15	'blに関する宣言文	
16	Public b As Variant, b1 As Variant	

●宣言文の順序も差分として分析するために、同様な宣言文であるのに関わらず、一方では追加。他方では削除の判定となります。

☐ プロセス及び共通宣言文の順序を差分比較する

《宣言文の分析例》

ID	比較・旧Code	比較・新Code
1	Option Explicit	Option Explicit
2	無造作に宣言文を配置	宣言文を整理しました
3	Public y As Variant, yy As String 'add yy	Public y As Variant, yy As String 'add yy
4	Public a As Variant '	Public a As Variant '
5	'blに関する宣言文	'blに関する宣言文
6	Public b As Variant, b1 As Variant	Public b As Variant, b1 As Variant
7	Public c As Variant	Public c As Variant
8	Public d As Variant 'Add	Public d As Variant 'Add
9	'xlに関する宣言文	'xlに関する宣言文
10	Public Const x = "xxx"	Public Const x = "xxx"

●同じ宣言文、コメントに関しては対応関係を維持させています。

●コメントの評価をオフにすると、本例では「差分」=無しとなります

(5) プロセスの順序分析に関して

本機能は、開発の最終ステージにおいて以降の保守に備えてプロセス等の記述位置を整理を行います。誤った変更をしていないかを確認する場合に効果を発揮します。

☒ プロセス及び共通宣言文の順序を差分比較する

ID	モジュール名 (WS:緑 FM:水 SM:紫 CL:)	プロセス名 (Bold:Sub)	相違	コンテンツの有無	
				比較・旧	比較・新
1	ThisWorkbook			●	●
2	Sheet1			●	●
3	Sheet2			●	●
4	Sheet3			●	●
5	Module1		有	●	●
6		X3	有	○	
7		X1	有		○
8		X2		○	○
9		X1	有	○	
10		X3	有		○
11		X4		○	○

●旧コード側
X3 ,X2, X1, X4 の順
●新旧コード側
X1, X2, X3 , X4 の順
のため、X3 と X2 を差分と認識

ID	比較・旧Code	比較・新Code
1	Option Explicit	Option Explicit
2	プロセスの記述位置整合前	プロセスの記述位置を整合した
3	Dim a As String	Dim a As String
4	X3の説明	
5	Sub X3() X3	
6	End Sub	
7		X1 の説明
8		Sub X1(x As String, wb As Workbook) 新しいX1
9		Dim xp1 As String
10		End Sub
11	X1 の説明	
12	Sub X1(x As String, wb As Workbook) 新しいX1	
13	Dim xp1 As String	
14	End Sub	
15		X3の説明
16		Sub X3() X3
17		End Sub

☐ プロセス及び共通宣言文の順序を差分比較する

ID	モジュール名 (WS:緑 FM:水 SM:紫 CL:)	プロセス名 (Bold:Sub)	相違	コンテンツの有無	
				比較・旧	比較・新
1	ThisWorkbook			●	●
2	Sheet1			●	●
3	Sheet2			●	●
4	Sheet3			●	●
5	Module1		有	●	●
6		X1		○	○
7		X2		○	○
8		X3		○	○
9		X4		○	○

●同じプロセスに関しては対応関係を維持させています。従って、差分は共通宣言文内にあるコメントのみです。

ID	比較・旧Code	比較・新Code
1	Option Explicit	Option Explicit
2	プロセスの記述位置整合前	プロセスの記述位置を整合した
3	Dim a As String	Dim a As String

4.3.3 表示制御

■ 分析結果表示の設定

☐ 印刷向け表示を選択☒ コメント文の配色を淡い色へ設定☒ 縮小表示をする

(1) 印刷向けの表示選択

本設定は結果レポートを画面表示するのか印刷用に表示するのかを指定します。画面表示指定では、差分のセルに背景色つけて差分のある場所を探し易くしています。一方、印刷表示指定では、トナー節約の観点から背景色を付けておりません。

● 画面表示

☐ 印刷向け表示を選択

ID	比較・旧Code	比較・新Code
1	Option Explicit	Option Explicit
2	無造作に宣言文を配置	宣言文を整理しました
3		Public y As Variant, yy As String 'add yy
4		Public a As Variant '
5		'bに関する宣言文
6		Public b As Variant, b1 As Variant
7		Public c As Variant
8		Public d As Variant 'Add
9	'xlに関する宣言文	'xlに関する宣言文
10	Public Const x = "xxx"	Public Const x = "xxx"
11	Public a As Variant '	
12	Public c As Variant	
13	Public y As Variant, yy As String 'add yy	
14	Public d As Variant 'Add	
15	'bに関する宣言文	
16	Public b As Variant, b1 As Variant	

● 差分部は太字(ボールド)表示されます。

● 赤字は文字列の差分です

● コメントとコードの区別は、(2)を参照。本例はオフの場合です。

《背景色の仕様》

差分区分表示:

変更:薄橙色

削除:蒼色

追加:青緑

変更無:白

☒ 印刷向け表示を選択

ID	比較・旧Code	比較・新Code
1	Option Explicit	Option Explicit
2	無造作に宣言文を配置	宣言文を整理しました
3		Public y As Variant, yy As String 'add yy
4		Public a As Variant '
5		'blに関する宣言文
6		Public b As Variant, b1 As Variant
7		Public c As Variant
8		Public d As Variant 'Add
9	'xlに関する宣言文	'xlに関する宣言文
10	Public Const x = "xxx"	Public Const x = "xxx"
11	Public a As Variant '	
12	Public c As Variant	
13	Public y As Variant, yy As String 'add yy	
14	Public d As Variant 'Add	
15	'blに関する宣言文	
16	Public b As Variant, b1 As Variant	

- 差分部分は太字の設定です。
- 赤字は文字列の差分です

(2) コメント文の配色を淡い色へ変更

本設定はコード部とコメント部の表示色を変える場合に利用します。デフォルトの設定では、オンになっています。

☐ コメント文の配色を淡い色へ設定

ID	比較・旧Code	比較・新Code
1	Option Explicit	Option Explicit
2	無造作に宣言文を配置	宣言文を整理しました
3	Public y As Variant, yy As String 'add yy	Public y As Variant, yy As String 'add yy
4	Public a As Variant '	Public a As Variant '
5	'blに関する宣言文	'blに関する宣言文
6	Public b As Variant, b1 As Variant	Public b As Variant, b1 As Variant
7	Public c As Variant	Public c As Variant
8	Public d As Variant 'Add	Public d As Variant 'Add
9	'xlに関する宣言文	'xlに関する宣言文
10	Public Const x = "xxx"	Public Const x = "xxx"

☒ コメント文の配色を淡い色へ設定

ID	比較・旧Code	比較・新Code
1	Option Explicit	Option Explicit
2	無造作に宣言文を配置	宣言文を整理しました
3	Public y As Variant, yy As String 'add yy	Public y As Variant, yy As String 'add yy
4	Public a As Variant '	Public a As Variant '
5	'blに関する宣言文	'blに関する宣言文
6	Public b As Variant, b1 As Variant	Public b As Variant, b1 As Variant
7	Public c As Variant	Public c As Variant
8	Public d As Variant 'Add	Public d As Variant 'Add
9	'xlに関する宣言文	'xlに関する宣言文
10	Public Const x = "xxx"	Public Const x = "xxx"

- コメントの基本の色は青です。
- 「変更」区分では文字列差分のある文字は赤となります
- 「削除・追加」区分では、コメント部は黄色となります

《「削除・追加」区分では、コメント部の色》

ID	比較・旧Code	比較・新Code
1	Option Explicit	Option Explicit
2	'無造作に宣言文を配置	'宣言文を整理しました
3		Public y As Variant, yy As String 'add yy
4		Public a As Variant '
5		'hに関する宣言文

注:コメントのみからなる文字列において、表示上は1つのシングルコーテーション“'”に見えますが、「式」表示窓には2つのシングルコーテーション“'”となっています。これは Excel の仕様上、1つのシングルコーテーション“'”は表示されないため、本ツールで意図的に連続する2つのシングルコーテーション“'”にしてあります。従って、元々コードに二つあった訳ではありません

※ "無造作に宣言文を配置"

(3) 縮小表示

本設定は、一行の文字数が長い場合に、その文字列に差分が無い場合に縮小表示をするものです。差分のある場合には折り返し表示となります。この為、コード差分表示を効率的に表示することが可能になります。デフォルトの設定ではオンになっています。

☐ 縮小表示をする

ID	比較・旧Code	比較・新Code
1	Sub Test()	Sub Test()
2	If (aaaa = "xxxxx" And bbbb = "yyyy" And cccc = "zzzzz" And dddd = "wwwww" And eeee = "vvvv")	If (aaaa = "xxxxx" And bbbb = "yyyy" And cccc = "zzzzz" And dddd = "wwwww" And eeee = "vvvv")
3	'Debug.Print aaaa	Debug.Print aaaa & ", " & bbbb & ", " & cccc & ", " & dddd & ", " & eeee
4	End If	End If
5	End Sub	End Sub

☒ 縮小表示をする

ID	比較・旧Code	比較・新Code
1	Sub Test()	Sub Test()
2		
3	'Debug.Print aaaa	Debug.Print aaaa & ", " & bbbb & ", " & cccc & ", " & dddd & ", " & eeee
4	End If	End If
5	End Sub	End Sub

● 縮小表示部のコードを確認したい場合には「式」窓をご利用下さい。

4.4 印刷設定

分析結果の印刷の設定仕様を以下に示します

(1) モジュール・プロセス一覧

≪A4縦の印刷書式≫

マクロ差分 結果(1/1) : モジュール一覧日時:2012/07/09-10/59

比較・旧		E¥■■棚卸シリーズ¥棚卸【単独 マクロ差分】¥VBATest¥Shrink表示¥Old.xls	コード量	
比較・新		E¥■■棚卸シリーズ¥棚卸【単独 マクロ差分】¥VBATest¥Shrink表示¥new.xls	14	
			14	
ID	モジュール名	プロセス名 (Bold:Sub)	相違	コンテンツの有無
2	Sheet1		有	● ●
3		Test	有	○ ○

(2) コード差分

≪A4横の印刷書式≫

コード差分結果(1/1) : Sheet1日時:2012/07/09-10/54

戻る		モジュール名	Sheet1 (Document)
ID	比較・旧Code	比較・新Code	
1	Sub Test()	Sub Test()	
2	If (aaaa = "xxxx" And bbbb = "yyyy" And cccc = "zzzz" And dddd = "wwwww" And eeee = "vvv" And fff = "vvv") = True Then	If (aaaa = "xxxx" And bbbb = "yyyy" And cccc = "zzzz" And dddd = "wwwww" And eeee = "vvv" And fff = "vvv") = True Then	
3	Debug.Print aaaa	Debug.Print aaaa	
4	End If	End If	
5	End Sub	End Sub	

(3) 一括印刷

印刷指定は「ファイル」・「印刷」にて「ブック全体」を指定してください。

両面印刷は、レポートのシート単位になります。

また、ページの通しは有効になります。

5. お問い合わせ

お問い合わせは、専用の[Webサイト](http://yuzu-miso.sakura.ne.jp)から以下のお問い合わせフォームにてお願いします。

<http://yuzu-miso.sakura.ne.jp>



棚卸【パスワード】、【関数・マクロ】、【個人情報】
の発売を開始しました

お問い合わせ

ホーム

概要

...

サポート

更新履歴

ご注文

■ クレームフォーム／ライセンスキー更新

1. 対象の製品を選択してください。
2. ご利用中のライセンスキーを記入してください。
3. 以下のフォームより、必要事項(*)は必須)を入力のうえ送信してください。

お名前(*)	<input type="text"/>
メールアドレス(*)	<input type="text"/> (確認用)
対象商品(*)	棚卸【単独 マクロ差分】
ライセンスキー(*)	<input type="text"/>
タイトル(*)	<input type="text"/>
内容(*)	<div><div></div></div>
添付ファイル (不具合画面 のキャプチャ用)	<div><div></div><div></div><div></div></div> <div>参照... 参照... 参照...</div>

送信する

リセット

●ライセンスキーはお客様管理上必要なため、購入時のキーを設定して下さい。
尚、評価版の不具合等に関しては「1」を設定して下さい。