

dbSheetClient のススメ

R1.2

原 作 : 小澤 光司

監 修 : 株式会社ニューコム

発行日 : 令和 2 年 4 月 16 日

《法的な注意事項》

本ドキュメント及び同時に提供される dbSheetClient のプロジェクトファイル、Excel ブック、データベース、データ、スクリプトファイルを含む一切の提供物の作成にあたっては細心の注意を払っていますが、記載事項の誤り、または欠落により発生した損害については責任を負いかねますのでご了承ください。前述の一切の提供物は、株式会社ニューコムが事前に承諾している場合を除き、製品の使用許諾契約書に記載の条件に同意をされたライセンス所有者に対してのみ供給されるもので、それらの利用に関しては製品使用許諾の関連する取り決めをお守りいただきます。

《著作権表示》

本ドキュメント及び同時に提供される一切の提供物に関する著作権は、株式会社ニューコムへ独占的に帰属します。株式会社ニューコムが事前に承諾している場合を除き、形態および手段を問わず、本書またはその一部を無断で複製・転載・改編・公開することを禁止します。

《商標表示》

本書に記載されている「dbSheetClient 実行版」、「dbSheetClient 開発版」、「dbSheetClient サーバー版」は株式会社ニューコムの製品です。

本書に記載されているマイクロソフト製品 (Windows、Office、Excel 等) は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。その他記載の各社の社名、製品名およびサービス名は、各社の商標または登録商標です。

目次

1.	はじめに.....	7
2.	dbSheetClient とは.....	7
2.1	dbSheetClient システム構成.....	7
2.2	Excel がデータベース連携すると何がかわるか.....	8
2.3	どんな業務に適用できるか.....	9
2.4	費用対効果の分岐点.....	10
2.5	サイボウズ社「kintone」との棲み分け.....	11
2.6	dbSheetClient の弱点.....	12
2.6.1	Excel が必須.....	12
2.6.2	複数プロジェクトの切替.....	12
2.6.3	起動時間.....	12
3.	システム開発運用のヒント.....	13
3.1	スモールスタート.....	13
3.2	現場の Excel を活用する.....	13
3.3	愛称を付ける.....	14
3.4	キラーコンテンツを見つける.....	14
4.	プロトタイプ作成編.....	15
4.1	新規プロジェクト作成.....	15
4.2	ユーザー、グループ登録.....	16
4.3	プロジェクトをアップロード.....	17
4.4	プロジェクトへの参加.....	18
4.5	画面のクローズボタン設定.....	19
4.6	Excel ファイルの準備.....	20
4.7	メニューの登録.....	23
4.7.1	新規メニュー登録.....	23
4.7.2	文書管理メニューの設定.....	24
4.7.3	デバッグモードで実行.....	25
4.7.4	プロジェクトを保存.....	26
4.8	ボタンを設置.....	26
4.8.1	メニューにボタンセットを割り付ける.....	26
4.8.2	新規ボタンを登録.....	27
4.8.3	プロジェクトを保存.....	29
4.9	タスクの登録.....	29
4.9.1	印刷ボタンの工夫.....	29

4.9.2	メニュー勤怠管理表の起動前処理にタスクをセット	32
4.9.3	保存ボタンの設定	34
4.9.4	PDF ファイル保存ボタン	39
4.9.5	その他様式のメニュー(シート)切替	44
4.9.6	プロジェクトを保存	48
4.10	本番環境へアップロード	49
4.11	ユーザー環境で実行	50
5.	経営の見える化	51
5.1	プロジェクトの準備	51
5.2	データベースの準備	52
1.2.1	エイリアス設定	52
1.2.2	データベース登録	53
1.2.3	売上テーブルの作成	55
5.3	パレート図	59
1.3.1	グラフの雛形を準備	59
1.3.2	メニュー、ボタン、タスクを作成	62
1.3.3	「入力画面」タスクを使った対象期間の指定	66
1.3.4	「データ照会」タスクを使った対象期間の指定	72
1.3.5	「検索付きデータ照会」タスクを使った対象期間の指定	75
1.3.6	部署、分類を絞り込む	81
1.3.7	抽出条件を表記	85
1.3.8	初期表示画面の調整	87
1.3.9	印刷ボタン、保存ボタンの設定	89
1.3.10	明細を展開する	94
1.3.11	「売上先」分析タスクをコピーして「分類」分析タスクを作成する	101
1.3.12	最終調整	105
5.4	Z 図	108
1.4.1	グラフの雛形を準備	108
1.4.2	メニュー、ボタンの追加	110
1.4.3	タスクの登録	111
1.4.4	デバッグモードで実行	120
1.4.5	条件判定分岐タスクの適用	121
1.4.6	メニュー番号を振り直す	124
1.4.7	ボタン番号を振り直す	125
1.4.8	タスク番号を振り直す	125
5.5	売上分析	126
6.	予算を収集する	128

6.1	予算収集の要点	128
2.1.1	ユーザー志向	128
2.1.2	取りまとめ部署作業負荷軽減の視点	129
2.1.3	情報システム部門の立ち位置	129
6.2	予算テーブルの設計	129
2.2.1	シンプル	130
2.2.2	顧客志向	130
2.2.3	商品志向	131
2.2.4	製造業、メーカー代理店	131
2.2.5	ハイブリッド	132
6.3	動かしてみよう	133
6.3.1	システム概要	133
6.3.2	データベース概要	134
6.3.3	動作環境(DB)の構築	134
6.3.4	動作環境(プロジェクト)の設定	136
6.3.5	デバッグ実行	137
6.3.6	デバッグ実行:予算収集	139
6.3.7	デバッグ実行:社長室メニュー:予算テーブル準備	142
6.3.8	デバッグ実行:社長室メニュー:入力確定テーブル準備	142
6.3.9	デバッグ実行:社長室メニュー:データバックアップ	143
6.3.10	デバッグ実行:社長室メニュー:査定データ準備	143
6.3.11	デバッグ実行:社長室メニュー:オフライン対応	144
6.3.12	デバッグ実行:管理者メニュー	146
6.3.13	マクロ利用とタスク利用	147
6.4	プロジェクトを作成しましょう	148
6.4.1	動作環境:DB「DBC 作成」の構築	149
6.4.2	動作環境:プロジェクト「予算収集」の新規作成	150
6.4.3	動作環境:Excel「予算収集.xlsx」の新規作成と設定	150
6.4.4	動作環境:プロジェクトに DB、Excel を定義	152
6.4.5	動作環境:DB ツールでテーブル作成	154
6.4.6	動作環境:画面のクローズボタン設定	158
6.4.7	動作環境:メニュー:TOP の設定	159
6.5	プロジェクト作成:管理者メニュー	161
6.5.1	予算入力ユーザー一覧(IOTG 利用)	162
6.5.2	予算テーブル(IOTG 利用)	168
6.5.3	予算入力確定テーブル(IOTG 利用)	169
6.5.4	メニュー切替で表示クリア	170

6.5.5	作成機能の動作確認(予算入力ユーザー一覧:担当部署コードの変更)	177
6.6	プロジェクト作成:予算収集(「読込」、「更新」)	178
6.6.1	Excel の前準備	178
6.6.2	メニュー「予算収集」の設定の前準備	179
6.6.3	メニュー「予算収集」の設定	184
6.6.4	「読込」①	185
6.6.5	「読込」②	189
6.6.6	「読込」③	192
6.6.7	ボタン「読込」の設定	194
6.6.8	「更新」①	195
6.6.9	「更新」②	196
6.6.10	ボタン「更新」の設定	198
6.7	プロジェクト作成:社長室メニュー	199
6.7.1	メニュー「社長室メニュー」の設定	200
6.7.2	メニュー「データバックアップ」の設定の前準備	200
6.7.3	メニュー「データバックアップ」の設定	202
6.7.4	データバックアップ	203
6.7.5	ボタン「バックアップ実行」の設定	207
6.7.6	メニュー「オフライン対応」の設定の前準備	208
6.7.7	メニュー「オフライン対応」の設定	211
6.7.8	オフラインファイル保存	212
6.7.9	ボタン「オフラインファイル保存」の設定	213
6.7.10	オフラインファイル読込	213
6.7.11	ボタン「オフラインファイル読込」の設定	217
6.8	プロジェクト作成:最終調整	218
6.8.1	更新忘れを防ぐ	218
6.8.2	重要なレコードに色を付ける	219
6.8.3	排他制御	220
6.9	未作成の機能の説明	221
6.9.1	ユーザーが複数支店を担当する場合の対応	221
6.9.2	「予算収集」のボタン「更新」無効化	223
6.9.3	社長室メンバー以外の「社長室メニュー」無効化	224
6.9.4	未登録ユーザーがログインした時の通知	225
6.10	マクロで実装する場合	226
6.10.1	マクロ実装の前準備	226
6.10.2	バックアップファイル名セット:マクロ実装	230
6.10.3	バックアップファイル名セット:マクロ起動	232

6.10.4	オフラインファイル読込:マクロ実装	234
6.10.5	オフラインファイル読込:マクロ起動	236
7.	開発運用の小技	237
7.1	Excel メニュー(タブやリボン)を解放する	237
7.2	ショートカットからの起動	238
7.3	マクロは是が非か?	238
7.4	SQL はこれだけ使えばいい.....	239
7.5	デバッグのコツ	241
7.6	RPA モドキ.....	243
7.7	ログの活用	244
8.	おわりに.....	246

1. はじめに

本書は著者が業務で活用している dbSheetClient (株式会社ニューコム社 開発・販売) について、ユーザー視点から具体的な活用事例を挙げて業務効率化の実践的な手法とノウハウを解説します。

Excel は誰もが知っていて使っている身近なソフトウェアです。

dbSheetClient は Excel の持つ潜在パフォーマンスをフルに発揮させるシステム開発ツールです。

本書は次の立場の方々に現状を打開する切り口として dbSheetClient 採用を推奨します。

- ① 「業務改善せよ」と上からも下からも責められて何から手をつけてよいか途方に暮れている方
- ② 業務課題は特定できているのに改善するための膨大な時間とコストに唖然としている方
- ③ ユーザーの要望に100%応えることができずいつも悶々としている方

本書は次の異なる立場の方を対象としているため内容の展開に違和感を持たれるかもしれません。

dbSheetClient 導入を検討されているユーザー

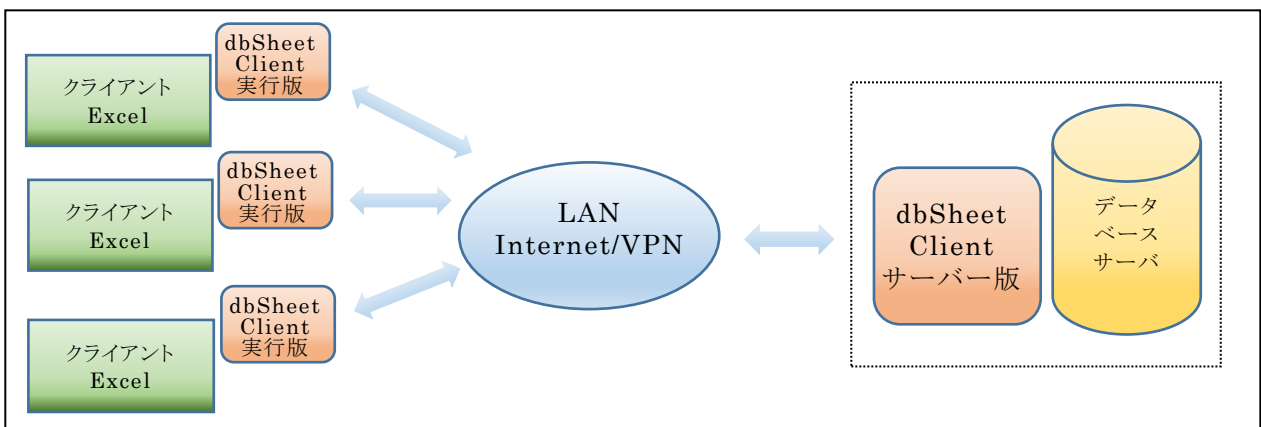
dbSheetClient を既に導入されていて活用をどう進めていったらよいか悩んでいる担当者の方

2. dbSheetClient とは

一言でいうと皆さんが普段お使いの Excel にデータベースと連携する機能を持たせるシステムです。

Excel の機能を損なうことなくデータベースに蓄積された情報を呼び出したり、入力した情報をデータベースに格納し共有することができます。

2.1 dbSheetClient システム構成



2.2 Excel がデータベース連携すると何がかわるか

Excel を使った見積書作成業務の場合では下記の違いがあります。

比較項目	Excel 単独	dbSheetClient 活用
見積番号	担当者や部署で別々に採番 AB-001、AB-002、・・・ ZA10001、ZA10002、・・・ M19A0001、M19A0002、・・・ 部署間で調整されていないと重複する 可能性がある。	全社統一の自動採番、重複しない。 M100001、S100002、T100003、・・・ 先頭のアルファベットを部署によって変える ことで見積番号から作成部署を特定する ことも可能(連番部分は全社共通でカウン トアップ)
共有	担当者や部署のフォルダに保管 サブフォルダの体系やファイル名の付け 方がバラバラ 取引先からの問合せ時に、いつ、どこに、 どんな見積書が提示されたのか調べるの に時間がかかる。 担当者パソコンに保存されていると共有不 可	作成時期や提出先、商品名等の情報で 検索可 保管先を指定することで担当者不在時で も軽微な修正に対応可 一元管理されるため引継ぎ時の負荷が軽 減
商品マスタ	商品価格や新商品情報が担当者や部署 で共有されているため価格表の改訂作業 が重複 改訂情報の共有が徹底されていないと旧 価格が誤って使用されてしまうリスクあり	商品価格や新商品情報を担当部署が一 元管理できる。 価格改訂作業の重複を防げる。 改訂がタイムリーに反映される。
様式	担当者や部署で様式がバラバラ 間違った関数式が設定されていたり 担当者しかメンテナンスできないマクロが 実装されていたりする。	全社で様式を統一可 (注) 部署や商品による様式の個性をいか に包括できるかがカギ 業務標準化が進む。 中途入社社員への業務手順説明や退社 社員の業務引継ぎ時に作業が軽減
次工程	提出した見積が発注になった時 次工程システムに連携されていないと 見積書に記載された情報(物件名や商品 名、価格等)を再度、別システムに入力す る二度手間が発生する。	データベースに格納された情報は次工程 に流用できる。 次工程とは仕入先への発注書であったり 社内の製作工程表であったり取引先との 契約書を指す。

<まとめ>

担当者のパソコン内、あるいは、部署の共有フォルダ内の Excel データが Excel 表示で全社で共有できるように変わります。

蓄積された情報は検索し再利用可能な状態となり二重管理による負荷やミスがなくなります。

属人化された業務手順が標準化され担当者不在時や業務引継ぎ時の対応が円滑になります。

2.3 どんな業務に適用できるか

今まで dbSheetClient 上で開発したシステム(以下、プロジェクトという)を分類してみました。
ご覧の通り、様々な業務に適用可能です。

適用業務	分類	プロジェクト
データベースに格納された情報を活用する	社外向け	会計監査用母集団データ抽出 税務署提出用消費税照合表集計 納品書、請求書、注文書作成(各部署別)
	社内	Z 図、パレット図 延滞債権管理表 債権残高推移表 仮受残高明細 再販実績集計
情報を収集し集計する	日次	販売管理システム(各部署別) 見積書作成 工事経歴管理 工事契約台帳
	月次	業務報告 営業資金予算 物件情報共有リスト 売上目標管理
	年次	予算申請 与信限度申請 業務内容紹介シート 自己申告書
	都度	労災保険申請 取引先属性登録(調書より) 不動産担保一覧
その他		保管用ダンボールラベル作成 複合機出力ログ分析(費用部署振替) 消込処理(計上データと客先支払い明細との照合) メーカー EDI データ変換 IT 全般統制(端末リスト、予約ジョブ差異抽出) 振替伝票起票 リスク情報等発生報告データ分析

作業時間短縮を目的とした場合、負荷軽減効果を測定することは容易です。

しかし、業務標準化や情報共有を狙った場合、経過を観察しないと導入効果の測定は難しいです。

スモールスタートが容易なシステムなので現場の反応を見て抜げるか撤収するかを判断してもよいです。

2.4 費用対効果の分岐点

dbSheetClient ソフトウェア & 保守費用

単位:円

項目	ユーザー数			
	100名	300名	500名	1000名
①サーバー版	800,000	1,400,000	1,400,000	1,400,000
②実行版	1,000,000	1,500,000	1,500,000	3,000,000
③開発版 1ライセンス 20万円	400,000	400,000	400,000	400,000
④オプション(クリックワンス対応)	100,000	サーバー版に含む	サーバー版に含む	サーバー版に含む
⑤基本・応用セミナー(4日)	400,000	400,000	400,000	400,000
⑥サーバー初期設定費用 (基本設定)	200,000	200,000	200,000	200,000
⑦保守費用(サポート含む) ※(①+②+③+④)×16%+10万円	468,000	628,000	628,000	868,000
初年度(イニシャル) ※①～⑦	2,900,000	3,900,000	3,900,000	5,400,000
次年度以降(年額) ※⑦	468,000	628,000	628,000	868,000
1ユーザー当たり 次年度以降(年額) ※⑦÷ユーザー数	4,680	2,093	1,256	868

(注1) 上記費用には次の費用が含まれていません。

- ・dbSheetClient が稼働するサーバーのハードウェア、OS、保守費用
- ・データベースサーバーが稼働するサーバーのハードウェア、OS、保守費用

※最近、サブスクリプション契約(年単位)がメニューに追加されました。

※クラウドサービス(アマゾン AWS やマイクロソフト Azure)を利用してサーバーを構築することも可

IT 投資を「守り」と取るか「攻め」と取るかによって費用対効果の分岐点は変わります。

「守り」すなわちコスト低減、「攻め」は顧客満足度向上、販路拡大、新商品、新サービスの開発。

現状の延長線で「働き方改革」を試みても業務改善効果は小さく、現場の実感が伴わないと思います。

劇的な業務改善を実行する手段は IT の力をどう上手く活用するかにかかっています。

2.5 サイボウズ社「kintone」との棲み分け

最近、通勤電車の車内広告にもよく見かけるサイボウズ社の「kintone」

著者の会社でも前バージョン「デヂエ」の時代から導入し活用しています。

導入企業が増えるにしたがって周辺業務をサポートするベンダーも増え活況を呈しています。

今後も手軽に情報共有するツールとして普及していくことでしょう。

「kintone」に適した業務や企業風土であればスピーディーに業務改革を実現できるでしょう。

しかし、「kintone」にも実現できることに限界があり、適用が難しい業務もあります。

本項では「kintone」の弱点を dbSheetClient がどう補えるかについて説明していきます。

比較項目	kintone	dbSheetClient
既存データベースとのデータ連携	データ連携ツール(EAI)を別途導入することで既存データベースとデータ交換することができます。 それなりの技術、時間、費用が発生します。 データ量や同期タイミング、スピードにも実運用上の課題が残ります。	データベースとは連携ではなく一体となります。 レスポンスが落ちません。 運用(データバックアップ等)がシンプル
表形式でのデータ編集	Excel ライクなデータ入力・編集が苦手 プラグインを使えばある程度の編集はできるが制約が多い。	データの入力規則や条件書式、オートフィルター等 Excel の機能がすべてそのまま使えるため、新たな機能の作り込みが不要
Excel 様式への印刷	Excel で作成した様式にデータを流し込むプラグインがあるが別途費用が発生する。また設定に負荷がかかりユーザー側の手順もスムーズではない。	Excel シートの様式は「印刷タスク」でそのまま印刷できる。
Excel 形式での保存	データのアウトプットは CSV 形式 Excel 形式で保存する場合、印刷同様に別途プラグインが必要	「保存タスク」でそのまま保存できる。 指定したシートのみを保存可 保存ファイル名も予め設定可
コスト	ユーザー数が増えるとコスト増大 <スタンダード> 1,500 円/月/ユーザーID → 17,640 円/年/ユーザー ※300 ユーザー:年間約 530 万円 <ライト>(カスタマイズ限定バージョン) 780 円/月/ユーザーID → 9,170 円/年/ユーザー ※300 ユーザー:年間約 270 万円 サーバー等ハードウェアは不要 保守費用も上記料金に含まれる。 (クラウドサービス)	前項「2.4.費用対効果の分岐点」参照 ※300 ユーザー:年間約 63 万円 初年度 390 万円 (注)上記金額にはサーバーのハードウェアとその保守費用は含まれていません。 (注)上記金額にはデータベースサーバーのソフトウェア、ハードウェアとその保守費用は含まれていません。

2.6 dbSheetClient の弱点

ユーザーの立場としては、メーカーが触れたがらない dbSheetClient の弱い部分にも言及せざるを得ません。

改善が困難な点もあれば今後克服できる点もあると思いますので、敢えて書かせて頂きます。

2.6.1 Excel が必須

dbSheetClient は Excel が動作する端末でないと利用できません。

いま流行のマルチデバイス対応(機種や OS 等に依存しない)ではありません。

わかりやすく言うとスマートフォンやタブレット(iOS や Android)では利用できません。

2.6.2 複数プロジェクトの切替

dbSheetClient はマルチプロジェクト対応オプションを導入すれば複数のプロジェクトを同時に起動することができます。

しかし、ブラウザのタブで切替するような操作性(切替スピードや画面サイズ変更等)ではありません。

同じプロジェクトを複数起動することはできません。

2.6.3 起動時間

プロジェクト起動時にバックグラウンドで Excel も同時に起動するため若干時間がかかります。

ユーザー環境(パソコンのスペックやメモリサイズ等)にもレスポンスは左右されます。

3. システム開発運用のヒント

著者は、新卒で商社に入社し、営業部門に17年在籍後、情報システム部門に異動しました。システム開発未経験者でしたが dbSheetClient 導入から7年間で約200のプロジェクトを起動しました。本章では、システムを開発する上で、あるいはシステム開発後の運用におけるヒントをまとめてみました。

3.1 スモールスタート

dbSheetClient 導入担当部署はすぐに効果を上げようと焦ってしまう傾向にあります。いきなり会社の柱となる大規模なワークフローの改善に着手したり、仕様打合せと称してユーザー部門に長時間の負荷をかけてしまったりして、かえって社内で浮いてしまいかねません。まずは小さな成功を積み上げていくことをオススメします。dbSheetClient は期待を裏切らない素晴らしいシステムです。最後にはユーザーを楽にしてあげられると信じてください。スモールスタートとは特定の部署・担当者にターゲットを絞ることです。さらにその部署・担当者が関わっている特定の業務に絞ることです。業務改善は劇的である必要はなく、こんなことで喜んでもらえるのかと拍子抜けするぐらいで良いのです。そして dbSheetClient を体感したユーザーが想像力を発揮して別の業務への適用を相談してきたらシメタものです。

3.2 現場の Excel を活用する

Excel をベースにユーザー部門と打合せができるということが dbSheetClient 最大の強味です。既存の業務に Excel が使われていたら、その手順、様式をなるべく変えないで、さらには現場で生きたデータをそのまま使ってシステムのタタキ台を作成することができます。いかにユーザーに違和感、抵抗感を持たれないようにするか、円滑な仕様打合せの重要なポイントです。運よく現場の Excel を入手できたら、あとはスピードが大切です。打合せした当日、あるいは翌日にはタタキ台を見せられるぐらいのスピード感を実現します。タタキ台は中途半端でも早ければ早いほどシステム開発のポイントがどこにあるか気づくことができます。

3.3 愛称を付ける

dbSheetClient ユーザー会の事例発表で感心したのはシステムに親しみやすい愛称を付けている会社があったことでした。

「でーびーしーとくらいあんと」ではちょっと長すぎて業務中の会話では舌を噛んでしまいます。(笑)

例えば社長の名前から「センちゃん」とか、横文字でカッコよく「D シート」とか

『センちゃんに入れといたから』とか『D シート見て』と会話が交わされるようになったら普及が進みます。

著者の会社でも某部門向けに開発した計上システムに愛称を募集して付けました。

今ではその愛称が共通語として流通して情報共有の効果を上げています。

3.4 キラーコンテンツを見つける

キラーコンテンツとは全社員が関わる業務のことです。

そのような業務に dbSheetClient を適用すると社内での認知度が格段にアップします。

勤怠管理や自己申告書などが候補となるでしょう。

dbSheetClient の存在を全く知らない部署や階層ができないようにすることでスポットライトの当たっていなかった業務が改善対象として浮かび上がってくることもあります。

4. プロトタイプ作成編

まずはデータベースと連携しないプロジェクトを例として、dbSheetClient でどのようにシステム開発を行うのか具体的な手順を紹介していきます。

開発のスピード感、開発後にも機能アップしていけるフレキシビリティをイメージして頂ければ幸いです。本章では複数の人事関連 Excel 様式を印刷、保存する簡単な文書管理システムについて説明します。本章では次の手順についてご確認ください。

- ①dbSheetClient を利用するためのユーザー、グループ、プロジェクト登録
- ②プロジェクトの初期設定
- ③メニュー、ボタン、タスクの関係

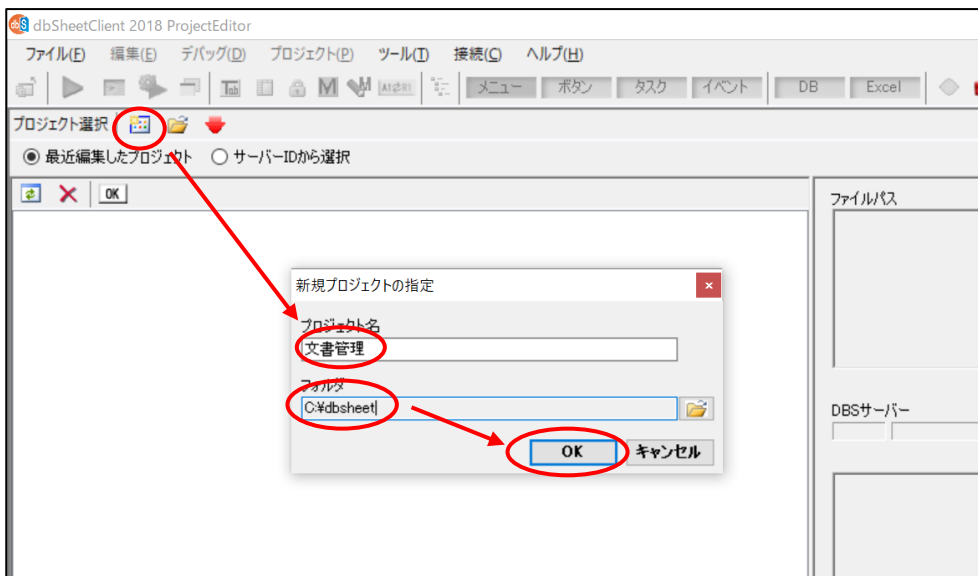
以下に説明するシステム動作環境は下記の通りです。

- ・dbSheetClient 開発版 Ver8.0.7.0 以上
- ・データベース:Microsoft SQL Server2012 以上
- ・Excel 2013 以上または Office365

4.1 新規プロジェクト作成

dbSheetClient 開発版にてプロジェクト新規作成ボタンをクリックしてプロジェクト名を入力、フォルダの指定は「C:¥dbsheet」とします。

プロジェクトファイルは「C:¥dbsheet¥文書管理¥文書管理.dbsPx」に作成されます。



4.2 ユーザー、グループ登録

システム管理画面にて「開発グループ」を作成しユーザーを所属させます。

(注:開発体験版では本作業不可)

(注) 開発ユーザーの場合は「新規開発ユーザーの登録」となります。

ユーザー登録

dbSheetClient / サーバーシステム Ver8.0.5.1 (ORG:6)

【全ユーザー情報】 新規実行ユーザーの登録 460 (FL:100)

ユーザー検索: [] (ユーザーID、氏名、カナ)

CSVアップ: [] 参照...

無効ユーザー表示

ユーザーID	氏名(漢字)	氏名(カナ)	権限	管理者	稼働	開始日	終了日
admin	管理システム		0		0	2016/08/27	2100/12/31
admin	管理システム		0		0	2019/03/19	2100/12/31

グループ登録

dbSheetClient / サーバーシステム Ver8.0.5.1 (ORG:6)

【グループ情報】 新規登録

ツリー表示 リスト表示

グループコード	グループ名	開始日	終了日
2	管理システム	2012/03/29	2100/12/31
3	開発グループ	2012/03/30	2100/12/31
4	管理システム	2012/03/30	2100/12/31

ユーザーをグループに所属させる

dbSheetClient / サーバーシステム Ver8.0.5.1 (ORG:6)

【グループユーザー情報】

グループ選択: GrCD:3_開発グループ [全ユーザー追加]

ユーザー選択: [ユーザーを選択してください] 絞り込み: []

	ユーザーID	氏名(漢字)
削除	admin	管理システム
削除	admin	管理システム

4.3 プロジェクトをアップロード

プロジェクトをサーバーにアップロードします。

(注: 開発体験版では本作業不可)

(注: 開発ユーザー権限 10+システム管理者権限ありの場合のみ新規プロジェクトをサーバーに作成可能)

アップロードにはサーバーとの接続が必要

dbSheetClient

プロジェクトをサーバーにアップロードします。よろしいですか?

OK キャンセル

【サーバーのプロジェクト管理情報】
このプロジェクトはローカルプロジェクトです。

プロジェクトをアップロード 編集済プロジェクトの定義内容を、サーバーにアップロードします。

プロジェクトをダウンロード 現在の定義内容を、サーバーにアップされている内容で上書きします。

サーバーのプロジェクトを上書き 初期状態になっているサーバーのプロジェクトに対して、現在の定義内容で上書きします。

dbSheetClient

このプロジェクトはサーバー未登録です。新規登録しますか?

はい(Y) いいえ(N)

プロジェクトを所属させるグループを選択してください

ID	グループ名
3	開発グループ
68	ALL 全登録ユーザー

OK

dbSheetClient

プロジェクトのアップロードが正常に終了しました。

OK

4.4 プロジェクトへの参加

作成したプロジェクトを「開発グループ」以外のグループに利用させるにはシステム管理画面にてグループをプロジェクトに参加させる設定が必要です。(注: 開発体験版では本作業不可)

(注) プロジェクトへの参加はグループ単位となります。ユーザー単位の参加は不可

グループをプロジェクトに参加させる

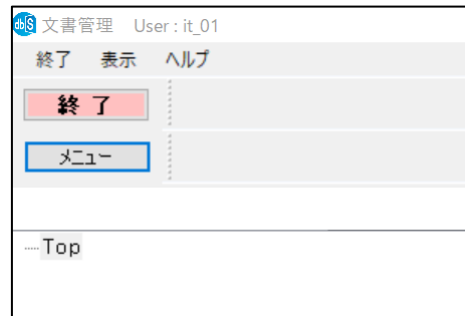
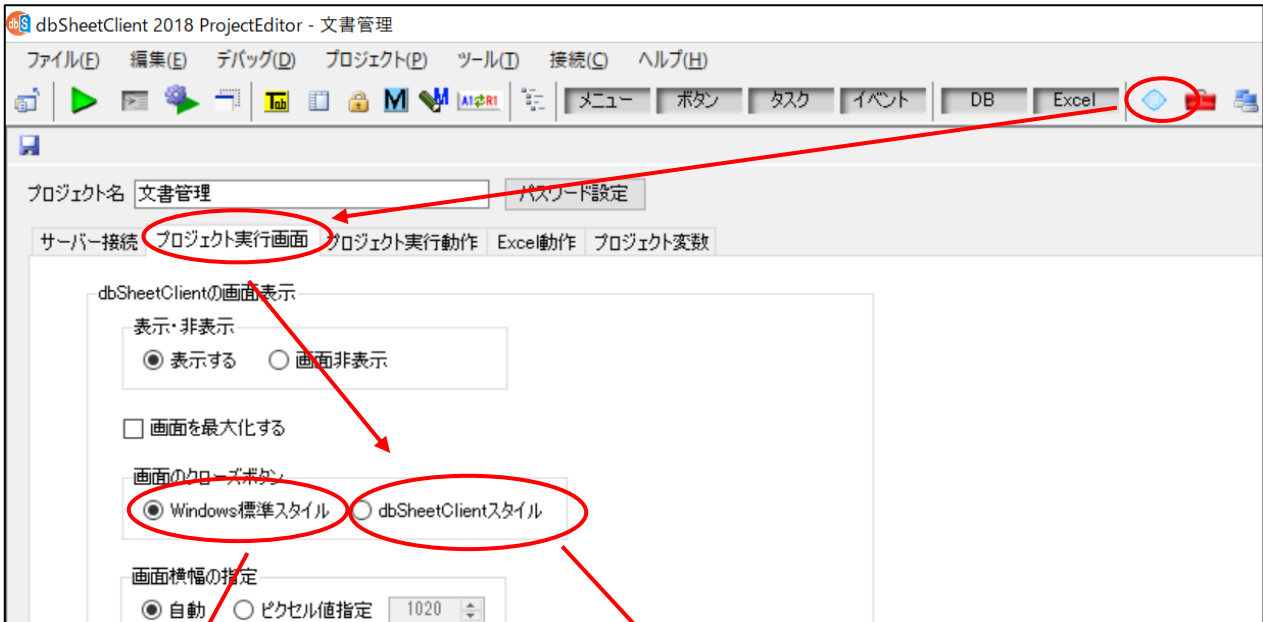
The screenshot shows the 'dbSheetClient / サーバシステム Ver8.0.5.1' interface. On the left is a navigation menu with categories: (1) リソース管理, (2) グループ管理, and (3) プロジェクト管理. Under (3), '1. グループ参加' is circled in red. A red arrow points from this menu item to the main content area. The main content area has a breadcrumb 'トップ > グループ参加' and a section titled '【プロジェクトグループ情報】'. It contains two dropdown menus: 'プロジェクト選択:' with 'PID:186 文書管理' selected, and 'グループ選択:' with 'グループを選んでください' selected. Below these is a table with columns 'グループコード' and 'グループ名'. The table has one row with '削除' in the first column, '3' in the second, and '開発グループ' in the third.

	グループコード	グループ名
削除	3	開発グループ

4.5 画面のクローズボタン設定

作成したプロジェクト起動時の画面設定を次の2種類から選択します。(初期設定は「Windows 標準スタイル」)

※本書では dbSheetClient スタイルを採用



4.6 Excel ファイルの準備

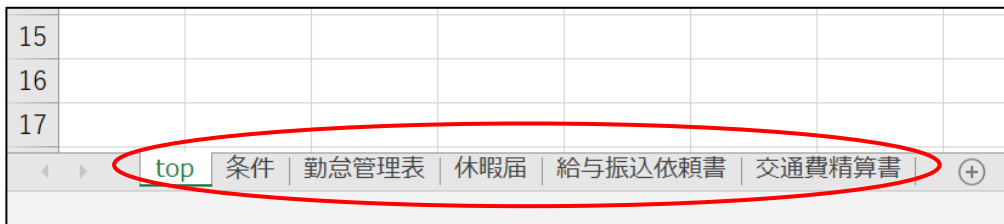
プロジェクトのベースとなる Excel ファイルを用意します。

Excel を起動してメニューより「新規」>「空白のブック」を選択

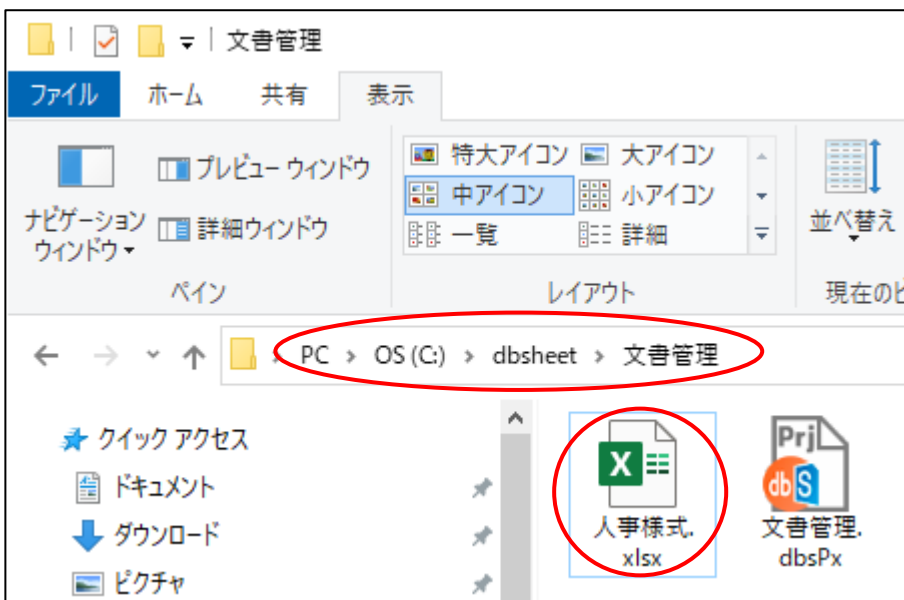
初期設定されたシート「Sheet1」を「top」に変更します。

さらに「条件」というシートを「top」の右隣りに挿入します。(空白のシート、利用目的は後述)

次に社内で利用している様式をこのブックにコピーします。



C:\¥dbsheet¥文書管理 フォルダ (プロジェクトファイルと同じフォルダ) に「人事様式.xlsx」と名前を付けて保存します。

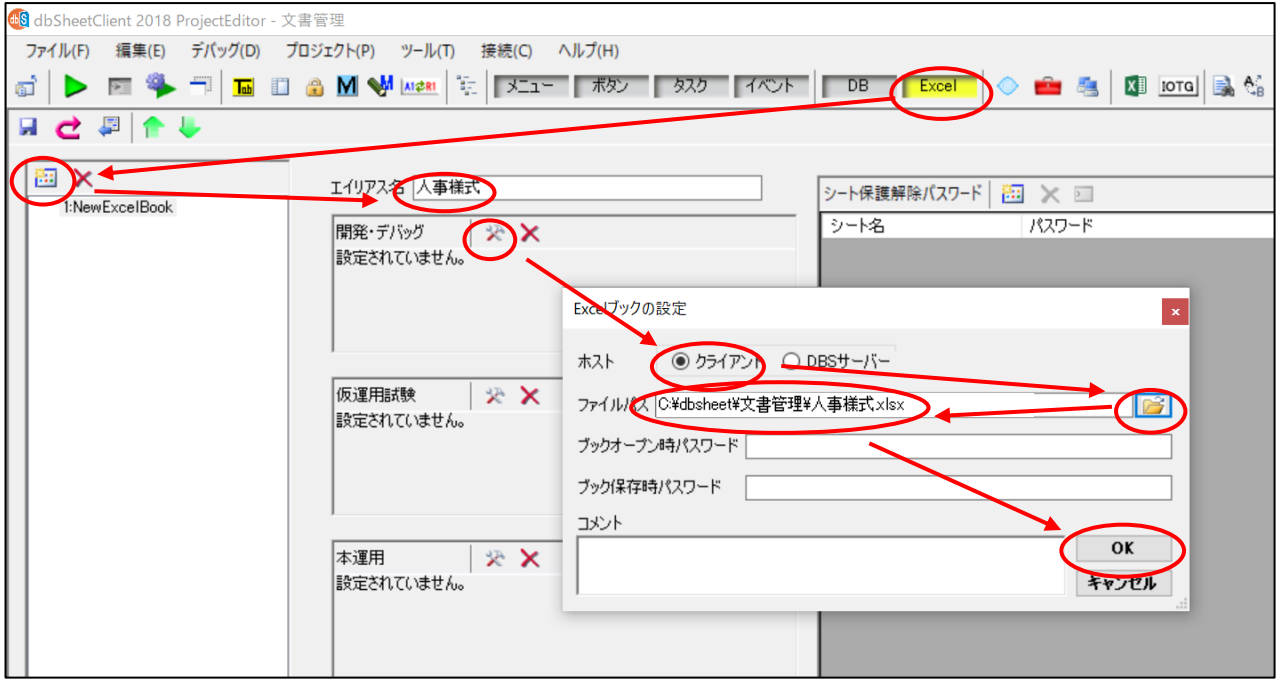


Excel ファイル設定画面にて設定します。

新規 Excel 追加ボタンからエイリアス名を「人事様式」とします。

開発・デバッグの設定ボタンをクリックして、ホストに「クライアント」を選択

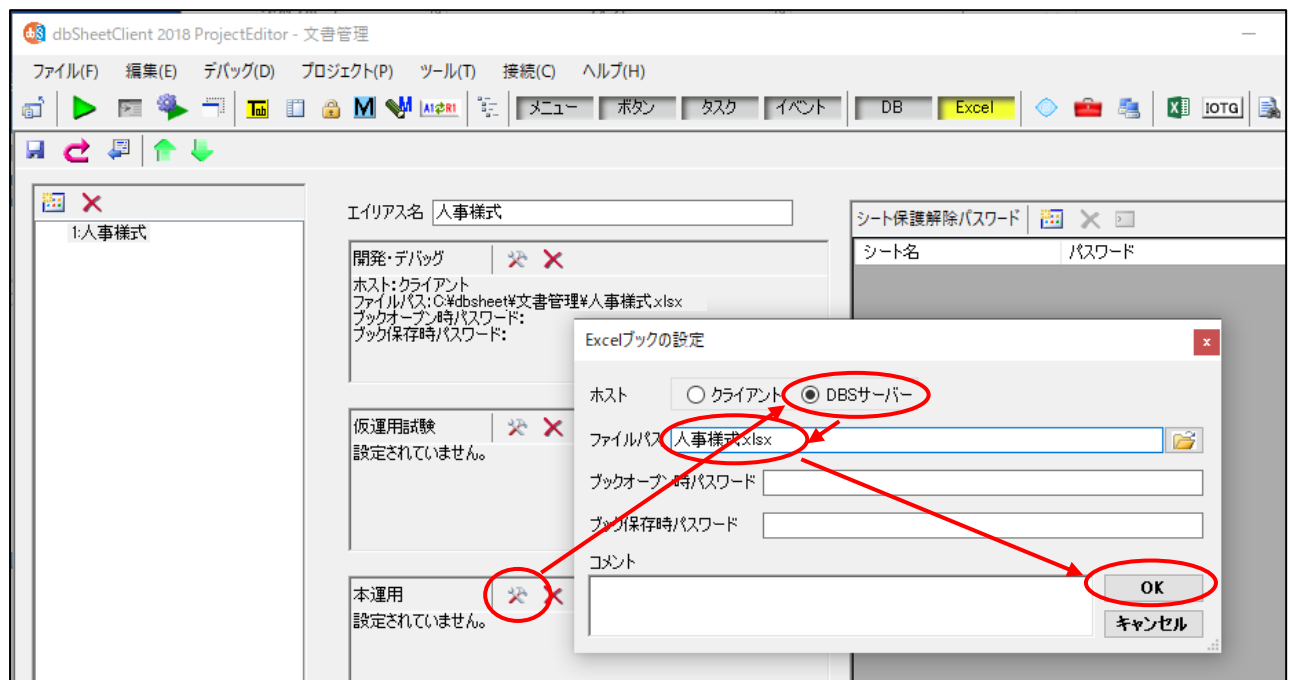
ファイル参照ボタンにて前項の「C:\dbsheet\文書管理\人事様式.xlsx」を指定します。



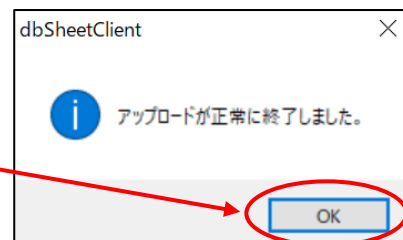
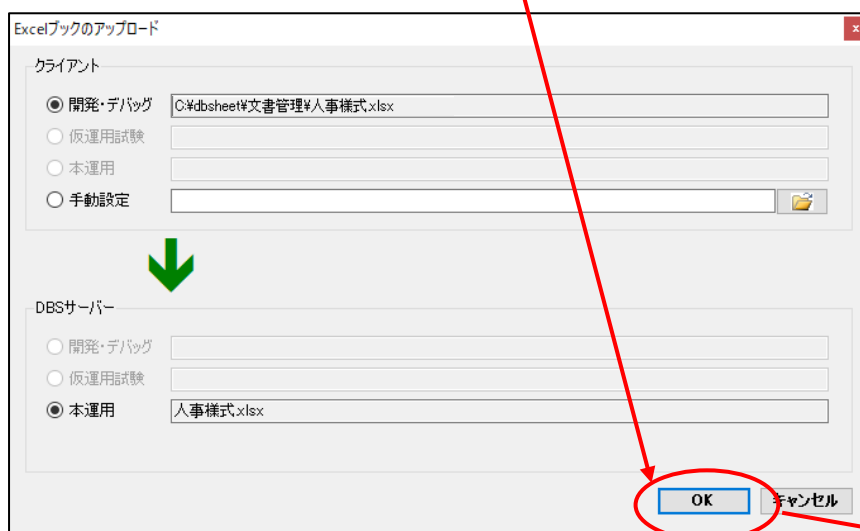
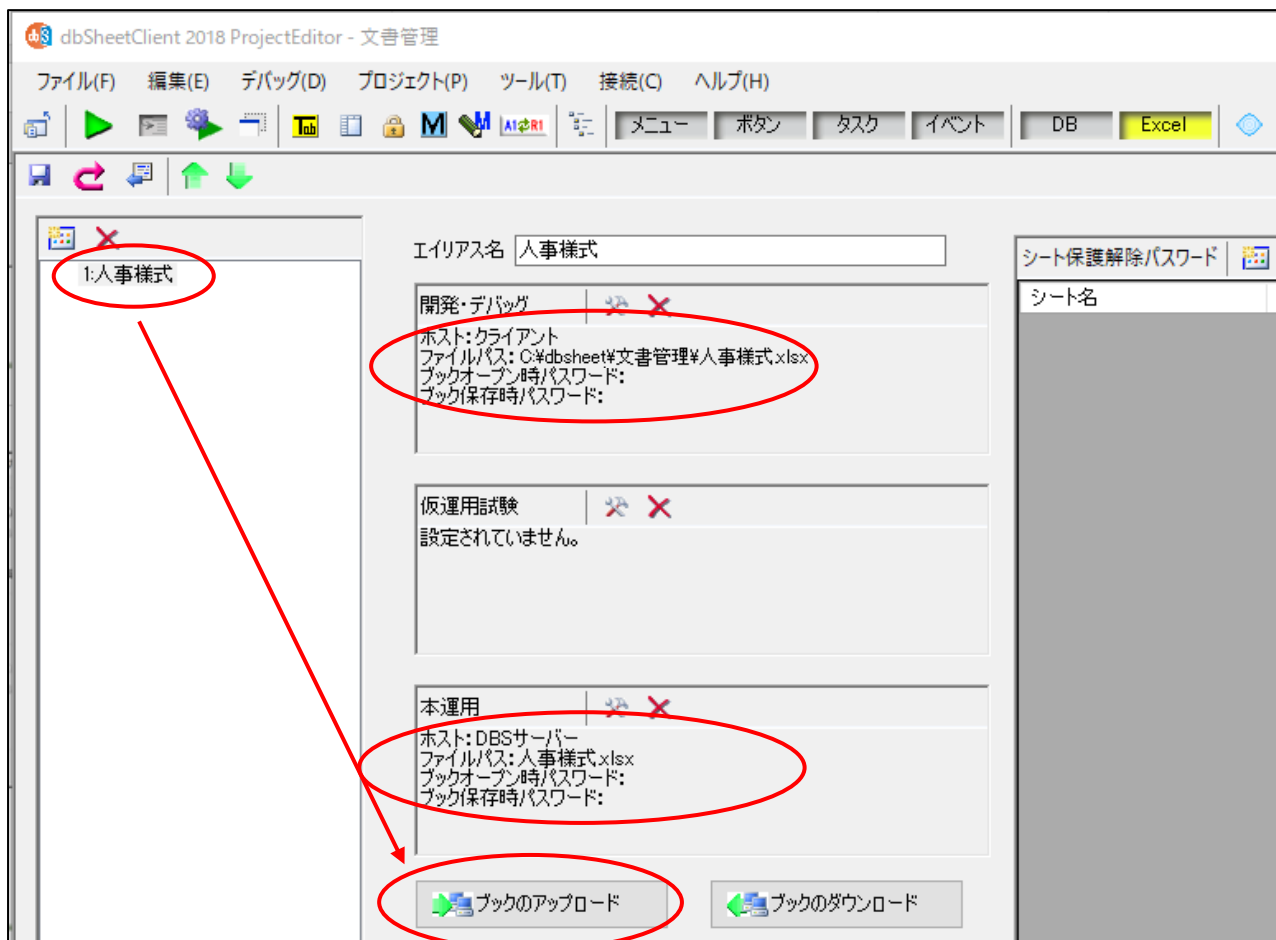
次に本運用の設定ボタンをクリックして、ホストに「DBS サーバー」を選択

ファイルパスには、フォルダは指定しないでファイル名のみ入力します。

(フォルダ指定しない場合、サーバー上のプロジェクトフォルダの初期値(サーバー依存)のフォルダに格納されます。)



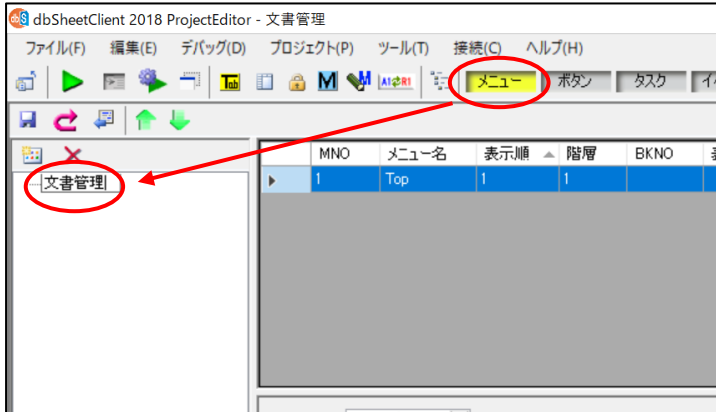
本運用環境に Excel ファイルをアップロードします。(注:開発体験版では本作業不可)



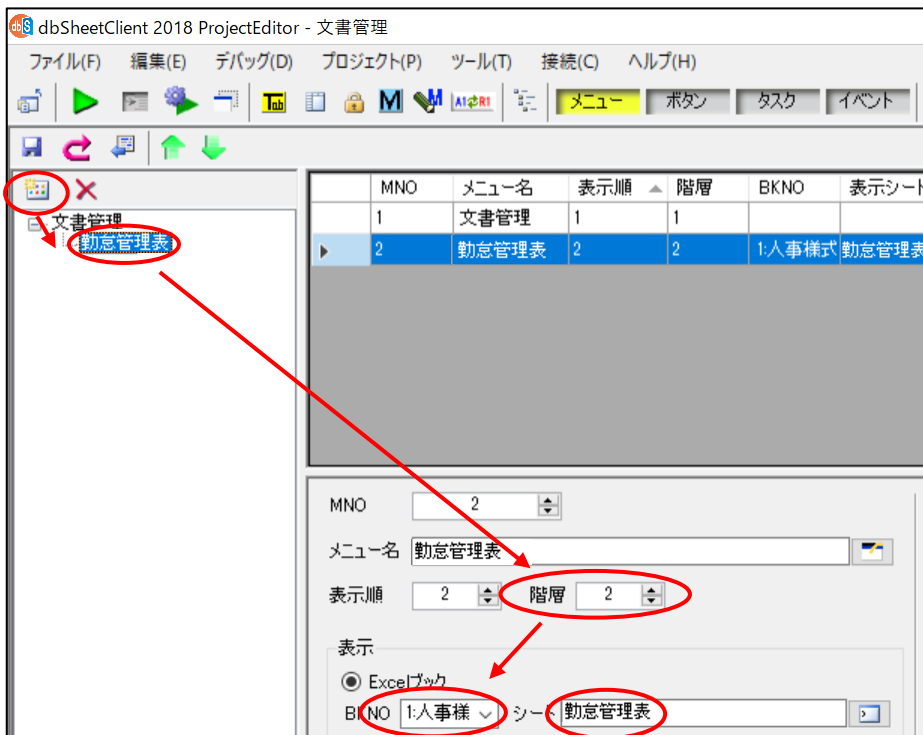
4.7 メニューの登録

4.7.1 新規メニュー登録

「メニュー」はユーザーが dbSheetClient プロジェクトを起動した時に画面左側に表示されます。
初期メニュー「top」の名称を「文書管理」に変更します。



新規メニュー作成ボタンをクリックして「勤怠管理表」メニューを追加します。
階層を「2」に変更、Excel ブックを「人事様式」、シートを「勤怠管理表」に変更



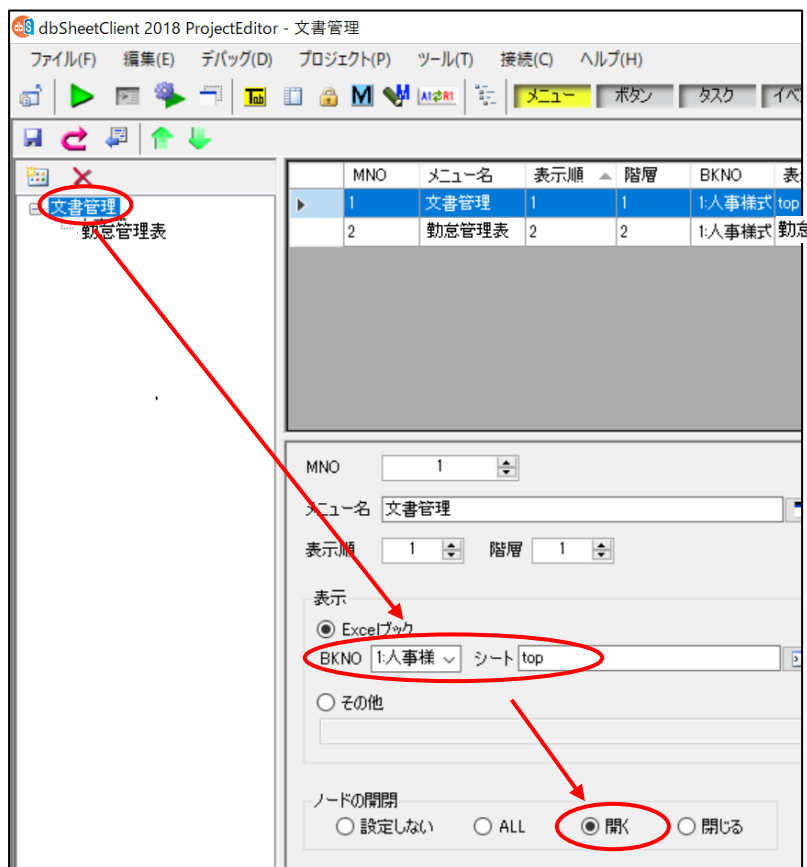
4.7.2 文書管理メニューの設定

「文書管理」メニューを選択して、ノードの開閉を「開く」に変更します。

→ プロジェクト起動時に階層下の「勤怠管理表」メニューが折りたたまれずに表示されます。

Excel ブックを「人事様式」、シートを「top」に変更

→ メニュー「文書管理」が選択された時に表示するブックとシートになります。



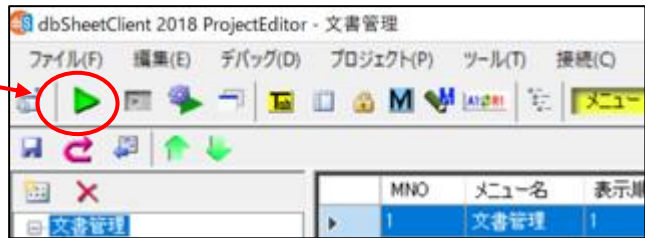
シート「top」には、文書の改訂履歴や利用手順を記載しておくが良いです。

	1	2	3	4
1	R1.2			
2		改訂履歴		
3				
4		日付	改訂内容	
5		2019/9/1	勤怠管理表、休暇届、給与振込依頼書、交通費精算書を追加	
6		2020/4/13	印刷範囲の修正（勤怠管理表、休暇届）	
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				

Sheet tabs at the bottom: top | 条件 | 勤怠管理表 | 休暇届 | 給与振込依頼書 | 交通費精算書

4.7.3 デバッグモードで実行

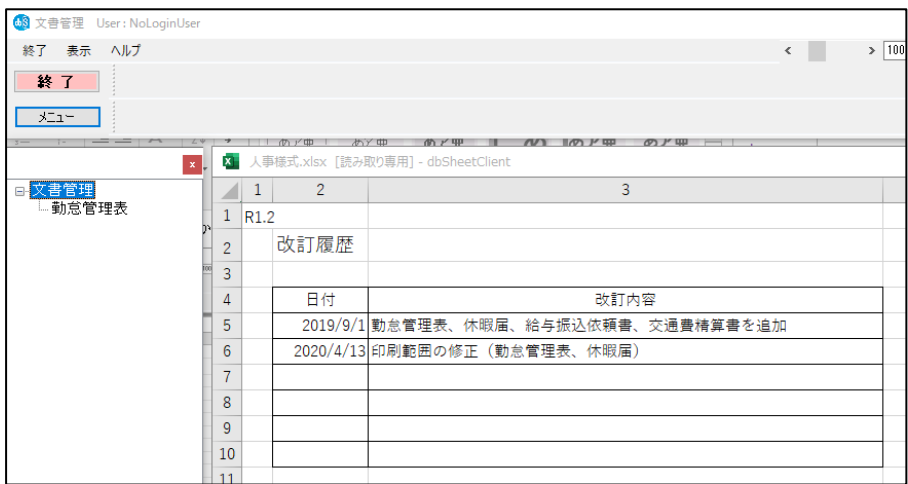
デバッグボタンをクリック



メニュー表示を確認

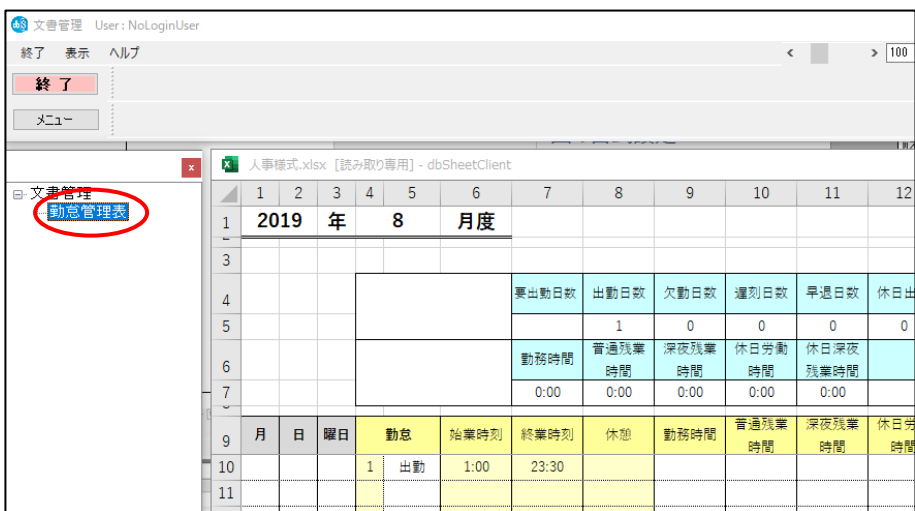
→ 「勤怠管理表」メニューが「文書管理」メニューの階層にあって表示されているか
初期シートの表示を確認

→ 「人事様式.xlsx」のシート「top」が画面右の Excel エリアに表示されているか



「勤怠管理表」メニューの選択で「勤怠管理表」シートの表示を確認

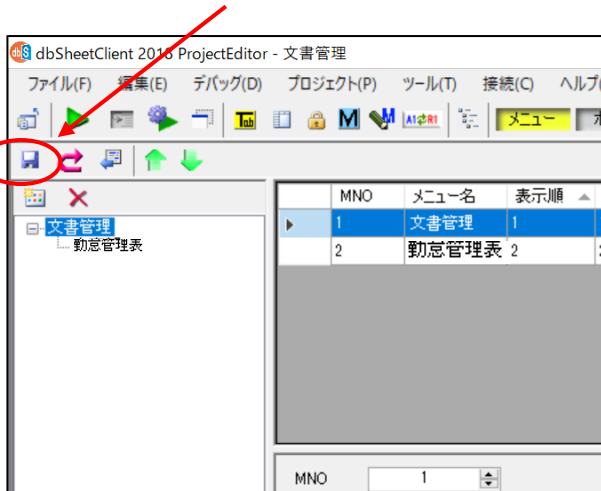
→ 「人事様式.xlsx」のシート「勤怠管理表」が画面右の Excel エリアに表示されているか



画面左上の「終了」ボタンにてデバッグモードを終了します。

4.7.4 プロジェクトを保存

ここまででいったんプロジェクトを保存します。

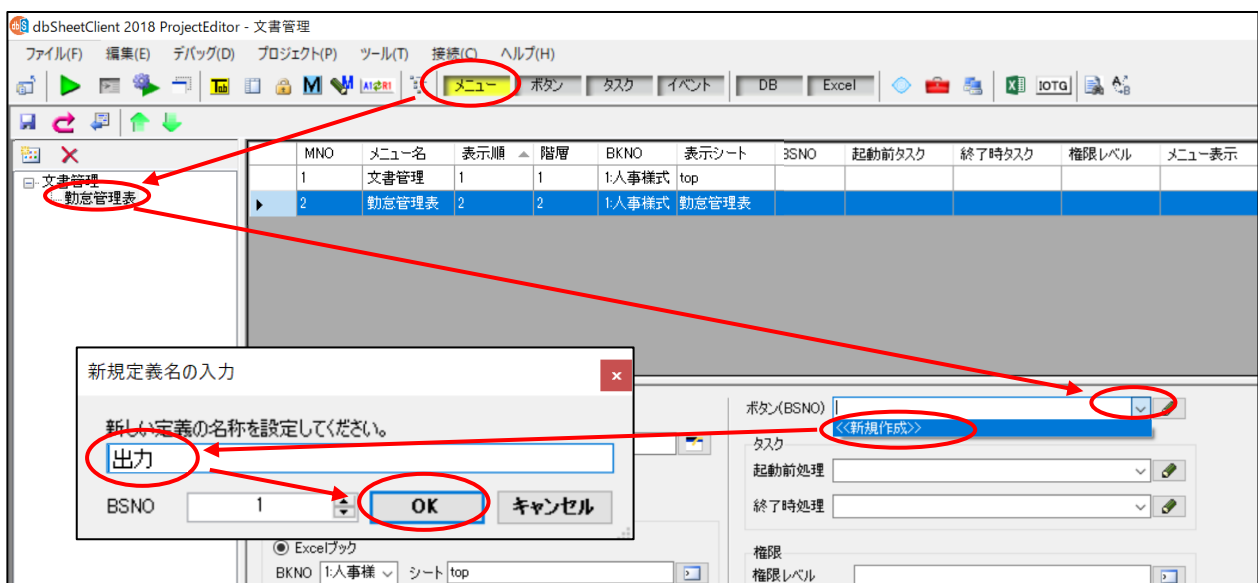


4.8 ボタンを設置

4.8.1 メニューにボタンセットを割り付ける

「ボタン」は dbSheetClient プロジェクトを起動した時に Excel 画面の上部に表示されます。

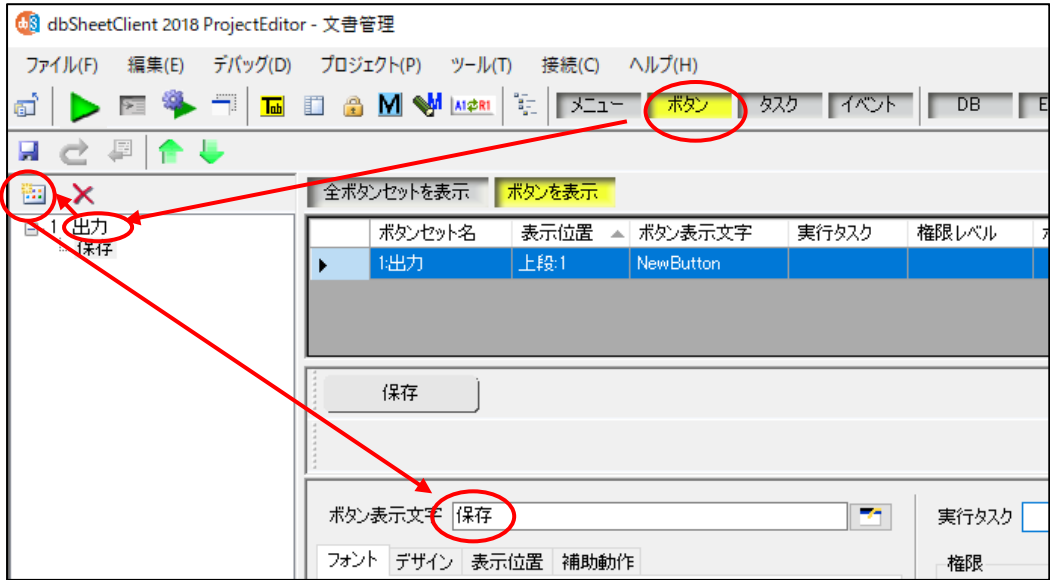
「勤怠管理表」メニューにボタンセット「出力」を設定します。



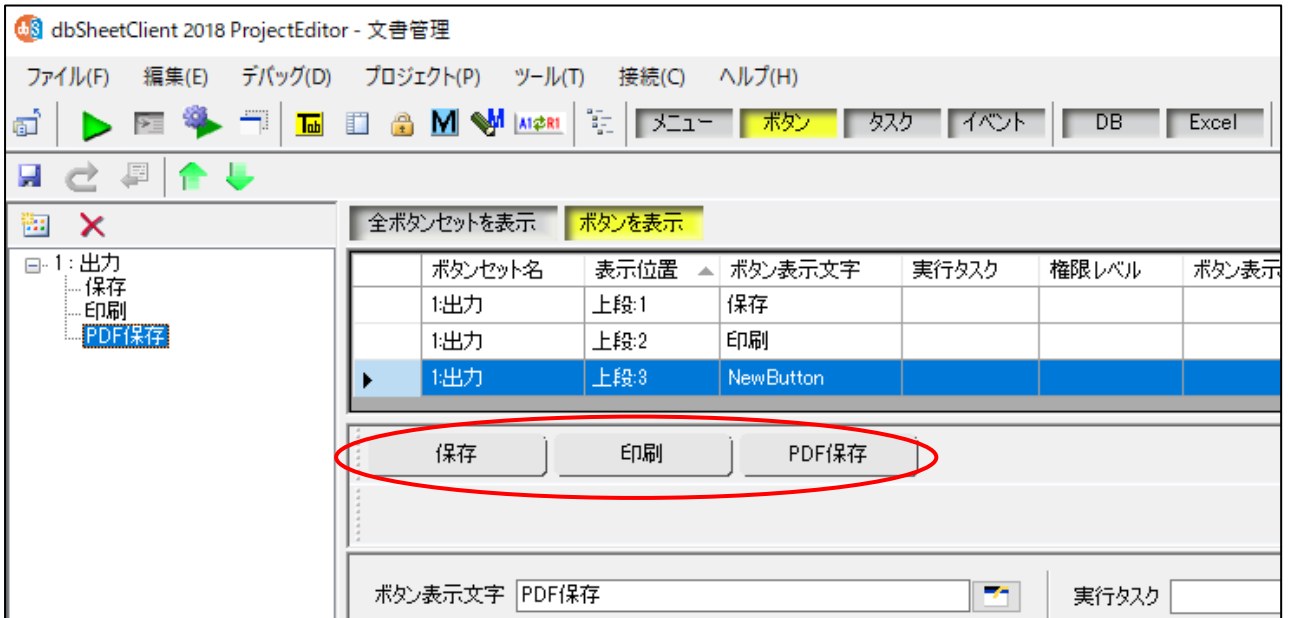
4.8.2 新規ボタンを登録

作成したボタンセット「出力」にボタンを登録します。

新規ボタン作成ボタンをクリックして「勤怠管理表」ボタンを追加します。



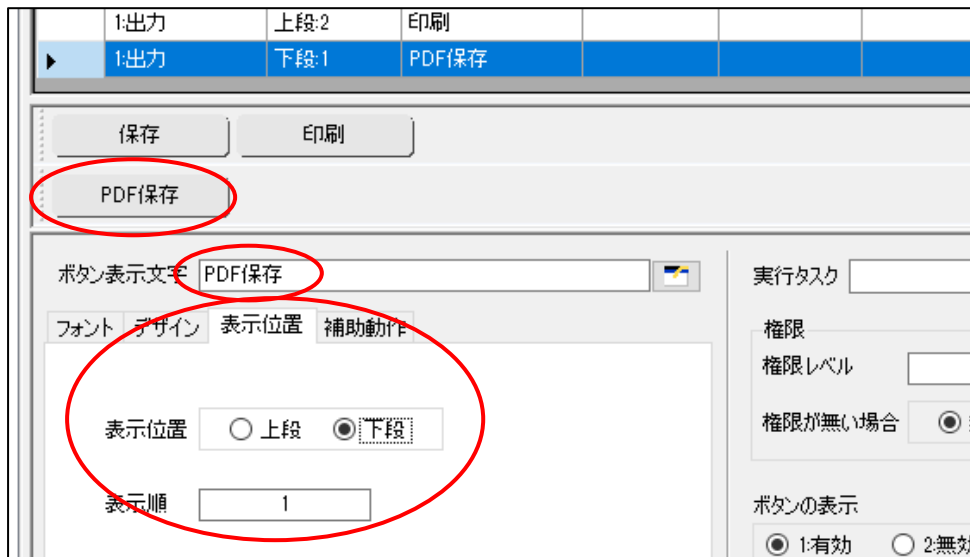
同様に「印刷」「PDF 保存」ボタンを登録します。



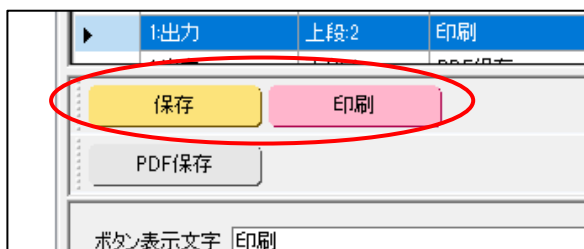
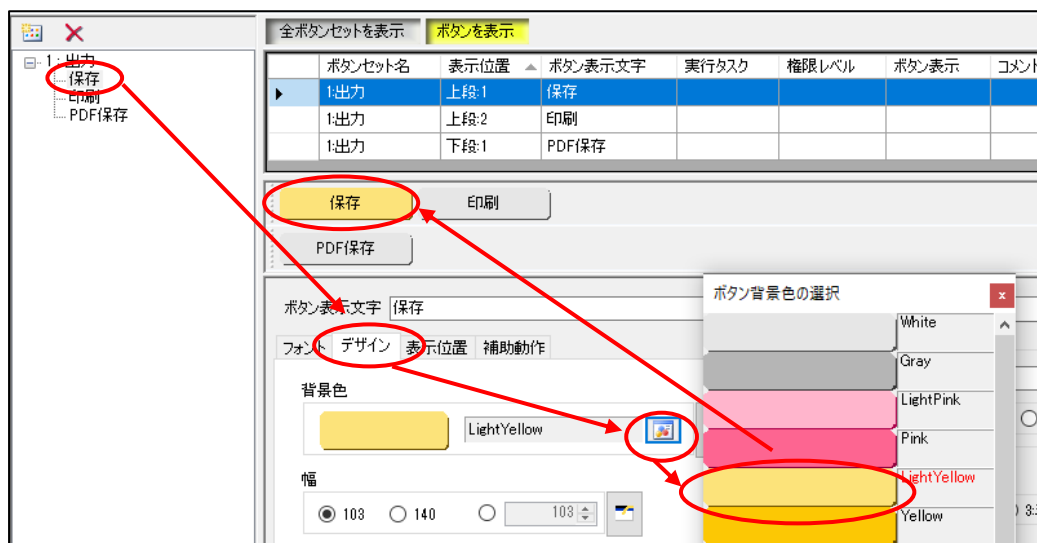
ボタンの配置を変更します。

ボタンをマウスで直接ドラッグすると表示位置を変更できます。

表示位置タブから変更することもできます。

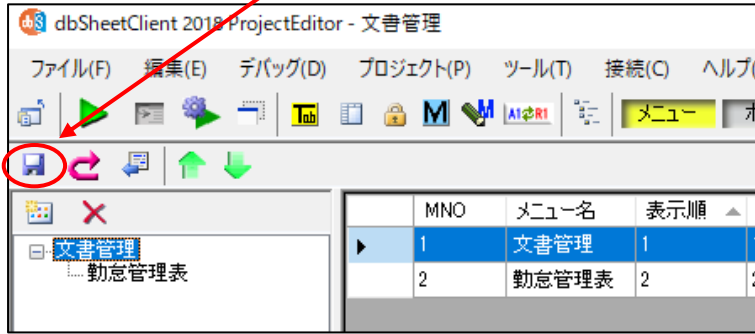


ボタンに色を付けてみます。



4.8.3 プロジェクトを保存

ここまででいったんプロジェクトを保存します。



4.9 タスクの登録

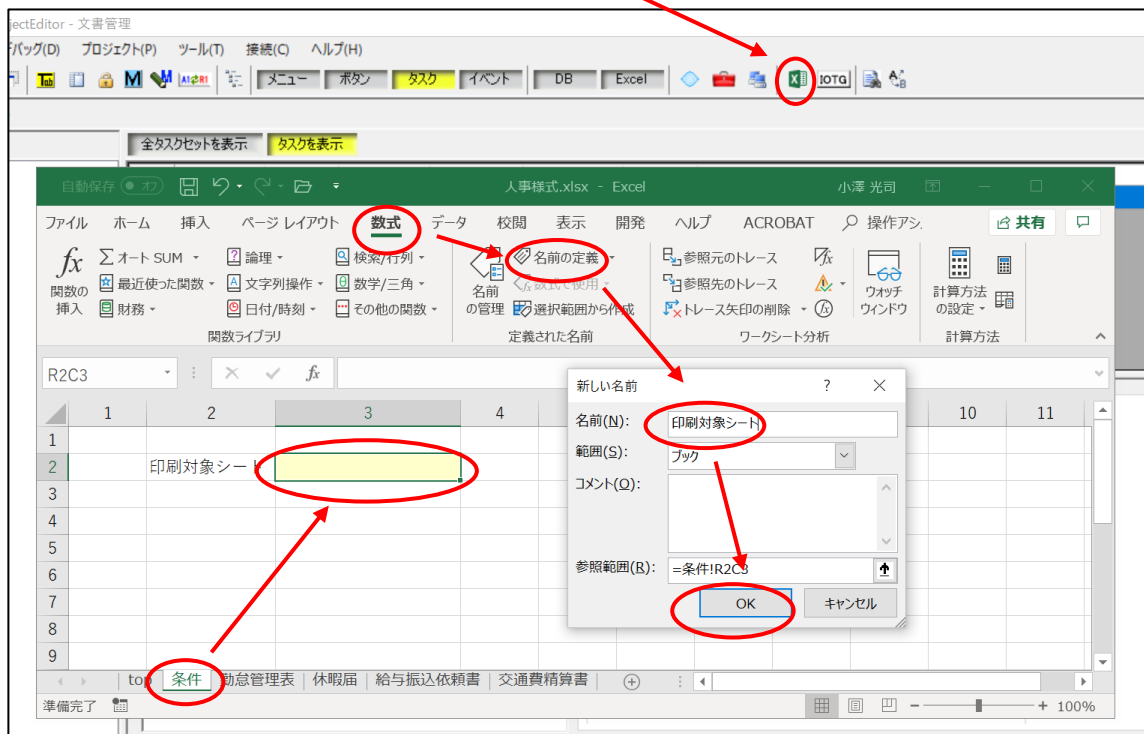
4.9.1 印刷ボタンの工夫

印刷ボタンには「印刷」タスクを割り付けます。

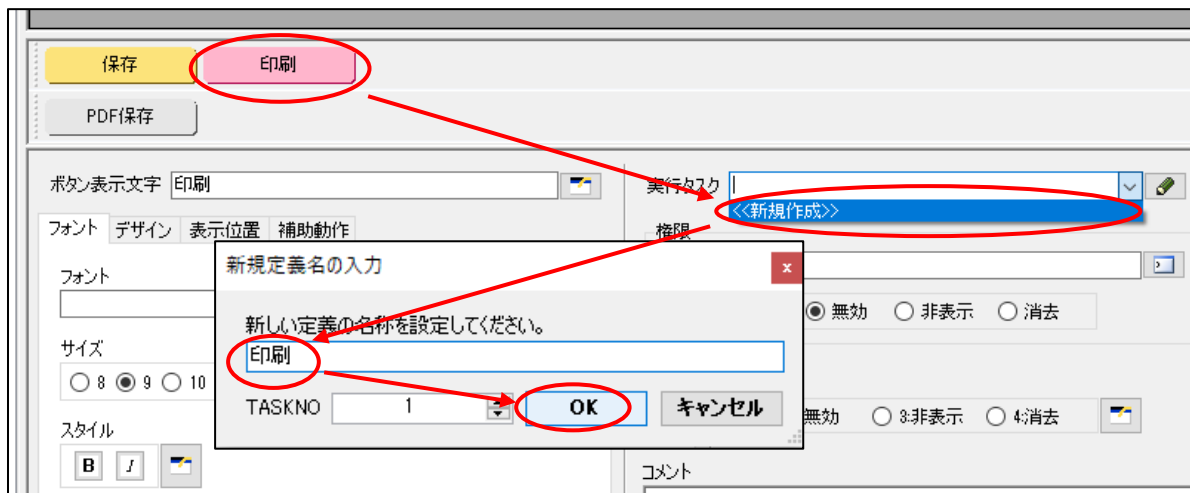
複数の様式に対して一つの「印刷」ボタンで対応できるようタスクを工夫します。

Excel ボタンで Excel ファイルを呼び出します。

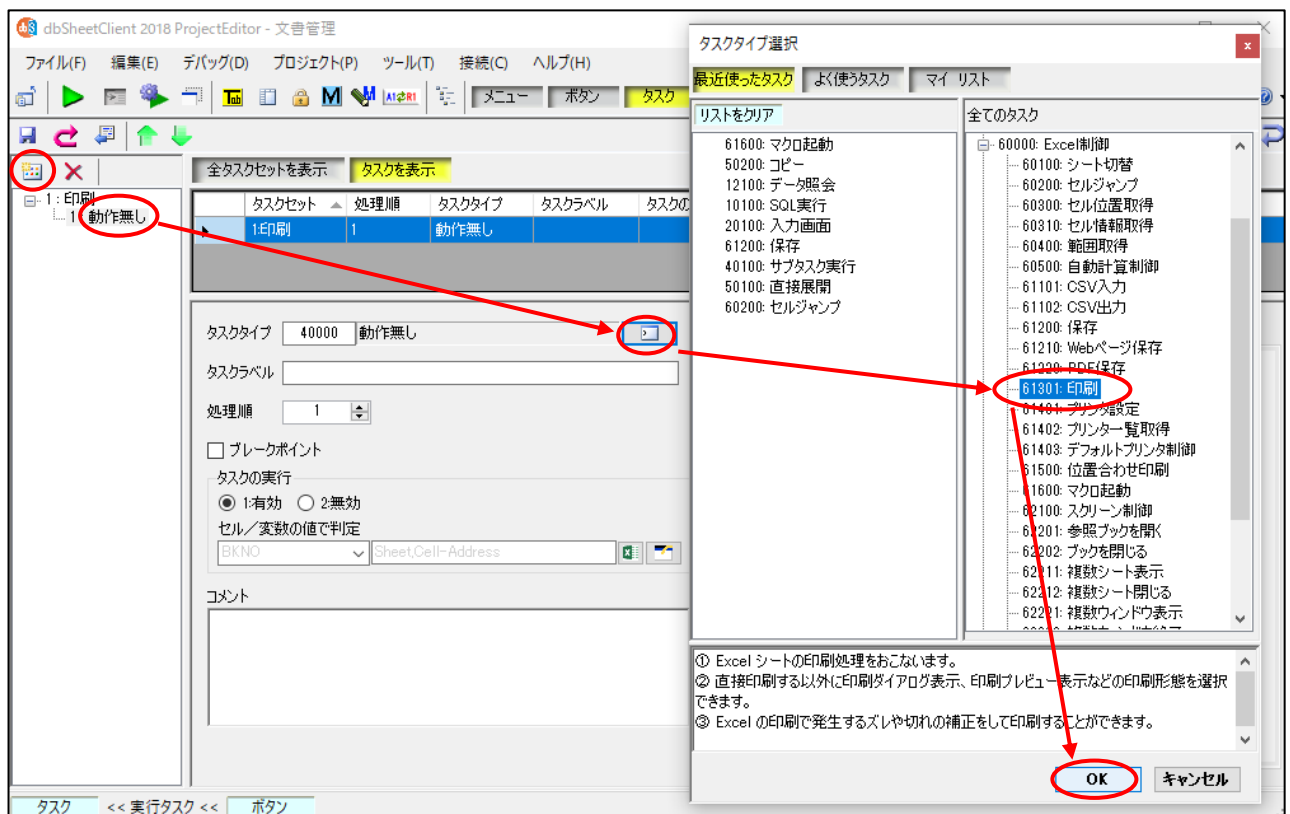
「条件」シートに「印刷対象シート」という名前を定義します。



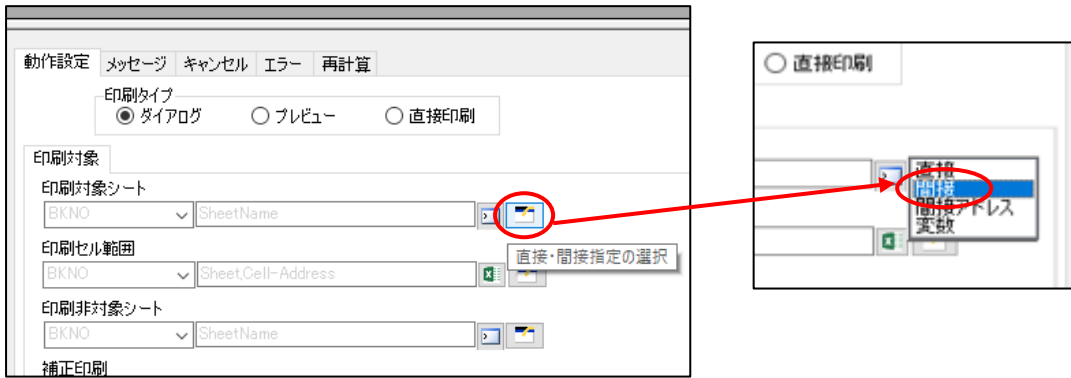
作成した「印刷」ボタンを選択してタスクを割り付けます (タスク名は「印刷」)



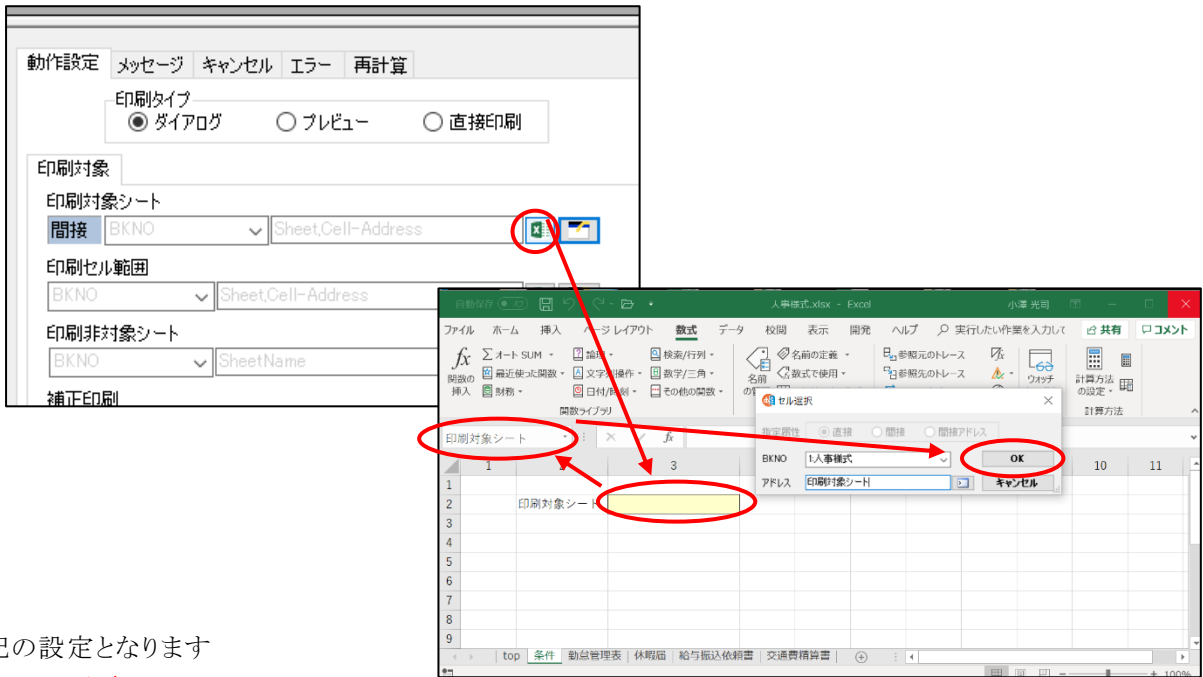
「印刷」タスク (タスクタイプ:61301) をセットします。



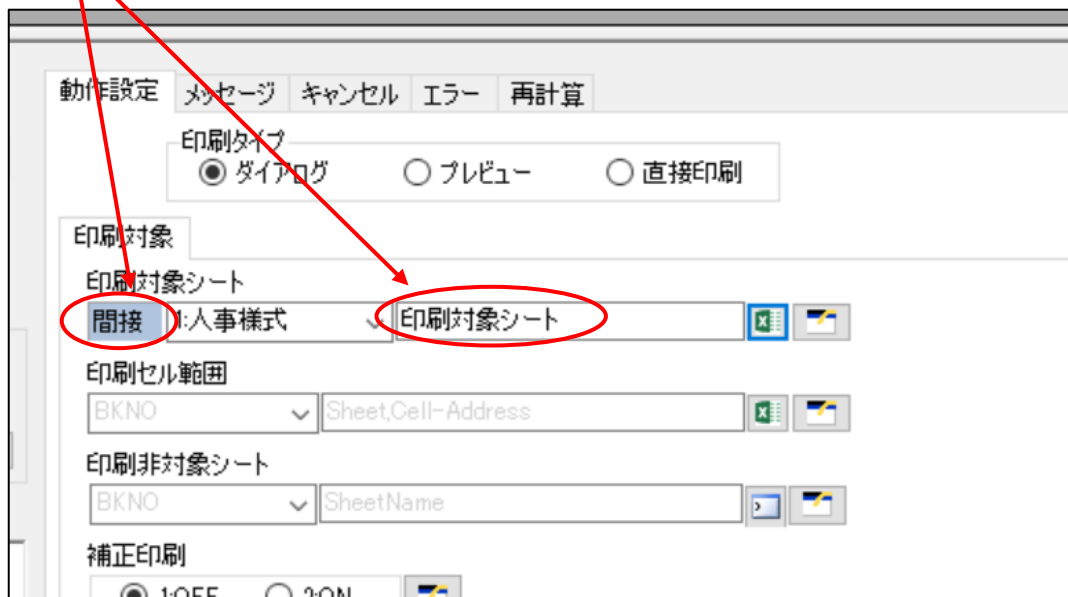
印刷対象シートの指定にて「間接」を選択



→ 自動表示する Excel から条件シートの「印刷対象シート」セルを指定します

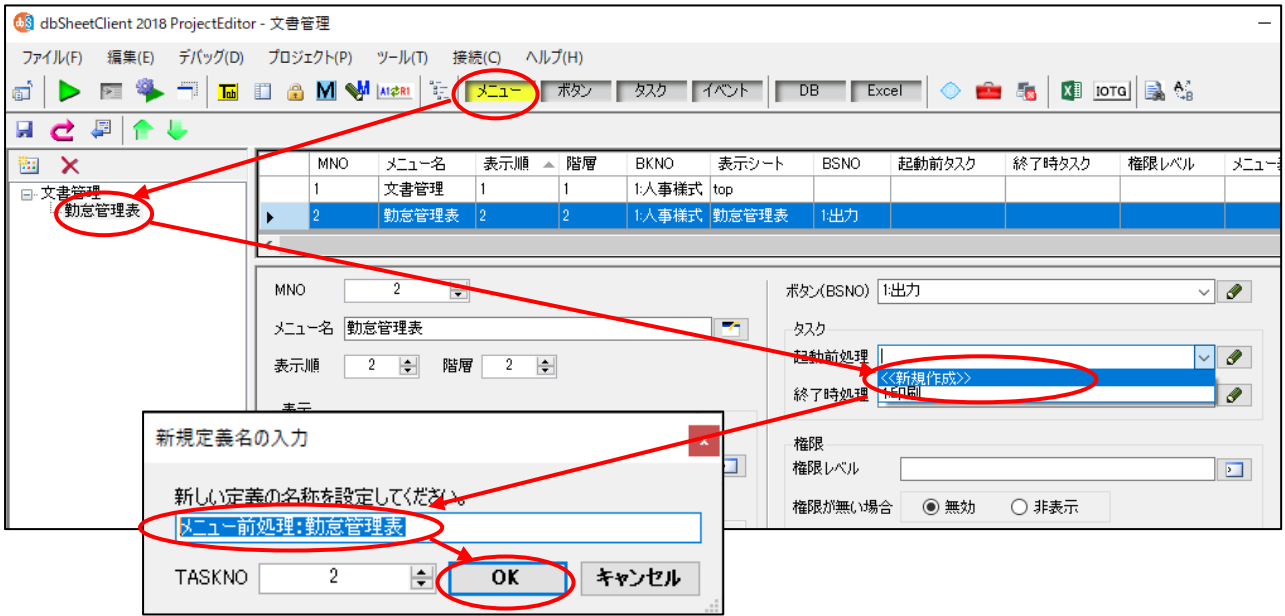


下記の設定となります

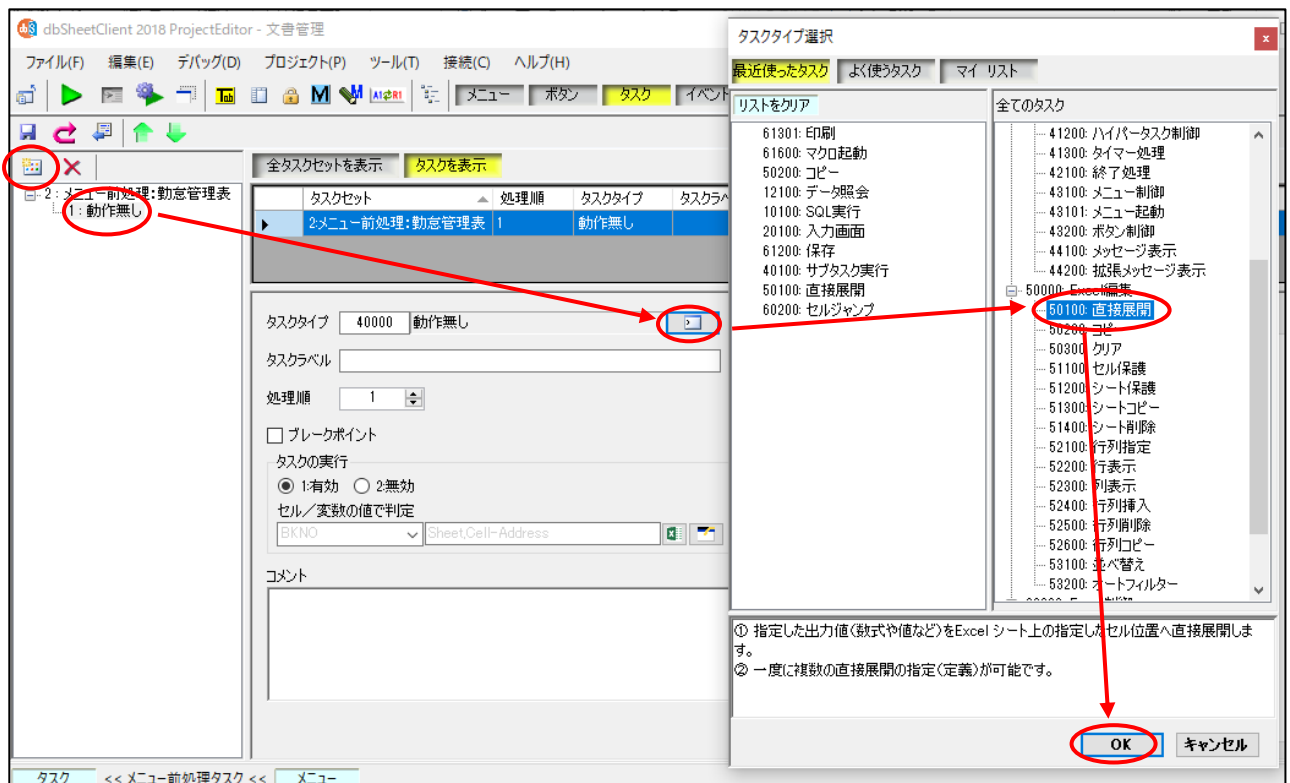


4.9.2 メニュー勤怠管理表の起動前処理にタスクをセット

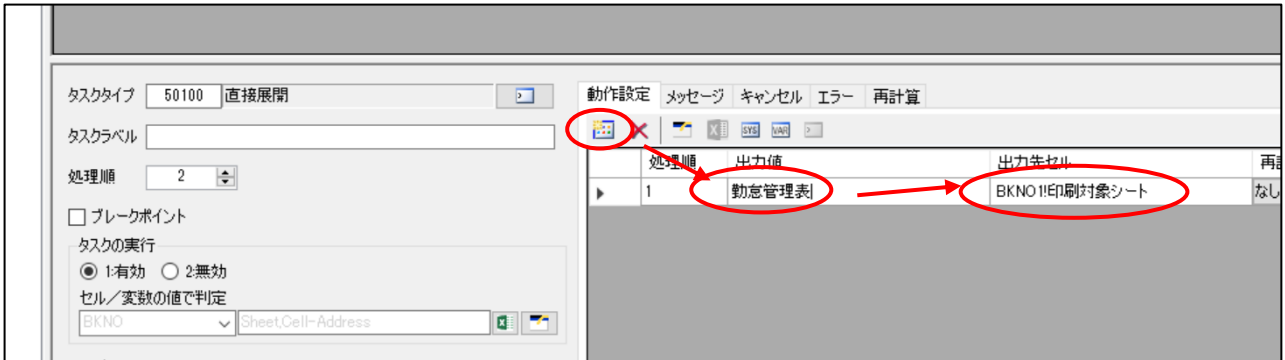
作成した「勤怠管理表」メニューの起動前処理にタスクを割り付けます。(タスク名は「メニュー前処理:勤怠管理表」)



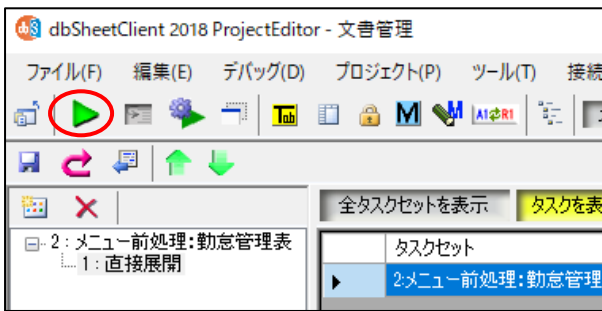
「直接展開」タスク(タスクタイプ:50100)をセットします。



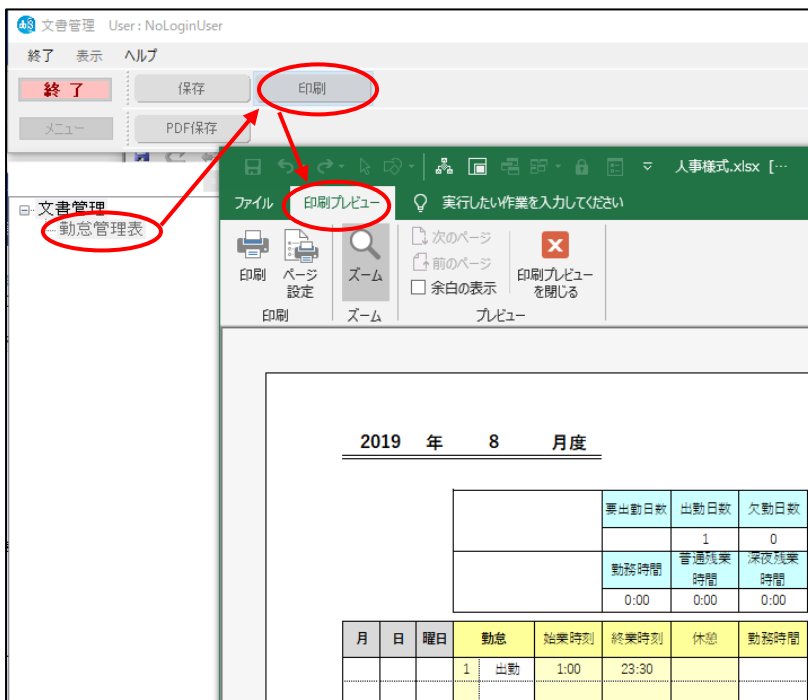
出力値 → 「勤怠管理表」、出力先セル → 条件シートの「印刷対象シート」セル



デバッグモードで実行



左メニューより「勤怠管理表」メニューを選択して様式「勤怠管理表」が表示された後、「印刷」ボタンをクリックして「印刷プレビュー」した時に様式「勤怠管理表」が印刷対象になっていることを確認します。



「印刷プレビューを閉じる」後、画面左上の「終了」ボタンにてデバッグモードを終了します。

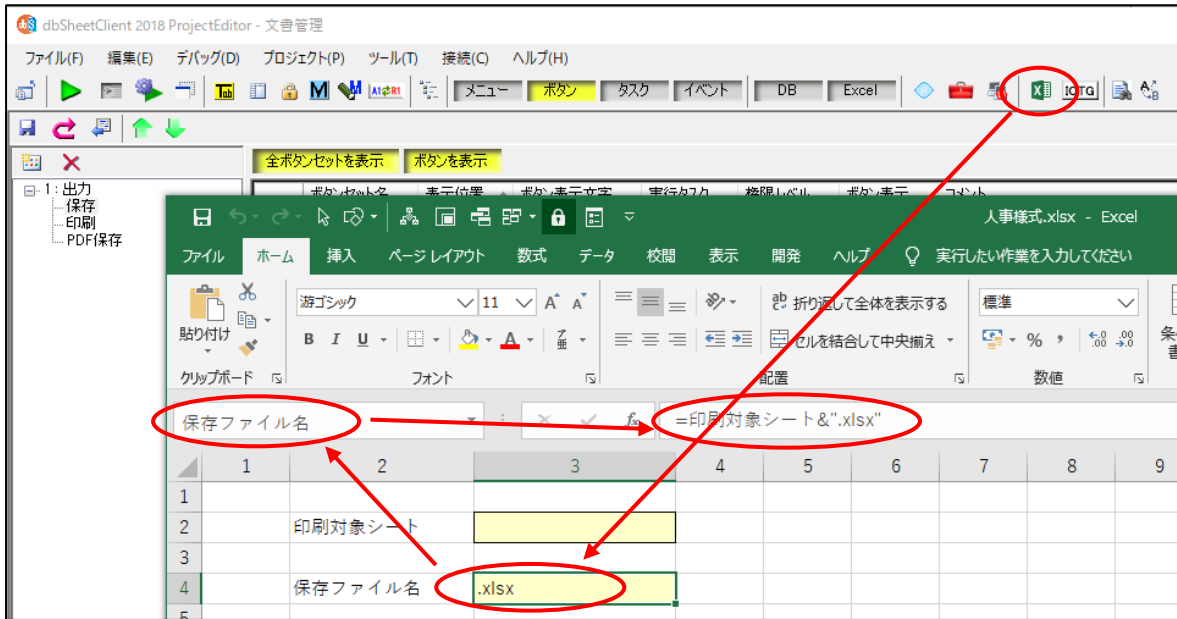
4.9.3 保存ボタンの設定

保存ボタンに「保存」タスクを割り付けます。

印刷ボタンと同様に複数の様式に対して1つの「保存」ボタンで対応できるよう工夫します。

Excel ボタンでシートを呼び出し、「条件」シートに「保存ファイル名」という名前を定義します。

「保存ファイル名」セルには、関数式「=印刷対象シート&".xlsx"」をセットします。

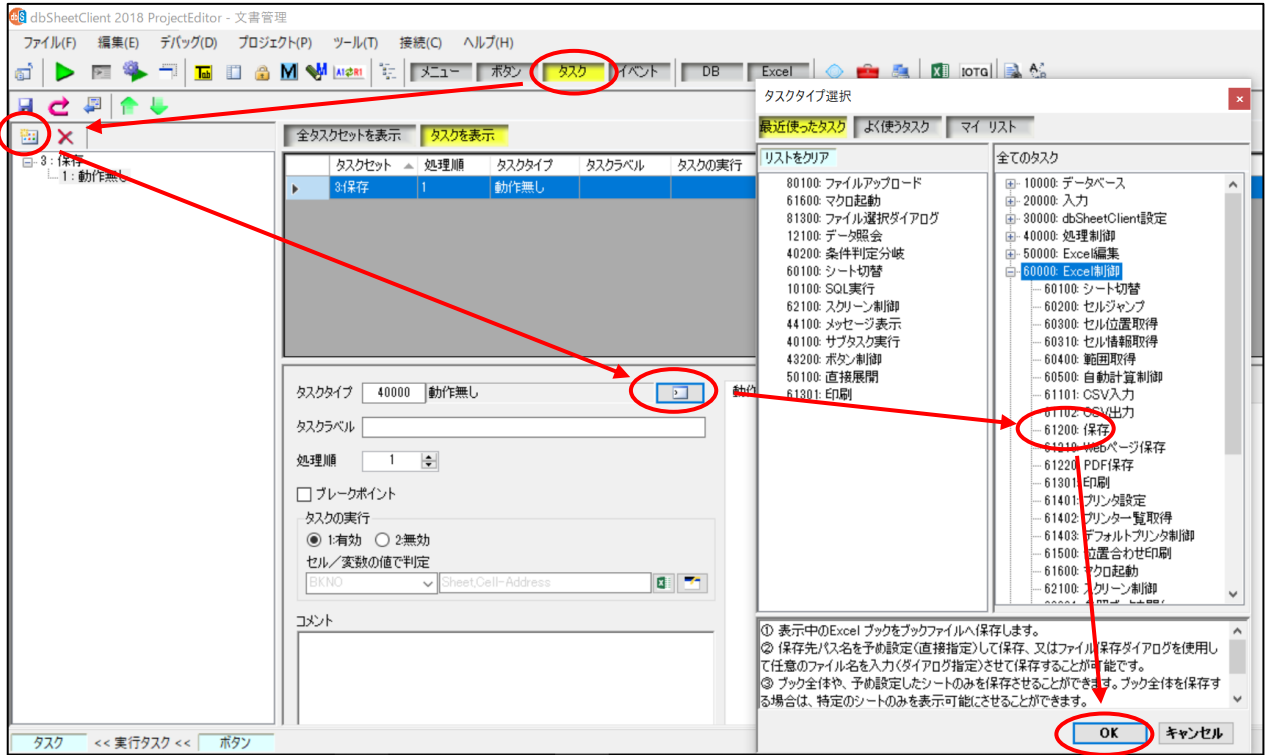


保存ボタンに「保存」という名前前でタスクセットを定義します。



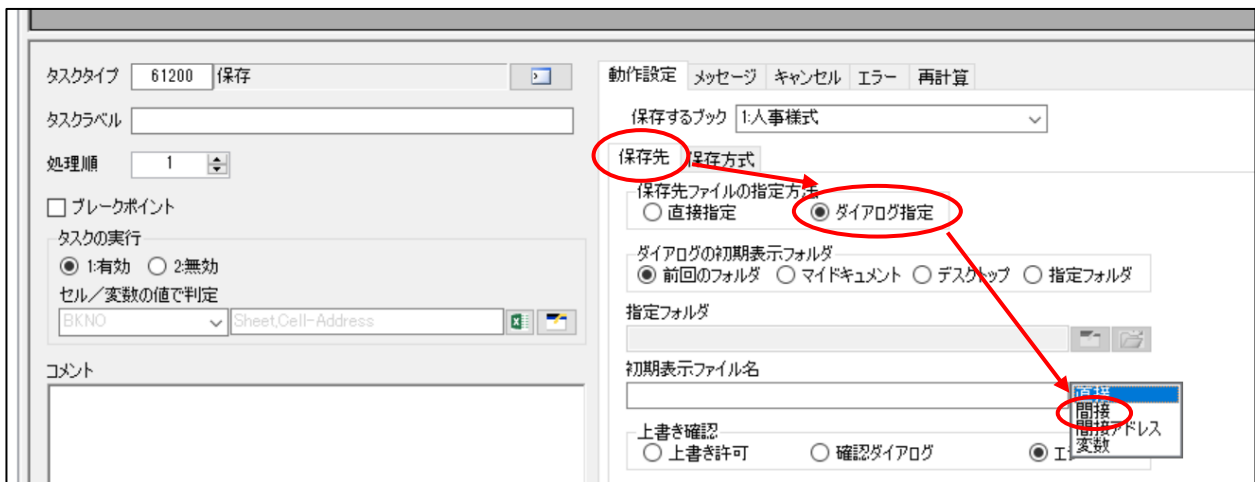
作成したタスクセット「保存」にタスクをセットします。

新規タスクボタンをクリックして「保存」タスク(タスクタイプ:61200)を追加します。

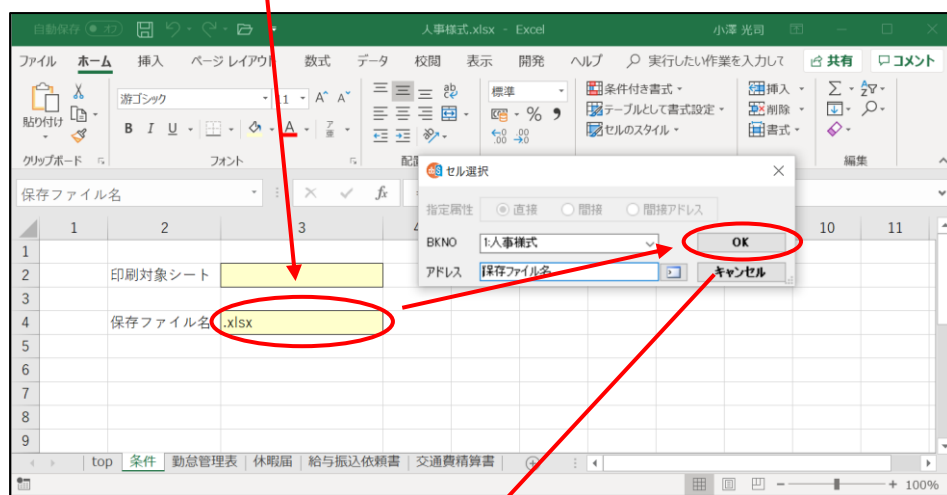
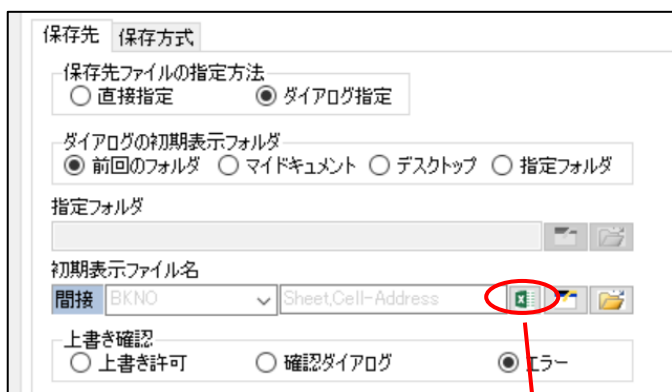


「保存」タスクの保存先の設定を行います。

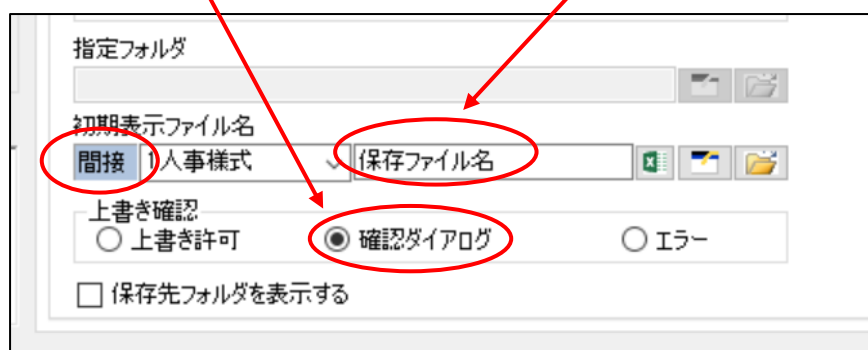
初期表示ファイル名の設定にて「間接」を選択



初期表示ファイル名を条件シートの「保存ファイル名」に設定します。



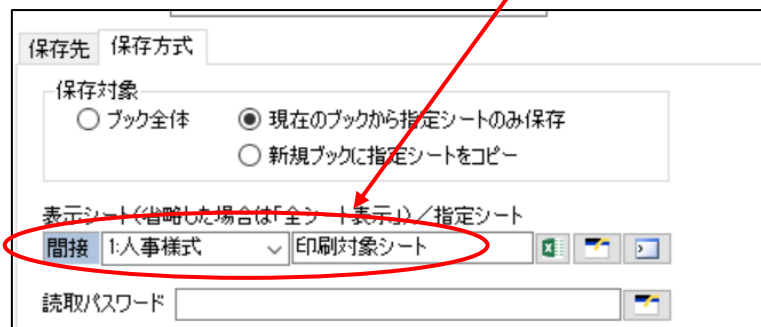
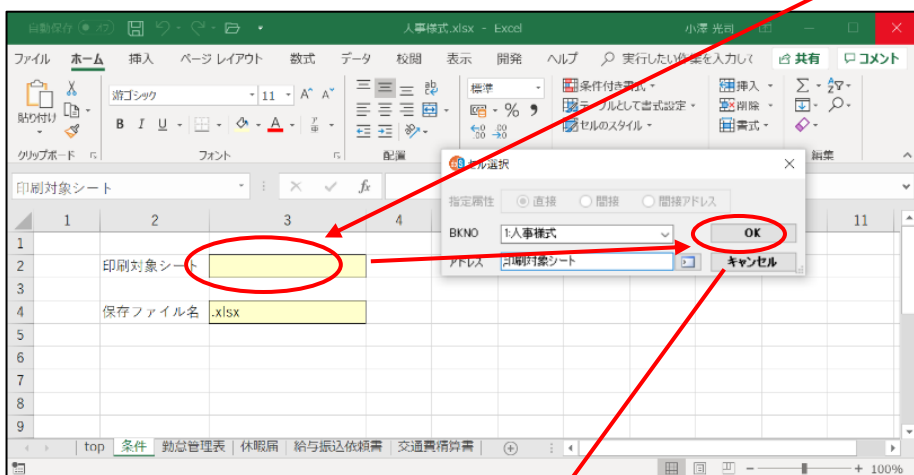
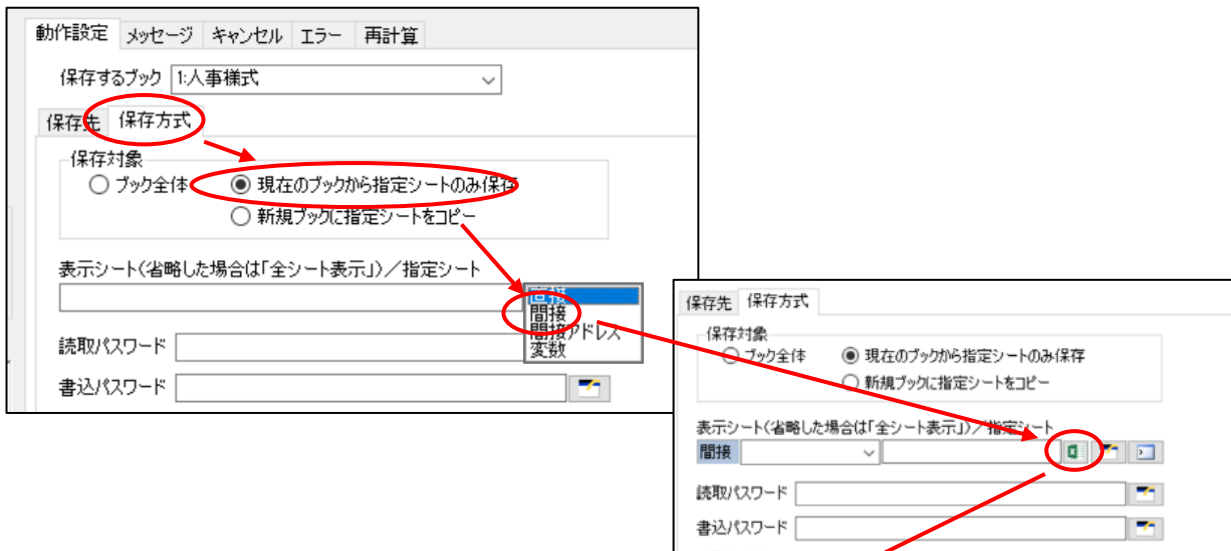
上書き確認は「確認ダイアログ」にします。



保存方式を設定します。

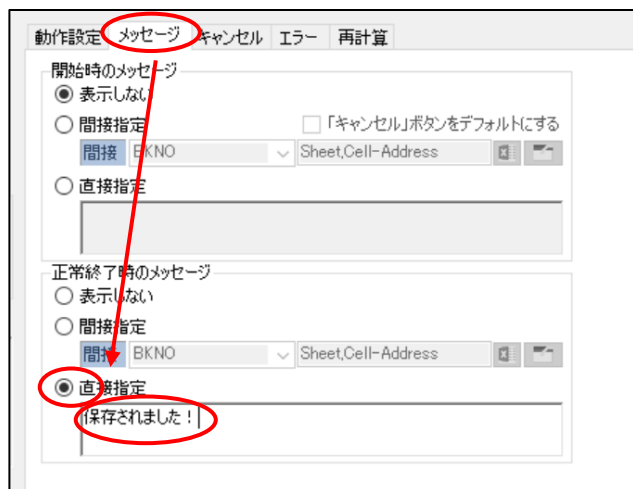
印刷対象シートを条件シートの「印刷対象シート」に設定します。

表示シートから「間接」を選択します。



メッセージ設定を行います。

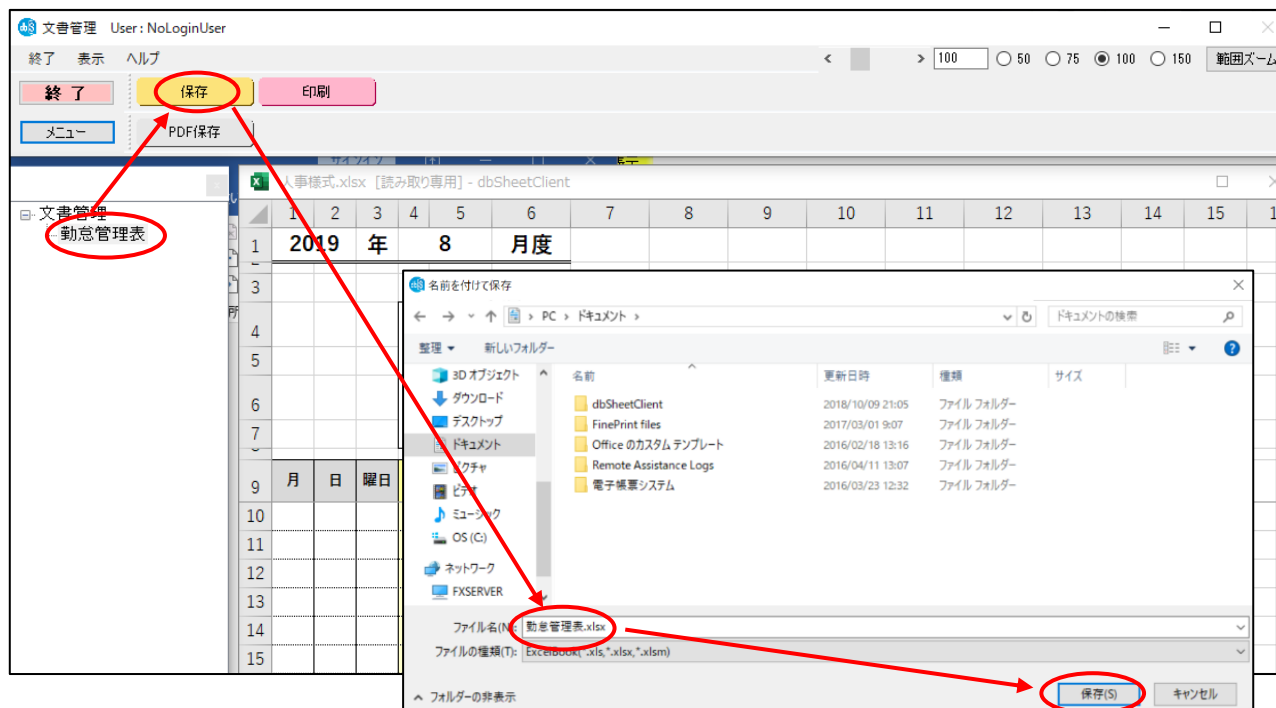
保存が完了した時に「保存されました！」と表示



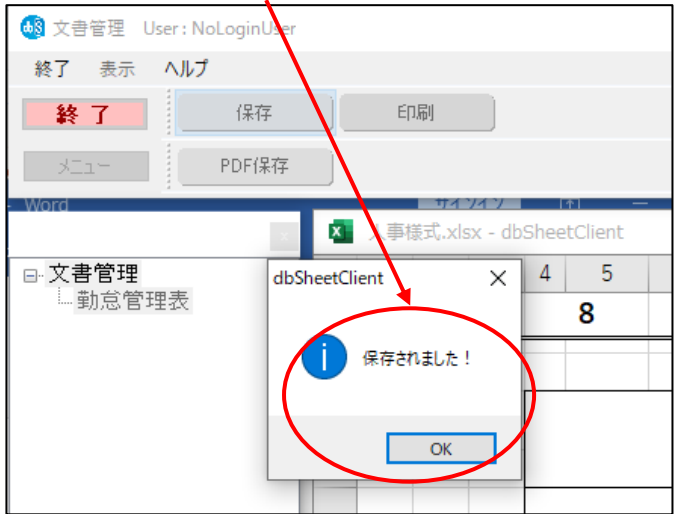
デバッグモードで実行

左メニューより「勤怠管理表」メニューを選択して様式「勤怠管理表」が表示された後、「保存」ボタンをクリックします。

保存ファイル名が「勤怠管理表.xlsx」になっていることを確認し保存します。



保存が完了した時に「保存されました！」と表示されることを確認します。



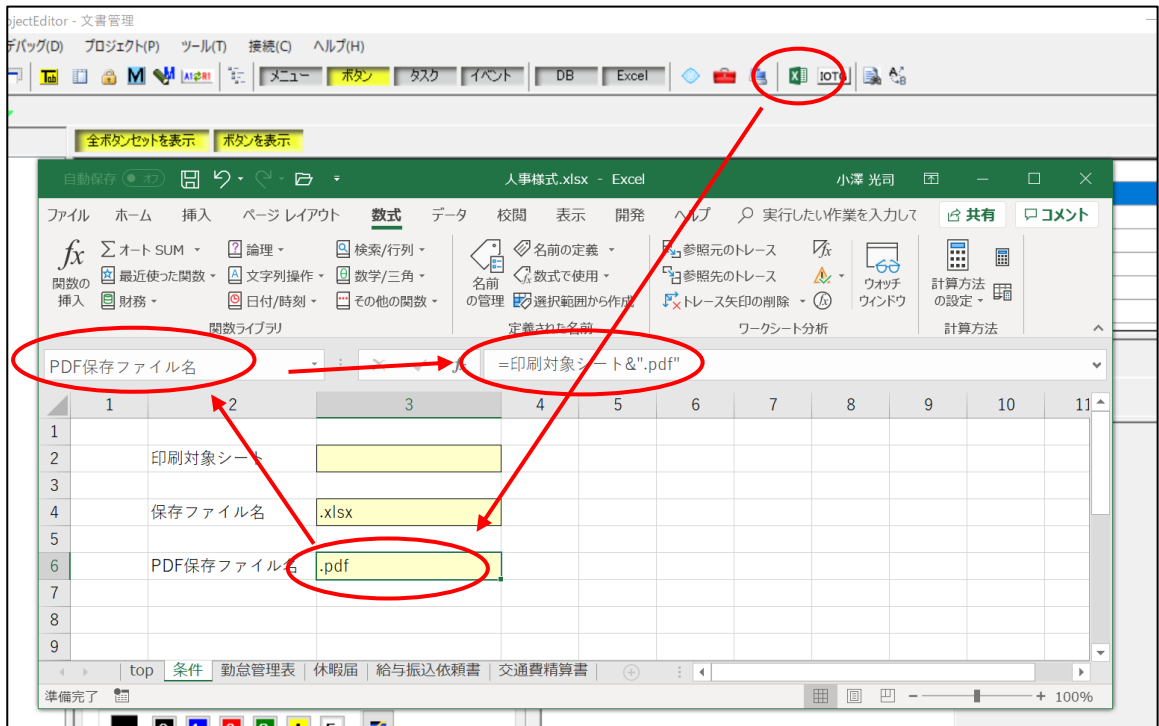
4.9.4 PDF ファイル保存ボタン

PDF 保存ボタンに「PDF 保存」タスクを割り付けます。

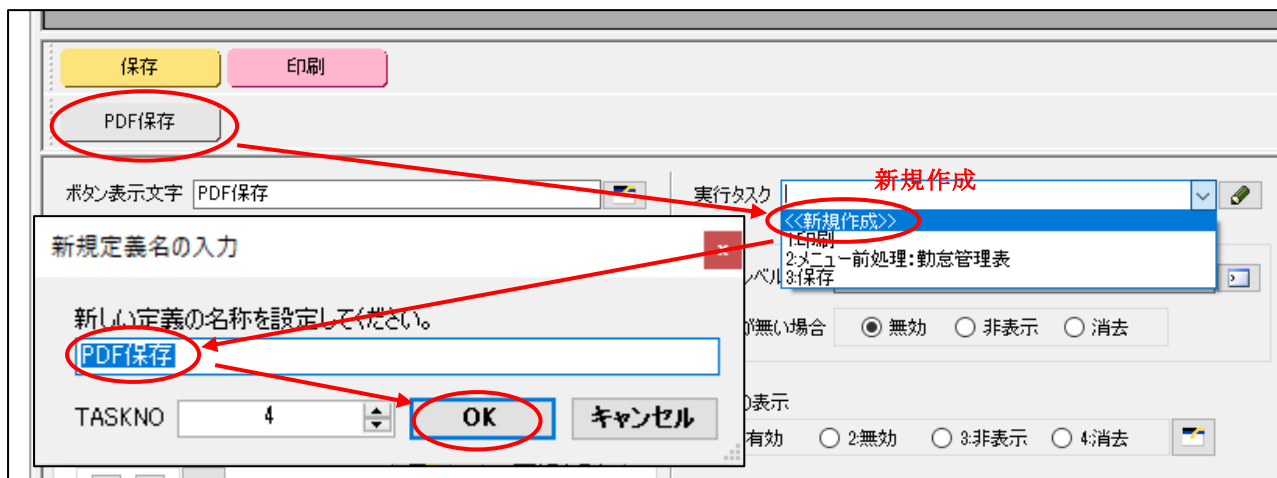
「印刷」ボタン、「保存」ボタンと同様の設定を行います。

Excel ファイルを呼び出し、「条件」シートに「PDF 保存ファイル名」という名前を定義

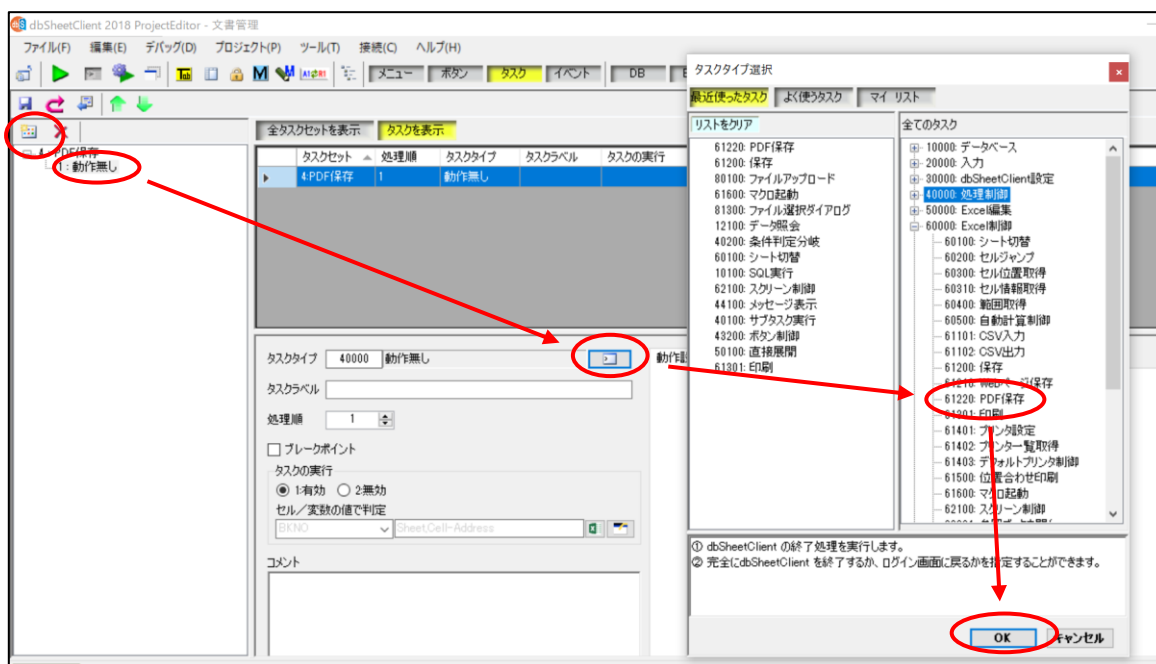
セルに関数式 「=印刷対象シート&".pdf"」をセットします。



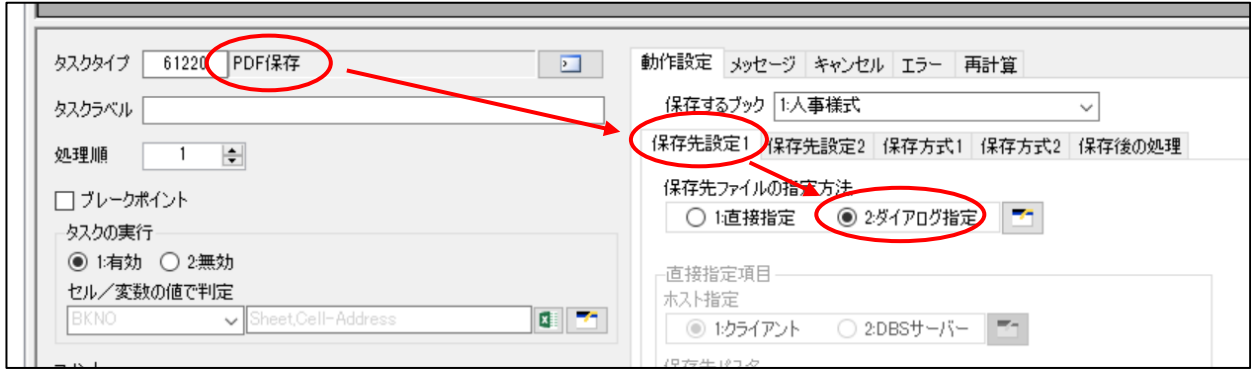
「PDF 保存」ボタンを選択して「PDF 保存」という名前でタスクセットを設定します。



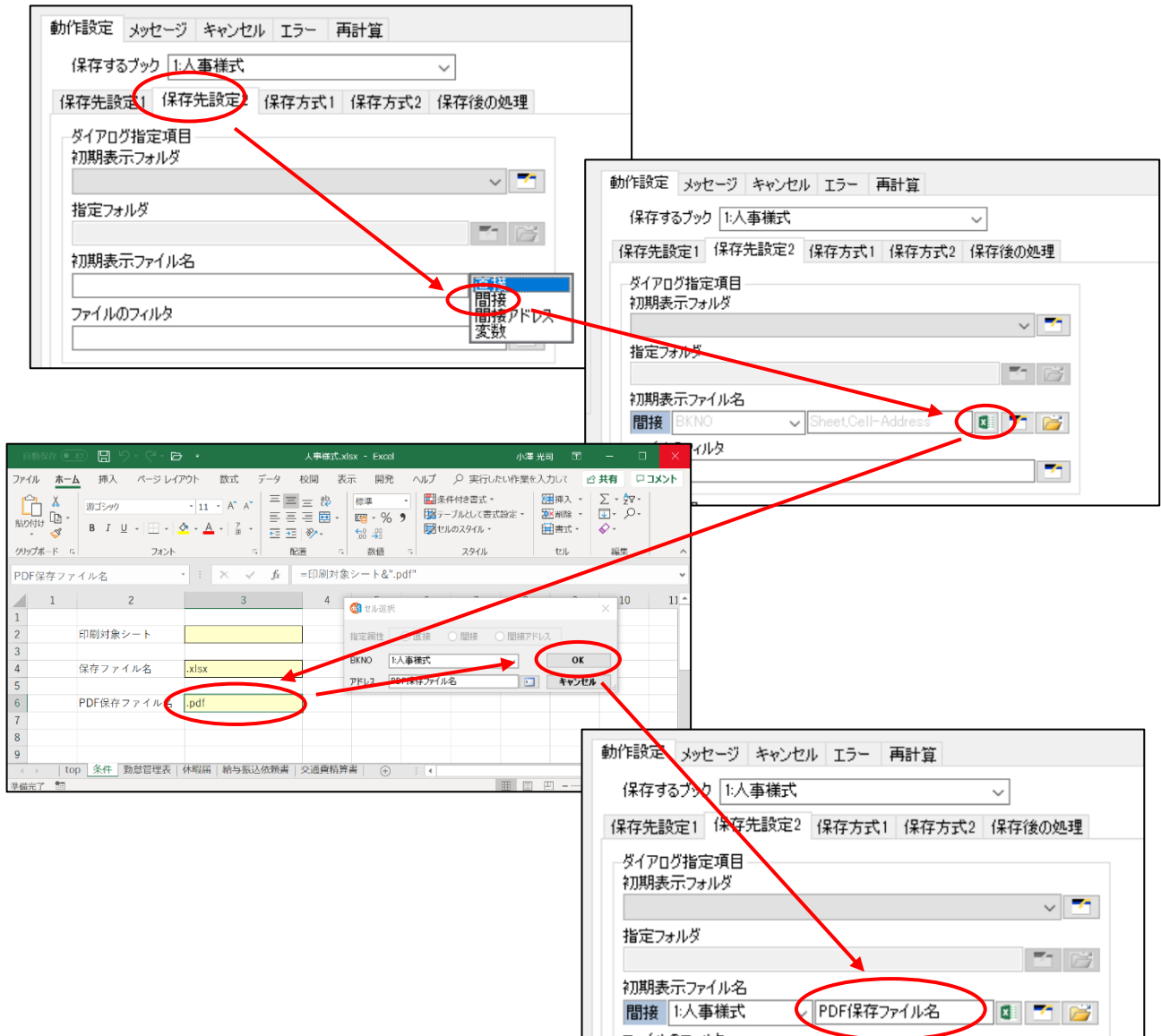
タスクセット「PDF 保存」に「PDF 保存」タスク(タスクタイプ:61220)をセットします。



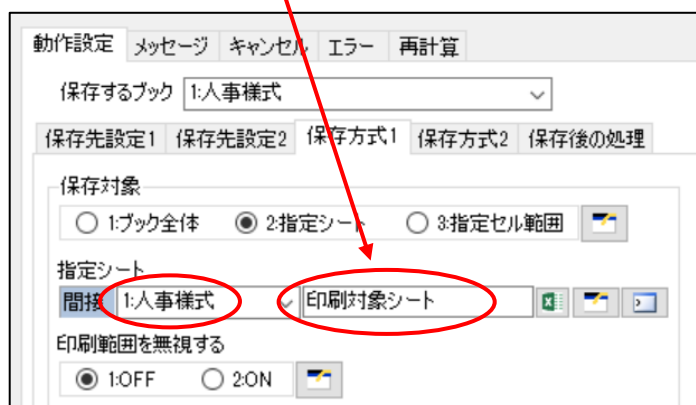
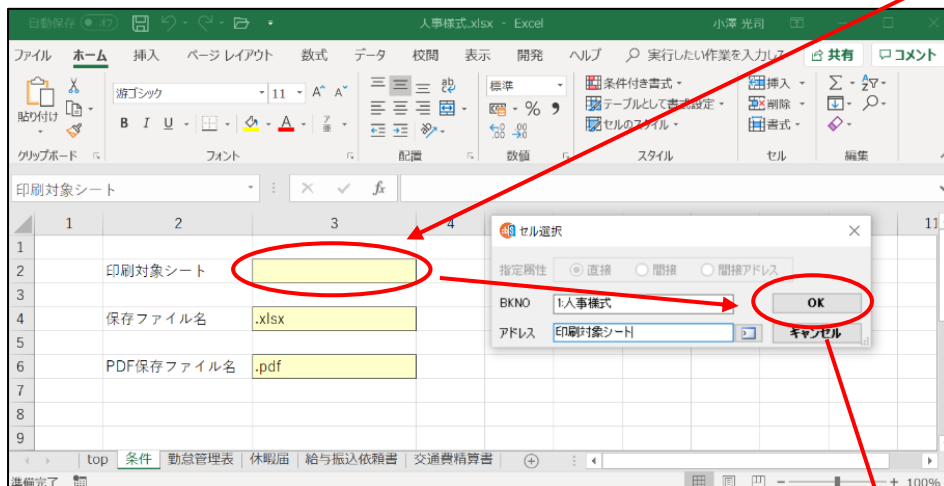
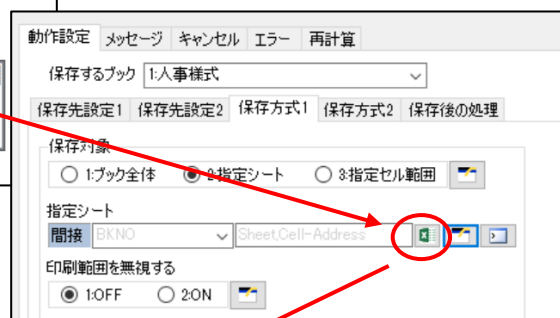
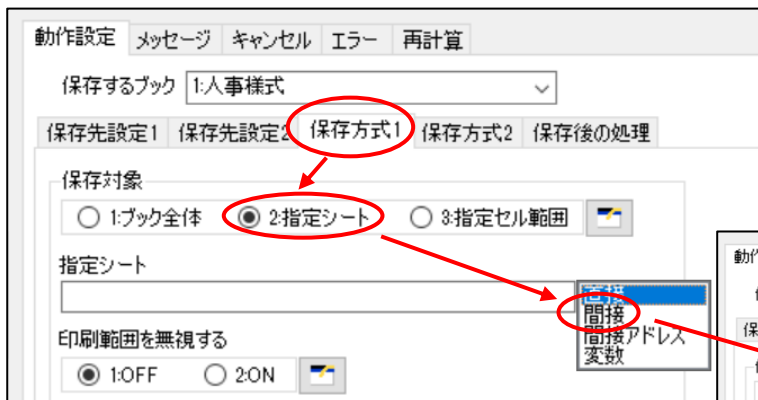
保存先設定1にて「ダイアログ指定」を選択します。



保存先設定2にて「初期表示ファイル名」を指定します。



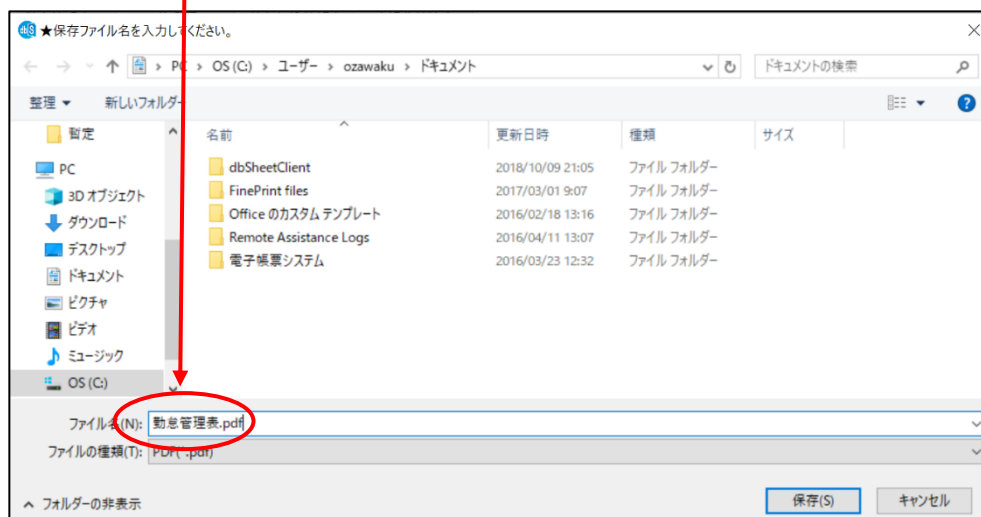
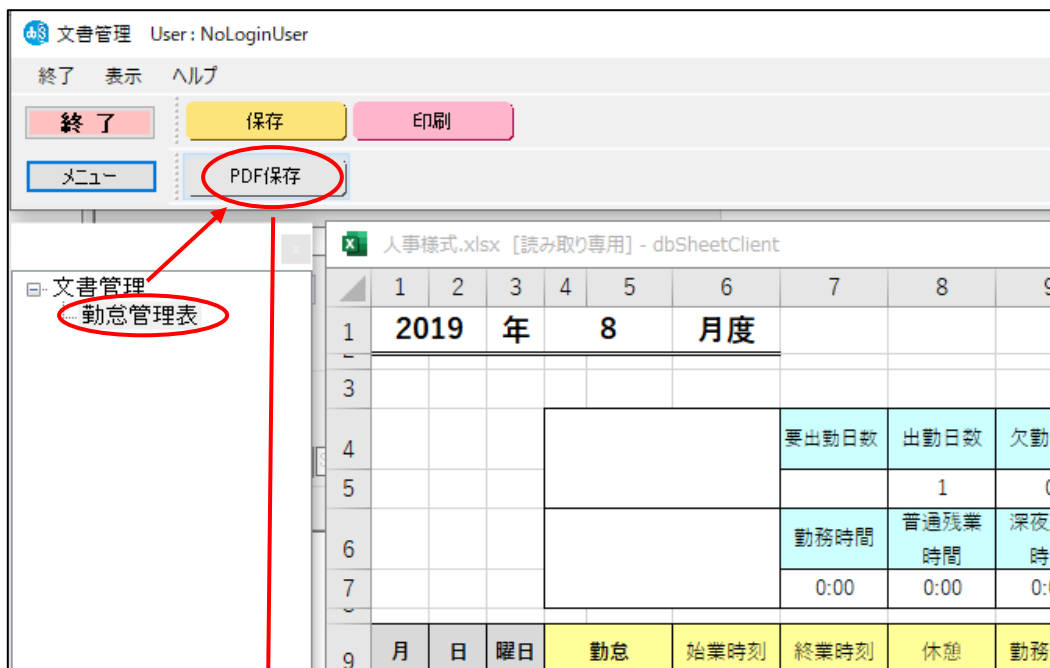
保存方式1にて保存対象シートを指定します。



デバッグモードで実行

左メニューより「勤怠管理表」メニューを選択して様式「勤怠管理表」が表示された後、「PDF 保存」ボタンをクリックします。

PDF 保存ファイル名が「勤怠管理表.pdf」になっていることを確認します。

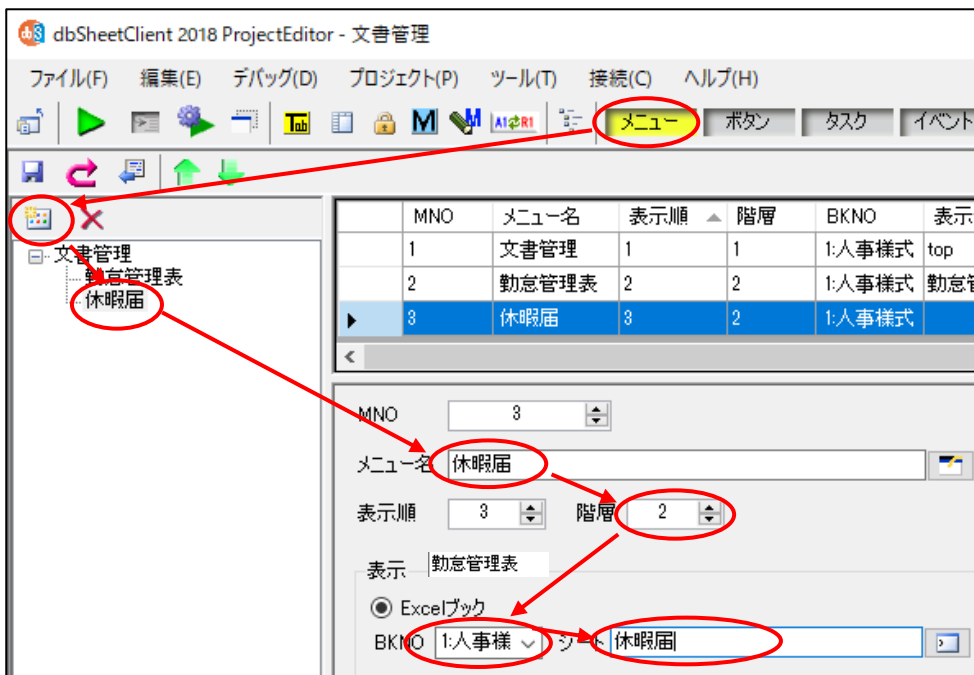


4.9.5 その他様式のメニュー（シート）切替

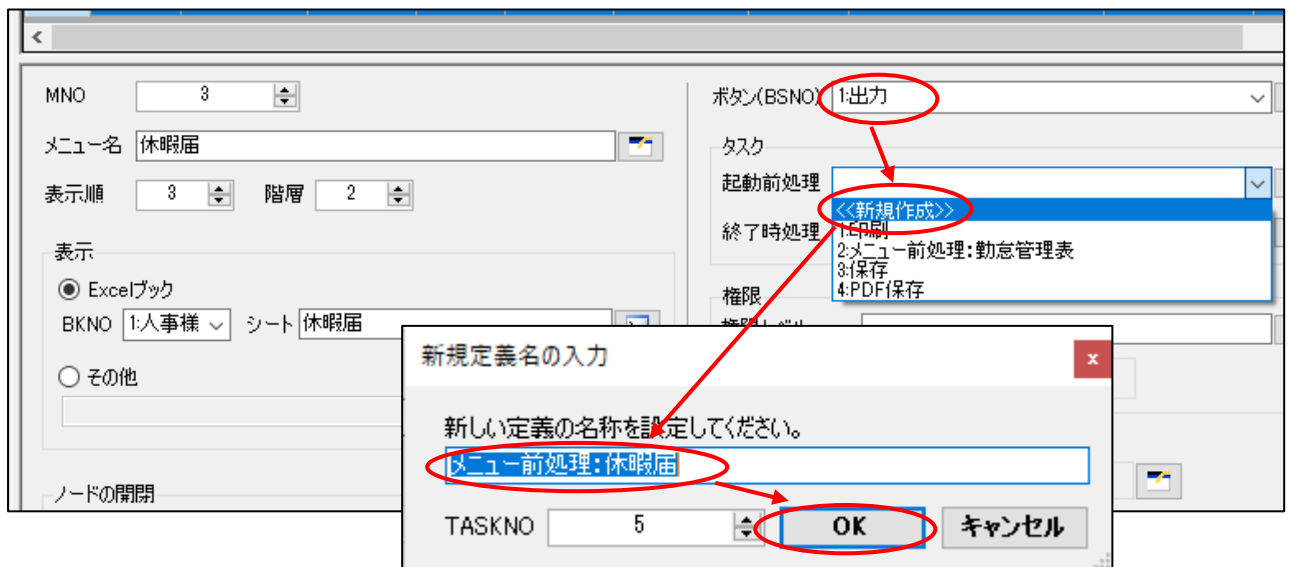
メニュー「勤怠管理表」のように、「休暇届」、「給与振込依頼書」、「交通費精算書」をメニューに追加します。

まず、新規メニュー作成ボタンをクリックして「休暇届」メニューを追加します。

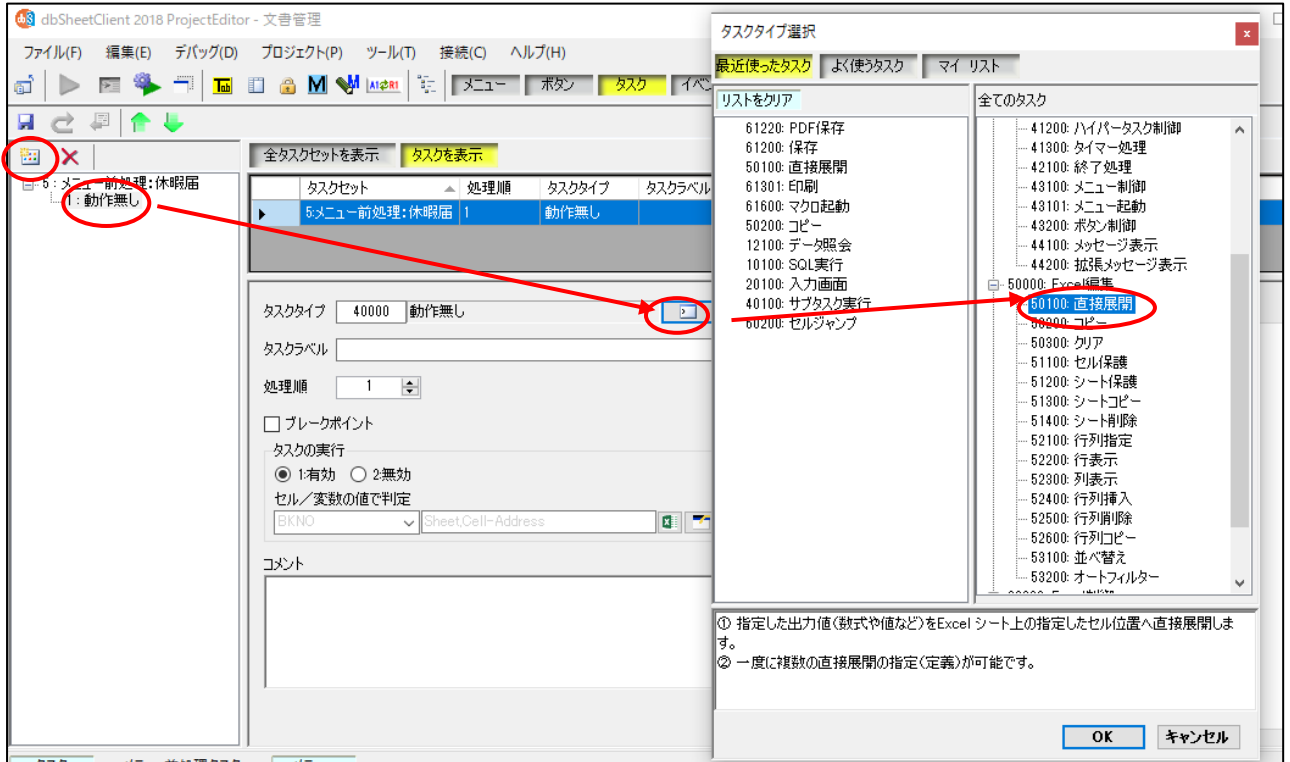
階層を「2」に変更、Excel ブックを「人事様式」、シートを「休暇届」に変更



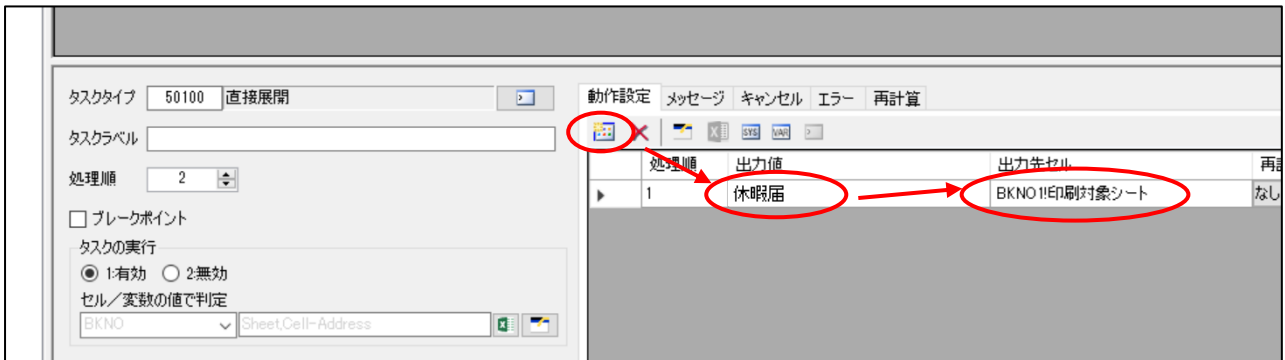
ボタンに「1:出力」を選択し、起動前処理にタスクを割り付けます。(タスク名は「メニュー前処理:休暇届」)



「直接展開」タスク(タスクタイプ:50100)をセットします。

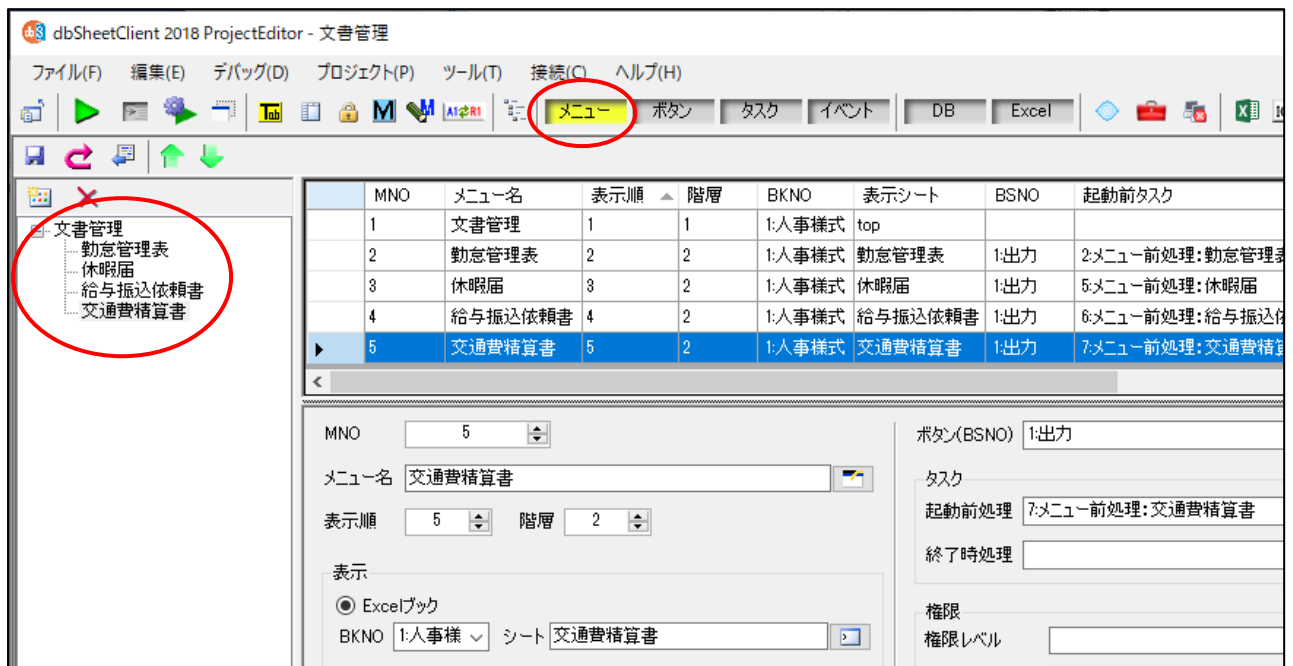


出力値 → 「休暇届」、出力先セル → 条件シートの「印刷対象シート」セル

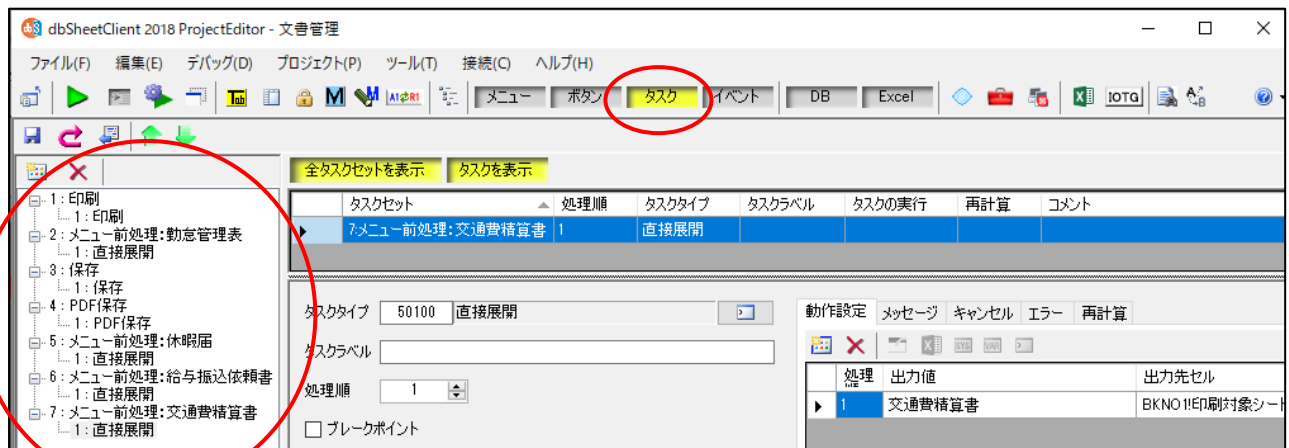


上記のメニュー「休暇届」の追加操作を同様にして、メニュー「給与振込依頼書」、「交通費精算書」を追加します。

メニューは以下の通りとなります。

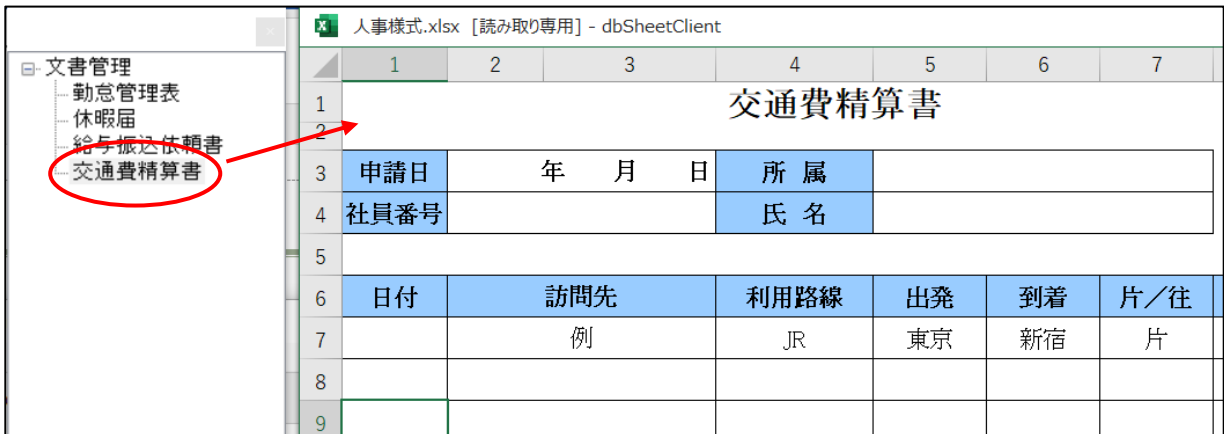
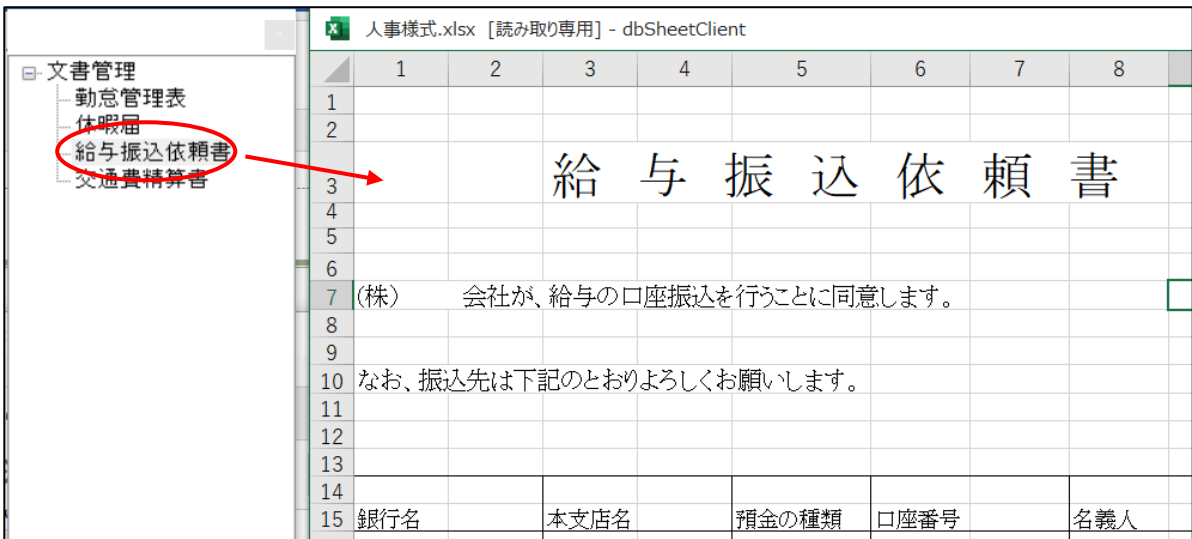
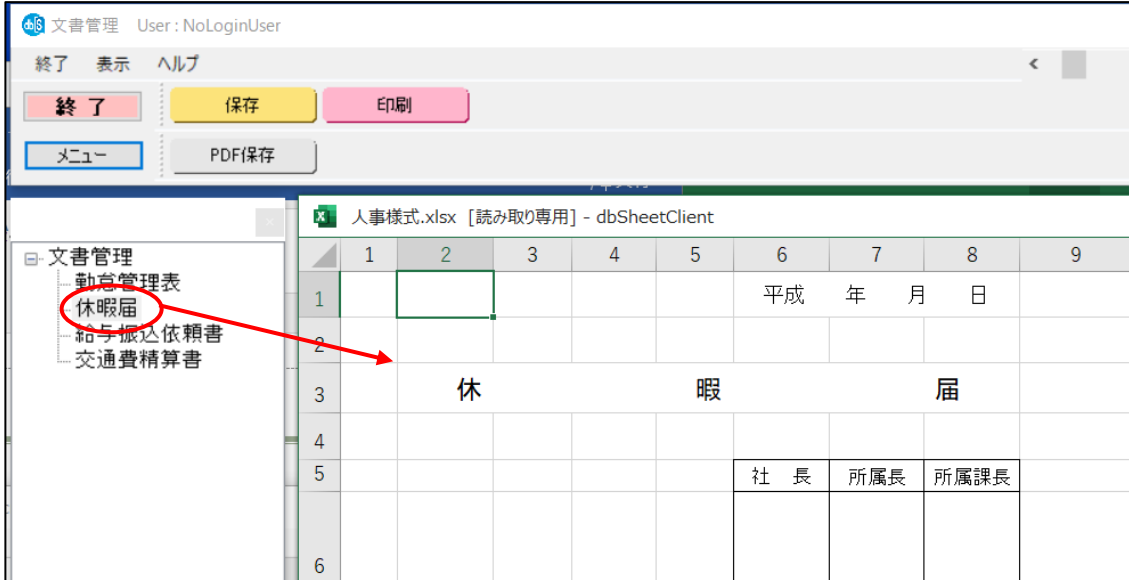


タスクは以下の通りとなります。



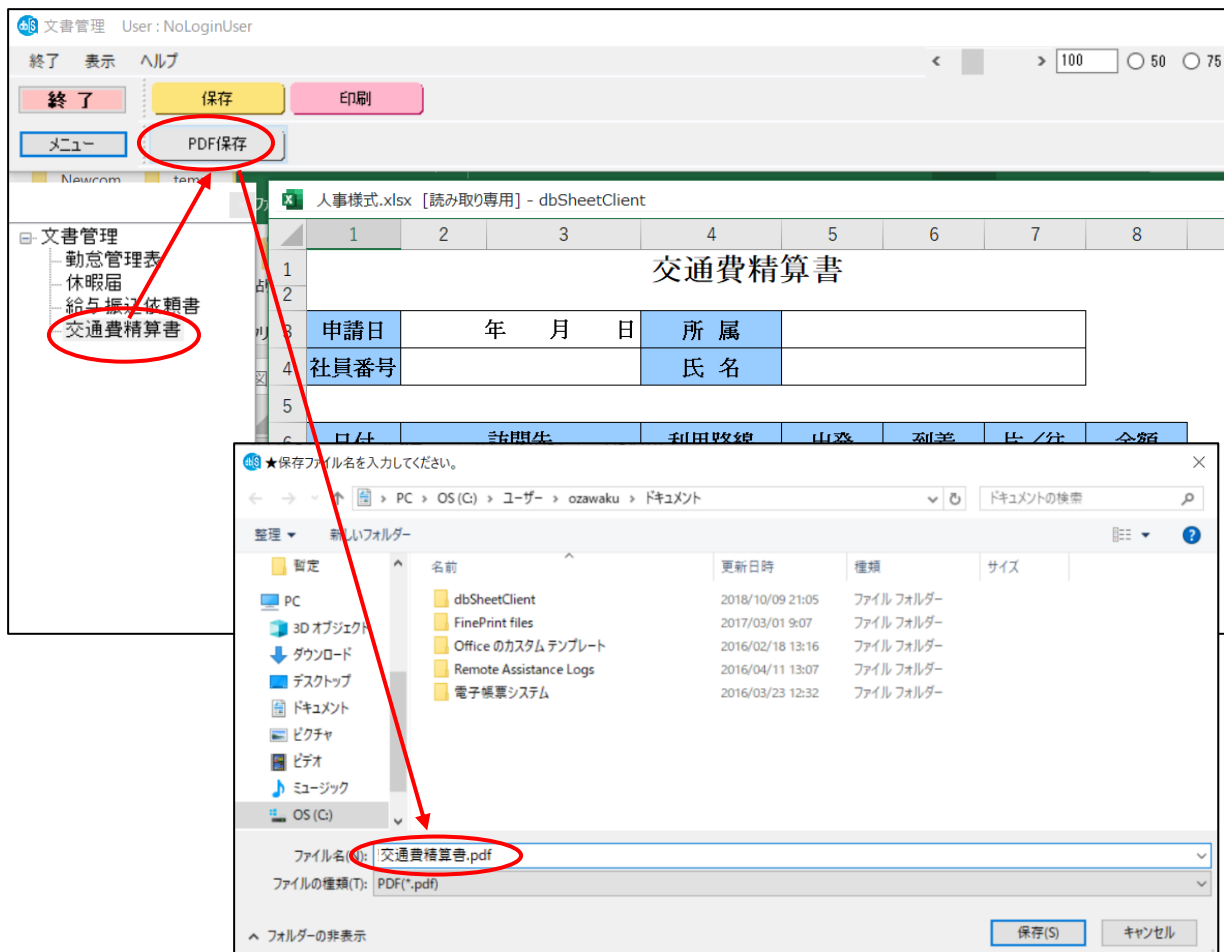
デバッグモードで実行

左メニューより各様式メニュー（「勤怠管理表」、「休暇届」、「給与振込依頼書」、「交通費精算書」）を選択して様式が切り替わることを確認します。



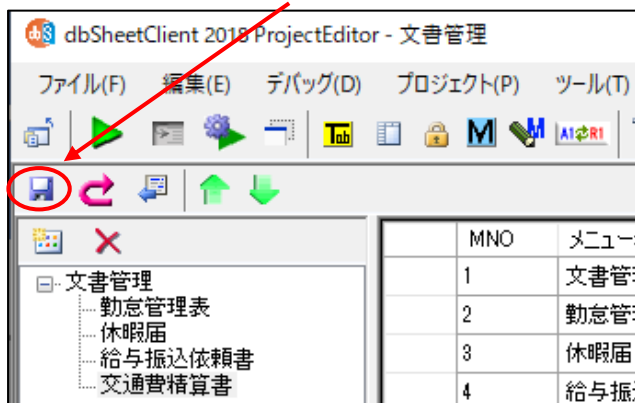
「印刷」「保存」「PDF 保存」ボタンで切り替えた様式を対象に印刷、保存ができることを確認します。

例:「交通費精算書」メニューで「PDF 保存」ボタン



4.9.6 プロジェクトを保存

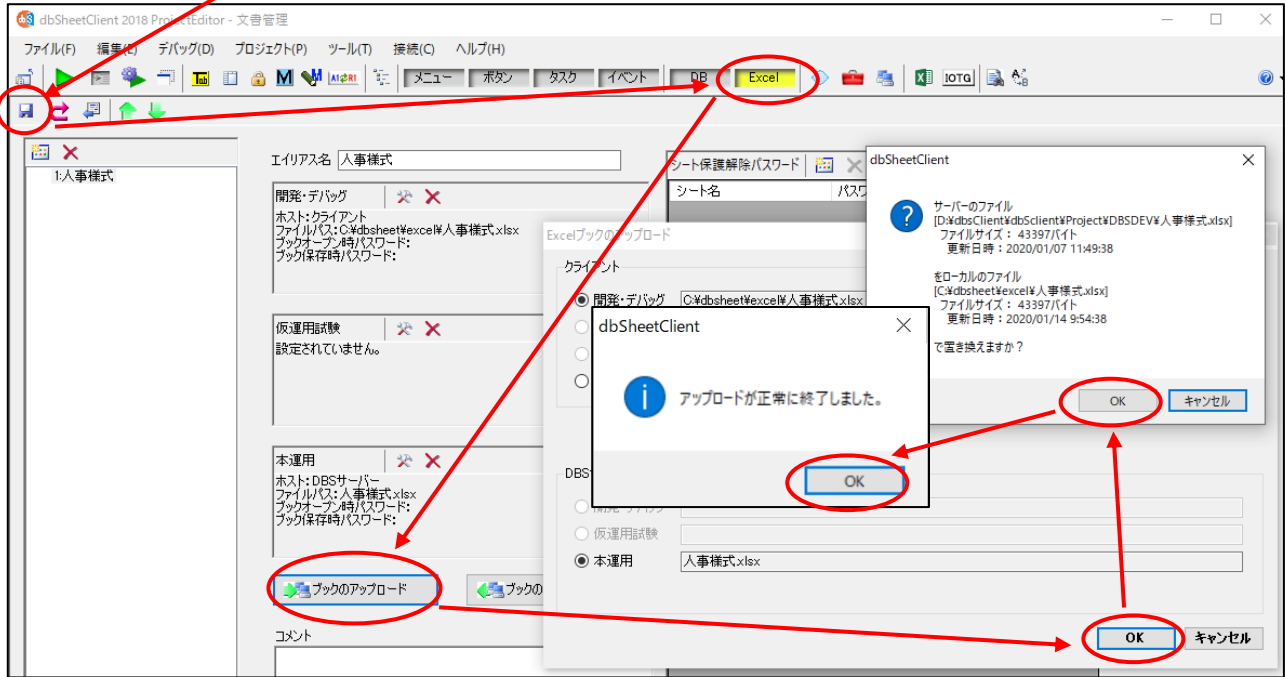
ここまででいったんプロジェクトを保存します。



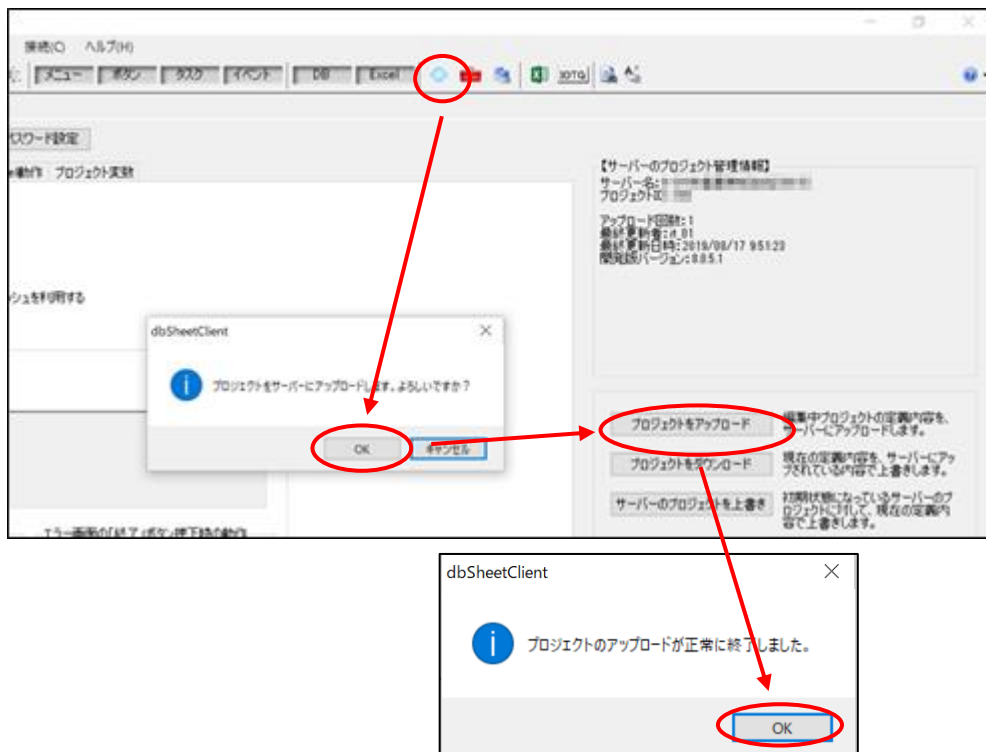
4.10 本番環境へアップロード

プロジェクトを上書き保存して、Excel ファイルを本番環境にアップロードします。

(注: 開発体験版では本作業不可)



プロジェクトをサーバーにアップロードします。(注: 開発体験版では本作業不可)

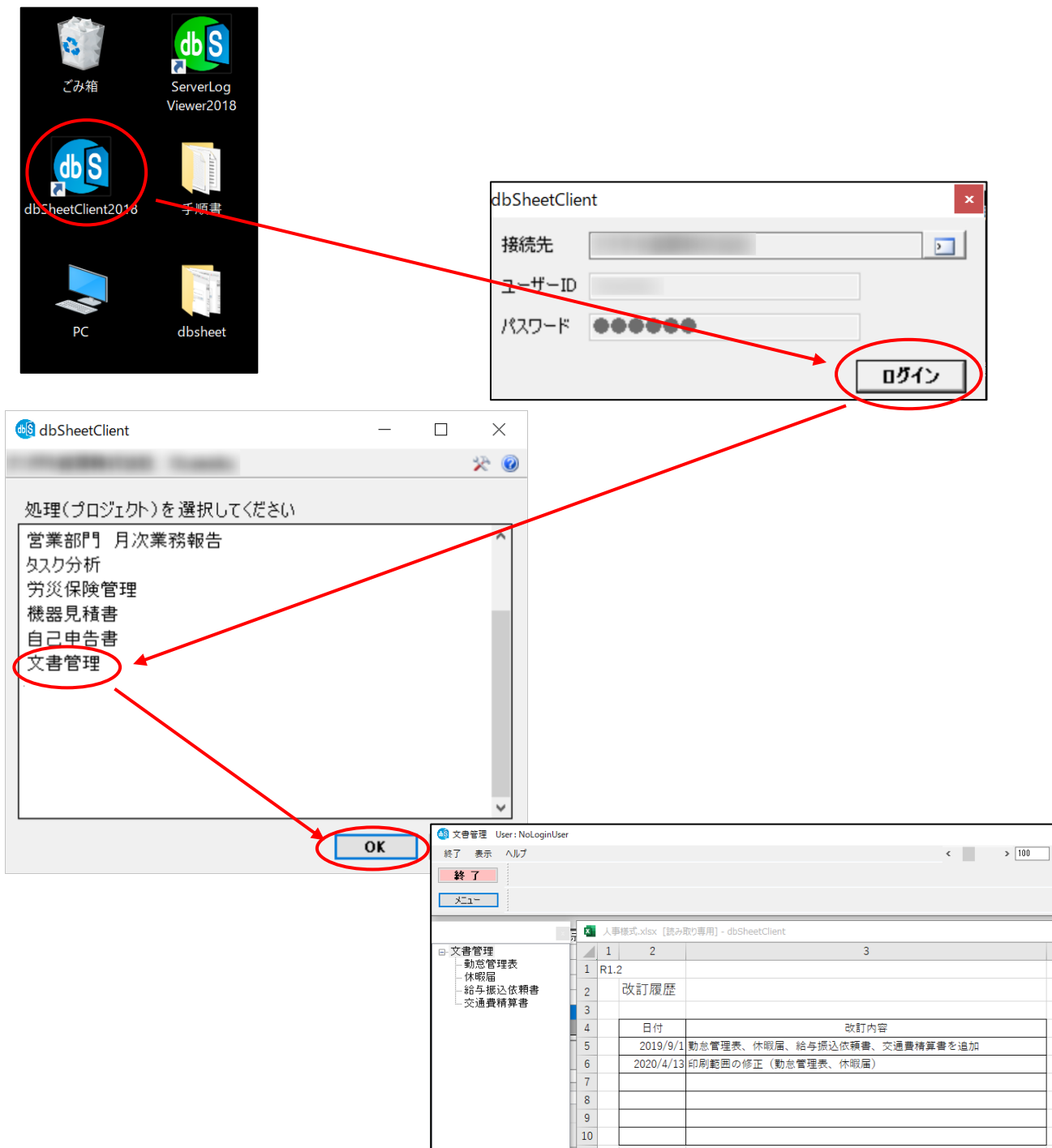


4.11 ユーザー環境で実行

ユーザーのパソコンから dbSheetClient を実行して開発環境と同じ動作を確認します。

(注:開発体験版では本作業不可)

dbSheetClient ショートカットをクリック → ログイン → メニューより「文書管理」を選択



5. 経営の見える化

時代はますます変化が激しくなっています。インターネットによる情報伝達スピード、経済のグローバル化、会社を舵取りする経営者にとって頭の痛いことばかりです。

少しでも早く変化の兆しを見つけ、それに対応する戦略・戦術を立案し実行する。

社員が一丸となって戦略・戦術を実行するためにはその目的・目標を共有しなければなりません。

進むべき道を共有するには文字や数字の羅列よりも直感的に訴えかけてくる図表の方が効果的です。

いま、IT 関連製品として BI ツールが流行っているのもそういった理由からでしょう。

本章では、売上明細データを使って、パレート図(売上構成比の見える化)と Z 図(売上推移の見える化)を作成するシステムを開発します。

前章では使わなかったデータベース連携に対応したシステムを紹介します。

Excel 単独のグラフがデータベース連携すると、どのような機能アップを実現するか体感してください。

本章では次の手順についてご確認ください。

- ①データベースと Excel の連携
- ②各種タスクの設定
- ③システムのデバッグ

計上年月	部署コード	部署	売上先コード	売上先	分類コード	分類	売上金額
201704	A01	西東京支店	103	(株)新潟糸魚川	301	総菜	607,725
201704	A01	西東京支店	1039	(株)福岡田川添田	405	水	40,000
201704	A01	西東京支店	1066	(株)富山南砺	402	清涼飲料	2,438,000
201704	A01	西東京支店	7927	(株)静岡御前崎	405	水	10,800
201704	A01	西東京支店	9366	(株)千葉葦里	402	清涼飲料	88,800
201704	A01	西東京支店	10167	(株)広島安芸坂	405	水	4,276,800
201704	A01	西東京支店	11420	(株)静岡浜松南	411	ゴマ	1,099,660
201704	A01	西東京支店	12526	(株)鳥取東北栄	402	清涼飲料	88,800
201704	A01	西東京支店	13462	(株)北海上川美瑛	405	水	40,000
201704	A02	東東京支店	103	(株)新潟糸魚川	305	納豆	187,000
201704	A02	東東京支店	119	(株)福島伊達	307	農産瓶詰	28,800
201704	A02	東東京支店	134	(株)埼玉東松山	305	納豆	266,600

5.1 プロジェクトの準備

「4.1 新規プロジェクトの作成」を参考にして「売上分析」というプロジェクトを準備します。

「4.6 Excel ファイルの準備」を参考に Excel ファイル名は「売上分析.xlsx」にしてください。

※拡張子 xlsx はマクロ有効ブック



(注)本章では VBA(マクロ)を多用した事例を紹介していますがマクロをほとんど使わずシステムを構築することも可能です。

5.2 データベースの準備

1.2.1 エイリアス設定

管理画面よりプロジェクト「売上分析」にデータベースを割り当てます。

システム管理者 dbSheetClient / サーバシステム Ver8.0.5.1 ユーザーID: Home

管理者トップ

【プロジェクトに参加するデータベース】

プロジェクト選択: PID:187 売上分析

エイリアス選択: エイリアスを選択してください

リンク削除	エイリアスNO	エイリアス名	DB名	DB種類	更新者	更新日
削除	7	DBC	dbc	MSSQL	SETUPUSER	2019/08/31 12:41:03

管理メニュー:

- 1.グループ登録
- 2.プロジェクト登録
- 3.ユーザー登録
- 4.データベース登録
- (2)グループ管理
 - 1.ユーザー参加
 - 2.ユーザーコピー
- (3)プロジェクト管理
 - 1.グループ参加
 - 2.エイリアス参加
 - 3.グループコピー

エイリアス設定の必要性についてです。

DB エイリアス設定で以下が可能になります。

- ・プロジェクト定義の変更無しで、接続 DB の変更が可能。

(dbSheetClient 開発版の DB 定義を変更しないで、サーバー管理画面の設定で接続 DB の変更が可能)

例 1) 開発、本番運用時の接続 DB の変更が容易。

サーバー管理画面で設定したエイリアスを開発版で指定しておくことで、開発時の DB 定義のまま、本番運用時にはサーバー管理画面で設定を変更するだけで、接続先をテスト用の DB から本番運用で使用する DB に変更することができます。

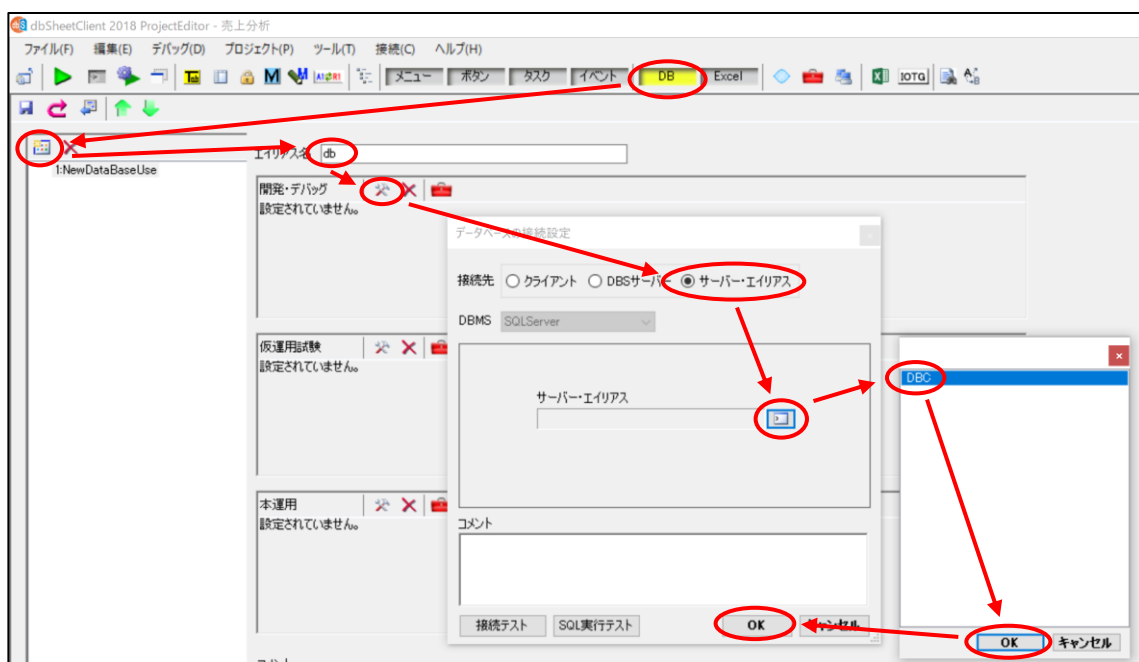
もし、エイリアスを利用しない場合には、開発版で DB 定義を変更したのちサーバーにプロジェクト定義をアップロードする必要があります。

例 2) 多数のプロジェクトで同一 DB を利用する場合、接続 DB の変更が容易。

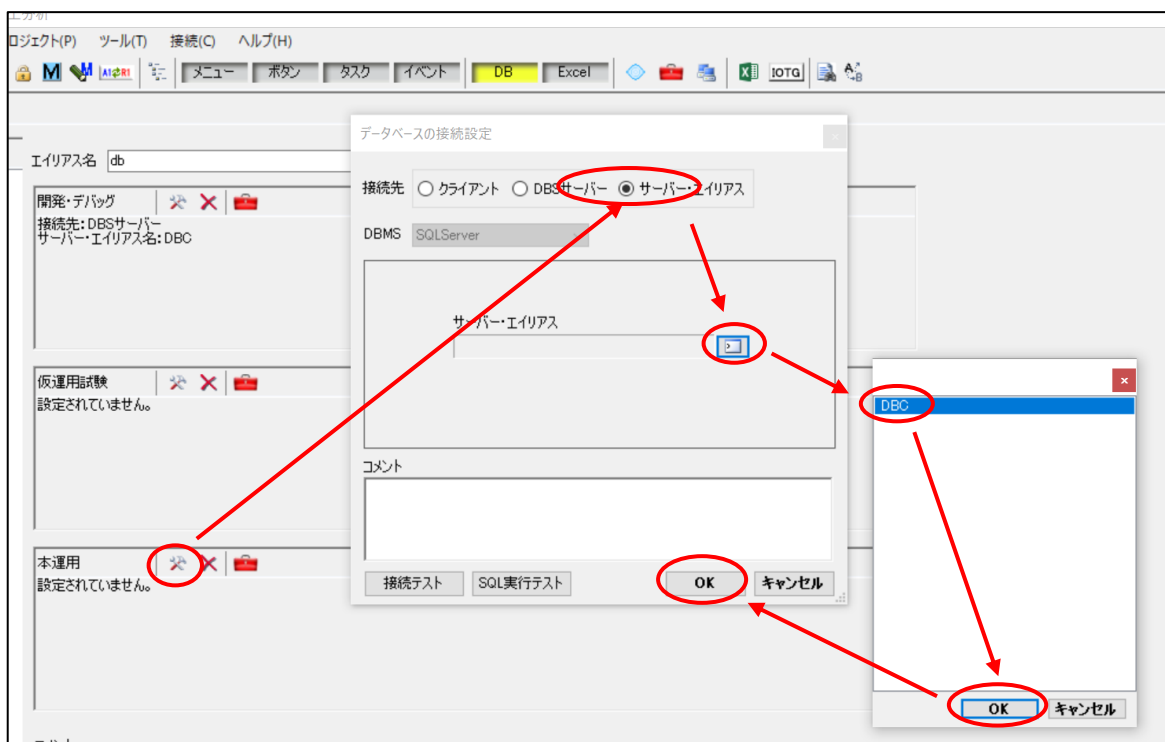
多数のプロジェクト定義で同一 DB の接続先が変更となった場合、サーバー管理画面で設定を変更するだけで、それら多数のプロジェクトの接続先 DB を即座に変更できます。

1.2.2 データベース登録

dbSheetClient 開発版にて開発環境のDB設定に登録したエイリアスを割り付けます。



本運用環境のDB設定にも登録したエイリアスを割り付けます。



下記のように設定されます。



1.2.3 売上テーブルの作成

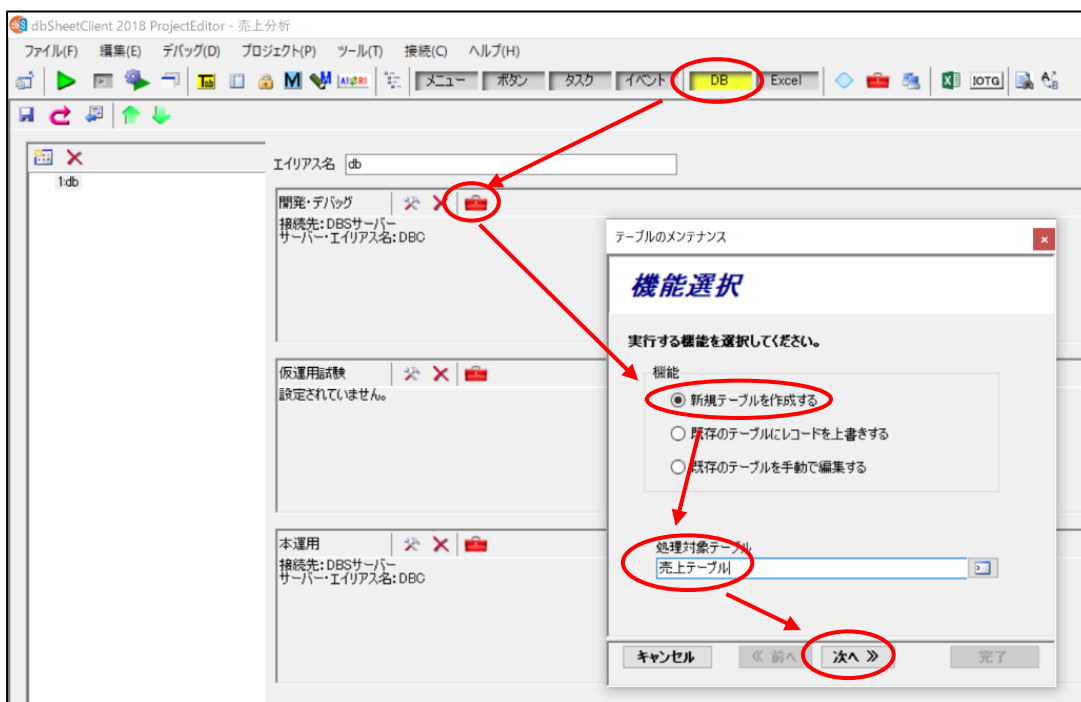
売上を分析するための適当なテーブルが無い、あるいはデータベースを売上管理に利用されていない場合 Excel の実績ファイルからデータベースのテーブルを作成することができます。

既に運用されているテーブルがある場合、本項は読み飛ばしてください。

売上実績が展開されたシートを「売上分析.xlsxm」に適当なシートを作成してコピーします。

1	2	3	4	5	6	7	8	9
計上年月	部署コード	部署	売上先コード	売上先	分類コード	分類	売上金額	
201704	A01	西東京支店	103	(株)新潟糸魚川	305	総菜	607,725	
201704	A01	西東京支店	1039	(株)福岡田川赤田	405	水	40,000	
201704	A01	西東京支店	1066	(株)富山南砺	402	清涼飲料	2,438,000	
201704	A01	西東京支店	7927	(株)静岡御前崎	405	水	10,800	
201704	A01	西東京支店	9366	(株)千葉富里	402	清涼飲料	88,800	
201704	A01	西東京支店	10167	(株)広島安芸坂	405	水	4,276,800	
201704	A01	西東京支店	11420	(株)静岡浜松南	411	ゴマ	1,099,660	
201704	A01	西東京支店	12526	(株)鳥取東伯北栄	402	清涼飲料	88,800	
201704	A01	西東京支店	13462	(株)北海上川美瑛	405	水	40,000	
201704	A02	東東京支店	103	(株)新潟糸魚川	305	納豆	187,000	
201704	A02	東東京支店	119	(株)福島伊達	307	農産瓶詰	28,800	
201704	A02	東東京支店	144	(株)埼玉東松山	305	納豆	266,600	
201704	A02	東東京支店	153	(株)東京国分寺	307	農産瓶詰	1,200	
201704	A02	東東京支店	205	(株)奈良御所	305	納豆	106,000	

dbSheetClient 開発版にて DB ツールを起動 → テーブルの新規作成を選択




売上実績が展開されたシートを指定します。

テーブルのメンテナンス

データ取込の対象

何からデータを取得するかを選択してください。

Excelから

取込開始セル: BKNO Sheet,Cell-Address 

取込範囲

取込列数: 0 取込行数: 0 自動設定

1行目は列名

MDBから

ファイルパス:

パスワード:

テーブル:

キャンセル << 前へ 次へ >> 完了

先頭行、1列目のセルを選択

売上分析.xlsxm - Excel

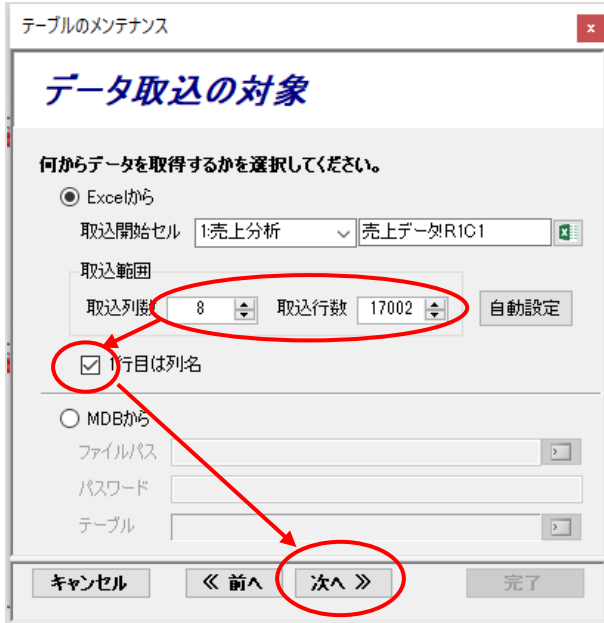
ファイル ホーム 挿入 ページレイアウト 数式 データ 校閲 表示 開発 実行したい作業を入力してください...

R1C1

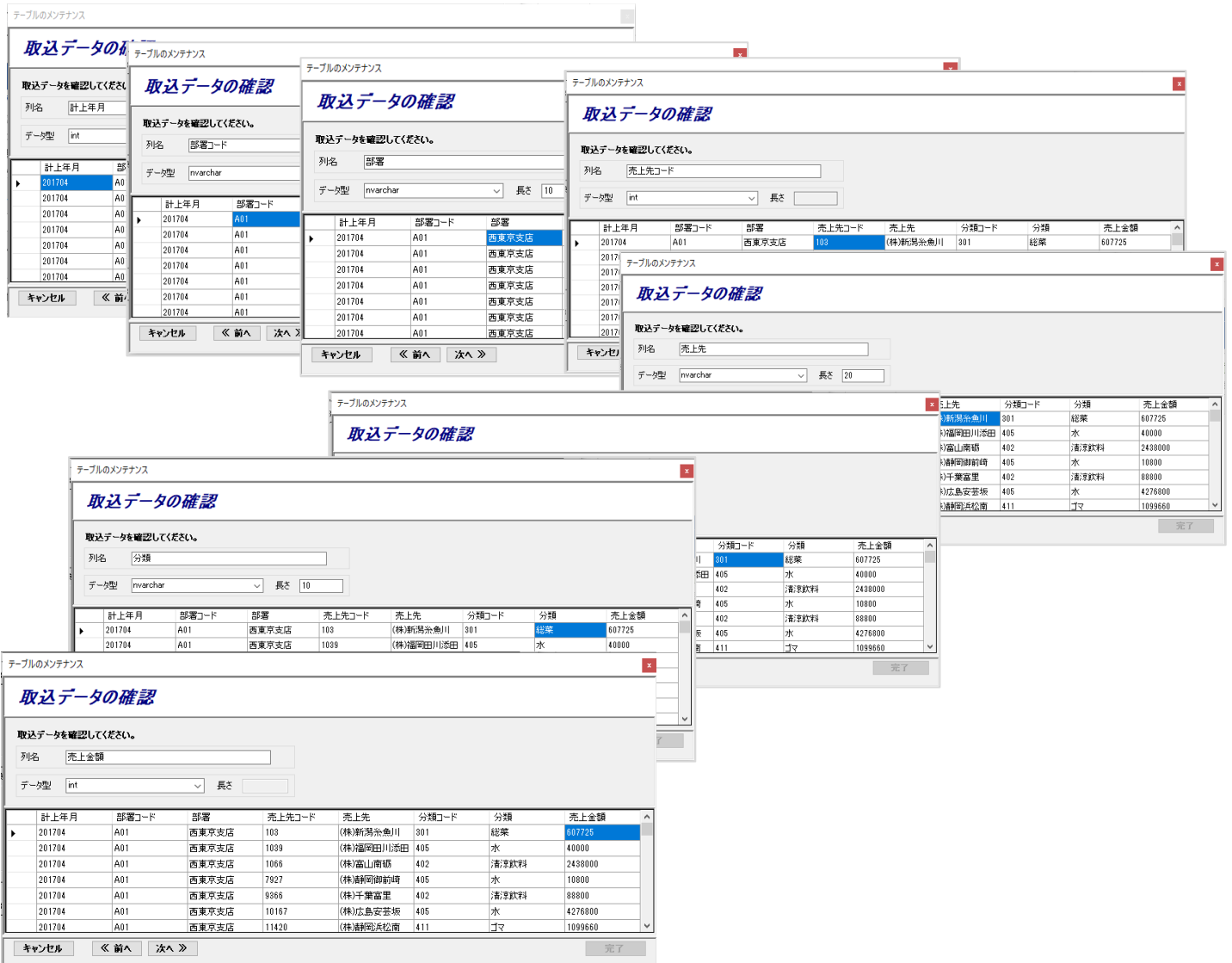
1	2	3	4	5	6	7	8
計上年月	部署 コード	部署	売上先 コード	売上先	分類 コード	分類	売上金額
201704	A01	西東京支店	103	(株)新潟糸魚川	301	総菜	607,725
201704	A01	セル選択				水	40,000
201704	A01					清涼飲料	2,438,000
201704	A01					水	10,800
201704	A01	BKNO				清涼飲料	88,800
201704	A01	アドレス				水	4,276,800
201704	A01	西東京支店	11420	(株)静岡浜松南	411	ゴマ	1,099,660
201704	A01	西東京支店	12526	(株)鳥取東伯北栄	402	清涼飲料	88,800
201704	A01	西東京支店	13462	(株)北海上川美瑛	405	水	40,000
201704	A02	東東京支店	103	(株)新潟糸魚川	305	納豆	187,000
201704	A02	東東京支店	119	(株)福島伊達	307	農産瓶詰	28,800
201704	A02	東東京支店	134	(株)埼玉東松山	305	納豆	266,600
201704	A02	東東京支店	153	(株)東京国分寺	307	農産瓶詰	1,200
201704	A02	東東京支店	205	(株)奈良御所	305	納豆	106,000

top 条件 売上データ

売上実績のフィールド名やレコード数を自動的に認識します。



各フィールドのデータ型を選択セルを変えながら確認します。(変更も可)



主キーを設定します。

テーブルのメンテナンス

新規テーブル作成の詳細設定

テーブル作成の詳細を設定してください。

主キーの設定

- 計上年月(int)
- 部署コード(nvarchar(10))
- 部署(nvarchar(10))
- 売上先コード(int)
- 売上先(nvarchar(20))
- 分類コード(int)
- 分類(nvarchar(20))
- 売上金額(int)

dbS制御フィールドが無い場合は自動追加する

データ型変換でエラーになった値はNULL値として扱う

キャンセル << 前へ 次へ >> 完了

dbSheetClient

処理が正常に終了しました。

OK

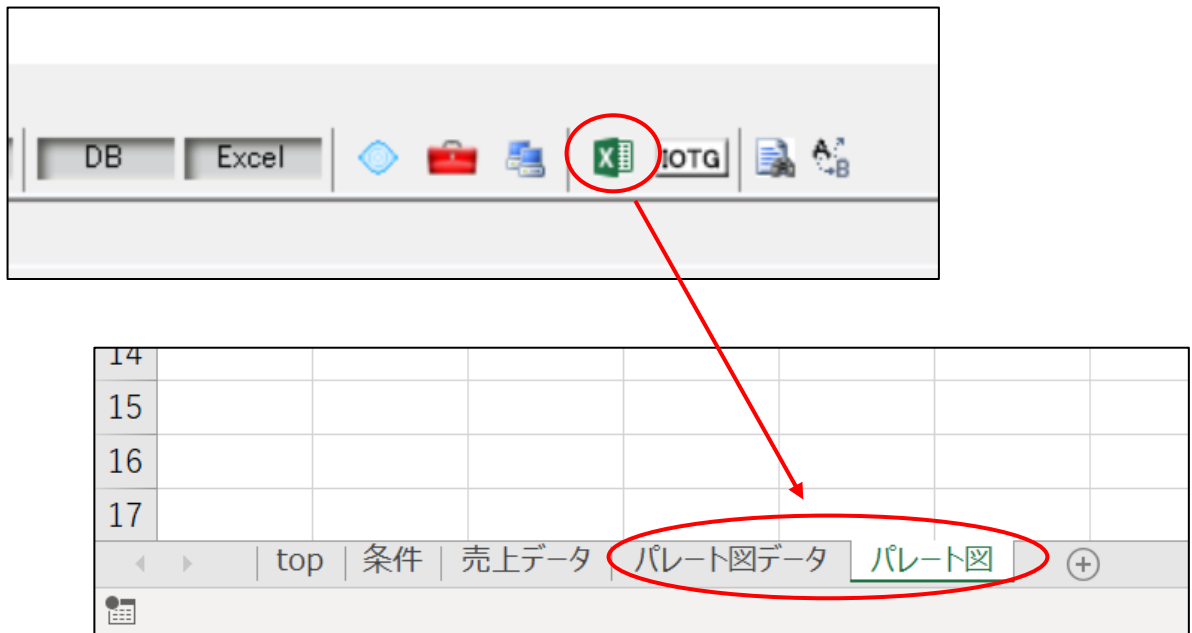
テーブルが作成されます。(実績データも同時にデータベースに登録されます。)

5.3 パレート図

パレート図はいわゆるABC分析と呼ばれる分析手法をビジュアル化するグラフです。
部署の売上を構成する売上先や分類の内訳とその全体に占める比率を表します。
Excel に標準実装された2軸グラフ(売上を縦棒グラフ、構成比率を折れ線グラフ)
売上全体の80%が何社の売上先で構成されているかを一目で認識することができます。
本章では売上期間や部署、売上先、分類を切り替えてシミュレーションするプロジェクトを作成します。

1.3.1 グラフの雛形を準備

dbSheetClient 開発版の Excel ボタンで「売上分析.xlsm」を呼び出し、
シート「パレート図データ」「パレート図」を追加します。



シート「パレート図データ」にダミーのデータを入力してグラフを作成します。

関数式 =RC[-1]
累計初期値

関数式 =SUM(R[2]C:R[101]C)
100 件の合計売上

ダミーデータ
100件

関数式 =RC[-1]/R1C3
累計÷合計(%表示)

	1	2	3	4	5
1			7,525,000		
2	No	項目	売上	累計	比率
3	1	テスト001	100,000	100,000	1%
4	2	テスト002	99,500	199,500	3%
5	3	テスト003	99,000	298,500	4%
6	4	テスト004	98,500	397,000	5%
7	5	テスト005	98,000	495,000	7%
8	6	テスト006	97,500	592,500	8%
9	7	テスト007	97,000	689,500	9%
10	8	テスト008	96,500	786,000	10%
11	9	テスト009	96,000	882,000	12%
12	10	テスト010	95,500	977,500	13%
13	11	テスト011	95,000	1,072,500	14%
14	12	テスト012	94,500	1,167,000	16%
15	13	テスト013	94,000	1,261,000	17%
16	14	テスト014	93,500	1,354,500	18%
17	15	テスト015	93,000	1,447,500	19%

top | 条件 | 売上データ | **パレート図データ** | パレート図 | (+)

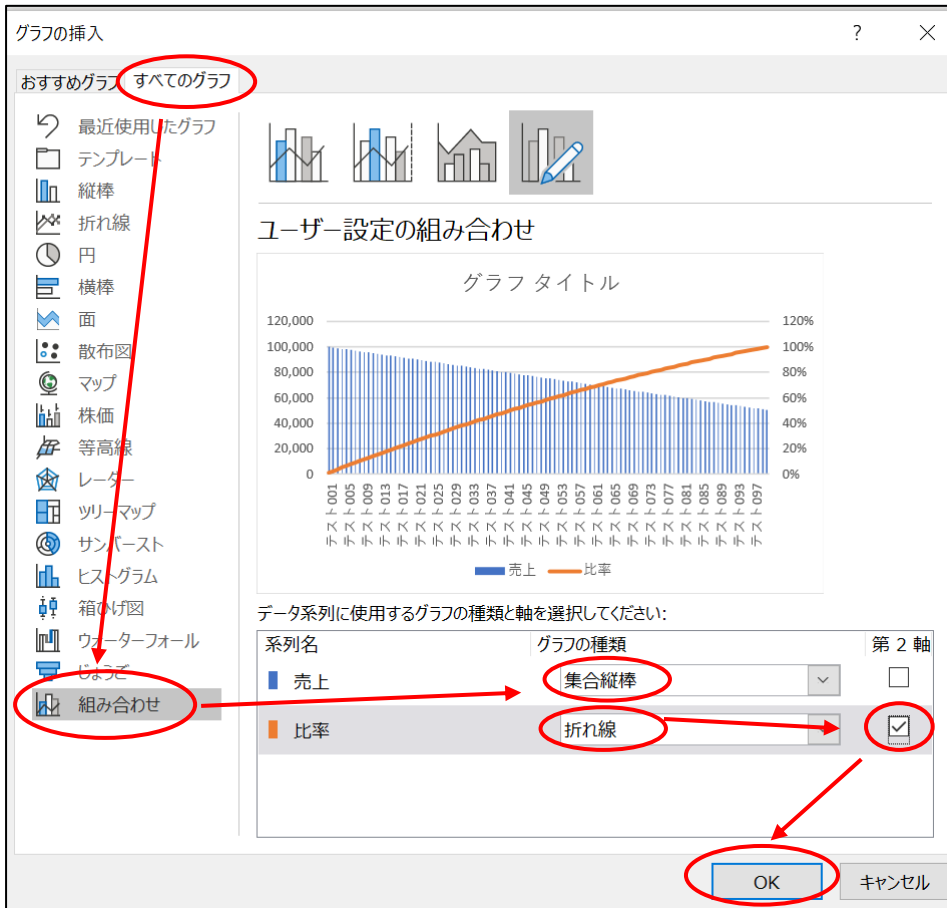
「項目」「売上」「比率」列を選択します。

関数式 =R[-1]C+RC[-1]
累計

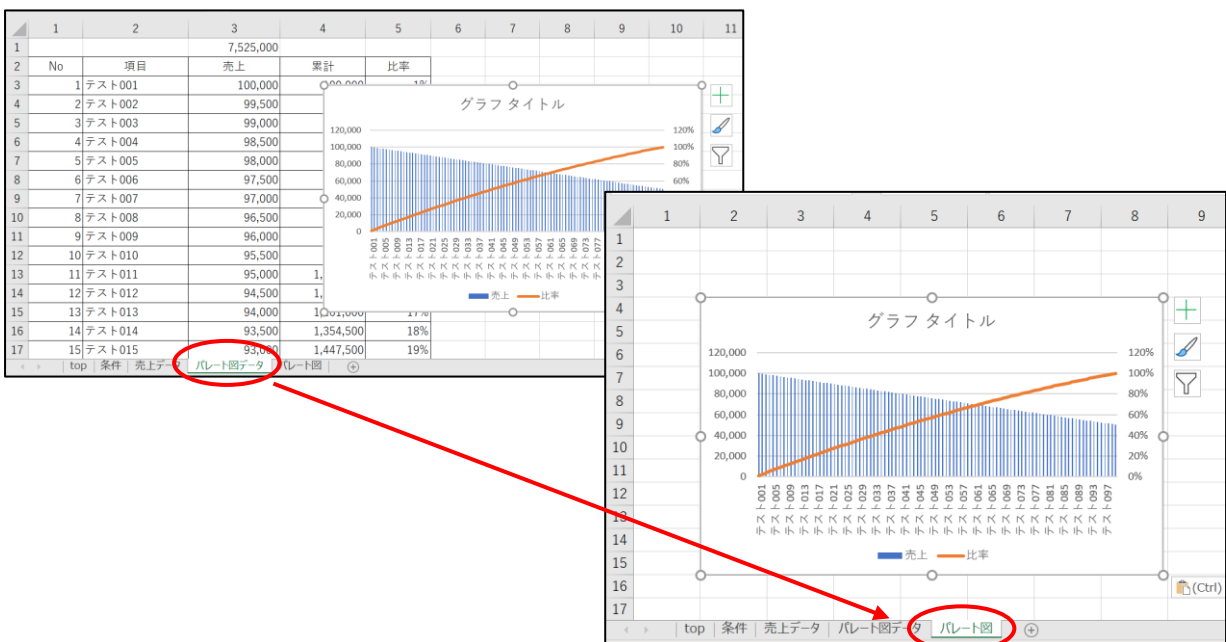
	1	2	3	4	5
1			7,525,000		
2	No	項目	売上	累計	比率
3	1	テスト001	100,000	100,000	1%
4	2	テスト002	99,500	199,500	3%
5	3	テスト003	99,000	298,500	4%
6	4	テスト004	98,500	397,000	5%
7	5	テスト005	98,000	495,000	7%
8	6	テスト006	97,500	592,500	8%
9	7	テスト007	97,000	689,500	9%
10	8	テスト008	96,500	786,000	10%
11	9	テスト009	96,000	882,000	12%
12	10	テスト010	95,500	977,500	13%
13	11	テスト011	95,000	1,072,500	14%
14	12	テスト012	94,500	1,167,000	16%
15	13	テスト013	94,000	1,261,000	17%
16	14	テスト014	93,500	1,354,500	18%
17	15	テスト015	93,000	1,447,500	19%

top | 条件 | 売上データ | **パレート図データ** | パレート図 | (+)

メニュー「挿入」 → 「グラフ」を選択

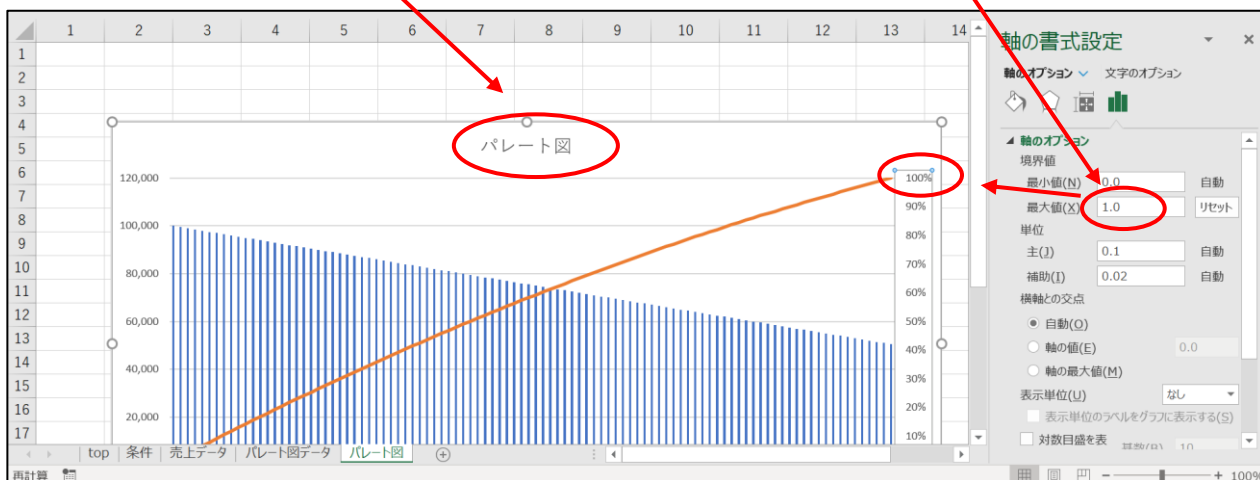


シート「パレート図データ」に挿入されたグラフを切り取ってシート「パレート図」に貼り付けます。



グラフのタイトルを「パレート図」に変更します。

比率の軸の最大値を「1.0」(100%)に変更します。



1.3.2 メニュー、ボタン、タスクを作成

メニュー「パレート図」を追加 → ボタンセット「パレート図」にボタンを登録します。

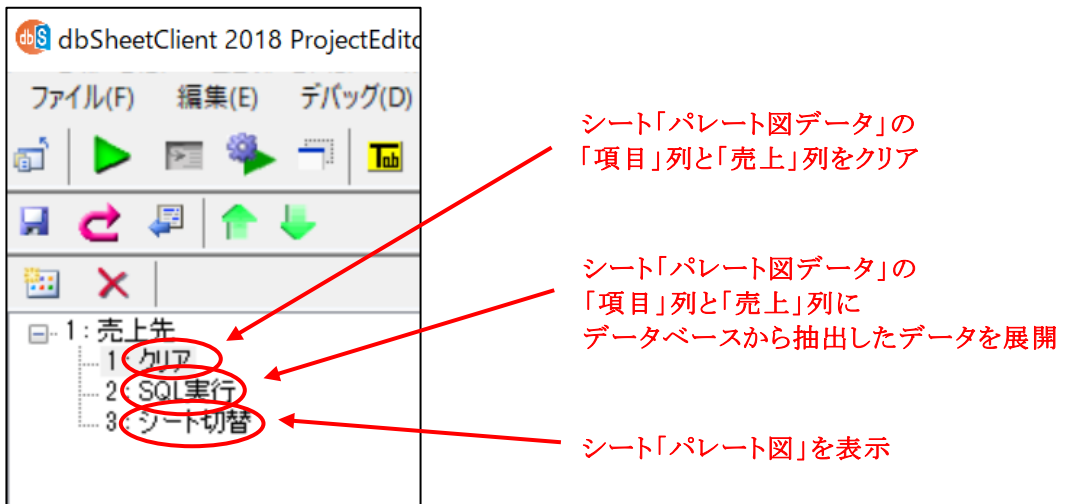
MNO	メニュー名	表示順	階層
1	売上分析	1	1
11	パレート図	2	2

BSNO	ボタンセット名	コメント
1	パレート図	

「売上先」ボタンに「売上先」という名前でタスクセットを作成します。



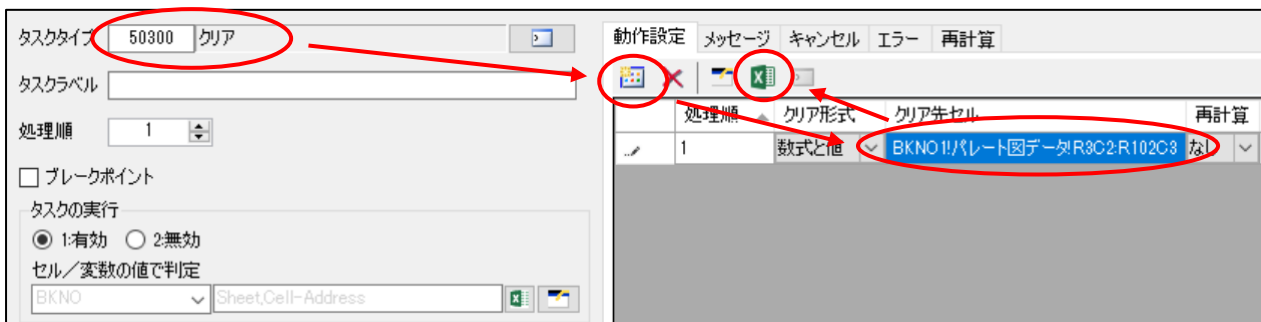
タスクセット「売上先」には次の3つのタスクを作成します。



タスクタイプ「クリア」を選択します。

処理を1つ追加してクリア先セルにシート「パレート図データ」の「項目」と「売上」が展開される範囲を指定します。

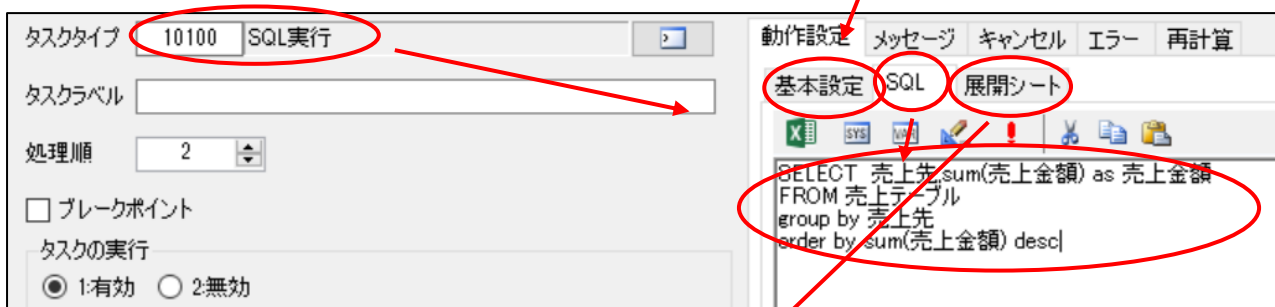
※2列目と3列目の3行から102行目まで



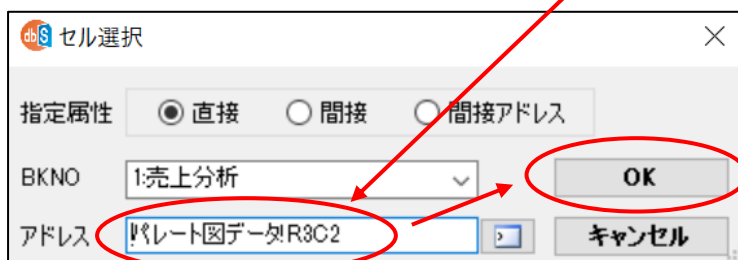
5.2.2 で登録したデータベースを選択

タスクタイプ「SQL実行」を選択します。

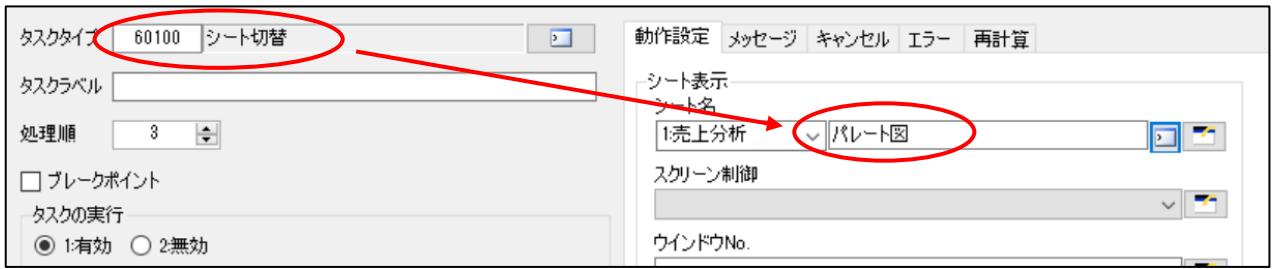
売上テーブルから売上先と売上先ごとの売上金額合計を売上金額が大きい順に抽出します。



展開先はシート「パレート図データ」の3行2列目のセル

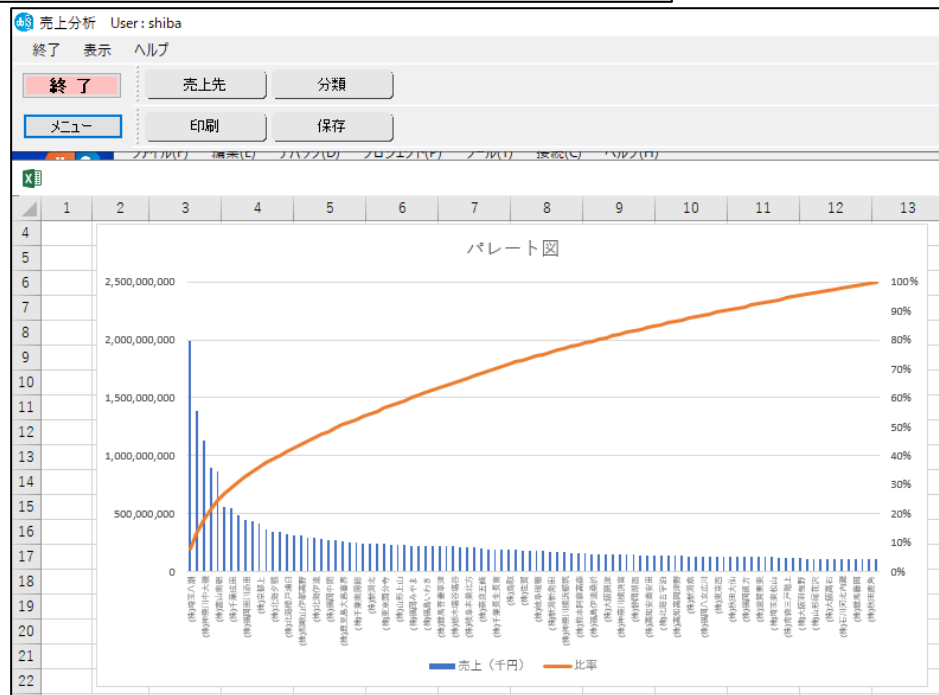
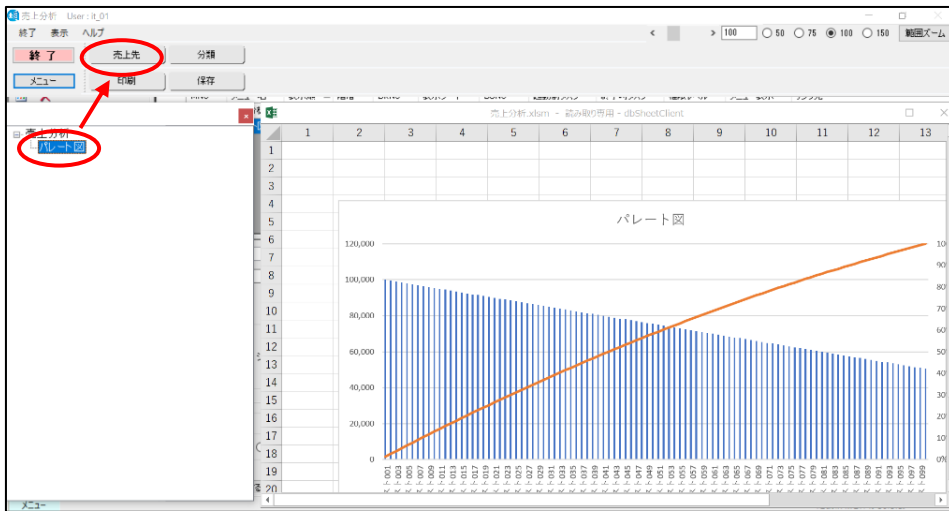


タスクタイプ「シート切替」でシート「パレート図」に切替



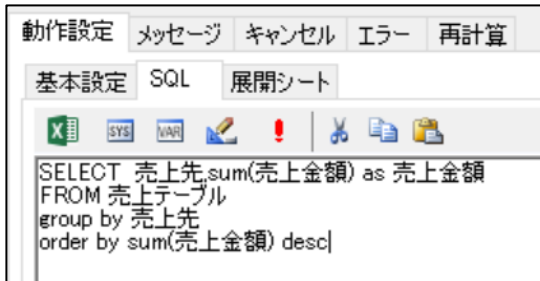
デバッグモードで実行します。

左メニュー「パレート図」→ 「売上先」ボタンをクリック→ 売上テーブルがグラフに反映されているか確認



1.3.3 「入力画面」タスクを使った対象期間の指定

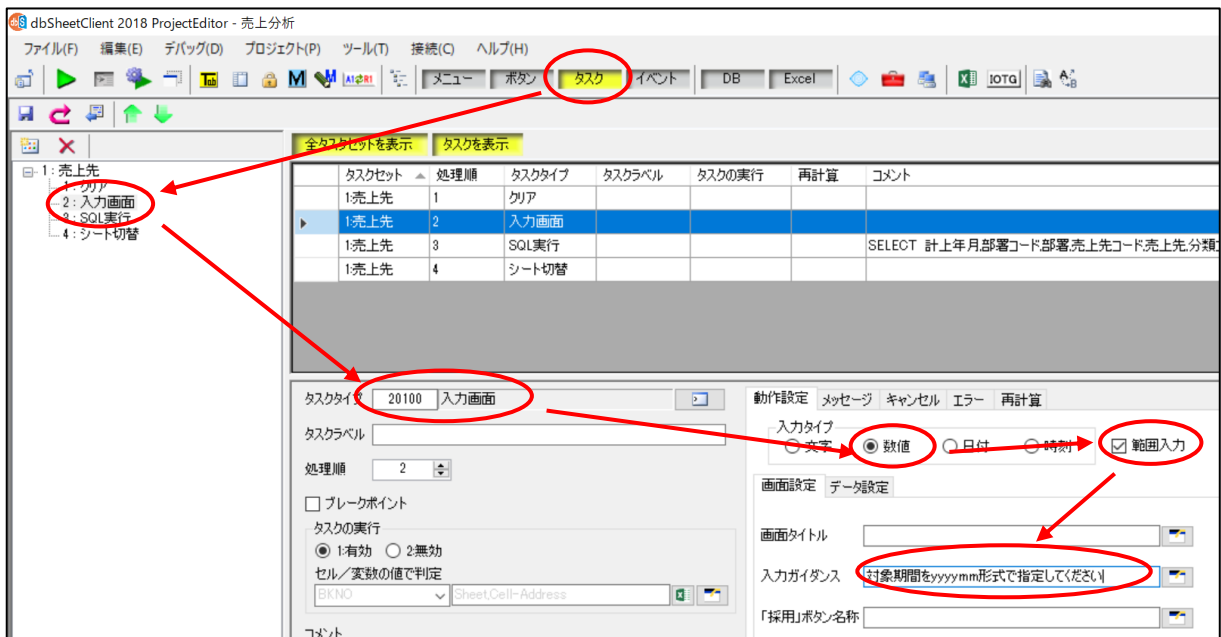
前項のSQL実行タスクでは売上テーブルに格納されたすべてのレコードが集計されてしまいます。



対象期間、部署、分類を指定して絞り込んだデータを集計するようにタスクセットを変更していきます。少し難易度が上がります。

まず対象期間を指定するタスクを「クリア」と「SQL実行」の間に「入力画面」タスクを挿入します。

入力タイプを「数値」、「範囲入力」にチェックを入れて入力ガイダンスを設定します。



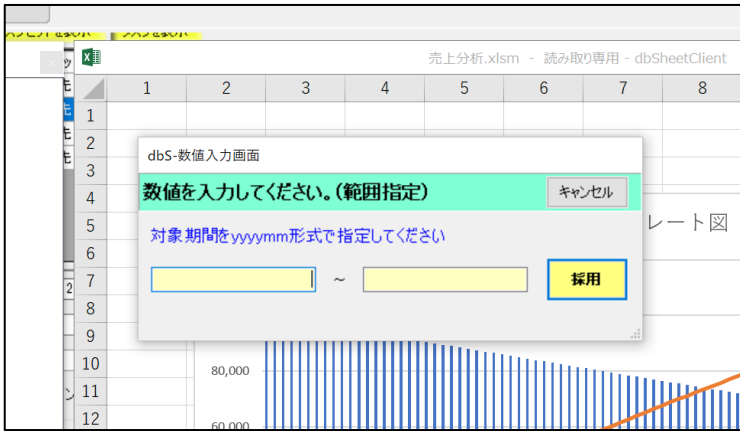
入力した対象期間を「条件」シートに出力します。
 「開始年月」「終了年月」と名前を付けたセルに出力

The image illustrates the configuration process for dbSheetClient to output data to specific cells. It consists of four main parts:

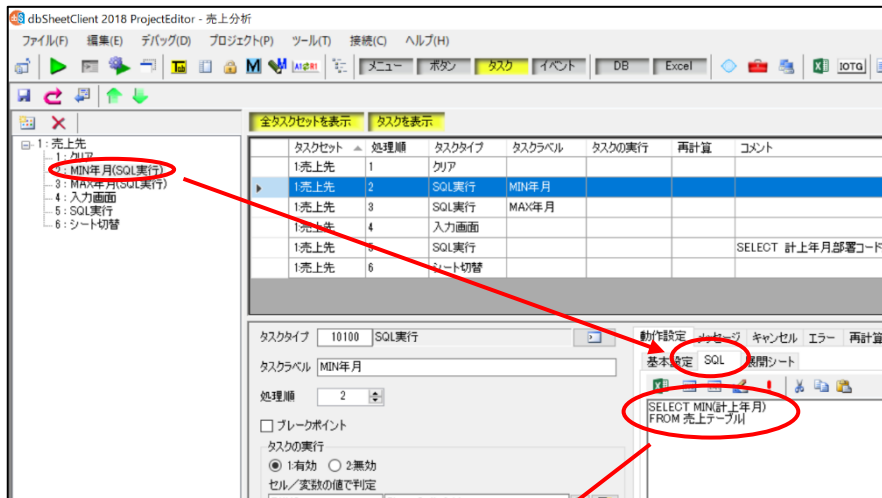
- 動作設定 (Action Settings) Dialog:** Shows the initial configuration. Under '入力タイプ' (Input Type), '数値' (Number) is selected. '範囲入力' (Range Input) is checked. Under '出力先' (Output Destination), both '(From)' and '(To)' are set to 'Sheet,Cell-Address'.
- Spreadsheet View:** Shows a grid with columns 1-10 and rows 1-10. Cells B3 (開始年月) and B5 (終了年月) are highlighted in yellow.
- セル選択 (Cell Selection) Dialog:** Shows the process of selecting the target cell. The 'アドレス' (Address) is set to '終了年月' and 'OK' is clicked.
- 動作設定 (Action Settings) Dialog (Final):** Shows the final configuration. Under '出力先' (Output Destination), both '(From)' and '(To)' are set to '1:売上分析'. The specific cells '開始年月' and '終了年月' are selected in the address fields.

デバック実行で動作を確認

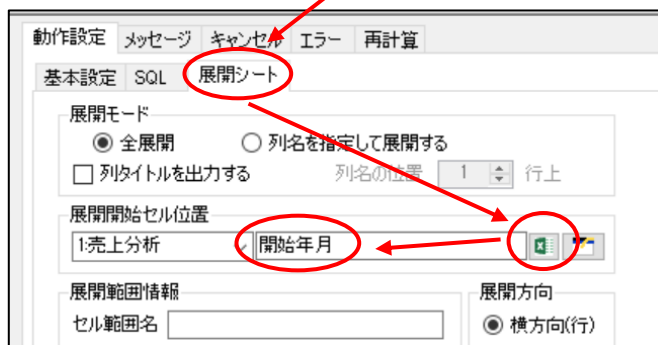
左メニュー「パレート図」を選択 → 「売上先」ボタンをクリック



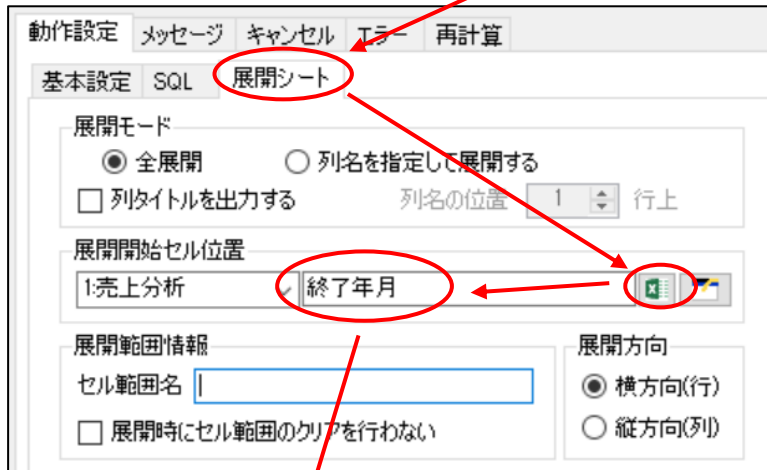
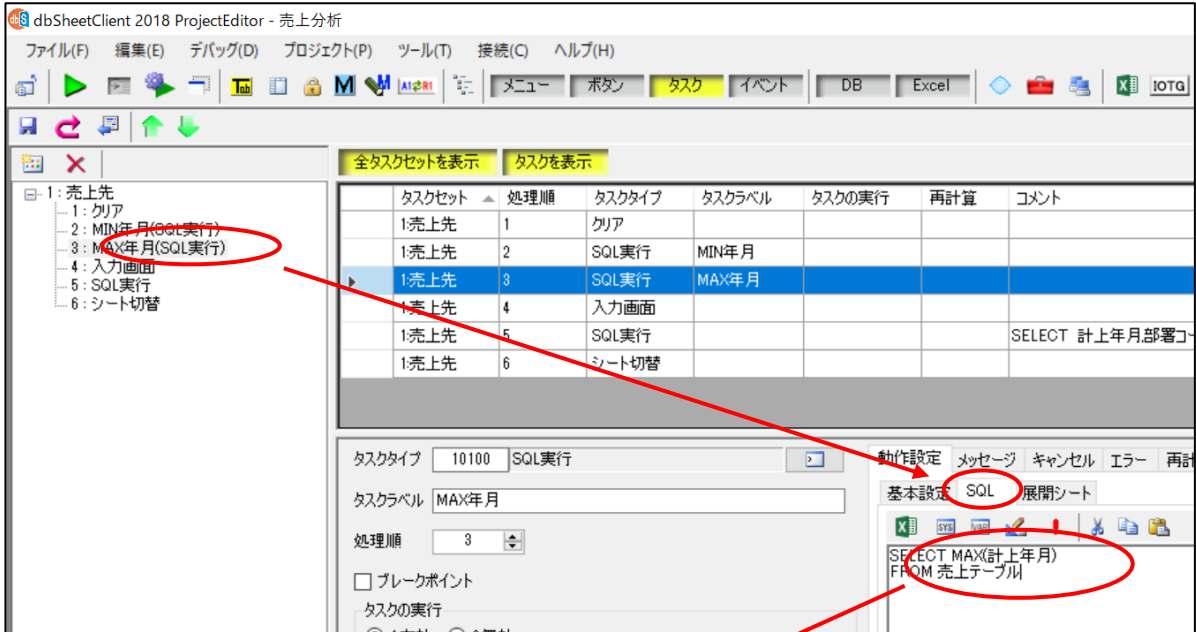
入力作業を補助するために売上テーブルの計上年月の最小値と最大値を初期表示させてみましょう。
 タスク「入力画面」の前にタスク「SQL実行」を2つ追加します。



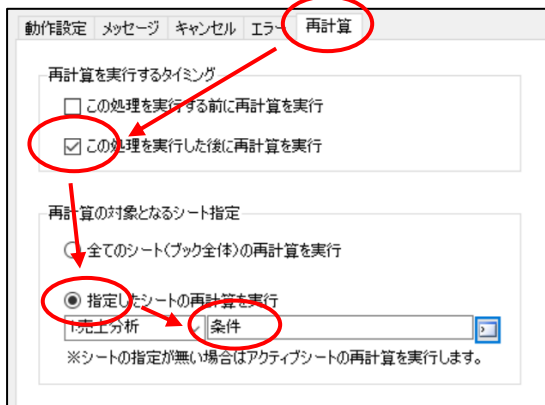
計上年月の最小値を抽出して名前「開始年月」に展開



計上年月の最大値を抽出して名前「終了年月」に展開



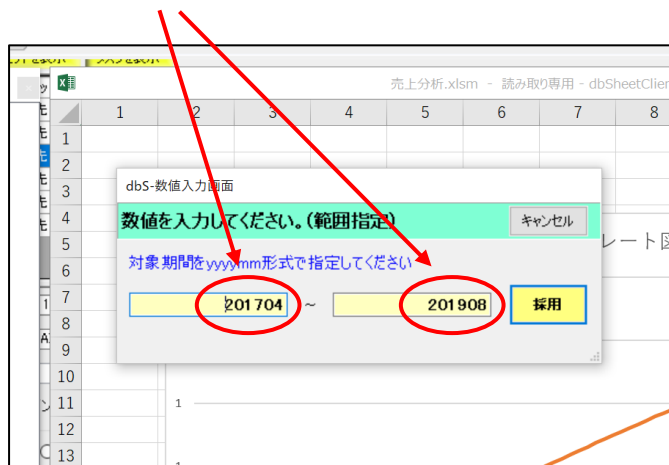
シート「条件」を再計算



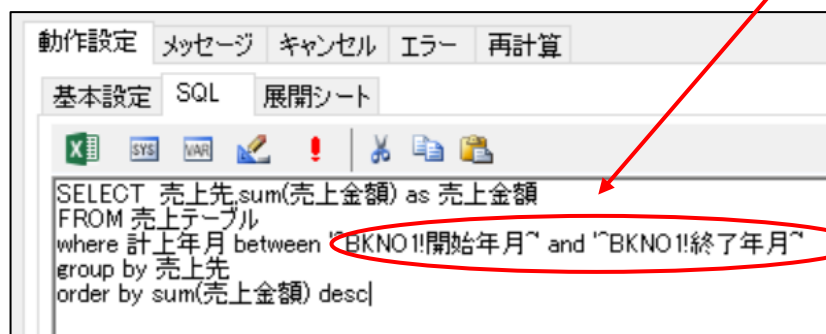
デバック実行で動作を確認

左メニュー「パレート図」を選択 → 「売上先」ボタンをクリック

対象期間範囲の初期値が入力された状態になりました。



入力された対象期間に該当する売上のみが集計されるようにSQL実行タスクを変更します。

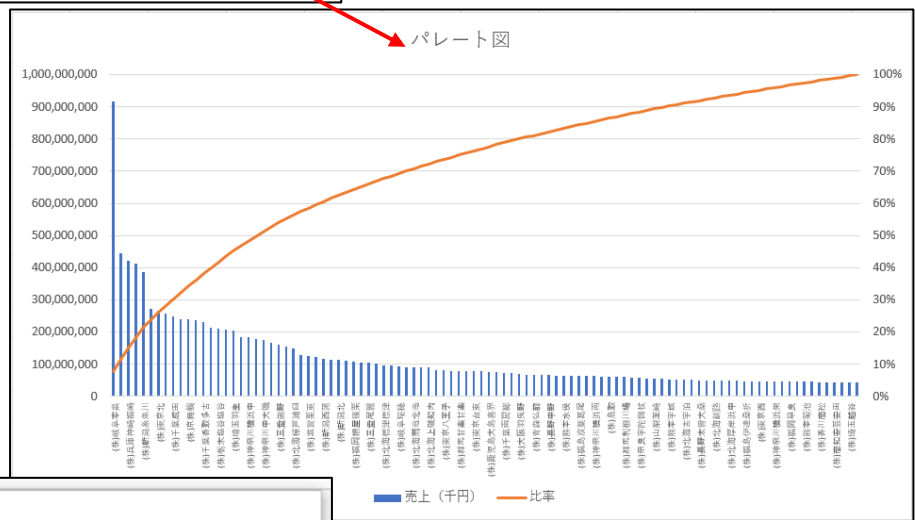


デバック実行で動作を確認

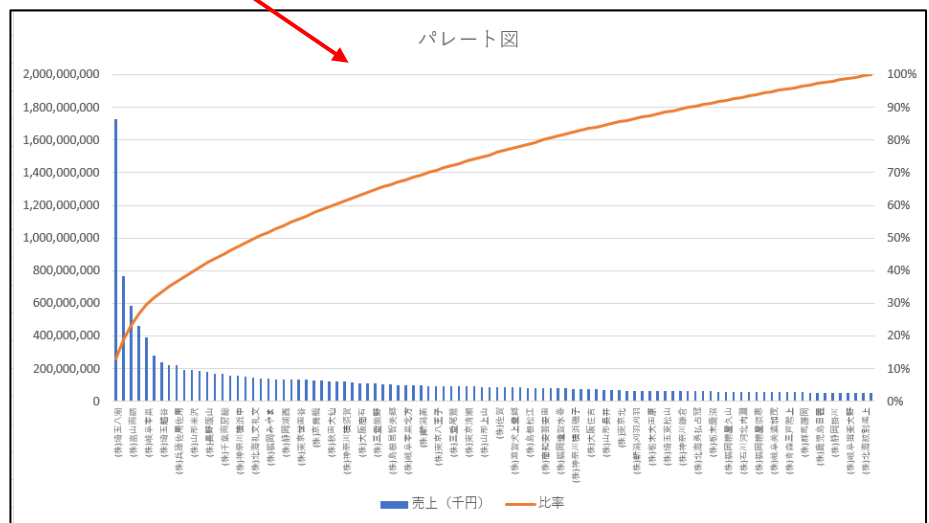
左メニュー「パレート図」を選択 → 「売上先」ボタンをクリック

対象期間を変更して「採用」ボタンをクリック → 期間によってグラフが変化することを確認します

dbS-数値入力画面
数値を入力してください。(範囲指定) キャンセル
対象期間をyyyymm形式で指定してください
201704 ~ 201803 採用



dbS-数値入力画面
数値を入力してください。(範囲指定) キャンセル
対象期間をyyyymm形式で指定してください
201804 ~ 201903 採用

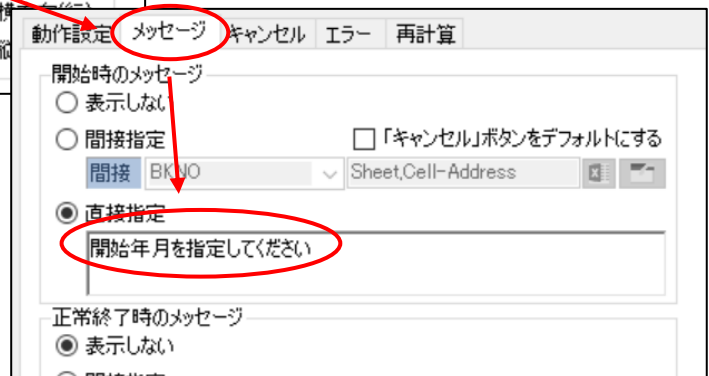
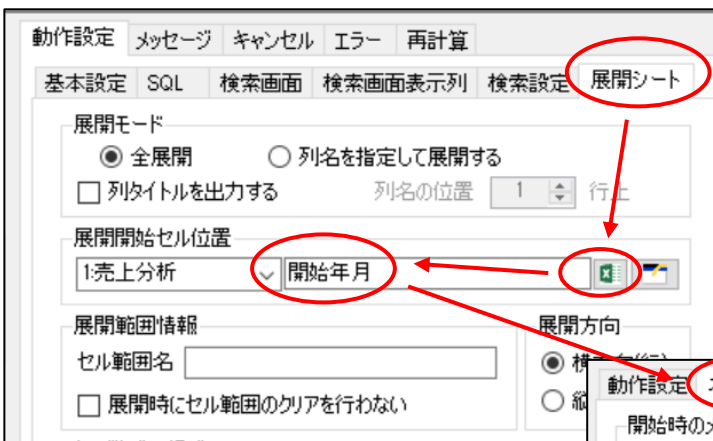
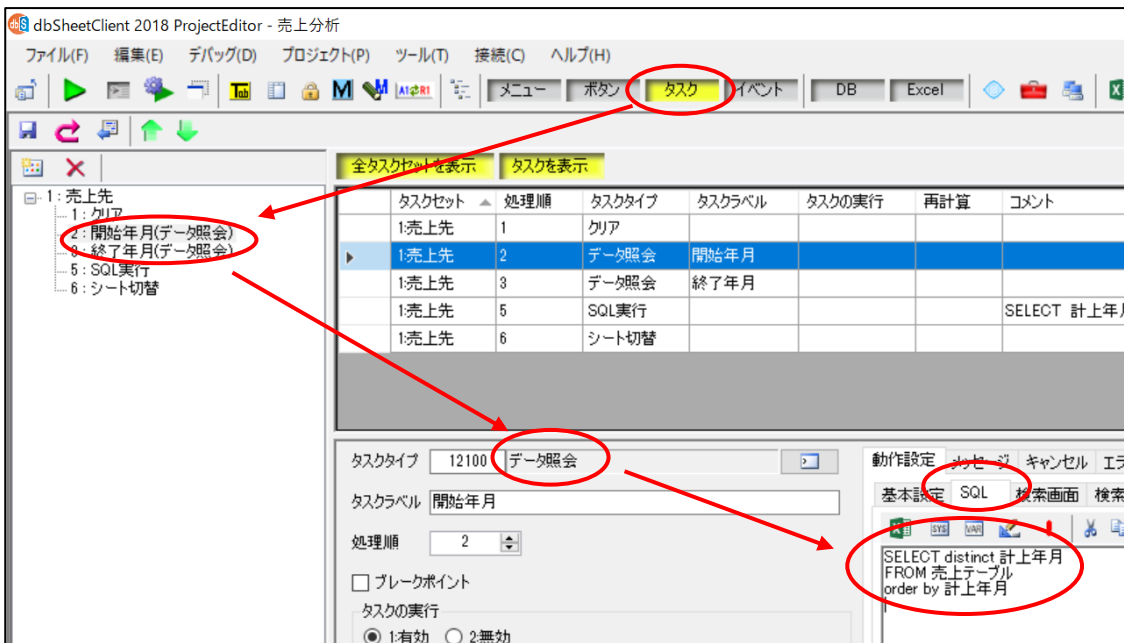


1.3.4 「データ照会」タスクを使った対象期間の指定

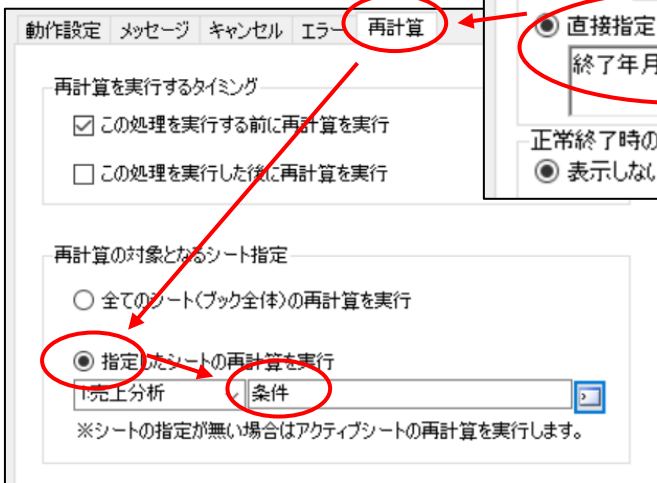
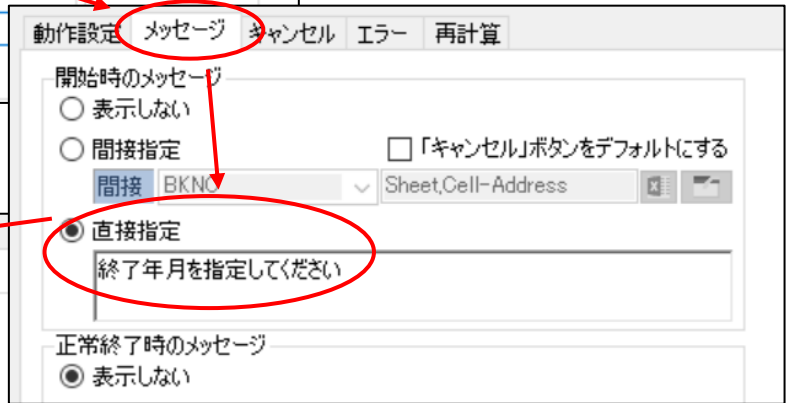
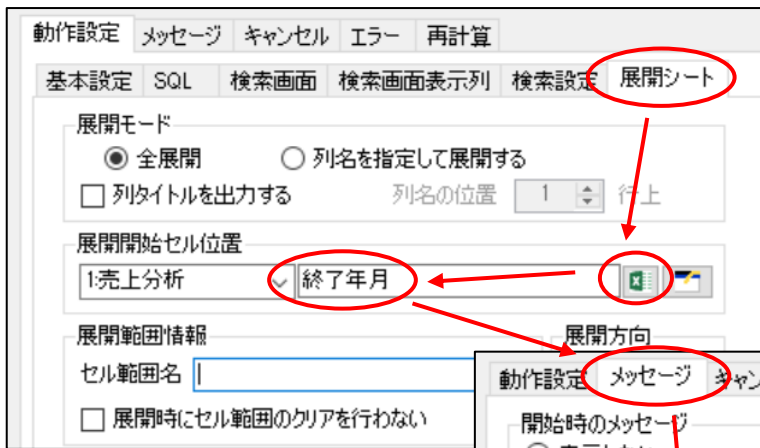
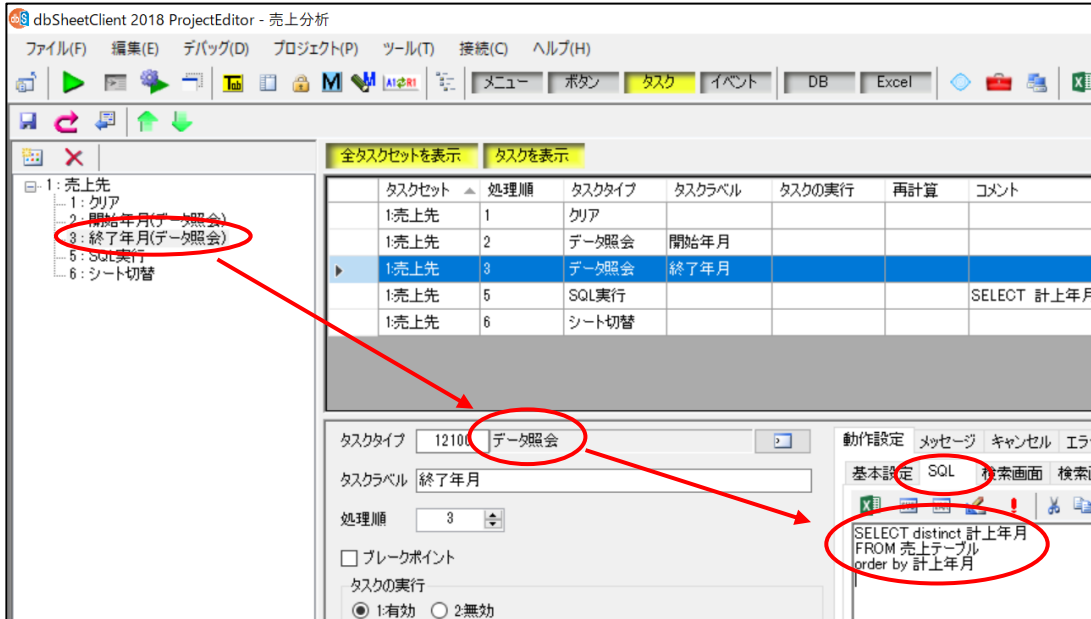
対象期間をキーボードからyyyyymm形式で入力する場合、入力ミスが発生しやすくなります。

「データ照会」タスクを使ってみましょう。

MIN 年月、MAX 年月 (SQL 実行)、入力画面タスクを削除 (対象タスクを選択して×ボタン) 代わりに「データ照会」タスクを2つ追加します。(開始年月、終了年月)



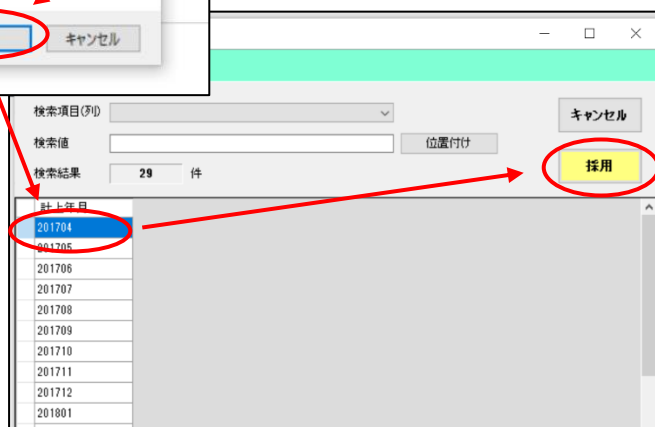
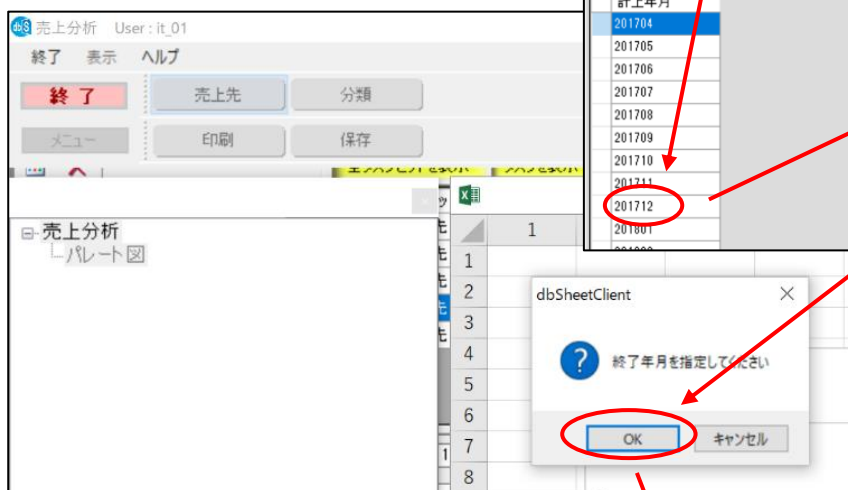
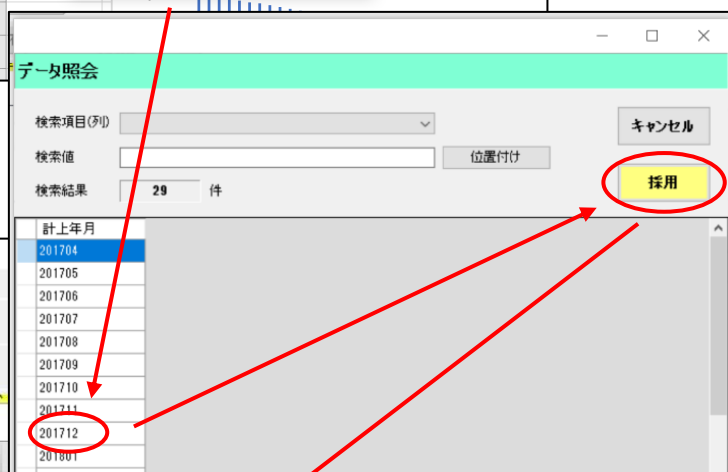
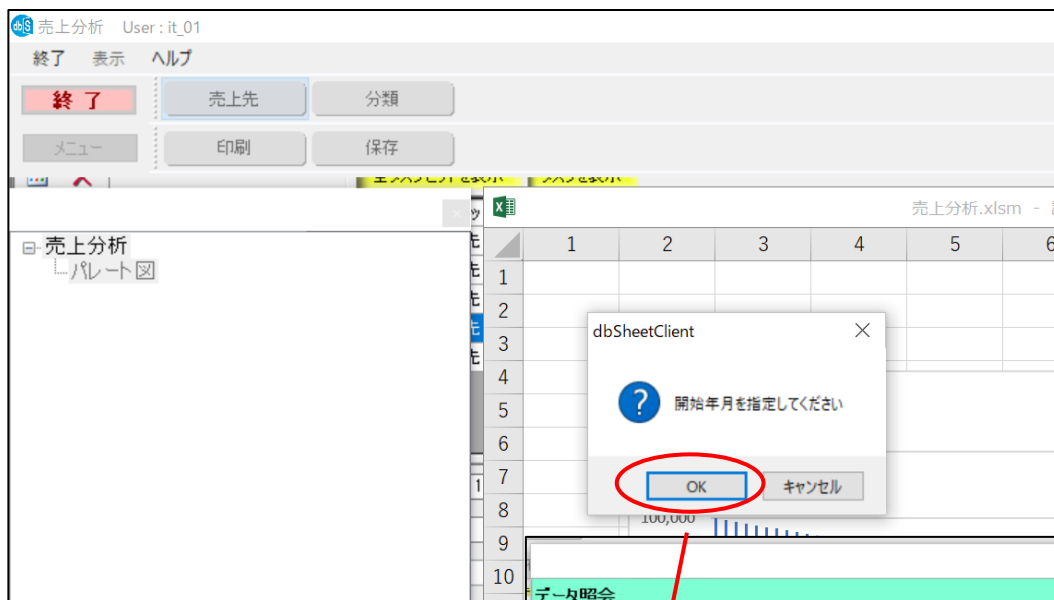
同様に終了年月を「データ照会」タスクで指定します。



デバック実行で動作を確認

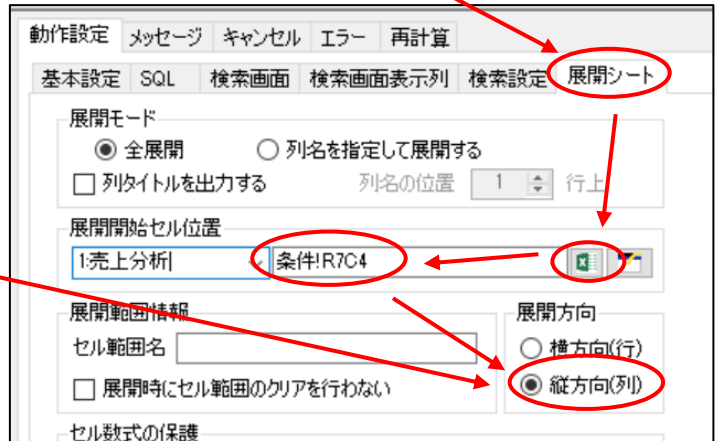
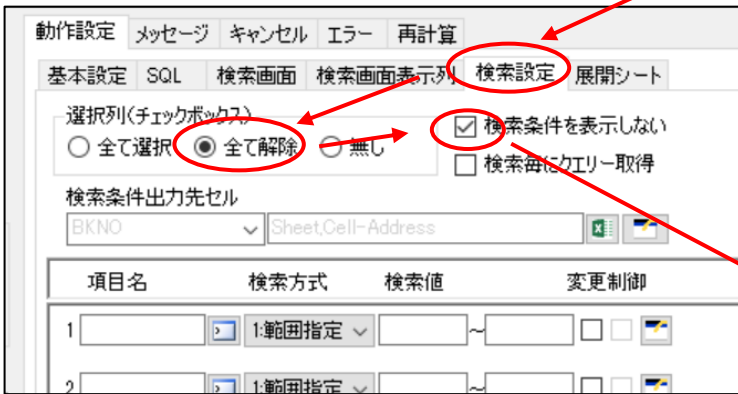
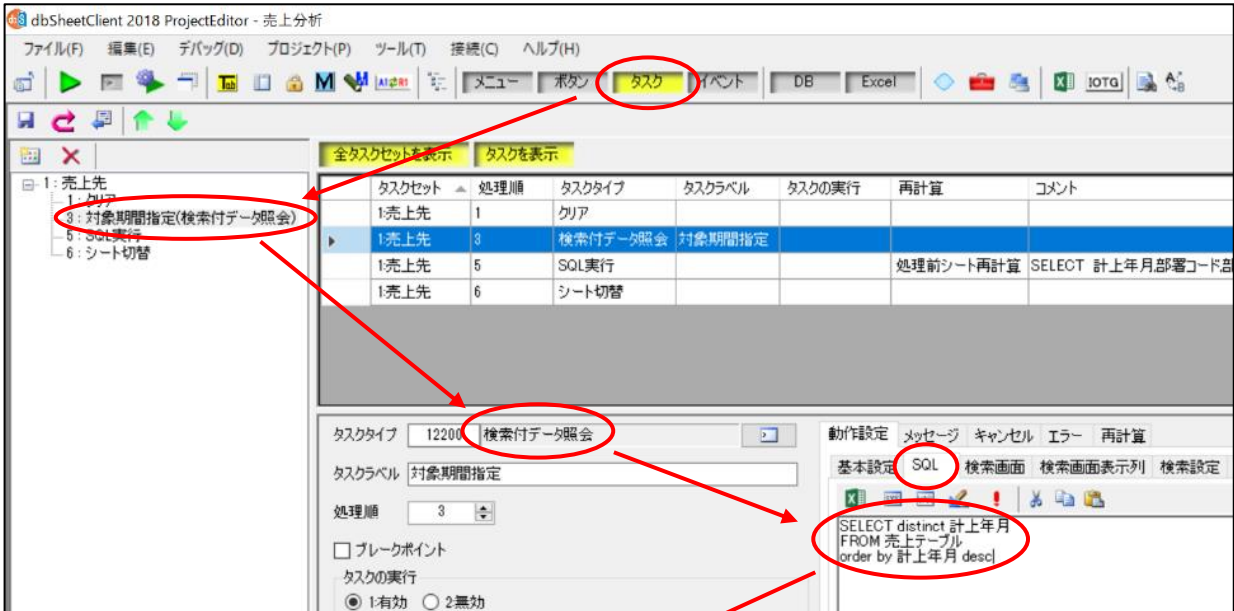
左メニュー「パレート図」を選択 → 「売上先」ボタンをクリック

設定したメッセージ「開始年月を指定してください」が表示されるか確認



1.3.5 「検索付きデータ照会」タスクを使った対象期間の指定

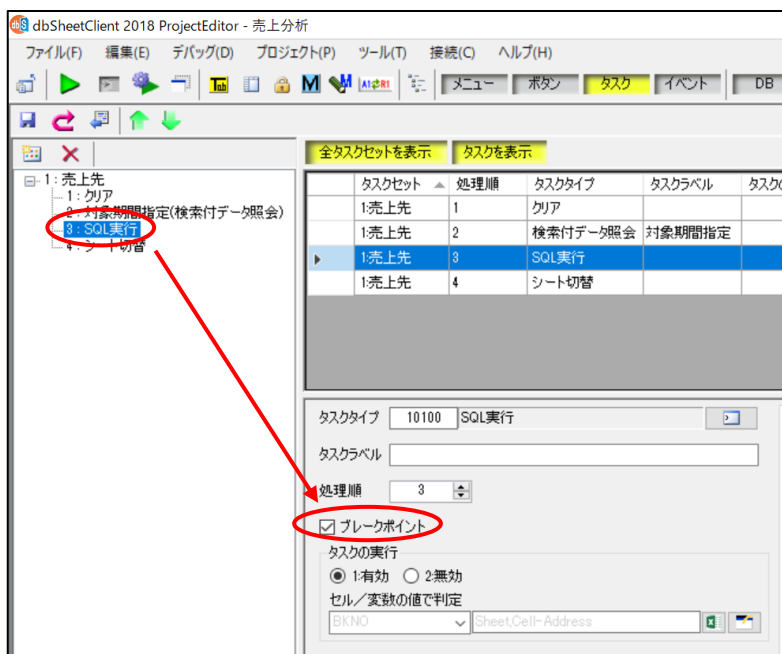
前項で開始、終了年月を指定した「データ照会」タスクの代わりに「検索付きデータ照会」タスクをセット



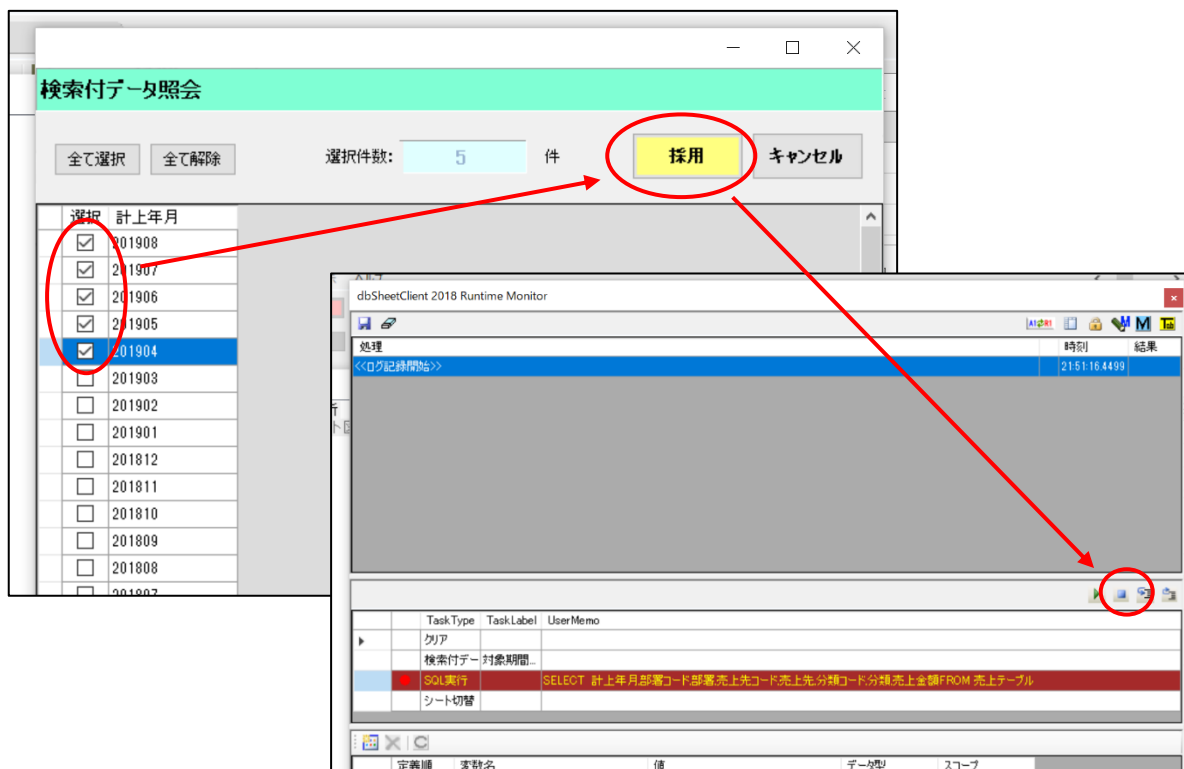
ポイント

「検索付きデータ照会」タスクで選択した年月が指定した「展開先(シート「条件」の7行4列目)」にどのように展開されるか確認してみます。

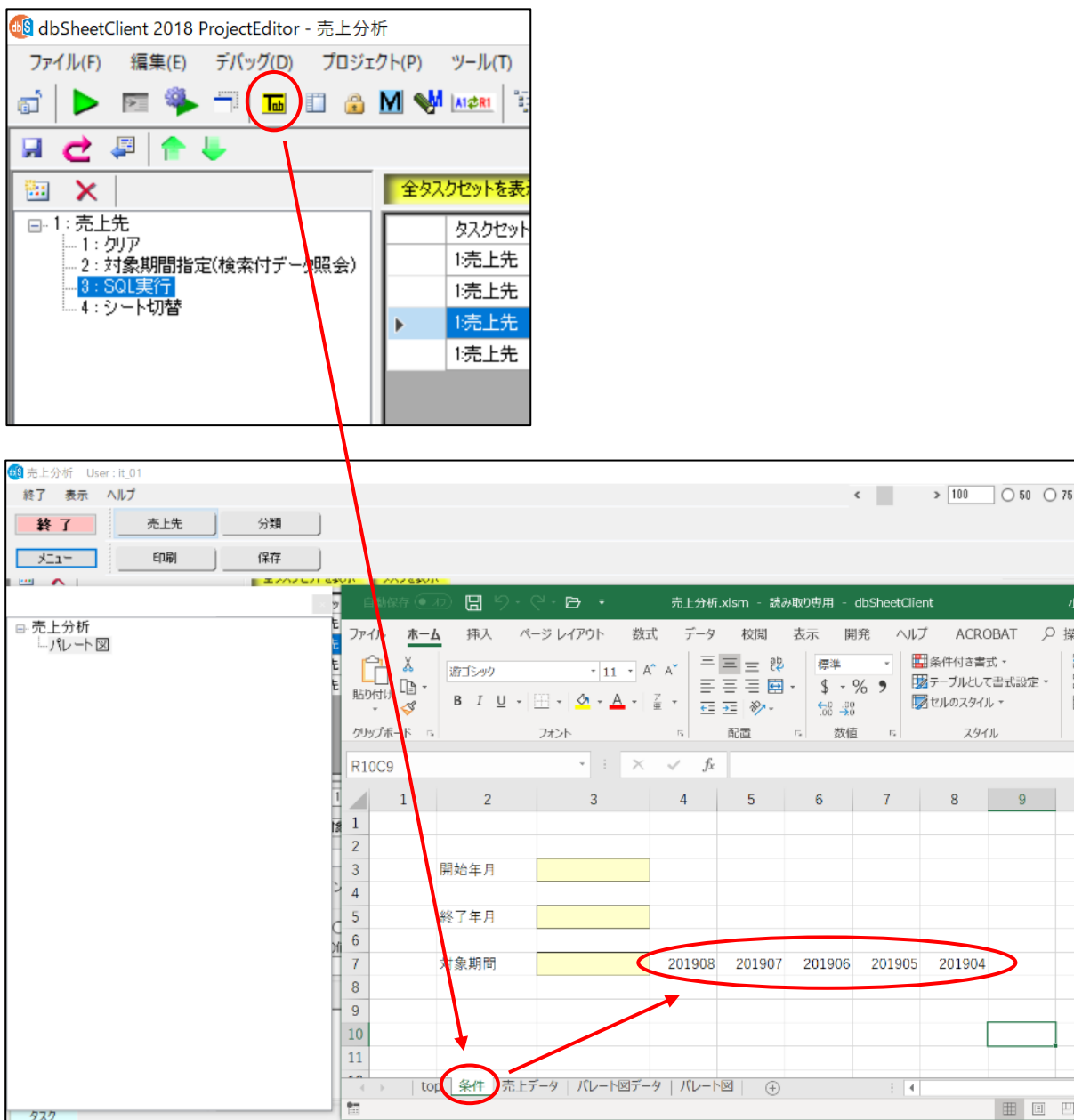
直後の「SQL実行」タスクにて「ブレークポイント」にチェックを入れます。



デバッグモードで実行 → 中断ボタン ■ をクリック



Excel シートタブ表示ボタンをクリックして「条件」シートを開きます。



「条件」シートの7行目に展開された選択した年月を「対象期間」という名前を付けたセル(7行3列)に「」で囲んで「,」区切りに結合するマクロ「bind_item」を「検索付きデータ照会」タスクの直後に追加します。

「マクロ起動」タスクを追加、実行後、「条件」シートを再計算

The screenshot shows the dbSheetClient 2018 ProjectEditor interface. On the left, a task list is shown with '3: bind_item(マクロ起動)' circled in red. The main window displays the configuration for this task, including 'タスクタイプ: 61600 マクロ起動', 'タスクラベル: bind_item', and '処理順: 3'. The '動作設定' (Action Settings) tab is active, showing 'シート名: 1売上分析' and '条件' selected. The '再計算' (Re-calculation) sub-tab is also active, showing options for when to re-calculate: 'この処理を実行した後に再計算を実行' (checked) and 'この処理を実行する前に再計算を実行'. Under '再計算の対象となるシート指定' (Specify sheets to be recalculated), '指定したシートの再計算を実行' (checked) is selected, with '1売上分析' and '条件' listed. A red arrow points from the '条件' sheet name in the task configuration to the '条件' sheet name in the re-calculation settings.

The screenshot shows an Excel spreadsheet with a formula in cell B7: `"201908","201907","201906","201905","201904"`. A red circle highlights this formula. A red arrow points from the '条件' sheet name in the re-calculation settings (from the previous screenshot) to this formula. Another red arrow points from the text 'マクロによって結合された選択年月' (Selected years combined by macro) to the formula. The spreadsheet also shows a table with columns for '開始年月' (Start Year) and '終了年月' (End Year), and a row for '対象期間' (Target Period) with the same formula.

マクロによって結合された
選択年月

マクロ「bind_item」の内容は以下の通りです。

```
Sub bind_item()  
,  
'検索付きデータ照会で選択された項目を一まとめに  
sheet_name = ActiveSheet.Name  
  
Call bind(7) '対象期間  
  
Sheets(sheet_name).Select  
  
End Sub
```

```
Function bind(r)  
  
If Sheets("条件").Cells(r, 4) <> "" Then  
Dim item_name As String  
Sheets("条件").Select  
Cells(r, 2000).Select  
Selection.End(xlToRight).Select  
Selection.End(xlToLeft).Select  
  
cc = ActiveCell.Column  
item_name = "" & Sheets("条件").Cells(r, 4) & ""  
For c = 5 To cc  
item_name = item_name & "," & Sheets("条件").Cells(r, c) & ""  
Next c  
Sheets("条件").Cells(r, 3) = item_name  
Range(Cells(r, 4), Cells(r, 2000)).Select  
Selection.ClearContents  
  
Else  
End If  
  
End Function
```

売上を集計する「SQL実行」タスクのSQL文を変更します。

The screenshot shows the 'dbSheetClient 2018 ProjectEditor' interface. The 'タスク' (Task) menu is highlighted in the top toolbar. In the task list, 'SQL実行' is selected. The configuration panel shows 'タスクタイプ' as 'SQL実行' and '処理順' as '4'. The 'SQL' tab is active, showing the following query:

```
SELECT 売上先 sum(売上金額) as 売上金額
FROM 売上テーブル
where 計上年月 in ("BKNO!対象期間")
group by 売上先
order by sum(売上金額) desc
```

The 'SQL' tab and the query text are circled in red. A zoomed-in view of the SQL editor shows the same query with red circles around the 'SQL' tab and the query text.

デバッグモードで実行

The screenshot shows the '検索付データ照会' (Search Data Query) dialog box. The '採用' (Apply) button is highlighted in red. The dialog shows a list of years with checkboxes, and the years 201906, 201905, and 201904 are selected. The '採用' button is circled in red. The Pareto chart shows sales amounts for various departments, with the top few departments having significantly higher sales.

1.3.6 部署、分類を絞り込む

「検索付きデータ照会」タスクをもう一つ追加します。(部署選択)

The screenshot shows the 'Task Editor' window with a list of tasks. The 4th task, '検索付きデータ照会' (Search with Data Query), is selected. The configuration for this task is shown below the list:

- タスクタイプ: 12:00 検索付きデータ照会
- タスクラベル: 対象部署指定
- 処理順: 4
- SQL: `SELECT distinct 部署コード部署 FROM 売上テーブル where 計上年月 in ("BKNO"!!対象期間) order by 部署コード`

部署を選択するSQL文に
指定した対象期間を加えること

The '検索設定' (Search Settings) dialog box is shown with the following settings:

- 基本設定: 検索条件を表示しない (checked)
- 検索条件出力先セル: BKNO, Sheet,Cell-Address

The '展開シート' (Expand Sheet) dialog box is shown with the following settings:

- 展開モード: 全展開 (selected)
- 展開開始セル位置: 条件!R9C4
- 展開方向: 縦方向(列) (selected)

対象期間指定と同様に直後に「マクロ起動」タスク (bind_item) を追加します。

「条件」シートの9行目に展開された選択部署を1つのセルに統合するマクロ

売上を集計する「SQL実行」タスクのSQL文に部署を指定する条件文を加えます。

The screenshot shows the 'dbSheetClient 2018 ProjectEditor - 売上分析' interface. A task list table is visible with the following data:

タスクセット	処理順	タスクタイプ	タスクラベル	タスクの実行	再計算	コメント
1.売上先	1	クリア				
2.対象期間指定(検索付データ照会)	2	検索付データ照会	対象期間指定		処理後シート再計算	
3. bind_item(マクロ起動)	3	マクロ起動	bind_item		処理後シート再計算	
4.対象部署指定(検索付データ照会)	4	検索付データ照会	対象部署指定		処理後シート再計算	
5. bind_item(マクロ起動)	5	マクロ起動	bind_item		処理後シート再計算	
6. SQL実行	6	SQL実行			処理前シート再計算	SELECT 計上年月部署コード部
7.シート切替	7	シート切替				

The 'SQL実行' task (row 6) is selected, and its configuration panel is shown below. The 'SQL' tab is active, displaying the following SQL query:

```
SELECT 売上先,sum(売上金額) as 売上金額
FROM 売上テーブル
where 計上年月 in ('BKN0!対象期間')
and 部署コード in ('BKN0!対象部署コード')
group by 売上先
order by sum(売上金額) desc
```

Red circles in the original image highlight the 'SQL実行' task in the list and the '部署コード in' condition in the SQL query.

デバッグモードで実行 → 対象期間選択の後、部署を選択する画面が表示されたら OK

The screenshot shows the '検索付データ照会' dialog box. It has a title bar with a close button. Below the title bar, there are buttons for '全て選択' (Select All), '全て解除' (Deselect All), and '採用' (Apply), along with a 'キャンセル' (Cancel) button. A label '選択件数:' (Selected Item Count) shows '0' items. Below this is a table with columns '選択' (Select), '部署コード' (Department Code), and '部署' (Department):

選択	部署コード	部署
<input checked="" type="checkbox"/>	A01	西東京支店
<input type="checkbox"/>	A02	東東京支店
<input type="checkbox"/>	B01	千葉支店
<input type="checkbox"/>	B02	埼玉支店
<input type="checkbox"/>	C01	名古屋支店
<input type="checkbox"/>	C02	大阪支店
<input type="checkbox"/>	D01	九州支店
<input type="checkbox"/>	D02	四国支店
<input type="checkbox"/>	D03	北陸支店
<input type="checkbox"/>	D04	北海道支店

さらに「検索付データ照会」タスクをもう一つ追加します。(分類選択)

タスクセット	処理順	タスクタイプ	タスクラベル	タスクの実行	再計算	コメント
1:売上先	3	マクロ起動	bind_item		処理後シート再計算	
1:売上先	4	検索付データ照会	対象部署指定		処理後シート再計算	
1:売上先	5	マクロ起動	bind_item		処理後シート再計算	
1:売上先	6	検索付データ照会	対象分類指定		処理後シート再計算	
1:売上先	7	マクロ起動	bind_item		処理後シート再計算	
1:売上先	8	SQL実行			処理前シート再計算	SELECT 計上年月,部署コー
1:売上先	9	シート切替				

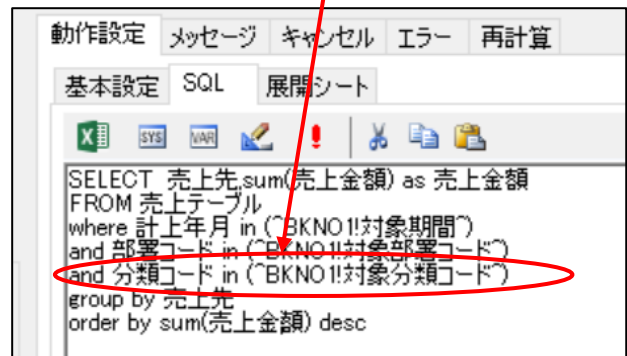
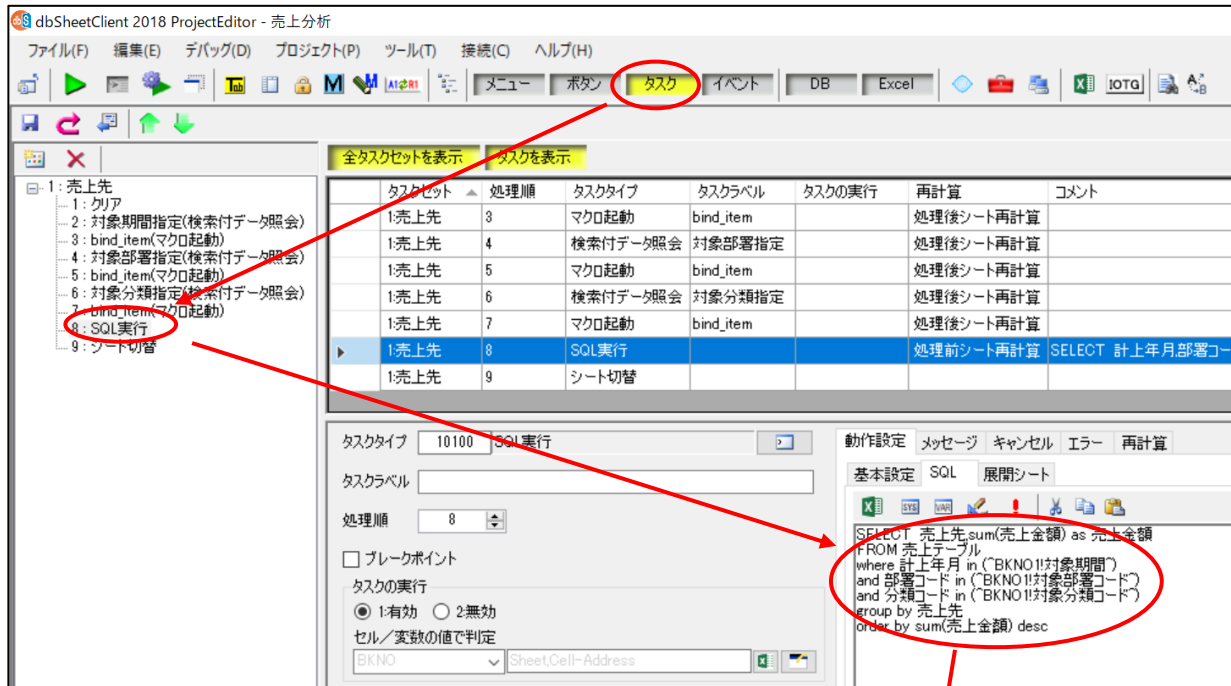
タスクタイプ: 12:00 検索付データ照会
 タスクラベル: 対象分類指定
 処理順: 6
 基本設定: SQL
 SQL: SELECT distinct 分類コード,分類 FROM 売上テーブル where 計上年月 in ('BKN0!!対象期間') and 部署コード in ('BKN0!!対象部署コード') order by 分類コード

動作設定: メッセージ, キャンセル, エラー, 再計算
 基本設定: SQL, 検索画面, 検索画面表示列, 検索設定, 展開シート
 展開モード: 全展開
 展開開始セル位置: 1:売上分析, 条件!R12C4
 展開方向: 縦方向(列)
 セル数式の保護: 保護しない

1	2	3	4	5
1				
2				
3	開始年月			
4				
5	終了年月			
6				
7	対象期間			
8				
9	対象部署コード			
10	対象部署			
11				
12	対象分類コード			
13	対象分類			
14				
15				
16				
17				

条件 | 売上データ | パレト図データ | パレト図

売上を集計する「SQL実行」タスクのSQL文に分類を指定する条件文を加えます。

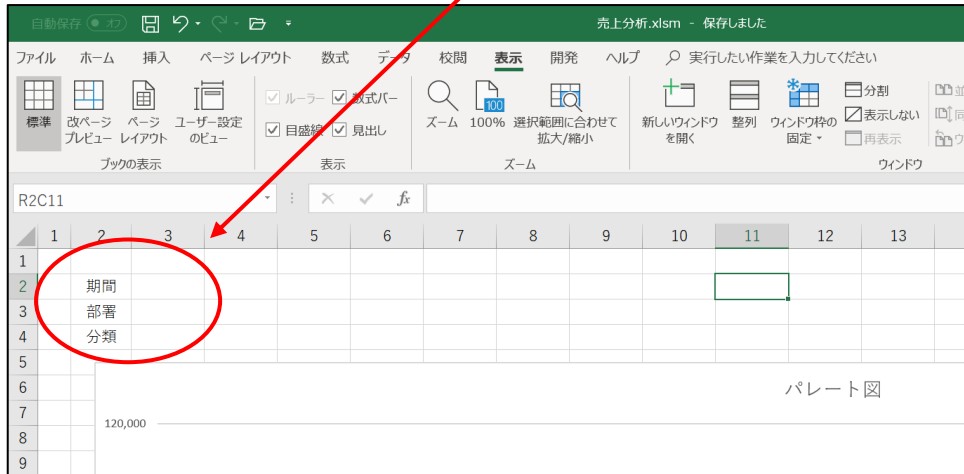


デバッグモードで実行 → 部署選択の後、分類を選択する画面が表示されたらOK

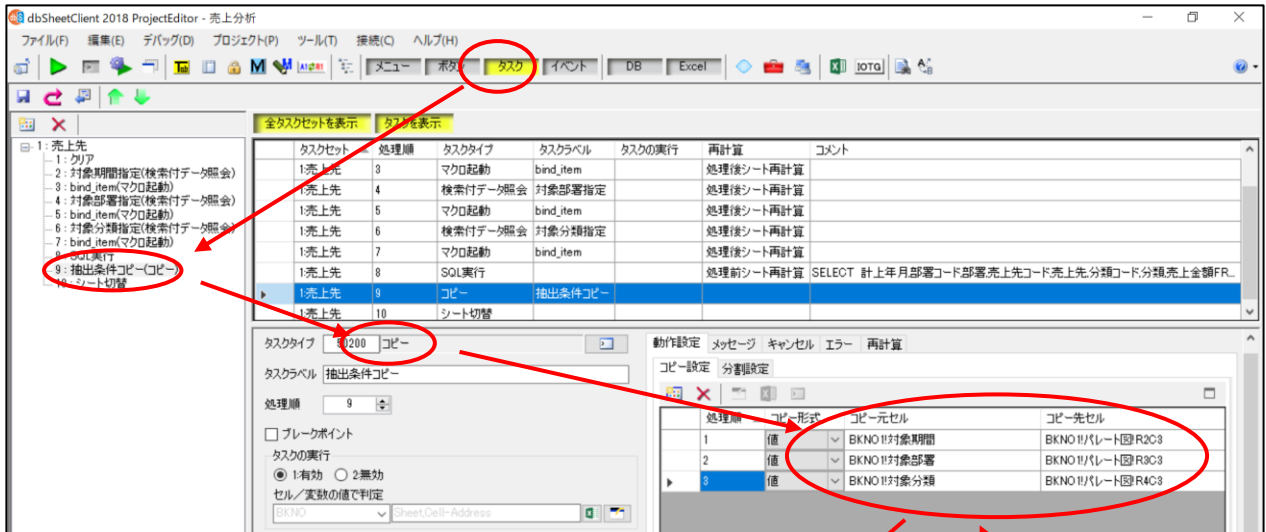


1.3.7 抽出条件を表記

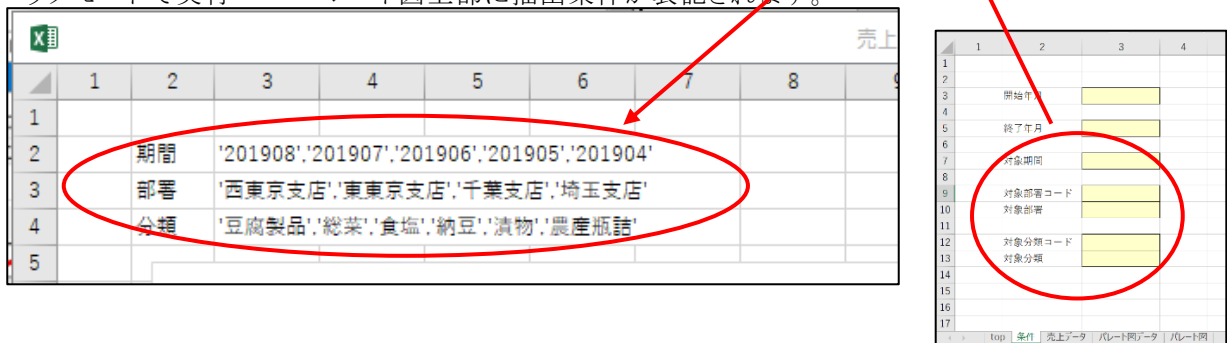
シート「パレット図」の上部に抽出条件を表記するスペースを準備します。



「コピー」タスクにて、シート「条件」に展開された抽出条件をコピーします。

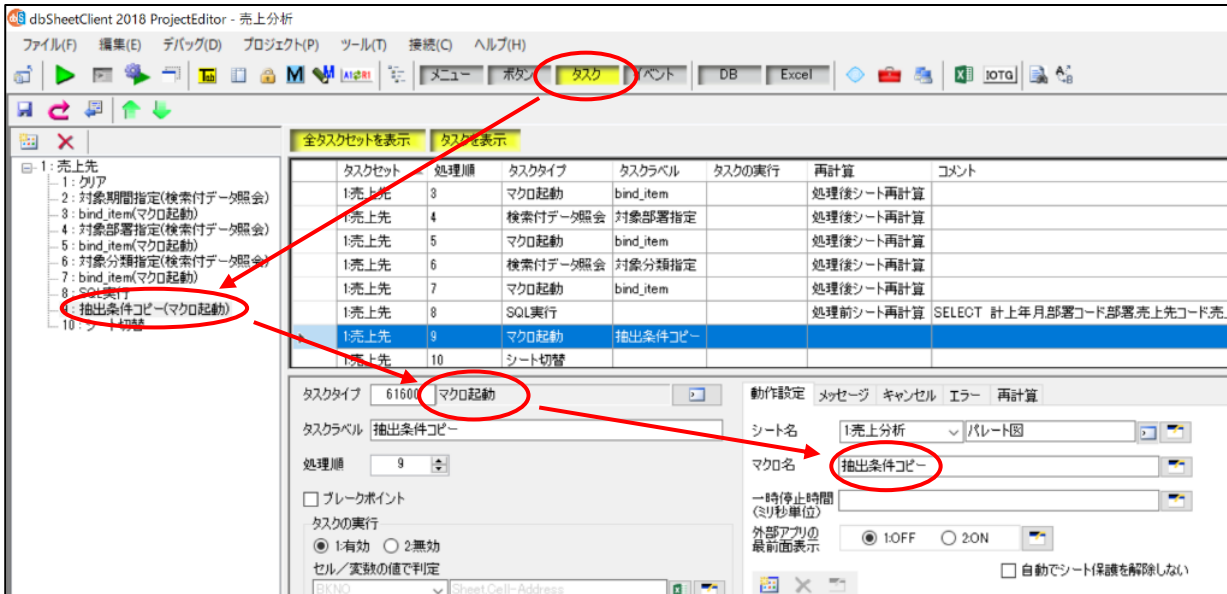


デバッグモードで実行 → パレット図上部に抽出条件が表記されます。

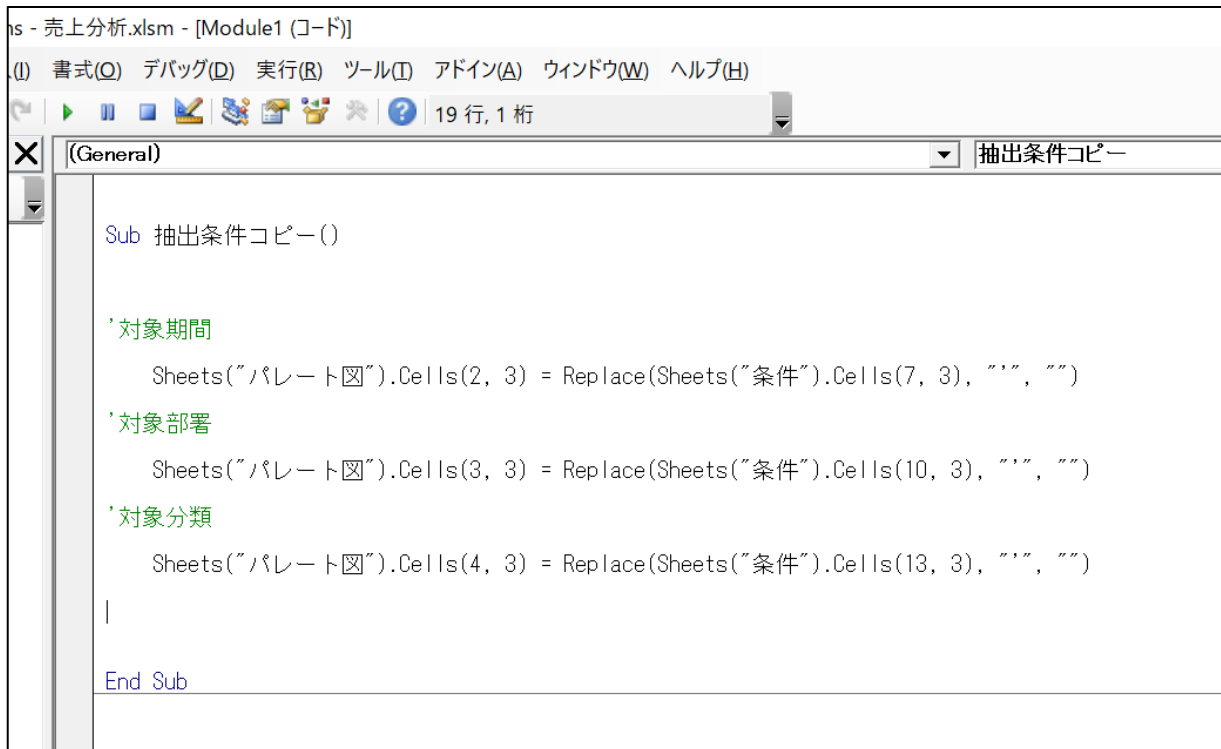


「コピー」タスクの代わりに「マクロ起動」タスクを使ってみます。

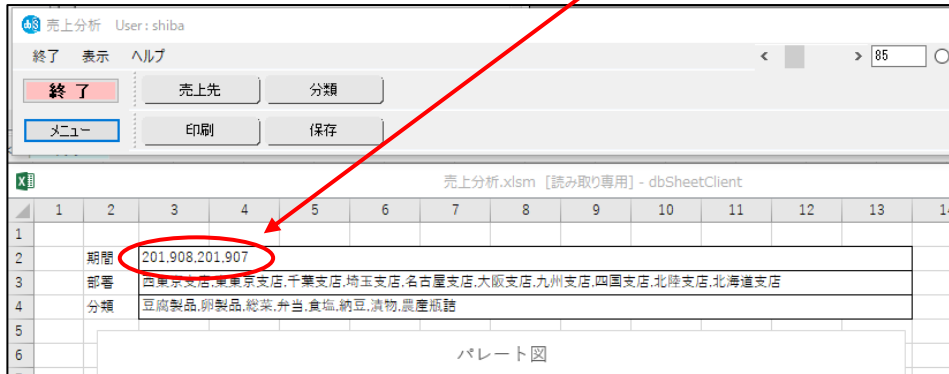
※コピーと同時に「'」が見苦しいのでとってしまう。



マクロプログラム



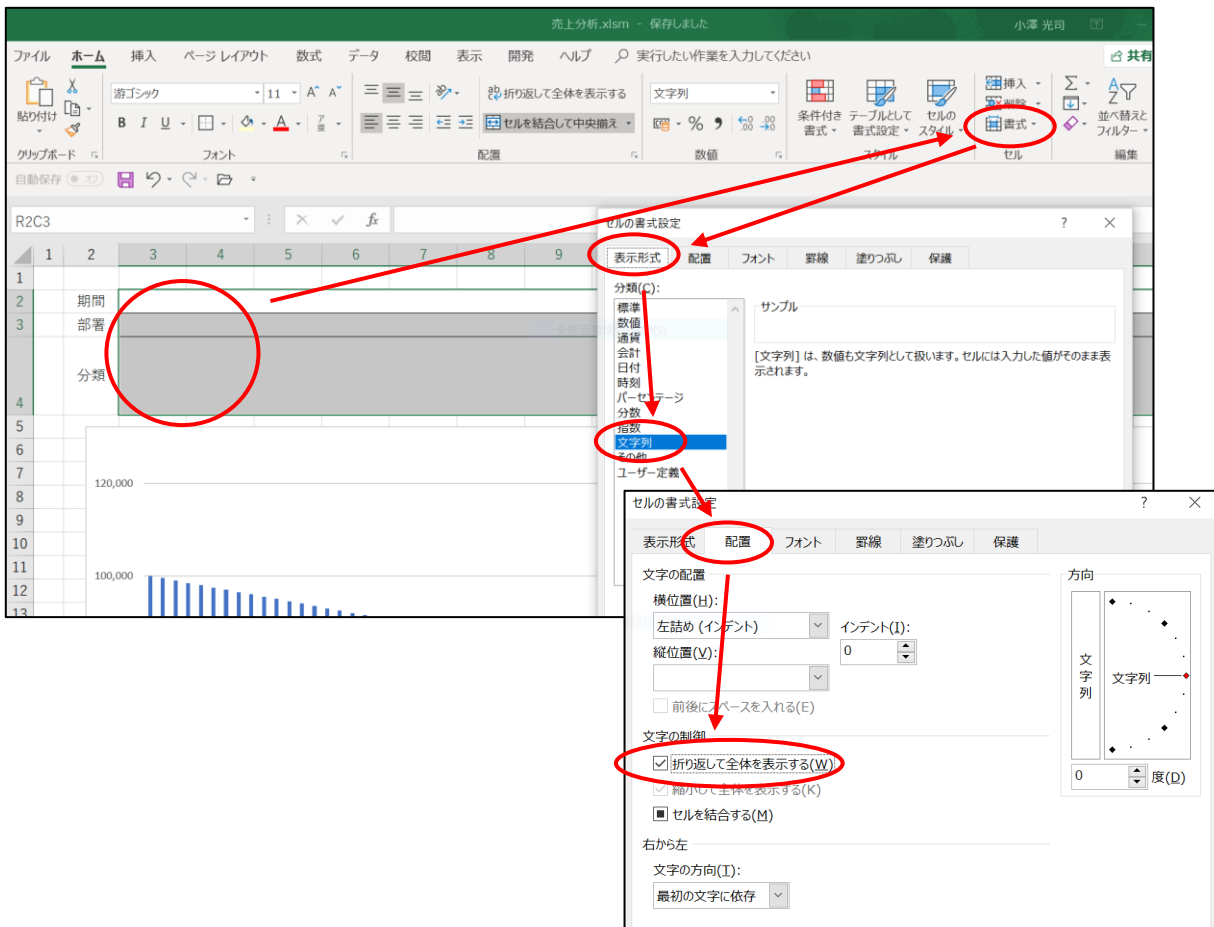
デバッグモードで実行 → 「'」が取れましたが、期間の表記がおかしくなっています。(次項で修正)



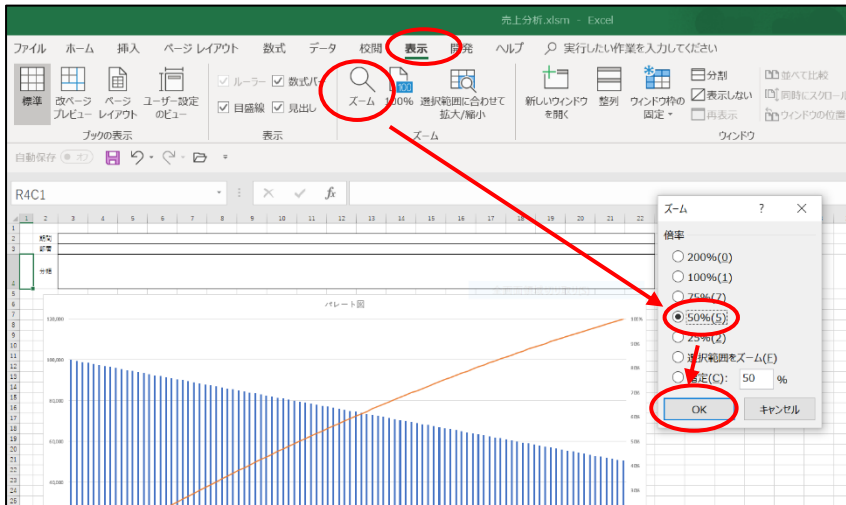
1.3.8 初期表示画面の調整

シート「パレート図」のセル(2,3)～セル(4,3)の書式を文字列に変更します。

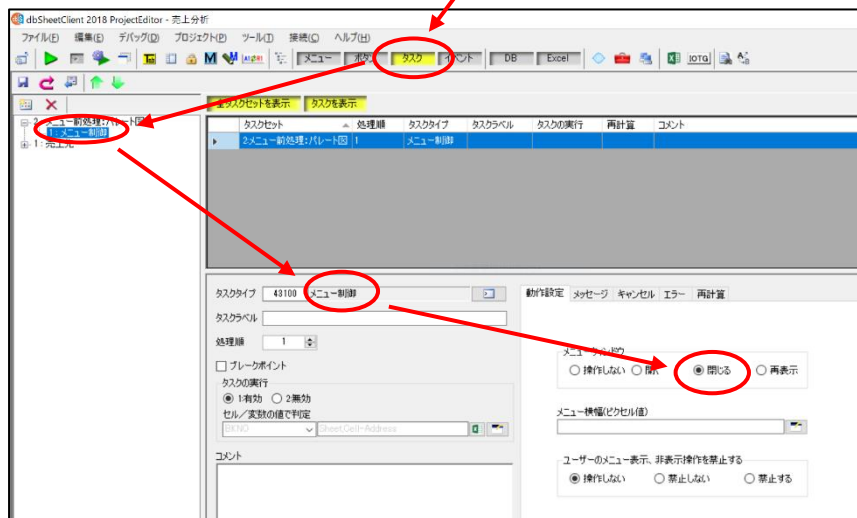
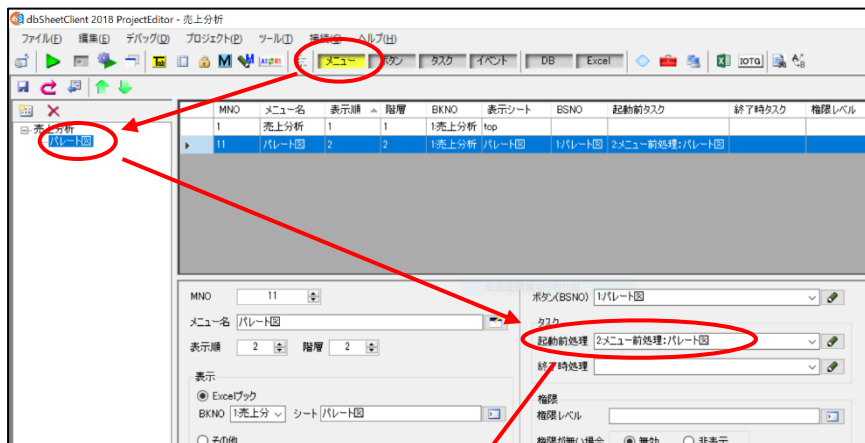
「折り返して全体を表示する」にチェックを入れます。分類の入る行の高さを大きくします。



シート「パレート図」の「ズーム」を 50%に変更します。

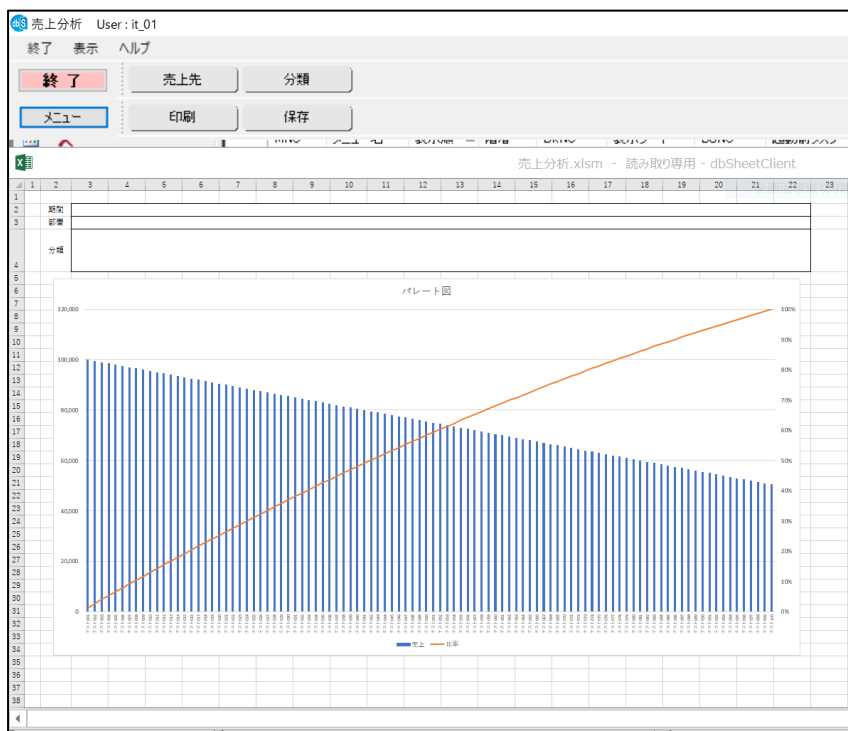


メニュー「パレート図」に「起動前処理」タスクを追加します。 → メニューを閉じるタスク



デバッグモードで実行

左メニューが非表示になりパレート図が画面いっぱいに表示されることを確認します。

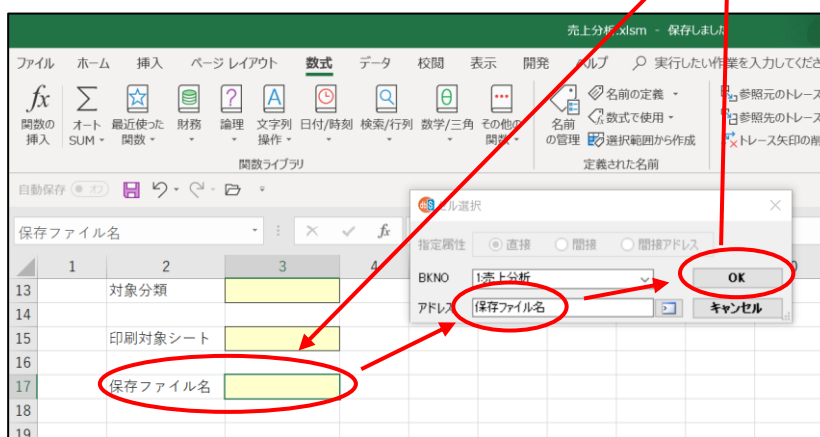
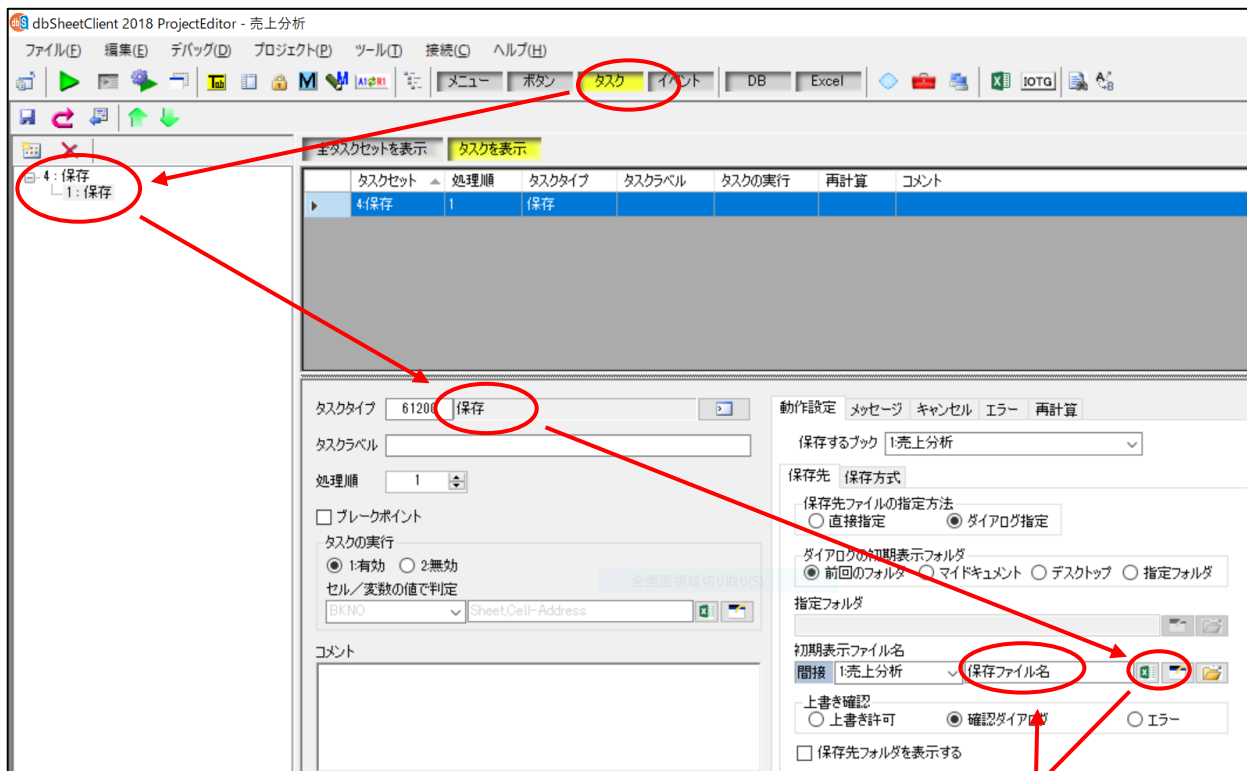


1.3.9 印刷ボタン、保存ボタンの設定

印刷ボタンに割り付けるタスクセットを設定します → 「印刷」タスク

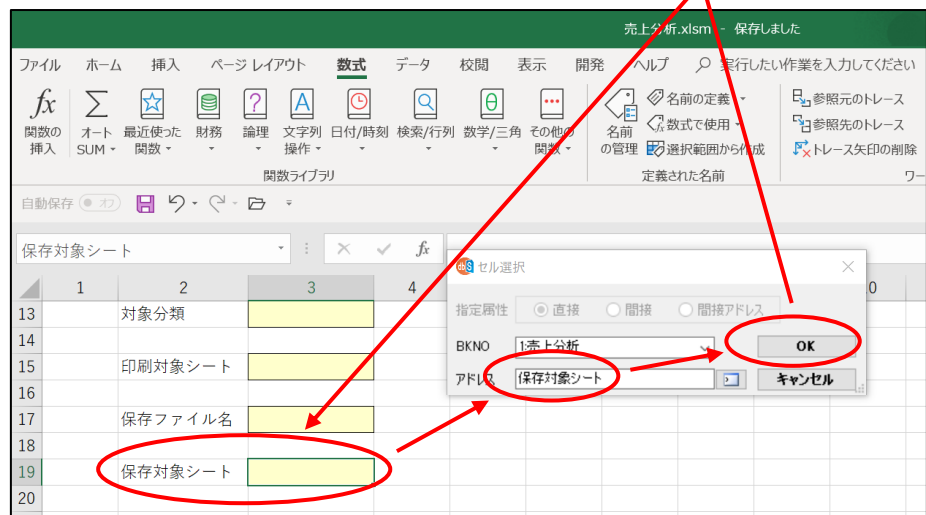
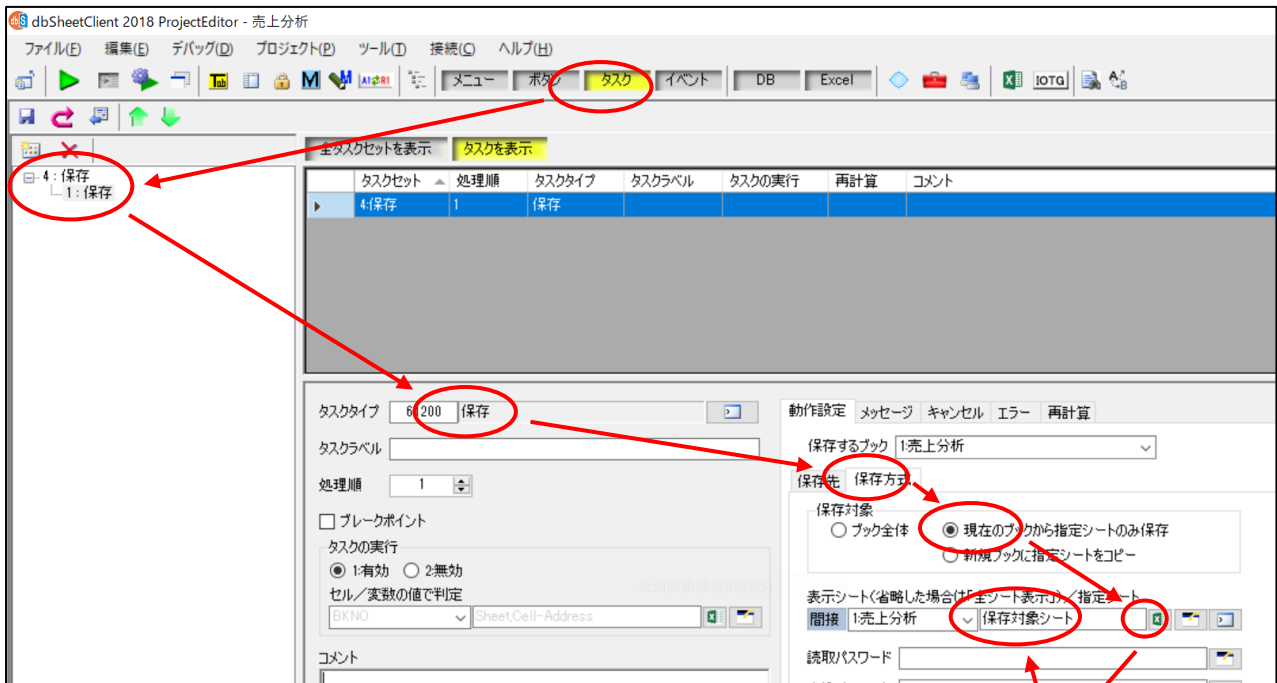
The screenshot shows the 'Task Editor' window. The 'タスク' (Task) button in the menu bar is circled in red. Below it, the task list shows '3:印刷' (3:Print) with '1:印刷' (1:Print) as a sub-task. The task properties for '印刷' are shown, with 'タスクタイプ' (Task Type) set to '印刷' (Print). In the '印刷対象' (Print Target) section, the '印刷対象シート' (Print target sheet) is set to '印刷対象シート' (Print target sheet), which is also circled in red. The '印刷対象セル範囲' (Print target cell range) is set to 'BKNO' and '印刷非対象シート' (Print non-target sheet) is set to 'BKNO'.

保存ボタンに割り付けるタスクセットを設定します → 「保存」タスク
 「条件」シートに「保存ファイル名」という名前を付けたセルを作ります。

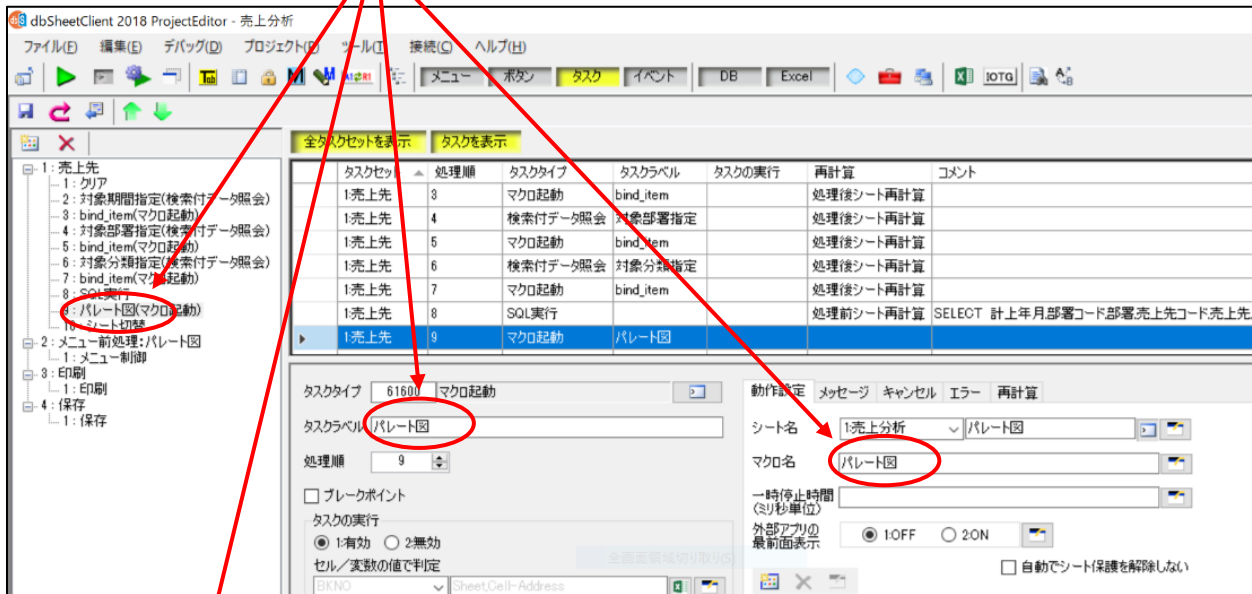


「保存方式」に「現在のブックから指定シートのみ保存」を選択します。

「条件」シートに「保存対象シート」という名前を付けたセルを作成して指定シートに設定します。



「抽出条件コピー」マクロを「パレート図」という名前に変えます。



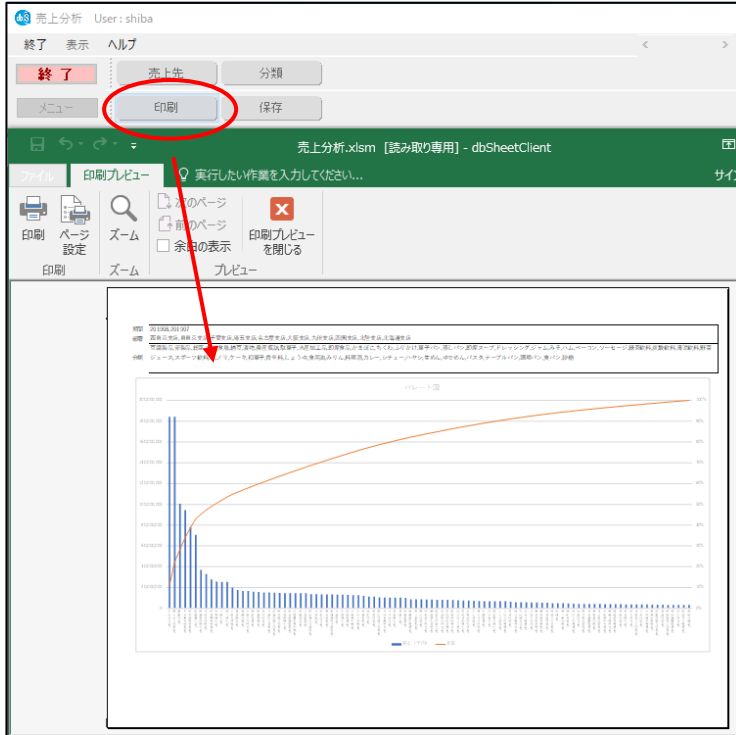
マクロに「印刷対象シート」、「保存ファイル名」、「保存対象シート」をセットする記述を追加します。

```

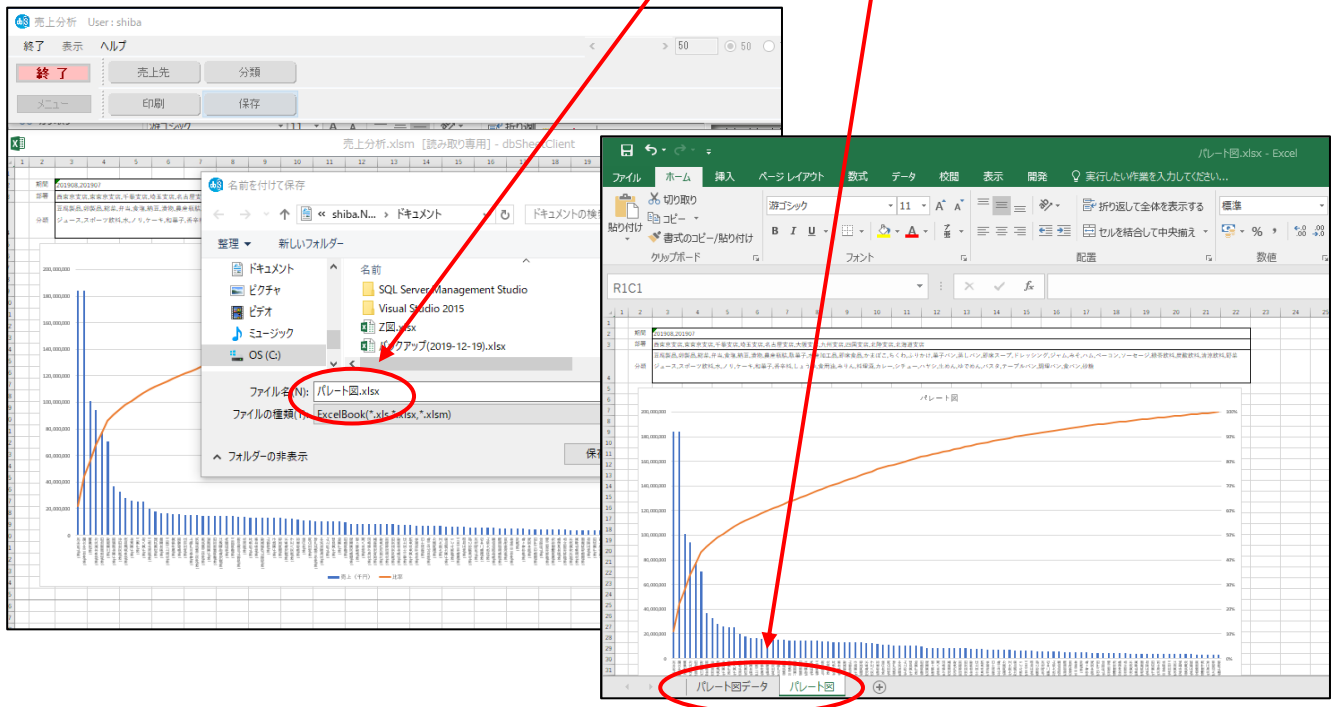
Sub パレート図()
'抽出条件をコピー
'対象期間
Sheets("パレート図").Cells(2, 3) = Replace(Sheets("条件").Cells(7, 3), "", "")
'対象部署
Sheets("パレート図").Cells(3, 3) = Replace(Sheets("条件").Cells(10, 3), "", "")
'対象分類
Sheets("パレート図").Cells(4, 3) = Replace(Sheets("条件").Cells(13, 3), "", "")
'印刷対象シート
Sheets("条件").Cells(15, 3) = "パレート図"
'保存ファイル名
Sheets("条件").Cells(17, 3) = "パレート図.xlsx"
'保存対象シート
Sheets("条件").Cells(19, 3) = "パレート図,パレート図データ"
End Sub
    
```

デバッグモードで実行

パレート図作成後、「印刷」ボタンをクリックして、シート「パレート図」の印刷プレビューを確認します。



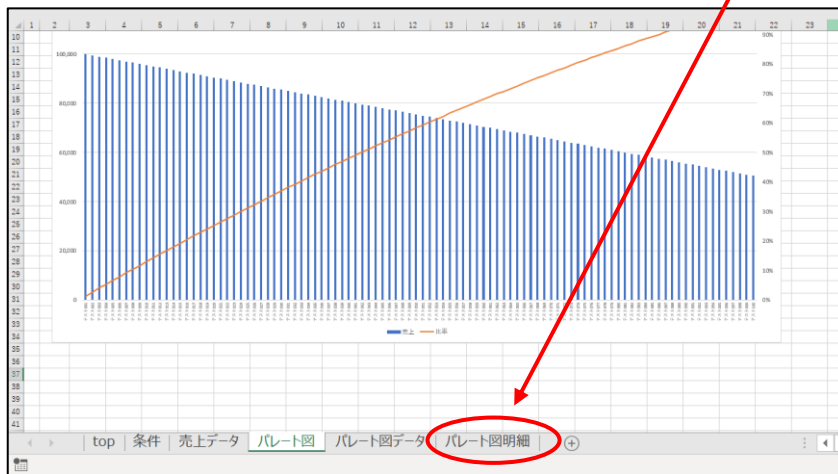
「保存」ボタンをクリックして保存ファイル名の初期表示が「パレート図.xlsx」になっているか保存されたファイルを開いてシート「パレート図」と「パレート図データ」が存在するか確認します。



1.3.10 明細を展開する

ユーザーは意外に先入観を持っていて客観的データを見せられると違和感を覚えるようです。「このグラフはおかしい。この取引先はもっと売上が多いはずだとか」等々、問合せがあります。いちいち説明するより、根拠となるデータを一緒に提供することをオススメします。

Excel ボタンをクリックして Excel ファイルを呼び出して「パレート図明細」というシートを追加します。



集計データを抽出する「SQL実行」タスクをコピーして、すぐ後ろに貼り付けます。

SQL文を変更します。

The image shows the dbSheetClient 2018 ProjectEditor interface. The main window displays a task execution table with columns for task ID, task name, task type, task label, task execution, recalculation, and comments. The 'SQL実行' (SQL Execution) task is highlighted. Below the table, the 'SQL' tab is active, showing a SQL query. The query is circled in red, and a red arrow points to it from the 'SQL実行' task in the table above.

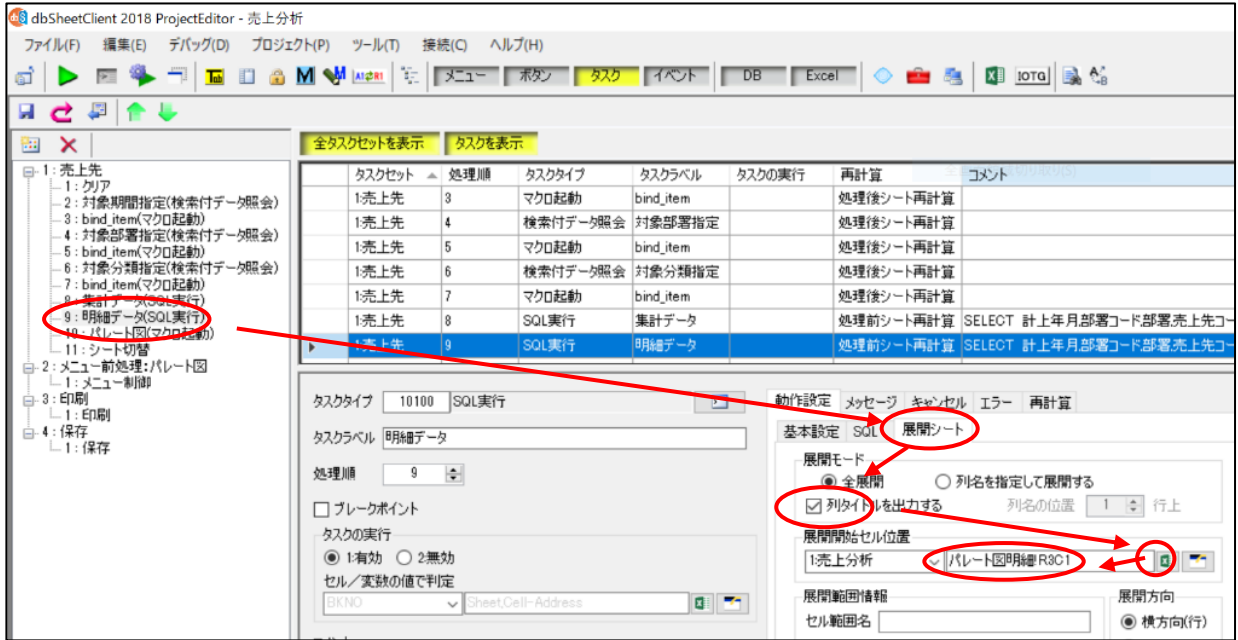
タスクID	処理種	タスクタイプ	タスクラベル	タスクの実行	再計算	コメント
1	売上先	8	マクロ起動	bind_item		
1	売上先	4	検索付データ照会	対象部署指定	処理後シート再計算	
1	売上先	5	マクロ起動	bind_item	処理後シート再計算	
1	売上先	6	検索付データ照会	対象分類指定	処理後シート再計算	
1	売上先	7	マクロ起動	bind_item	処理後シート再計算	
1	売上先	8	SQL実行	集計データ	処理後シート再計算	SELECT 計上年月部署コード部署売上先コード売上先分類
1	売上先	9	SQL実行	明細データ	処理後シート再計算	SELECT 計上年月部署コード部署売上先コード売上先分類

```

SELECT 計上年月部署コード部署売上先コード売上先分類売上金額
FROM 売上テーブル
where 計上年月 in ("BKNO!!対象期間")
and 部署コード in ("BKNO!!対象部署コード")
and 分類コード in ("BKNO!!対象分類コード")
order by 計上年月部署コード売上先コード分類コード
    
```


抽出結果の展開先を新たに追加した「パレート図明細」シートにします。

開始セルを3行1列目とします。「列タイトルを出力する」にチェックを入れます。



マクロ「パレート図」の「保存対象シート」に「パレート図明細」を追加します。

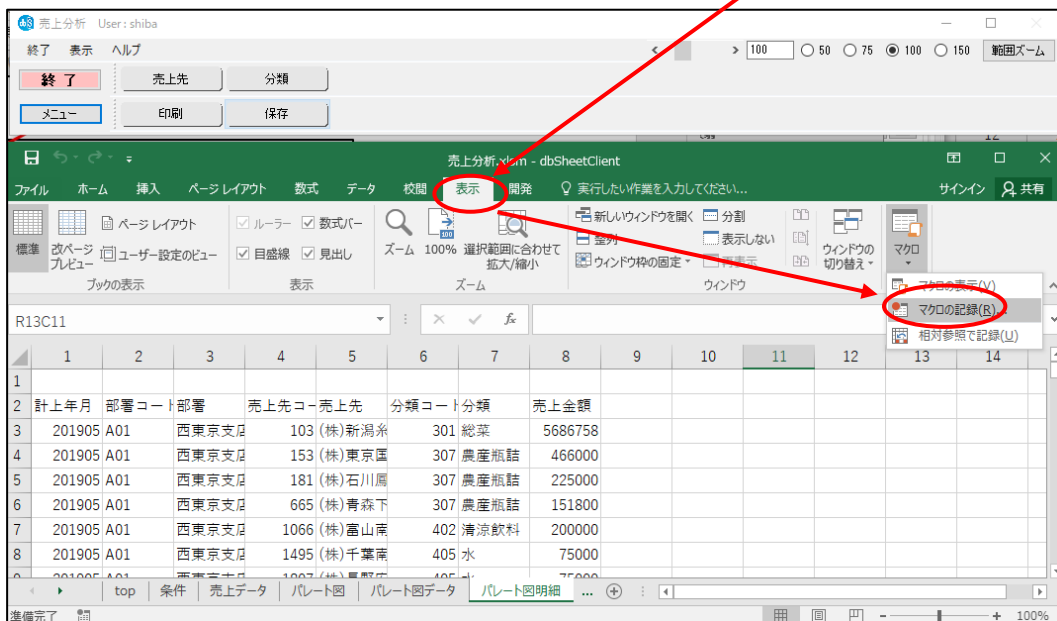
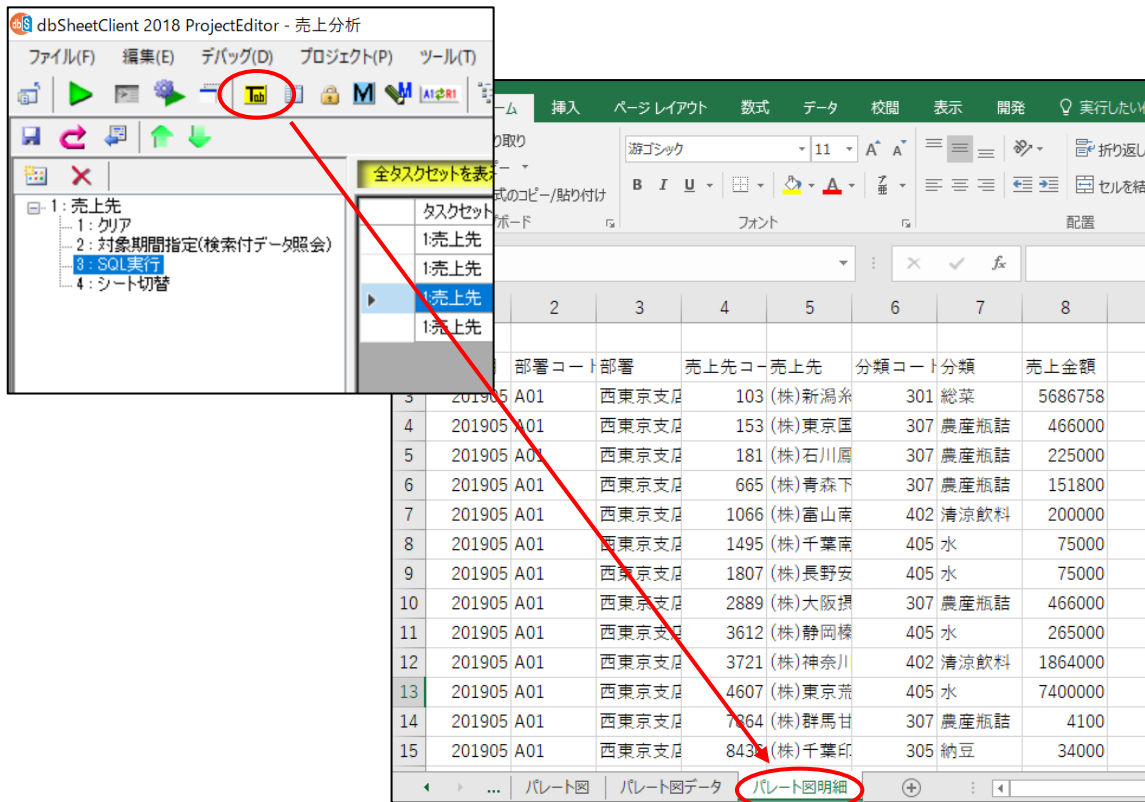
```

'印刷対象シート
    Sheets("条件").Cells(15, 3) = "パレート図"
'保存ファイル名
    Sheets("条件").Cells(17, 3) = "パレート図.xlsx"
'保存対象シート
    Sheets("条件").Cells(19, 3) = "パレート図,パレート図データ,パレート図明細"
    
```

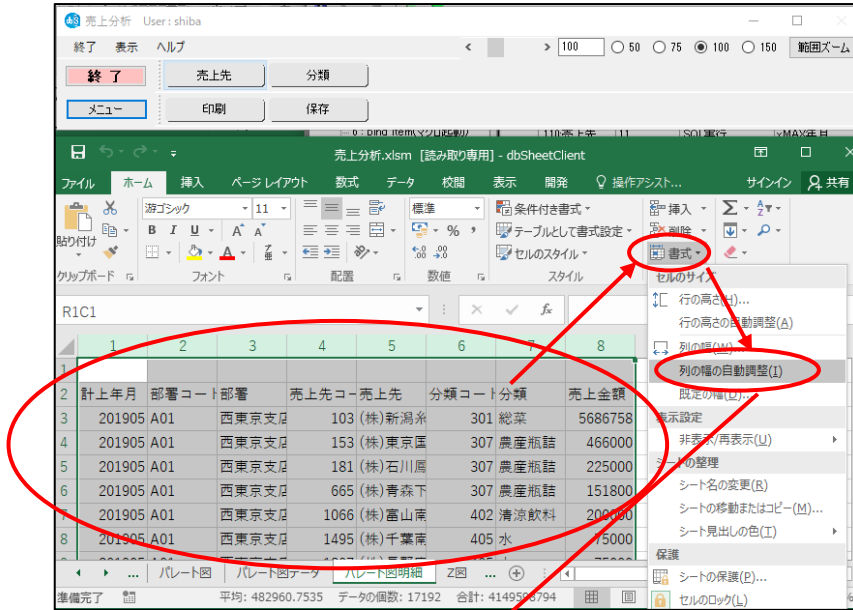
シート「パレート図明細」に展開されたデータを見やすくする記述をマクロ「パレート図」に追加します。
かなりマニアックなテクニックですが覚えておくと重宝します。

まずデバッグモードで実行して適当なパレート図を作成します。

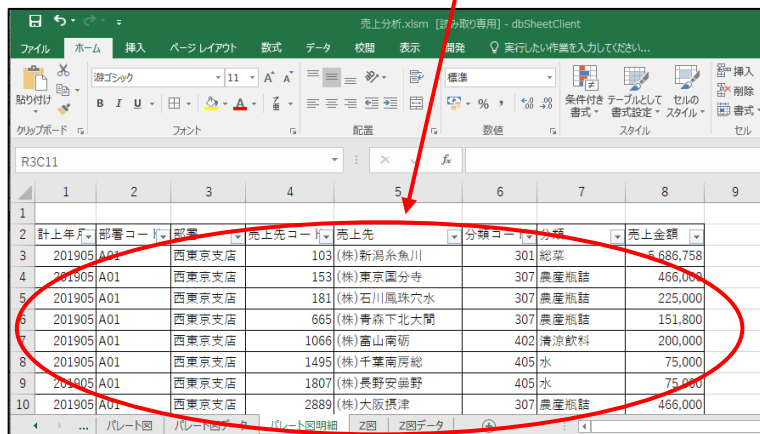
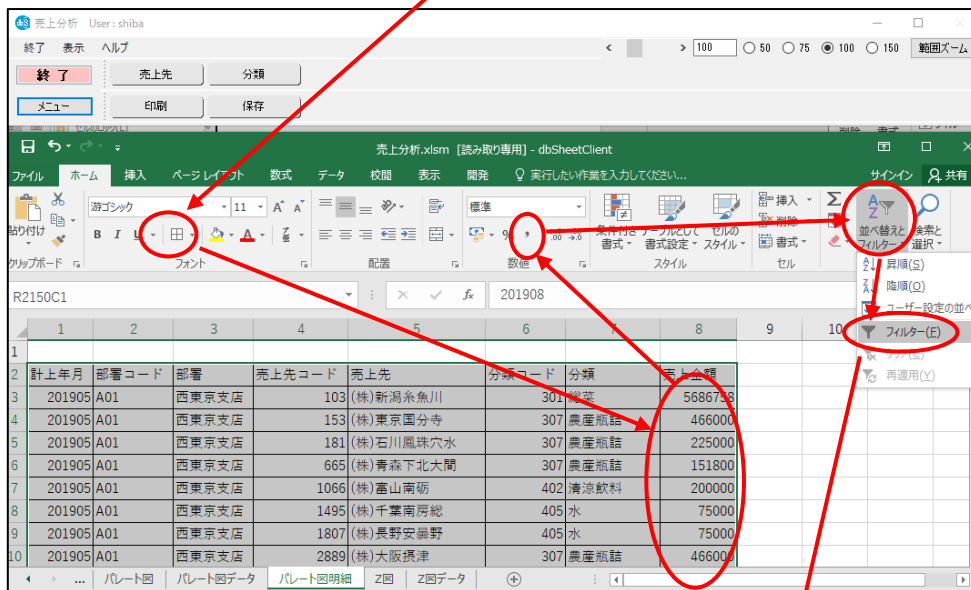
Excel シートタブ表示ボタンをクリックして「パレート図明細」シートを開き、マクロ記録を実行します。



明細データが展開された範囲を選択して「列の幅の自動調整」をかけます。

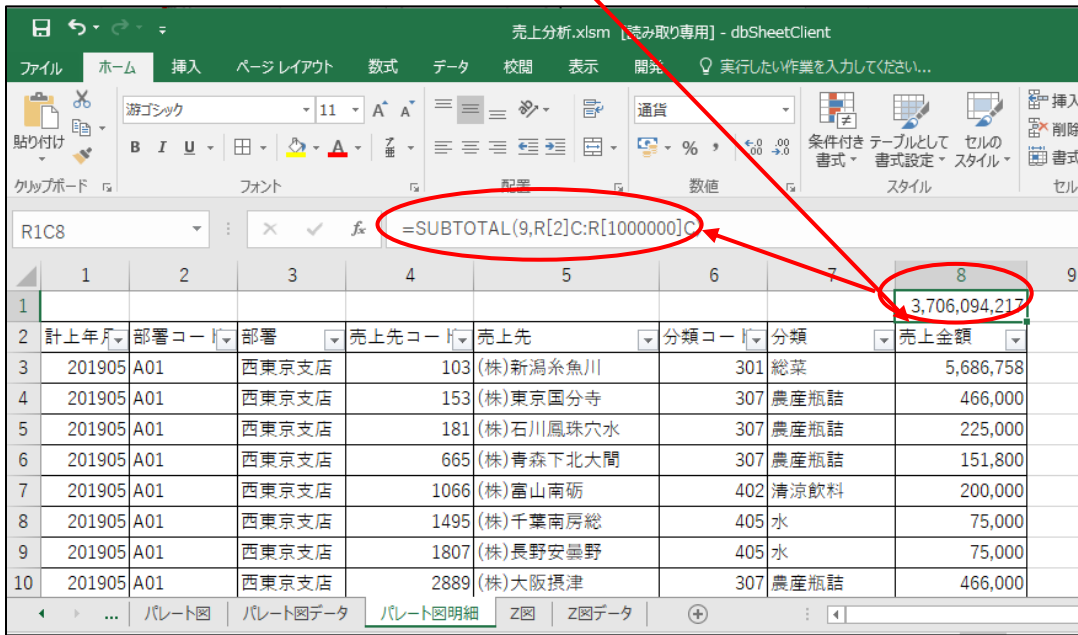


罫線を引いて、オートフィルターをかけて、「売上金額」列を桁区切りにします。

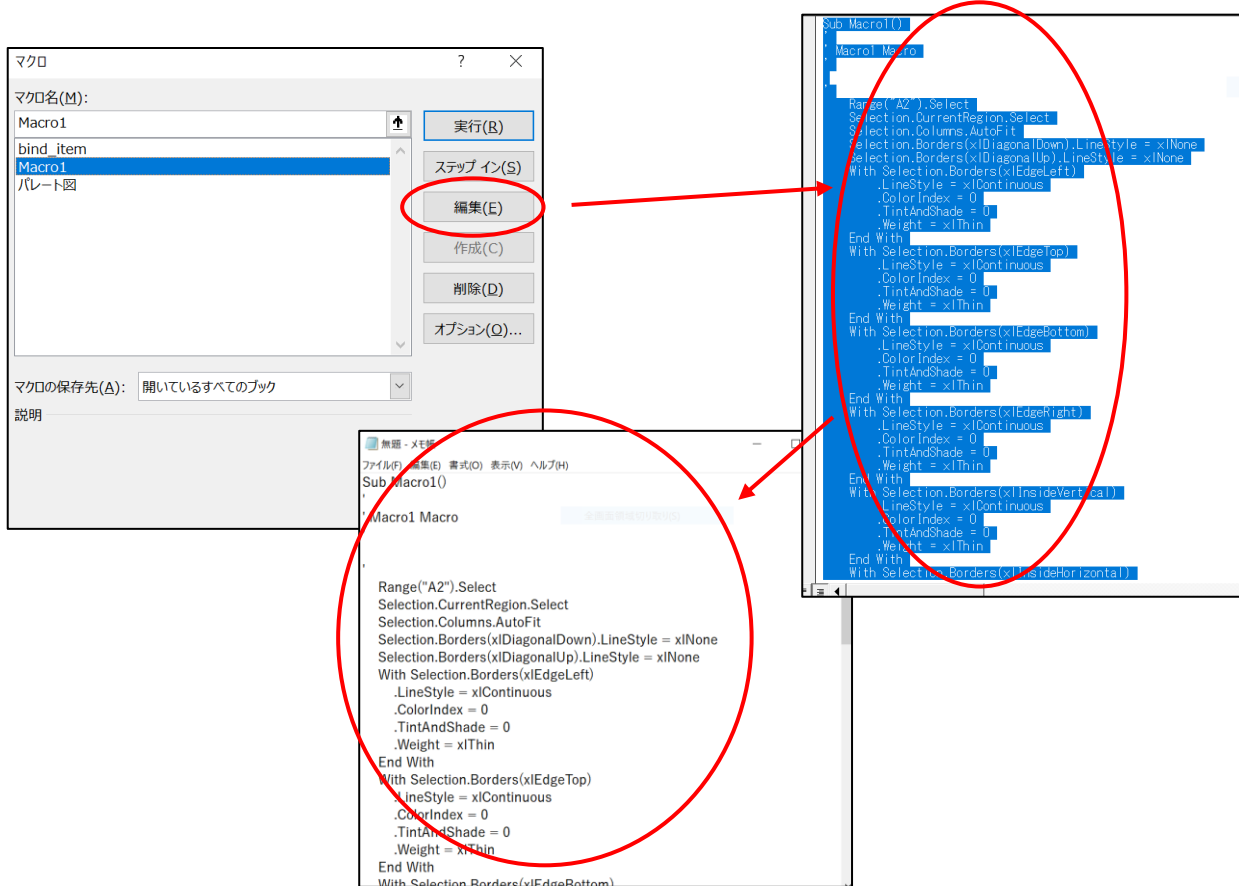


「売上金額」列の1行目のセルに **subtotal** 関数をセットします。

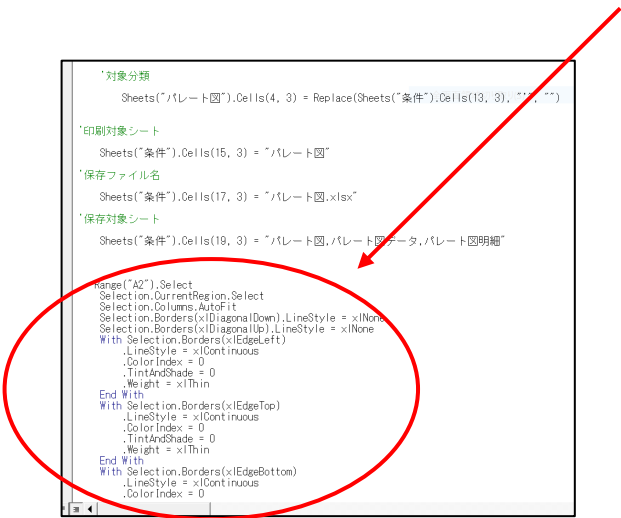
※オートフィルターで「部署」や「売上先」を絞り込んだ時、売上合計金額が自動計算されるようにします。



マクロ記録を終了して記録されたマクロをメモ帳にコピーします。



デバッグモードを終了し、Excel ボタンをクリックして Excel ファイルを呼び出します。
 マクロ「パレット図」の末尾にメモ帳に貼り付けたマクロをコピーします。



マクロ記録で生成されたプログラムを適当に変更を加えます。

'パレット図明細' シートを調整する

```

Sheets("パレット図明細").Select

'データが展開された範囲を調べる
Cells(100000, 2).Select
Selection.End(xlDown).Select
Selection.End(xlUp).Select

rr = ActiveCell.Row

Cells(2, 10000).Select
Selection.End(xlToRight).Select
Selection.End(xlToLeft).Select

cc = ActiveCell.Column

Range(Cells(2, 1), Cells(rr, cc)).Select

'セルの幅を自動調整
Selection.Columns.AutoFit

'罫線を引く
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft)
.LineStyle = xlContinuous
.ColorIndex = 0
.TintAndShade = 0
.Weight = xlThin
End With
    
```

'罫線を引く

```

Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft)
.LineStyle = xlContinuous
.ColorIndex = 0
.TintAndShade = 0
.Weight = xlThin
End With
With Selection.Borders(xlEdgeTop)
.LineStyle = xlContinuous
.ColorIndex = 0
.TintAndShade = 0
.Weight = xlThin
End With
With Selection.Borders(xlEdgeBottom)
.LineStyle = xlContinuous
.ColorIndex = 0
.TintAndShade = 0
.Weight = xlThin
End With
With Selection.Borders(xlEdgeRight)
.LineStyle = xlContinuous
.ColorIndex = 0
.TintAndShade = 0
.Weight = xlThin
End With
With Selection.Borders(xlInsideVertical)
.LineStyle = xlContinuous
.ColorIndex = 0
.TintAndShade = 0
.Weight = xlThin
End With
    
```

```

.Weight = xlThin
End With
With Selection.Borders(xlInsideHorizontal)
.LineStyle = xlContinuous
.ColorIndex = 0
.TintAndShade = 0
.Weight = xlThin
End With

'オートフィルターをかける
Selection.AutoFilter

'売上金額列を3桁区切にする
Columns("H:H").Select
Selection.Style = "Comma [0]"

'売上金額の合計を計算する関数をセット
Cells(1, 8) = "=SUBTOTAL(9,R[2]C:R[100000]C)"

'売上金額合計セルの列幅を調整
Cells(1, 8).Select
Selection.Columns.AutoFit

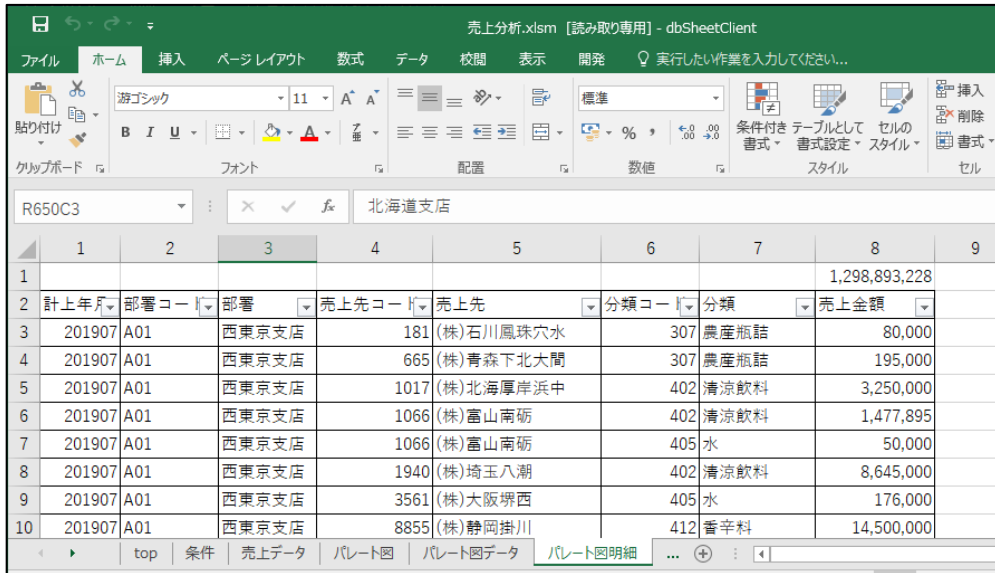
End Sub
    
```

デバッグモードで実行

パレート図を作成して「保存」ボタンで結果を保存します。

「パレート図明細」シートと一緒に保存されているか確認します。

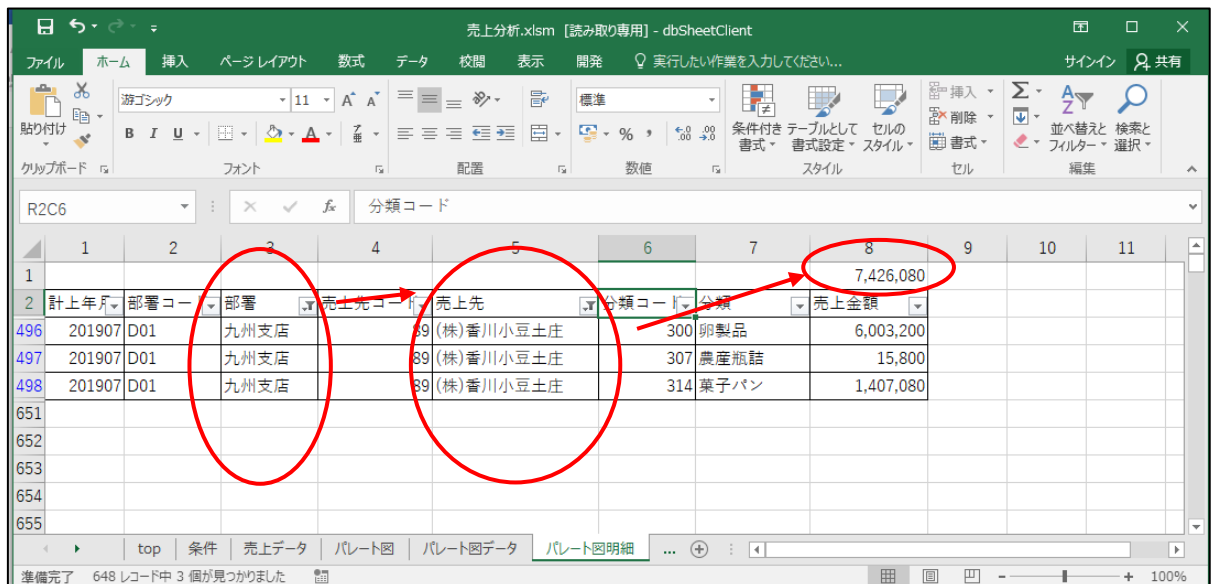
また、自動列幅調整や罫線、売上金額が桁区切りになっているか確認します。



The screenshot shows an Excel spreadsheet with a table of sales data. The table has columns for fiscal year, department code, department, sales company code, sales company, category code, category, and sales amount. The total sales amount for the department is 1,298,893,228. The table is filtered to show data for the department '北海道支店'.

1	2	3	4	5	6	7	8	9
1							1,298,893,228	
2	計上年月	部署コード	部署	売上先コード	売上先	分類コード	分類	売上金額
3	201907	A01	西東京支店	181	(株)石川風珠穴水	307	農産瓶詰	80,000
4	201907	A01	西東京支店	665	(株)青森下北大間	307	農産瓶詰	195,000
5	201907	A01	西東京支店	1017	(株)北海道厚岸浜中	402	清涼飲料	3,250,000
6	201907	A01	西東京支店	1066	(株)富山南砺	402	清涼飲料	1,477,895
7	201907	A01	西東京支店	1066	(株)富山南砺	405	水	50,000
8	201907	A01	西東京支店	1940	(株)埼玉八潮	402	清涼飲料	8,645,000
9	201907	A01	西東京支店	3561	(株)大阪堺西	405	水	176,000
10	201907	A01	西東京支店	8855	(株)静岡掛川	412	香辛料	14,500,000

オートフィルターで明細を絞り込んで、売上金額合計が自動計算されることを確認します。



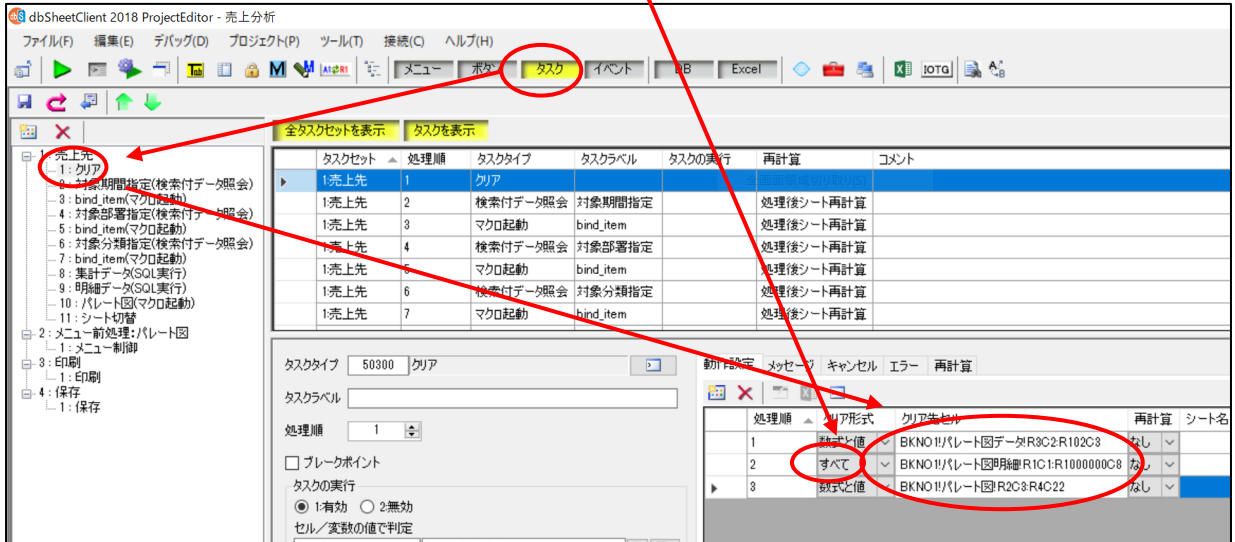
The screenshot shows the same Excel spreadsheet with the Pareto chart summary table. The table is filtered to show data for the department '九州支店'. The total sales amount for the department is 7,426,080. Red circles highlight the department code 'D01', the sales company code '89', and the total sales amount '7,426,080'.

1	2	3	4	5	6	7	8	9	10	11
1							7,426,080			
2	計上年月	部署コード	部署	売上先コード	売上先	分類コード	分類	売上金額		
496	201907	D01	九州支店	89	(株)香川小豆土庄	300	卵製品	6,003,200		
497	201907	D01	九州支店	89	(株)香川小豆土庄	307	農産瓶詰	15,800		
498	201907	D01	九州支店	89	(株)香川小豆土庄	314	菓子パン	1,407,080		
651										
652										
653										
654										
655										

タスクセット「売上先」の「クリア」タスクを変更します。

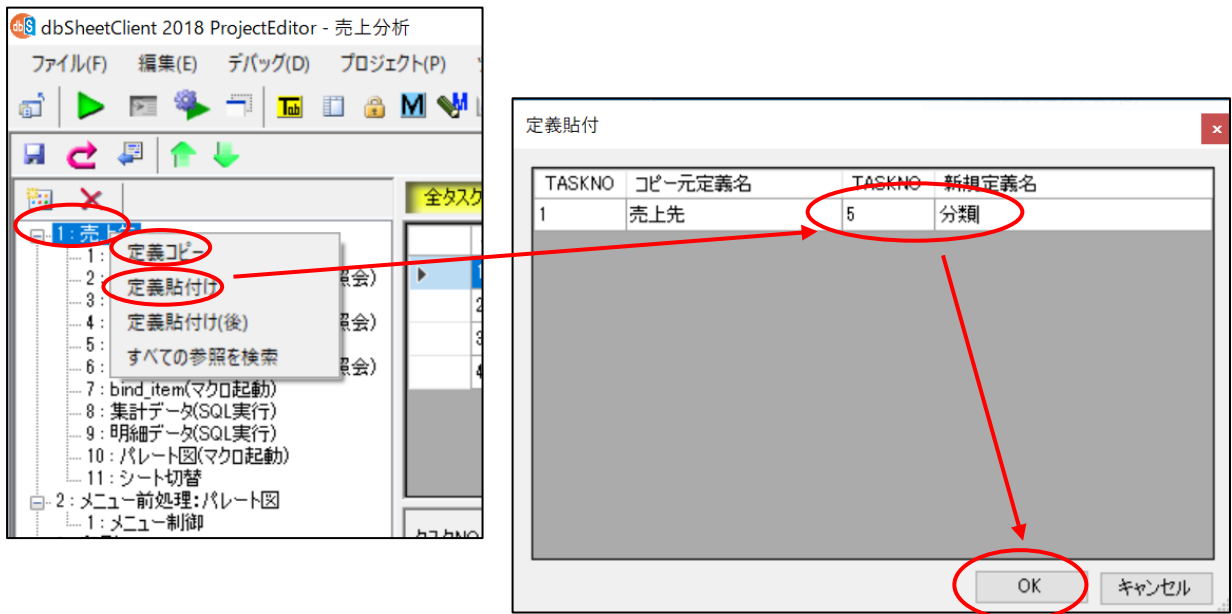
シート「パレート図」の「期間」「部署」「分類」、シート「パレート図明細」のデータ展開範囲を追加します。

(注)シート「パレート図明細」では「クリア形式」を「すべて」にしないと罫線が残ります。

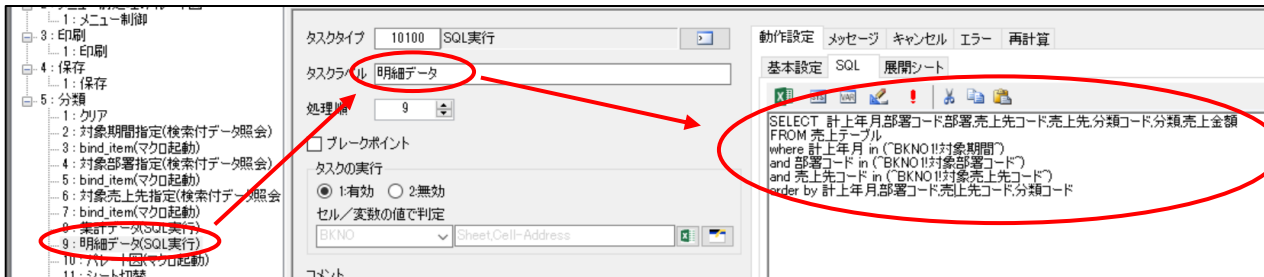
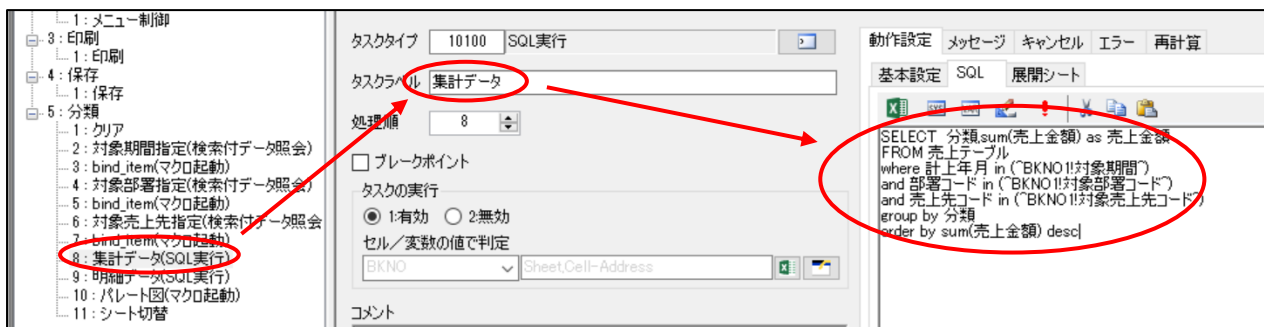
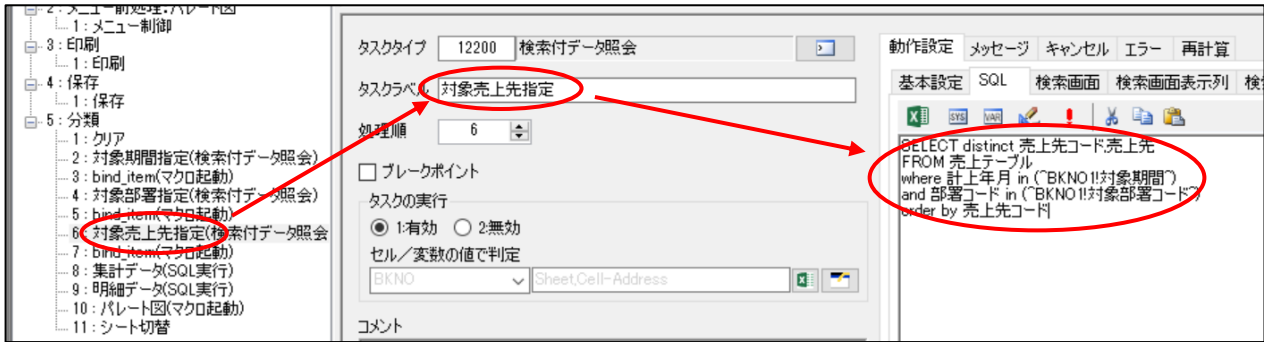


1.3.11 「売上先」分析タスクをコピーして「分類」分析タスクを作成する

タスクメニューよりタスクセット「売上先」を右クリックして「定義コピー」を選択して「定義貼付け」をクリック
 タスクセット名を「分類」にします。(TASKNO は「5」)



コピーしたタスクセット「分類」の「SQL 実行」タスクを「分類」分析用に変更します。(3ヶ所)



マクロ「bind_item」も変更します。

```

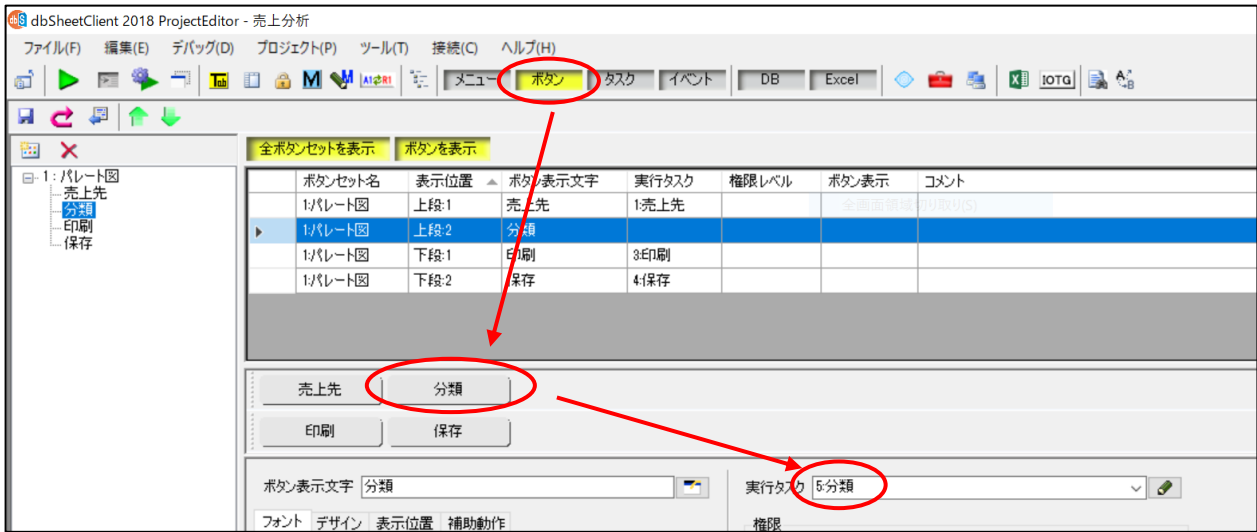
Sub bind_item()
↓
'検索付きデータ照会で選択された項目を一まとめに
sheet_name = ActiveSheet.Name

Call bind(7) '対象期間
Call bind(9) '対象部署コード
Call bind(10) '対象部署
Call bind(12) '対象分類コード
Call bind(13) '対象分類
Call bind(21) '対象売上先コード
Call bind(22) '対象売上先

Sheets(sheet_name).Select

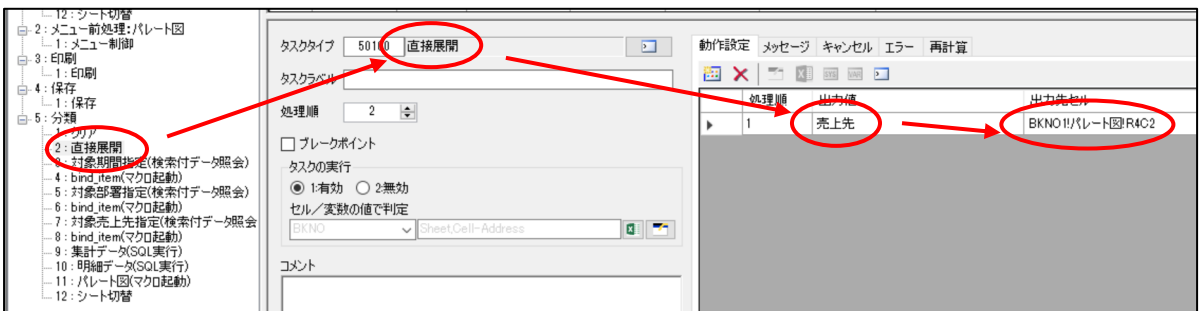
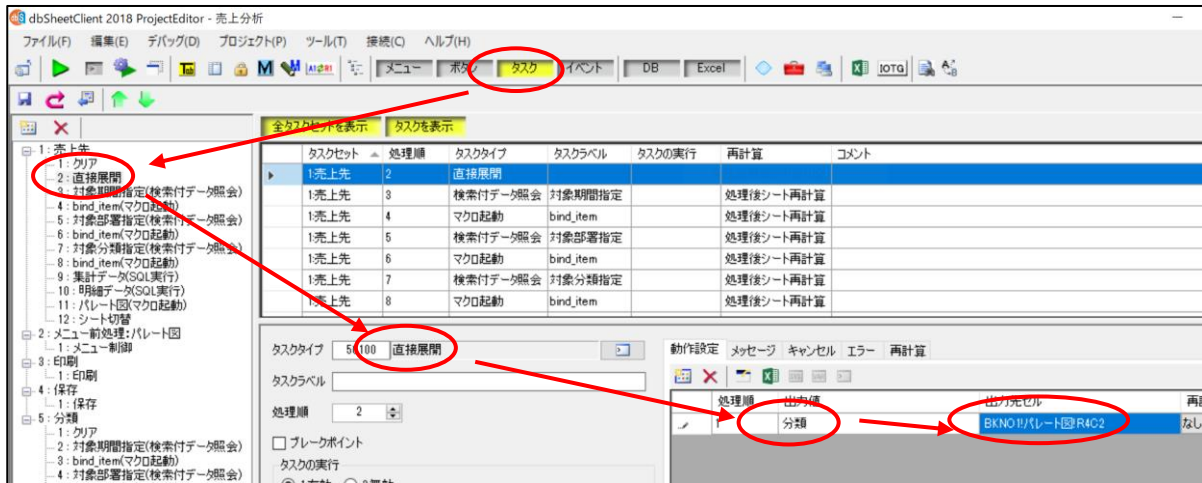
End Sub
    
```


「分類」ボタンにタスクセット「分類」を割り付けます。



「売上先」ボタンと「分類」ボタンでパレット図への抽出条件展開を切替えます。

タスクセット「売上先」、「分類」に「直接展開」タスクを追加します。



マクロ「パレート図」にコピー切替記述を追加します。

```

Sub パレート図()
'抽出条件をコピー
'対象期間
Sheets("パレート図").Cells(2, 3) = Replace(Sheets("条件").Cells(7, 3), "", "")
'対象部署
Sheets("パレート図").Cells(3, 3) = Replace(Sheets("条件").Cells(10, 3), "", "")
'「売上先」ボタンと「分類」ボタンでコピーする項目を切り替える
syubetu = Sheets("パレート図").Cells(4, 2)
Select Case syubetu
Case "売上先"
'対象分類をコピー
Sheets("パレート図").Cells(4, 3) = Replace(Sheets("条件").Cells(13, 3), "", "")
Case "分類"
'対象売上先をコピー
Sheets("パレート図").Cells(4, 3) = Replace(Sheets("条件").Cells(22, 3), "", "")
Case Else
End Select
End Sub
    
```

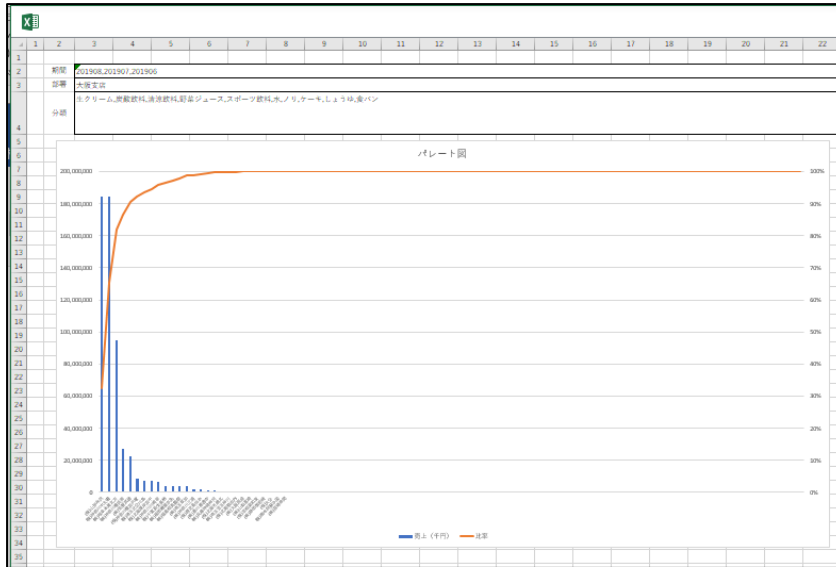
デバッグモードで実行

「部署」選択の次に「売上先」選択となっているか確認します。

The screenshot shows the application's search dialog and a Pareto chart. The search dialog, titled '検索付データ照会', has a table with columns '選択' (Selected), '売上先コード' (Sales Company Code), and '売上先' (Sales Company Name). The '売上先' column is circled in red. The Pareto chart, titled 'パレート図', shows sales data with a legend at the bottom right where '売上 (千円)' (Sales in thousands of yen) is represented by a blue bar and '比率' (Ratio) by an orange line. The '売上先' label in the legend is also circled in red. Red arrows point from the text above to these specific elements.

1.3.12 最終調整

本章はかなり長くなってしまいましたが、もう少しお付き合いください。
 抽出された分類数が少ない場合にグラフの左余白を詰めて間が抜けないようにします。
 ※初期値100件を必要に応じて減らす機能をマクロに組み込む
 また、売上金額を円単位から千円単位に変更します。



シート「パレート図データ」に展開された売上先件数を調べてグラフのデータ範囲を可変にする記述をマクロ「パレート図」に組み込みます。

「5.3.10 明細を展開する」で行った手法で簡単に生成できます。

```

'パレート図の対象数を調整する
Sheets("パレート図明細").Select
'データが展開された最下行を調べる
Cells(100000, 3).Select
Selection.End(xlDown).Select
Selection.End(xlUp).Select
rr = ActiveCell.Row
'最下行が102行（初期値）より小さければ変更する
If rr < 102 Then
Else
rr = 102
End If
'グラフ描画のデータ範囲を調整する
Sheets("パレート図").Select
ActiveSheet.ChartObjects("グラフ 1").Activate
ActiveChart.FullSeriesCollection(1).Select
Selection.Formula =
=SERIES(パレート図データ!R2C3,パレート図データ!R3C2:R & rr & "C2,パレート図データ!R3C3:R & rr & "C3,1)"
Selection.Formula =
=SERIES(パレート図データ!R2C3,パレート図データ!R3C2:R & rr & "C2,パレート図データ!R3C3:R & rr & "C3,1)"
ActiveChart.FullSeriesCollection(2).Select
Selection.Formula =
=SERIES(パレート図データ!R2C5,パレート図データ!R3C2:R & rr & "C2,パレート図データ!R3C5:R & rr & "C5,2)"
End Sub
    
```

シート「パレート図データ」のタイトル行を変更します。

「売上」 → 「売上(千円)」

No	項目	売上 (千円)	累計	比率
1		7,525,000		
1		7,525,000		
2	テスト001	100,000	100,000	1%
3	テスト002	99,500	199,500	3%
4	テスト003	99,000	298,500	4%
5	テスト004	98,500	397,000	5%
6	テスト005	98,000	495,000	7%
7	テスト006	97,500	592,500	8%
8	テスト007	97,000	689,500	9%
9	テスト008	96,500	786,000	10%
10	テスト009	96,000	882,000	12%
11	テスト010	95,500	977,500	13%
12	テスト011	95,000	1,072,500	14%
13	テスト012	94,500	1,167,000	16%
14	テスト013	94,000	1,261,000	17%
15	テスト014	93,500	1,354,500	18%

集計データを展開する「SQL実行」タスクのSQL文を変更します。

dbSheetClient 2018 ProjectEditor - 売上分析

メニュー ボタン タスク イベント DB Excel IoTG

タスクセット	処理順	タスクタイプ	タスクラベル	タスクの実行	再計算	コメント
1: 売上先						
1: 売上先	3	検索付データ照会	対象期間指定		処理後シート再計算	
1: 売上先	4	マクロ起動	bind_item		処理後シート再計算	
1: 売上先	5	検索付データ照会	対象部署指定		処理後シート再計算	
1: 売上先	6	マクロ起動	bind_item		処理後シート再計算	
1: 売上先	7	検索付データ照会	対象分類指定		処理後シート再計算	
1: 売上先	8	マクロ起動	bind_item		処理後シート再計算	
1: 売上先	9	SQL実行	集計データ		処理前シート再計算	SELECT 計上年月部署コード部署売
1: 売上先	10	SQL実行	明細データ		処理前シート再計算	SELECT 計上年月部署コード部署売

タスクタイプ: 10100 SQL実行

タスクラベル: 集計データ

処理順: 9

基本設定 SQL 展開シート

```

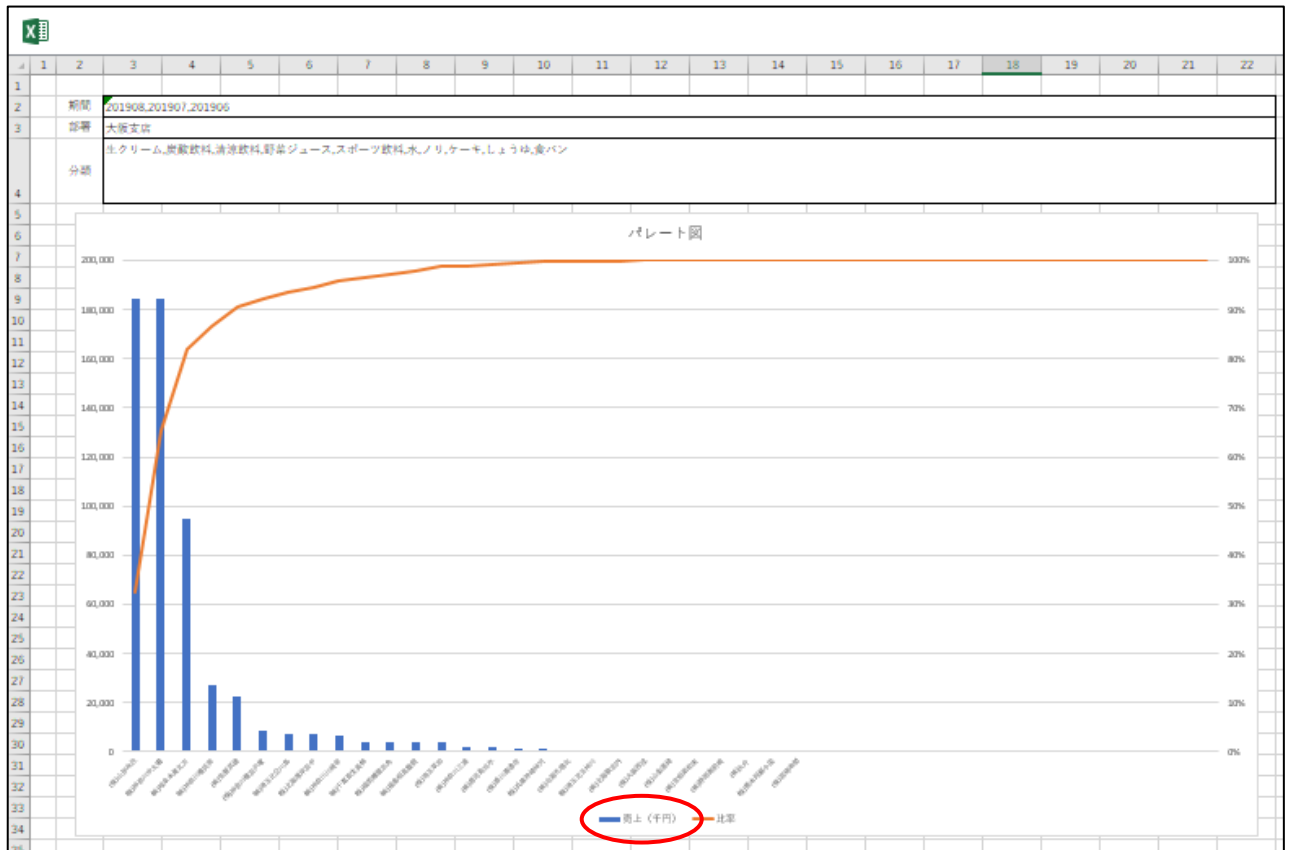
SELECT 売上先, round(sum(売上金額/1000),0) as 売上金額
FROM 売上先
where 計上年月 in ('BKNO1|対象期間')
and 部署コード in ('BKNO1|対象部署コード')
and 分類コード in ('BKNO1|対象分類コード')
group by 売上先
order by round(sum(売上金額/1000),0) desc
    
```

明細データを展開する「SQL実行」タスクのSQL文を変更します。

The screenshot shows the 'dbSheetClient 2018 ProjectEditor' interface. The 'タスク' (Task) menu is highlighted. A table lists tasks, with task 10 'SQL実行' selected. The configuration panel for this task shows 'タスクタイプ' as 'SQL実行' and 'タスクラベル' as '明細データ'. The 'SQL' tab is active, displaying the following SQL query:

```
SELECT 計上年月部署コード部署売上先コード売上
round(売上金額/1000,0) as 売上金額(千円)
FROM 売上先
where 計上年月 in ('BKNO' ||対象期間)
and 部署コード in ('BKNO' ||対象部署コード)
and 分類コード in ('BKNO' ||対象分類コード)
order by 計上年月部署コード売上先コード分類コード
```

デバッグモードで実行



5.4 Z 図

Z 図とは次の3つの実績値の増減を折れ線グラフに重ねて描いたものです。

- ① 単月に発生した実績
- ② ①の累計
- ③ 当月を含む過去12ヶ月の合計値

上記の3つの折れ線グラフは必ずZ形になることからZ図と呼ばれています。

実績値は売上でも販売利益でも在庫金額でもよいです。

折れ線グラフで実績と予算の推移を表したり、棒グラフで今年度実績と前年度実績を対比します。

経営層が一番気になるのは、「現在、上昇基調にあるのか下降線を辿っているのか」ではないでしょうか。

Z図の解説はインターネット情報に譲り、これから dbSheetClient を使った作成方法を紹介します。

1.4.1 グラフの雛形を準備

Excel ファイルに「Z図」「Z図データ」シートを追加します。

ダミーデータを展開し関数式をセットします。(インターネットにたくさん事例があります。)

=LEFT(R[11]C[-4],4) & "年" & RIGHT(R[11]C[-4],2) & "月"

=R[11]C[-4]

=RC[-1]

1	2	3	4	5	6	7	8	9	10
No	計上年月	売上金額		No	計上年月	単月売上	累計売上	12ヶ月移動和	
1	200101	110		1	2001年12月	220	220	1,980	
2	200102	120		2	2002年01月	230	450	2,100	
3	200103	130		3	2002年02月	240	690	2,220	
4	200104	140		4	2002年03月	250	940	2,340	
5	200105	150		5	2002年04月	260	1,200	2,460	
6	200106	160		6	2002年05月	270	1,470	2,580	
7	200107	170		7	2002年06月	280	1,750	2,700	
8	200108	180		8	2002年07月	290	2,040	2,820	
9	200109	190		9	2002年08月	300	2,340	2,940	
10	200110	200		10	2002年09月	310	2,650	3,060	
11	200111	210		11	2002年10月	320	2,970	3,180	
12	200112	220		12	2002年11月	330	3,300	3,300	
13	200201	230							
14	200202	240							
15	200203	250							
16	200204	260							
17	200205	270							
18	200206	280							
19	200207	290							
20	200208	300							
21	200209	310							
22	200210	320							
23	200211	330							
24									
25									
26									
27									
28									

=SUM(RC[-6]:R[11]C[-6])

=R[-1]C+RC[-1]

=R[11]C[-4]

=RC[-1]

=LEFT(R[11]C[-4],4) & "年" & RIGHT(R[11]C[-4],2) & "月"

=SUM(RC[-6]:R[11]C[-6])

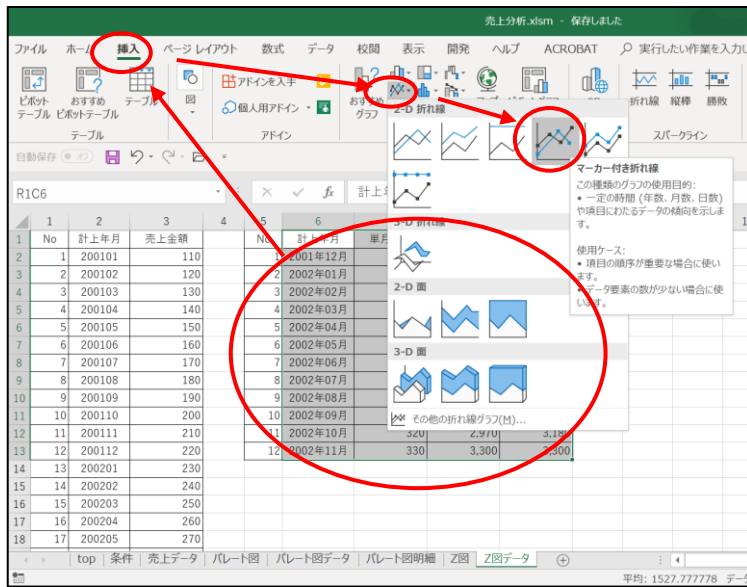
=R[-1]C+RC[-1]

=R[11]C[-4]

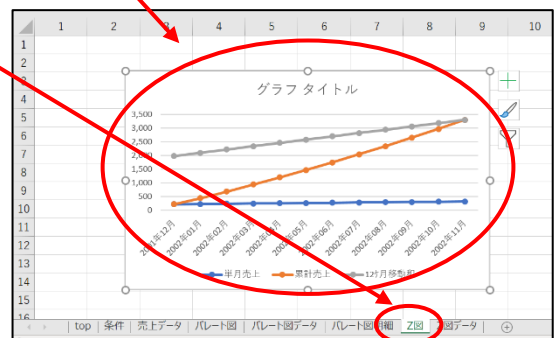
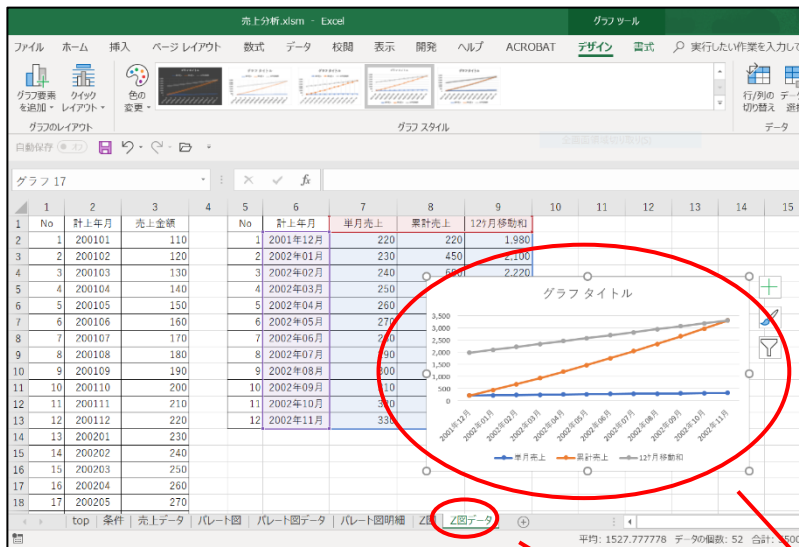
=RC[-1]

=LEFT(R[11]C[-4],4) & "年" & RIGHT(R[11]C[-4],2) & "月"

グラフを生成するデータを範囲指定して折れ線グラフを指定します。

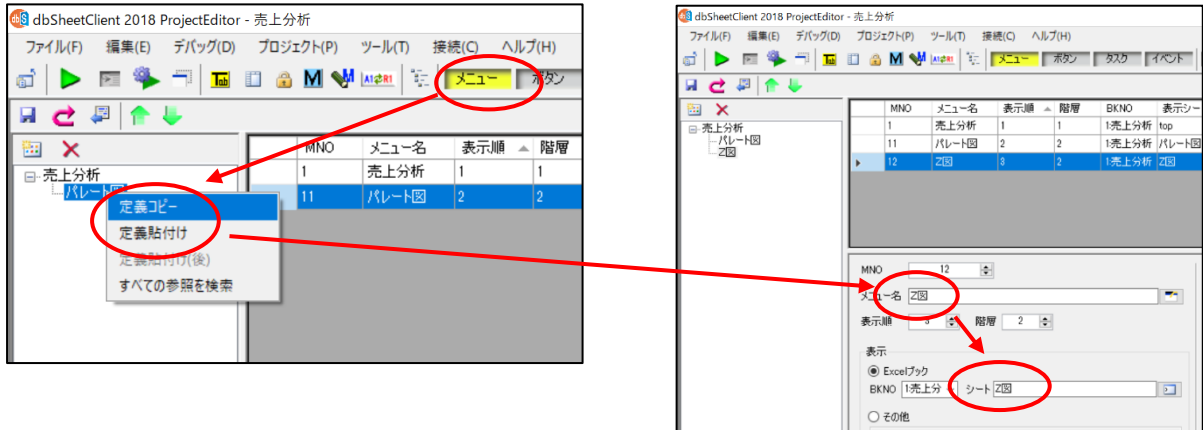


シート「Z図データ」に生成されたグラフを切り取って、シート「Z図」に貼り付けます。



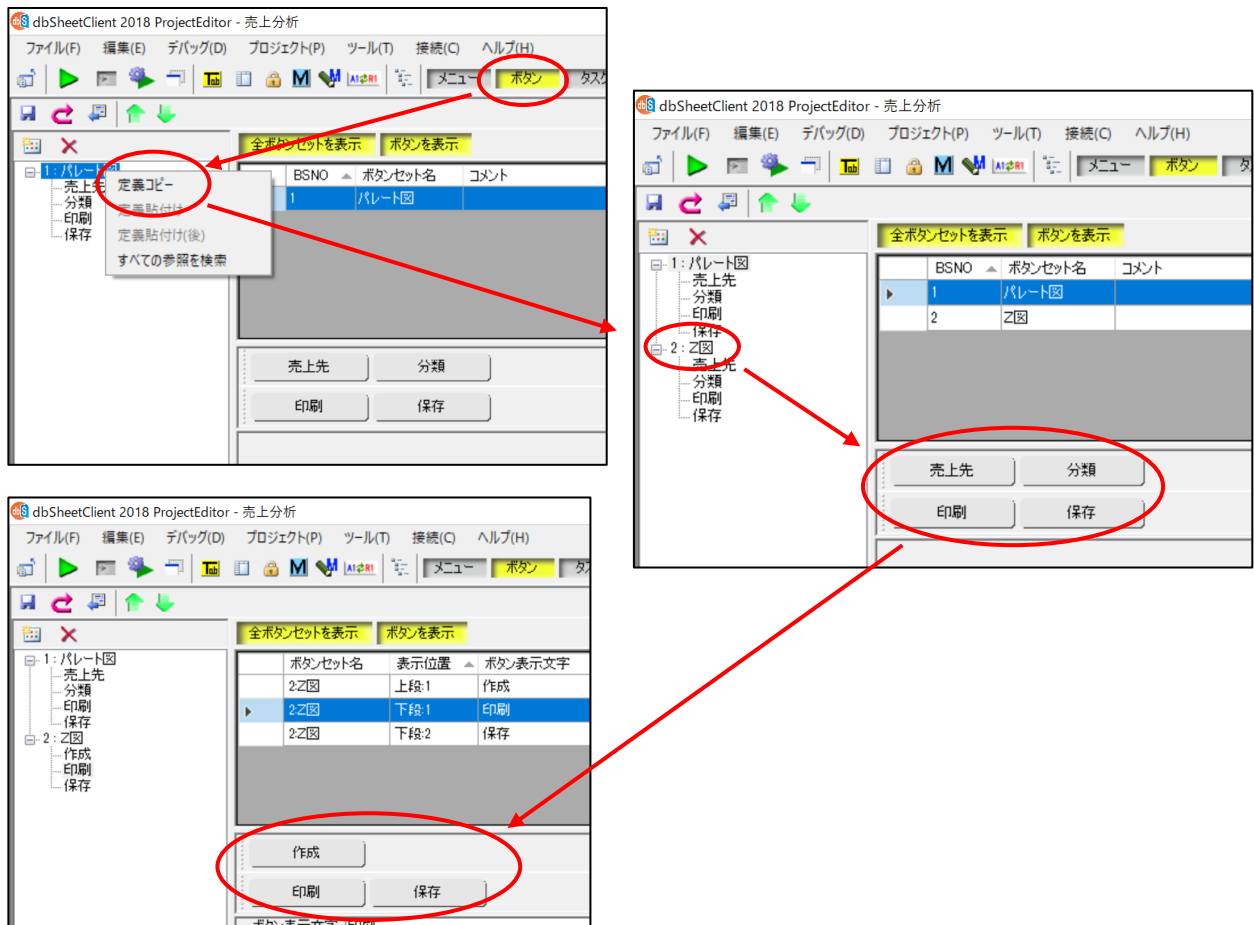
1.4.2 メニュー、ボタンの追加

メニュー「パレート図」をコピーします。メニュー名と切替シートも「Z図」に変更します。

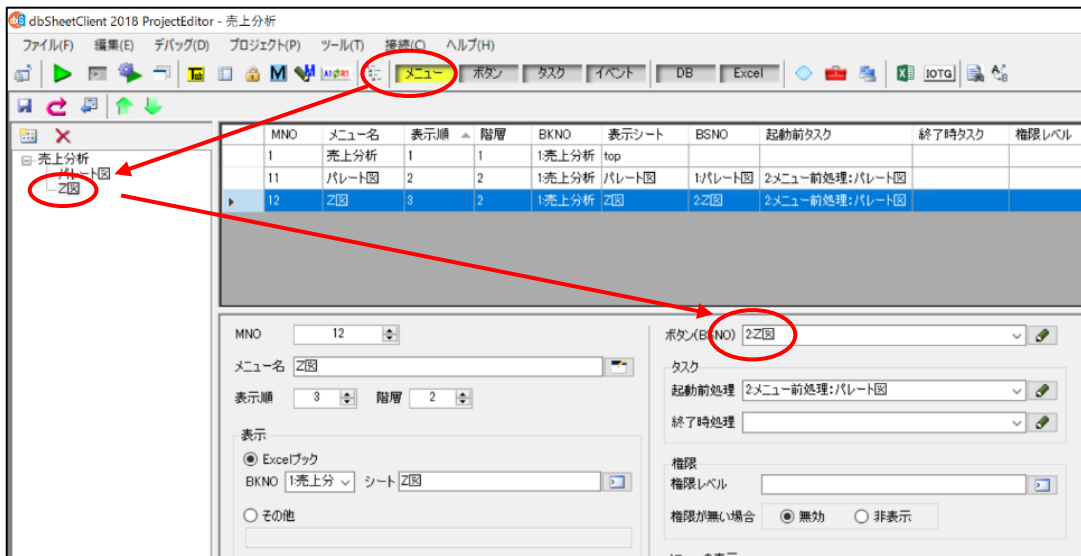


ボタンセット「パレート図」をコピーします。

ボタンセット名を「Z図」に変更、「売上先」を「作成」に変更、「分類」ボタンは削除します。

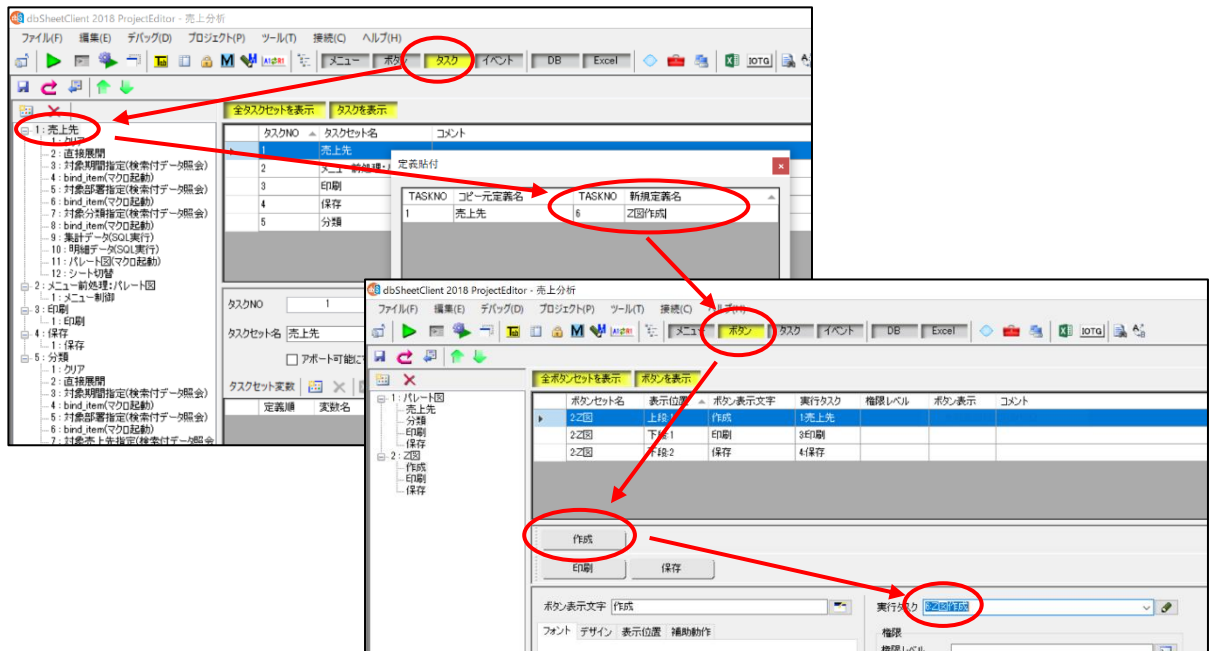


メニュー「Z図」に作成したボタンセット「Z図」を割り付けます。



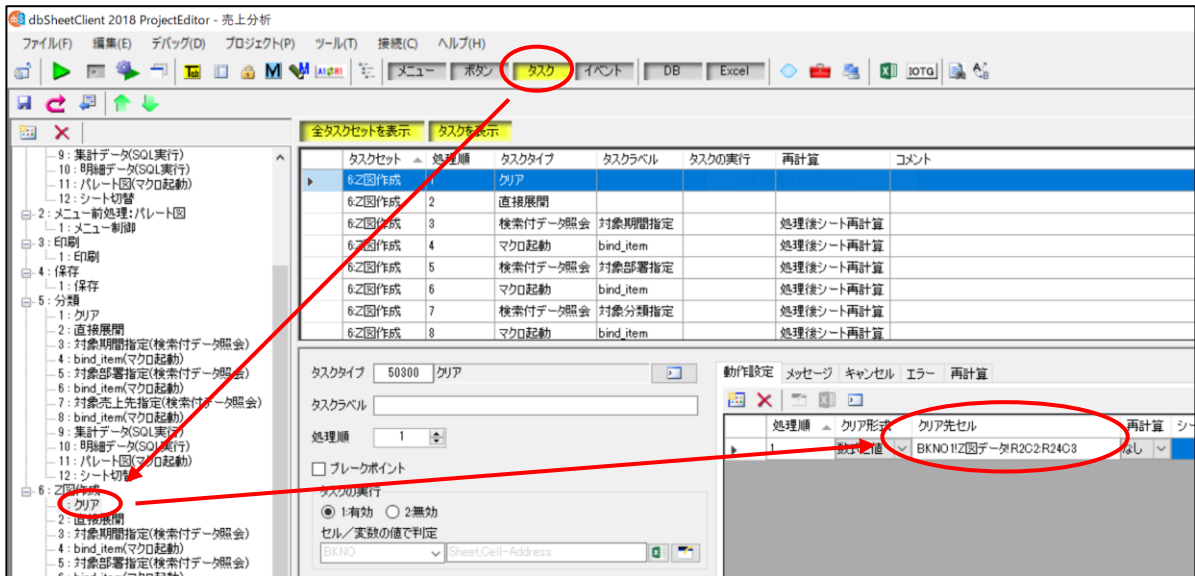
1.4.3 タスクの登録

タスクセット「売上先」をコピーしてタスクセット「Z図作成」を作成します。
 ボタン「作成」にタスクセット「Z図作成」を割り付けます。

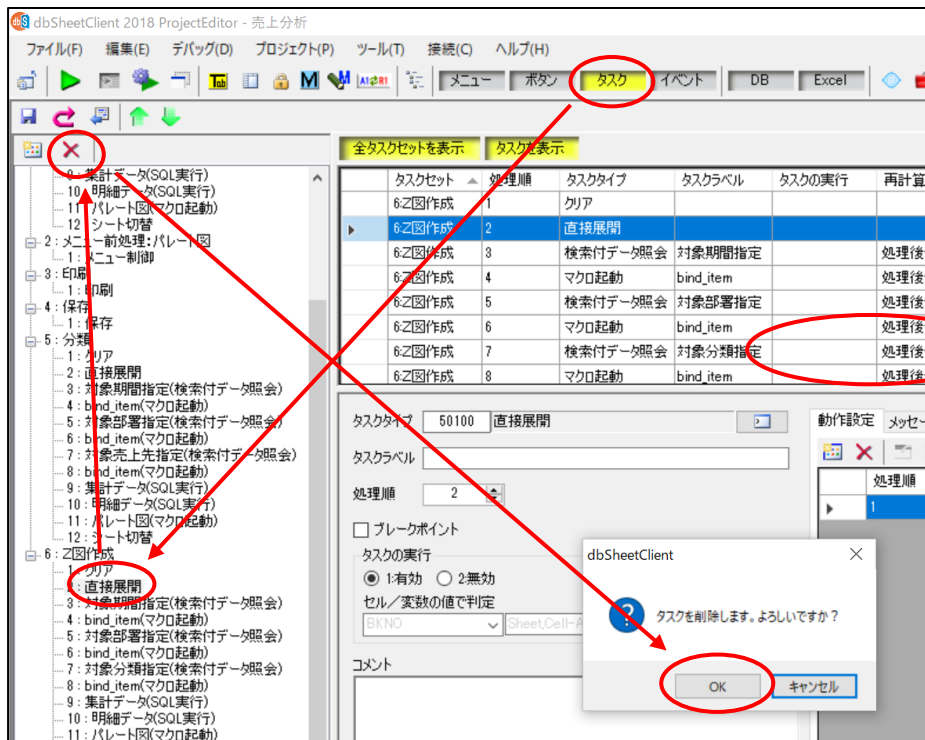


タスクセット「Z図作成」の「クリア」タスクの設定を変更します。

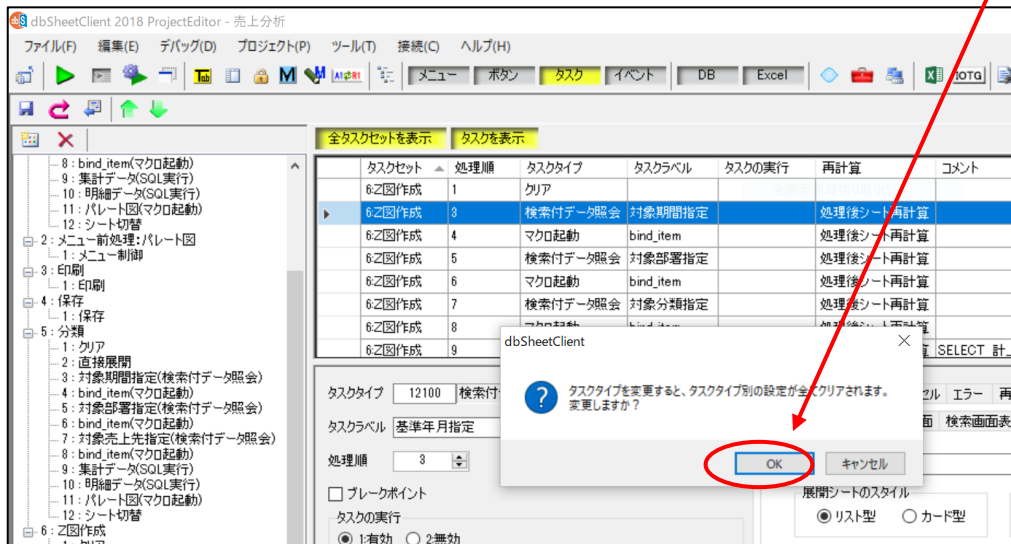
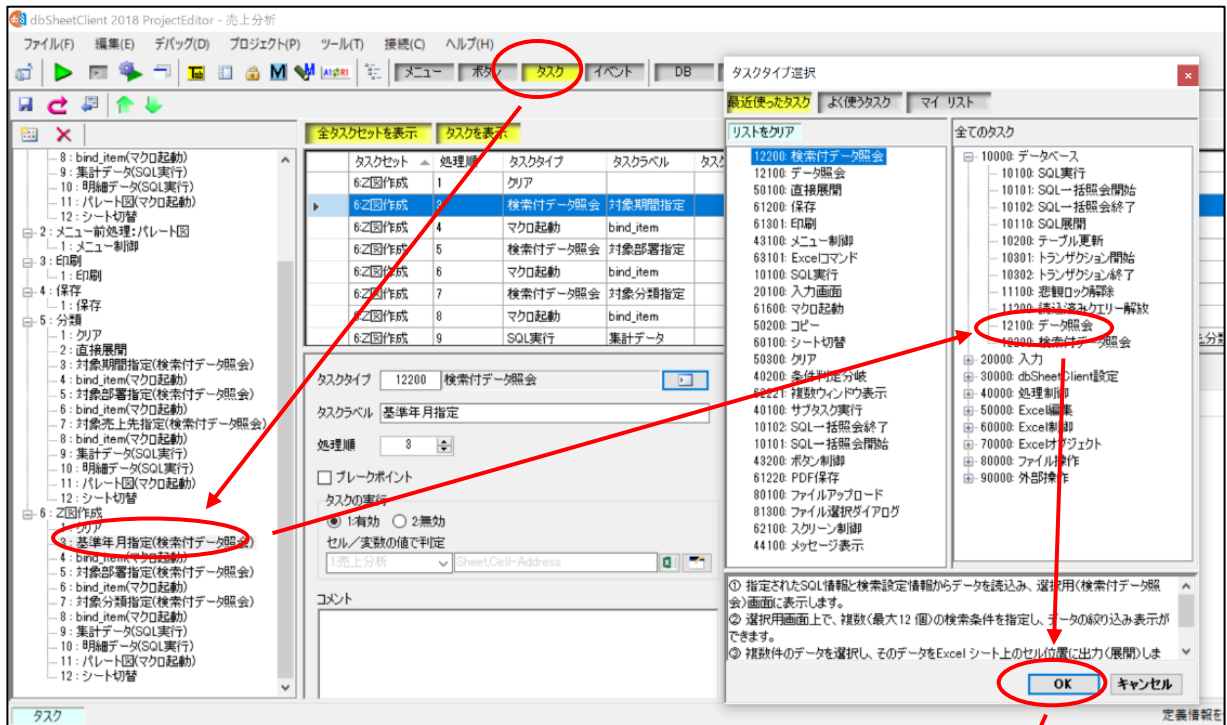
→ クリア範囲をシート「Z図データ」の「2行2列目～23行3列目」に変更



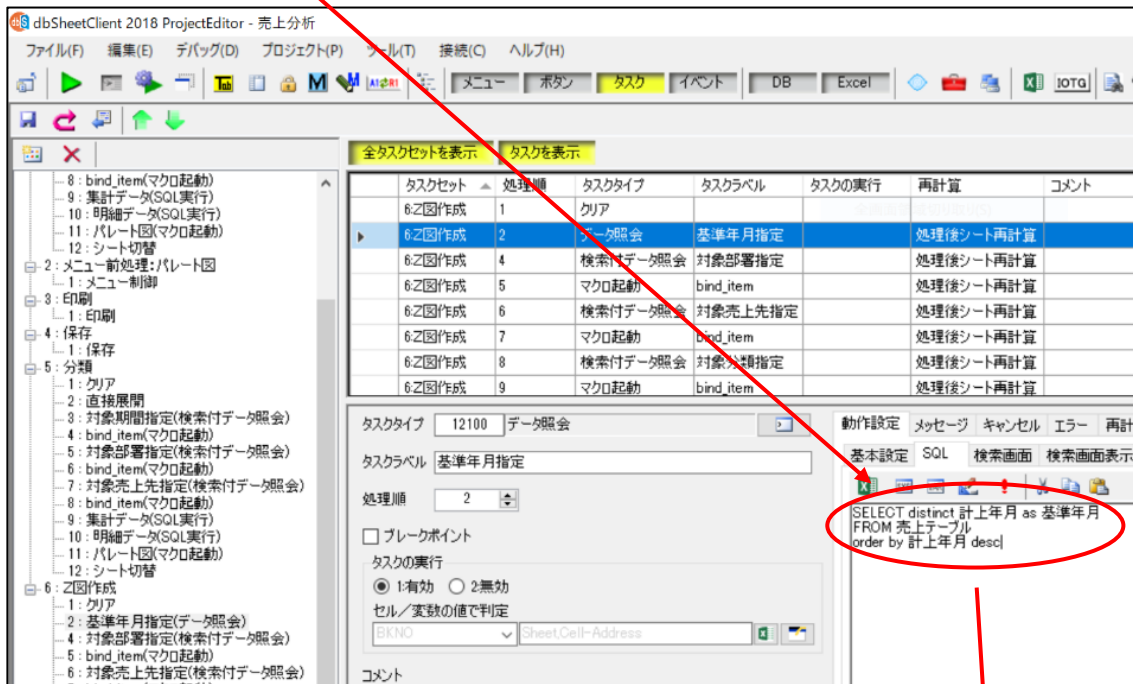
「直接展開」タスクを削除します。



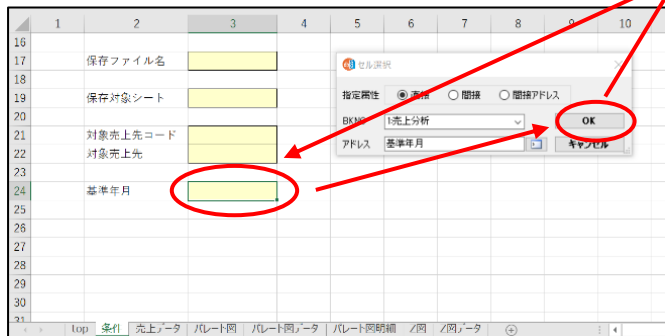
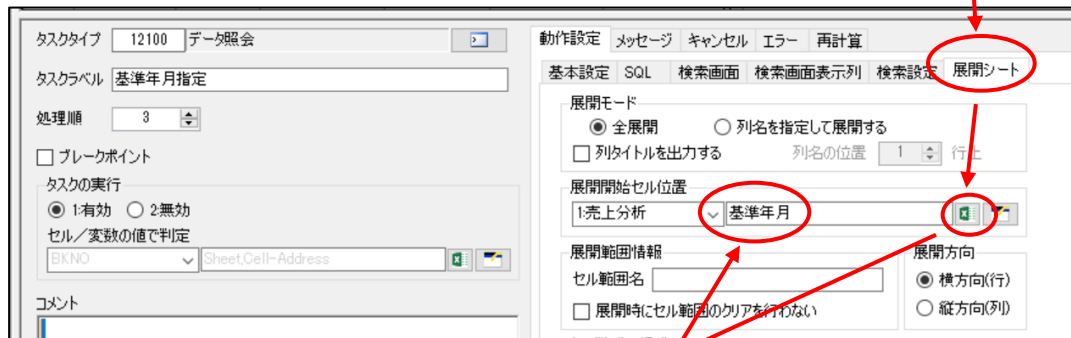
対象期間を指定するタスクのタスクタイプを「データ照会」に変更します。(変更前は「検索付データ照会」)



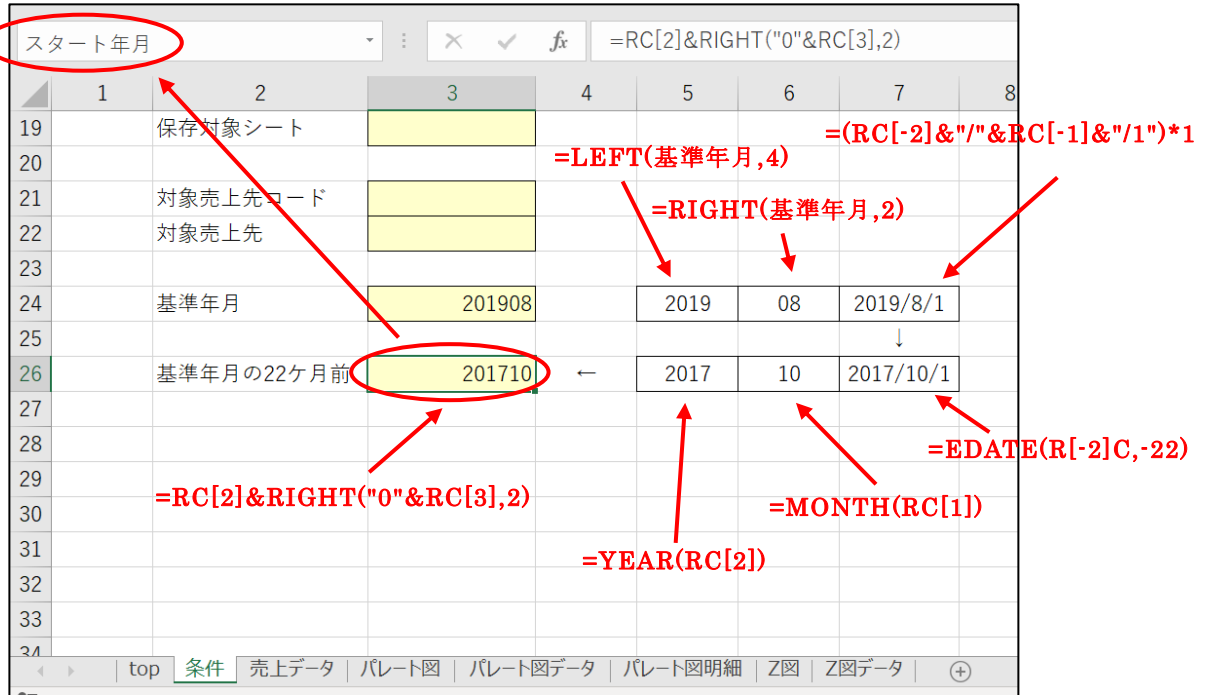
基準年月を指定するSQL文をセットします。



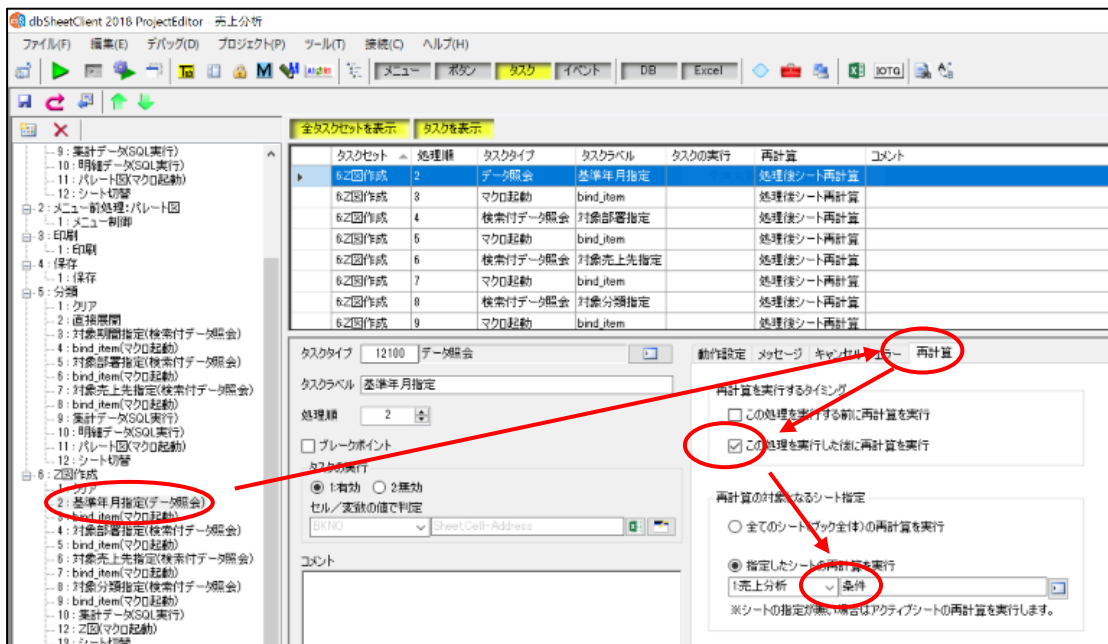
「条件」シートに「基準年月」という名前のセルを作成し展開先に設定します。



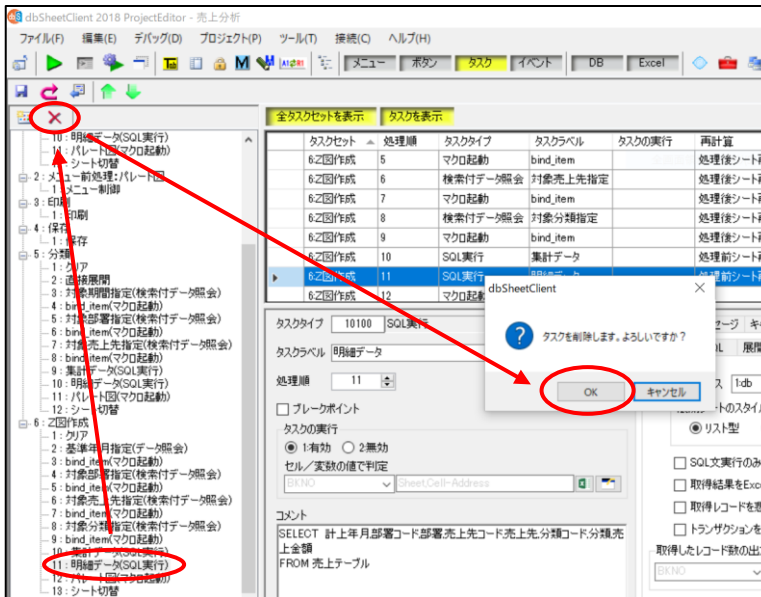
シート「条件」の「基準年月」にデータがセットされたら、その22ヶ月前の年月が自動計算されるように関数式をセットします。22ヶ月前の年月が入るセルに「スタート年月」という名前を付けます。



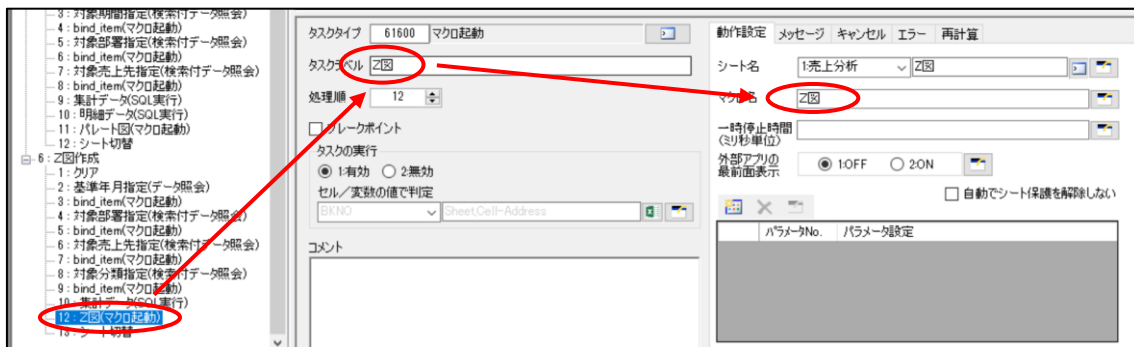
タスク実行後に「条件」シートを再計算するよう設定します。



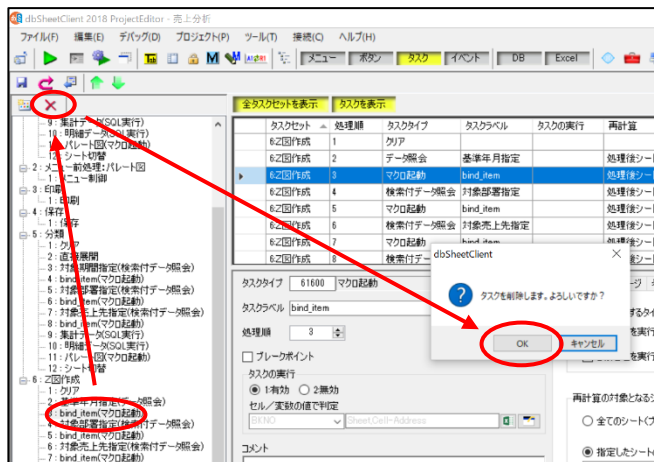
タスク「明細データ」を削除します。(※Z図では明細の提示は省略します。)



「マクロ起動」のマクロ名を「Z図」に変更します。



「基準年月指定」(検索付データ照会)直後の「マクロ起動」タスク「bind_item」を削除します。



「対象部署指定」「対象売上先指定」「対象分類」のSQL文の「計上年月」抽出条件を変更します。

このスクリーンショットは、dbSheetClientの「対象部署指定」設定画面を示しています。左側のタスクリストで「対象部署指定(検索付データ照会)」が4番目に選択されています。右側のSQLエディタには、以下のクエリが表示されています。

```
SELECT distinct 部署コード部署
FROM 売上マスタ
where 計上年月 between "BKNO!!スタート年月" and "BKNO!!基準年月"
order by 部署コード
```

このクエリ中の「計上年月 between "BKNO!!スタート年月" and "BKNO!!基準年月"」部分が赤い円で囲まれています。

このスクリーンショットは、dbSheetClientの「対象分類指定」設定画面を示しています。左側のタスクリストで「対象分類指定(検索付データ照会)」が8番目に選択されています。右側のSQLエディタには、以下のクエリが表示されています。

```
SELECT distinct 分類コード分類
FROM 売上マスタ
where 計上年月 between "BKNO!!スタート年月" and "BKNO!!基準年月"
and 部署コード in ("BKNO!!対象部署コード")
and 売上先コード in ("BKNO!!対象売上先コード")
order by 分類コード
```

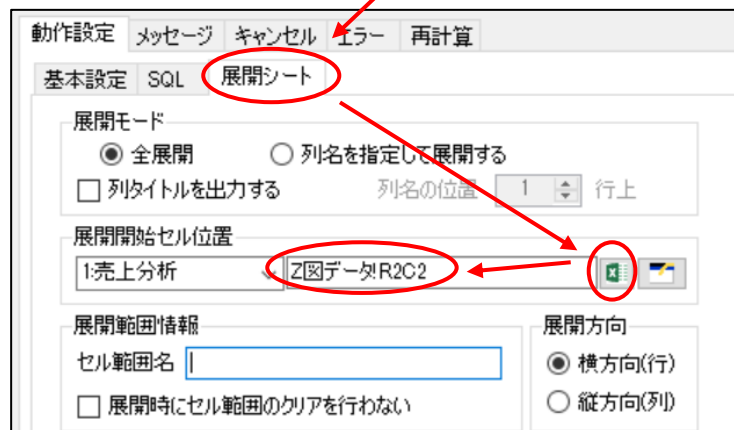
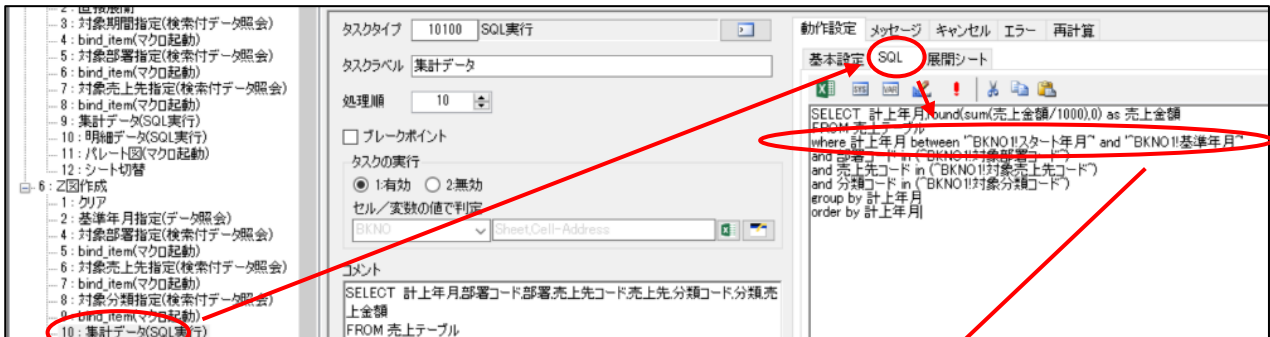
このクエリ中の「計上年月 between "BKNO!!スタート年月" and "BKNO!!基準年月"」部分が赤い円で囲まれています。

このスクリーンショットは、dbSheetClientの「対象売上先指定」設定画面を示しています。左側のタスクリストで「対象売上先指定(検索付データ照会)」が6番目に選択されています。右側のSQLエディタには、以下のクエリが表示されています。

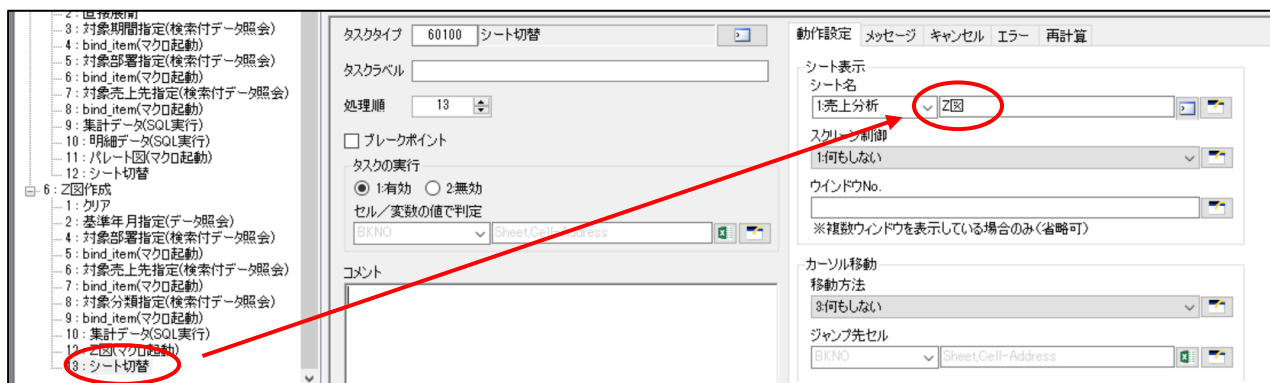
```
SELECT distinct 売上先コード売上先
FROM 売上マスタ
where 計上年月 between "BKNO!!スタート年月" and "BKNO!!基準年月"
and 部署コード in ("BKNO!!対象部署コード")
order by 売上先コード
```

このクエリ中の「計上年月 between "BKNO!!スタート年月" and "BKNO!!基準年月"」部分が赤い円で囲まれています。

「集計データ」のSQL文の「計上年月」抽出条件を変更します。
 展開先をシート「Z図データ」の 2 行 2 列目に設定します。

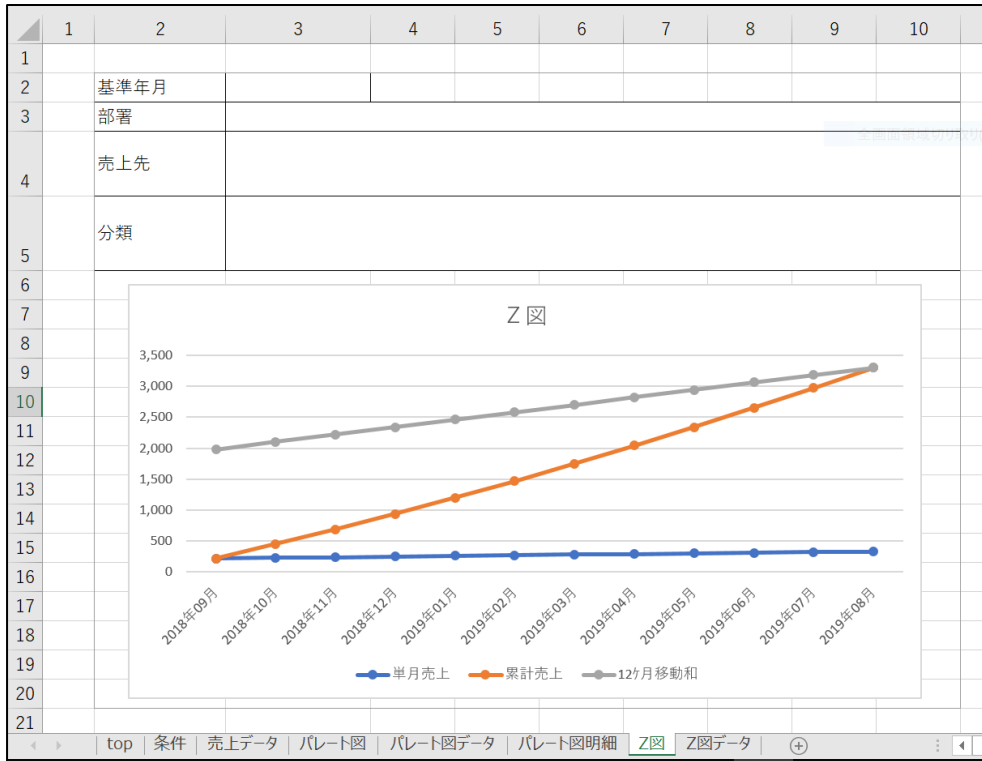


「シート切替」タスクのシート名を「Z図」に変更します。



Excel ファイルを呼び出してシート「Z図」の体裁を整えます。

A4横で印刷した時に綺麗に収まるよう文字サイズや印刷範囲、用紙向き、拡大・縮小設定を行います。



マクロ「パレート図」のプログラムをコピーしてマクロ「Z図」を作成します。

抽出条件コピーと印刷・保存設定のみ残します。

```

Sub Z図()
    '抽出条件をコピー
    '基準年月
    Sheets("Z図").Cells(2, 3) = Left(Sheets("条件").Cells(24, 3), 4) & "年" & Right(Sheets("条件").Cells(24, 3), 2) & "月"
    '対象部署
    Sheets("Z図").Cells(3, 3) = Replace(Sheets("条件").Cells(10, 3), "", "")
    '対象売上先をコピー
    Sheets("Z図").Cells(4, 3) = Replace(Sheets("条件").Cells(22, 3), "", "")
    '対象分類をコピー
    Sheets("Z図").Cells(5, 3) = Replace(Sheets("条件").Cells(13, 3), "", "")

    '印刷対象シート
    Sheets("条件").Cells(15, 3) = "Z図"
    '保存ファイル名
    Sheets("条件").Cells(17, 3) = "Z図.xlsx"
    '保存対象シート
    Sheets("条件").Cells(19, 3) = "Z図,Z図データ"

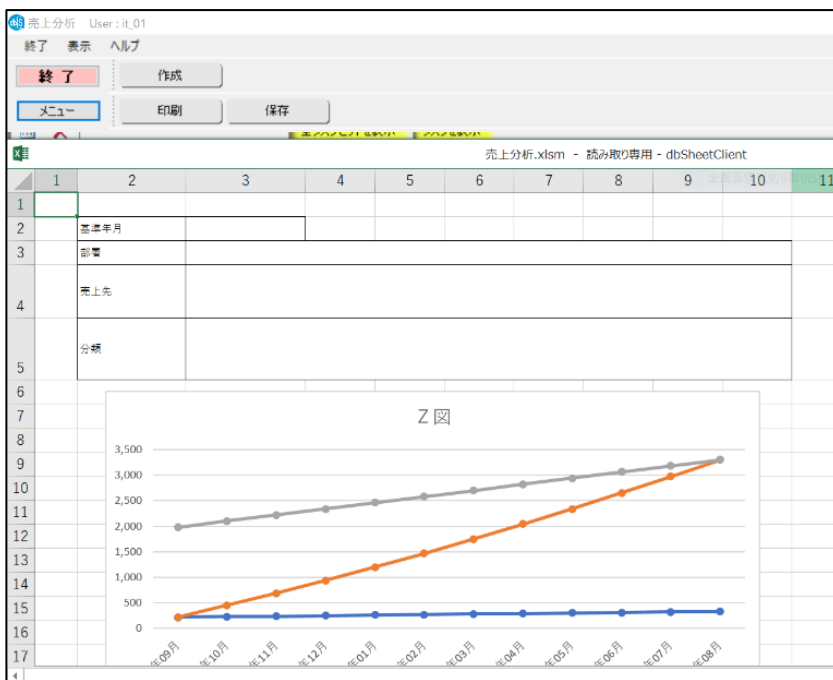
    Cells(1, 1).Select
End Sub
    
```

1.4.4 デバッグモードで実行

左メニュー「Z図」をクリックします。

左側のメニュースペースが閉じられ、シート「Z図」に切り替わっているか確認します。

画面上部の「作成」ボタンをクリックします。

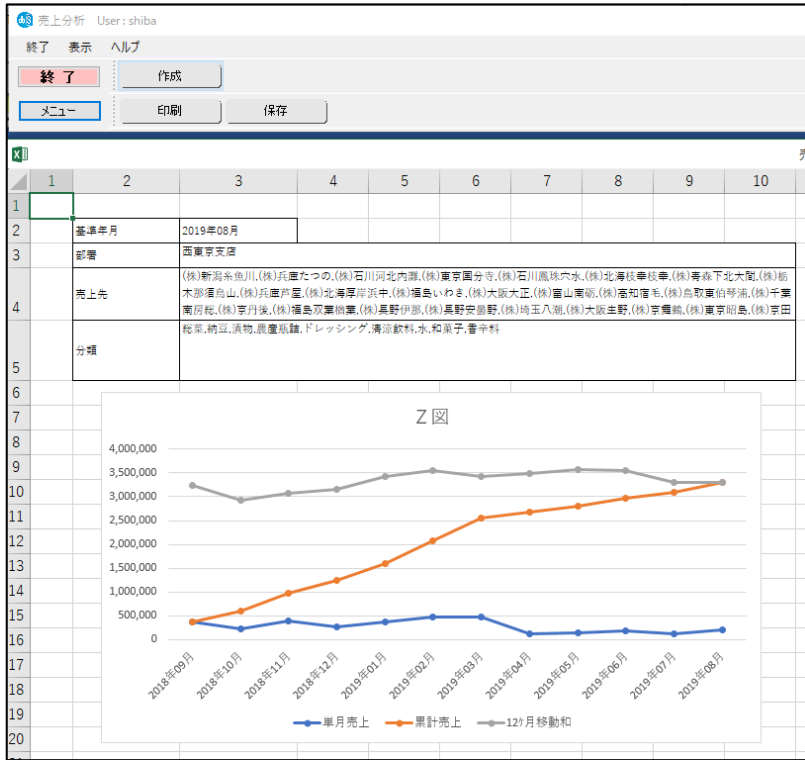


「基準年月」→「部署」→「売上先」→「分類」の順に選択画面が表示されるか確認します。

The sequence shows the following selection steps:

- 部署 (Department) Selection:** A dialog box with a list of departments. '西東京支店' (Western Tokyo Branch) is selected.
- 売上先 (Sales Customer) Selection:** A dialog box with a list of sales customers. '(株)新島谷川' (Shinimigawa Co., Ltd.) is selected.
- 分類 (Classification) Selection:** A dialog box with a list of classifications. '結露' (Condensation) is selected.

指定した抽出条件でZ図が描画されるか確認します。

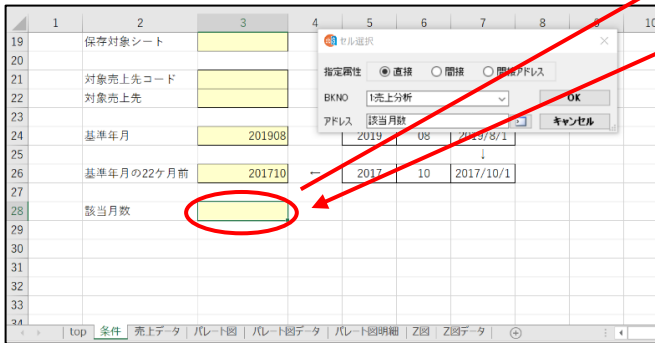
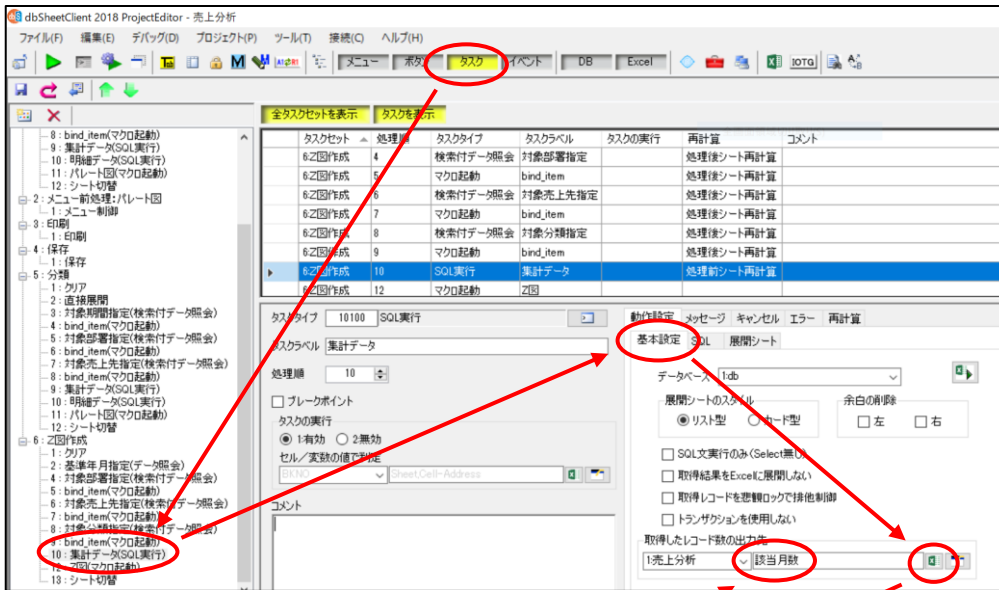


1.4.5 条件判定分岐タスクの適用

Z図の描画には基準年月を含めて過去22ヶ月、全部で23ヶ月分のデータが必要です。
 新規に実績が上がり始めた売上先や分類、もしくは抽出条件の指定内容によっては23ヶ月分が揃いません。
 そこで、23ヶ月分が揃わなかった場合に警告メッセージを表示するようになっています。

1	2	3	4	5	6	7	8	9	10
1	No	計上月	完	余額	No	計上年月	単月売上	累計売上	12ヶ月移動和
2	1	200101		110	1	2001年12月	220	220	1,980
3	2	200102		120	2	2002年01月	230	450	2,100
4	3	200103		130	3	2002年02月	240	690	2,220
5	4	200104		140	4	2002年03月	250	940	2,340
6	5	200105		150	5	2002年04月	260	1,200	2,460
7	6	200106		160	6	2002年05月	270	1,470	2,580
8	7	200107		170	7	2002年06月	280	1,750	2,700
9	8	200108		180	8	2002年07月	290	2,040	2,820
10	9	200109		190	9	2002年08月	300	2,340	2,940
11	10	200110		200	10	2002年09月	310	2,650	3,060
12	11	200111		210	11	2002年10月	320	2,970	3,180
13	12	200112		220	12	2002年11月	330	3,300	3,300
14	13	200201		230					
15	14	200202		240					
16	15	200203		250					
17	16	200204		260					
18	17	200205		270					
19	18	200206		280					
20	19	200207		290					
21	20	200208		300					
22	21	200209		310					
23	22	200210		320					
24	23	200211		330					

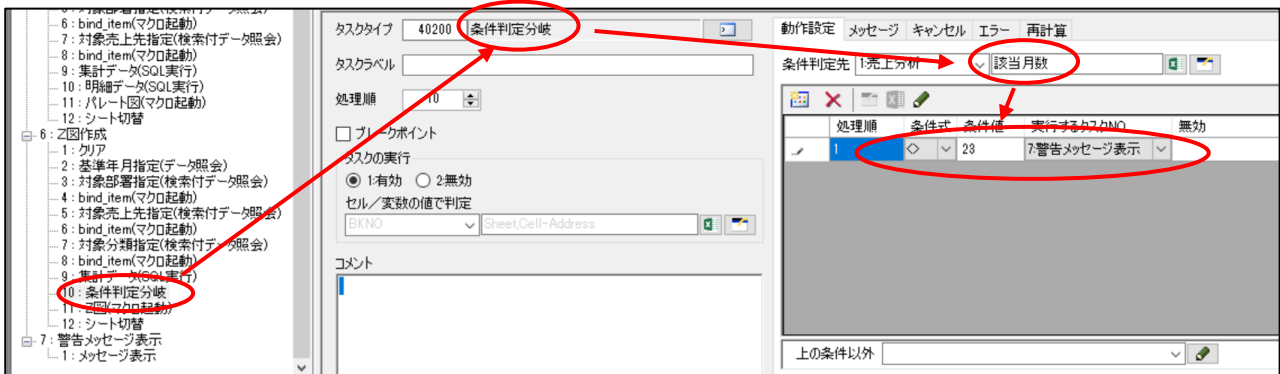
集計データを抽出する「SQL実行」タスクの「取得したレコード数の出力先」を設定



タスク「集計データ」の後ろに「条件判定分岐」タスクを追加します。

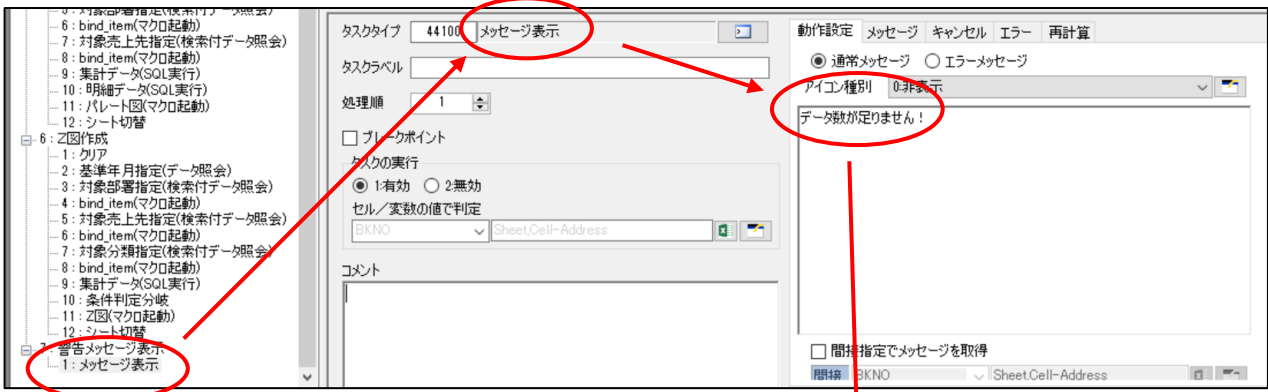
「条件判定先」を「該当月数」に設定し条件を「23 でない場合」は警告メッセージを表示するタスクを実行します。

※23の場合は何も設定しない。

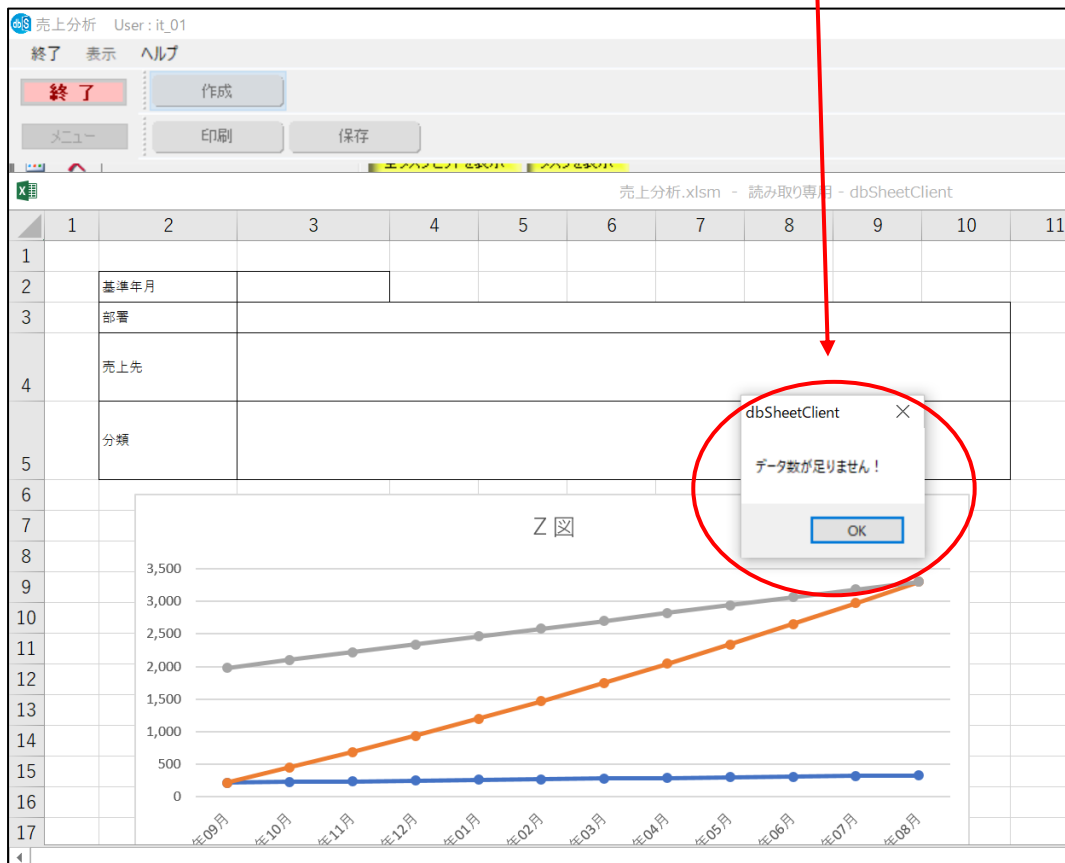


警告メッセージを表示するタスクセットを追加します。

タスクタイプは「メッセージ表示」を選択し、メッセージ内容に「データ数がたりません！」と入力します。

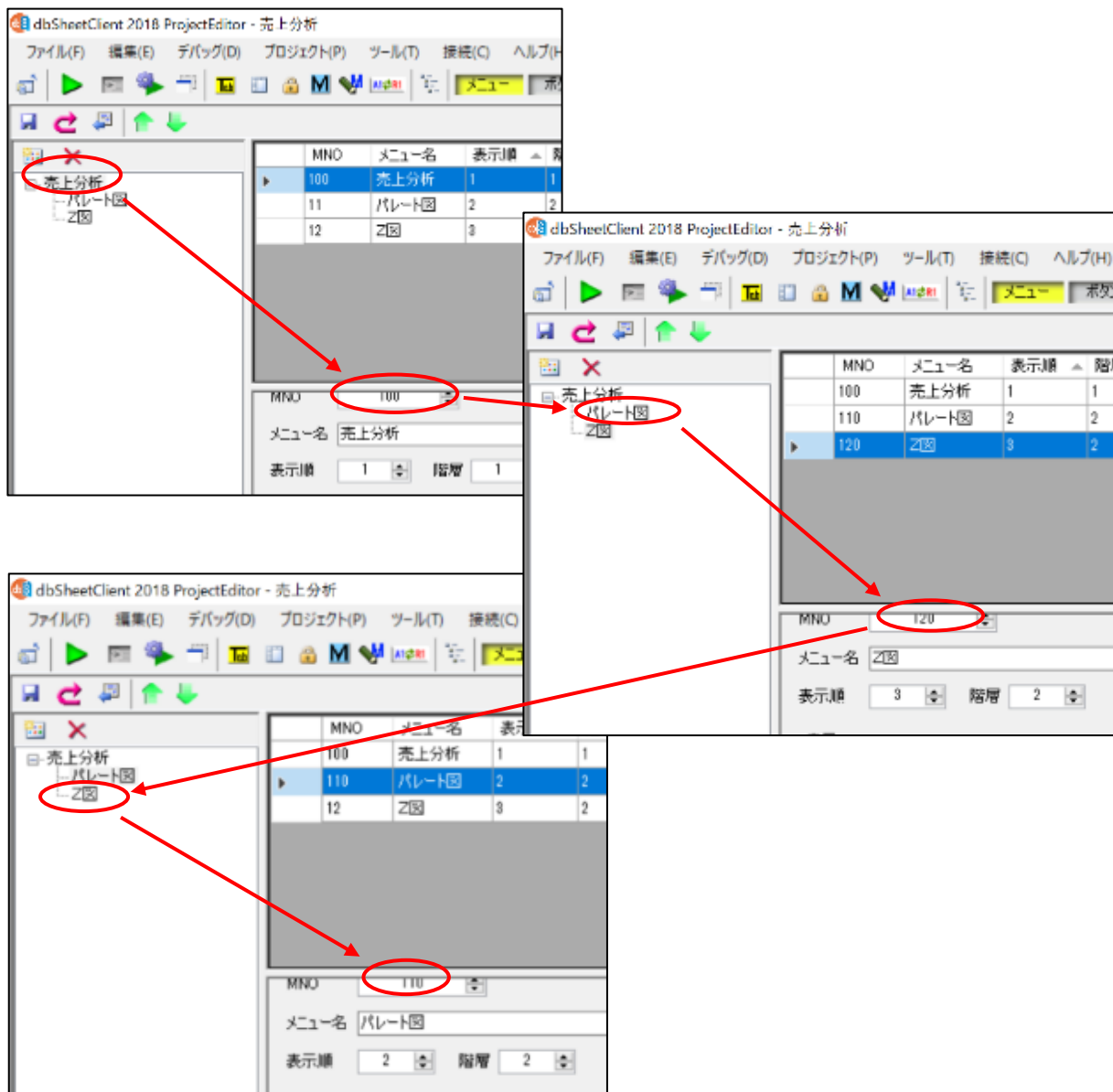


条件を満たさない抽出条件でZ図を作成してみて警告メッセージが表示されるか確認します。

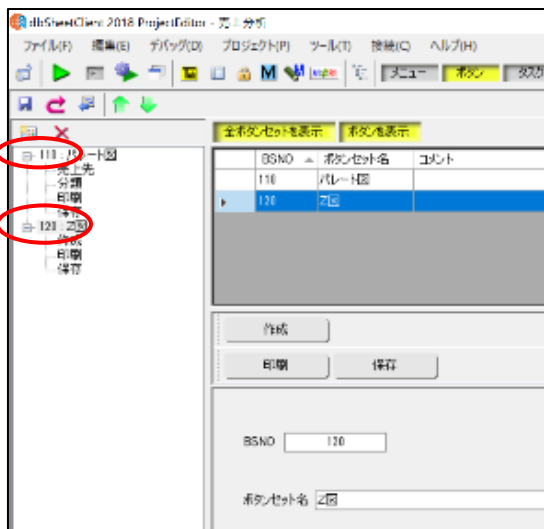


1.4.6 メニュー番号を振り直す

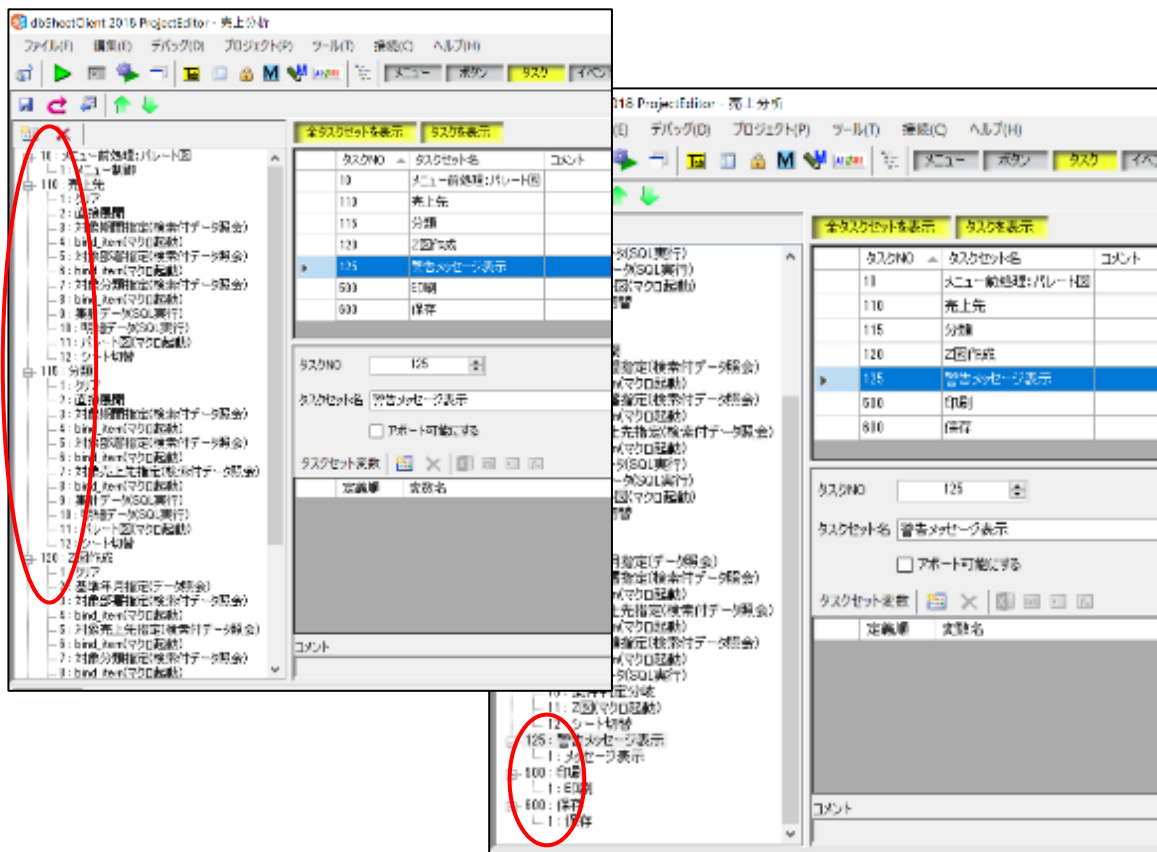
開発後のメンテナンスがしやすいようにメニュー番号を振りなおします。



1.4.7 ボタン番号を振り直す



1.4.8 タスク番号を振り直す

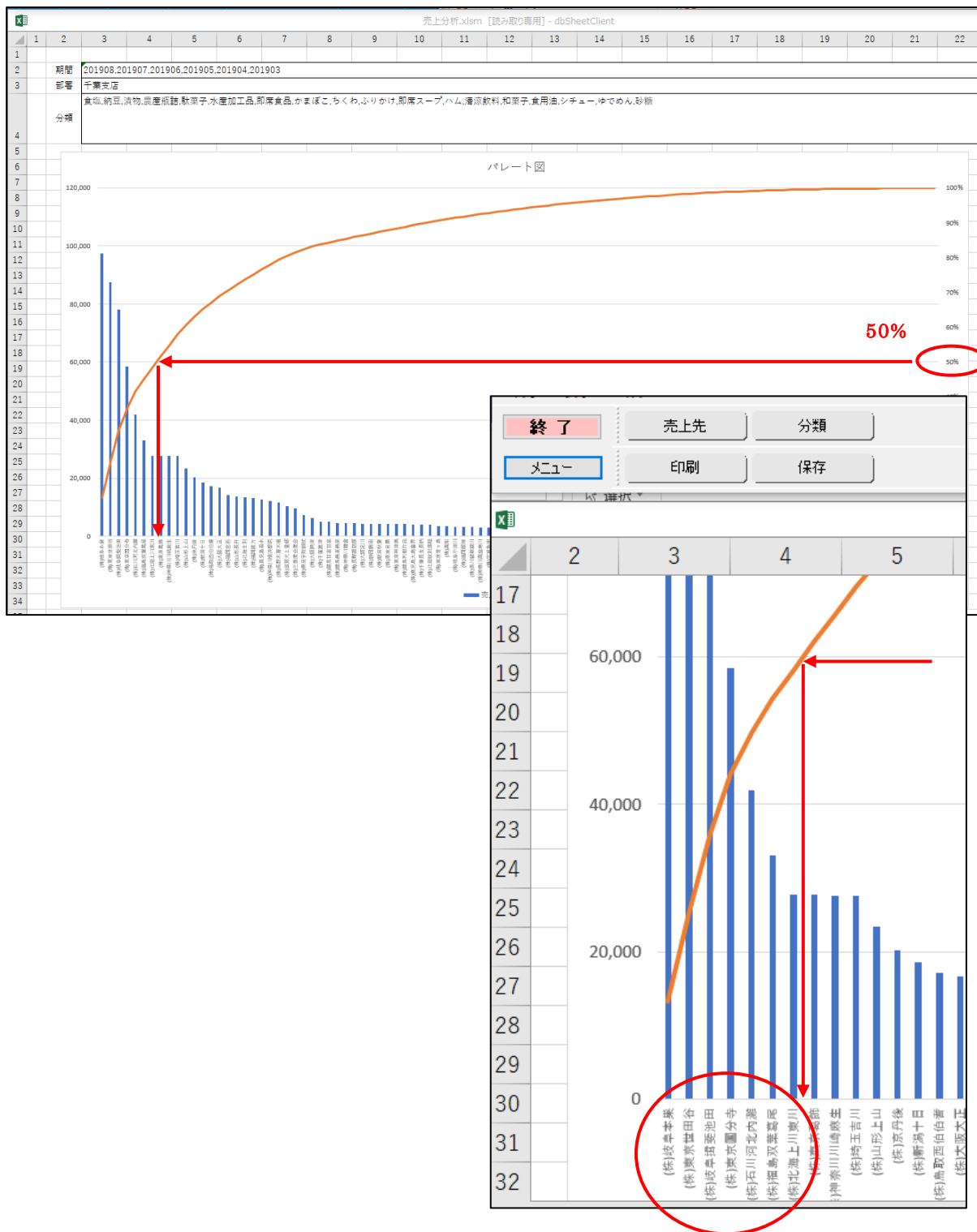


5.5 売上分析

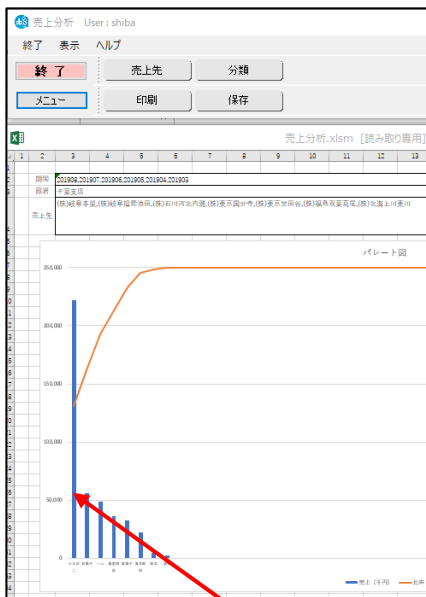
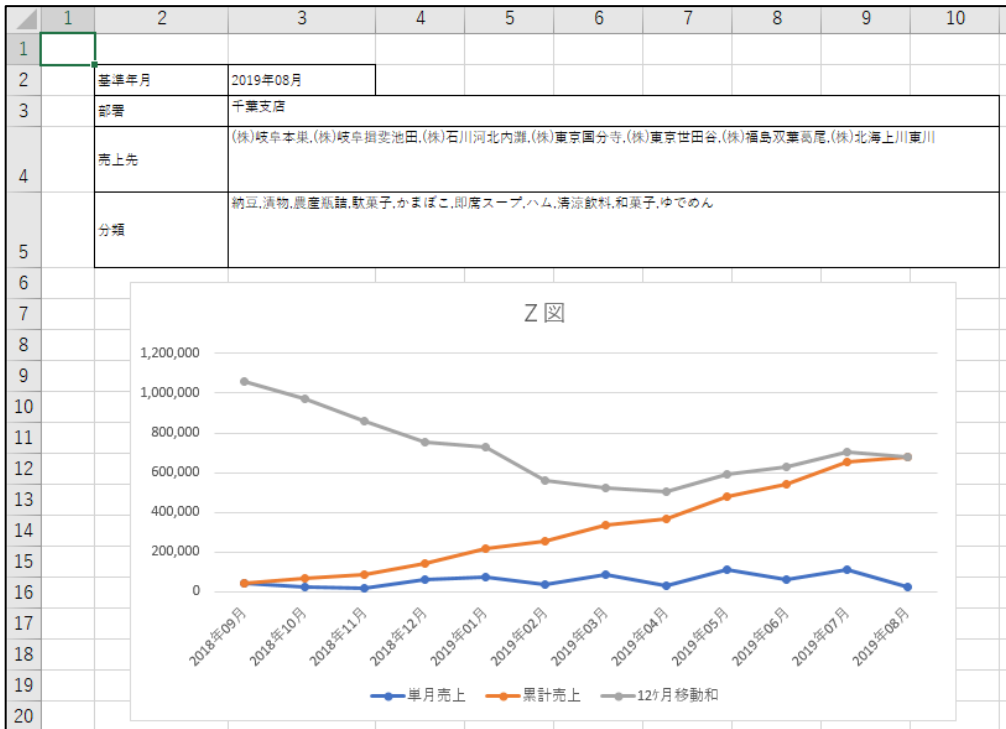
それでは完成した「売上分析」プロジェクトを使って具体的な売上分析を行ってみましょう。

千葉支店の直近6ヶ月の「売上先」を構成するパレート図を作成します。

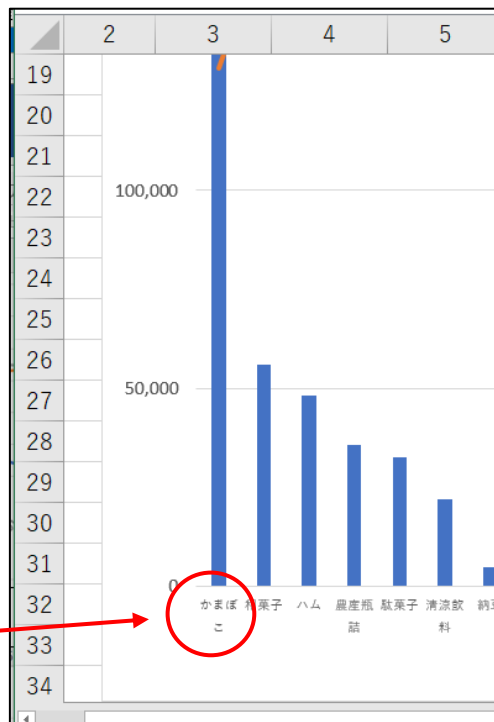
全売上の50%を構成する売上先を確認します。→ 7社あります。



全売上の50%を構成する売上先7社を指定して、Z図を作成します。
 12ヶ月の移動和を見ると下降を辿っていたが、4ヶ月前から持ち直してきたことが確認できます。
 単月の売上が隔月で上下しています。
 さらに、この7社で直近6ヶ月の分類構成をパレート図を使って確認すると
 分類名「かまぼこ」が圧倒的な構成比率であることが確認できます。
 今後の対策を策定する上で作成したグラフが役立つでしょう。



かまぼこ



6. 予算を収集する

経営層が最も欲しい情報は未来の情報です。

すなわち、売上に繋がる案件がどのくらいあるか、それはいつ売上になるか、その確度は？

受注案件にリスクが発生する可能性はないか、年度末までに経費はどのくらい発生するか等々です。

これらの各現場にしかない情報を速く正確に収集し分析することが変化に対応した経営に必須です。

dbSheetClient ユーザー会の事例発表を見ていても、この情報収集業務に Excel と dbSheetClient の組合せが絶大な効果を発揮していることがわかります。

新しいシステムに抵抗感のある年配者(役職者が多い)も「Excel なら」という安堵感があるようです。

本章では、各部署から年度末までの予算を収集し、年度末の達成予算を算出するシステムを例として次の事項についてご紹介します。

- ① 予算収集に潜在する現場ニーズにどう対応できるか
- ② データベースへのアクセス制御
- ③ メール通知機能を使ったワークフロー

6.1 予算収集の要点

著者の在籍する会社でも dbSheetClient 導入後、早い時期にこの「情報収集業務」に適用しました。情報収集と言っても様々な形態が存在します。

たとえば次年度の予算(売上&経費)は年に一回、すべての部署が対象となります。

業務報告では営業部門が対象で毎月、行います。

様々な情報収集システムを開発する過程で気が付いた要点を挙げてみます。

2.1.1 ユーザー志向

何の説明も受けず何も参照しないで直感的に入力作業に取り掛かれるシステム

キャッシュディスクペンスーのようなシステムが理想です。

具体的には次の点を考慮することです。

- ① 画面遷移やボタンクリックの回数を減らす
- ② 初歩的な入力ミスや更新忘れを予防する
- ③ 入力の参考になる関連情報をリンクする

2.1.2 取りまとめ部署作業負荷軽減の視点

Excel ファイルを配布して現場で予算入力したものを回収し集計する作業。

この業務に dbSheetClient を適用すると早く正確にかつ作業負荷が軽減されます。

と言われても取りまとめ部署(一般的には社長室とか経営企画部)の担当者はピンときません。

なぜなら裏に隠れた事前事後の作業や周辺業務がどう改善されるのか不明だからだと思います。

- ① 組織改編や入力担当者異動への対応
- ② 期限に間に合わせるための進捗管理(ユーザーへの督促)
- ③ IT操作が得意でないユーザーに対するフォロー
- ④ ネットワークが利用できない部署(海外現地法人等)への対応
- ⑤ 予算回収後の集計作業
- ⑥ 経営層からの査定指示(ユーザーに再度入力作業を強いる)
- ⑦ 予算精度を上げる対策(予算と実績の乖離を監視)

これらの作業をバックグラウンドで行なう取りまとめ部署の負荷軽減への視点が大切です。

2.1.3 情報システム部門の立ち位置

情報システム部門は、リリース後、ユーザーが自立して運用できるシステムを構築することです。

予算収集は毎年継続して発生する業務です。

そして、ビジネス環境の変化、それに伴う組織改編や人事異動は避けられません。

少々の環境変化があってもユーザーだけで対応できるシステムにするという視点を持つことが大切です。

そしてユーザーからの改善要望に短期間で応えられるようなシステム構造にしておくことです。

6.2 予算テーブルの設計

経営層から社長室に下半期の売上見込を各部署から収集するよう指示がありました。

売上見込が収集され次第、事業戦略会議を開催するとのことでした。

売上見込をどのような様式で収集するかについて考えてみましょう。

2.2.1 シンプル

部署別に、前年度実績を表示して、その右隣りに今年度見込を入力する様式
見込を裏付ける資料は各部署で作成して事業戦略会議に臨む
予算入力の負荷は無に等しいですが、資料作りが大変かもしれません。

	1	2	3	4	5	6
1					単位：千円	
2		部署コード	部署	前年度 売上	今年度 売上見込	
3		A01	西東京支店	60,000	65,000	
4		A02	東東京支店			
5		B01	千葉支店			
6		B02	埼玉支店			
7		C01	名古屋支店			
8		C02	大阪支店			
9		D01	九州支店			
10		D02	四国支店			
11		D03	北陸支店			
12		D04	北海道支店			

2.2.2 顧客志向

主要売上先別に売上見込入力

	1	2	3	4	5	6	7
13							
14					単位：千円		
15		部署コード	部署	売上先	前年度 売上	今年度 売上見込	
16		A01	西東京支店	A社	20,000	22,000	
17		A01	西東京支店	B社			
18		A01	西東京支店	C社			
19		A01	西東京支店	・			
20		A01	西東京支店	・			
21		A01	西東京支店	・			
22							

2.2.3 商品志向

主要商品分類別に売上見込を入力

	1	2	3	4	5	6	7
13							
14						単位：千円	
15		部署コード	部署	商品分類	前年度 売上	今年度 売上見込	
16		A01	西東京支店	X商品	40,000	41,000	
17		A01	西東京支店	Y商品			
18		A01	西東京支店	Z商品			
19		A01	西東京支店	・			
20		A01	西東京支店	・			
21		A01	西東京支店	・			
22							

2.2.4 製造業、メーカー代理店

商品分類別、月別に売上見込を入力

需要に合わせた生産計画を策定するためには売上時期を考慮しなくてはなりません。

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1															
2															単位：千円
3		部署 コード	部署	商品分類	前年度 売上	今年度 売上見込	①今年度確 定売上	②今年度見 込み入力計	2019年9月	2019年10月	2019年11月	2019年12月	2020年1月	2020年2月	2020年3月
4		A01	西東京支店	X商品	40,000	43,000	25,000	18,000	3,000	2,300	3,300	2,000	1,200	4,400	1,800
5		A01	西東京支店	Y商品											
6		A01	西東京支店	Z商品											
7		A01	西東京支店	・											
8		A01	西東京支店	・											
9		A01	西東京支店	・											
10															

2.2.5 ハイブリッド

理想形は売上先別、商品分類別、月別に売上見込を入力

しかし、売上先と商品分類の組合せは無数にあり、それを月別に展開すると、ユーザー部門からクレームが出ることは間違いありません(笑)

本書では敢えてこの理想形を目指すことにします。

売上先、商品、計上時期を特定して予算を立てるには戦略戦術が必要です。

予算を立てる過程で事業戦略会議に向けた準備が進むのではないのでしょうか。

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	1
1																	
2																	単位：千円
3		部署 コード	部署	売上先	商品分類	前年度 売上	今年度 売上見込	①今年度確 定売上	②今年度見 込み入力計	2019年9月	2019年10月	2019年11月	2019年12月	2020年1月	2020年2月	2020年3月	
4		A01	西東京支店	A社	X商品	40,000	43,000	25,000	18,000	3,000	2,300	3,300	2,000	1,200	4,400	1,800	
5		A01	西東京支店	A社	Y商品												
6		A01	西東京支店	B社	Z商品												
7		A01	西東京支店		・												
8		A01	西東京支店		・												
9		A01	西東京支店		・												
10																	

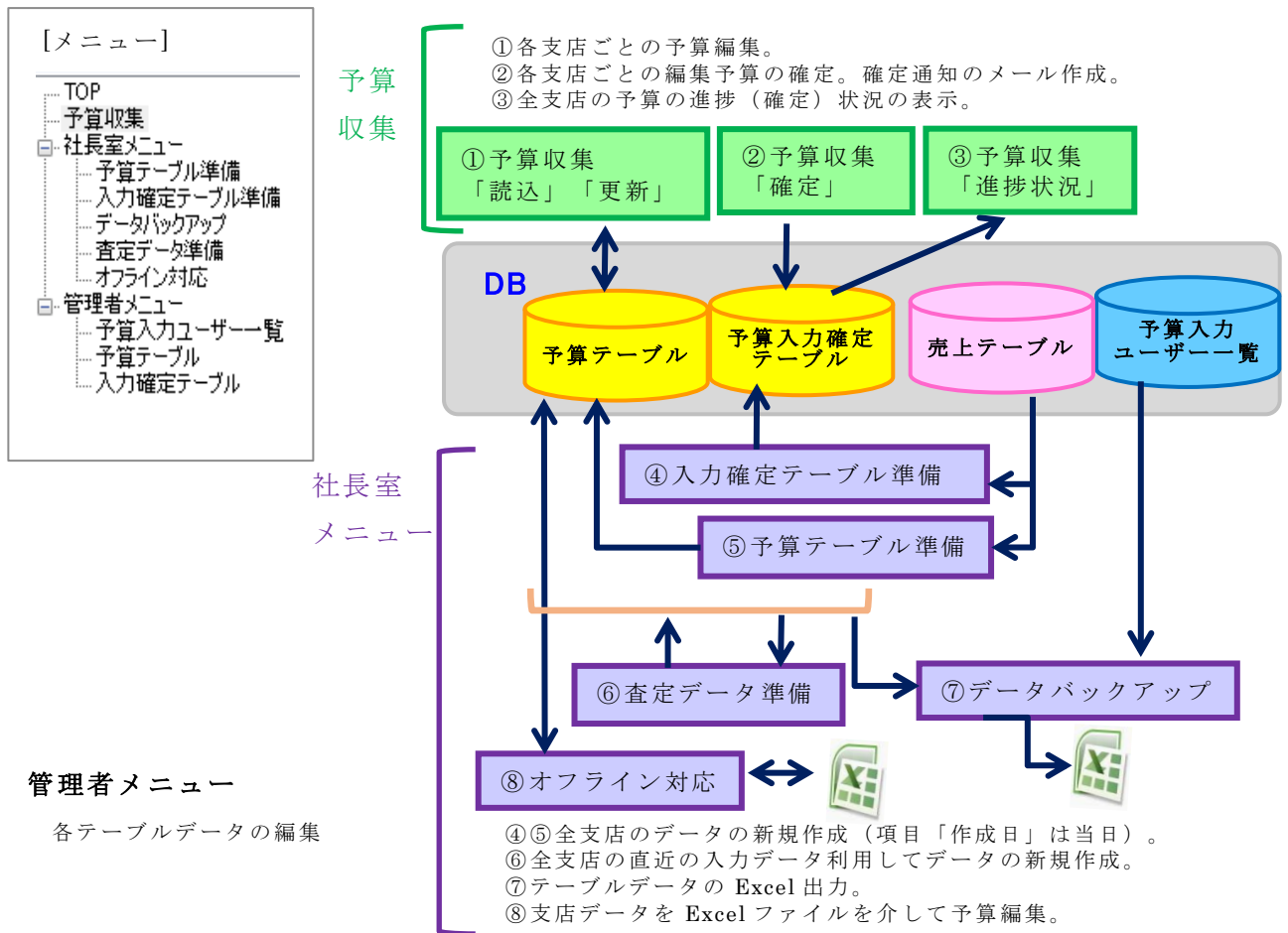
6.3 動かしてみましよう

既に作成済みのサンプルのプロジェクト[予算収集_完成]を動作させてみましょう。
 サンプルプロジェクトのシステム概要は以下の通りです。

6.3.1 システム概要

動作環境は以下の通りです。

- ・ローカル環境(1 台の PC 内で dbSheetClient サーバー利用無し)の開発版のデバッグ実行。
- ・dbSheetClient 開発版 Ver8.0.7.0 以上
- ・データベース:Microsoft SQL Server 2012 Express 以上(Express 利用)
- ・Excel 2013 以上または Office365
- ・プロジェクト[予算収集_完成]の機能は以下の通りです。



6.3.2 データベース概要

①データベース名： DBC 完成 (SQL Server Express 2016)

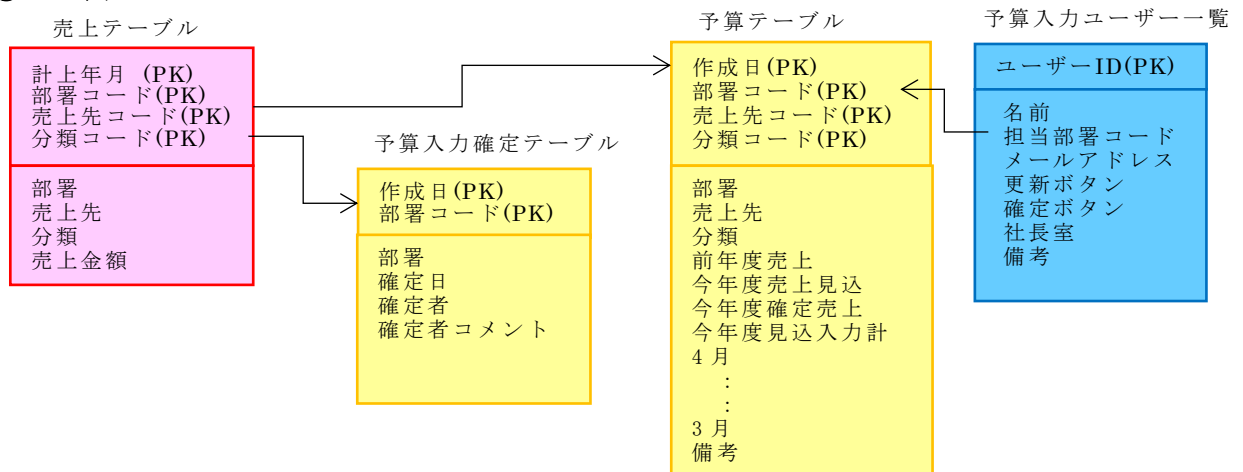
②テーブル名

DBC完成

- データベースダイアグラム
- テーブル
 - システムテーブル
 - FileTables
 - 外部テーブル
 - dbo.売上テーブル
 - dbo.予算テーブル
 - dbo.予算入力ユーザー一覧
 - dbo.予算入力確定テーブル

NO	テーブル名	目的
1	売上テーブル	売上データから全部署(支店)の初期予算データを作成
2	予算テーブル	各部署(支店)の予算データを管理。予算データを作成した日(項目[作成日])が最新のデータが現在の編集対象データ。
3	予算入力ユーザー一覧	利用ユーザーを管理。機能の権限等の管理。
4	予算入力確定テーブル	各部署(支店)の確定状況を管理。

③ER図

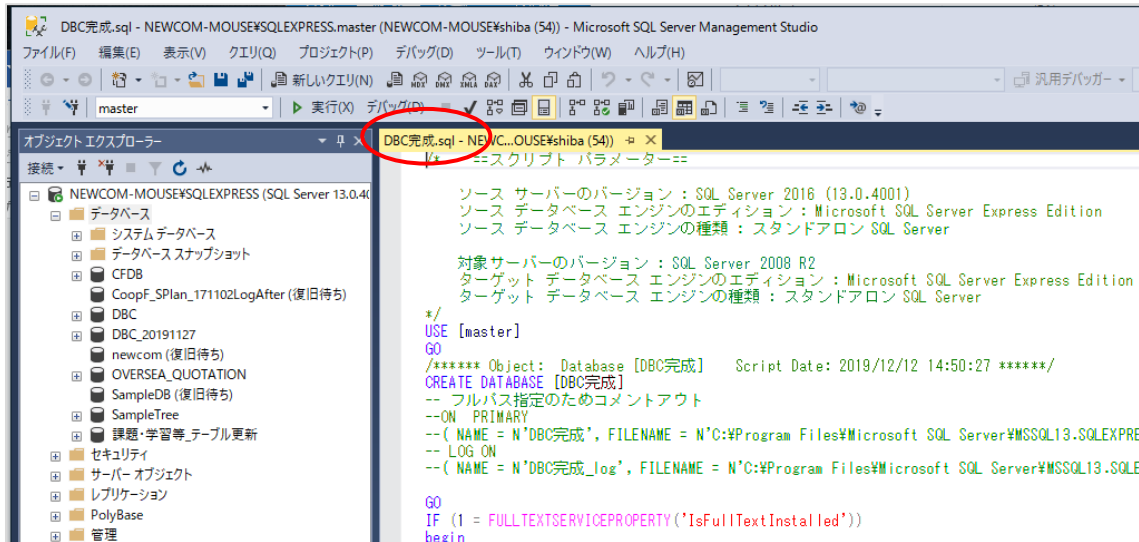


6.3.3 動作環境 (DB) の構築

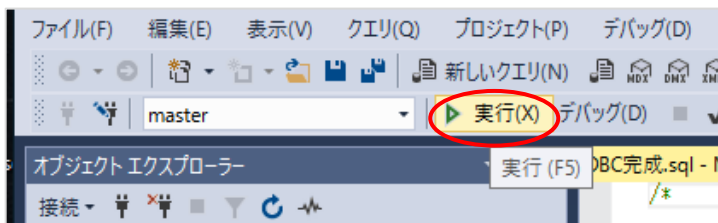
データベース[DBC 完成]を以下の手順で作成します。

SQL Server Management Studio から [DBC 完成.sql]を実行します。

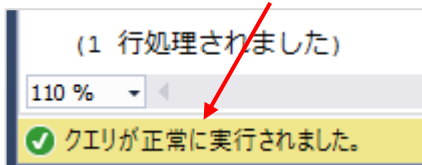
SQL Server Management Studio に [DBC 完成.sql]ファイルをドラッグ&ドロップしてファイルを開きます。



実行します。

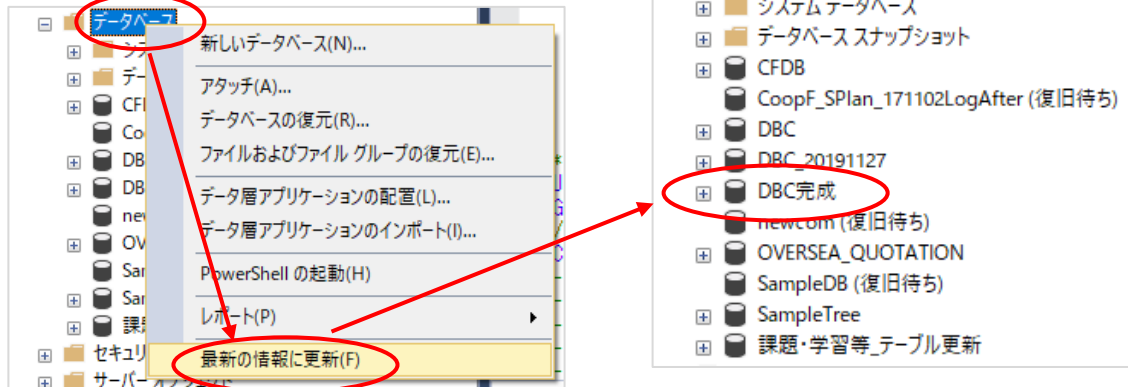


正常に実行すると以下を表示します。



実行後、作成した[DBC 完成]は「データベース」の中にはすぐには表示しないため、「最新の情報に更新」し、作成できたことを確認します。

「データベース」を右クリック



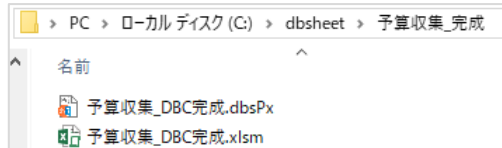
DBC 完成には以下のテーブルとデータが設定済です。

- ・「予算テーブル」、「予算入力確定テーブル」、「予算入力ユーザー一覧」、「売上テーブル」

6.3.4 動作環境（プロジェクト）の設定

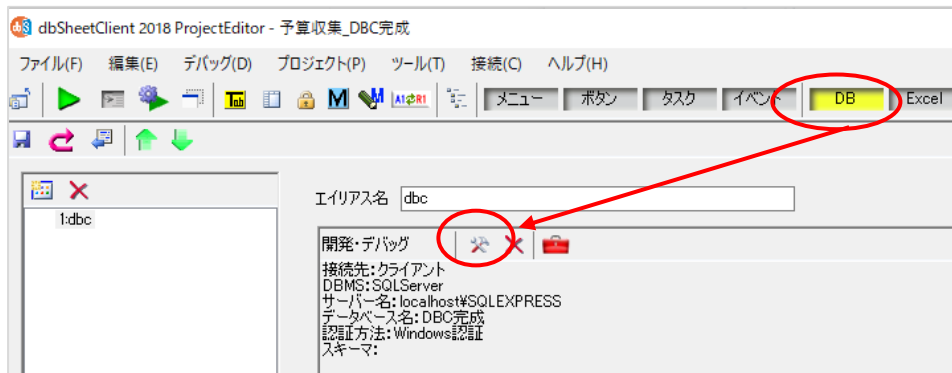
プロジェクトファイル[予算収集_完成]のフォルダを PC 内の任意フォルダに設定します。

例) C:\¥dbsheet のフォルダの下に設定



作成した[DBC 完成]の DB に接続する設定を行います。

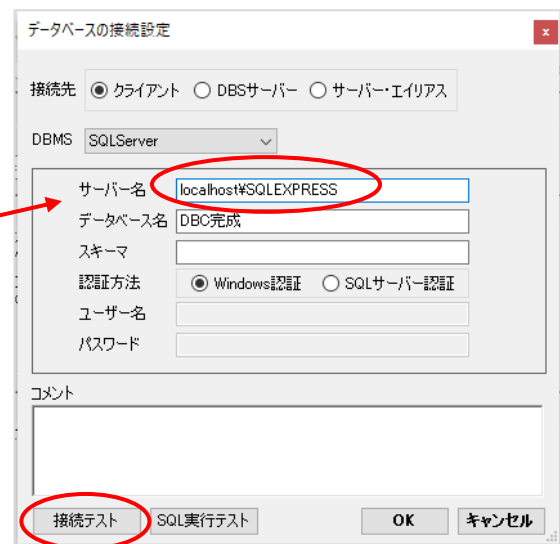
プロジェクトファイルを開き「DB」→「データベースの設定」を選択します。



「サーバー名」が localhost(実行 PC)の
SQLEXPRESS となっています。

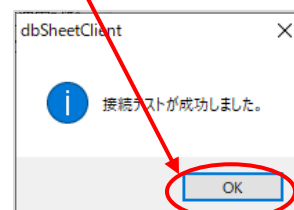
SQL Server Management Studio を参照して、
サーバー名を適切に設定します。

(SQL Server Express のインストール時に
設定した名称となっています)



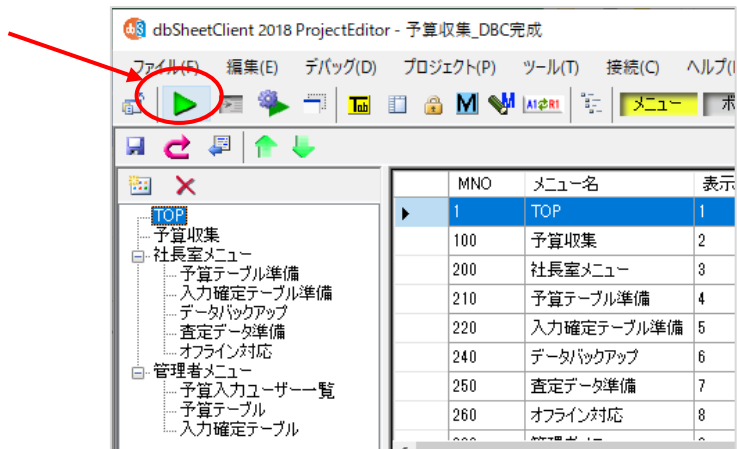
設定後、「確認テスト」で DB 接続の確認をします。

正常に接続できた場合は右図を表示します。



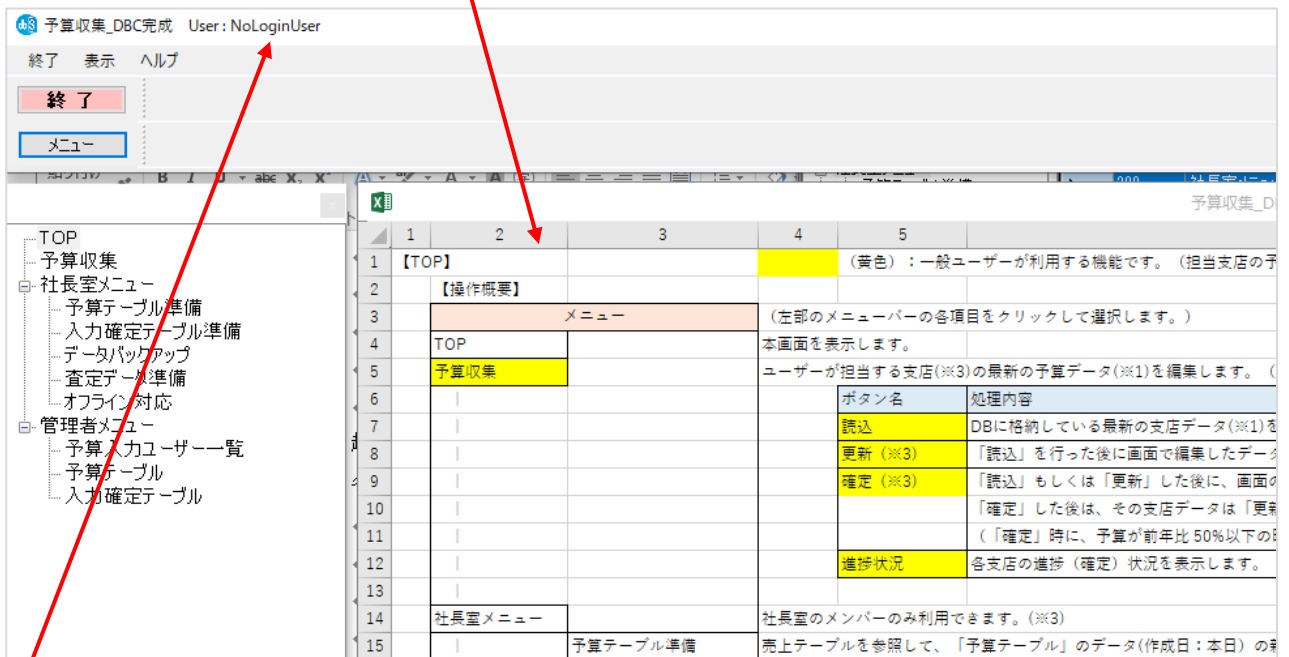
6.3.5 デバッグ実行

「デバッグ実行」します。



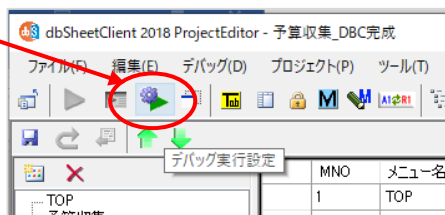
起動後、以下を表示します。

各メニュー、各ボタンの内容は本シート[**TOP**]に記載していますので、参考にして操作をお願いします。

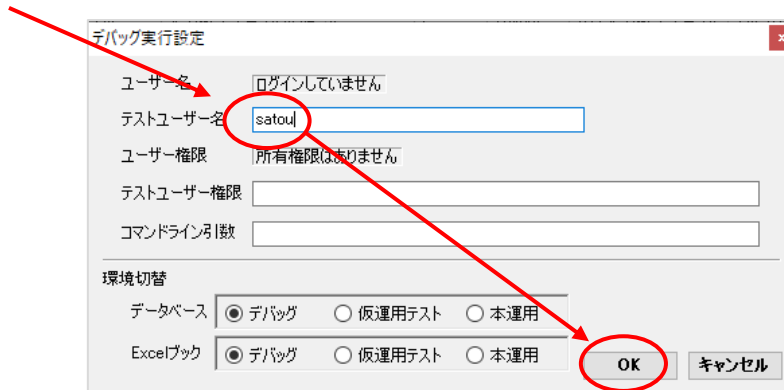


サーバー接続(ログイン)をしていない状態で、かつ「デバッグ実行設定」を行っていない場合は、User は NoLoginUser として実行します。

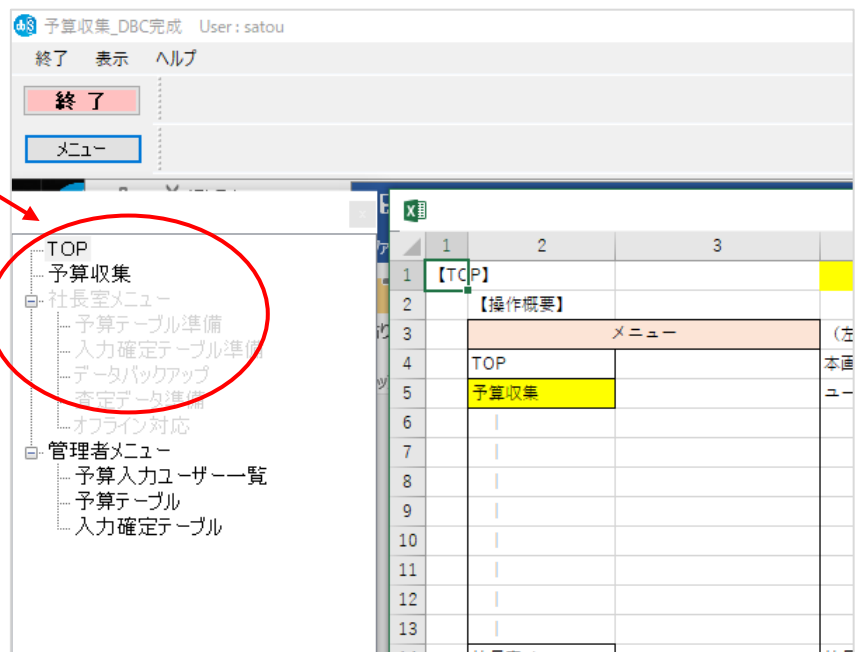
「デバッグ実行設定」では、デバッグ実行時の User の設定ができます。



User を satou で実行したい場合は、デバッグ実行前に「デバッグ実行設定」を行います。



User:satou に設定後、起動するとメニュー「社長室メニュー」が無効となります。



ユーザーによって利用できる機能に制限をかけることが可能です。

登録している User、権限等はメニュー:管理者メニュー>予算入力ユーザー一覧 で参照、編集できます。

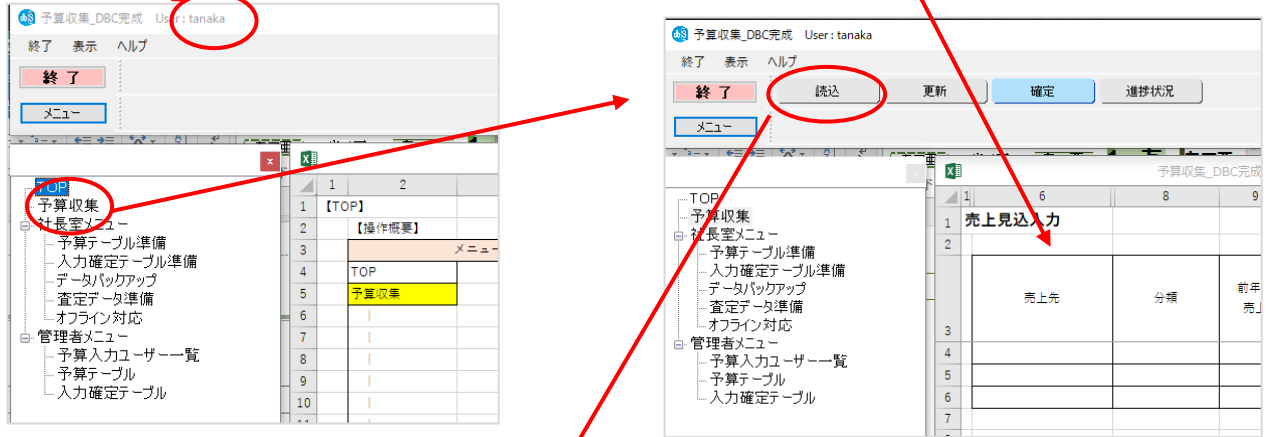
削除	ユーザーID	名前	担当部署コード	メールアドレス	更新ボタン	確定ボタン	社長室
	NoLoginUser	NoLogin	all	nologin@xxxxxx.co.jp	1	1	1
	satou	佐藤 三郎	C02.D01	satou@xxxxxx.co.jp			
	suzuki	鈴木 三郎	B01	suzuki@xxxxxx.co.jp	1		1
	tanaka	田中 一郎	A01	tanaka@xxxxxx.co.jp	1	1	1
	yamada	山田 四郎	all	yamada@xxxxxx.co.jp			1

User の tanaka は担当部署コード A01(西東京支店)、予算収集の更新ボタン 1 (更新可)、確定ボタン 1(確定可)となっています。(以降で User:tanaka を利用します)

6.3.6 デバッグ実行：予算収集

User を tanaka として「予算収集」をします。(前記「デバッグ実行設定」参照)

初期表示ではデータ表示はありません。



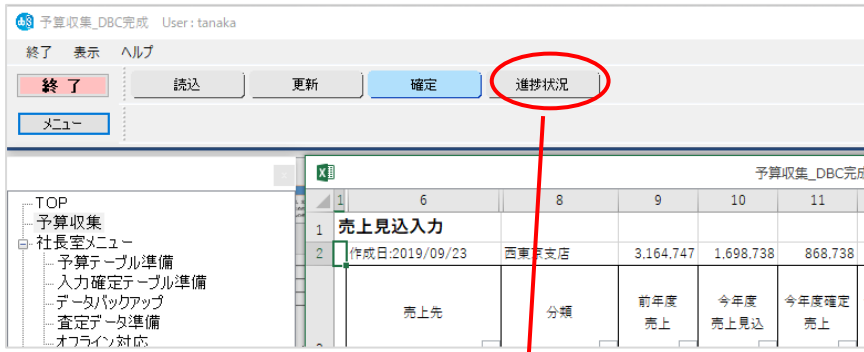
「読み込」で tanaka の担当部署：西東京支店 のデータを表示します。



予算データ(例:8行目の9月、10月)を入力して「更新」します。



現在の全支店の「進捗状況」(確定状況)を確認します。



部署コード	部署	前年度売上	今年度売上見込	確定日	確定者
A01	西東京支店	3,164,747	1,648,738		
A02	東東京支店	1,621,946	461,333		
B01	千葉支店	3,228,480	2,338,230		
B02	埼玉支店	1,116,452	540,604		
C01	名古屋支店	1,882,794	731,857		
C02	大阪支店	2,216,809	772,554		
D01	九州支店	1,011,000	449,586		
D02	四国支店	844,694	194,134		
D03	北陸支店	973,102	2,752,412	2019/10/4	山田四郎
D04	北海道支店	594,849	111,633		

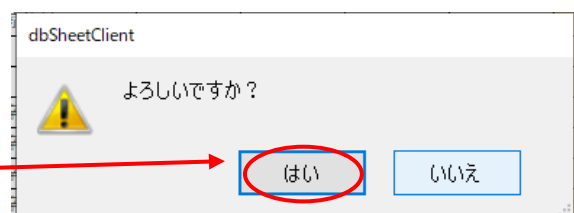
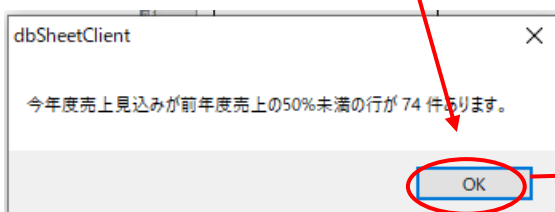
サンプルのデータでは、「北陸支店」のみが確定し、tanaka の西東京支店を含め他の支店は未確定(「更新」可能)な状態です。

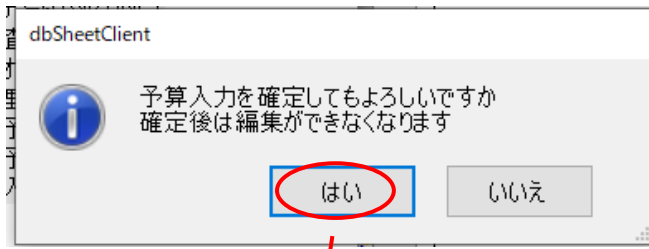
再度「読み込み」して、tanaka の西東京支店のデータ確認し「確定」をします。



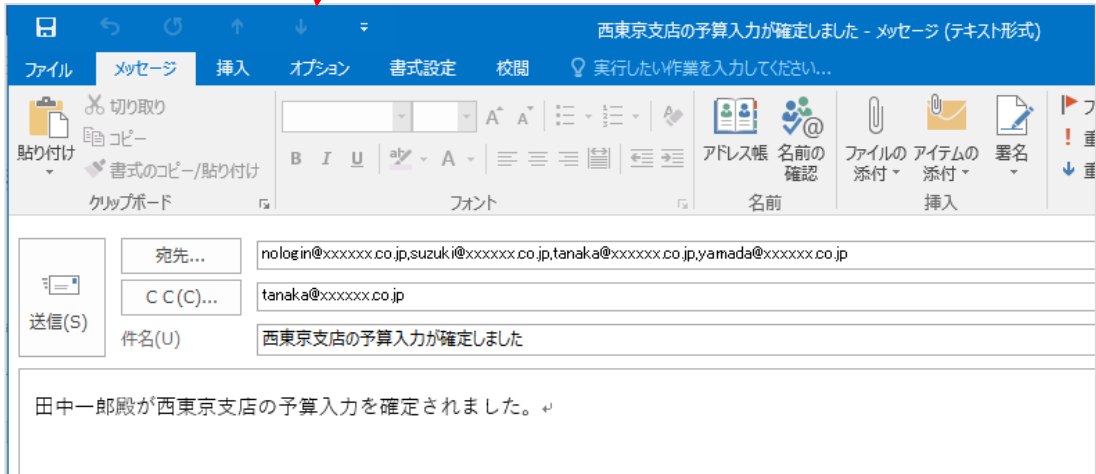
右端の列「今年度売上見込 check (前年比 50%以上)」が「×」(50%未満)が 74 件あるため、メッセージを表示します。

「確定」する場合は、「はい」。

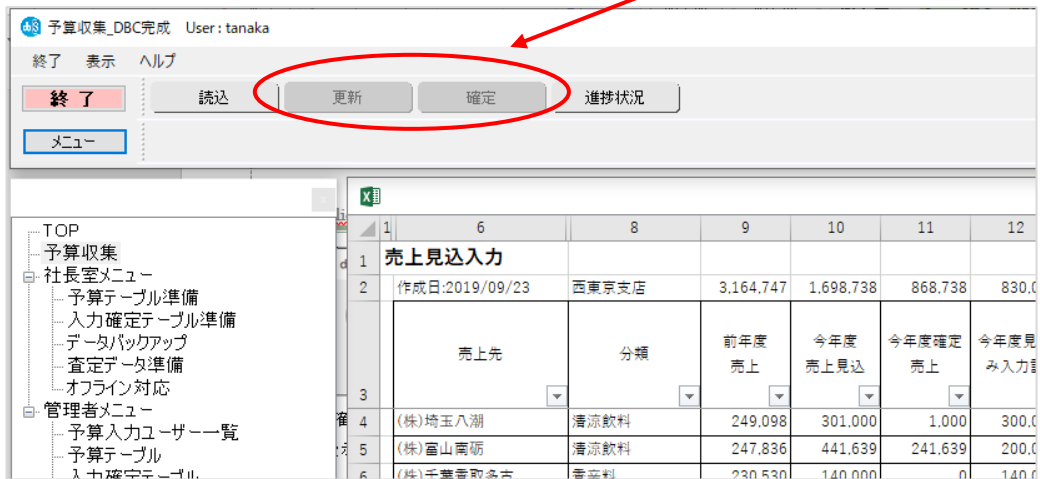




「確定」の最終確認で「はい」の選択で、確定します。確定を通知するメールを自動作成して表示します。
 (宛先はダミーであるため、このメールは破棄します。メール作成の機能紹介まで)



西東京支店のデータが「確定」したため、ボタン「更新」、「確定」が無効になります。

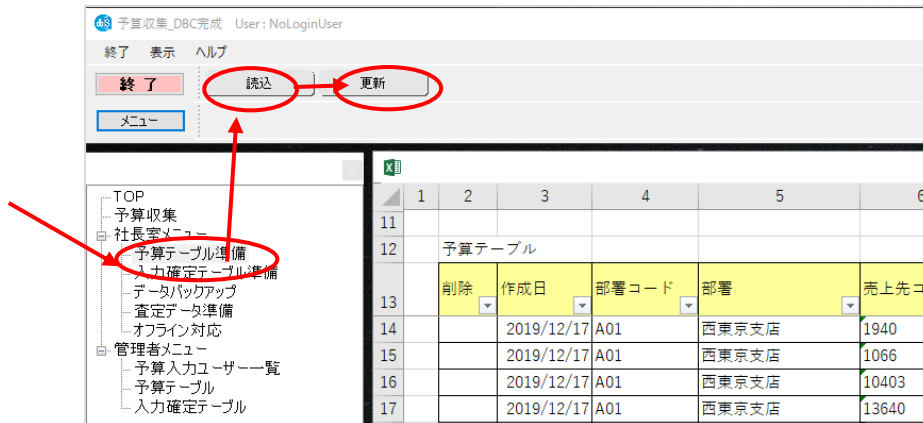


進捗状況では、西東京支店は確定状態になります。

作成日: 2019/09/23		前年度売上	今年度売上見込	確定日	確定者
部署コード	部署				
A01	西東京支店	3,164,747	1,698,738	2019/12/16	田中一郎
A02	東東京支店	1,621,946	461,333		
B01	千葉支店	3,228,480	2,338,230		

6.3.7 デバッグ実行：社長室メニュー：予算テーブル準備

売上した実績(「売上テーブル」)を元にして、新しく全支店分の「予算テーブル」データを作成します。
メニューから「予算テーブル準備」を選択し、「読込」で作成するデータを表示します。
(作成日は実行当日(例:実行日 2019/12/17)として作成します。)



「更新」で表示データを「予算テーブル」に追加します。

今回の「予算収集」から、最新の作成日(2019/12/17)のデータが編集対象となります。

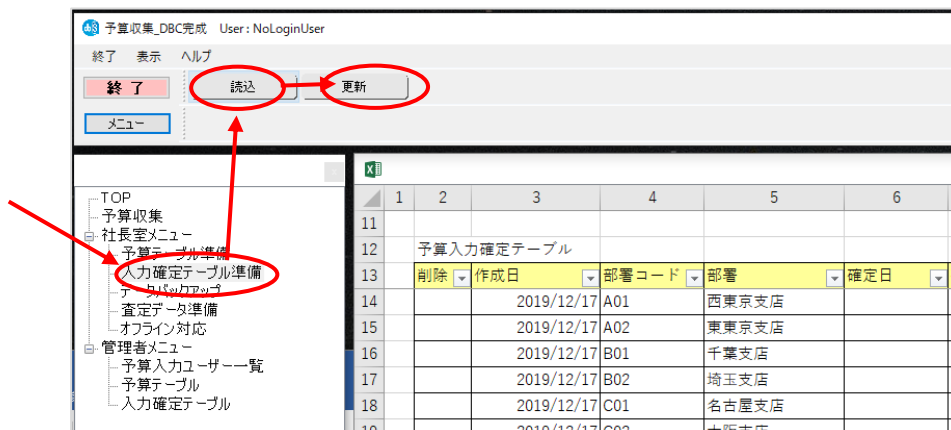
6.3.8 デバッグ実行：社長室メニュー：入力確定テーブル準備

「予算テーブル準備」を行ったのち、確定状況を管理する「予算入力確定テーブル」も作成日が当日のデータを追加作成します。

(「予算テーブル」と「予算入力確定テーブル」は同じ作成日データを持つことが本システムの仕様となっています。)

メニューから「入力確定テーブル準備」を選択し、「読込」で作成データを表示します。

(作成日は実行当日(例:2019/12/17)として作成します。)



「更新」で表示データを「予算入力確定テーブル」に追加します。

今回の「予算収集」の「進捗状況」から、最新の作成日(2019/12/17)のデータが表示対象となります。

6.3.9 デバッグ実行：社長室メニュー：データバックアップ

本システムで作成した全データ(「予算テーブル」、「予算入力確定テーブル」、「予算入力ユーザー一覧」)を Excel に書出します。

削除	作成日	部署コード	部署	売上先コード	売上先
	2019/09/15	A01	西東京支店	10072	(株)栃木
	2019/09/15	A01	西東京支店	10094	(株)大阪
	2019/09/15	A01	西東京支店	10125	(株)徳島
	2019/09/15	A01	西東京支店	10167	(株)広島

各シートに書出し。

6.3.10 デバッグ実行：社長室メニュー：査定データ準備

再度予算を練り直すため、既存の入力済の最新データを生かして新規に「予算テーブル」、「予算入力確定テーブル」のデータを作成します。

作成日は実行当日(例:2019/12/18)として「読込」で作成します。

注：既に同一の「作成日」データが予算テーブル等にある場合は、KEY が重なるため「更新」で追加できません。(「予算テーブル準備」、「入力確定テーブル準備」を本日も行っている場合など)

動作確認として利用する場合は既存の該当作成日のデータ削除をお願いします。

削除	作成日	部署コード	部署
	2019/12/18	A02	東東京支店
	2019/12/18	C01	名古屋支店
	2019/12/18	C01	名古屋支店
	2019/12/18	A02	東東京支店

「更新」で「予算テーブル」、「予算入力確定テーブル」に追加します。

次回の「予算収集」の対象は、最新の作成日(2019/12/18)のデータが表示対象となります。

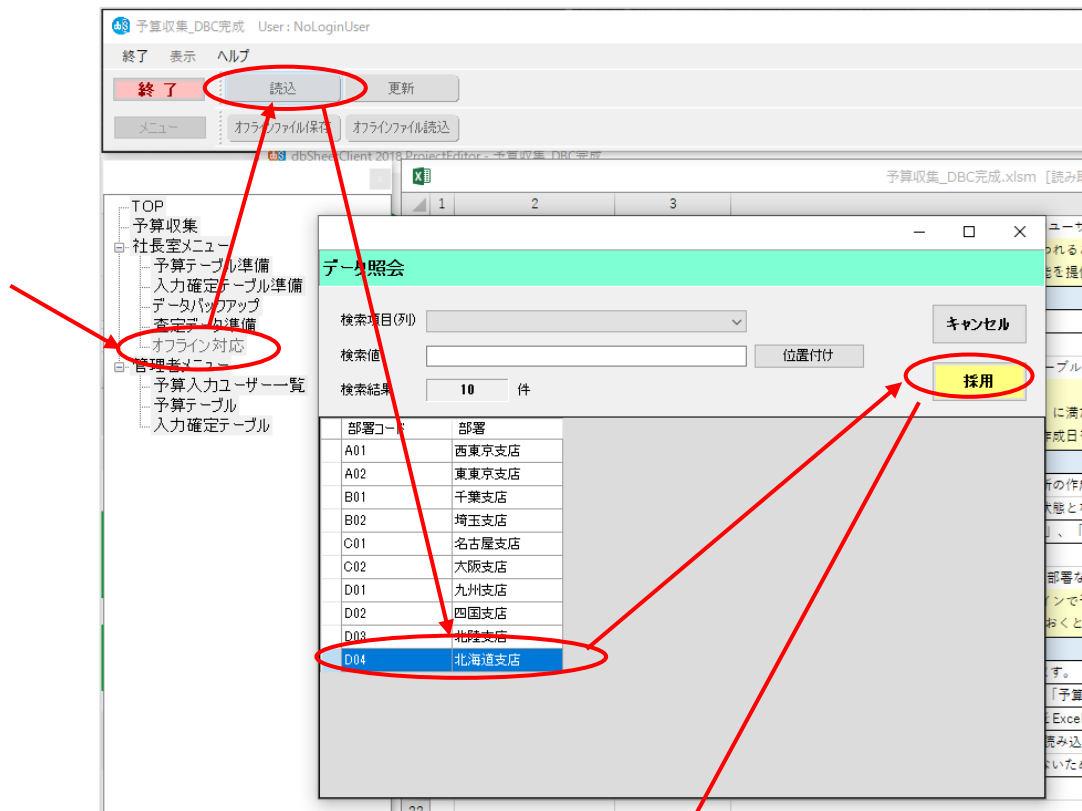
6.3.11 デバッグ実行：社長室メニュー：オフライン対応

本システムを利用できない部署などに対応するため、Excel ファイルを介して予算収集をする機能です。

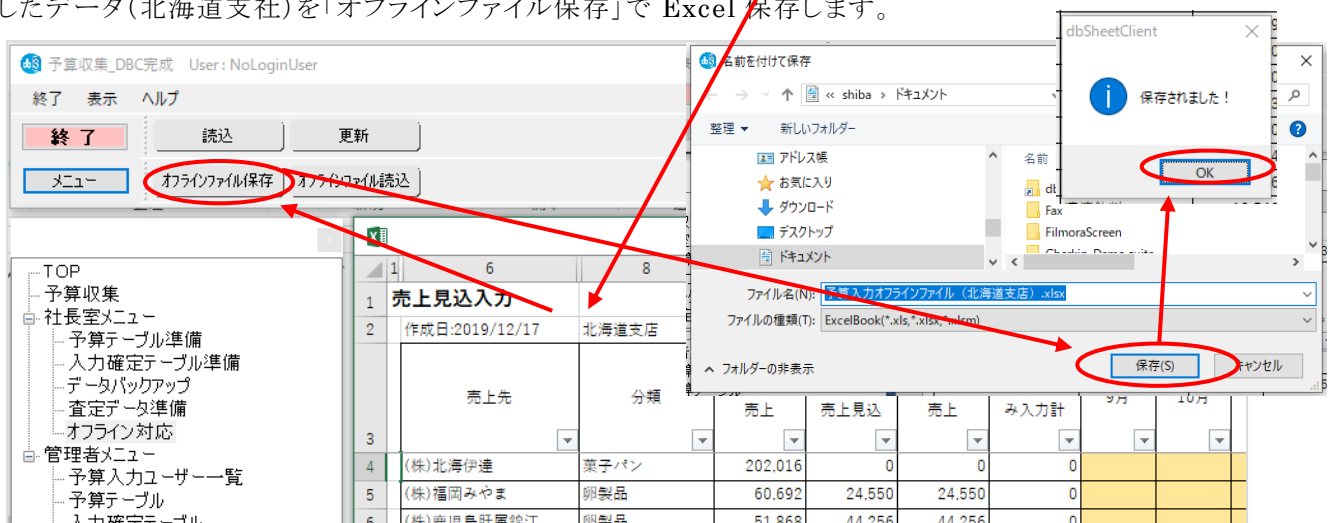
例) 北海道支社の対応する場合。

部署コード D04(北海道支社)を読み込めるのは、「担当部署コード」:all (全部署担当)です。「デバッグ実行設定」での設定は行わず、User:NoLoginUser を利用します。

複数部署を担当する User は「読み込」時に「データ照会」のウィンドウから、対象部署を選択します。



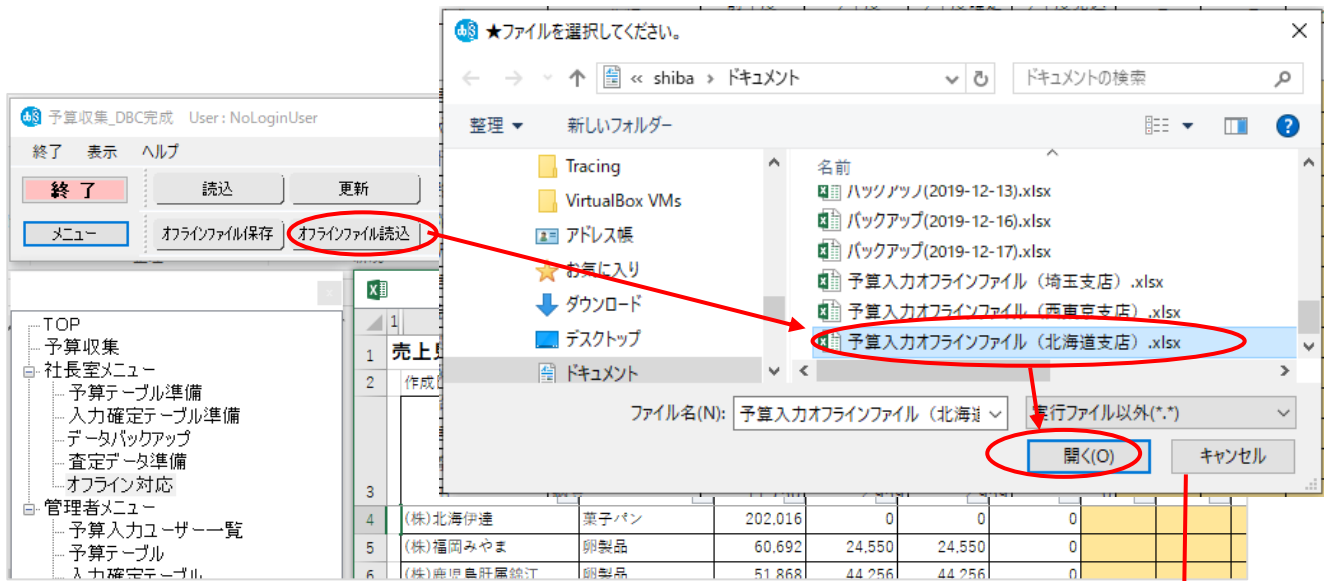
読み込んだデータ(北海道支社)を「オフラインファイル保存」で Excel 保存します。



オフライン保存した Excel ファイルを編集し保存します。(9月、10月の 2 行のデータを追加)

売上見込入力								
作成日:2019/12/17		北海道支店	594,849	205,273	111,633	93,640	35,740	57,900
売上先	分類	前年度売上	今年度売上見込	今年度確定売上	今年度見込み入力計	9月	10月	
(株)北海伊達	菓子パン	202,016	35,740	0	35,740	12,340	23,400	
(株)福岡みやま	卵製品	60,692	82,450	24,550	57,900	23,400	34,500	
(株)鹿児島肝属錦江	卵製品	51,868	44,256	44,256	0			
(株)兵庫栄	菓子パン	20,800	0	0	0			

「オフラインファイル読込」で編集したファイルを読込します。



先ほど編集したデータが画面に反映されます。

「更新」で表示データを DB に更新します。



6.3.12 デバッグ実行：管理者メニュー

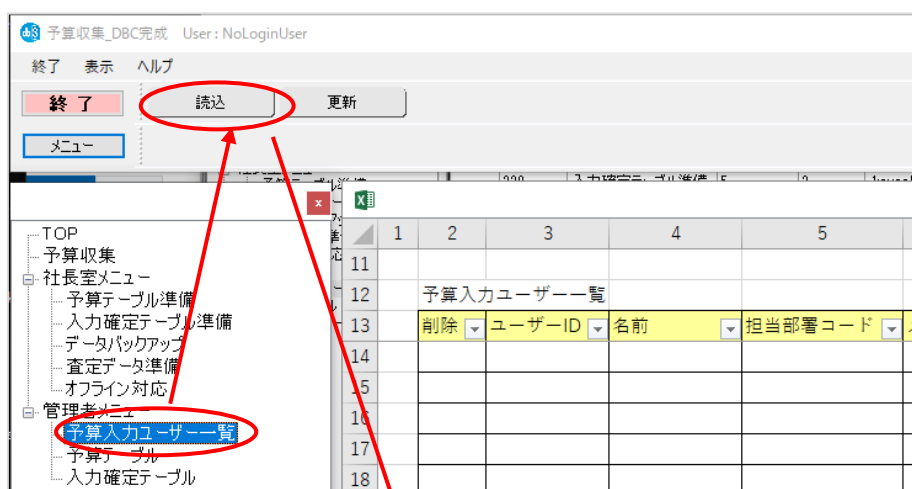
DB の各テーブルで管理している全データを表示、編集します。

管理者メニュー内の各テーブル名のボタン「読み」、「更新」操作は、全て同じです。

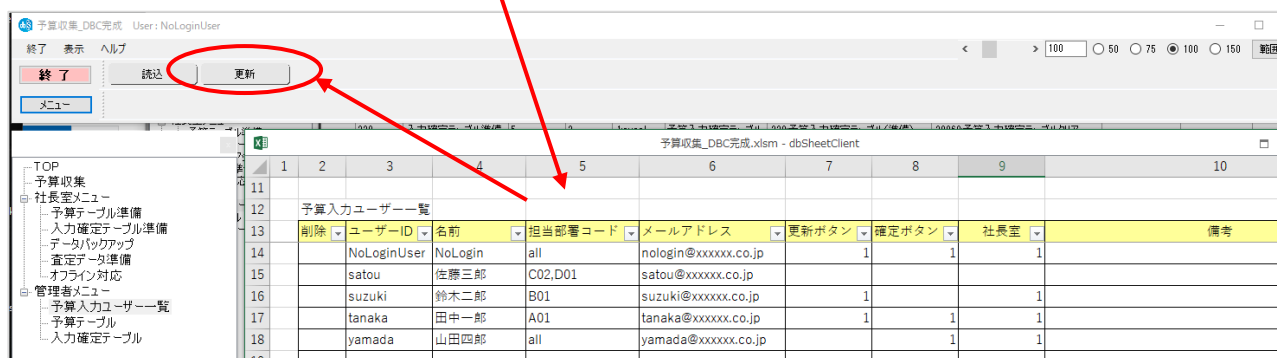
例)「予算入力ユーザー一覧」

メニュー:管理者メニュー>予算入力ユーザー一覧。

初期表示ではデータ表示はありません。



「読み」後、データを編集して「更新」します。



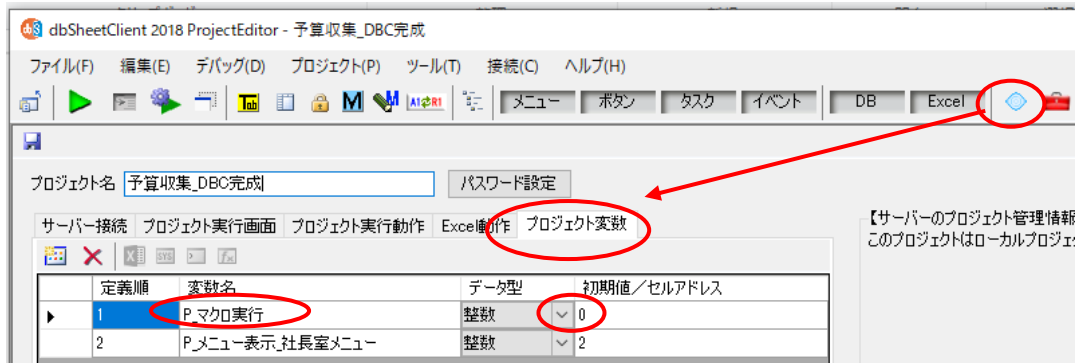
6.3.13 マクロ利用とタスク利用

本プロジェクトでは、初期設定の状態では Excel マクロを利用していません。

マクロに相当する機能を dbSheetClient の「タスク」で実装しています。

プロジェクトの設定状態を変更することで、「タスク」で実装した一部の機能を「マクロ」で実行することができます。(見た目の動作は変わりませんが、タスクではなくマクロで実行することになります。)

変更する設定:

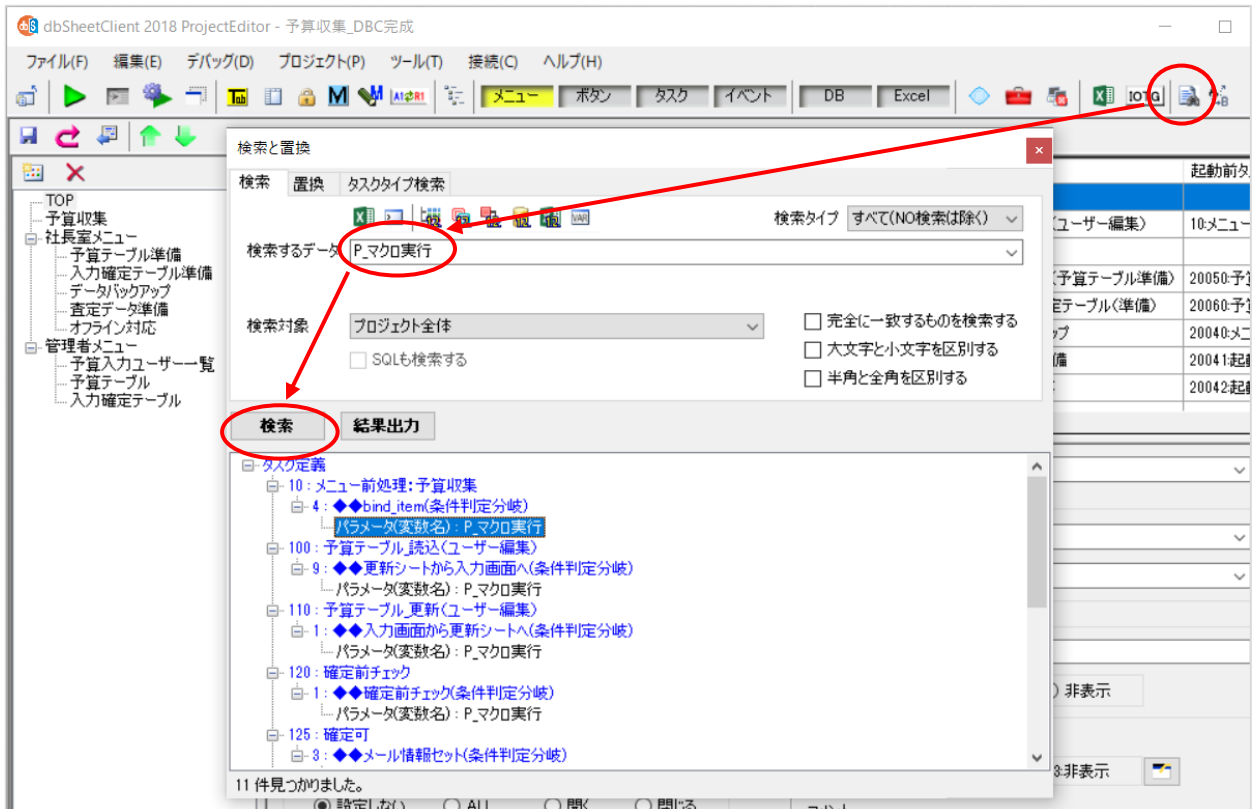


変数「P_マクロ実行」の初期値の設定。

0:タスク実行

1:マクロ実行

どのタスクの処理がマクロ実行になるかは、以下で検索ができます。

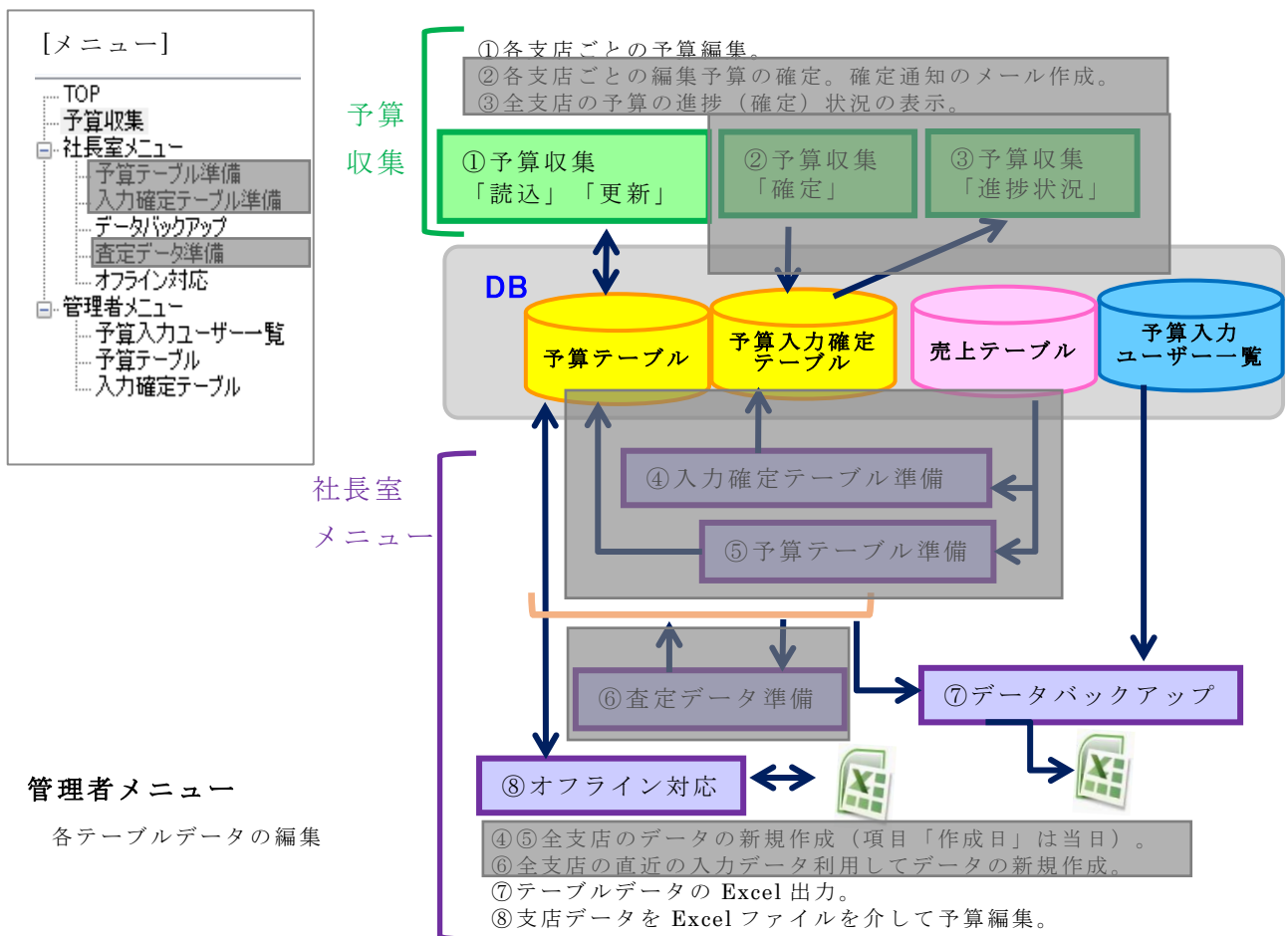


6.4 プロジェクトを作成しましょう

前述のプロジェクト[予算収集_完成]の一部の機能(以下)を作成します。

- ・管理者メニュー(予算入力ユーザー一覧、予算テーブル、入力確定テーブル)
- ・①予算収集(「読込」、「更新」)
- ・社長室メニュー(⑦データバックアップ、⑧オフライン対応)

以下のグレーで塗りつぶした機能は作成しません。



まず、動作環境の構築から始めます。

6.4.1 動作環境 : DB「DBC 作成」の構築

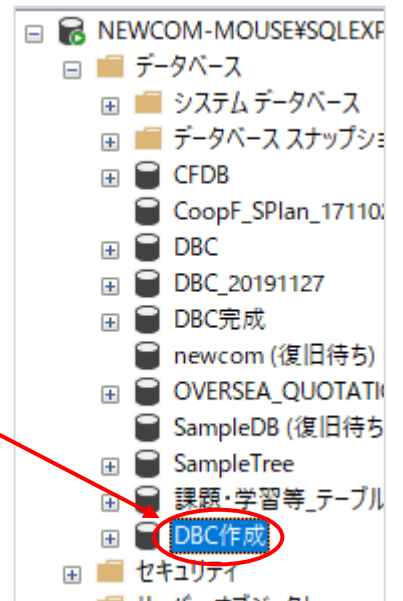
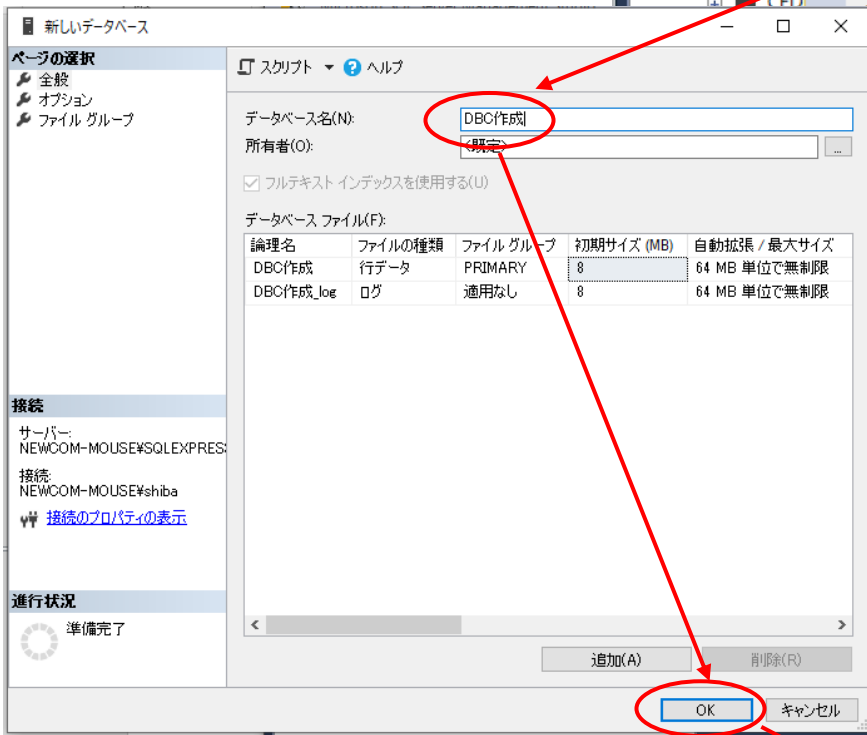
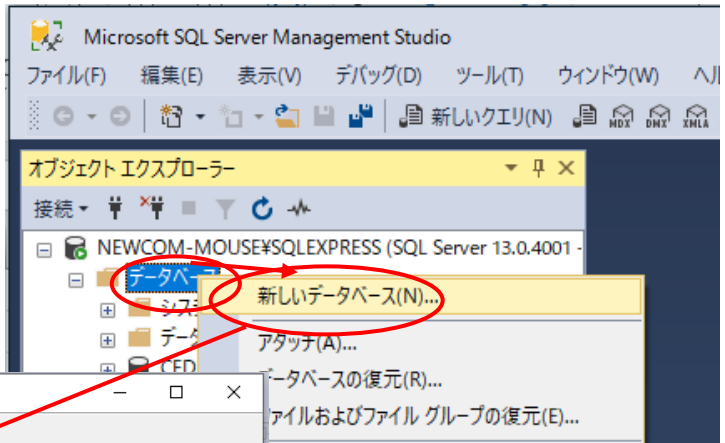
利用するデータベース[DBC 作成]を以下の手順で作成します。

SQL Server Management Studio を起動します。

「データベース」を選択して、
右クリックします。

「新しいデータベース」を選択します。

「DBC 作成」を作成します。



データベース「DBC 作成」が作成されました。

6.4.2 動作環境 : プロジェクト「予算収集」の新規作成

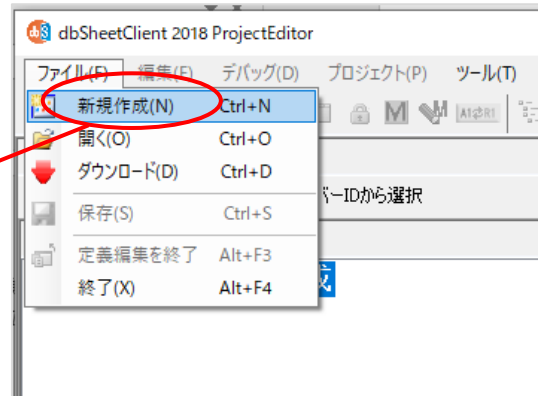
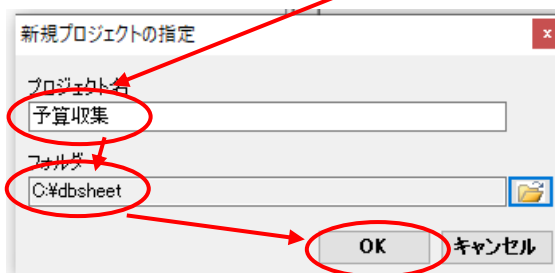
プロジェクト[予算収集]を以下の手順で新規に作成します。

dbSheetClient 開発版を起動します。

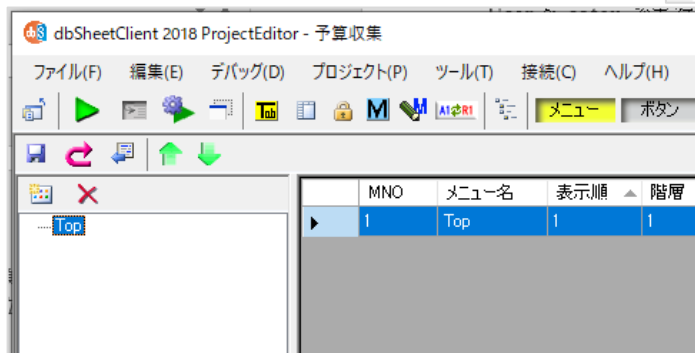
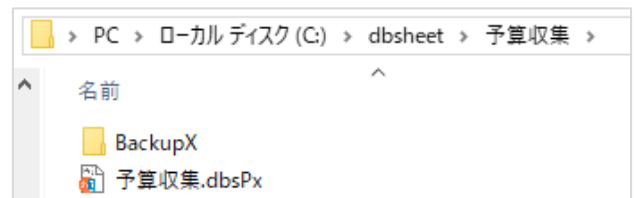
「新規作成」を選択し、

プロジェクト名「予算収集」、

フォルダ「C:\dbsheet」とし、OK します。

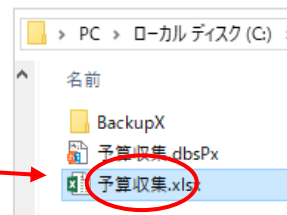
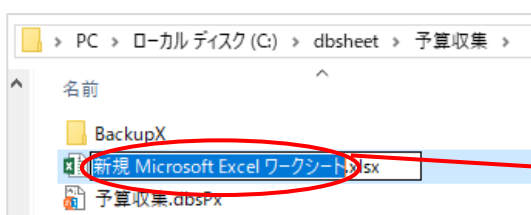


プロジェクトファイルが指定のフォルダに作成されます。



6.4.3 動作環境 : Excel「予算収集.xlsx」の新規作成と設定

利用する Excel を「予算収集.xlsx」として新規作成します。(プロジェクトと同じフォルダ)



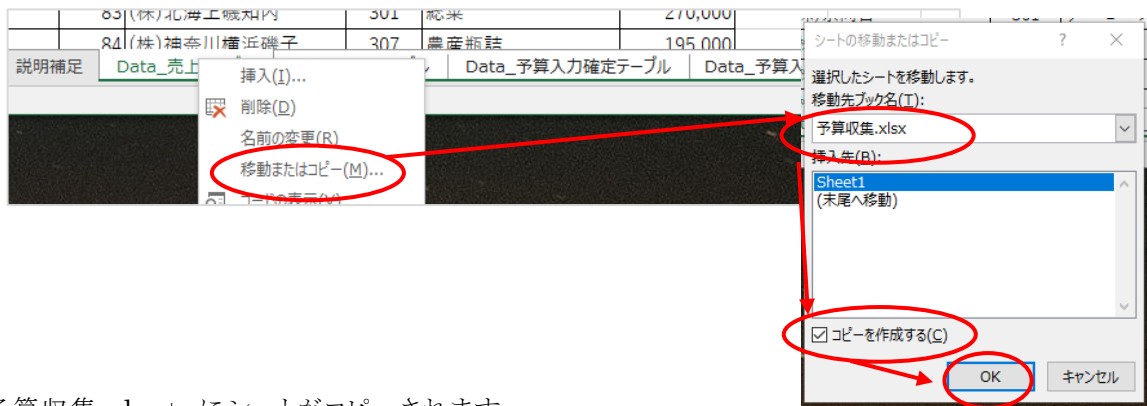
データベース「DBC 作成」にテーブルを新規作成する前準備として、「予算収集.xlsx」に作成するテーブルのデータシートコピーをします。

コピー元シート: 「6.3 動かしてみよう」で利用した「予算収集_DBC 完成.xlsx」から以下の 4 シートをコピーします。

- 「Data_売上テーブル」、「Data_予算テーブル」、
- 「Data_予算入力確定テーブル」、「Data_予算入力ユーザー一覧」

「予算収集.xlsx」、「予算収集_DBC 完成.xlsx」の2ファイルを開きます。

「予算収集_DBC 完成.xlsx」にて、4 シートを選択し右クリックし、「移動またはコピー」を選択。



「予算収集.xlsx」にシートがコピーされます。

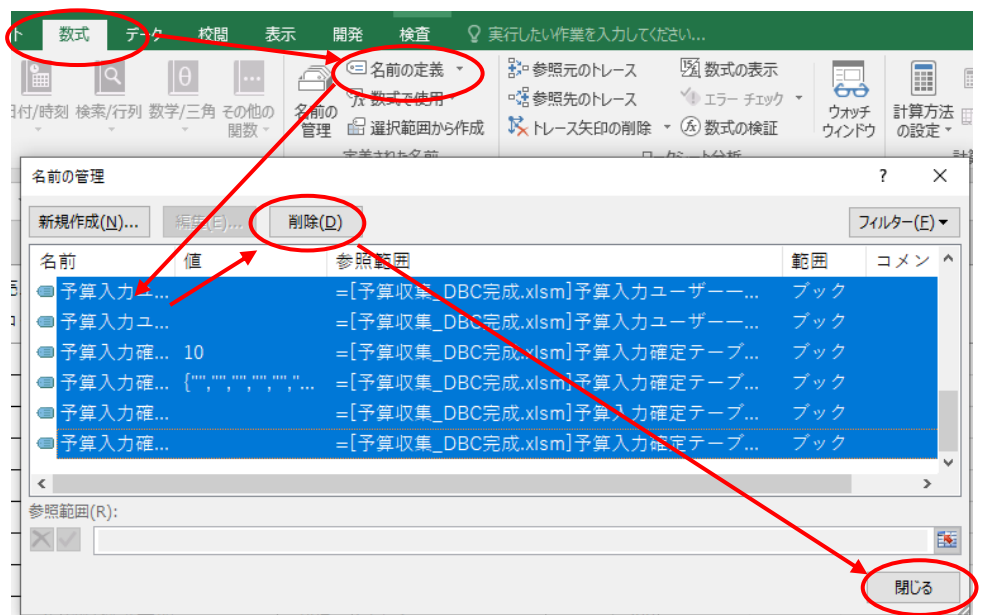
4	C01	名古屋支店	103	(株)新潟系魚川	700	食パン	2,269,531
Data_売上テーブル Data_予算テーブル Data_予算入力確定テーブル Data_予算入力ユーザー一覧 Sheet1							

このコピーで不要なセルの名前定義がコピーされているので、以下で全ての名前定義を削除します。

予算収集.xlsx

全ての項目を選択して「削除」します。

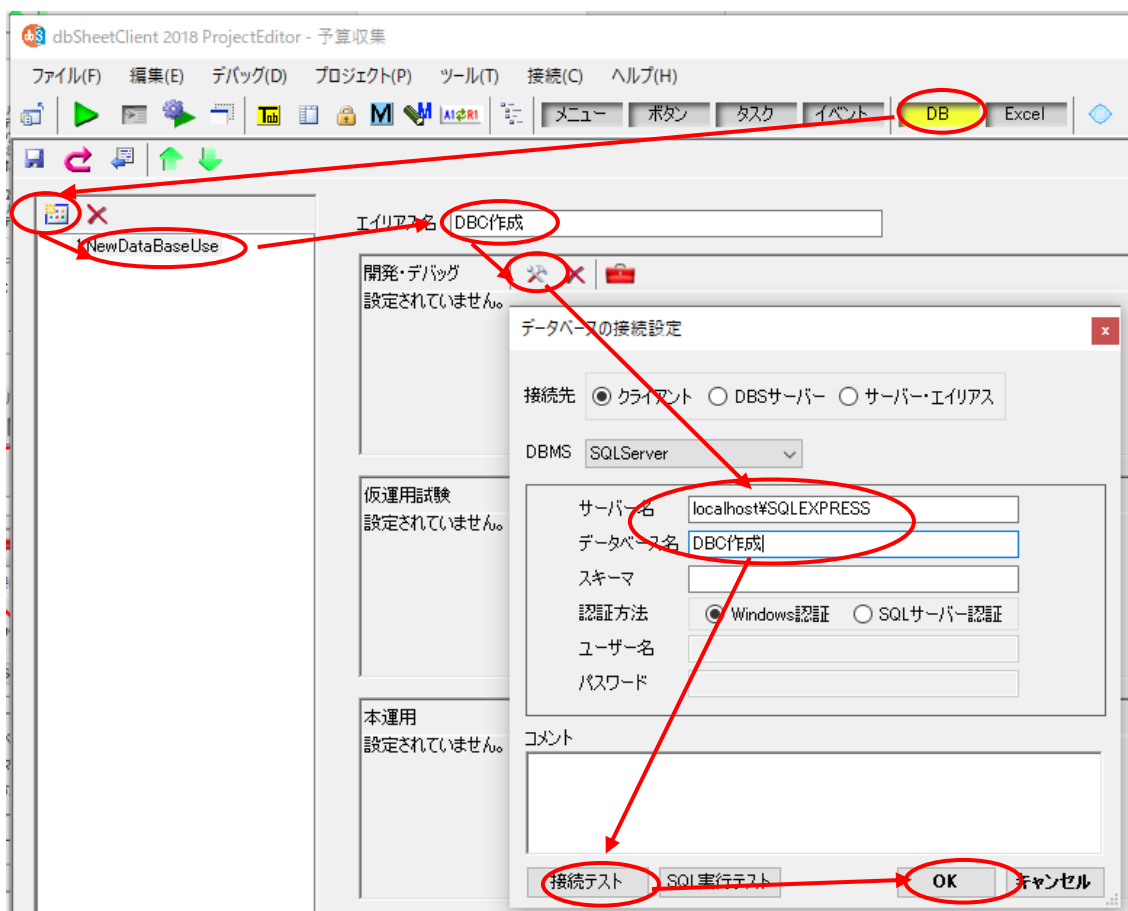
削除後、ファイルを保存して閉じます。



6.4.4 動作環境 : プロジェクトに DB、Excel を定義

プロジェクトにDB(エイリアス名 : DBC 作成)を定義します。

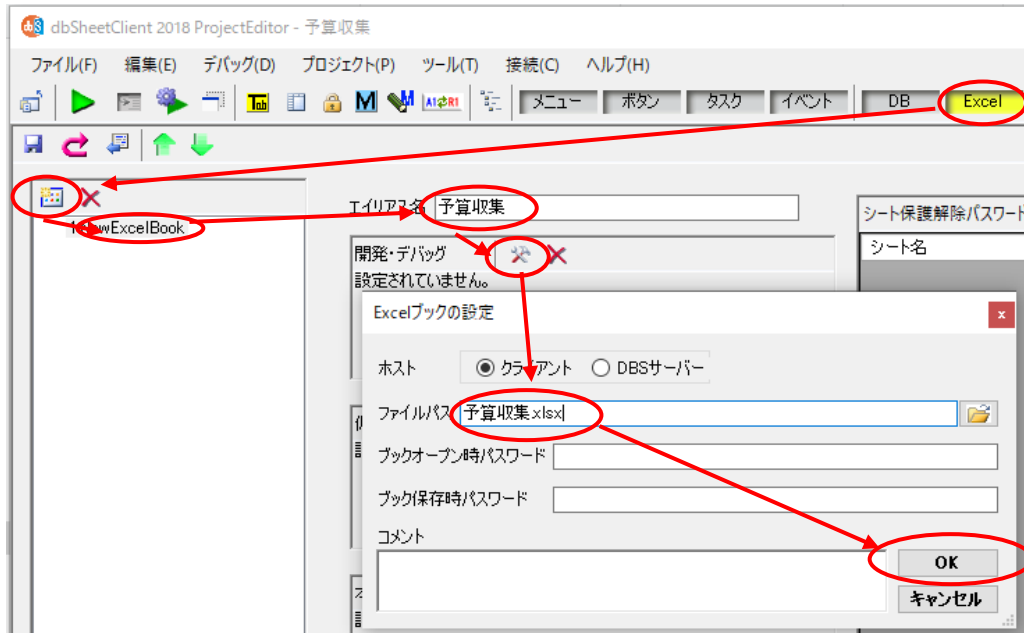
- ・サーバー名は SQL Server Management Studio で確認してください。
- ・データベース名は、前の章で作成した「DBC 作成」とします。



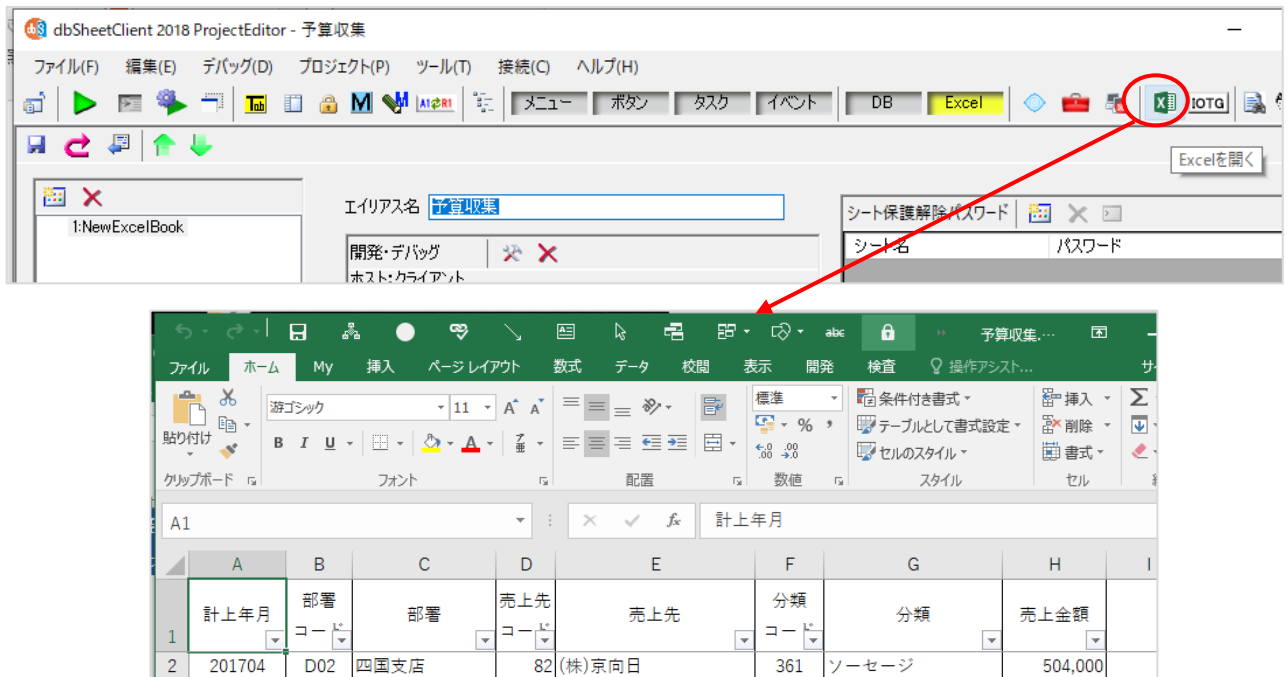
「接続テスト」で接続を確認して OK します。

プロジェクトにExcel(エイリアス名: 予算収集)を定義します。

- ・ファイルパスでフォルダ指定をしない場合は、プロジェクトファイルと同じ場所となります。



正しく定義できたかを以下で確認します。

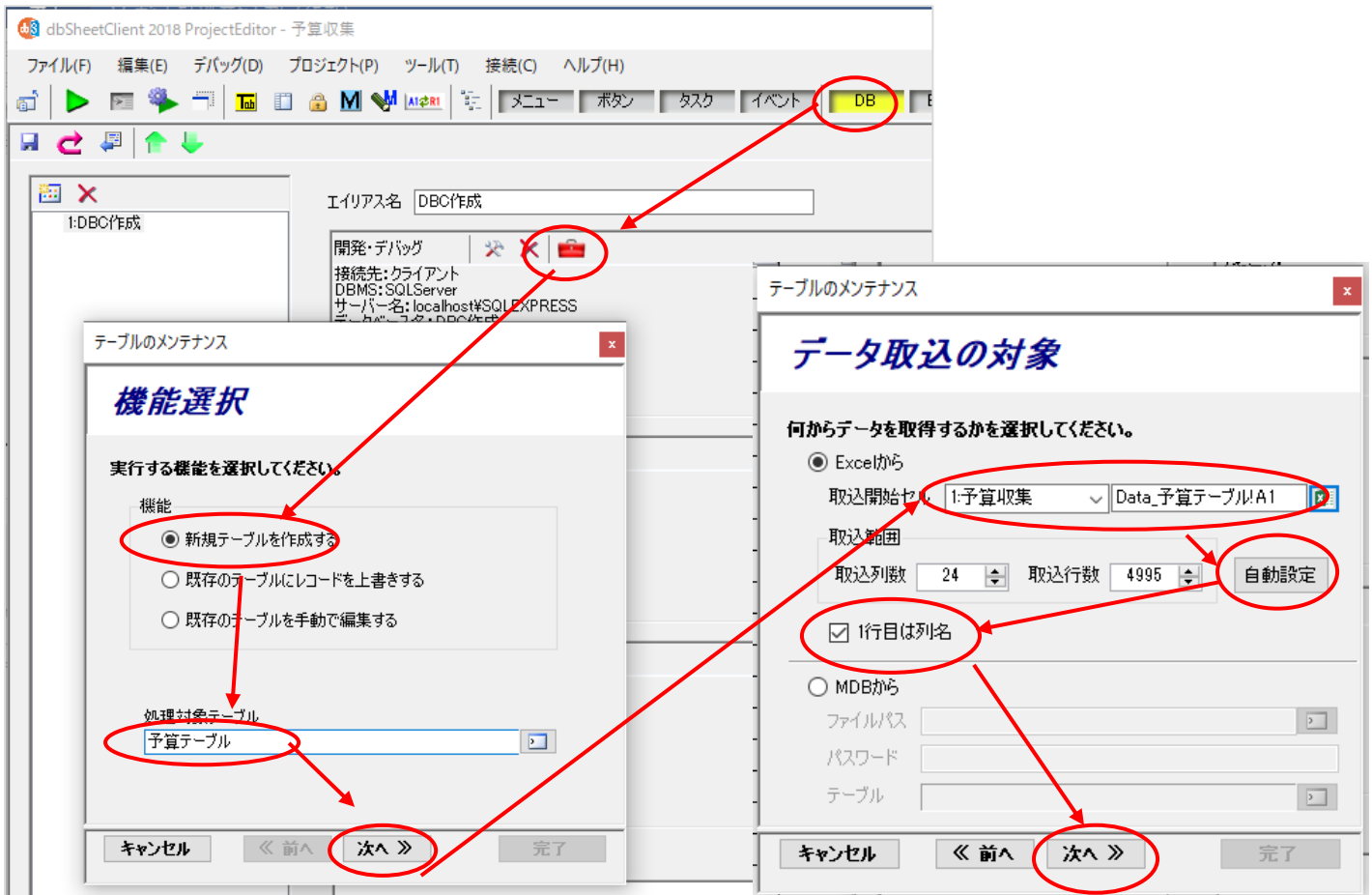


Excel を開くことができれば OK です。(確認後、Excel は閉じます)

6.4.5 動作環境 : DB ツールでテーブル作成

dbSheetClient 開発版の機能「DB ツール」を利用して、各テーブル、テーブル内データを作成します。

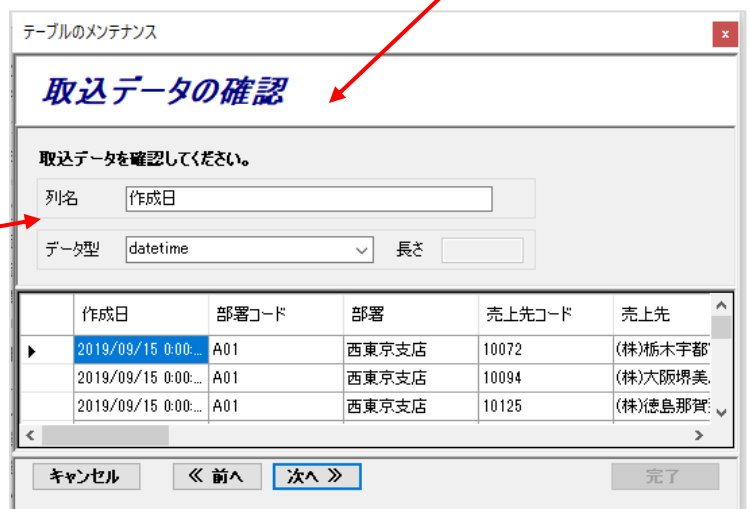
「予算テーブル」をまず作成します。



指定した Excel シートのデータから「列名」、「データ型」を自動認識してテーブルの項目定義を作成します。

「列名」、「データ型」は変更が可能です。

予算テーブルの場合、列名「備考」のデータ型が nvarchar(10)でサイズが小さいため、nvarchar(200)に変えます。



列名「備考」を選択するために
「備考」の項目の 1 行目を選択
します。

テーブルのメンテナンス

取込データの確認

取込データを確認してください。

列名

データ型 長さ

11月	12月	1月	2月	3月	備考

キャンセル << 前へ 次へ >> 完了

nvarchar(10) を (200) に変えます。

テーブルのメンテナンス

取込データの確認

取込データを確認してください。

列名

データ型 長さ

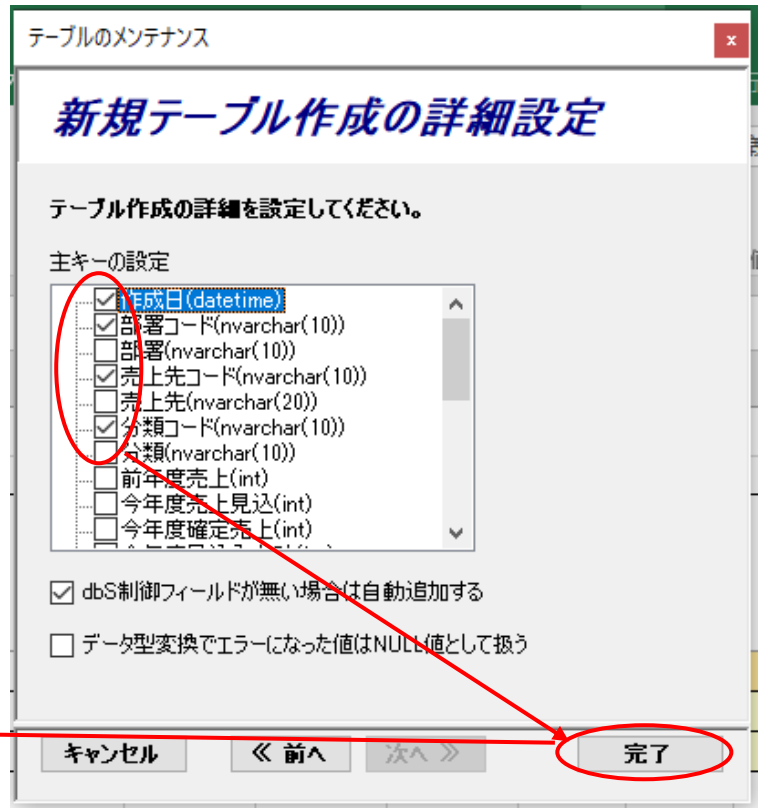
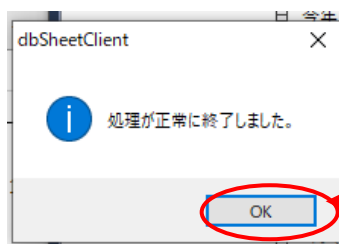
11月	12月	1月	2月	3月	備考

キャンセル << 前へ 次へ >> 完了

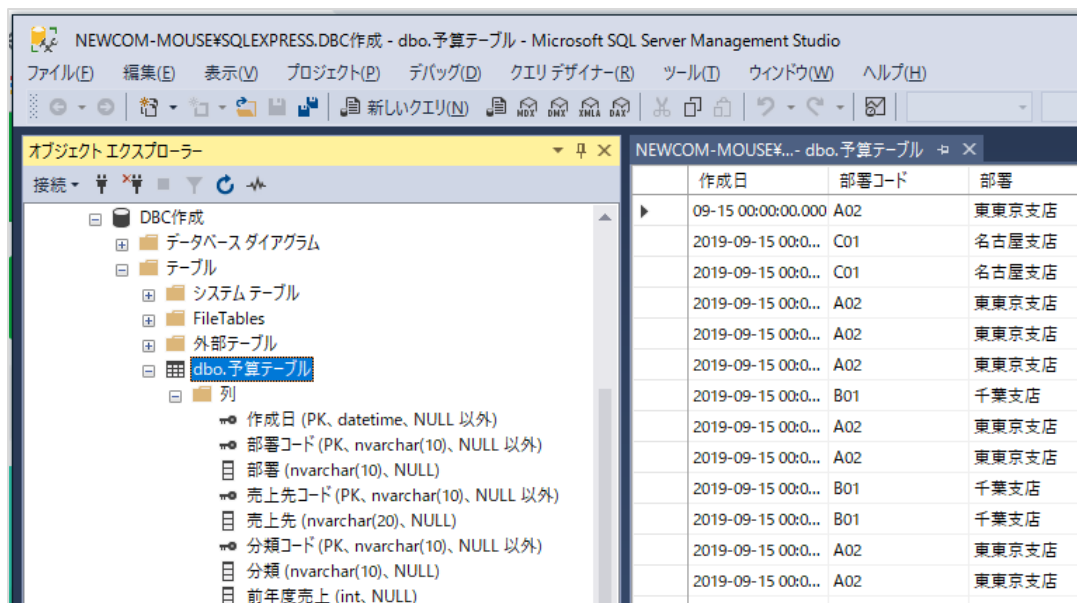
この画面は、項目定義の最終確認の画面です。

予算テーブルの以下の 4 項目にチェックを入れて主キーを設定します。

作成日、部署コード、
売上先コード、分類コード



SQL Server Management Studio で「予算テーブル」が作成され、データ設定していることを確認します。



他のテーブル(予算入力確定テーブル、予算入力ユーザー一覧、売上テーブル)も同様の手順でテーブル作成、データ設定を行います。

作成する各テーブルで利用するシート、利用する主キーは以下の通りです。

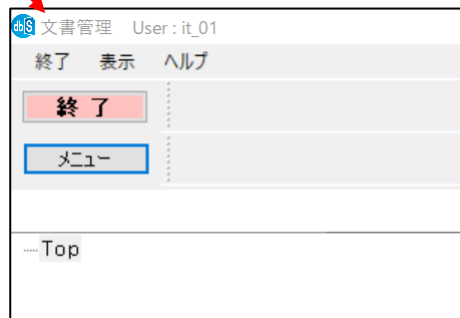
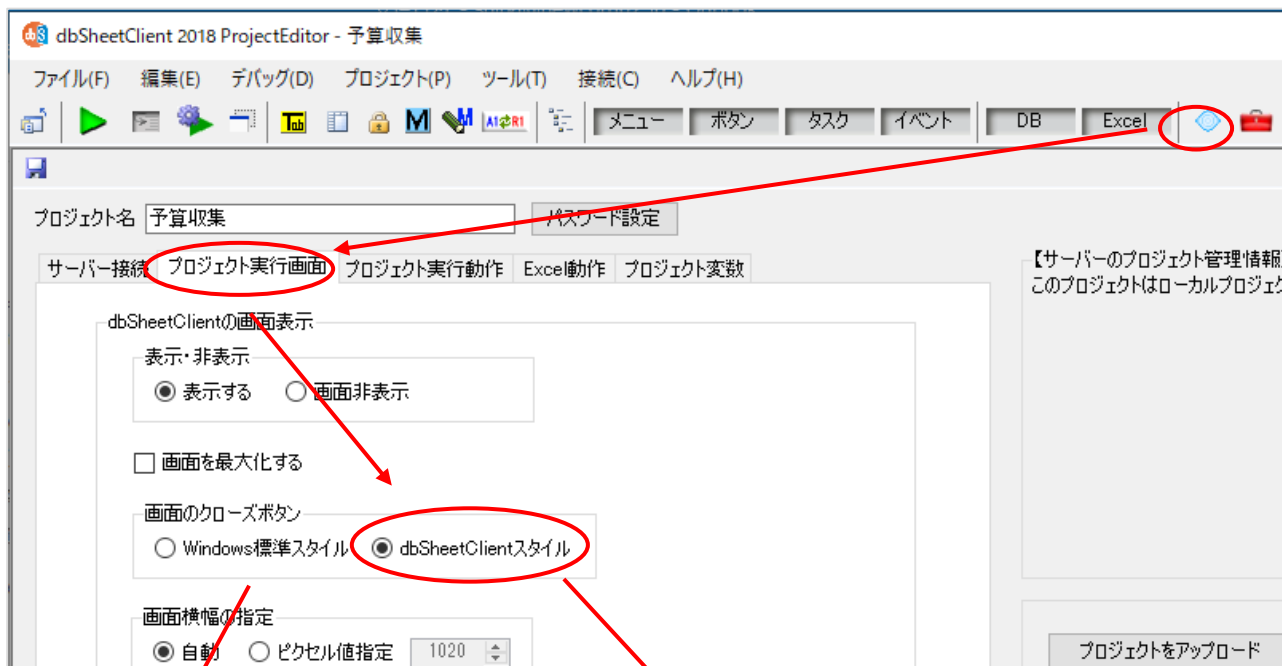
項番	テーブル名	利用する Excel シート	主キー	変更する項目
①	予算テーブル	Data_予算テーブル	作成日 部署コード 売上先コード 分類コード	「備考」 nvarchar(200)
②	予算入力確定テーブル	Data_予算入力確定テーブル	作成日 部署コード	(無し)
③	予算入力ユーザー一覧	Data_予算入力ユーザー一覧	ユーザーID	(無し)
④	売上テーブル	Data_売上テーブル	計上年月 部署コード 売上先コード 分類コード	(無し)

上記の①「予算テーブル」は作成しましたので、同様の手順で②、③、④を作成します。
これら全4テーブルは作成しないと、以降のプロジェクト作成ができません。

6.4.6 動作環境：画面のクローズボタン設定

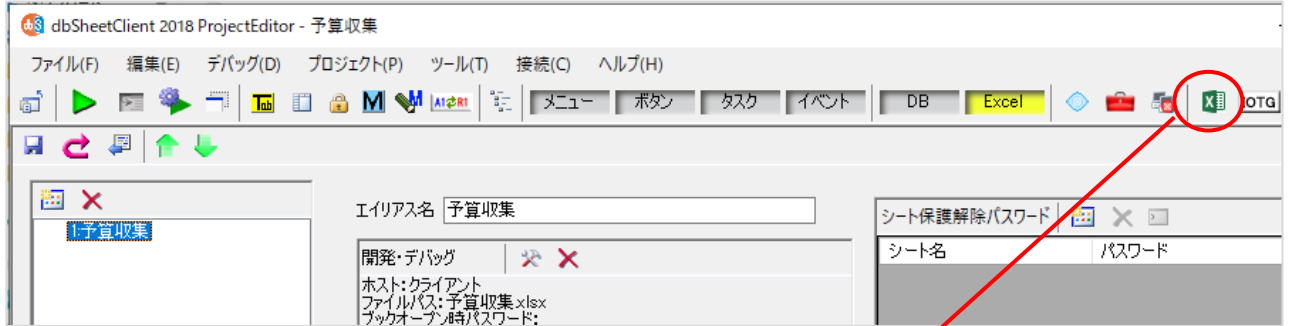
プロジェクト起動時の画面設定を次の 2 種類から選択します。(初期設定は「Windows 標準スタイル」)

※本書では dbSheetClient スタイルを採用

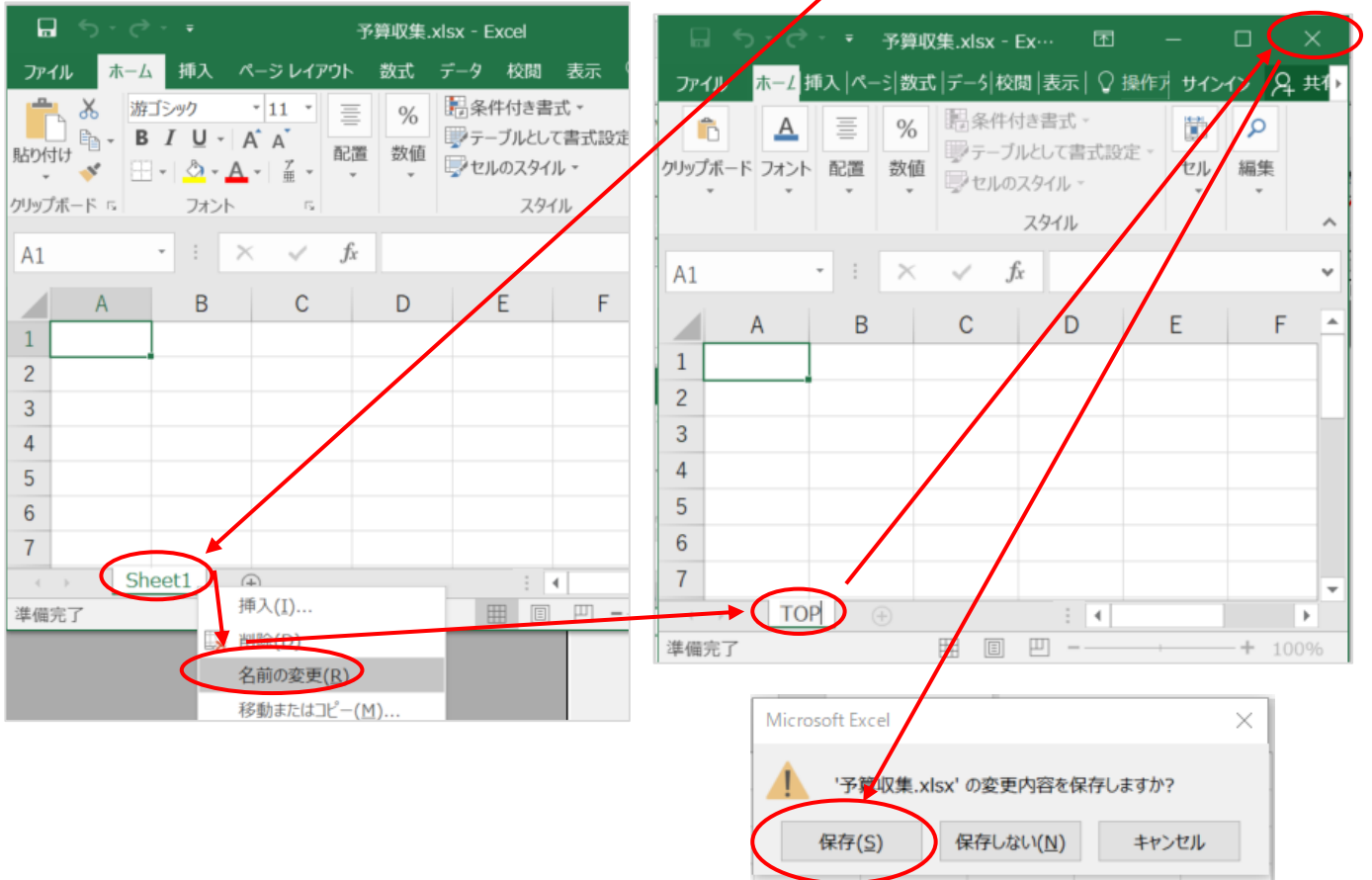


6.4.7 動作環境：メニュー：TOP の設定

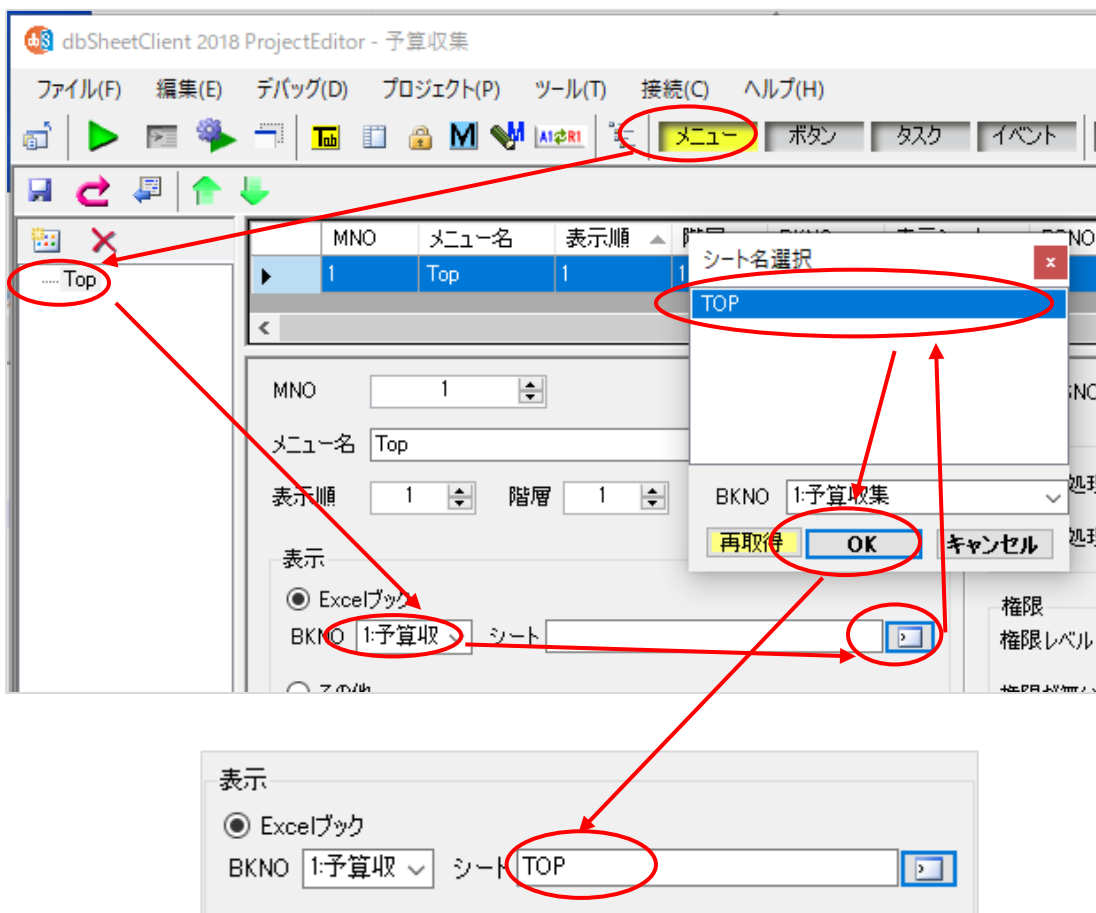
予算収集.xlsx の シート「Sheet1」の名前を「TOP」に変更します。



名前を変更して、「保存」してから閉じます。

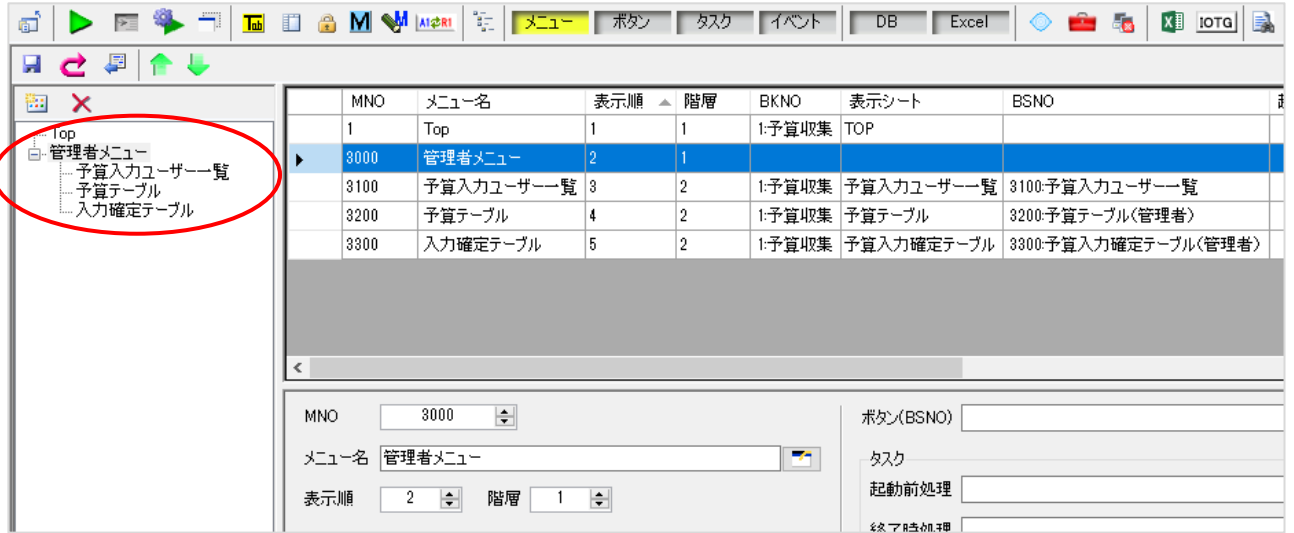


初期表示、またはメニュー「TOP」を選択時にシート「TOP」を表示する設定をします。

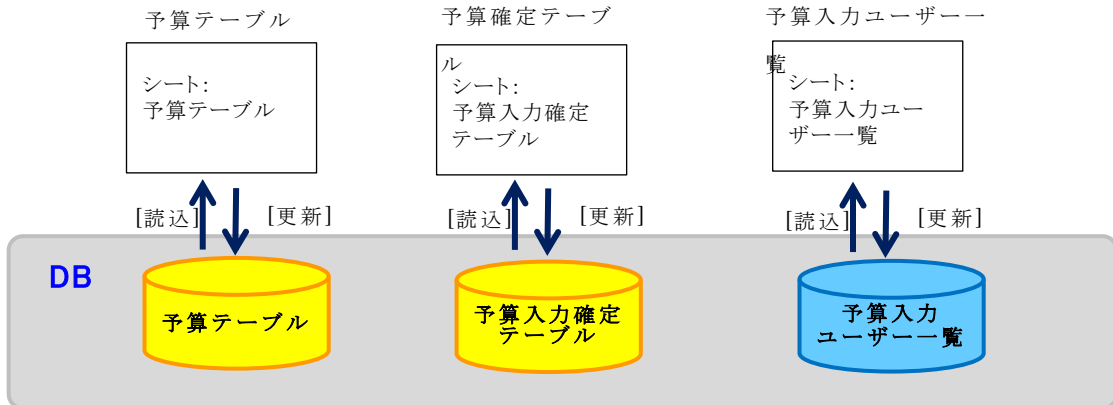


6.5 プロジェクト作成：管理者メニュー

管理者メニュー（予算入力ユーザー一覧、予算テーブル、入力確定テーブル）の機能を作成します。
作成後のメニューは以下の通りとなります。



作成する処理の概要は以下です。



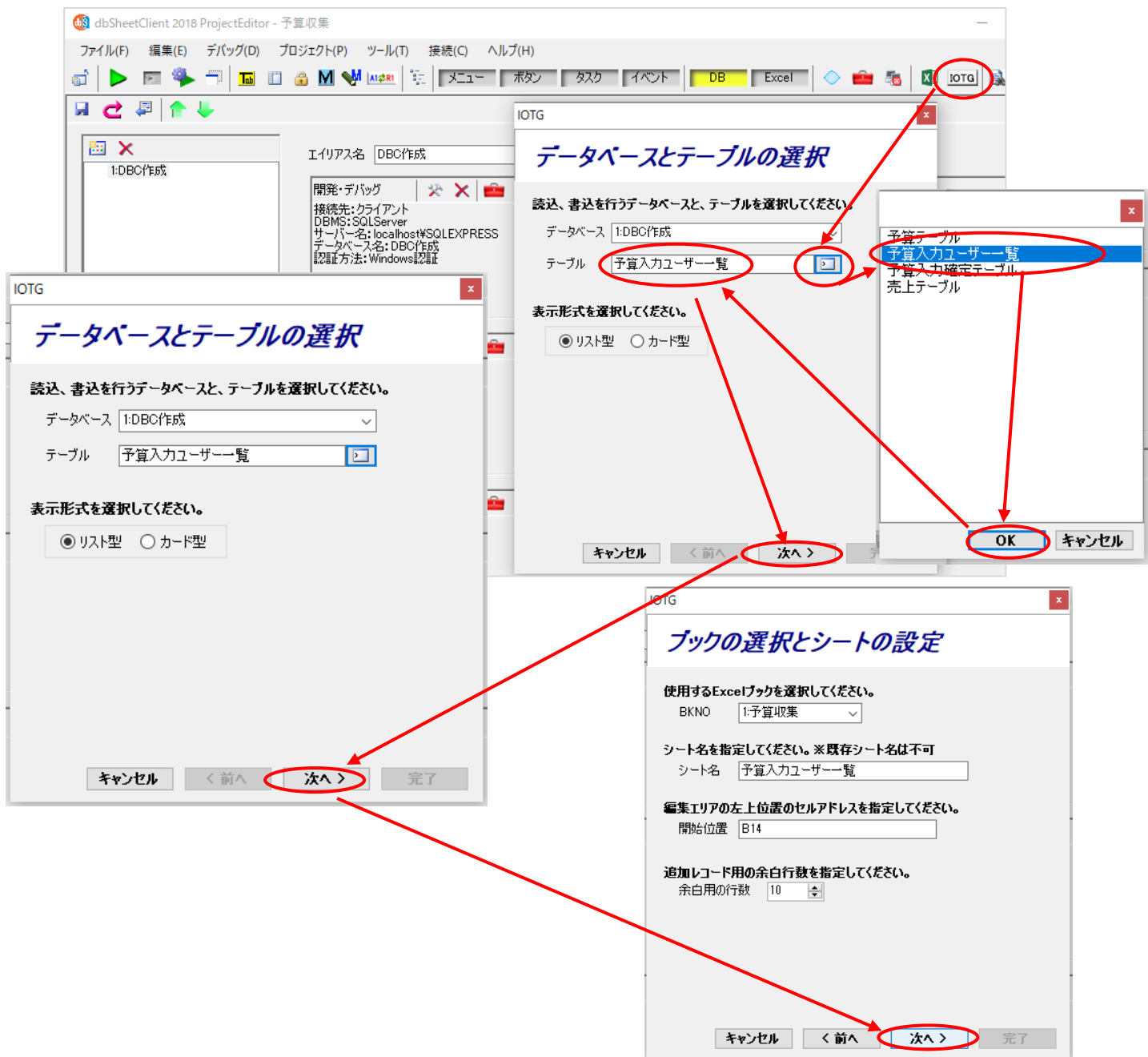
6.5.1 予算入力ユーザー一覧 (IOTG 利用)

dbSheetClient 開発版の IOTG(Input Output Task Generator)機能を利用します。

- ・IOTG : DB テーブルを参照して、プロジェクトにテーブルデータを参照、更新するメニュー、ボタン、タスクを自動生成する機能。

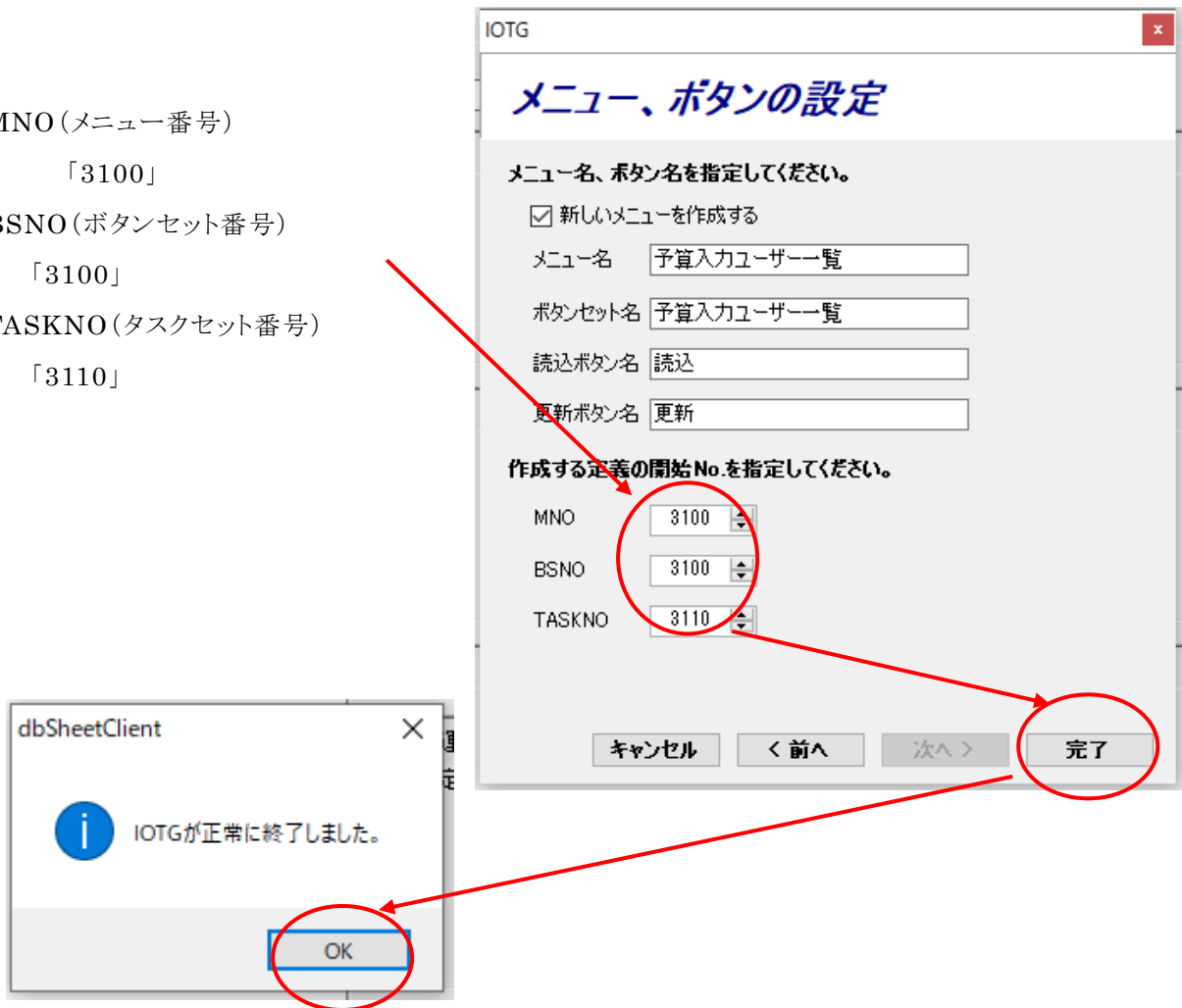
前の章で DB テーブル「予算入力ユーザー一覧」が作成済であることが前提です。

作成していない場合は、「6.4.5. 動作環境:DB ツールでテーブル作成」を参照して作成します。

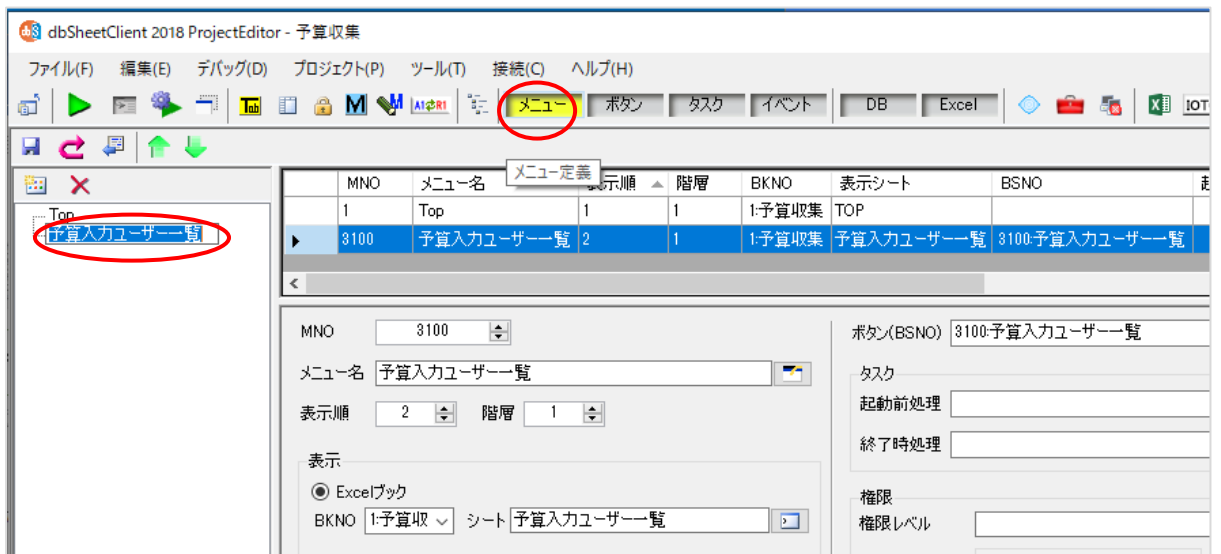


既に完成しているプロジェクト「予算収集_完成」の構成(全体整理した構成)に合わせるため、以下のように設定を変更します。

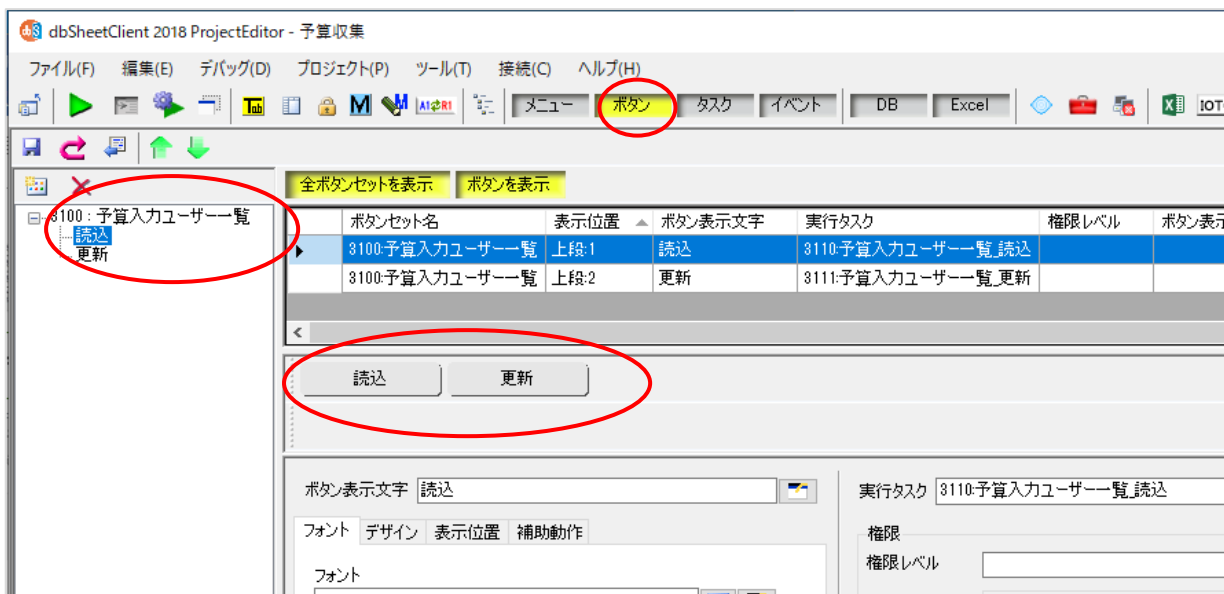
- ・MNO (メニュー番号)
「3100」
- ・BSNO (ボタンセット番号)
「3100」
- ・TASKNO (タスクセット番号)
「3110」



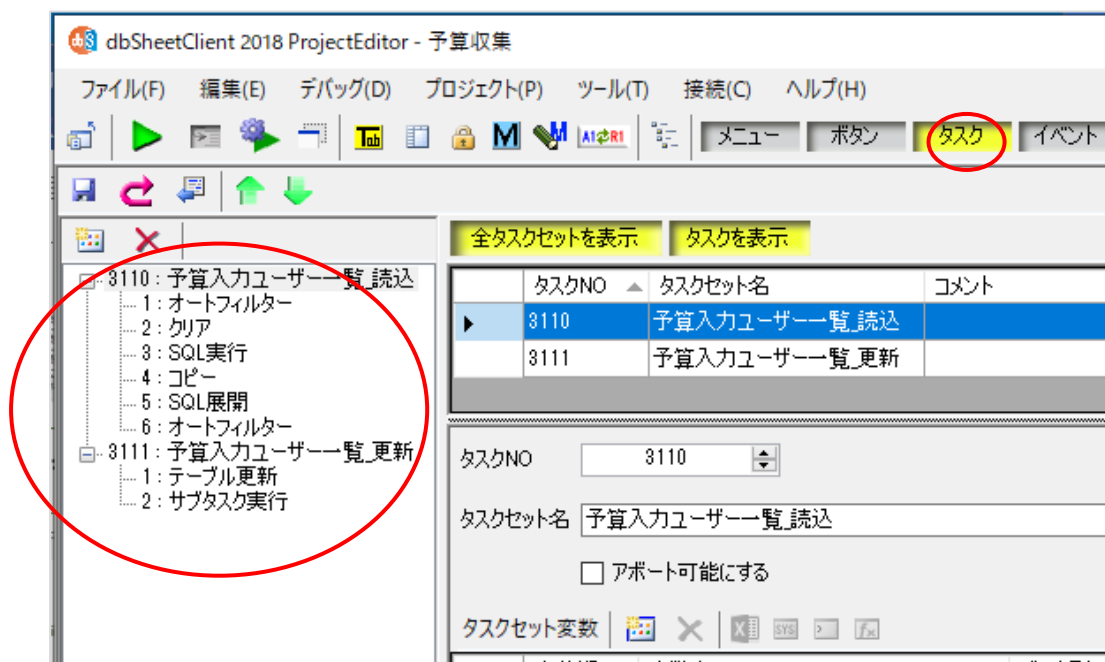
作成された「メニュー」は以下です。



作成された「ボタン」は以下です。

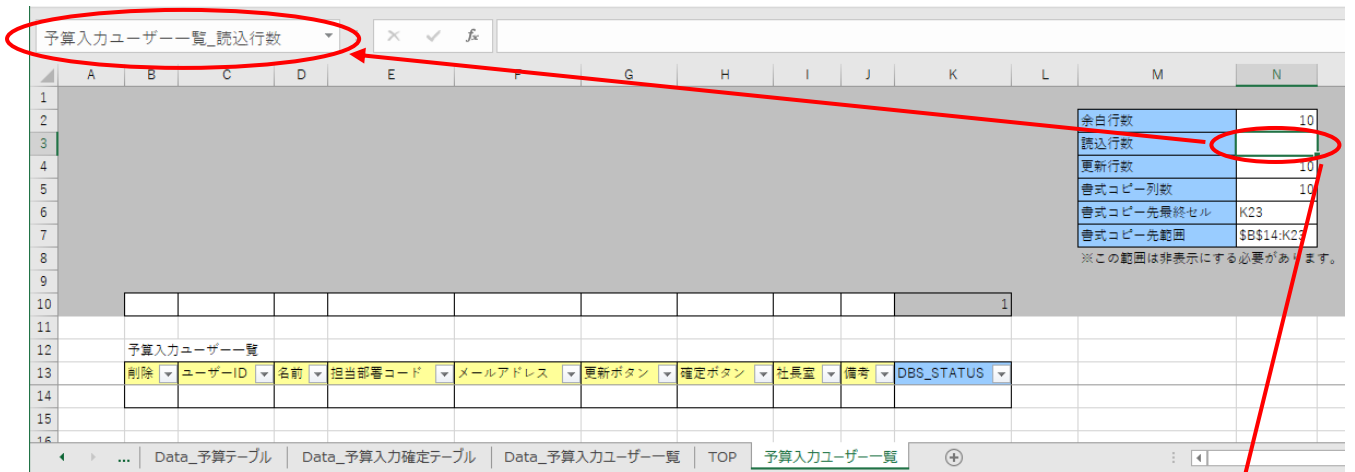


作成された「タスク」は以下です。



作成された Excel シート「予算入力ユーザー一覧」は以下です。

(赤線は定義追加した名前付きセル(6個)の中の1例:「予算入力ユーザー一覧_読込行数」です。)

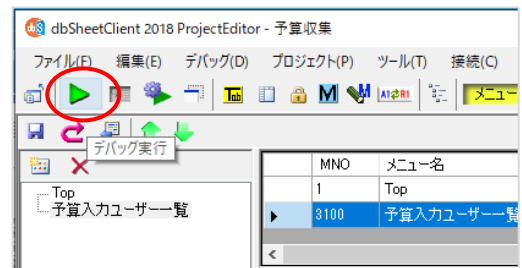


上記シートには、以下のセルの名前定義が追加されます。(タスク等で本定義を利用しています)

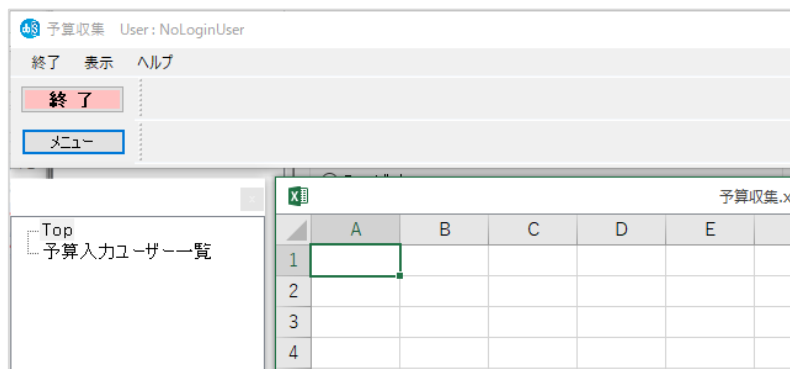
名前の管理

名前	値	参照範囲	範囲
予算入力ユーザー一覧_更新行数	10	=予算入力ユーザー一覧!\$N\$4	ブック
予算入力ユーザー一覧_書式コピー...	\$B\$14:K23	=予算入力ユーザー一覧!\$N\$7	ブック
予算入力ユーザー一覧_書式範囲	{",", ",", ",", ",", ",", ",", ",", ",", ",", "1"}	=予算入力ユーザー一覧!\$B\$10:\$K\$10	ブック
予算入力ユーザー一覧_展開位置		=予算入力ユーザー一覧!\$C\$14	ブック
予算入力ユーザー一覧_読込行数		=予算入力ユーザー一覧!\$N\$3	ブック
予算入力ユーザー一覧_編集位置		=予算入力ユーザー一覧!\$B\$14	ブック

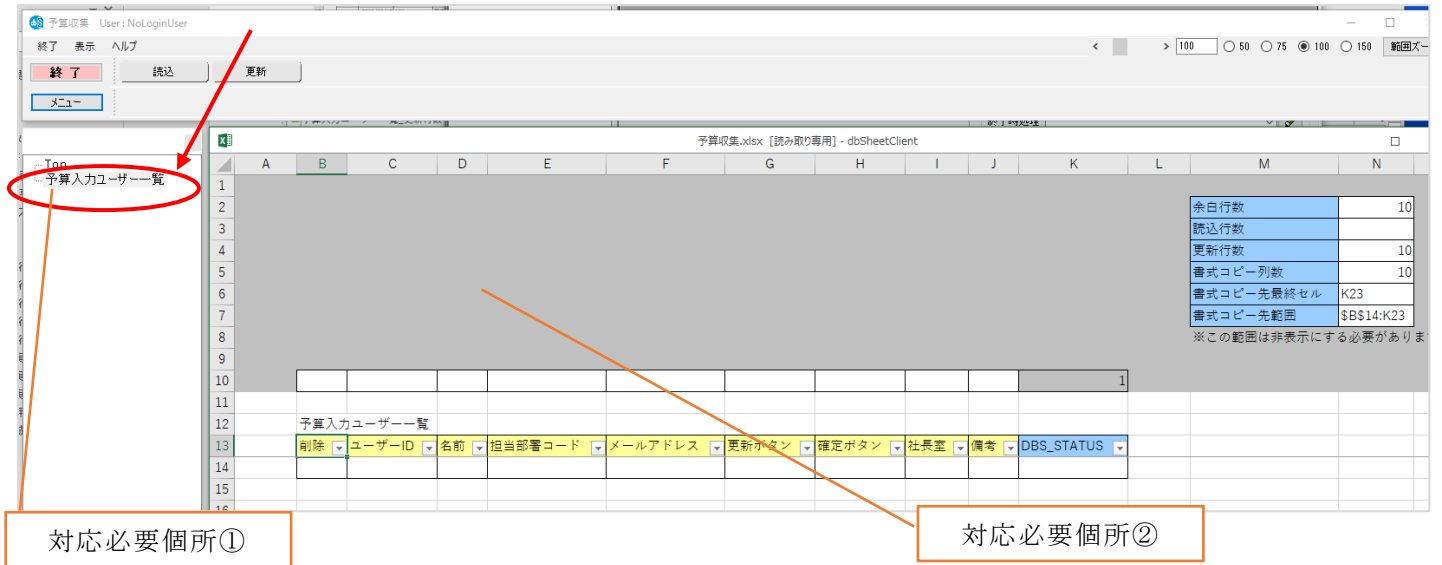
作成直後の状態で「デバッグ実行」して動作を確認します。



「デバッグ実行」直後の表示。(メニュー「TOP」のシート「TOP」を表示します。)



メニュー「予算入力ユーザー一覧」を選択します。



IOTG で作成直後の上記の表示状態では、以下①～③の対応が必要な個所があります。

対応必要個所①: メニュー階層の設定

「管理者メニュー」(上図には無し)の階層の下に
「予算入力ユーザー一覧」 を設定します。
本対応の方法は、後ほど説明します。

対応必要個所②: シートの 1～10 行の灰色行を非表示化

1～10 行の灰色部は作業領域であり、ユーザーには
非表示にします。
シート「予算入力ユーザー一覧」の 1～10 行を非表示化して
保存します。
(参考:プロジェクト「予算収集_完成」の Excel ファイル)

また、ボタン「読込」を押下すると、以下を表示します。

予算入力ユーザー一覧									
削除	ユーザーID	名前	担当部署コード	メールアドレス	更新ボタン	確定ボタン	社長室	備考	DBS
	NoLoginUser	NoLogin	all	nologin@xxxxxx.co.jp	1	1	1		
	satou	佐藤三郎	C02,D01	satou@xxxxxx.co.jp					
	suzuki	鈴木二郎	B01	suzuki@xxxxxx.co.jp	1			1	
	tanaka	田中一郎	A01	tanaka@xxxxxx.co.jp	1		1	1	
	yamada	山田四郎	all	yamada@xxxxxx.co.jp			1	1	

対応必要個所③: 列幅の調整が必要です。(文字がセル幅をオーバー)

シート「予算入力ユーザー一覧」の列幅を調整して保存します。
(参考:プロジェクト「予算収集_完成」の Excel ファイル)

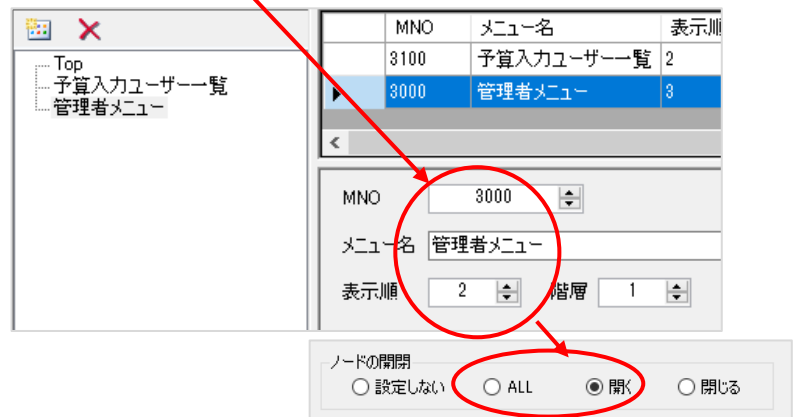
対応必要個所③

対応必要箇所①:メニュー階層の設定 についての説明です。

まず、新規に「管理者メニュー」を追加します。

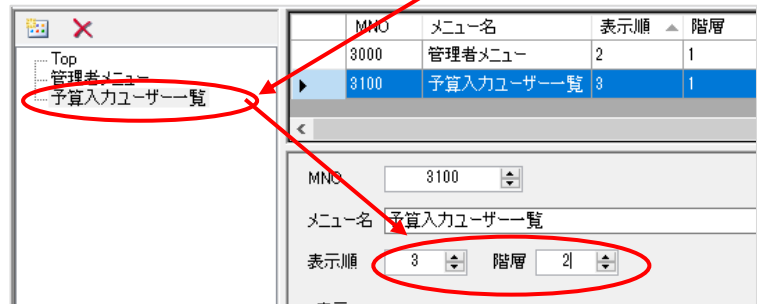


- ・MNO「3000」
- ・メニュー名「管理者メニュー」
- ・表示順「2」
- ・ノードの開閉「ALL」または「開く」



メニュー「予算入力ユーザー一覧」を選択。

- ・表示順「3」
- ・階層「2」



「管理者メニュー」の下に「予算入力ユーザー一覧」を設定できました。

メニューの追加、表示順、階層の設定は上記の要領で行います。

6.5.2 予算テーブル (IOTG 利用)

前の章で DB テーブル「予算テーブル」が作成済であることが前提です。

作成していない場合は、「6.4.5. 動作環境:DB ツールでテーブル作成」を参照して作成します。

前の「予算入力ユーザー一覧 (IOTG 利用)」とテーブル名、シート等が異なるだけで操作は基本的に同じです。

The screenshot shows the following steps in the configuration process:

- ブックの選択とシートの設定:** Selects the Excel book '1:予算収集' (BKNO), sheet name '予算テーブル', and starting cell 'B14'.
- データベースとテーブルの選択:** Selects the database '1:DBC作成' and the table '予算テーブル'.
- メニュー、ボタンの設定:** Specifies the menu name '予算テーブル', button set name '予算テーブル(管理者)', and defines the start numbers for MNO (3200), BSNO (3200), and TASKNO (3210).

Red circles and arrows indicate the following specific actions and fields:

- The 'IOTG' button in the top toolbar.
- The '予算テーブル' field in the 'データベースとテーブルの選択' dialog.
- The '次へ >' button in the 'データベースとテーブルの選択' dialog.
- The '次へ >' button in the 'ブックの選択とシートの設定' dialog.
- The '新しいメニューを作成する' checkbox and '予算テーブル' menu name field in the 'メニュー、ボタンの設定' dialog.
- The '予算テーブル(管理者)' button set name field in the 'メニュー、ボタンの設定' dialog.
- The '3200' values for MNO and BSNO, and '3210' for TASKNO in the 'メニュー、ボタンの設定' dialog.
- The '完了' button in the 'メニュー、ボタンの設定' dialog.
- The 'OK' button in the 'dbSheetClient' notification dialog.

- ・ボタンセット名
「予算テーブル(管理者)」
- ・MNO(メニュー番号)
「3200」
- ・BSNO(ボタンセット番号)
「3200」
- ・TASKNO(タスクセット番号)
「3210」

対応必要個所①～③も行います。

6.5.3 予算入力確定テーブル (IOTG 利用)

前の章で DB テーブル「予算入力確定テーブル」が作成済であることが前提です。

作成していない場合は、「6.4.5. 動作環境:DB ツールでテーブル作成」を参照して作成します。

前の「予算入力ユーザー一覧 (IOTG 利用)」とテーブル名、シート等が異なるだけで操作は基本的に同じです。

The screenshot shows the 'dbSheetClient 2018 ProjectEditor - 予算収集' interface. The main menu has 'IOTG' circled in red. The 'ブックの選択とシートの設定' dialog is open, with '1:予算収集' selected in the '使用するExcelブック' dropdown. The 'データベースとテーブルの選択' dialog is also open, with '1:DB作成' selected in the 'データベース' dropdown and '予算入力確定テーブル' selected in the 'テーブル' dropdown. The 'メニュー、ボタンの設定' dialog is open, with '新しいメニューを作成する' checked, '入力確定テーブル' in the 'メニュー名' field, '予算入力確定テーブル(管理者)' in the 'ボタンセット名' field, '3300' in the 'MNO' field, '3300' in the 'BSNO' field, and '3310' in the 'TASKNO' field. The '完了' button is circled in red. A small 'dbSheetClient' dialog box at the bottom left shows 'IOTGが正常に終了しました。' with an 'OK' button circled in red.

- メニュー名
「入力確定テーブル」
- ボタンセット名
「予算入力確定テーブル(管理者)」
- MNO (メニュー番号)
「3300」
- BSNO (ボタンセット番号)
「3300」
- TASKNO (タスクセット番号)
「3310」

対応必要個所①～③も行います。

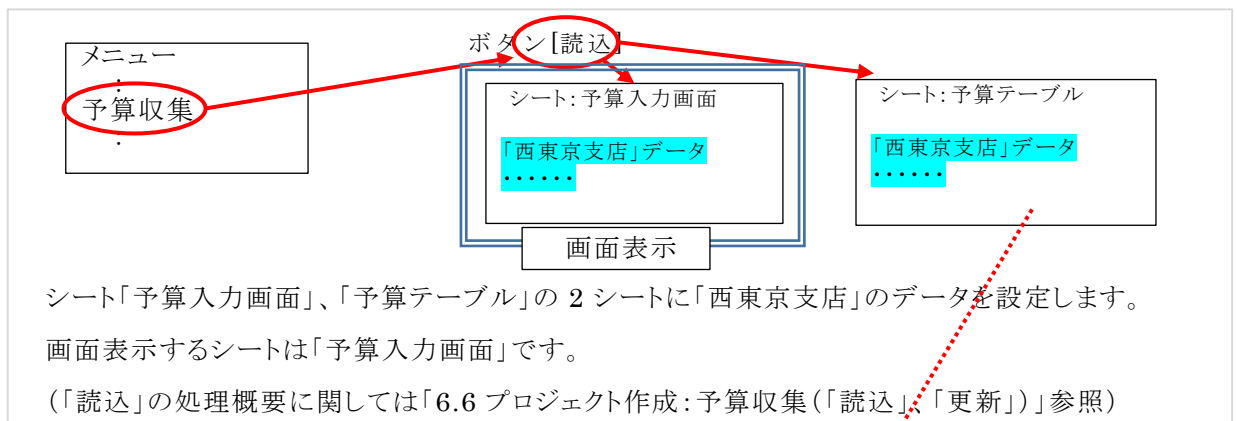
6.5.4 メニュー切替で表示クリア

メニューでシートを切替した時、以前に設定していた内容をクリアする処理を追加します。

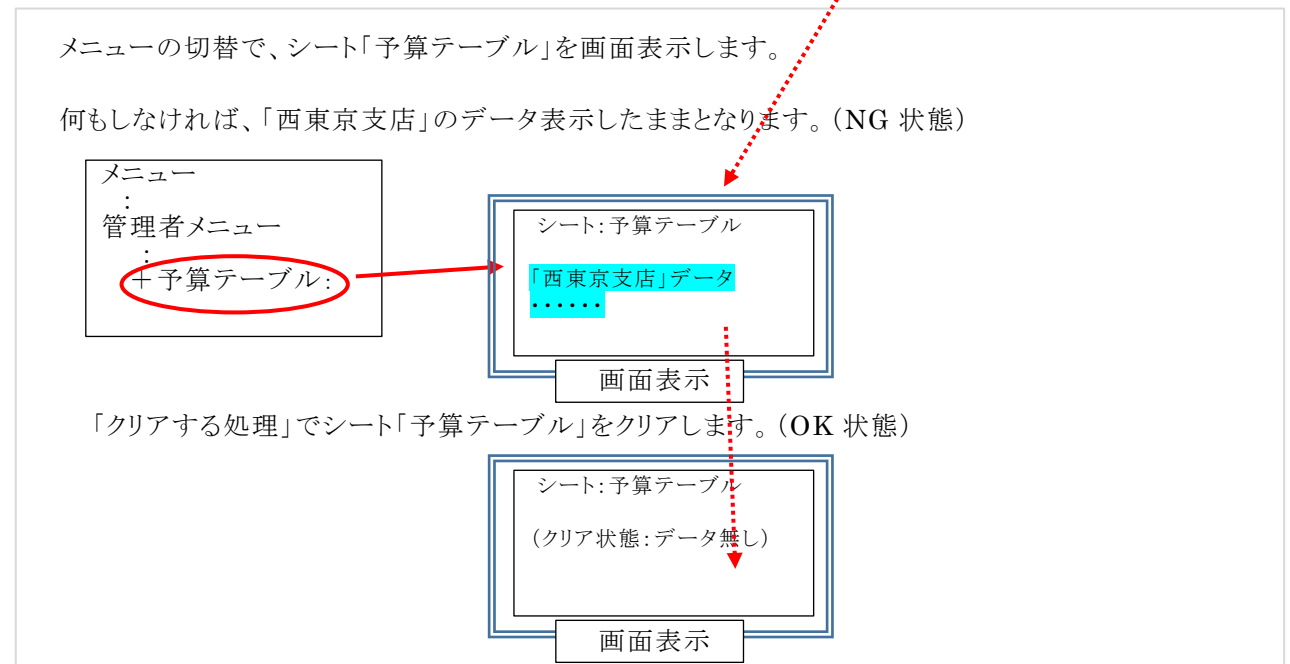
(本システムの仕様です。複数のメニューの機能でシート「予算テーブル」などを共有して利用しているためです。)

「クリアする処理」が必要なシーンを以下の例で説明します。

例： メニュー「予算収集」のボタン「読込」で「西東京支店データ」を表示した場合。



その後、メニュー「管理者メニュー> 予算テーブル」に切替した場合。



管理者メニューで利用する以下の 3 テーブルの各クリア処理を新規に作成します。

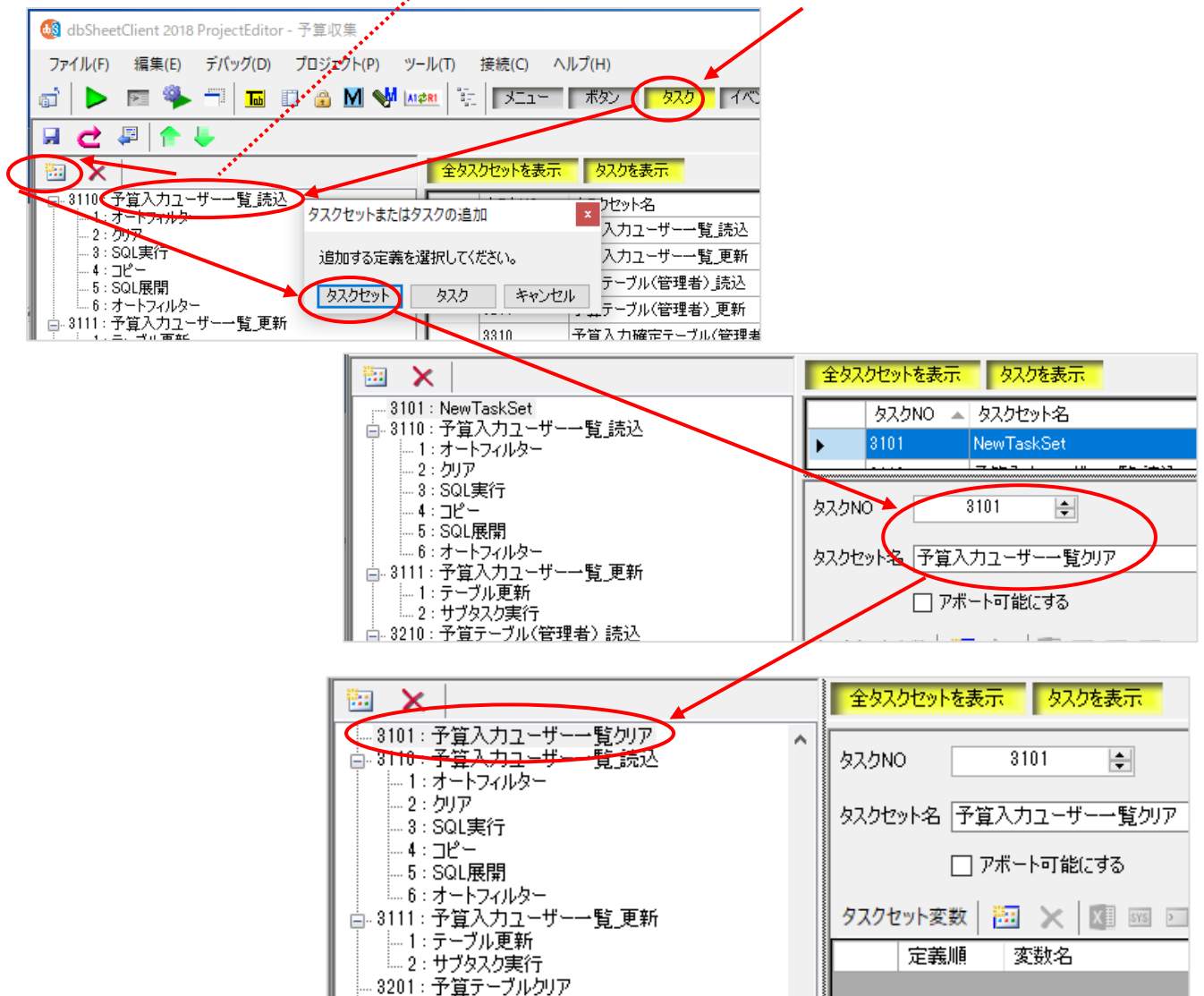
- ・① 予算入力ユーザー一覧、② 予算テーブル、③ 予算入力確定テーブル

「① 予算入力ユーザー一覧」のクリア処理を新規に作成します。
 (既に IOTG で作成済のクリア処理のタスクを利用します。)

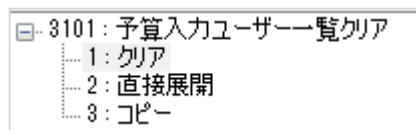
TASKNO:3101(予算入力ユーザー一覧クリア)

タスクセット「予算入力ユーザー一覧クリア」(TASKNO:3101)を新規追加します。

(注:「追加ボタン」押下の前にタスクセットを選択します。タスクが選択されている場合、タスクの追加となりますので注意が必要です。)



このタスクセットにタスク(処理順:1~3)を追加します。



処理順: 1(クリア)

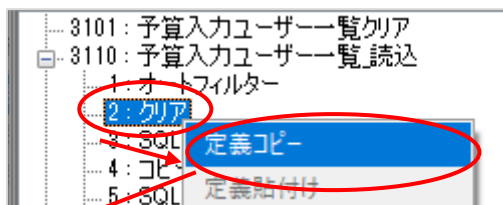
処理の目的: 既存の展開データ、書式をクリアします。

クリア範囲は、既に定義してある名前付きセルで自動設定しています。

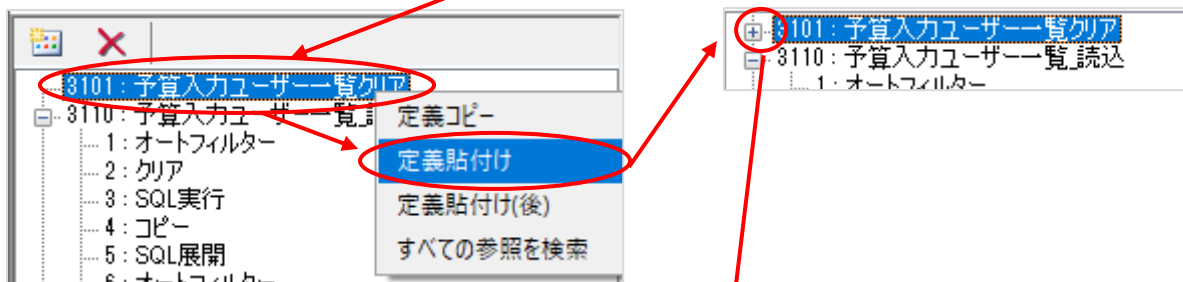
クリアする処理のタスクは IOTG で既に作成済です。

IOTG で作成したタスクを以下の手順でコピーします。

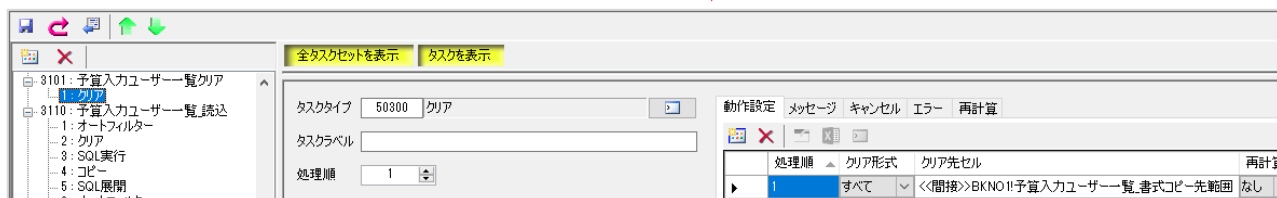
コピー元のタスク「クリア」を右クリックし、「定義コピー」します。



コピー先のタスクセットを選択、右クリックして「定義貼付け」で、タスクが張付きます。

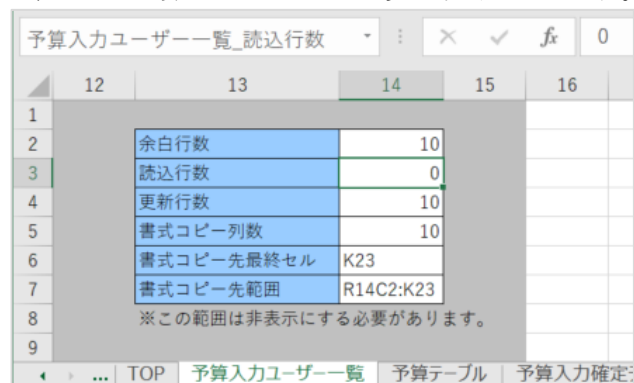


コピー後のタスクは、以下のようになります。

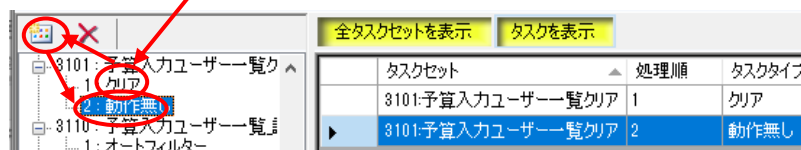


処理順: 2(直接展開)

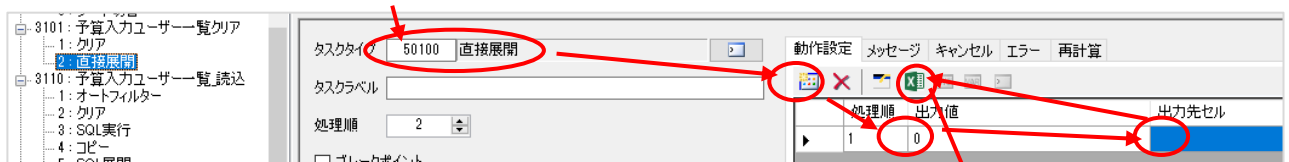
処理の目的: この次の処理順:3(コピー)で書式(罫線あり)のコピーを行います。
 ここでは、データが 0 件の状態のコピー範囲を設定するために、セル名「予算入力ユーザー一覧_読込行数」を 0 に設定します。
 (書式コピーする行数は、「読込行数」+「余白行数」です。
 「余白行数」が 10 に設定してあるため「読込行数」が 0 の場合は、書式コピーする行数は 10 行分になります。
 これらの定義は、シート:予算入力ユーザー一覧の以下にあります。)



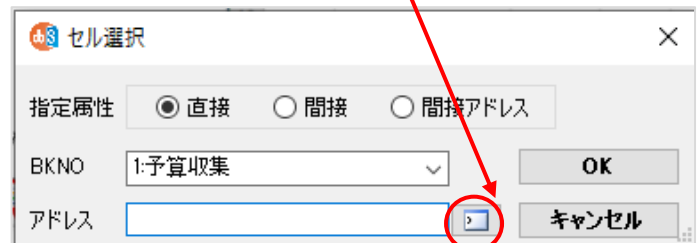
先ほどコピーしたタスク:(クリア)を選択し、「追加」アイコンでタスク:(動作無し)が追加されます。



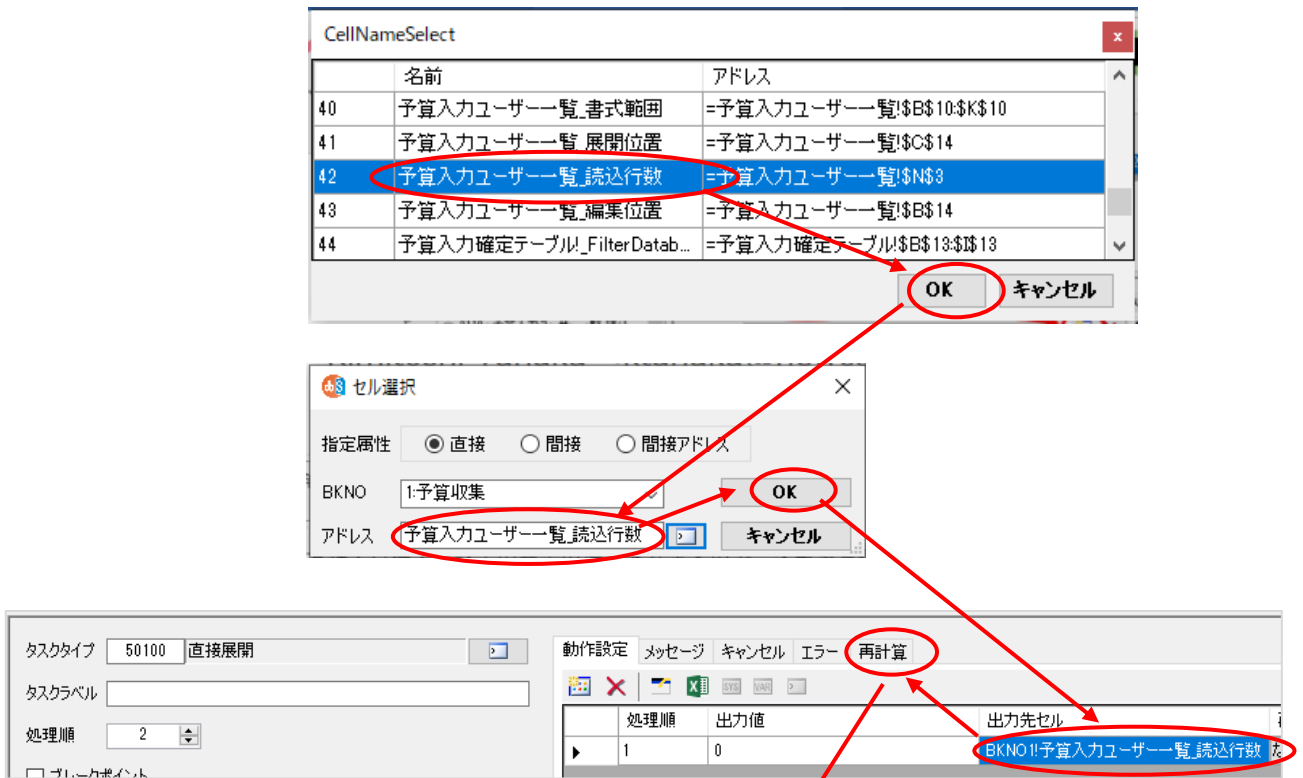
追加したタスク:(動作無し)を(直接展開)にして、0 を設定します。



出力先セルを選択後、Excel のアイコンを押下して定義します。(手入力でも可能です)



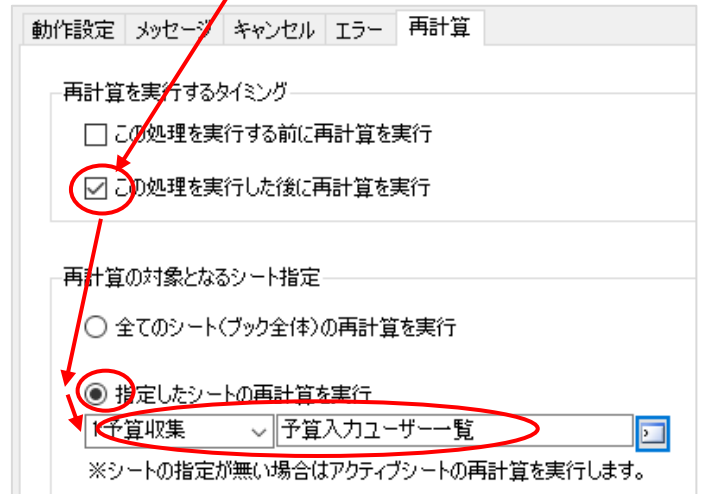
名前付きセルの一覧から「予算入力ユーザー一覧_読込行数」を選択、OK します。



タブ: 再計算の設定を行います。

(シート: 予算入力ユーザー一覧のセル式の自動計算を行うように設定します。

次の処理で再計算結果を利用するため必要です。)



処理順:3(コピー)

処理の目的: 空データの書式をコピーします。

コピー範囲は、既に定義してある名前付きセルで自動設定しています。

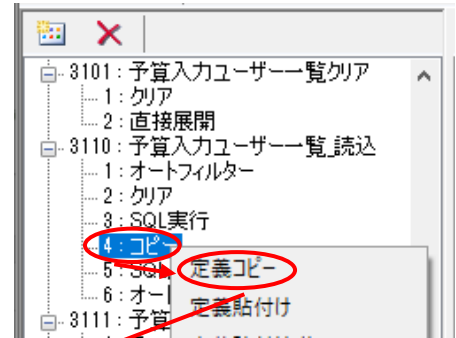
コピーする処理のタスクは IOTG で既に作成済です。

IOTG で作成したタスクを以下の手順でコピーします。

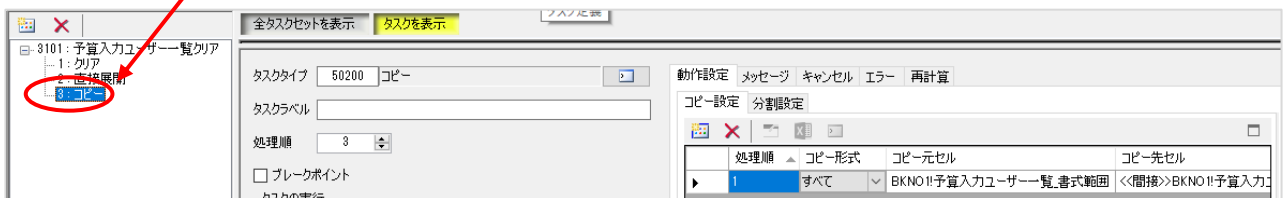
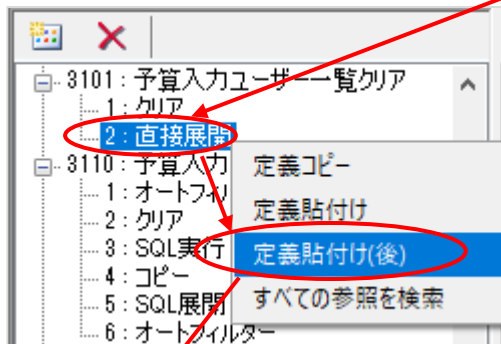
コピー元のタスク「コピー」

(TASKNO:3110 の処理順:4)

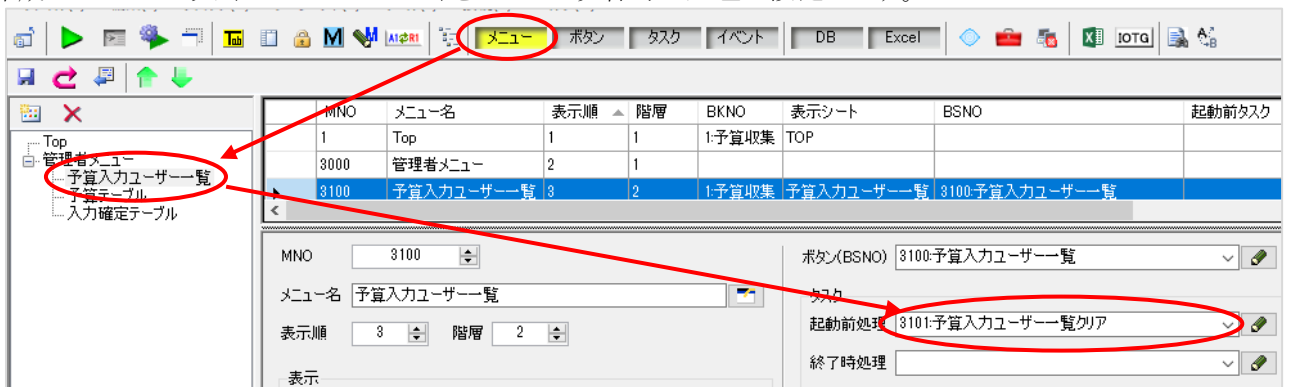
を右クリックし、「定義コピー」します。



コピー先のタスク(TASKNO:3101 の処理順:2 直接展開)を選択し、右クリックして「定義貼付け(後)」で、処理順:3 にタスクが張付きます。



作成したタスクセット(TASKNO:3101)をメニュー切替時の処理に設定します。



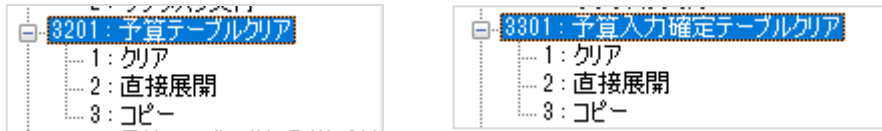
以上で「予算入力ユーザー一覧」のシート切替時のクリア設定は終了です。

メニュー:「予算テーブル」、「予算入力確定テーブル」のクリア処理に関しても同様に行います。

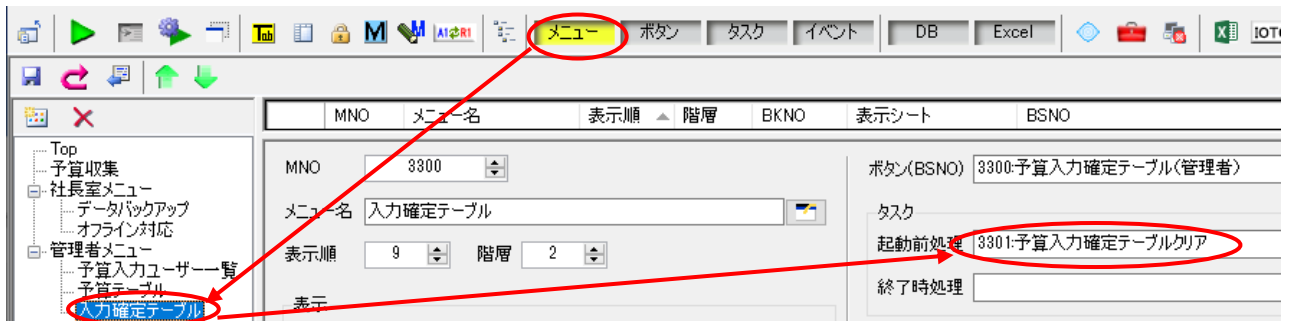
No	追加 タスクセット名	追加 TASKNO	コピー元 TASKNO	値:0 を直接展開する セル名	再計算シート 名
①	予算入力ユーザー 一覧クリア	3101	3110	予算入力ユーザー 一覧_読込行数	予算入力ユーザー 一覧
②	予算テーブルクリア	3201	3210	予算テーブル_読込行 数	予算テーブル
③	予算入力確定テー ブルクリア	3301	3310	予算入力確定テーブ ル_読込行数	予算入力確定 テーブル

No.①は、前記で作成しました。No.②、③を①に倣って作成します。

No.②、③の作成後のタスクセットは以下の通りとなります。



また、No.②、③をメニュー切替時の処理に追加します。



6.5.5 作成機能の動作確認（予算入力ユーザー一覧：担当部署コードの変更）

作成した機能の一部の動作確認をしながら、データ変更します。

これから作成する「予算収集」の機能では、ユーザーの担当部署コードが「複数」、「all」には対応しないため、既存データの「複数（部署コードのカンマ区切り）」、「all」を変更します。

「all」→「A01」に変更。

「C02,D01」→「C02」に変更。

変更後、「更新」します。

削除	ユーザー-ID	名前	担当部署コード	メー
	NoLoginUser	NoLogin	all	no
	satou	佐藤三郎	C02.D01	sa
	suzuki	鈴木二郎	B01	su
	tanaka	田中一郎	A01	ta
	yamada	山田四郎	all	ya

削除	ユーザー-ID	名前	担当部署コード	メー
	NoLoginUser	NoLogin	A01	nolog
	satou	佐藤三郎	C02	satou
	suzuki	鈴木二郎	B01	suzu
	tanaka	田中一郎	A01	tanaka
	yamada	山田四郎	A01	yamada

6.6 プロジェクト作成：予算収集（「読込」、「更新」）

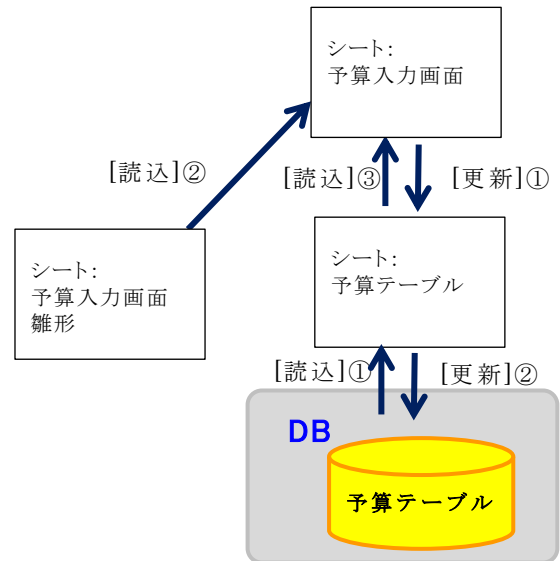
予算収集（「読込」、「更新」）の機能を作成します。
作成する処理の概要は以下の通りです。

「読込」：[読込]①～③を行います。

- ① : IOTG の機能で作成した処理を元として読込
- ② : ユーザー入力画面の雛形(空データ)を設定
- ③ : 入力画面(雛形設定済)に
読込データを設定(①と同じデータを設定)

「更新」：[更新]①～②を行います

- ① : 編集データを DB に更新するシートにコピー
- ② : IOTG の機能の更新

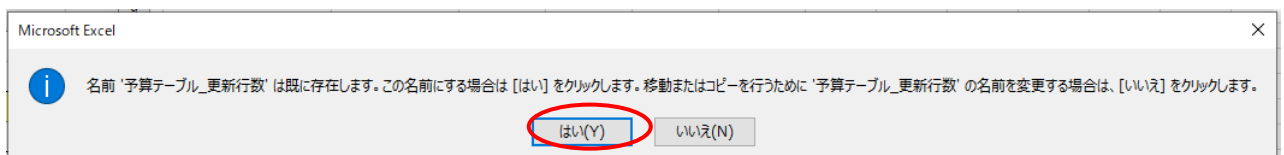


6.6.1 Excel の前準備

以下のシートをプロジェクト「予算収集_完成」の「予算収集_完成.xlsm」から「予算収集.xlsx」にコピーします。

- ・「予算入力画面」 : 「予算収集」で利用。(前記の概要を参照)
- ・「予算入力画面雛形」 : 「予算収集」で利用。(前記の概要を参照)
- ・「条件」 :
各機能を利用するにあたり必要な情報を格納、共有するためのシートです。
「予算収集」ではユーザーの情報を本シートに展開し、展開した担当部署を読込時に利用します。
- ・「work1」 : 「進捗状況」で利用するシート
- ・「work2」 :
空のシート。メニュー「社長室メニュー」、「管理者メニュー」選択時に表示するシート
- ・「説明補足」 :
メニュー「社長室メニュー」の「データバックアップ」、「査定データ準備」、「オフライン対応」で表示するシート

シートコピー時に、以下の画面が表示されますが、すべて「はい」とします。



6.6.2 メニュー「予算収集」の設定の前準備

メニュー「予算収集」の設定をする前に、メニュー選択時のタスクセット、ボタンの設定を行います。

タスクセットの追加は以下です。(以降で詳細を説明します)

タスクセットは、TASKNO:1、1002、1000 の順に作成します。

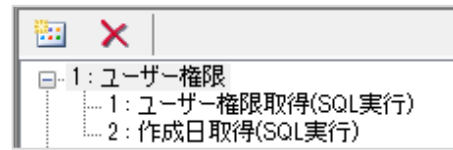
追加 TASKNO	タスクセット名	概要
1	ユーザー権限	ユーザー情報をシート条件に展開。 予算テーブル等を参照する「作成日」(最新の作成日)を取得。
1002	入力画面クリア	入力画面の初期化(データ無しの初期表示状態)
1000	メニュー前処理: 予算収集	タスクセット:1,1002 の呼び出し。 メッセージ(読込、更新の説明)の表示。

ボタンの追加は以下です。

追加 BSNO	ボタンセット名	概要
1000	予算テーブル(ユーザー編集)	ボタン「読込」、「更新」

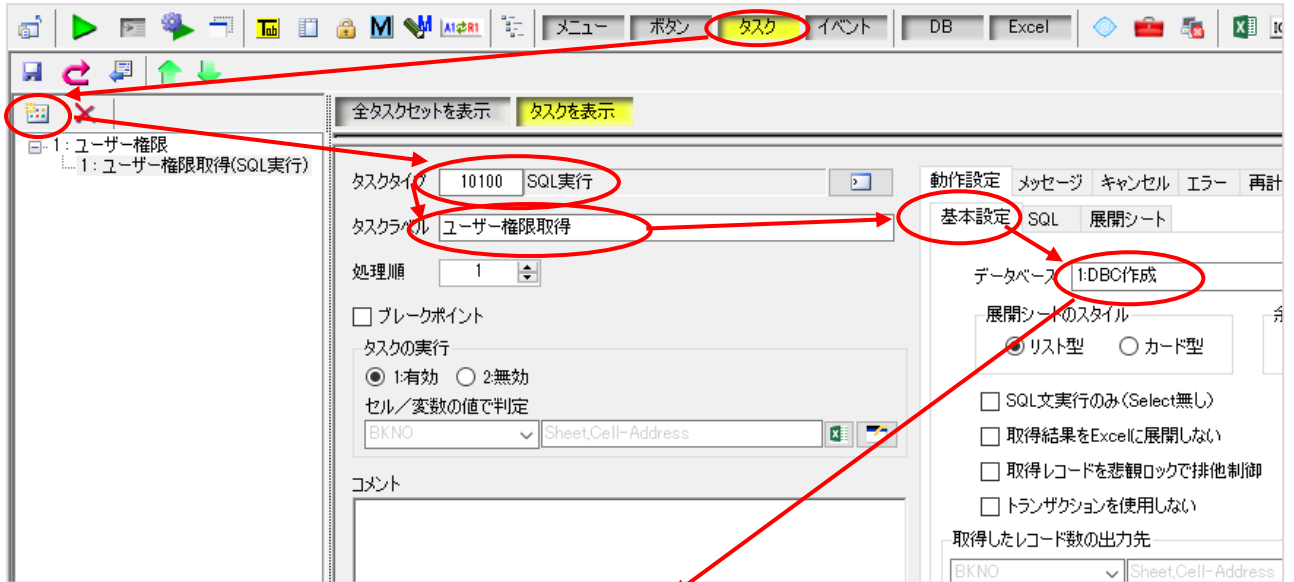
TASKNO:1(ユーザー権限)

タスクセットを新規追加します。



処理順:1(SQL 実行)

処理の目的: DB のユーザー情報をシート:条件に展開します。

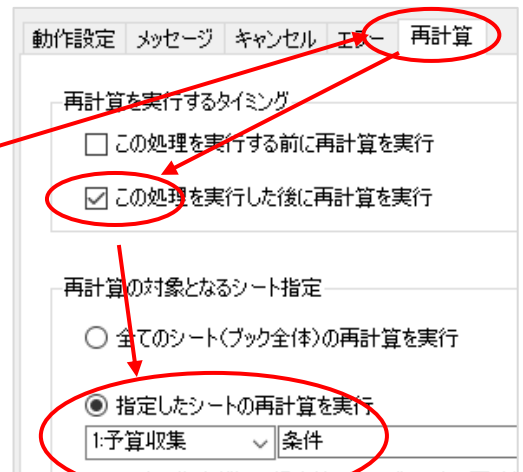
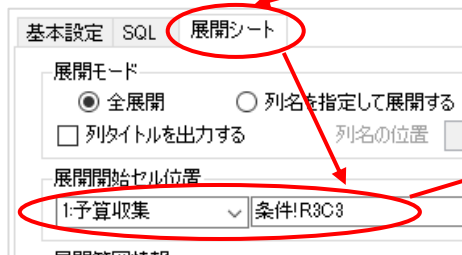
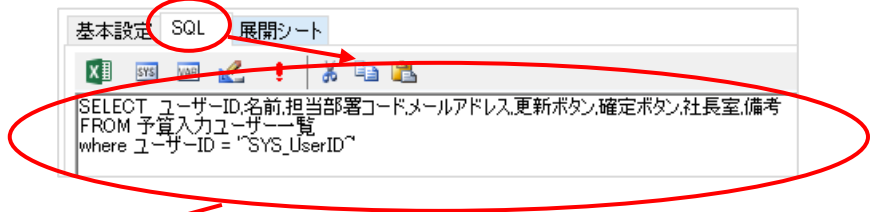


SQL 文はプロジェクト

「予算収集_完成」から

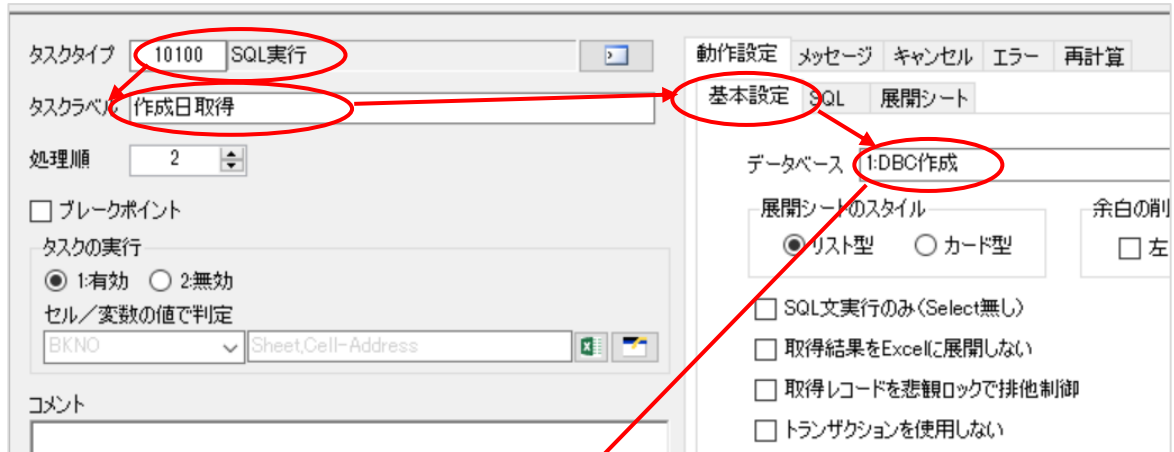
コピーできます。

(TASKNO:1 は同じです)

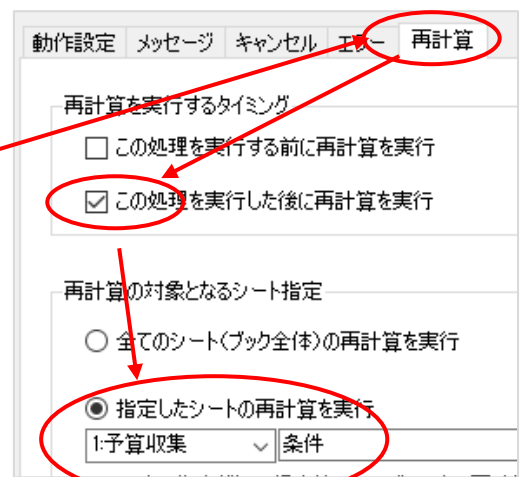
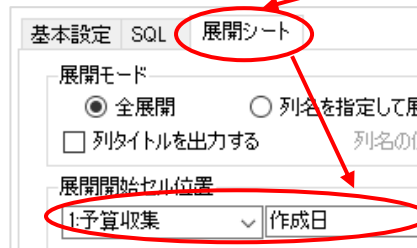
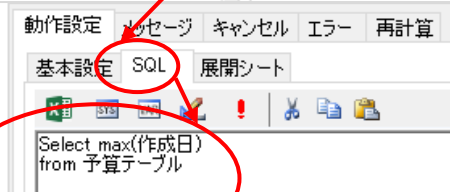


処理順: 2(SQL 実行)

処理の目的: DB から予算テーブル等を参照する「作成日」(最新の作成日)を取得します。



SQL 文はプロジェクト「予算収集_完成」からコピーできます。



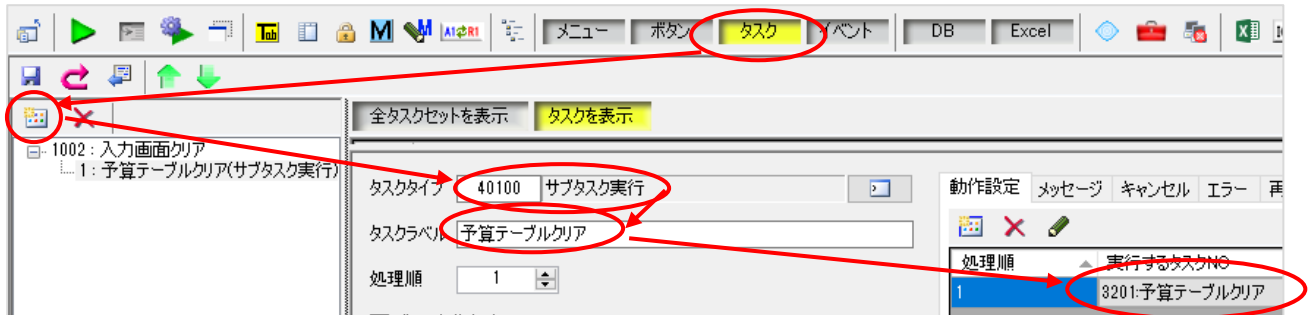
TASKNO:1002(入力画面クリア)

タスクセットを新規追加します。

処理順:1(サブタスク実行)

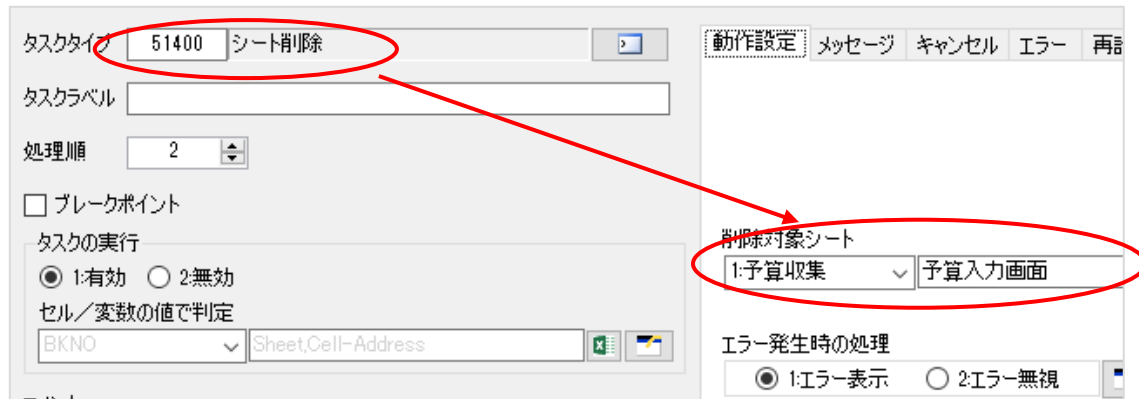
処理の目的: シート「予算テーブル」をクリアします。

- 1002: 入力画面クリア
 - 1: 予算テーブルクリア(サブタスク実行)
 - 2: シート削除
 - 3: シートコピー
 - 4: セルジャンプ



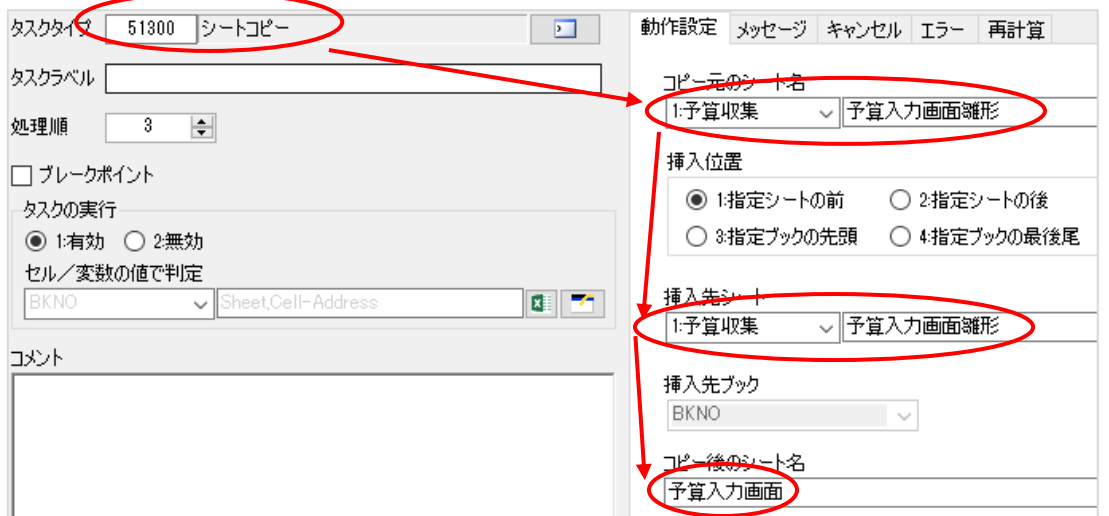
処理順:2(シート削除)

処理の目的: シート「予算入力画面」を処理順:3 でコピーする前にシート削除します。



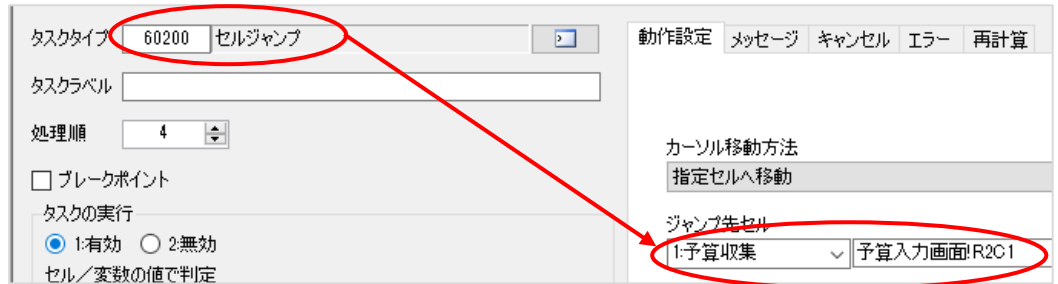
処理順:3(シートコピー)

処理の目的: シート「予算入力画面」をコピーして作成します。



処理順: 4(セルジャンプ)

処理の目的: 先頭行から表示するようにセルジャンプします。

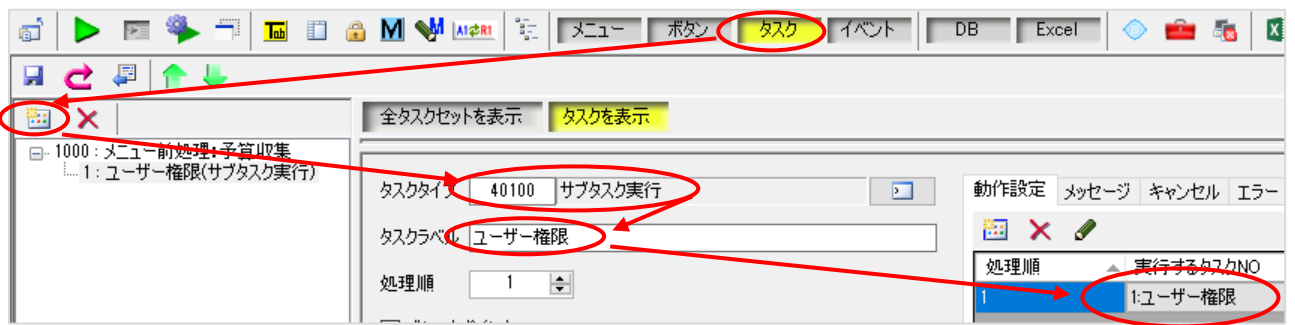
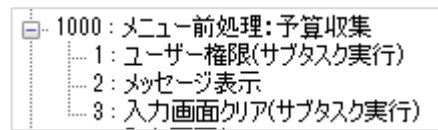


TASKNO: 1000(メニュー前処理: 予算収集)

タスクセットを新規追加します。

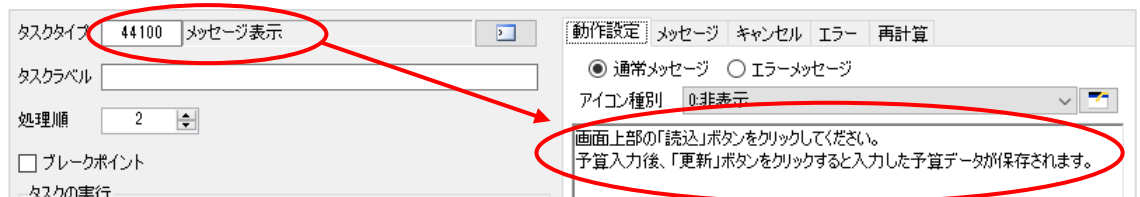
処理順: 1(サブタスク実行)

処理の目的: 最新のユーザー情報を取得して、予算収集の準備をします。



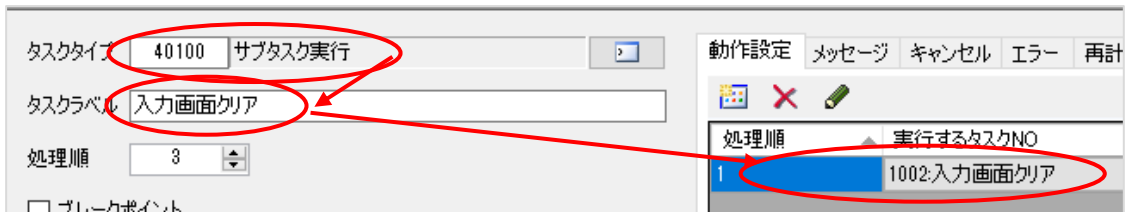
処理順: 2(メッセージ表示)

処理の目的: ユーザーへの操作方法のメッセージ表示。



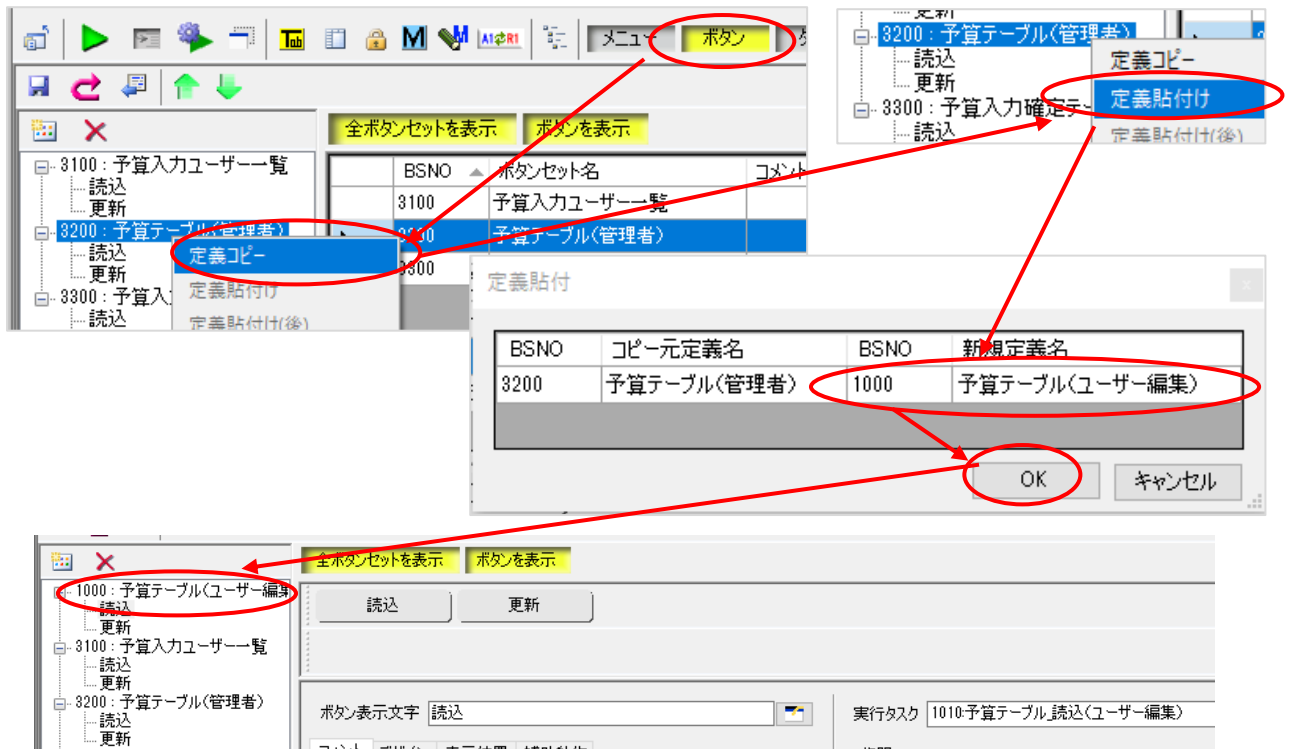
処理順: 3(サブタスク実行)

処理の目的: メニュー切替後にクリア状態の入力画面を表示します。



BSNO:1000(予算テーブル(ユーザー編集))

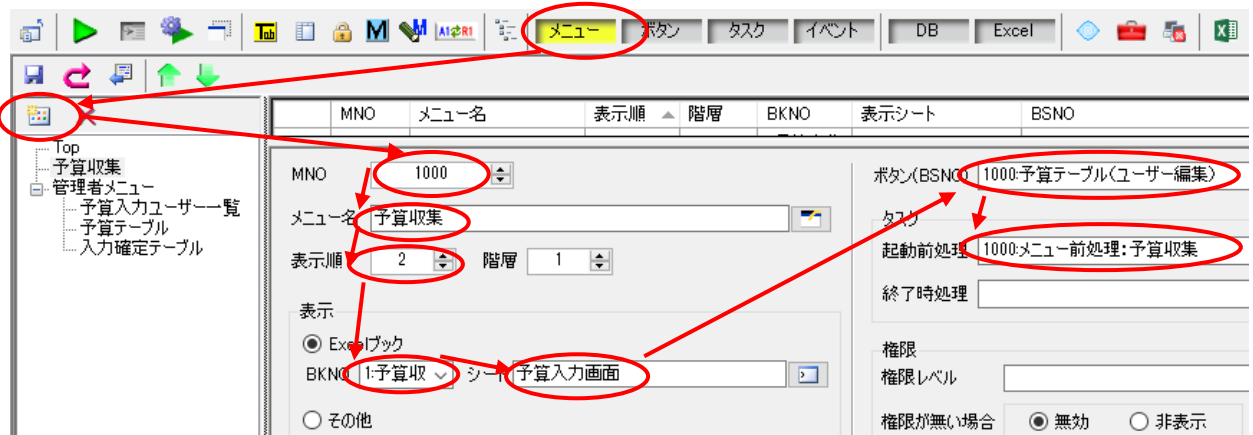
既存の BSNO:3200 をコピーし、BSNO:1000 予算テーブル(ユーザー編集) を作成します。



ボタン「読込」、「更新」の処理は、後ほど作成します。

6.6.3メニュー「予算収集」の設定

メニュー「予算収集」を以下の手順で新規作成します。

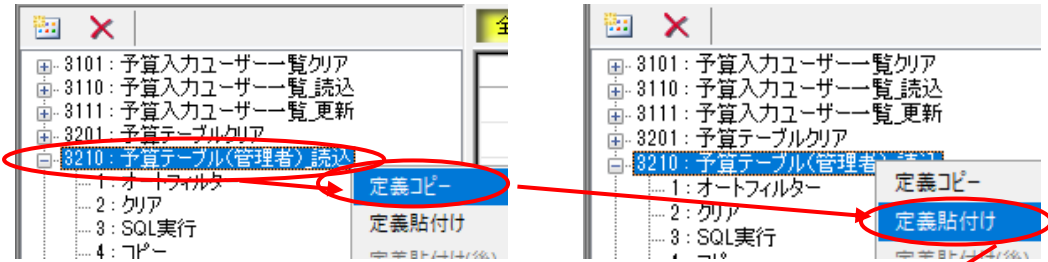


本設定で、メニュー「予算収集」を押下した時に初期表示するまでの処理の設定は終了です。

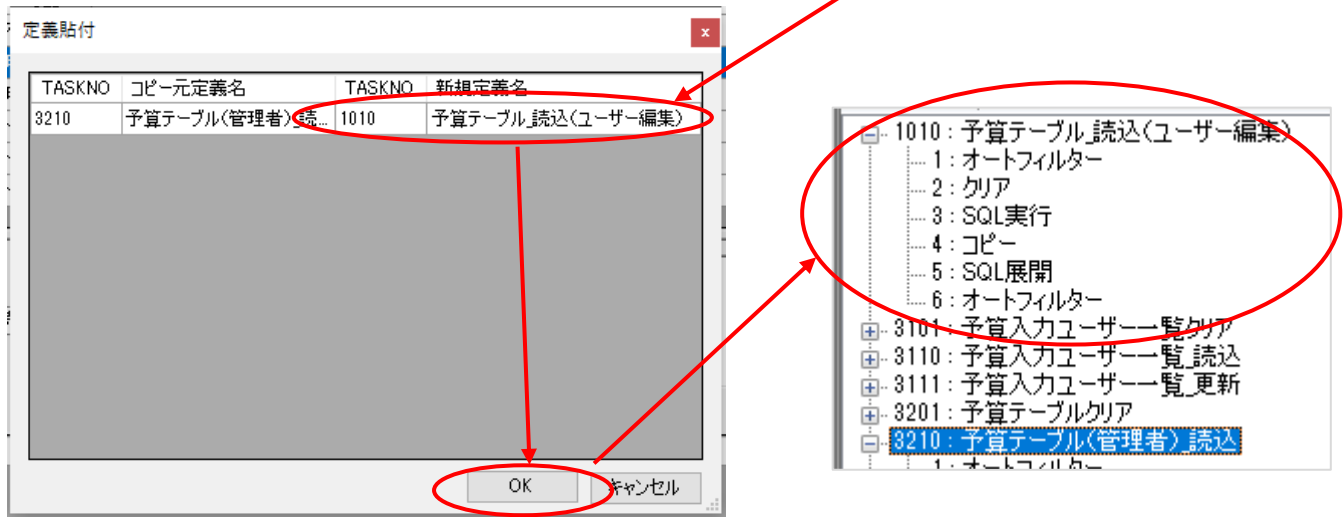
6.6.4 「読込」 ①

「読込」ボタンを押下した時の処理 (概要説明の[読込]①～③) の作成に入ります。

IOTG で作成済の「予算テーブル」関連タスクセットをコピーし、「予算収集」用にカスタマイズします。



TAKNO:1010、新規定義名: 予算テーブル_読込(ユーザー編集)とします。



カスタマイズ前の TASKNO:1010 のタスクセットが作成されました。
このタスクセットに対して、以下の処理の追加、変更をします。

- ・処理順: 1 (コピー) タスクを新規追加
- ・処理順: 2 (SQL 実行) タスクを新規追加
- ・処理順: 5 (SQL 実行) タスク内容を変更 (元々の処理順: 3)
- ・処理順: 7 (SQL 展開) タスク内容を変更 (元々の処理順: 5)

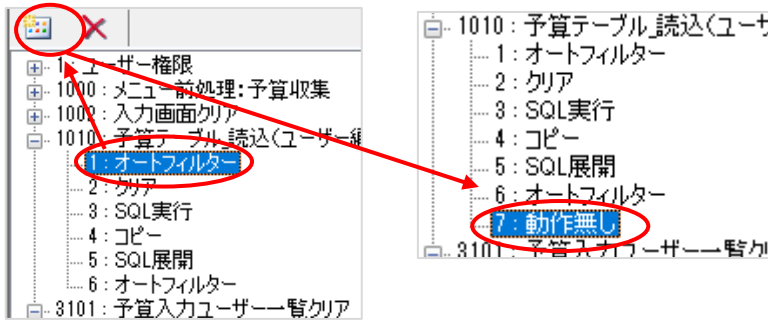
TASKNO:1010(予算テーブル 読込(ユーザー編集))

処理順:1(コピー)

処理の目的: 読込の検索時に利用する「部署コード」を Excel シート内でコピーします。

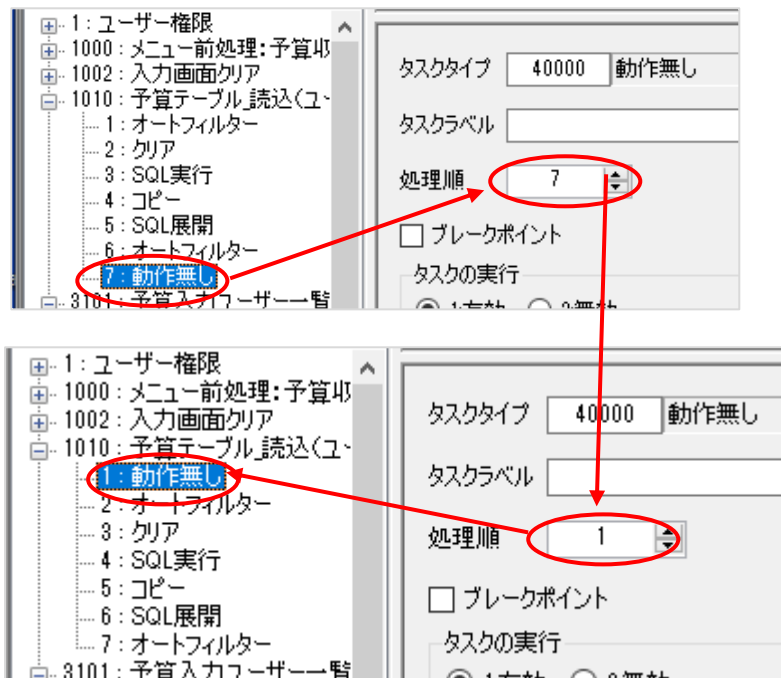
処理順:1にタスク「コピー」を新規に追加します。

タスクセット内の任意のタスクを選択(例:処理順:1)し、タスクを新規追加します。

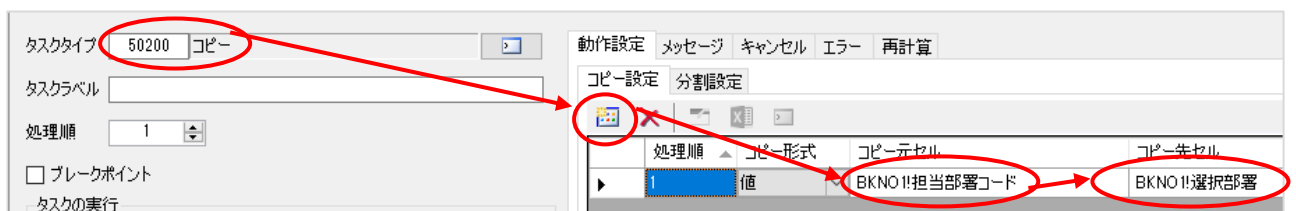


処理順:7 に新規 追加されます。

処理順:7 を処理順:1 に変更します。



処理順:1 に変更後、タスクタイプ「コピー」にして、コピー設定をします。

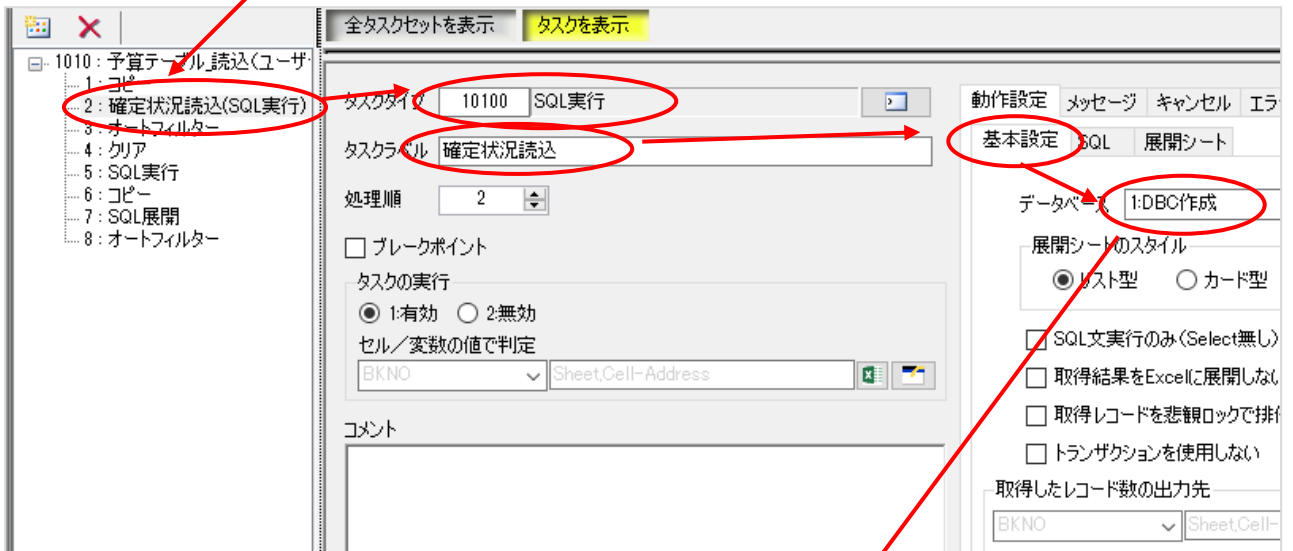


処理順:2(SQL 実行)

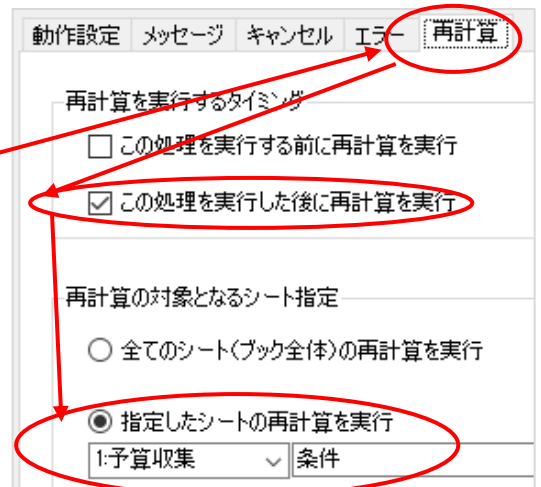
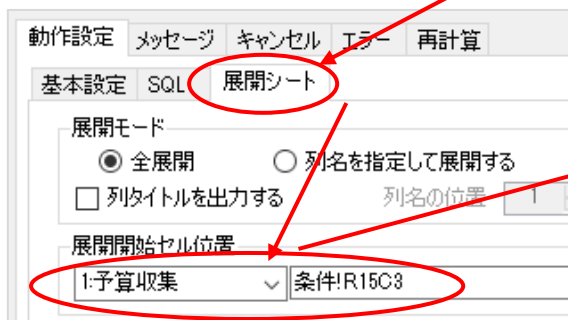
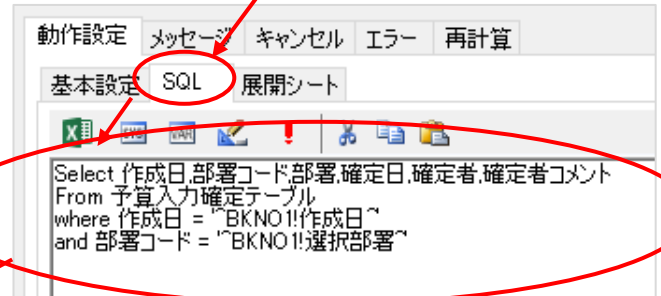
処理の目的: 部署コードの部署の確定状況を Excel シートに展開します。

(ここで展開する項目「部署」の名称が、後述する「オフラインファイル保存」のファイル名に利用されます)

タスクを新規追加します。(上記の処理順:1 参照)

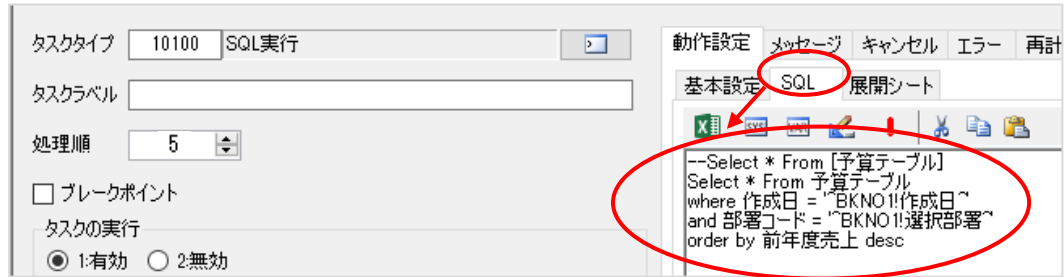


SQL 文はプロジェクト「予算収集_完成」からコピーできます。



処理順:5(SQL 実行)

既存の SQL 文を
変更します。

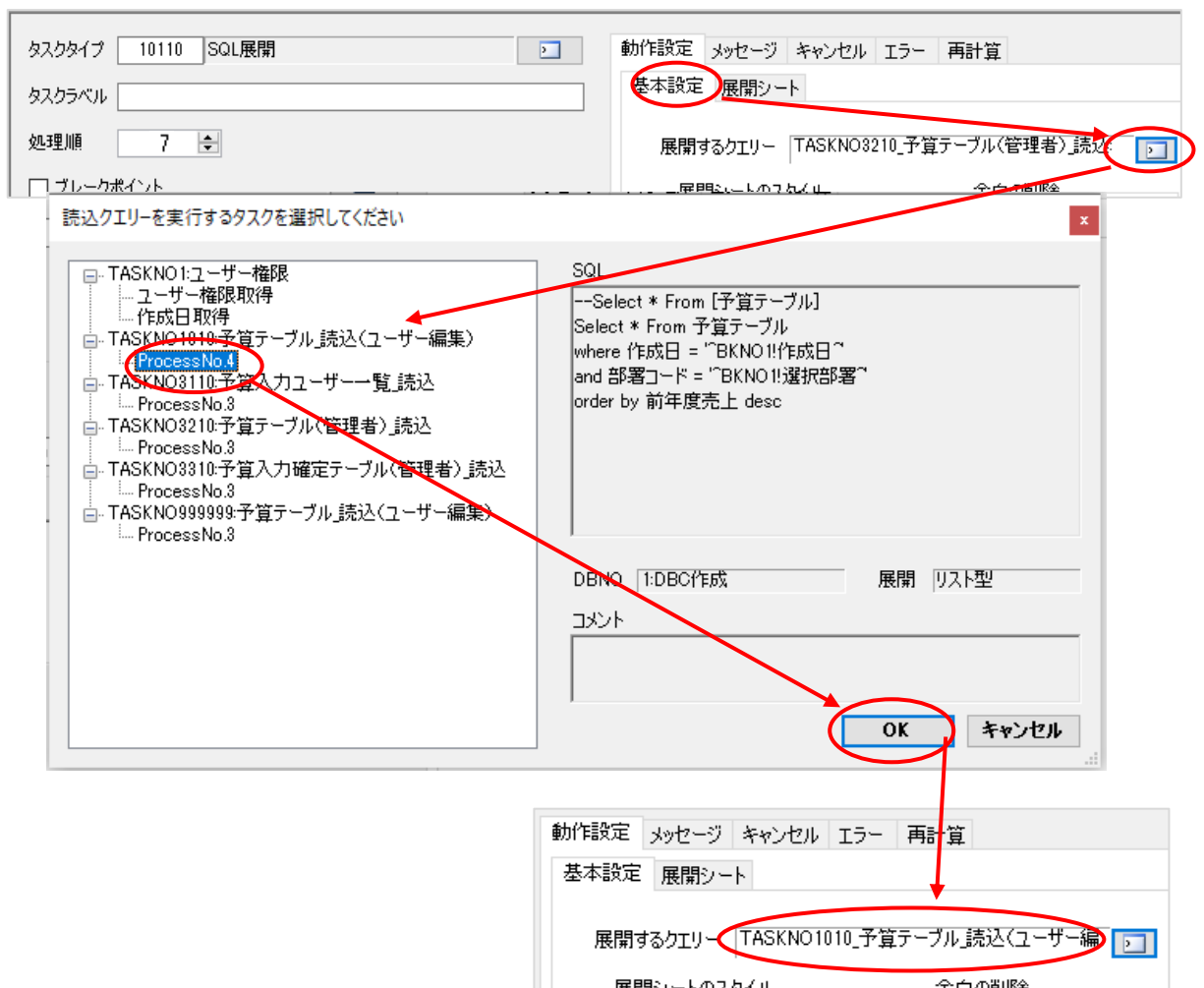


SQL 文はプロジェクト「予算収集_完成」からコピーできます。 (TASKNO:1010 は同じです。)
TASKNO:1 で取得済の「作成日」、処理順:1 でコピーした「選択部署」のデータを利用して、
DB から抽出します。

(変更前の SQL 文(-- でコメントアウト)では、全データを取得していました)

処理順:7(SQL 展開)

展開するクエリ (TASKNO:3210) がタスクセットのコピー前のものであるので、
(TASKNO:1010)に変更します。



6.6.5 「読込」②

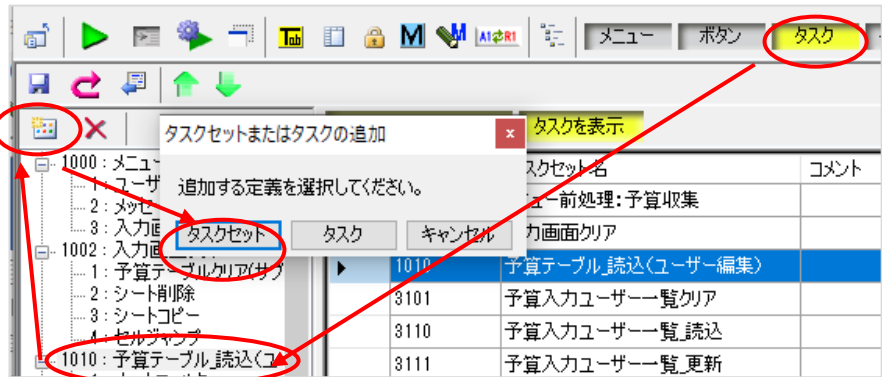
読込②、③の処理は別のタスクセット(TASKNO:99015)を新規追加して設定します。

(「予算収集_完成」に合わせます。「予算収集_完成」では本処理をタスク、マクロのいずれかで処理できるようにしています。ここでは、タスクの処理(TASKNO:99015)の設定を行います。)

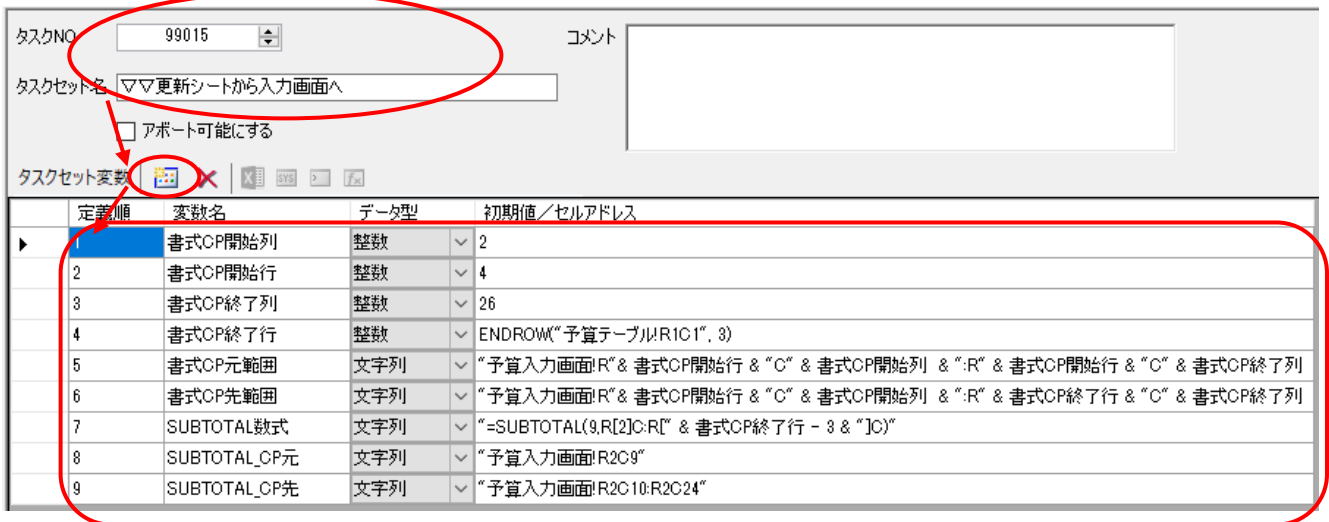
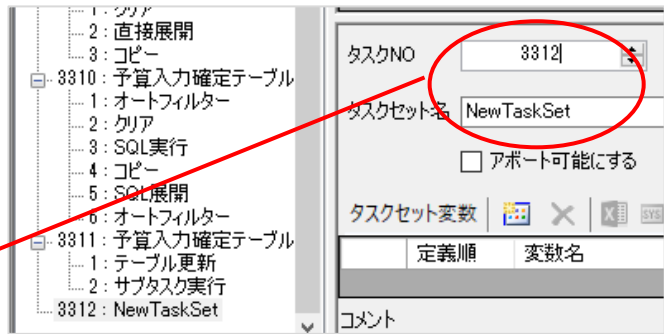
タスクセットを選択

(例:TASKNO:1010)後、「追加」のボタンを押下します。

(タスクセットが未選択の場合は、タスクセットの追加ができません)



追加した直後のタスクセット(右)をタスク NO から設定をします。



タスクセット変数の「追加」のボタンで行追加し、9個のタスクセット変数を追加します。

(「予算収集_完成」から文字列の選択コピーができます)

これらの変数はこのタスクセット内で利用可能な変数です。データの取得件数が変わると書式コピーする範囲が変わります。そのような時などに利用できます。(各変数の詳細な説明は割愛します。利用用途は変数名を参考にしてください。)

TASKNO:99015(▽▽更新シートから入力画面へ)

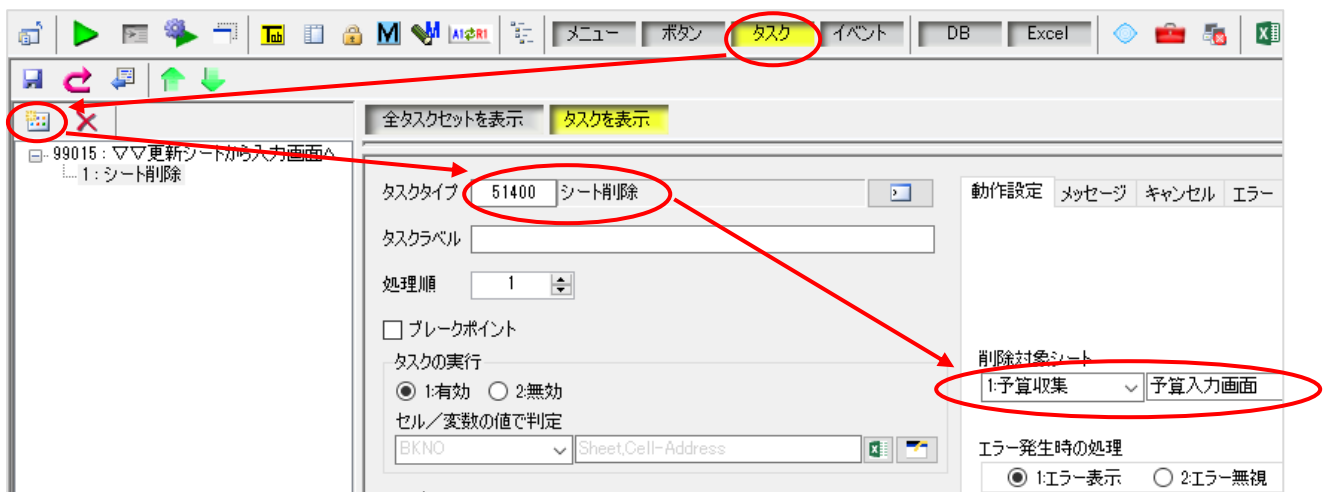
処理順:1~8 を追加します。

- 99015: ▽▽更新シートから入力画面へ
 - 1: シート削除
 - 2: シートコピー
 - 3: コピー
 - 4: SQL展開
 - 5: オートフィルター
 - 6: SUBTOTAL 数式 設定(直接展開)
 - 7: SUBTOTAL 数式 コピー(コピー)
 - 8: セルジャンプ

処理順:1(シート削除)

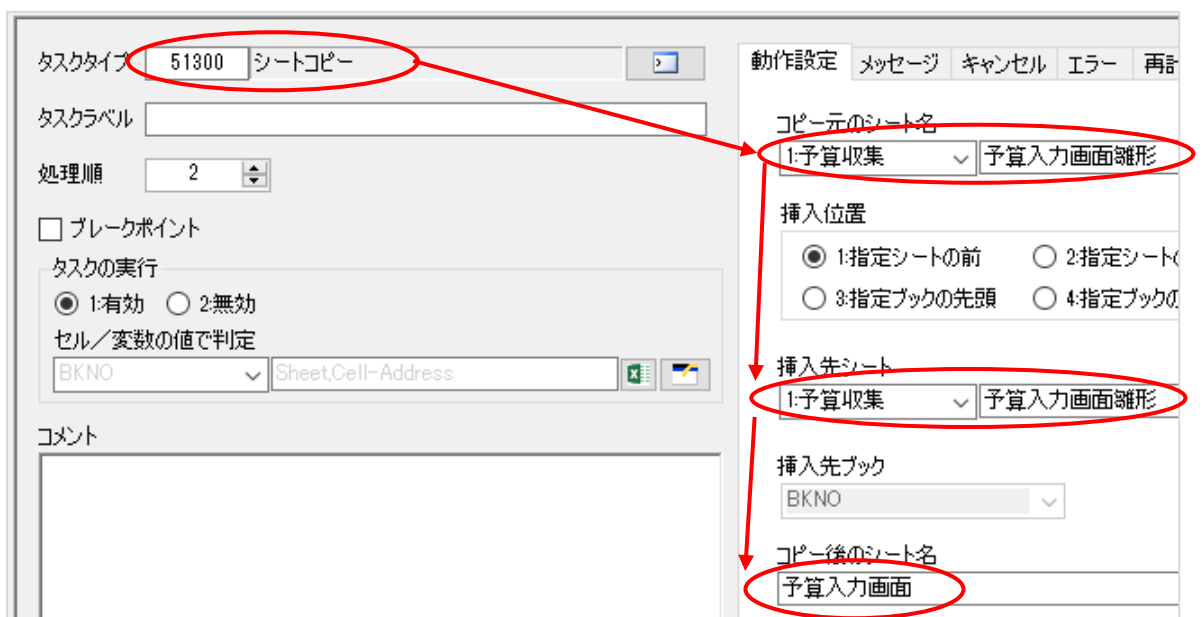
処理の目的: シート「予算入力画面」を処理順:2 でコピーする前にシート削除します。

(シート切替時と同様に、読込時にも既存データクリアのためコピーします)



処理順:2(シートコピー)

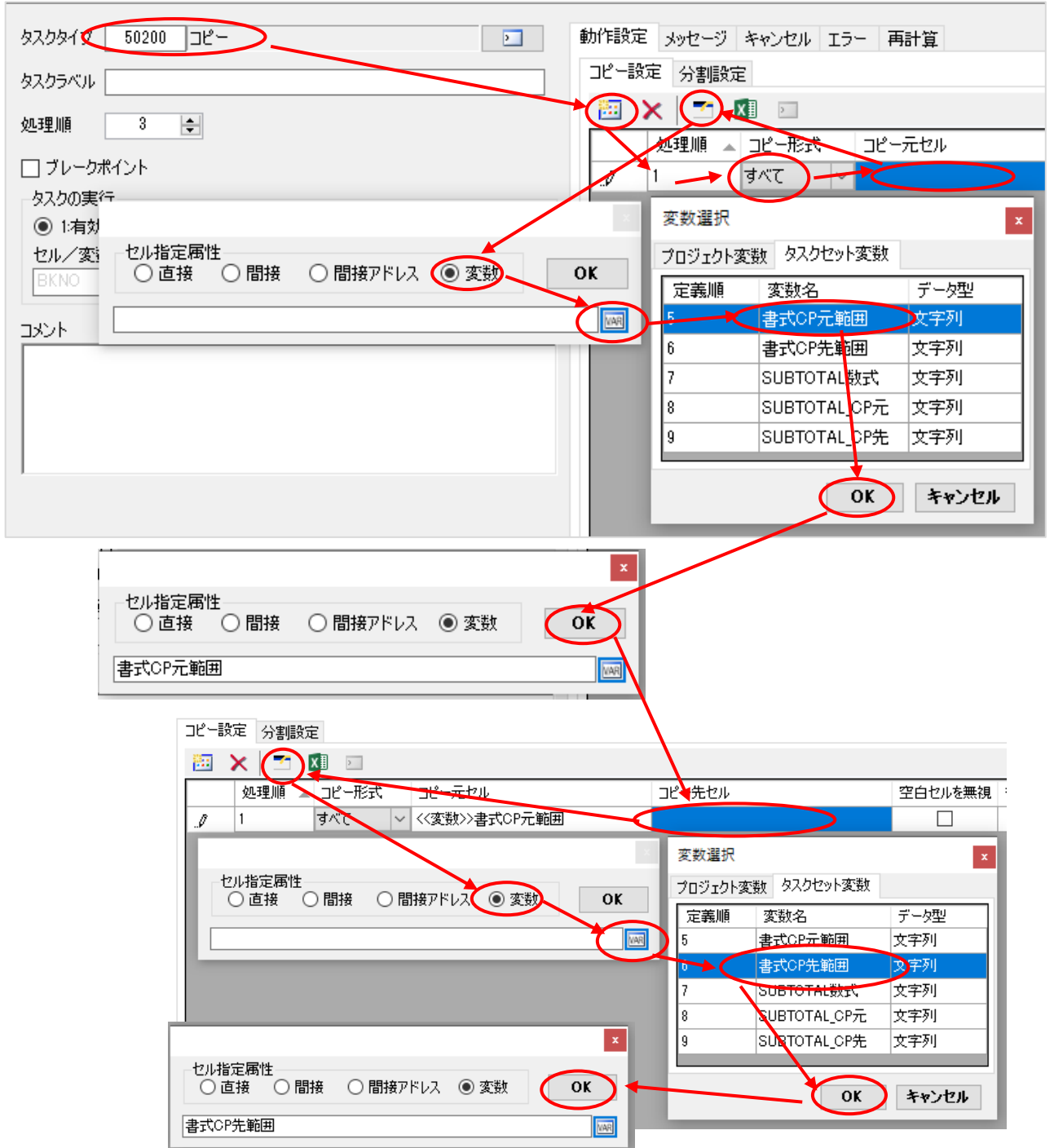
処理の目的: シート「予算入力画面」をコピーして作成します。



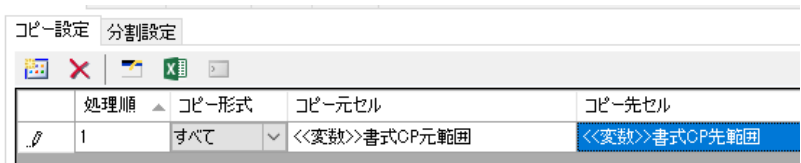
処理順: 3(コピー)

処理の目的: 検索データを展開する前に罫線など書式をコピーして設定します。

コピー元セル、コピー先セルに変数を利用します。(以下では定義済変数を選択して設定します)



以下の設定となります。



6.6.6 「読込」③

読込①でシート「予算テーブル」に展開したデータと同じデータをシート「予算入力画面」に展開します。展開方法として、既に展開済みのシート「予算テーブル」から「コピー」も考えられますが、タスク「SQL 展開」を利用します。

処理順: 4(SQL 展開)

タスク「SQL 展開」では、すでに dbSheetClient 開発版のシステム内部に保持しているデータから展開するため、Excel シートからコピーするより処理速度が速いです。

タスクタイプ: 10110 SQL展開

タスクラベル:

処理順: 4

動作設定: 基本設定 展開シート

展開するクエリー:

読込クエリーを実行するタスクを選択してください

- TASKNO1:ユーザー権限
 - ユーザー権限取得
 - 作成日取得
- TASKNO10110:予算テーブル 読込(ユーザー編集)
 - ProcessNo.3
- TASKNO31110:予算入力ユーザー一覧 読込
 - ProcessNo.3
- TASKNO3210:予算テーブル(管理者) 読込
 - ProcessNo.3
- TASKNO3310:予算入力確定テーブル(管理者) 読込
 - ProcessNo.3

SQL

```
--Select * From [予算テーブル]
Select * From 予算テーブル
where 作成日 = "BKNO!!作成日"
and 部署コード = "BKNO!!選択部署"
order by 前年度売上 desc
```

DBNO: 1:DBC作成 展開 リスト型

コメント

OK キャンセル

動作設定: 基本設定 展開シート

展開モード

全展開 列名を指定して展開する

列タイトルを出力する 列名の位置: 1 行上

展開開始セル位置

1:予算収集 予算入力画面!R4C2

展開範囲情報

セル範囲名:

展開時にセル範囲のクリアを行わない

展開方向

横方向(行) 縦方向(列)

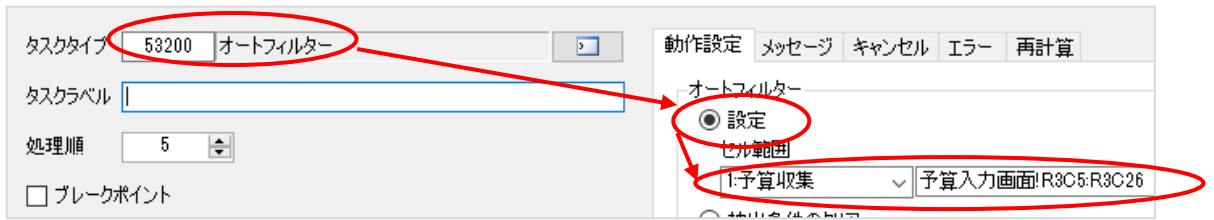
セル数式の保護

保護しない 保護する NULL値の場合のみ保護

自動でシート保護を解除しない dbS制御フィールドを出力しない

処理順:5(オートフィルター)

処理の目的: 検索データの展開後のデータに対してフィルター設定します。



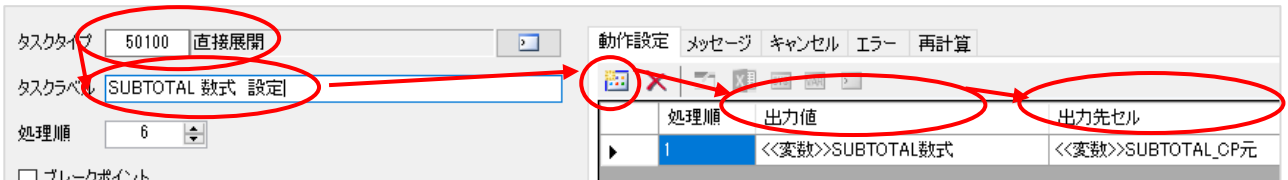
処理順:6(直接展開)

処理の目的: リスト上部に表示するフィルター結果の合計を計算する数式を設定します。

(1列分の上部のセル)

このセルをコピー元として、次の処理で複数列分の複数セルにコピーします。

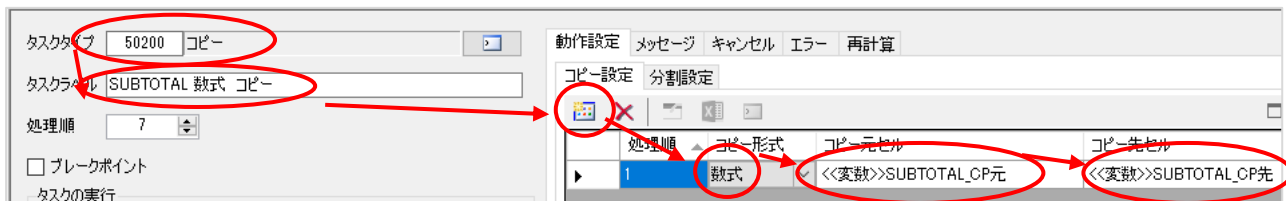
(変数を利用しています。変数の設定方法は、処理順:3を参照してください。)



処理順:7(コピー)

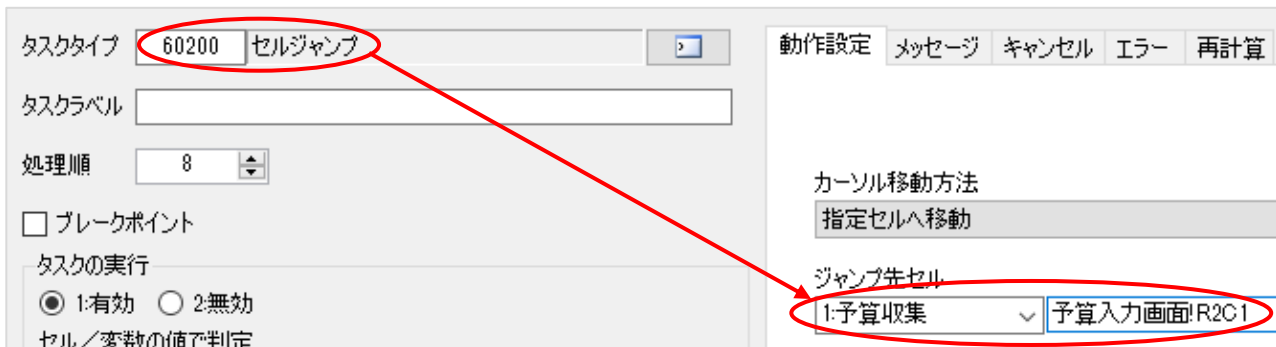
処理の目的: 処理順:6で設定した1列分の数式を複数セルにコピーします。

このセルをコピー元として、次の処理で複数列分の複数セルにコピーします。



処理順:8(セルジャンプ)

処理の目的: 前のコピー処理でセル選択状態がコピー先に残るため、セル位置を変更します。

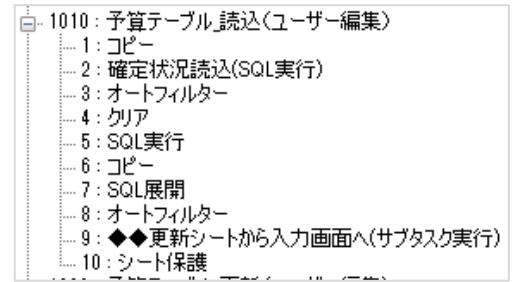


以上で TASKNO:99015 の作成は終了です。

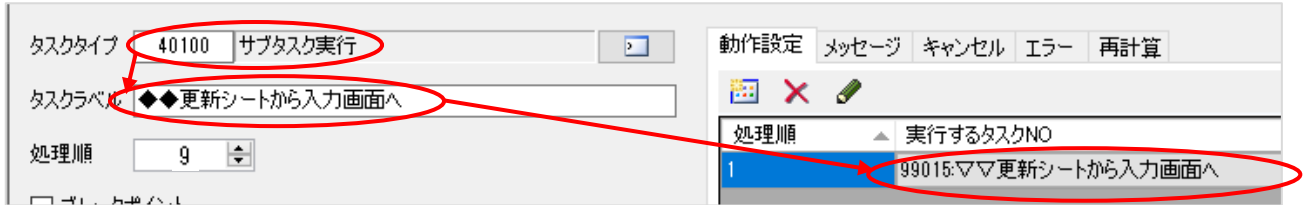
作成した TASKNO:99015 ([読込]②、③)を読込①(TASKNO:1010)から呼び出す設定をします。

TASKNO:1010(予算テーブル 読込(ユーザー編集))

処理順:9 以降のタスクを追加します。(右は追加後)

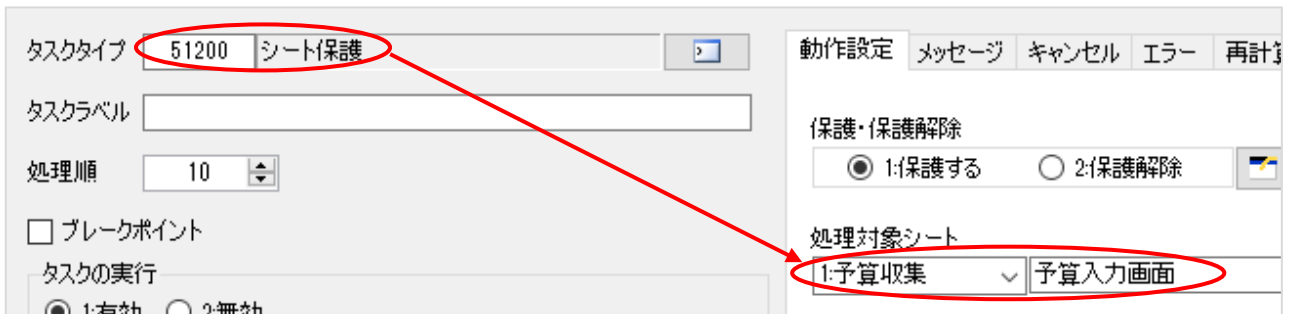


処理順:9(サブタスク実行)



処理順:10(シート保護)

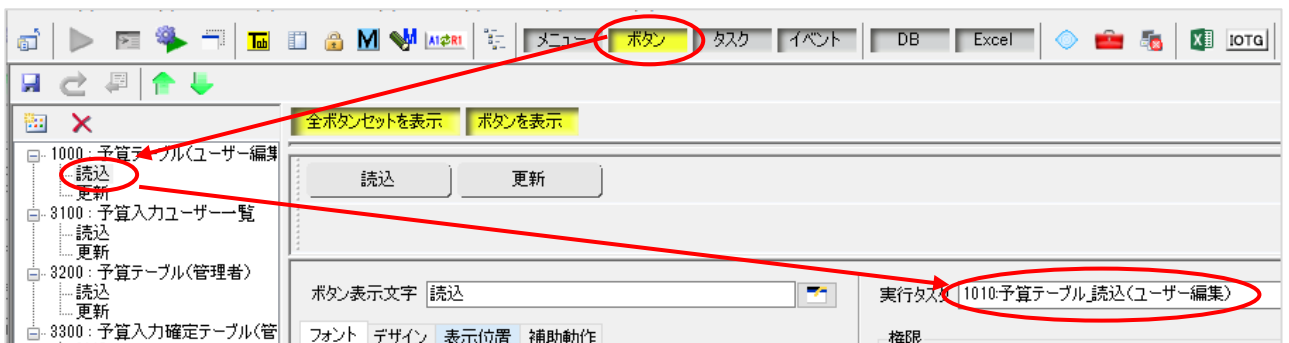
入力画面のシートの入力可能なセル以外を編集不可にするため、シート保護をします。



以上で「読込」①～③の処理は終了です。

6.6.7 ボタン「読込」の設定

メニュー「予算収集」で表示する「読込」ボタンを押下したとき、読込①～③ (TASKNO:1010)を実行するように定義します。



6.6.8 「更新」①

「更新」ボタンを押下した時の処理(概要説明の[更新]①、②)の作成をします。

更新①の処理はタスクセット(TASKNO:99025)を新規追加して設定します。

(「予算収集_完成」に合わせます。「予算収集_完成」では本処理をタスク、マクロのいずれかで処理できるようにしています。ここでは、タスクの処理(TASKNO:99025)の設定を行います。)

タスクセットを選択

(例:TASKNO:1010)後、

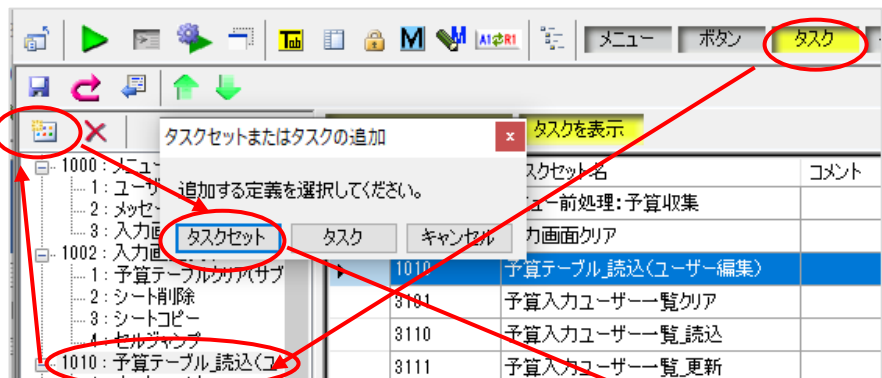
「追加」のボタンを押下

します。

(タスクセットが未選択

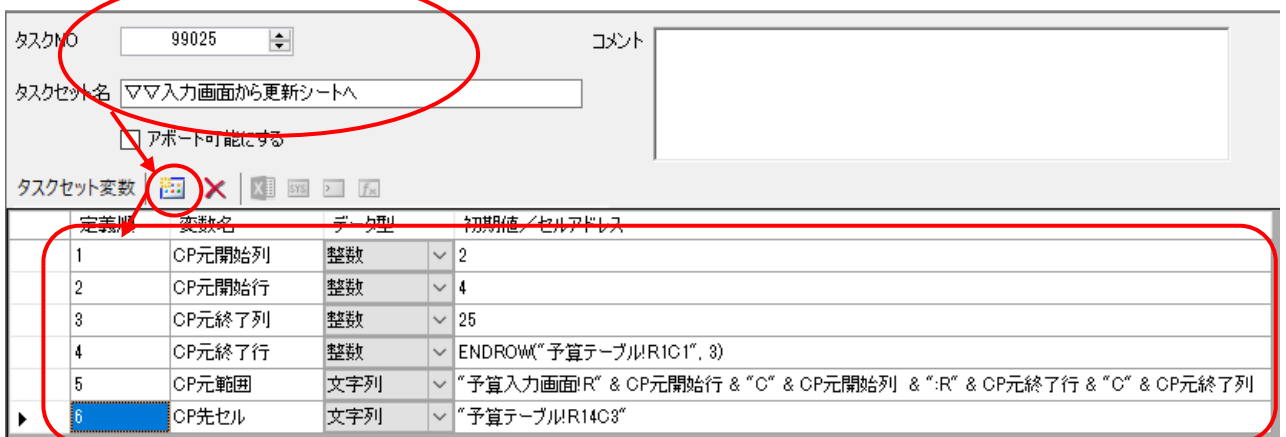
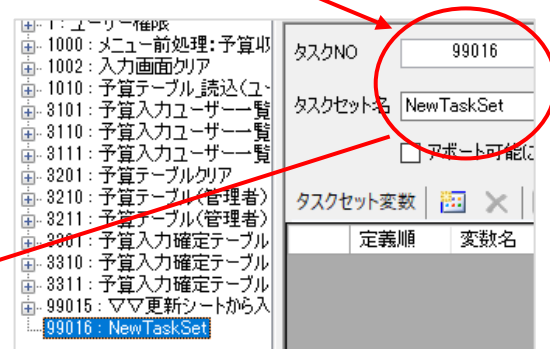
の場合は、タスクセットの

追加ができません)



追加した直後のタスクセット(右)を

タスク NO から設定をします。



タスクセット変数の「追加」のボタンで行追加し、6個のタスクセット変数を追加します。

(「予算収集_完成」から文字列の選択コピーができます)

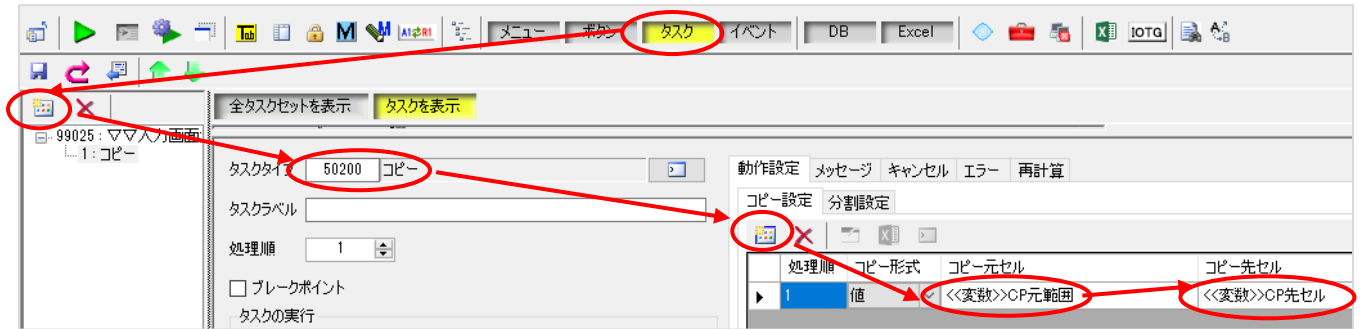
TASKNO:99025(▽▽更新シートから入力画面へ)

処理順: 1(コピー)



処理の目的: シート:予算入力画面から、シート:予算テーブルに更新データをコピーします。

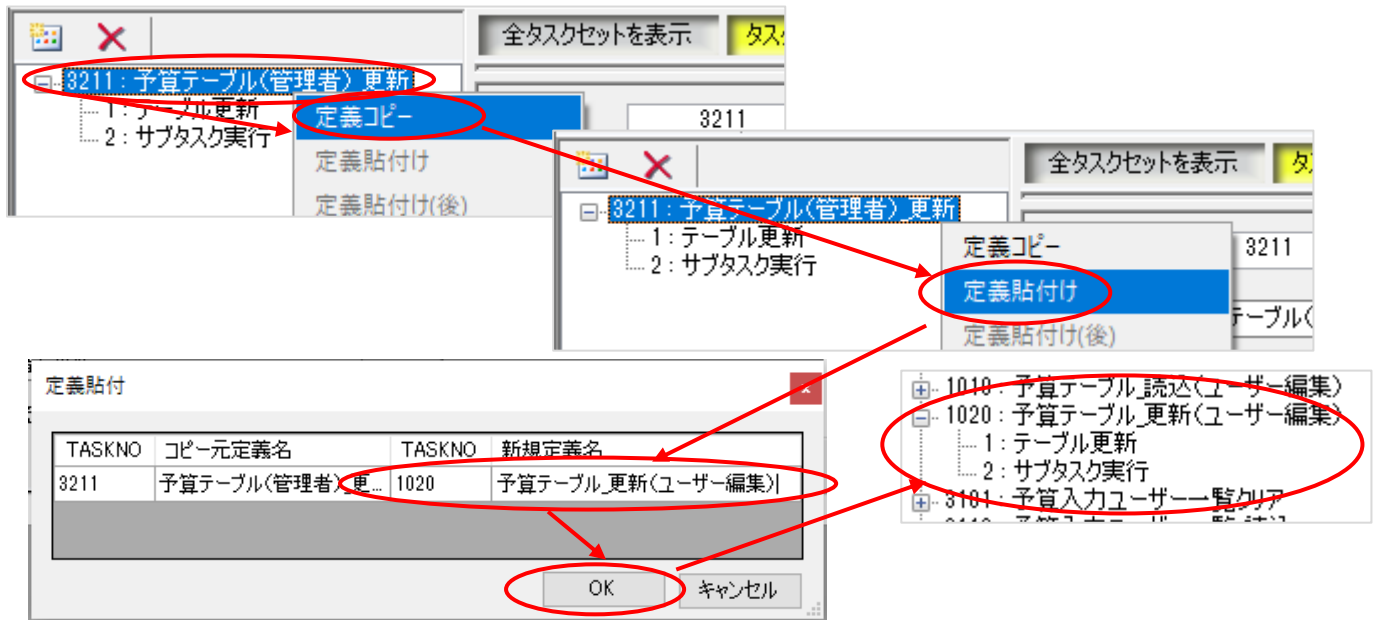
コピー元セル、コピー先セルに変数を利用します。(定義済の変数を選択して設定します)



6.6.9 「更新」②

更新②の処理を作成します。

IOTG で作成済の「予算テーブル」関連タスクセットをコピーし、「予算収集」用にカスタマイズします。



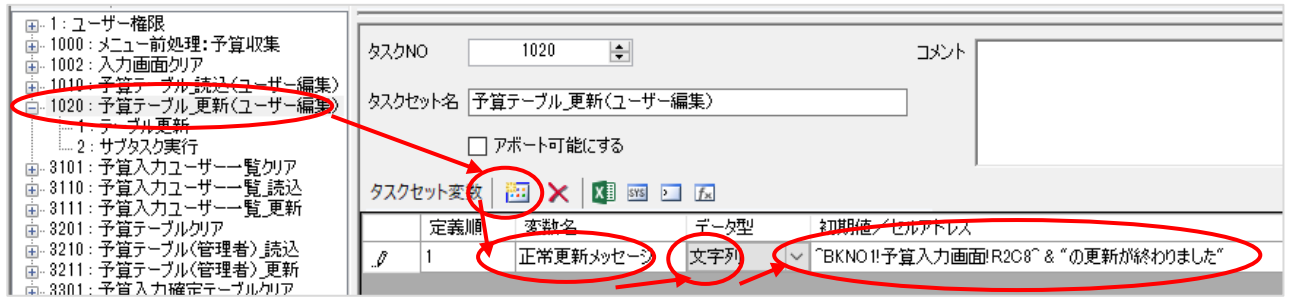
カスタマイズ前の TASKNO:1020 のタスクセットが作成されました。

このタスクセットに対して、以下の処理の追加、変更をします。

- ・処理順: 1(サブタスク実行) タスクを新規追加
- ・処理順: 2(テーブル更新) タスク内容を変更(元々の処理順:1)
- ・処理順: 3(サブタスク実行) タスク内容を変更(元々の処理順:2)

TASKNO:1020(予算テーブル 更新(ユーザー編集))

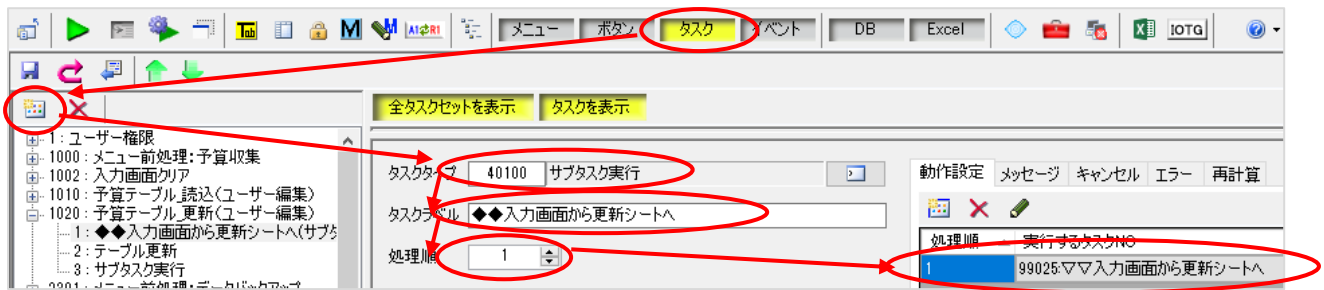
タスクセット変数を追加します。



処理順:1(サブタスク実行)

処理の目的: 作成済の TASKNO:99025 ([更新]①)を呼び出す設定をします。

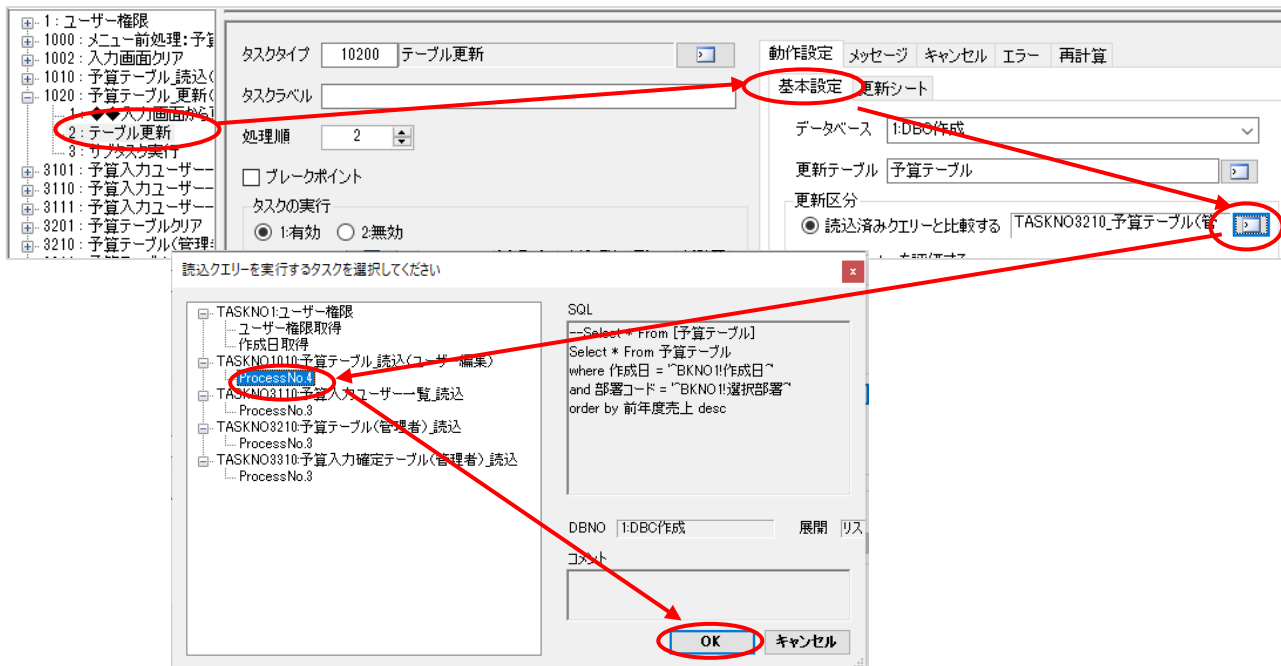
タスクセットの先頭(処理順:1)にタスクを新規追加して設定します。



DB を更新する処理の設定です。

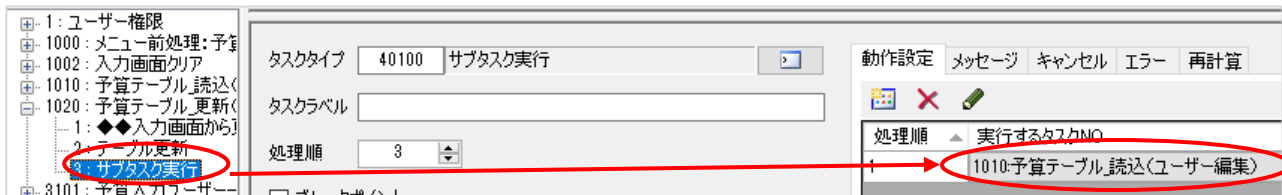
処理順:2(テーブル更新)

読込済みクエリーと比較する (TASKNO:3210) がタスクセットのコピー前のままであるので、(TASKNO:1010)に変更します。



処理順:3(サブタスク実行)

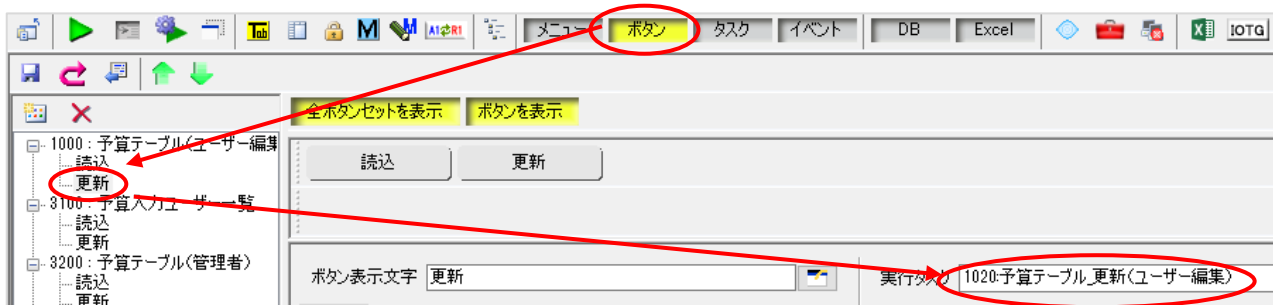
サブタスク実行(TASKNO:3210)がタスクセットのコピー前のもので、
(TASKNO:1010)に変更します。
更新後の再読み込みの処理です。



以上で「更新」①、②の処理は終了です。

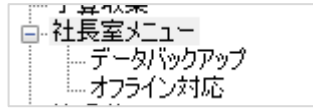
6.6.10 ボタン「更新」の設定

メニュー「予算収集」で表示する「更新」ボタンを押下したとき、更新①、②(TASKNO:1020)を実行するように定義します。

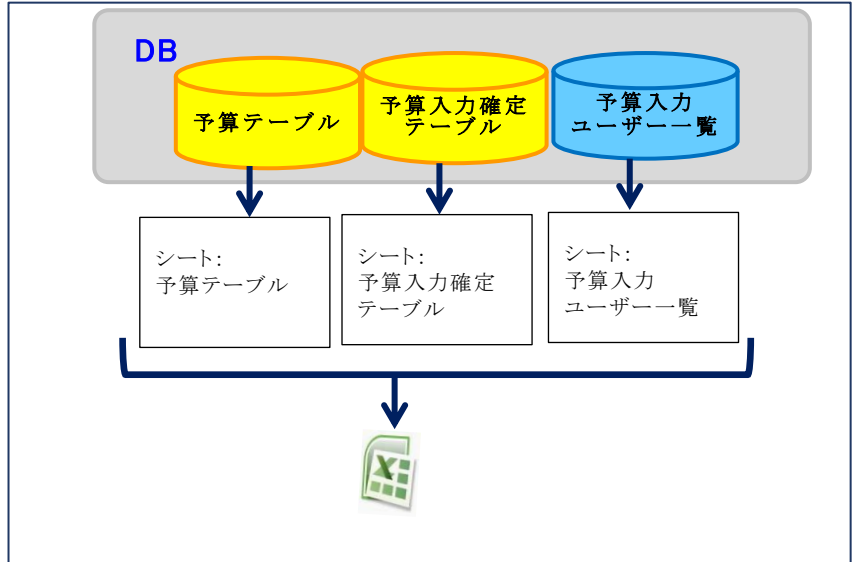


6.7 プロジェクト作成：社長室メニュー

作成する処理の概要は以下の通りです。

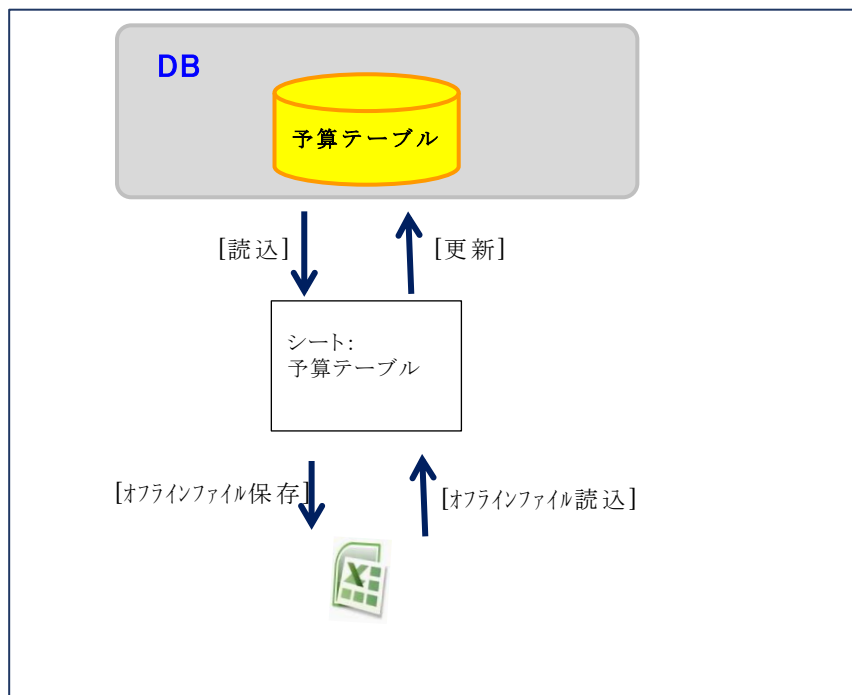


「データバックアップ」:
全支店データが対象です。



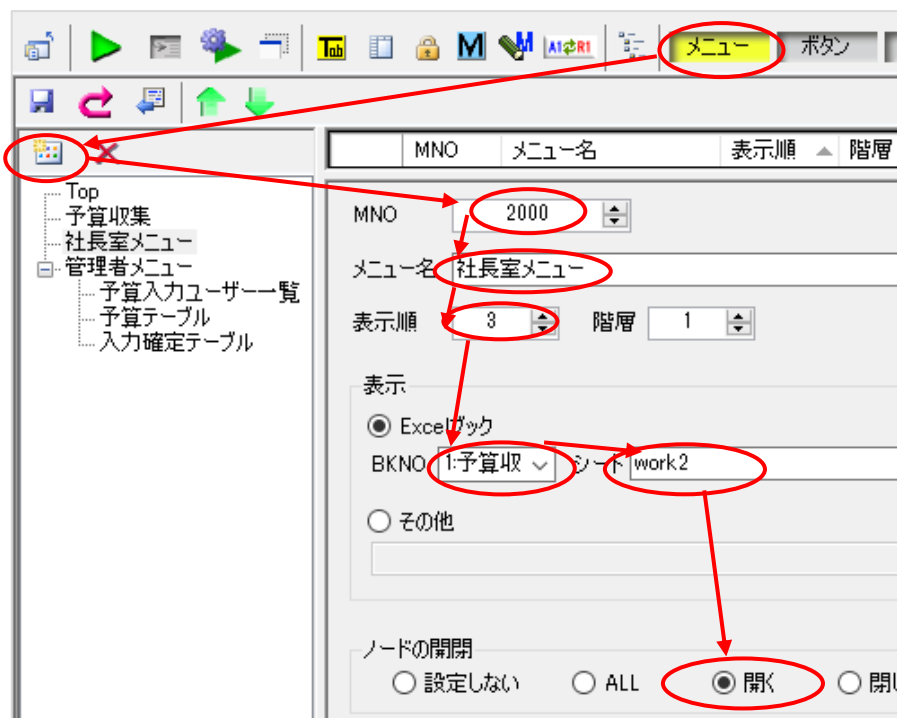
「オフライン対応」:
1 支店のデータが対象です。

「読込」、「更新」は
既存処理を利用します。



6.7.1 メニュー「社長室メニュー」の設定

メニュー「予算収集」を以下の手順で新規作成します。



6.7.2 メニュー「データバックアップ」の設定の前準備

メニュー「データバックアップ」の設定をする前に、メニュー選択時のタスクセット、ボタンの設定を行います。

タスクセットの追加は以下です。(以降で詳細を説明します)

追加 TASKNO	タスクセット名	概要
2301	メニュー前処理:データバックアップ	データバックアップの説明個所へセルジャンプ

ボタンの追加は以下です。

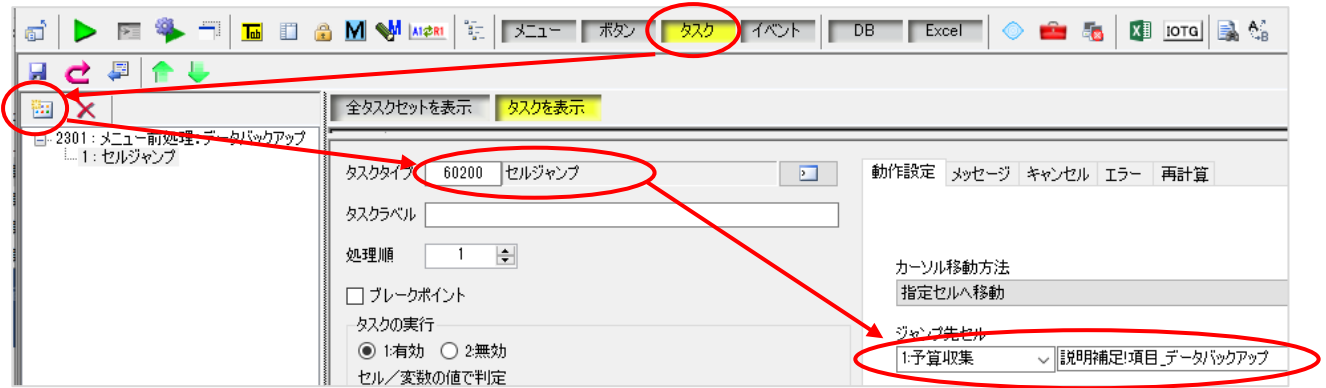
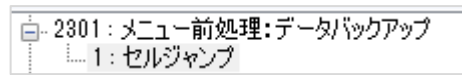
追加 BSNO	ボタンセット名	概要
2300	データバックアップ	ボタン「バックアップ実行」

TASKNO:2301(メニュー前処理:データバックアップ)

タスクセットを新規追加します。

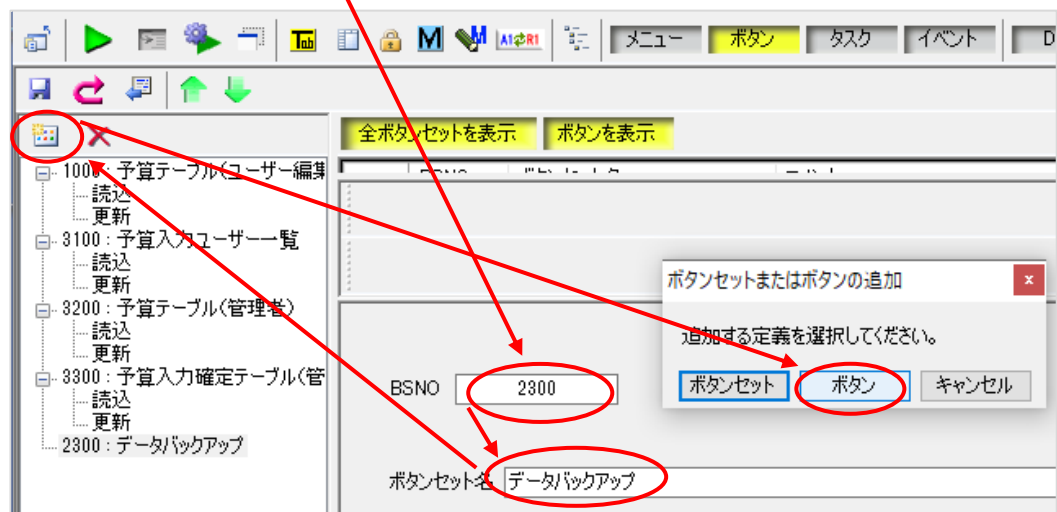
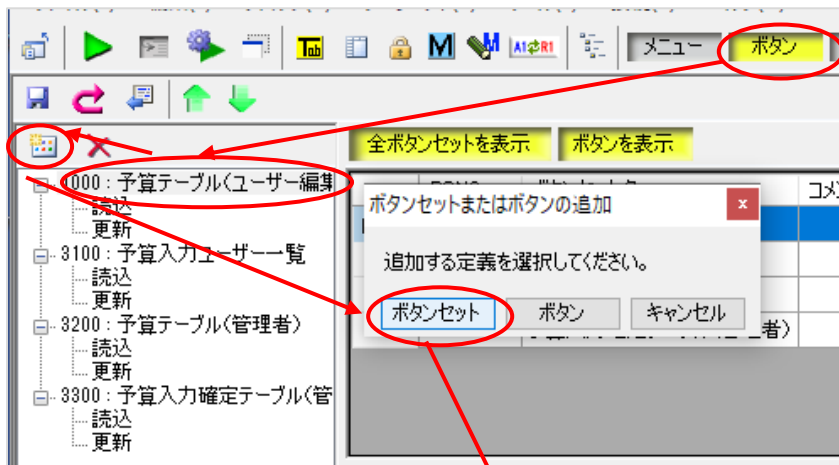
処理順:1(セルジャンプ)

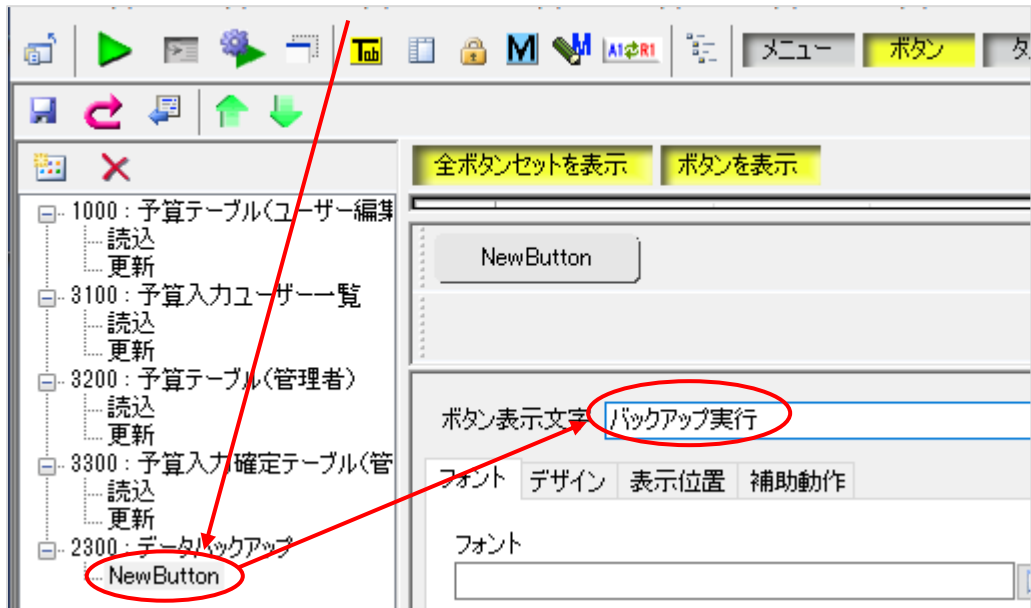
処理の目的: カレントセルをデータバックアップの説明の項目に移動します。



BSNO:2300(データバックアップ)

データバックアップのボタンセットを新規追加します。

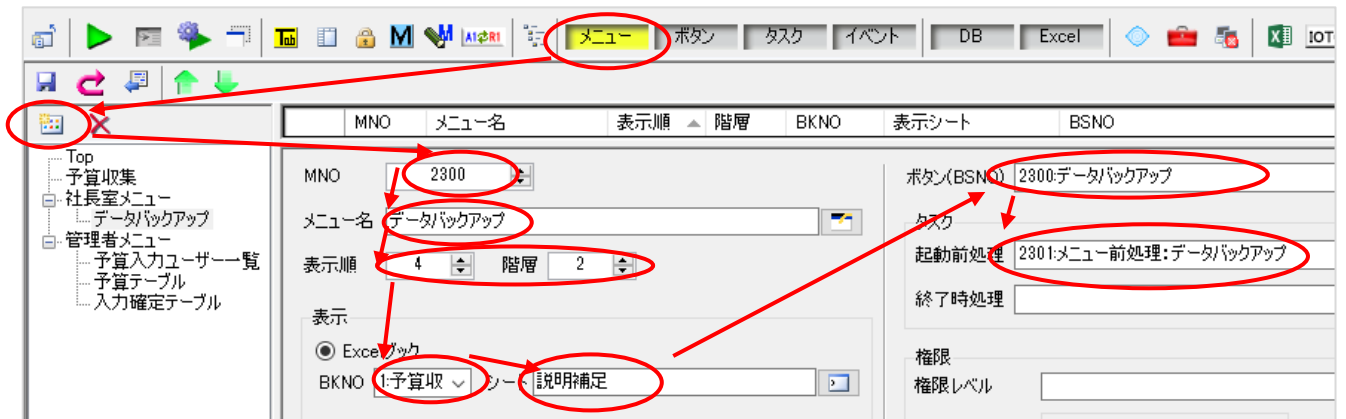




ボタン押下時の処理は後ほど設定します。

6.7.3 メニュー「データバックアップ」の設定

メニュー「データバックアップ」を以下の手順で新規作成します。

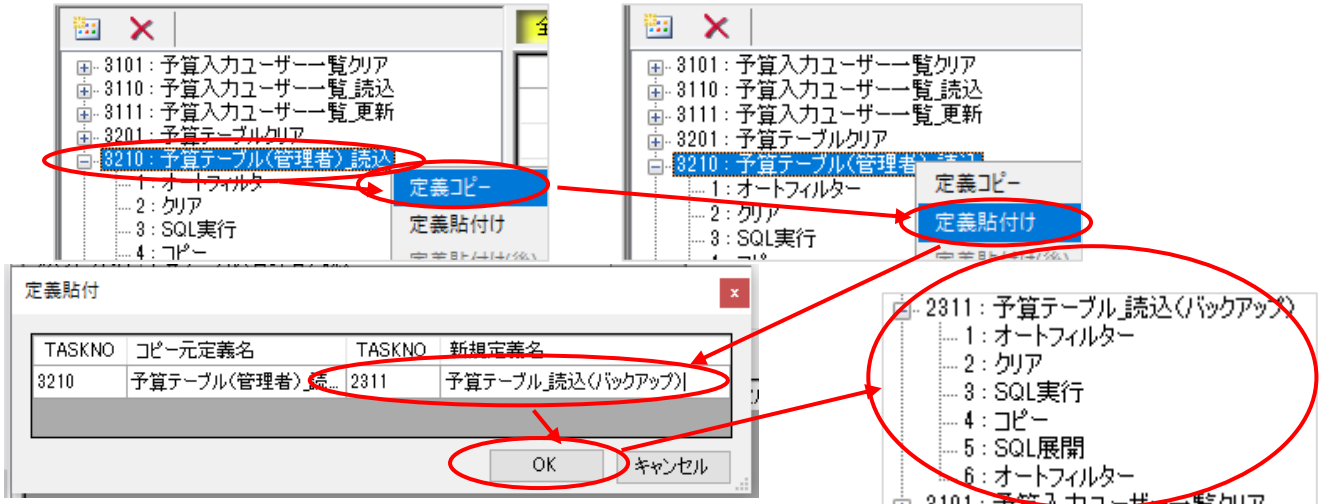


本設定で、メニュー「データバックアップ」を押下した時に初期表示するまでの処理の設定は終了です。

6.7.4 データバックアップ

「バックアップ実行」ボタンを押下した時の処理の作成に入ります。

IOTG で作成済の「予算テーブル」関連タスクセットをコピーし、「予算収集」用にカスタマイズします。



カスタマイズ前の TASKNO:2311 のタスクセットが作成されました。

このタスクセットに対して、以下の処理の変更をします。

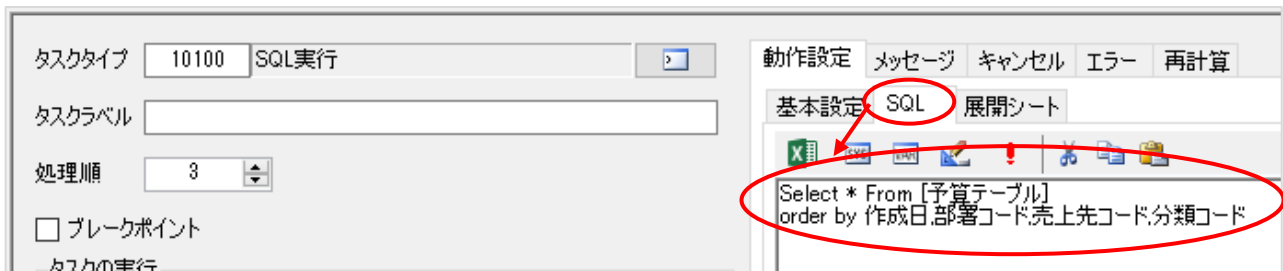
- ・処理順: 3 (SQL 実行) タスク内容を変更
- ・処理順: 5 (SQL 展開) タスク内容を変更

TASKNO: 2311(予算テーブル 読込(バックアップ))

処理順: 3(SQL 実行)

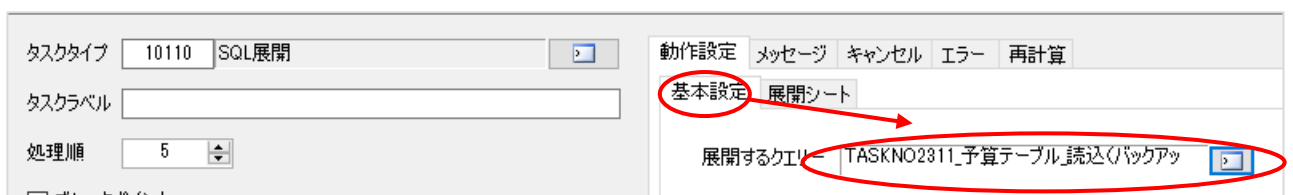
SQL 文に「order by 作成日,部署コード,売上先コード,分類コード」を追加します。

SQL 文はプロジェクト「予算収集_完成」からコピーできます。



処理順: 5(SQL 展開)

展開するクエリー (TASKNO: 3210) がタスクセットのコピー前のままであるので、(TASKNO: 2311)に変更します。

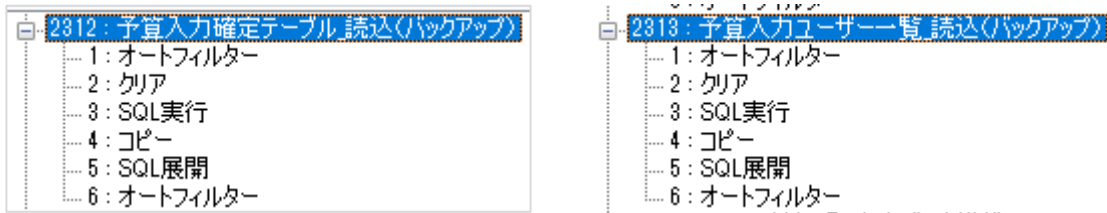


前記の「予算テーブル」同様に、「予算入力確定テーブル」、「予算入力ユーザー一覧」も IOTG のタスクセットのコピー、変更を同様に行います。

No	コピー元 TASKNO	新規 TASKNO	新規 タスク名	処理順:3(SQL 実行)の SQL 文 追加	処理順:5(SQL 展開)の展開する クエリー
①	3210	2311	予算テーブル_読込(バックアップ)	order by 作成 日,部署コード,売上先コード,分類コード	2311
②	3310	2312	予算入力確定テーブル_読込(バックアップ)	order by 作成 日,部署コード	2312
③	3110	2313	予算入力ユーザー一覧_読込(バックアップ)	(追加無し)	2313

No.①は、前記で作成しました。No.②、③を①に倣って作成します。

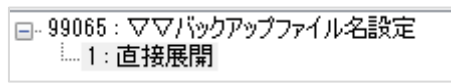
No.②、③の作成後のタスクセットは以下の通りとなります。



以上でバックアップのためのシートへのデータ展開の処理が作成できました。

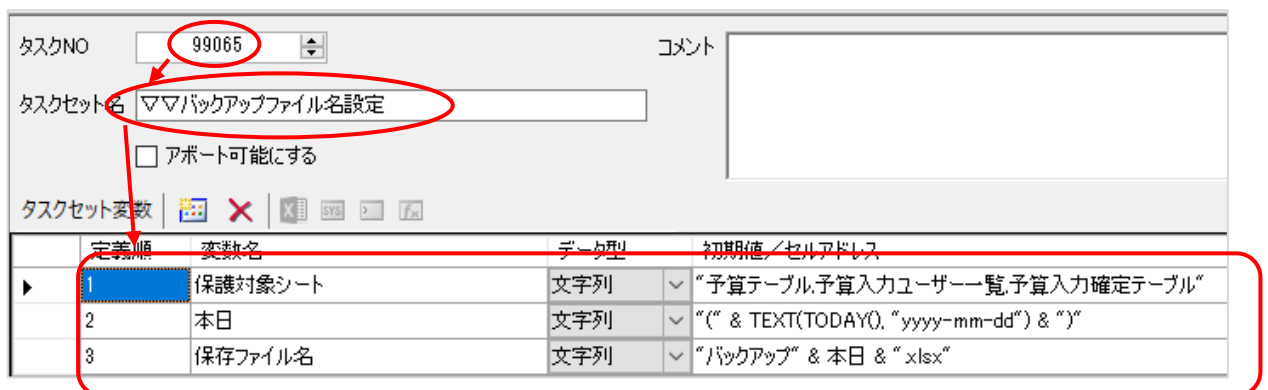
次にバックアップ時のファイル名を作成し、Excel にセルに設定する処理をタスクセットに定義します。

TASKNO:99065(▽▽バックアップファイル名設定)



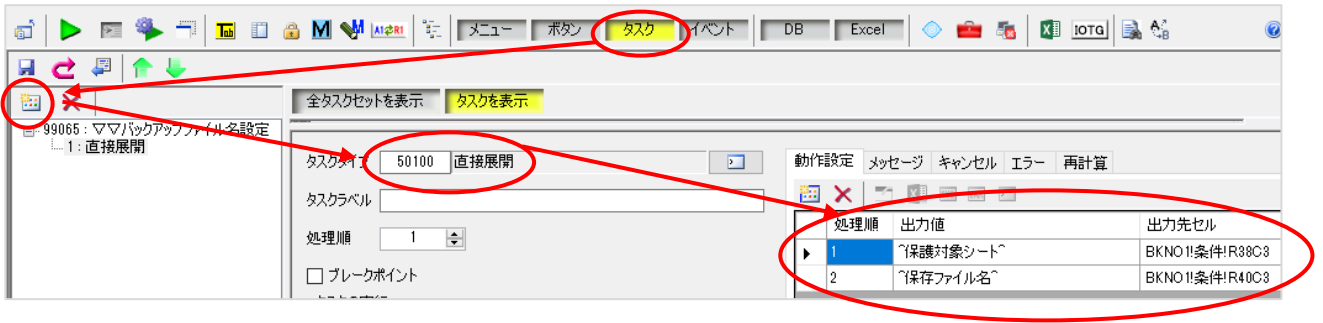
タスクセットを新規追加します。

タスクセット変数(3 個)を定義します。



処理順: 1(直接展開)

処理の目的: バックアップファイル名を Excel に展開します。



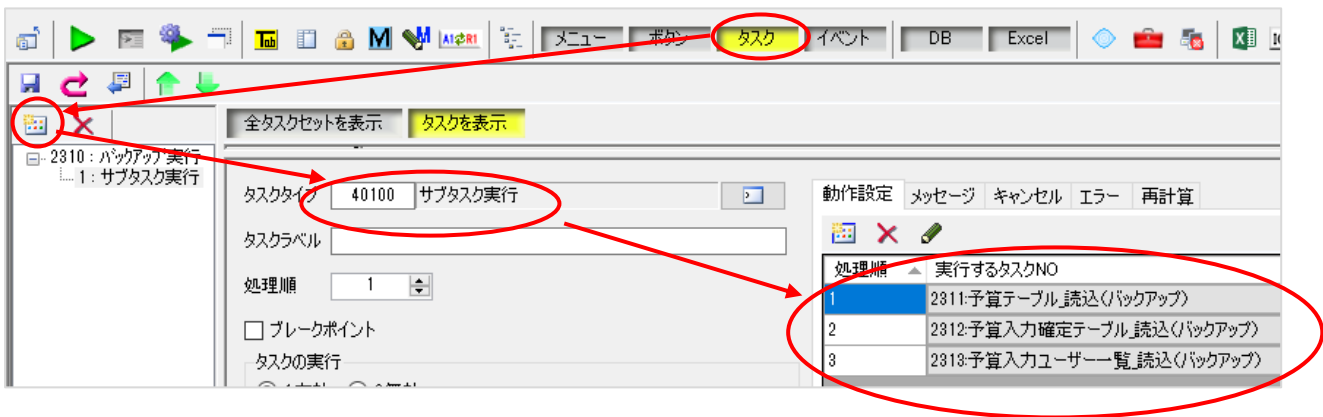
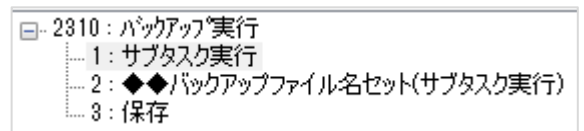
タスクセットを新規に作成して、バックアップ実行の処理を作成します。

TASKNO: 2310(バックアップ実行)

タスクセットを新規追加します。

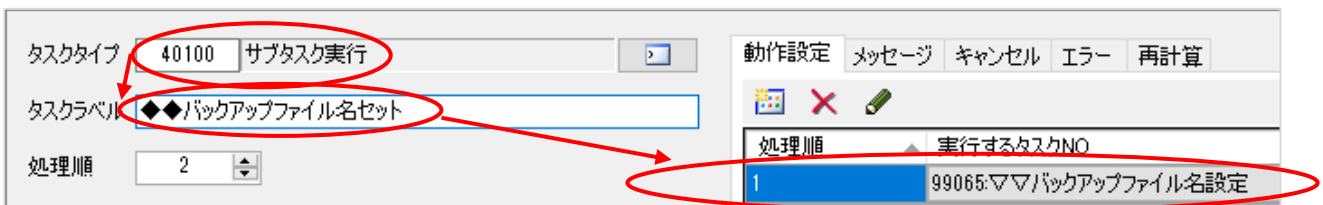
処理順: 1(サブタスク実行)

処理の目的: DB からバックアップするデータをシートに展開します。



処理順: 2(サブタスク実行)

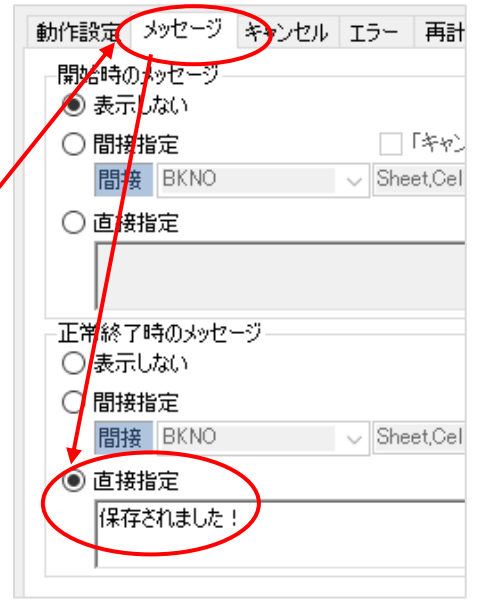
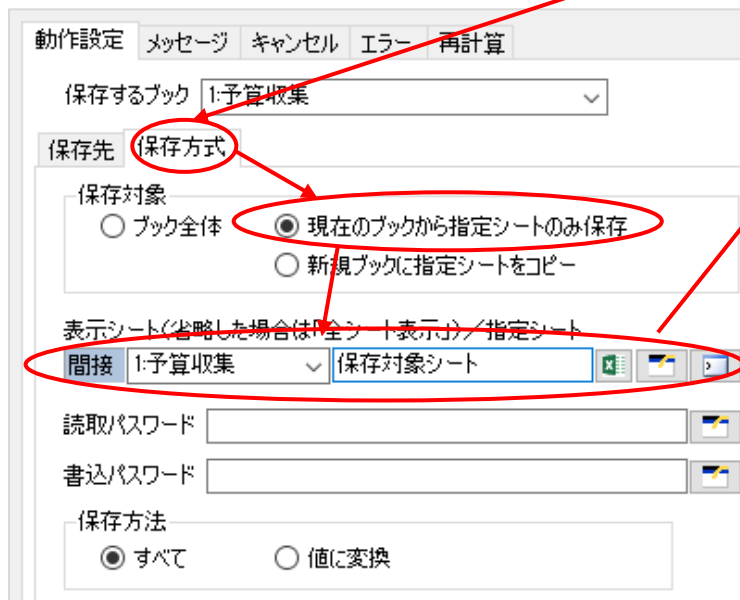
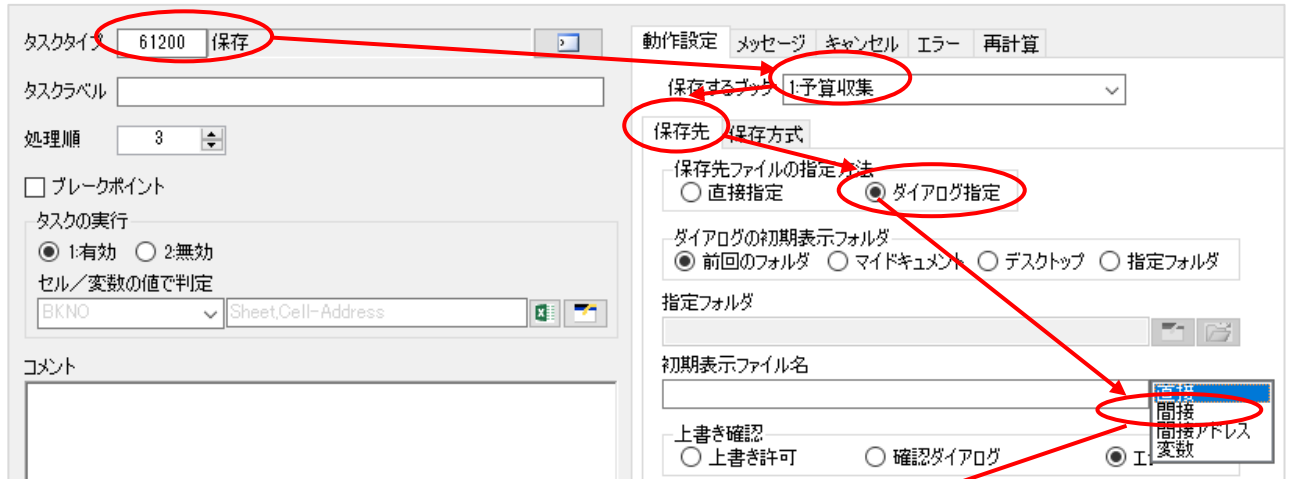
処理の目的: バックアップファイル名を Excel に設定します。



処理順: 3(保存)

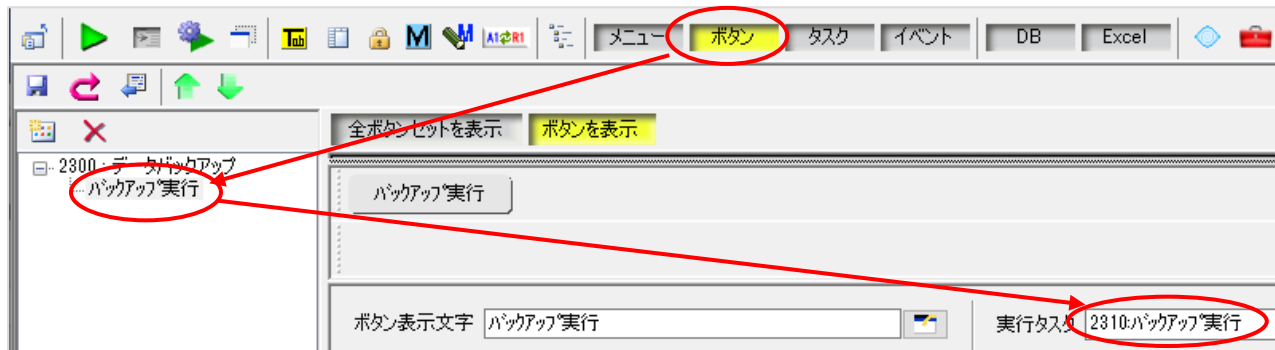
処理の目的: バックアップデータを設定したシートを Excel ファイルに保存します。

(初期表示ファイル名は「間接」を指定します。)



6.7.5 ボタン「バックアップ実行」の設定

ボタン押下時の処理を設定します。



以上でバックアップ実行の設定は終わりです。

6.7.6 メニュー「オフライン対応」の設定の前準備

メニュー「オフライン対応」の設定をする前に、メニュー選択時のタスクセット、ボタンの設定を行います。

タスクセットの追加は以下です。(以降で詳細を説明します)

追加 TASKNO	タスクセット名	概要
2501	メニュー前処理: オフライン対応	ユーザー情報取得 オフライン対応の説明個所へセルジャンプ

ボタンの追加は以下です。

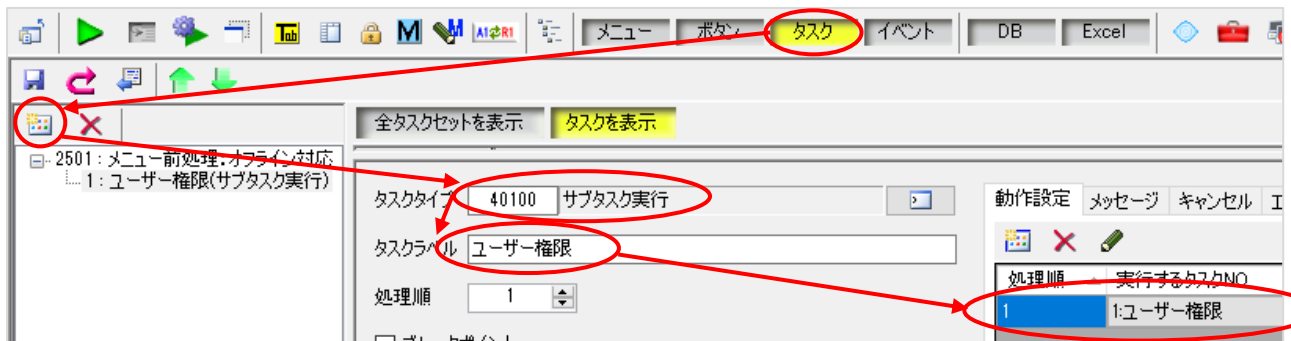
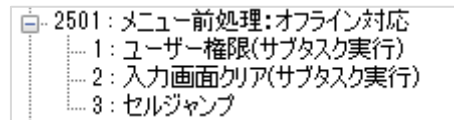
追加 BSNO	ボタンセット名	概要
2500	オフライン対応	ボタン「読込」、「更新」、「オフラインファイル保存」、「オフラインファイル読込」

TASKNO:2501(メニュー前処理:オフライン対応)

タスクセットを新規追加します。

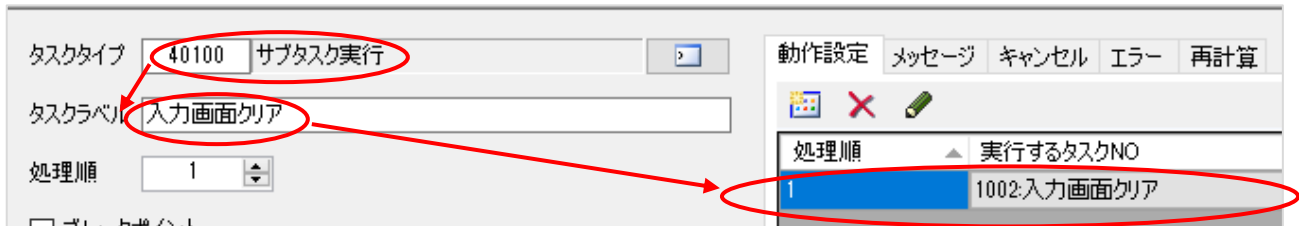
処理順:1(サブタスク実行)

処理の目的: 最新のユーザー情報を取得して、オフライン対応の準備をします。



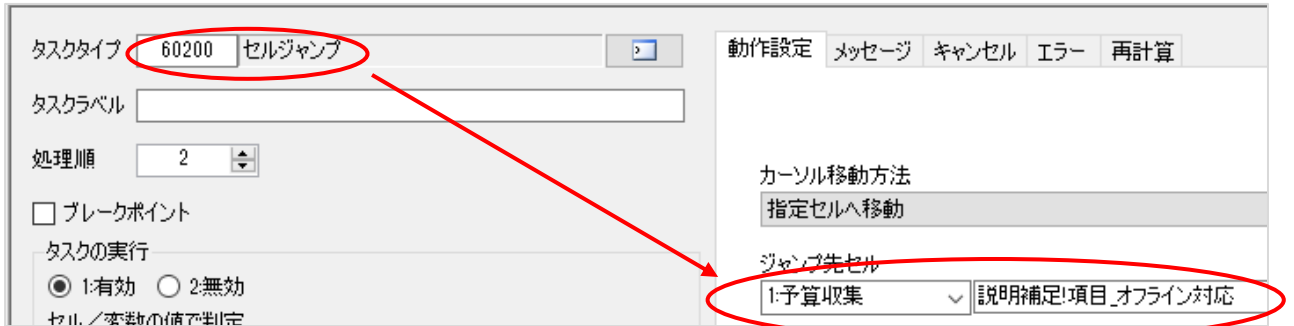
処理順:2(サブタスク実行)

処理の目的: 読込後に入力画面を表示するため、予め入力画面をクリアします。



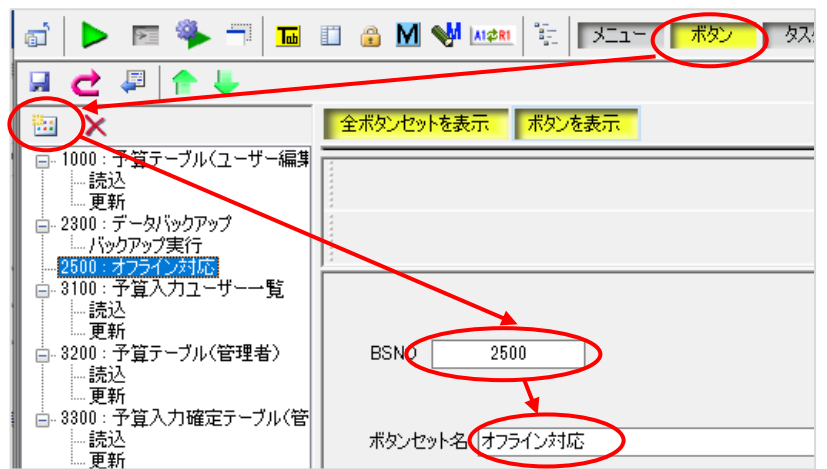
処理順: 3(セルジャンプ)

処理の目的: カレントセルをオフライン対応の説明の項目に移動します。



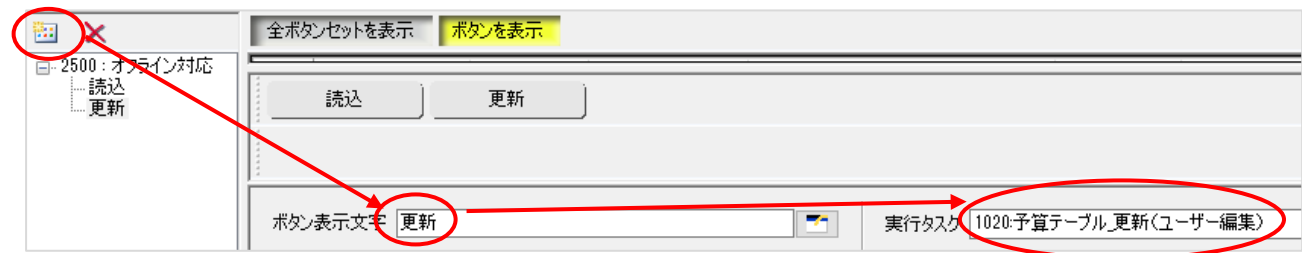
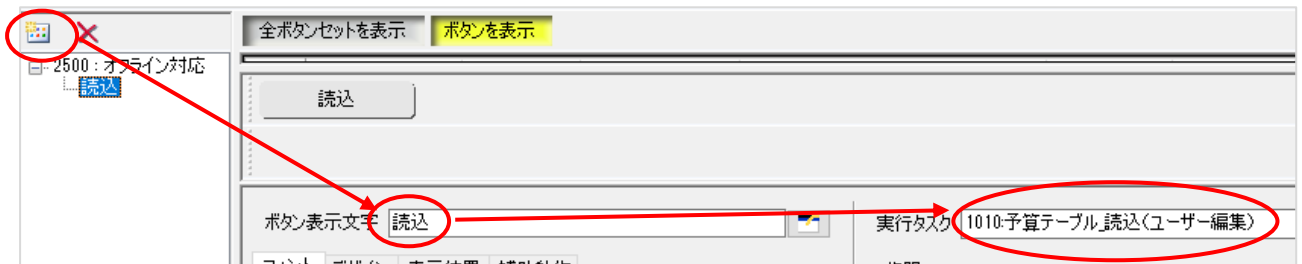
BSNO: 2500(オフライン対応)

ボタンセットを新規作成します。

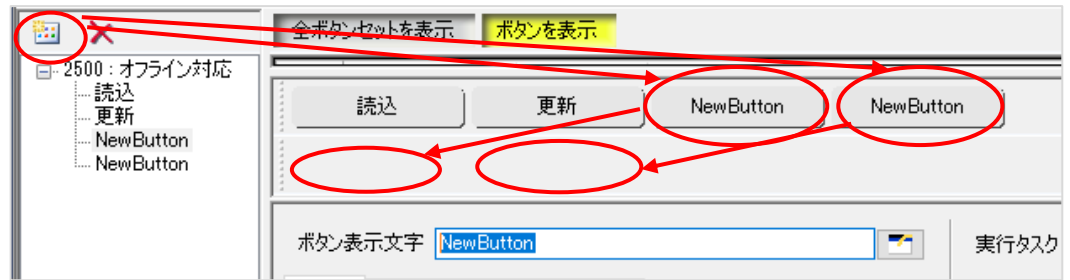


ボタン「読込」、「更新」を追加します。

各ボタンの実行タスクは予算収集と同じ処理のため、予算収集で利用したタスクを設定します。

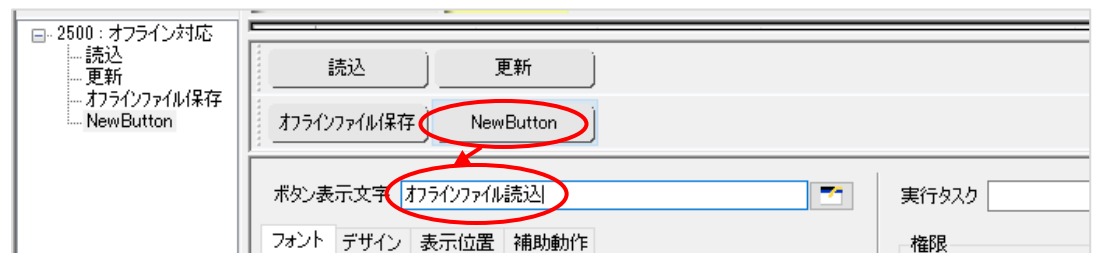
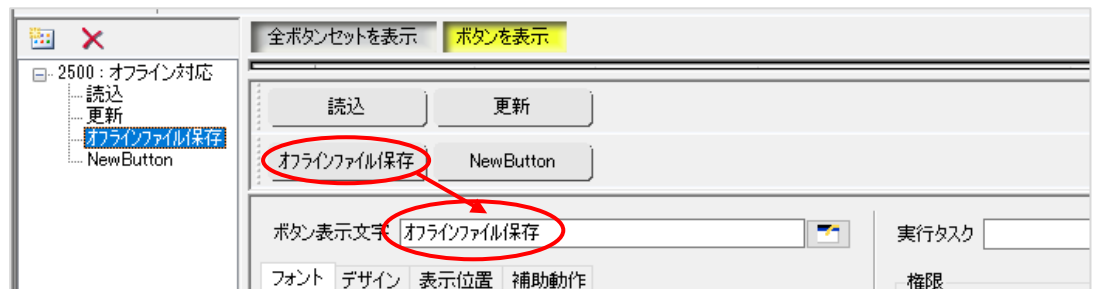


さらにボタンを 2 個追加し、個々 Drag&Drop して移動します。



移動したボタンのボタン名「オフラインファイル保存」、「オフラインファイル読込」の定義をします。

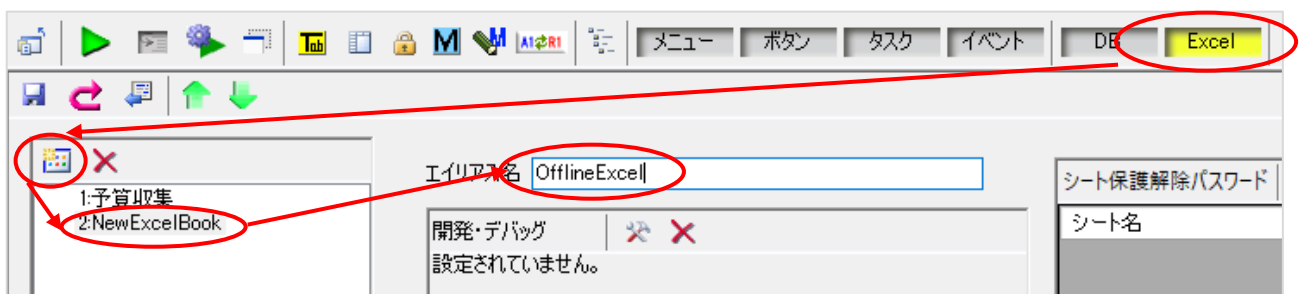
(ボタン押下時の実行タスクは、後ほど設定します)



また、Excel 定義を追加します。

追加する Excel 定義はオフラインファイル読込時に参照する Excel ファイルに割り当てします。

dbSheetClient のタスクで Excel 操作をするためには、Excel 定義が必要です。



エイリアス名「OfflineExcel」として定義します。

Excel ファイル本体(ファイル名)の定義は行いません。

オフラインファイル読込の処理でユーザーが選択した読込ファイルをこの Excel 定義に割り当てします。

そのようにすることで、ユーザーが選択した Excel を dbSheetClient のタスクで操作可能となります。

6.7.7 メニュー「オフライン対応」の設定

メニュー「オフライン対応」を以下の手順で新規作成します。



本設定で、メニュー「オフライン対応」を押下した時に初期表示するまでの処理の設定は終了です。

6.7.8 オフラインファイル保存

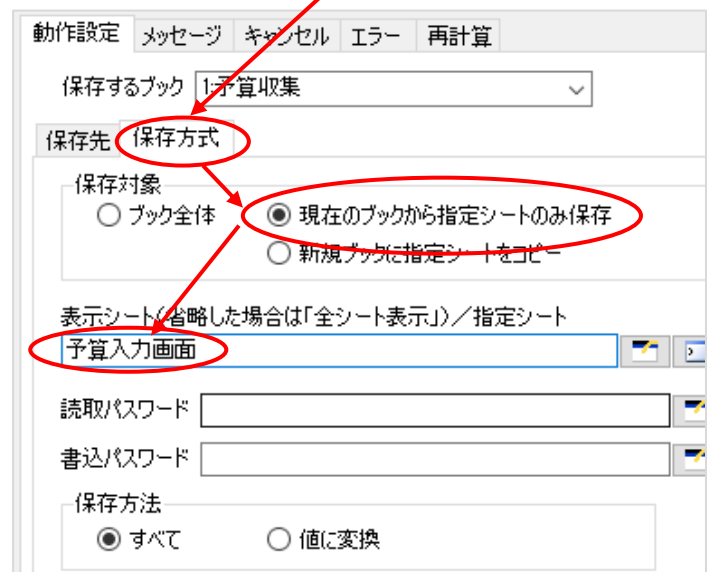
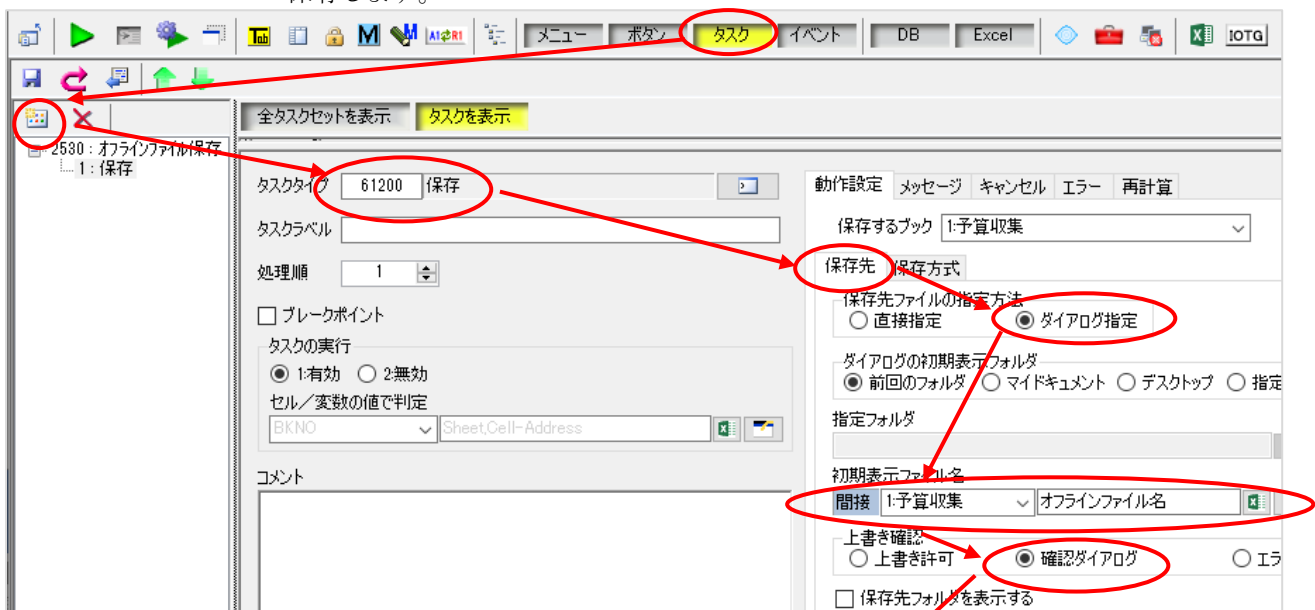
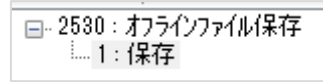
タスクセットを新規に作成して、オフラインファイル保存の処理を作成します。

TASKNO:2530(オフラインファイル保存)

タスクセットを新規追加します。

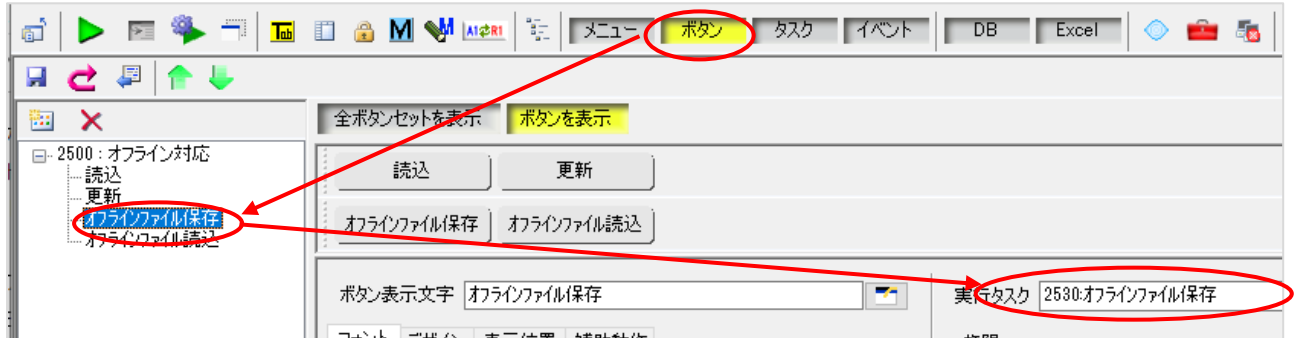
処理順:1(保存)

処理の目的: ダイアログを表示して指定したファイルにシート: 予算入力画面を保存します。



6.7.9 ボタン「オフラインファイル保存」の設定

「オフラインファイル保存」ボタンを押下したとき、TASKNO:2530 を実行するように定義します。



6.7.10 オフラインファイル読込

タスクセットを新規に作成して、オフラインファイル読込の処理を作成します。

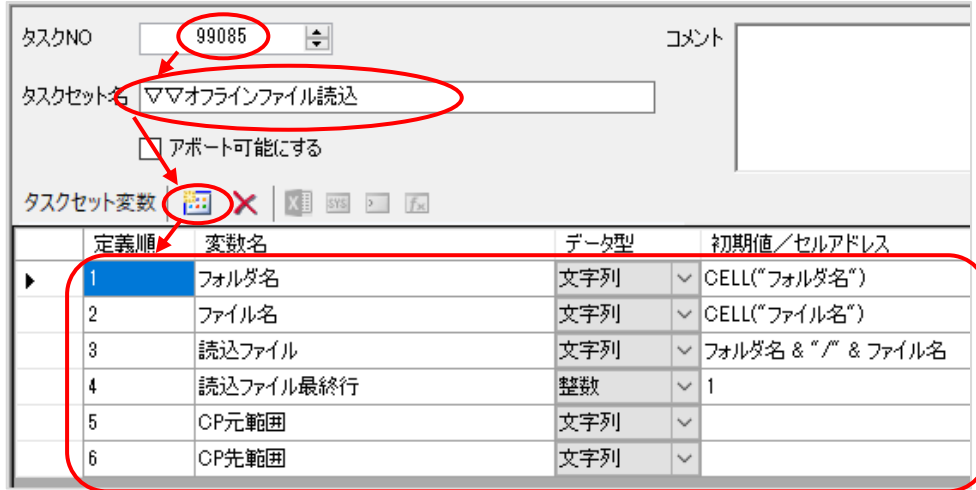
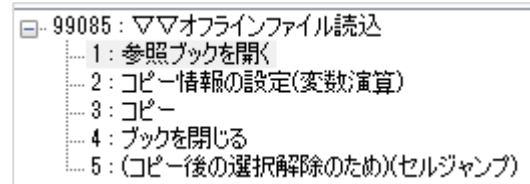
以下 2 個のタスクセットを作成します。(以降で詳細を説明します)

追加 TASKNO	タスクセット名	概要
99085	▽▽オフラインファイル 読込	参照ブックを利用してオフラインファイルから読込
2540	オフラインファイル読込	読込オフラインファイルの選択と TASKNO:99085 の呼び出し

TASKNO:99085(▽▽オフラインファイル読込)

タスクセットを新規追加します。

タスクセット変数(6 個)も定義します。



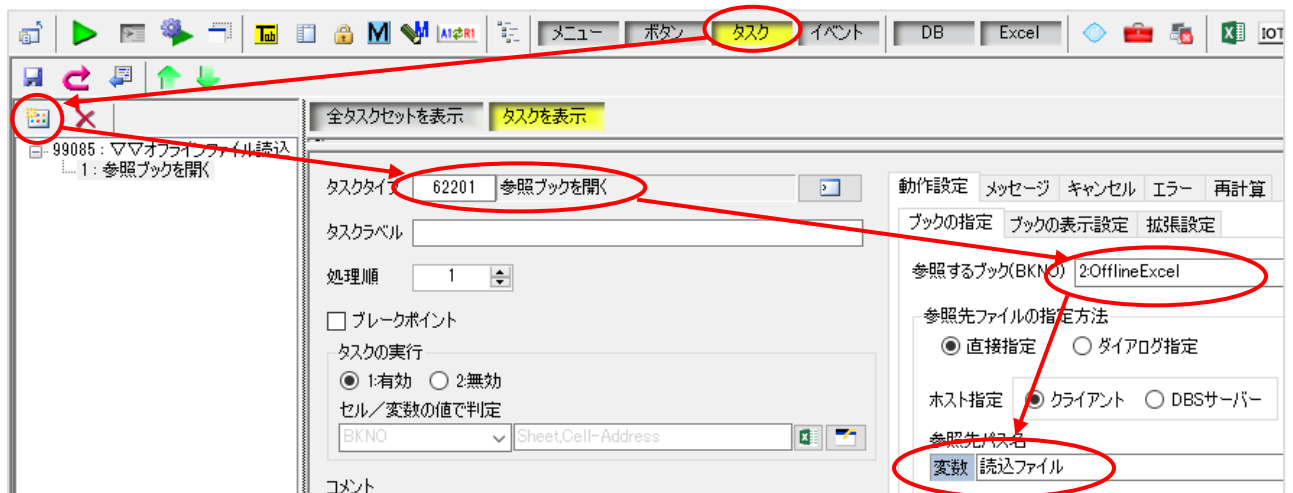
定義順:1~3 で Excel からフォルダ名、ファイル名を取得して、「読込ファイル」にフルパスを設定しています。

定義順:4~6 は、タスクセット内の処理順:2(変数演算)以降で利用します。

処理順:1(参照ブックを開く)

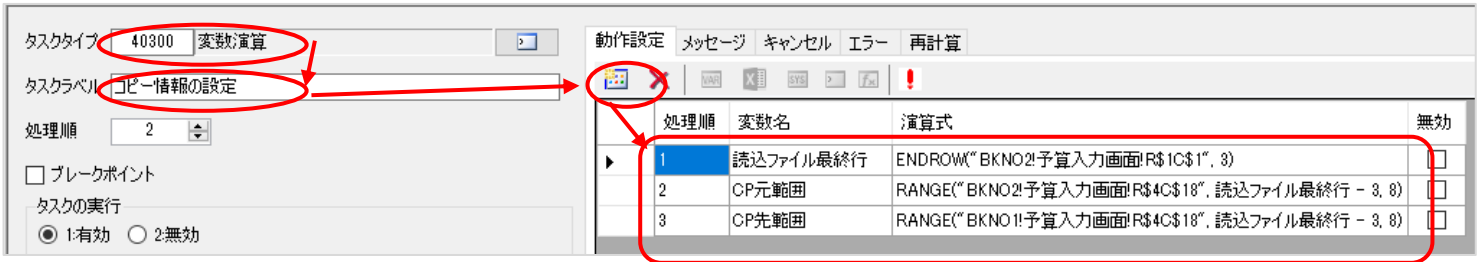
処理の目的: 変数「読込ファイル」に設定された Excel を参照ブックとして開きます。

「読込ファイル」がタスクで操作が可能となります。



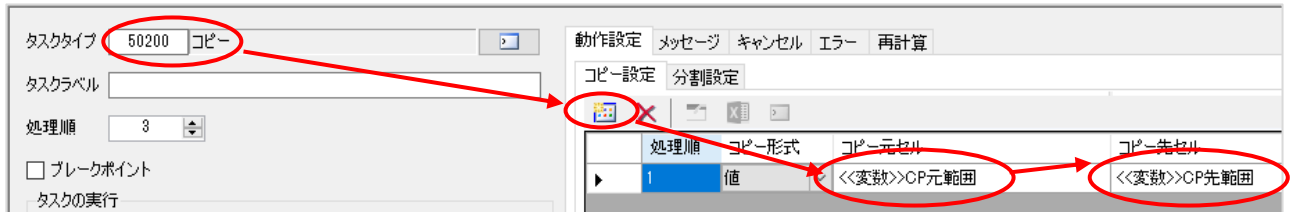
処理順: 2(変数演算)

処理の目的: 変数を利用して参照ブックの更新データのコピーする範囲を計算します。



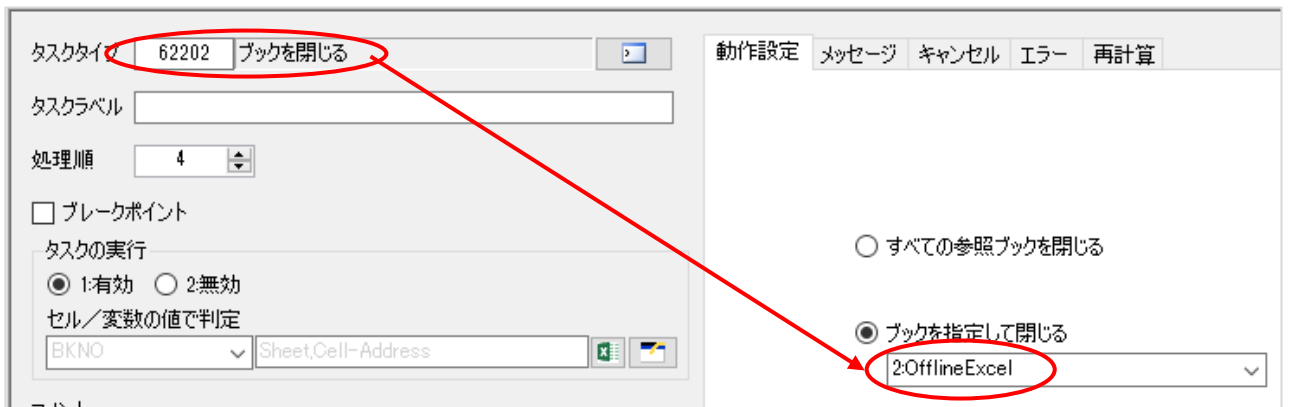
処理順: 3(コピー)

処理の目的: 読込ファイルから更新データを表示している予算入力画面のシートにコピーします。(読込ファイルのデータを画面表示)



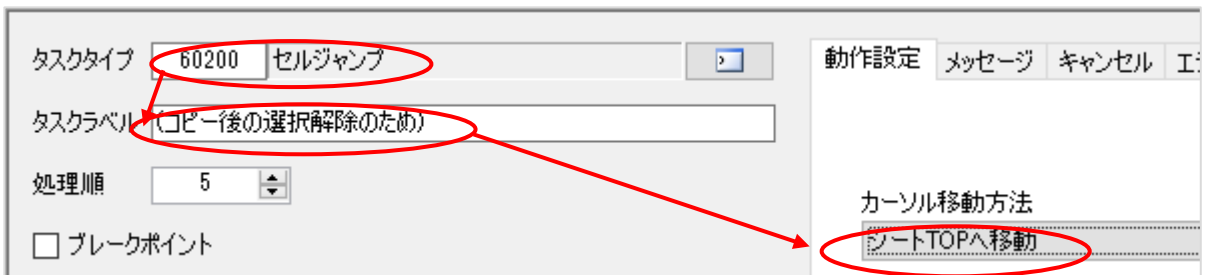
処理順: 4(ブックを閉じる)

処理の目的: 「読込」で開いた参照ブックを閉じます。



処理順: 5(セルジャンプ)

処理の目的: コピー後のコピー先範囲が選択状態のままであるので、選択解除のためにセルジャンプを行います。



TASKNO: 2540(オフラインファイル読込)

新規にタスクセットを作成します。

- 2540: オフラインファイル読込
 - 1: ファイル選択ダイアログ
 - 2: ◆◆オフラインファイル読込(サブタスク実行)
 - 3: シート切替

定義順	変数名	データ型	初期値/セルアドレス
-----	-----	------	------------

処理順: 1(ファイル選択ダイアログ)

処理の目的: ファイル選択ダイアログを表示して読込ファイルを選択します。

選択したファイルのフォルダ名、ファイル名は Excel に出力します。

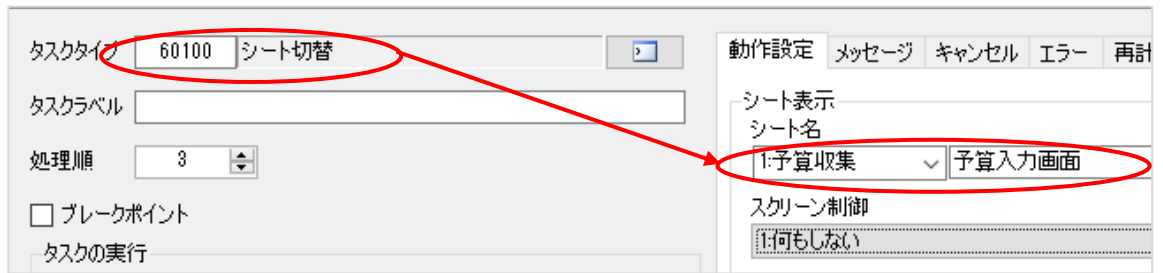
処理順: 2(サブタスク実行)

処理の目的: 選択済のオフラインファイルを読込し、予算入力画面シートに設定します。

処理順	実行するタスクNO
1	99085:▽▽オフラインファイル読込

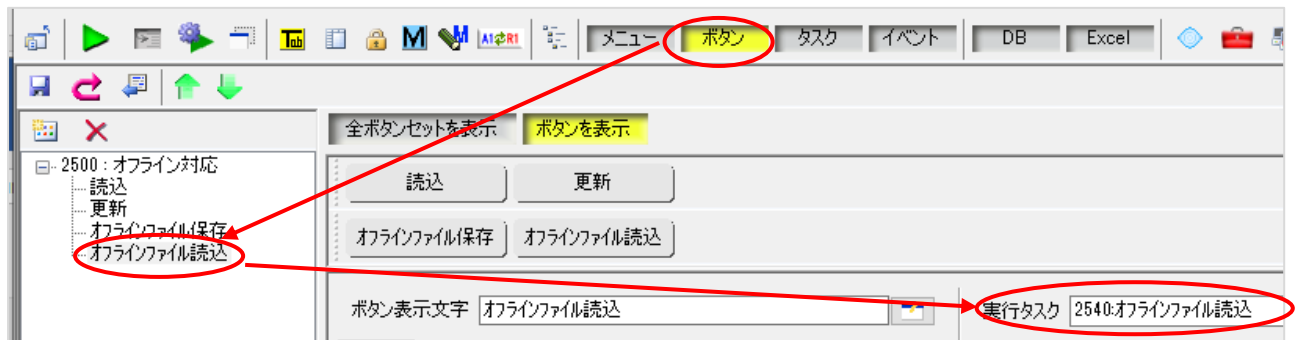
処理順: 3(シート切替)

処理の目的: 読込データを反映した予算入力画面シートを表示します。



6.7.11 ボタン「オフラインファイル読込」の設定

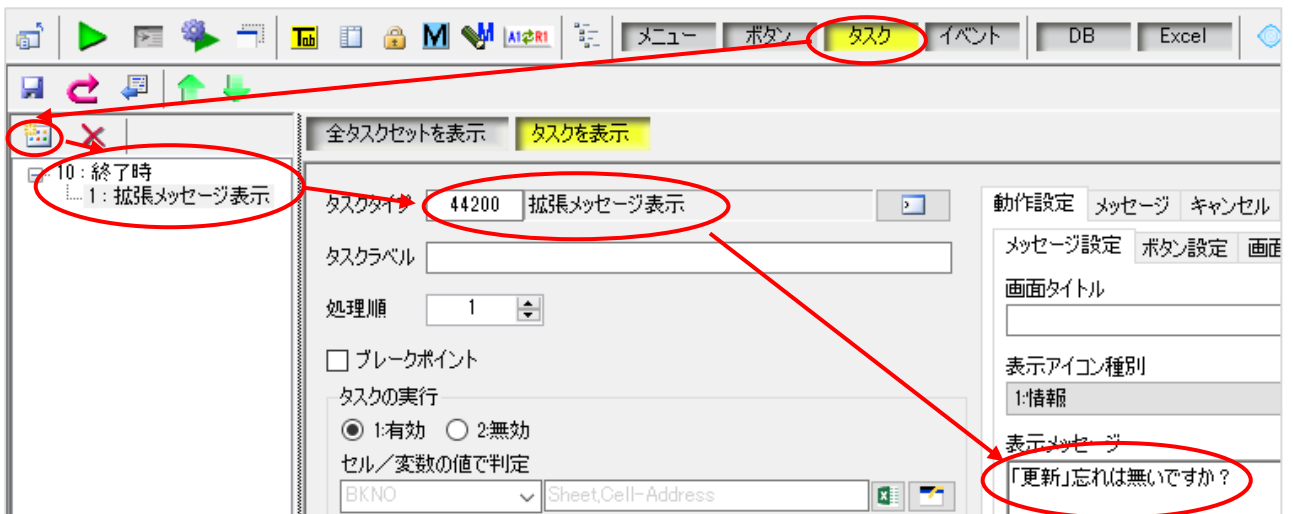
「オフラインファイル読込」ボタンを押下したとき、TASKNO:2540 を実行するように定義します。



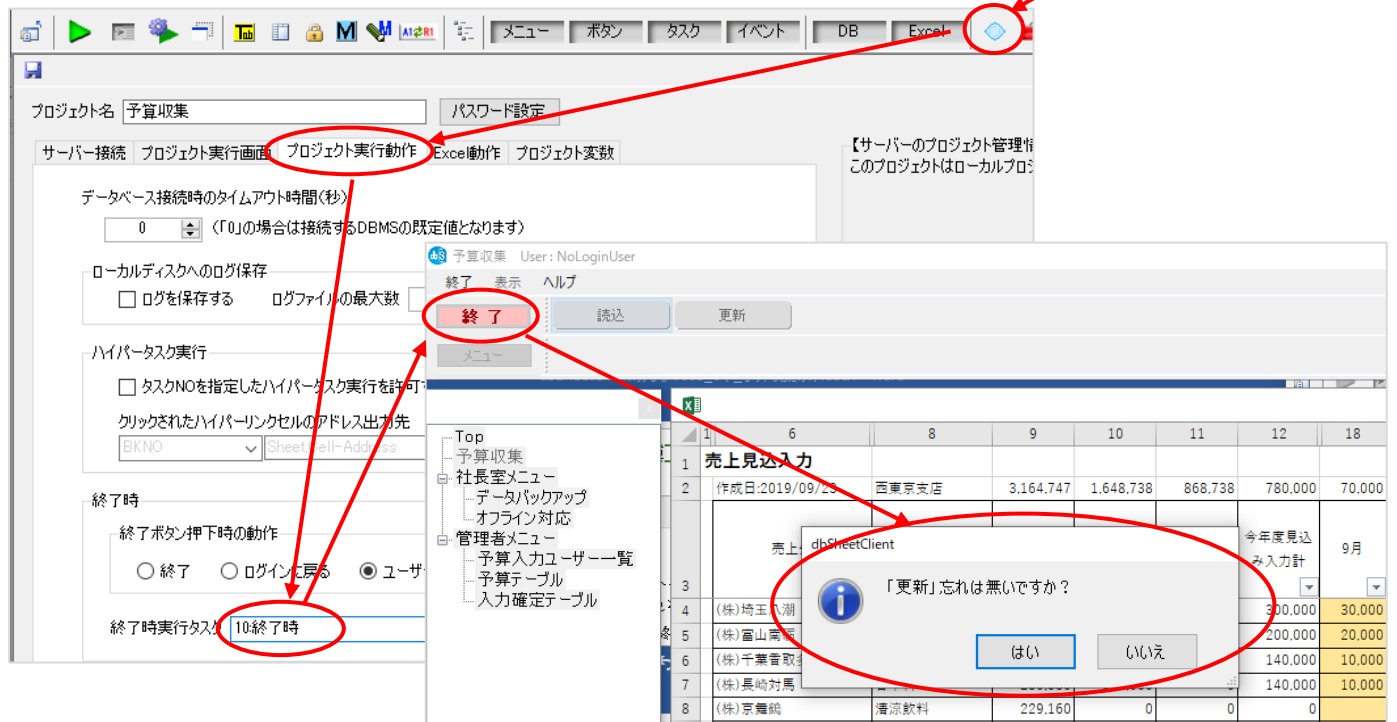
6.8 プロジェクト作成：最終調整

6.8.1 更新忘れを防ぐ

予算データ入力後、「更新」ボタンを押してデータを保存せずに終了してしまうユーザーがいます。
この更新忘れを防ぐため、終了時に確認画面を表示するようにします。
タスクセット「終了時」を追加し「拡張メッセージ表示」タスクを設定します。



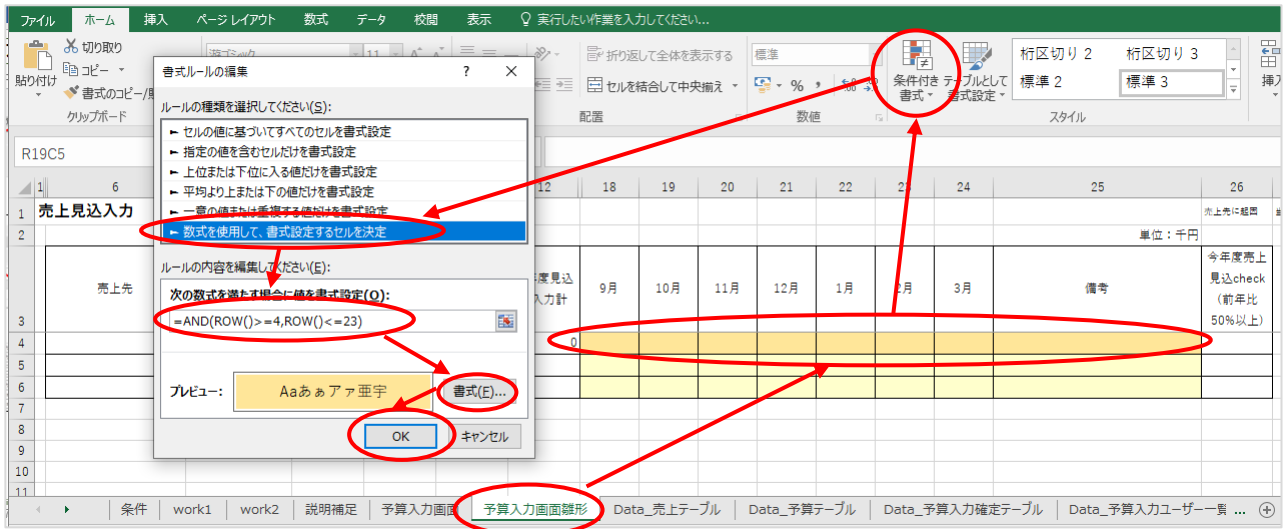
「プロジェクト実行動作」の「終了時実行タスク」にタスクセット「終了時」を割り付けます。



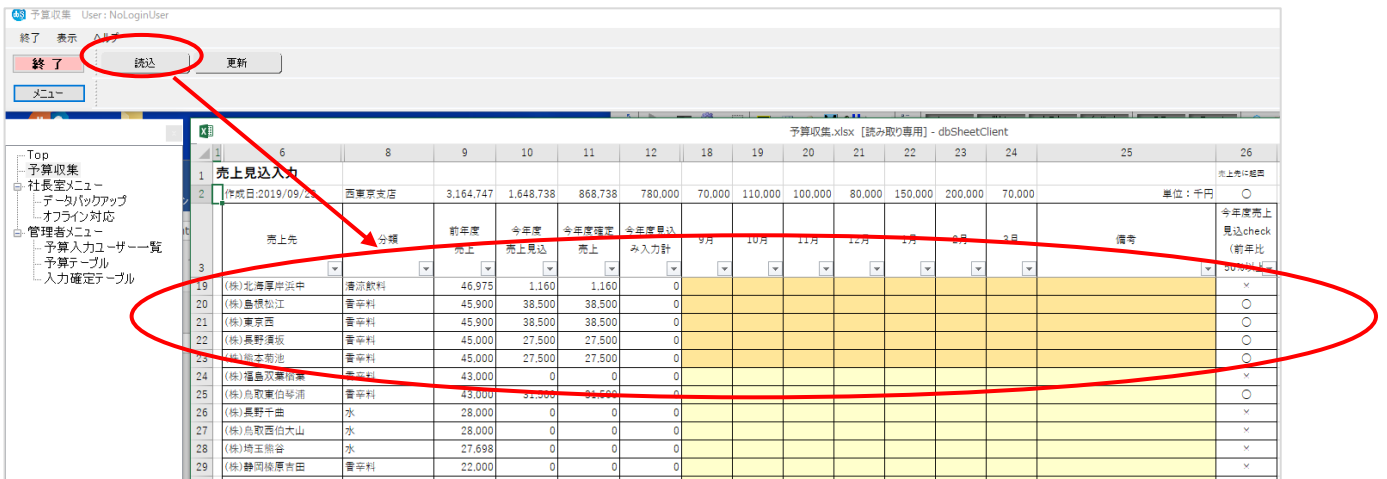
6.8.2 重要なレコードに色を付ける

予算入力画面では前年度売上の大きい順に表示しています。

上位20レコードについては入力セルの色を条件書式で変えています。(既に設定済です)



デバッグモードで実行

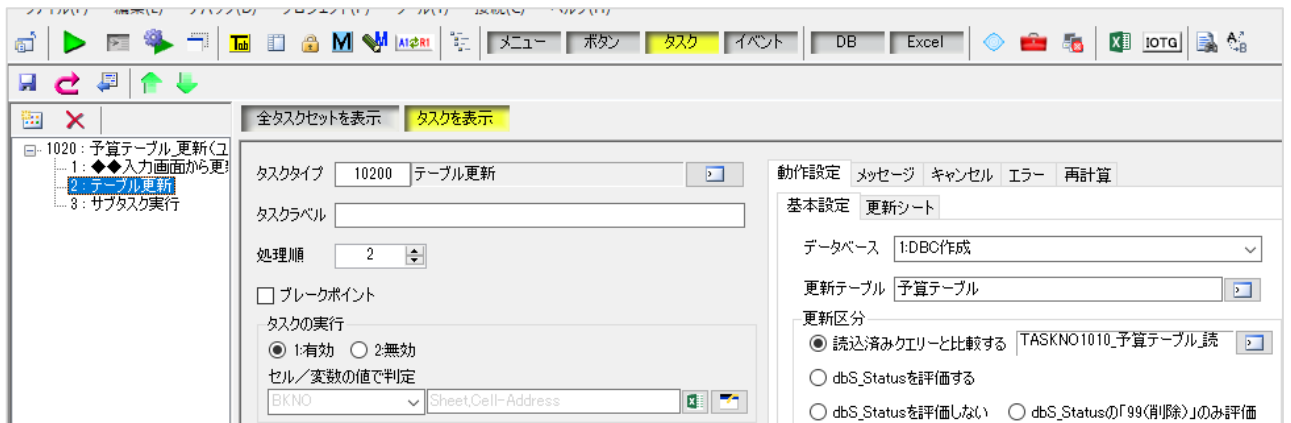


6.8.3 排他制御

dbSheetClient には「楽観ロック」と「悲観ロック」という2種類の排他制御機能があります。

デフォルトの「楽観ロック」は同じレコードを複数のユーザーが読み込んだ時、先に更新したユーザーがいれば後から別のユーザーが更新しようするとエラーとなります。(タスク:テーブル更新)

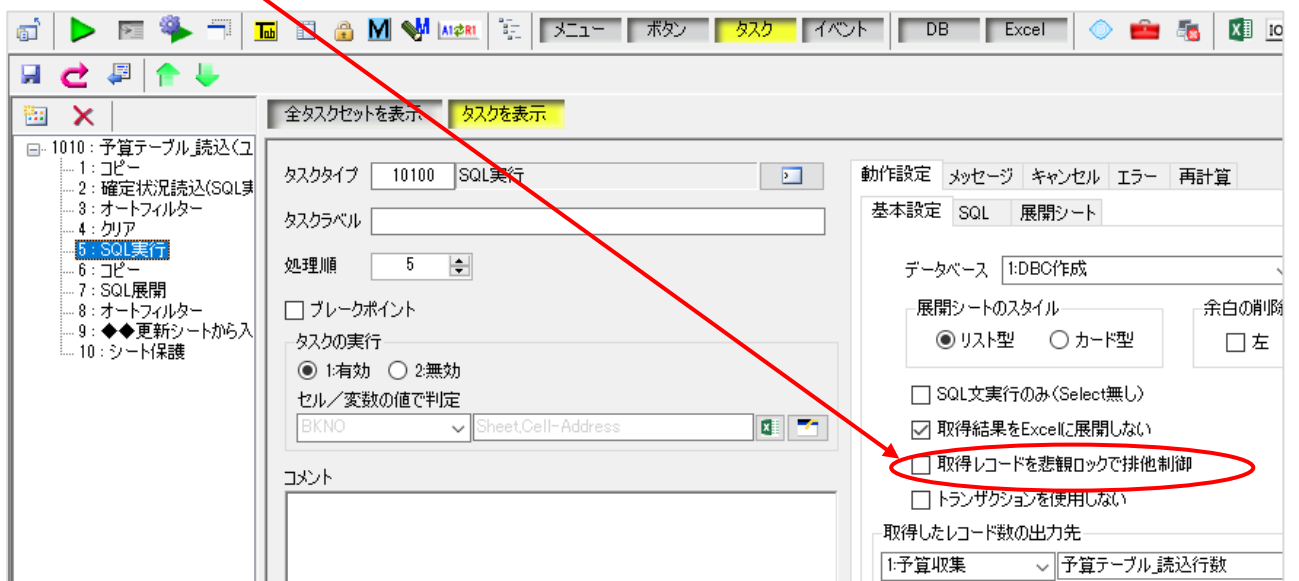
(作成したプロジェクトは「楽観ロック」となっています。)



「悲観ロック」は先に読み込んだユーザーがいれば後から別のユーザーが読み込もうとした時、エラーとなります。(タスク:SQL実行)

部署の予算を複数のユーザーが同時に編集しようするとエラーが発生する確率が高くなります。

部署選択の後、売上先を指定してさらに編集する予算データを絞り込む等の工夫が必要となります。



6.9 未作成の機能の説明

前記の「プロジェクト作成」では、プロジェクト「予算収集_完成」の以下の機能は作成していません。

- ・メニュー「予算収集」のボタン「確定」の処理
- ・メニュー「予算収集」のボタン「進捗状況」の処理
- ・メニュー「社長室メニュー>予算テーブル準備」の処理
- ・メニュー「社長室メニュー>入力確定テーブル準備」の処理
- ・メニュー「社長室メニュー>査定データ準備」の処理
- ・ユーザーが複数支店を担当する場合の対応 (★)
- ・「予算収集」のボタン「更新」無効化 (★)
- ・社長室メンバー以外の「社長室メニュー」無効化 (★)
- ・未登録ユーザーがログインした時の通知 (★)

これら機能の内、一部の機能(★)の概要の説明をします。

(プロジェクト「予算収集_完成」を元にして説明します)

6.9.1 ユーザーが複数支店を担当する場合の対応

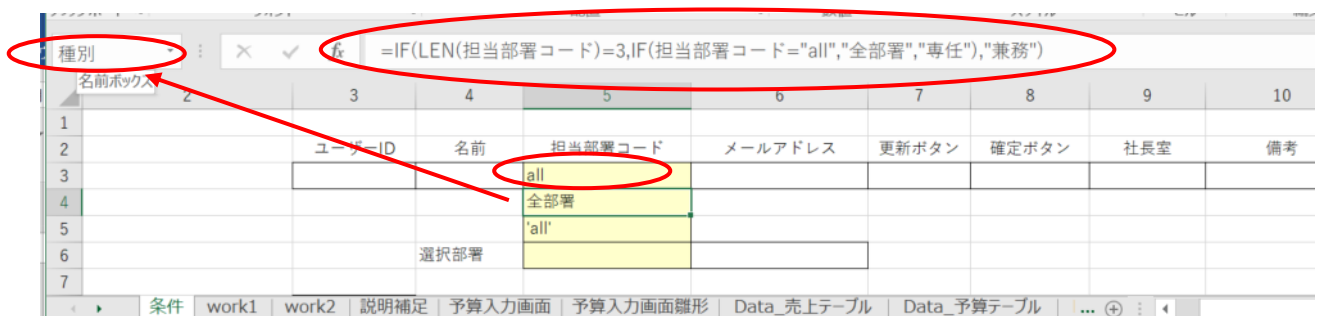
ユーザーの担当支店はテーブル「予算入力ユーザー一覧」の項目「担当部署コード」で管理しています。

各支店の部署コードは 3 文字が本システムの仕様となっています。

項目「担当部署コード」には、以下の 3 ケースを想定しています。

ケース	担当部署	項目「担当部署コード」設定例
専任	1 部署のみ	A01
兼務	指定の複数の部署 (カンマ区切りで指定)	B01,C01
全部署	全部署。(all で指定)	all

ユーザー情報は以下の Excel シート「条件」の R3C3～に展開し、R3C5 に「担当部署コード」が設定されます。(下では all)



名前付きセル「種別」には「担当部署コード」の内容によってセル式で以下のいずれかを表示します。

「全部署」、「専任」、「兼務」

メニュー「予算収集」のボタン「読込」(TASKNO:1010)のタスク「条件判定分岐」で上記を判定します。

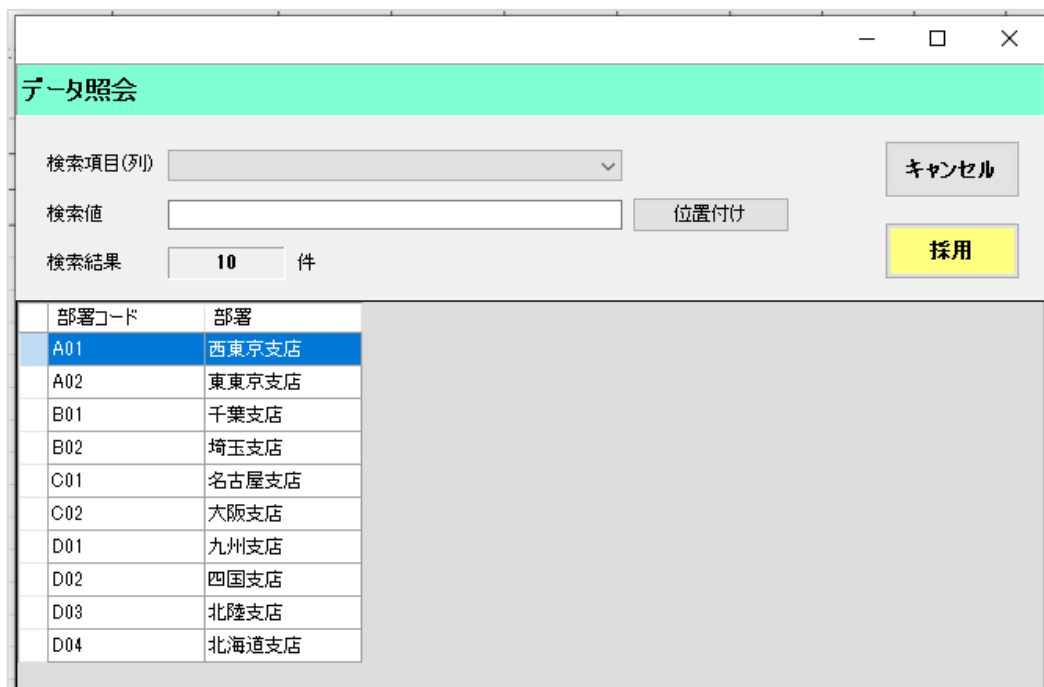


判定で、TASKNO:1011, 1012, 1013 のいずれかのタスクセットを実行します。

これらタスクセットの実行結果として、セル「選択部署」に、読込を行う1個の部署名を設定します。

「兼務」、「全部署」の場合は、複数の部署から1部署を選択するため、タスク「データ照会」を利用します。

タスク「データ照会」では、以下のウィンドウを表示して、ユーザーに1部署選択を促します。



読込のSQLではセル「選択部署」の内容を参照しています。

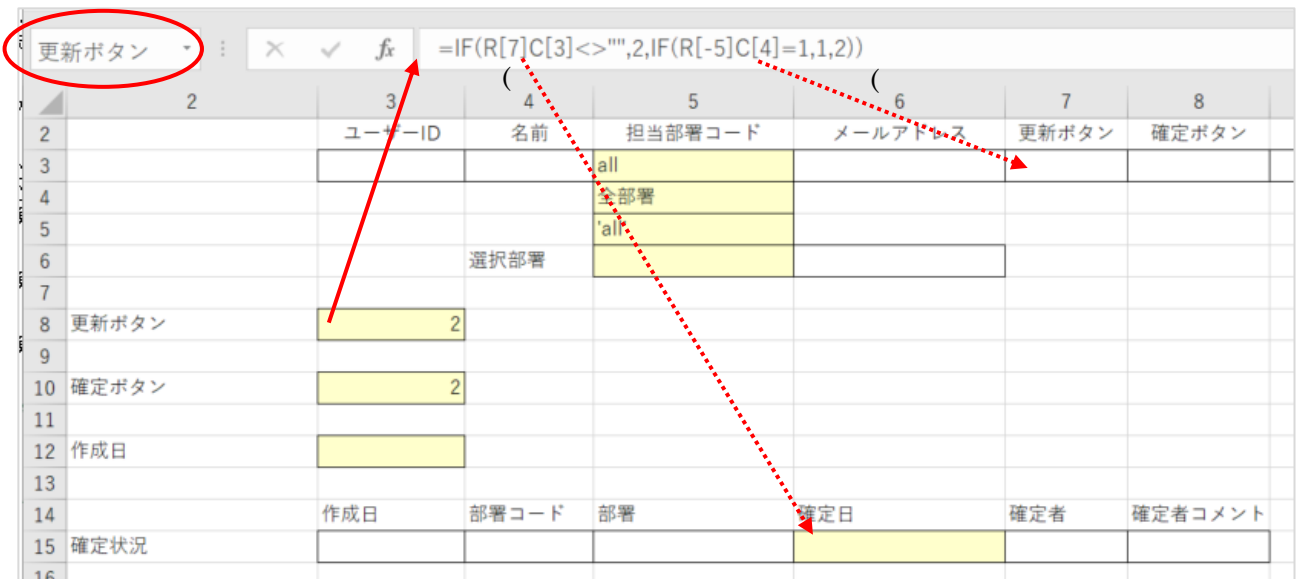
6.9.2 「予算収集」のボタン「更新」無効化

メニュー「予算収集」のボタン「更新」は以下のいずれかの場合に「無効」となります。

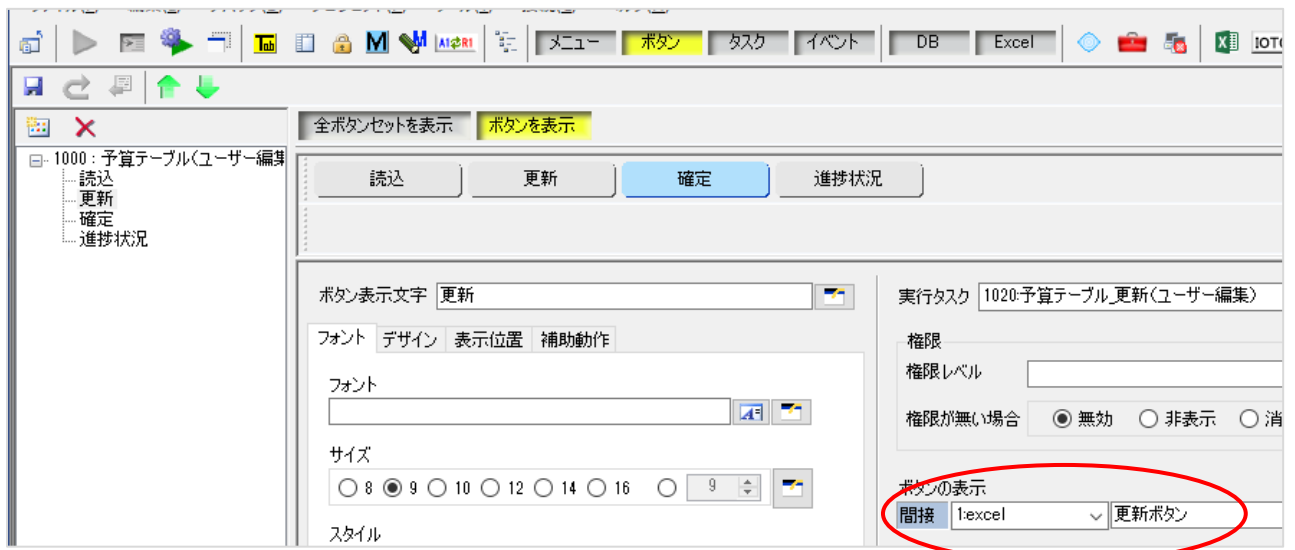
- (a) 予算入力ユーザー一覧で、ユーザーの「更新ボタン」が 1 以外の場合。
- (b) 予算入力確定テーブルで、表示支店の「確定日」が設定済の場合。

この判定は以下の Excel シート「条件」の R8C3「更新ボタン」で行っています。(セル式)

(1:有効、2:無効)



ボタン定義にて、上記判定値を参照して 1:有効、2:無効の設定をしています。



6.9.3 社長室メンバー以外の「社長室メニュー」無効化

メニュー「社長室メニュー」は以下の場合に「無効」となります。

- ・予算入力ユーザー一覧で、ユーザーの「社長室」が 1 以外の場合。

この判定は以下の Excel シート「条件」の R3C9 の「社長室」の値を参照して、プロジェクト変数「P_メニュー表示_社長室メニュー」に判定値を計算結果として設定しています。

(1:有効、2:無効)

	3	4	5	6	7	8	9
1							
2	ユーザーID	名前	担当部署コード	メールアドレス	更新ボタン	確定ボタン	社長室
3			all				
4			全部署				

全タスクセットを表示 | タスクを表示

タスクタイプ: 40300 変数演算

タスクラベル: 社長室メニューの表示

処理順: 3

動作設定: メッセージ | キャンセル | エラー | 再計算

処理順	変数名	演算式
1	P_メニュー表示_社長室メニュー	IF("BKNO"条件!R3C9"=1, 1, 2)

MNO: 2000

メニュー名: 社長室メニュー

表示順: 3 | 階層: 1

表示: Excelブック | BKNO: 1excel | シート: work2

その他

ボタン(BSNO):

タスク: 起動前処理 | 終了時処理

権限: 権限レベル

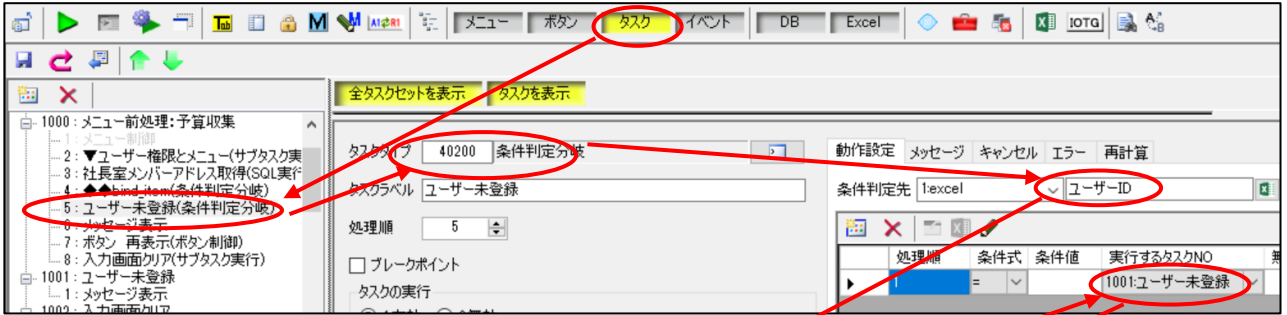
権限が無の場合: 無効 | 非表示

メニューの表示: 変数 P_メニュー表示_社長室メニュー

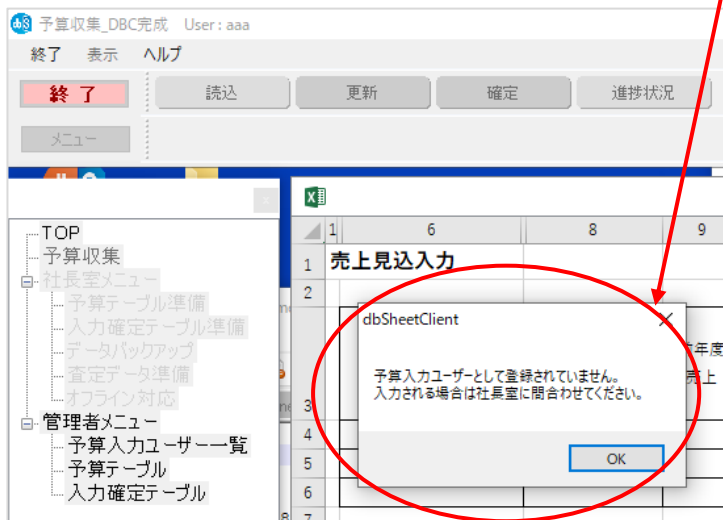
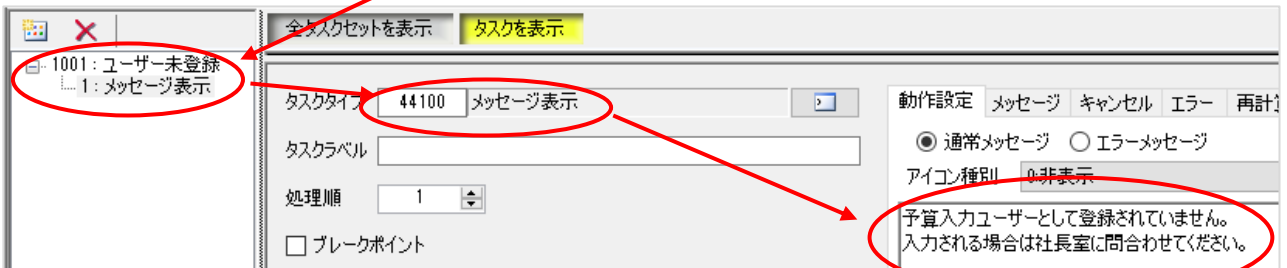
プロジェクト変数はタスクセット変数とは異なり、タスクセットを超えてプロジェクト内全体で利用可能な変数です。

6.9.4 未登録ユーザーがログインした時の通知

「予算入力ユーザー一覧」テーブルに登録されたユーザー以外がログインし、メニュー：予算収集選択時、メッセージを表示する機能です。



ユーザーID	名前	担当部署コード	メールアドレス	更新
		B01,A02		
		兼務		
		B01','A02'		
	選択部署			
	更新ボタン			4



6.10 マクロで実装する場合

前記の「プロジェクト作成」では、Excel マクロは利用していません。

TASKNO:99000 以降のタスクセットの処理はマクロで置き換えることが可能です。

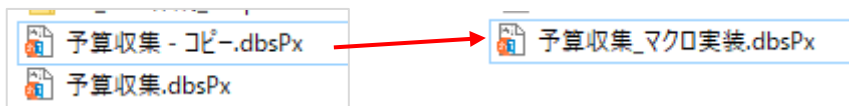
その中の以下 2 個のタスクセットをマクロで置き換えてみます。

- ・TASKNO:99065 ▽▽バックアップファイル名設定
- ・TASKNO:99085 ▽▽オフラインファイル読込

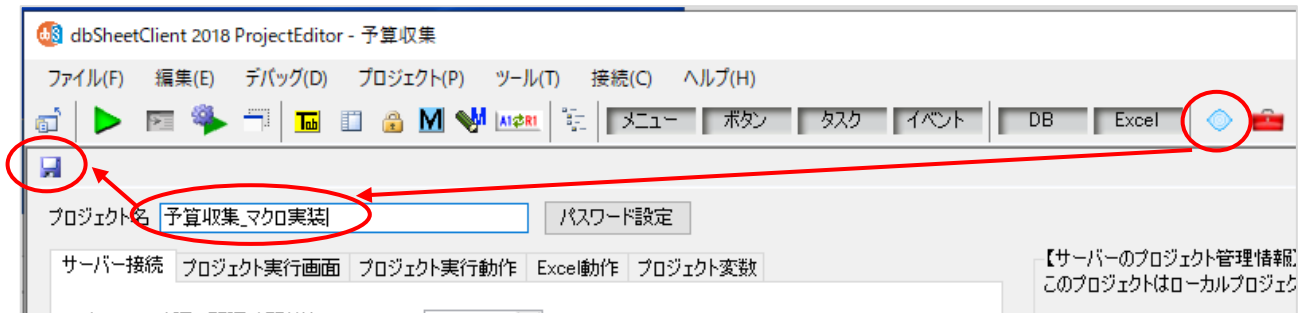
6.10.1 マクロ実装の前準備

既存のプロジェクトファイルをコピーしてマクロ実装用のプロジェクトを作成します。

ファイル名は「予算収集_マクロ実装.dbsPx」とします。

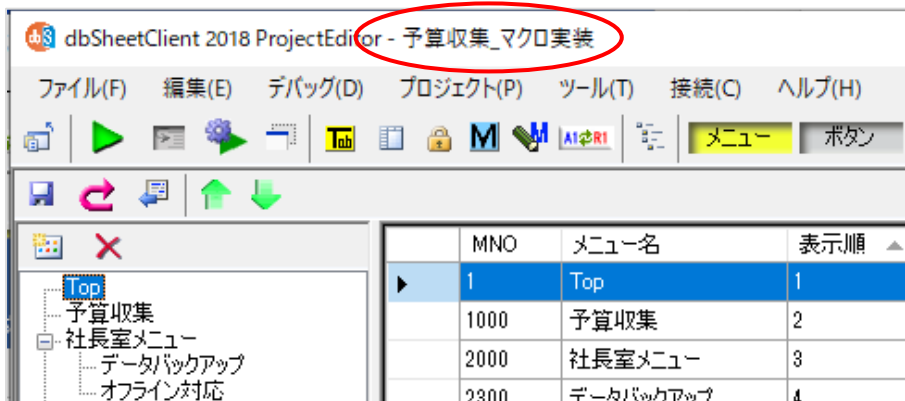


プロジェクトファイル「予算収集_マクロ実装.pdxPx」を開き、プロジェクト名を「予算収集_マクロ実装」に変更します。



プロジェクト名を変更後、「保存」をして、再度、プロジェクトを開きます。

dbSheetClient 開発版の上部のプロジェクト名が「予算収集_マクロ実装」になります。



「予算収集.xlsx」 ファイルからマクロ有効ブック:「予算収集_マクロ実装.xlsm」を作成します。

予算収集.xlsx ファイルを開きます。

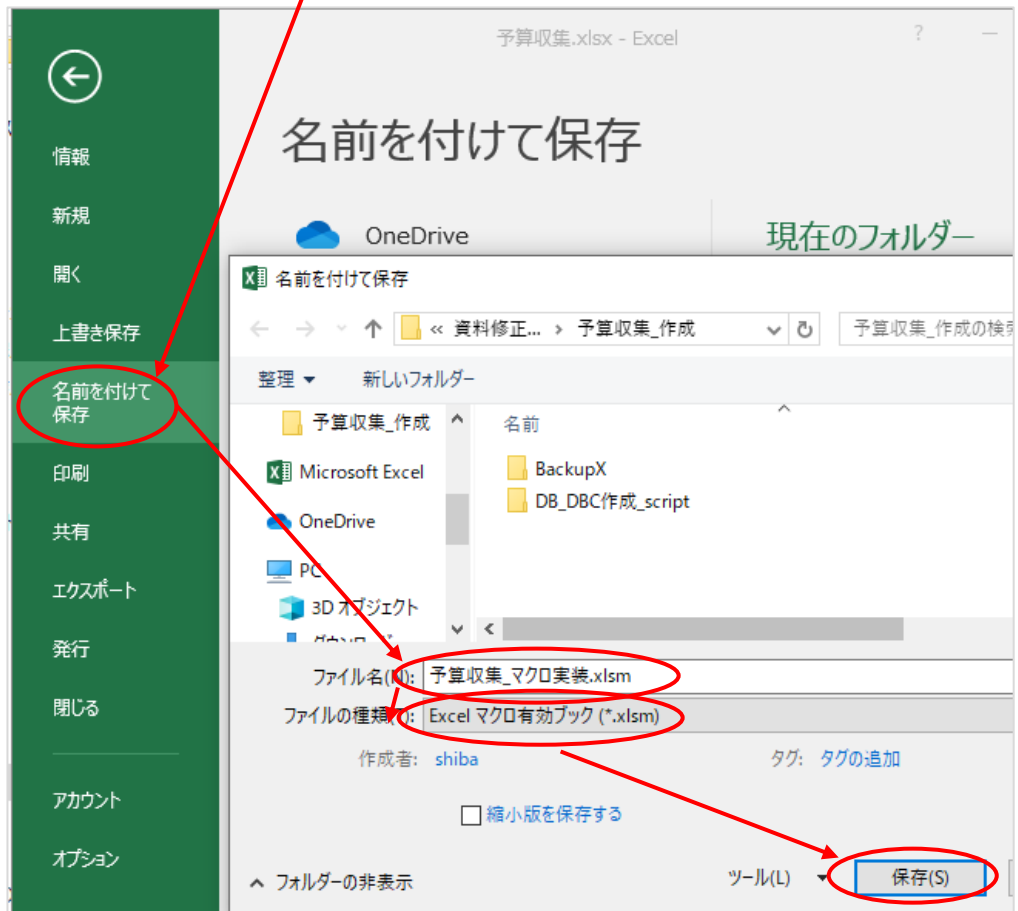
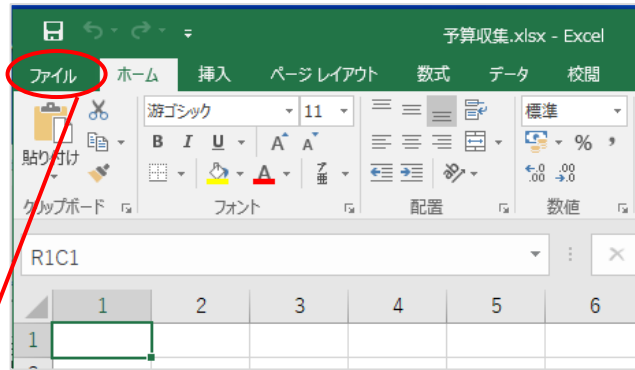
「ファイル」→「名前を付けて保存」

「ファイル名」: 予算収集_マクロ実装

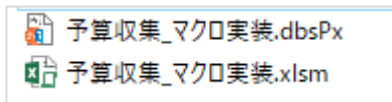
「ファイルの種類」:

Excel マクロ有効ブック(*.xlsm)

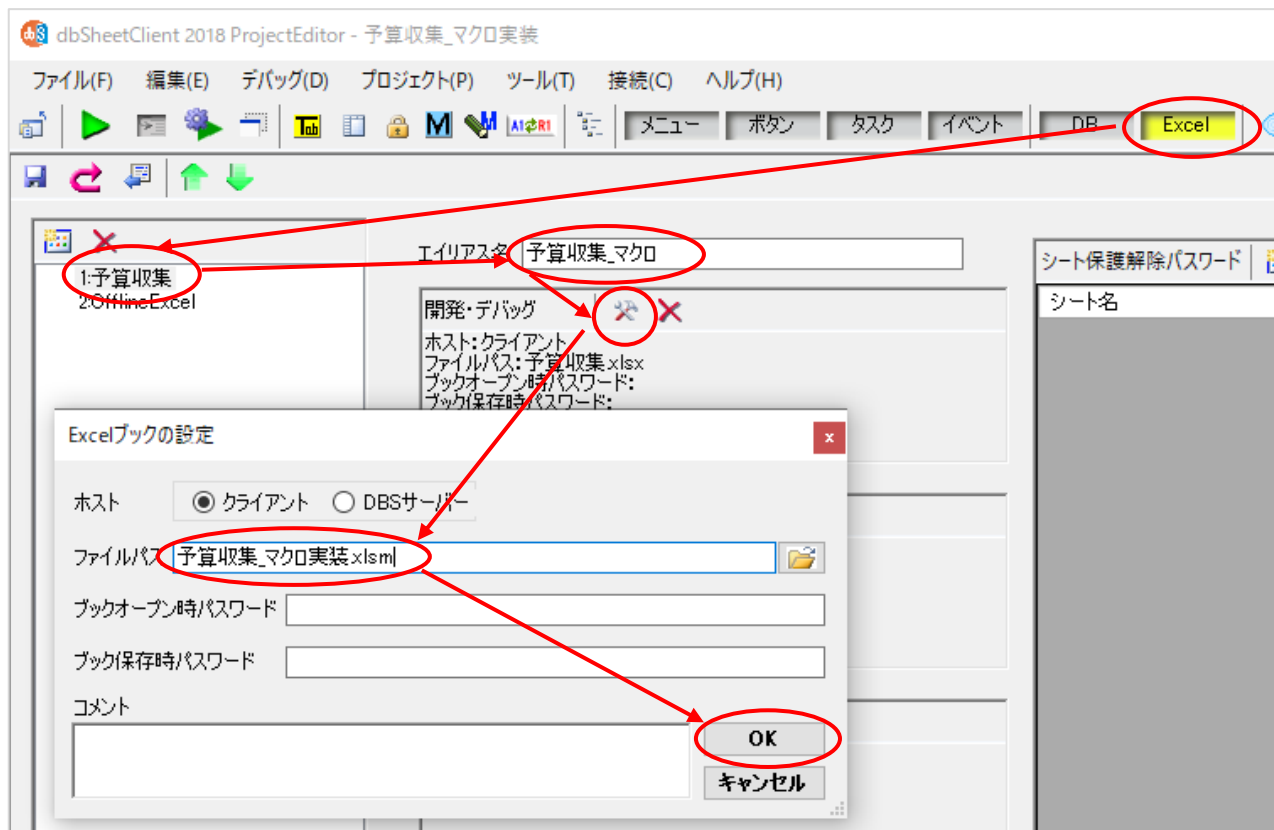
として「保存」し、終了します。



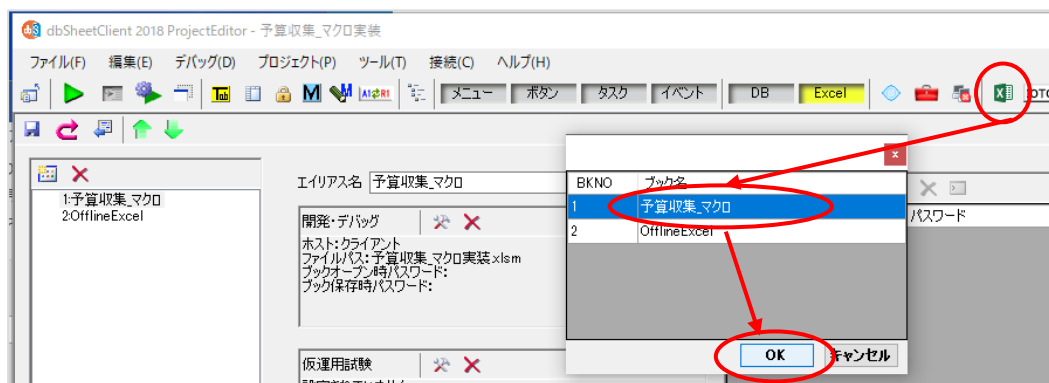
以下の 2 ファイルを作成しました。



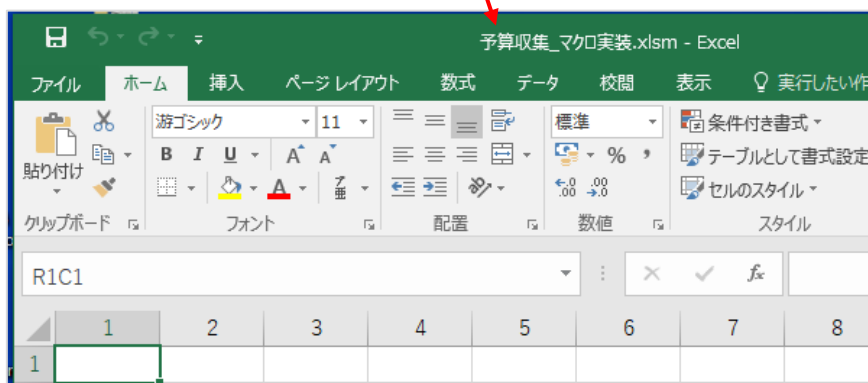
Excel 定義を作成した「予算収集_マクロ実装.xlsm」に変更します。



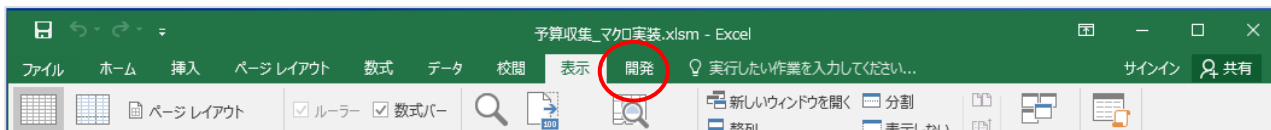
dbSheetClient 開発版から「予算収集_マクロ実装.xlsm」ファイルを開くことを確認します。



マクロの Excel が
開ければ OK です。

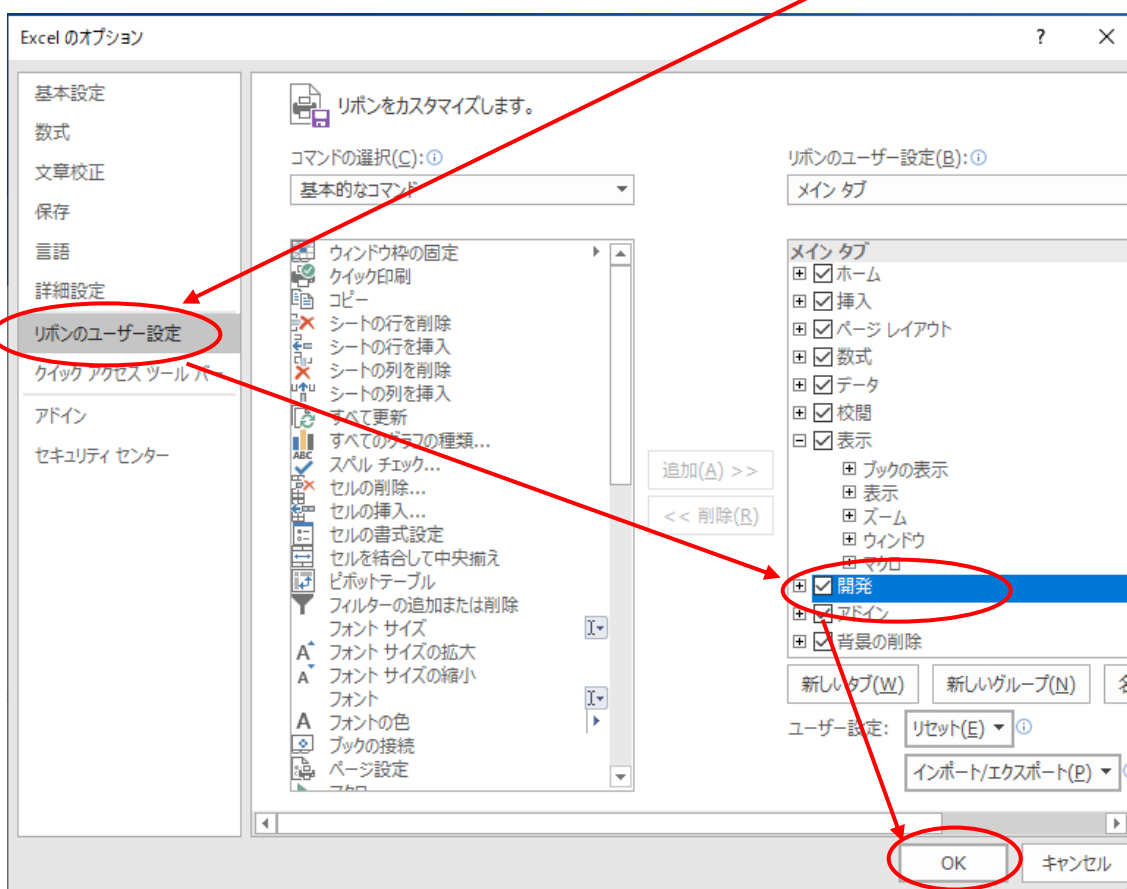
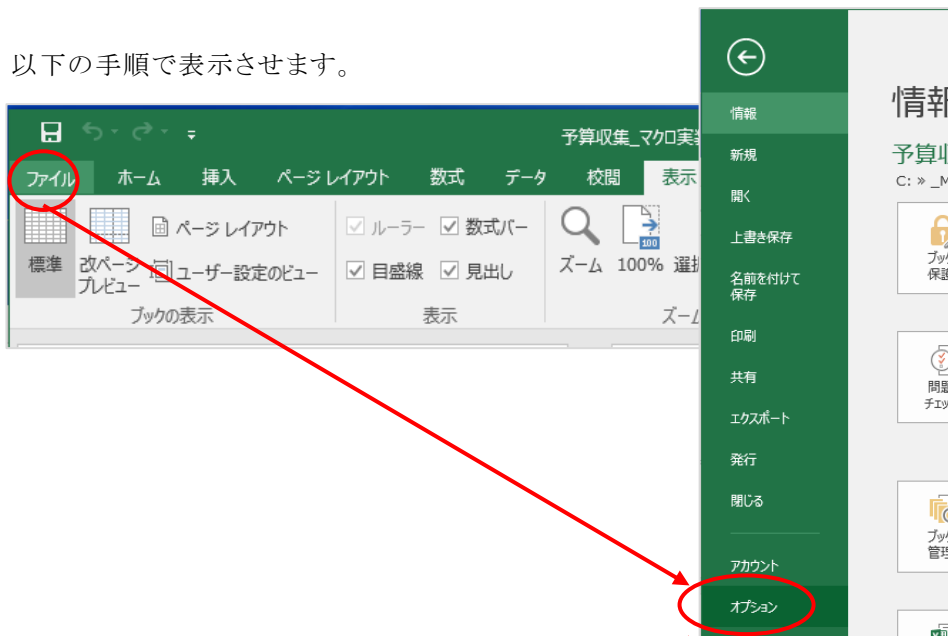


開いている Excel にタブ「開発」があるか確認します。



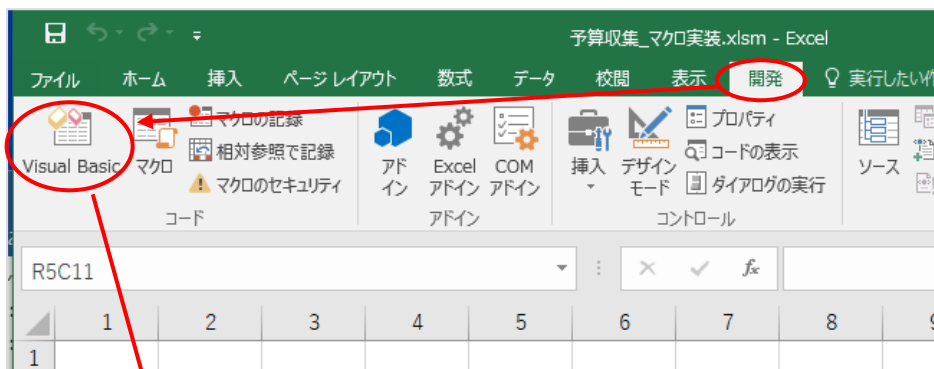
タブ「開発」が無い場合は、以下の手順で表示させます。

「ファイル」
→「オプション」

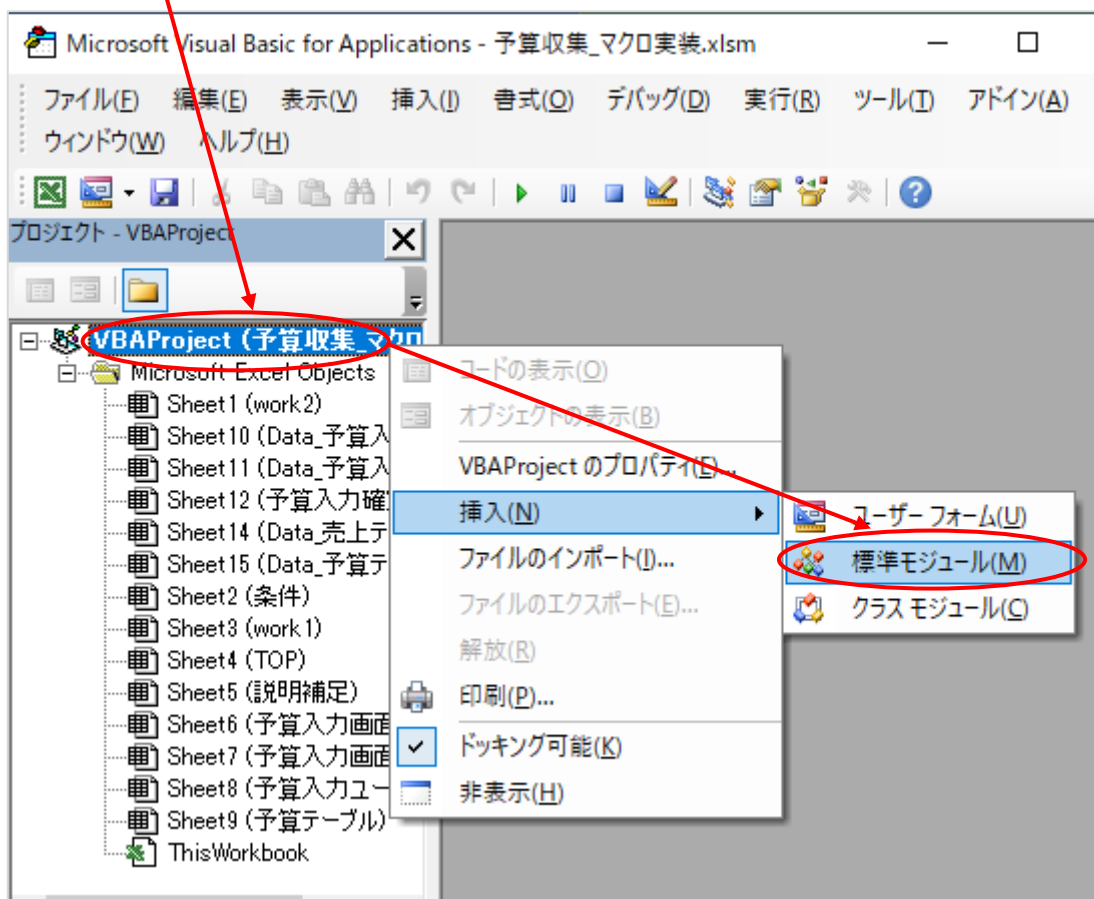


6.10.2 バックアップファイル名セット：マクロ実装

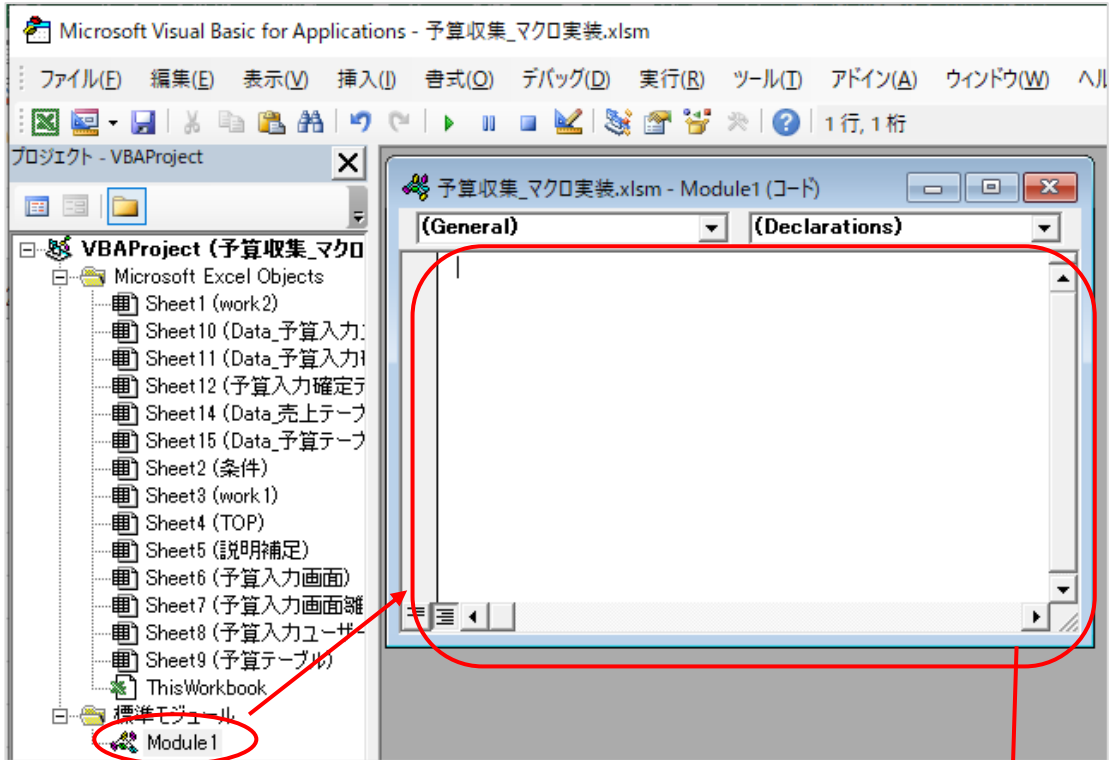
Excel ファイルにマクロ「バックアップ名セット」を実装します。



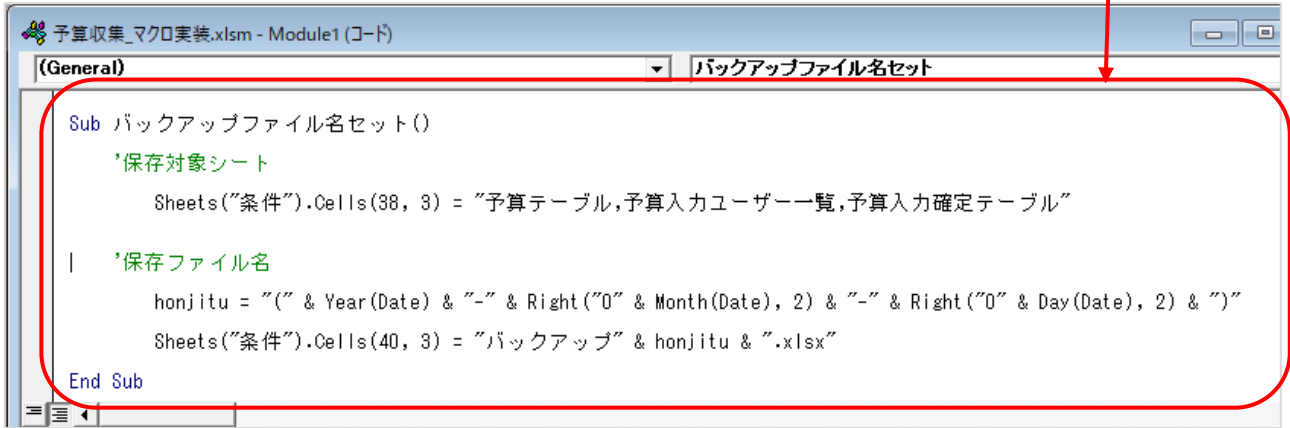
標準モジュールを追加します。



空の標準モジュール: Module1 が追加されます。



標準モジュール: Module1 に「Sub バックアップファイル名セット()」を定義します。
マクロのコードはプロジェクト「予算収集_完成」の「予算収集_DBC 完成.xlsm」からコピーできます。



Excel ファイルを保存して閉じます。

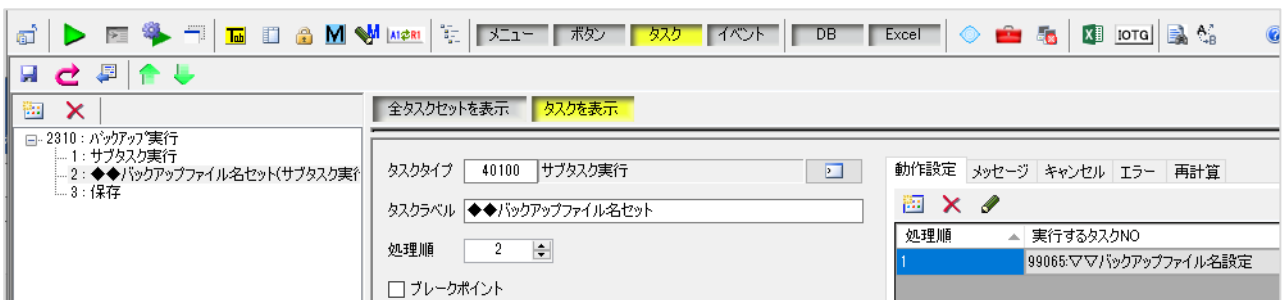
6.10.3 バックアップファイル名セット：マクロ起動

既存の以下のタスクセットの呼び出しの代わりに、マクロを実行するようにします。

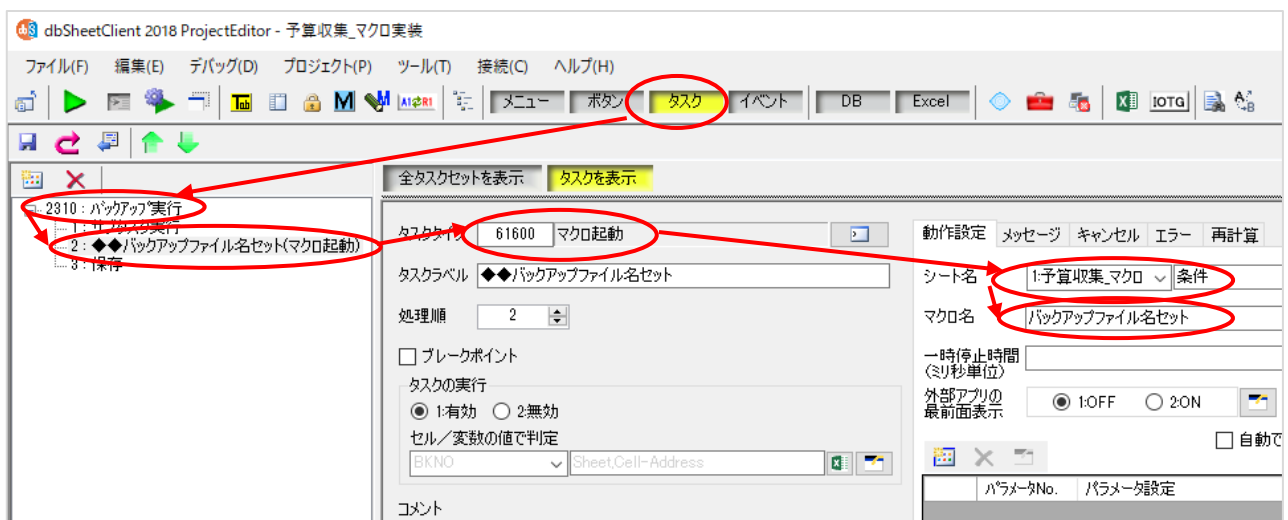
・TASKNO:99065 ▽▽バックアップファイル名設定

既存の処理では「TASKNO:2310 バックアップ実行」の処理順:2 で上記のタスクセットを呼び出しています。

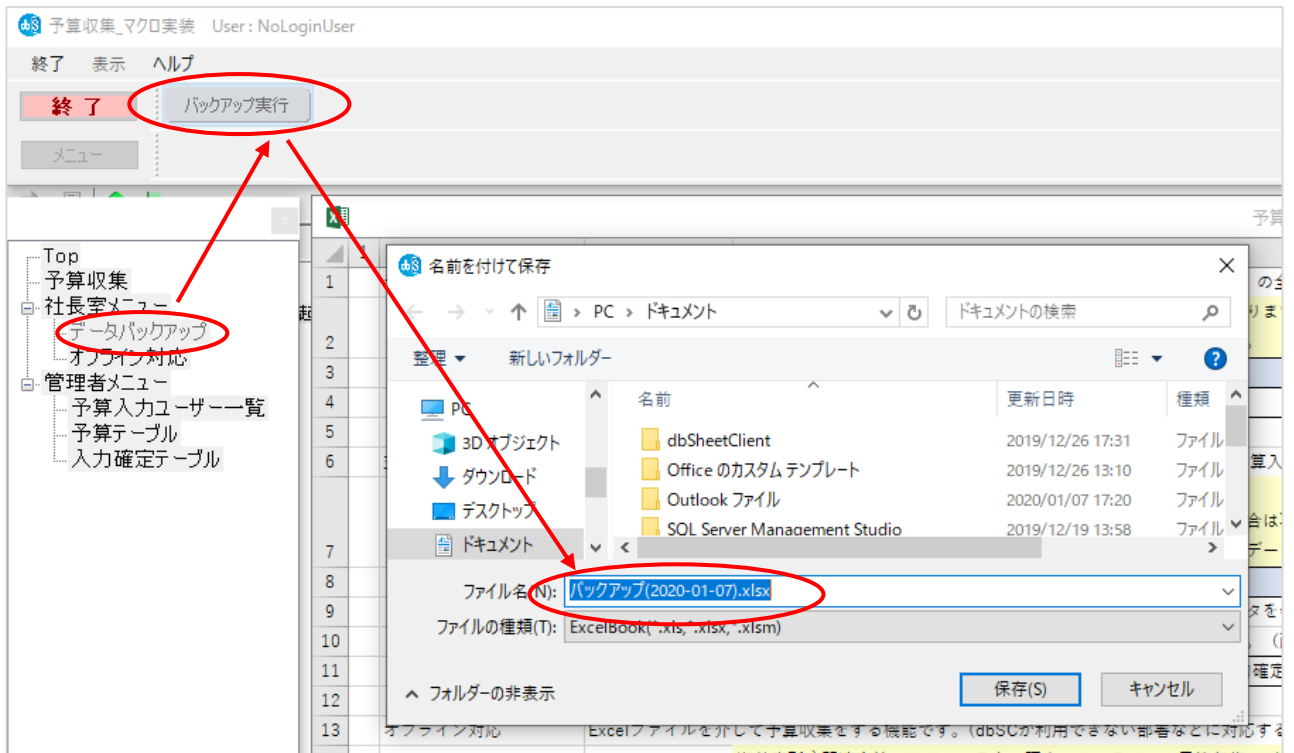
(既存は以下)



タスクタイプを「マクロ起動」に変えてマクロ「バックアップファイル名セット」を実行するように変更します。



プロジェクトを保存して、デバッグ実行で動作確認します。

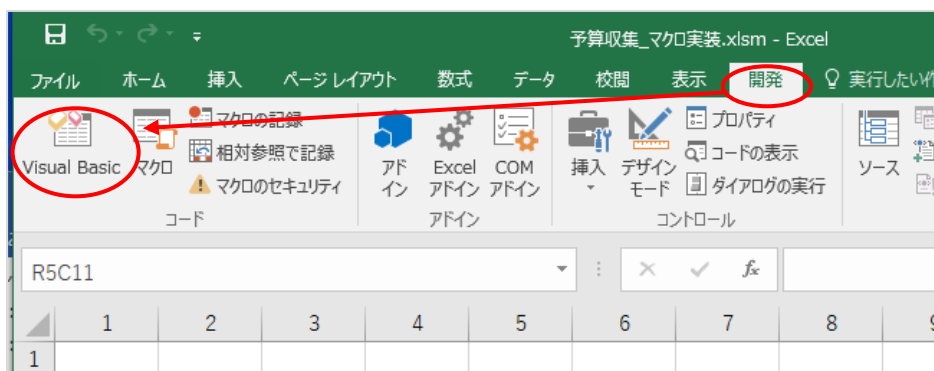


ファイル名で日付ありのファイル名が表示されれば OK です。

6.10.4 オフラインファイル読込：マクロ実装

Excel ファイルにマクロ「オフラインファイル読込」を実装します。

既に作成済の標準モジュール:Module1 にマクロを追加します。



標準モジュール:Module1 に「Sub オフラインファイル読込()」を定義します。

マクロのコードはプロジェクト「予算収集_完成」の「予算収集_DBC 完成.xlsx」からコピーできます。

```
予算収集_マクロ実装.xlsm - Module1 (コード)
(General) オフラインファイル読込

Sub バックアップファイル名セット()
    '保存対象シート
    Sheets("条件").Cells(38, 3) = "予算テーブル,予算入力ユーザー一覧,予算入力確定ラ

    '保存ファイル名
    honjitu = "(" & Year(Date) & "-" & Right("0" & Month(Date), 2) & "-" & Right("
    Sheets("条件").Cells(40, 3) = "バックアップ" & honjitu & ".xlsx"
End Sub

Sub オフラインファイル読込()
    '元のブック名を記録
    Dim ReturnBook As String
    ReturnBook = ActiveWorkbook.Name

    '読込ファイルを開く
    file_pass = Sheets("条件").Cells(42, 3)
    file_name = Sheets("条件").Cells(43, 3)
    ChDir file_pass
    Workbooks.Open Filename:=file_pass & "*" & file_name

    '読込ファイルから9月～備考までをコピーする
    Workbooks(file_name).Activate
    Sheets("予算入力画面").Select

    Cells(10000, 3).Select
    Selection.End(xlDown).Select
    Selection.End(xlUp).Select
    rr = ActiveCell.Row

    Range(Cells(4, 18), Cells(rr, 25)).Select
    Selection.Copy
    Workbooks(ReturnBook).Activate
    Sheets("予算入力画面").Select
    Range(Cells(4, 18), Cells(rr, 25)).Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False

    '読込ファイルを閉じる
    Windows(file_name).Activate
    Application.DisplayAlerts = False
    ActiveWindow.Close
    Application.DisplayAlerts = True
End Sub
```

Excel ファイルを保存して閉じます。

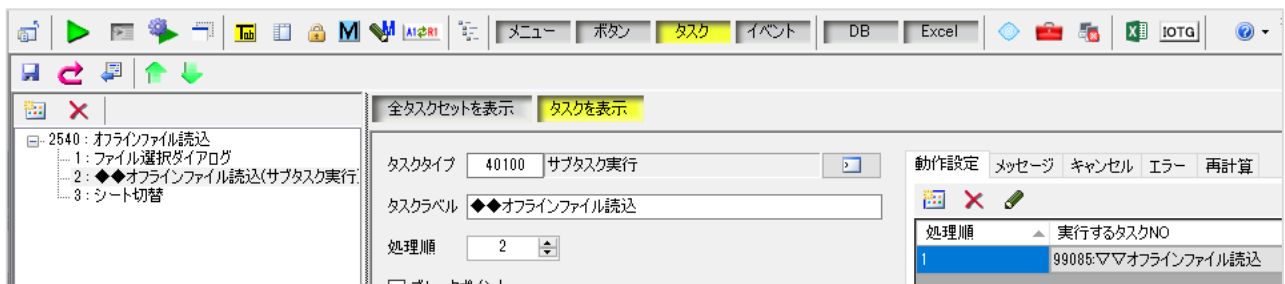
6.10.5 オフラインファイル読込：マクロ起動

既存の以下のタスクセットの呼び出しの代わりに、マクロを実行するようにします。

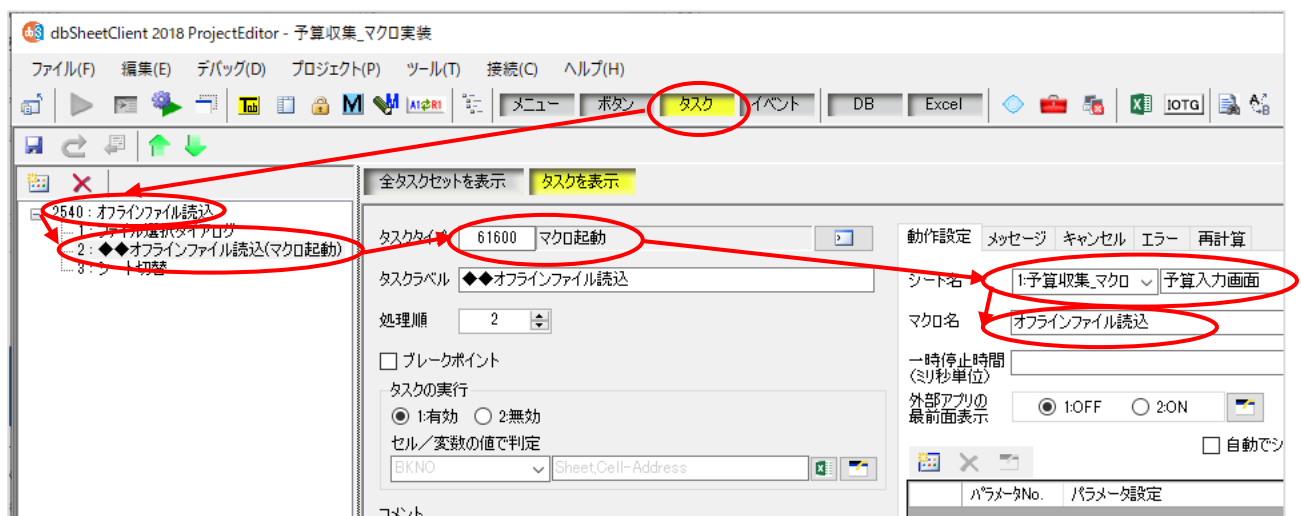
・TASKNO:99085 ▽▽オフラインファイル読込

既存の処理では「TASKNO:2540 オフラインファイル読込」の処理順:2 で上記のタスクセットを呼び出しています。

(既存は以下)



タスクタイプを「マクロ起動」に変えてマクロ「オフラインファイル読込」を実行するように変更します。



プロジェクトを保存して、デバッグ実行で動作確認します。

(動作方法については「6.3.11 デバッグ実行:社長室メニュー:オフライン対応」を参照)

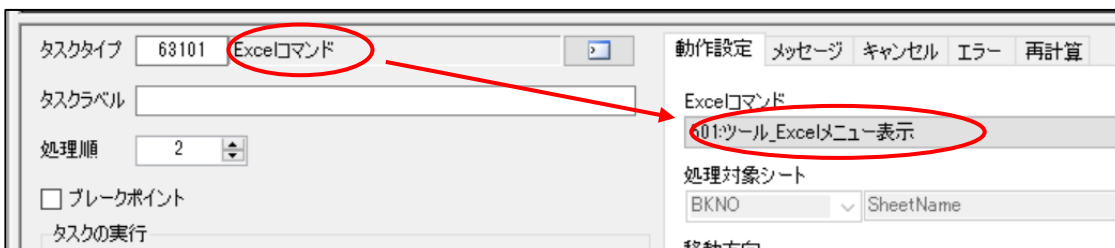
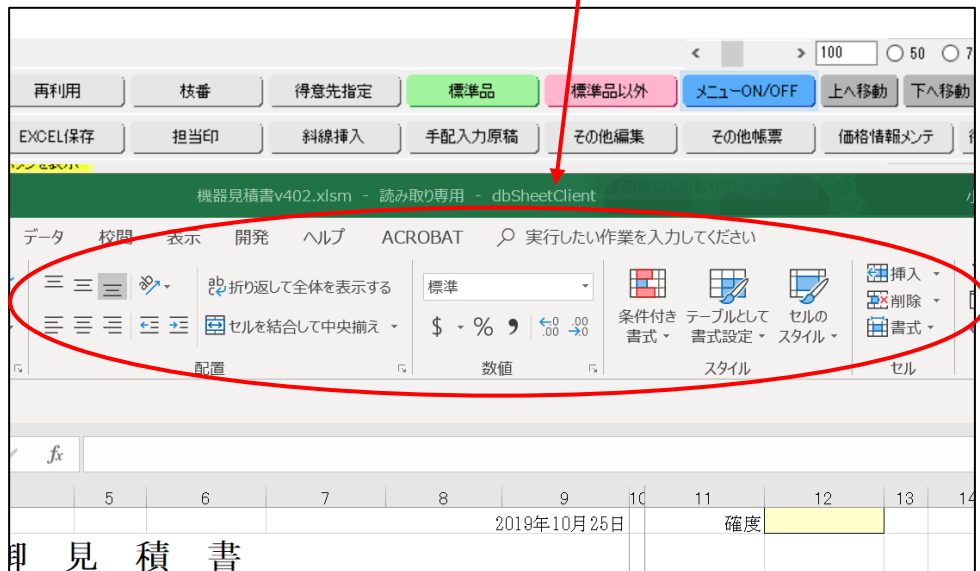
以上でマクロでの実装の説明は終わります。

7. 開発運用の小技

最後に、システム開発する上でのちょっとした小技をご紹介します。

7.1 Excel メニュー（タブやリボン）を解放する

Excel メニュー（タブやリボン）はデフォルトでは表示されませんが、「Excel コマンド」タスクにて表示可
dbSheetClient に違和感を抱いていたユーザーもひと安心(笑)



7.4 SQL はこれだけ使えばイイ

私の個人的な感想としては複雑な SQL 文は嫌いです。^^;

複雑な SQL とは「join」やら「union」やらやたら長文になる命令です。

dbSheetClient を使ってシステム開発する上で SQL の習得は必須か？

その答えは駆使できればできるに越したことはないができなくても最低限の文法を知っていれば OK。

それでは最低限の文法について紹介します。

<例1>

SQL を使うタスクは「SQL 実行」「データ照会」「検索付きデータ参照」の3つです。

SQL は Select 文がメインです。

The screenshot shows the SQL editor interface with the following SQL query: `SELECT * FROM 案件管理基本 where 部署コード = "BKNO!!選択部署" order by record_no desc`. Red arrows point from text annotations to parts of the query: **Where** is the record extraction condition (points to the where clause), **order by** is the record order (points to the order by clause), **desc** is the descending order specification (points to the desc keyword), ***印** is used to expand all items (points to the asterisk in the select clause), and **展開元テーブル名** is the source table name (points to the table name).

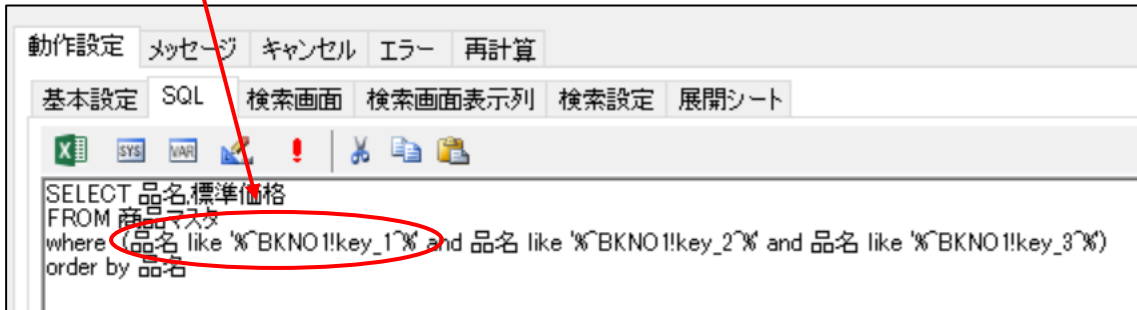
<例2>

展開する項目を指定

The screenshot shows the SQL editor interface with the following SQL query: `select ログインID,名前,部署コード,管理メニュー,登録更新ボタン,役割,メールアドレス FROM 案件管理ユーザー一覧 where ログインID = "SYS_UserID"`. A red circle highlights the column list in the select clause, with an arrow pointing to the text "展開する項目を指定".

< 例3 >

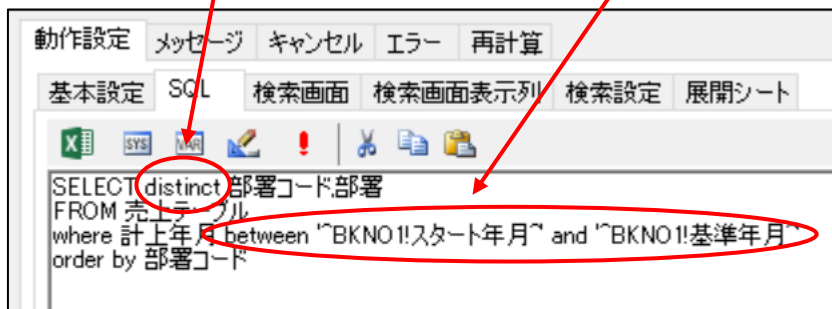
曖昧な文字列検索



< 例4 >

重複データを除外して取得

範囲を指定して抽出



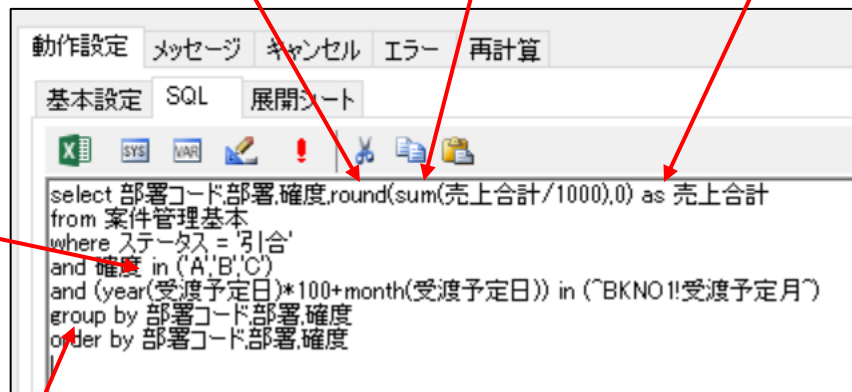
< 例5 >

四捨五入する関数

値を合計する関数

項目名を指定

複数の条件を指定



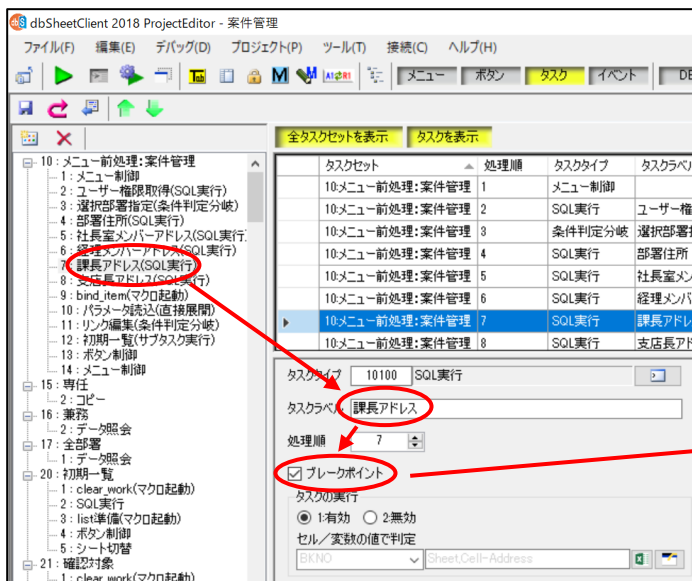
SUM関数で集計するグループ単位

7.5 デバッグのコツ

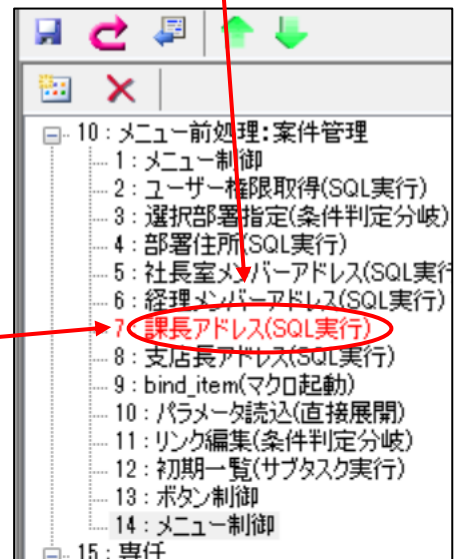
dbSheetClient のデバッグ補助機能を紹介します。

<ブレイク>

タスクセット内の任意のタスクで処理を中断させることができます。

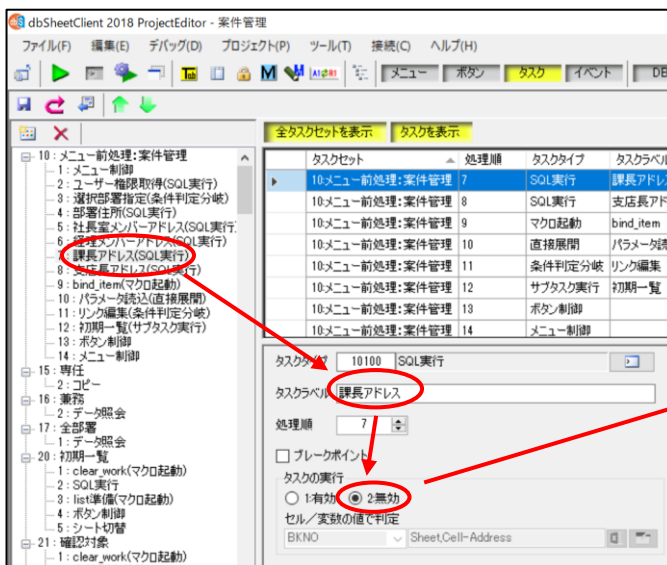


ブレイク設定されると赤くなります

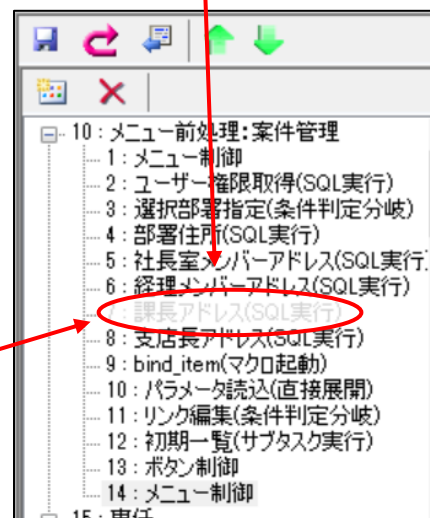


<無効化>

タスクセット内の任意のタスクを一時的に無効化させることができます。



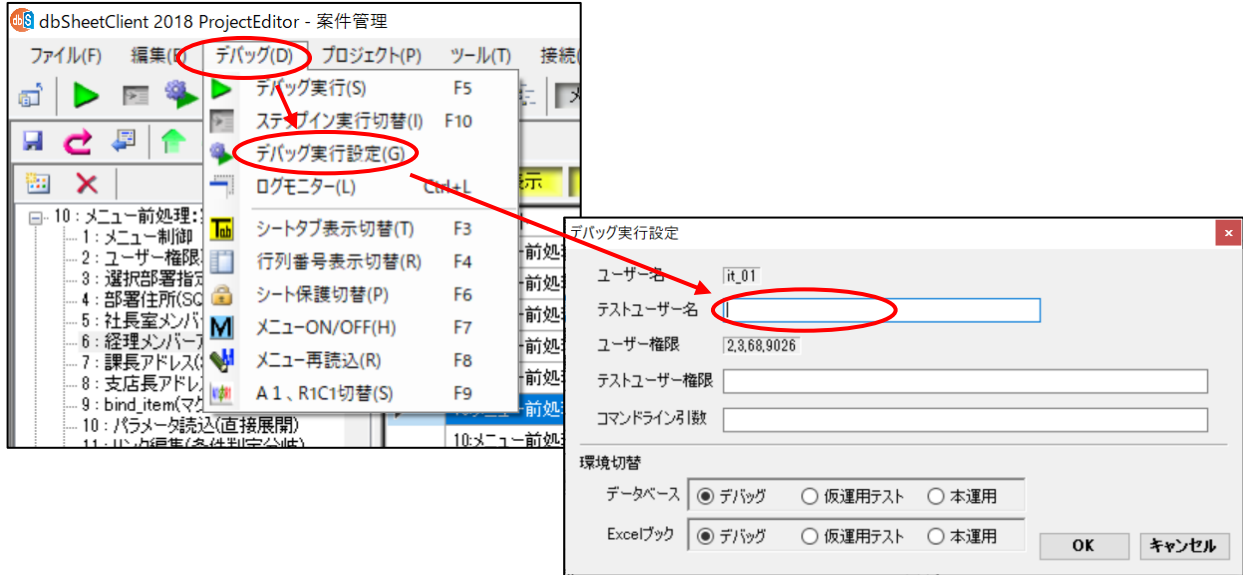
無効化されるとグレーになります



<テストユーザー変更>

ユーザーを切り替えてプロジェクトの動作確認ができます。

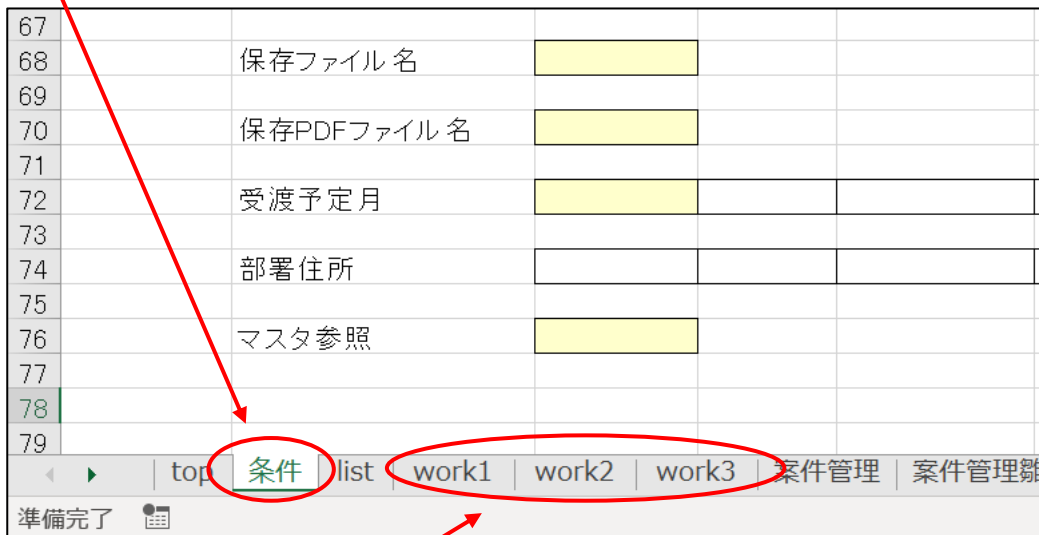
アクセス権によるメニューの表示・非表示が正しく設定されているか等を確認



<ワークシート活用>

「条件」シートを使って変数値を確認

(注) dbSheetClient はワークシートを使わずに変数管理する機能もあります。

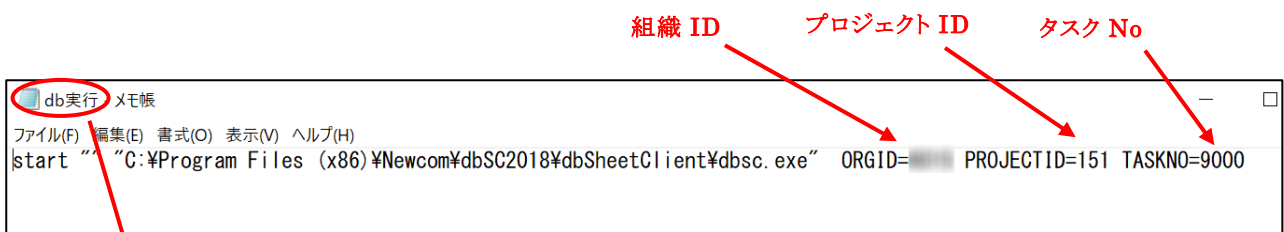


一次的にワークシートを使って中間結果を確認しつつ調整していくことができます。

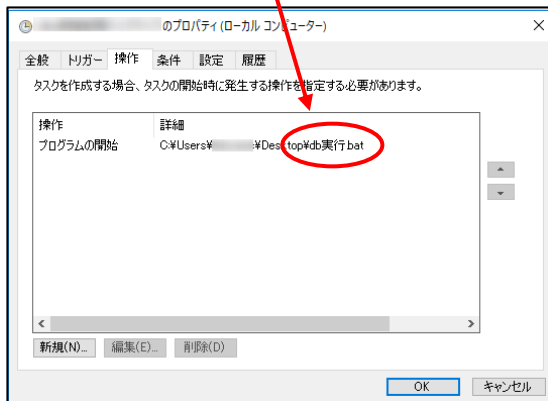
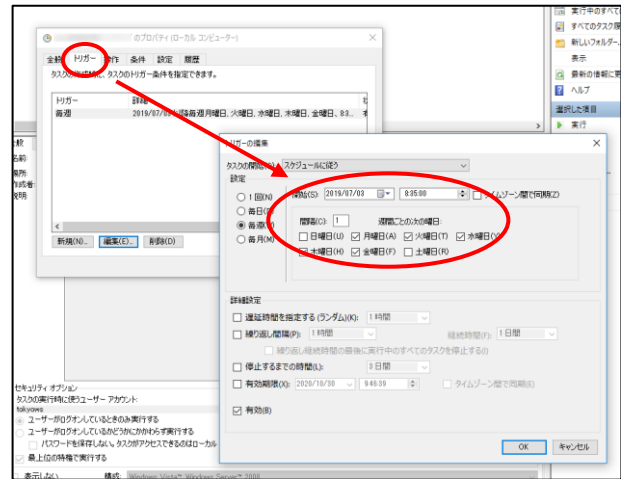
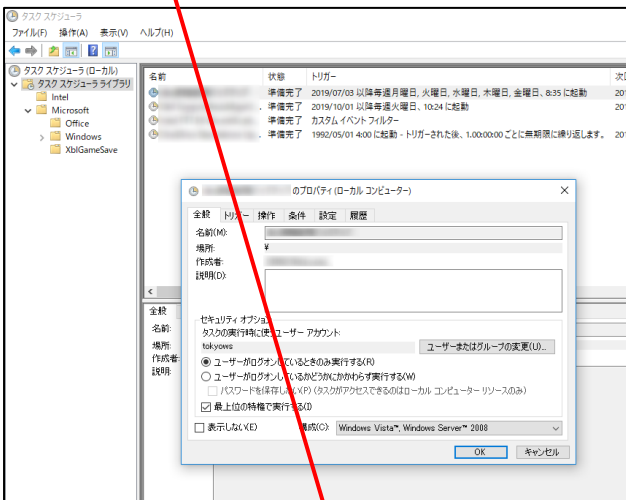
7.6 RPA モドキ

RPA (Robotic Process Automation) が働き方改革の切り札のように宣伝されています。定型業務の自動化ツールとして様々な RPA が登場しています。ここでは、dbSheetClient を使って業務を自動化する手法について、ご紹介します。

特定プロジェクトの特定タスクを直接実行するバッチファイルを作成します。



作成したバッチファイルをタスクスケジューラーに登録して定期的に自動実行させます。



7.7 ログの活用

dbSheetClient には標準でログ解析ツール「ServerLogViewer」が付いています。期間や処理内容、プロジェクトやユーザーを指定してログを抽出することができます。抽出したログは CSV ファイルに保存して様々な分析に役立てます。

プロジェクト開始件数 → ヘビーユーザーを発掘したり、稼働率の低いプロジェクトの見直し等

The screenshot shows the 'ServerLogViewer' interface. In the search conditions section, the date range is set to '2019年 9月28日' to '2019年 10月27日'. The '処理概要' (Process Summary) dropdown is set to '実行開始' (Execution Start). The '検索実行' (Execute Search) button is highlighted with a red circle. Red arrows point from the '1ヶ月' radio button, the '実行開始' dropdown, and the '検索実行' button.

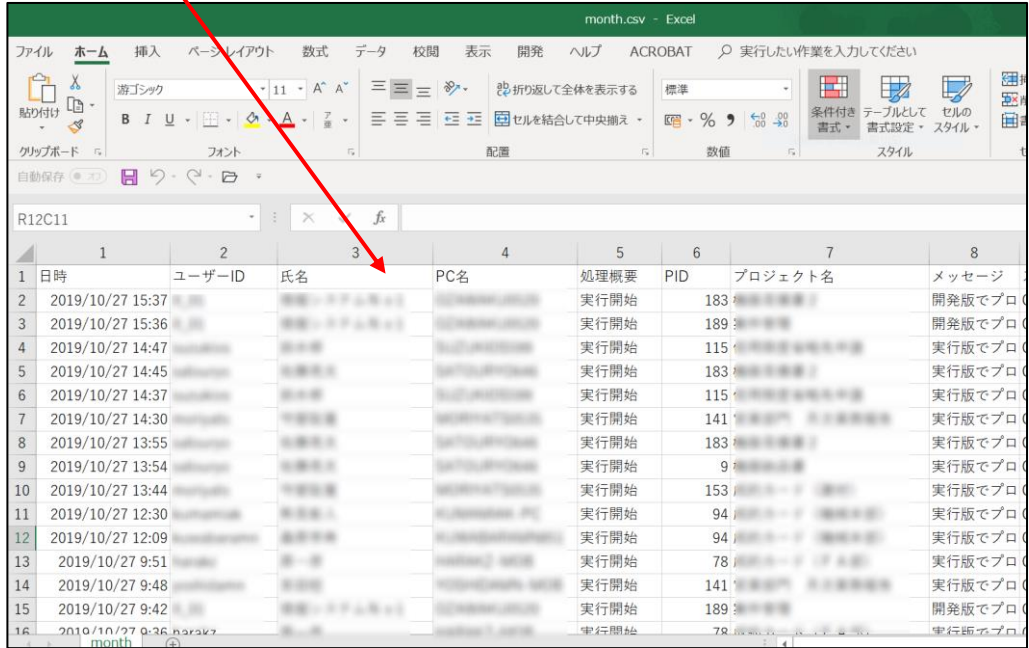
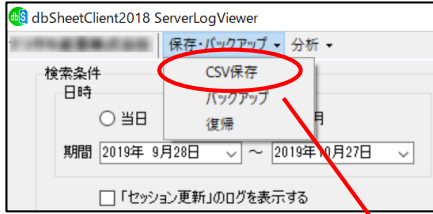
日時	ユーザーID	氏名	PC名	処理概要	PID	プロジェクト名	メッセージ
2019/10/27 15:37:36				実行開始	183		開発版でプロジェクトを実行開始しました。
2019/10/27 15:36:17				実行開始	189		開発版でプロジェクトを実行開始しました。
2019/10/27 14:47:03				実行開始	115		実行版でプロジェクトを実行開始しました。
2019/10/27 14:45:59				実行開始	183		実行版でプロジェクトを実行開始しました。
2019/10/27 14:37:17				実行開始	115		実行版でプロジェクトを実行開始しました。
2019/10/27 14:30:24				実行開始	141		実行版でプロジェクトを実行開始しました。
2019/10/27 13:55:24				実行開始	183		実行版でプロジェクトを実行開始しました。
2019/10/27 13:54:54				実行開始	9		実行版でプロジェクトを実行開始しました。
2019/10/27 13:44:39				実行開始	153		実行版でプロジェクトを実行開始しました。
2019/10/27 12:30:49				実行開始	94		実行版でプロジェクトを実行開始しました。

エラー発生ログ → エラーの原因や頻度を解析

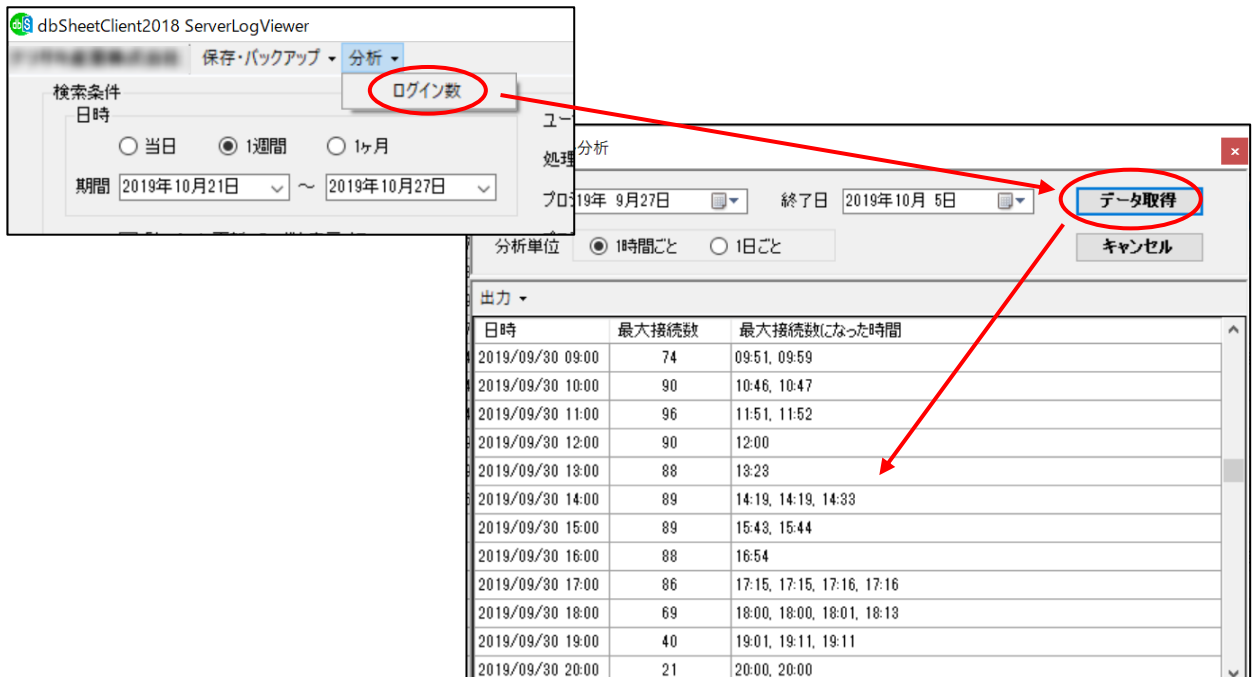
The screenshot shows the 'ServerLogViewer' interface with search filters set to '2019年 10月11日' to '2019年 10月11日'. The '処理概要' (Process Summary) dropdown is set to 'クライアントの処理結果' (Client Processing Results). The '検索実行' (Execute Search) button is highlighted with a red circle. Red arrows point from the '1週間' radio button, the 'クライアントの処理結果' dropdown, and the '検索実行' button.

日時	ユーザーID	氏名	PC名	処理概要	PID	プロジェクト名	メッセージ	ステータス
2019/10/11 9:50:07				印刷	153	ProjectName	ectNo:153LogStatus...	OK
2019/10/11 9:49:59				印刷	9	ProjectName	OKDateTime:2019/10...	OK
2019/10/11 9:48:57				クライアントの処理結果	31	エラー発生Proc...	ProjectNo:31LogStat...	Exception
2019/10/11 9:48:17				保存	85	ProjectName	ProjectNo:35LogStat...	OK
2019/10/11 9:48:12				保存	94	ProjectName	ProjectNo:94LogSta...	OK
2019/10/11 9:47:09				印刷	94	ProjectName	ProjectNo:94LogSta...	OK
2019/10/11 9:47:06				印刷	94	ProjectName	ProjectNo:94LogSta...	OK
2019/10/11 9:46:24				クライアントの処理結果	31	エラー発生Proc...	ProjectNo:31LogStat...	Exception
2019/10/11 9:45:50				印刷	153	ProjectName	ectNo:153LogStatus...	OK
2019/10/11 9:45:37				クライアントの処理結果	31	エラー発生Proc...	ProjectNo:31LogStat...	Exception
2019/10/11 9:44:44				印刷	9	ProjectName	OKDateTime:2019/10...	OK
2019/10/11 9:43:42				保存	162	ProjectName	162LogStatus:OKDa...	OK
2019/10/11 9:43:33				印刷	182	ProjectName	182LogStatus:OKDa...	OK

抽出したログはメニューより CSV ファイルに保存できます。



時間毎の同時接続数を調べることもできます。 → サーバー負荷やライセンス管理に有効



8. おわりに

世の中が絶え間なく変化し続けることを誰も止めることはできない。

企業は世の変化の影響を受けて、これまた変化し続けないと衰退する。

企業の変化は社内業務の変化に直結する。

システムも永遠ではなく生きた有機体のようにしなやかに強かに成長し続ける宿命にある。

さらにその変化のスピードも一段と速くなっている時代である。

このような忙しい時代を **dbSheetClient** を活用して楽しく乗り切ろうではありませんか。

本書が皆様方の業務改善に少しでもお役に立てることを心より期待いたします。

《原作者紹介》

小澤 光司(おざわ こうじ)

新卒で商社に入社し、営業部門に17年在籍。営業に関する業務に精通。その後、情報システム部門に異動、現在に至る。元営業マンでシステム開発未経験者の原作者が、現場の要請に密着した業務改善の必要に迫られて導入したのが「dbSheetClient」。業務のツボを見抜き、それを Excel 上で表現する技法は利用者をして感嘆させ続けてきた。dbSheetClient に出会い、その技法をシステム構築へと昇華できることに無常の喜びを感じ、作りに作ったプロジェクト200本。導入から7年間でこの実績は超々パワーユーザー級。趣味は業務改善。

株式会社ニューコム Copyright 2020 NEWCOM Inc. All right reserved.

本 社 〒330-0061 さいたま市浦和区常盤 7-3-16 ジブラルタ生命浦和ビル

HP:<https://www.newcom07.jp> Email:ncm.contact@newcom07.jp

TEL:048-815-8460 FAX:048-825-5518
