チュートリアル

■ ごあいさつ

このたびは弊社製品 Datamart for WebReport2.0 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、Datamart for WebReport2.0 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。

■ 版権／ご注意

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

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/BI 管理ツール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
・　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	・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、

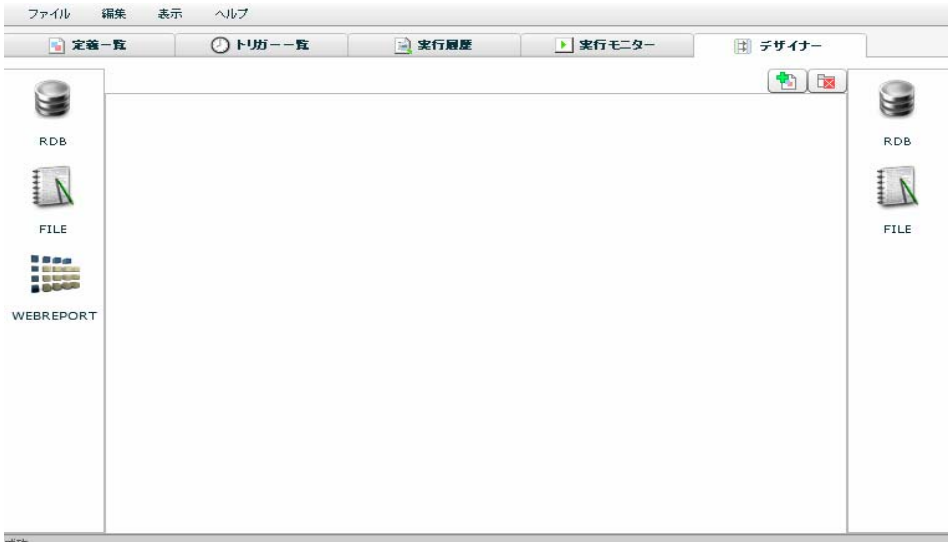
“ 西日本 DB ” データベースの “ 商品マスター ” テーブルへデータを新規挿入/更新します。

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/BI 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。

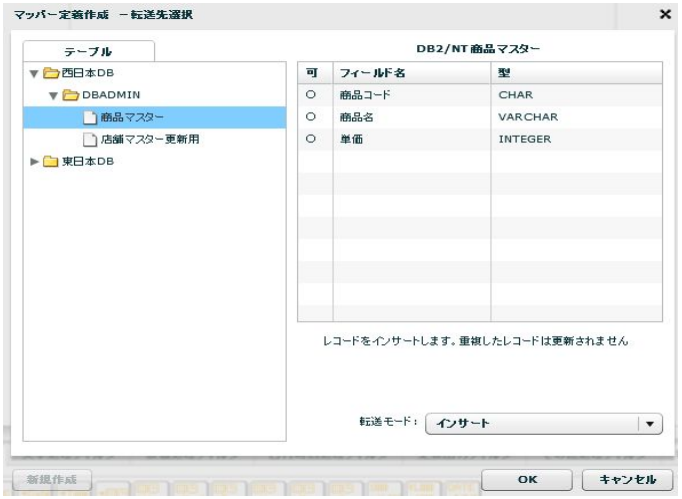
手順 3：　RDB から出力先テーブルを選択します

データの出力先テーブルを選択します。

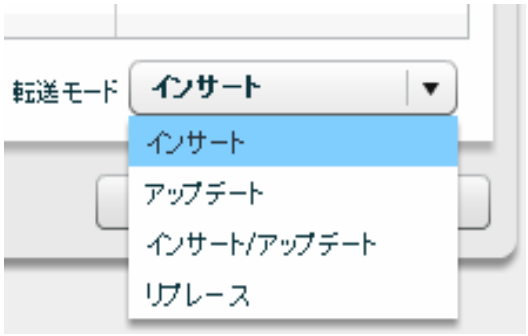
手順 2 では“東日本 DB”からデータを読み込むよう設定しました。手順 3 では出力先として“西日本 DB”に取得したデータを転送するように指定します。

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

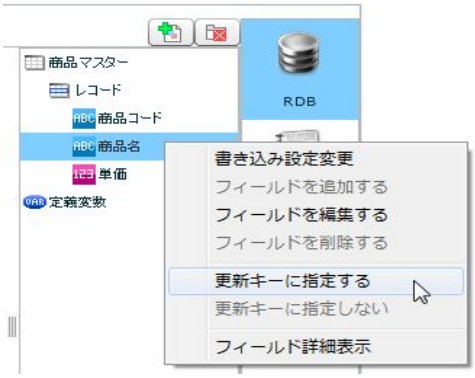
[OK]ボタンをクリックし転送先選択画面を閉じます。

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。

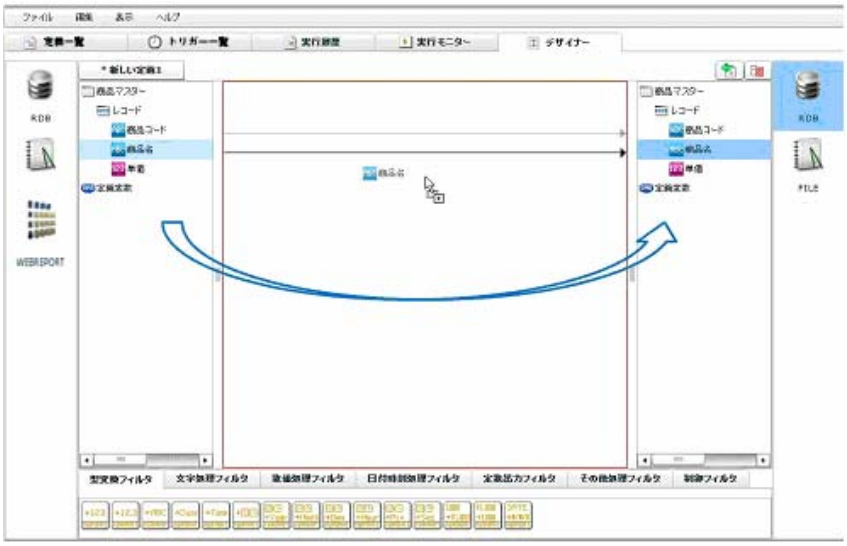
出力先のフィールドのうち、更新キーとなるフィールドを右クリックし、メニューから[更新キーに指定する]をクリックします。入力元と出力先で一致する更新キーが見つかった場合は更新し、一致しないデータの場合は新規に挿入されます。



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。

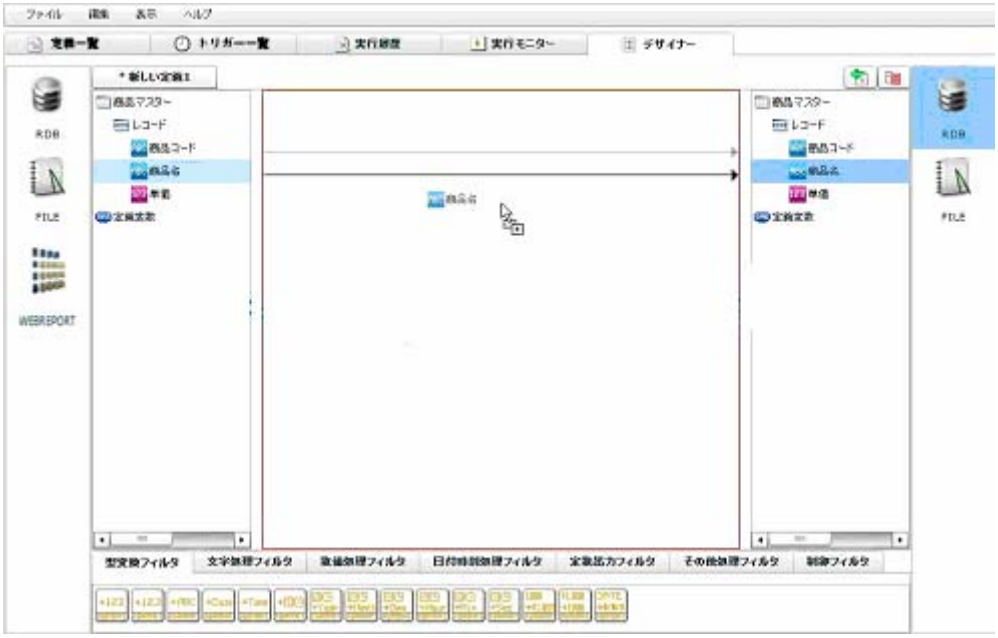
この実習では“東日本 DB の商品マスター”から“西日本 DB の商品マスター”へデータを転送することです。そのためには画面左側のフィールドを、右側の同名のフィールドに対してマウスでドラッグ＆ドロップしマッピングを一つ一つ作成します。



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 の単価にその分上乗せした値を西日本 DB の単価フィールドにマッピングします。

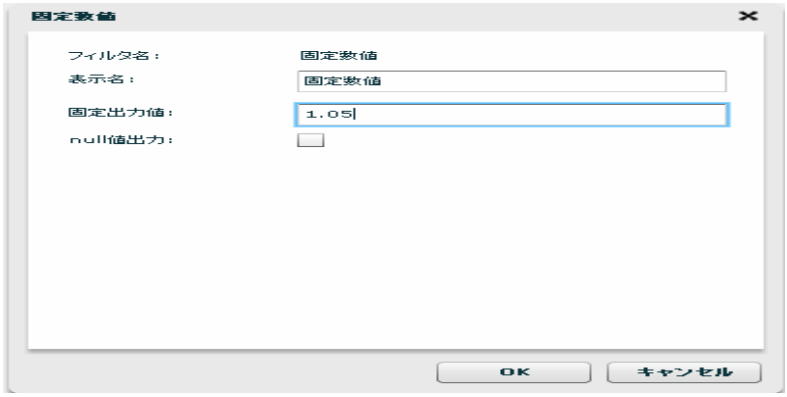
このマッピングには “ 5% ” という値を持たせた固定数値フィルタと、掛け算フィルタを利用し表現します。

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。

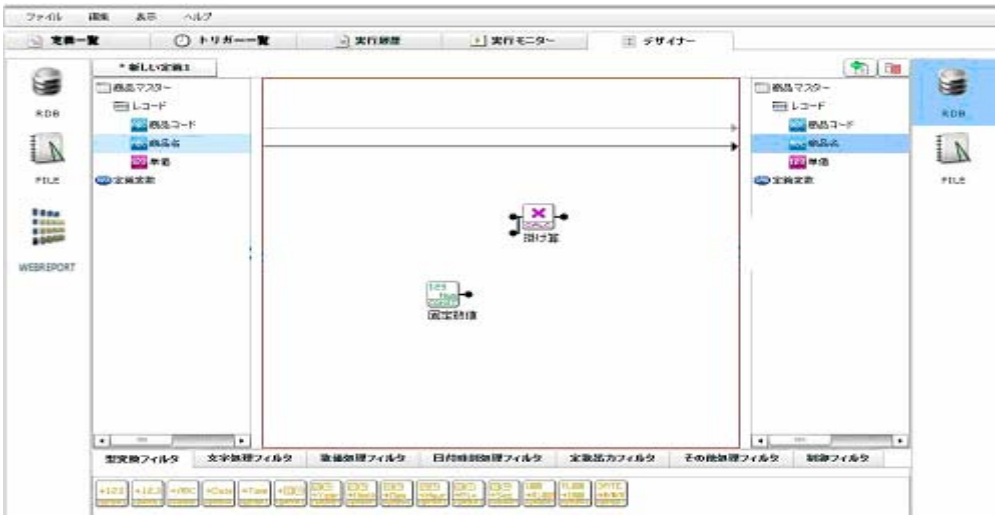


固定数値フィルタを配置すると次のようなプロパティ画面が表示されます。

[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。

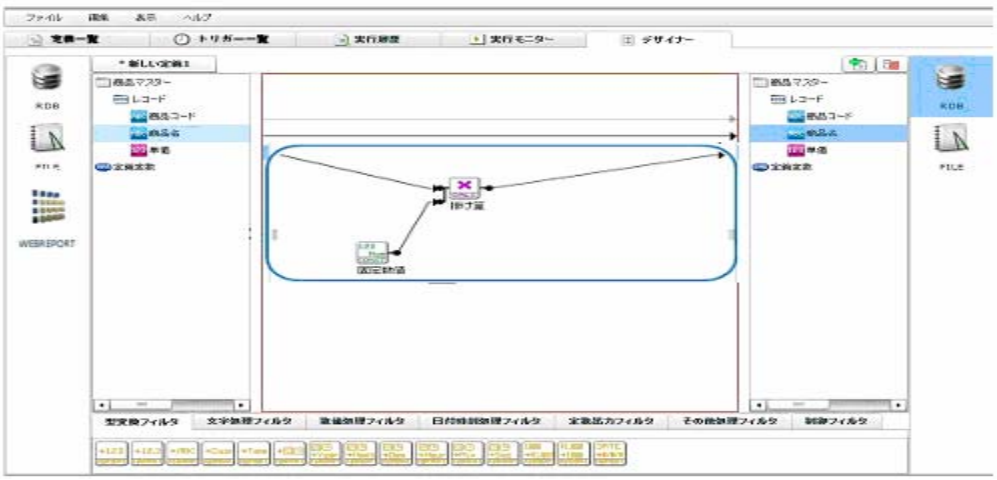


次に[数値処理フィルタ]タブから、掛け算フィルタを選択し、同様にドラッグ＆ドロップし、画面中央付近に配置します。



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

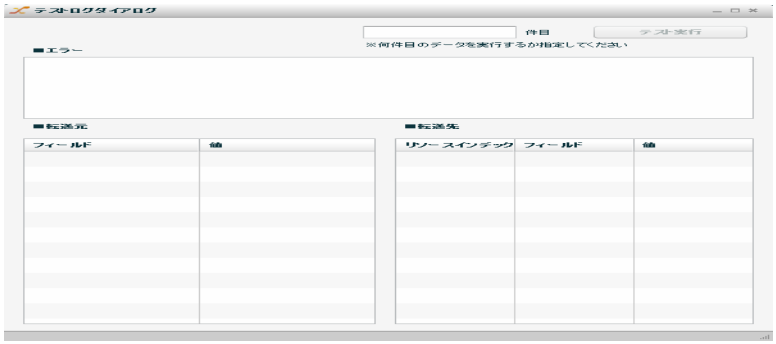
メニューバーの[ファイル]→[名前を付けて保存]をクリックします。次の画面ように保存先を選択し[名前]欄に定義名称、[コメント]欄にその定義の説明を入力し[保存]をクリックします。

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。

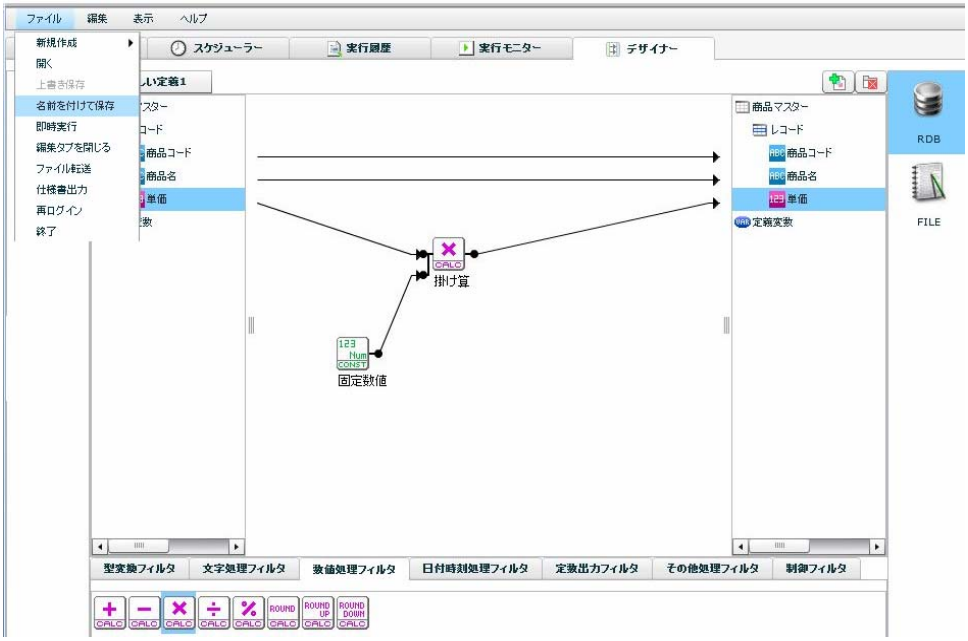
画面下部のデータ表示部に結果が表示されることを確認します。

［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。

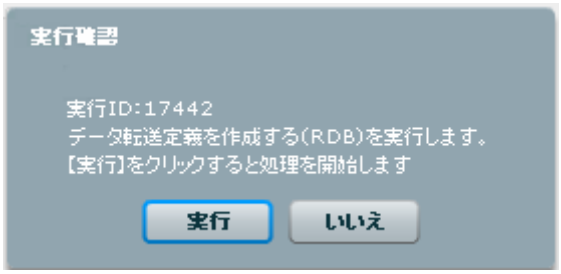
テスト実行では出力先リソースは更新されません。

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
・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	・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

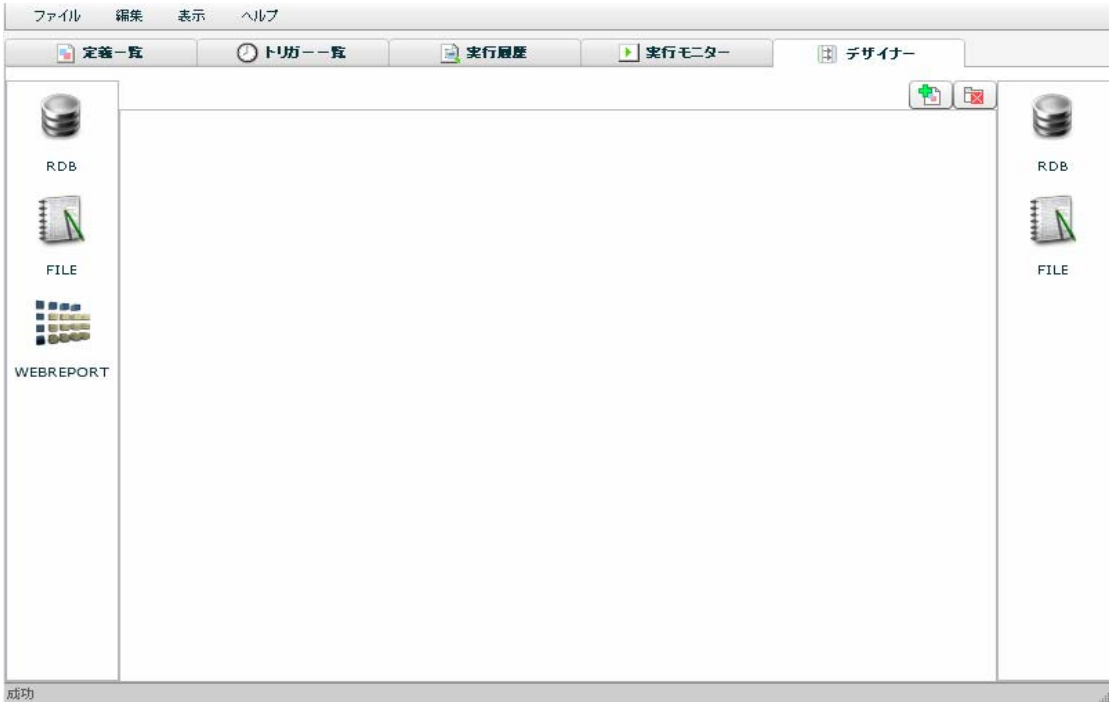
“ 東日本ファイルリソース ” フォルダの “ 商品マスター ” ファイルからデータを読み込み、

“ 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。

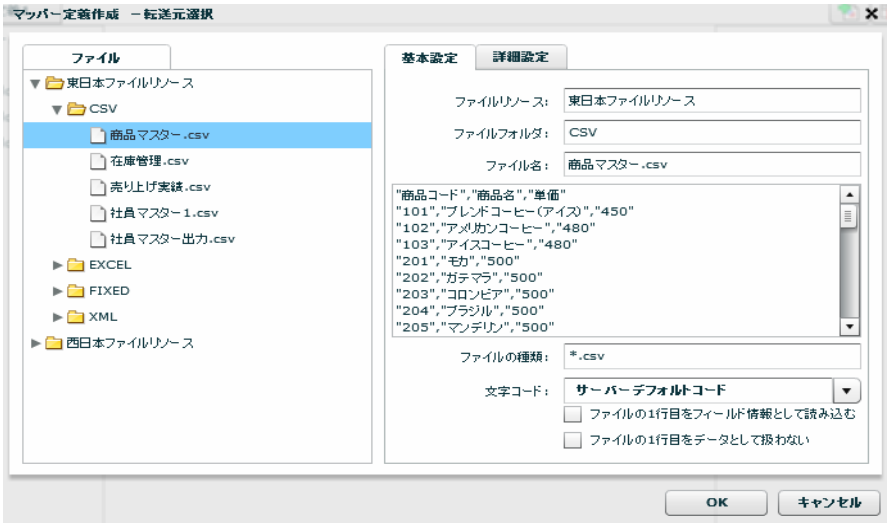
まず、入力元ではデータの読み込み先であるファイルを一つ選択します。

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

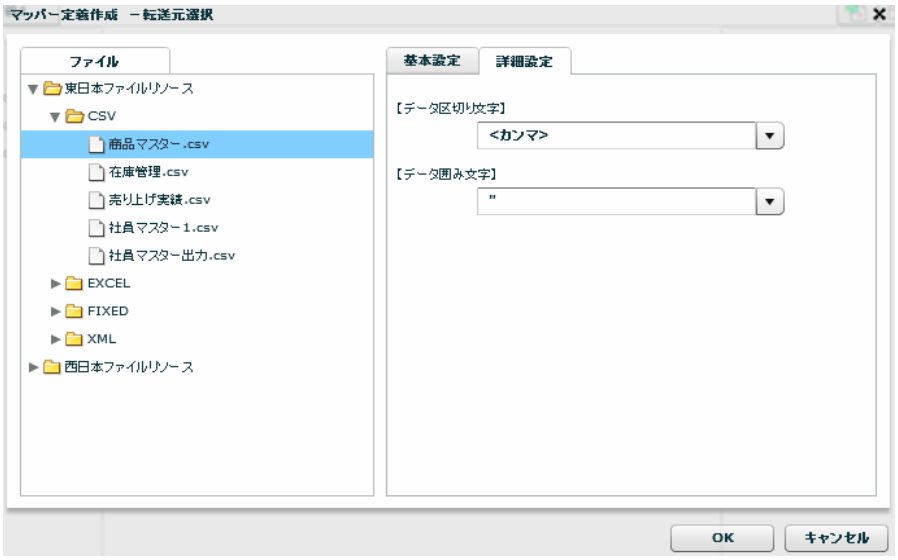
[FILE]と書かれたアイコンを右クリックし[CSV ファイル選択]をクリックします。

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］のチェックをいれます。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。

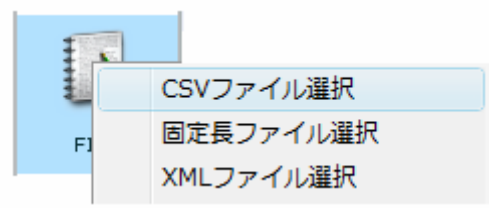
データ区切り文字や囲み文字がコンボボックスの一覧にない場合、任意の文字を入力することで別の文字を指定することができます。

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

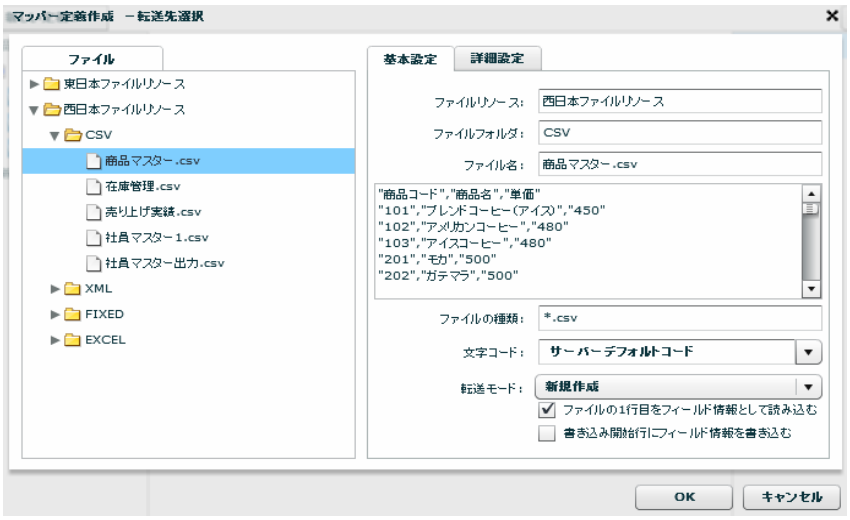
データの出力先ファイルを選択します。

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。

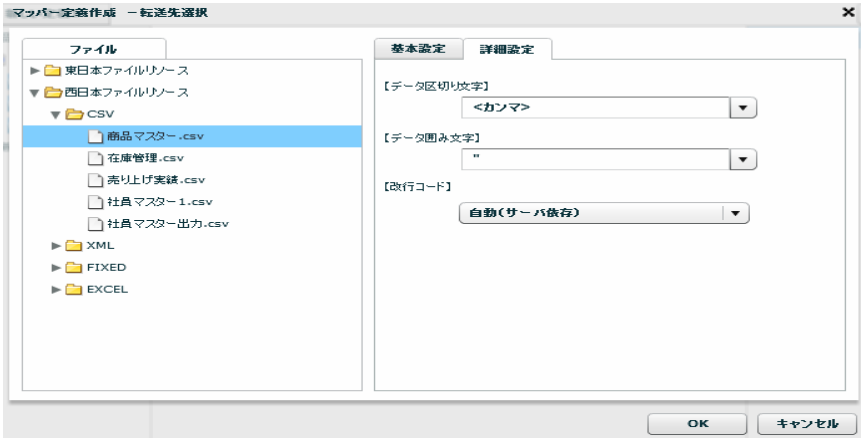
ヘッダ情報は不要になるため、［書き込み開始行にフィールド情報を書き込む］のチェックを外します。

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。

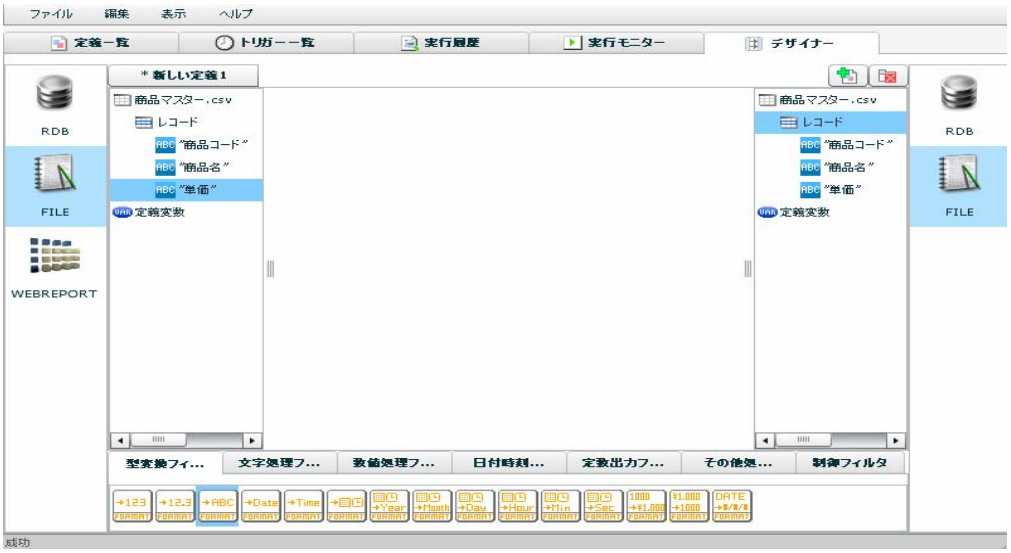


データ区切り文字や囲み文字がコンボボックスの一覧にない場合、任意の文字を入力することで別の文字を指定することができます。

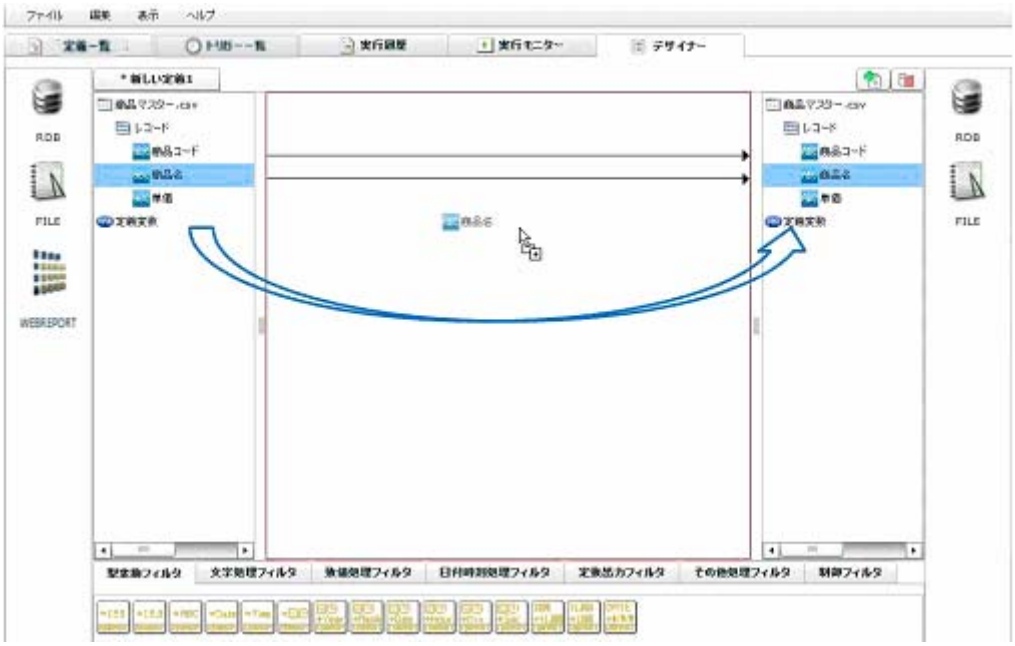
［OK］ボタンをクリックし転送先選択画面を閉じます。

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。



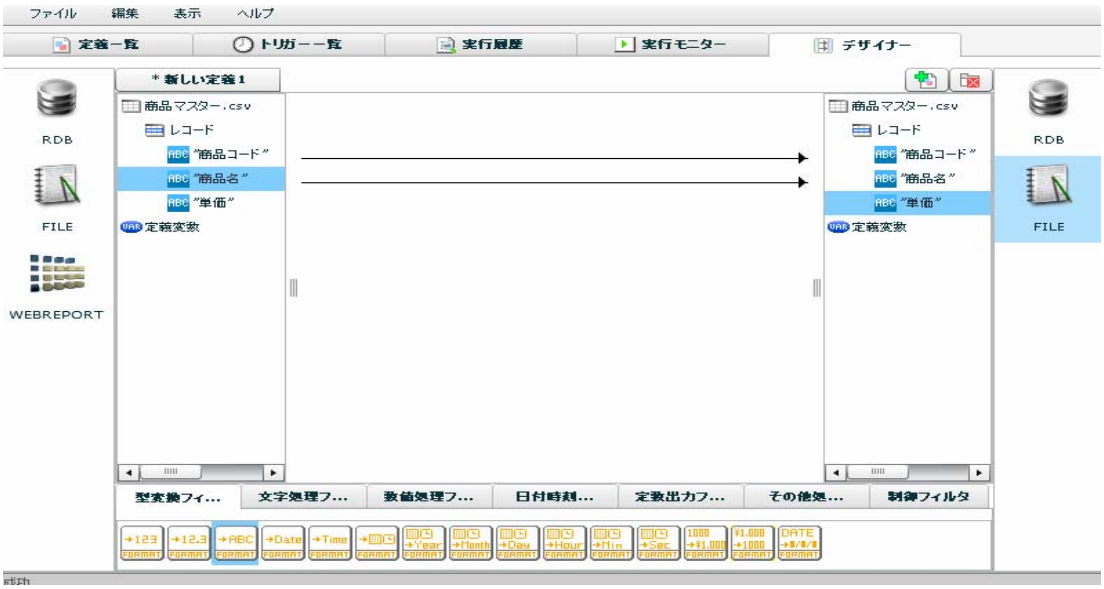
この実習では“東日本リソースフォルダの商品マスターファイル”から“西日本リソースフォルダの商品マスターファイル”へデータを転送することです。そのためには画面左側のフィールドを、右側の同名のフィールドに対してマウスでドラッグ＆ドロップしマッピングを一つ一つ作成します。



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。

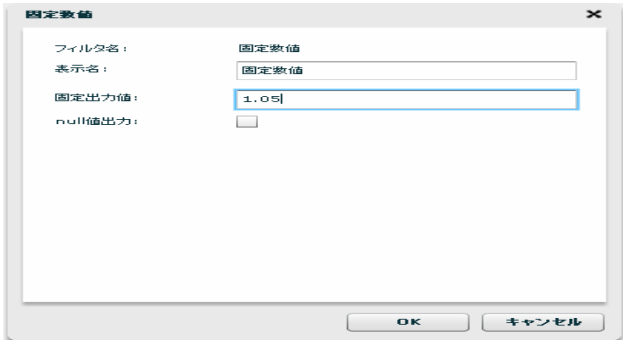
このマッピングには“ 5% ”という値を持たせた固定数値フィルタと、掛け算フィルタを利用し表現します。

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。

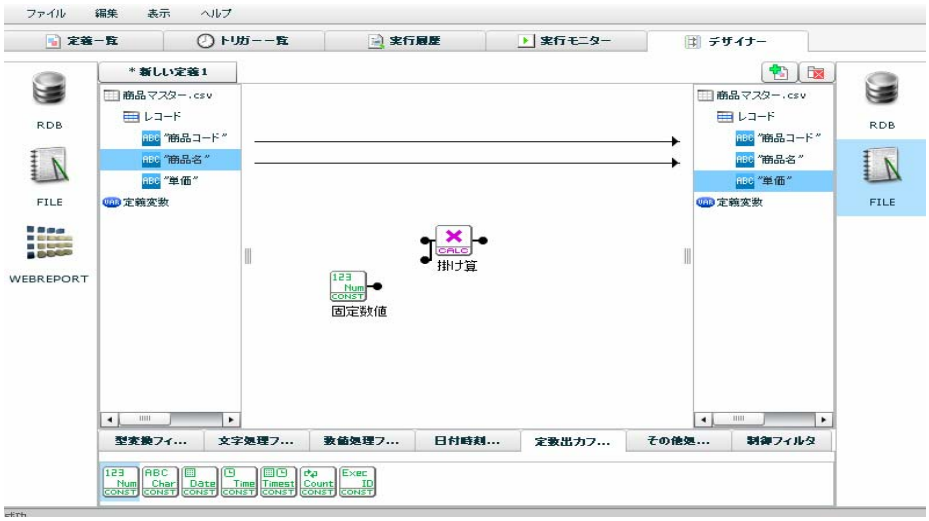


固定数値フィルタを配置すると次のようなプロパティ画面が表示されます。

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。

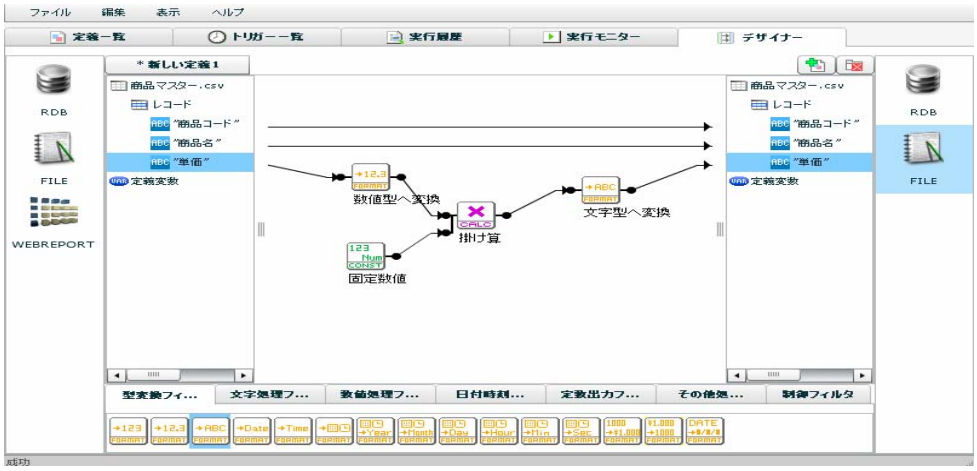


次に[数値処理フィルタ]タブから、掛け算フィルタを選択し、同様にドラッグ＆ドロップし、画面中央付近に配置します。



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
・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	・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

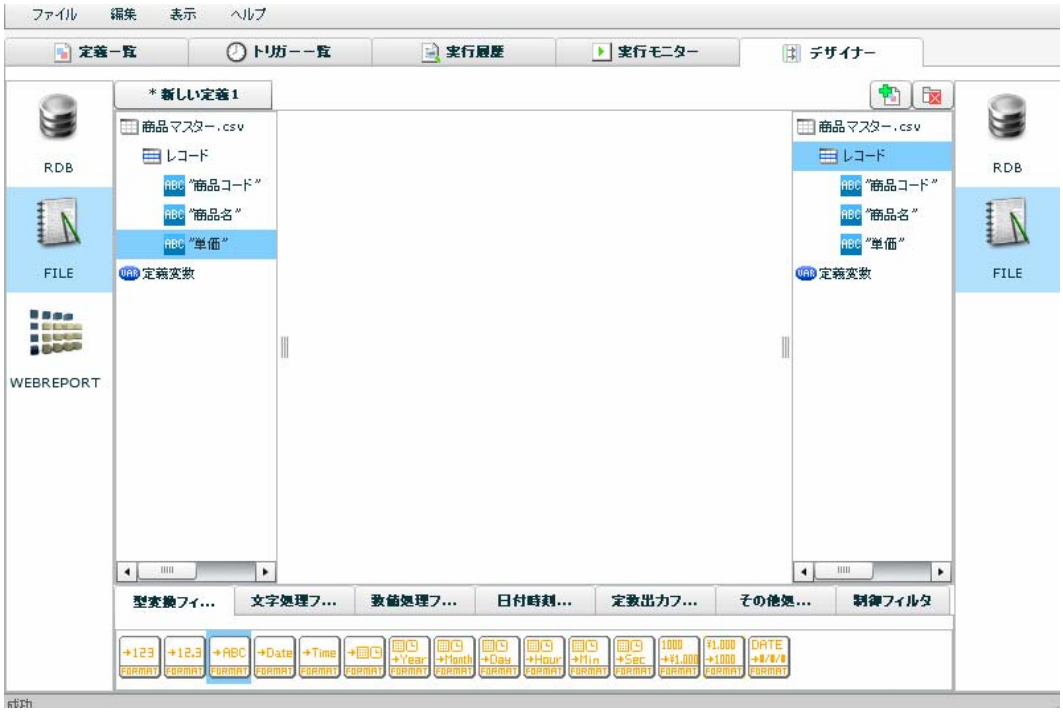
“ XML ” フォルダの “ 売上情報 ” ファイルからデータを読み込み、

“ 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。

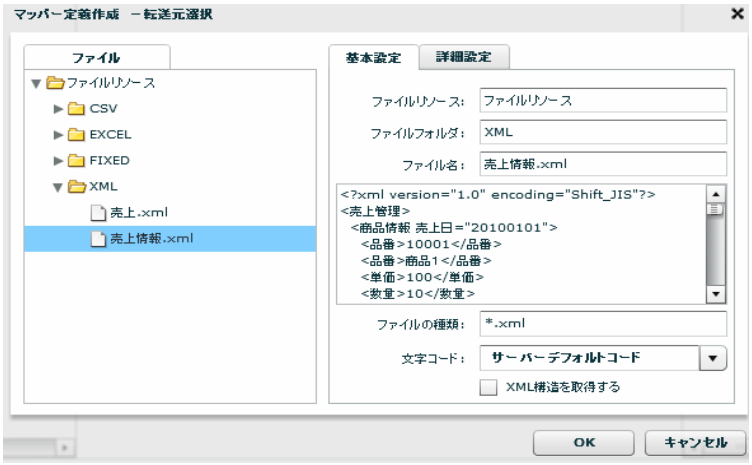
まず、入力元ではデータの読み込み先であるファイルを一つ選択します。

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

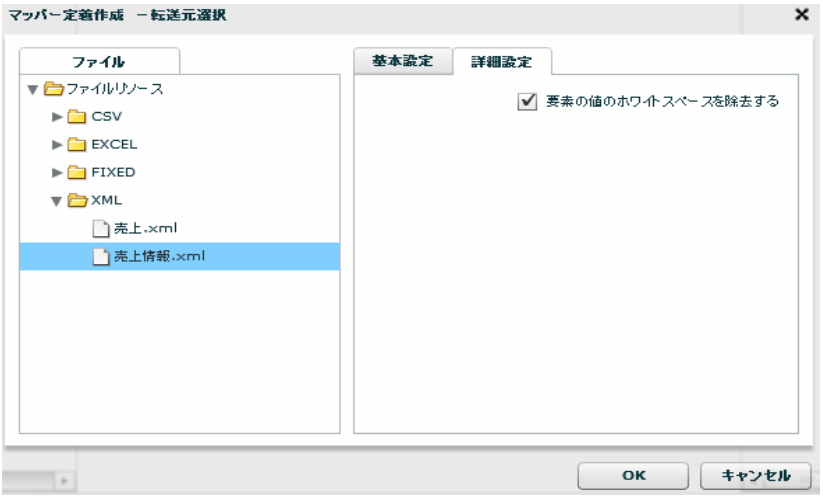
[FILE]と書かれたアイコンを右クリックし[XML ファイル選択]をクリックします。

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の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。



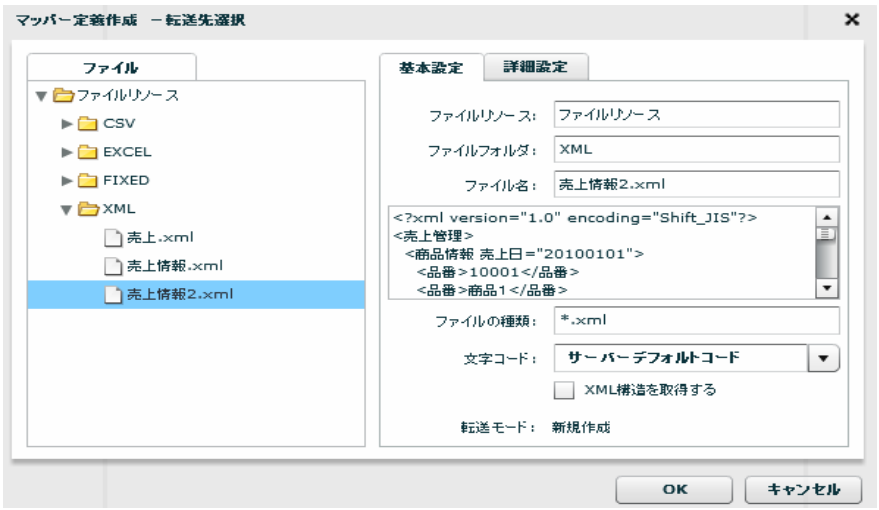
［OK］ボタンをクリックし転送元選択画面を閉じます。

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。

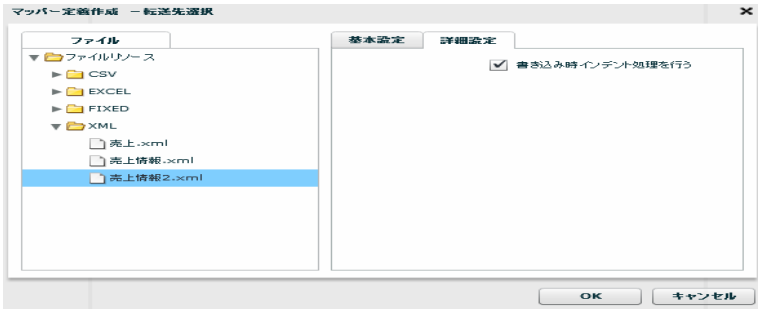
[転送先選択]画面で“ XML ”フォルダから“ 売上情報 2 ”テーブルを選択し転送先ファイルを指定します。



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。

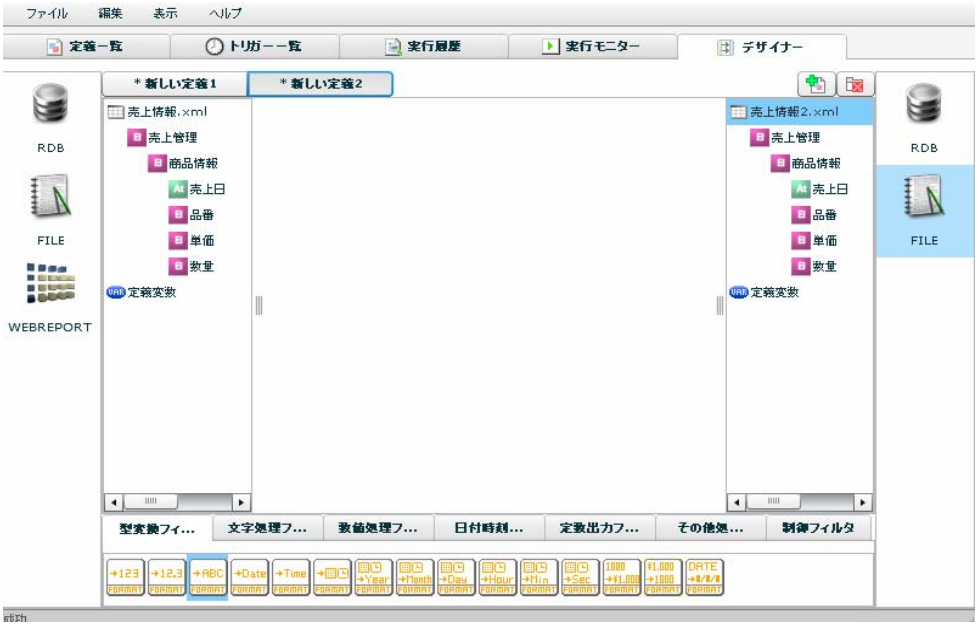
インデント処理を行うと要素の先頭が字下げされます。



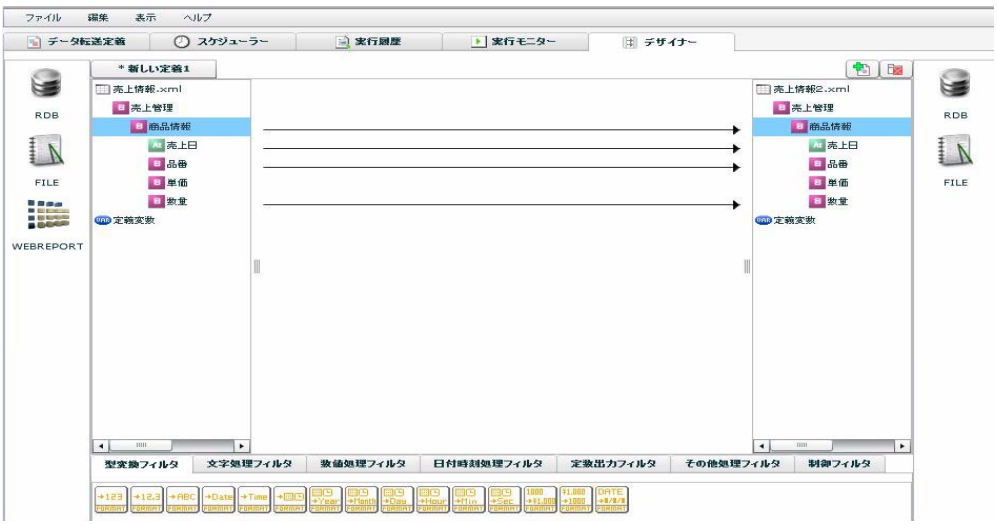
［OK］ボタンをクリックし転送先選択画面を閉じます。

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。



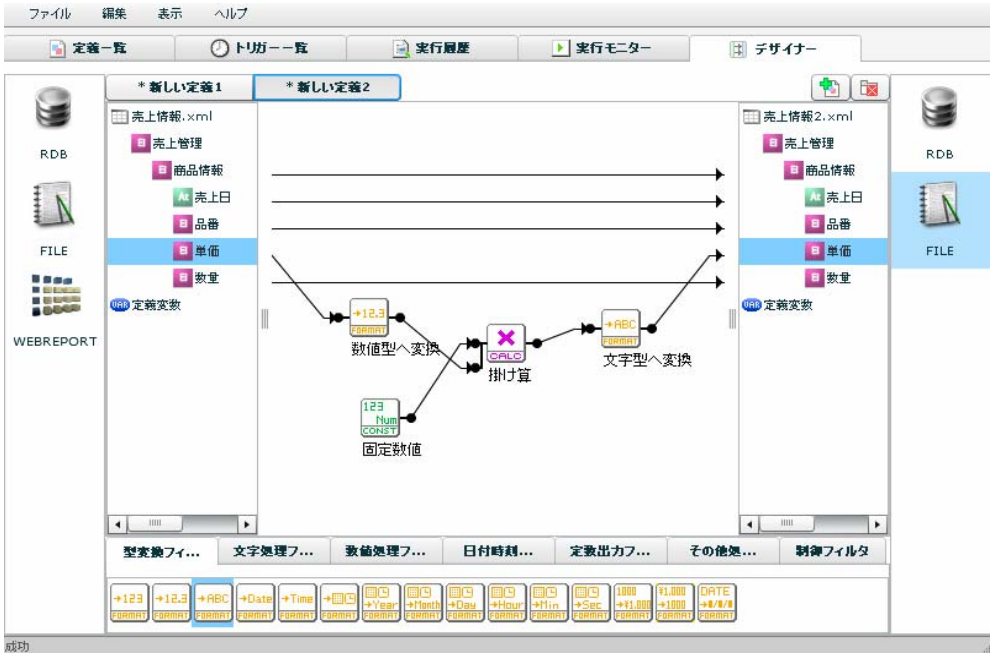
そのためには画面左側の転送したいフィールドを、右側の同名のフィールドに対してマウスでドラッグ＆ドロップしマッピングを一つひとつ作成します。



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。

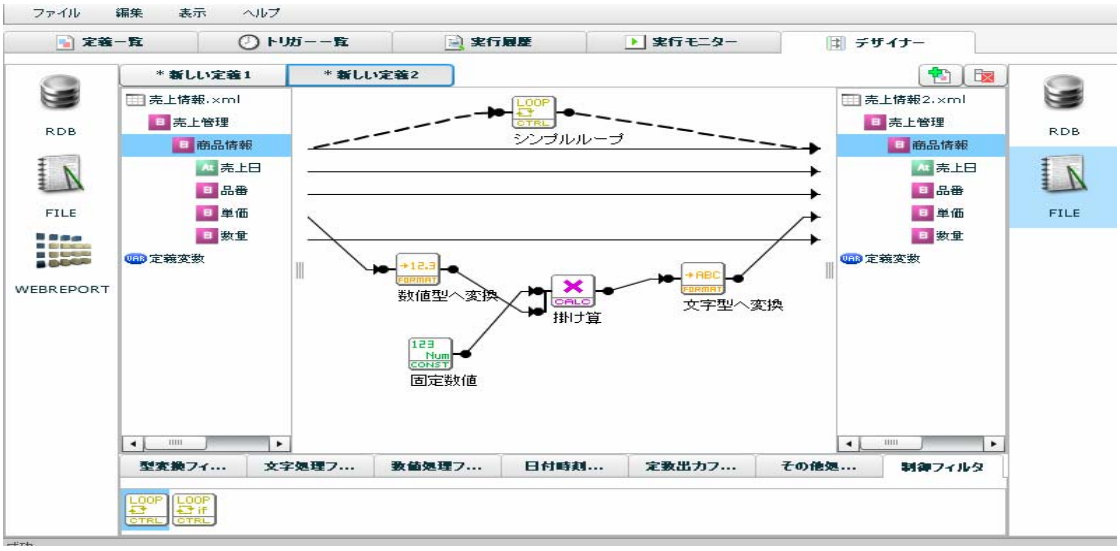
（詳しくは 1-1 データ転送定義を作成する(データベース)参照）



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。

1-3: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った場合があります。

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
・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	・スクリプト定義

● 事前修了しておきたい実習項目

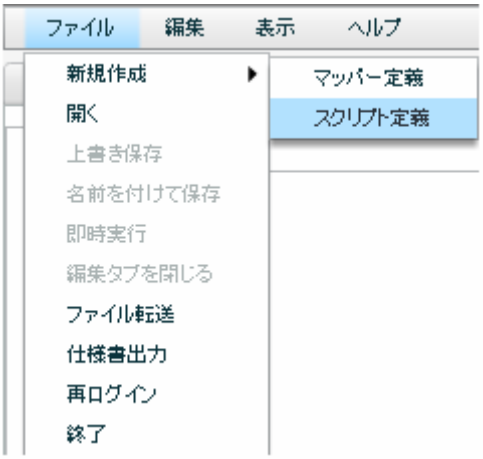
ありません。

● 実習のシナリオ

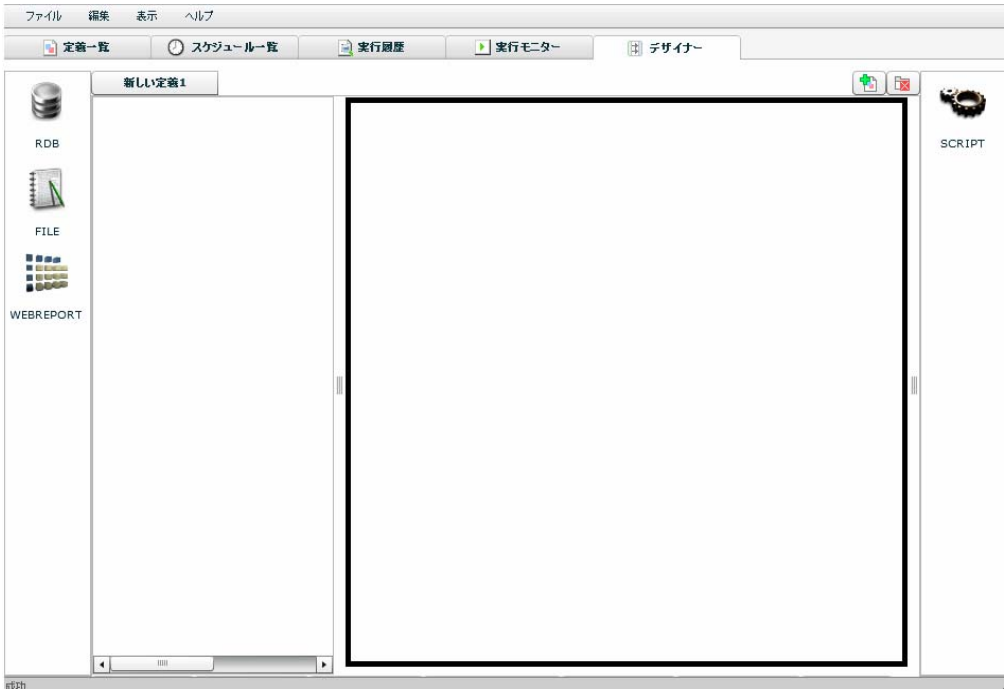
“ 売り上げ実績.csv ” からデータを読み込み、年度ごとの売り上げ、粗利益の合計を算出し CSV ファイルに出力する。

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



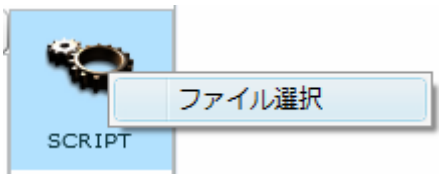
スクリプト定義新規作成画面

手順２：　入力リソースに CSV として[売り上げ実績.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。

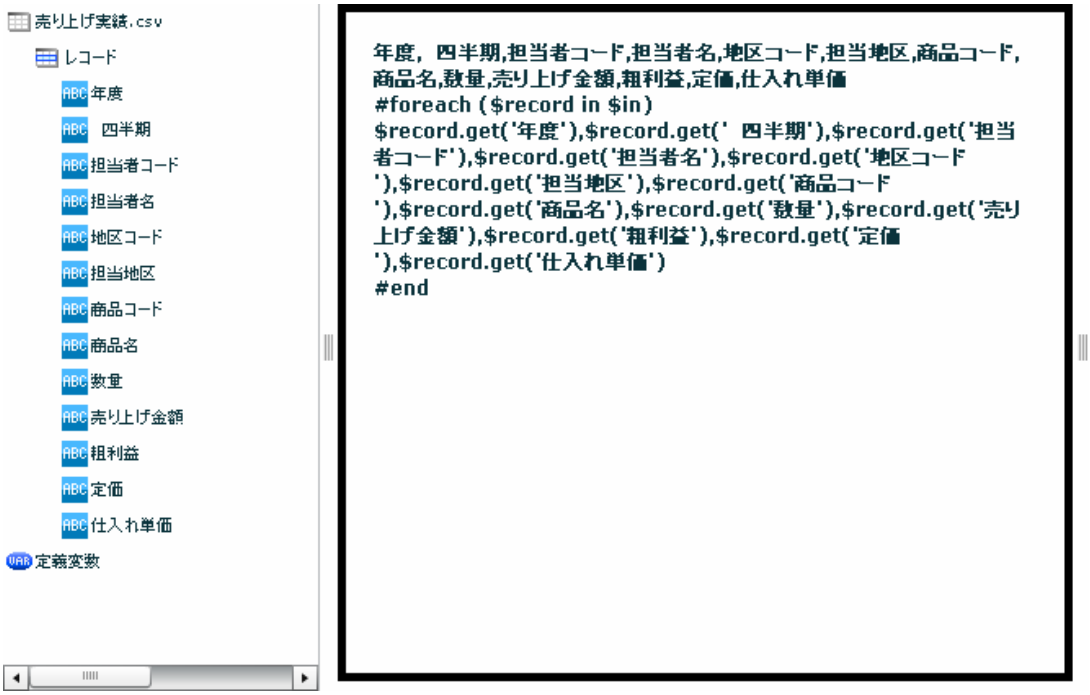
入力リソース選択

手順３：　出力ファイルに書き込むファイルを選択する。



出力リソース選択

手順４：　自動でデフォルト記述が書き込まれる。



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
```

①　②　③　④

処理の記述

① 【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。

② 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荒利益の合計を出力し、その後変数を初期化する。

③ そのレコードの【売り上げ】、【粗利益】を加える

②、③の繰り返し

④ 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くれない 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。

1-4: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。

第 2 章

第 2 章 ー応用編ー

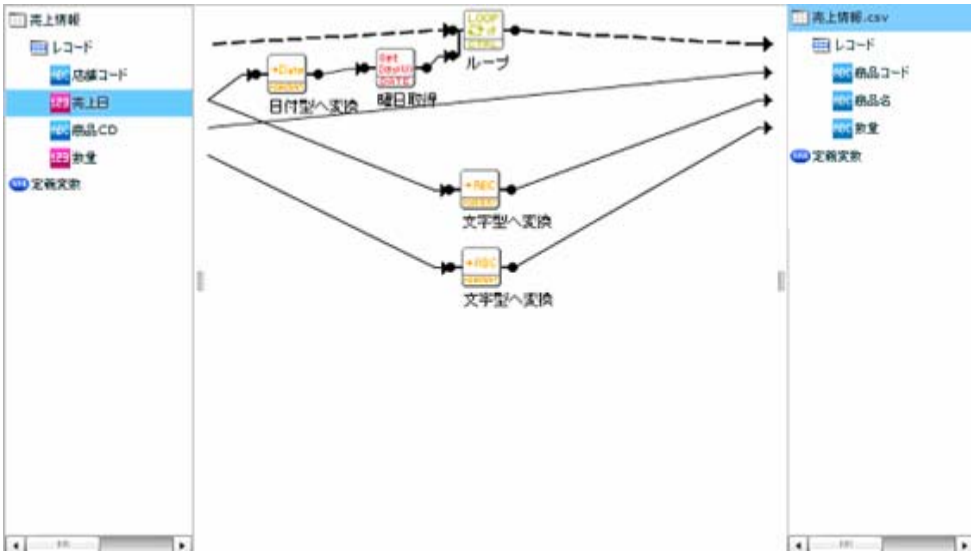
2-1:便利なフィルタ（応用）

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



① 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）

② 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。

③ 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。

④ 文字型へ変換フィルタ

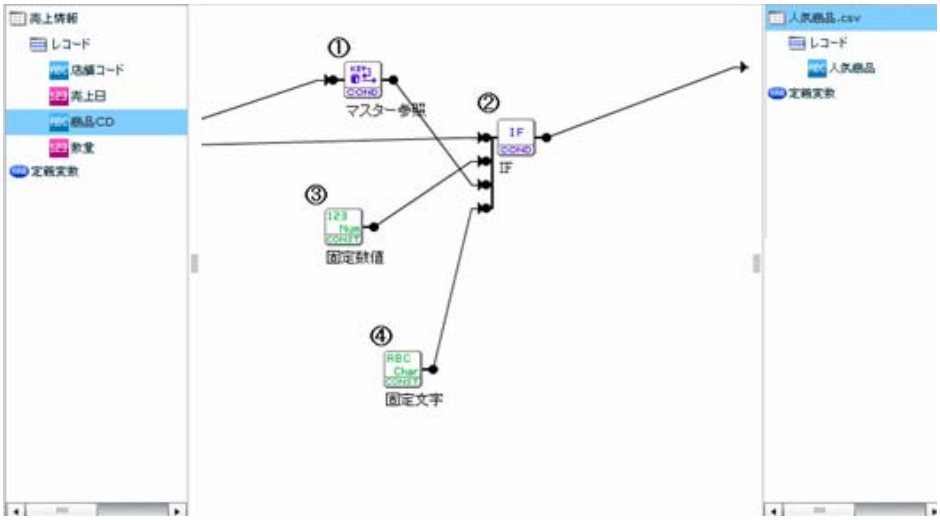
今回の場合、数量を文字型に変換して出力します。

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



① 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ＣＤ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。

② 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。

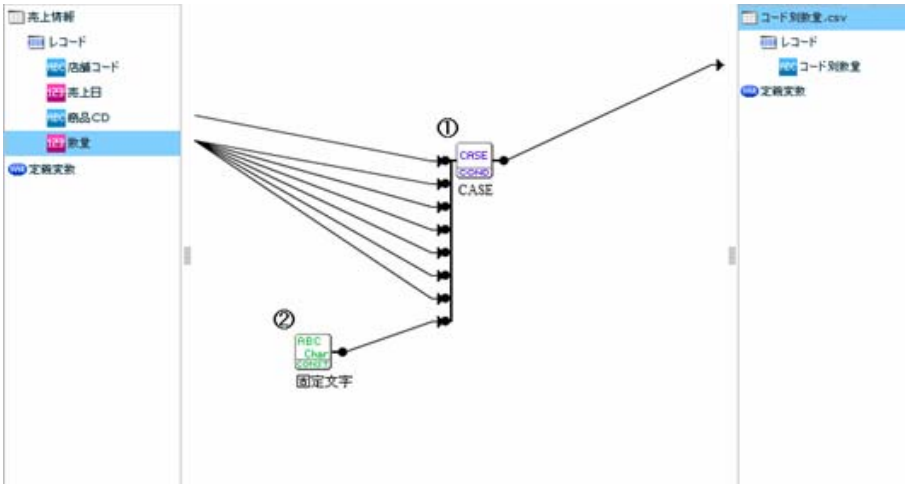
③ 固定数値、固定文字フィルタ

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



① 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。

② 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。

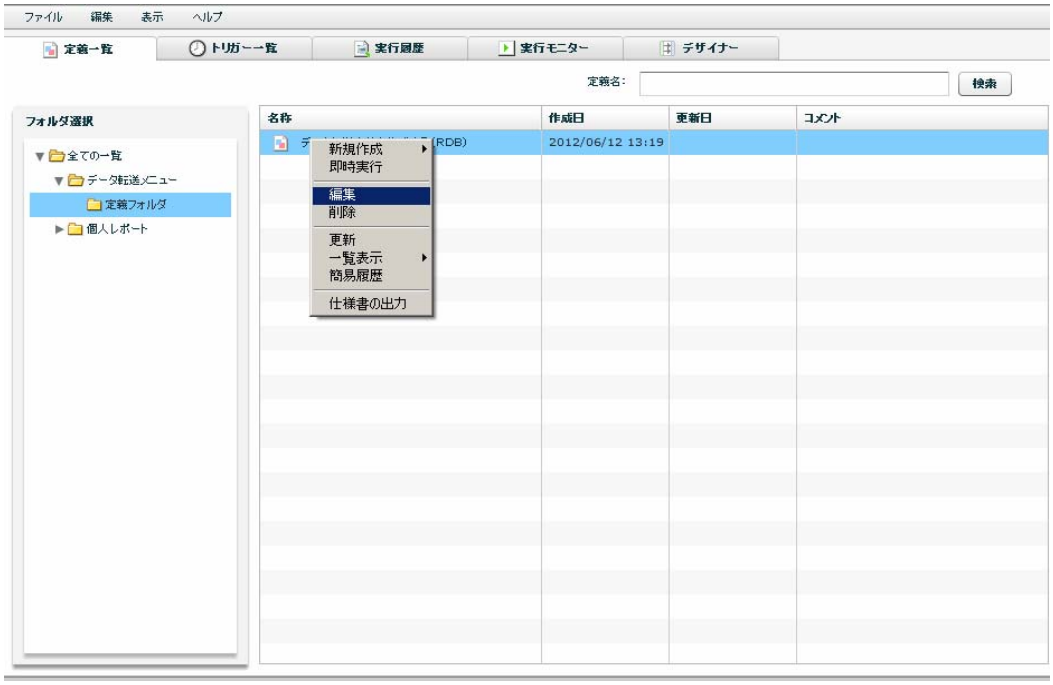
出力先リソースに指定されている西日本 DB の“ 商品マスター ”テーブルを外し、新たに“ 在庫情報 ”テーブルを作成します。

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。

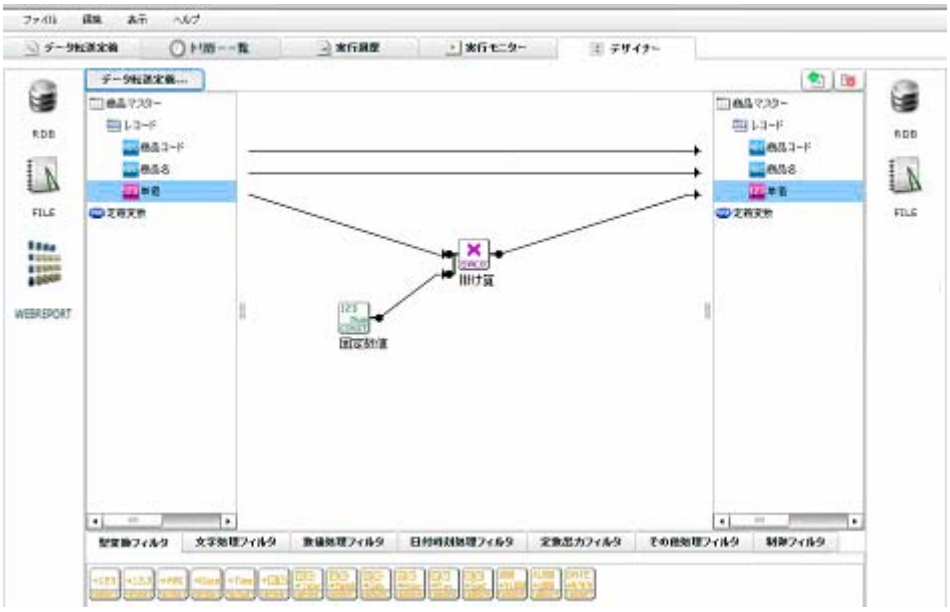
手順１：　データ転送定義を選択し編集します

[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。



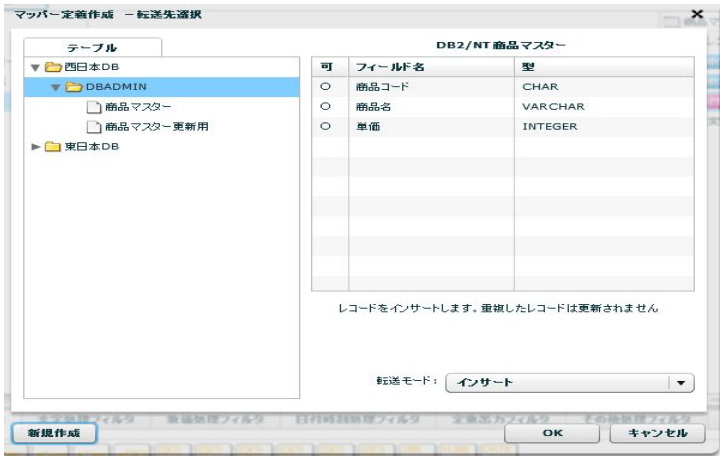
手順２：　作成するテーブル情報を入力します

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。

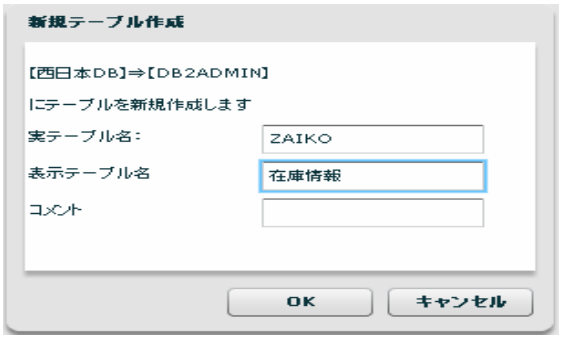


転送先選択画面が表示されます。

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。

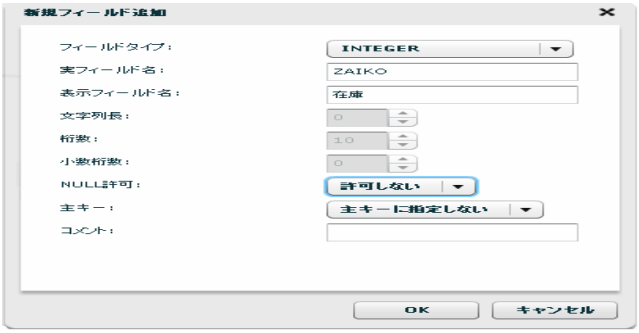


新規テーブル作成画面では、実テーブル名と画面上に表示される表示テーブル名を入力し[OK]ボタンをクリックします。



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。

次のフィールドを出力先に追加します。

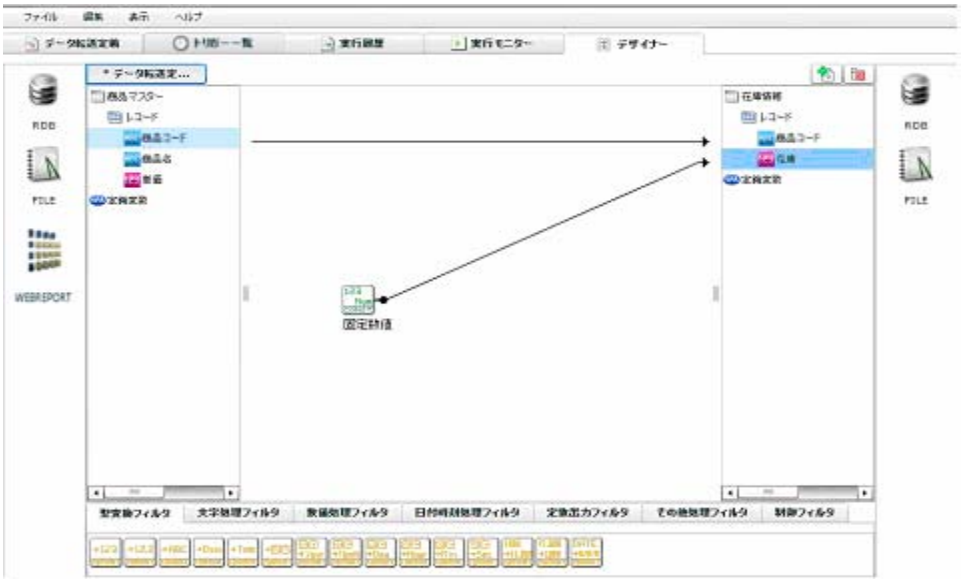
商品コード（文字型）

在庫（数値型）

手順３：　各フィールドをマッピングします

画面上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。

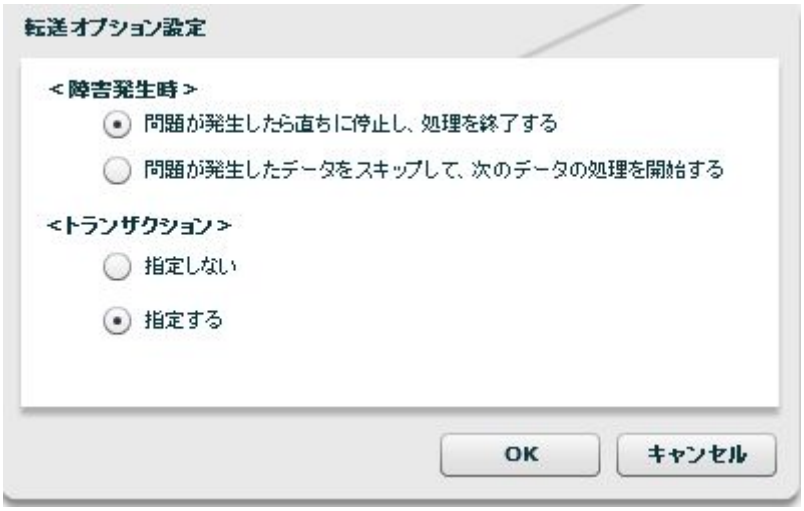
キャンバス上に固定数値フィルタを配置し、そのプロパティの固定出力値欄に 100 を指定し、在庫フィールドと結線します。



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。

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

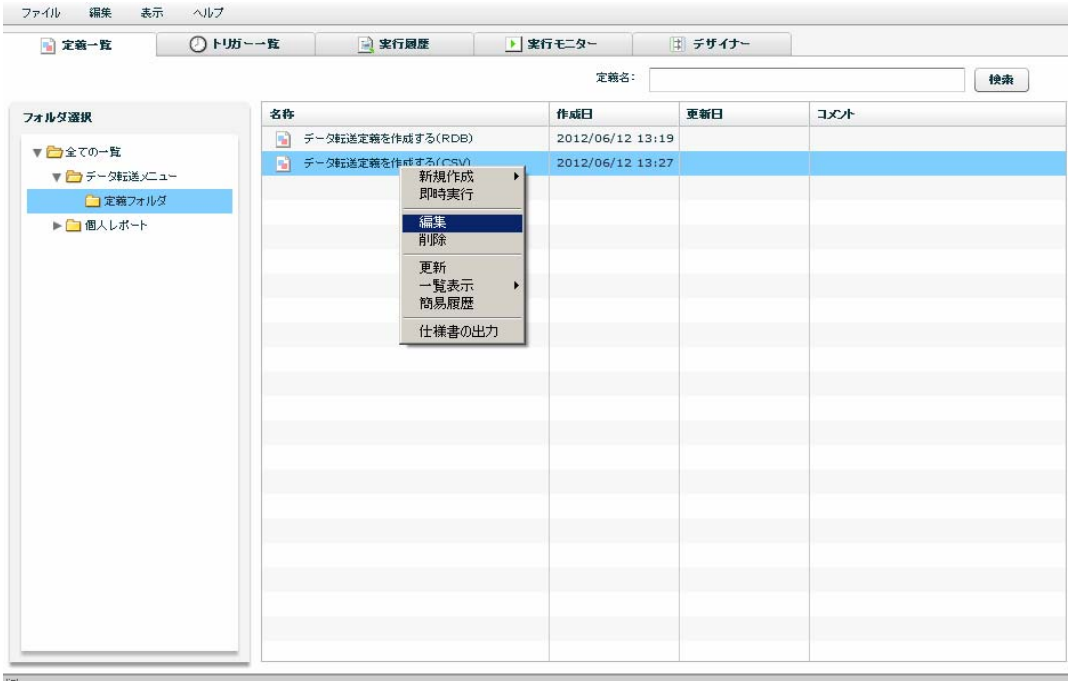
『1-2：データ転送定義を作成する（ファイル）』で作成したデータ転送定義を編集します。

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。

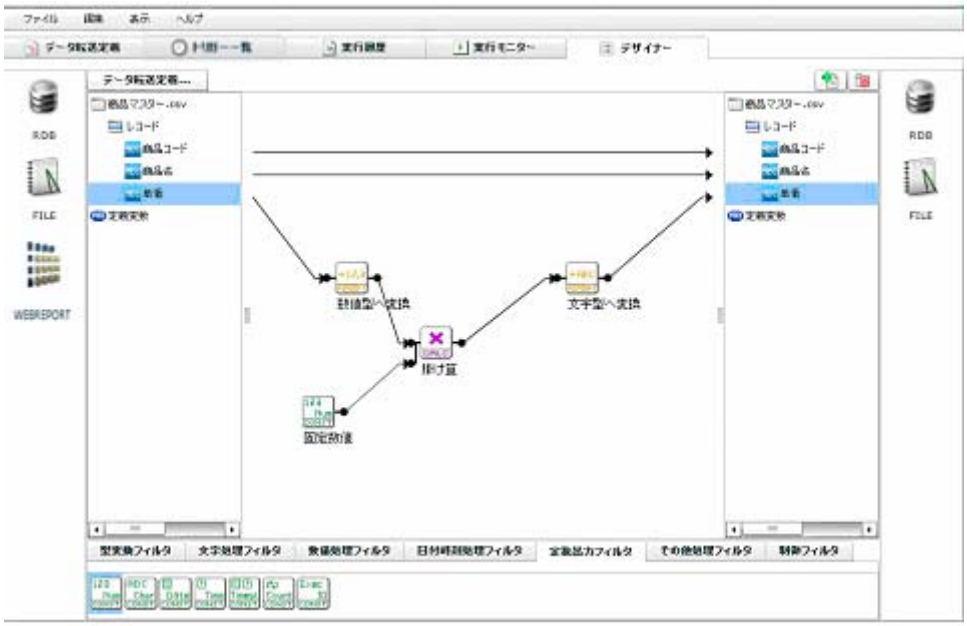
手順１：　データ転送定義を選択し編集します

[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。



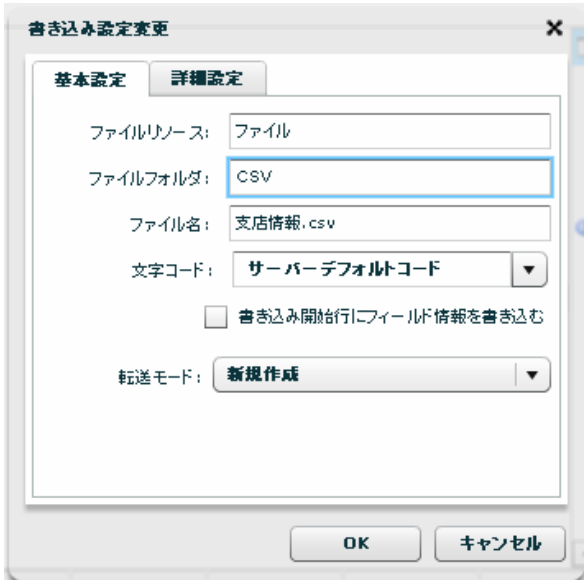
手順 2:　ファイル名を変更します

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。

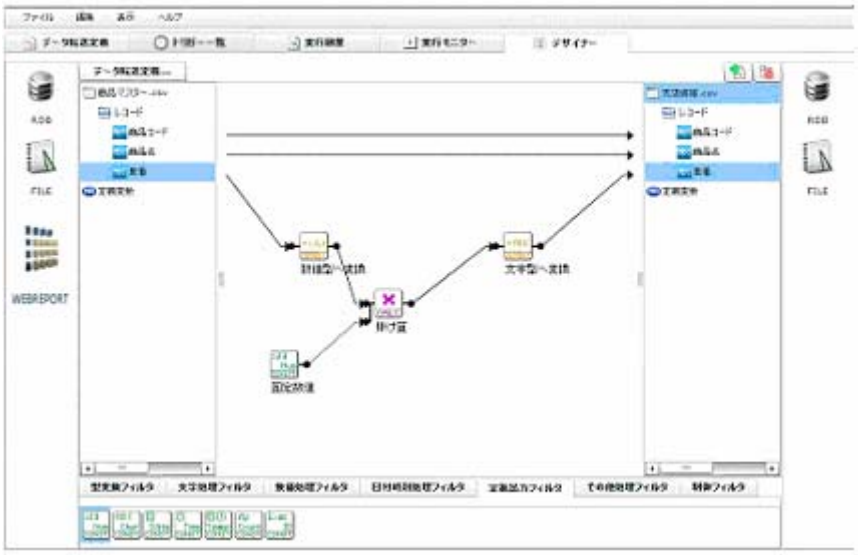


書き込み設定変更画面が表示されます。

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。

[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

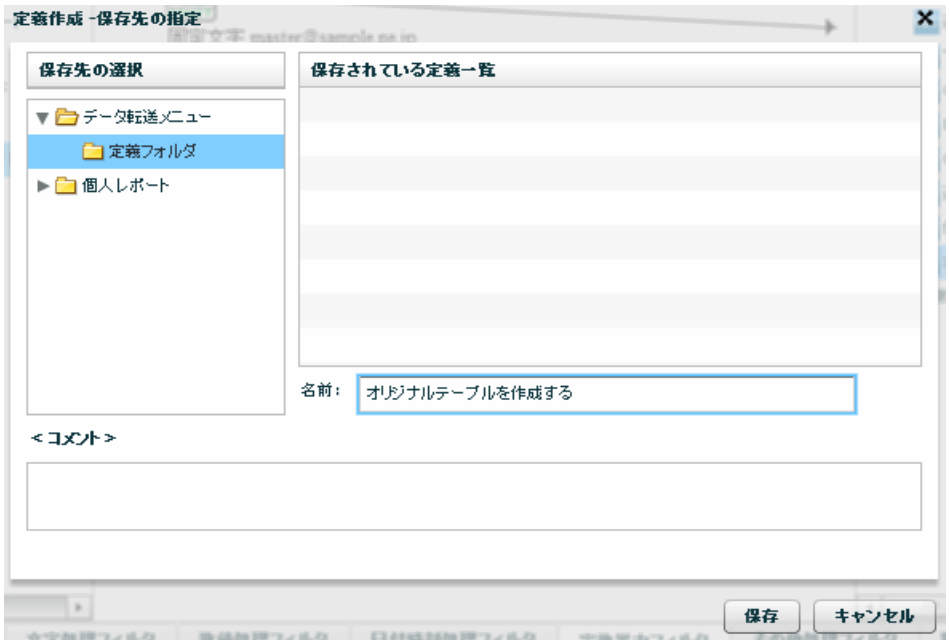
→コマンド実行の結果、入力した任意の値が返された場合は「障害」と判断されます。

[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ことでしょう。

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
・タスク定義を作成する ・スケジュール実行を作成する	・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

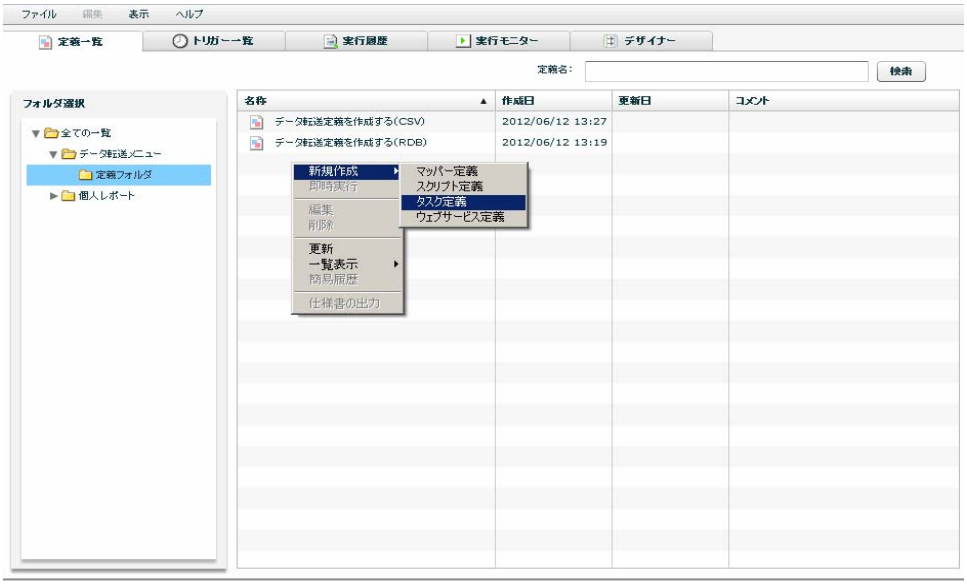
まず、二つのデータ転送定義を含めたタスク定義を一つ作成します。

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。



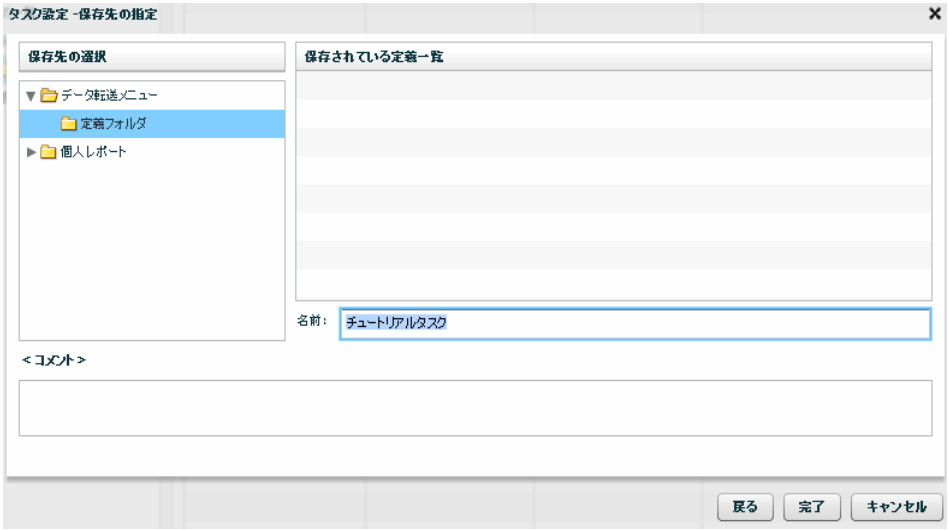
タスク定義作成では、まとめたい定義を保存先フォルダから選択し[追加]ボタンをクリックします。

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。

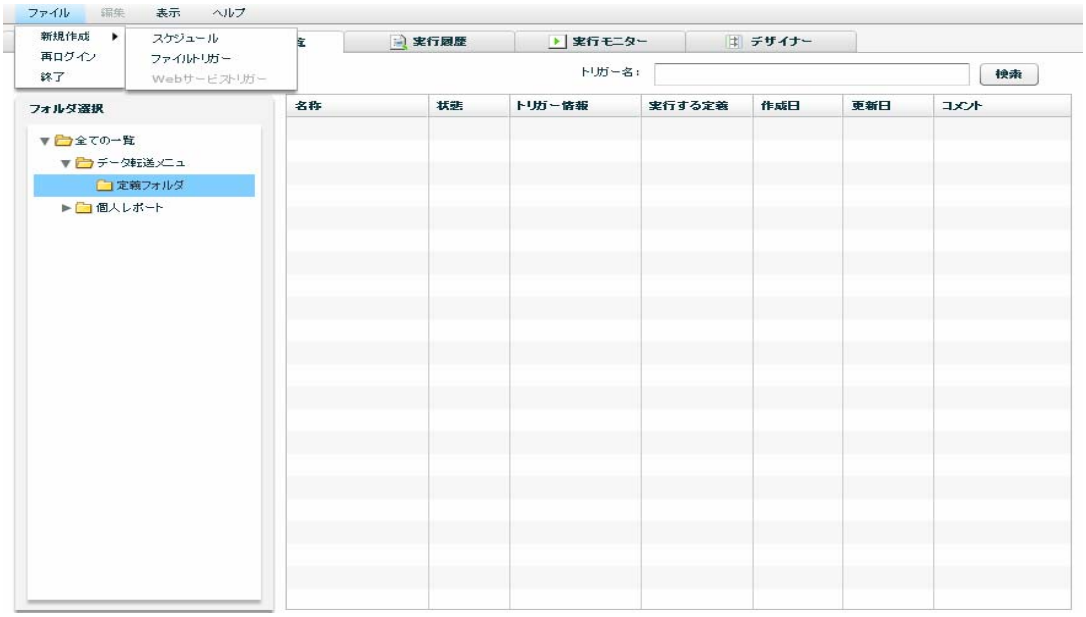


次に、作成した『チュートリアルタスク』をスケジュール実行します。

手順２：　スケジュール定義を作成します

[スケジューラー]タブを開きます。

画面の空白箇所で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。

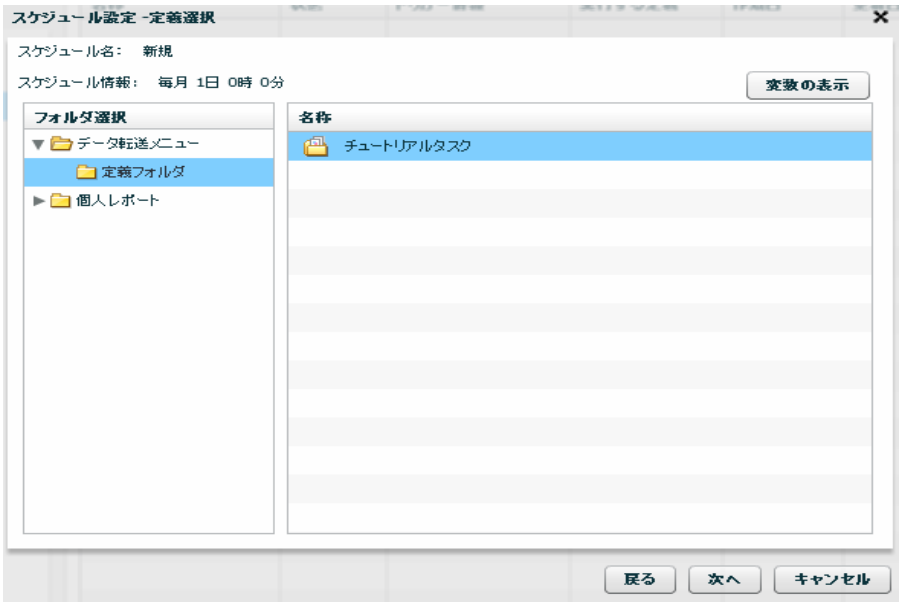


実行タイプ画面では自動実行するタイミングを指定します。

この実習では『毎月１日の午前０時』を指定します。

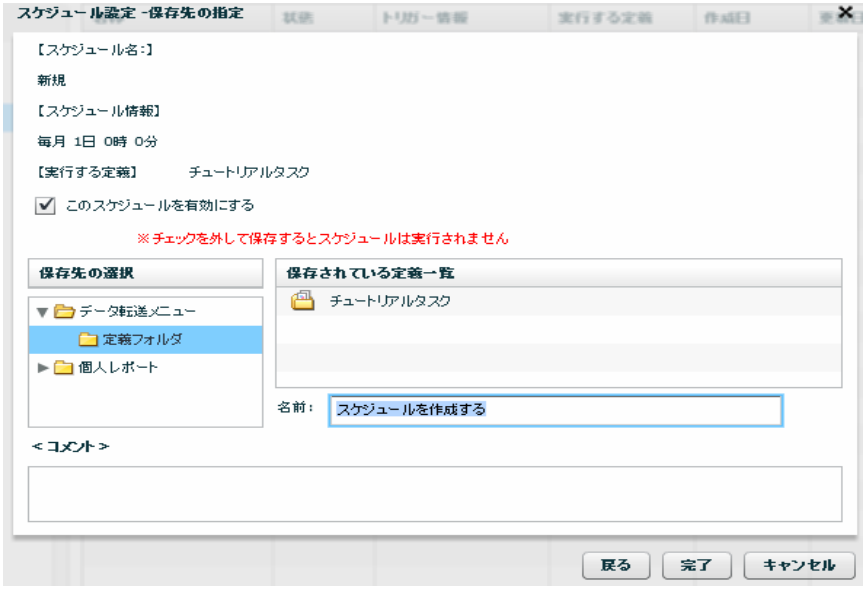
[次へ]をクリックします。

実行させたい定義を選択し、[次へ]をクリックします。

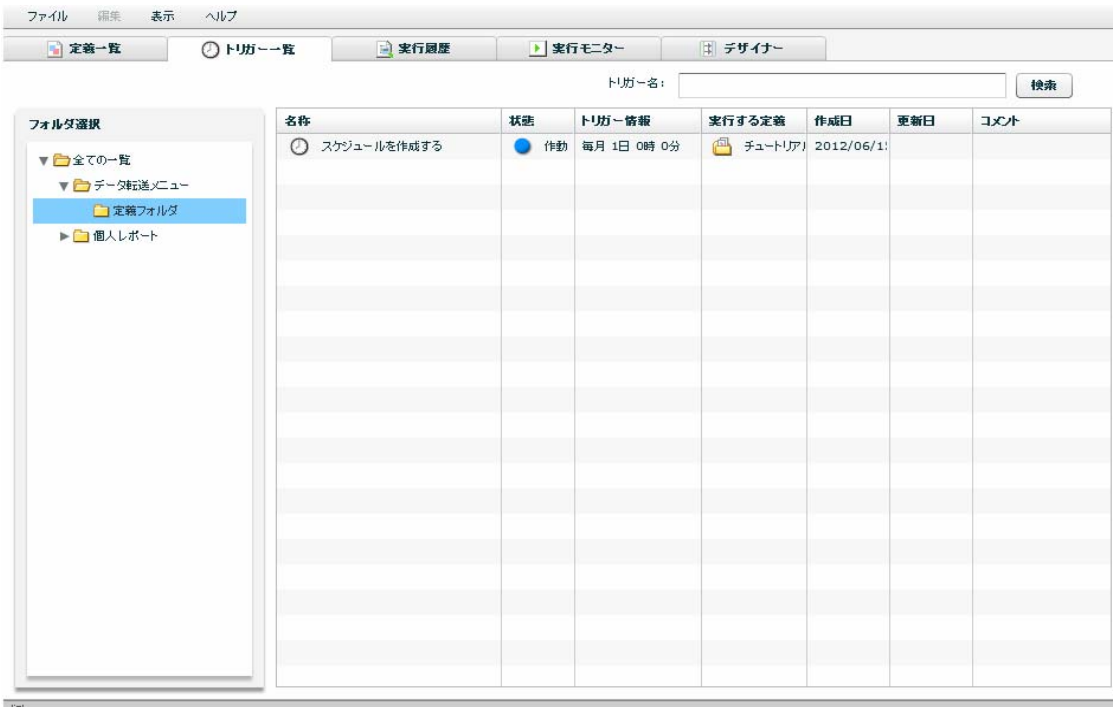


スケジュール名を入力し保存先を指定します。

[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、スケジューラー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。

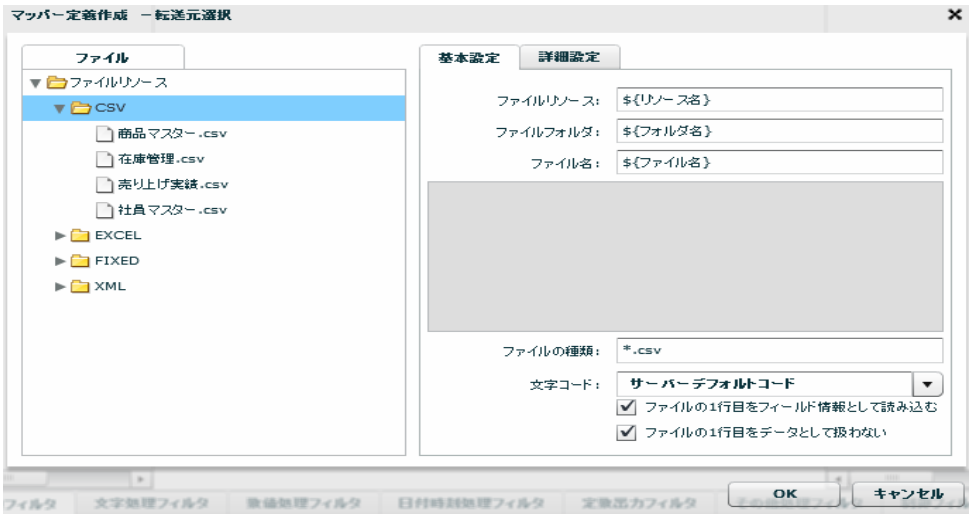
本実習では、ファイルトリガーが感知したファイルのリソース名・フォルダ名・ファイル名をトリガー変数に指定し、それを実行定義に渡すことで動的な処理を実現します。

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。

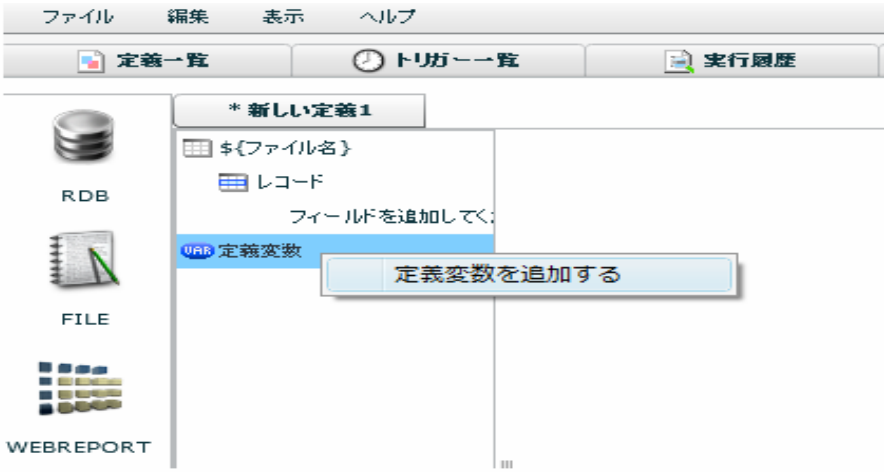
今回は簡単の為、リソース名、フォルダ名、ファイル名という名前の定義変数を使用します。

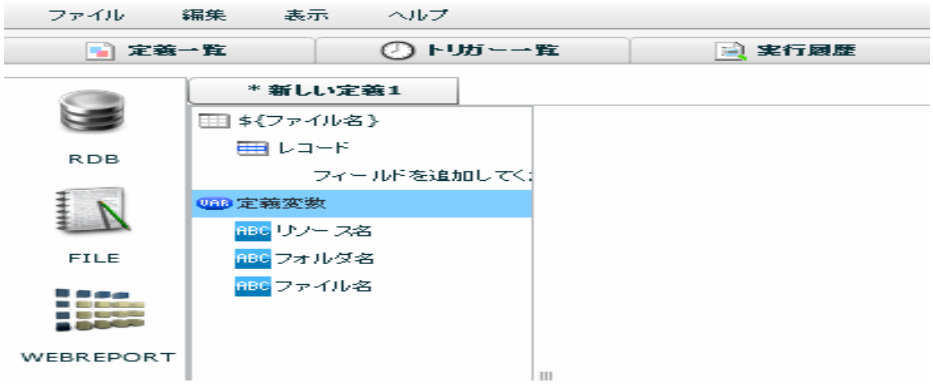
（※定義変数を使用する場合${定義変数名}と表記します。）



[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。

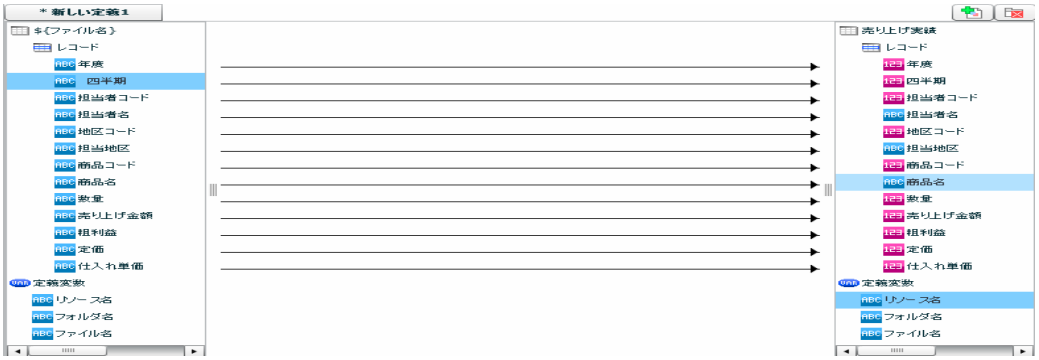
今回はリソース名、フォルダ名、ファイル名の３つを追加します。





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。

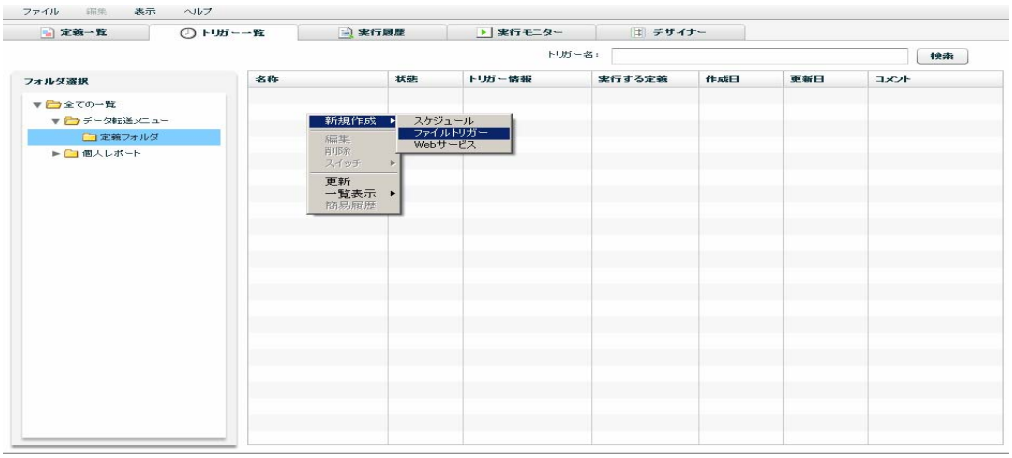


定義の作成ができましたので保存します。（例では売上更新定義という名前で保存）

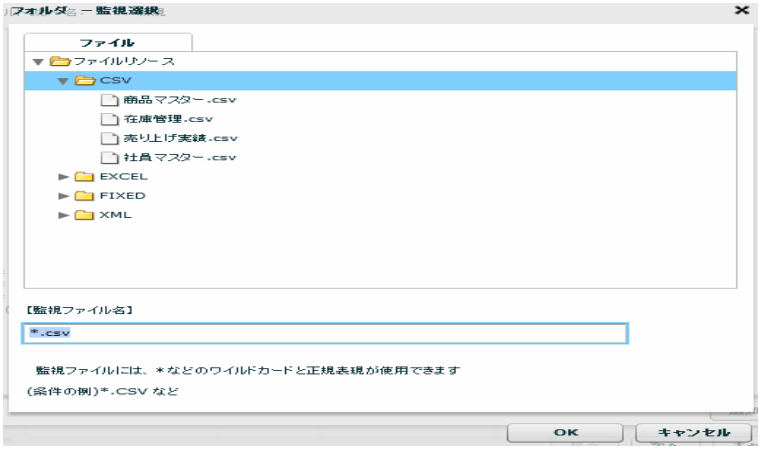
手順２：　ファイルトリガーを作成します

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。

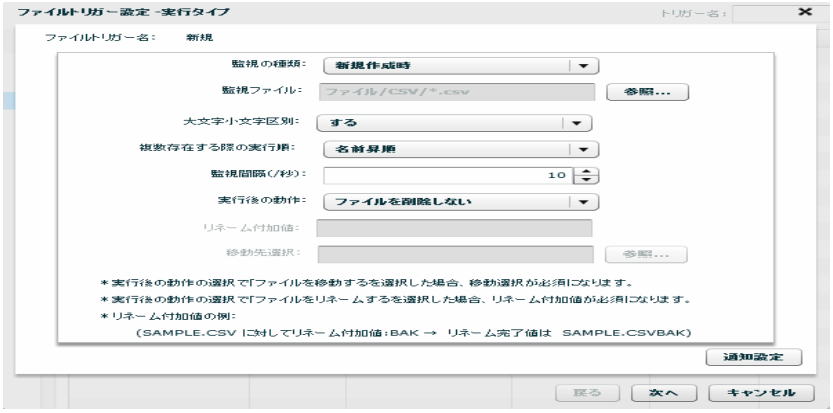
[トリガー一覧]タブから[新規作成]－[ファイルトリガー]を選択してください。



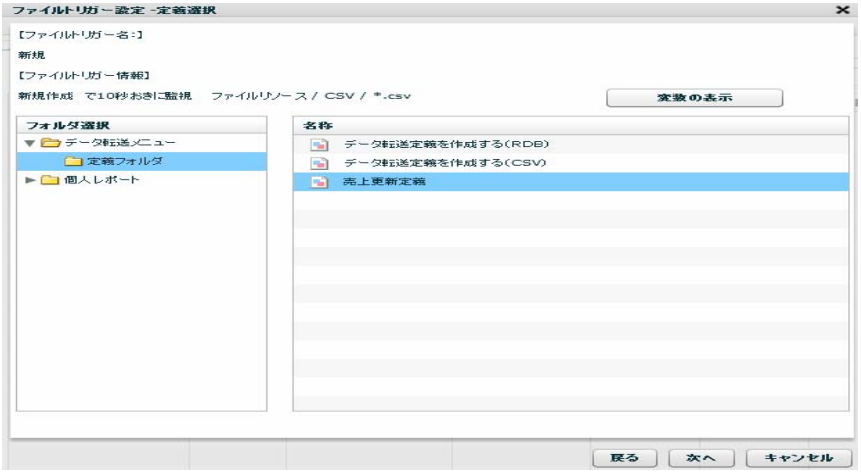
監視ファイルで[参照]をクリックし監視するフォルダを選択しファイル名を*.csv とします。



設定が終了したら[次へ]をクリックします。

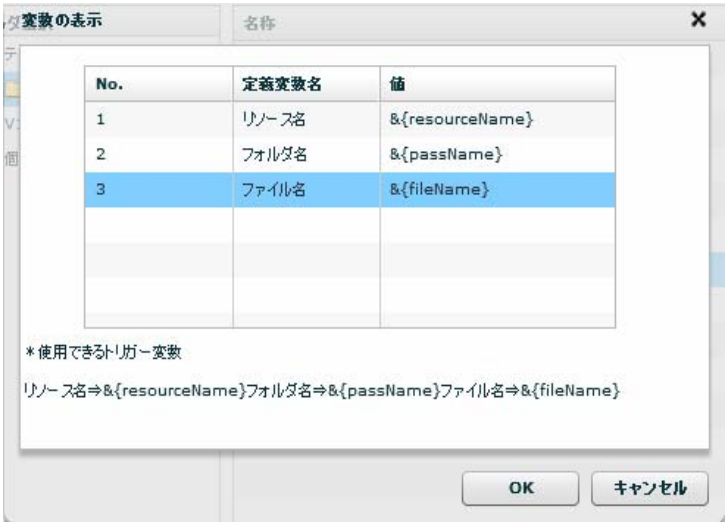


実行する定義には手順１で作成した転送定義を選択します。



[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

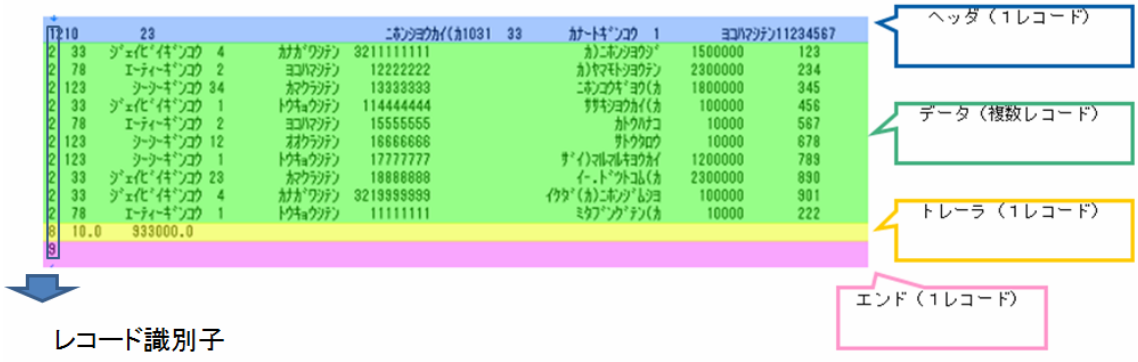
２． データベース ⇒ 全銀協の固定長ファイルへの出力

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

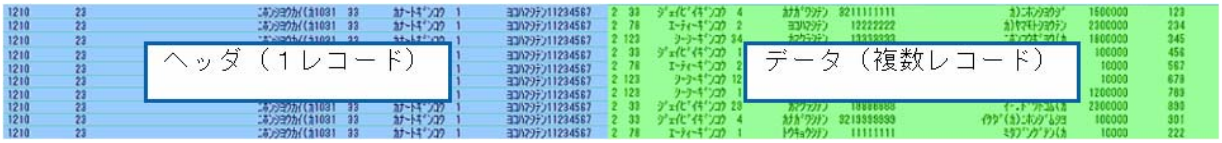
■全銀協の固定長ファイル（振込データ）

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。



■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。



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	ヘッダーレコード:1
2	種別コード	2	半角数字	総合:21、給与:11、賞与:12
3	コード区分	1	半角数字	省略可（スペース）
4	依頼人コード	10	半角数字	当金庫の指定するコード
5	依頼人名	40	半角カナ	省略可（スペース）
6	振込指定日	4	半角数字	月日の４桁（MMDD）で指定
7	仕向金融機関番号	4	半角数字	省略可（スペース）
8	仕向金融機関名	15	半角カナ	省略可（スペース）、左詰スペース埋め
9	仕向支店番号	3	半角数字	-
10	仕向支店名	15	半角カナ	省略可（スペース）、左詰スペース埋め
11	預金種目（依頼人）	1	半角数字	普通:1、当座:2
12	口座番号（依頼人）	7	半角数字	-
13	ダミー	17	半角英数字	スペース

■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	ヘッダーレコード: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

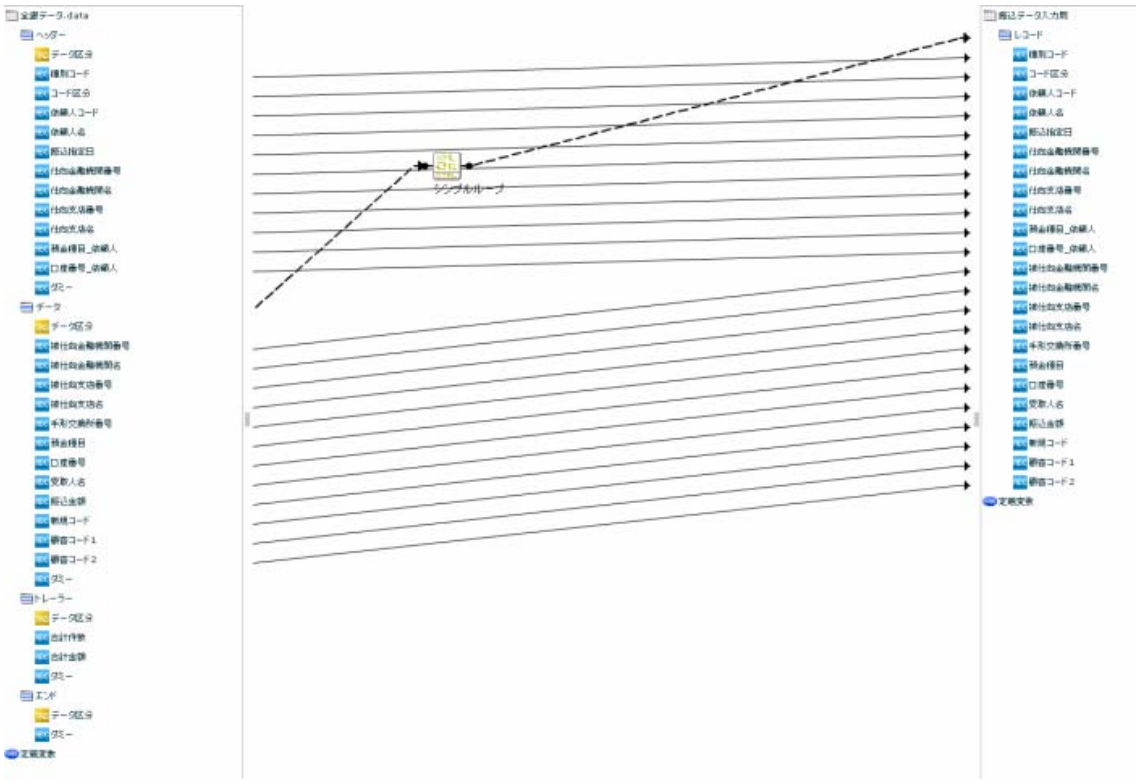
■トレーラ

項番	項目名	桁数	文字形態	備考
1	データ区分	1	半角数字	ヘッダーレコード: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

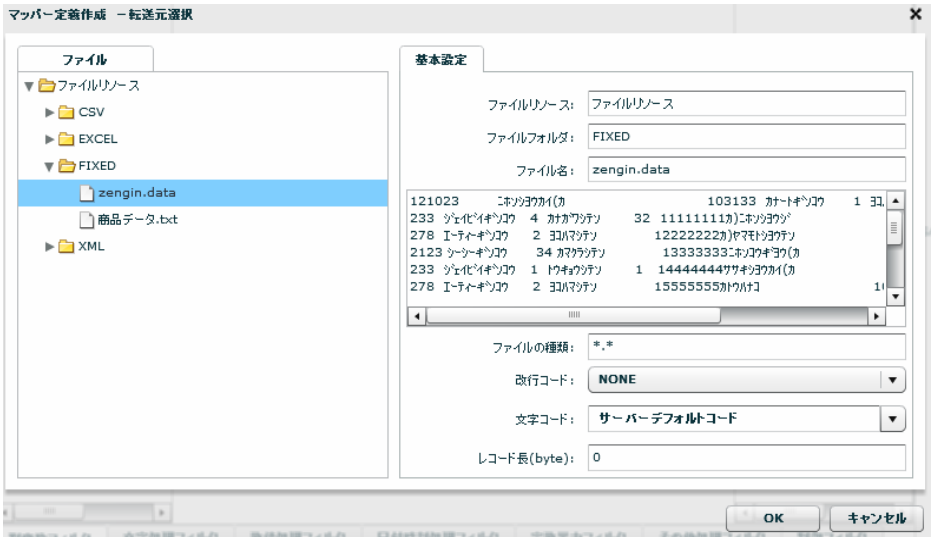
項番	項目名	桁数	文字形態	備考
1	データ区分	1	半角数字	ヘッダーレコード:9
2	ダミー	119	半角英数字	スペース

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：

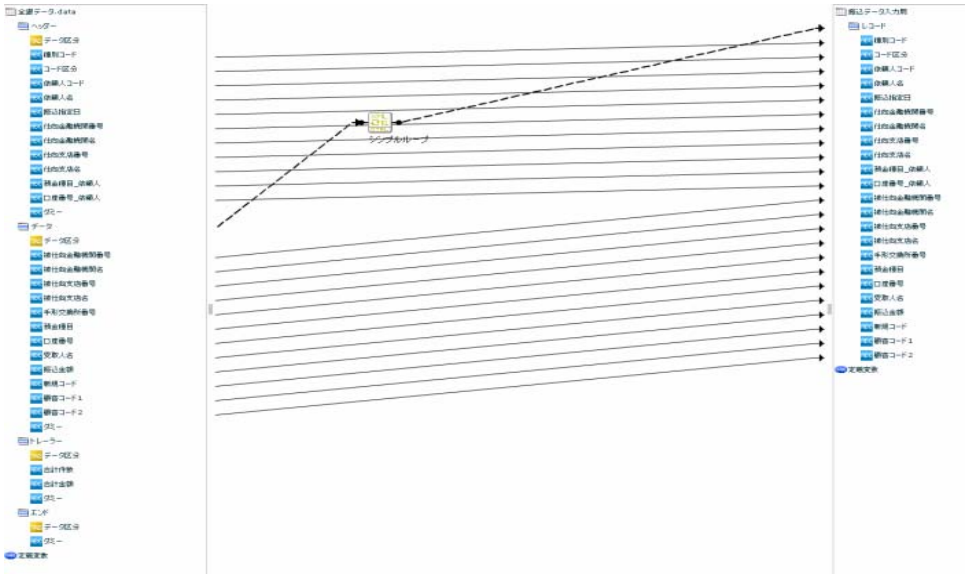
【フィールドタイプ】に【文字列】を選択して、【揃え】に【左揃え】、【パディング文字】に【半角スペース】を設定、【フィールド長（byte）】には指定サイズを設定

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。

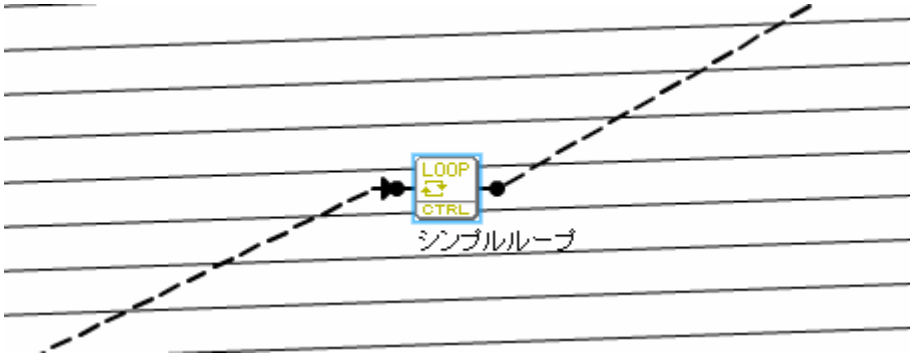
３．　マッピング



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。

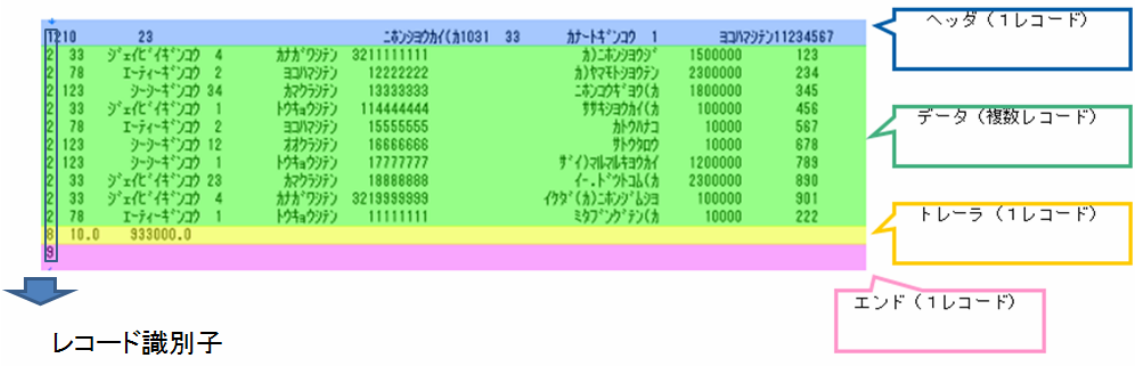


以上で、固定長ファイル（振込データ）のデータベースへの取り込みの設定は終了です。定義を保存して実行してみてください。

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**

転送先全銀協ファイル　※**全銀データ**.data



「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。**

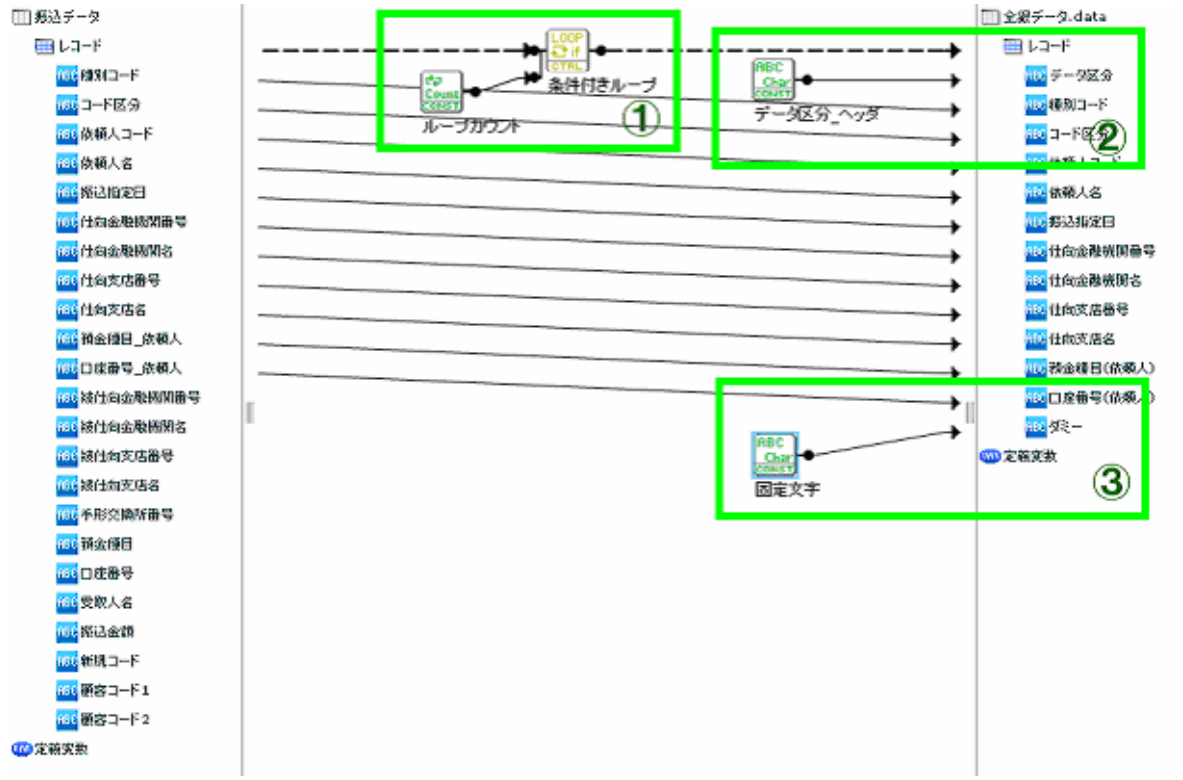
・全銀データ_データ作成：データベースの全レコードからデータ情報を抜き出し、「全銀データ.data**」に追記します。データ情報から「合計件数」と「合計金額」を計算し、定義変数に渡します。**

・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

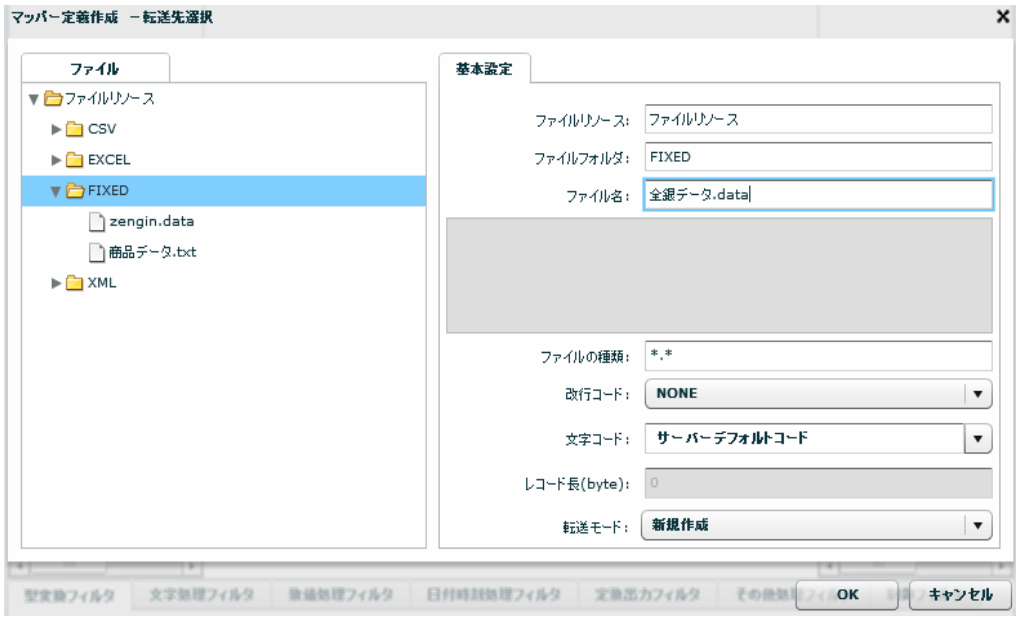


転送元には事前に用意してあるデータベースの「振込データ」を選択します。転送先には全銀協の固定長ファイル「全銀データ.data」を指定し、レコードパターンに全銀協のレコードフォーマットの「ヘッダ」部分を用意します。

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：

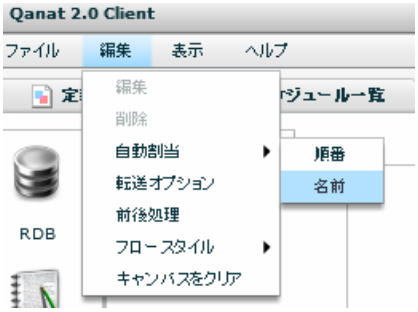


転送先に固定長ファイルの「全銀データ.data」を指定します。

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

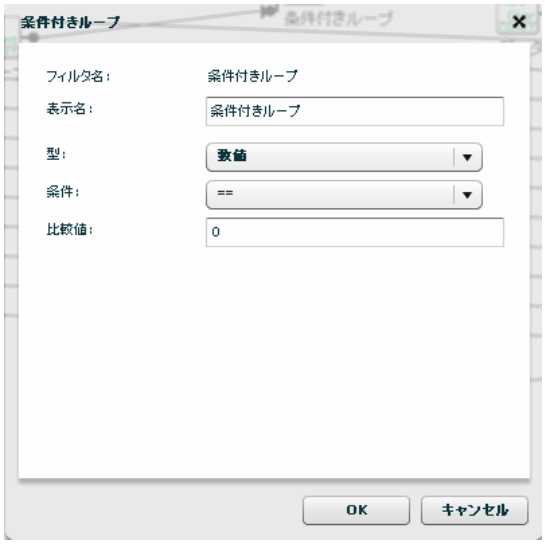


同名フィールドを結び付けます

(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。

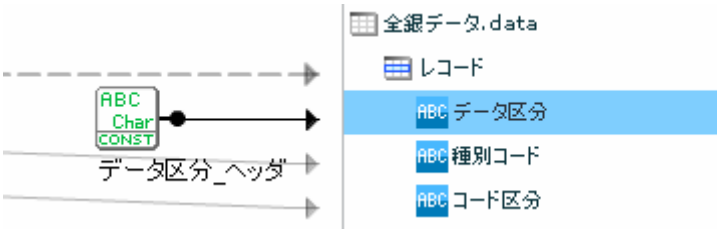
「条件付ループ」でループカウントが 0 の、1 レコード目のヘッダ情報のみ出力するようにします。

1 レコード目のヘッダ情報のみ出力



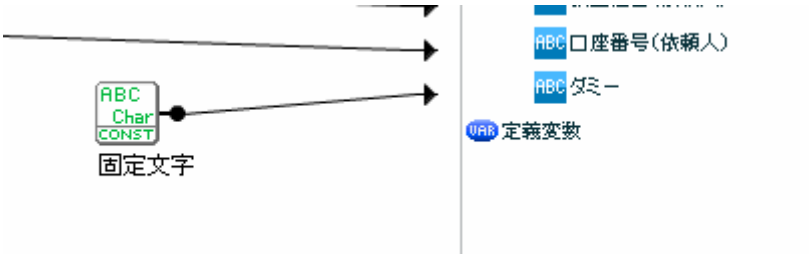
条件付きループで 1 レコード目のみ出力させる

(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。



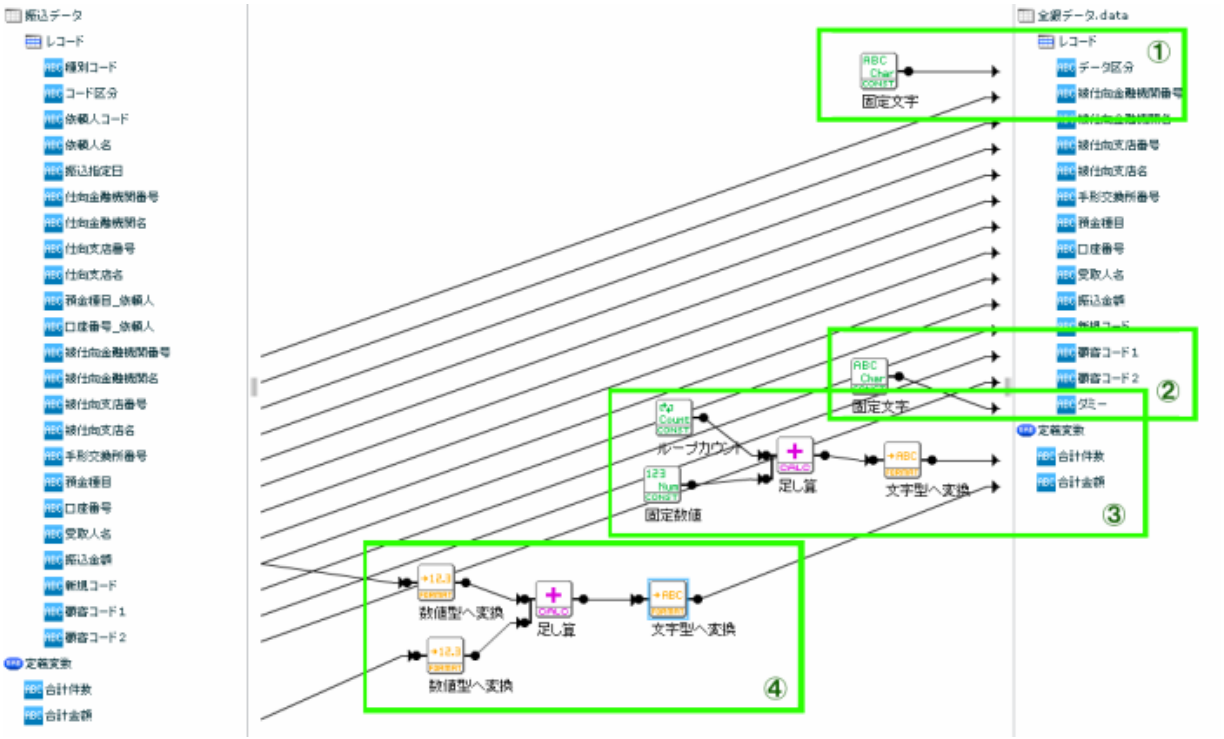
データ区分は固定で 1 を出力

(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。

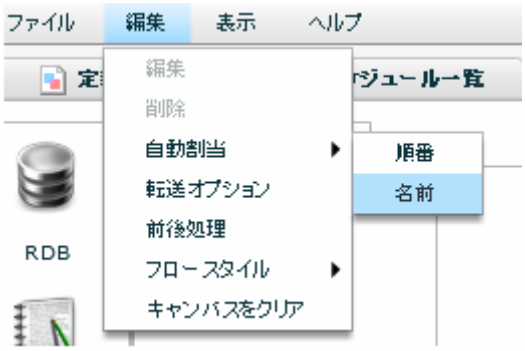
転送モードは【追加書き込み】としてください。

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

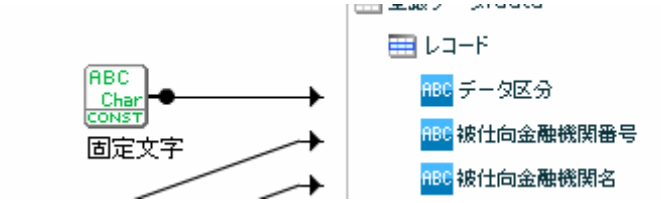
定義変数

「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。



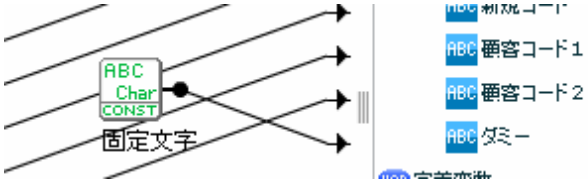
同名フィールドを結び付けます

(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

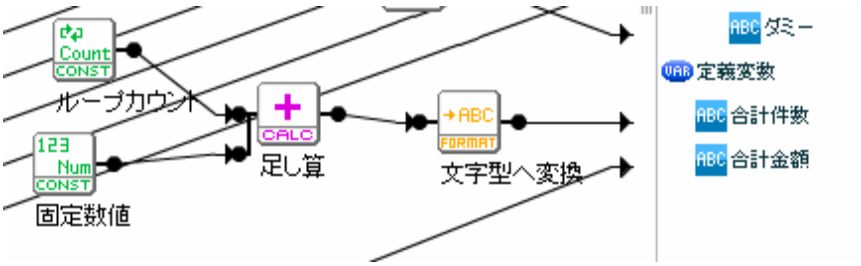
(2)ダミーに半角スペースを１文字指定します。



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

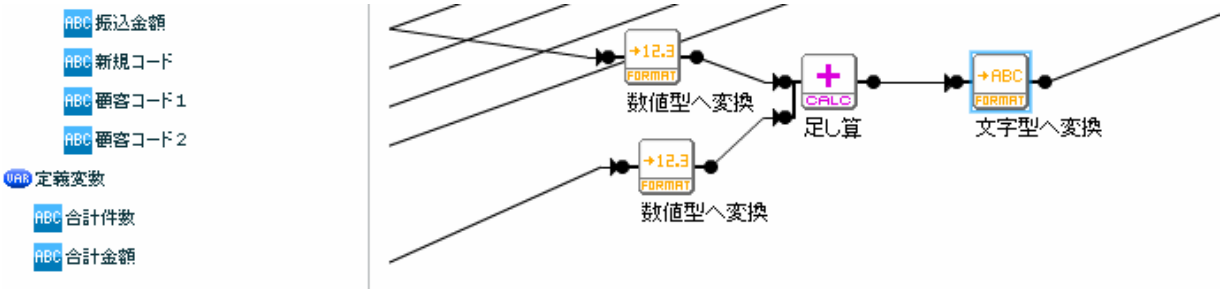
(3)レコード数を定義変数の「合計件数」に設定します。ループカウントを使用していますが、ループカウントは 0 スタートなので固定数値フィルターに 1 を設定してプラスして出力しています。

「合計件数」の値は次のトレーラ出力で出力します。

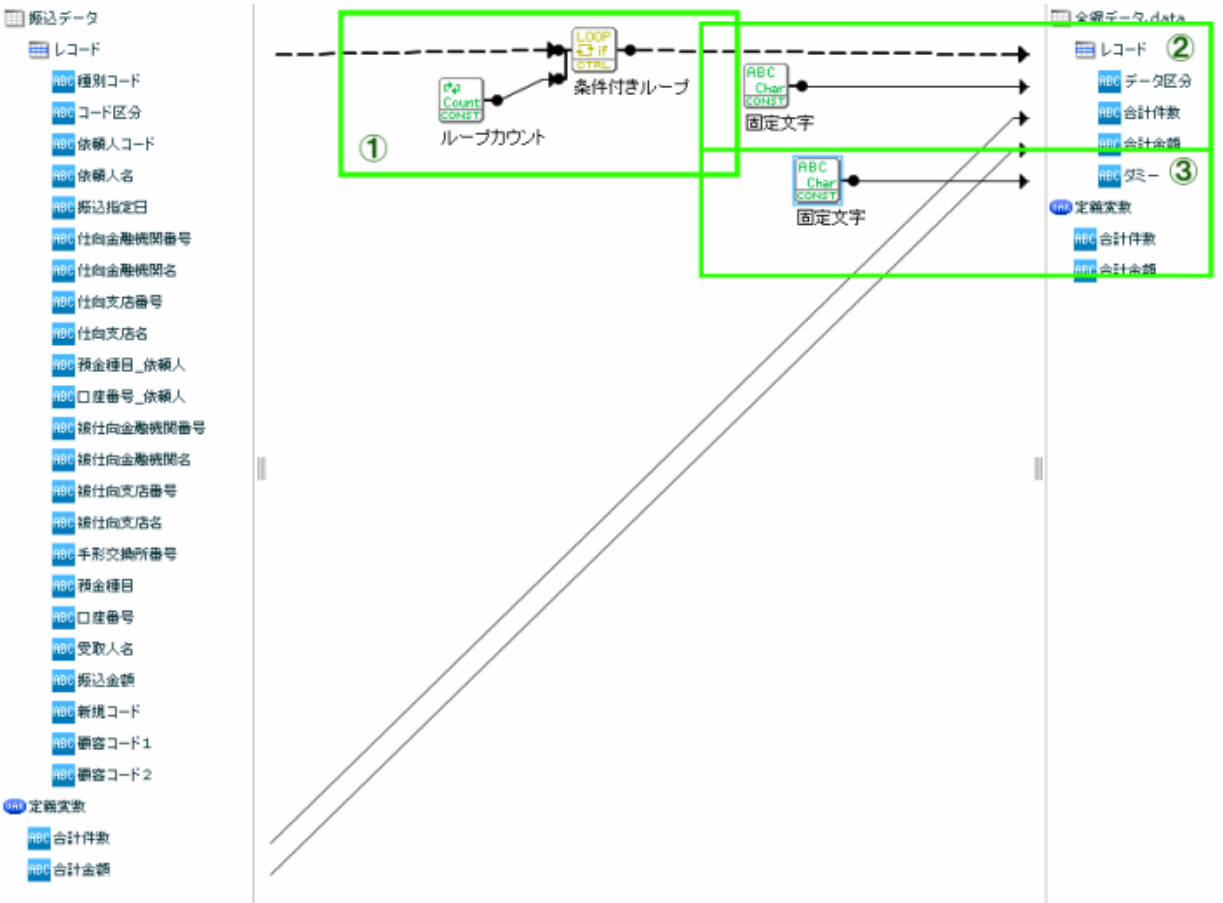


(4)全レコードの合計振込金額を定義変数の「合計金額」に設定します。転送元データベースの「振込金額」を定義変数の「合計金額」とたし合わせて、定義変数の「合計金額」に出力します。「合計金額」のデフォルト値は「0」です。

「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

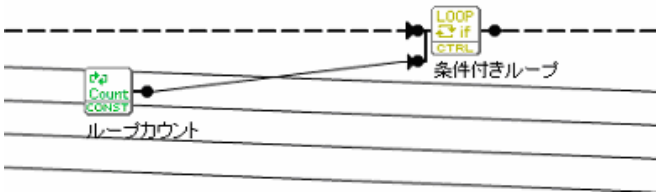
定義変数

「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。

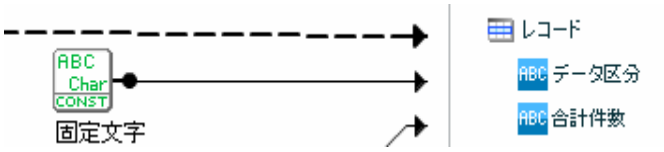
「条件付ループ」でループカウントが０の、１レコード目のヘッダ情報のみ出力するようにします。



１レコード目のヘッダ情報のみ出力

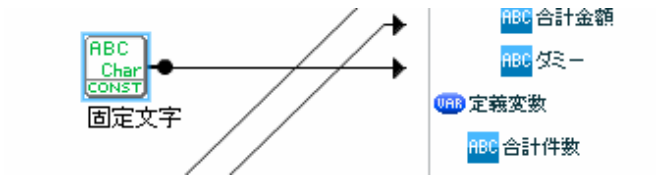
条件付きループで１レコード目のみ出力させる

(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。



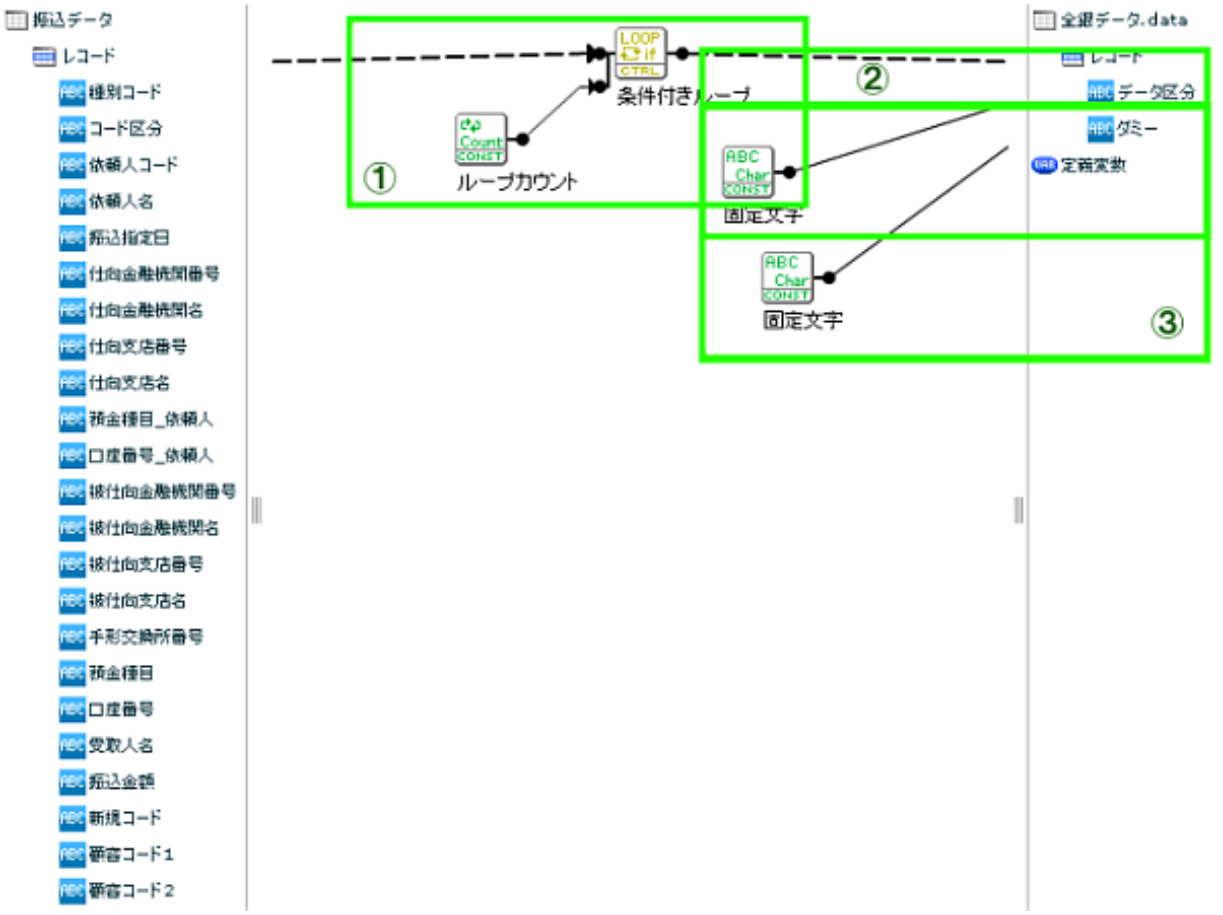
データ区分は固定で８を出力

(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

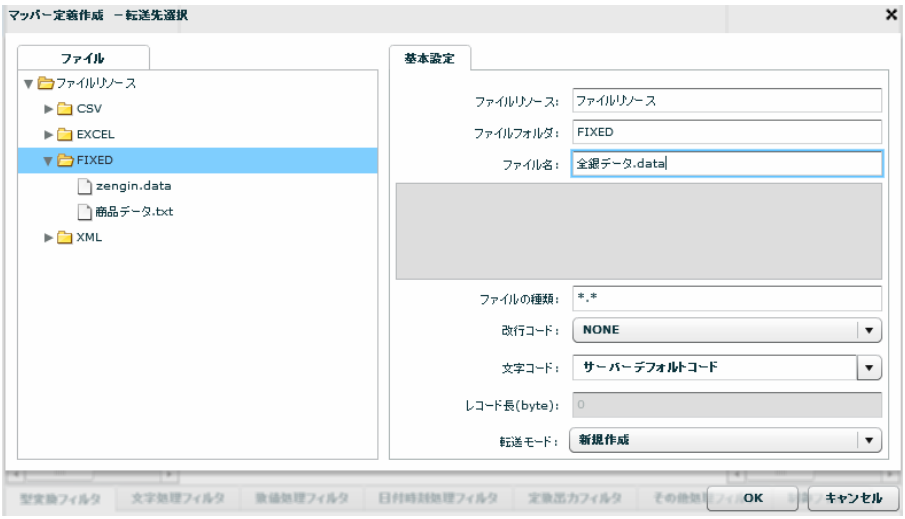


転送元には事前に用意してあるデータベースの「振込データ」を選択しますが、出力するのは「データ区分」への「9」のみになるため、転送元はどのリソースでも問題はありません。レコード転送先には全銀協の固定長ファイル「全銀データ.data」を指定し、レコードパターンに全銀協のレコードフォーマットの「エンド」部分を用意します。

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。

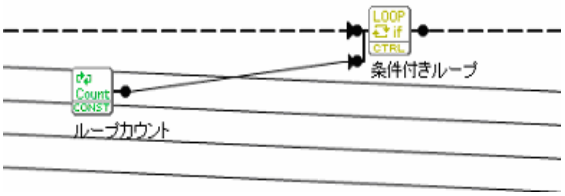
転送モードは【追加書き込み】としてください。

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。

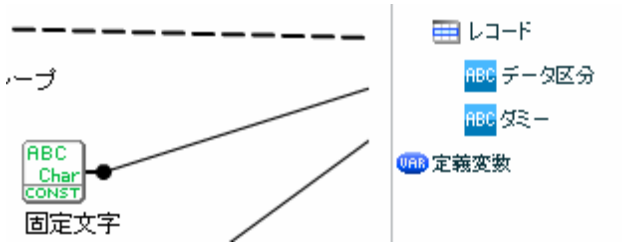
「条件付ループ」でループカウントが 0 の、1 レコード目のヘッダ情報のみ出力するようにします。



1 レコード目のヘッダ情報のみ出力

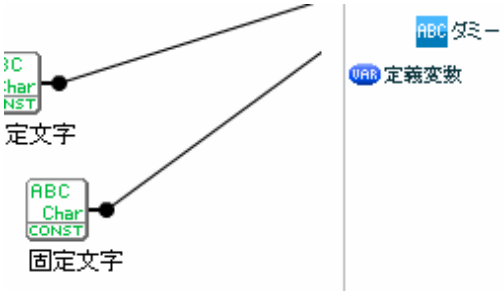
条件付きループで 1 レコード目のみ出力させる

(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。

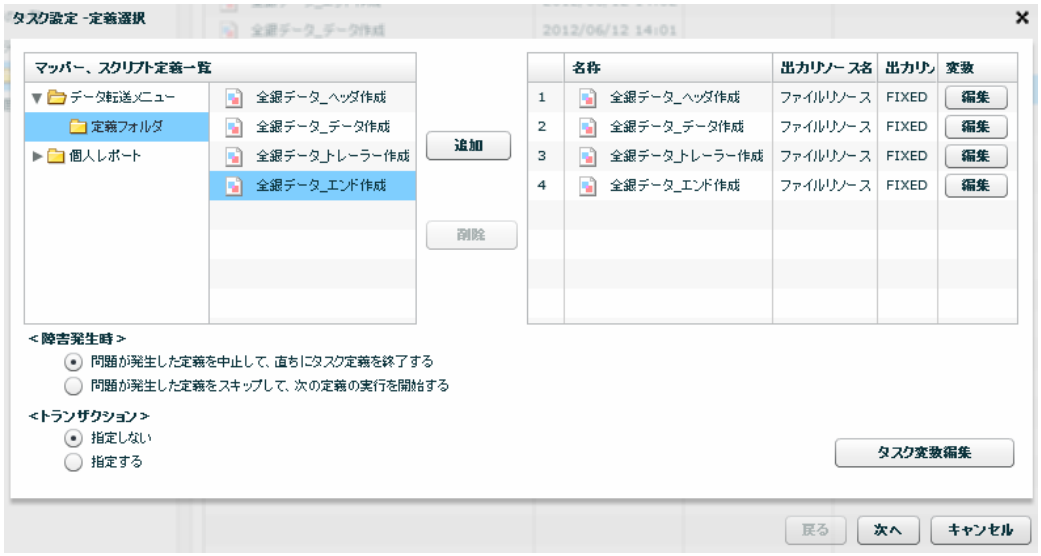


半角スペースを固定文字フィルターに設定してダミーに出力

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。

【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。

「全銀データ_データ作成」の【変数】-【編集】をクリックします。

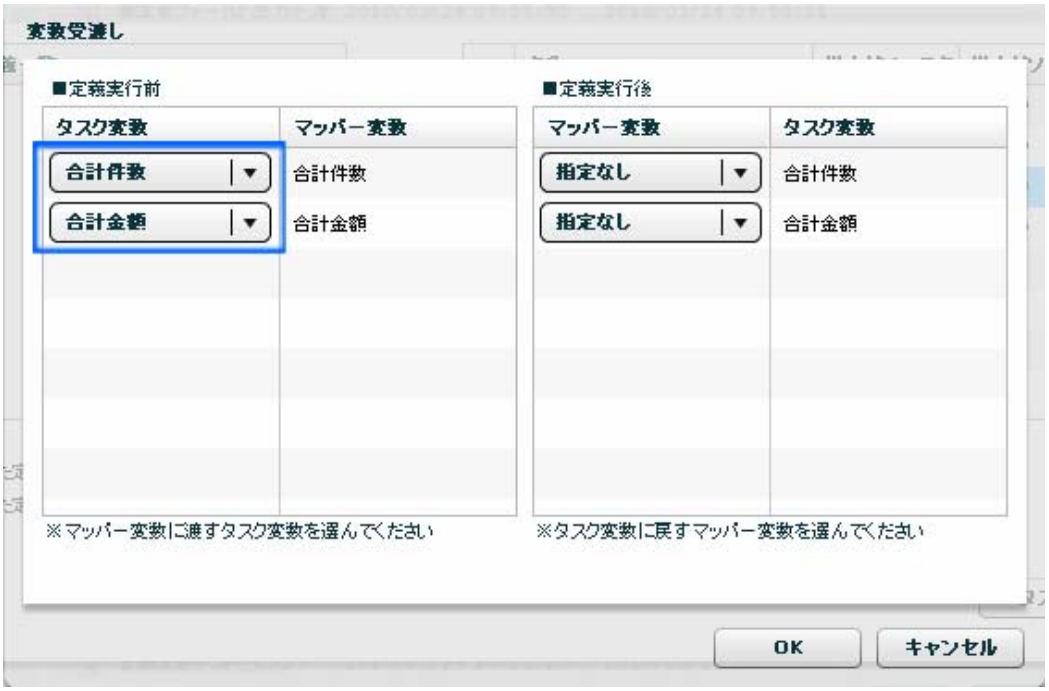
定義実行後に「全銀データ_データ作成」の集計されたマッパー変数「合計件数」、「合計金額」の値をそれぞれ、先ほど設定したタスク変数に渡します。

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。

「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。



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。

Datamart for WebReport2.0　チュートリアル

第 1 版　　2009 年　7 月 29 日

第 2 版　　2012 年 12 月 14 日

発行　JB アドバンスト・テクノロジー株式会社

〒221-0022　神奈川県横浜市神奈川区守屋町 3 丁目 9 番地 C 号ビル

お問い合わせ　　弊社ホームページより、お問い合わせください。
http://www.jbat.co.jp/

資料番号：CVWRR 50U-02