

JBAT

 **Customer Vision**
Qanat 2.0



ごあいさつ

このたびは弊社製品 Qanat 2.0 をご購入いただきまして、誠にありがとうございます。

(以下、Qanat 2.0 を「本製品」と呼び、ご説明して参ります。)

本製品は、基幹系や情報系のデータベースにある様々なデータをフィルタ群やスクリプトを使って自由に加工し、転送するデータ転送ツールです。

なお、表示画面などは操作の一例として掲載しているものです。お客様のご使用環境によっては、画面に表示される内容が異なる場合がありますので、ご了承ください。

著作権 / ご注意



本書に記載されている会社名、製品名は、それぞれ各社の商標または登録商標です。

本書の内容の一部または全部を無断で複写転載することを禁じます。

本書に掲載の内容および製品の仕様などは、予告なく変更されることがあります。

本書の内容は万全を期して作成しておりますが、万一ご不明な点や誤り、記載もれ、乱丁、落丁などお気づきの点がございましたら、弊社までご連絡ください。

本書の表記方法について

	ハードウェアやソフトウェアの損害やエラーの発生を防止するために、必ず守っていただきたい情報を記載しています。
	特定のテーマに関する補足情報を記載しています。
メニュー、アイコン、ボタン、ウィンドウ、タブ	[] で囲んで表記します。 (例) [OK] ボタンをクリックします。
キーボード上のキー	< > で囲んで表記します。 (例) キーボードの <Tab> キーを押します。
参照先	章、節、項は『』、見出しは「」で囲んで表記します。 (例) 『2章1 ログインとログアウト』を参照してください。

目次

ごあいさつ	i
版權 / ご注意	i
本書の表記方法について	i
目次	ii
はじめに	1
本書について	1
実習いただく前に	1
第 1 章 基本編	3
1-1:データ転送定義を作成する(データベース)	4
1-2:データ転送定義を作成する(ファイル).....	13
1-3:データ転送定義を作成する(SALESFORCE).....	38
1-4:スクリプト定義を作成する.....	45
1-5:QanatExecuteを使用する	49
1-6:メール送信定義を作成する.....	51
第 2 章 - 応用編 -	56
2-1:便利なフィルタ (応用)	57
2-2:オリジナルテーブルを作成する	62
2-3:オリジナルファイルを作成する	68
2-4:スケジュールを作成する	73
2-5:ファイルトリガーを作成する	78
2-6:全銀協データを扱う	83
2-7:作成したファイルを送信する	105
2-8:Webサービスを実行する	123

はじめに

本書について

本書では、本製品の基本操作や、より便利な機能の使用方法について手順を追って紹介しています。実際に製品を操作しながら読み進めていただくことで、よりいっそうご理解いただくことができます。

実習いただく前に

次の点にご注意ください。

運用中データ（本番業務データ）を利用した実習は推奨できません。実習には本書サンプルデータをご利用いただくか、またはお客様ご自身でサンプルデータをご用意ください。データベース名、スキーマ名、テーブル名、フィールド名は CV 管理ツールで追加する必要があります。

サンプルデータは CDROM の以下のフォルダに収められています。

- ・ **DB テーブル** : **商品マスター**
/Sample/Tutorial/db2_syoumas.sql
- ・ **DB テーブル** : **売り上げ実績**
/Sample/Tutorial/db2_uriage.sql
/Sample/Tutorial/db2_uriagetable.sql (空データ)
- ・ **DB テーブル** : **振込データ (全銀協用データ)**
/Sample/Tutorial/db2_zengin.sql
- ・ **DB テーブル** : **振込データ入力用 (全銀協用データ テーブル作成のみ)**
/Sample/Tutorial/db2_zengin_nodata.sql
- ・ **DB テーブル** : **取引先**
/Sample/Tutorial/db2_customer.sql
- ・ **CSV** : **商品マスター**
/Sample/Tutorial/商品マスター.csv
- ・ **CSV** : **売り上げ実績**
/Sample/Tutorial/売り上げ実績.csv
- ・ **XML** : **売上情報**
/Sample/Tutorial/売上.xml
/Sample/Tutorial/売上情報.xml
- ・ **固定長** : **全銀協フォーマット**
/Sample/Tutorial/zengin.data
- ・ **EXCEL** : **商品マスター**
/Sample/Tutorial/商品マスター.xls
- ・ **EXCEL** : **売上データ**
/Sample/Tutorial/売上データ.xls
- ・ **EXCEL** : **売上データテンプレート**
/Sample/Tutorial/売上データテンプレート.xls

- ・ **MAIL** : 社員マスター
/Sample/Tutorial/社員マスター.csv

DB テーブルのサンプルデータは IBM DB2 V9.5 (EE/WE) に対応しています。

実習の結果生じたデータ破損やデータ環境および運用中のデータ転送定義等への障害については弊社では責任を負いかねます。よって、影響を避けるために本書で利用する様々なデータや設定、生成物等 (データ転送定義など) については本書でのみ利用するとの前提のもと、ご説明を進めてまいります。

第 1 章

第 1 章 基本編

1-1: データ転送定義を作成する(データベース)

ここではデータベースに対してデータの読み取り/書き込みを行うデータ転送定義の作成を習得します。

読取ったデータはフィルタを利用してデータの加工を行った後、対象のデータベースへ挿入します。

目的	作成されるもの
<ul style="list-style-type: none">データベースに対して読み書きをするフィルタを利用してデータ加工をする	<ul style="list-style-type: none">データ転送定義

事前修了しておきたい実習項目

ありません。

実習のシナリオ

“東日本 DB”データベースの“商品マスター”テーブルからデータを読み込み、

“西日本 DB”データベースの“商品マスター”テーブルへデータを新規挿入/更新します。

商品マスターには単価フィールドがありますが、西日本では東日本より5%高く設定します。

作成したデータ転送定義を保存しテスト実行を利用して検証した後、手動で実行します。

手順1： 本製品を起動し [デザイナー] タブを選択します。

入力元（データ読み込み側）と出力先（データ書き込み側）には RDB を利用します。

**手順2： RDBから入力元テーブルを選択します**

まず、入力元ではデータの読み込み先であるデータベースのテーブルを一つ選択します。



[RDB]と書かれたアイコンを右クリックし[複数テーブル選択]をクリックします。



表示されているフォルダ名は、CV 管理ツールで付けられたデータベースの論理名です。目的のデータベースからスキーマ名を選択し、データを読み込みたいテーブルを選択して中央にドラッグ&ドロップします。

- ・ 選択フィールドを絞る場合 中央の配置したテーブルのフィールドをフィールド選択にドラッグ&ドロップ
- ・ フィールド全選択 中央右上の全設定をクリック

今回は全設定をクリックします。(全設定をクリックすると選択テーブルの全フィールドが入力フィールドに選択されます。)



[OK]を押下するとフィールド選択に配置したフィールドがマッパー画面に表示されます。

ここで分離レベルを選択します。この実習では分離レベルは既定値の[指定しない]を指定します。

[OK]ボタンをクリックし転送元選択画面を閉じます。

! 実際の運用ではデータの特性に応じて適切な分離レベルを選択してください。

手順3： RDBから出力先テーブルを選択します

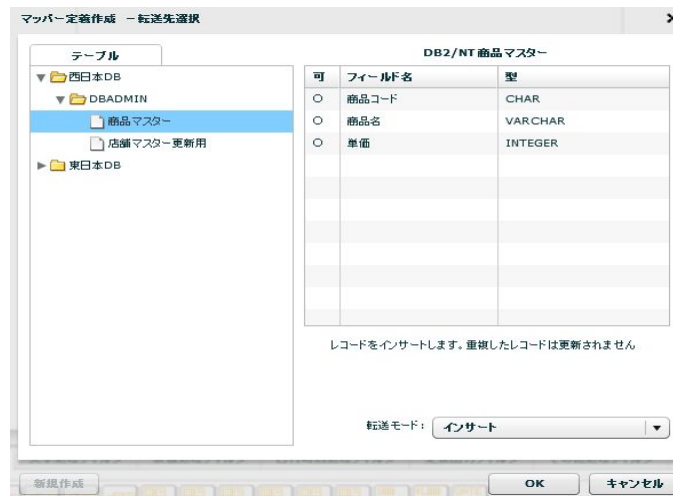
データの出力先テーブルを選択します。

手順2では“東日本DB”からデータを読み込むよう設定しました。手順3では出力先として“西日本DB”に取得したデータを転送するように指定します。

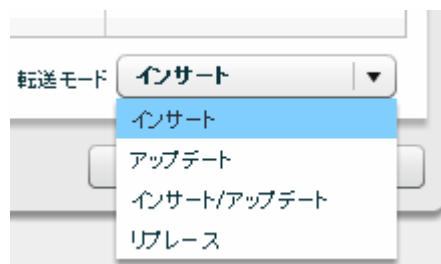
画面右側の[RDB]アイコンを右クリックし[1 テーブル選択]をクリックします。



[転送先選択]画面で“西日本 DB”を選択し“商品マスター”テーブルを選択することで転送先を指定します。



[転送先選択]画面では、転送モードを指定します。この実習では[インサート/アップデート]を選択します。



転送モードは次から一つ選択します。

- ・インサート : データの新規挿入
- ・アップデート : 更新キー指定によるデータ更新
- ・インサート/アップデート : 更新キーが一致する既存データは更新し、それ以外は新規挿入
- ・リプレース : 対象のテーブル内のデータを全て削除した後、新規挿入

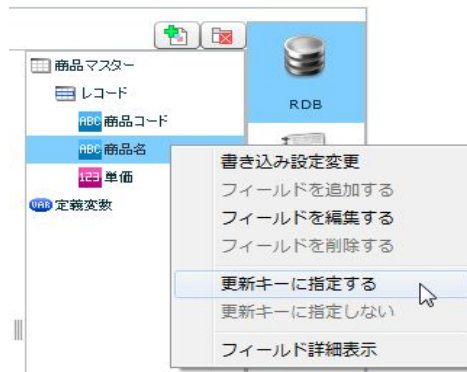
[OK]ボタンをクリックし転送先選択画面を閉じます。

手順4： データマッピングをします

入力元と出力先のそれぞれのデータフィールドをつなぎ、データ転送設定をします。

手順3で転送モードを[インサート/アップデート]を選択しました。転送モードにアップデートを含む場合更新キーを指定する必要があります。

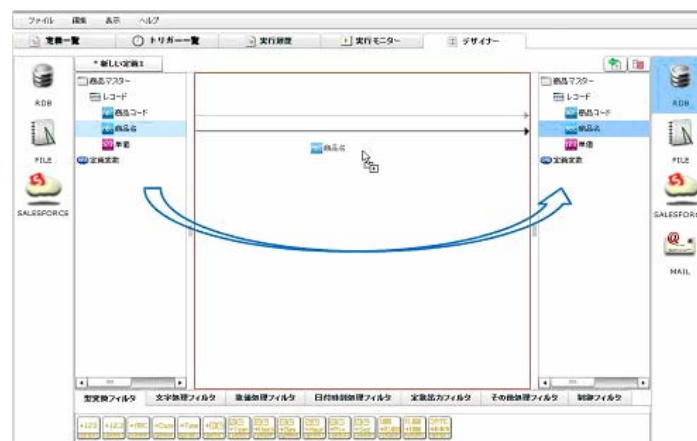
出力先のフィールドのうち、更新キーとなるフィールドを右クリックし、メニューから[更新キーに指定する]をクリックします。入力元と出力先で一致する更新キーが見つかった場合は更新し、一致しないデータの場合は新規に挿入されます。



出力先に主キー設定があるにも関わらず更新キーを指定しなかった場合や、転送モードが[インサート]である場合などに同一のキー値を持つデータを転送しようとするとき一意キー違反が発生する場合があります。この場合、本製品はエラーを記録し場合により処理が停止します。

更新キーは忘れずに必ず指定するようにしてください。

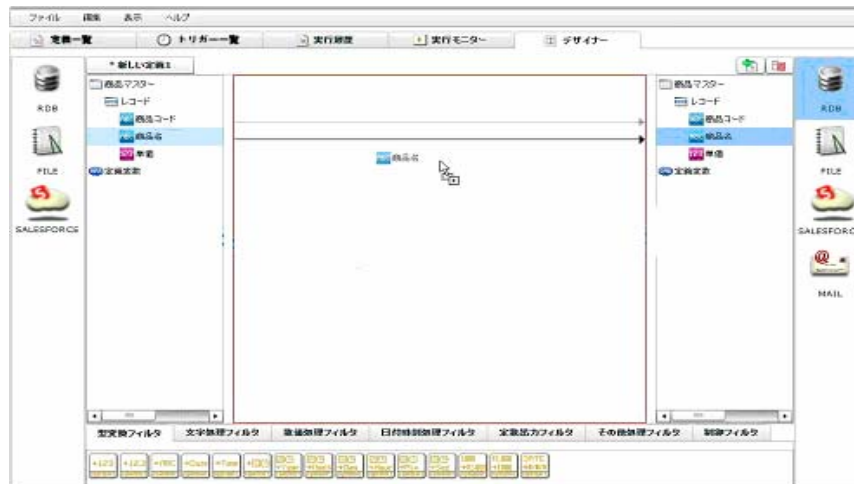
この実習では“東日本 DBの商品マスター”から“西日本 DBの商品マスター”へデータを転送することです。そのためには画面左側のフィールドを、右側の同名のフィールドに対してマウスでドラッグ&ドロップしマッピングを一つ一つ作成します。



単価フィールドは西日本のほうを5%高く設定したいため、東日本DBの単価にその分上乘せた値を西日本DBの単価フィールドにマッピングします。

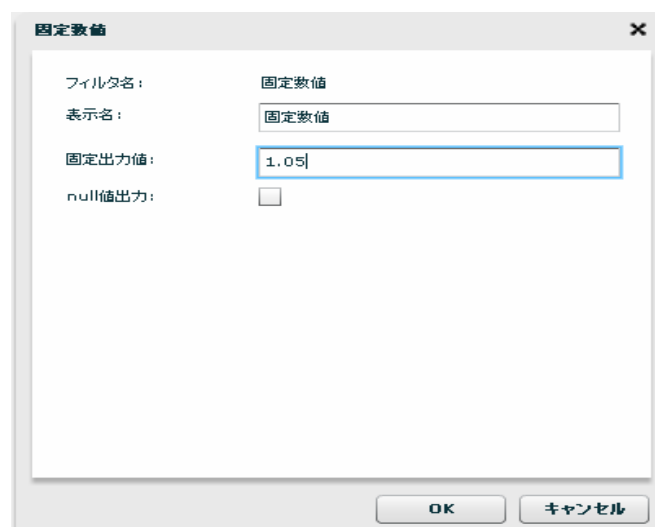
このマッピングには“5%”という値を持たせた固定数値フィルタと、掛け算フィルタを利用し表現します。

まず、画面下部の[定数出力フィルタ]タブを選択し、固定数値フィルタを画面中央のキャンバスにドラッグ&ドロップします。

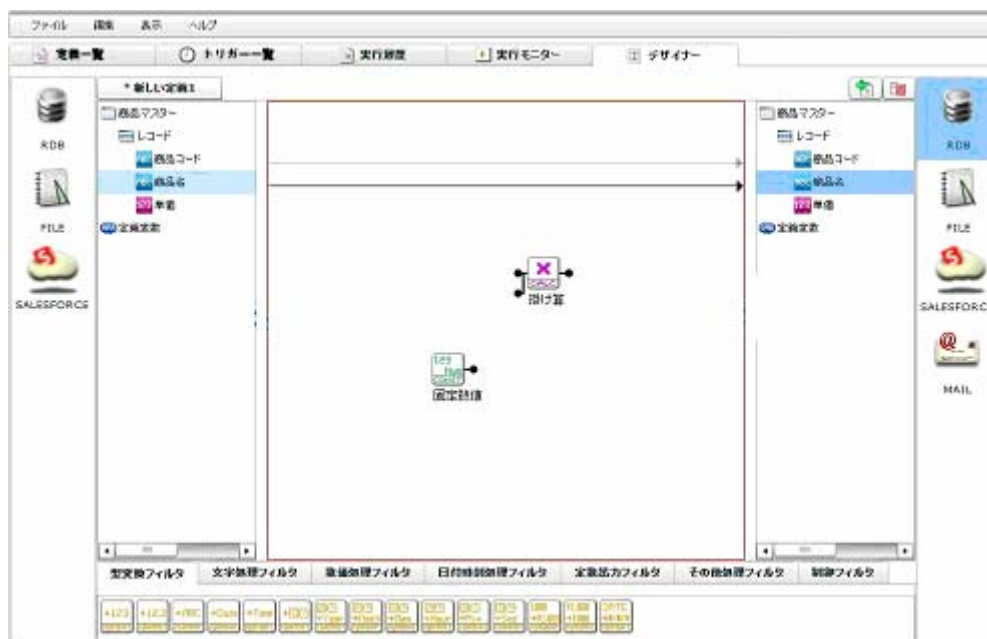


固定数値フィルタを配置すると次のようなプロパティ画面が表示されます。

[固定出力値]欄に5%である“1.05”と半角で入力し、[OK]をクリックします。

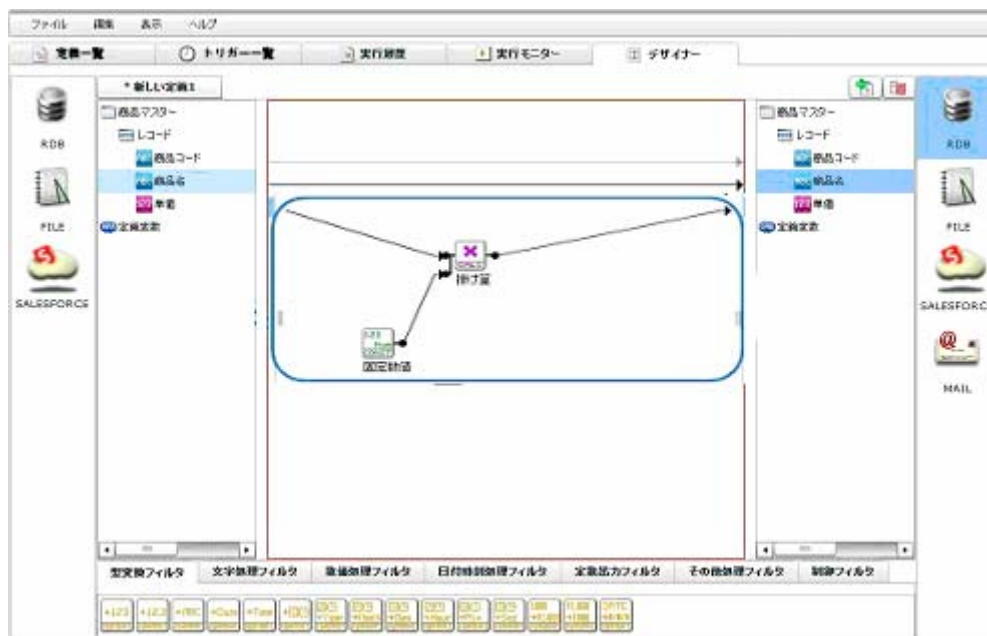


次に[数値処理フィルタ]タブから、掛け算フィルタを選択し、同様にドラッグ&ドロップし、画面中央付近に配置します。



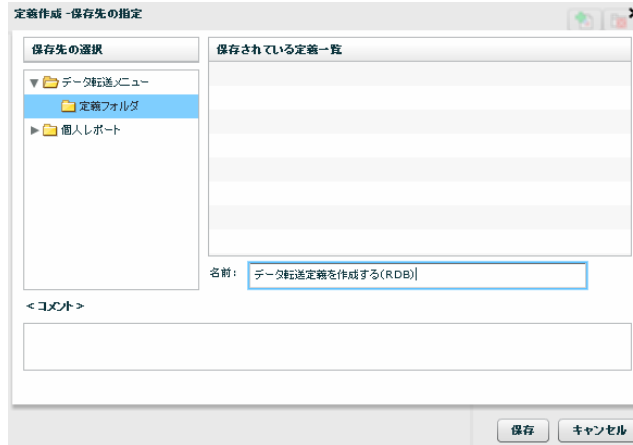
入力元の単価フィールドと、固定数値フィルタをそれぞれ掛け算フィルタにドラッグ&ドロップしお互いを結線（マッピング）します。

そして、掛け算フィルタを出力先の単価フィールドに対してマッピングすることで5%を乗じた値を出力先の単価フィールドにセットすることができます。



手順5: データ転送定義を保存します

メニューバーの[ファイル] [名前を付けて保存]をクリックします。次の画面のように保存先を選択し[名前]欄に定義名称、[コメント]欄にその定義の説明を入力し[保存]をクリックします。



手順6: データ転送定義の検証をします

データ転送定義をテスト実行します。

保存を終えたらテスト実行機能を利用し、エラーなく正常に動作することを確認します。

メニューバーの[表示] [テストログダイアログ]をクリックします。



テスト実行に利用するデータを指定し、[テスト実行]ボタンをクリックします。

画面下部のデータ表示部に結果が表示されることを確認します。



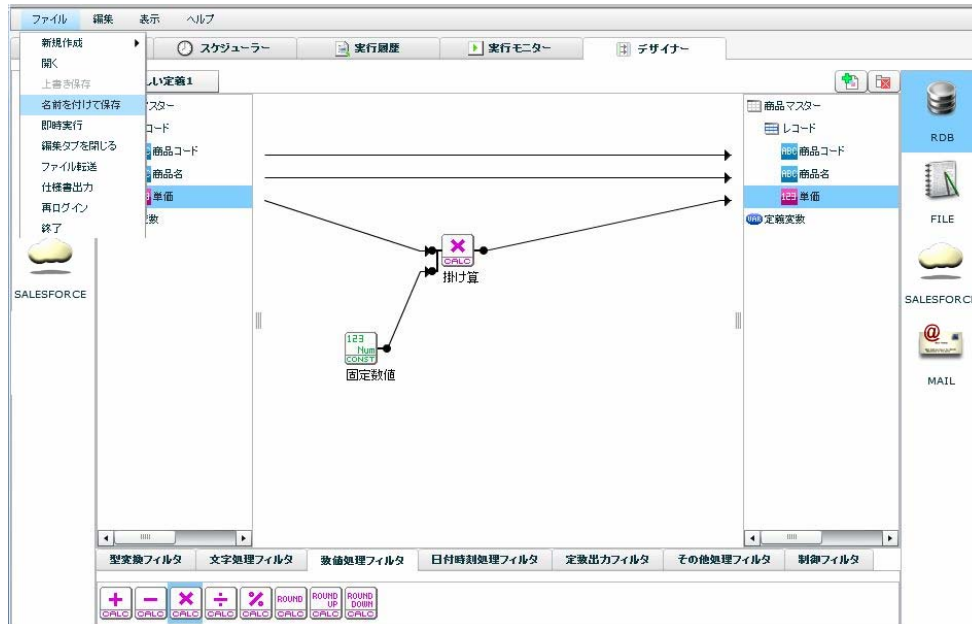
[エラー]欄に赤字でエラー内容が表示される場合、データ転送定義に誤りがあるか、または存在しないデータを指定している場合があります。



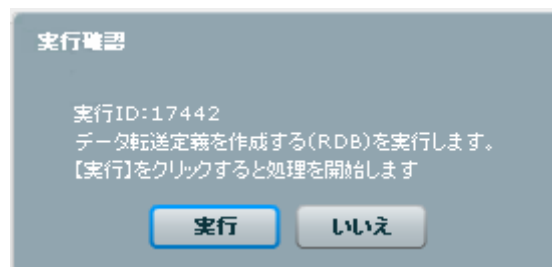
テスト実行では出力先リソースは更新されません。

手順 7: データ転送定義を実行します

先ほど作成/保存した定義を画面のように開いた状態でメニューバーの[ファイル] [即時実行]をクリックします。



実行確認画面では[実行]をクリックし実行を開始します。



西日本 DB データベースの商品マスターテーブルを参照し、結果を確認します。

東日本 DB の商品マスターテーブルの各商品データと、それぞれ 5% だけ高く設定された単価が表示されているはずです。

1-2:データ転送定義を作成する(ファイル)

CSV ファイル

CSV ファイルリソースに対してデータの読み取り/書き込みを行うデータ転送定義の作成を習得します。

読取ったデータはフィルタを利用してデータの加工を行った後、対象の CSV ファイルリソースへ更新します。

目的	作成されるもの
<ul style="list-style-type: none">・ CSV ファイルリソースを利用しデータの読み書きをする・ フィルタを利用してデータ加工をする	<ul style="list-style-type: none">・ データ転送定義

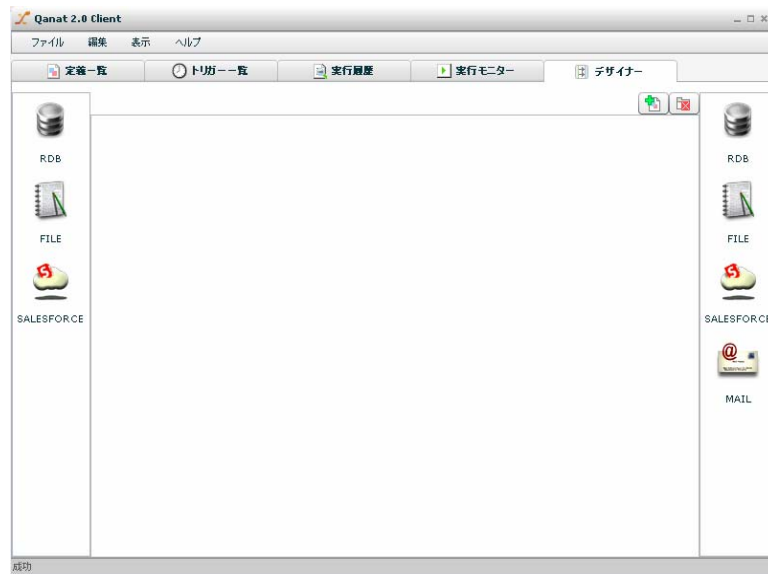
事前修了しておきたい実習項目

ありません。

実習のシナリオ

“東日本ファイルリソース”フォルダの“商品マスター”ファイルからデータを読み込み、
“西日本ファイルリソース”フォルダの“商品マスター”ファイルへデータを新規挿入します。
商品マスターには単価フィールドがありますが、西日本では東日本より5%高く設定します。

作成したデータ転送定義を保存し手動で実行します。

手順 1： 本製品を起動し [デザイナー] タブを選択します。

入力元（データ読み込み側）と出力先（データ書き込み側）にはファイルを利用します。

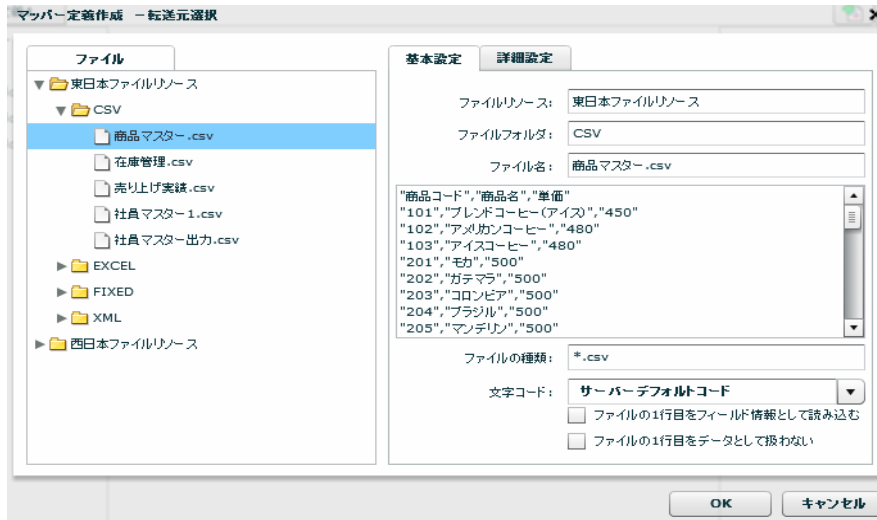
まず、入力元ではデータの読み込み先であるファイルの一つを選択します。

手順 2： ファイルリソースから入力元ファイルを選択します

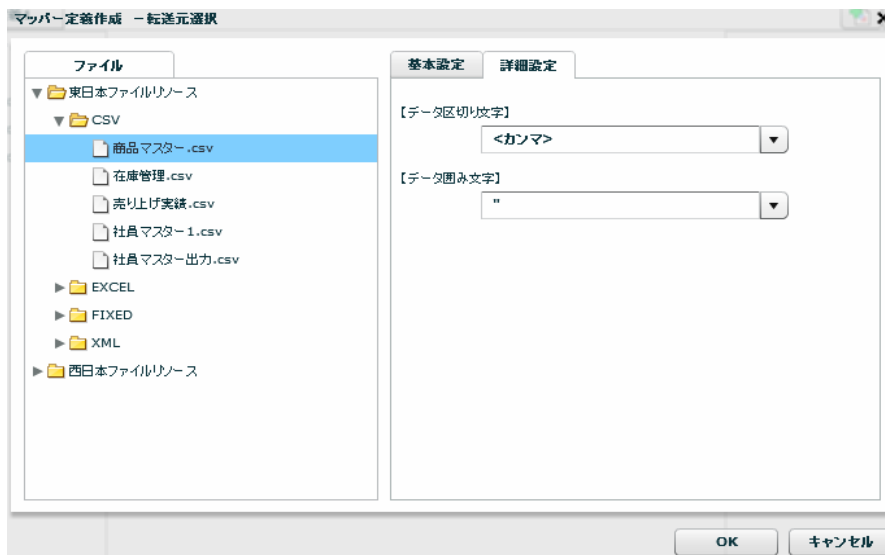
[FILE]と書かれたアイコンを右クリックし[CSV ファイル選択]をクリックします。

表示されているフォルダから目的のファイルを選択します。右側にはプレビューが表示され、データの区切り文字や囲み文字を確認します。

CSV ファイルにヘッダ行が含まれる場合は [ファイルの1行目をデータとして扱わない] のチェックを入れます。また、[ファイルの1行目をフィールド情報として読み込む] にチェックを入れることで、ファイルの1行目のデータを利用して読み込みフィールドを作成します。



[詳細設定] タブをクリックし、先ほど確認したデータ区切り文字や囲み文字を指定します。



[OK] ボタンをクリックし転送元選択画面を閉じます。



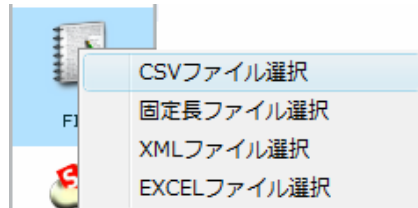
データ区切り文字や囲み文字がコンボボックスの一覧にない場合、任意の文字を入力することで別の文字を指定することができます。

手順3: CSVファイルリソースから出力先ファイルを選択します

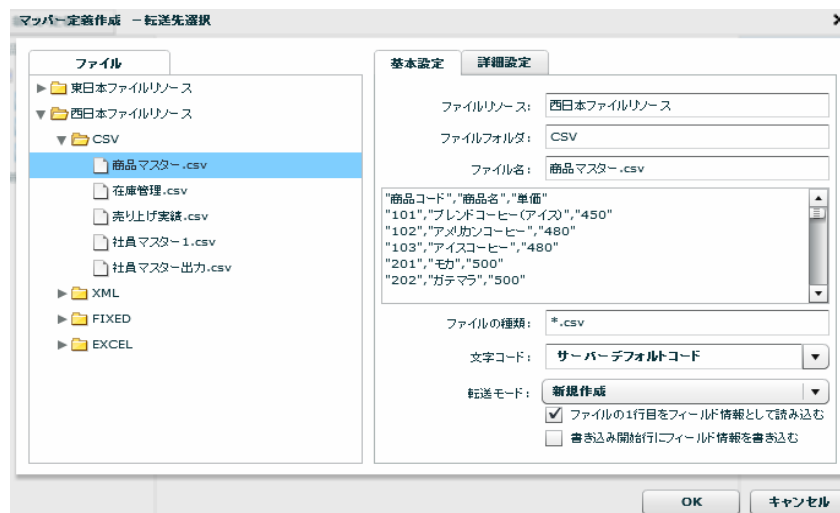
データの出力先ファイルを選択します。

手順2では“東日本リソースフォルダからデータを読み込むよう設定しました。手順3では出力先として“西日本リソースフォルダに取得したデータを転送するように指定します。

画面右側の[FILE]アイコンを右クリックし[CSVファイル選択]をクリックします。



[転送先選択]画面で“西日本リソースフォルダから“商品マスター”テーブルを選択し転送先ファイルを指定します。



この実習では西日本リソースフォルダに既に存在する商品マスターファイルに追加書き込みします。

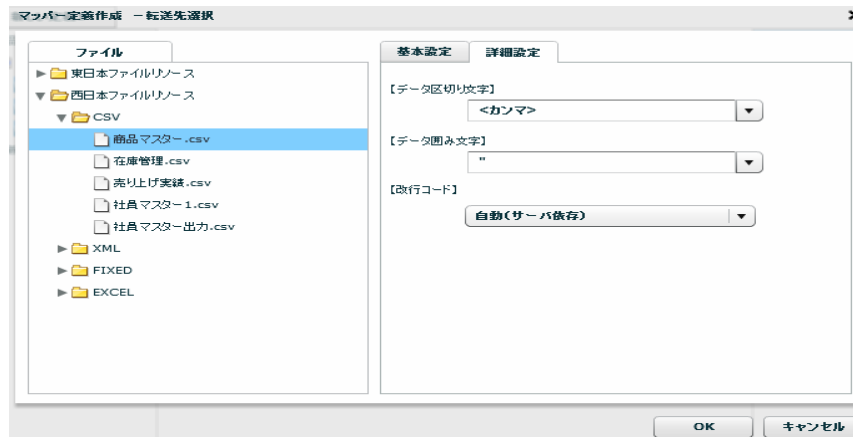
ヘッダ情報は不要になるため、[書き込み開始行にフィールド情報を書き込む]のチェックを外します。

また、[ファイルの1行目をフィールド情報として読み込む]にチェックを入れることで、

ファイルの1行目のデータを利用して書き込みフィールドを作成します。

[詳細設定]タブをクリックしデータ区切り文字と囲み文字を指定します。

では、転送モードを指定します。この実習では[追加書き込み]を選択します。

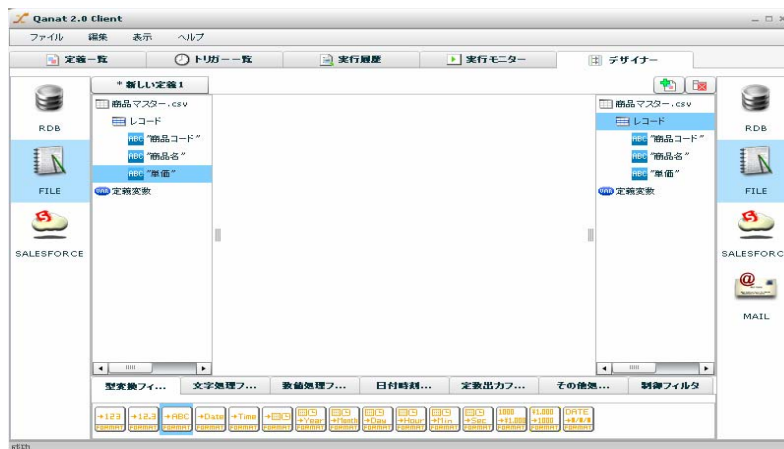


❗ データ区切り文字や囲み文字がコンボボックスの一覧にない場合、任意の文字を入力することで別の文字を指定することができます。

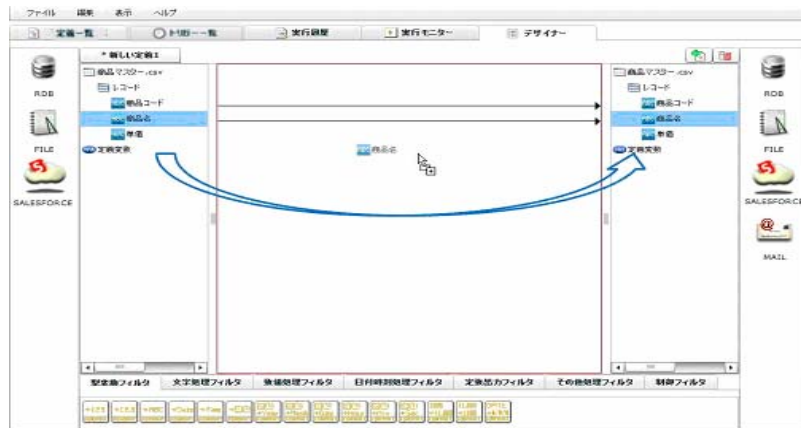
[OK] ボタンをクリックし転送先選択画面を閉じます。

手順4： データマッピングをします

入力元と出力先のそれぞれのデータフィールドをつなぎ、データ転送設定をします。



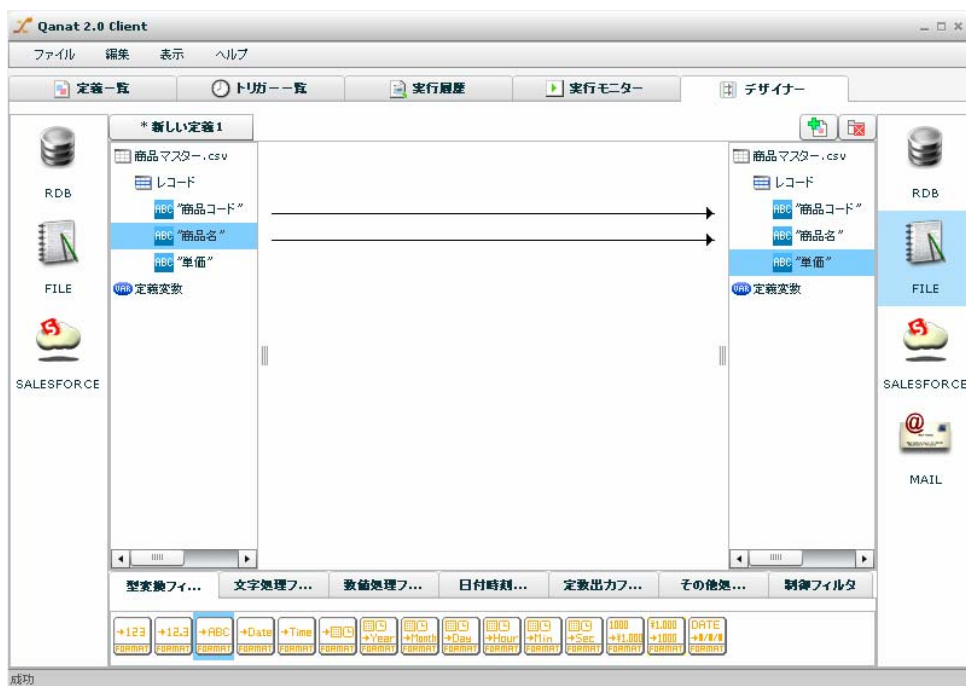
この実習では“東日本リソースフォルダの商品マスターファイル”から“西日本リソースフォルダの商品マスターファイル”へデータを転送することです。そのためには画面左側のフィールドを、右側の同名のフィールドに対してマウスでドラッグ&ドロップしマッピングを一つ一つ作成します。



単価フィールドは西日本のほうを5%高く設定したいため、東日本リソースフォルダの商品マスターの単価にその分上乗せした値を、西日本リソースフォルダの商品マスターの単価フィールドにマッピングします。

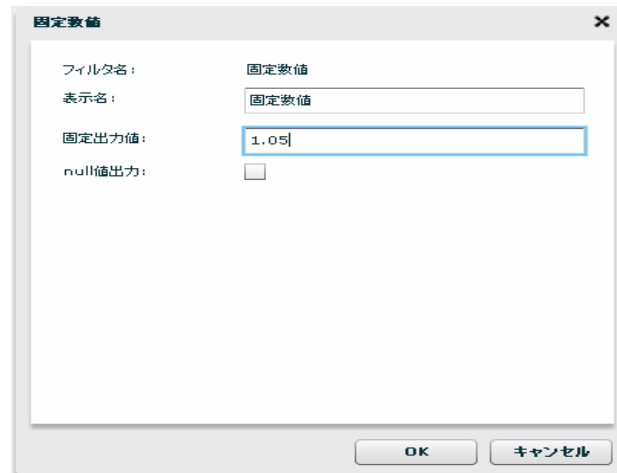
このマッピングには“5%”という値を持たせた固定数値フィルタと、掛け算フィルタを利用し表現します。

まず、画面下部の[定数出力フィルタ]タブを選択し、固定数値フィルタを画面中央のキャンバスにドラッグ&ドロップします。

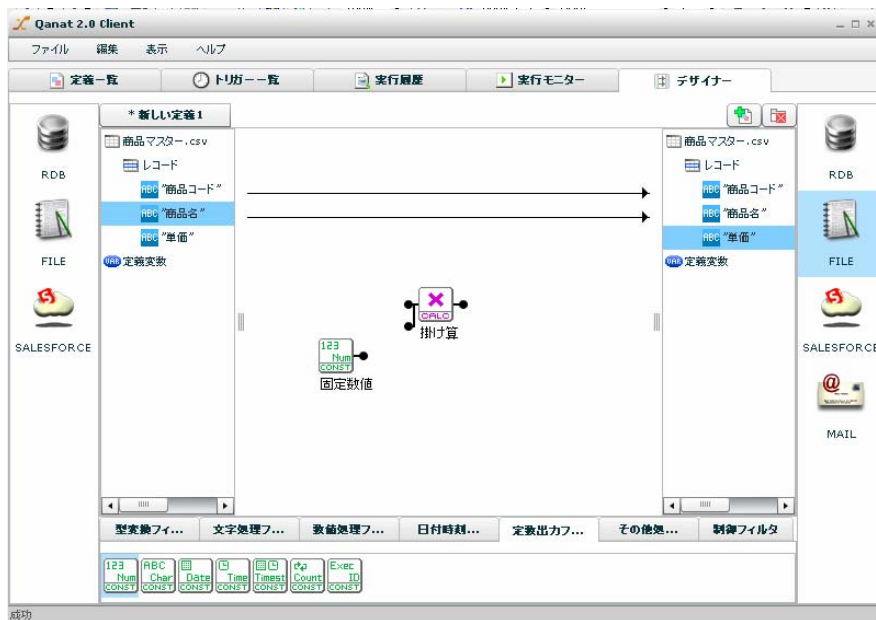


固定数値フィルタを配置すると次のようなプロパティ画面が表示されます。

固定出力値欄に5%である“1.05”と半角で入力し、[OK]をクリックします。

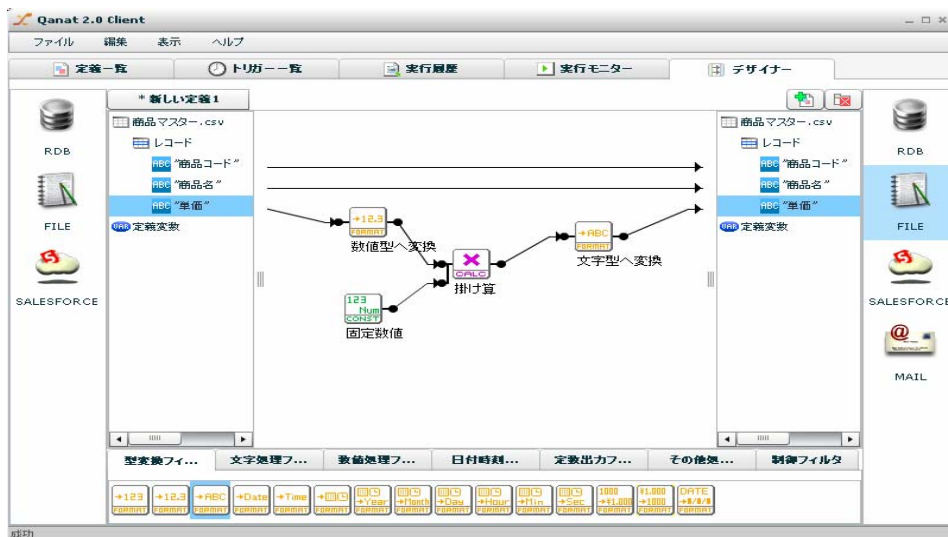


次に[数値処理フィルタ]タブから、掛け算フィルタを選択し、同様にドラッグ&ドロップし、画面中央付近に配置します。



入力元の単価フィールドと、固定数値フィルタをそれぞれ掛け算フィルタにドラッグ&ドロップしお互いを結線（マッピング）します。

そして、掛け算フィルタを出力先の単価フィールドに対してマッピングすることで5%を乗じた値を出力先の単価フィールドにセットすることができます。



CSV ファイルリソースでは各データフィールドを文字型として扱うため、整数型への変換フィルタを挿入します。

- ❗ 定義保存、テスト実行、定義実行はチュートリアル『1-1：データ転送定義を作成する（データベース）』を参照してください。

XML ファイル

XML ファイルリソースに対してデータの読み取り/書き込みを行うデータ転送定義の作成を習得します。

読取ったデータはフィルタを利用してデータの加工を行った後、対象の XML ファイルリソースへ更新します。

目的	作成されるもの
<ul style="list-style-type: none">XML ファイルリソースを利用しデータの読み書きをするフィルタを利用してデータ加工をする	<ul style="list-style-type: none">データ転送定義

事前修了しておきたい実習項目

ありません。

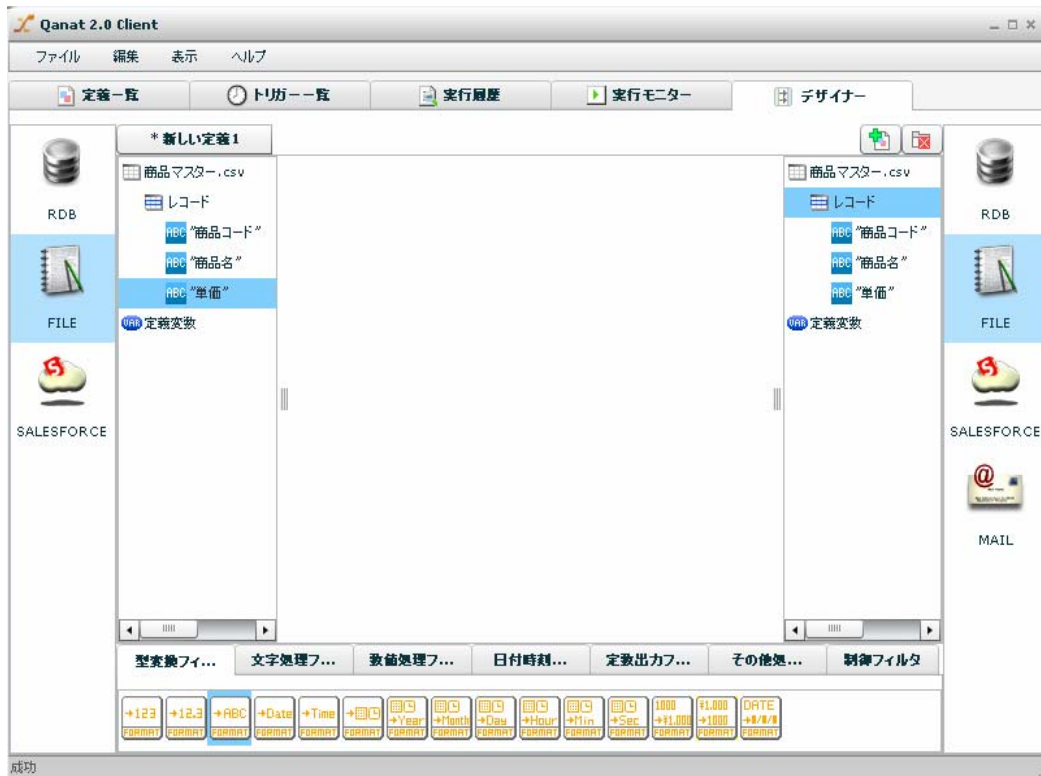
実習のシナリオ

“ XML ” フォルダの “ 売上情報 ” ファイルからデータを読み込み、

“ XML ” フォルダの “ 売上情報 2 ” ファイルへデータを新規挿入します。

売上情報には単価フィールドがありますが、西日本では東日本より 5% 高く設定します。

作成したデータ転送定義を保存し手動で実行します。

手順 1： 本製品を起動し [デザイナー] タブを選択します。

入力元（データ読み込み側）と出力先（データ書き込み側）にはファイルを利用します。

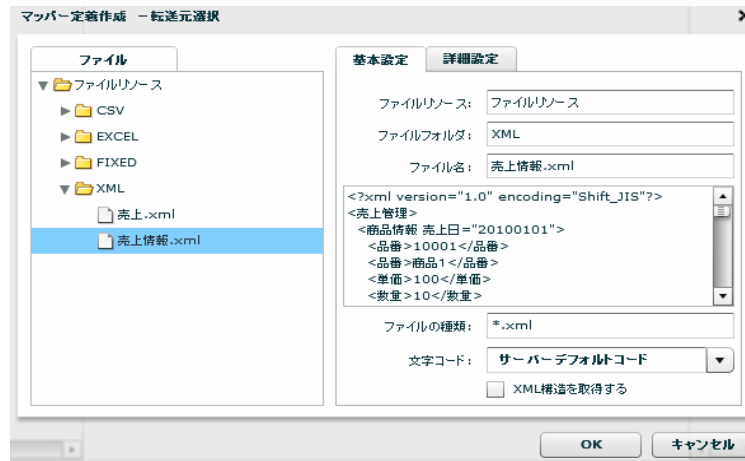
まず、入力元ではデータの読み込み先であるファイルの一つを選択します。

手順 2： ファイルリソースから入力元ファイルを選択します

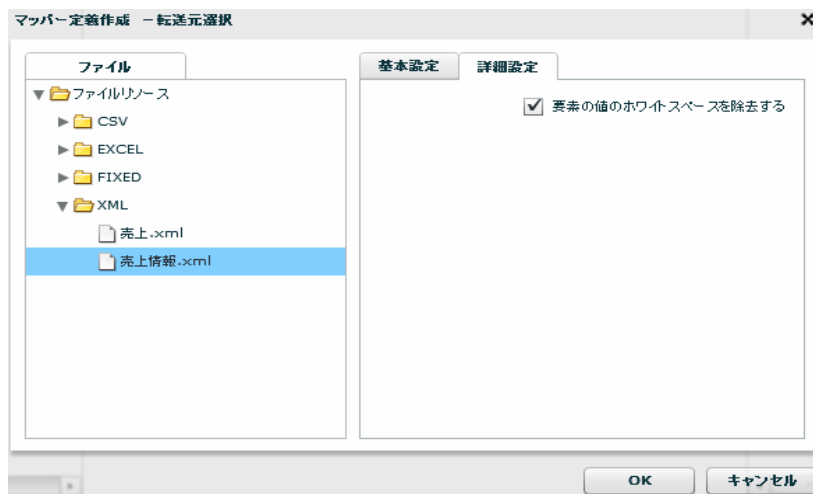
[FILE]と書かれたアイコンを右クリックし[XML ファイル選択]をクリックします。

表示されているフォルダから目的のファイルを選択します。右側にはプレビューが表示されるので要素内のホワイトスペース（タブやスペース）を確認します。

XML ファイルの構造を取得するには時間がかかる場合があるのであらかじめ [XML 構造を取得する] にチェックを入れます。



[詳細設定] タブをクリックし、先ほど確認した要素の値のホワイトスペースを除去するかを指定します。

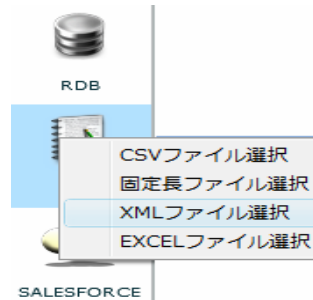


[OK] ボタンをクリックし転送元選択画面を閉じます。

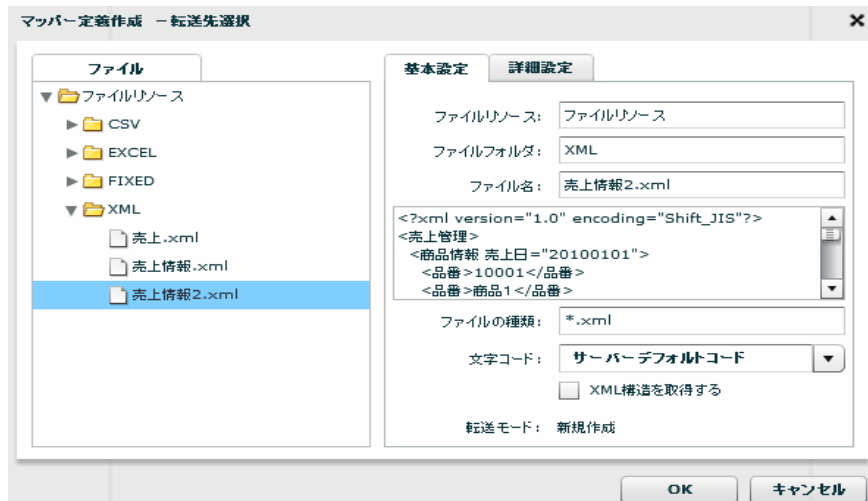
手順 3： XML ファイルリソースから出力先ファイルを選択します

データの出力先ファイルを選択します。

画面右側の[FILE]アイコンを右クリックし[XML ファイル選択]をクリックします。



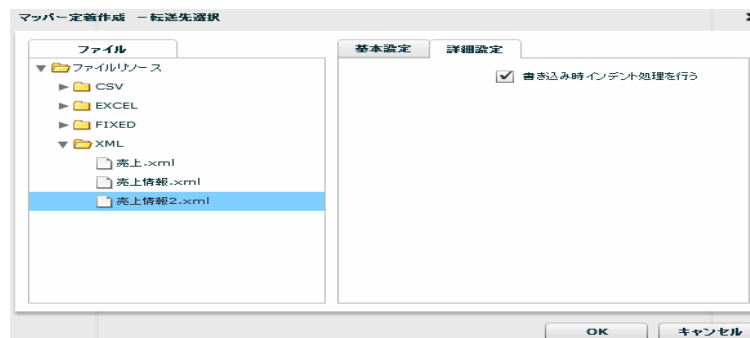
[転送先選択]画面で“XML”フォルダから“売上情報 2”テーブルを選択し転送先ファイルを指定します。



転送元と同様に [XML 構造を取得する] にチェックを入れます。

[詳細設定]タブをクリックし書き込み時にインデント処理を行うかを指定します。

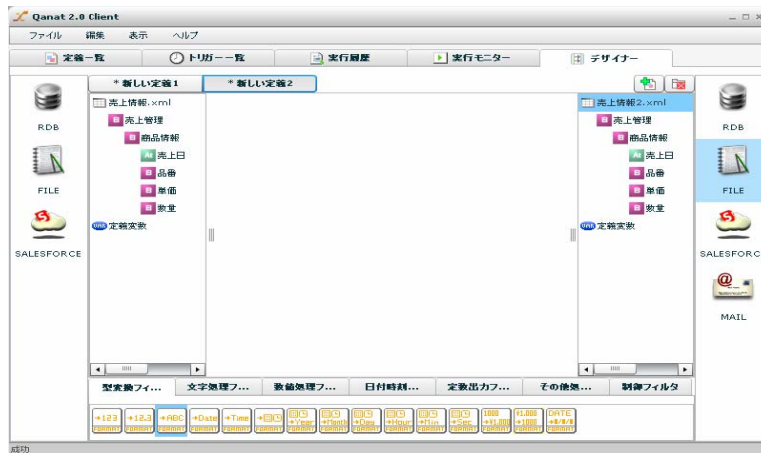
インデント処理を行うと要素の先頭が字下げされます。



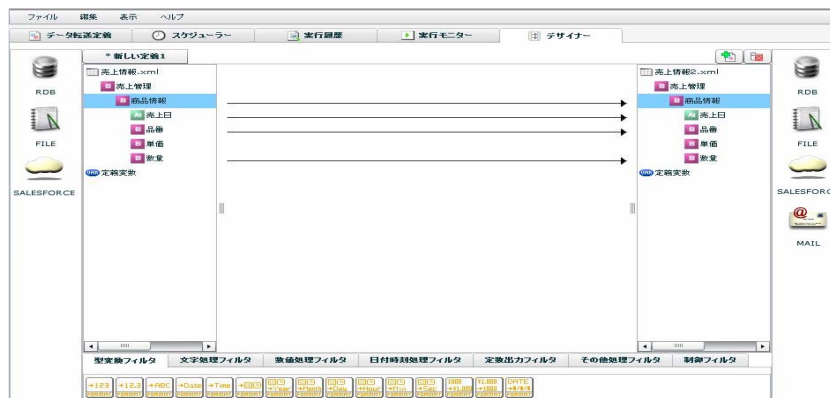
[OK] ボタンをクリックし転送先選択画面を閉じます。

手順4： データマッピングをします

入力元と出力先のそれぞれのデータフィールドをつなぎ、データ転送設定をします。



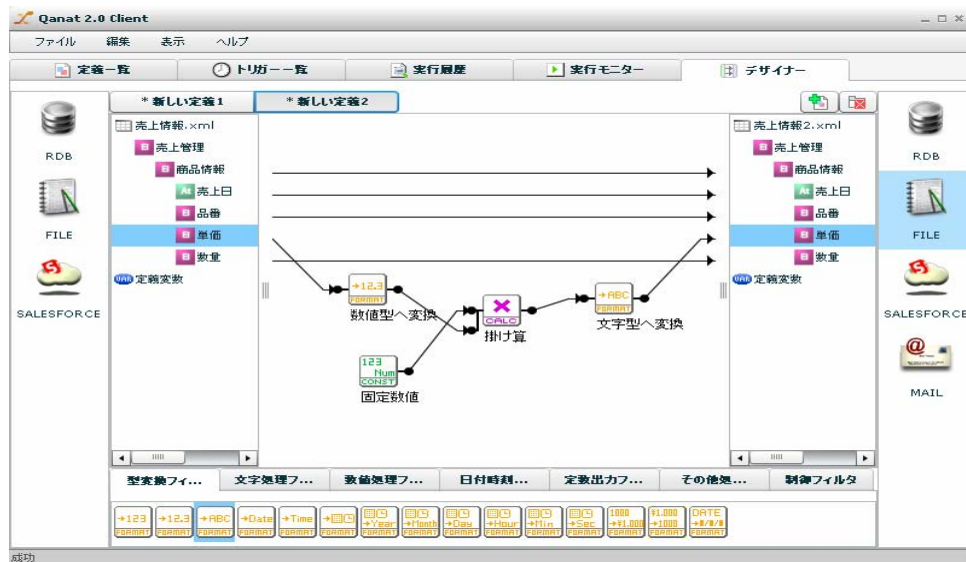
そのためには画面左側の転送したいフィールドを、右側の同名のフィールドに対してマウスでドラッグ&ドロップしマッピングを一つひとつ作成します。



単価フィールドは売上情報2のほうを5%高く設定したいため、売上情報の単価にその分上乗せした値を、売上情報2の単価フィールドにマッピングします。

このマッピングには“5%”という値を持たせた固定数値フィルタと、掛け算フィルタを利用し表現します。

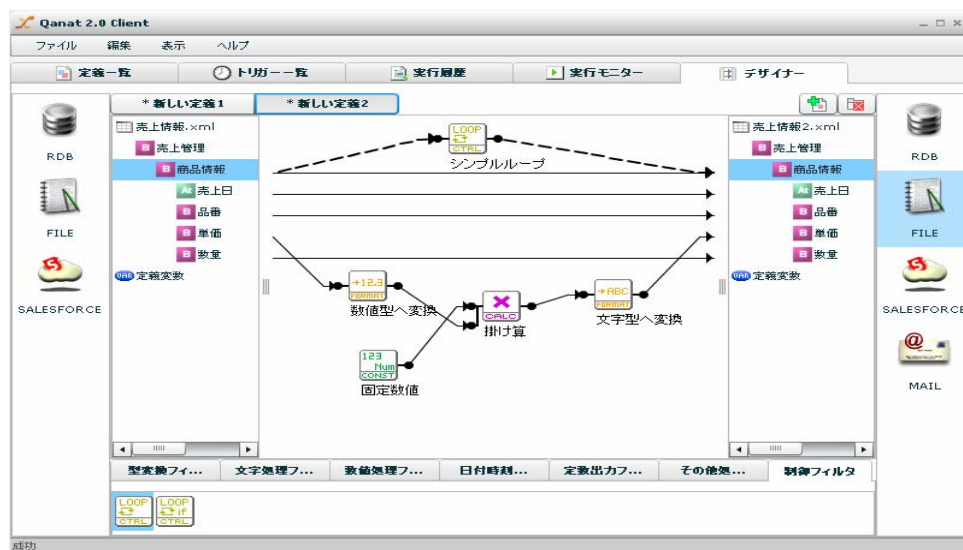
(詳しくは 1-1 データ転送定義を作成する(データベース)参照)



XML ファイルリソースでは各データフィールドを文字型として扱うため、数値型へ変換フィルタ、文字型へ変換フィルタを挿入してください。

最後に[制御フィルタ]タブをクリックし[シンプルループフィルタ]をマッパー上にドラッグ&ドロップします。

この[シンプルループフィルタ]と入出力の売上情報レコードを結線し、マッピングした全てのデータフィールドが転送されるようにします。



定義保存、テスト実行、定義実行はチュートリアル『1-1：データ転送定義を作成する（データベース）』を参照してください。

EXCEL ファイル

EXCEL ファイルリソースに対してデータの読み取り/書き込みを行うデータ転送定義の作成を習得します。

読取ったデータはフィルタを利用してデータの加工を行った後、対象の EXCEL ファイルリソースへ更新します。

目的	作成されるもの
<ul style="list-style-type: none">EXCEL ファイルリソースを利用しデータの読み書きをするフィルタを利用してデータ加工をする	<ul style="list-style-type: none">データ転送定義

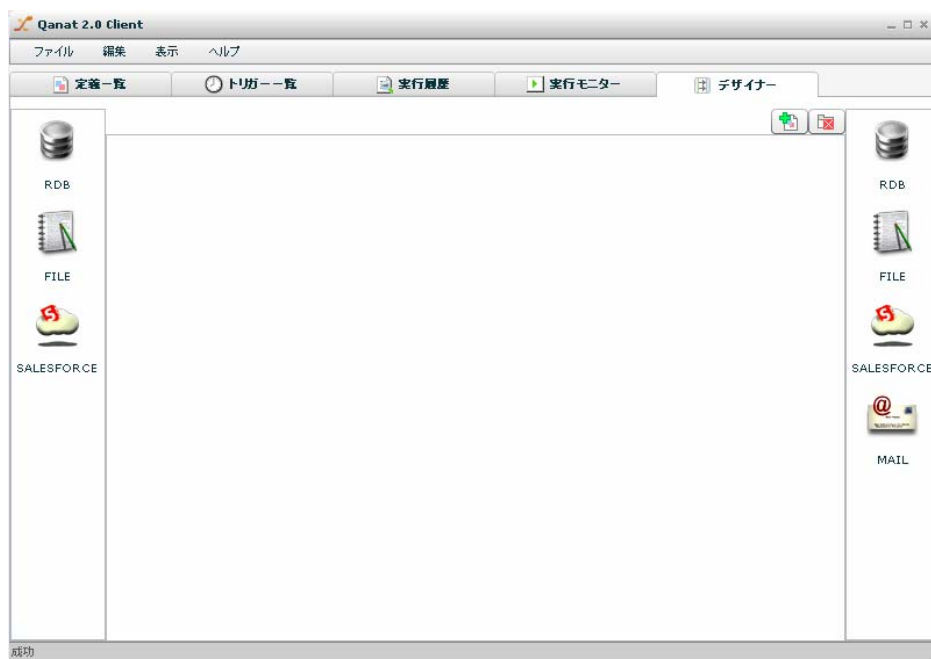
事前修了しておきたい実習項目

ありません。

実習のシナリオ

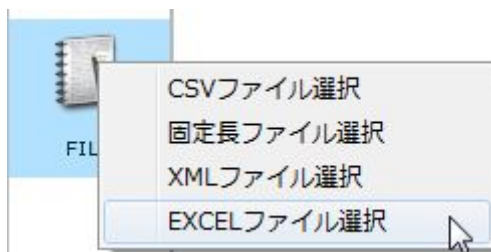
“ EXCEL ” フォルダの “ 商品マスター.xls ” ファイルからデータを読み込み、
“ EXCEL ” フォルダの “ 西日本商品マスター.xls ” ファイルへデータを新規作成します。
情報には単価フィールドがありますが、西日本では 5% 高く設定します。

作成したデータ転送定義を保存し手動で実行します。

手順 1： 本製品を起動し [デザイナー] タブを選択します。

入力元（データ読み込み側）と出力先（データ書き込み側）にはファイルを利用します。

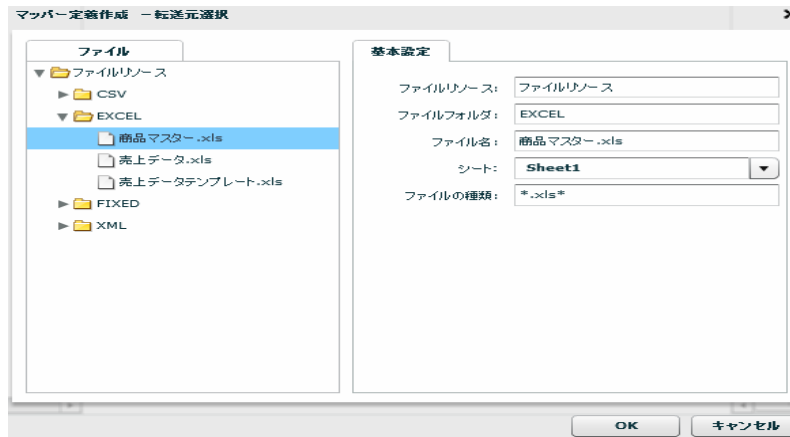
まず、入力元ではデータの読み込み先であるファイルの一つを選択します。

手順 2： ファイルリソースから入力元ファイルを選択します

[FILE]と書かれたアイコンを右クリックし[EXCEL ファイル選択]をクリックします。

表示されているフォルダから目的のファイルを選択します。

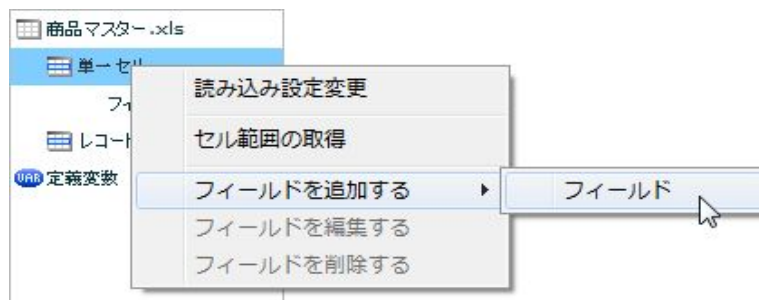
ファイルを選択するとシート名が選択されます。今回は「Sheet1」を選択します。



[OK] ボタンをクリックし転送元選択画面を閉じます。

読み込みセルを指定します。セルを指定するには、手動で行う方法と、EXCEL ファイルから読み取る方法が用意されています。読み込みでは、手動で行う方法で設定します。

はじめに単一セルの設定を行います。入力元の「単一セル」フィールドを選択状態にして右クリックメニューの「フィールドを追加する」「フィールド」をクリックします。

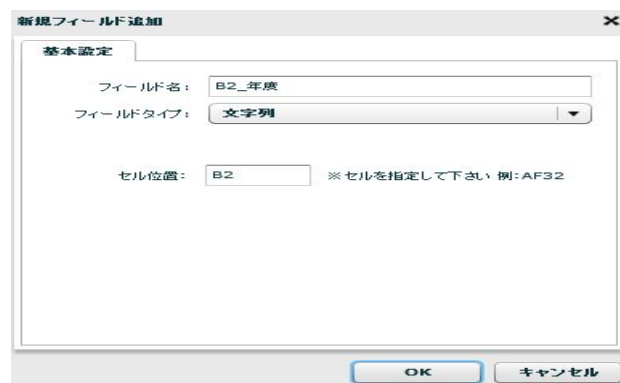


「新規フィールド追加」画面が表示されます。各プロパティを以下のように設定します。

フィールド名 B2_年度

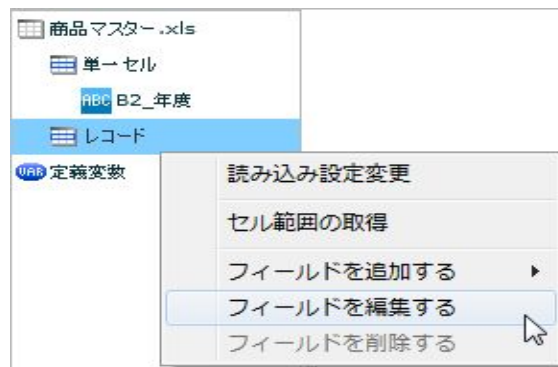
フィールドタイプ 文字列

セル位置 B2



[OK] ボタンをクリックし転送元選択画面を閉じます。

続いて、レコードセルを指定します。入力元の「レコード」フィールドを選択状態にして右クリックメニューの「フィールドを編集する」をクリックします。



各プロパティを以下のように設定します。

セル範囲 B4:D43

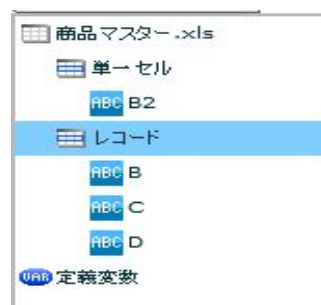
読み込み方向 縦

指定されたセル範囲を超えてもレコードが空になるまで読み込む いいえ

指定されたセル範囲内でレコードが空のとき 読込



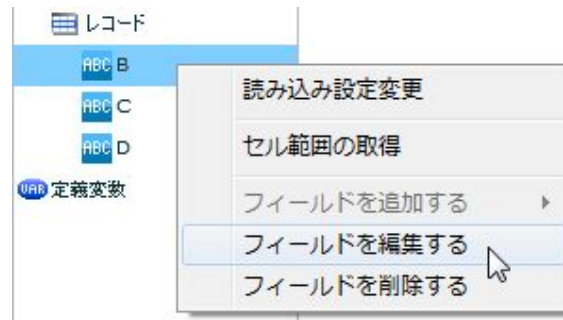
[OK]ボタンをクリックすると、「設定したレコード範囲で、フィールドを再設定しますか?」というメッセージが表示されるので、[はい]をクリックしてください。入力側の「レコード」フィールドの下に「B」「C」「D」フィールドが表示されます。



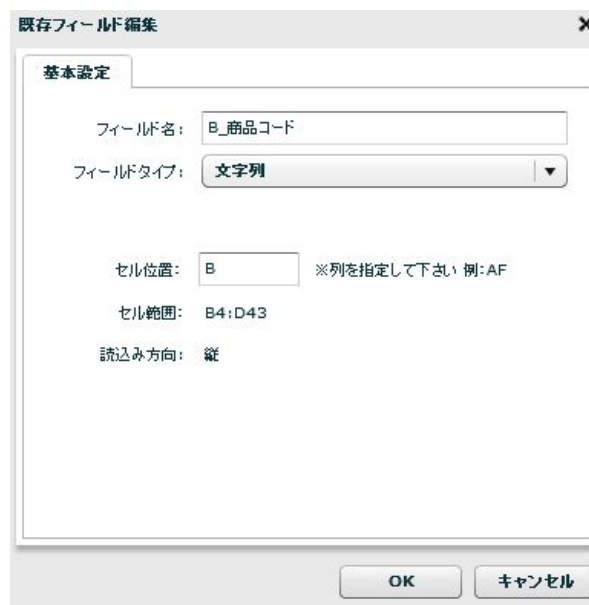
このままでも定義の実行に問題はありませんが、わかりやすくするため、各フィールドの表示名を変更します。以下のように変更します。

変更前の表示名	変更後の表示名
B	B_商品コード
C	C_商品名
D	D_単価

変更するフィールドを選択状態にして「フィールドを編集する」をクリックします。



「既存フィールド編集」画面が表示されるので、「フィールド名」を変更します。フィールド「B」は「B_商品コード」に変更します。



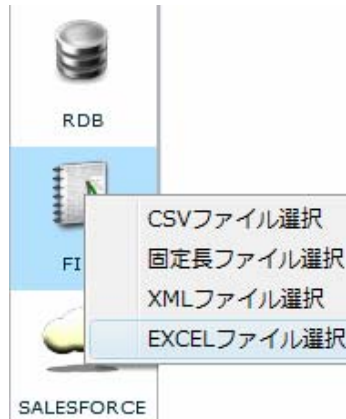
同様の手順で、フィールド「C」「D」のフィールド名を変更してください。

以上で入力元ファイルの選択は終了です。引き続き、出力先ファイルの選択へ進んでください。

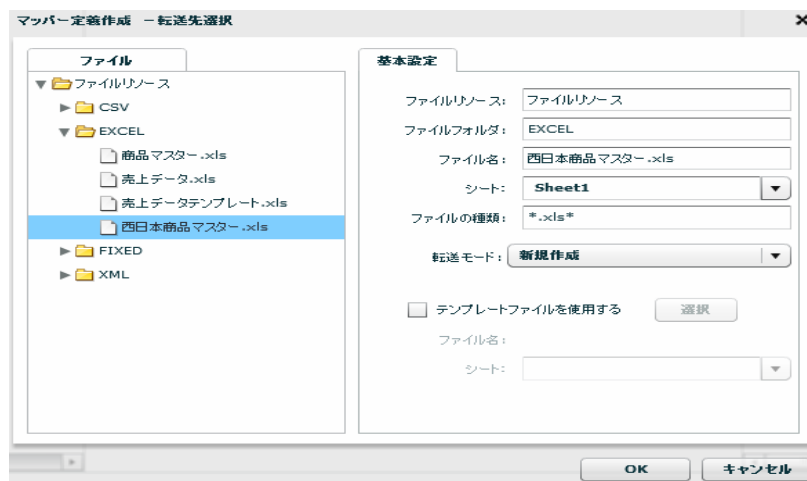
手順3: EXCEL ファイルリソースから出力先ファイルを選択します

データの出力先ファイルを選択します。

画面右側の[FILE]アイコンを右クリックし[EXCEL ファイル選択]をクリックします。



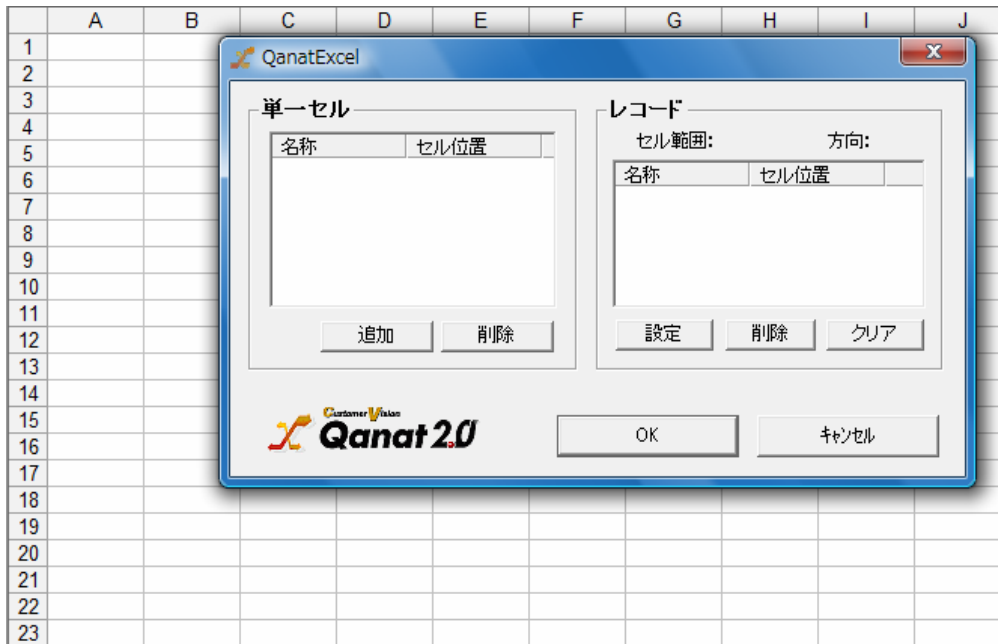
[転送先選択]画面で 出力するファイルを選択します。今回は既存のファイル「西日本商品マスター.xls」に書き込みを行うこととして記述します。「西日本商品マスター.xls」は新規作成された EXCEL ファイルです。出力ファイルを選択します。



[OK] ボタンをクリックし転送先選択画面を閉じます。

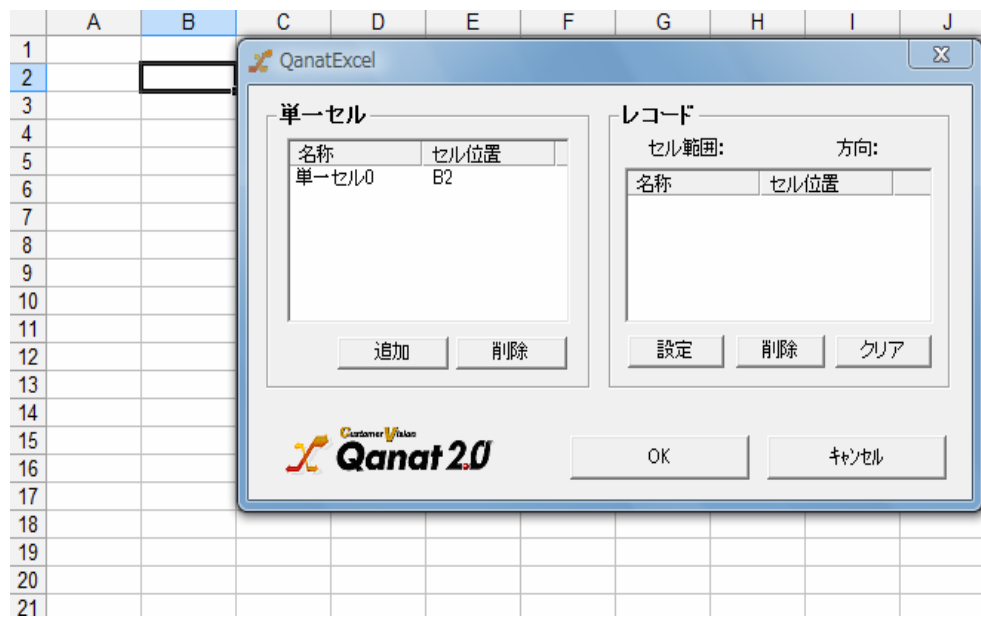
書き込みセルを指定します。書き込みでは、EXCEL ファイルから読み取る方法で設定します。

転送先の「西日本商品マスター.xls」を選択し、右クリックメニューの「セル範囲の取得」をクリックします。ファイル選択画面が表示されるので、「西日本商品マスター.xls」を選択して、[OK]ボタンをクリックしてください。「ダウンロードが完了しました」のメッセージが表示されるので[OK]ボタンをクリックします。EXCEL と QanatExcel が実行されます。



はじめに、単一セルの指定を行います。EXCEL ファイル上のセル「B2」を選択し、QanatExcel の[追加] ボタンをクリックします。「単一セル」に以下の情報が表示されます。

名称	セル位置
単一セル0	B2



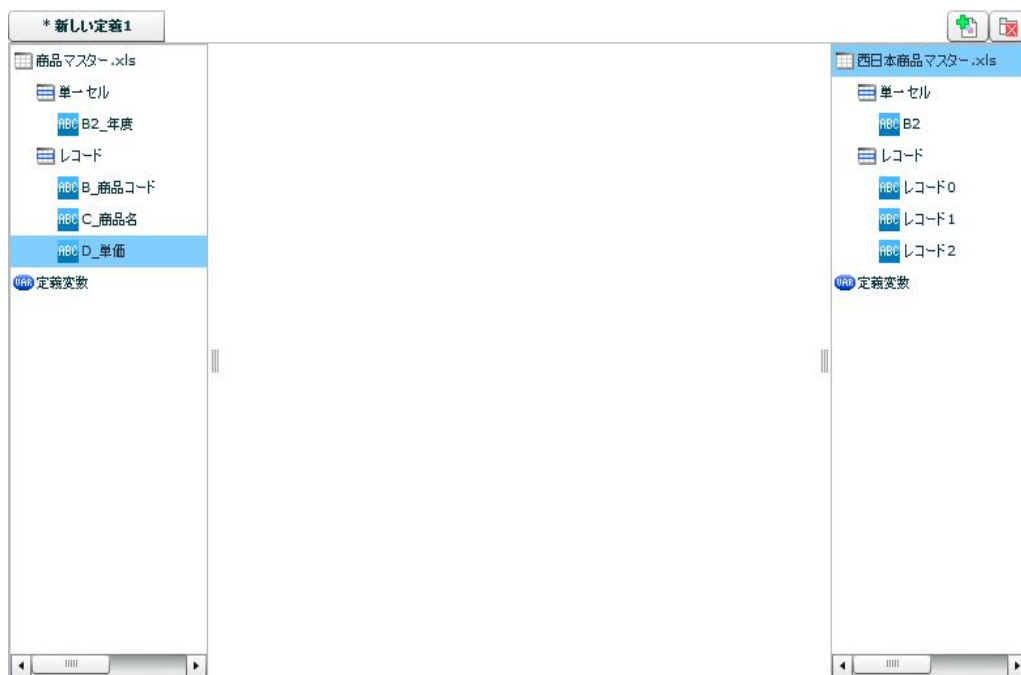
名称はフィールド名として扱われます。分かりやすさのため、名称を変更します。「単一セル0」をクリックすると、入力できる状態になるので、「B2」と名称を変更します。



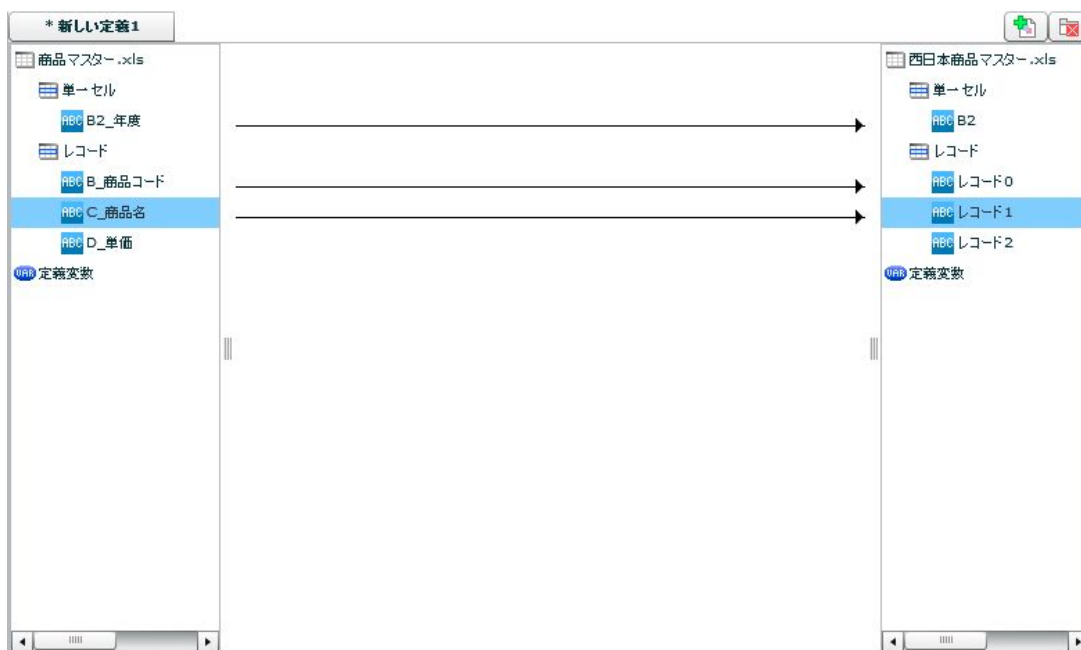
以上の操作が終了したら、[OK]ボタンをクリックします。出力先のフィールドが作成されたことを確認した後、データマッピング作業を行います。

手順4： データマッピングをします

入力元と出力先のそれぞれのデータフィールドをつなぎ、データ転送設定をします。



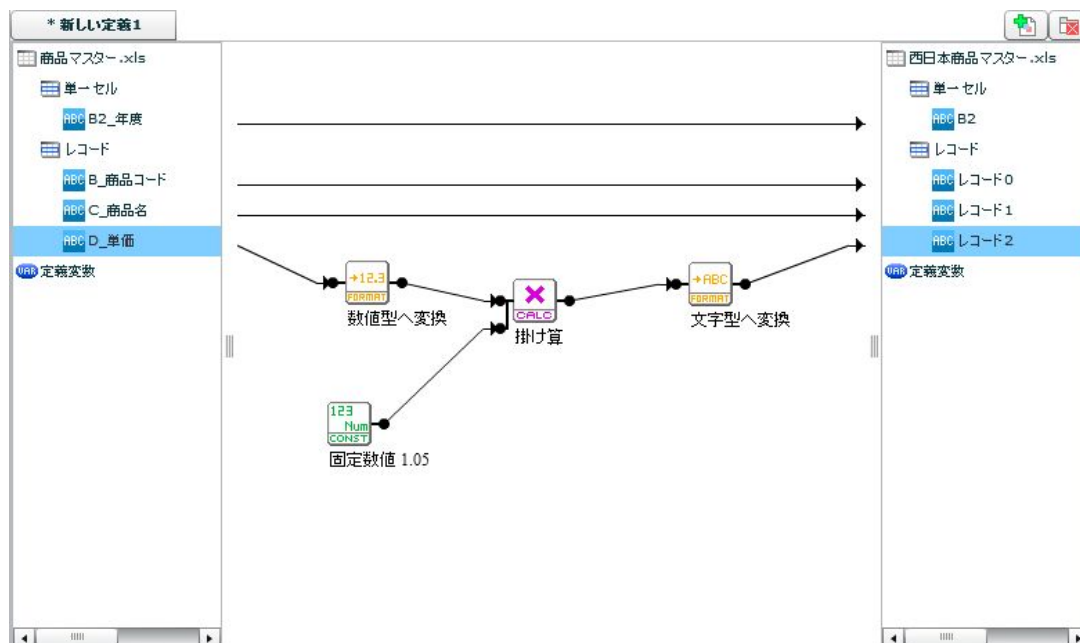
そのためには画面左側の転送したいフィールドを、右側の同名のフィールドに対してマウスでドラッグ&ドロップしマッピングを一つひとつ作成します。



単価フィールドは「西日本商品マスター.xls」のほうを 5% 高く設定したいため、売上情報の単価にその分上乘せした値を、売上情報 2 の単価フィールドにマッピングします。

このマッピングには“5%”という値を持たせた固定数値フィルタと、掛け算フィルタを利用し表現します。

(詳しくは 1-1 データ転送定義を作成する(データベース)参照)



以上で定義の作成は終了となります。定義を保存し、実行してください。実行結果は以下のようになります。

	A	B	C	D
1				
2		2010年度		
3		101	ブレンド	472.5
4		102	アメリカ	504.0
5		103	アイスコ	504.0
6		201	モカ	525.0
7		202	ガテマラ	525.0
8		203	コロンビ	525.0
9		204	ブラジル	525.0
10		205	マンデリ	525.0
11		206	キリマン	577.5
12		207	ブルーマ	661.5
13		301	カフェ・	577.5
14		302	アイスカ	609.0
15		303	ウィンナ	609.0
16		304	ロシアコ	630.0
17		305	コーヒー	609.0
18		401	レモンテ	472.5
19		402	ミルクテ	472.5
20		403	アイステ	504.0
21		404	ブランデ	630.0
22		405	ウーロン	504.0
23		501	ココア	556.5
24		502	アイスコ	609.0
25		503	ミルク	472.5
26		504	アイスミ	504.0
27		505	コーラ	504.0
28		506	ソーダ	504.0
29		507	レモンス	556.5
30		508	オレンジ	577.5
31		509	トマトジ	577.5
32		510	ヨーグル	577.5
33		511	クリーム	609.0
34		512	オレンジ	609.0
35		513	コーヒー	609.0
36		514	コーラフ	609.0
37		515	バニラア	556.5
38		516	ブルーベ	609.0
39		517	ラズベリ	609.0
40		601	トースト	262.5
41		602	タマゴト	399.0
42		603	ハムトー	399.0

1-3: データ転送定義を作成する(SALESFORCE)

ここでは salesforce に対してデータの書き込みを行うデータ転送定義の作成を習得します。

目的	作成されるもの
<ul style="list-style-type: none">salesforce に対して書きこみをする	<ul style="list-style-type: none">データ転送定義

事前修了しておきたい実習項目
ありません。

実習のシナリオ

Salesforce アカウントを所持し、アカウント定義作成ができていますものとして。

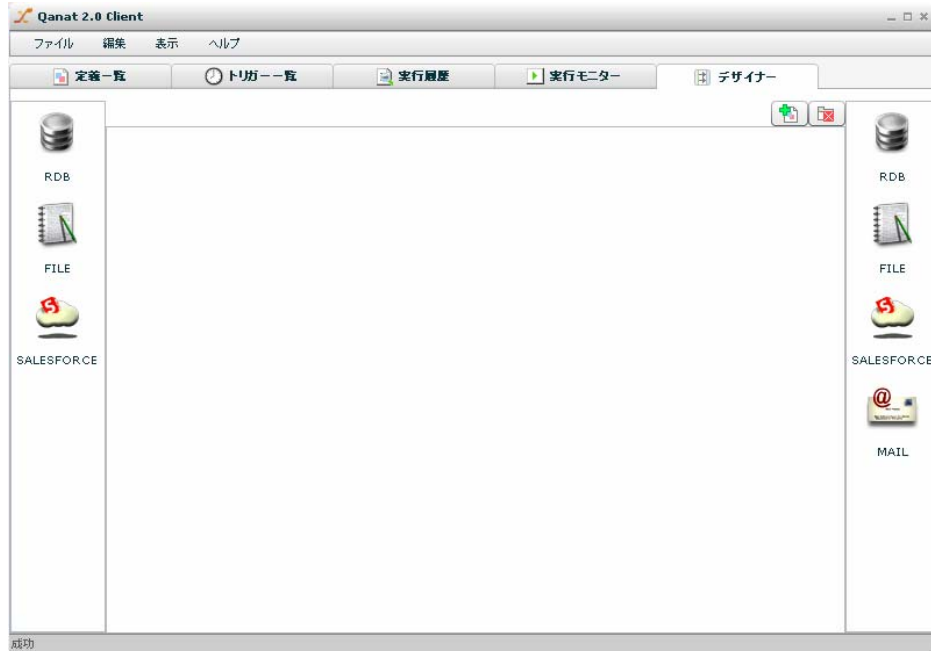
“ XML ” フォルダの売上.xml からデータを読み込み、

Salesforce リソースの “ 取引先 ” テーブルヘデータを更新します。

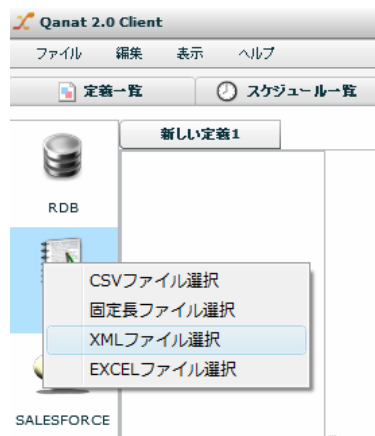
作成したデータ転送定義を保存しテスト実行を利用して検証した後、手動で実行します。

手順 1： 本製品を起動し [デザイナー] タブを選択します。

入力元（データ読み込み側）には XML ファイル、出力先（データ書き込み側）には SALESFORCE を利用します。

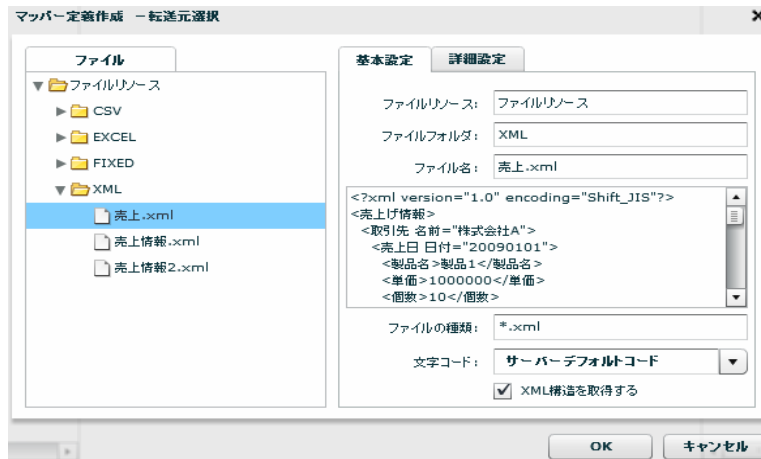
**手順 2： FILE から XML ファイルを選択します**

まず、入力元ではデータの読み込み先である XML ファイルを選択します。

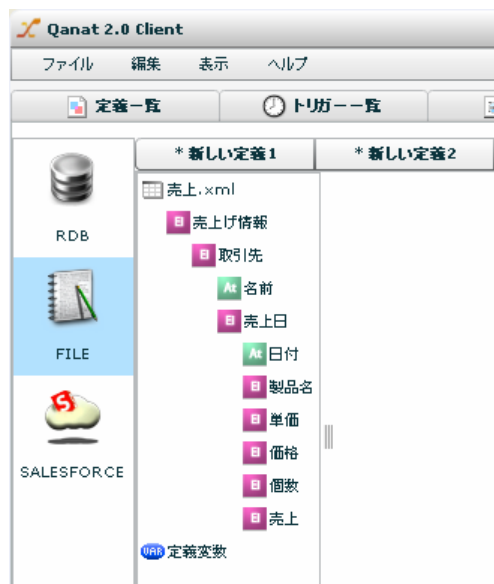


[FILE]と書かれたアイコンを右クリックし[XML ファイル選択]をクリックします。

売上.xml を選択し、XML 構造を取得するにチェックを入れます。



ファイル選択



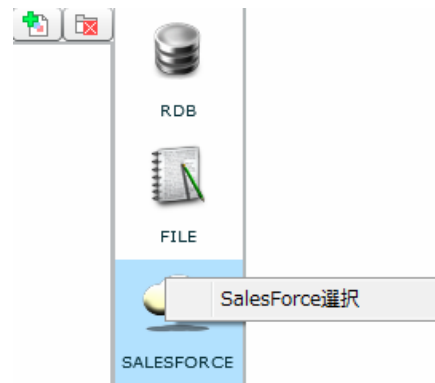
入力 - 売上 xml

手順3: SALESFORCE から出力先テーブルを選択します

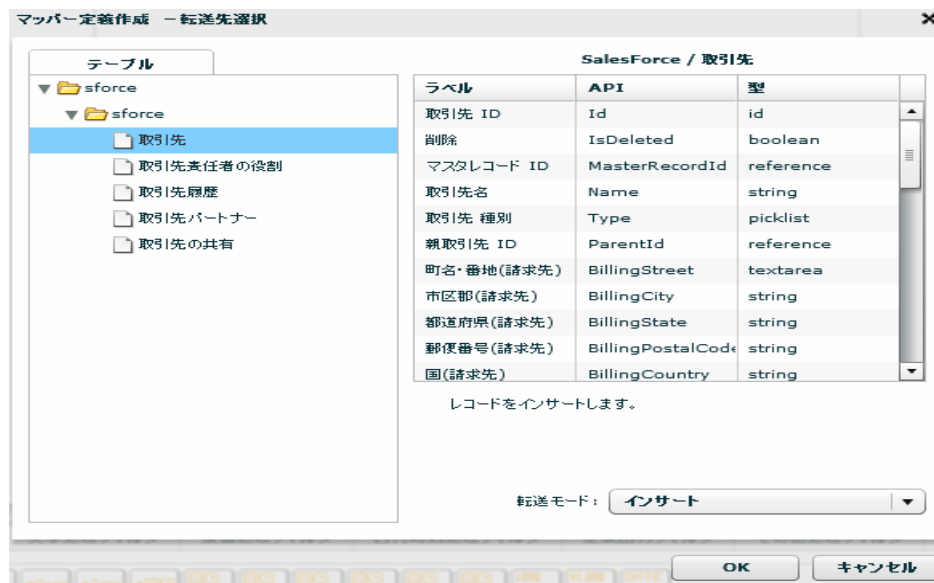
データの出力先テーブルを選択します。

手順2では売上.xmlを読み込むよう設定しました。手順3では出力先として SALESFORCE リソースの取引先テーブルに取得したデータを転送するように指定します。

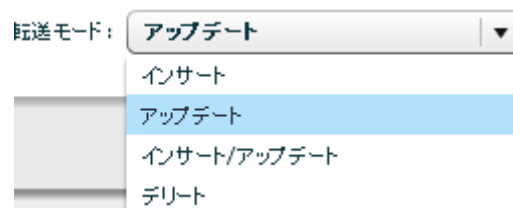
画面右側の[SALESFORCE]アイコンを右クリックし[SalesForce 選択]をクリックします。



[転送先選択]画面で“取引先”テーブルを選択し転送先を指定します。



[転送先選択]画面では、転送モードを指定します。この実習では[アップデート]を選択します。



転送モードは次から一つ選択します。

- ・インサート : データの新規挿入
- ・アップデート : 更新キー指定によるデータ更新
- ・インサート/アップデート : 更新キーが一致する既存データは更新し、それ以外は新規挿入
- ・デリート : 対象のテーブル内のデータを全て削除

[OK]ボタンをクリックし転送先選択画面を閉じます。



手順 4： データマッピングをします

入力元と出力先のそれぞれのデータフィールドをつなぎ、データ転送設定をします。

手順 3 で転送モードを[アップデート]を選択しました。インサート以外の転送モードでは更新キーを指定する必要があります。

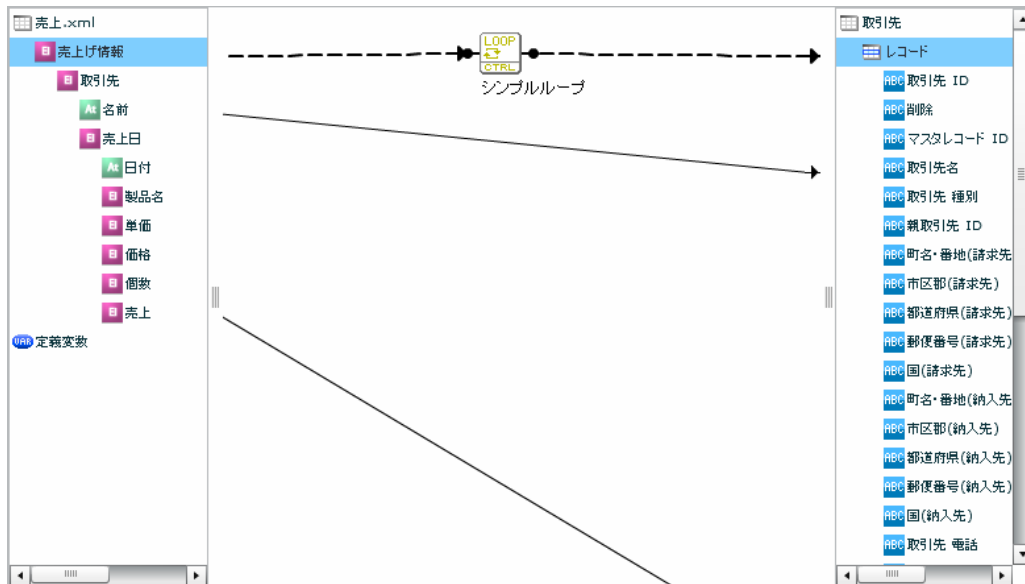
また、[インサート/アップデート]に限り、更新キーに指定できるのは salesforce の外部キーのみとなります。

出力先のフィールドのうち、更新キーとなるフィールドを右クリックし、メニューから[更新キーに指定する]をクリックします。入力元と出力先で一致する更新キーが見つかった場合は更新します。



- ❗ 出力先に主キー設定があるにも関わらず更新キーを指定しなかった場合はエラーが発生します。
更新キーは忘れずに必ず指定するようにしてください。

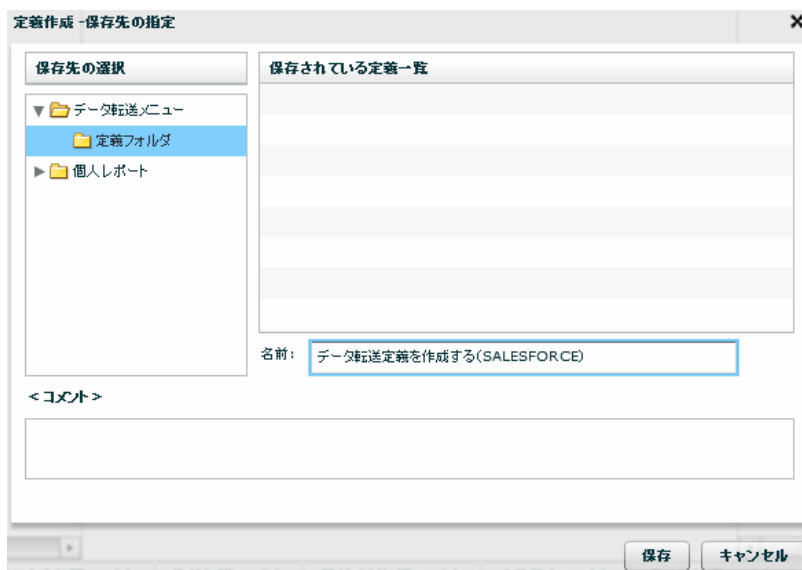
この実習は“売上.xml”から“SALESFORCEの取引先テーブル”ヘデータを転送することです。そのためには画面左側のフィールドを、右側の出力させたいフィールドに対してマウスでドラッグ&ドロップしマッピングを一つ一つ作成します。



入力の名前フィールドを出力の取引先名フィールドに、入力の売上フィールドを出力の年間売上フィールドへ結線し売上情報要素と出力レコードをシンプルループフィルタで結線します。

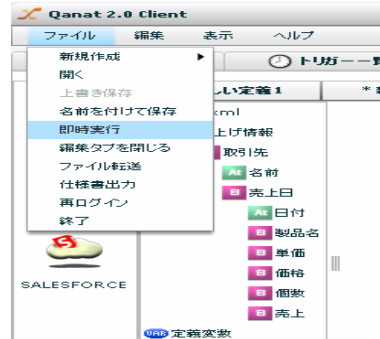
手順5： データ転送定義を保存します

メニューバーの[ファイル] [名前を付けて保存]をクリックします。次の画面のように保存先を選択し[名前]欄に定義名称、[コメント]欄にその定義の説明を入力し[保存]をクリックします。

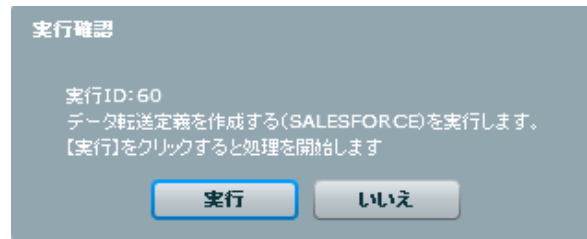


手順6： データ転送定義を実行します

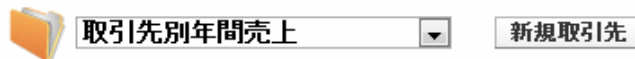
先ほど作成/保存した定義を画面のように開いた状態でメニューバーの[ファイル] [即時実行]をクリックします。



実行確認画面では[実行]をクリックし実行を開始します。



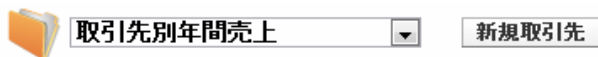
SALESFORCE の取引先テーブルを参照し、結果を確認します。



新規ビューの作成 | 編集 | 削除 | **更新** | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y

<input type="checkbox"/> アクション	取引先名 ↑	都道府県(請求先)	電話	種別	年間売上
<input type="checkbox"/> 編集 削除	株式会社A	NY	(212) 555-5555	Prospect	10,000,000
<input type="checkbox"/> 編集 削除	株式会社B	Ontario	(905) 555-1212	Prospect	10,000,000

Salesforce 取引先別年間売上ビュー（定義実行前）



新規ビューの作成 | 編集 | 削除 | **更新** | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | ア |

<input type="checkbox"/> アクション	取引先名 ↑	都道府県(請求先)	電話	種別	年間売上
<input type="checkbox"/> 編集 削除	株式会社A	NY	(212) 555-5555	Prospect	30,000,000
<input type="checkbox"/> 編集 削除	株式会社B	Ontario	(905) 555-1212	Prospect	100,000,000

Salesforce 取引先別年間売上ビュー（定義実行後）

売上.xml のデータにより、salesforce の取引先テーブルが更新されているはずです。

1-4: スクリプト定義を作成する

マッパー定義ではできない or とても複雑なものもスクリプト定義で作成すればシンプルにできるという場合があります。

ここではそういった一例を記述します。

目的	作成されるもの
<ul style="list-style-type: none">・スクリプトによるデータ転送定義を作成する・スクリプトの書き方を理解する	<ul style="list-style-type: none">・スクリプト定義

事前修了しておきたい実習項目

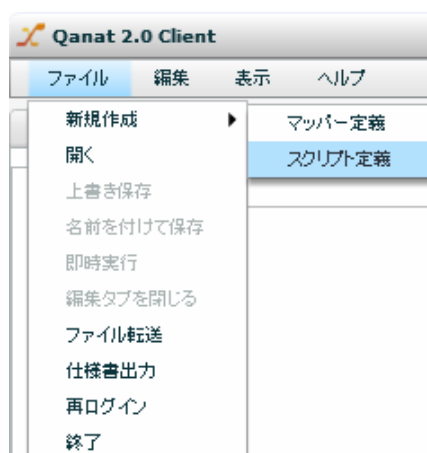
ありません。

実習のシナリオ

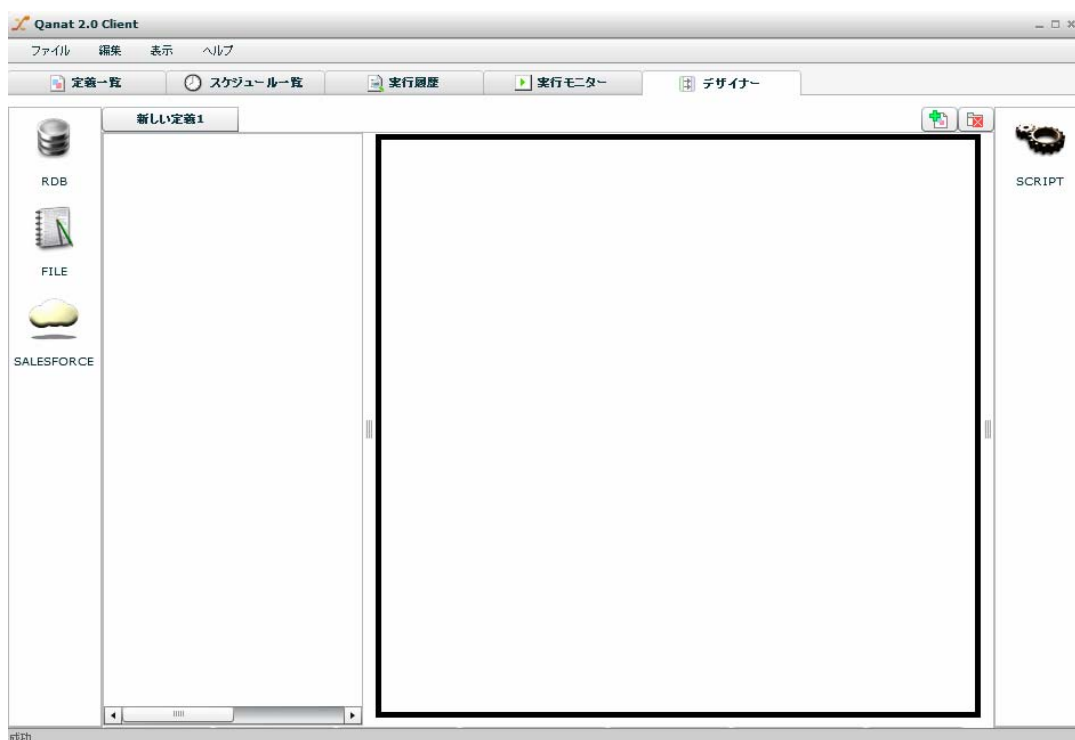
“ 売り上げ実績.csv ” からデータを読み込み、年度ごとの売り上げ、粗利益の合計を算出し CSV ファイルに出力する。

作成したスクリプト定義を保存し手動で実行します。

手順1: [ファイル]メニューの新規作成からスクリプト定義を選択します。



スクリプト定義作成



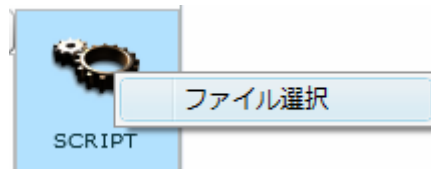
スクリプト定義新規作成画面

手順2： 入力ソースに CSV として[売上げ実績.csv]ファイルを選択する。

検索条件で年度を基準に並び替え設定をし、全フィールドを選択する。



入力ソース選択

手順3： 出力ファイルに書き込むファイルを選択する。

出力ソース選択

手順4： 自動でデフォルト記述が書き込まれる。

売上げ実績.csv

レコード

- ABC 年度
- ABC 四半期
- ABC 担当者コード
- ABC 担当者名
- ABC 地区コード
- ABC 担当地区
- ABC 商品コード
- ABC 商品名
- ABC 数量
- ABC 売上げ金額
- ABC 粗利益
- ABC 定価
- ABC 仕入れ単価

VAR 定義変数

```

年度, 四半期,担当者コード,担当者名,地区コード,担当地区,商品コード,
商品名,数量,売上げ金額,粗利益,定価,仕入れ単価
#foreach ($record in $in)
$record.get('年度'),$record.get(' 四半期'),$record.get('担当
者コード'),$record.get('担当者名'),$record.get('地区コード
'),$record.get('担当地区'),$record.get('商品コード
'),$record.get('商品名'),$record.get('数量'),$record.get('売り
上げ金額'),$record.get('粗利益'),$record.get('定価
'),$record.get('仕入れ単価')
#end

```

デフォルト記述

手順5： 処理の記述をする。

売り上げ、金額、粗利益を1レコードずつ足しこんでいき、年度の値が変わったところで値を初期化し、また足しこんでいくという処理を記述します。

Verocityの規則に従ってこの処理を記述していくと以下のようになります。

```

年度,売り上げ金額,粗利益
#set($cnt = 0)
#set($year = 0)
#set($sell = 0)
#set($profit = 0)
#foreach ($record in $in)
#if($record.get('年度')!=$year && $cnt > 0)
$year,$sell,$profit
#set($sell = 0)
#set($profit = 0)
#end
#set($cnt = $cnt + 1)
#set($year = $record.get('年度'))
#set($s = 0)
#set($sell = $sell + $s.parseInt($record.get('売り上げ金額')))
#set($p = 0)
#set($profit = $profit + $p.parseInt($record.get('粗利益')))
#end
#if($cnt > 0)
$year,$sell,$profit
#end

```

処理の記述

【レコード数】、【年度】、【売り上げ】、【粗利益】の4つの変数をつくり0で初期化する。

変数は英字のみ、売り上げと粗利益には年度ごとの合計が入る。

年度が変わったタイミングで、変わる直前の年度の売り上げと粗利益の合計を出力し、その後変数を初期化する。

そのレコードの【売り上げ】、【粗利益】を加える

、 の繰り返し

最終レコードを出力する。



定義保存、テスト実行、定義実行はチュートリアル『1-1：データ転送定義を作成する（データベース）』を参照してください。

このスクリプト定義を保存し、実行すると年度と年度ごとの売り上げ、粗利益の合計が CSV ファイルに出力されます。

```

年度,売り上げ金額,粗利益
2002,30000,4500
2003,25000,7500
2004,30000,10000
2008,20000,5000

```

出力結果

このように、マッパー定義ではつけれない or 複雑なルーチンになってしまうような処理でもスクリプト定義で比較的簡単に作成することが可能になります。

1-5:QanatExecute を使用する

ここでは QanatExecute を利用して外部アプリケーション（今回は HULFT）と連携する例をあげます。

例：本製品で、データベース CSV ファイルというデータ転送処理を行いその出力データを HULFT で加工する。

手順 1： HULFT サーバーに QanatExecute をインストールする。

手順 2： インストールした QanatExecute フォルダ内のバッチファイルを編集する。

```

QanatExecute02.bat - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
@rem FOLDER フォルダ名
@rem DEF_NAME 定義名
@rem DEF_TYPE qanmapper:マッパー定義 qantask:タスク定義 qanscript:スクリプト定義
@rem VARIABLE 定義変数 (変数名=値,変数名=値,...)
@rem =====
SET INST_PATH=C:\Program Files\QanatExecute

SET HOST=10.40.70.196
SET PORT=6100
SET USER=cvadmin
SET PASSWORD=cvadmin
SET MENU=テスト1
SET FOLDER=フォルダー1
SET DEF_NAME=hulft
SET DEF_TYPE=qanmapper
SET VARIABLE=

"%INST_PATH%\jre\bin\java" -jar "%INST_PATH%\QanatExecute.jar" %HOST% %PORT% %USER% %PASSWORD% %MENU% %FOLDER% %DEF_NAME% %DEF_TYPE% %VARIABLE%

echo %errorlevel%

@rem 以下は、1/-1の時はバッチの実行結果がエラーになる

if %errorlevel%==1 goto :ERREND
if %errorlevel%=-1 goto :ERREND

:TRUEEND

PAUSE

```

HULFT と本製品連携時の QanatExecute バッチファイルの例

SET HOST :	ホスト名 or IP
SET PORT :	ポート番号
SET USER :	本製品ログイン時ユーザーID
SET PASS :	本製品ログイン時パスワード
SET MENU :	実行させる転送定義を含むメニュー名
SET FOLDER :	実行させる転送定義を含むフォルダ名
SET DEF_NAME :	実行させる転送定義名
SET DEF_TYPE :	実行させる転送定義のタイプ
SET VARIABLE :	定義変数 (変数名=値 変数名=値 . . .)

本製品では実行定義が正常終了の場合"errorlevel"=0、異常終了の場合"errorlevel" =1 or -1 になります。しかし、バッチ処理が終了するとバッチ処理が正常終了したという値として"errorlevel"=0 になります。つまり、定義が正常終了したか異常終了したかに関わらず、"errorlevel"=0 になってしまいます。HULFT では Execute バッチファイルが正常終了 ("errorlevel"が 0) の場合に配信処理が実行されます。(本製品の実行結果がエラーでも Execute バッチファイルとしては正常終了です)

"errorlevel"は直前の実行結果を持つため、本製品の定義の実行結果によって HULFT の配信の有無を制御したいのであれば、バッチ処理自体がエラーになるようにバッチを組んでください。例のように、"errorlevel"が 1 または-1 の場合は存在しないラベルに goto させるなどの処理を書き込むことで"errorlevel" =1 or -1 になったタイミングでバッチファイルを異常終了させ、"errorlevel"=0 になるのを回避します。

手順 3： バッチファイルを実行する。

上記の QanatExecute バッチファイルを実行すると、データベース CSV ファイルの転送処理がなされ、その処理が正常終了した場合だけ、HULFT で配信処理を行うという連携が可能になります。

1-6:メール送信定義を作成する

データベースやファイルリソースからデータを取得し、メール送信を行うデータ転送定義の作成を習得します。

目的	作成されるもの
・読み取ったデータを利用しメール送信を行う	・ありません

事前修了しておきたい実習項目

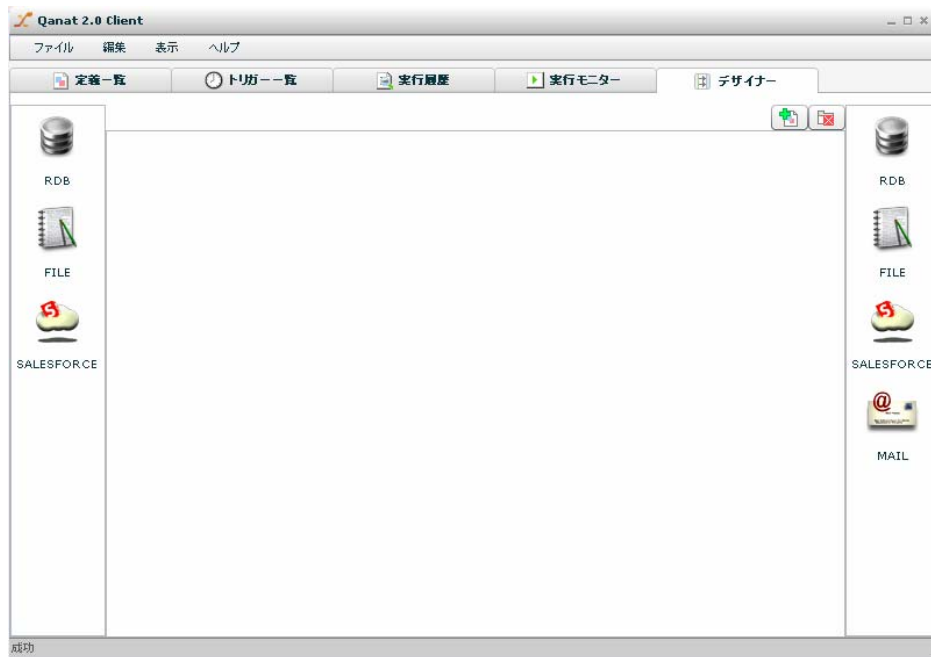
ありません。

ただし、使用するメールサーバーを事前に準備し、管理ツールを使用して、データリソースの「メールサーバー」を作成しておく必要があります。

実習のシナリオ

“社員マスター.csv”ファイルからデータを読み込み、各社員にメール送信を行います。

作成したデータ転送定義を保存し手動で実行します。

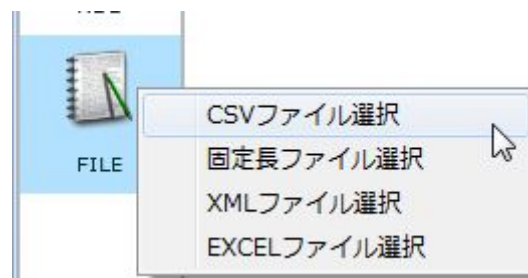
手順1： 本製品を起動し [デザイナー] タブを選択します。

入力元（データ読み込み側）にはファイルを利用します。

まず、入力元ではデータの読み込み先であるファイルの一つを選択します。

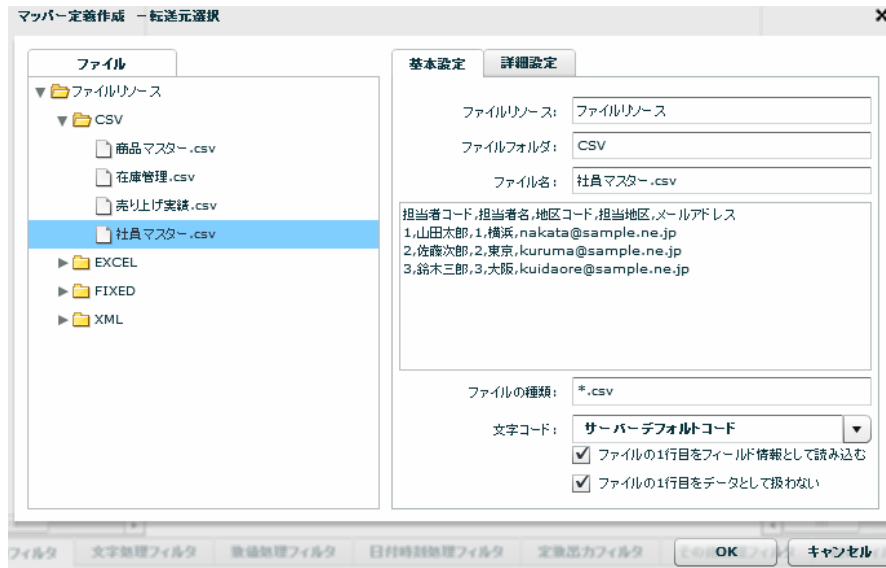
手順2： ファイルリソースから入力元ファイルを選択します

[FILE]と書かれたアイコンを右クリックし[CSV ファイル選択]をクリックします。



表示されているフォルダから目的のファイルを選択します。右側にはプレビューが表示され、データの区切り文字や囲み文字を確認します。

「社員マスター.csv」にはヘッダ行が含まれているので [ファイルの一行目を見出し行として扱う] のチェックを入れます。



[OK] ボタンをクリックし転送元選択画面を閉じます。転送元フィールドは以下のように表示されます。



転送元の設定は終了です。引き続き、転送先の設定を行います。

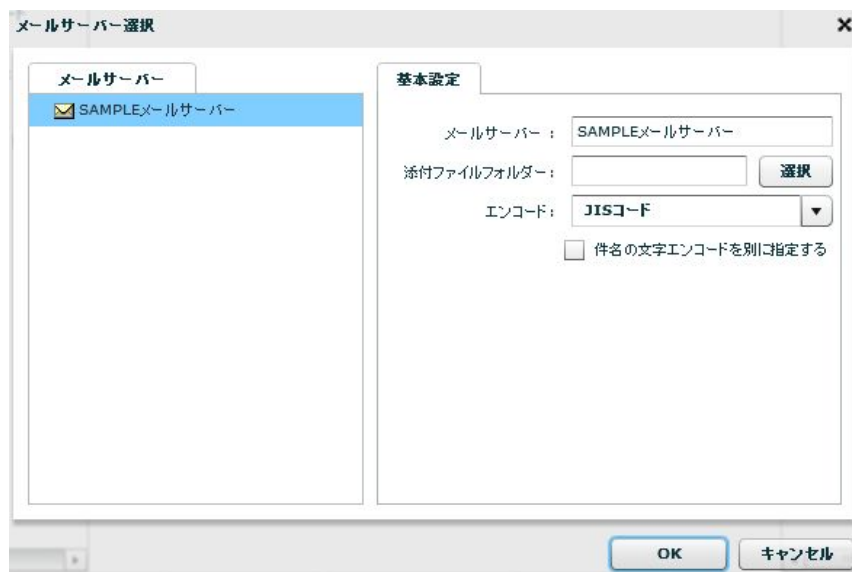
手順3： 転送先からメール送信を選択します

メール送信を選択します。

画面右側の[MAIL]アイコンを右クリックし[メールサーバー選択]をクリックします。



[転送先選択]画面でメール送信を行うメールサーバーを指定します。今回は事前に作成した“SAMPLEメールサーバー”を選択します。



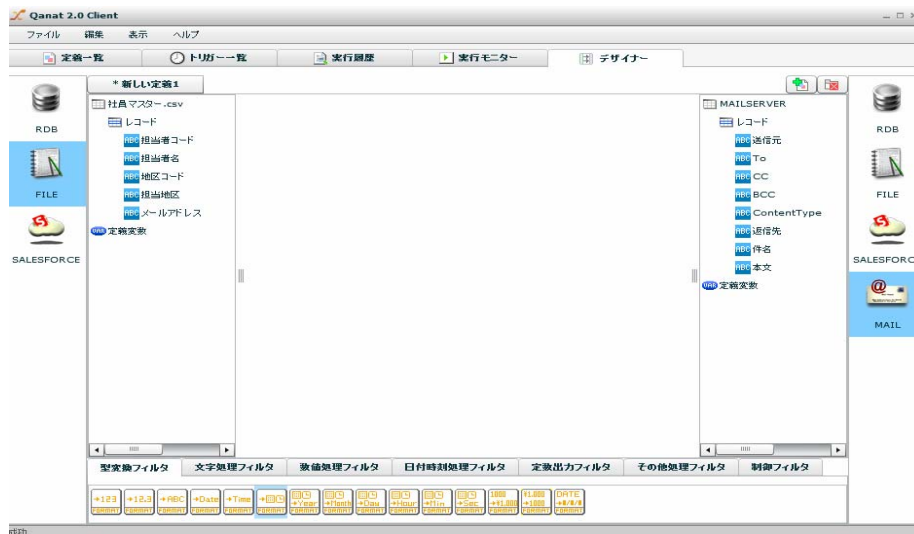
[OK] ボタンをクリックし転送先選択画面を閉じます。

以下のフィールドが転送先に表示されます。引き続き、データマッピングを行います。

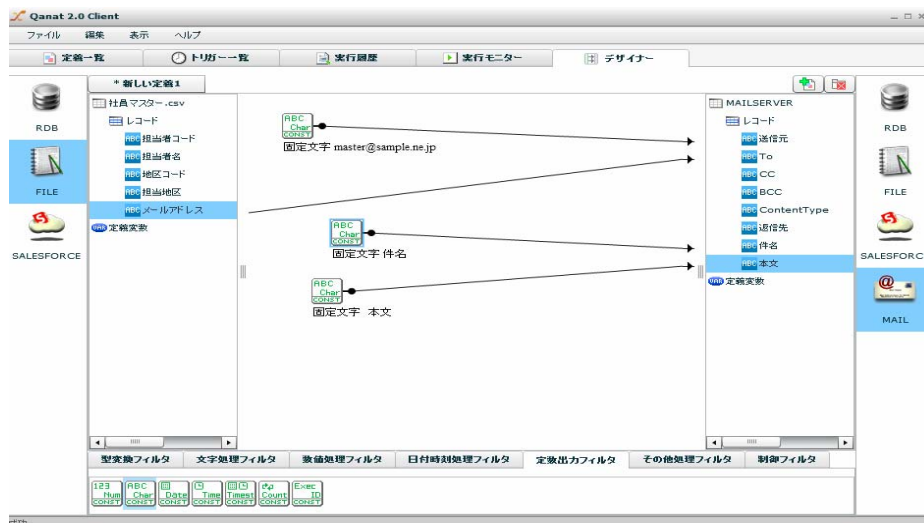


手順4： データマッピングをします

入力元と出力先のそれぞれのデータフィールドをつなぎ、データ転送設定をします。



メール送信に必要な設定値は「送信元」フィールドと「To」フィールドになります。「送信元」フィールドは[固定文字]フィルタから結線します。「To」フィールドは読み込みを行った「メールアドレス」フィールドと結線します。メールの確認のため、「件名」フィールドと「本文」フィールドにもそれぞれ[固定文字]フィルタから結線します。



以上でメール送信の定義作成は終了です。メール送信処理ではファイルフォルダー内にあるファイルを添付することや、HTMLメールを送信することも可能です。手順については [2-7:作成したファイルを送信する](#) を参照してください。

第 2 章

第2章 - 応用編 -

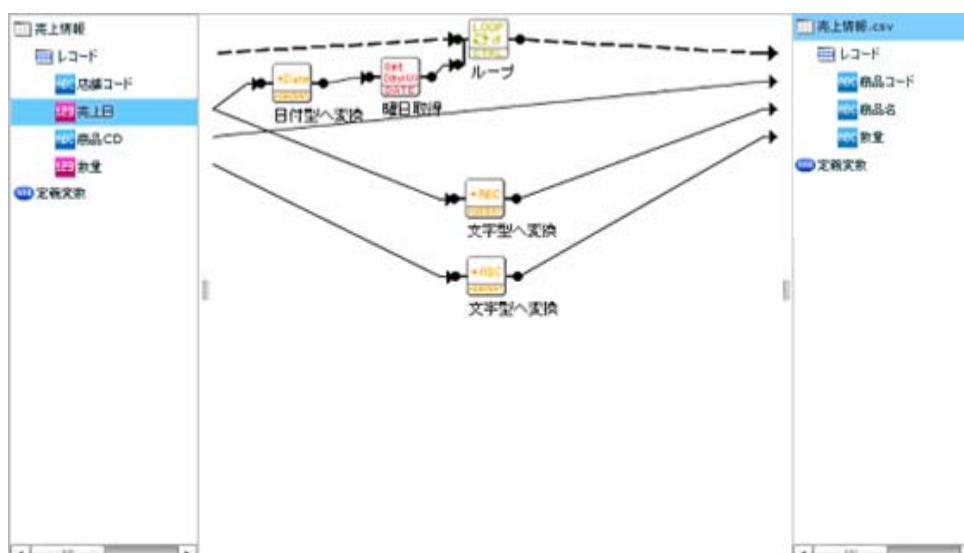
2-1:便利なフィルタ（応用）

ここから便利なフィルタ群を使った実用的な例を挙げていきます。

例1：売上データを読み込み木曜日の売上数量をCSVに書きこむ。

入力元データ = 東日本 DB 売上情報テーブル

出力先データ = CSV ファイル



ループフィルタ

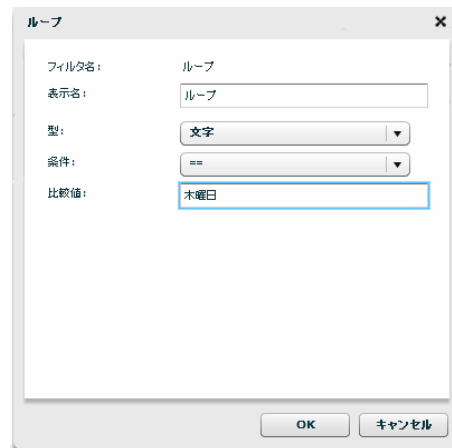
入力値が[木曜日]のレコードだけ繰り返し転送を行うよう設定します。

型 : 文字

条件 : ==

比較値 : 木曜日

に各々設定します。(下図参照)

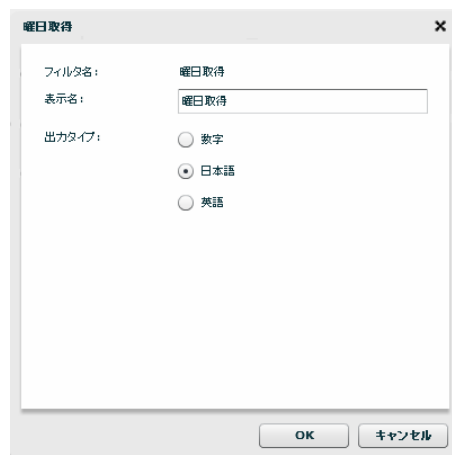


日付型へ変換フィルタ

今回の場合売上データの日付が数値型なので日付型に変換します。

曜日取得フィルタ

入力日付からその曜日を出力します。今回は分かりやすく曜日名が日本語で出力されるように設定します。



文字型へ変換フィルタ

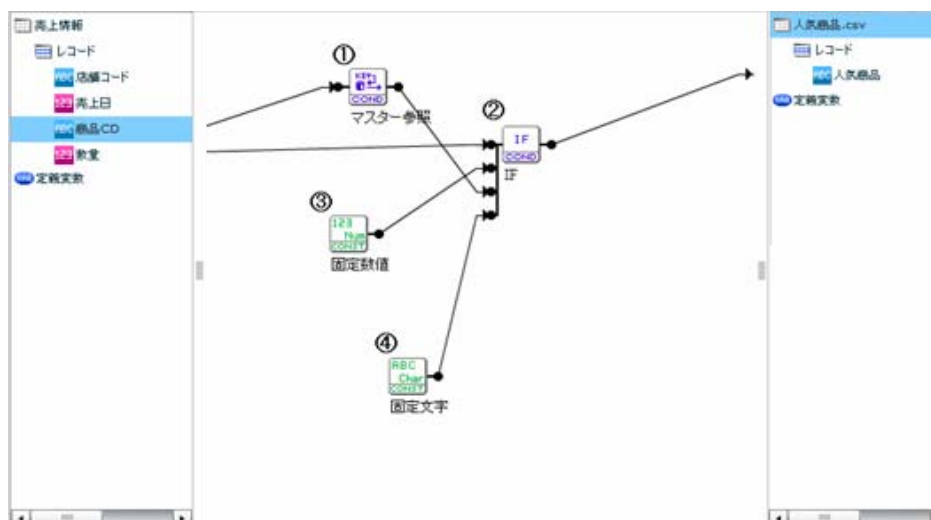
今回の場合、数量を文字型に変換して出力します。

例 2：売上データを読み込み、1 日で 150 個以上売れた商品名をマスター参照して出力。150 個売れなかった商品は「非該当」と出力。

入力元データ = 東日本 DB 売上情報テーブル

マスター参照 = 東日本 DB 商品マスターテーブル

出力先データ = CSV ファイル



マスター参照フィルタ

参照マスター情報：東日本 DB-商品マスターテーブル

キー項目：商品コード

出力フィールド：商品名

に設定する。

これにより、売上情報の 150 個以上売れた商品 C D と商品マスターの商品コードが等しければその商品名を出力する。

マスター参照

フィルタ名: マスター参照

表示名: マスター参照

参照マスター情報: 東日本DB >> DBADMIN >> 商品マスター

選択

No.	キー項目
1	商品コード

追加 削除

出力フィールド名: 商品名

OK キャンセル

IF フィルタ

条件：>

フィールド 1 (今回の場合数量)

フィールド 2 (固定数値)

真の値 (今回の場合 150 個以上売り上げた商品名)

偽の値 (該当せず)

と設定。

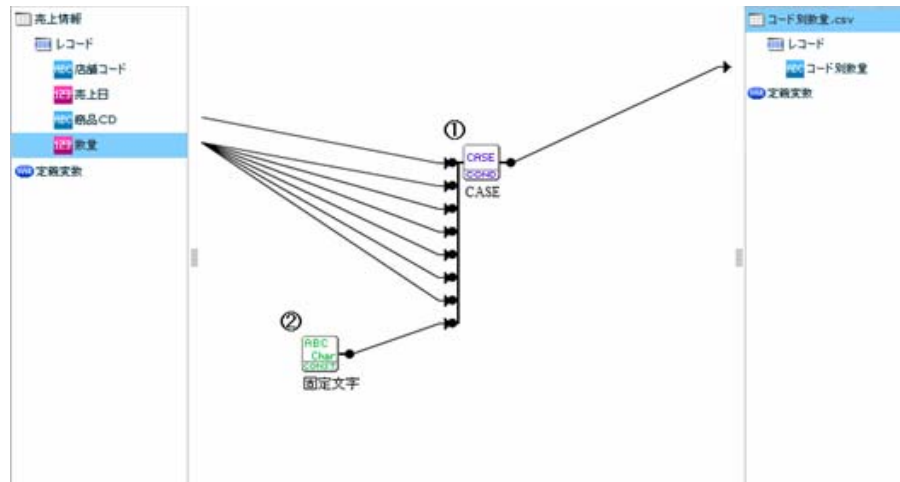
固定数値、固定文字フィルタ

それぞれ今回の場合、“150”、“該当せず”と設定

例3：売上情報テーブルの商品CDを分類し、その数量をCSVに出力。

入力元データ = 東日本 DB 売上情報テーブル

出力先データ = CSV ファイル



CASE フィルタ

CASE 値に分類条件を書く。今回の場合正規表現を使って“1~”、“2~”・・・を設定。これらの分類条件に合わなければデフォルト値を使用。

CASE ✕

フィルタ名: CASE

表示名:

条件フィールド:

No.	Case値	正規表現
1	1.*	<input checked="" type="checkbox"/>
2	2.*	<input checked="" type="checkbox"/>
3	3.*	<input checked="" type="checkbox"/>
4	4.*	<input checked="" type="checkbox"/>
5	5.*	<input checked="" type="checkbox"/>
6	6.*	<input checked="" type="checkbox"/>

デフォルト値を使用する

固定文字フィルタ

デフォルト値として“該当せず”を設定。

2-2:オリジナルテーブルを作成する

テーブル作成機能を利用することで、データ転送先を作成することができます。

この機能を利用することでデータベースに詳しくないユーザーも簡単にテーブル作成でき、データ転送先に利用することができます。

目的	作成されるもの
・ データベースを指定し新たにテーブル作成をする	・ データ転送定義 ・ データベーステーブル

事前修了しておきたい実習項目

1-1: データ転送定義を作成する(データベース)

実習のシナリオ

『1-1: データ転送定義を作成する(データベース)』で作成したデータ転送定義を編集します。

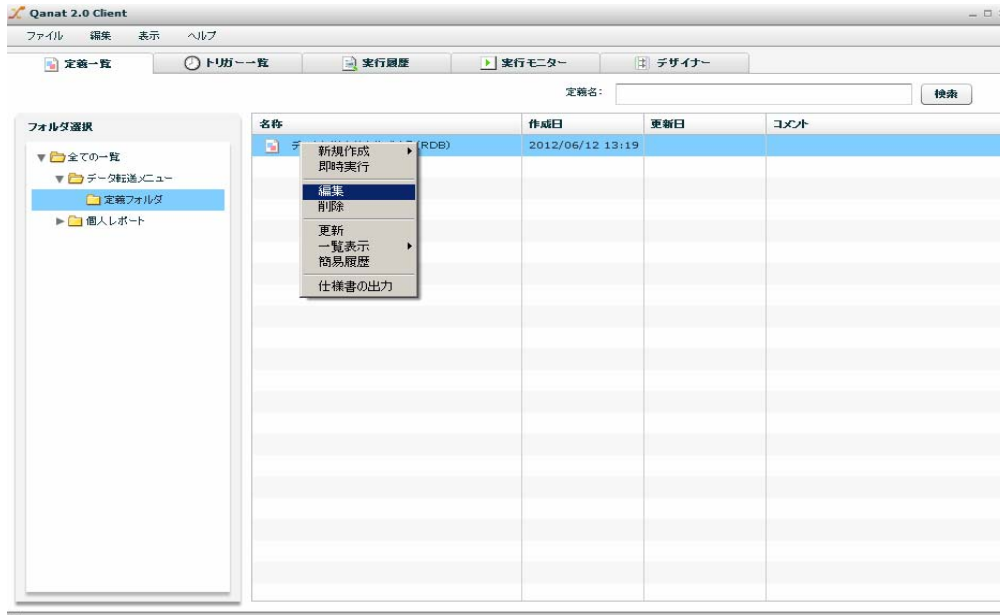
出力先リソースに指定されている西日本 DB の“商品マスター”テーブルを外し、新たに“在庫情報”テーブルを作成します。

在庫情報テーブルの主キーは商品コードとし、在庫フィールドには既定値として在庫数 100 を指定します。

また、この実習では実行時にトランザクションと障害発生時についての転送オプションを指定します。

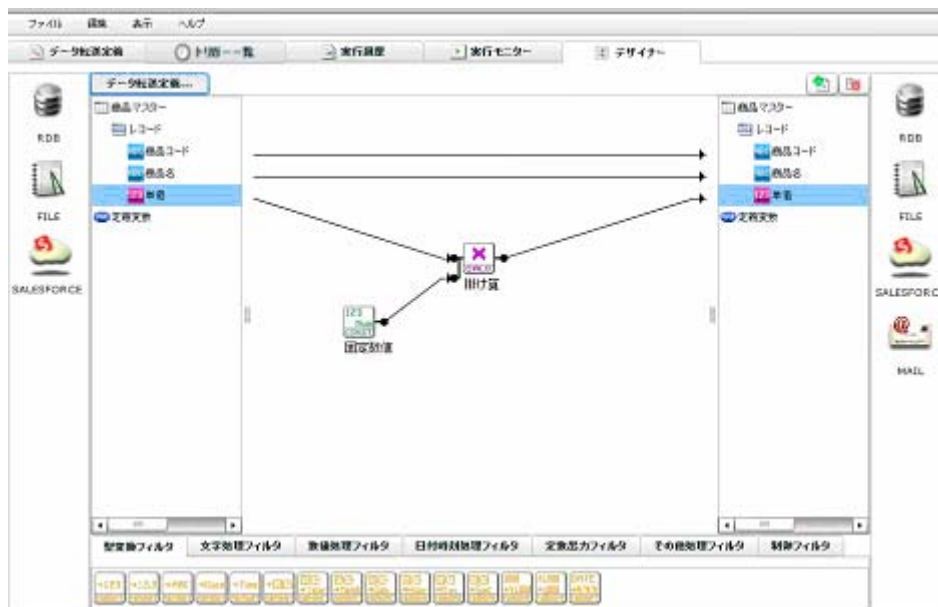
手順1： データ転送定義を選択し編集します

[データ転送定義]タブをクリックし『1-1：データ転送定義を作成する（データベース）』で作成したデータ転送定義を右クリックし[編集]をクリックします。



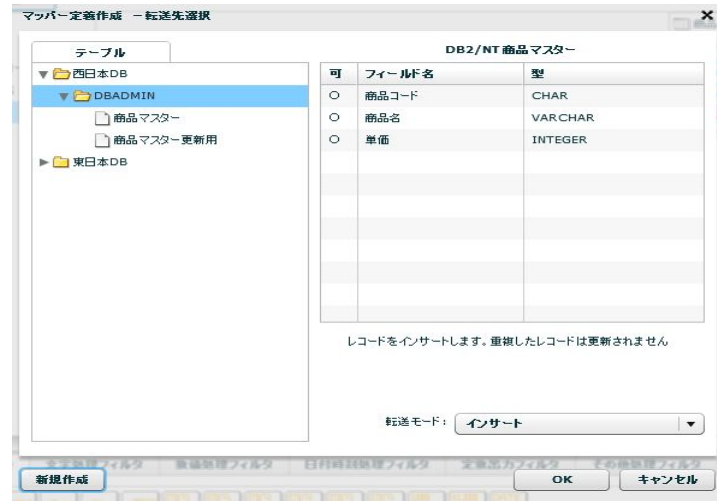
手順2： 作成するテーブル情報を入力します

データ転送定義が復元されます。出力先リソースの[RDB]を右クリックし[1テーブル選択]をクリックします。

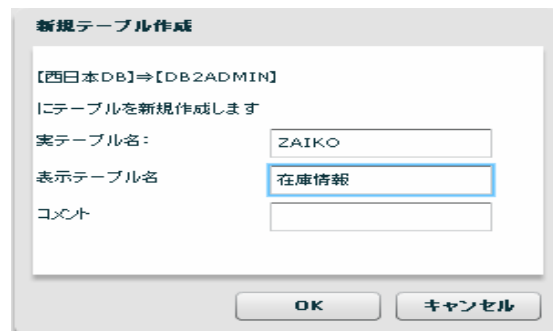


転送先選択画面が表示されます。

商品マスターテーブルのスキーマである[DB2ADMIN]と書かれたフォルダアイコンをクリックし、画面左下の有効になった[新規作成]ボタンをクリックします。

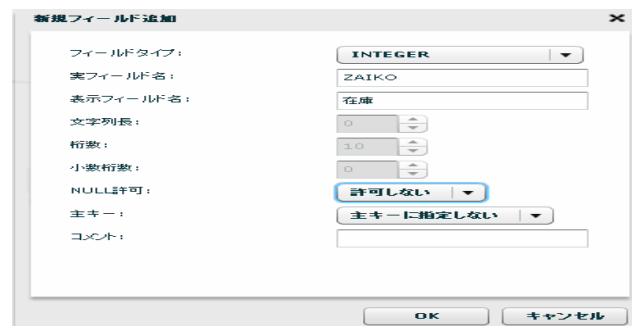


新規テーブル作成画面では、実テーブル名と画面上に表示される表示テーブル名を入力し[OK]ボタンをクリックします。



デザイナー画面に戻り、出力先テーブル情報がクリアされた代わりに[在庫情報]と表示されます。

[在庫情報]の[レコード]を右クリックし表示されたメニューから[フィールドを追加する]をクリックします。



新規フィールド追加画面では次のように実フィールド名と表示フィールド名を入力します。

新規フィールド追加

フィールドタイプ: BIGINT

実フィールド名: STOCK

表示フィールド名: 在庫

文字列長: 0

桁数: 19

小数桁数: 0

NULL許可: 許可する

主キー: 主キーに指定しない

コメント:

OK キャンセル

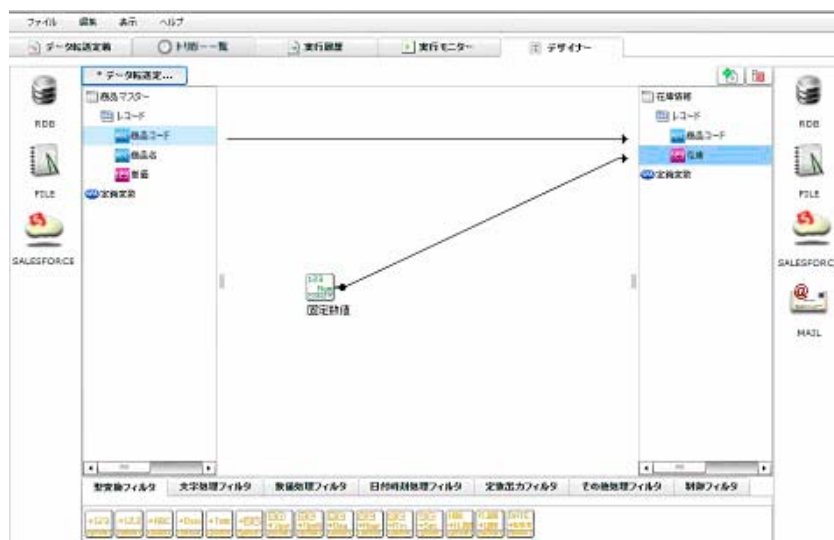
次のフィールドを出力先に追加します。

商品コード (文字型)

在庫 (数値型)

手順3: 各フィールドをマッピングします

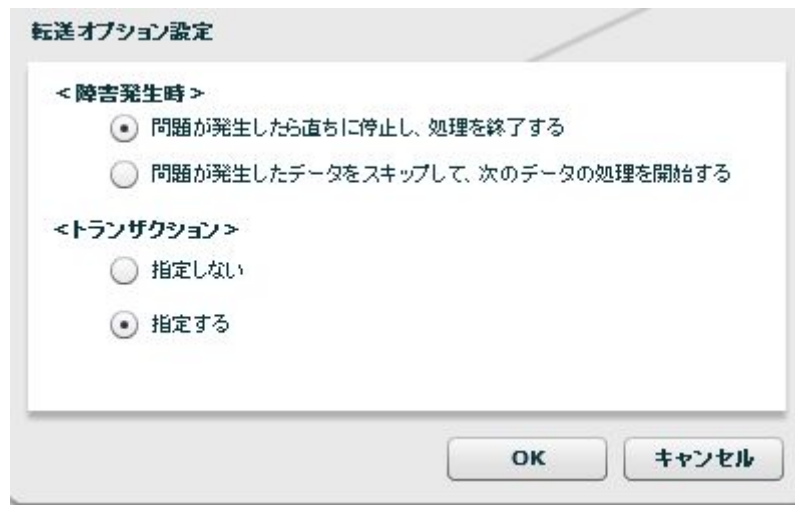
画面に追加された出力先の各フィールドに、それぞれ対応する入力元フィールドをマッピングします。キャンパス上に固定数値フィルタを配置し、そのプロパティの固定出力値欄に 100 を指定し、在庫フィールドと結線します。



データ転送定義を保存した時点ではじめて在庫情報テーブルが作成されます。テーブル作成に関わる各種制約はデータベースの仕様に従います。

手順4： 転送オプションを指定します

ツールバーの[編集] [転送オプション]をクリックします。



転送元が商品マスターとはいえ、在庫情報テーブルへのデータ転送です。処理中にエラーが発生した場合に備えデータの一貫性についても考慮すべきです。

この場合、『転送障害が発生した場合、その処理は無効とし処理前の状態に戻す』のが良いでしょう。

これを転送オプションで次のように指定します。

- ・[障害発生時]：問題が発生したら直ちに停止し、処理を終了する

更新データまたはシステムにエラーが発生した場合以降の処理を行いません。エラーを記録します。

- ・[トランザクション]：指定する

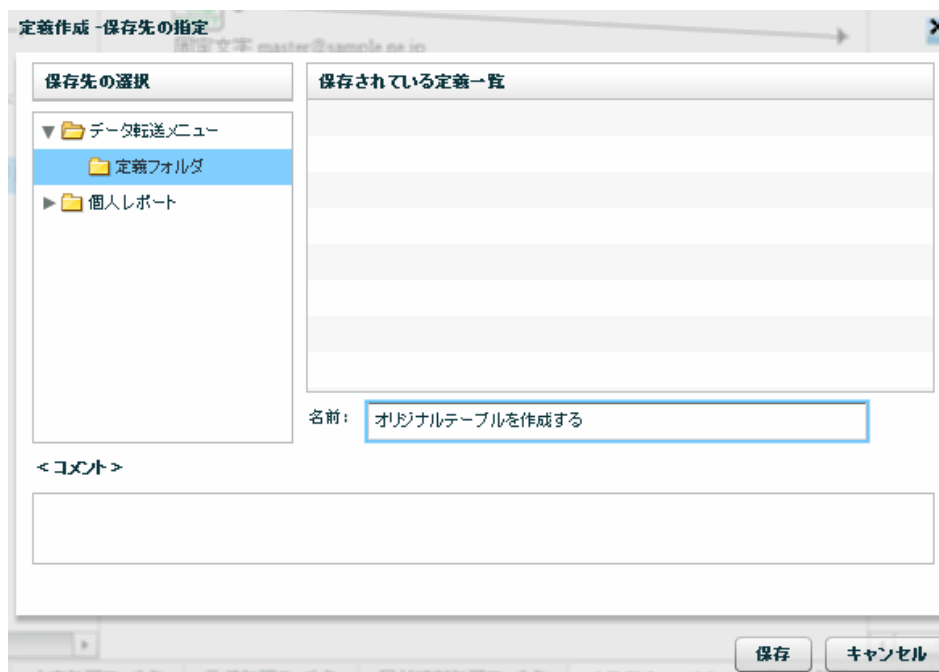
処理とともにトランザクションが開始されます。エラーが記録され処理が停止した場合、ロールバックを行い、処理前の状態に戻します。



[障害発生時]を **問題が発生したデータをスキップして、次のデータの処理を開始する** を選択し、[トランザクション]を **指定する** にした場合、エラーは全て無視されるためロールバックが起こることはありません。

手順 5: データ転送定義を保存し実行します

データ転送定義名を入力し[保存]ボタンをクリックします。



在庫情報テーブルは保存が完了した時点で対象のデータベースに作成されます。

保存後、データ転送定義『オリジナルテーブルを作成する』を実行します。新しく作成された在庫情報テーブルには入力元からの商品データと在庫数 100 がセットされていることを確認できることでしょうか。

2-3:オリジナルファイルを作成する

CSV ファイル

ファイル作成機能を利用することで、新規に CSV ファイルを作成することができます。

目的	作成されるもの
・フォルダを指定し新たに CSV ファイル作成をする	・データ転送定義 ・新規 CSV ファイル

事前修了しておきたい実習項目

1-2：データ転送定義を作成する（ファイル）

実習のシナリオ

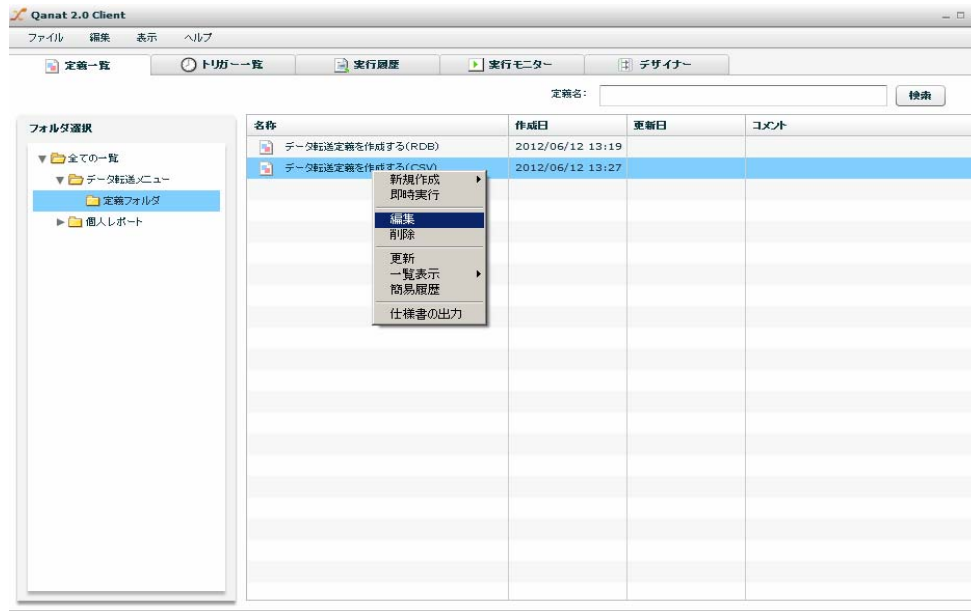
『1-2：データ転送定義を作成する（ファイル）』で作成したデータ転送定義を編集します。

出力先リソースに指定されている西日本ファイルリソースフォルダの“店舗情報”ファイルを外し、新たに“支店情報”ファイルを作成しデータを転送します。

また、この実習ではデータ転送定義の“実行前と実行後”に処理を指定するための“前後処理”を指定します。

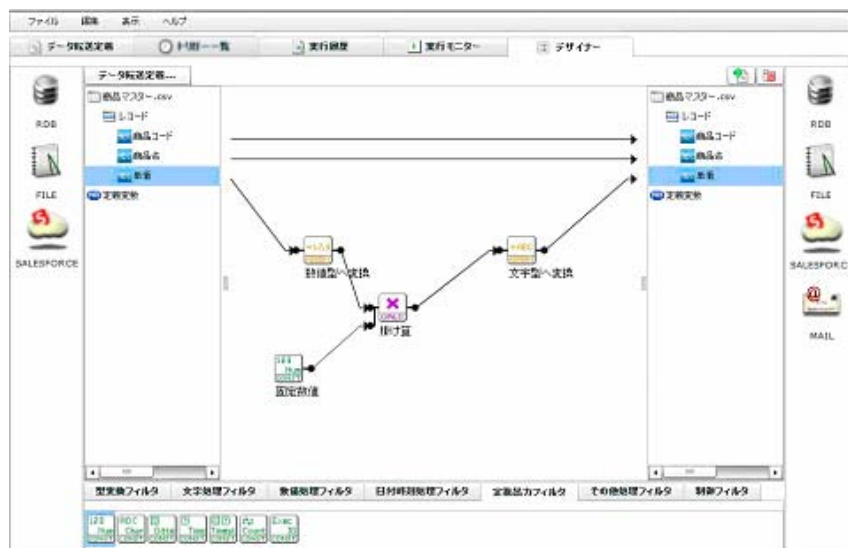
手順1: データ転送定義を選択し編集します

[データ転送定義]タブをクリックし『1-2: データ転送定義を作成する(ファイル)』の **CSVファイル** で作成したデータ転送定義を右クリックし[編集]をクリックします。



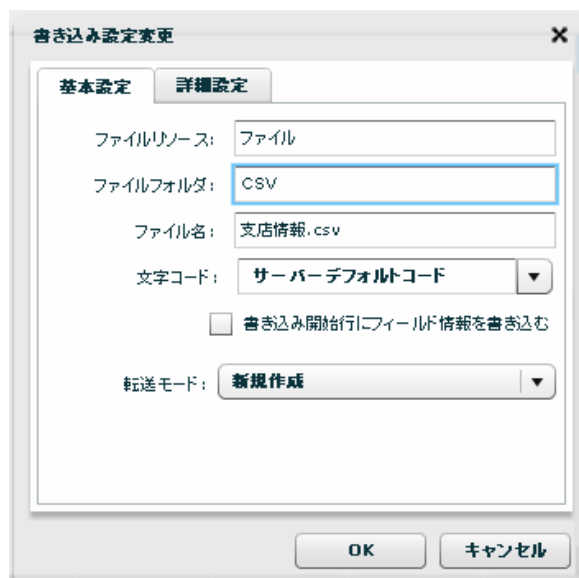
手順2: ファイル名を変更します

データ転送定義が復元されます。出力先リソースの[商品マスター.csv]を右クリックし[書き込み設定変更]をクリックします。

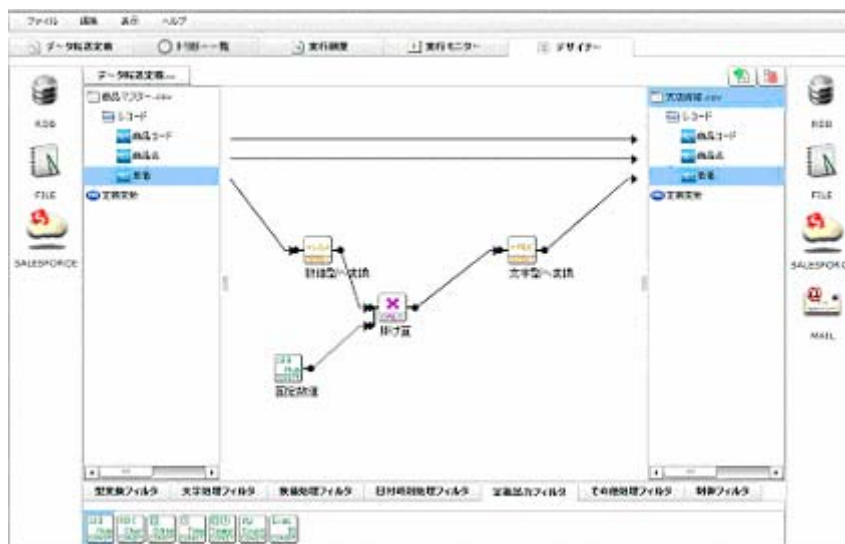


書き込み設定変更画面が表示されます。

新たに作成したいファイル名を入力し、[OK]ボタンをクリックします。



デザイナー画面に戻り、出力先ファイル名が[支店情報.csv]と表示されます。



❗ XML ファイルでも同様にオリジナルファイルの作成が可能です。

手順3： 前後処理を指定します

メニューバーの[編集] [前後処理]をクリックします。

前後処理では、データ転送定義を実行する前に実施する“前処理”と、実行した後に実施する“後処理”の二つに分かれます。

この実習では後処理を実行させます。

[後処理]タブをクリックし、画面上部の[有効にする]にチェックを付けます。

前後処理

前処理 後処理

有効にする

コマンドを実行するサーバーを選択してください

Qanat 2.0サーバー上で行う

入力元OS/400上で行う

ユーザーID

パスワード

コマンド

コマンド戻り値 障害発生時 成功時(他の値が戻った場合は障害)

※カンマ区切りで、複数指定できます

障害発生時 停止 スキップ

OK キャンセル

[コマンド]には本製品サーバー上で実行するコマンドを入力します。

例えば、次のように日付でファイル名をリネームするようなバッチファイルが『rename.bat』という名称であるとしてます。

```
set yy=%date:~0,4%
set mm=%date:~5,2%
set dd=%date:~8,2%

copy c:\qanat\csv\コピー元.csv c:\qanat\out\%yy%_%mm%_%dd%.csv
```

『rename.bat』を本製品サーバーと同じ Windows 環境の、C ドライブ直下に配置した場合、[コマンド] 欄には次のように入力します。

```
C:\rename.bat
```

[コマンド戻り値]には任意の値を入力します。

これは次のそれぞれのチェックボックスのうちどちらを選択するかにより本製品サーバーの解釈が異なります。

[障害発生時]をチェックした場合

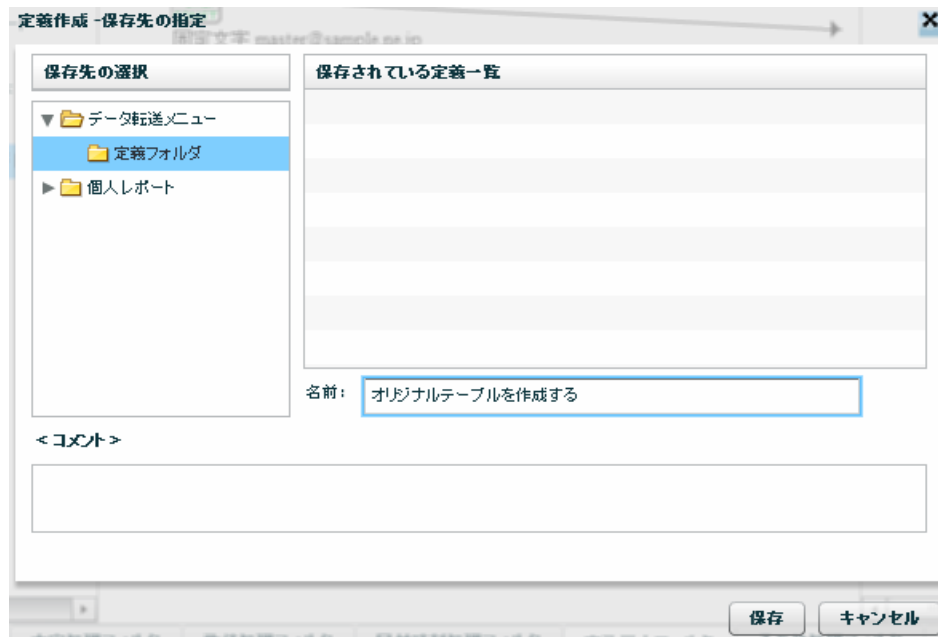
コマンド実行の結果、入力した任意の値が返された場合は「障害」と判断されます。

[成功時（他の値が戻った場合は障害）]をチェックした場合

コマンド実行の結果、入力した任意の値が返された場合は「処理成功」と判断されます。

手順4： 保存し実行します

データ転送定義を保存します。



データ転送定義名を入力し[保存]ボタンをクリックします。

支店情報ファイルは保存が完了した時点で対象のフォルダに作成されます。

保存後、データ転送定義『オリジナルファイルを作成する』を実行します。新しく作成された支店情報ファイルには入力元からの店舗情報がセットされていることを確認できるでしょう。

さらに、後処理が有効であるためコマンドが実行され、その結果新たなファイルが生成されていることも確認することができます。

2-4: スケジュールを作成する

これまで作成した複数の異なるデータ転送定義は一つのタスク定義にまとめ、一つの実行単位にすることができます。

また、データ転送定義や、タスク定義はタイマー起動を指定することで自動実行することができます。

目的	作成されるもの
<ul style="list-style-type: none">・タスク定義を作成する・スケジュール実行を作成する	<ul style="list-style-type: none">・タスク定義・スケジュール

事前修了しておきたい実習項目

1-1: データ転送定義を作成する (データベース)

1-2: データ転送定義を作成する (ファイル)

実習のシナリオ

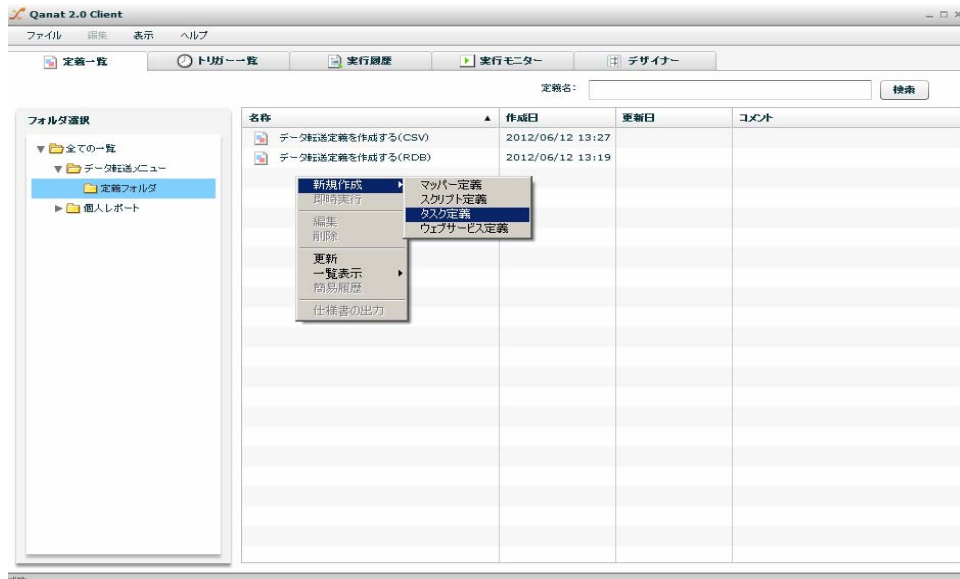
まず、二つのデータ転送定義を含めたタスク定義を一つ作成します。

作成したタスク定義は決められた日時で自動実行するようにスケジュール実行を指定します。

手順1： タスク定義を作成します

タスク定義では、複数のデータ転送定義を一つの意味のある実行単位にまとめ、一度の実行でそれら複数のデータ転送定義を順序を指定して実行します。

右クリックメニューの[新規作成] [タスク定義]をクリックします。



タスク定義作成では、まとめたい定義を保存先フォルダから選択し[追加]ボタンをクリックします。

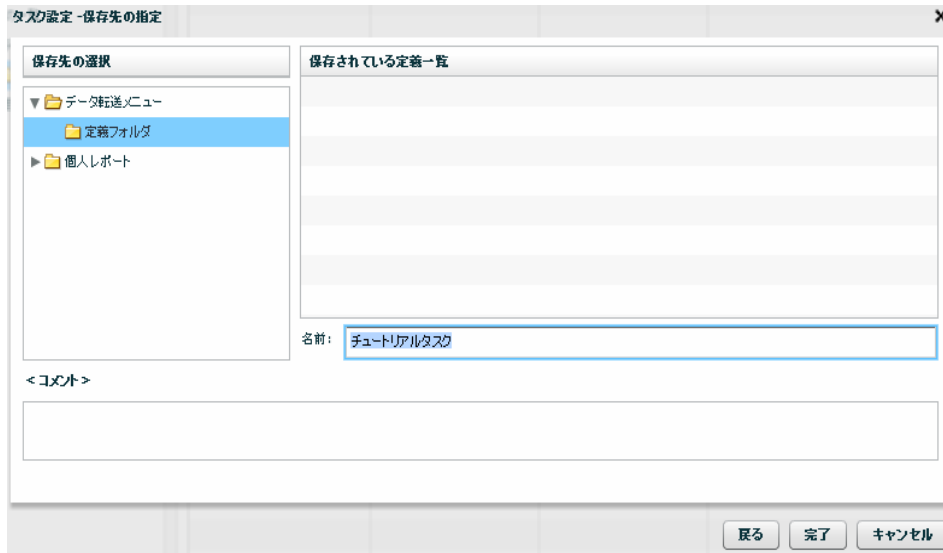
データ転送定義と同様に障害発生時とトランザクション指定が行えます。



[次へ]ボタンをクリックし保存先の指定と名前を入力します。

- ❗ トランザクション指定では出力リソースが異なる今回のチュートリアルのような場合は指定することができません。

[完了]をクリックし保存します。

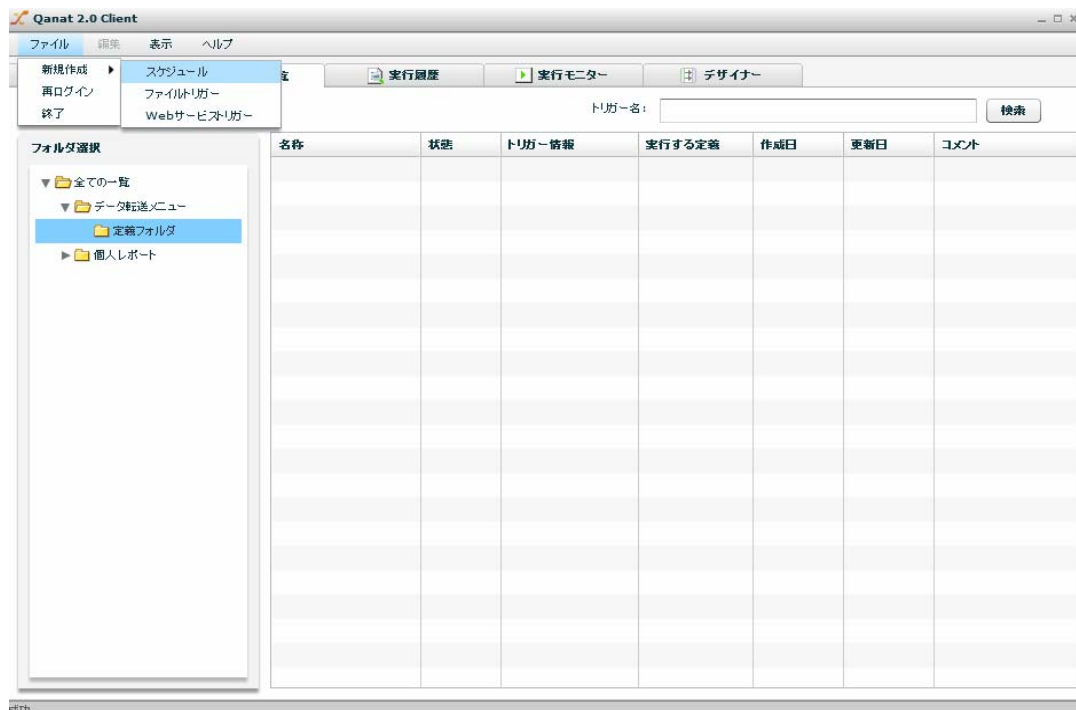


次に、作成した『チュートリアルタスク』をスケジュール実行します。

手順2: スケジュール定義を作成します

[トリガー一覧]タブを開きます。

画面の空白箇所でも右クリックしメニューから[新規作成] [スケジュール]をクリックします。



実行タイプ画面では自動実行するタイミングを指定します。

この実習では『毎月1日の午前0時』を指定します。

[次へ]をクリックします。

スケジュール設定-実行タイプ

スケジュール名: 新規

毎年 毎月 毎週 毎日 インターバル 一回

1 日 0 時 0 分

通知設定

戻る 次へ キャンセル

実行させたい定義を選択し、[次へ]をクリックします。

スケジュール設定-定義選択

スケジュール名: 新規

スケジュール情報: 毎月 1日 0時 0分

変数の表示

フォルダ選択

- データ転送メニュー
 - 定額フォルダ
- 個人レポート

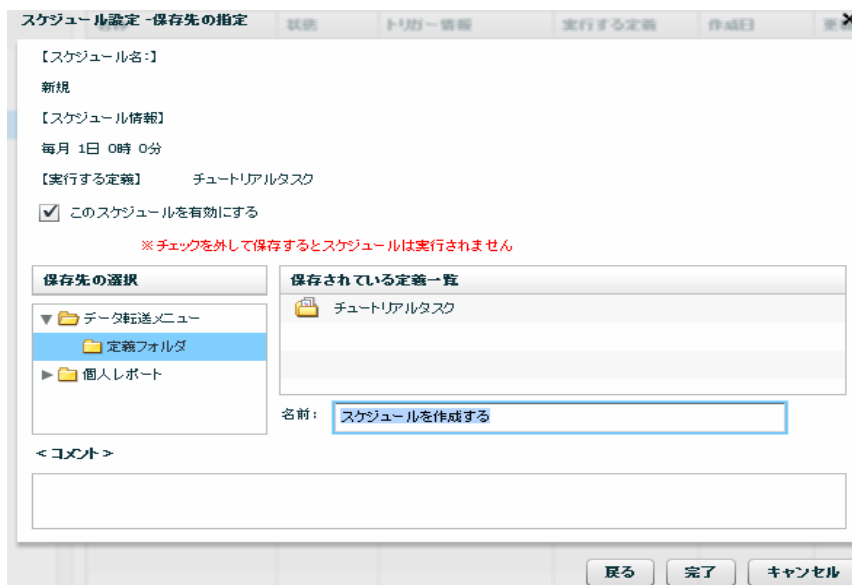
名称

- チュートリアルタスク

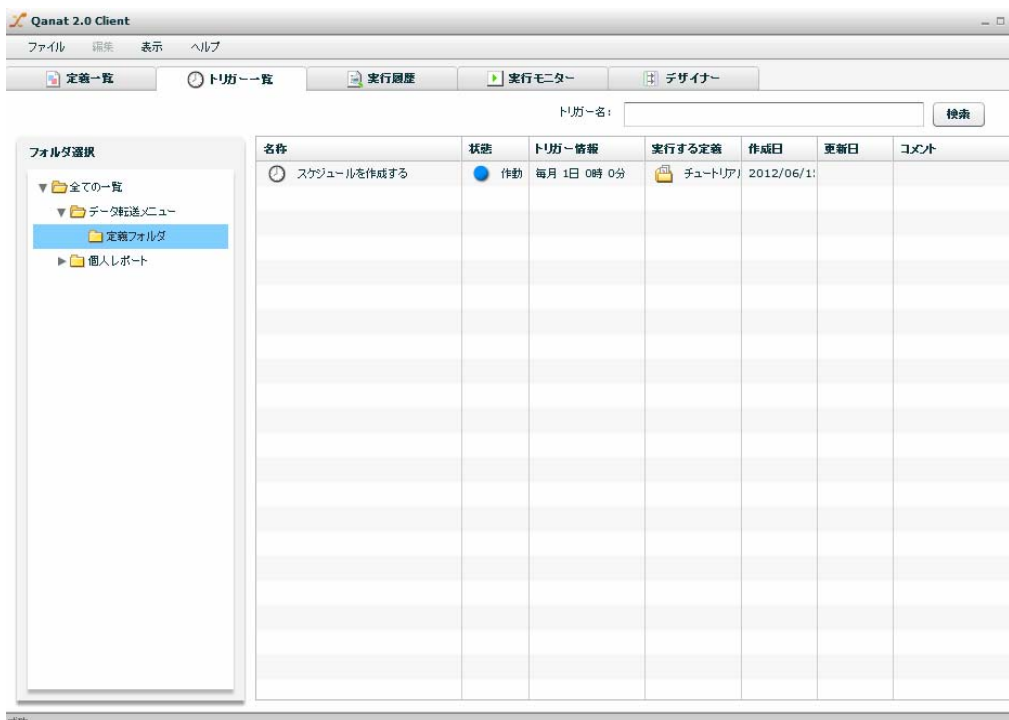
戻る 次へ キャンセル

スケジュール名を入力し保存先を指定します。

[完了]をクリックします。このとき画面上部の[このスケジュールを有効にする]をチェックします。



保存後、スケジューラタブの一覧画面で作成したスケジュール定義が有効となっていることを確認します。



決められた日時に指定されたデータ転送定義が実行されます。

2-5: ファイルトリガーを作成する

ファイルトリガーで指定されたフォルダの CSV ファイルを監視し、フォルダにファイルが作成されたタイミングで CSV ファイルのデータを元に RDB を更新します。

目的	作成されるもの
・ ファイルトリガーを作成する	・ データ転送定義 ・ ファイルトリガー

事前修了しておきたい実習項目

1-2 : データ転送定義を作成する (ファイル)

実習のシナリオ

まず、入力リソースのリソース、フォルダ、ファイルに定義変数を使用した CSV のデータ転送定義を一つ作成します。

作成したデータ転送定義をファイルトリガーで実行し、実行時の監視ファイル名をファイルトリガーからデータ転送定義の定義変数に渡します。

手順1： ファイルトリガーで実行する転送定義を作成します

ファイルトリガーは、作成時にトリガー変数を作成することができ、実行する転送定義の定義変数にトリガー変数を渡すことができます。

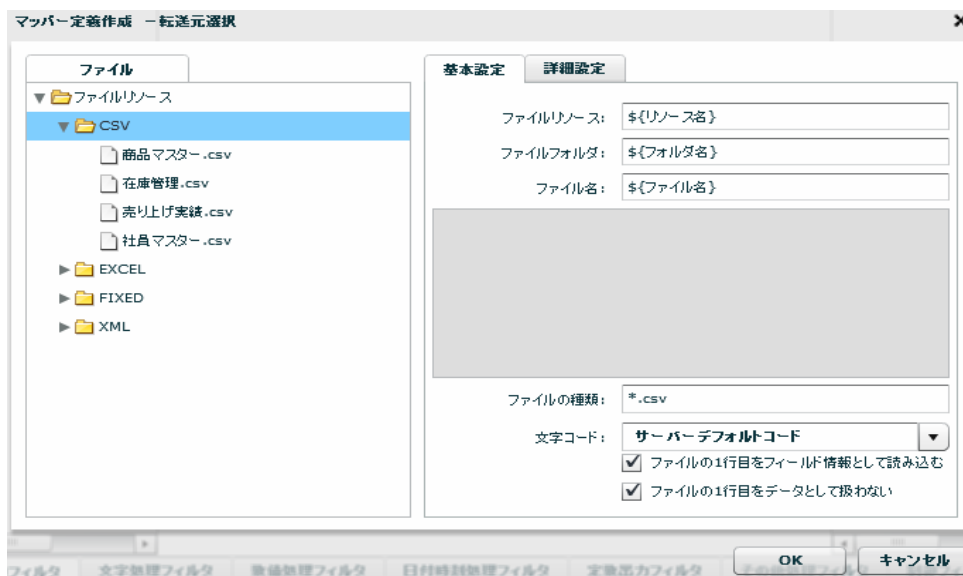
本実習では、ファイルトリガーが感知したファイルのリソース名・フォルダ名・ファイル名をトリガー変数に指定し、それを実行定義に渡すことで動的な処理を実現します。

まず、ファイルトリガーで実行する定義を作成します。

デザイナ画面で入力リソースに CSV ファイルを選択し、ファイルリソース、ファイルフォルダ、ファイル名を以下のように設定します。

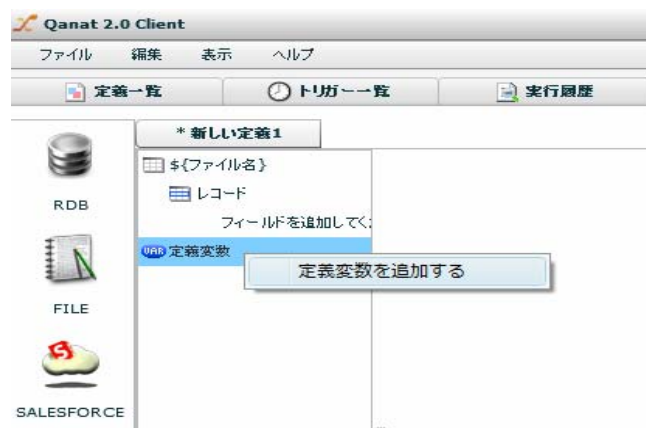
今回は簡単な為、リソース名、フォルダ名、ファイル名という名前の定義変数を使用します。

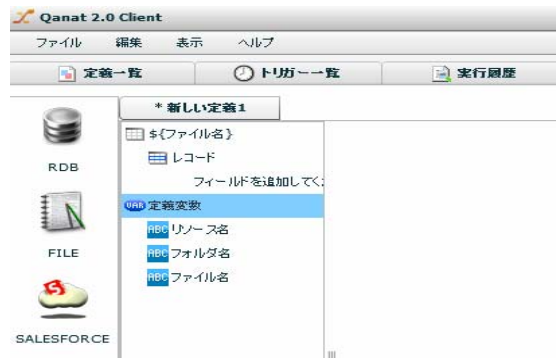
(定義変数を使用する場合\${定義変数名}と表記します。)



[OK]をクリックし、デザイナ画面で使用する定義変数を追加していきます。

今回はリソース名、フォルダ名、ファイル名の3つを追加します。





CSV ファイルのデータに合わせてフィールドを追加し出力に RDB (インサート) を選択し結線していきます。

データベースの作成は CD 内の/Sample/Tutorial/db2_uriagetable.sql をご利用ください。

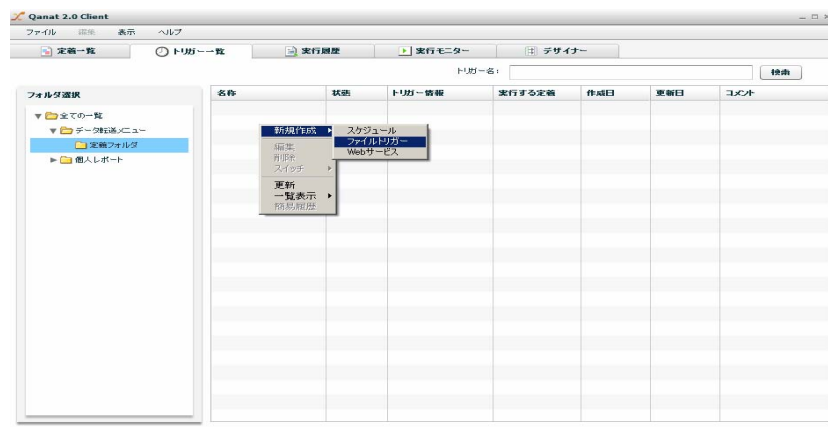


定義の作成ができましたので保存します。(例では売上更新定義という名前で保存)

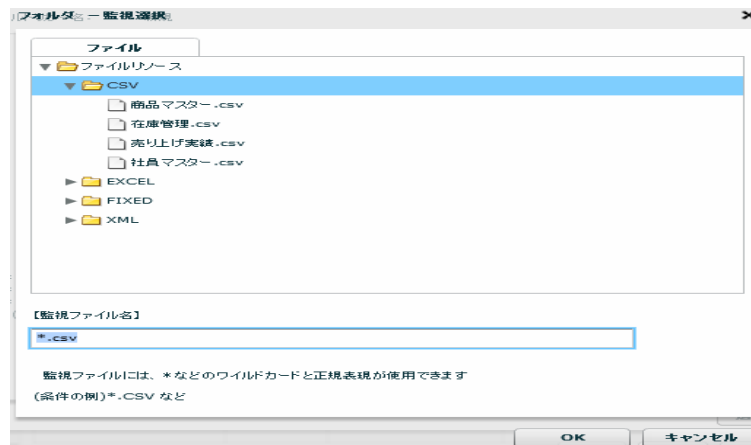
手順2: ファイルトリガーを作成します

次に、作成した『売上更新定義』を実行するファイルトリガーを作成します。

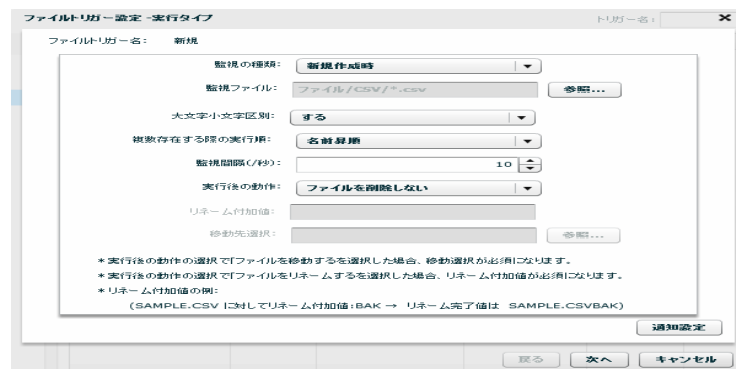
[トリガー一覧]タブから[新規作成] - [ファイルトリガー]を選択してください。



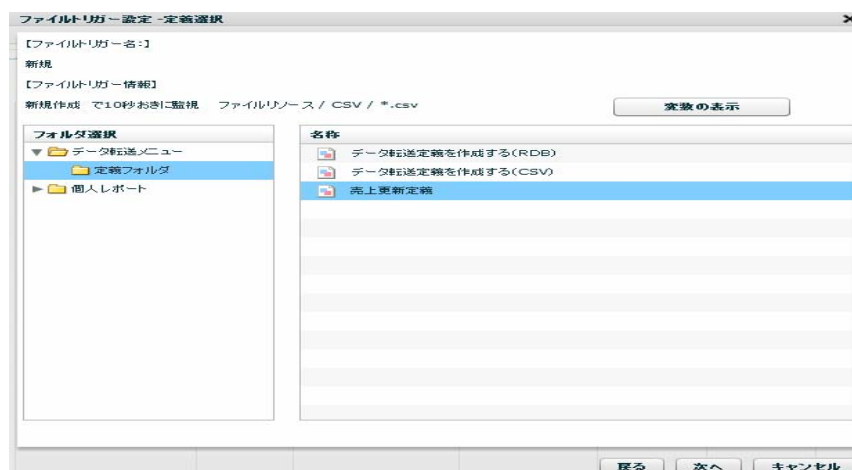
監視ファイルで[参照]をクリックし監視するフォルダを選択しファイル名を*.csvとします。



設定が終了したら[次へ]をクリックします。

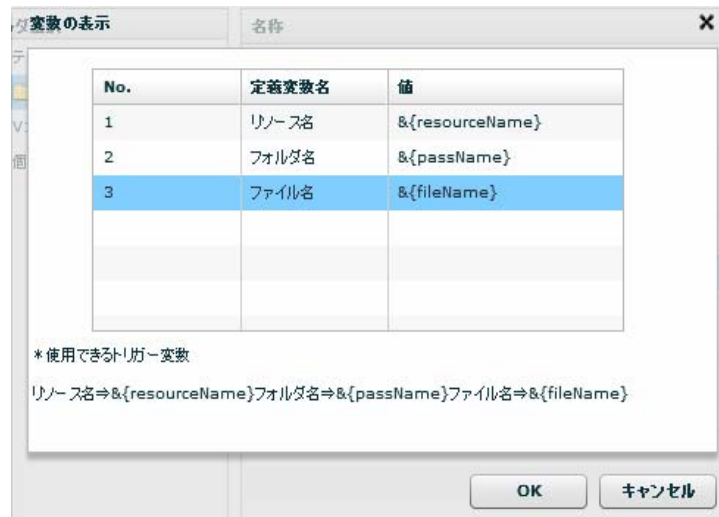


実行する定義には手順1で作成した転送定義を選択します。



[変数の表示]をクリックし、渡すトリガー変数を設定します。

定義変数名には実行する定義の定義変数が自動でセットされます。



今回はリソース名に&{resourceName}、フォルダ名に&{passName}、ファイル名に&{fileName}を渡します。

トリガーの発火条件となったファイルのファイルリソース名、ファイルフォルダ名、ファイル名が各々実行定義の定義変数にセットされることになります。

[OK]をクリックし[次へ]をクリックします。

トリガー名を指定して保存します。(例では売上更新トリガーとします。)

手順3: ファイルトリガーを発火します

それではファイルトリガーを発火させてみましょう。

ファイルトリガーの監視ファイルに選択したフォルダにサンプルデータの売り上げ実績.csvを追加してみましょう。

```

年度, 四半期, 担当者コード, 担当者名, 地区コード, 担当地区, 商品コード, 商品名, 数量, 売り上げ金額, 粗利益, 定価, 仕入れ単価
2002, 2, 1, '山田太郎', 1, '横浜', 2, 'ラーメン', 10, 20000, 3000, 2000, 1000
2002, 3, 1, '山田太郎', 1, '横浜', 2, 'ラーメン', 5, 10000, 1500, 2000, 1000
2003, 1, 1, '山田太郎', 1, '横浜', 1, '目玉焼き', 30, 15000, 5000, 500, 150
2003, 3, 3, '鈴木三郎', 3, '大阪', 4, 'たこ焼き', 10, 10000, 2500, 1000, 400
2004, 1, 2, '佐藤次郎', 2, '東京', 3, '寿司', 20, 30000, 10000, 1500, 500
2008, 4, 3, '鈴木三郎', 3, '大阪', 4, '焼そば', 20, 20000, 5000, 1000, 400

```

このCSVファイルのデータがRDBの売り上げ実績テーブルに出力されていきます。

2-6:全銀協データを扱う

全銀協データとは、全国銀行協会連合会がデータ伝送を行うために定めたデータです。特定口座への振り込み情報などが1レコード120Byteの固定長のファイルで扱われます。この章では本製品を使用した次の手順

1. 全銀協の固定長ファイル データベースへの取り込み
2. データベース 全銀協の固定長ファイルへの出力

について説明します。

固定長ファイルフォーマットとデータベース構成

全銀協の固定長ファイル（振込データ）

以下の4種類のレコードフォーマットから構成されています。

1210	23	送金元(1091) 33		送金先(1091) 33	送金種別 1	送金元口座番号11234567			ヘッダ (1レコード)
2	33	振込元(1091) 33	振込先(1091) 33	振込種別 4	振込額	振込手数料	振込残高	振込日	データ (複数レコード)
2	78	振込元(1091) 33	振込先(1091) 33	振込種別 2	振込額	振込手数料	振込残高		
2	123	振込元(1091) 33	振込先(1091) 33	振込種別 34	振込額	振込手数料	振込残高		
2	33	振込元(1091) 33	振込先(1091) 33	振込種別 1	振込額	振込手数料	振込残高		
2	78	振込元(1091) 33	振込先(1091) 33	振込種別 2	振込額	振込手数料	振込残高		
2	123	振込元(1091) 33	振込先(1091) 33	振込種別 12	振込額	振込手数料	振込残高		
2	123	振込元(1091) 33	振込先(1091) 33	振込種別 1	振込額	振込手数料	振込残高		
2	33	振込元(1091) 33	振込先(1091) 33	振込種別 23	振込額	振込手数料	振込残高		
2	33	振込元(1091) 33	振込先(1091) 33	振込種別 4	振込額	振込手数料	振込残高	トレーラ (1レコード)	
2	78	振込元(1091) 33	振込先(1091) 33	振込種別 1	振込額	振込手数料	振込残高		
8	10..0	933000.0							エンド (1レコード)

レコード識別子

データベースの振込データ

以下のように1レコードは全銀協固定長ファイルのヘッダとデータが結合した構成とします。

1210	23	送金元(1091) 33	送金先(1091) 33	送金種別 1	送金元口座番号11234567	2	33	振込元(1091) 33	振込先(1091) 33	振込種別 4	振込額	振込手数料	振込残高	振込日
1210	23	送金元(1091) 33	送金先(1091) 33	送金種別 1	送金元口座番号11234567	2	78	振込元(1091) 33	振込先(1091) 33	振込種別 2	振込額	振込手数料	振込残高	振込日
1210	23	送金元(1091) 33	送金先(1091) 33	送金種別 1	送金元口座番号11234567	2	123	振込元(1091) 33	振込先(1091) 33	振込種別 34	振込額	振込手数料	振込残高	振込日
1210	23	送金元(1091) 33	送金先(1091) 33	送金種別 1	送金元口座番号11234567	2	33	振込元(1091) 33	振込先(1091) 33	振込種別 1	振込額	振込手数料	振込残高	振込日
1210	23	送金元(1091) 33	送金先(1091) 33	送金種別 1	送金元口座番号11234567	2	78	振込元(1091) 33	振込先(1091) 33	振込種別 2	振込額	振込手数料	振込残高	振込日
1210	23	送金元(1091) 33	送金先(1091) 33	送金種別 1	送金元口座番号11234567	2	123	振込元(1091) 33	振込先(1091) 33	振込種別 12	振込額	振込手数料	振込残高	振込日
1210	23	送金元(1091) 33	送金先(1091) 33	送金種別 1	送金元口座番号11234567	2	123	振込元(1091) 33	振込先(1091) 33	振込種別 1	振込額	振込手数料	振込残高	振込日
1210	23	送金元(1091) 33	送金先(1091) 33	送金種別 1	送金元口座番号11234567	2	33	振込元(1091) 33	振込先(1091) 33	振込種別 23	振込額	振込手数料	振込残高	振込日
1210	23	送金元(1091) 33	送金先(1091) 33	送金種別 1	送金元口座番号11234567	2	33	振込元(1091) 33	振込先(1091) 33	振込種別 4	振込額	振込手数料	振込残高	振込日
1210	23	送金元(1091) 33	送金先(1091) 33	送金種別 1	送金元口座番号11234567	2	78	振込元(1091) 33	振込先(1091) 33	振込種別 1	振込額	振込手数料	振込残高	振込日

ヘッダ (1レコード)

データ (複数レコード)

全銀協の固定長ファイル（振込データ）のレコードフォーマット

ヘッダ

項番	項目名	桁数	文字形態	備考
1	データ区分	1	半角数字	1
2	種別コード	2	半角数字	21 11 12
3	コード区分	1	半角数字	
4	依頼人コード	10	半角数字	
5	依頼人名	40	半角カナ	
6	振込指定日	4	半角数字	4 MMDD
7	仕向金融機関番号	4	半角数字	
8	仕向金融機関名	15	半角カナ	
9	仕向支店番号	3	半角数字	-
10	仕向支店名	15	半角カナ	
11	預金種目（依頼人）	1	半角数字	1 2
12	口座番号（依頼人）	7	半角数字	-
13	ダミー	17	半角英数字	

データ

項番	項目名	桁数	文字形態	備考
1	データ区分	1	半角数字	2
2	被仕向金融機関番号	4	半角数字	-
3	被仕向金融機関名	15	半角カナ	左詰、スペース埋め
4	被仕向支店番号	3	半角数字	-
5	被仕向支店名	15	半角カナ	左詰、スペース埋め
6	手形交換所番号	4	半角数字	省略可（スペース）
7	預金種目	1	半角数字	普通：1、当座：2、貯蓄：4
8	口座番号	7	半角数字	右詰、0埋め
9	受取人名	30	半角カナ	-
10	振込金額	10	半角数字	右詰、0埋め
11	新規コード	1	半角数字	省略可（スペース）
12	顧客コード1	10	半角数字	省略可（スペース）
13	顧客コード2	10	半角数字	省略可（スペース）
14	ダミー	9	半角英数字	スペース

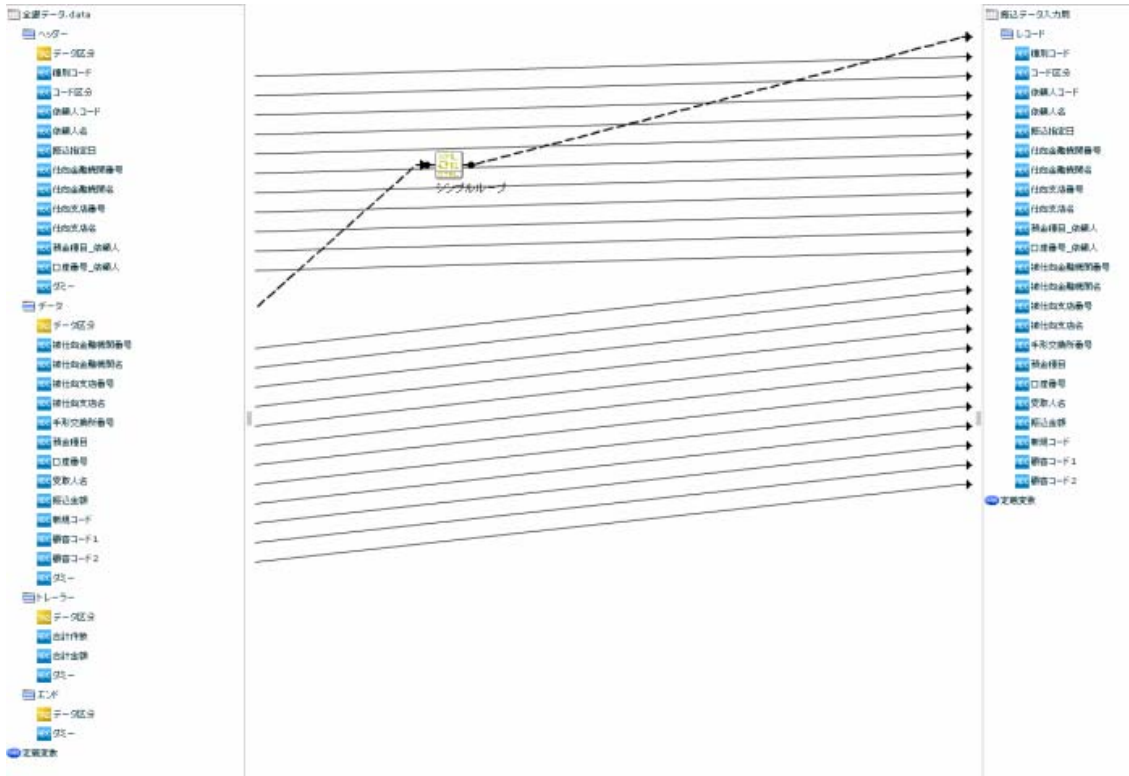
トレーラ

項番	項目名	桁数	文字形態	備考
1	データ区分	1	半角数字	8
2	合計件数	6	半角数字	データ・レコードの合計。右詰、0埋め
3	合計金額	12	半角数字	データ・レコードの合計。右詰、0埋め
4	ダミー	101	半角英数字	スペース

エンド

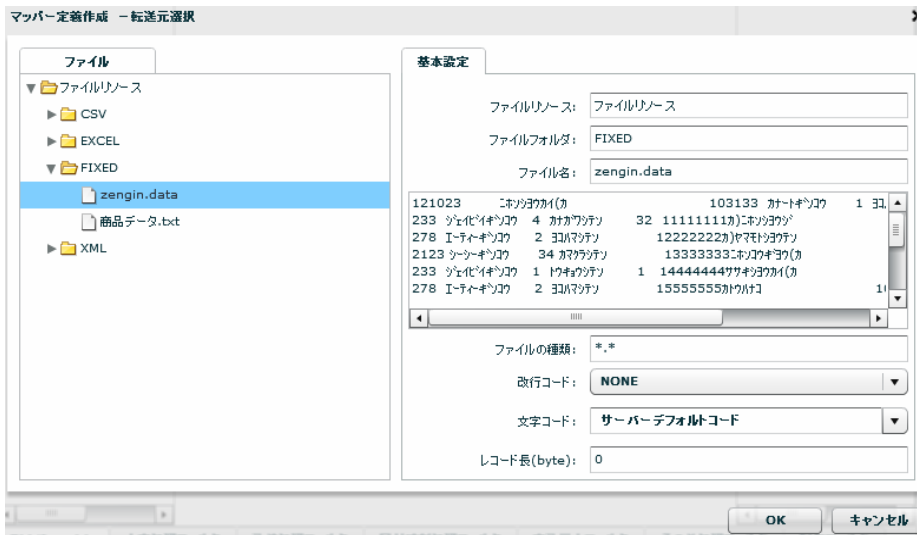
項番	項目名	桁数	文字形態	備考
1	データ区分	1	半角数字	9
2	ダミー	119	半角英数字	スペース

固定長ファイル（振込データ）のデータベースへの取り込み



転送元には全銀協の固定長ファイルを指定し、レコードパターンに全銀協のレコードフォーマット4種類を用意します。転送先には事前に用意してある振込みテーブルを選択します。

1. 入力固定長ファイルの設定



サンプルに付属している全銀協フォーマットの固定長ファイル（zengin.data）を指定します。

【改行コード】：CR+LF

【文字コード】：SHIFT_JIS

【レコード長】：改行コードも含めるため、122byteを設定します。

各レコードパターンとフィールドの設定

ヘッダー**データ区分：**

【フィールドタイプ】に【レコード識別子】を選択して、【レコード識別子】には「1」を設定

**それ以外のフィールド：**

【フィールドタイプ】に【文字列】を選択して、【揃え】に【左揃え】、【パディング文字】に【半角スペース】を設定、【フィールド長 (byte) 】には指定されたサイズを設定

**データ**

基本的にヘッダーと同じです。

データ区分：

【フィールドタイプ】に【レコード識別子】を選択して、【レコード識別子】には「2」を設定

それ以外のフィールド：

【フィールドタイプ】に【文字列】を選択して、【揃え】に【左揃え】、【パディング文字】に【半角スペース】を設定、【フィールド長 (byte) 】には指定サイズを設定

トレーラー

基本的にヘッダーと同じです。

データ区分：

【フィールドタイプ】に【レコード識別子】を選択して、【レコード識別子】には「8」を設定

それ以外のフィールド：

【フィールドタイプ】に【文字列】を選択して、【揃え】に【左揃え】、【パディング文字】に【半角スペース】を設定、【フィールド長 (byte) 】には指定サイズを設定

エンド

基本的にヘッダーと同じです。

データ区分：

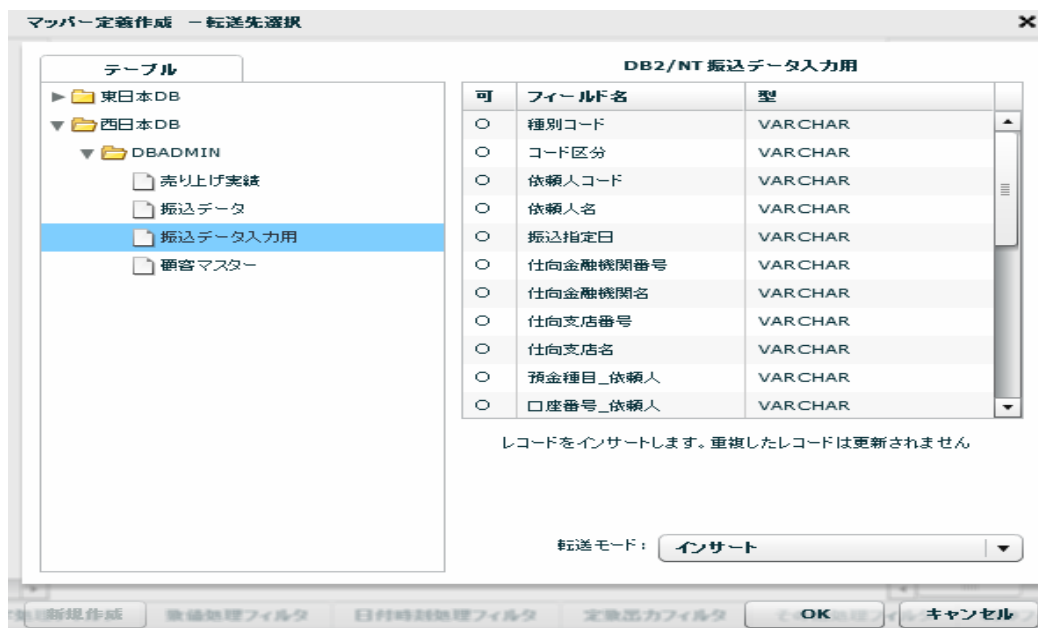
【フィールドタイプ】に【レコード識別子】を選択して、【レコード識別子】には「9」を設定

それ以外のフィールド：

【フィールドタイプ】に【文字列】を選択して、【揃え】に【左揃え】、【パディング文字】に【半角スペース】を設定、【フィールド長 (byte) 】には指定サイズを設定

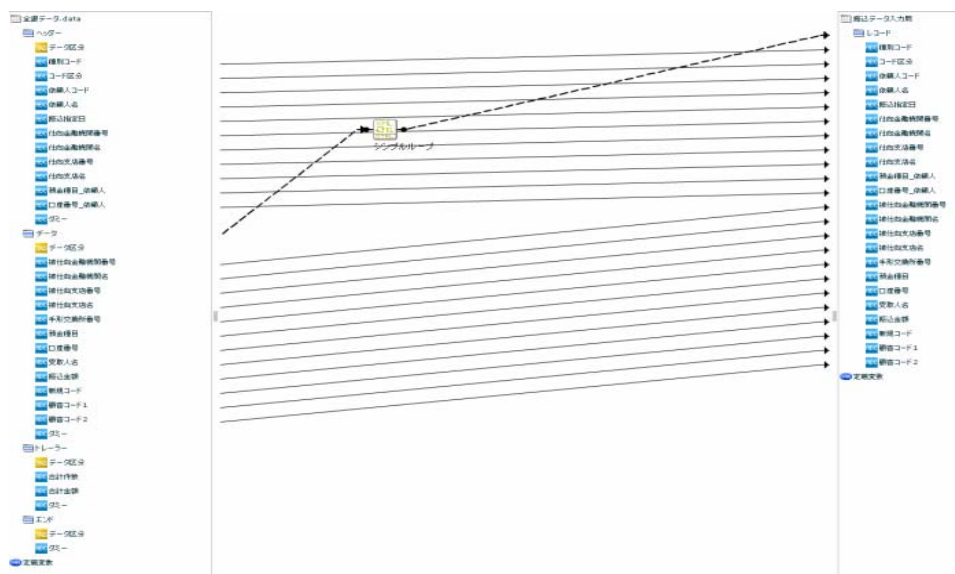
2. 出力データベースの設定

データベースの作成は CD 内の/Sample/Tutorial/db2_zengin_nodata.sql をご利用ください。



サンプルに付属している SQL で作成したテーブルを指定します。

3. マッピング

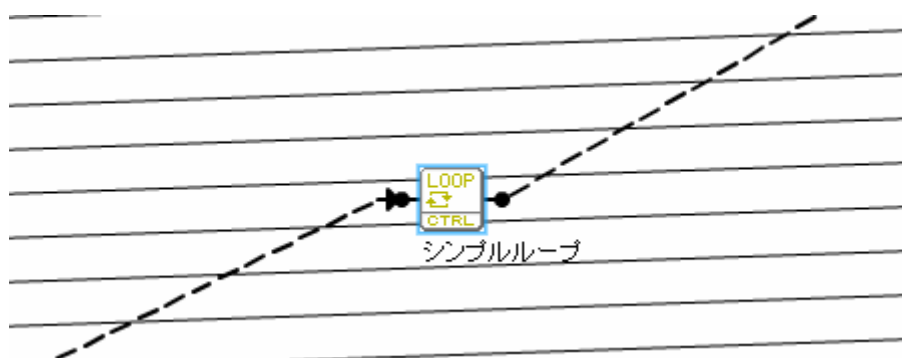


上記のように接続します

[編集] - [自動割当] - [名前] を実行して、同盟の入出力フィールドを結び付けます。



転送元のレコードパターン【データ】と転送先のレコードパターンをシンプルループで繋げます。



以上で、固定長ファイル（振込データ）のデータベースへの取り込みの設定は終了です。定義を保存して実行してみてください。

データベース 全銀協の固定長ファイルへの出力

転送元データベース /Sample/Tutorial/db2_zengin.sql にて作成してください

1210	23	ニホコカ(カ)1031	33	カトキコカ	1	ヨコワケ(カ)11234567	2	33	ジュエ(イ)ンコ	4	カカ(カ)ワケ	3211111111	カ(カ)ニホコカ	1500000	123
1210	23	ニホコカ(カ)1031	33	カトキコカ	1	ヨコワケ(カ)11234567	2	78	エチ(キ)ンコ	2	ヨコワケ	12222222	カ(カ)ヤマト(コ)ワケ	2300000	234
1210	23	ニホコカ(カ)1031	33	カトキコカ	1	ヨコワケ(カ)11234567	2	123	ツツ(キ)ンコ	34	カマ(カ)ワケ	13333333	ニホコカ(カ)ヨ(カ)	1800000	345
1210	23	ニホコカ(カ)1031	33	カトキコカ	1	ヨコワケ(カ)11234567	2	33	ジュエ(イ)ンコ	1	トク(カ)ワケ	1144444444	サ(カ)キヨ(カ)イ(カ)	1000000	456
1210	23	ニホコカ(カ)1031	33	カトキコカ	1	ヨコワケ(カ)11234567	2	78	エチ(キ)ンコ	2	ヨコワケ	15555555	カ(カ)ク(カ)	10000	567
1210	23	ニホコカ(カ)1031	33	カトキコカ	1	ヨコワケ(カ)11234567	2	123	ツツ(キ)ンコ	12	オク(カ)ワケ	16666666	オク(カ)ワケ	10000	678
1210	23	ニホコカ(カ)1031	33	カトキコカ	1	ヨコワケ(カ)11234567	2	123	ツツ(キ)ンコ	1	トク(カ)ワケ	17777777	サ(カ)イ(カ)ニホ(カ)キヨ(カ)イ(カ)	1200000	789
1210	23	ニホコカ(カ)1031	33	カトキコカ	1	ヨコワケ(カ)11234567	2	33	ジュエ(イ)ンコ	23	カマ(カ)ワケ	18888888	イ(カ)ト(カ)コ(カ)ム(カ)	2300000	890
1210	23	ニホコカ(カ)1031	33	カトキコカ	1	ヨコワケ(カ)11234567	2	33	ジュエ(イ)ンコ	4	カカ(カ)ワケ	3213333333	イ(カ)カ(カ)ニホ(カ)ム(カ)	1000000	901
1210	23	ニホコカ(カ)1031	33	カトキコカ	1	ヨコワケ(カ)11234567	2	78	エチ(キ)ンコ	1	トク(カ)ワケ	11111111	ミ(カ)ア(カ)ワケ(カ)	10000	222

ヘッダ (1レコード)

データ (複数レコード)



転送先全銀協ファイル 全銀データ.data

1210	23	ニホコカ(カ)1031	33	カトキコカ	1	ヨコワケ(カ)11234567									
2	33	ジュエ(イ)ンコ	4	カカ(カ)ワケ	3211111111	カ(カ)ニホコカ	1500000	123							
2	78	エチ(キ)ンコ	2	ヨコワケ	12222222	カ(カ)ヤマト(コ)ワケ	2300000	234							
2	123	ツツ(キ)ンコ	34	カマ(カ)ワケ	13333333	ニホコカ(カ)ヨ(カ)	1800000	345							
2	33	ジュエ(イ)ンコ	1	トク(カ)ワケ	1144444444	サ(カ)キヨ(カ)イ(カ)	1000000	456							
2	78	エチ(キ)ンコ	2	ヨコワケ	15555555	カ(カ)ク(カ)	10000	567							
2	123	ツツ(キ)ンコ	12	オク(カ)ワケ	16666666	オク(カ)ワケ	10000	678							
2	123	ツツ(キ)ンコ	1	トク(カ)ワケ	17777777	サ(カ)イ(カ)ニホ(カ)キヨ(カ)イ(カ)	1200000	789							
2	33	ジュエ(イ)ンコ	23	カマ(カ)ワケ	18888888	イ(カ)ト(カ)コ(カ)ム(カ)	2300000	890							
2	33	ジュエ(イ)ンコ	4	カカ(カ)ワケ	3213333333	イ(カ)カ(カ)ニホ(カ)ム(カ)	1000000	901							
2	78	エチ(キ)ンコ	1	トク(カ)ワケ	11111111	ミ(カ)ア(カ)ワケ(カ)	10000	222							
8	10.0	933000.0													
9															

ヘッダ (1レコード)

データ (複数レコード)

トレーラ (1レコード)

エンド (1レコード)

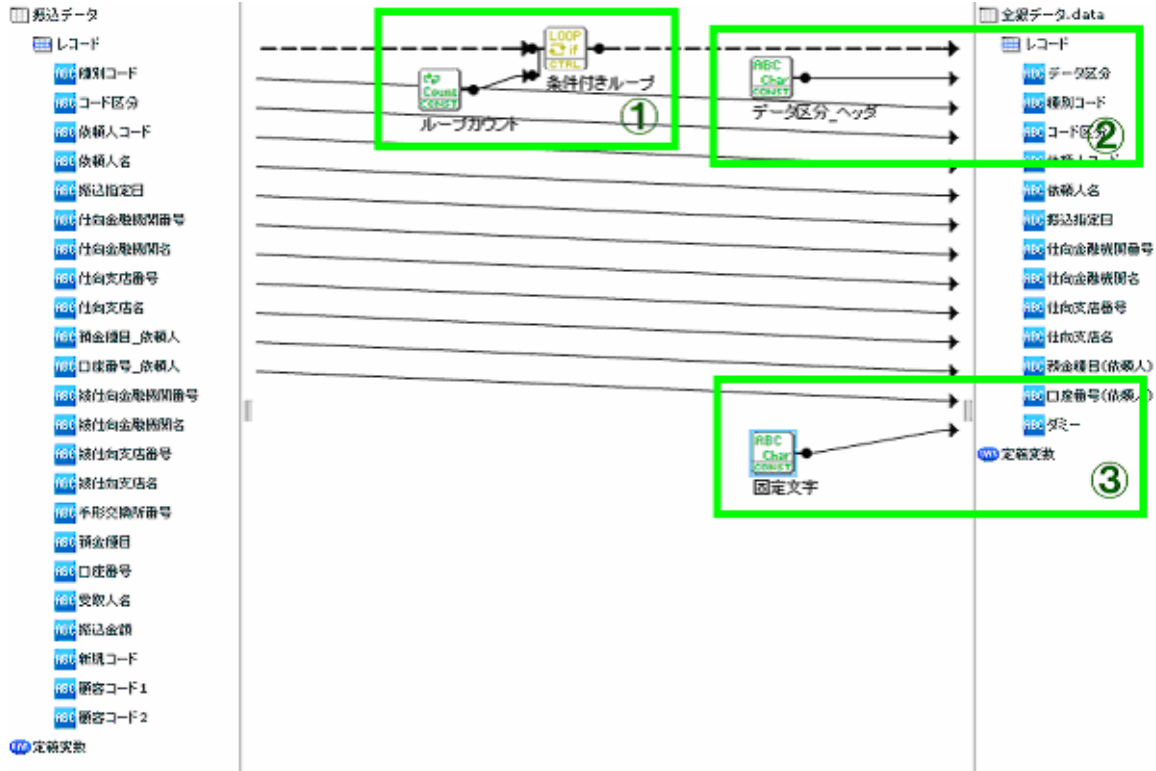
レコード識別子

「全銀データ_ヘッダ作成」、「全銀データ_データ作成」、「全銀データ_トレーラ作成」、「全銀データ_エンド作成」の4つの定義を作成します。

- ・全銀データ_ヘッダ作成：データベースの1レコード目からヘッダ情報を抜き出し、「全銀データ.data」に1行出力します。
- ・全銀データ_データ作成：データベースの全レコードからデータ情報を抜き出し、「全銀データ.data」に追記します。データ情報から「合計件数」と「合計金額」を計算し、定義変数に渡します。
- ・全銀データ_トレーラ作成：「データ出力」で渡された「合計件数」と「合計金額」を「全銀データ.data」に1行追記します。
- ・全銀データ_エンド作成：終了データを zengin_out.data に1行追記します。

上記4つの定義をタスクにて順番に実行していき、一つの「全銀データ.data」に出力していきます。

1. 全銀データ_ヘッダ作成



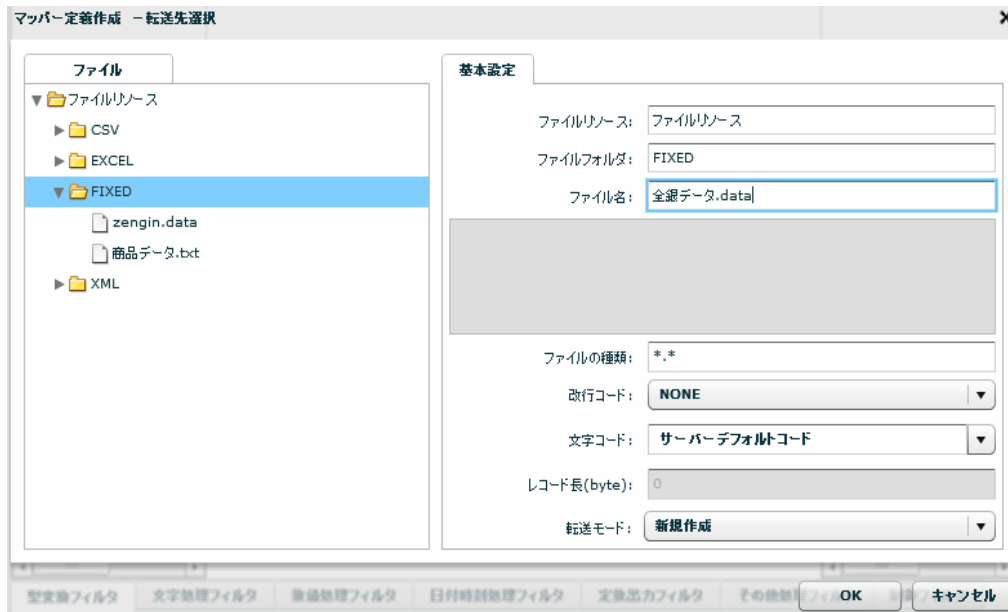
転送元には事前に用意してあるデータベースの「振込データ」を選択します。転送先には全銀協の固定長ファイル「全銀データ.data」を指定し、レコードパターンに全銀協のレコードフォーマットの「ヘッダ」部分を用意します。

転送元：



サンプルに付属している SQL で作成したテーブルを指定します。

転送先：



転送先に固定長ファイルの「全銀データ.data」を指定します。

転送モードは【新規作成】としてください。

各フィールドの設定

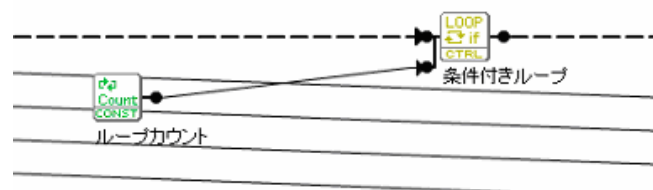
【フィールドタイプ】に【文字列】を選択して、【揃え】に【左揃え】、【パディング文字】に【半角スペース】を設定、【フィールド長 (byte) 】には指定サイズを設定



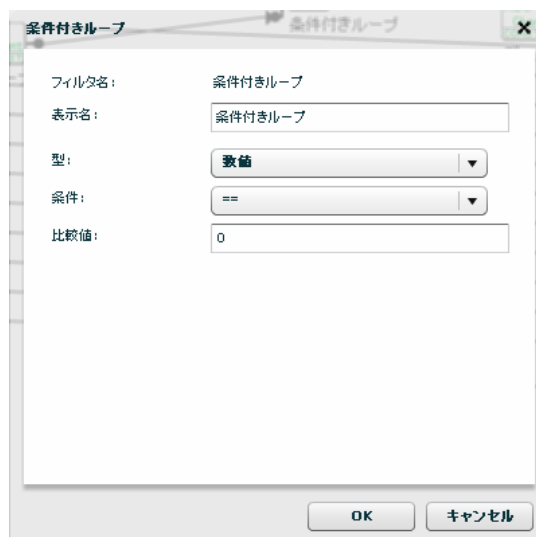
同名フィールドを結び付けます

(1) ヘッダは1行だけ出力します。データベースのヘッダ情報は全てのレコードで同一です。

「条件付ループ」でループカウントが0の、1レコード目のヘッダ情報のみ出力するようにします。

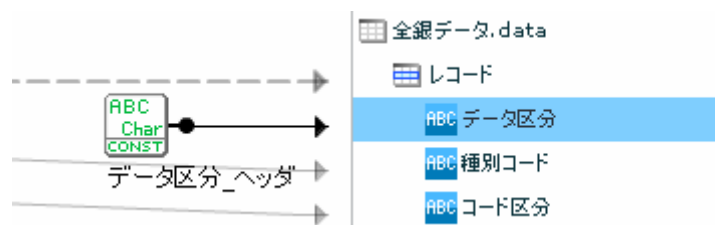


1レコード目のヘッダ情報のみ出力

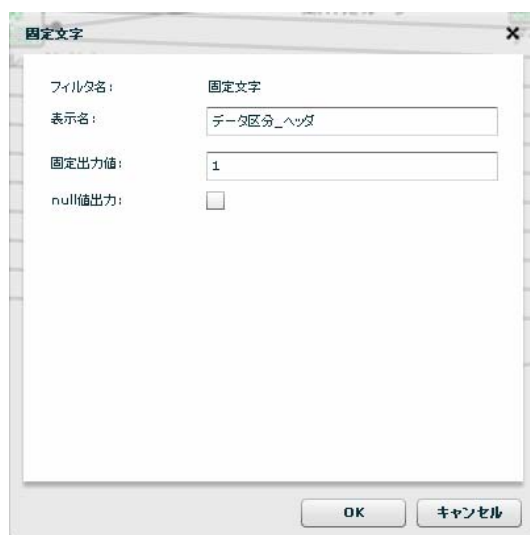


条件付きループで1レコード目のみ出力させる

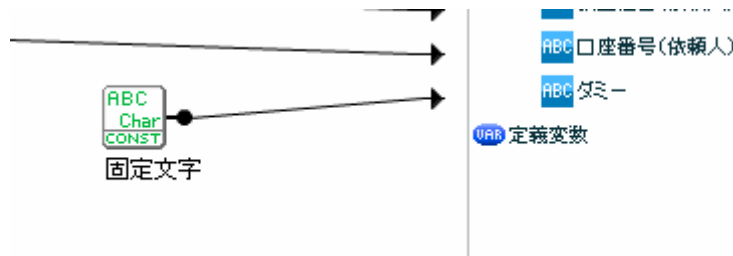
(2) データ区分は固定で1を出力します。固定長ファイル固有の情報のため、データベースには含まれていないため、マッピングで対応します。



データ区分は固定で1を出力

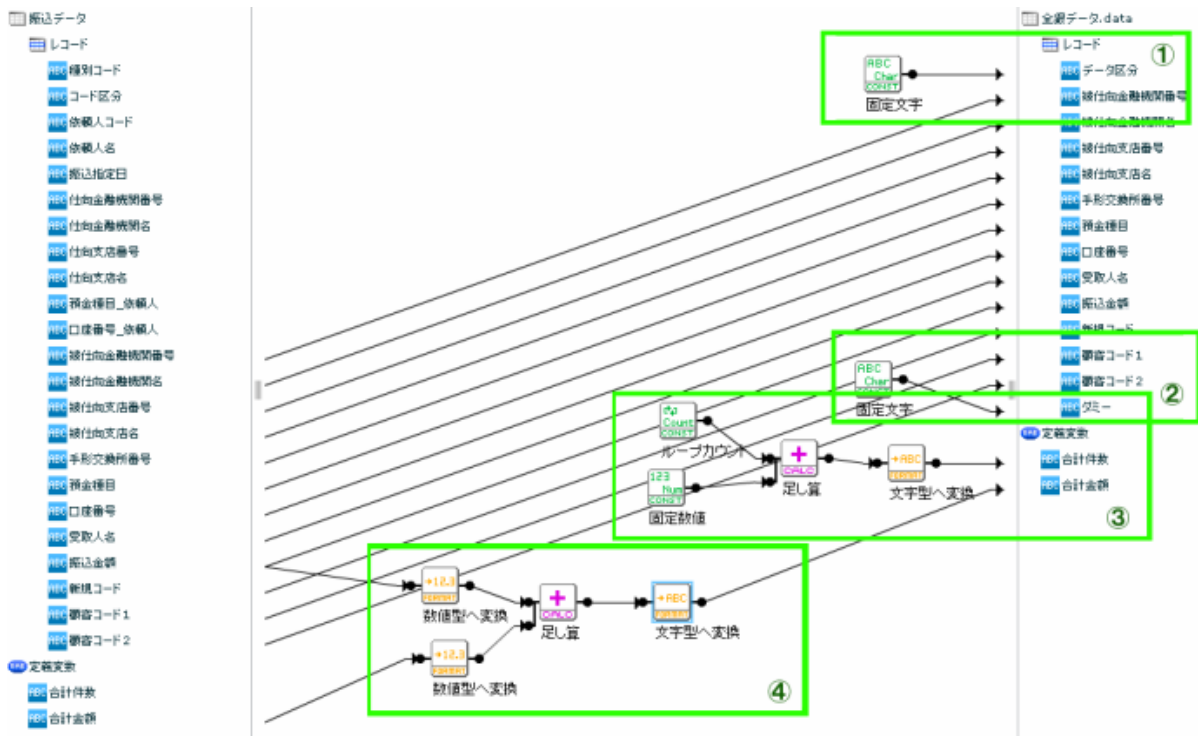


(3) ダミーに半角スペースを 1 文字指定します。何も指定されていない場合は NULL と判定されて NULL 文字でフィールドが埋められてしまい、テキストファイルになりません。半角スペースを一つ入れることで文字列と判定され、「パディング文字」に指定した半角スペースがフィールドを埋めます。



半角スペースを固定文字フィルターに設定してダミーに出力

2. 全銀データ_データ作成



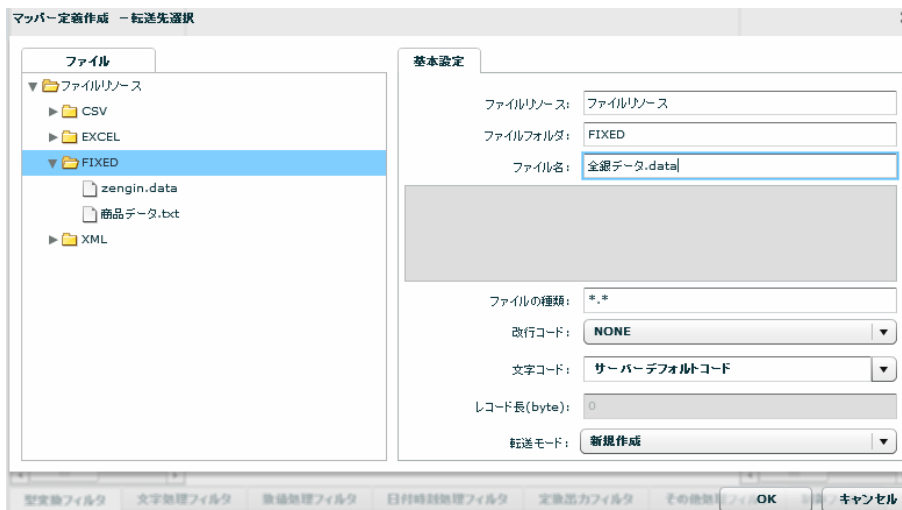
転送元には事前に用意してあるデータベースの「振込データ」を選択します。転送先には全銀協の固定長ファイル「全銀データ.data」を指定し、レコードパターンに全銀協のレコードフォーマットの「データ」部分を用意します。

転送元：



サンプルに付属している SQL で作成したテーブルを指定します。

転送先：



転送先に固定長ファイルの「全銀データ.data」を指定します。

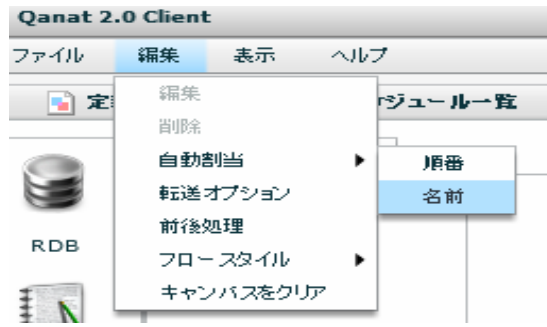
転送モードは【追加書き込み】としてください。

各フィールドの設定

【フィールドタイプ】に【文字列】を選択して、【揃え】に【左揃え】、【パディング文字】に【半角スペース】を設定、【フィールド長 (byte)】には指定サイズを設定

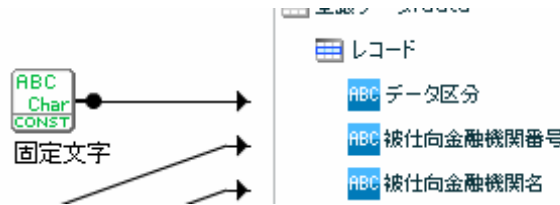
定義変数

「合計件数」と「合計金額」の定義変数を設定します。この定義内で合計件数と合計金額を集計し、次のトレーラ出力時に使用します。合計金額のデフォルト値は「0」を指定します。



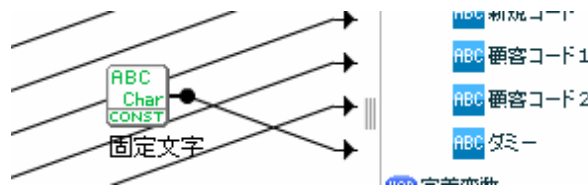
同名フィールドを結び付けます

(1) データ区分は固定で 2 を出力します。固定長ファイル固有の情報のため、データベースには含まれていないため、マッピングで対応します。



データ区分は固定で 2 を出力

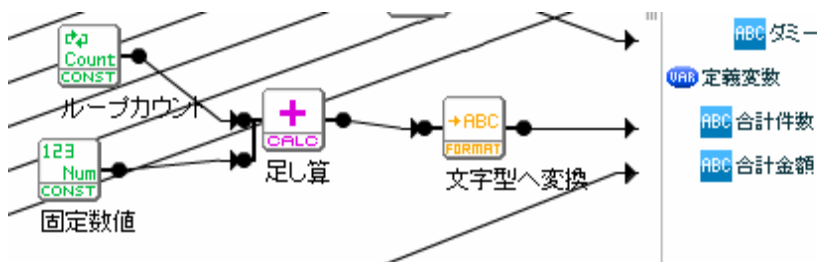
(2) ダミーに半角スペースを 1 文字指定します。



半角スペースを固定文字フィルターに設定してダミーに出力

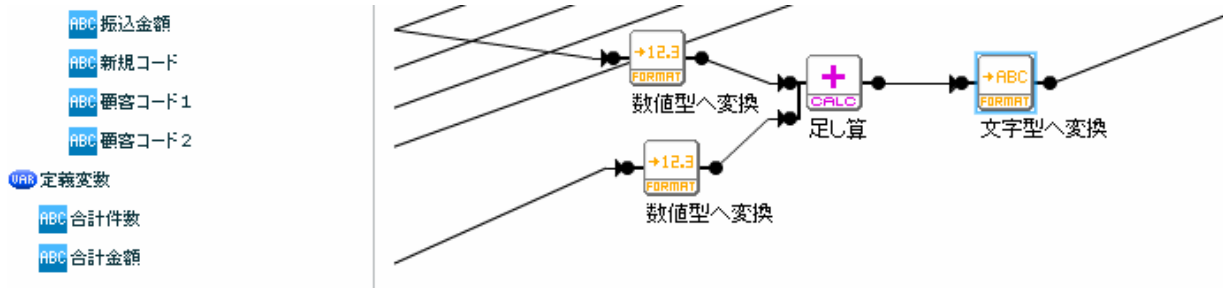
(3) レコード数を定義変数の「合計件数」に設定します。ループカウントを使用していますが、ループカウントは 0 スタートなので固定数値フィルターに 1 を設定してプラスして出力しています。

「合計件数」の値は次のトレーラ出力で出力します。

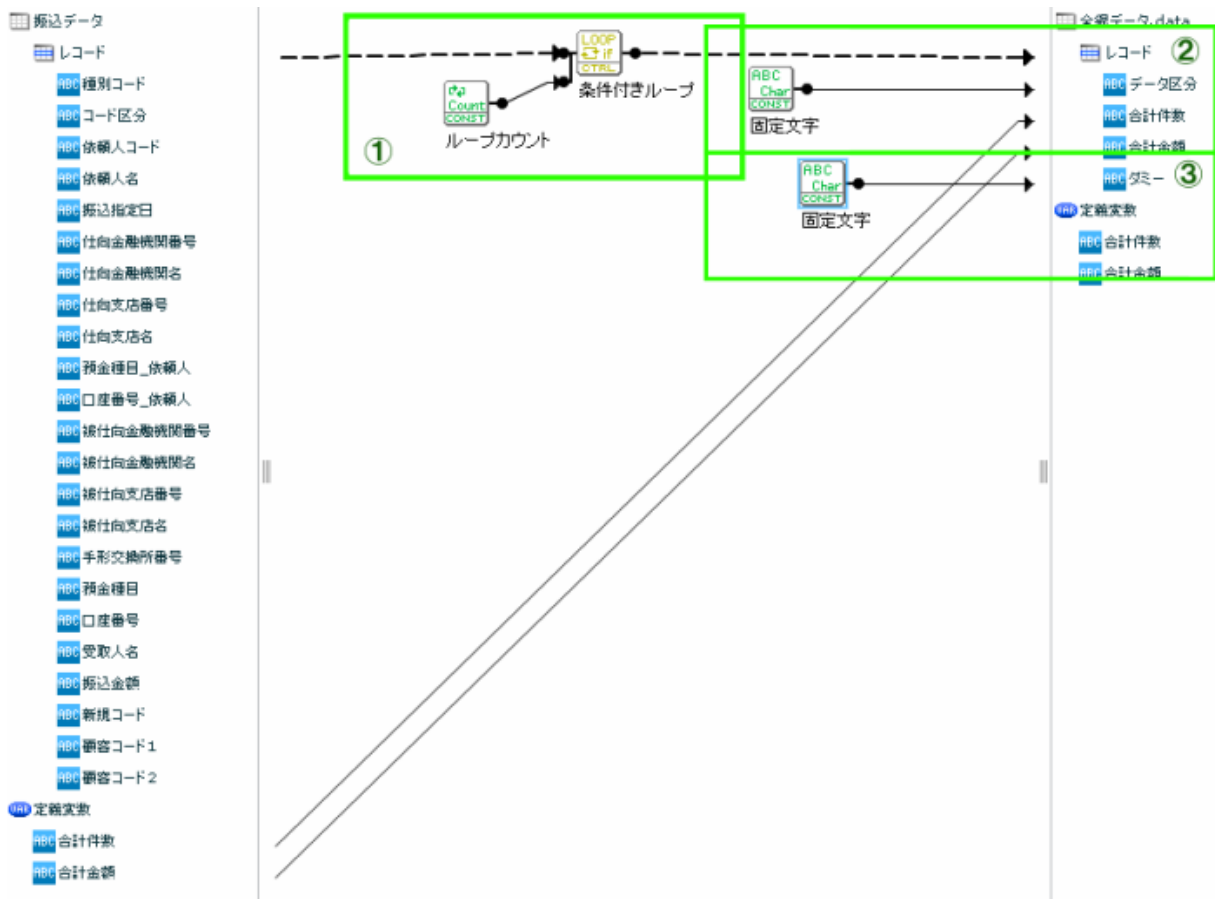


(4)全レコードの合計振込金額を定義変数の「合計金額」に設定します。転送元データベースの「振込金額」を定義変数の「合計金額」とたし合わせて、定義変数の「合計金額」に出力します。「合計金額」のデフォルト値は「0」です。

「合計金額」の値は次のトレーラ出力で出力します。



3. 全銀データ_トレーラ作成



転送元には事前に用意してあるデータベースの「振込データ」を選択しますが、出力するのは直前の定義「データ出力」時に集計された定義変数の「合計件数」と「合計金額」1レコードのみになるため、転送元はどのリソースでも問題はありません。レコード転送先には全銀協の固定長ファイル「全銀データ.data」を指定し、レコードパターンに全銀協のレコードフォーマットの「トレーラ」部分を用意します。

転送元：



サンプルに付属している SQL で作成したテーブルを指定します。

転送先：



転送先に固定長ファイルの「全銀データ.data」を指定します。

転送モードは【追加書き込み】としてください。

各フィールドの設定

【フィールドタイプ】に【文字列】を選択して、【揃え】に【左揃え】、【パディング文字】に【半角スペース】を設定、【フィールド長 (byte)】には指定サイズを設定

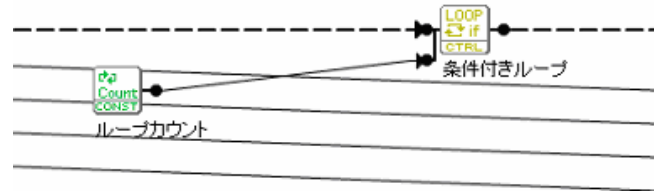
定義変数

「合計件数」と「合計金額」の定義変数を設定します。この定義内で「データ出力」で集計された「合計件数」と「合計金額」を出力します。

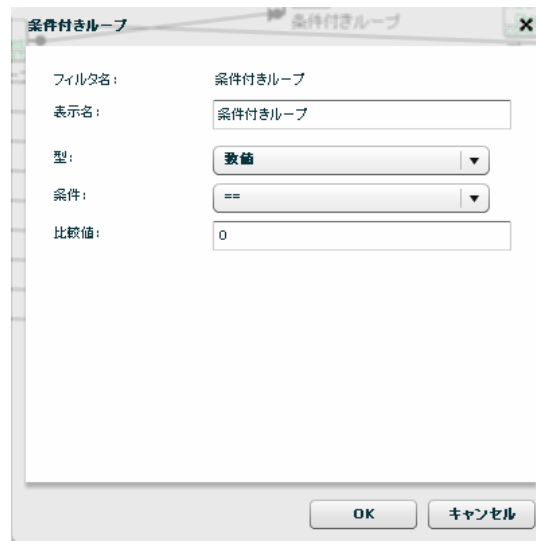
定義変数の「合計件数」と「合計金額」を転送先に繋げます。

(1)ヘッダは1行だけ出力します。データベースのヘッダ情報は全てのレコードで同一です。

「条件付ループ」でループカウントが0の、1レコード目のヘッダ情報のみ出力するようにします。

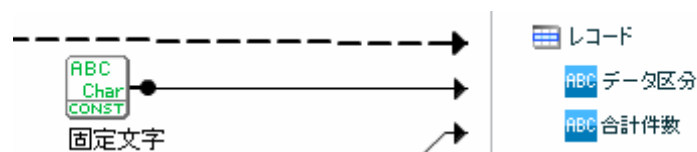


1レコード目のヘッダ情報のみ出力



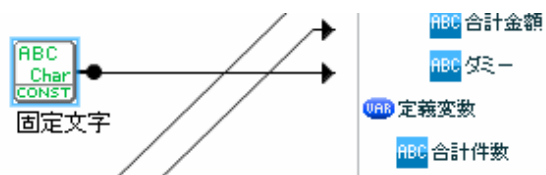
条件付きループで1レコード目のみ出力させる

(2)データ区分は固定で8を出力します。固定長ファイル固有の情報のため、データベースには含まれていないため、マッピングで対応します。



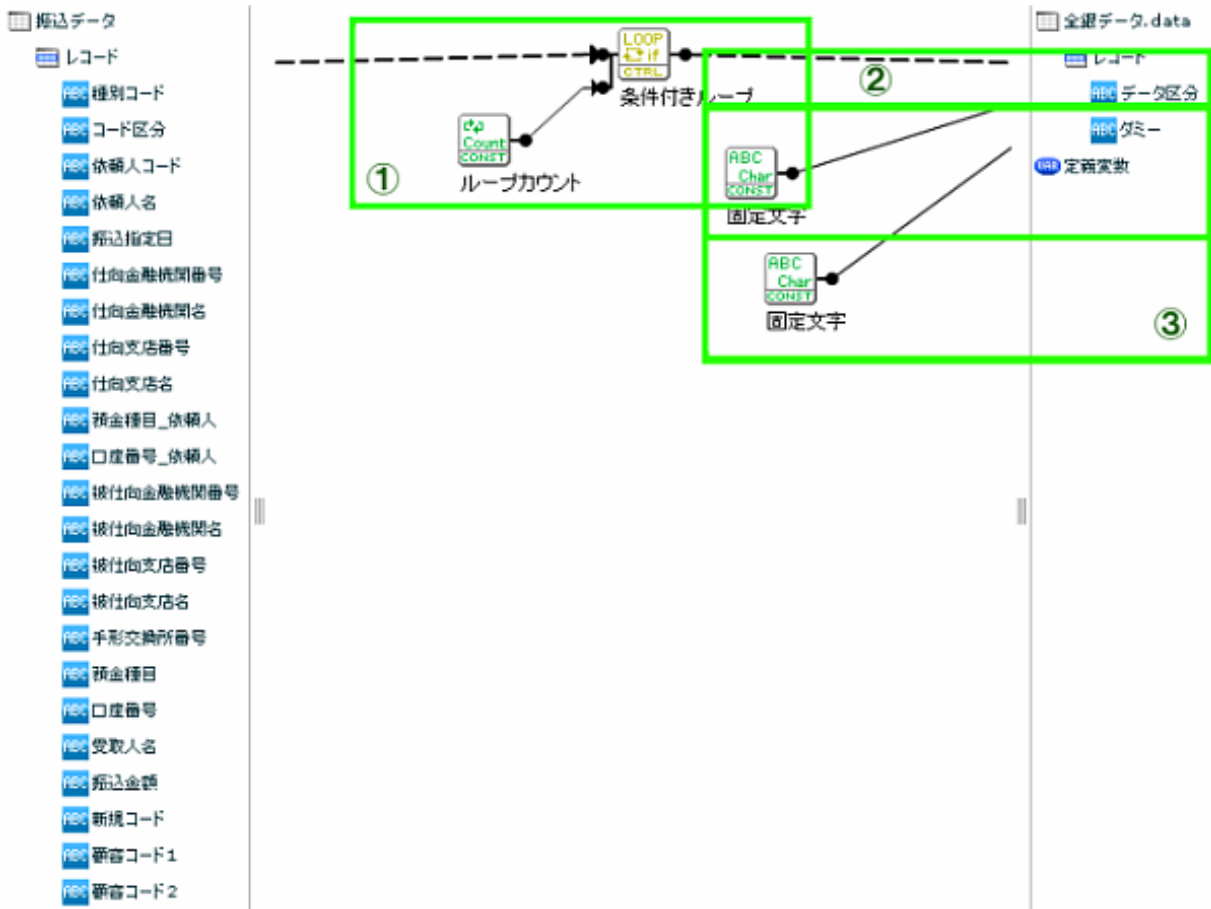
データ区分は固定で8を出力

(3)ダミーに半角スペースを1文字指定します。



半角スペースを固定文字フィルターに設定してダミーに出力

4. 全銀データ_エンド作成



転送元には事前に用意してあるデータベースの「振込データ」を選択しますが、出力するのは「データ区分」への「9」のみになるため、転送元はどのリソースでも問題はありません。レコード転送先には全銀協の固定長ファイル「全銀データ.data」を指定し、レコードパターンに全銀協のレコードフォーマットの「エンド」部分を用意します。

転送元：



サンプルに付属している SQL で作成したテーブルを指定します。

転送先：



転送先に固定長ファイルの「全銀データ.data」を指定します。

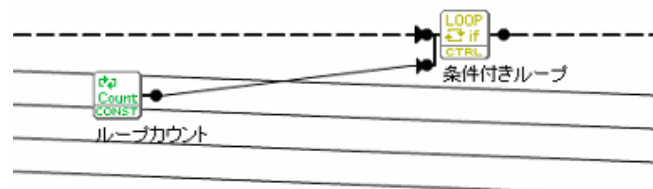
転送モードは【追加書き込み】としてください。

各フィールドの設定

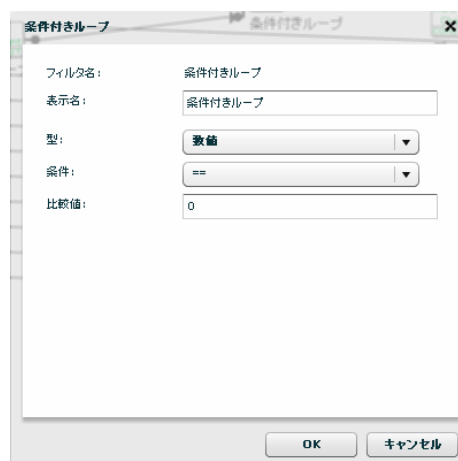
【フィールドタイプ】に【文字列】を選択して、【揃え】に【左揃え】、【パディング文字】に【半角スペース】を設定、【フィールド長 (byte)】には指定サイズを設定

(1)ヘッダは1行だけ出力します。データベースのヘッダ情報は全てのレコードで同一です。

「条件付ループ」でループカウントが0の、1レコード目のヘッダ情報のみ出力するようにします。

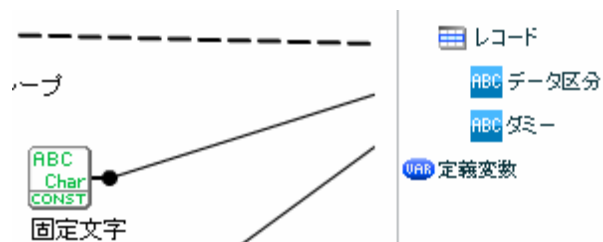


1レコード目のヘッダ情報のみ出力



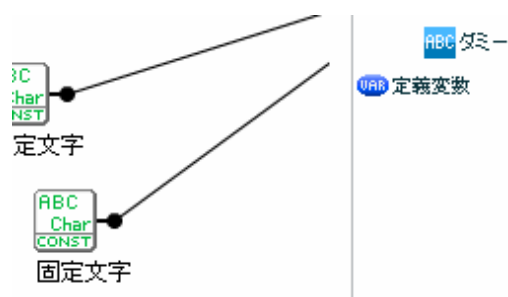
条件付きループで1レコード目のみ出力させる

(2) データ区分は固定で9を出力します。固定長ファイル固有の情報のため、データベースには含まれていないため、マッピングで対応します。



データ区分は固定で9を出力

(3) ダミーに半角スペースを1文字指定します。

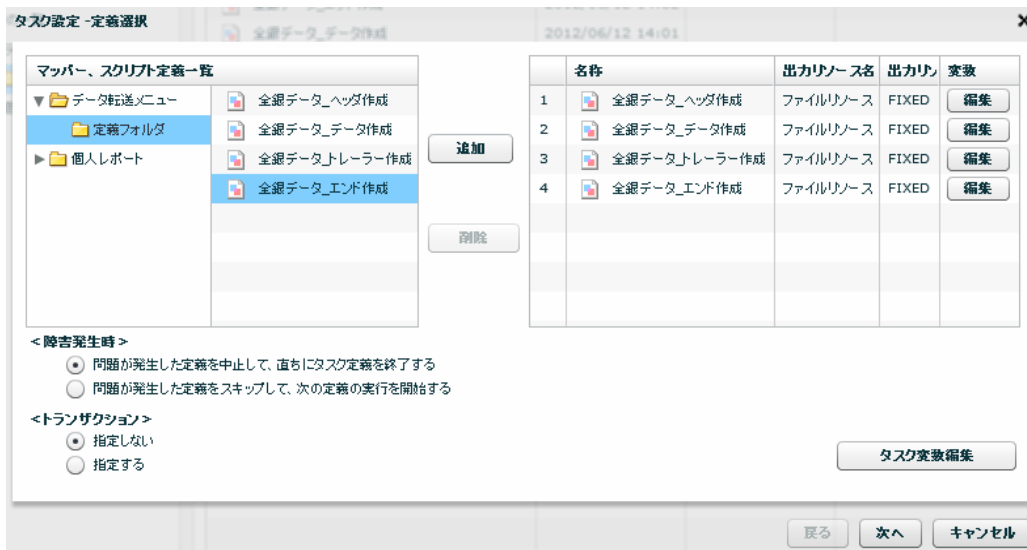


半角スペースを固定文字フィルターに設定してダミーに出力

これらの「全銀データ_ヘッダ作成」、「全銀データ_データ作成」、「全銀データ_トレーラ作成」、「全銀データ_エンド作成」の定義を保存します。

5. タスク設定

【データ転送定義】画面でタスクを新規作成し、【タスク設定-定義選択】画面で以下の設定を行います。



(1) 作成した「全銀データ_ヘッダ作成」、「全銀データ_データ作成」、「全銀データ_トレーラ作成」、「全銀データ_エンド作成」をタスクで順番に設定します。

(2) データ作成時に集計した「合計件数」と「合計金額」をトレーラ作成に使用するために、受け渡し用のタスク変数を設定します。

1	全銀データ_ヘッダ作成	固定長	FIXED	編集
2	全銀データ_データ作成	固定長	FIXED	編集
3	全銀データ_トレーラ作成	固定長	FIXED	編集
4	全銀データ_エンド作成	固定長	FIXED	編集

合計件数
合計金額

【タスク変数編集】をクリックして、タスク変数編集画面で「合計件数」、「合計金額」を追加して【OK】を押します。値は0を設定します。



(3) 「全銀データ_データ作成」で集計した「合計件数」と「合計金額」をタスク変数に渡します。

	名称	出力リソース名	出力リソース	変数
1	 全銀データ_ヘッダ作成	固定長	FIXED	<input type="button" value="編集"/>
2	 全銀データ_データ作成	固定長	FIXED	<input type="button" value="編集"/>
3	 全銀データ_トレーラー作成	固定長	FIXED	<input type="button" value="編集"/>
4	 全銀データ_エンド作成	固定長	FIXED	<input type="button" value="編集"/>

「全銀データ_データ作成」の【変数】-【編集】をクリックします。

定義実行後に「全銀データ_データ作成」の集計されたマッパー変数「合計件数」、「合計金額」の値をそれぞれ、先ほど設定したタスク変数に渡します。

変数受渡し

■定義実行前		■定義実行後	
タスク変数	マッパー変数	マッパー変数	タスク変数
<input type="button" value="指定なし"/> ▼	合計件数	<input type="button" value="合計件数"/> ▼	合計件数
<input type="button" value="指定なし"/> ▼	合計金額	<input type="button" value="合計金額"/> ▼	合計金額

※マッパー変数に渡すタスク変数を選んでください ※タスク変数に戻すマッパー変数を選んでください

マッパーで設定された値をタスクに戻し、【OK】を押します。

(4) タスク変数の値を「全銀データ_トレーラ作成」の「合計件数」と「合計金額」に渡します。

	名称	出力リソース名	出力リソース	変数
1	 全銀データ_ヘッダ作成	固定長	FIXED	<input type="button" value="編集"/>
2	 全銀データ_データ作成	固定長	FIXED	<input type="button" value="編集"/>
3	 全銀データ_トレーラ作成	固定長	FIXED	<input type="button" value="編集"/>
4	 全銀データ_エンド作成	固定長	FIXED	<input type="button" value="編集"/>

「全銀データ_トレーラ作成」の【変数】-【編集】をクリックします。

定義実行前にタスク変数の値を「全銀データ_トレーラ作成」マッパー変数「合計件数」、「合計金額」に渡します。

変数受渡し

■定義実行前		■定義実行後	
タスク変数	マッパー変数	マッパー変数	タスク変数
<input type="button" value="合計件数"/> ▼	合計件数	<input type="button" value="指定なし"/> ▼	合計件数
<input type="button" value="合計金額"/> ▼	合計金額	<input type="button" value="指定なし"/> ▼	合計金額

※マッパー変数に渡すタスク変数を選んでください ※タスク変数に戻すマッパー変数を選んでください

タスクに保存した値をマッパーに設定し、【OK】を押します。

【次へ】を押して、【タスク設定-保存先の指定】画面でタスクを保存して終了です。

タスクを実行し、「全銀データ.data」が全銀協フォーマットの固定長ファイルとして出力されていることを確認してください。

2-7:作成したファイルを送信する

この章では本製品で作成したファイルを送信する次の手順

1. 売上げ情報 EXCEL ファイルを読み込み、地区コードごとに別の EXCEL ファイルに出力する
2. 作成されたファイルを、地区の担当者に送信する

について説明します。

作成する定義

- ・ 1. 売上情報地区別出力定義
- ・ 2. 売上情報地区別送信定義

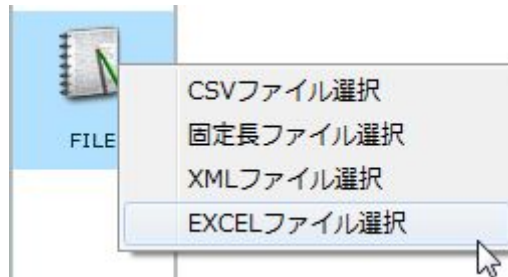
使用するデータ

- ・ 売上データ.xls 「1. 売上情報地区別出力定義」の定義作成時読み込みファイルとして使用します。
- ・ 売上データテンプレート 「1. 売上情報地区別出力定義」の定義作成時、テンプレートファイルとして使用します。
- ・ 社員マスター.csv 「2. 売上情報地区別送信定義」の定義作成時、読み込みファイルとして使用します。

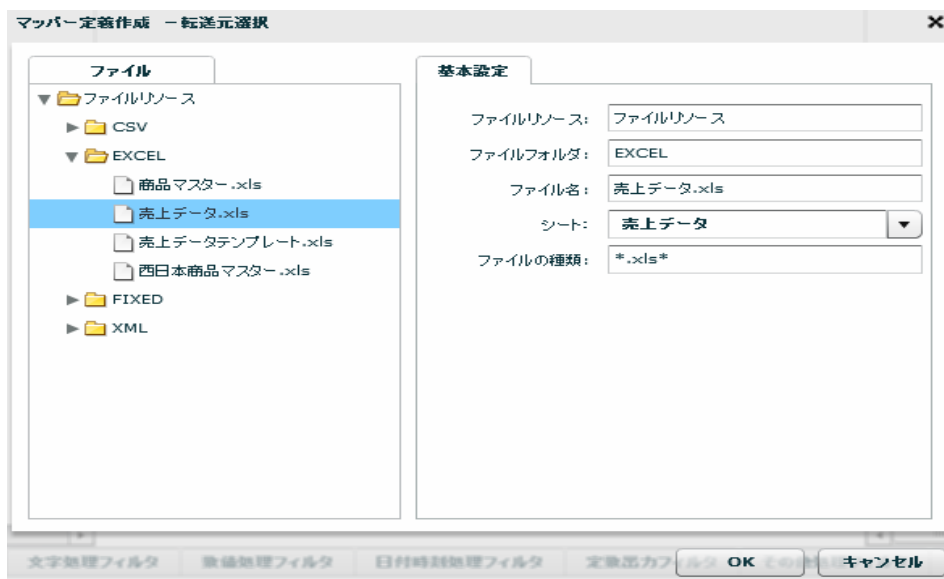
売上げ情報を読み取り、地区コードごとに違う EXCEL ファイルに出力する

1. 売上げ情報の読み取り

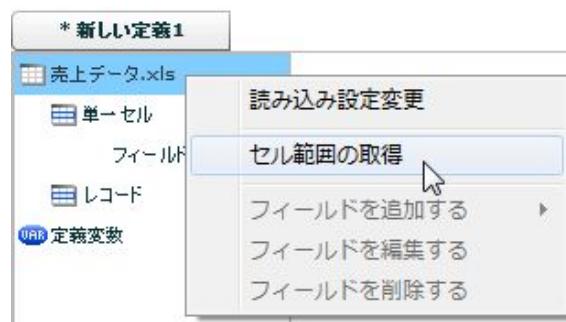
転送元は EXCEL ファイルを指定します。[FILE]アイコン上で右クリックメニュー「EXCEL ファイル選択」をクリックします。



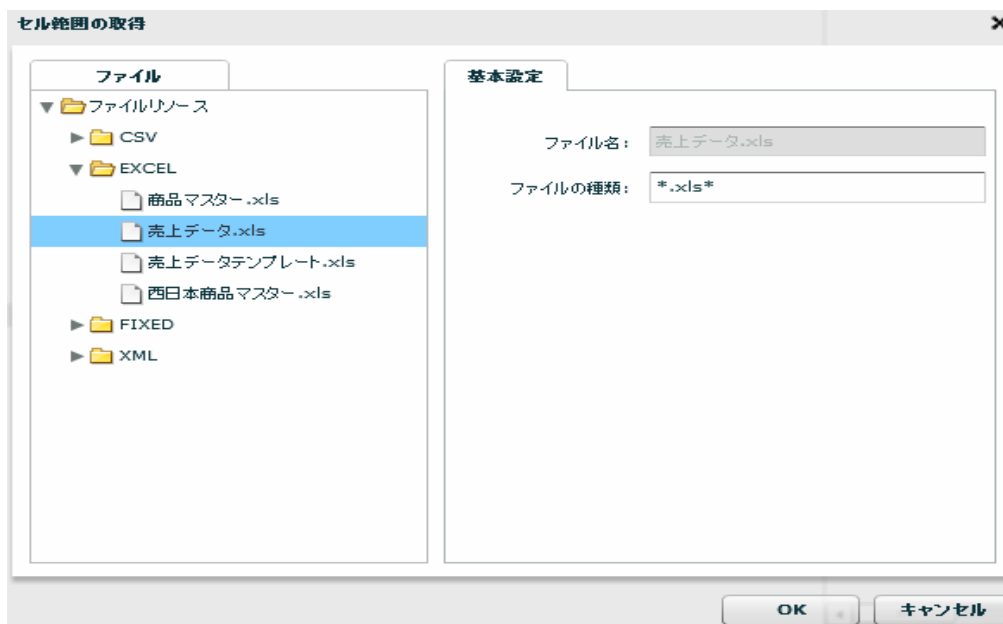
「転送元選択画面」が表示されます。転送元には EXCEL ファイル「売上データ.xls」を指定します。シートは「売上データ」を選択し、[OK]ボタンをクリックします。



入力元に「単一セル」フィールドと「レコード」フィールドが表示されます。入力元のファイル名を選択状態にして、右クリックメニューの「セル範囲の取得」をクリックします。



「セル範囲の取得」画面が表示されます。転送元ファイルと同様に、「売上データ.xls」を指定し、[OK] ボタンをクリックします。



「ダウンロードが完了しました。」というメッセージが表示されるので、[OK]ボタンをクリックします。本製品サーバーよりダウンロードした EXCEL ファイルと、「QanatExcel」が表示されます。

年度	四半期	担当者コード	担当者名	地区コード	担当地区	商品コード	商品名	数量	売り上げ金額	粗利益	定価	仕入れ単
2003	1	1	山田太郎	1	横浜	1	ラーメン	30	15000	5000	500	1
2002	2	1	山田太郎	1	横浜	2	ラーメン	10	20000	3000	2000	10
2004	1	2	佐藤次郎	2	東京	3	寿司	20	30000	10000	1500	5
2008	4	3	鈴木三郎	3	大阪	4	たこ焼き	20	20000	5000	1000	4
2002	3	1	山田太郎	1	横浜	2	目玉焼き	5	10000	1500	2000	10
2003	3	3	鈴木三郎	3	大阪	4	焼きそば	10	10000	2500	1000	4

単一セルの設定を行います。単一セルとして扱うのは資料作成日の値が入力されているセル「D2」と資料作成者の値が入力されている「D3」となります。それぞれのセルを選択状態にして、「QanatExcel」の[追加]ボタンをクリックしてください。単一セルの名称に以下の2つの項目が表示されている状態となります。

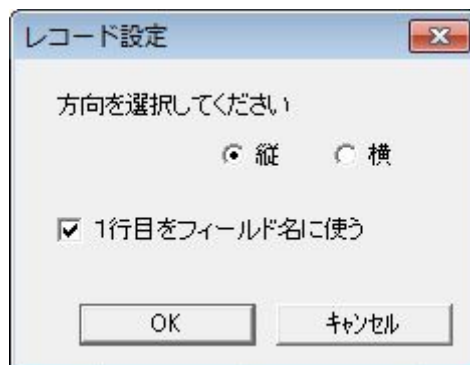
A	B	C	D	E	F	G	H
			資料作成日				
			資料作成者				
			2010/1/1				
			山田太郎				
年度	四半期	担当者コード	担当者名	地区コード	担当地区	商品コード	商品名
2003	1	1	山田太郎	1	横浜	1	ラーメン
2002	2	1	山田太郎	1	横浜	2	ラーメン
2004	1	2	佐藤次郎	2	東京	3	寿司
2008	4	3	鈴木三郎	3	大阪	4	たこ焼
2002	3	1	山田太郎	1	横浜	2	目玉焼
2003	3	3	鈴木三郎	3	大阪	4	焼そば

「名称」は定義内でフィールド名として表示されます。このままでも処理に問題はありませんが、分かりやすさのため、「名称」を変更します。「単一セル 0」をダブルクリックすると、入力モードに切り替わります。このセルの値は「資料作成日」なので、名称を「資料作成日」に変更します。同様に、「単一セル 1」を「資料作成者」に変更します。

続いて、レコードの設定を行います。レコードとして扱うのは売上げ情報が入力されているセル「B6」から「N12」までの範囲となります。セル「B6」からセル「N12」までをドラッグし、選択状態にした後、「QanatExcel」の[設定]ボタンをクリックしてください。

A	B	C	D	E	F	G	H	I	J	K	L	M	N
	資料作成日		2010/1/1										
	資料作成者		山田太郎										
年度	四半期	担当者コード	担当者名	地区コード	担当地区	商品コード	商品名	数量	売り上げ金額	粗利益	定価	仕入れ単価	
2003	1	1	山田太郎	1	横浜	1	ラーメン	30	15000	5000	500	150	
2002	2	1	山田太郎	1	横浜	2	ラーメン	10	20000	3000	2000	1000	
2004	1	2	佐藤次郎	2	東京	3	寿司	20	30000	10000	1500	500	
2008	4	3	鈴木三郎	3	大阪	4	たこ焼き	20	20000	5000	1000	400	
2002	3	1	山田太郎	1	横浜	2	目玉焼き	5	10000	1500	2000	1000	
2003	3	3	鈴木三郎	3	大阪	4	焼きそば	10	10000	2500	1000	400	

「レコード設定」画面が表示されます。今回は、行をレコードとして扱う為、方向をデフォルトの設定である「縦」に設定します。また、1行目にフィールド名が記述されているため、「1行目をフィールド名に使う」にチェックを入れ、[OK]ボタンをクリックしてください。



レコードの項目に、以下の項目が表示されます。確認した後[OK]ボタンをクリックします。「QanatExcel」が終了し、入力元に設定したフィールド情報が反映されます。

名称	セル位置	名称	セル位置
年度	B	商品名	I
四半期	C	数量	J
担当者コード	D	売り上げ金額	K
担当者名	E	粗利益	L
地区コード	F	定価	M
担当地区	G	仕入れ定価	N
商品コード	H		

引き続き、転送先の設定を行います。

2. 売上げ情報を担当地区コード毎に別ファイルに出力する

転送先は、レコードの担当地区の値によって出力するファイルを変更する処理を行います。出力ファイルを動的に変更する為に、定義変数の設定を行います。入力元の「定義変数」フィールドを選択状態にして、右クリックメニュー「定義変数を追加する」をクリックします。



「定義変数のセット」画面が表示されます。「名前」に「転送先ファイル名」と入力し、[OK]ボタンをクリックします。

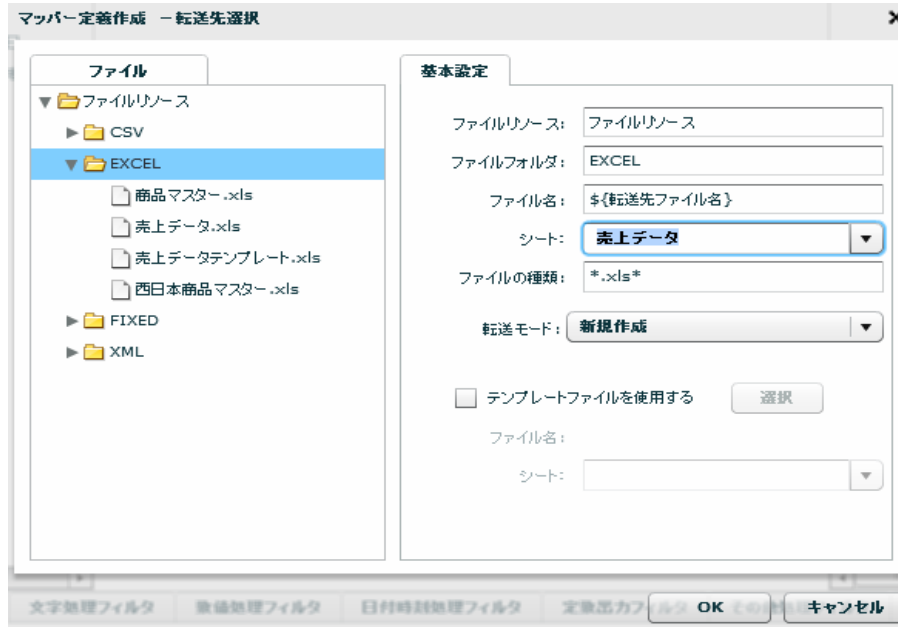


「定義変数」フィールドの下に「出力ファイル名」フィールドが追加されます。

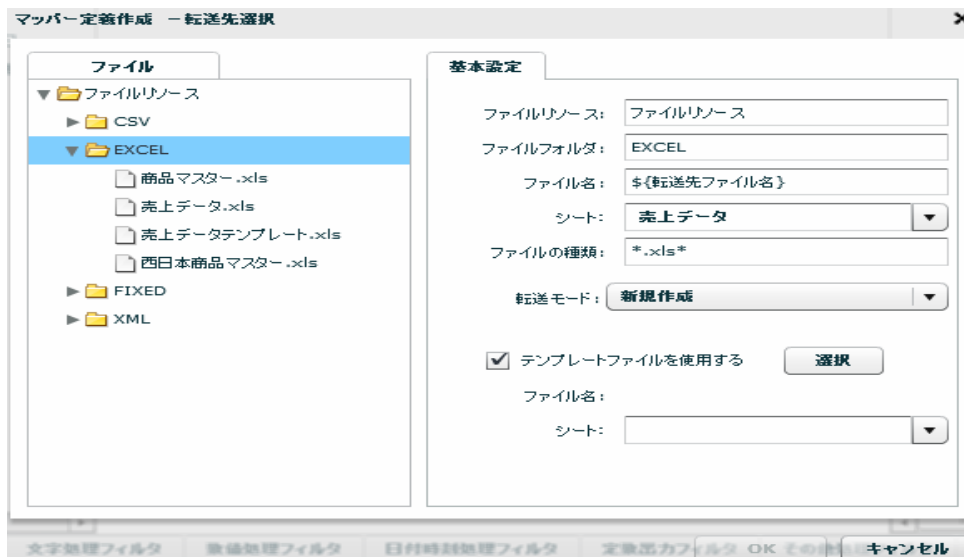
続いて、転送先ファイル名を設定します。画面右の[FILE]アイコン上で右クリックメニュー「EXCELファイル選択」をクリックします。



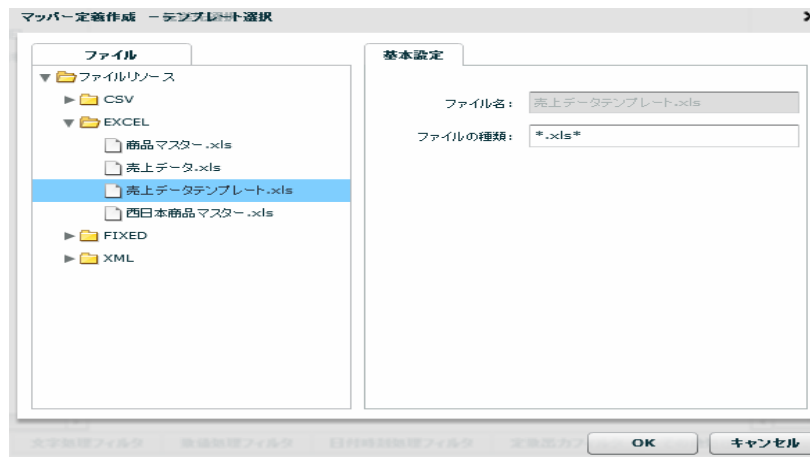
「転送先選択」画面が表示されます。画面左のフォルダから、フォルダの2階層目である「ファイルフォルダー」を選択し、画面右の「ファイル名」に値を入力します。出力ファイル名は担当地区によって変更するため、先ほど作成した定義変数「転送先ファイル名」を指定します。定義変数の指定は\${[定義変数名]}で行います。ファイル名に定義変数を指定しているため、「シート」を手動入力する必要があります。シート名は「売上げデータ」と入力します。転送モードはデフォルトの「新規作成」のまま設定してください。



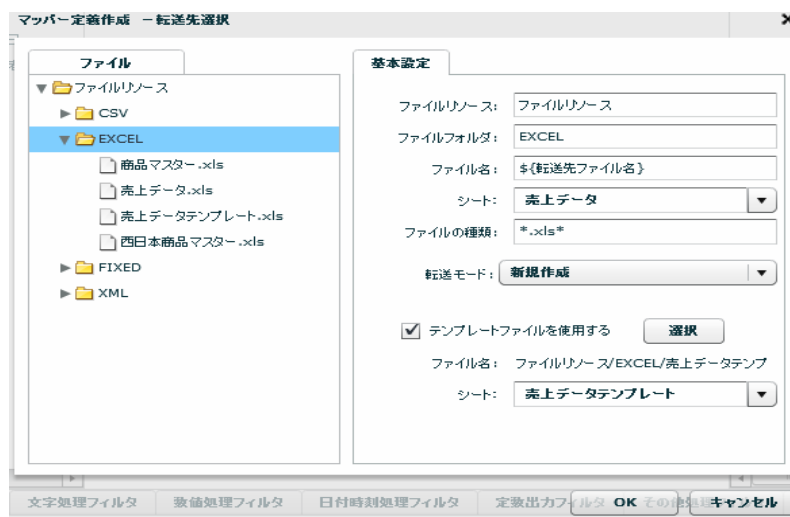
このまま出力しても問題はありませんが、今回はテンプレートを使用して、整形された EXCEL ファイルに出力するように設定します。「テンプレートファイルを使用する」にチェックを入れ、[選択] ボタンをクリックしてください。



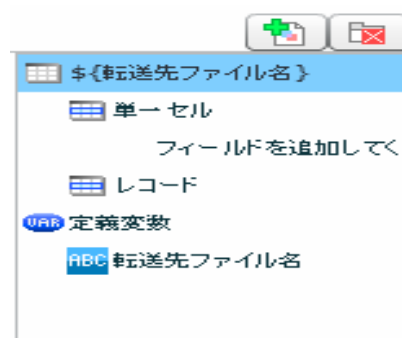
「テンプレート選択」画面が表示されます。テンプレート用のファイルである「売上データテンプレート.xls」を選択し、[OK]ボタンをクリックしてください。



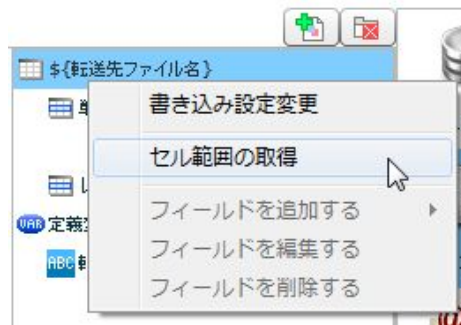
「テンプレート選択」画面が表示されます。「シート」は「売上データテンプレート」が自動で選択されていることを確認してください。確認の後、[OK]ボタンをクリックしてください。



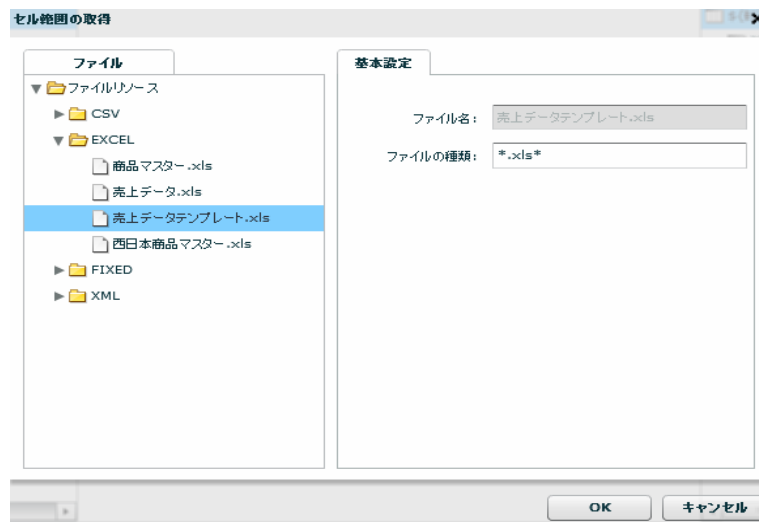
転送先に「単一セル」フィールドと「レコード」フィールドが表示されます。



続いて、転送元ファイル選択時と同様に、書き込みを行うセルを指定します。転送先上で右クリックメニュー「セル範囲の取得」をクリックします。



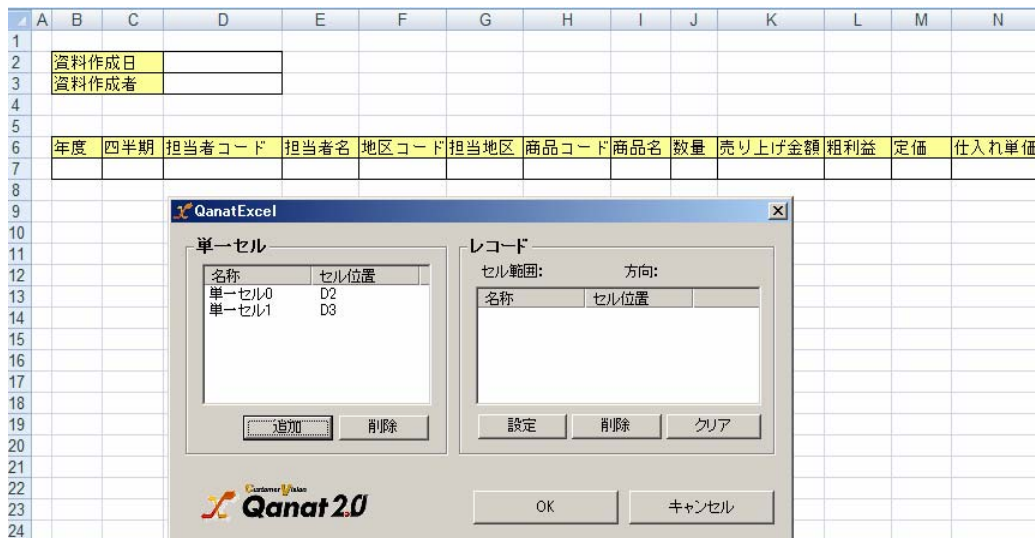
「テンプレート選択」画面が表示されるので、前の作業でテンプレートに指定した「売上データテンプレート」を指定して、[OK]ボタンをクリックしてください。



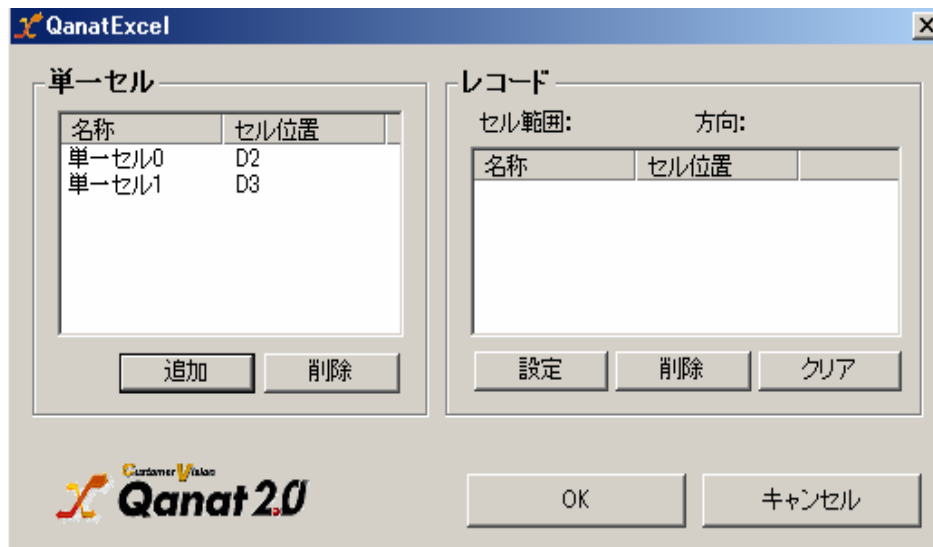
「ダウンロードが完了しました。」というメッセージが表示されるので、[OK]ボタンをクリックします。本製品サーバーよりダウンロードした EXCEL ファイルと、「QanatExcel」が表示されます。

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1														
2		資料作成日												
3		資料作成者												
4														
5														
6		年度	四半期	担当者コード	担当者名	地区コード	担当地区	商品コード	商品名	数量	売り上げ金額	粗利益	定価	仕入れ単価
7														
8														
9														
10														
11														
12														
13														
14														
15														
16														
17														
18														
19														
20														
21														
22														
23														
24														

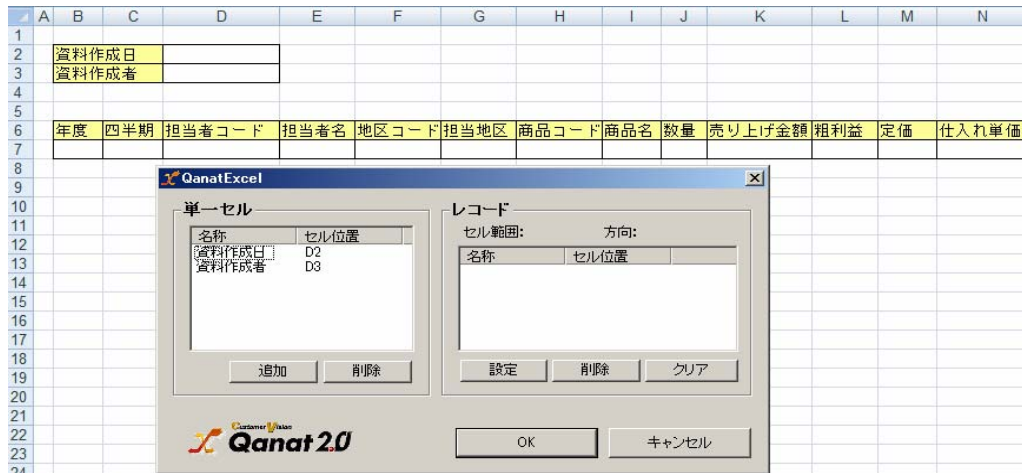
単一セルの設定を行います。単一セルとして扱うのは資料作成日の値が入力されているセル「D2」と資料作成者の値が入力されている「D3」となります。それぞれのセルを選択状態にして、「QanatExcel」の[追加]ボタンをクリックしてください。単一セルの名称に以下の2つの項目が表示されている状態となります。



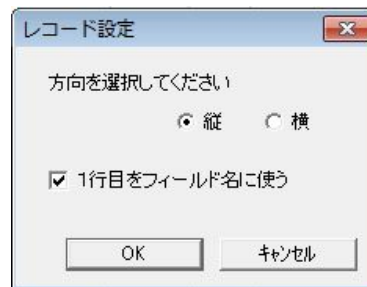
「名称」は定義内でフィールド名として表示されます。このままでも処理に問題はありますが、わかりやすくするため、「名称」を変更します。「単一セル0」をダブルクリックすると、入力モードに切り替わります。このセルの値は「資料作成日」なので、名称を「資料作成日」に変更します。同様に、「単一セル1」を「資料作成者」に変更します。



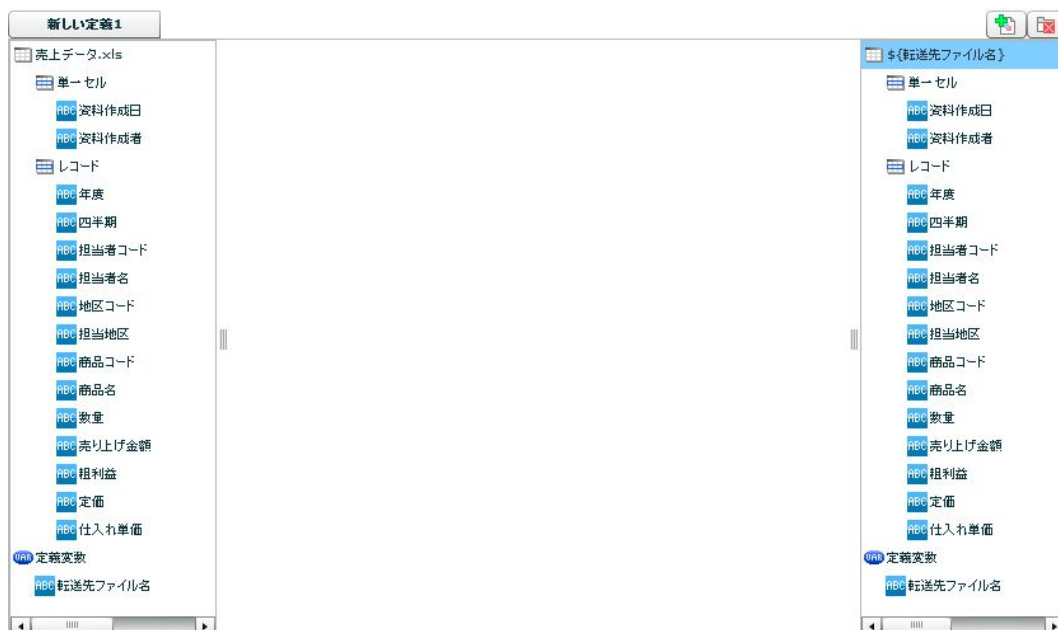
続いて、レコードの設定を行います。今回の書き込みの場合、担当地区によって書き込まれるレコード数が異なる為、一様にレコード範囲を指定することができません。そのため、「指定されたセル範囲を超えても書き込む」設定を利用して書き込みを行います。レコードとして扱うのは売上げ情報が入力されているセル「B6」から「N7」までの範囲となります。セル「B6」からセル「N7」までをドラッグし、選択状態にした後、「QanatExcel」の[設定]ボタンをクリックしてください。



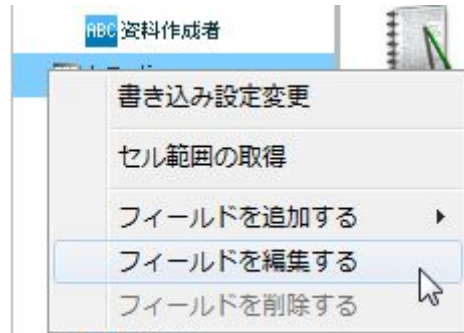
「レコード設定」画面が表示されます。今回は、行をレコードとして扱う為、方向をデフォルトの設定である「縦」に設定します。また、1行目にフィールド名が記述されているため、「1行目をフィールド名に使う」にチェックを入れ、[OK]ボタンをクリックしてください。



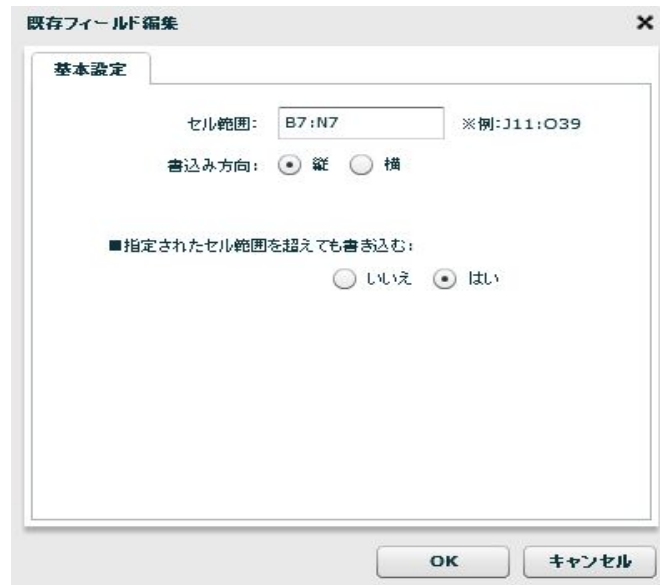
レコードの項目に、以下の項目が表示されます。確認した後[OK]ボタンをクリックします。「QanatExcel」が終了し、入力元に設定したフィールド情報が反映されます。



転送先の「レコード」フィールドを選択状態にして、右クリックメニュー「フィールドを編集する」をクリックします。

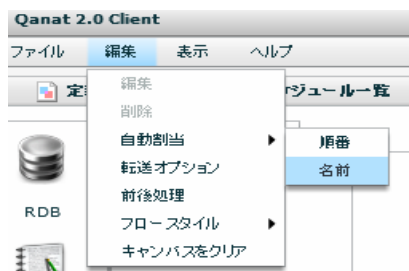


「セル範囲」「書き込み方向」に関しては変更の必要はありません。「指定されたセル範囲を超えても書き込む」の設定がデフォルトでは「いいえ」に選択されているので、「はい」に変更し、[OK]ボタンをクリックします。「設定したレコード範囲で、フィールドを再設定しますか？」というメッセージが表示されますが、再設定の必要はないため、[いいえ]ボタンをクリックしてください。

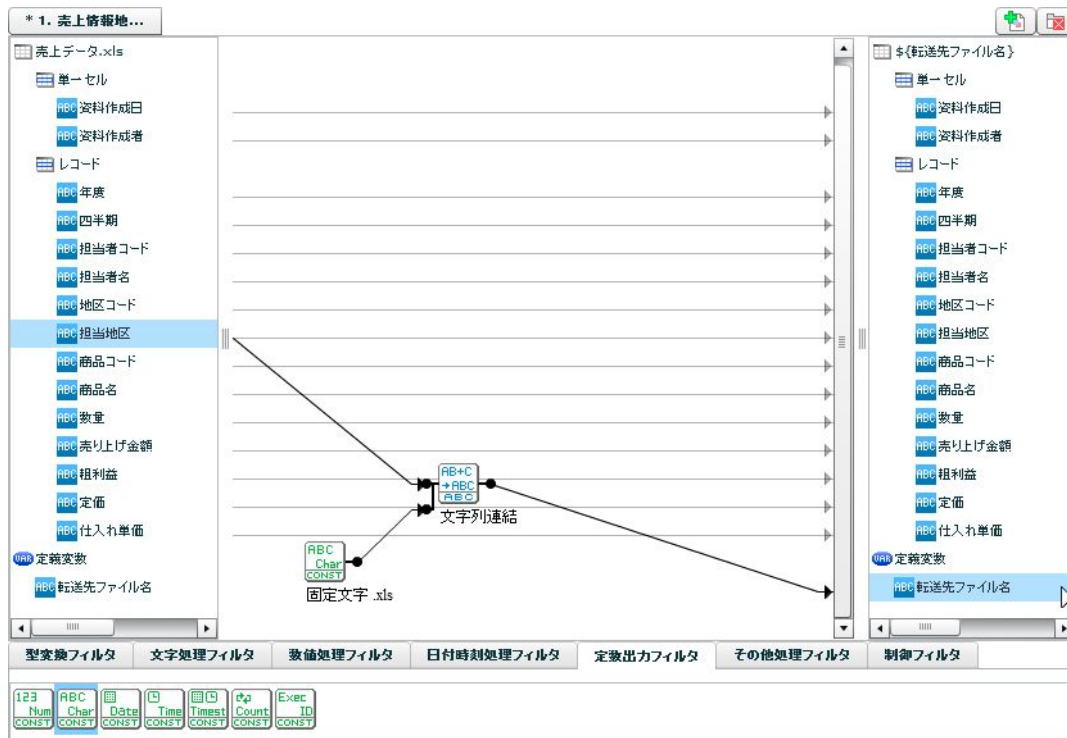


ここまでで転送元、転送先の設定が完了しました。続いて、結線を行います。

ツールメニューの[編集]-[自動割当]-[名前]をクリックし、結線を行ってください。



名前が一致したフィールドが結線されます。続いて、転送ファイル名を担当地区にするので、転送元の「担当地区」フィールドを転送先の定義変数「転送ファイル名」に結線します。担当地区には拡張子が付いていないため、フィルタ[文字列フィルタ]-[文字列連結] と [定数出力フィルタ]-[固定文字]を利用して拡張子を付加します。



以上で売上げ情報を担当地区コード毎に別ファイルに出力する処理が完成しました。ツールメニューの[ファイル]-[名前をつけて保存]をクリックし、定義を「1. 売上情報地区別出力定義」という名前で保存してください。保存後、ツールメニューの[ファイル]-[即時実行]から定義を実行し、以下の3ファイルが作成されることを確認してください。それぞれ出力されているレコード数は異なりますが、「指定されたセル範囲を超えても書き込む」が有効になっているため、テンプレートを使用して出力されていることを確認してください。引き続き、送信定義の作成を行います。

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1														
2		資料作成日		2012/6/12										
3		資料作成者		山田太郎										
4														
5														
6		年度	四半期	担当者コード	担当者名	地区コード	担当地区	商品コード	商品名	数量	売り上げ金額	粗利益	定価	仕入れ単価
7		2002	2	2	佐藤次郎	2	東京	3	寿司	20	30000	10000	1500	500
8														
9														
10														
11														
12														

東京.xls

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1														
2		資料作成日		2012/6/12										
3		資料作成者		山田太郎										
4														
5														
6		年度	四半期	担当者コード	担当者名	地区コード	担当地区	商品コード	商品名	数量	売り上げ金額	粗利益	定価	仕入れ単価
7		2003	1	1	山田太郎	1	横浜	1	目玉焼き	30	15000	5000	500	150
8		2002	2	1	山田太郎	1	横浜	2	ラーメン	10	20000	3000	2000	1000
9		2002	3	1	山田太郎	1	横浜	2	ラーメン	5	10000	1500	2000	1000
10														

横浜.xls

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1														
2		資料作成日		2012/6/12										
3		資料作成者		山田太郎										
4														
5														
6		年度	四半期	担当者コード	担当者名	地区コード	担当地区	商品コード	商品名	数量	売り上げ金額	粗利益	定価	仕入れ単価
7		2008	4	3	山田太郎	1	大阪	4	焼そば	20	20000	5000	1000	400
8		2003	3	3	山田太郎	1	大阪	4	たこ焼き	10	10000	2500	1000	400

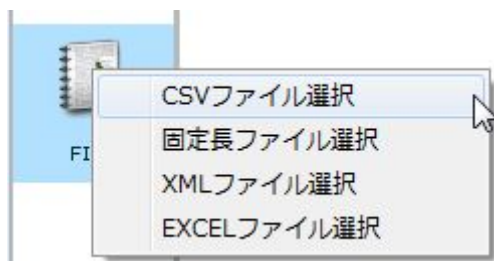
大阪.xls

3. 送信定義の作成

「1. 売上情報地区別出力定義」で作成されたファイルをそれぞれの地域の担当者に送信する定義を作成します。ツールメニューの[ファイル]-[新規作成]-[マッパー定義]をクリックしてください。



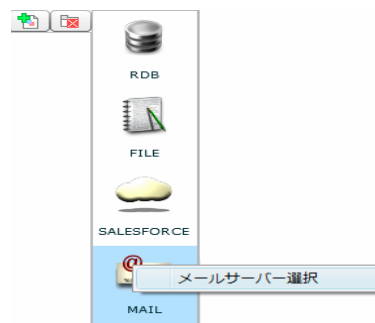
「新しい定義1」タブが表示されます。転送元の設定を行います。画面左の「FILE」アイコン上で右クリック「CSVファイル選択」をクリックしてください。



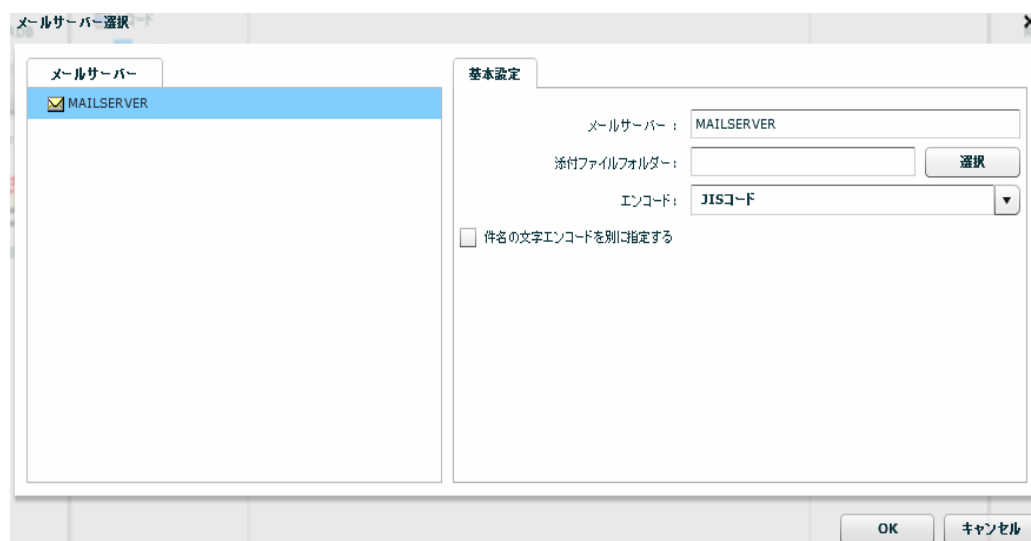
「転送元選択」画面が表示されます。「社員マスター.csv」を選択してください。ファイルの1行目に見出し行が設定されているので、「ファイルの1行目を見出し行として扱う」にチェックを入れた後、[OK]ボタンをクリックしてください。



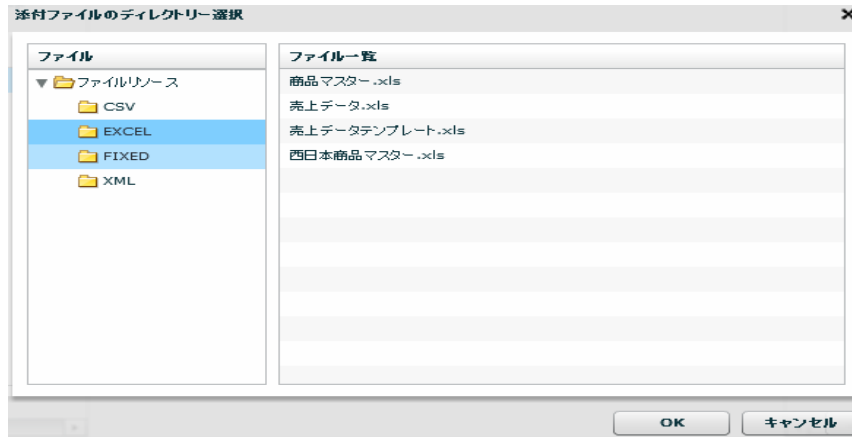
転送元にフィールド情報が表示されます。続いて、転送先の設定を行います。画面右の「MAIL」アイコン上で右クリック「メールサーバー選択」をクリックします。



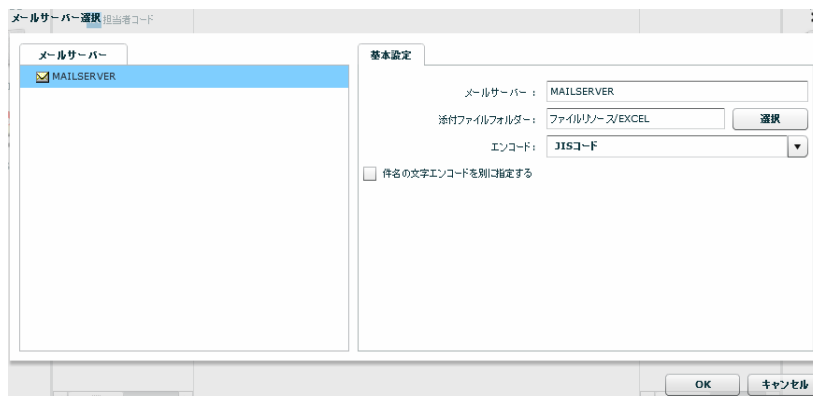
メールサーバーを選択します。今回はファイルを添付するため、[選択]ボタンをクリックし、添付するファイルが存在するファイルフォルダーを指定します。



「添付ファイルのディレクトリ選択」画面が表示されます。「1. 売上情報地区別出力定義」で出力先に指定したファイルフォルダーを選択してください。選択すると、画面右にファイルフォルダーに含まれるファイル一覧が表示されます。確認した後、[OK]ボタンをクリックしてください。

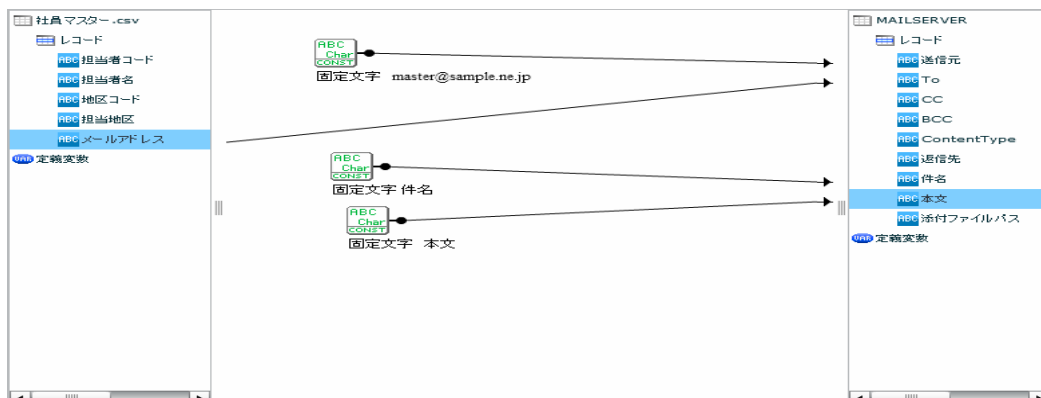


「メールサーバー選択」画面に戻ります。「添付ファイルフォルダー」に選択したファイルフォルダーの名前が表示されていることを確認した後、[OK]ボタンをクリックしてください。

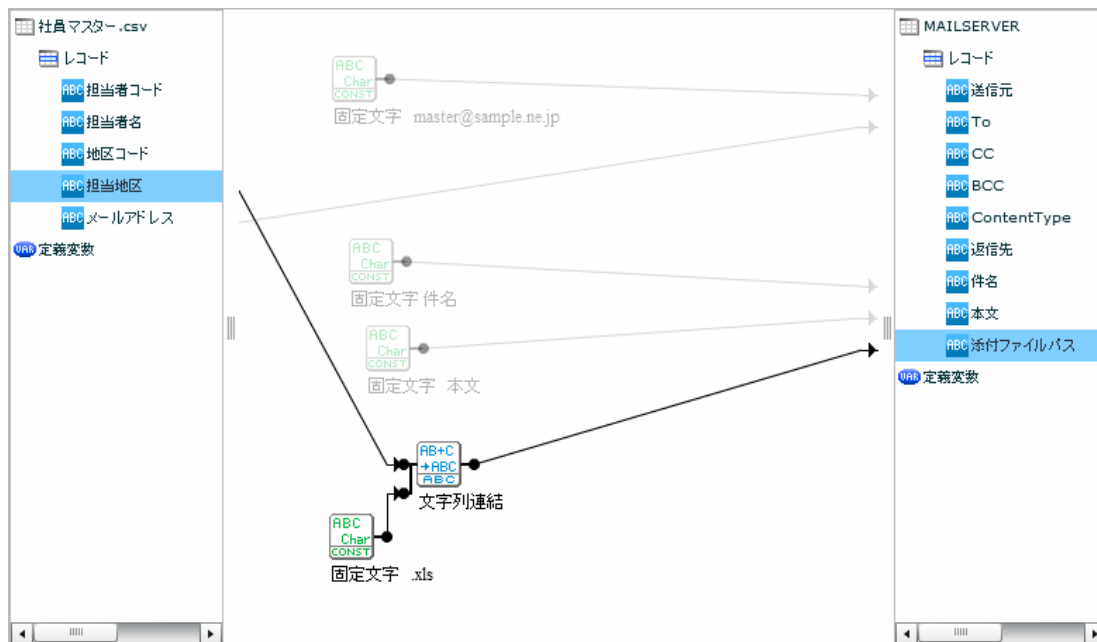


転送先にフィールドが表示されます。続いてデータマッピングを行います。

メール送信に必要な設定値は「送信元」フィールドと「To」フィールドになります。「送信元」フィールドは[固定文字]フィルタから結線します。「To」フィールドは読み込みを行った「メールアドレス」フィールドと結線します。メールの確認のため、「件名」フィールドと「本文」フィールドにもそれぞれ[固定文字]フィルタから結線します。



添付するファイルを担当地区の値によって変更するため、「添付ファイル」のフィールドは、転送元のフィールド「担当地区」の値に拡張子「.xls」を連結させた値を結線します。

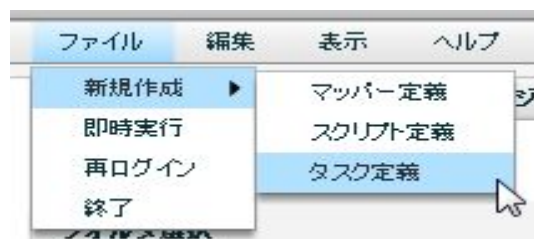


以上でメール送信定義の作成は終了となります。定義を「2. 売上情報地区別送信定義」という名前で保存してください。保存した後、ツールメニューの[ファイル]-[即時実行]から定義を実行し、各メールアドレスに担当地区のファイルが添付されていることを確認してください。

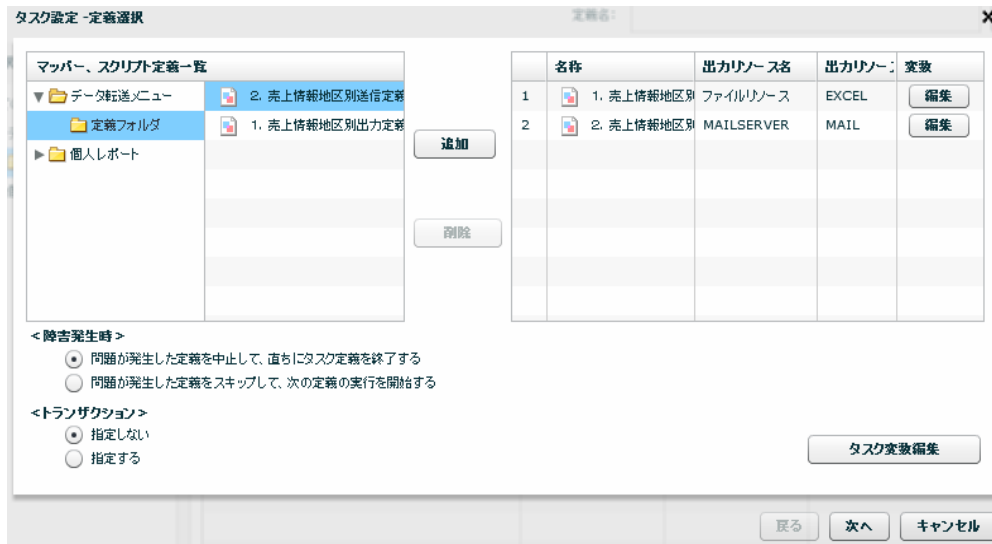
これで、必要な定義の作成が終了しました。最後にこの定義を「ファイル作成 メール送信」を行うタスク定義として登録します。

4. タスク定義の作成

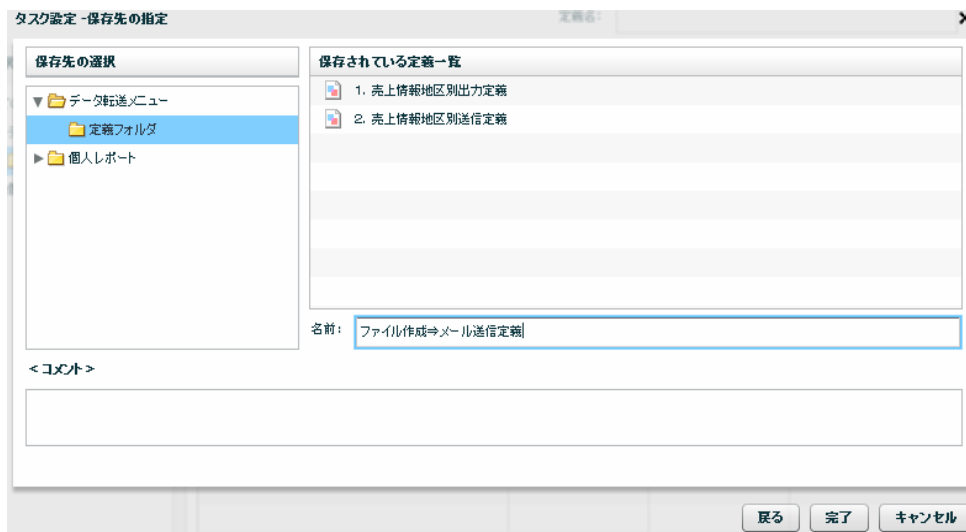
作成したマッパー定義を使用して、タスク定義を作成します。「データ転送定義」タブを表示してください。画面右からタスク定義を作成するフォルダを選択し、ツールメニューの[ファイル]-[新規作成]-[タスク定義]を選択してください。



「定義選択」画面が表示されます。先ほど作成した「1. 売上情報地区別出力定義」、「2. 売上情報地区別送信定義」を選択し、[追加]ボタンをクリックしてください。画面右のリストに定義が追加されたことを確認してください。その他の項目はデフォルト値のまま、[次へ]ボタンをクリックしてください。



「保存先の指定」画面が表示されます。タスク定義の保存フォルダを選択します。タスク名は「ファイル作成 メール送信定義」として、[完了]ボタンをクリックしてください。



以上でタスク定義の作成は終了です。「定義一覧」タブから、作成したタスク定義を実行し、正常に終了することを確認してください。

2-8:Web サービスを実行する

この章では本製品で Web サービスを実行する手順をご紹介します。

今回は SugarCRM の login サービス、getEntryList サービス、logout サービスを順に実行し、SugarCRM から取得した顧客情報をスクリプト定義で顧客マスター形式にカスタマイズし、csv ファイルにおとします。

最後にできあがった csv ファイルで DB の取引先テーブルを更新します。

今回の手順は以下のようになります。

- 1 . Web サービスリクエスト定義で SugarCRM の login、getEntryList、logout それぞれ 3 つの転送定義を作成します。
- 2 . 1 で作成した 3 つの転送定義を順に実行するタスク定義を作成し、login で得た sessionID をタスク変数に渡し、getEntryList,logout でその sessionID を使用できるように設計します。
- 3 . タスク定義を実行してできた sugarCRMgetEntryList 顧客リスト.csv を顧客マスター形式に変換するスクリプト定義を作成します。

(本製品で多重 Array 形式のデータを取得する場合、ループの基準としたい要素をシンプルループフィルタで結線することで実現します。今回の場合 module_name=Accounts を基準にループさせるため、出力させるデータは以下のように module_name 配下が繰り返し取得されたデータになります。)

本チュートリアルでは取得した情報を顧客 (Accounts) マスターとして使用したい為、以下のような顧客情報リストデータを顧客マスター形式に変換することをスクリプト定義で実現しています。)

```
module_name,name,value
Accounts,assigned_user_name,tanaka
Accounts,modified_by_name,admin
Accounts,created_by_name,admin
Accounts,id,f01915c7-2f42-e70f-e4aa-4df70310f77b
Accounts,name,オープンソースCRM 618509
Accounts,date_entered,2011-06-14 06:47:24
Accounts,date_modified,2011-06-14 06:47:24
Accounts,modified_user_id,1
Accounts,created_by,1
Accounts,description,
Accounts,deleted,0
Accounts,assigned_user_id,seed_sally_id
Accounts,account_type,顧客
Accounts,industry,
Accounts,annual_revenue,
Accounts,phone_fax,
Accounts,billing_address_street,金岡町54854
Accounts,billing_address_city,川崎市麻生区
Accounts,billing_address_state,大阪府
Accounts,billing_address_postalcode,38713
Accounts,billing_address_country,日本
Accounts,rating,
Accounts,phone_office,(706) 723-9896
```

tanaka,admin,admin,f01915c7-2f42-e70f-e4aa-4df70310f77b,オープンソースCRM...

顧客マスター形式に変換

module_name 要素を基準に
ループ取得されたデータ

顧客マスター形式に変換

4. 顧客マスター形式に変換した csv ファイルから DB の取引先テーブルを更新する定義を作成します。

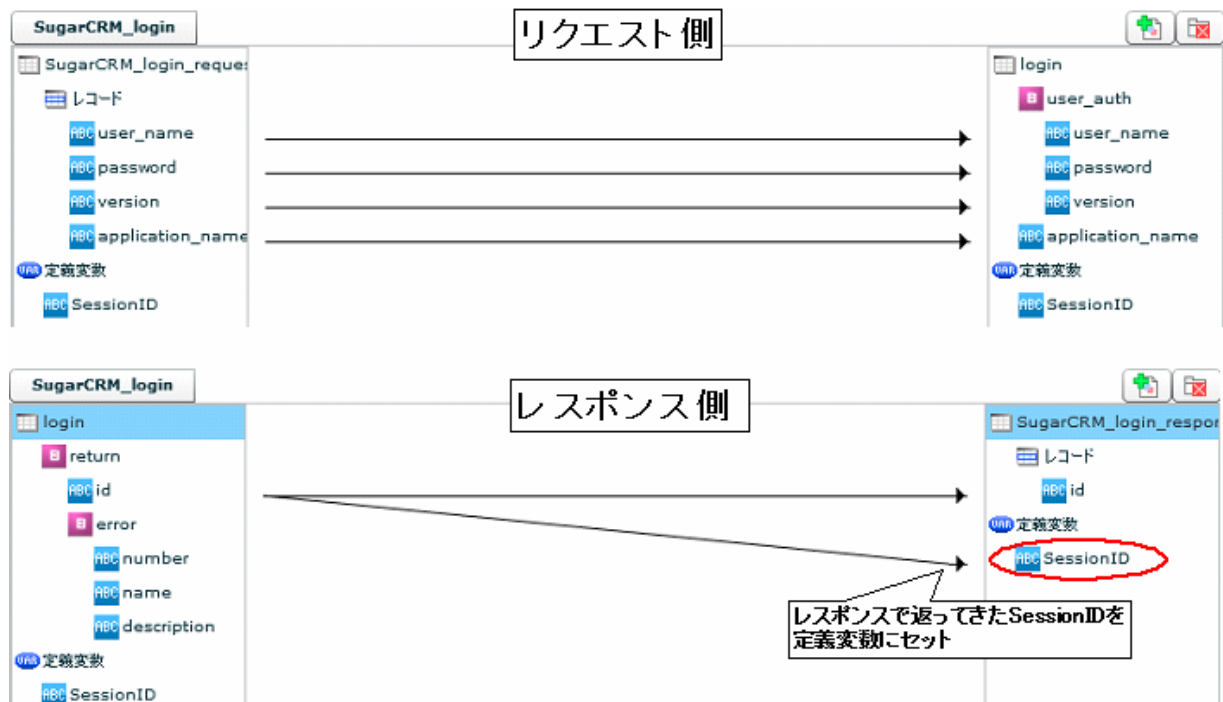
作成する定義

- ・ sugarCRM_login 定義
- ・ sugarCRM_getEntryList 定義
- ・ sugarCRM_logout 定義
- ・ sugarCRM_MakeAccounts マスターCSV 定義
- ・ SugarCRM_DB2UPDATE 定義
- ・ sugarCRM_getEntryListTASK 定義 (sugarCRMlogin、 getEntryList、 logout 定義)

[手順 1] sugarCRMlogin、 getEntryList、 logout の 3 定義を作成します。

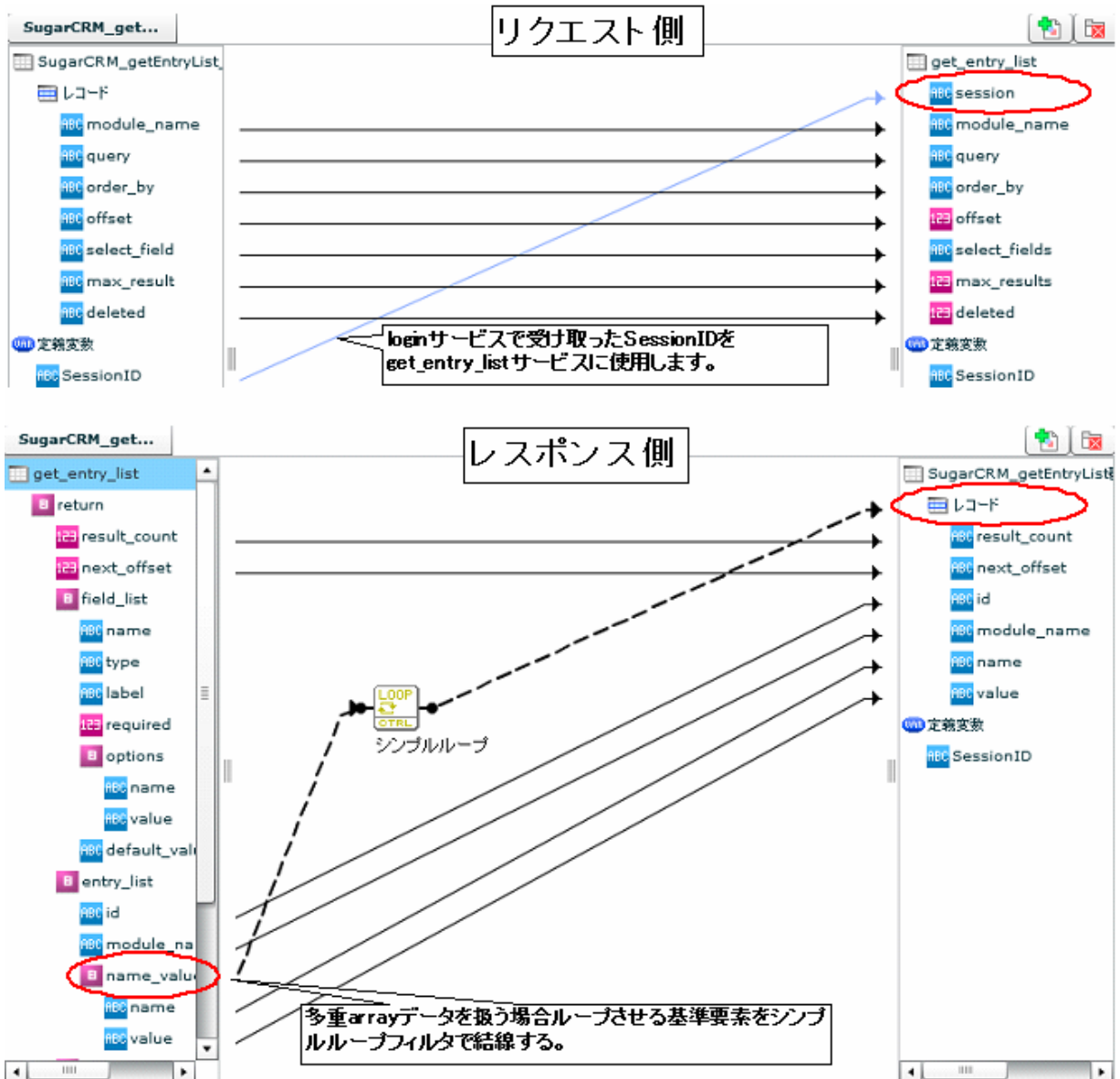
login サービスのレスポンスで得た sessionID を定義変数にセットし、後の 2 定義で使用するため全ての転送定義に定義変数 SessionID を追加しています。

- ・ まず、sugarCRM_login 定義を作成します。login に必要なユーザー名、パスワードなどを login サービスのリクエスト側にセットします。
- ・ レスポンス側で返ってきた SessionID を定義変数にセットします。



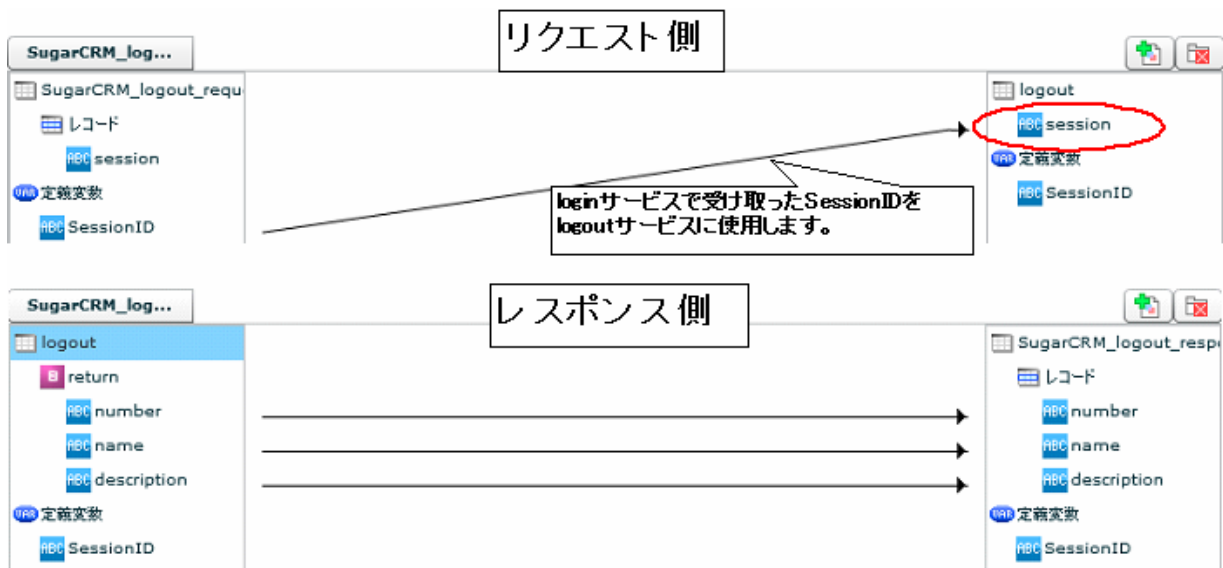
sugarCRM_login (リクエスト側及びレスポンス側)

- ・次に、sugarCRM_getEntryList 定義を作成します。先の login サービスで取得した SessionID をリクエスト側の Session パラメータにセットします。
- ・return として返ってくるデータが多重 array 構造になっているため、ループさせる基準要素を選択しシンプルループフィルタで結線する必要があります。



sugarCRM_getEntryList (リクエスト側及びレスポンス側)

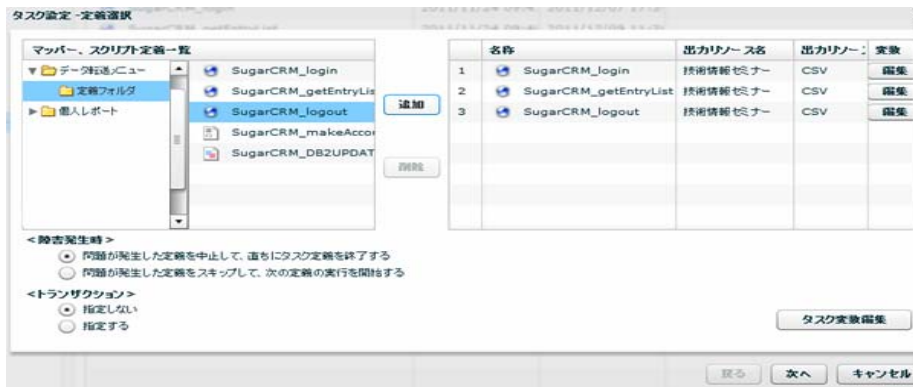
・次に、SugarCRM_logout 定義を作成します。先の login サービスで取得した SessionID をリクエスト側の Session パラメータにセットします。



sugarCRM_logout (リクエスト側及びレスポンス側)

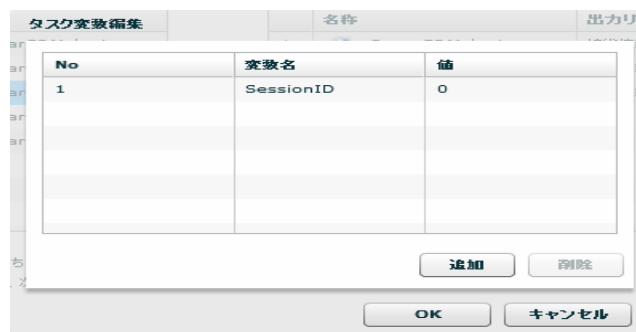
[手順 2] sugarCRMlogin、getEntryList、logout の3 定義を順に実行するタスク定義を作成します。

まず、手順 1 で作成した3 つの転送定義をタスクで実行する定義として順に追加します。



タスク定義 定義選択画面

タスク変数編集ボタンをクリックし、変数名 SessionID を追加します。



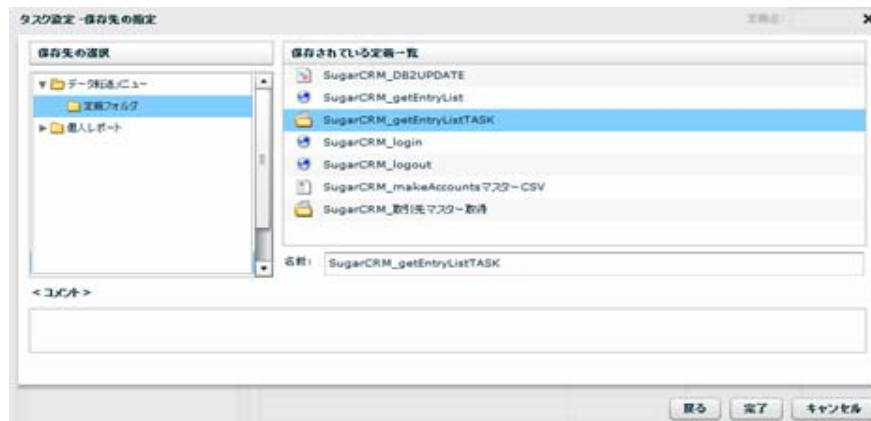
タスク変数編集画面

各転送定義の変数欄の編集ボタンをクリックし、login 定義で得た sessionID を後の2 定義で使用できるように変数受け渡しの設定をします。



変数受け渡し画面（各定義）

名前をつけ保存します。



[手順3] 手順2 で作成したタスク定義を実行してできた sugarCRM_getEntryList 顧客リスト.csv を顧客マスター形式に変換するスクリプト定義を作成します。SugarCRM から値を取得した状態では顧客データの【項目】、【値】で1レコードになっているため、1顧客が複数のレコードに分かれています。DBを更新する場合は各【項目】をまとめて1顧客を1レコードにする必要があります。

```
#set($name = "")
#set($value = "")
#set($cnt = 0)
#foreach ($record in $in)
#if($record.get('name')== 'assigned_user_name' && $cnt > 1)
$value
#set($value = "")
#end
#set($cnt = $cnt + 1)
#set($s = $record.get('value'))
#set($value = $value + $s)
#set($value = $value + ',')
#end
#if($cnt > 0)
$value
#end
```

SugarCRM_makeAccounts マスターCSV 定義

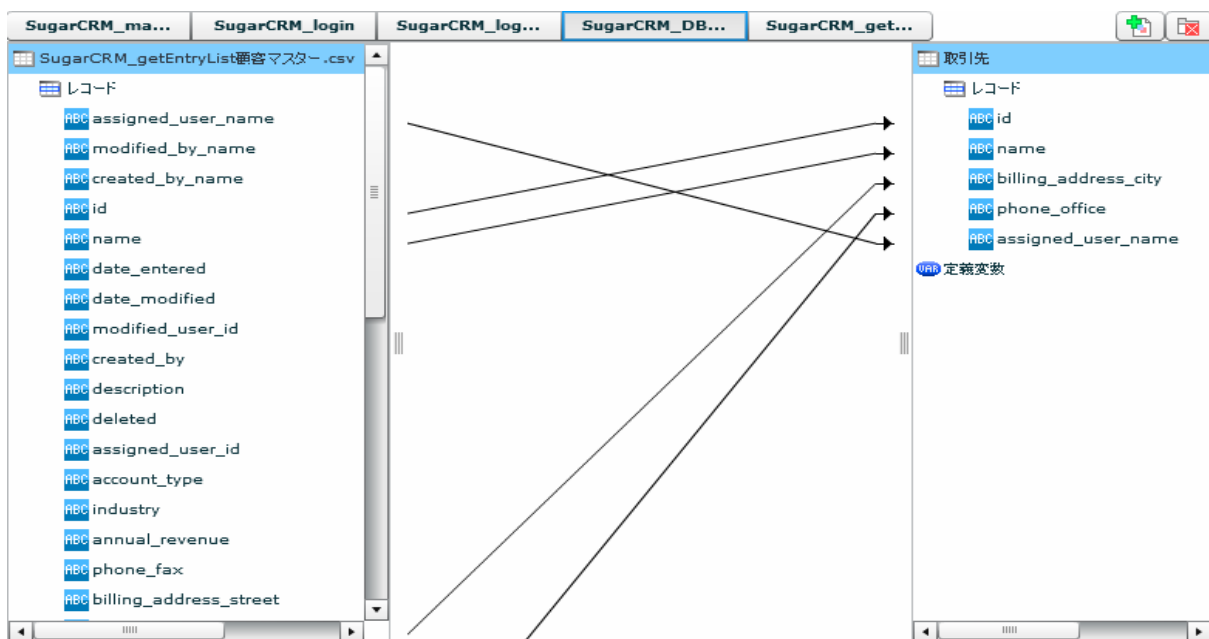
【name】、【value】、【cnt】の3つの変数をつくり初期化する。

変数は英字のみ、【name】は顧客マスターの各フィールド名、【value】は各フィールドの値をレコードごとにカンマ区切りで取得。

レコードが変わったタイミングで、【value】の値を出力して【value】の値を初期化

最終レコードを出力する。

[手順 4] 手順 3 で作成した SugarCRM_getEntryList 顧客マスター.csv で DB の取引先テーブルをアップサートする定義を作成します。



SugarCRM_DB2UPDATE 定義

以上の手順により、SugarCRM にログインし、取得した顧客情報を元に DB を更新することが可能になります。

Qanat 2.0

1	2010	1	29
3	2010	7	16
4	2011	2	18
5	2011	12	10

JB

221-0022

3 9 C

<http://www.jbat.co.jp/>

JB

