



Datamart
for WebReport 2.0

ユーザースマニュアル

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。

また、WebReport2.0は蓄積されたデータをWebブラウザから簡単に照会できる汎用検索ツールです。

本書は、基本的な操作方法、注意点などについて記載しております。ご使用の際にお読みください。

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。

版權 / 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

 注意	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、必ず守ってください
	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
第 1 章 システムの概要	1
1.1 WebReport と Datamart for WebReport 2.0	2
1.1.1 データマートとは	3
1.1.2 データマートを活用するメリット	3
1.2 Datamart for WebReport 2.0 のユーザー	4
1.3 各種転送定義とデータ転送ツールの関係	5
1.4 運用の流れ	6
第 2 章 操作方法	7
2.1 共通操作	8
2.1.1 ログイン	8
2.1.2 初期画面	11
2.1.3 機能メニュー	12
2.1.4 メニューバー	13
2.1.5 エラー	16
2.2 定義一覧	17
2.2.1 データ転送定義とは	17
2.2.2 メニューバー	17
2.2.3 機能メニュー	18
2.2.4 定義一覧画面について	19
2.2.5 利用可能メニュー	20
2.2.6 タスク定義設計画面	23
2.2.7 仕様書の出力	26
2.3 トリガー一覧	28
2.3.1 スケジュールとは	28
2.3.2 ファイルトリガーとは	28
2.3.3 トリガー一覧画面について	29
2.3.4 利用可能メニュー	30
2.3.5 スケジュール作成画面	33
2.3.6 ファイルトリガー作成画面	35
2.4 実行履歴	39
2.4.1 実行履歴とは	39

2.4.2	実行履歴画面について	39
2.4.3	利用可能メニュー	41
2.4.4	詳細表示画面	42
2.4.5	リストの操作	43
2.5	実行モニター	44
2.5.1	実行中とは	44
2.5.2	実行モニター画面について	44
2.5.3	利用可能メニュー	45
2.6	デザイナー	46
2.6.1	データ転送定義とは	46
2.6.2	デザイナー画面について	47
2.6.3	共通メニュー	48
2.6.4	リソース選択	51
2.6.5	RDB 選択	52
2.6.6	RDB 作成・変更	62
2.6.7	RDB を利用したときの各種メニュー	65
2.6.8	CSV ファイル選択	73
2.6.9	CSV ファイルを利用したときの各種メニュー	77
2.6.10	固定長ファイル選択	85
2.6.11	固定長ファイルを利用したときの各種メニュー	88
2.6.12	固定長ファイル転送におけるループ処理	100
2.6.13	XML ファイル選択	102
2.6.14	XML ファイルを利用したときの各種メニュー	105
2.6.15	XML ファイル転送におけるループ処理	114
2.6.16	前後処理	115
2.6.17	フロースタイル	118
2.6.18	フィルタのプロパティ	120
2.6.19	定義保存	121
2.6.20	テストログダイアログ - テスト実行	122
2.6.21	即時実行	123
2.6.22	ファイル転送	124
2.6.23	複数出力	125
2.7	スクリプト定義	126
2.7.1	スクリプト定義とは	126
2.7.2	使い方	126
2.8	定義変数	129
2.8.1	定義変数とは	129
2.8.2	使い方	129
2.9	QanatExecute	136
2.9.1	QanatExecute とは	136
2.9.2	使い方	136
付録 A フィルタリファレンス		138
A	フィルタリファレンス	1

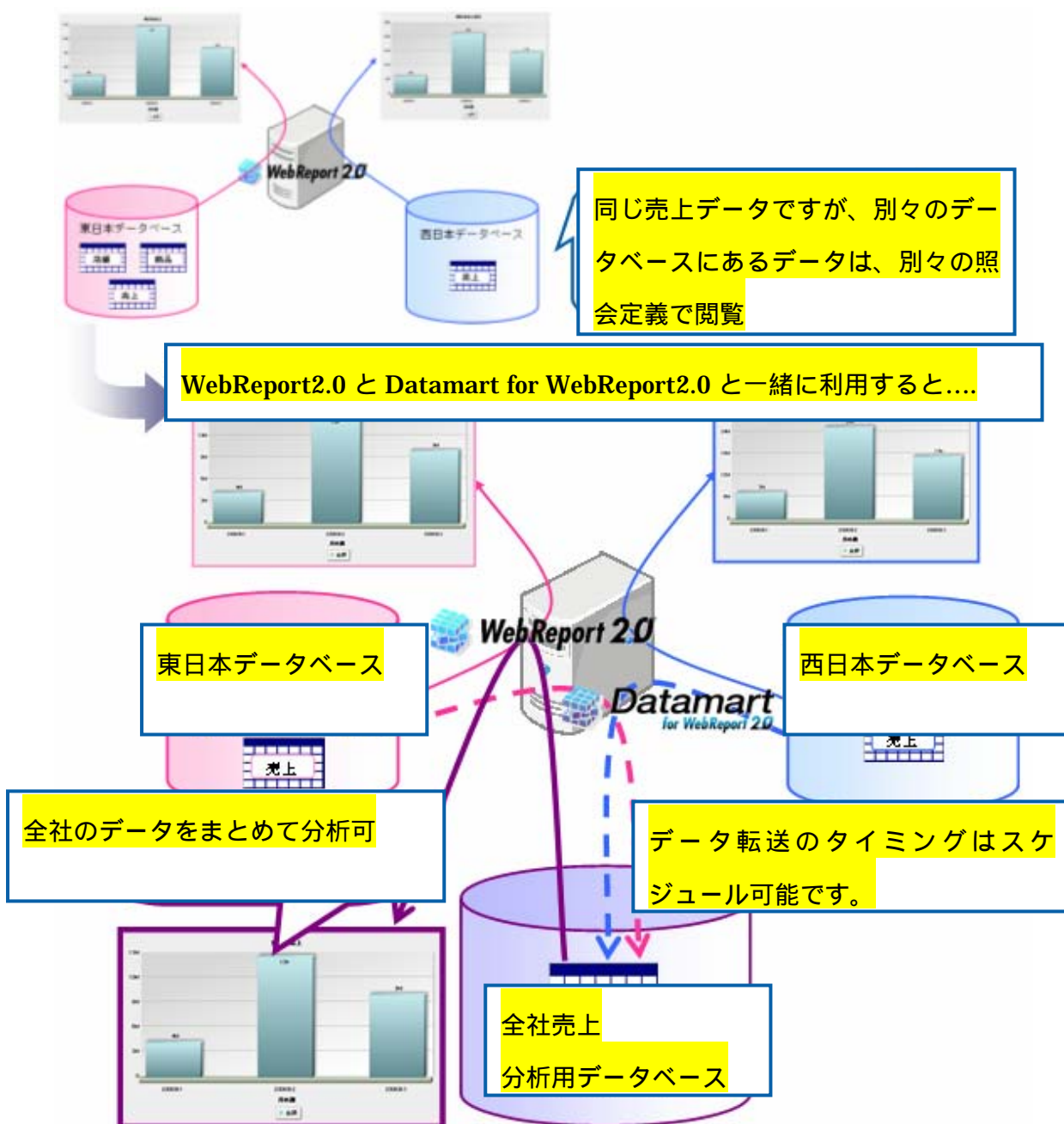
A-1	型変換フィルタ	2
A-2	文字変換フィルタ	9
A-3	数値処理フィルタ	15
A-4	日付時刻処理フィルタ	19
A-5	定数出力フィルタ	24
A-6	その他処理フィルタ	28
A-7	制御フィルタ	39
付録 B 対応データ型		41
B	対応データ型	1
B-1	文字型	1
B-2	数値型	2
B-3	日付型	3
B-4	時刻型	3
B-5	日付時刻型	3
付録 C 便利な機能		214
C	便利な機能	215

第 1 章

第1章 システムの概要

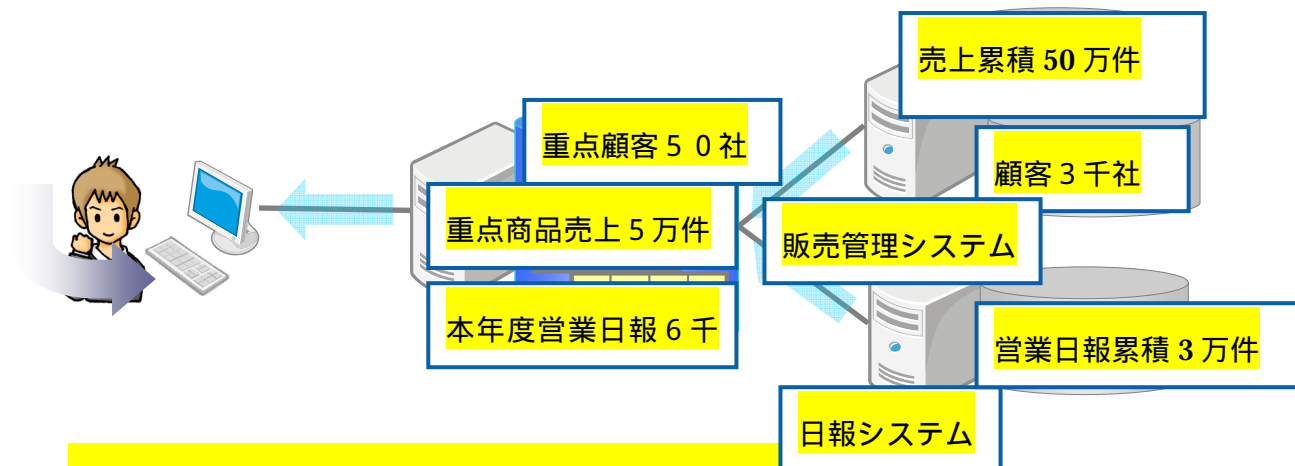
1.1 WebReport と Datamart for WebReport 2.0

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 Web ブラウザから簡単に照会できるデータベース照会機能に加え、より高度な分析を実現するため、データマートにデータを集めるためのデータ転送機能もあわせ持っています。



1.1.1 データマートとは

・情報活用ツールで扱いやすい形式で、データを収集しておくための照会専用のデータベースです。



本製品はさまざまなデータをこのデータマートに転送するための機能を提供します。

データ転送元とデータ転送先、その実行スケジュールを設定することで、データ転送が可能です。

1.1.2 データマートを活用するメリット

- ・データの収集時に、あらかじめ必要な情報だけを集計してから転送することにより、照会時に極力計算処理を抑えることができ、表示のレスポンスを向上することができます。
- ・これまで手作業で時間がかかっていた、異なる種類のデータを分析するためのデータ収集作業が自動化できます！

1.2 Datamart for WebReport 2.0 のユーザー

本製品のユーザーは、以下の3種類に分けることができます。

システム管理者 (データ転送定義者)

レポート設計者

レポート照会者

システム管理者

システム管理者は、照会に使用されるデータベースの接続情報の定義や、Web 照会画面の表示項目の設定を行います。データ転送ツールを使用して、以下の定義を作成します。

データベース定義の作成

データベース定義 (照会に使用されるデータベースの接続情報、ログイン情報の定義) を作成します。

スキーマ定義の作成

スキーマ定義 (データベースのどのテーブルを照会に使用するかという定義) を作成します。

テーブル/フィールド定義の作成

テーブル/フィールド定義 (スキーマ定義に含めるテーブル/フィールド情報の定義) を作成します。

メニュー定義の作成

メニュー定義 (Web 照会画面の以下の表示項目) を作成します。

照会メニュー (Web 照会画面のタブ)

照会サブメニュー (タブ選択時に表示されるフォルダ)

データ転送定義の作成

転送元テーブル または 照会定義、転送先テーブル間のデータ転送方法を設定します。

データ転送スケジュールの設定

データ転送定義の実行タイミングをスケジュールリングします。

レポート設計者

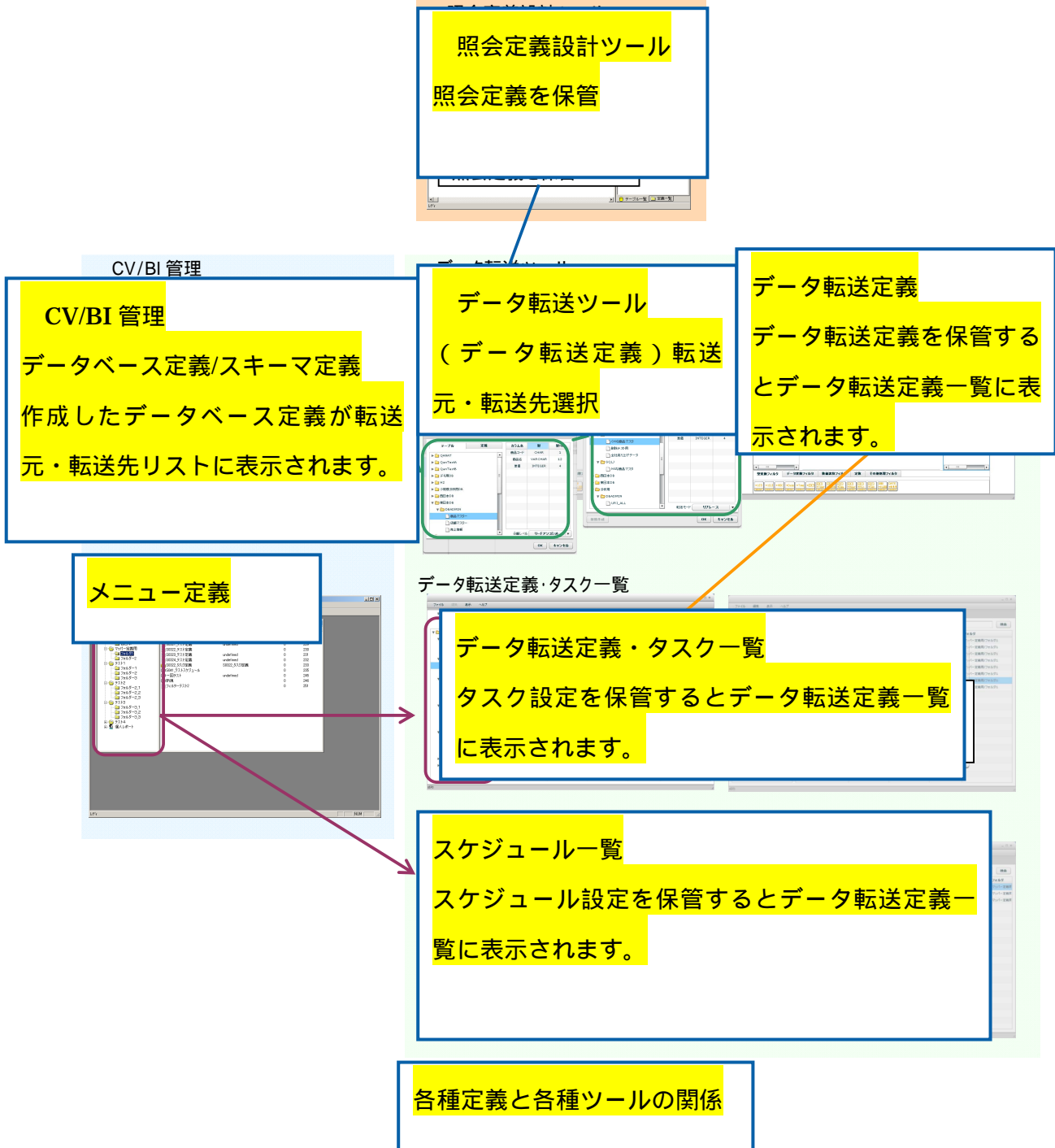
照会定義 (どのデータベースからどのような条件でデータを読み込み、表示するかの定義) を作成します。

レポート照会者

Web 照会画面で照会メニューを選択し、データを照会します。

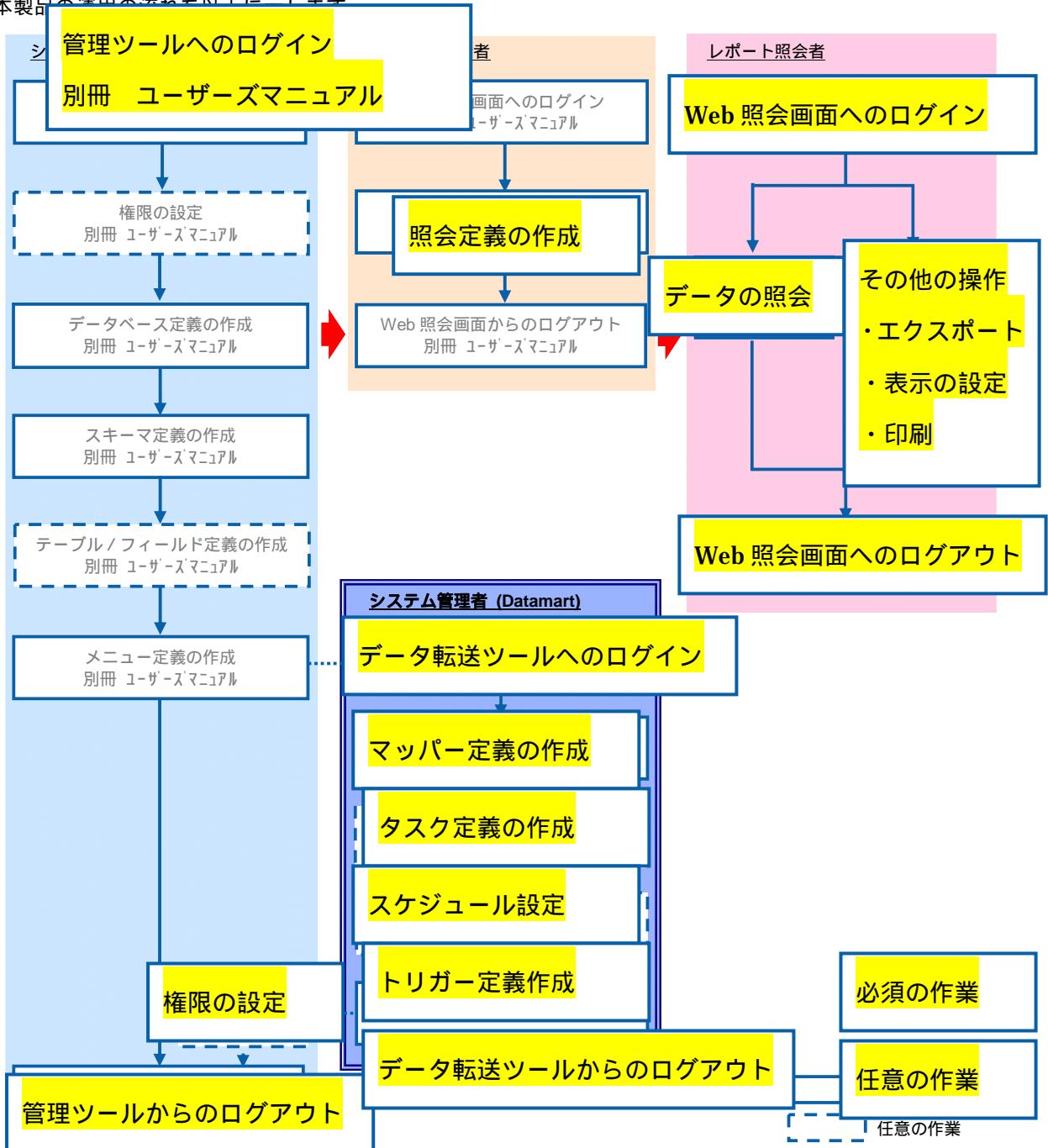
1.3 各種転送定義とデータ転送ツールの関係

システム管理者が作成する各種定義とデータ転送ツールの関係について、以下に示します。



1.4 運用の流れ

本製品の運用の流れを以下に示します。



運用の流れ
 このマニュアルでは、データ転送ツール部分に関連する部分の説明を行います。
 全体的な利用方法に関しては、別冊の WebReport 2.0 ユーザーマニュアルを参照してください。

第 2 章

第 2 章 操作方法

2.1 共通操作

ここでは、本製品へのログイン方法、ログアウト方法について説明します。また、ログインして最初に表示される初期画面についても説明します。なお、データ転送定義の作成には、CV/BI 管理ツールも必要です。あらかじめ導入しておいてください。

2.1.1 ログイン

本製品へのログインは、以下の手順で行います。

- 1 [スタート]メニューから [プログラム] - [Customer Vision] - [WebReport] - [DatamartTool] を選択します。



DatamartTool の選択

- 2 [サーバー]、[ユーザー]、[パスワード]を設定し、[ログイン]ボタンをクリックします。

サーバー ログインサーバー情報を入力します。
Server 名 or IP アドレス: (コロン) ポート番号

ユーザー 登録されているユーザー名を半角英数字で入力します。

パスワード 登録されているパスワードを半角英数字で入力します。
入力したパスワードは「*」で表示されます。

言語選択 プルダウンボックスより当製品の表示言語を選択します。



ログイン

Datamart
for WebReport 2.0

Qanatサーバー: 10.40.25.238:6200

ユーザー名:

パスワード:

Japanese ▼

ログイン キャンセル

ログイン画面

定義一覧画面が表示されます。



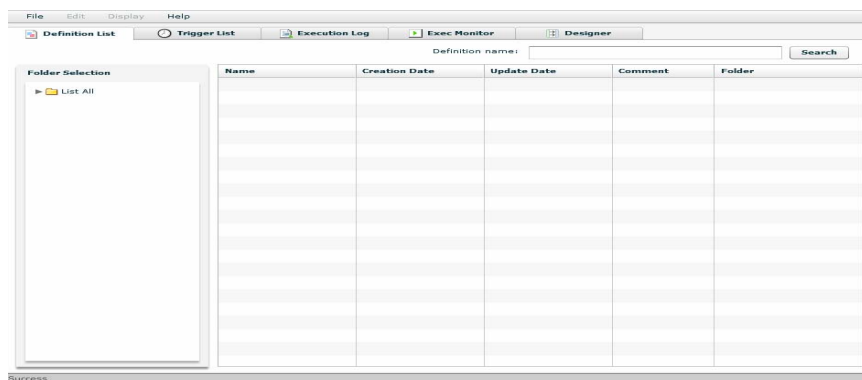
本製品にログインするためには、特別な設定が必要です。

ログインできない場合は、管理者の方にご連絡ください。

言語選択で選ばれた言語によっては本製品に表示される文字フォントに差異があります。



選択言語：Japanese の画面



選択言語：English の画面

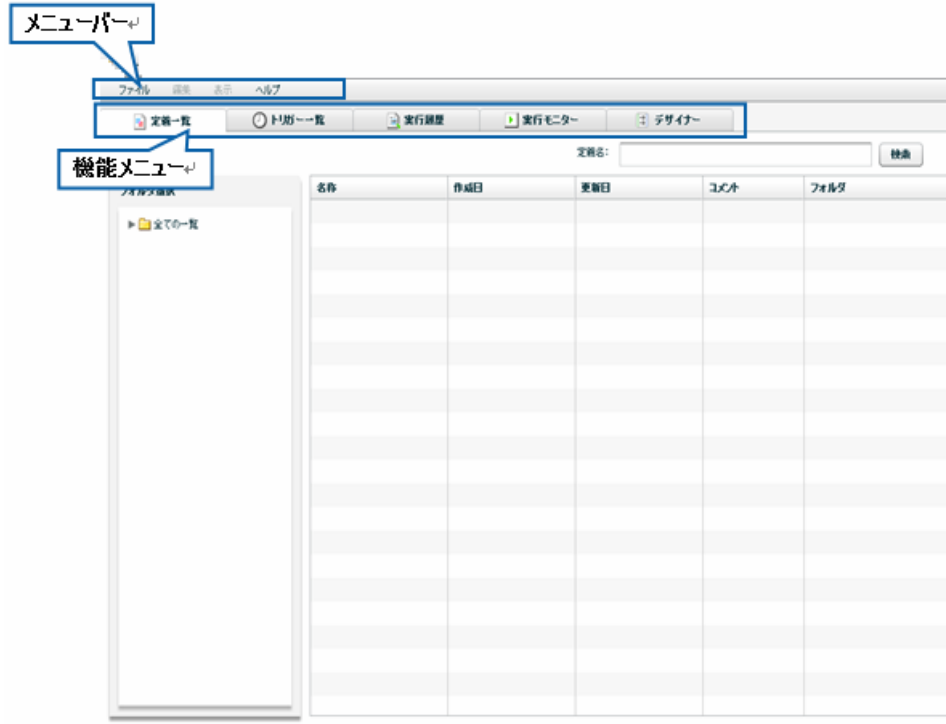


選択言語：Chinese の画面

2.1.2 初期画面

本製品にログインすると、定義一覧画面が表示されます。CV/BI管理で登録した、データリソース定義、スキーマ定義、メニュー定義を利用したデータ転送定義を作成することができます。

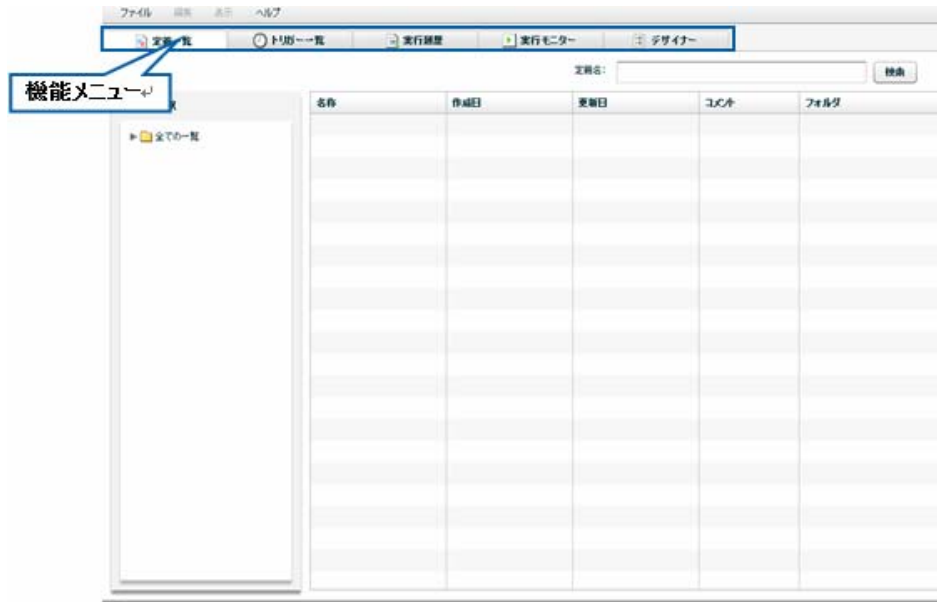
左側にある「フォルダ選択」にて、「全ての一覧」を選択すると、全てのデータ転送定義一覧を表示します。画面左側に「フォルダ」という列が表示され、各定義の保存先フォルダ情報を確認することができます。



定義一覧画面

2.1.3 機能メニュー

機能メニューには [定義一覧]、[トリガー一覧]、[実行履歴]、[実行モニター]、[デザイナー] の5つの機能メニューが表示されます。クリックすることにより、画面が切り替えられます。メニューバーも選択した機能により内容が異なります。



[定義一覧] 機能

データをどのように転送するかを設定するマッパー定義、スクリプト定義と、複数の転送定義を実行順にまとめたタスク定義の管理ができます。

[トリガー一覧] 機能

作成したデータ転送定義の実行トリガーを管理します。

[実行履歴] 機能

トリガーにて実行、または、即時実行された履歴を参照することができます。

[実行モニター] 機能

現在実行されている定義（トリガーによる起動・即時実行ともに）の一覧を表示します。また、権限がある定義は、中止も可能です。

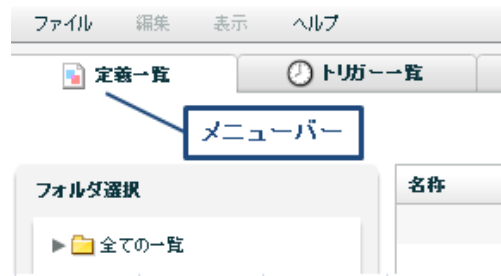
[デザイナー] 機能

データ転送定義をグラフィカルに作成・編集できます。

2.1.4 メニューバー

メニューバーには[ファイル]、[編集]、[表示]、[ヘルプ]の4つのメニューが表示されます。機能メニューごとに利用可能な内容が異なります。ヘルプは共通の内容になります。

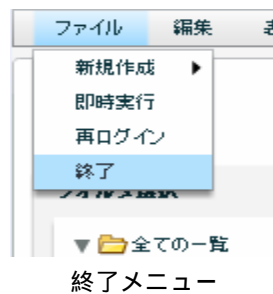
次に共通のメニューバーの機能を説明します。



[終了]

本製品の終了は、以下のいずれかの方法で行います。

- ・メニューから[ファイル] - [終了]を選択する。
- ・画面右上の[×]ボタンをクリックする。



[再ログイン]

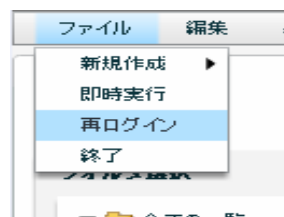
本製品を操作中に、何らかの理由によりサーバーとの接続が切断されたとき、以下のようなエラーが表示されることがあります。

このような場合に、メニューより、再ログインを行うことにより、作業を継続できます。



ログイン未エラー

メニューから[ファイル] - [再ログイン]を選択する。



再ログインメニュー

パスワードのみ入力し、「ログイン」ボタンをクリックします。



再ログイン



再ログインではサーバー、ユーザー名、選択言語は変更できません。

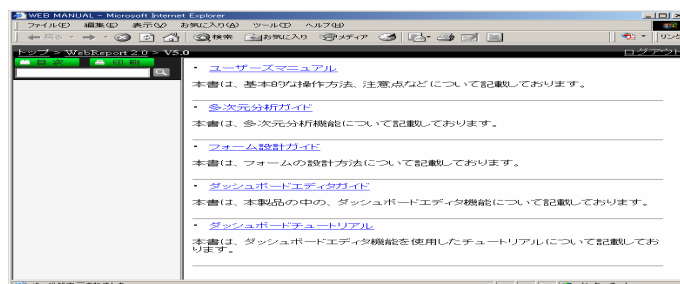
[マニュアル参照]

本製品のどの機能を利用しているときでも、メニューからマニュアル、バージョン情報の参照が可能です。



ヘルプメニュー

メニュー [ヘルプ] - [マニュアル] でマニュアルが参照可能です。

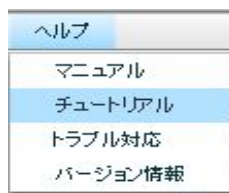


マニュアルサイト



マニュアル参照のためには、インターネット接続が必要です。

メニュー [ヘルプ] - [マニュアル] でマニュアルが参照可能です。



メニュー [ヘルプ] - [チュートリアル] ではチュートリアルが参照可能です。



メニュー [ヘルプ] - [トラブル対応] ではトラブル対応マニュアルが参照可能です。



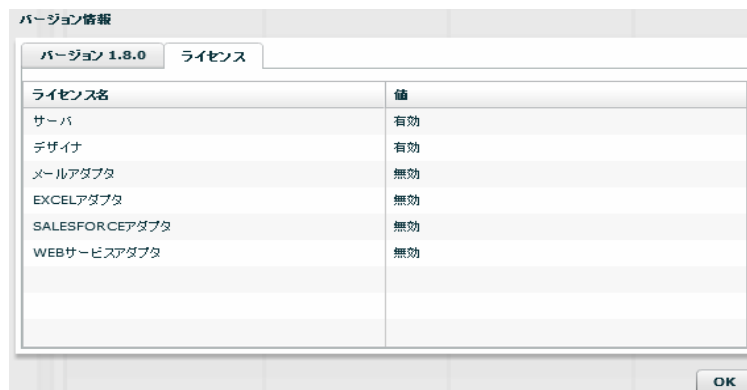
マニュアル、チュートリアル、トラブル対応の参照のためには、インターネット接続が必要です。

[バージョン情報]

メニュー [ヘルプ] - [バージョン情報] でバージョン情報が参照可能です。



バージョン情報



現在、導入されているライセンスの一覧を確認することもできます。



本マニュアルでは例としてバージョン 1.8.0 時の画面イメージを添付しています。実際の表示は導入されているバージョンが表示されます。

2.1.5 エラー

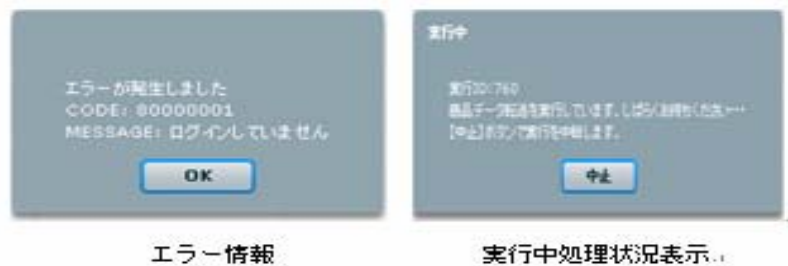
正常な値の場合は、フォーカスが当たっている入力フィールド枠は水色です。

入力値エラーの場合、枠が赤色でフィールドの横にエラーメッセージが表示されます。データ型が異なる場合には入力できません。



また、何らかの処理時にエラーが発生した場合には、画面にエラー情報が表示されます。

さらに、同様の方法で処理中等の現在の処理状況を表示することもあります。



2.2 定義一覧

ここでは、定義一覧画面の機能について説明します。定義一覧では保管されているデータ転送定義を管理できます。

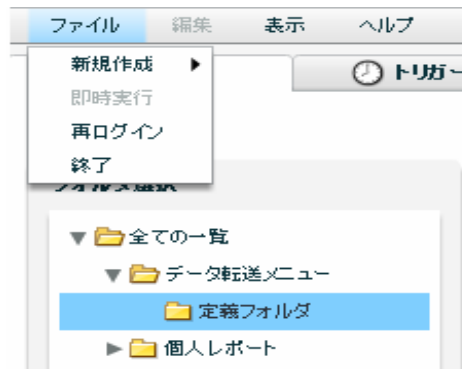
2.2.1 データ転送定義とは

データ転送定義にはマッパー定義、スクリプト定義があり、指定したデータベースのテーブルのデータやファイルリソースのデータを別のテーブルやファイルリソースへ転送することができます。

タスク定義を作成すると、複数のマッパー定義、スクリプト定義を任意の順序で実行することができます。

2.2.2 メニューバー

メニューバーには[ファイル]、[編集]、[表示]、[ヘルプ]の4つのメニューが表示されます。操作中に、利用可能なメニューのみ、利用可能状態になります。



メニューバー ファイルメニュー

[ファイル]メニュー

- | | |
|-------|-------------------------------------|
| 新規作成 | 新規にデータ転送定義を作成します。 |
| 即時実行 | 選択しているデータ転送定義を実行します。 |
| 再ログイン | 何らかの理由によりサーバーとの接続が切断されたとき、再ログインします。 |
| 終了 | 本製品からログアウトし、終了します。 |

[編集]メニュー

- | | |
|----|------------------------|
| 編集 | 表示中の設定画面で選択した項目を変更します。 |
| 削除 | 表示中の設定画面で選択した項目を削除します。 |

[表示]メニュー

- 更新 一覧に表示しているデータを最新状態に表示しなおします。
- 一覧表示 一覧に表示するデータの絞込みをします。
- 簡易履歴 選択された定義の直近の履歴情報のみ表示します。

[ヘルプ]メニュー

- マニュアル インターネット経由で本製品の最新マニュアルを表示します。
- チュートリアル インターネット経由でチュートリアルを表示します。
- トラブル対応 インターネット経由で障害発生時のトラブル対応表を表示します。
- バージョン情報 本製品のバージョン情報を表示します。

2.2.3 機能メニュー

機能メニューには[定義一覧]、[トリガー一覧]、[実行履歴]、[実行モニター]、[デザイナー]の5つの機能メニューが表示されます。クリックすることにより、機能が切り替えられます。利用可能な機能のみ、メニューが利用可能状態になります。

[定義一覧]機能

データをどのように転送するかを設定するマッパー定義・スクリプト定義と、複数の転送定義を実行順にまとめたタスク定義の管理ができます。

[トリガー一覧]機能

作成した、データ転送定義の実行スケジュール、ファイルトリガーを管理します。

[実行履歴]機能

各トリガーにて実行、または、即時実行された履歴を参照することができます。

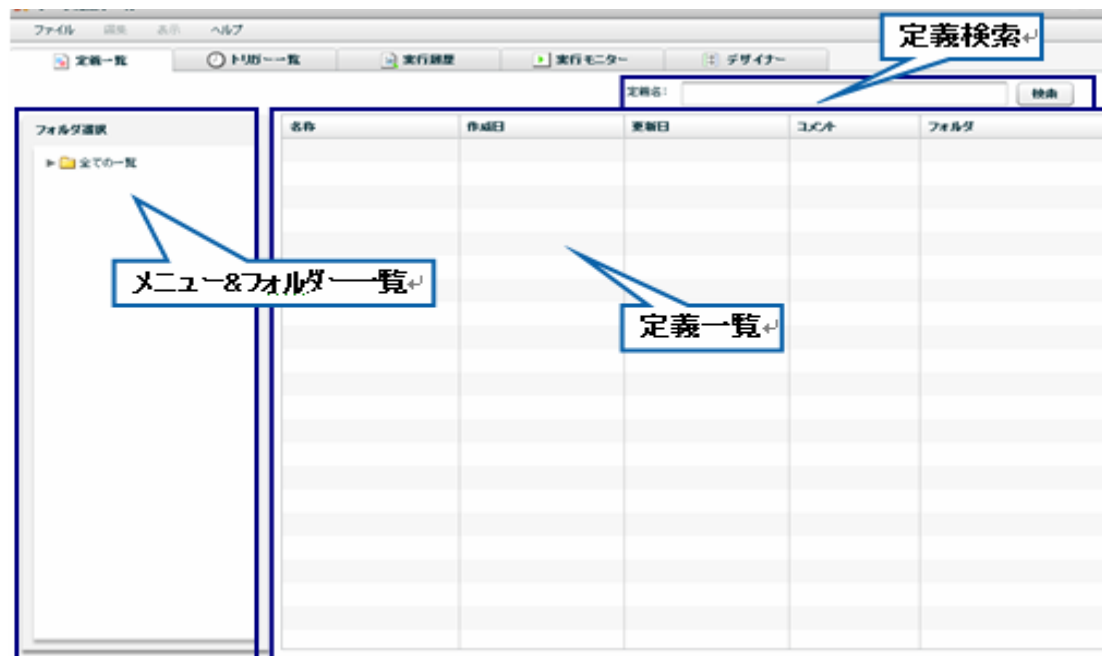
[実行モニター]機能

現在実行されている定義(各トリガーによる起動・即時実行ともに)の一覧を表示します。また、権限がある定義は、中止も可能です。

[デザイナー]機能




マッパー定義・スクリプト定義を作成・編集できます。

2.2.4 定義一覧画面について



定義一覧画面

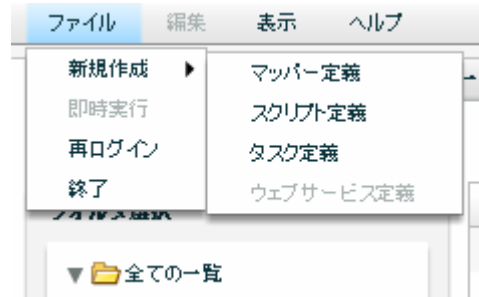
メニュー&フォルダ一覧使用可能なメニューとフォルダを表示します。

全ての一覧	フォルダに関係なく、すべての定義を定義一覧に表示します。
定義一覧	すべてのデータ転送定義を表示します。
 マッパー定義アイコン	定義の種類がマッパーであることを示します。
 タスクアイコン	定義の種類がタスクであることを示します。
 スクリプト定義アイコン	定義の種類がスクリプトであることを示します。
名称	保存時に設定した定義の名称
作成日	定義を作成した日
更新日	定義を更新した日
コメント	定義に設定されたコメント(128文字まで)
フォルダ	定義が格納されているメニュー&フォルダ情報
定義検索	定義名の部分一致で使用権限があるすべての定義検索し表示します。

2.2.5 利用可能メニュー

ここでは、定義一覧で利用可能な定義一覧固有メニューを中心に記述します。

ファイル



定義一覧で何らかの定義が選択されているときのファイルメニュー

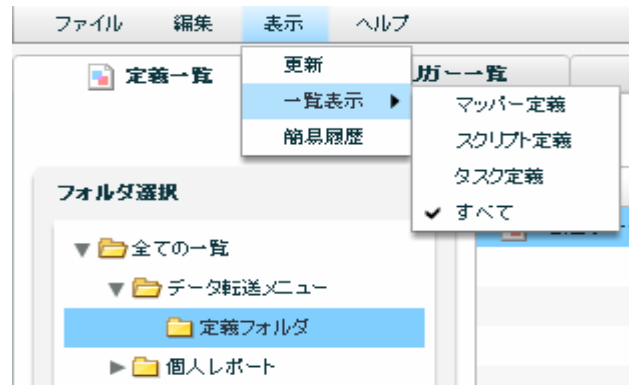
- | | |
|---------|------------------------------------|
| 新規作成： | 定義の新規作成画面に遷移します。 |
| マッパー定義 | 新規マッパー定義を作成するため、デザイナーに遷移します。 |
| スクリプト定義 | 新規スクリプト定義を作成するため、デザイナーに遷移します。 |
| タスク定義 | 新規タスク定義画面が表示されます。 |
| 即時実行 | 選択されているマッパー定義、スクリプト定義、タスク定義を実行します。 |
| 再ログイン | 何らかの理由によりサーバーとの接続が切断されたとき再ログインします。 |
| 終了 | 本製品からログアウトし、終了します。 |



定義一覧で何らかの定義が選択されているときの編集メニュー

- | | |
|-----|-----------------------------|
| 編集： | 選択されている定義を編集可能な状態にします。 |
| 削除： | 選択されている定義を、削除確認画面表示後、削除します。 |

表示

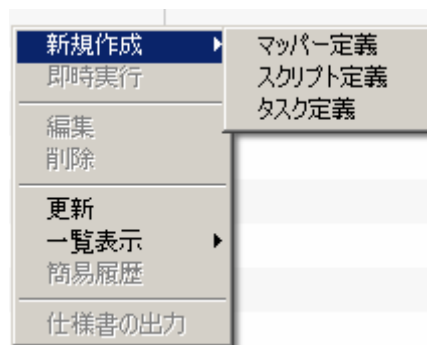


定義一覧で何らかの定義が選択されているときの表示メニュー

更新	一覧の更新をします。
一覧表示	一覧に表示している情報の制御を行います。
マッパー定義	一覧にマッパー定義のみ表示します。
スクリプト定義	一覧にスクリプト定義のみ表示します。
タスク定義	一覧にタスク定義のみ表示します。
すべて	一覧にマッパー定義・スクリプト定義・タスク定義すべてを表示します。
簡易履歴	直近の履歴情報のみ表示します。

右クリックメニュー

ここからは、定義一覧領域で利用可能な右クリックで表示されるメニューについて記述します。



定義一覧領域にて右クリックで表示されるメニュー

新規作成	定義の新規作成ができます。(定義はフォルダの下のみ作成可能です。)
マッパー定義	新規マッパー定義を作成します。
スクリプト定義	新規スクリプト定義を作成します。
タスク定義	新規タスク定義を作成します。

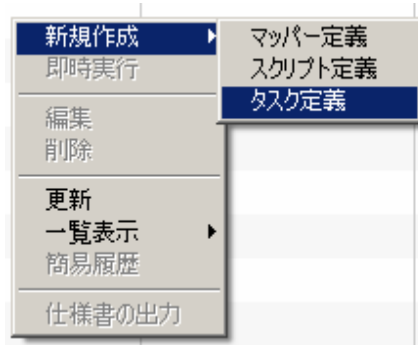
即時実行	定義を実行します。
編集	マッパー or スクリプト or タスク定義を編集できる状態にします。
削除	マッパー or スクリプト or タスク定義を削除します。
更新	一覧の更新をします。
一覧表示	一覧に表示している情報の制御を行います。
マッパー定義	一覧にマッパー定義のみ表示します。
スクリプト定義	一覧にスクリプト定義のみ表示します。
タスク定義	一覧にタスク定義のみ表示します。
すべて	一覧にすべてのデータ転送定義を表示します。
簡易履歴	直近の履歴情報のみ表示します。
仕様書の出力	選択した転送定義の仕様書を出力します。



定義に適切な権限がない場合は、編集・削除メニューが利用できなくなります。

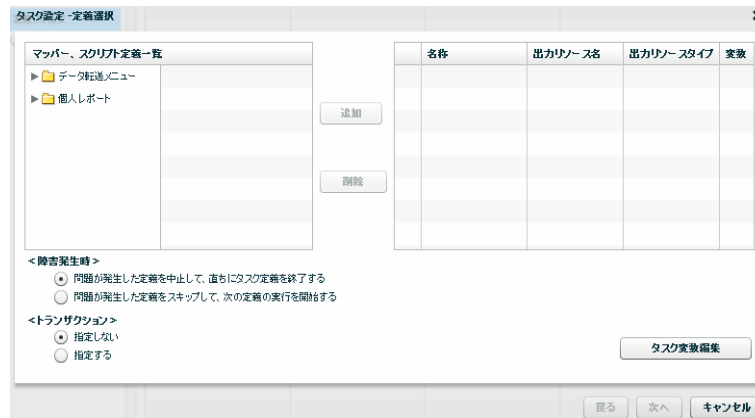
2.2.6 タスク定義設計画面

フォルダを選択後、表示された一覧の右クリックメニュー - [新規作成] - [タスク定義]より、タスク定義を作成することができます。



タスクの新規作成

- 1 マッパー定義を選択します。



マッパー or スクリプト選択

<マッパー、スクリプト定義一覧>

タスクに含めることができるマッパー定義とスクリプト定義のリスト

追加 マッパー定義一覧のマッパー定義 or スクリプト定義をタスク設定に追加します。

削除 タスク設定一覧のマッパー定義 or スクリプト定義をタスク設定から削除します。

<タスク設定一覧> タスクに含まれるマッパー定義 or スクリプト定義のリスト

No マッパー定義 or スクリプト定義の実行順番号

名称 定義名

出力リソース名 転送定義の出力先リソース名 (データベース名、ファイルリソース等)

出力リソースタイプ 転送定義の出力先データタイプ (RDB、CSV 等)

変数 タスクとマッパー or スクリプト間で定義変数の受け渡し設定を行います。

<障害発生時>

障害発生時の処理方法を指定します。

問題が発生した定義をスキップして、次の定義の実行を開始する。

問題が発生した定義を中止して、直ちにタスク定義を終了する。

マッパー定義・スクリプト定義では、レコード単位でのエラー発生時にスキップするか、停止するかを設定できます。



タスク定義では、“マッパー定義とその前後処理を一つの単位”としてエラー発生時にスキップするか、停止するかを設定できます。1回でもスキップが行われた場合、処理結果は警告となります。

<トランザクション>

出力リソース名が同じデータベースのときのみ利用可能です。

定義の途中でエラーが発生したときに、データを元に戻すか、処理したところまでを反映するかを選択できます。

指定しない・・・ 処理したところまでを反映

照会定義に新旧のデータが混じる危険性があります。



転送中も基本的に、照会定義によるデータの閲覧ができます。しかし、転送中の新しいデータと、前回転送されたデータが混在して表示される危険性があります。

指定する・・・ データベースを実行前の状態に戻す

基本的に、データ転送中はデータを閲覧できません。



転送中は、データベースの設定にもよりますが、基本的に照会定義による閲覧ができません。しかし、転送中の中途半端なデータが照会されることはありません。

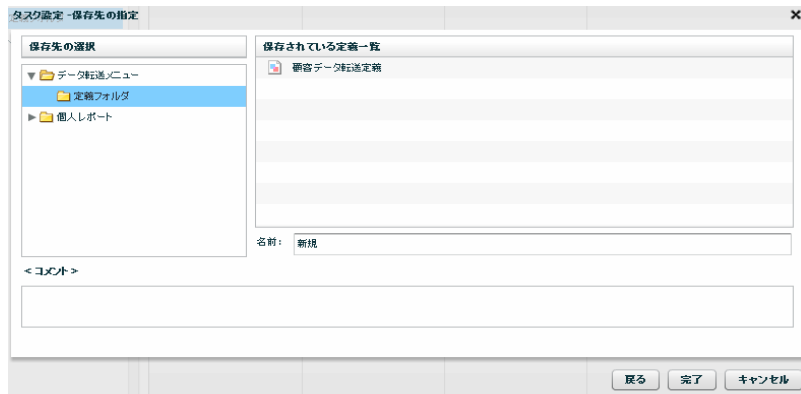


出力するデータベースが異なる場合、出力先がデータベース以外の場合はトランザクションの指定ができません。

<タスク変数編集>

タスクの定義変数を設定します。

2 保存先・タスク定義名を指定します。



タスク保存

名前は、64文字までで指定してください。（半角全角混在でも64文字）
定義を作成する権限があるフォルダにのみ保存できます。
すでにある別の定義から、保存権限を有している定義への上書きも可能です。

トランザクション設定の有効範囲について

同じデータベースへの出力の場合のみ、タスクにて、マッパー定義間に渡るトランザクション設定が可能です。タスクにて、トランザクションを指定したときは、マッパー定義で指定されたトランザクション設定は無視されます。

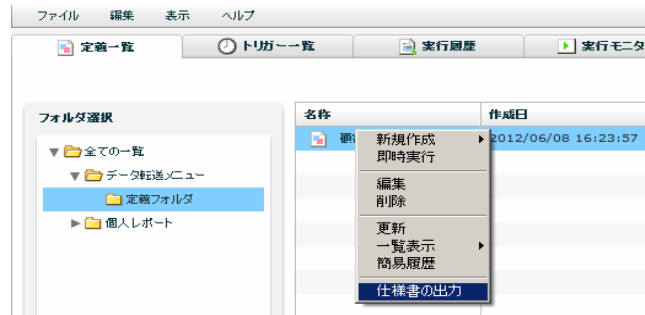


トランザクション設定を”指定する”にすると、データベースサーバーのメモリを多く消費する可能性があります。テスト時に十分な考慮をお願いします。

あまりにもメモリの消費量が多い場合には、対象件数を絞るような条件を設定した WebReport 定義を作成し、転送件数の制限をつける等の対策をお願いします。

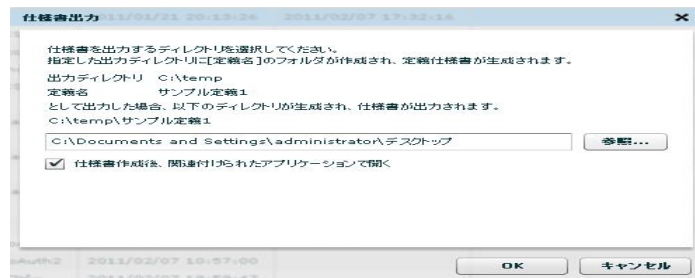
2.2.7 仕様書の出力

フォルダを選択後、表示された一覧からマッパー定義 or スクリプト定義 or タスク定義を選択し、右クリックメニュー - [仕様書の出力]より、仕様書を HTML 形式で作成することができます。



仕様書の出力

- 1 「仕様書出力」画面が表示されます。仕様書出力先ディレクトリを指定して、[OK]ボタンをクリックしてください。「仕様書作成後、関連付けられたアプリケーションで開く」にチェックを入れると、仕様書作成後、拡張子 HTML に関連付けられたアプリケーションで仕様書を開きます。



仕様書出力画面

- 2 仕様書の出力が完了すると、以下のダイアログが表示されます。



仕様書の出力が終了した際に表示されるダイアログ

- 3 仕様書作成後、関連付けられたアプリケーションで開く」にチェックを入れている場合、仕様書を関連付けられたアプリケーションで開きます。仕様書は HTML 形式で出力され、以下の情報と定義のマッピング画面で構成されています。

定義変数

前後処理

入力リソース

出力リソース

転送オプション

フィルター一覧

定義イメージ 戻る

定義のマッピング画面

定義変数 戻る

定義変数

名前	デフォルト値	コメント
変数A	変数A	変数A変数A
変数B	変数B	変数B
変数C	変数C	変数C

前後処理 戻る

前後処理

処理種別 処理種別
処理の実行/無効

処理種別 処理種別
処理の実行/無効

入力リソース 戻る

入力リソース

リソース種別(DB種別) リソース名(スキーマ名) 接続種別
RDS DBC/NT/EAST_DB DBADMIN READ UNCOMMITTED

フィールド情報

フィールド名	実フィールド名	型名	文字列長	桁数	小数桁数	NULL許可	主キー	コメント	表示
ID1	KZKERRTBL1.ID1	INTEGER	0	10	0	はなし	はなし	はなし	標準
DATE1	KZKERRTBL1.DATE1	TIMESTAMP0	26	8		はなし	はなし	はなし	標準
REGION1	KZKERRTBL1.REGION1	VARCHAR	50	50	0	はなし	はなし	はなし	標準
NAME1	KZKERRTBL1.NAME1	VARCHAR	1000	10000		はなし	はなし	はなし	標準
PRICE1	KZKERRTBL1.PRICE1	INTEGER	0	10	0	はなし	はなし	はなし	標準
SALENUM1	KZKERRTBL1.SALENUM1	INTEGER	0	10	0	はなし	はなし	はなし	標準
DATE	KZKTSTTBL2.DATE	TIMESTAMP0	26	8		はなし	はなし	はなし	標準

ジョイン情報

入力テーブル	入力フィールド	出力テーブル	出力フィールド	接続種別
KZKERRTBL1	ID1	KZKTSTTBL2	ID	等価結合

出力リソース 戻る

出力リソース

リソース種別(ce種別) リソース名(スキーマ名) テーブル名 転送モード
RDS DBC/NT/EAST_DB DBADMIN KZKTSTTBL2 インサート

フィールド情報

※結核したフィールドのみ表示します

フィールド名	実フィールド名	型名	文字列長	桁数	小数桁数	NULL許可	主キー	コメント
ID	ID	INTEGER	0	10	0	はなし	はなし	はなし
DATE	DATE	TIMESTAMP0	26	8		はなし	はなし	はなし
REGION	REGION	VARCHAR	100	100	0	はなし	はなし	はなし
NAME	NAME	VARCHAR	200	200	0	はなし	はなし	はなし
PRICE	PRICE	INTEGER	0	10	0	はなし	はなし	はなし
SALENUM	SALENUM	INTEGER	0	10	0	はなし	はなし	はなし

転送オプション

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
トランザクション/確定しない

フィルター一覧 戻る

フィルター一覧

フィルター名	文字列連結
表示名	文字列連結
フローの数を	
区切り文字/アルファ	

出力される仕様書

2.3 トリガー一覧

ここでは、トリガー一覧画面の機能について説明します。

データ転送処理を自動実行するための手段、「スケジュール」と「ファイルトリガー」の 2 種類の設定、管理を行います。

2.3.1 スケジュールとは

データ転送定義を日時で自動的に実行できる機能です。

スケジュール定義を作成すると、データ転送定義の実行を自動化することができます。

例えば、毎週金曜日の午後 1 時に指定した定義を実行することができます。

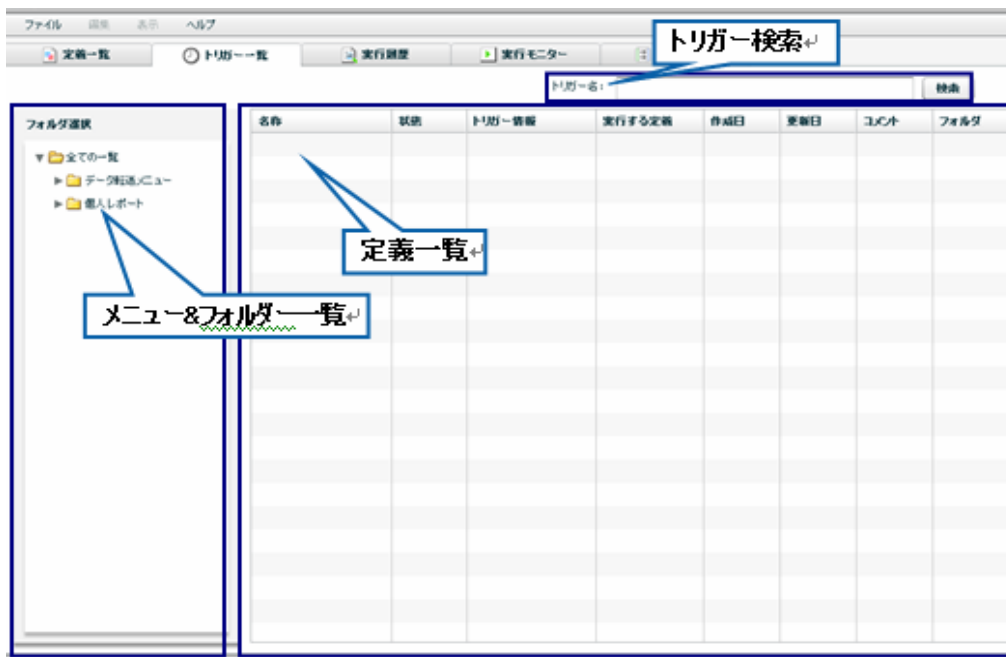
2.3.2 ファイルトリガーとは

データ転送定義をファイル監視で自動的に実行できる機能です。

ファイルトリガー定義を作成すると、データ転送定義の実行を自動化することができます。

例えば、指定した CSV ファイルが更新されるたびに、その CSV ファイルを転送することができます。

2.3.3 トリガー一覧画面について



トリガー一覧画面

メニュー&フォルダ一覧	使用可能なメニューとフォルダを表示します。
すべての一覧	フォルダに関係なく、すべての定義を定義一覧に表示します。
定義一覧	設定した各トリガーを表示します。
名称	保存時に設定した各トリガー名
状態	現在このトリガーが有効か無効かを表示します。停止 or 作動
トリガー情報	スケジュールは実行日時、ファイルトリガーは監視ファイルを表示します。
実行する定義	トリガーにより実行する定義を表示します。
作成日	定義を作成した日
更新日	定義を更新した日
コメント	定義に設定されたコメント（128文字まで）
フォルダ	定義が格納されているメニュー&フォルダ情報
検索フィールドとボタン	トリガー名を入力し、部分一致で検索できます。

2.3.4 利用可能メニュー

ここでは、トリガー一覧で利用可能なトリガー一覧固有メニューを中心に記述します。



定義一覧で何らかの定義が選択されているときのファイルメニュー

ファイル

新規作成

スケジュール

新規スケジュール設定をします。

ファイルトリガー

新規ファイルトリガー設定をします。

再ログイン

何らかの理由によりサーバーとの接続が切断されたとき再ログイン
します。

終了

本製品からログアウトし、終了します。

編集



定義一覧で何らかの定義が選択されているときの編集メニュー

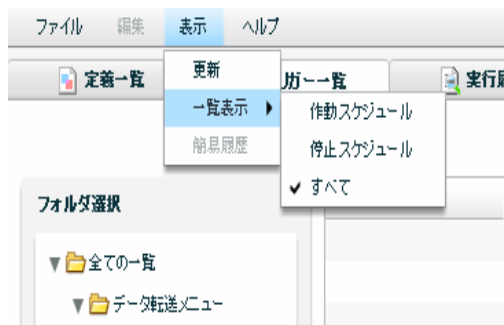
編集

選択されているトリガーを編集可能な状態にします。

削除

選択されているトリガーを削除確認画面表示後、削除します。

表示

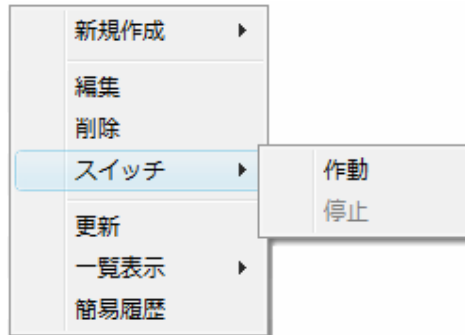


定義一覧で何らかの定義が選択されているときの表示メニュー

更新	一覧の更新をします。
一覧表示	一覧に表示する情報を切り替えます。
作動スケジュール	一覧に作動スケジュールのみ表示します。
停止スケジュール	一覧に停止スケジュールのみ表示します。
すべて	一覧に作動スケジュール・停止スケジュールの両方を表示します。
簡易履歴	過去に実行された履歴情報のうち直近のデータのみ表示します。

右クリックメニュー

ここからは、トリガー一覧領域で利用可能な右クリックで表示されるメニューについて記述します。



定義一覧領域にて右クリックで表示されるメニュー

新規作成

スケジュール	スケジュールを新規に作成します。
ファイルトリガー	ファイルトリガーを新規に作成します。

編集

トリガーを編集できる状態にします。

削除

トリガーを削除します。

スイッチ 起動

トリガーを有効にします。

スイッチ 停止

トリガーを無効にします。

更新

一覧の更新をします。

一覧表示

一覧に表示する情報を切り替えます。

作動トリガー

一覧に作動トリガーのみ表示します。

停止トリガー

一覧に停止トリガーのみ表示します。

すべて

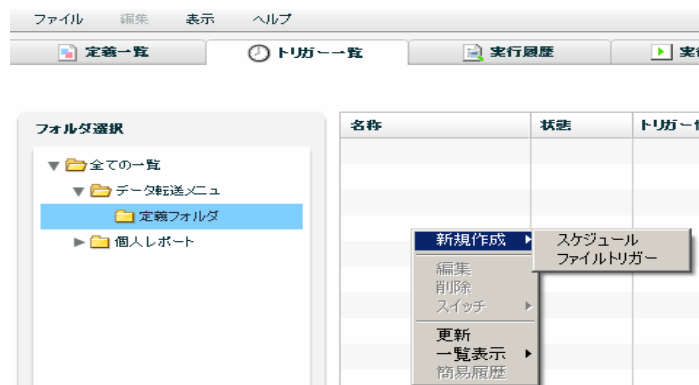
一覧に作動トリガー・停止トリガーの両方を表示します。

簡易履歴

過去に実行された履歴情報のうち直近のデータのみ表示します。

2.3.5 スケジュール作成画面

フォルダを選択後、表示された一覧の右クリックメニュー - [新規作成] - [スケジュール]より、スケジュールを作成することができます。



スケジュールの新規作成

- 1 実行タイミングを設定します。

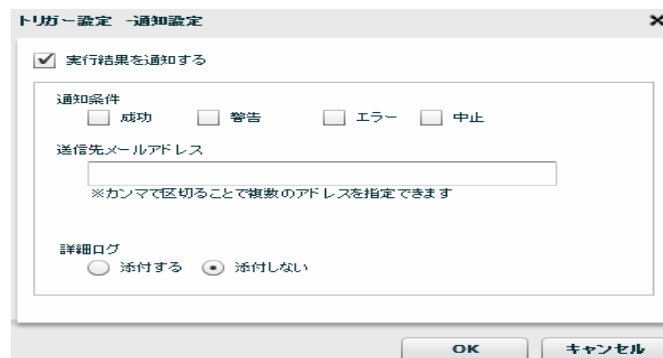


スケジュール設定-実行タイプ

実行タイミングを選択し、タイミングにあった値を設定します。

通知設定

通知設定画面が開き、エラー時などのメール転送内容を設定できます。



スケジュール設定-通知設定

「実行結果を通知する」にチェックを入れ、通知条件、送信先メールアドレス、詳細ログを設定します。

通知条件

成功	スケジュールが正常に終了したとき通知します。
警告	スケジュールが警告を伴って終了したとき通知します。
エラー	スケジュールが異常終了したとき通知します。
中止	スケジュールが中止されたときに通知します。

送信先メールアドレス

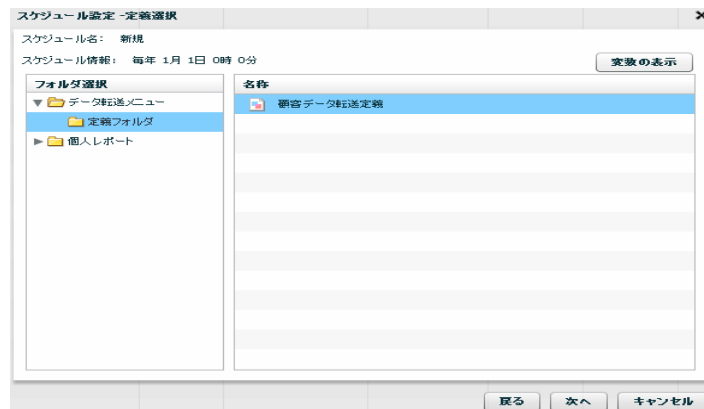
通知メールを送信するメールアドレス。

カンマで区切ることで複数のアドレスを指定できます。

詳細ログ エラー時の情報を出力したファイルです。（エラー時のみ作成されます。）

添付する・添付しない 通知メールに詳細ログファイルを添付するかの設定

2 実行させたい転送定義を指定します。

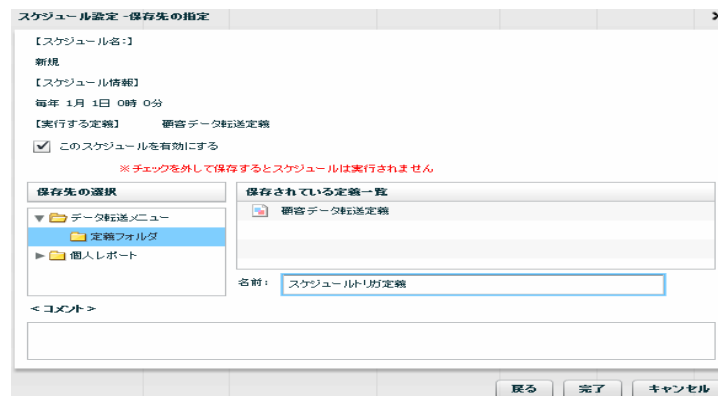


実行定義指定

スケジュール設定により、実行させたい転送定義を選択します。

使用権限のない定義は表示されません。

3 保存先・スケジュール名を指定します。



スケジュール保存

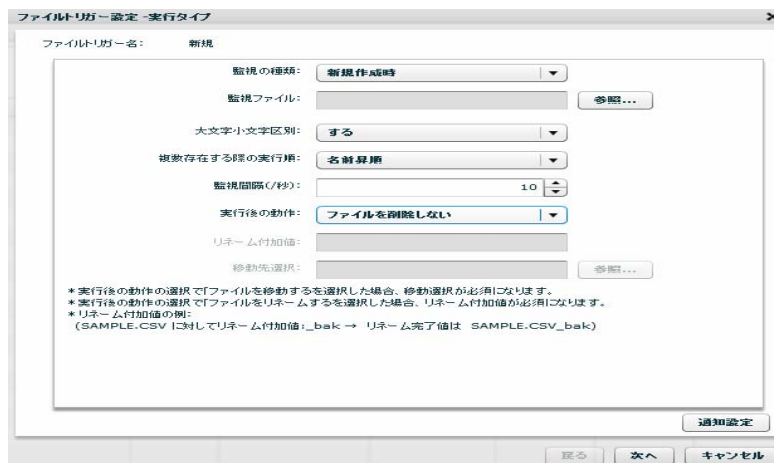
2.3.6 ファイルトリガー作成画面

フォルダを選択後、表示された一覧の右クリックメニュー - [新規作成] - [ファイルトリガー]より、ファイルトリガーを作成することができます。



ファイルトリガーの新規作成

1 実行タイミングを設定します。



ファイルトリガー設定-実行タイプ

ファイルトリガーを設定します。

監視の種類

データ転送定義を実行するタイミングを監視ファイルの「新規作成時」「日付更新時」から選べます。

監視ファイル

ファイルトリガーで監視するファイルを指定します。

「参照」ボタンから「監視ファイル選択」画面を表示して選択してください。

大文字小文字区別

上で指定した監視ファイル名の 大文字小文字を区別するか指定します。

複数ファイルの実行順

「名前昇順」「名前降順」「日付昇順」「日付降順」が選択できます。

監視ファイルにはワイルドカードや正規表現が使えるため、合致するファイルが複数存在する場合があります。例えば、受注*.csv を監視対象にしていた場合に、「受注 1.csv」「受注 2.csv」「受注 3.csv」「受注 4.csv」の4つのファイルが一度に新規作成されると4回定義が実行されます。この時の実行順を指定することができます。名前降順でデータを転送することなどができます。

監視間隔

指定されたファイルを監視する間隔です。下限は10秒です。

実行後の動作

実行後に監視ファイルへの対応を設定できます。「ファイルを削除しない」「ファイルを削除する」「成功時のみ削除する」「ファイルを移動する」「ファイルをリネームする」の5つの中から選択できます。

リネーム付加値

「実行後の動作」で「ファイルをリネームする」を選択した場合に、監視ファイルの拡張子の後に文字列を付加することでリネームを行います。例えば監視ファイルが「受注 1.csv」「リネーム付加値」が「_bak」の場合はファイルトリガーでデータ転送定義が実行された後に「受注 1.csv_bak」というファイルにリネームされます。

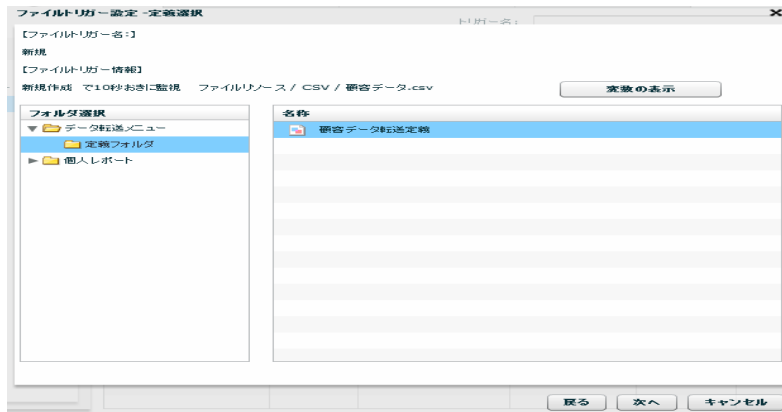
移動先選択

「実行後の動作」で「ファイルを移動する」を選択した場合に、監視ファイルを指定したフォルダに移動することができます。

通知設定

通知設定画面が開き、エラー時などのメール転送内容を設定できます。設定できる内容はスケジュールと同じになります。

2 実行させたい転送定義を指定します。

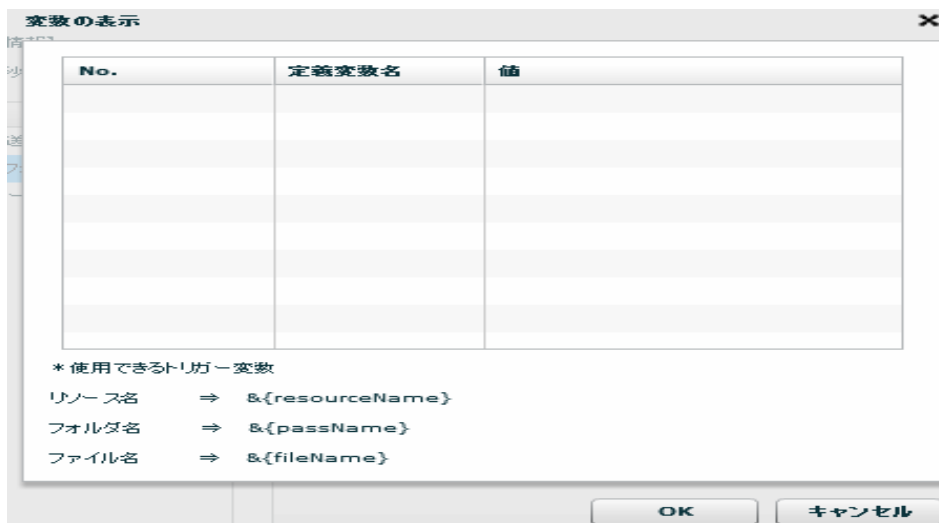


実行定義指定

ファイルトリガーにより、実行させたい転送定義を選択します。使用権限のない定義は表示されません。

変数の表示

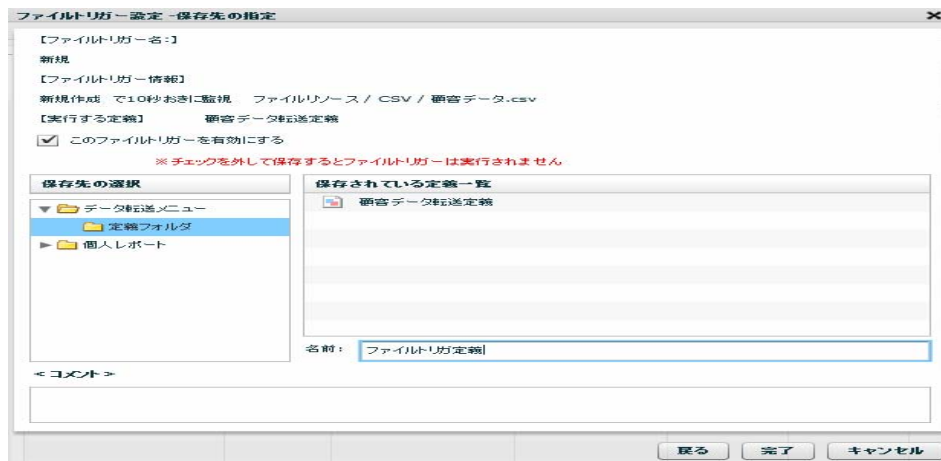
実行させたい転送定義に監視ファイルのリソース、フォルダ、ファイル名を渡すことができます。



変数の表示

「実行定義指定」で設定した転送定義の定義変数が表示されるので、「値」に引き渡すトリガー変数を指定してください。トリガー変数はそれぞれ「&{resourceName}」「&{passName}」「&{fileName}」の3種類があり、それぞれファイルトリガー実行時の監視ファイルのリソース、フォルダ、ファイル名が入っています。これにより、実行時のファイルをデータ転送定義に渡すことができます。例えばファイルトリガーに引っかかったファイルが「受注 3.csv」の場合、このファイルをデータ転送定義でRDBに転送することができます。

3 保存先・ファイルトリガー名を指定します。



ファイルトリガー保存

ファイルトリガーで監視ファイルをワイルドカードや、正規表現で指定している場合に、複数のファイルが一度に合致する場合、合致したファイルの数だけ定義を実行することになります。この時に全ての定義が終了しない間に、同名のファイルが作成されてもトリガーで感知することができません。

例えば、次のような設定の場合

- ・監視の種類：新規作成時
- ・監視ファイル：売り上げ*.csv
- ・実行後の動作：ファイルを削除する



[売り上げ1.csv]、[売り上げ2.csv]、[売り上げ3.csv]と新規作成され一度に実行されたとします。[売り上げ1.csv]が削除された後に、外部から新しく[売り上げ1.csv]が作成されても[売り上げ3.csv]の処理が終了し、ファイルトリガーの動作が終了した時点で、[売り上げ1.csv]が残っているため、次の監視では新規作成と認識されません。

このような状態になると[売り上げ.csv]が残り続けて、ファイルトリガーに反応しなくなってしまいます。一度、ファイルトリガーを「停止」して、[売り上げ1.csv]を削除し、再度「作動」させてください。



監視間隔が長すぎる場合は、間隔の間に同じファイルが複数回更新されても、途中の更新状態を認識することができません。監視間隔はファイルの更新間隔より短い時間を指定してください。



ファイルトリガーの編集は即座に反映されません。次の監視タイミング後に反映されます。

2.4 実行履歴

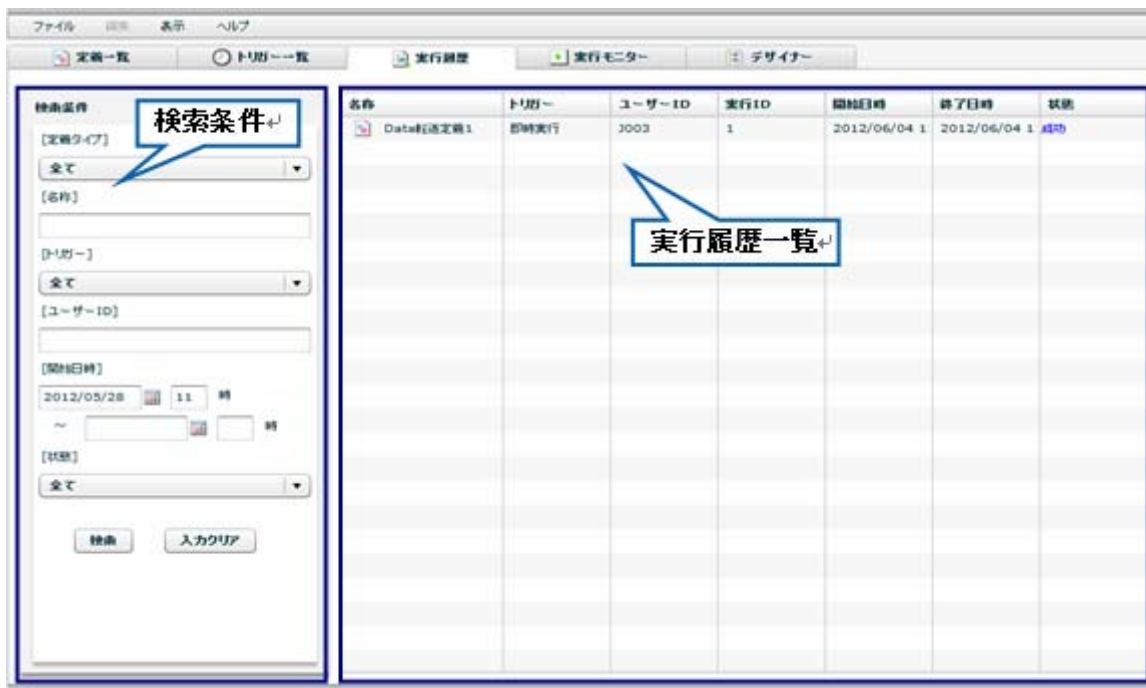
ここでは、実行履歴画面の機能について説明します。

データ転送処理の実行結果を参照できます。

2.4.1 実行履歴とは

転送定義を即時実行した場合、トリガーから自動実行した場合の実行結果のことです。

2.4.2 実行履歴画面について



実行履歴画面



使用権限のある定義の実行履歴のみ一覧に表示されます。

【実行履歴一覧】

名称	定義の名称
トリガー	何から実行されたかを表します。 スケジュール or ファイルトリガー or 即時実行
ユーザーID	即時実行の場合、実行したユーザーの ID。 スケジュール、ファイルトリガーの場合はその作成者の ID。
実行 ID	実行単位に振られる一意の番号
開始日時	実行が開始された日時
終了日時	実行が終了した日時
状態	成功
	警告
	エラー
	中止
	サーバーダウン
	何のエラーなく終了した場合
	何らかのエラーは発生したが、スキップ設定により 最後まで実行され、終了した場合
	何らかのエラーにより途中で停止して終了した場合
	明示的に中止ボタンで終了させられて途中で終了した場合
	実行中にサーバーが停止し、途中で終了した場合

【検索条件】 入力し、ログ検索ボタンをクリックすると、検索条件として有効になります。

名称	スケジュール名部分一致
定義タイプ	定義の種別 マッパー定義・スク립ト定義・タスク定義
トリガー	何から実行されたか 各トリガー・即時実行
実行者 ID	定義を即時実行したユーザー 各トリガー定義の作成ユーザーのログイン ID
開始日時	開始日時の範囲の始まりを検索条件として設定
終了日時	開始日時の範囲の終わりを検索条件として設定
状態	各トリガーの状態を検索条件として設定 (全て・成功・警告・エラー・中止・サーバーダウン)



開始日時とは、実行が開始された日時情報のことです。

2.4.3 利用可能メニュー

ここでは、実行履歴で利用可能な実行履歴固有メニューを中心に記述します。

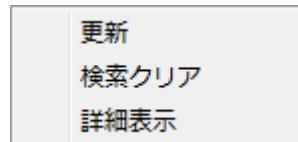
表示



実行履歴一覧で何らかの定義が選択されているときの表示メニュー

更新	一覧の表示を更新します。
検索クリア	設定されている検索条件をクリアし、初期表示に戻します。 (検索条件指定時のみ利用可能)
詳細表示	選択された定義の詳細実行履歴を表示します。

ここからは、実行履歴一覧領域で利用可能な右クリックで表示されるメニューについて記述します。



定義一覧領域にて右クリックで表示されるメニュー

更新	一覧の表示を更新します。
検索クリア	設定されている検索条件をクリアします。
詳細表示	選択された定義の詳細実行履歴を表示します。

2.4.4 詳細表示画面

一覧のメニューより、実行結果の詳細情報が閲覧できます。



実行履歴詳細

基本情報 実行履歴一覧に表示されている情報と同様の実行結果の概要を表示します。

結果情報 スケジュールまたは即時実行で実行された、転送定義の実行状況を表示します。

詳細情報 転送定義の実行時の詳細情報を表示します。

Error 出力について

- ・ Mail に添付
- ・ サーバーのフォルダを直接見る

Windows でのデフォルト導入ドライブ：¥Program Files¥Customer Vision¥QanarSvr¥error¥実行 ID.err というファイルになります。



付録に実行時に出現しやすいエラー情報があります。

2.4.5 リストの操作

項目名をクリックすることにより、表示リストの並び替えが可能です。

名称	トリガ-	ユーザーID	実行ID	開始日時	終了日時	状態
 Data転送定義2	即時実行	J003	2	2012/06/04 1	2012/06/04 1	成功
 Data転送定義1	即時実行	J003	1	2012/06/04 1	2012/06/04 1	成功

実行履歴一覧画面

2.5 実行モニター

2.5.1 実行中とは

転送定義を即時実行した場合、トリガーから自動実行した場合の実行状態のことです。

2.5.2 実行モニター画面について

ここでは、実行モニター画面の機能について説明します。

現在、実行中のデータ転送処理の実行状況を参照できます。自分で即時実行した定義、または、停止権限がある定義の場合は、中止ができます。



名称	ユーザーID	実行ID	開始日時	処理済みデータ件数	強制中止
largeCSV	cvadmin	69132	2011/06/07 11:41:	2010count	中止

実行モニター画面



すべての実行中定義が一覧に表示されます。

名称	マッパー定義名 または タスク名
ユーザーID	実行を開始したユーザーID / トリガーの場合はその設定ユーザー名
実行 ID	実行単位に振られる一意の番号
開始日時	実行が開始された日時
強制中止	即時実行の場合 自分が実行した処理のみ中止できます。 スケジュールの場合 定義に停止権限がある処理のみ中止できます。

2.5.3 利用可能メニュー

実行モニターで利用可能な実行モニター固有メニューはありません。実行中一覧領域で利用可能な右クリックで表示されるメニューについて記述します。



定義一覧領域にて右クリックで表示されるメニュー

更新	一覧の表示を更新します。
中止	選択された定義を中止します。

2.6 デザイナー

ここでは、デザイナー画面の機能について説明します。

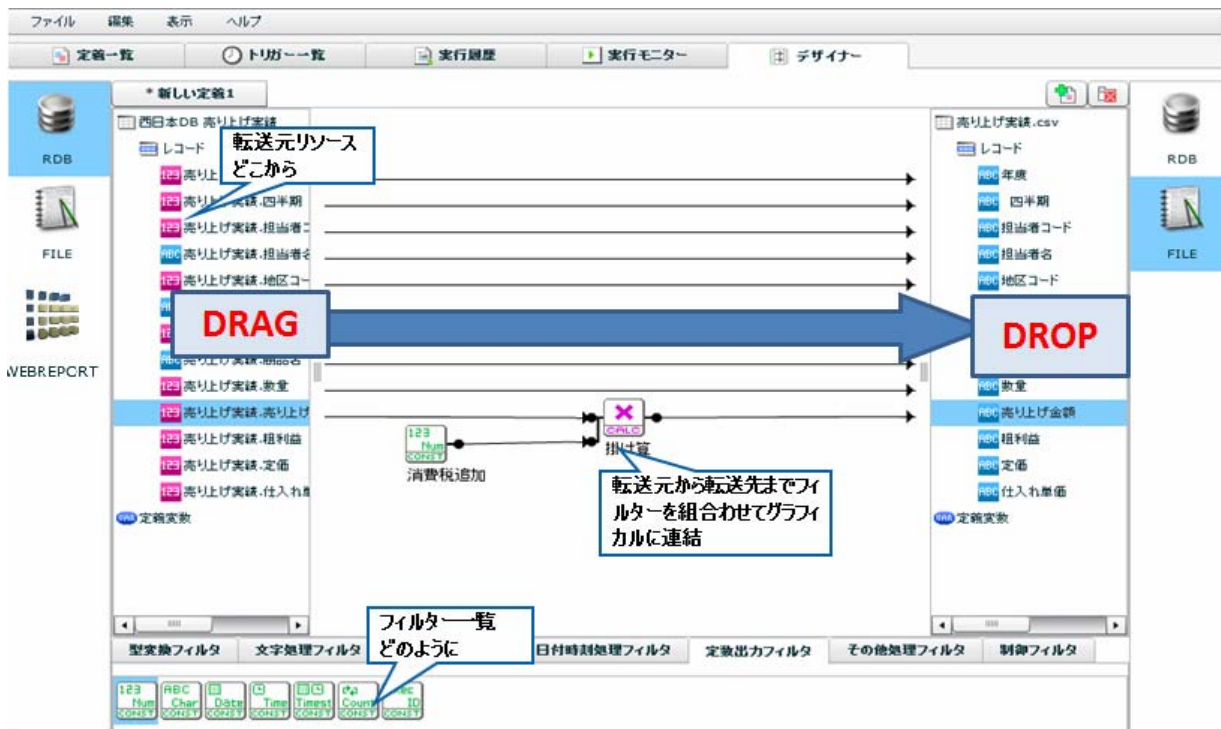
デザイナーでは、転送元から転送先への転送設定であるデータ転送定義である「マッパー定義」の作成ができます。

2.6.1 データ転送定義とは

データを転送する際の主要な設定のことです。

転送元、転送先、転送するデータ、転送方法の情報を含みます。

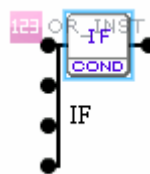
2.6.2 デザイナー画面について



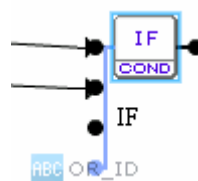
デザイナー画面

左側の転送元テーブル（ファイル）のフィールドを右側の転送先テーブル（ファイル）のフィールドへ Drag & Drop することにより、転送の設定ができ、線（フロー）が描画されます。

フィルタの入口にフローをつなぎたい場合は、フィルタの上に転送元フィールドを Drag & Drop するか、フロー接続部に転送元フィールドを Drag & Drop してください。



フィルタ上に Drag & Drop（上から順にフローが引かれます。）

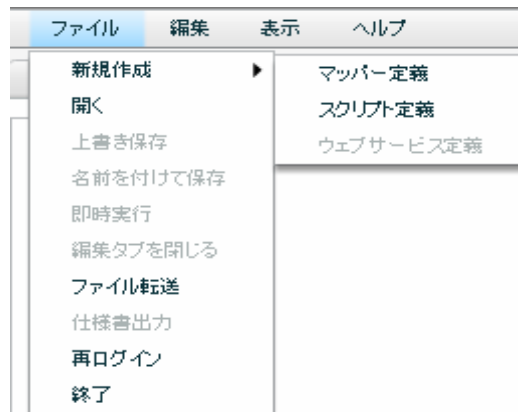


フロー接続部に Drag & Drop（青くハイライトされている接続部にフローが引かれます。）

2.6.3 共通メニュー

ここでは、デザイナーで利用可能なデザイナー固有メニューを中心に記述します。

ファイル



ファイルメニュー

新規作成

マッパー定義	新規マッパー定義を作成します。
スクリプト定義	新規スクリプト定義を作成します。

開く

既存のマッパー定義、スクリプト定義を編集するために開きます。

上書き保存

既存のマッパー定義、スクリプト定義を同じ名前で保存します。

名前を付けて保存

編集中のマッパー定義、スクリプト定義に名前を付けて (または変更して)保存します。

即時実行

現在開いているマッパー定義、スクリプト定義を即時実行します。

編集タブを閉じる

現在開いているマッパー定義、スクリプト定義を閉じます。

ファイル転送

本製品サーバーに登録しているファイルリソースに対して、ファイルのアップロード/ダウンロードを行います。

仕様書出力

マッパー定義、スクリプト定義の仕様書を出力します。

再ログイン

何らかの理由によりサーバーとの接続が切断されたとき再ログイン
します。

終了

本製品を終了します。

編集



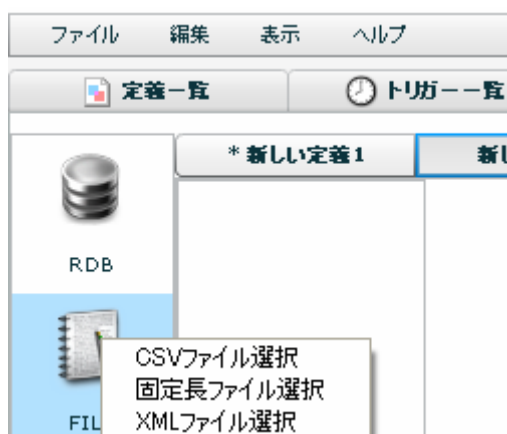
編集 メニュー

編集		デザイナーでは利用できません。
削除		デザイナーでは利用できません。
自動割当	順番	転送元と転送先が指定されているときに、上から順番に同じ型に対して、転送の線を自動で引きます。
	名前	転送元と転送先が指定されているときに、上から同じ名前、同じ型に対して、転送の線を自動で引きます。
転送オプション		マッパー定義、スクリプト定義の転送オプション指定画面を表示します。
前後処理		データ転送を実行する前と実行した後にコマンドを指定します。
フロースタイル		フローのスタイルを指定します。
	直線	フローが直線で表示されます。
	カギ線	フローがカギ線で表示されます。
	曲線	フローが曲線で表示されます。
キャンバスをクリア		設定済の線とフィルタを一括削除します。

2.6.4 リソース選択



データ転送元・転送先には RDB を選択できます。データ転送元では RDB のテーブルの結合結果をリソースとして指定できます。



ファイル選択には CSV ファイル・固定長ファイル・XML ファイルが選択できます。

2.6.5 RDB 選択

データ転送元に RDB を選択します。



RDB 選択



転送元選択

データ転送元では複数のテーブルを で結合して、転送元として参照出来ます。

売上げ情報と顧客マスタ情報を結合することなどができます。

では参照するフィールドを指定します。

と では検索条件、並び替えを行えます。

【テーブル一覧】 転送したいデータが格納されているテーブルを へのドラッグ&ドロップで選択します。

【結合】 中央部分にあるマッピング画面にテーブルを複数出力し、フィールドからフィールドにドラッグ&ドロップすることで、同じデータベース内のテーブルを結合することができます。



結合したフィールド同士が結線される

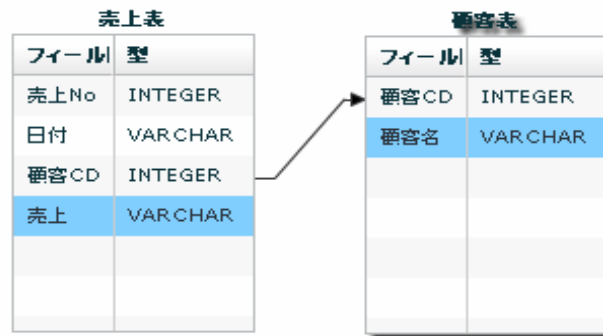
【矢印選択】

結合したフィールド同士が結線され、その結線上で右クリックすることで矢印方向で参照元、参照先を指定できる。以下に例を記載致します。

売上表				顧客表	
売上No	日付	顧客表.顧客CD	売上	顧客CD	顧客名
1	2001/05/27	101	300	101	二島商店
2	2001/05/27	101	1750	102	姫路商事
3	2001/05/27	103	5000	103	大阪物産
4	2001/06/27	104	1750	104	神戸商店
5	2001/06/27	104	2500	105	福岡商事

・顧客CD同士で連結しテーブルを結合し、マスタのデータを取得します。

矢印の方向が「 」の場合： 売上表が参照元で顧客表マスタを参照します。

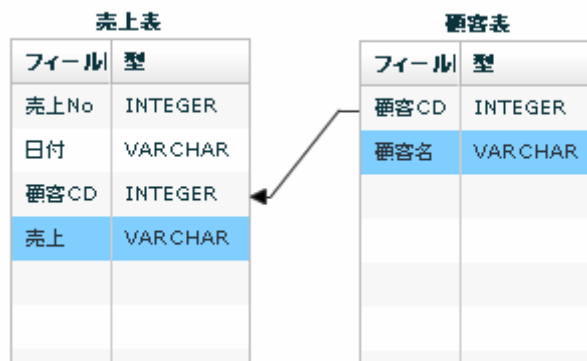


実行結果は以下の通りです。

実行結果

売上No	日付	売上表.顧客CD	顧客表.顧客CD	顧客名	売上:
1	2001/05/27	101	101	二島商店	300
2	2001/05/27	101	101	二島商店	300
3	2001/05/27	103	103	大阪物産	5000
4	2001/06/27	104	104	神戸商店	1750
5	2001/06/27	104	104	神戸商店	1750

矢印の方向が「 」の場合： 顧客表マスタが参照元で売上表を参照します。



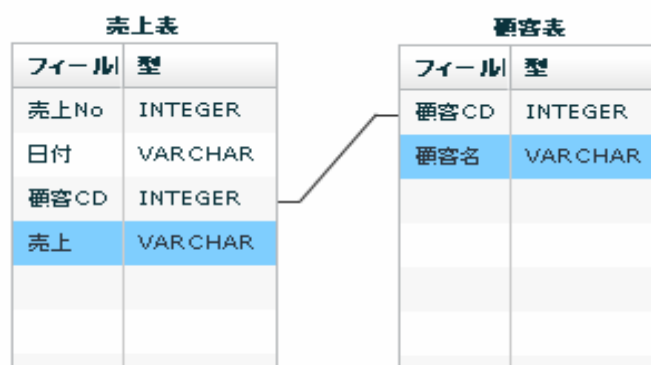
実行結果は以下の通りです。

実行結果

顧客名	顧客表.顧客CD	売上表.顧客CD	売上No	日付	売上:
二島商店	101	101	1	2001/05/27	300
二島商店	101	101	2	2001/05/27	300
姫路商事	102				
大阪物産	103	103	3	2001/05/27	5000
神戸商店	104	104	4	2001/06/27	1750
神戸商店	104	104	5	2001/06/27	1750
福岡商事	105				

矢印の方向が「-」の場合： 売上表の顧客CDと顧客表マスタの顧客CDが一致したものを

出力します。



実行結果は以下の通りです。

実行結果

売上No	日付	売上表.顧客CD	顧客表.顧客CD	顧客名	売上:
1	2001/05/27	101	101	二島商店	300
2	2001/05/27	101	101	二島商店	300
3	2001/05/27	103	103	大阪物産	5000
4	2001/06/27	104	104	神戸商店	1750
5	2001/06/27	104	104	神戸商店	1750

- 【フィールド選択】** テーブル一覧にて選択されたテーブルのフィールド一覧を表示します
 フィールド選択するにはマッパー定義上にあるフィールドを
 ドラッグ&ドロップするか画面上の全設定ボタンで行えます。



- 【フィールド名】** フィールド定義として保管されている表示名です。

- 【型】** データの型です。

- 【表示】**



標準	選択されたフィールドの値を返します。
最大	選択されたフィールドの最大値をレコードから返します。
最小	選択されたフィールドの最小値をレコードから返します。
平均値	選択されたフィールドの平均値をレコードから返します。
合計	選択されたフィールドの合計値をレコードから返します。
集計キー	指定されたフィールドをグループでまとめて1レコード出力します。

例： 下記のデータで出身別で年齢の平均値を出したい場合、

レコードNO	出身	名前	年齢
1	東京	山田	17歳
2	東京	鈴木	25歳
3	兵庫	佐藤	45歳



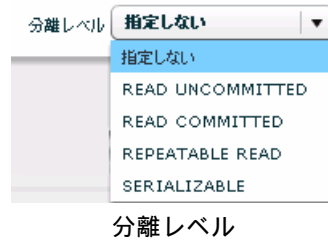
フィールド選択で出身のデータが入ってフィールドに集計キーをセットし、
次に年齢のデータが入っているフィールドに平均値をセットする
上記の出力結果

レコードNO	出身	年齢
1	東京	21歳
2	兵庫	45歳

【全設定ボタン】 画面上部にある全設定ボタンを押下することで中央部で選択されているテーブルのフィールドを全て選択することができます。



【分離レベル】 定義実行時の転送元の分離レベルを設定します。



指定しない	転送元データベースに指定された分離レベルを利用します。
READ UNCOMMITTED	読み取り用ロックを使用しません。
READ COMMITTED	データ読み取り時に読み取り用ロックをかけます。
REPEATABLE READ	トランザクションが終了する前に複数回読み込む場合でも、その間に他のトランザクションが実行されても影響がないようなロックをかけます。
SERIALIZABLE	REPEATABLE READ に加え、トランザクション中は他のトランザクションからの行の追加もできないようなロックをかけます。

分離レベルについて

データ転送元の分離レベルの設定により、送信されるデータの " 鮮度 " が異なります。

作成したい処理に適した分離レベルの設定を行ってください。



データベースにより、サポートされない分離レベルもあります。詳細は、転送元データベースの仕様を確認してください。

【検索条件】

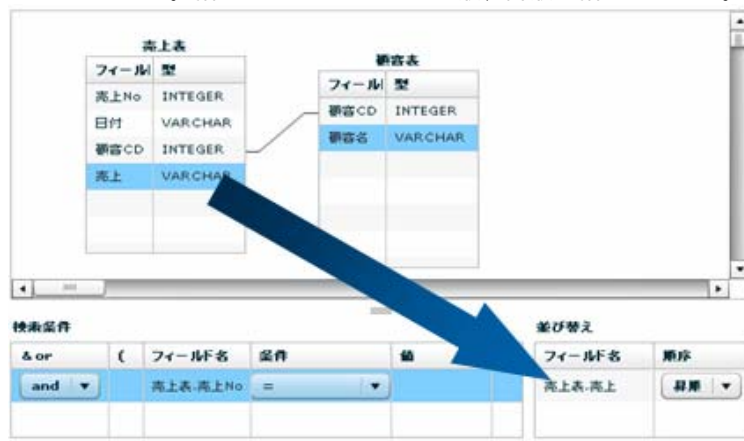
のテーブル配置領域に配置したテーブルから検索条件に使用したいフィールドを の検索条件領域にドラッグ&ドロップします。SQL の Where 句と同じ構文を作成します。

&or	複数の条件を指定したいとき and 条件か or 条件を選択できます。
(条件を囲む (の部分です。
フィールド名	検索条件に使用するフィールド名です。
条件	" =、! =、> =、>、< =、< "、文字含む、文字含まない、文字始まる、文字終わる、"is Null、is not Null"の中から使用する検索条件を選択します。
値	検索条件に使用する値をここに入力します。
)	条件を囲む) の部分です。

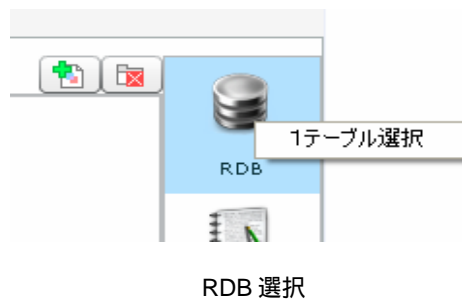


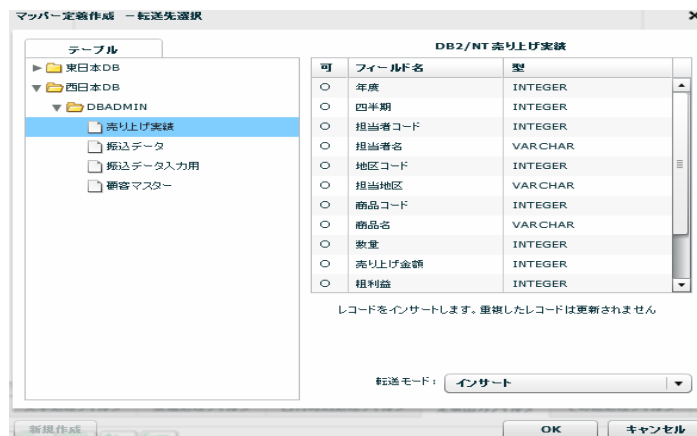
並び替え

のテーブル配置領域に配置したテーブルから並び替えをしたいフィールドを の並び替え領域にドラッグ&ドロップします。指定したフィールドの昇順、降順を指定できます。

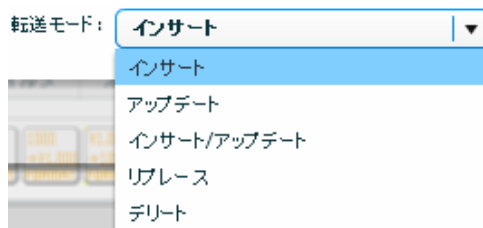


データ転送先に RDB を選択します。





- テーブル一覧 データを転送したいテーブルを選択します
- フィールド一覧 テーブル一覧にて選択されたテーブルのフィールド情報
- 可 フィールドが転送対象として設定可能かどうかを表示します。
- フィールド名 フィールド定義として保管されている表示名
- 型 データ型
- 転送モード 転送先への処理モードを選択します。



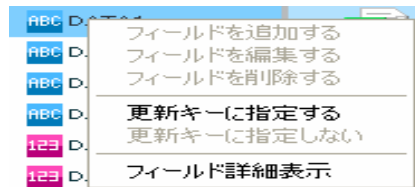
転送モード

- インサート データを追加だけします。キーの重複時にはエラーになります。
- アップデート データを更新だけします。更新対象がない場合エラーになります。
- インサート/アップデート データを更新キーにより、追加または更新します。
- リプレース データを全件削除後、追加します。
- デリート データを自動選択されるキーにより、削除します。

アップデート、インサート/アップデート時には更新キーを指定する必要があります。

デリート時の場合は結線したフィールド全てが自動的にデリートのキーになります。

更新キーは、フィールドを選択状態にし、右クリックで表示されるメニュー [更新キーに指定する] にて設定可能です。

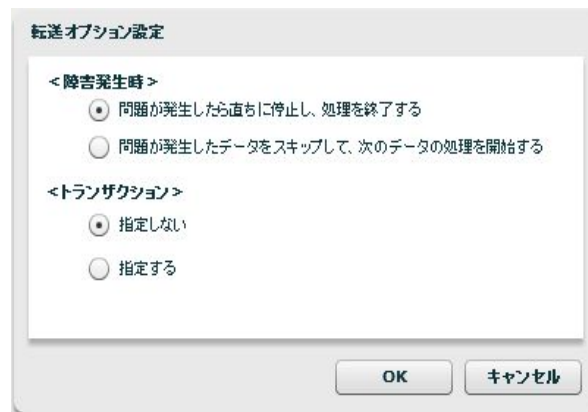


更新キー指定メニュー

メニューの[編集] [転送オプション]で設定の変更を行うことができます。

この画面では、問題発生時のスキップ設定、トランザクションを設定できます。

スキップを設定した場合は、問題が発生した場合にそのデータを無視するためエラーとは判定されません。



2.6.6 RDB 作成・変更

転送先テーブルをマッパー定義作成時に作成することもできます。



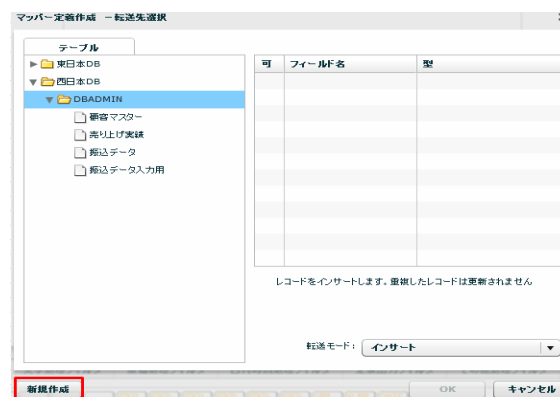
権限の章に書いてある権限がユーザーに与えられている場合のみ利用可能です。

スキーマに必要な権限があれば、データ転送先選択画面下部左の[新規作成]ボタンが利用可能になります。

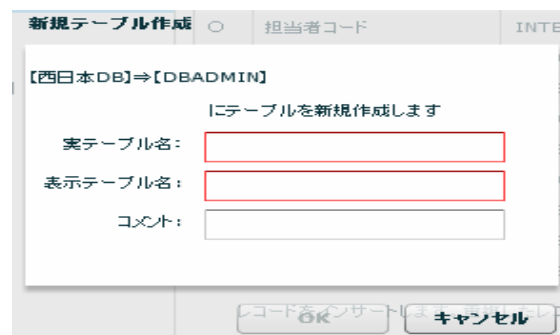
以下の手順にて必要な項目を入力&設定していくことにより、テーブル作成ができます。

1 テーブル情報の設定

転送先選択画面にて、作成したいスキーマを選択し、[新規作成]ボタンをクリックします。



転送先選択

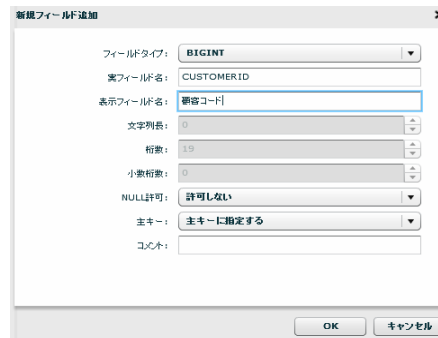


新規テーブル作成

- | | |
|---------|---|
| 実テーブル名 | データベースに実際に作成されるテーブル名です。半角英数文字のみ有効
設定可能な実テーブル名の長さは、データベースの種類により異なります。 |
| 表示テーブル名 | 本製品内で利用されるテーブル定義の表示用の名称です。 |
| コメント | テーブル定義のコメント（128文字まで） |

2 フィールド情報の設定

転送元のフィールドをドラッグし、転送先の“レコード”にドロップすることでフィールドを追加できます。



新規フィールド追加

転送元のフィールドの情報が設定されている、新規フィールド作成画面が表示されます。

何の変更の必要もなければ、「OK」ボタンをクリックすると、フィールド設定は完了です。

設定の変更が必要であれば、設定を変更してください。

「データ型」について



転送元データベースと転送先データベースの DBMS が異なる場合、データ型に互換性がない場合があります。この場合は正しく情報が設定されませんので手動での設定をお願いします。

設定値について

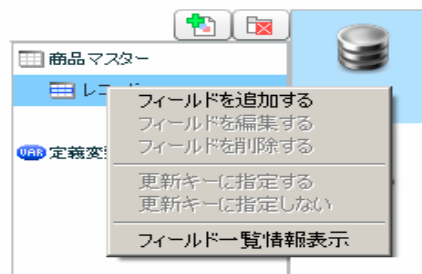


転送元データベースと転送先データベースの DBMS が異なる場合、フィールドの設定値を利用可能な範囲で利用しますが、データベースの種類により設定可能な値が異なるため、テーブルを作成するデータベースの上限値に制限されます。

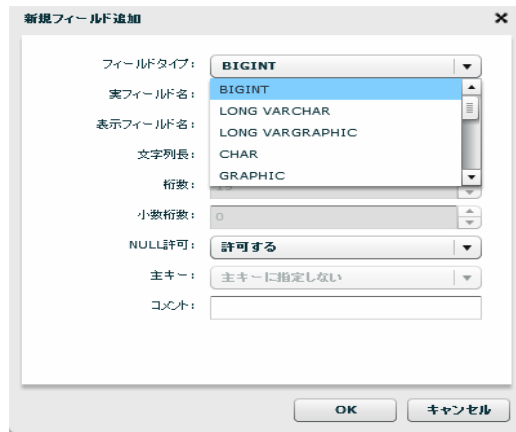
データ型を変更するなどの処理が必要になることがありますのでご注意ください。

また、手動で設定することにより新たなフィールドを作成することも可能です。

転送先のレコードをクリックし、右クリックメニューのフィールドの追加で何も設定されていないフィールドの追加画面が表示されます。



レコードの右クリックメニュー



新規フィールド作成

フィールドタイプ	本製品で利用可能かつデータベースで利用可能なデータ型のリスト
実フィールド名	テーブルに作成されるフィールド名(半角大文字英数と半角_のみ)設定可能な実フィールド名の長さは、データベースの種類により異なります。
表示フィールド名	本製品内で利用されるフィールド定義の表示用の名称です。
文字列長	文字型の文字列長(DBMSの種類、データ型によりバイト or 文字数)
桁数	数値データ型の全体の桁数
小数桁数	数値データ型のうち少数を格納可能な場合の少数以下の桁数
NULL許可	NULL値の格納が可能かどうか
キー	作成されたテーブルの主キーかどうか(テーブル作成時のみ設定可能)
コメント	フィールド定義のコメント(128文字まで)

3 テーブルの作成

テーブル作成は、マッパー定義保存時に同時に行われます。

エラーが発生時は、エラー情報が画面に表示されますので、内容の修正を行い、再度マッパー定義保存操作を行ってください。

4 テーブル作成後のフィールド情報の変更

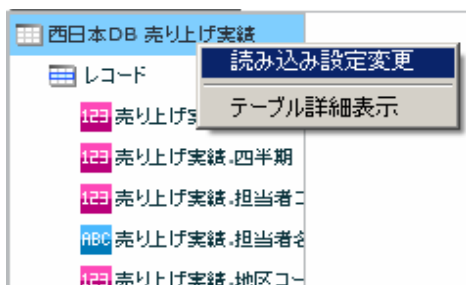
本製品で作成されたテーブルで実テーブル変更権限がある場合、テーブルに対する少々の変更が可能です。可能なことは以下の2点です。

- ・新規フィールドの追加(すでにあるフィールドの下に作成されます)
- ・文字列フィールドの文字列長の拡張

フィールド情報変更の際も、マッパー定義保存時にテーブル変更操作を実行します。

2.6.7 RDB を利用したときの各種メニュー

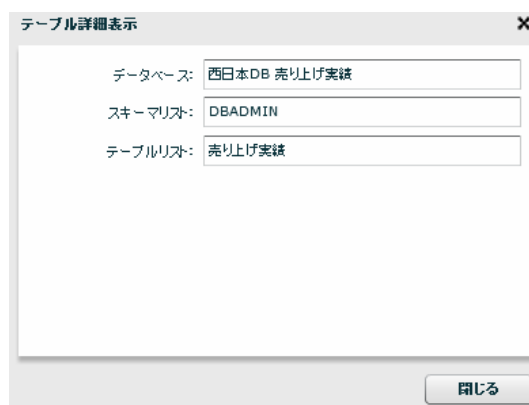
- ・ 転送元テーブル関連



転送元テーブルを選択時に右クリックで表示されるメニュー

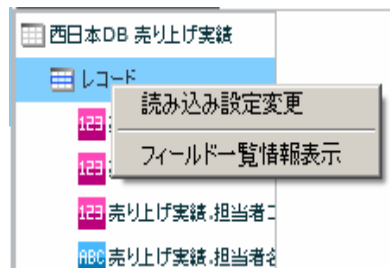


読み込み設定変更



テーブル詳細表示

・転送元レコード関連

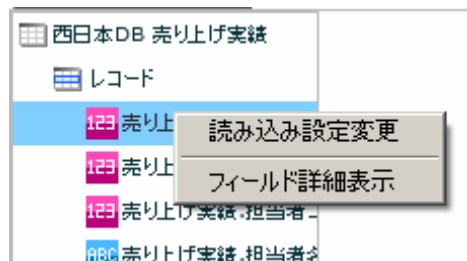


転送元レコードを選択時に右クリックで表示されるメニュー

ポジション	テーブル名	フィールド名	型名	文字列長	桁数	小数桁数	NULL許可	主キー	コメント
1	売り上げ実績	年度	INTEGER	-	10	0	し	し	
2	売り上げ実績	四半期	INTEGER	-	10	0	し	し	
3	売り上げ実績	担当者コード	INTEGER	-	10	0	し	し	
4	売り上げ実績	担当者名	VARCHAR	40	-	-	し	し	
5	売り上げ実績	地区コード	INTEGER	-	10	0	し	し	
6	売り上げ実績	担当地区	VARCHAR	40	-	-	し	し	
7	売り上げ実績	商品コード	INTEGER	-	10	0	し	し	
8	売り上げ実績	商品名	VARCHAR	40	-	-	し	し	
9	売り上げ実績	数量	INTEGER	-	10	0	し	し	
10	売り上げ実績	売り上げ金額	INTEGER	-	10	0	し	し	
11	売り上げ実績	粗利益	INTEGER	-	10	0	し	し	

フィールド一覧情報表示

・転送元フィールド関連



転送元フィールドを選択時に右クリックで表示されるメニュー

フィールド表示名: 売り上げ実績.年度
 フィールド実名: 売り上げ実績.年度
 型: INTEGER
 桁数: 10 小数桁数: 0
 主キー: いいえ

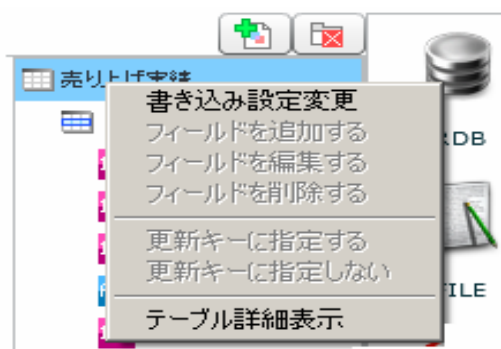
フィールド詳細表示

- ・ 転送先リソース関連

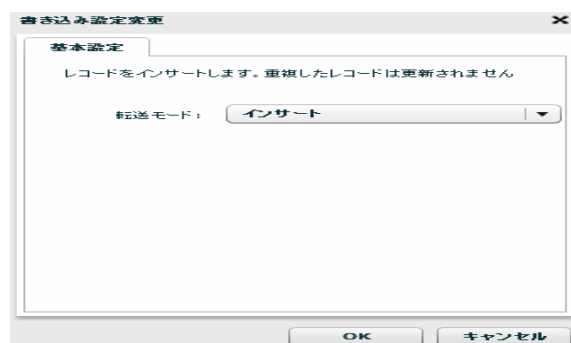


転送先 RDB を選択時に右クリックで表示されるメニュー

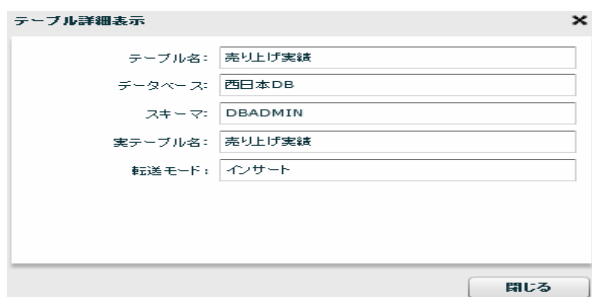
- ・ 転送先テーブル関連



転送先テーブルを選択時に右クリックで表示されるメニュー

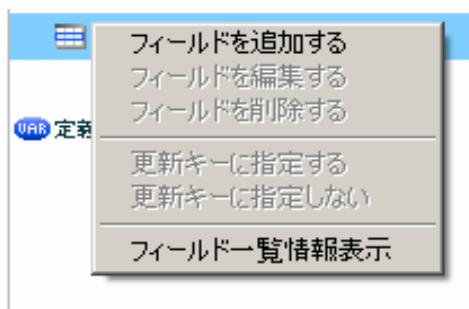


書き込み設定変更

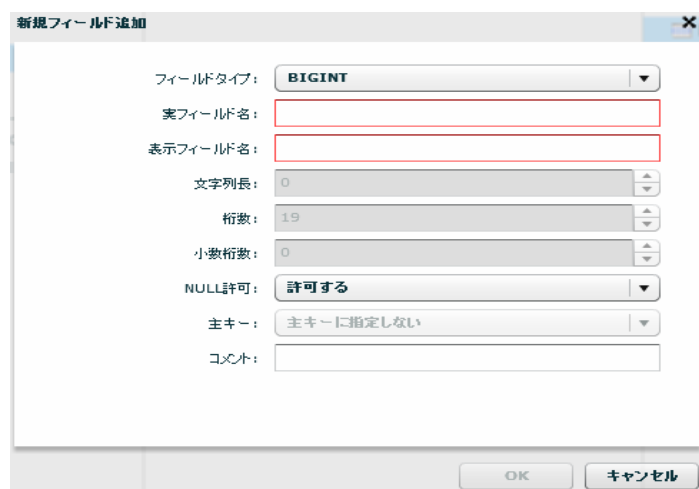


テーブル詳細表示

・ 転送先レコード関連



転送先レコードを選択時に右クリックで表示されるメニュー

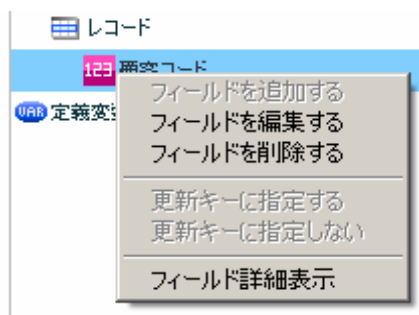


フィールドを追加する



フィールド一覧情報表示

・ 転送先フィールド関連

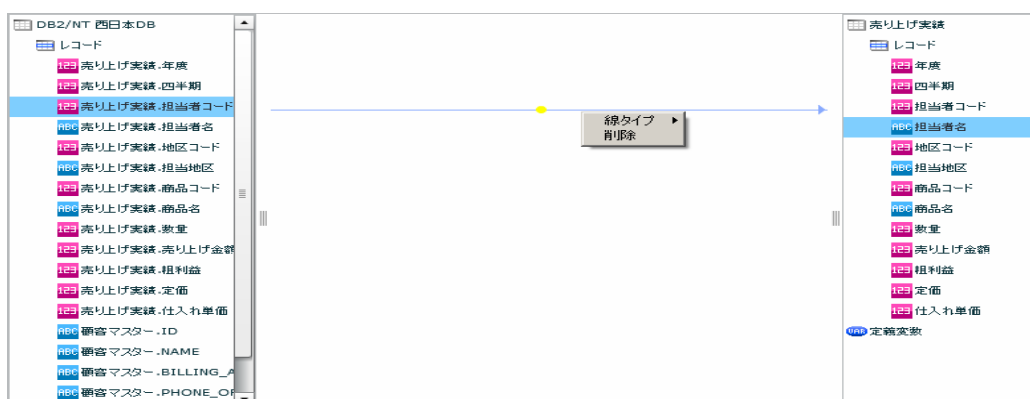


転送先フィールドを選択時に右クリックで表示されるメニュー



フィールド詳細表示

・ 線関連

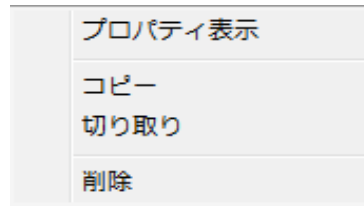


線を選択時に右クリックで表示されるメニュー

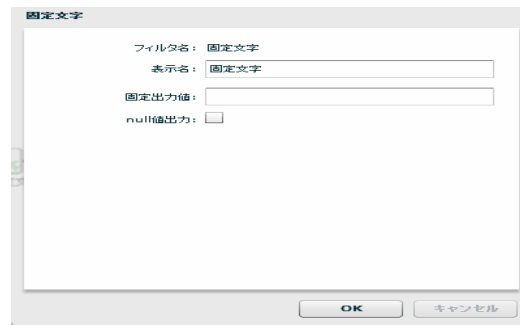
線タイプ 線のスタイルを変更します。

削除 結線を切断します。

- ・フィルタ関連



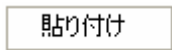
フィルタを選択時に右クリックで表示されるメニュー



フィルタ プロパティ

プロパティ表示	フィルタの設定画面を表示します。
コピー	選択されているフィルタをコピーします。
切り取り	選択されているフィルタを切り取ります。(後でペースト可能です)
削除	選択されているフィルタを削除します。

- ・何もないところで右クリックしたとき

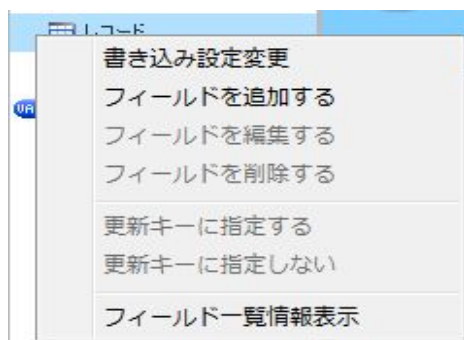


直前の切り取り または コピーをしたフィルタを貼り付けます。

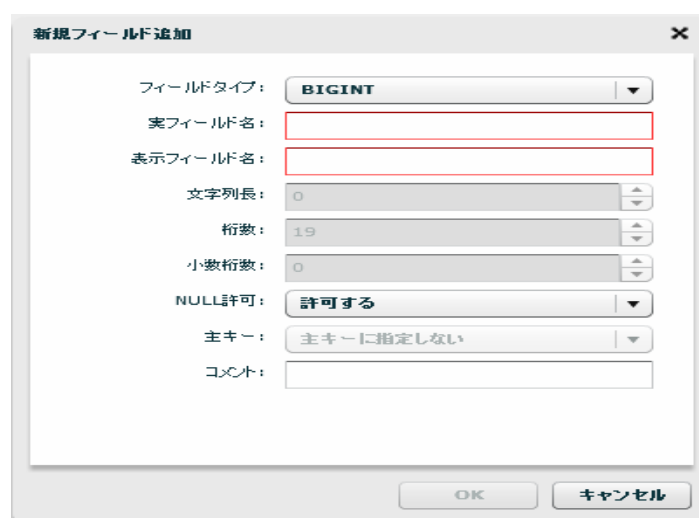
テーブル作成機能で作成されたテーブルの場合は、フィールド作成系の機能も利用可能です。

テーブルの新規作成時(一度もマッパー定義を保存していないとき)

- ・テーブルの新規作成時でレコード関連

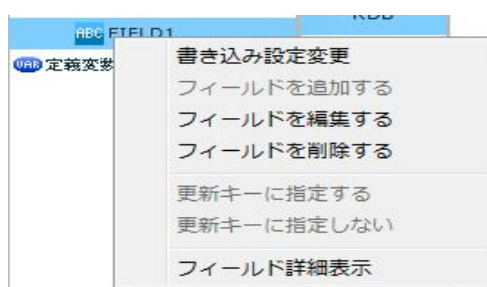


テーブルの新規作成時でレコードを選択時に右クリックで表示されるメニュー



新規フィールド追加

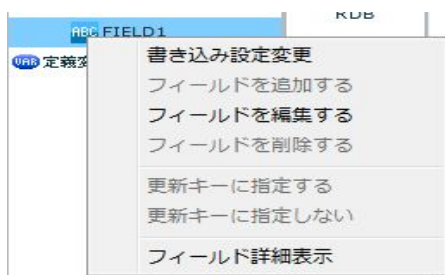
- ・ テーブルの新規作成時で作成されたフィールド関連



テーブルの新規作成時で作成されたフィールドを選択時に右クリックで表示されるメニュー

新規フィールド編集

テーブル作成機能にて作成済テーブルのフィールドの場合(一度以上マッパー定義を保存した後)



テーブル作成機能にて作成済テーブルのフィールドを選択時に右クリックで表示されるメニュー

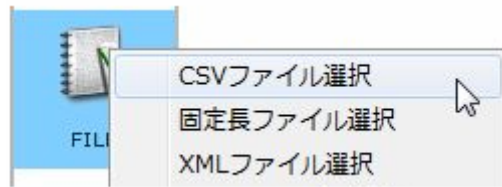
フィールド編集



編集可能な項目はデータ型により異なります。
文字型の文字列長が拡張できる場合があります。

2.6.8 CSV ファイル選択

CSV ファイルをデータ転送元を選択



CSV ファイル選択



転送元ファイル選択

ファイル

転送したいデータが格納されているファイルを選択します。

基本設定

<ファイル名>

<ファイル> で指定したファイルの名称と、一部データをプレビュー表示します。

<ファイルの種類>

<ファイル> で指定したフォルダのうち利用して表示するファイルの絞り込みをします。絞り込みにはウィルドカードを利用します。CSV ファイル選択の場合 “*.csv” と入力されています。

<文字コード>

サーバーデフォルトコードとは、本製品サーバーを導入した OS のデフォルトを意味します。

<ファイルの1行目をフィールド情報として読む>

ファイルの1行目のデータを読み込んで、マッピング画面でフィールド名として使用します。

<ファイルの1行目をデータとして扱わない>

画面例のようにチェックがない場合は、ファイルの1行目をデータとみなし、1行目のデータから読み

込みが開始されます。チェックを入れると1行目を見出し行とみなし、2行目のデータから読み込みが開始されます。

詳細設定



転送元選択 詳細設定

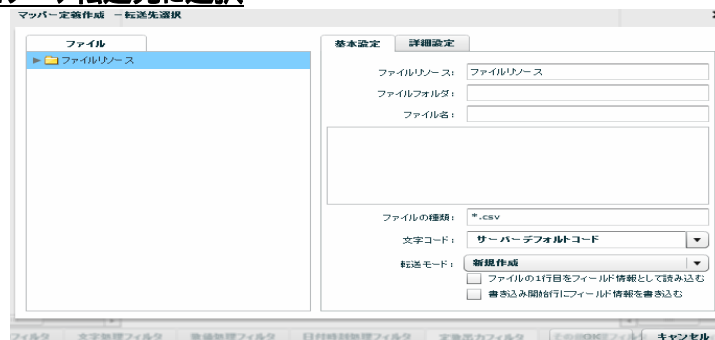
<データ区切り文字>

ファイル中の各データを区切る文字を指定します。基本設定タブのプレビュー表示を利用し、正しい区切り文字を「カンマ/タブ/スペース」から選択するかまたは、入力し指定します。

<データ囲み文字>

ファイル中の各データを囲む文字を指定します。基本設定タブのプレビュー表示を利用し、正しい囲み文字を「' / " / なし」から選択するかまたは、入力し指定します。

CSV ファイルをデータ転送先に選択



転送先ファイル選択

ファイル

データ転送先のファイルを指定します。

基本設定

データ転送先のファイルを指定します。

<ファイルの種類>

<ファイル>で指定したフォルダのうち利用して表示するファイルの絞込みをします。絞込みにはワールドカードを利用します。CSVファイル選択の場合“*.csv”と入力されています。

<書き込み文字コード>

サーバーデフォルトコードとは、本製品サーバーを導入したOSのデフォルトを意味します。

<ファイルの1行目をフィールド情報として読む>

ファイルの1行目のデータを読み込んで、フィールド名として使用します。

<書き込み開始行にヘッダ情報を書き込む>

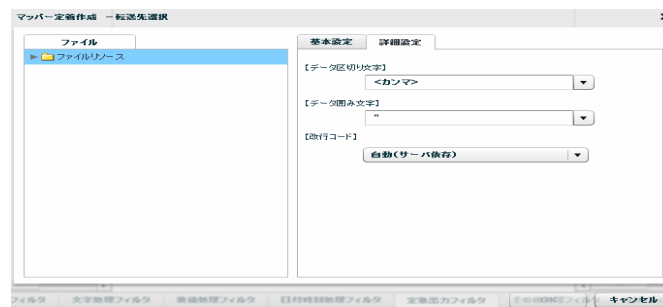
書き込みを開始する際にヘッダ情報書き込みの有無を指定します。チェックを外すとヘッダ情報は書き込まれません。ヘッダ情報には出力フィールドに指定した設定が使われます。

<転送モード>

データ転送時の書き込み方法を指定します。

- ・上書き(ない場合新規) : 既存データはクリアされ、転送されたデータで更新されます。ファイルが見つからない場合は同名のファイルが作成されます。
- ・追加書き込み : 既存データの最後尾に追記されます。ファイルが見つからない場合は同名のファイルが作成されます。

詳細設定



転送先選択 詳細設定

<データ区切り文字>

ファイル中の各データを区切る文字を指定します。区切り文字は「カンマ/タブ/スペース」から選択するかまたは、新たに入力し指定します。

<データ囲み文字>

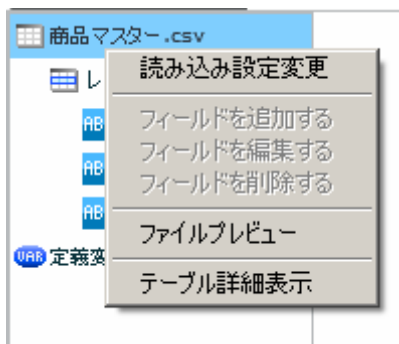
ファイル中の各データを囲む文字を指定します。囲み文字は「' / " / なし」から選択するかまたは、入力し指定します。

<改行コード>

ファイル中の各データ行の改行コードを指定します。「自動(サーバー依存)/CR+LF/CR/LF」から選択します。

2.6.9 CSV ファイルを利用したときの各種メニュー

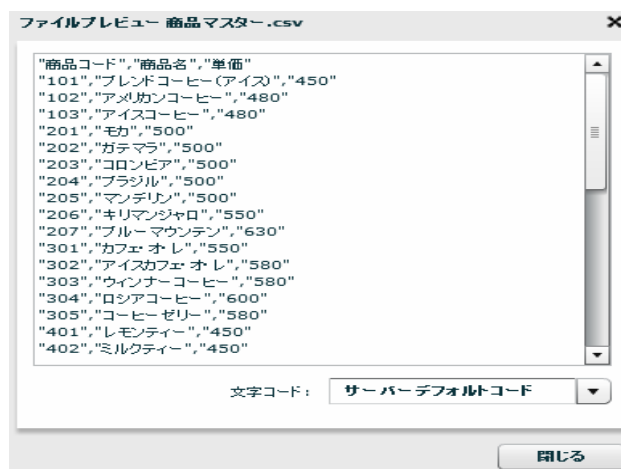
- ・転送元ファイル関連



転送元ファイルを選択時に右クリックで表示されるメニュー

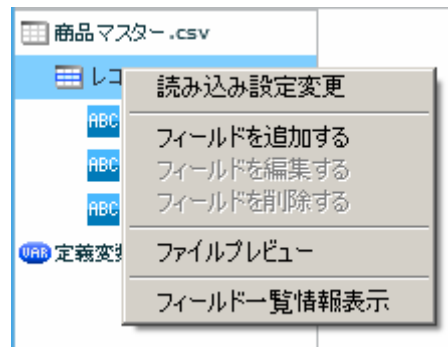


読み込み設定の「基本設定」タブと「詳細設定」タブ

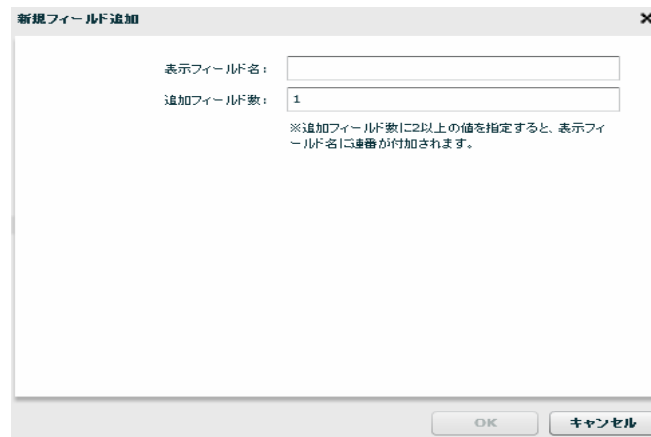


ファイルプレビュー表示

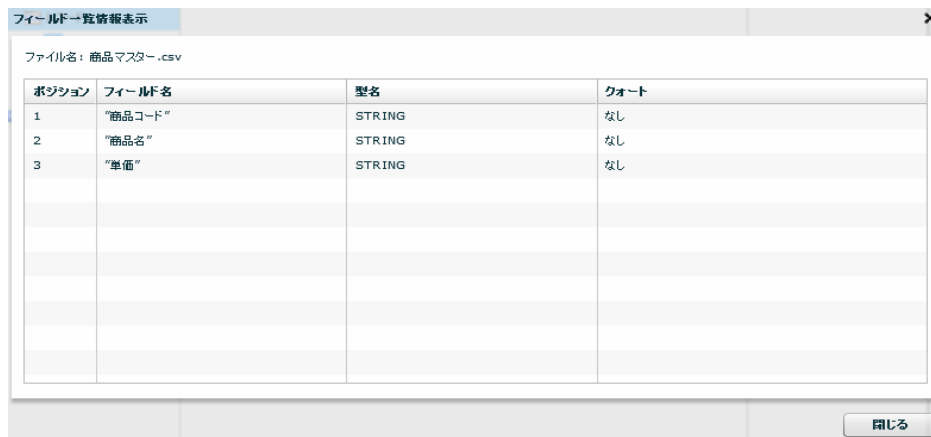
・転送元レコード関連



転送元レコードを選択時に右クリックで表示されるメニュー

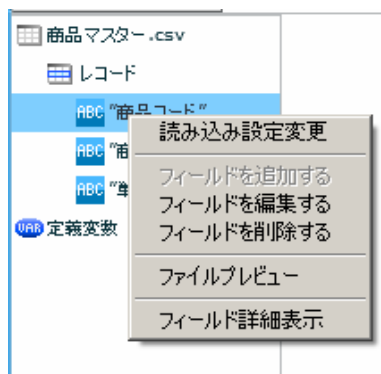


フィールドを追加

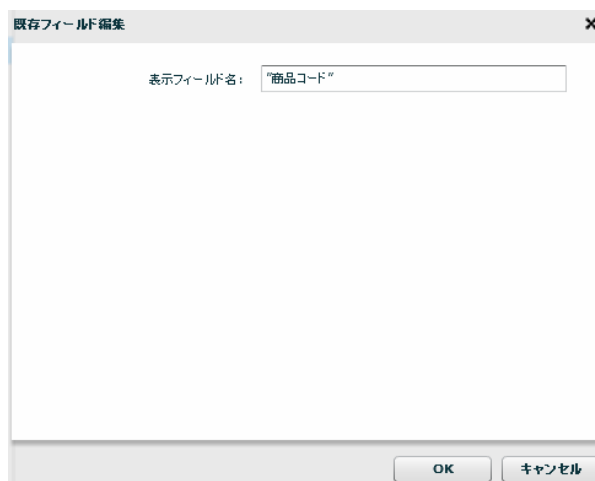


フィールド一覧情報

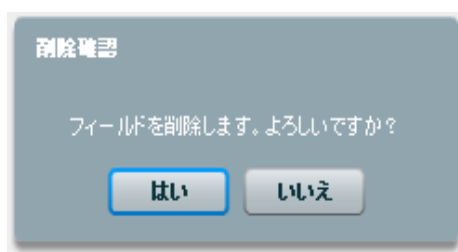
・転送元フィールド関連



転送元フィールドを選択時に右クリックで表示されるメニュー



フィールド編集



フィールド削除



フィールド詳細

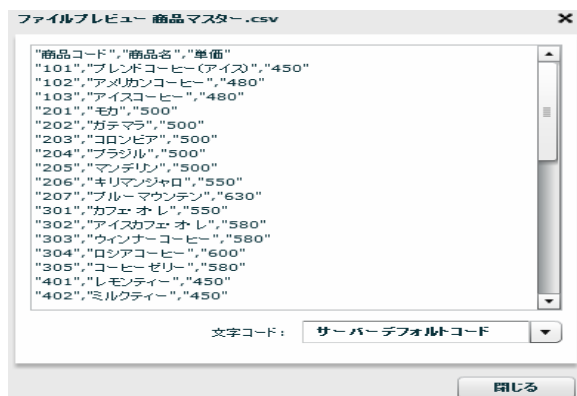
・転送先テーブル関連



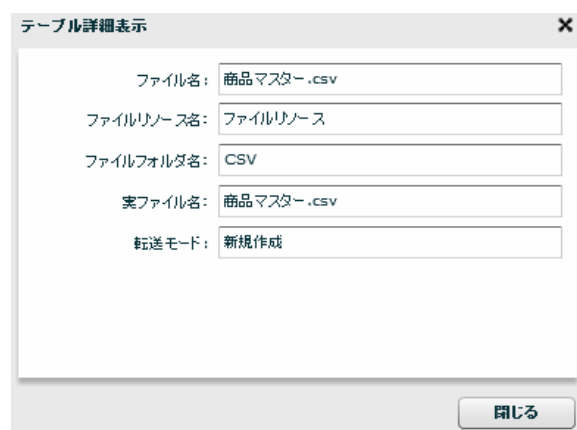
転送先テーブルを選択時に右クリックで表示されるメニュー



書き込み設定変更の「基本設定」タブと「詳細設定」タブ

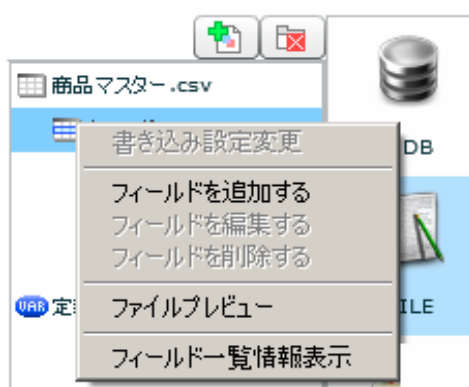


ファイルプレビュー

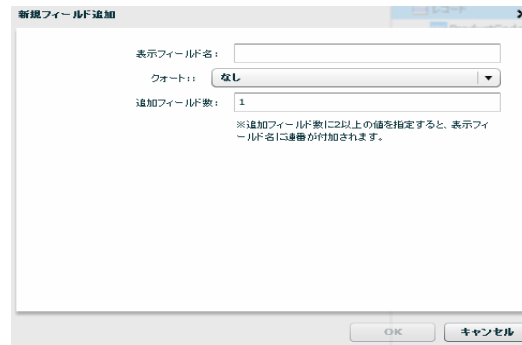


テーブル詳細表示

- ・転送先レコード関連



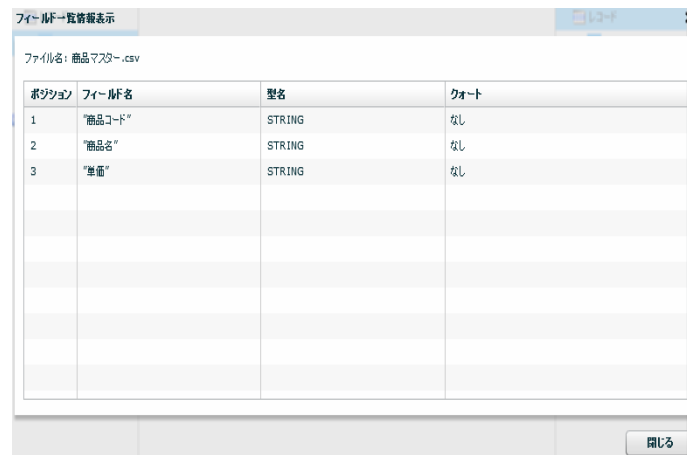
転送先レコードを選択時に右クリックで表示されるメニュー



フィールドを追加

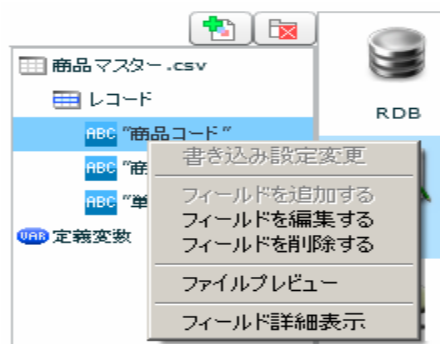


ファイルプレビュー

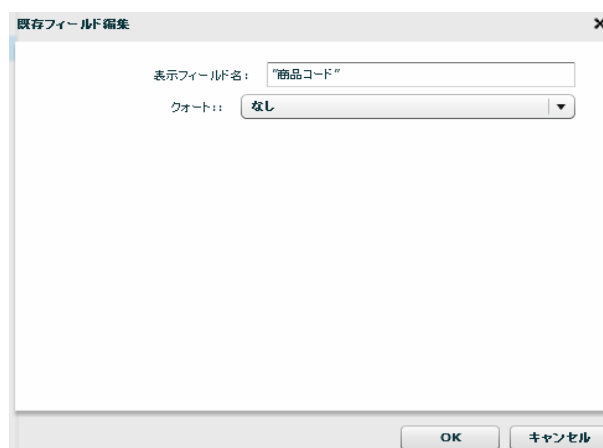


フィールド一覧情報表示

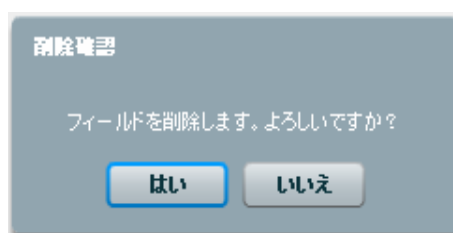
・転送先フィールド関連



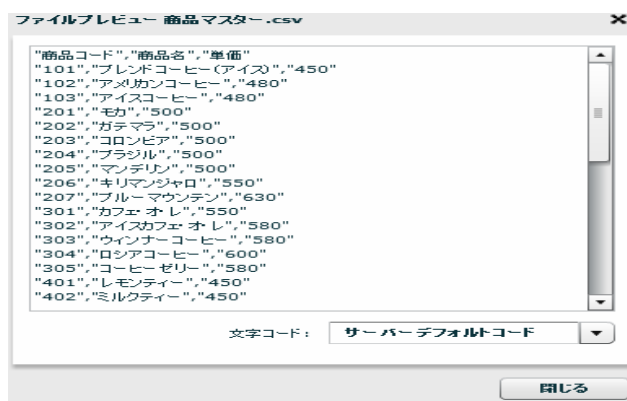
転送先フィールドを選択時に右クリックで表示されるメニュー



フィールドを編集



フィールドを削除



ファイルプレビュー



フィールド詳細表示

2.6.10 固定長ファイル選択

固定長ファイルをデータ転送元を選択



固定長ファイル選択



転送元ファイル選択

ファイル

転送したいデータが格納されているファイルを選択します。

基本設定

<ファイルリソース> <ファイルフォルダ> <ファイル名>

<ファイル> で指定したファイルのリソース、フォルダ、名称と、一部データをプレビュー表示します。

定義変数を指定できます。

<ファイルの種類>

<ファイル> で指定したフォルダのうち利用して表示するファイルの絞込みをします。絞込みにはワールドカードを利用します。固定長ファイル選択の場合 “*.*” と入力されています。

<改行コード>

改行コードを NONE,CR+LF,CR,LF の中から指定します。

NONE : 指定したレコード長でレコードを区切っていきます。

CR+LF、CR、LF : 指定したレコード長でレコードを区切っていきます。ただし、レコード内に、ここで指定した改行コードがあった場合は、改行コードまでをレコードとして区切ります。

レコード長: 46byte = 10 + 12 + 20 + 4

	10byte	12byte	20byte	4byte
1				CR+LF
2			CR+LF	
3				CR+LF

上記のように、レコードによってレコード長が異なり、改行されている場合は【レコード長】に 46byte、

【改行コード】に CR+LF を指定することで、上記のような固定長ファイルにも対応可能です。

<文字コード>

読み込み時の文字コードを設定します。

サーバーデフォルトコードとは、本製品サーバを導入した OS のデフォルトを意味します。

<レコード長>

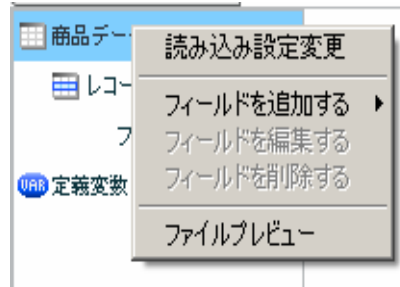
1レコードとして読み込むバイト長を指定します。

固定長ファイルをデータ転送先に選択

転送先ファイル選択

2.6.11 固定長ファイルを利用したときの各種メニュー

- ・転送元ファイル関連



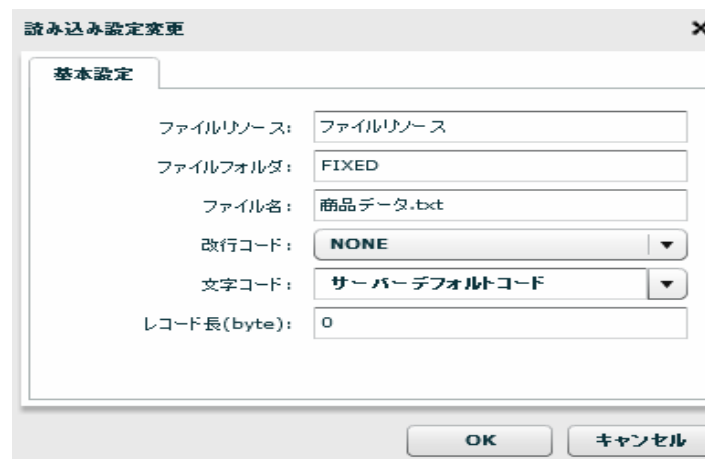
転送元ファイルを選択時に右クリックで表示されるメニュー

[読み込み設定変更]

入力ファイル設定を変更します。

基本設定

ファイルリソース、ファイルフォルダ、ファイル名、改行コード、文字コード、レコード長の変更をすることができます。

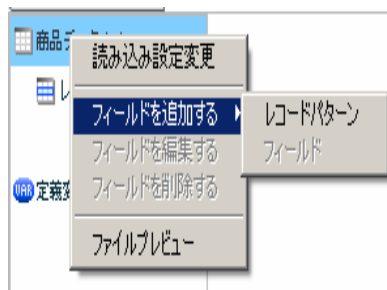


読み込み設定変更 基本設定

[フィールドを追加する] [レコードパターン]

レコードパターンを追加します。転送元では複数のレコードパターンを設定できます。

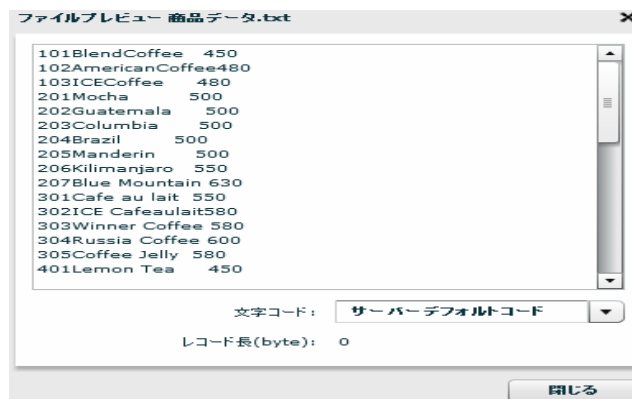
一つの固定長ファイル内に複数のフォーマットがある場合でも対応可能です。



フィールドを追加する レコードパターン

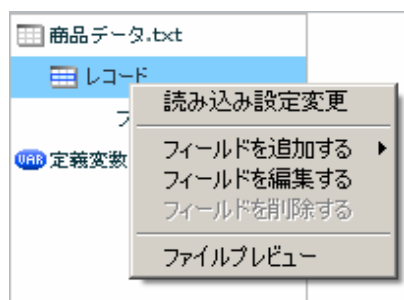
[ファイルプレビュー]

ファイルプレビューを表示します。



ファイルプレビュー

・転送元レコード関連



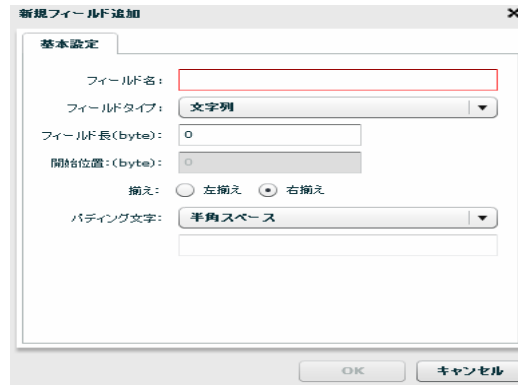
転送元レコードを選択時に右クリックで表示されるメニュー

[読み込み設定変更]

入力ファイル設定を変更します。

[フィールドを追加する] [フィールド]

フィールドを追加します。



新規フィールド追加

フィールド名	フィールドの名前を指定します。
フィールドタイプ	レコード識別子、文字列、ゾーン 10 進、パック 10 進、バイナリからフィールドの型を選択します。
<ul style="list-style-type: none"> ・レコード識別子 ・文字列 ・ゾーン 10 進 ・パック 10 進 ・バイナリ 	レコードパターンを識別する文字列です。複数のレコードパターンを設定した時に、この値からどのレコードパターンか判定します。詳細は 2.6.12 をご参照ください。
フィールド長	フィールドのバイト長を指定します。
開始位置	そのフィールドの開始位置 (byte) が表示されます。
揃え	指定したフィールドデータがフィールドに対し左揃えか右揃えかを指定します。
パディング文字	指定したフィールドがパディング文字で埋められているときそのパディング文字を指定することでデータだけを読むことができます。パディング文字として半角スペース、全角スペース、Null 文字、直接入力、文字コード入力を選択できます。

フィールドタイプにゾーン 10 進、バック 10 進を選択した場合詳細設定画面が追加されます。



フィールド詳細設定 (ゾーン 10 進)

ゾーン 10 進数について

小数部桁数

小数点以下の桁数です。小数部桁数 4 のゾーン 10 進フィールドは 00115.1109 などと表されます。

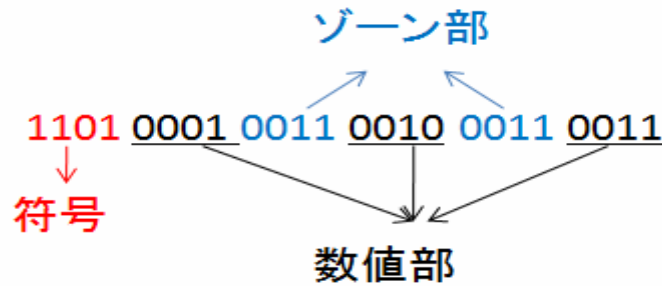
符号形式

符号を表す 4 ビットの位置を指定します。(ゾーン 10 進数のみ)

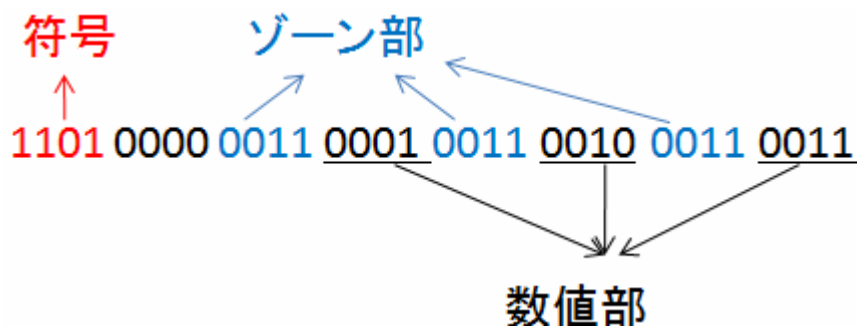
符号

数値の負を決定する 4 ビットを指定します。デフォルトでは D(16 進数) = 1101(2 進数)に設定されています。

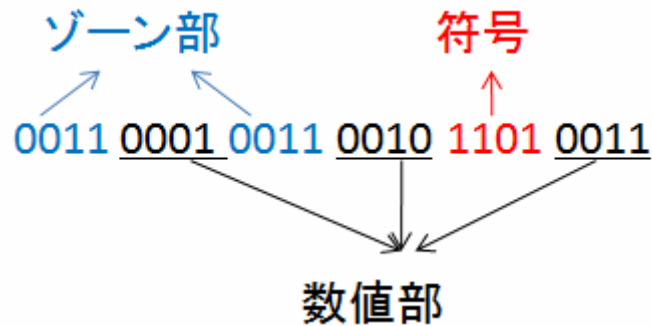
以下に-123 という数値をそれぞれの符号形式で表したバイナリ情報を説明します。



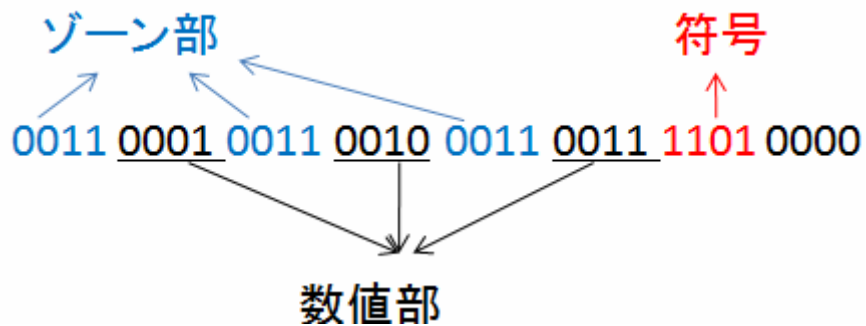
符号形式：前



符号形式：前(単独)



符号形式：後



符号形式：後（単独）

開始位置、揃え、パディング文字はフィールドタイプが文字列の時のみ指定できます。

例：以下のように、商品番号(10byte)、商品名(22byte)、価格(4byte)で空白部分はパディング文字として半角スペースで埋められている入力データを読み込みます。

商品番号	商品名	価格
102	アメリカンコーヒ	480
103	アイスコーヒー	500

← 10byte
← 22byte
← 4byte

入力固定長データ

フィールドを以下の設定で追加していきます。

新規フィールド追加

基本設定

フィールド名: 商品番号

フィールドタイプ: 文字列

フィールド長(byte):

開始位置:(byte): 0

揃え: 左揃え 右揃え

パディング文字: 半角スペース

OK キャンセル

商品番号フィールド

新規フィールド追加

基本設定

フィールド名: 商品名

フィールドタイプ: 文字列

フィールド長(byte): 22

開始位置:(byte): 10

揃え: 左揃え 右揃え

パディング文字: 半角スペース

OK キャンセル

商品名フィールド

新規フィールド追加

基本設定

フィールド名: 価格

フィールドタイプ: 文字列

フィールド長(byte): 4

開始位置:(byte): 32

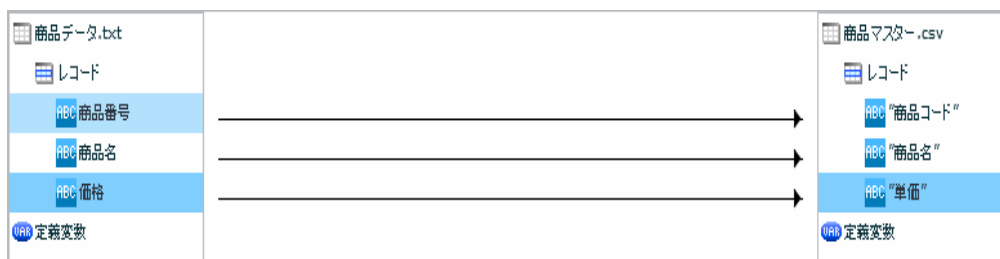
揃え: 左揃え 右揃え

パディング文字: 半角スペース

OK キャンセル

価格フィールド

入力のフィールドが追加できたら CSV ファイルに書き込みます。



CSV 書き込み

出力されたデータは以下です。

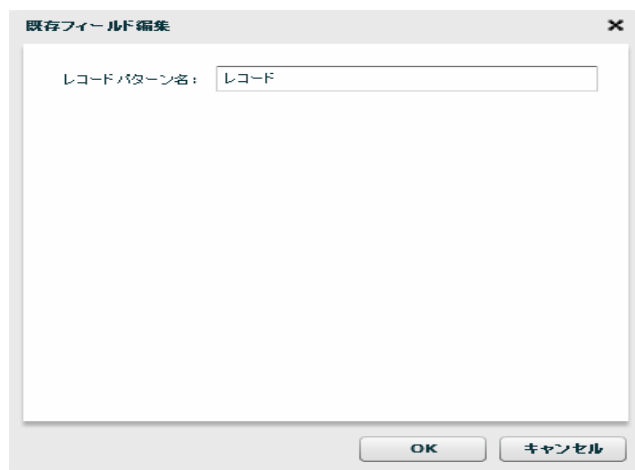
```
商品番号,商品名,価格  
102,アメリカンコーヒ,480  
103,アイスコーヒー,500
```

出力データ

パディング文字（今回の場合半角スペース）を除いたデータだけを出力することができたのが分かります。

[フィールドを編集する]

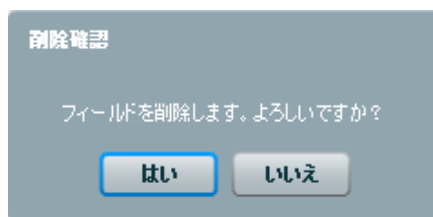
レコードパターンをの名前を編集することができます。



レコードパターン編集

[フィールドを削除する]

レコードパターンが複数ある場合のみレコードパターンを削除することができます。

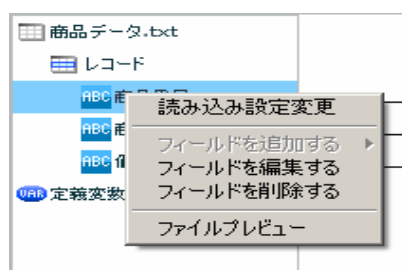


レコードパターン削除

[ファイルプレビュー]

ファイルプレビューを表示します。

・転送元フィールド関連



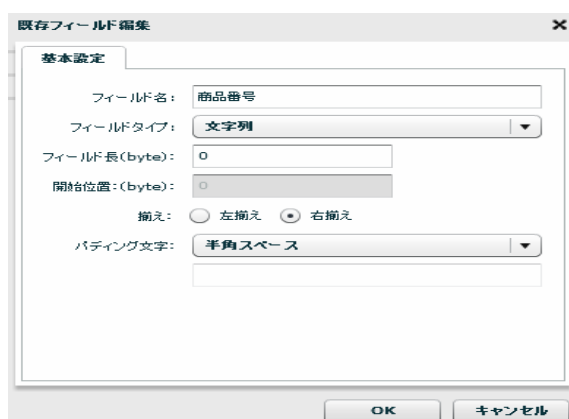
転送元フィールドを選択時に右クリックで表示されるメニュー

[読み込み設定変更]

入力ファイル設定を変更します。

[フィールドを編集する]

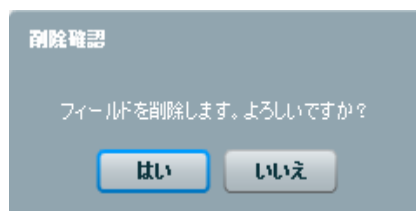
入力フィールドを編集します。



フィールド編集

[フィールドを削除する]

フィールドを削除します。

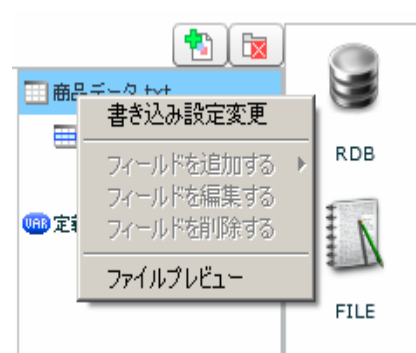


フィールド削除

[ファイルプレビュー]

ファイルプレビューを表示します。

・転送先ファイル関連

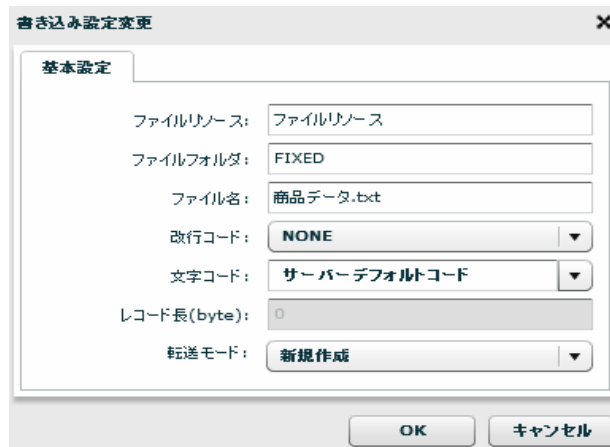


転送先ファイルを選択時に右クリックで表示されるメニュー

[書き込み設定変更]

基本設定

新たに作成したいファイル名を入力し、改行コード、文字コード、転送モードの設定をします。



書き込み設定変更 基本設定

[OK]を押すと新たなファイル名の固定長ファイルが作成されます。

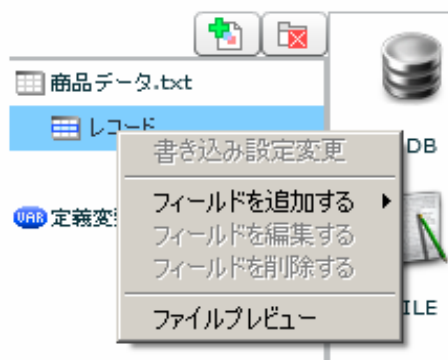
[ファイルプレビュー]

ファイルプレビューを表示します。



ファイルプレビュー

・ 転送先レコード関連



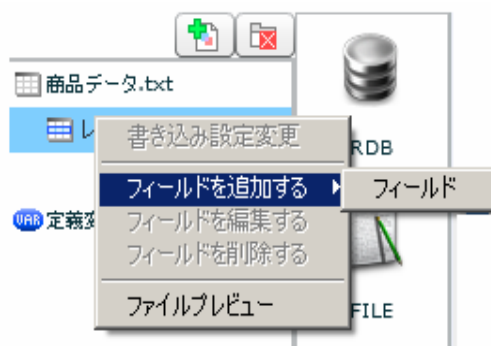
転送先レコードを選択時に右クリックで表示されるメニュー

[書き込み設定変更]

出力ファイル設定を変更します。

[フィールドを追加する] - [フィールド]

フィールドを追加します。転送先ではレコードパターンは一つで固定のため、レコード識別子はありません。

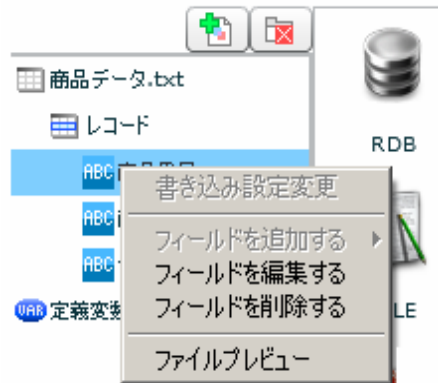


フィールド追加

[ファイルプレビュー]

ファイルプレビューを表示します。

・ 転送先フィールド関連



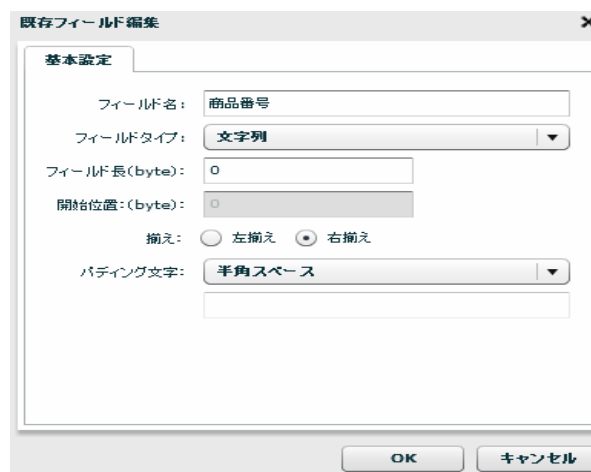
転送先フィールドを選択時に右クリックで表示されるメニュー

[書き込み設定変更]

出力ファイル設定を変更します。

[フィールドを編集する]

出力フィールドを編集します。



フィールド編集

[フィールドを削除する]

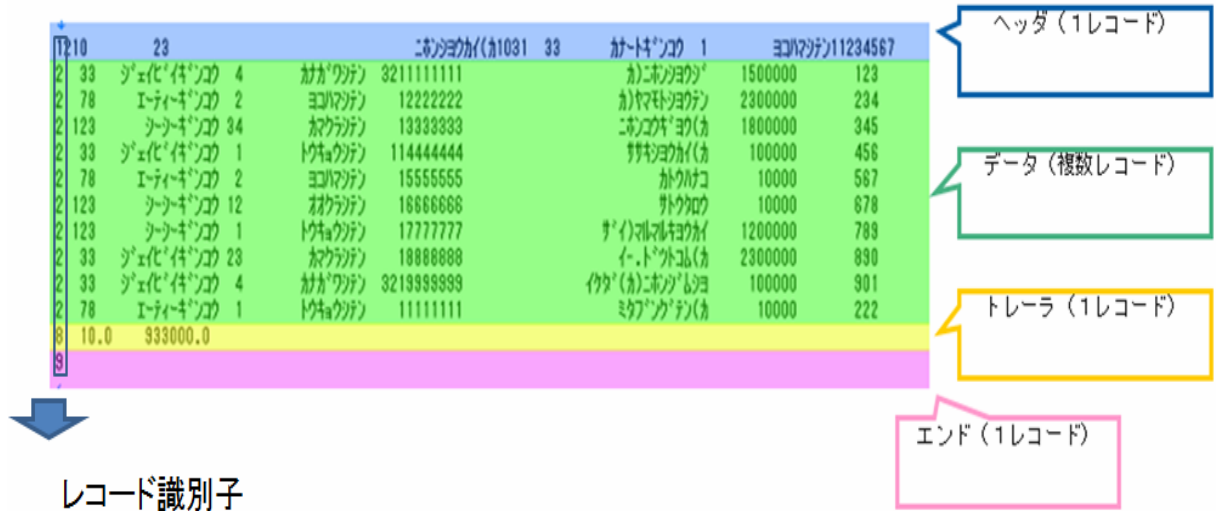
出力フィールドを削除します。

[ファイルプレビュー]

ファイルプレビューを表示します。

2.6.12 固定長ファイル転送におけるのループ処理

固定長ファイル転送では以下のような複数のレコードパターンをもつ入力データを転送することもできます。
 (詳しい固定長ファイル転送定義作成手順は本製品のチュートリアルをご参照ください。)



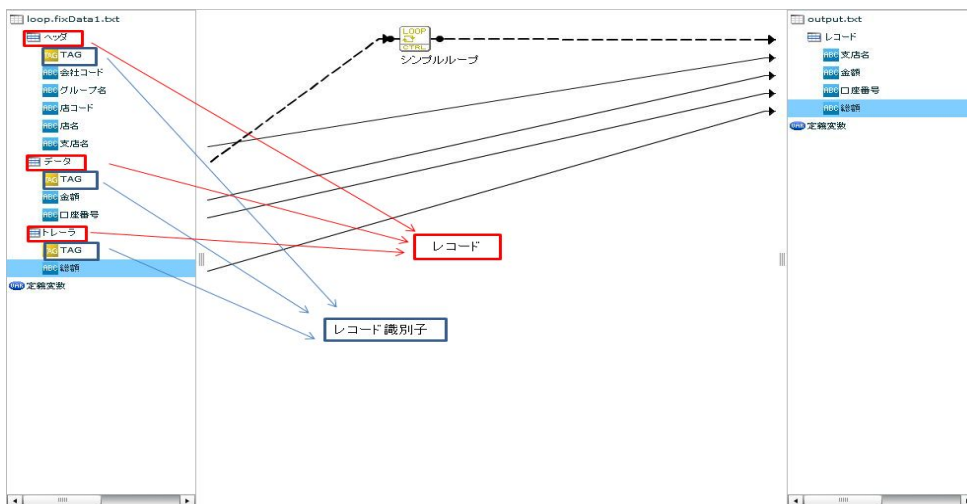
固定長データには複数レコードからなるデータにヘッダ、トレーラ、エンドなどの付加情報がのっているものがあります。

こういった入力データを読み込むには、複数レコードをもつデータだけをループさせる必要があります。そのためレコード識別子と呼ばれるレコードを区別するためのフィールドを用意し、ループさせるレコードとそのフィールドを識別可能にします。

(レコードパターンを複数もつ固定長データを扱う場合必ずレコード識別子を指定してください。)

下図に複数レコードをもつデータだけをループさせる定義の例を挙げます。

[制御フィルタ]のシンプルループフィルタをループさせたい入力レコードと出力レコードの間に結線します。



固定長ファイル転送におけるのループ処理

この定義で出力されるレコードは、下記のフォーマットで出力されます。

1レコード目：ヘッダ・データ1・トレーラ

2レコード目：ヘッダ・データ2・トレーラ

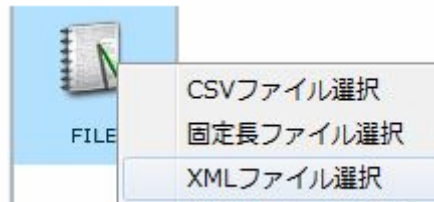
.....

最終レコード：ヘッダ・最終データ・トレーラ

というように、ループをつないだレコードだけがループ処理を受け、ヘッダとトレーラの固有データがそれに付加され出力されます。

2.6.13 XML ファイル選択

XML ファイルをデータ転送元を選択



XML ファイル選択



転送元ファイル選択

ファイル

転送したいデータが格納されているファイルを選択します。

基本設定

<ファイル名>

<ファイル> で指定したファイルの名称と、一部データをプレビュー表示します。

<ファイルの種類>

<ファイル> で指定したフォルダのうち利用して表示するファイルの絞込みをします。絞込みにはウィルドカードを利用します。XML ファイル選択の場合 “*.xml ” と入力されています。

<読み込み文字コード>

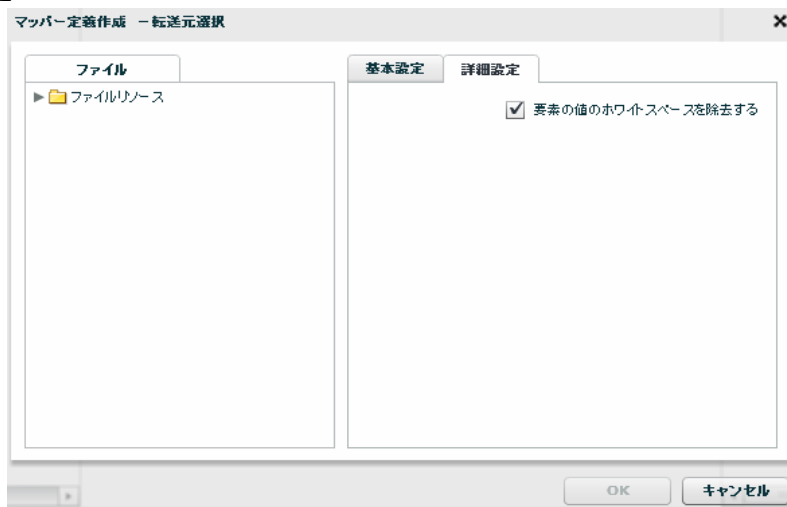
読み込み時の文字コードを指定します。

サーバーデフォルトコードとは、本製品サーバを導入した OS のデフォルトを意味します。

< XML 構造を取得する >

指定したファイルの XML 構造を取得します。

詳細設定



転送元 詳細設定

< 要素の値のホワイトスペースを除去する >

要素中の文字列にが含まれるスペースやタブを削除します。

XML ファイルをデータ転送先に選択

転送先ファイル選択

ファイル

データ転送先のファイルを指定します。

基本設定

データ転送先のファイルを指定します。

<ファイルの種類>

<ファイル> で指定したフォルダのうち利用して表示するファイルの絞込みをします。絞込みにはワールドカードを利用します。XML ファイル選択の場合 “*.xml ” と入力されています。

<書き込み文字コード>

サーバーデフォルトコードとは、本製品サーバを導入した OS のデフォルトを意味します。

<XML 構造を取得する>

指定したファイルの XML 構造を取得します。

<転送モード>

データ転送時の書き込み方法を指定します。転送先が XML の場合、転送モードは「新規作成」が固定で指定されています。

- ・新規作成 既存データはクリアされ、転送されたデータで更新されます。ファイルが見つからない場合は同名のファイルが作成されます。

詳細設定



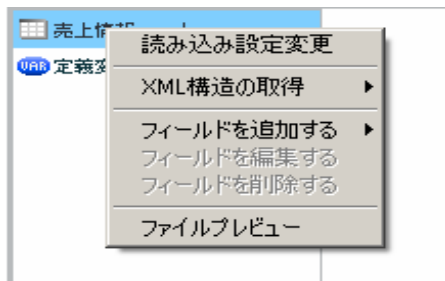
転送先 詳細設定

<書き込み時インデント処理を行う>

要素の先頭の字下げを行います。

2.6.14 XML ファイルを利用したときの各種メニュー

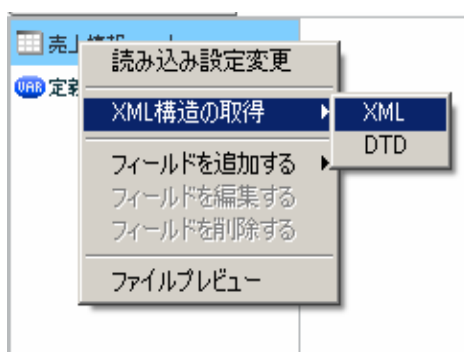
- ・転送元ファイル関連



転送元ファイルを選択時に右クリックで表示されるメニュー



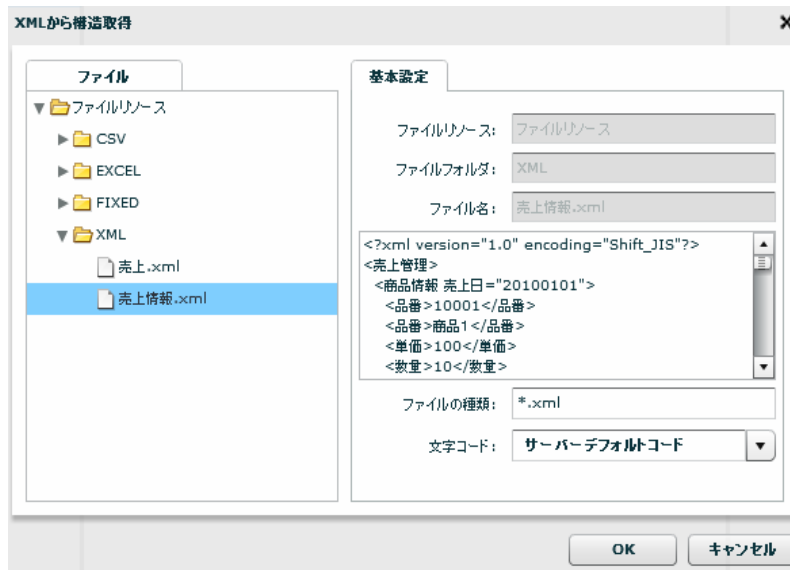
読み込み設定変更の「基本設定」タブと「詳細設定」タブ



XML 構造の取得

[XML 構造の取得] [XML]

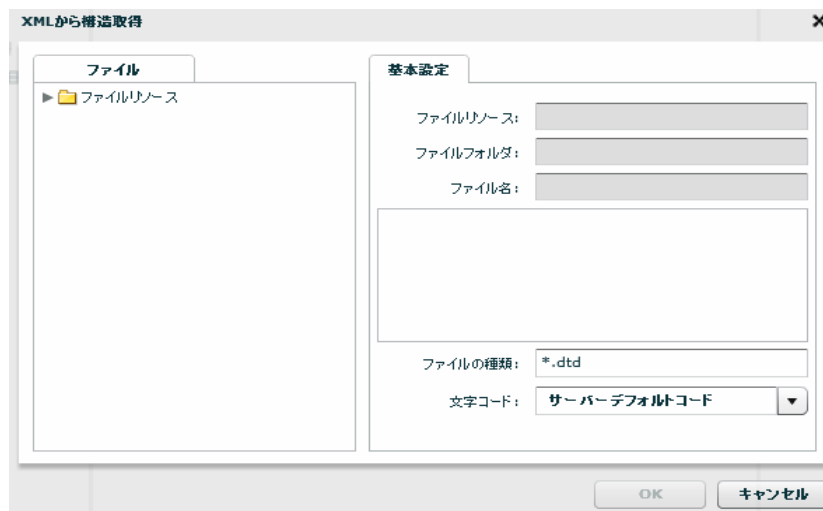
指定した XML ファイルから XML 構造を取得します。



XML 構造取得先指定 (XML ファイル)

[XML 構造の取得] [DTD]

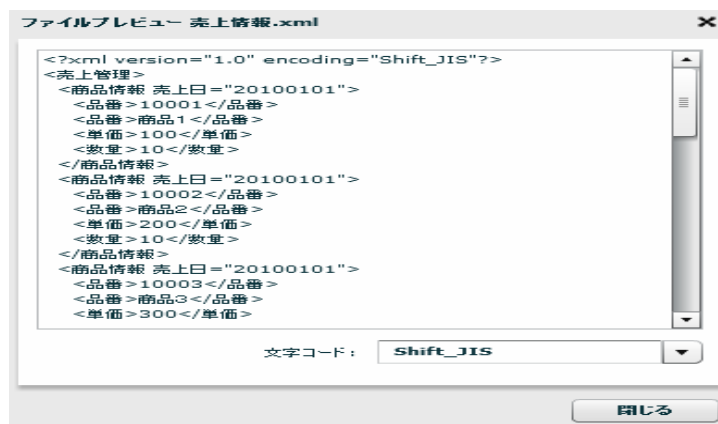
指定した DTD ファイルから XML 構造を取得します。



XML 構造取得先指定 (DTD ファイル)

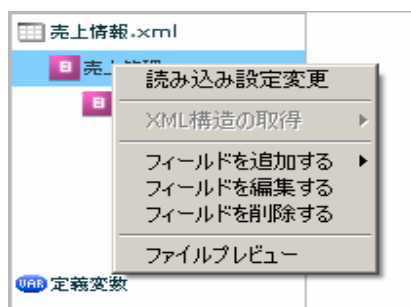
[ファイルプレビュー]

ファイルプレビューを表示します。

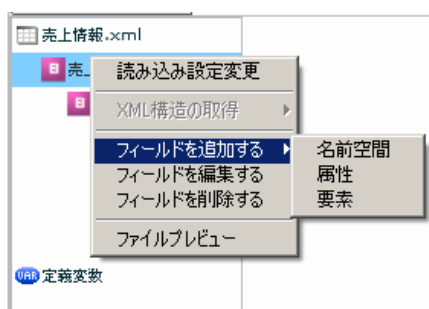


ファイルプレビュー

・転送元レコード関連



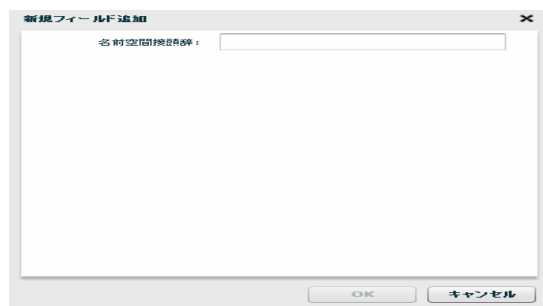
転送元レコードを選択時に右クリックで表示されるメニュー



フィールドを追加する

[フィールドを追加する] [名前空間]

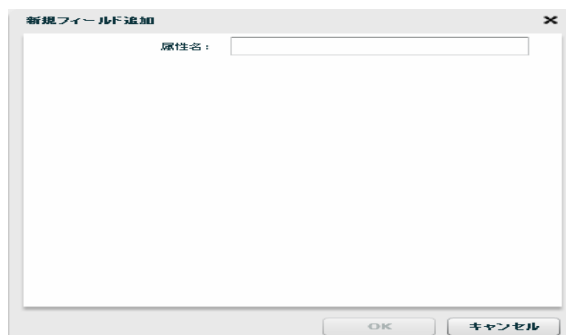
入力した接頭辞の名前空間をフィールドに追加します。



フィールド追加 名前空間

[フィールドを追加する] [属性]

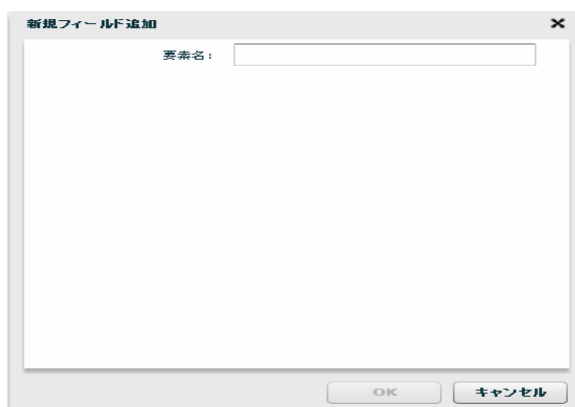
入力した名前の属性をフィールドに追加します。



フィールド追加 属性

[フィールドを追加する] [要素]

入力した名前の要素をフィールドに追加します。



フィールド追加 要素

[フィールドを編集する]

選択したフィールドの名前を変更できます。

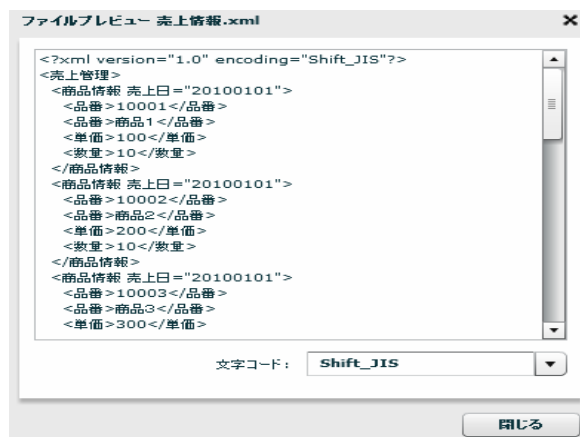


フィールド編集

[フィールドを削除する]

選択したフィールドを削除できます。

[ファイルプレビュー]



ファイルプレビュー

・ 転送先ファイル関連

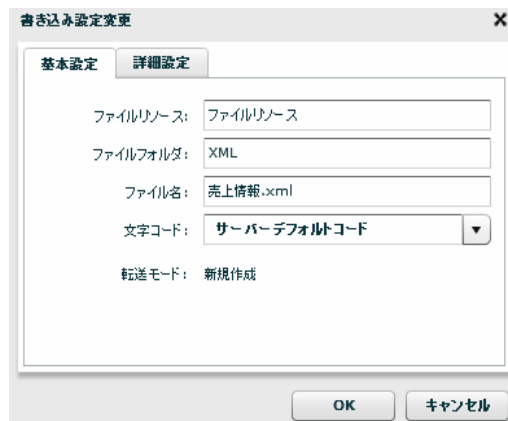


転送先ファイルを選択時に右クリックで表示されるメニュー

[書き込み設定変更]

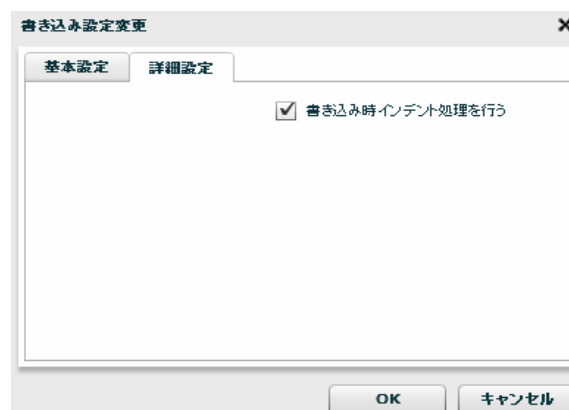
基本設定

新たに作成したいファイル名を入力し、書き込み文字コード、転送モードの設定をします。



書き込み設定変更 基本設定

詳細設定



書き込み設定変更 詳細設定

書き込み時インデント処理の有無を設定します。

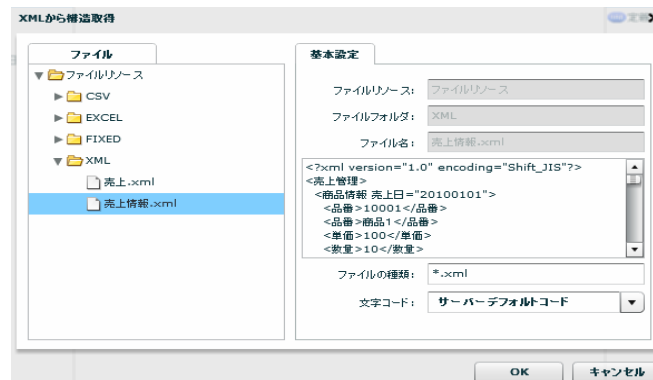
[OK]を押すと新たなファイル名のXMLファイルが作成されます。



XML 構造の取得

[XML 構造の取得] [XML]

指定したXMLファイルからXML構造を取得します。



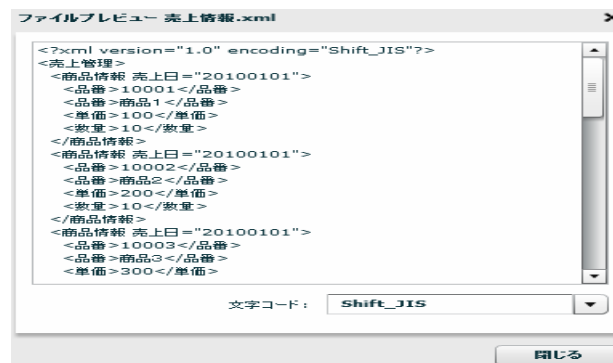
XML 構造取得先指定 (XML ファイル)

[XML 構造の取得] [DTD]

指定したDTDファイルからXML構造を取得します。

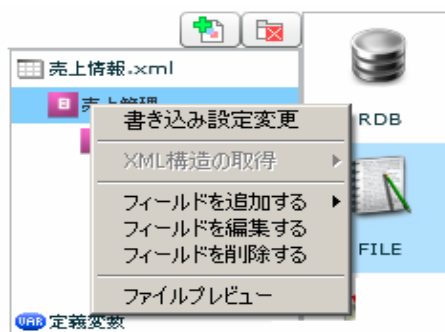
[ファイルプレビュー]

ファイルプレビューを表示します。



ファイルプレビュー

・転送先レコード関連



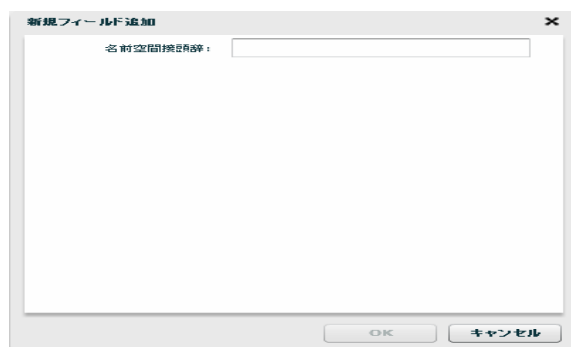
転送先レコードを選択時に右クリックで表示されるメニュー



フィールドを追加する

[フィールドを追加する] [名前空間]

入力した接頭辞の名前空間をフィールドに追加します。



フィールド追加 名前空間

[フィールドを追加する] [属性]

入力した名前の属性をフィールドに追加します。



フィールド追加 属性

[フィールドを追加する] [要素]

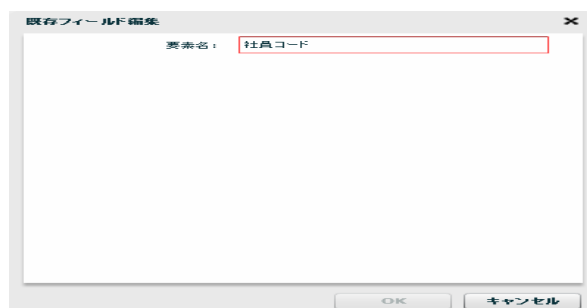
入力した名前の要素をフィールドに追加します。



フィールド追加 要素

[フィールドを編集する]

選択したフィールドの名前を変更できます。



フィールド編集

[フィールドを削除する]

選択したフィールドを削除できます。

[ファイルプレビュー]



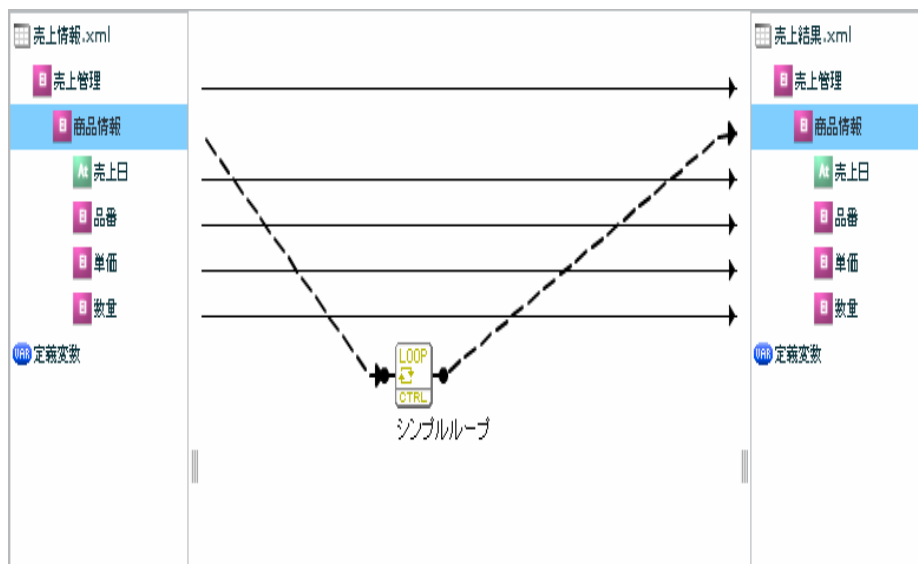
ファイルプレビュー

2.6.15 XML ファイル転送におけるのループ処理

一般的なデータ転送ではデフォルトでループ処理を行います、XML ファイル転送の場合はループの基準をマッパー上で選択して行います。

手順は、XML ファイルの中でループの基準にしたい要素を[制御フィルタ]タブ内の[シンプルループフィルタ]で下図のように結線します。

下図の場合単価フィールドがループの基準要素となります。



XML ファイル転送におけるのループ処理

2.6.16 前後処理

前後処理では、マッパー定義を実行する前と、実行した後の処理を指定することができます。処理はコマンドで指定します。



前後処理

< 前処理 >



前処理

前処理では、データ転送を実行する前に“実行したい処理”を指定します。

- ・有効にする
チェックをすることで前処理が実行されます
- ・本製品サーバー上で行う
指定したコマンドを本製品サーバーを導入した OS 上で実行します。(Windows/Linux)
- ・入力元 OS/400 上で行う
入力元リソースに本製品がサポートする OS/400 を指定している場合に選択できます。
データベース名が表示されますので、必要なユーザーID/パスワードを入力します。
(入力リソースに OS/400 以外の OS や File を選択した場合は利用できません)
- ・コマンド
実行させる OS がサポートするコマンドラインを入力します
- ・コマンド戻り値

- ・ 障害発生時 コマンド実行の結果、入力した値が返された場合、前処理は障害発生と判断されます。それ以外の値が返された場合、前処理は成功と判断されます。
- ・ 成功時 コマンド実行の結果、入力した値が返された場合前処理は成功と判断されます。それ以外の値が返された場合、前処理は障害発生と判断されます。

コマンドの入力形式は、対象の OS によって異なります。戻り値が複数ある場合カンマで区切ることで複数指定できます。

以下に Widnows と Linux のコマンド実行例を記載します。

Windows 環境でディレクトリ作成コマンド「mkdir」を実行するコマンド

```
CMD /C mkdir "C:¥NewDirectory"
```

Linux 環境でディレクトリ作成コマンド「mkdir」を実行するコマンド

```
/bin/mkdir /NewDirectory
```



コマンド戻り値は、実行するコマンドによって異なります。事前に、どのような場合にこういった値が返されるのか把握しておく必要があります。

- ・ 障害発生時 前処理で障害が発生した場合の対処について指定します。
- ・ 停止 障害が発生した時点で処理を停止し次の処理(データ転送)は実行されません。
- ・ スキップ 障害を無視し、次の処理(データ転送)を実行します。

<後処理>

後処理

後処理では、データ転送を実行した後に“実行したい処理”を指定します。

- ・ 有効にする
 チェックをすることで後処理が実行されます
- ・ マッパー定義の実行に失敗した場合は、後処理を行わない(停止)

チェックをするとデータ転送でエラーになった場合後処理は実行されません。

- ・本製品サーバー上で行う

指定したコマンドを、本製品サーバーを導入した OS 上で実行します。(Windows/Linux)

- ・入力元 OS/400 上で行う

入力元リソースに本製品がサポートする OS/400 を指定している場合に選択できます。

データベース名が表示されますので、必要なユーザーID/パスワードを入力します。

(リソースに OS/400 以外の OS や File を選択した場合は利用できません)

- ・コマンド 実行させる OS がサポートするコマンドラインを入力します

- ・コマンド戻り値

- ・障害発生時 コマンド実行の結果、入力した値が返された場合、後処理で障害発生と判断されます。それ以外の値が返された場合、後処理は成功と判断されます。

- ・成功時 コマンド実行の結果入力した値が返された場合後処理は成功と判断されます。それ以外の値が返された場合、後処理は障害発生と判断されます。

コマンドの入力形式は、対象の OS によって異なります。戻り値が複数ある場合カンマで区切ることで複数指定できます。

以下に Windows と Linux のコマンド実行例を記載します。

Windows 環境でディレクトリ削除コマンド「rmdir」を実行するコマンド

```
CMD /C rmdir "C:¥NewDirectory"
```

Linux 環境でディレクトリ削除コマンド「rmdir」を実行するコマンド

```
/bin/rmdir /NewDirectory
```



コマンド戻り値は、実行するコマンドによって異なります。事前に、どのような場合にこういった値が返されるのか把握しておく必要があります。

2.6.17 フロースタイル

フロースタイルでは、マッパー定義を作成する際のフローの形状を指定することができます。デフォルトでは「直線」が選択されています。



フロースタイル

< 直線 >

フローの始点と終点の最短距離を直線で結線します。

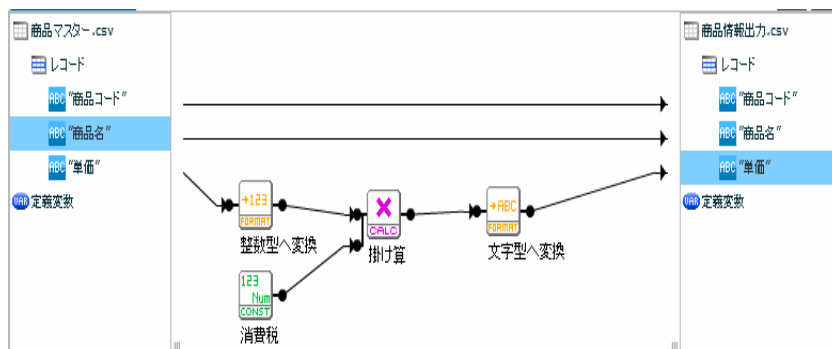
< カギ線 >

フローの形状がカギ型で表示されます。

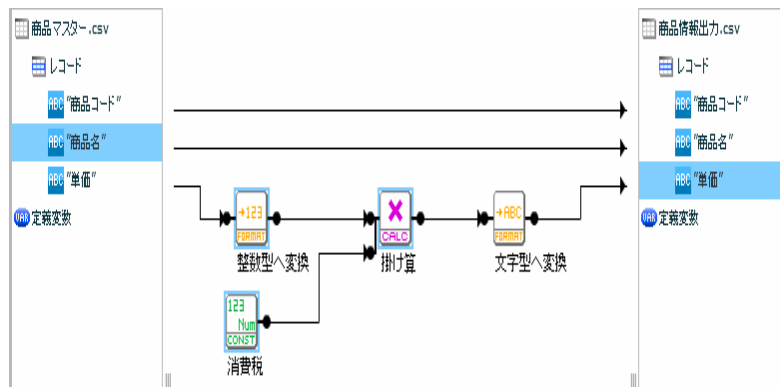
< 曲線 >

フローの形状が曲線で表示されます。

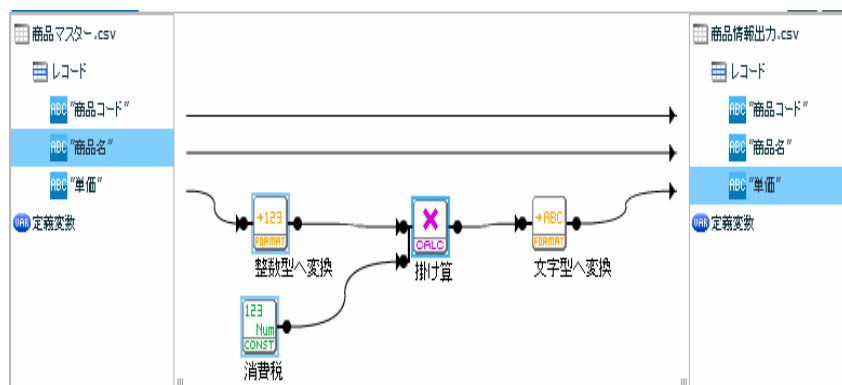
フロースタイルの変更は既に結線されているフローには適用されません。既存フローのスタイルを変更する場合は、フローを選択状態にした後、右クリックメニュー「線タイプ」をクリックして行います。



フロースタイル：直線

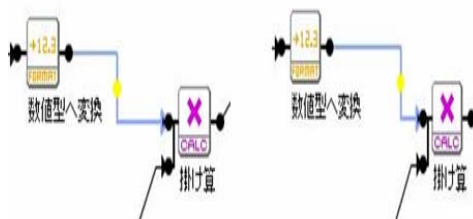


フロースタイル：カギ線



フロースタイル：曲線

フロースタイルが「カギ線」「曲線」に指定されている場合、フローを選択状態にするとフローの中点に黄色い点が表示されます。この点をドラッグすることで、フローの位置を調整することができます。

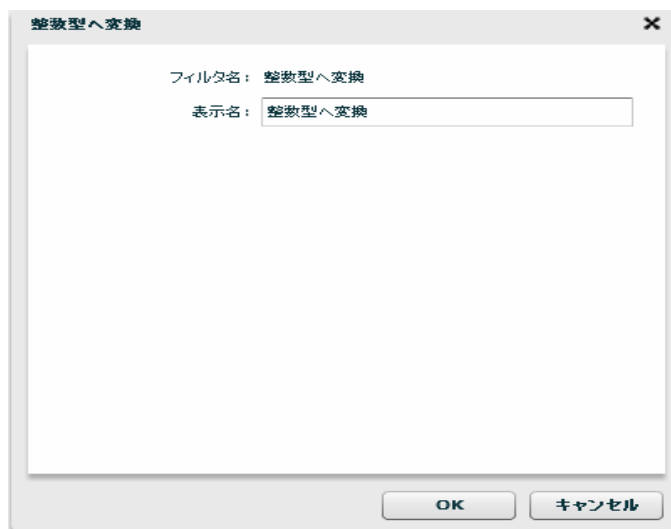


中点をドラッグすることでフローの位置を調整することが可能

2.6.18 フィルタのプロパティ

プロパティ画面は、フィルタの設定項目により異なります。

一番単純なフィルタのプロパティ画面は、表示名のみの指定です。



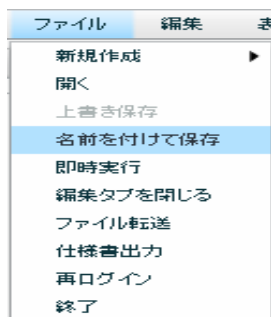
単純な フィルタ プロパティ画面



設定が必要なフィルタでは、キャンバスへフィルタをドラッグ&ドロップした瞬間に設定を入力するプロパティ画面が自動で表示されます。

2.6.19 定義保存

メニュー [ファイル] - [名前を付けて保存] より定義の保存ができます。



名前を付けて保存 メニュー



マップパー定義保存ダイアログ

- | | |
|------|-------------------|
| 名前 | 保存する定義の名前を入力します。 |
| 保存 | クリックで定義を保存します。 |
| コメント | コメントは 128 文字までです。 |

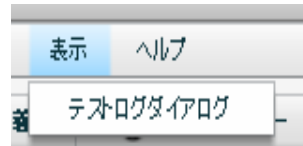


定義保管権限のないフォルダには保存できません。

2.6.20 テストログダイアログ - テスト実行

テストログダイアログとは、データ転送定義をテストする機能です。データ転送定義の作成ミスによるエラーなどを防ぐことができるため、運用開始前の実施が効果的です。

メニュー [表示] - [テストログダイアログ] より定義のテスト実行ができます。



テスト実行



テストログダイアログ

画面上部のテキストボックスにはテスト実行に使用するデータレコードを指定します。例えば1レコード目のデータでテストするにはテキストボックスに“1”と入力し[テスト実行]ボタンをクリックします。

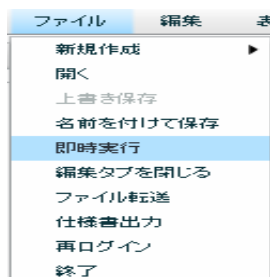
テストに使用するデータは入力元に設定したリソースから取得していますが、テスト実行の結果は出力先のリソースには影響を与えません。テスト実行結果は画面下部の [転送先] 欄にのみ表示されます。

< エラーが発生した場合 >

テスト実行結果がエラーの場合、[エラー] 欄にエラー情報が赤色で出力されます。画面下部の [転送元] [転送先] の各フィールドにはそのとき扱われた値が出力されます。データ転送定義やテストに利用したデータによっては値が出力されない場合があります。転送先のリソースインデックスは出力先のアダプターのインデックスになります。出力先には複数のアダプターが指定できるため、出力された値とフィールドが、どのアダプターに含まれているかが出力されています。

2.6.21 即時実行

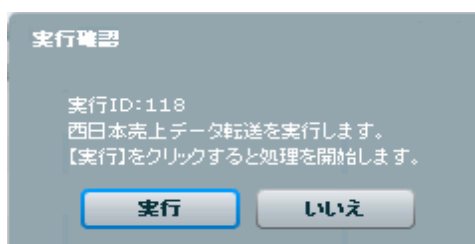
メニュー [ファイル] - [即時実行] より定義の実行ができます。



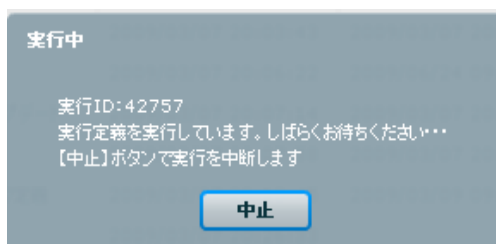
即時実行 メニュー



マッパー定義を保管してからでないと実行できません。
定義を変更した状態でも、保管するまで変更は実行に反映されません。

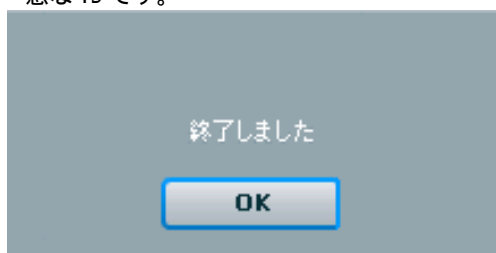


実行確認



実行中

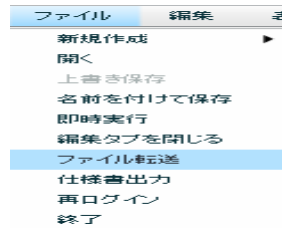
実行 ID : 実行時に振られる一意な ID です。



実行完了

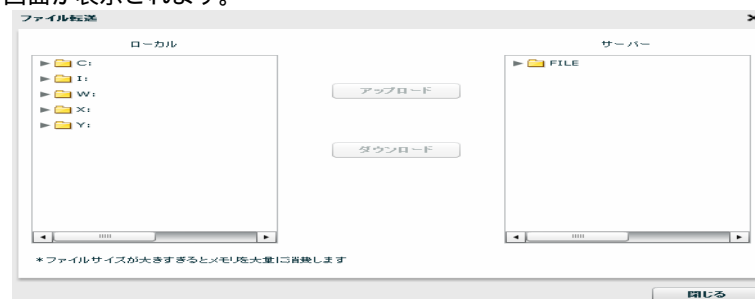
2.6.22 ファイル転送

メニュー[ファイル] - [ファイル転送]よりサーバーとのファイル転送(アップロード、ダウンロード)を行うことができます。



即時実行 メニュー

「ファイル転送」画面が表示されます。



「ローカル」には本製品が動作している環境のドライブが表示されます。

「サーバー」には本製品サーバーに登録されたファイルリソースが表示されます。

<サーバーへファイルのアップロードを行う>

ローカルファイルをサーバーへアップロードする場合は、ローカル側のファイルを選択状態にし、サーバー側のファイルフォルダー(2階層目)を選択状態にした後、[アップロード]ボタンをクリックすることで実行されます。

<ローカルへファイルのダウンロードを行う>

サーバーからローカルへダウンロードする場合は、ローカル側のフォルダを選択状態にし、サーバー側のファイル(3階層目)を選択状態にした後、[ダウンロード]ボタンをクリックすることで実行されます。

アップロード/ダウンロードの制限事項について

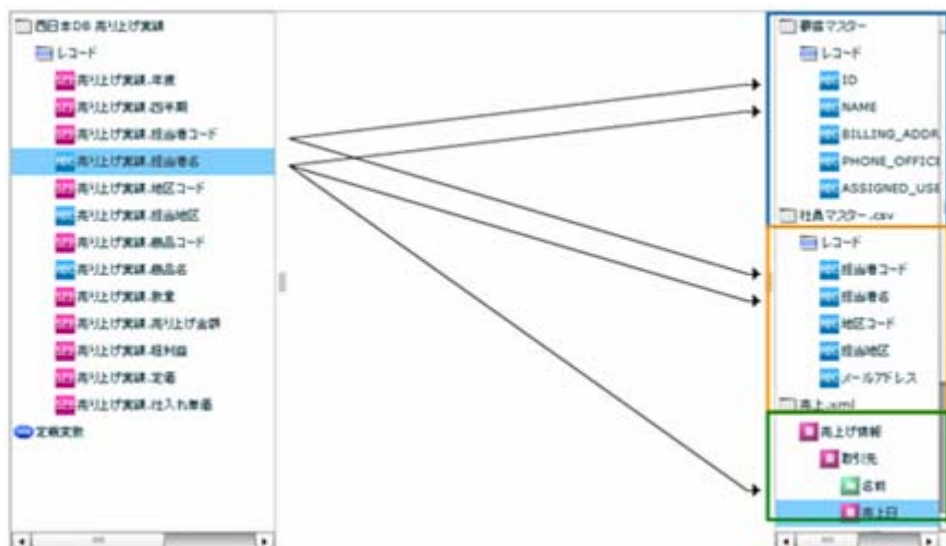


・ファイルのみアップロードおよびダウンロードが可能となっています。フォルダを対象とすることはできません。

・本機能を使用して、ファイル名の変更や削除、および新規ファイルの作成などは行うことができません。

2.6.23 複数出力

Datamartでは出力先を複数選択することができます。一回の定義実行で複数のリソースに出力できます。追加でリソースを選択することにより、マップ上の方方向にテーブルが追加されていきます。



定義変数 複数出力を行う際の定義変数は選択されたテーブルの下方方向に表示されます。

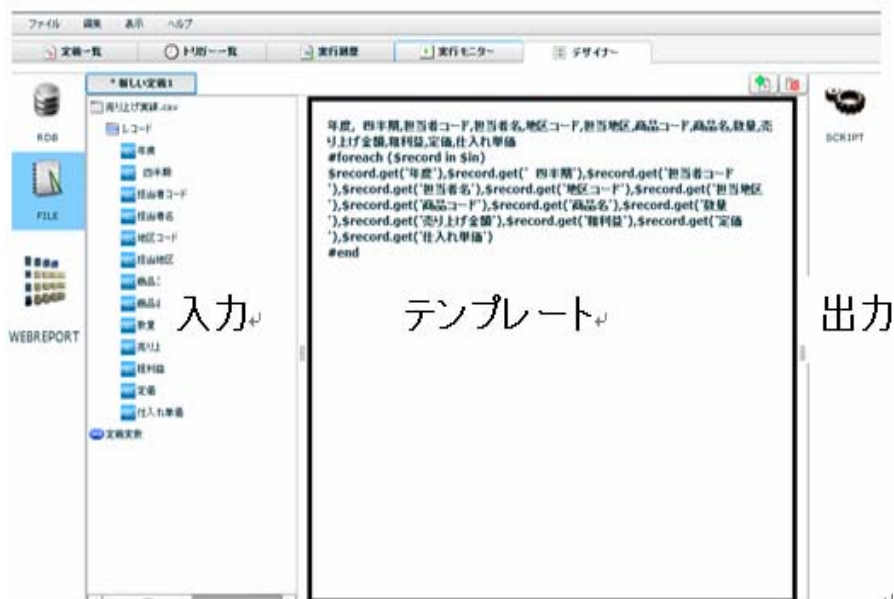
2.7 スクリプト定義

2.7.1 スクリプト定義とは

本製品で扱うことができるスクリプト定義とは、Verocityの規則に従いVTL (Velocity Template Language) で処理内容をかくことによって、マッパー定義ではできないようなファイル出力を可能にするものです。マッパー定義との大きな違いは、1レコードずつではなく、全レコードに対して処理を行う点です。VelocityとはJavaベースの「汎用テンプレートエンジン」です。

2.7.2 使い方

[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。



スクリプト定義新規作成画面

デザイナー画面の黒太枠内に転送処理内容を VTL (Velocity Template Language) で書き込みます。
 また、入力リソースを選択すると下図のように自動でデフォルト記述が書き込まれます。
 デフォルトで書き込まれる記述の内容は、入力リソースをそのままファイル転送するというものです。

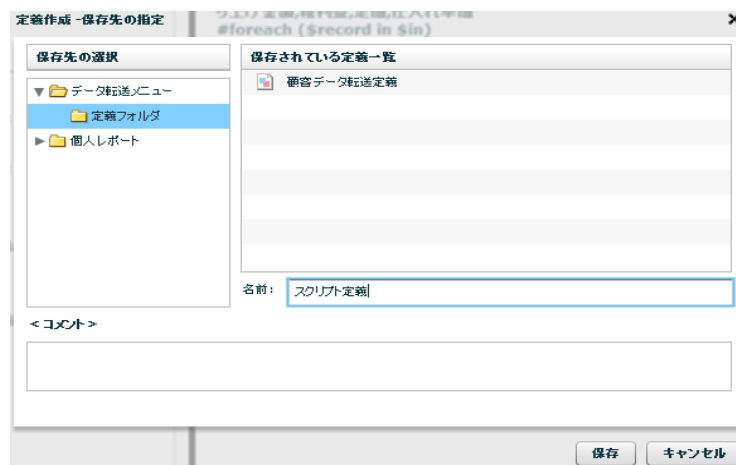
```

年度,数量,売上金額,粗利益
#foreach ($record in $in)
$record.get('年度'),$record.get('数量'),$record.get('売上金額'),$record.get('粗利益')
#end
  
```

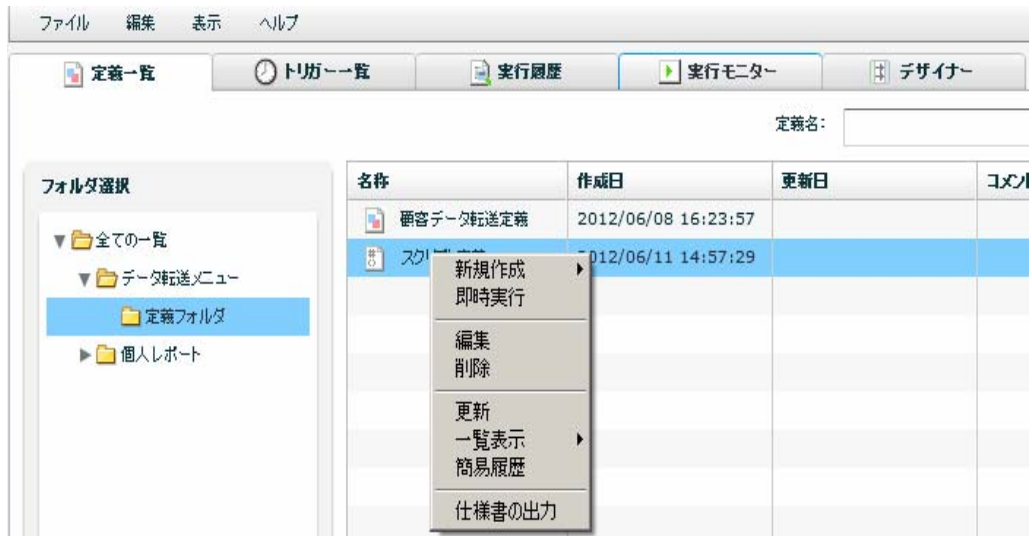
スクリプト定義のデフォルト記述

RDB、CSV ファイルを入力リソースに選択した場合は、**\$in** が入力リソースの全レコードにあたります。XML ファイルの場合は**\$root** になります。変数 \$cnt を数値型で扱う場合は \$cnt.parseInt(値) のように記述します。

出力ファイルを選択したら [ファイル] メニューで [名前を付けて保存] を押し、実行します。
 今回は定義名を「スクリプト定義例」にします。



保存名入力



スクリプト定義の実行

VTL でよく使われる構文と機能

構文	機 能
==コメント	1 行分のコメントです。出力結果には含まれません。
== コメント *#	複数行のコメントです。出力結果には含まれません。
\$変数名	変数の参照するオブジェクトの文字列表現を返します。 変数が null の場合、「\$変数名」という文字列が出力されます。
#foreach(\$変数 1 in \$変数 2) #end	繰り返し処理を行います。 参照 2 の保持する各値を順番に取り出し、参照 1 がそれを参照します。 この構文内では、\$velocityCount と記述することにより、現在の繰り返し処理の回数を表示します。(デフォルトでは、1 からカウントします。)

#if(条件式) [#else if(式)] [#else] #end	条件文 条件式には、以下の論理式が使用出来ます。 #if(\$hoge == \$foo) 等号演算子 #if(\$hoge > 28) #if(\$hoge >= 28) #if(\$hoge <= 28) #if(\$hoge = 28) 数値の等号 #if(\$hoge = "hoge") 文字列等号
#include(引 数 1,引 数 2, ...)	vm ファイル内から他のテキストファイルをインクルードします。複数指定の際は、カンマで区切ります。
#parse(引 数)	#include と基本的には同じですが、引数には指定できるテキストファイルは 1 つです。しかし、そのファイル内で記述された VTL も Velocity により解析されます。
#set(\$変数 = 値)	#set 指示子は、リファレンスの値をセットするために使われます。値はリファレンス変数かプロパティリファレンスに割り当てられることができます。左辺の変数に右辺の値がセットされます。
\$変数.get(文字列)	変数名(文字列)の値を取得します。

2.8 定義変数

2.8.1 定義変数とは

定義変数とは1つのマッパー定義・タスク定義内でもつ変数です。定義に対してユーザーが自由に設定することができ、

- ・外部から渡される値を格納する。
- ・値を変換する。
- ・その変数を内部で使う。

ことができます。

また、定義変数にはマッパー、スクリプトで定義されるマッパー変数と、タスクで定義されるタスク変数があります。

【タスク定義】 【マッパー定義、スクリプト定義】間では定義実行の前後で値を受け渡すことができます。

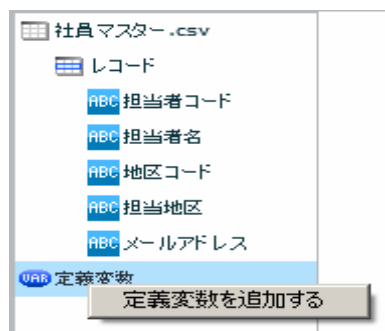
マッパー変数を転送定義内で変換してタスク変数に渡し、そのタスク変数を次に実行されるマッパーのマッパー変数に渡すなど動的な利用が可能になります。

2.8.2 使い方

- ・定義変数の設定[マッパー定義]

デザイナー画面で入出力リソースの定義変数を右クリックし、定義変数を追加するを選択します。

入出力リソースどちらかに定義変数を追加するともう1つにも自動で追加されます。



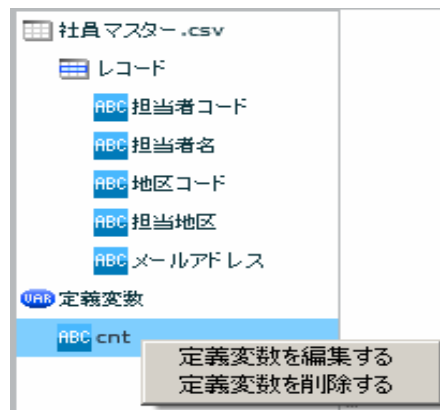
定義変数の追加

定義変数の名前とそのデフォルト値を入力し、[OK]を押します。

今回は変数名 cnt、デフォルト値 1 と設定します。

定義変数設定

追加された定義変数フィールドを右クリックすると以下の画面が表示されます。



定義変数を右クリック時のメニュー

[定義変数を編集する]

定義変数の名前、デフォルト値を変更することができます。

[定義変数を削除する]

定義変数を削除します。

・定義内で定義変数を使用する場合

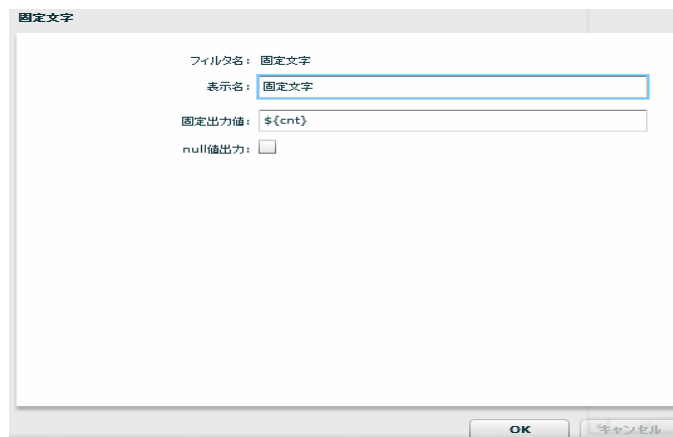
ユーザーが入力できるフィールドに対して、 $\{\text{変数名}\}$ と入力することで変数の値を使用することができます。

定義変数は以下の項目で使用できます。

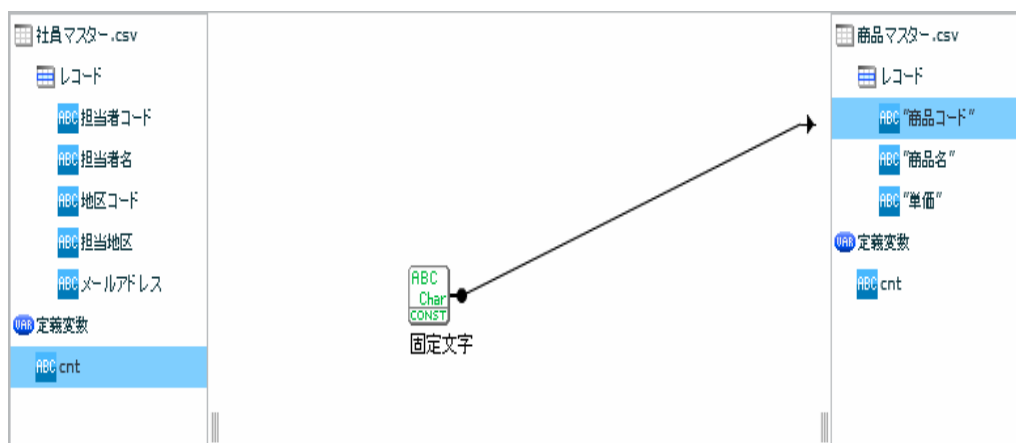
- ・フィルタの固定出力値
- ・入出力ファイルリソースのリソース名、フォルダ名、ファイル名
- ・検索条件の値
- ・スクリプトのテンプレート内

フィルタの固定出力値

マッパー定義に設定した定義変数を固定文字フィルタを使って出力します。



固定文字フィルタのプロパティ

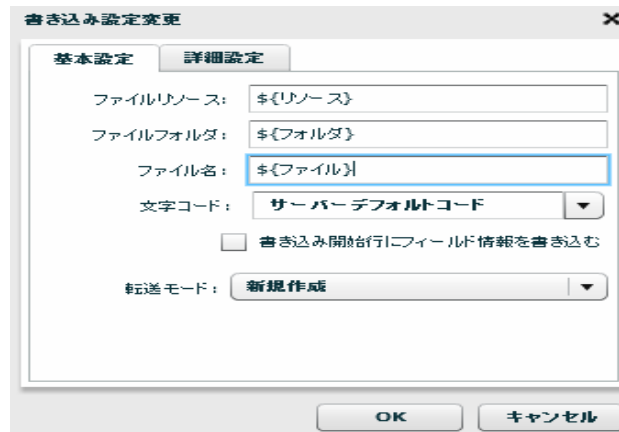


定義変数使用例

固定文字フィルタをマッピングし、その値を`${cnt}`と設定します。

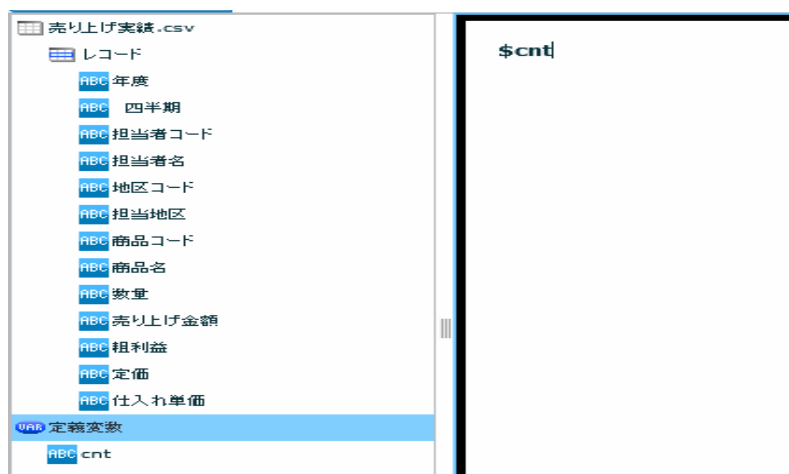
この固定文字フィルタの固定出力値は定義変数 `cnt` の値である `1` という文字になります。

出力リソースのファイル名



出力リソース、フォルダ、ファイル名

スクリプトのテンプレート内



スクリプト

スクリプトでは定義変数に英字の小文字で指定してください。

- ・変数を使えない場面

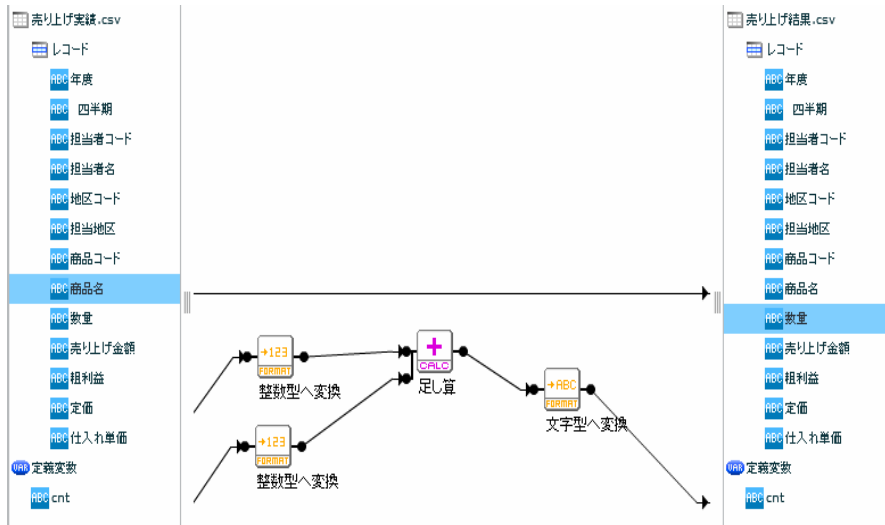
入出力 RDB テーブル名、フィルタのコメント、フィルタの表示名など

- ・定義変数の設定[タスク定義]

タスク変数とマッパー変数の受け渡しで動的な転送定義作成例をみていきます。

以下2つのマッパー定義をタスク定義に追加します。

マッパー定義1



マッパー定義1

入力 CSV ファイルデータ :

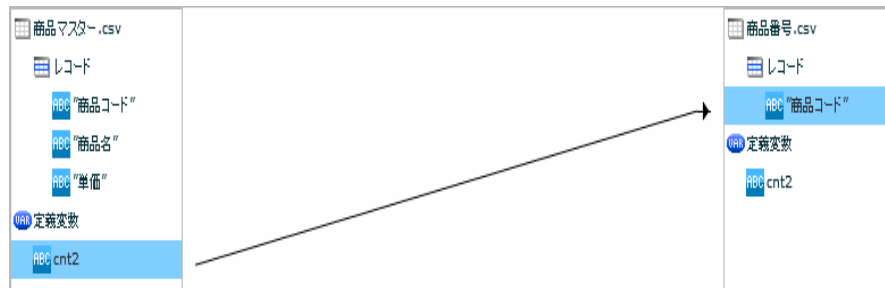
num 1~10

str one~ten

定義変数

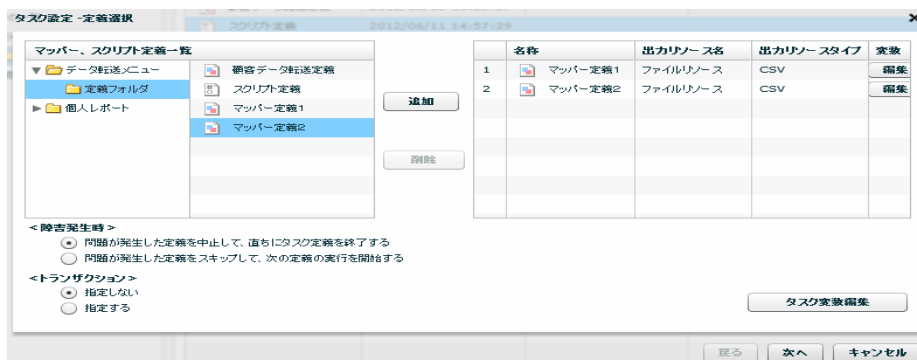
Cnt 1

マッパー定義2



マッパー定義2

タスク定義を作成します。



タスク設定

タスク定義作成の手順は「2.2.6 タスク定義設計画面」を参照してください。
タスク設定画面の[タスク変数編集]をクリックします。



タスク変数編集

[追加] タスク変数を追加します。変数名と値を入力し[OK]をクリックします。
今回は以下のように設定します。

変数名 : task 値 : 設定せず



タスク変数追加

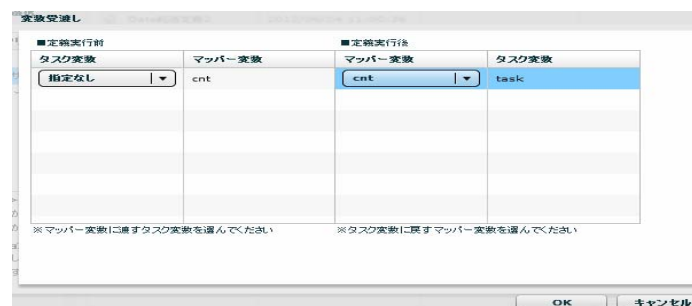
[削除] タスク変数を削除します。

・タスク定義 マッパー定義間での定義変数の受け渡し

タスク設定画面で追加したマッパー定義の変数欄の[編集]をクリックします。

プルダウンから

1. 定義実行前に定義変数に渡したいタスク変数を指定します。
2. 定義実行後にタスク変数に戻したいマッパー変数を指定します。



変数受け渡し画面[定義変数]

■ 定義実行前		■ 定義実行後	
タスク変数	マップパー変数	マップパー変数	タスク変数
task	cnt2	指定なし	task

※マップパー変数に連ずるタスク変数を選んでください

※タスク変数に戻すマップパー変数を選んでください

OK キャンセル

変数受け渡し画面[定義変数 2]

今回は上図のように設定します。

マップパー定義「定義変数」の cnt の値を実行後に task にセット。

マップパー定義「定義変数 2」で実行前に task から cnt2 にセット。

3. [OK]をクリックします。

変数受け渡し設定は以上です。

タスク定義を保存して実行すると実行前後に指定した変数の受け渡しが行われます。

今回の場合、タスク定義を実行するとまずマップパー定義「定義変数」が実行され、その定義変数である cnt の値が実行後にタスク変数に渡されます。

cnt の初期値は 1 で、num の値が全て足しこまれる定義になっているので実行後タスク変数に渡される cnt の値は $1 + 1 + 2 + \dots + 10 = 56$ になります。

このタスク変数 56 という値が次に実行されるマップパー定義[定義変数 2]に実行前に渡されるので、[定義変数 2]のフィールドへの出力値は 56 となります。

2.9 QanatExecute

2.9.1 QanatExecute とは

QanatExecute とは外部から本製品の定義実行命令を送るためのツールです。

本製品以外のマシンに QanatExecute をインストールすることで、そのマシンから本製品の転送定義を実行させることができます。これにより、外部の別アプリケーションから本製品の転送定義を実行させることが可能になります。

インストールした QanatExecute フォルダ内には転送定義を実行するための QanatExecute.bat ファイルが存在します (Linux の場合は QanatExecute.sh を使用します)。このバッチファイルを編集することで簡単に QanatExecute の機能を利用することが可能です。

2.9.2 使い方

QanatExecute をインストールし、そのフォルダ内のバッチファイルをテキストで編集します。

```
@rem =====↓
@rem 引数説明↓
@rem =====↓
@rem HOST   ホスト名 or IP↓
@rem PORT   ポート番号↓
@rem USER   ユーザーID↓
@rem PASSWORD パスワード↓
@rem MENU   メニュー名↓
@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=↓
SET PORT=↓
SET USER=↓
SET PASSWORD=↓
SET MENU=↓
SET FOLDER=↓
SET DEF_NAME=↓
SET DEF_TYPE=↓
SET VARIABLE=↓
↓
"%INST_PATH%\jre\bin\java" -jar "%INST_PATH%\QanatExecute.jar" %HOST% %PORT% %USER% %PASSWORD% %MENU% %FOLDER% %DEF_NAME% %DEF_TYPE% %VARIABLE%↓
↓
PAUSE↓
```

QanatExecute.bat

実行する転送定義の情報を編集します。

```

SET HOST      ホスト名 or IP
SET PORT      ポート番号
SET USER      ユーザーID
SET PASSWORD  パスワード
SET MENU      メニュー名
SET FOLDER    フォルダ名
SET DEF_NAME  定義名
SET DEF_TYPE  qanmapper: マッパー定義 qantask: タスク定義 qanscript: スクリプト定義
SET VARIABLE  定義変数

```

```

@rem =====
@rem 引数説明
@rem =====
@rem HOST      ホスト名 or IP
@rem PORT      ポート番号
@rem USER      ユーザーID
@rem PASSWORD  パスワード
@rem MENU      メニュー名
@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=DBからCSV
SET DEF_TYPE=qanmapper
SET VARIABLE=hensu1=1 hensu2=2

```

実際の代入例

定義変数のセットは SET VARIABLE=変数名 1=値 1 変数名 2=値 2・・・といった形式で入力します。

編集したバッチファイルを実行すると、編集した情報で転送定義が実行されます。



Linux で使用する「QanatExecute.sh」は、導入後『INST_PATH=』の値を手動で設定する必要があります。

付録 A

付録 A フィルタリファレンス

A フィルタリファレンス

ここでは、データ転送定義の作成時に使用できるフィルタを記載します。



DBMS、データ型により制限値は異なりますが、フィルタを利用した処理を行う場合は、有効桁数は 15 桁になりますのでご注意ください。



注意

転送元のデータが Null 値の場合、多くのフィルタの出力結果は、Null 値になります。Null 値を含む可能性があるデータの処理にご注意ください。また、フィルタの処理にてエラーが発生した場合、後続のフィルタには、Error 値という特別な値が渡ります。この Error 値は Null と同様、多くのフィルタの出力値は Error 値となります。

どちらの場合も、IF フィルタを利用して、Null 値か？ Error 値かの判断が可能です。判断が必要な場合には、IF フィルタをお使いください。

A-1 型変換フィルタ

データ型の変換処理を行うフィルタ群です。



整数値へ変換

入力：文字値 または 数値

プロパティ：

表示名

出力：数値型

説明：入力値(文字/数値)を整数型に変換します。出力は数値型です。

例：文字の社員コードが整数転送されます

	入力	出力
レコード 1	19876	19876
レコード 2	19877	19877



数値型へ変換

入力：文字値 または 数値

プロパティ：

表示名

出力：数値型

説明：入力値(数値/文字)を数値(小数有り)に変換します。

例：文字の社員コードが数値転送されます

	入力	出力
レコード 1	19876	19876
レコード 2	19877	19877



文字型へ変換

入力：数値 または 文字値 または日付、時刻、日付時刻値

プロパティ：

表示名

出力：文字型

説明：入力値を(数値/文字/日付)を文字列へ変換します。

例：時刻の値が文字に変換されます。

	入力データ型	入力	出力
レコード 1	数値型(整数)	50150	50150
レコード 2	数値型(少数桁有)	50150.152	50150.152
レコード 3	日付型	2009/03/24	20090324
レコード 4	時刻型	09:37:45	093745
レコード 5	日付時刻型	2009/03/24 13:15:27.123	20090324131527123



出力の形式を指定して文字型に変換したい場合は、数値フォーマット、日付フォーマットを利用してください。



日付型へ変換

入力：数値 または 文字値 または日付値 または日付時刻値

プロパティ：

表示名

出力：日付型

説明：入力値(文字/数値/日付/日付時刻)を日付型へ変換します。

例：日付時刻文字の値が日付に変換されます。

	入力	出力
レコード 1	20010807	2001/08/07
レコード 2	20010721102530	2001/07/21



入力が数値、文字値の場合は最左桁から、YYYYMMDD(4桁の年 2桁の月 2桁の日)として処理します。9桁目以降は無視されます。

桁数が8桁に満たない場合は、エラーになります。



日付の存在チェックは行いません。極力日付型への変換を試みます。
例) 20090229(数値 または 文字) 20090301 として出力



Oracle の旧 Date 型(時刻情報を含むデータ型)への出力は、日付時刻型へ変換をご利用ください。



時刻型へ変換

入力：数値 または 文字値 または日付値 または日付時刻値

プロパティ：

表示名

出力：時刻型

説明：入力値(文字/数値/時刻/日付時刻)を時刻型へ変換します。

例：日付時刻文字の値が時刻に変換されます。

	入力	出力
レコード 1	134521	13:45:21
レコード 2	102530	10:25:30



入力が数値、文字値の場合は最左桁から、HHMMSS

(時刻 2 桁の分 2 桁の秒)として処理します。7 桁目以降は無視されます。
桁数が 6 桁に満たない場合は、エラーになります。



時刻の存在チェックは行いません。極力時刻型への変換を試みます。
例) 251080(数値 または 文字) 011120 として出力



日付時刻型へ変換

入力：数値 または 文字値 または日付値、時刻値、日付時刻値

プロパティ：

表示名：

出力：日付時刻型

説明：入力値(文字/数値/日付時刻)を日付時刻型へ変換します。

例：日付時刻文字の値が日付時刻に変換されます。

	入力	出力
レコード 1	20010807134521	2001/08/07 13:45:21
レコード 2	20010721102530	2001/07/21 10:25:30



入力が数値、文字値の場合は最左桁から、YYYYMMDDHHMMSS

(4 桁の年 2 桁の月 2 桁の日 2 桁の時刻 2 桁の分 2 桁の秒)として処理します。
15 桁目以降は無視されます。桁数が 14 桁に満たない場合は、エラーになります。

入力が時刻型の場合、日付は 1970/01/01 になります。



入力が日付値の場合、時刻は 00:00:00 になります。

任意の日付/時刻を指定したい場合は、文字列連結で文字列を作成して入力値としてください。



年

入力：日付値、または日付時刻値

プロパティ：

表示名：

出力：数値型

説明：入力値(日付/日付時刻)から年を抜き出し整数値に変換します。

例：日付データから年を取り出して数値型データに変換します

	入力	出力
レコード 1	2001/07/23	2001
レコード 2	2001/07/11	2001



月

入力：日付値、または日付時刻値

プロパティ：

表示名：

出力：数値型

説明：入力値(日付/日付時刻)から月を抜き出し整数値に変換します。

例：日付データから月を取り出して数値型データに変換します

	入力	出力
レコード 1	2001/07/23	7
レコード 2	2001/07/11	7



日

入力：日付値、または日付時刻値

プロパティ：

表示名

出力：数値型

説明：入力値(日付/日付時刻)から日を抜き出し整数値に変換します。

例：日付データから日を取り出して数値型データに変換します。

	入力	出力
レコード 1	2001/07/23	23
レコード 2	2001/07/11	11



時刻の時

入力：時刻項目、または日付時刻値

プロパティ：

表示名

出力：数値型

説明：入力値(時刻/日付時刻)から時を抜き出し整数値に変換します。

例：時刻データから時を取り出して数値型データに変換します。

	入力	出力
レコード 1	13:15:27	13
レコード 2	09:37:45	9



時刻の分

入力：時刻項目、または日付時刻値

プロパティ：

表示名

出力：数値型

説明：入力値(時刻/日付時刻)から分を抜き出し整数値に変換します。

例：時刻データから分を取り出して数値型データに変換します。

	入力	出力
レコード 1	13:15:27	15
レコード 2	09:37:45	37



時刻の秒

入力：時刻項目、または日付時刻値

プロパティ：

表示名

出力：数値型

説明：引数 1 の時刻の秒の部分を数値で返します。

例：時刻データから秒を取り出して数値型データに変換します。

	入力	出力
レコード 1	13:15:27	27
レコード 2	09:37:45	45



数値フォーマット

入力：数値

プロパティ：

表示名

フォーマット文字列 リストよりフォーマット選択

フォーマット文字列 入力フィールド

出力：文字型

説明： 入力値(数値)をフォーマットに従い文字値に変換します。

例： 数値の売上金額を「#,###」というフォーマットで文字列の売上金額データに変換します。

	入力	出力
コード 1	10500500	10,500,500
コード 2	5980	5,980

フォーマット文字列入力フィールドについて

フォーマットに指定可能な文字は、Java の DecimalFormat 入力規則に従います。

DecimalFormat の詳細情報は以下を参照ください。

<http://java.sun.com/j2se/1.5.0/ja/docs/ja/api/java/text/DecimalFormat.html>



文字フォーマット

入力：文字型

プロパティ：

表示名

フォーマット文字列 リストよりフォーマット選択

フォーマット文字列 入力フィールド

出力：数値型

説明： 指定したフォーマットで入力された入力値(文字型)を数値に変換します。

例： 文字型の売上金額データを「¥#,###;-¥#,###」というフォーマットで数値型の売上金額に変換します。

	入力	出力
コード 1	¥12,200	12200
コード 2	-¥1,234	1234

フォーマット文字列入力フィールドについて

フォーマットに指定可能な文字は、Java の DecimalFormat 入力規則に従います。

DecimalFormat の詳細情報は以下を参照ください。

<http://java.sun.com/j2se/1.5.0/ja/docs/ja/api/java/text/DecimalFormat.html>



日付フォーマット

入力：日付値、時刻項目、または日付時刻値

プロパティ：

表示名

フォーマット文字列 リストよりフォーマット選択

フォーマット文字列 入力フィールド

出力：文字型

説明：入力値(日付/日付時刻)をフォーマットに従い文字値に変換します。

例：数値の売上日から「yyyy/MM/dd」というフォーマットで文字列の売上日データに変換します。

	入力	出力
レコード 1	20090130	2009/01/30
レコード 2	20090201	2009/01/30

フォーマット文字列入力フィールドについて



フォーマットに指定可能な文字は、Java の SimpleDateFormat 入力規則に従います。
SimpleDateFormat の詳細情報は以下を参照ください。

<http://java.sun.com/j2se/1.5.0/ja/docs/ja/api/java/text/SimpleDateFormat.html>



変換後に出力可能な最小単位はミリ秒です

A-2 文字変換フィルタ

データの加工処理を行うフィルタ群です。



文字列連結

入力：任意の個数の文字値（最大 20 個）

プロパティ:

表示名

フローの数 連結する値の数（最大 20 個）

区切り文字 選択リストより選択

出力：文字型

説明：2 つ以上の入力値(文字)の文字列をつないで 1 つの文字列にします。

例： 商品コード`と商品名を連結し、商品コード`名フィールド`に変換します。その区切り文字は`/`です。

	入力 1	入力 2	出力
コード` 1	S10010	スムーズ コンパス	S10010/スムーズ コンパス
コード` 2	S10011	三角定規	S10011/三角定規



入力値が null 値の場合、長さ 0 の文字列（空白値）として処理します。



文字列置換

入力：文字値

プロパティ：

表示名

置換前 置換したい文字列

置換後 置換後の文字列

正規表現を使用する

出力：文字型

説明： 入力値(文字)の文字列の中で、2番目に指定した文字列を3番目に指定した文字列で置き換えます。正規表現を使用するにチェックを入れると2番目の文字列を正規表現で指定できません。

例： [顧客名]に登録されている文字のうち‘株’を検索し、‘株式会社’に変換します。

	入力	出力
レコード 1	(有)仏法僧	(有)仏法僧
レコード 2	シノリ株	シノリ株式会社



部分文字列

入力：文字値

プロパティ：

表示名

開始位置 開始する文字位置(1~)

文字数 抽出する文字数

値を指定しない場合、指定した文字位置から後ろ全てを取り出します。

出力：文字型

説明： 入力値(文字)の文字列から指定した位置から指定した文字数を抜き出します。

例： [組織名]に登録されている文字のうち1文字目から6文字分を抜き出します。

	入力	出力
レコード 1	営業部営業一課	営業部営業一
レコード 2	営業部営業三課	営業部営業三



開始に指定した位置より入力文字値が短い場合は長さ0の文字列を出力します。



文字列分割

入力：文字値

プロパティ：

表示名

区切り文字 区切り文字

指定位置 位置指定（分割された何番目を取得するか？1～）

出力：文字型

説明： 入力値(文字)の文字列を区切り文字で分割し、指定位置の文字列にします。

例： 電話番号の局番(2番目)を取り出します。その区切りは'-'です。

	入力	出力
レコード 1	03-5714-5136	5714
レコード 2	0584-83-7567	83



指定した位置の文字列が存在しない場合は長さ 0 の文字列を出力します。



全角変換

入力：文字値

プロパティ：

表示名

出力：文字型

説明： 入力値(文字)の半角データを全角へ変換します。

例： 半角の社員名データが全角に変換されます。

	入力	出力
レコード 1	ヤマダ タロウ	ヤマダ タロウ
レコード 2	タカシ ハナコ	タカナシ ハナコ



半角変換

入力：文字値

プロパティ：

表示名

出力：文字型

説明： 入力値(文字)の全角データが半角変換可能ならば、半角文字へ変換します。

例： 全角の商品名カナが半角に変換されます。

	入力	出力
コード 1	スムーズ コンパス	スムーズ コンパス
コード 2	サンカクジョウギ	サンカクジョウギ



大文字変換

入力：文字値

プロパティ：

表示名

出力：文字型

説明： 入力値(文字)の文字列中の英小文字を英大文字に変換します。英小文字以外の文字はそのまま転送されます。全角/半角共に変換されます。

例： 商品名の英小文字を英大文字に変換します。

	入力	出力
コード 1	New Work Friend	NEW WORK FRIEND
コード 2	ポット Type-S	ポット TYPE-S



小文字変換

入力：文字値

プロパティ：

表示名

出力：文字型

説明： 入力値(文字)の文字列中の英大文字を英小文字に変換します。英大文字以外の文字はそのまま転送されます。全角/半角共に変換されます。

例： 商品名の英大文字を英小文字に変換します。

	入力	出力
コード 1	New Work Friend	new work frined
コード 2	ポット Type-S	ポット type-s



数字以外の文字除去

入力：文字値

プロパティ：

表示名

出力：文字型

説明： 入力値(文字)の文字列の内、半角および全角の数字(0~9)以外の文字を取り除きます。

例： 顧客コード内の数字以外の文字が除去されます。

	入力	出力
レコード 1	C100000001	100000001
レコード 2	C100000002	100000003



空白文字の除去

入力：文字値

プロパティ：

表示名

出力：文字型

説明： 入力値(文字)の文字列の内、空白(半角および全角)の文字を取り除きます。

例： 商品名フィールドから 空白文字を取り除きます。

	入力	出力
レコード 1	スムーズ コンパス	スムーズコンパス
レコード 2	三角定規	三角定規



非文字コード除去

入力：文字値

プロパティ：

表示名

出力：文字型

説明： 入力値(文字)の文字列中の文字以外のコード(TAB、復帰、改行など)を取り除きます。

例： コメントから制御コードを取り除きます。

	入力	出力
レコード 1	New Work Friendの 新バージョンです。	New Work Friendの新バージョンです。
レコード 2	ポット Type-S	ポット Type-S



除去可能な非文字コードは以下の制御コードです。
 コード値 00~1F の C0 集合の制御コードと呼ばれるコードです。
 (ヌル、テキスト終了、改行、改ページ、復帰、シフトアウト、シフトインなど)



文字列長

入力：文字値

プロパティ：

表示名

出力：数値型

説明：文字列フィールドに入っているデータの文字数を返します。

例：住所フィールドに登録されているデータを文字数で返します。

	入力	出力
レコード 1	神奈川県横浜市守屋町3-9	13
レコード 2	東京都大田区蒲田5-37-1	14

A-3 数値処理フィルタ

数値の演算処理を行うフィルタ群です。



足し算

入力 1：数値

入力 2：数値

プロパティ：

表示名

出力：数値型

説明：2つの入力値（数値）を足し算します。

例：商品合計金額と送料を足し算した金額を出力します。

	入力		出力
レコード 1	8500	735	9235
レコード 2	3000	1050	4050



出力可能な数値の有効桁数は 15 桁です。



引き算

入力 1：数値

入力 2：数値

プロパティ：

表示名

出力：数値型

説明：2つの入力値（数値）を引き算します。

例：税込み合計金額から税相当分を引き算した金額を出力します。

	入力		出力
レコード 1	1050	50	1000
レコード 2	50250	250	50000



出力可能な数値の有効桁数は 15 桁です。



掛け算

入力 1：数値

入力 2：数値

プロパティ：

表示名

出力：数値型

説明：2つの入力値（数値）を掛け算します。

例：単価フィールドと数量フィールドを掛けて金額を出力します。

	入力		出力
レコード 1	100	10	1000
レコード 2	500	50	25000



出力可能な数値の有効桁数は 15 桁です。



割り算

入力 1：数値

入力 2：数値

プロパティ：

表示名

出力：数値型

説明：2つの入力値（数値）を割り算します。

例：合計フィールドを数量フィールドで割って単価を出力します。

	入力		出力
レコード 1	10000	10	1000
レコード 2	50000	25	2000



出力可能な数値の有効桁数は 15 桁です。



商余

入力 1：数値

入力 2：数値 割る値

プロパティ：

表示名

出力：数値型

説明：入力値(数値型)の値を指定した値で割った時の余りを求めます。

例：売上金額を 10000 で割った売上金額剰余を出力します。

	入力	出力
コード 1	36,012	6012
コード 2	40,000	0



四捨五入

入力：数値

プロパティ：

表示名

桁指定 四捨五入する桁(数値)

小数点以下はプラスの数値で、整数桁はマイナスの数値で指定できます。

出力：数値型

説明：入力値(数値型)の値を指定した桁で四捨五入します。

例：粗利率を四捨五入して 2 桁目までの結果を出力します。

	入力	出力
コード 1	41.457	41.46
コード 2	178.812	178.81



出力可能な数値の有効桁数は 15 桁です。



切り上げ

入力： 数値

プロパティ：

表示名

桁指定 切り上げ指定桁（数値）

出力： 数値型

説明： 入力値(数値型)の値を指定した桁で切り上げます。

例： 粗利率を整数の最終桁を切り上げた結果を出力します。パラメータは-1

	入力	出力
レコード 1	41.457	50
レコード 2	178.812	180



出力可能な数値の有効桁数は 15 桁です。



切り捨て

入力： 数値

プロパティ：

表示名

桁指定 切り捨て指定桁（数値）

出力： 数値型

説明： 入力値(数値型)の値を指定した桁で切り捨てます。指定桁は省略できますが、0 桁で切り捨てられます。

例： 粗利率を切り捨てた結果を出力します。パラメータは省略 または 0

	入力	出力
レコード 1	41.457	41
レコード 2	178.812	178



出力可能な数値の有効桁数は 15 桁です。

A-4 日付時刻処理フィルタ

日付/時刻/日付時刻の作成・演算・曜日取得などを行うフィルタ群です。



日付作成

入力：数値

プロパティ：

表示名

出力：日付型

説明：数値型で入力された年・月・日を文字連結して日付型に変換します。

例：1984,10,25 という年月日 3 つの入力値を結合して日付型で出力します。

	入力1	入力2	入力3	出力
レコード 1	1984	10	25	1984-10-25



時刻作成

入力：数値

プロパティ：

表示名

出力：時刻型

説明：数値型で入力された時・分・秒を文字連結して時刻型に変換します。

例：12,10,30 という時分秒 3 つの入力値を結合して時刻型で出力します。

	入力1	入力2	入力3	出力
レコード 1	12	10	30	12:10:30



日付時刻作成

入力 1 : 日付

入力 2 : 時刻

プロパティ:

表示名

出力 : 日付時刻型

説明 : 日付と時刻を文字連結して日付時刻型に変換します。

例 : 日付 1984-10-25 と時刻 12:10:30 を結合して日付時刻型で出力します。

	入力1	入力2	出力
コード 1	1984-10-25	12:10:30	1984-10-25 12:10:30



曜日取得

入力 : 日付、日付時刻

プロパティ:

表示名

出力タイプ : 数字 or 日本語 or 英語

出力 : 文字型

説明 : 入力された日付・日付時刻からその曜日を取得し、選択された出力タイプで出力します。

例 : ある日付・日付時刻の曜日を出力します。

	入力	出力
コード 1	2009-11-24	火曜日
コード 2	2009-12-25 7:30:25	Fri



日付差分

入力 1 : 日付

入力 2 : 日付、日付時刻

プロパティ:

表示名

出力タイプ : 年 or 月 or 日

出力 : 数値型

説明 : 入力 1 (日付) と入力 2 (日付 or 日付時刻) の差分を選択された出力タイプで出力します。

例 : 2つの日付の差分を出力します。

出力タイプ-レコード 1:日 レコード 2 : 月

	入力		出力
レコード 1	2009-11-20	2009-11-24	4
レコード 2	2009-10-25	2009-12-25 7:30:25	2



時刻差分

入力 1 : 時刻

入力 2 : 時刻、日付時刻

プロパティ:

表示名

出力タイプ : 時 or 分 or 秒

出力 : 数値型

説明 : 入力 1 (時刻) と入力 2 (時刻 or 日付時刻) の差分を選択された出力タイプで出力します。

例 : 2つの時刻の差分を出力します。

出力タイプ-レコード 1:秒 レコード 2 : 分

	入力		出力
レコード 1	11:22:33	11:24:41	128
レコード 2	6:34:14	2009-12-25 7:30:25	56



日付時刻差分

入力 1：日付時刻

入力 2：日付、時刻、日付時刻

プロパティ：

表示名

出力タイプ：年 or 月 or 日 or 時 or 分 or 秒

出力：数値型

説明：入力 1（日付時刻）と入力 2（日付 or 時刻 or 日付時刻）の差分を選択された出力タイプで出力します。

例： 2つの日付時刻の差分を出力します。

出力タイプ-レコード 1:月 レコード 2：年

	入力		出力	
レコード 1	2009-12-25 7:30:25	2008-11-14		13
レコード 2	1984-10-25 6:34:14	2009-12-25 7:30:25		25



日付演算

入力 1：日付

入力 2：数値

プロパティ：

表示名

演算項目：年 or 月 or 日

出力：日付型

説明：入力 1（日付）に入力 2（数値）を加算した日付を出力します。

例： 日付の演算結果を出力します。

演算項目-レコード 1:日 レコード 2：年

	入力		出力	
レコード 1	2009-12-25	2		2009-12-27
レコード 2	1984-10-25	-3		1981-10-25



時刻演算

入力 1 : 時刻

入力 2 : 数値

プロパティ:

表示名

演算項目 : 時 or 分 or 秒

出力 : 時刻型

説明 : 入力 1 (時刻) に入力 2 (数値) を加算した時刻を出力します。

例 : 時刻の演算結果を出力します。

演算項目-レコード 1:分 レコード 2 : 時

	入力		出力
レコード 1	12:11:30	2	12:13:30
レコード 2	3:10:40	10	13:10:40



日付時刻演算

入力 1 : 日付時刻

入力 2 : 数値

プロパティ:

表示名

演算項目 : 年 or 月 or 日 or 時 or 分 or 秒

出力 : 日付時刻型

説明 : 入力 1 (日付時刻) に入力 2 (数値) を加算した日付時刻を出力します。

例 : 日付時刻の演算結果を出力します。

演算項目-レコード 1:分 レコード 2 : 月

	入力		出力
レコード 1	1984-10-25 12:11:30	2	1984-10-25 12:13:30
レコード 2	2005-1-28 3:10:40	10	2005-11-28 3:10:40

A-5 定数出力フィルタ

出力項目やフィルタにセットしたい固定値を出力するフィルタ群です。



主に、転送先に固定の値を格納したいときや、他のフィルタに引数として値を与えたいときに使います。



固定日付時刻フィルタプロパティ画面



固定数値

入力：引数なし

プロパティ：

表示名

固定出力値 出力する数値（数値）

null 値出力 null 値を出力したい場合にチェックします。

出力：数値型

説明：プロパティに指定された、数値データを出力します。数値フィールドに null 値をセットしたい場合は、null 値出力にチェックします。



固定出力値に設定があった場合でも、null 値出力のチェックがされていると、null 値が出力されます。



出力可能な数値の有効桁数は 15 桁です。



固定文字

入力：引数なし

プロパティ：

表示名

固定出力値 出力したい文字列（文字）

null 値出力 null 値を出力したい場合にチェックします。

出力：文字型

説明：プロパティに指定された、文字データを出力します。文字フィールドに null 値をセットしたい場合は、null 値出力にチェックします。



固定出力値に設定があつた場合でも、null 値出力のチェックがされていると、null 値が出力されます。



固定日付

入力：引数なし

プロパティ：

表示名

固定出力値 出力したい日付データ（yyyyMMdd 形式）

現在日付出力 実行時の現在日付をセットしたい場合にチェックします。

null 値出力 null 値を出力したい場合にチェックします。

出力：日付型

説明：プロパティに指定された、日付データを出力します。日付フィールドに null 値をセットしたい場合は、null 値出力にチェックします。また、実行時の現在日付をセットしたい場合は、現在日付時刻出力にチェックします。



プロパティの優先度は null 値出力 > 現在日付出力 > 固定出力値になっています。null 値出力と現在日付出力にチェックを入れた場合、null 値出力が優先されます。



固定時刻

入力：引数なし

プロパティ：

表示名

固定出力値 出力したい時刻データ（HHmmss 形式）

現在時刻出力 実行時の時刻をセットしたい場合にチェックします。

null 値出力 null 値を出力したい場合にチェックします。

出力：時刻型説明： プロパティに指定された、時刻データを出力します。時刻フィールドに null 値をセットしたい場合は、null 値出力にチェックします。また、実行時の現在時刻をセットしたい場合は、現在日付時刻出力にチェックします。



プロパティの優先度は null 値出力 現在日付出力 固定出力値になっています。null 値出力と現在日付出力にチェックを入れた場合、null 値出力が優先されます。



固定日付時刻

入力：引数なし

プロパティ：

表示名

固定出力値 出力したい日付時刻データ（yyyyMMddHHmmss 形式）

現在日付時刻出力 実行時の日付時刻をセットしたい場合にチェックします。

null 値出力 null 値を出力したい場合にチェックします。

出力：日付時刻型

説明：プロパティに指定された、日付時刻データを出力します。日付時刻フィールドに null 値をセットしたい場合は、null 値出力にチェックします。また、実行時の現在日付時刻をセットしたい場合は、現在日付時刻出力にチェックします。



プロパティの優先度は null 値出力 現在日付出力 固定出力値になっています。null 値出力と現在日付出力にチェックを入れた場合、null 値出力が優先されます。



ループカウント

入力：引数なし

プロパティ：

表示名

出力：数値型

説明：転送処理中の現時点での処理件数番号を出力します。0～件数までの数字が出力されます。



転送先のデータを表示時に転送順に並べたい場合、転送先テーブルのフィールドに格納しておくくと便利です。



実行 ID

入力：引数なし

プロパティ：

表示名

出力：文字型

説明：実行時の識別コードを出力します。

基本的には実行順に連続して発番される数字です。



転送先テーブル内で一意のキーが必要な場合、ループカウントの出力値と実行 ID の両方を転送し、この二つをキーとしておくことで、一意性を保つことができます。

A-6 その他処理フィルタ

条件による出力制御、マスター参照などの処理フィルタ群です。



最大値

入力：任意の同じデータ型の 数値 または日付値 または 時刻項目 または日付時刻（20 個まで）

プロパティ:

表示名

フローの数 最大値を求める対象の入力値の数をセットします。

出力：引数に指定された型

説明：2 つ以上の入力値(数字/日付/時刻/日付時刻)のデータ群の最大値を求めます。数字/日付/時刻/日付時刻の混用は不可能です。出力は入力値属性です

例： 売上金額 5 月、売上金額 6 月、売上金額 7 月の各フィールドを引数として、一番売上金額が大きかった月の金額を出力します。

	入力			出力
レコード 1	5000000	200000	4500000	5000000
レコード 2	8000000	10000000	5000000	10000000

例： 売上日付が一番新しい日を出力します。

	入力1	入力2	出力
レコード 1	1999/10/21	1999/10/23	1999/10/23
レコード 2	2001/10/21	null値	2001/10/21



入力値が null 値の場合、判断対象からはずして判断します。



最小値

入力：任意の同じデータ型の 数値 または日付値 または 時刻項目 または日付時刻（20 個まで）

プロパティ：

表示名

フローの数 最小値を求める対象の入力値の数をセットします。

出力：引数に指定された型

説明：2 つ以上の入力値(数字/日付/時刻/日付時刻)のデータ群の最小値を求めます。数字/日付/時刻/日付時刻の混用は不可能です。

例： 売上金額 5 月、売上金額 6 月、売上金額 7 月の各フィールドを引数として、一番売上金額が小さかった月の金額を出力します。

	入力			出力
レコード 1	5000000	200000	4500000	200000
レコード 2	null値	10000000	5000000	5000000



入力値が null 値の場合、判断対象からはずして判断します。



マスター参照

マスター参照フィルタプロパティ画面

入力：複数個の文字 または 数値 または日付 または 時刻 または日付時刻 （参照キー項目）

プロパティ：

表示名

参照マスター情報 参照するマスターテーブルの情報を表示します。

No. 参照キー項目の番号。入力の個数と同一である必要があります。

キー項目 参照マスターのフィールド一覧より選択する参照キー

出力フィールド名 出力する参照マスターのフィールドを選択

出力： 出力フィールド名プロパティに指定されたフィールドのデータ型

説明： 参照テーブルにキー項目を指定してデータを参照し、指定フィールドの値を返します。

キー項目とキー値は複数組指定できますが、取得できるフィールド情報は1つのみです。

例： 顧客コードフィールドと企業コードフィールドを条件として、顧客マスターより

顧客名を取得します。

	入力		出力
レコード 1	U000010	C1010112	吉田 太郎
レコード 2	U050105	C0111022	山川 春菜

出力値について

キーと出力値の状況	出力
設定されたキーに合致するレコードが1件のみの場合	正常出力
設定されたキーに合致するレコードが0件の場合	null 値出力
設定されたキーに合致するレコードが2件以上の場合	エラー
設定されたキーのデータが null 値の時	エラー

正しくマスター参照ができなかった時の挙動を設定したい場合は IF フィルタを利用することで以下のようなことが可能です。

例) 商品マスター参照

マスターにキーが登録されていなかったレコードの商品名フィールドには"該当無し"を出力

IF フィルタ入力前(=マスター参照出力後)の状態

	入力	出力
レコード 1	U000010	花火 5号
レコード 2	U050105	null

IF フィルターで

1 番目 マスター参照フィルタの出力値

2 番目 結線無し

3 番目 固定文字列フィルタに"該当無し"をセットしたものの出力値

4 番目 再度、マスター参照フィルタの出力値

プロパティの条件 isNull

に設定します。

IF フィルタ出力後の状態

	入力	出力
レコード 1	花火 5 号	花火 5 号
レコード 2	null	該当無し

この値を出力側のフィールドに結線すれば、

マスターにキーが登録されていなかったレコードの商品名フィールドには"該当無し"を出力することができます。

同様に、IF フィルタには、is Error というプロパティの条件もありますので、

Error の挙動も設定することができます。

さらに、Error の場合は、マッパー定義のオプションの転送モードを“エラーをスキップ”という設定にしておけば、エラーが起きたレコードは、転送せず、正しく値が取得できたレコードのみを転送することもできます。



このフィルタは、転送処理ごとに毎回処理されます。よって、パフォーマンスに影響が出ることも想定されます。

ご利用になられる際は、テスト等にてパフォーマンスを考慮した上でお使いください。

一般的に、データと参照マスターを JOIN する転送定義を作成し、実行していただくことでパフォーマンスが向上する可能性が高いです。



CSVマスター参照

CSVマスター参照フィルタプロパティ画面

入力：複数個の文字（参照キー項目）

プロパティ：

表示名

参照マスター情報 参照するマスターファイルの情報を表示します。

No. 参照キー項目の番号。入力の個数と同一である必要があります。

キー項目 参照マスターのフィールド一覧より選択する参照キー

出力フィールド名 出力する参照マスターのフィールドを選択

出力：文字型

説明：参照ファイルにキー項目を指定してデータを参照し、指定フィールドの値を返します。

キー項目は複数組指定できますが、取得できるフィールド情報は1つのみです。

例：顧客コードフィールドと企業コードフィールドを条件として、顧客マスターより顧客名を取得します。

	入力		出力
レコード 1	U000010	C1010112	吉田 太郎
レコード 2	U050105	C0111022	山川 春菜

出力値について

キーと出力値の状況	出力
設定されたキーに合致するレコードが1件のみの場合	正常出力
設定されたキーに合致するレコードが0件の場合	null 値出力
設定されたキーに合致するレコードが2件以上の場合	エラー
設定されたキーのデータが null 値の時	エラー

正しくマスター参照ができなかった時の挙動を設定したい場合は IF フィルタを利用することで以下のようなことが可能です。

例) 商品マスター参照

マスターにキーが登録されていなかったレコードの商品名フィールドには"該当無し"を出力

IF フィルタ入力前(=マスター参照出力後)の状態

	入力	出力
レコード 1	U000010	花火 5 号
レコード 2	U050105	null

IF フィルターで

- 1 番目 マスター参照フィルタの出力値
 - 2 番目 結線無し
 - 3 番目 固定文字列フィルタに"該当無し"をセットしたものの出力値
 - 4 番目 再度、マスター参照フィルタの出力値
- プロパティの条件 isNull
に設定します。

IF フィルタ出力後の状態

	入力	出力
レコード 1	花火 5 号	花火 5 号
レコード 2	null	該当無し

この値を出力側のフィールドに結線すれば、
マスターにキーが登録されていなかったレコードの商品名フィールドには"該当無し"を出力
することができます。

同様に、IF フィルタには、is Error というプロパティの条件もありますので、
Error の挙動も設定することができます。

さらに、Error の場合は、マッパー定義のオプションの転送モードを“エラーをスキップ”という設定にしておけば、エラーが起きたレコードは、転送せず、正しく値が取得できたレコードのみを転送することもできます。

**注意**

このフィルタは、転送処理ごとに毎回処理されます。よって、パフォーマンスに影響が出ることも想定されます。

ご利用になられる際は、テスト等にてパフォーマンスを考慮した上でお使いください。

一般的に、データと参照マスターを JOIN する転送定義を作成し、実行していただくことでパフォーマンスが向上する可能性が高いです。

IF

IF フィルタのプロパティ画面

入力 1: 文字 または 数値 または日付 または 時刻 または日付時刻	比較項目
入力 2: 文字 または 数値 または日付 または 時刻 または日付時刻	比較項目
入力 3: 文字 または 数値 または日付 または 時刻 または日付時刻	真結果値
入力 4: 文字 または 数値 または日付 または 時刻 または日付時刻	偽結果値

パラメータ:

表示名

フィールド 1 入力 1 に結線されているフィールド (値) 情報 (表示項目)

条件 入力 1 と入力 2 の比較条件

選択可能な比較条件は以下のものがあります。

- "==" 引数 1 と 引数 2 が同じ場合真
- "!=" 引数 1 と 引数 2 が違う場合真
- "<" 引数 1 が 引数 2 より小さい場合真
- ">" 引数 1 が 引数 2 より大きい場合真
- "<=" 引数 1 が 引数 2 と同じか小さい場合真
- ">=" 引数 1 が 引数 2 と同じか大きい場合真
- "is Null" 引数 1 が Null 値 の場合真
- "is Error" 引数 1 が Error 値 の場合真
- "固定値" 引数 1 がテキストフィールド 2 の値の場合真
- "正規表現" 引数 1 がテキストフィールド 2 の正規表現の場合真

フィールド 2 入力 2 に結線されているフィールド (値) 情報 (表示項目)

"固定値"、"正規表現"の場合テキストフィールド (結線不可)

真の値 入力 3 に結線されているフィールド (値) 情報 (表示項目)

偽の値 入力 4 に結線されているフィールド (値) 情報 (表示項目)

出力： 入力 3、入力 4 に指定された型

説明： 入力 1 と入力 2 をパラメータの比較条件で比較し、真であれば入力 3 の値を、偽であれば入力 4 の値を返します。

入力 1 と入力 2 のデータ型、入力 3 と入力 4 のデータ型は同一である必要があります。

例： 性別フラグが 1 ならば男性、1 ではないなら女性を出力。固定値フィルタを併用して変換が可能です。

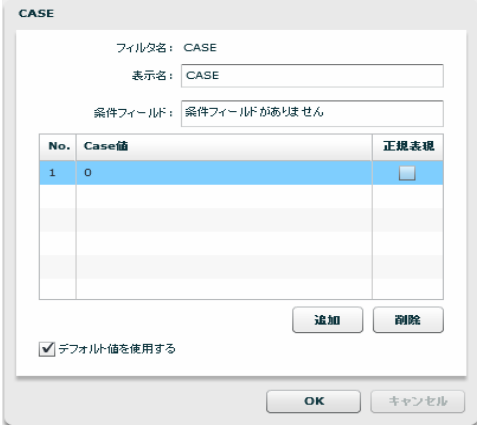
IF における ErrorValue と null 値の扱いについて

入力 1 の入力値が優先的に処理されます。



例) 引数 1 の入力値が Null 値で入力 2 の入力値が ErrorValue の場合出力値は真偽どちらの値でもなく Null です。また、入力 1 の入力値が ErrorValue 値で入力 2 の値が Null の場合出力値は真偽どちらの値でもなく ErrorValue です。

例) 入力 1 の入力値が Null 値で比較条件パラメータが isError の場合、出力値は真偽どちらの値でもなく Null です。また、入力 1 の入力値が ErrorValue で比較条件パラメータが isNull の場合は、出力値は真偽どちらの値でもなく ErrorValue です。


CASE


フィルタ名: CASE
表示名: CASE
条件フィールド: 条件フィールドがありません

No.	Case値	正規表現
1	0	<input type="checkbox"/>

デフォルト値を使用する

追加 削除

OK キャンセル

CASE フィルタのプロパティ

入力 1 : 文字 または 数値 または日付 または 時刻 または日付時刻 判断に使用する値

入力 2 ~ : N 個目の比較値(プロパティ)と等しかったときの結果値

デフォルト値を使用するプロパティが設定されている場合、最終はデフォルト値

プロパティ

表示名

条件フィールド 入力 1 に結線したフィールド (値) 情報

No. 比較値の番号 (引数 N-1 の値)

Case 値 入力 2 ~ の比較値 (入力 2 ~ 比較値のデータ型と同じ型)

正規表現 Case 値を正規表現で扱う。

追加 比較する対象の値を追加します。(その分の結線が必要)

削除 比較する対象の値を削除します。(結線情報も削除されます。)

デフォルト値を使用する どの Case 値にも合致しなかったときの出力値を利用する場合にチェックします。このチェックがありの場合、最後の入力値がデフォルト値として出力されます。

出力 : 入力 2 ~ N に指定された型

説明 : 入力 1 の項目の値により、出力項目を切り替えます。

入力 1 の値が、1 つめのプロパティ Case 値と等しいとき、入力 2 を結果値として返します。同様に、入力 1 の項目が 2 つめのプロパティ Case 値と等しいとき入力 3 の結果値を返します。正規表現使用の場合は入力 1 がプロパティ Case 値の正規表現になっている場合結果値を返します。”デフォルト値を使用する”チェックを設定することにより、どの Case 値にも当てはまらない場合の出力値を最終入力で設定することができます。

例 : 店舗コードの値により、出力する文字を切り替えます。

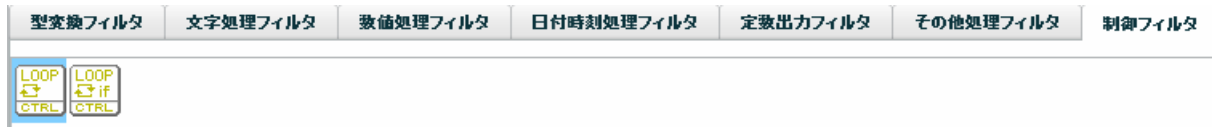
	入力	出力
レコード 1	TOKYO	東京
レコード 2	OSAKA	大阪
レコード 3	New York	その他

出力値について



入力値と比較値の状況	出力
入力値が Case の比較値に合致せず、default が設定されていない場合	エラー
入力値が NULL で比較値が任意の値の場合	null 値出力

A-7 制御フィルタ



シンプルループ

入力 1：入力リソースの“レコード”項目

プロパティ：

表示名 画面上に表示する名称です。

出力： True（出力リソースのレコード項目以外には結線できません）

説明： 指定したレコードを基準にループさせたい場合に使用するループフィルタです。



ループ

入力 1：入力リソースの“レコード”項目

入力 2：文字/数値/日付時刻/日付/時刻のうちのいずれか

プロパティ：

表示名 画面上に表示する名称です。

型 入力 2 に引かれた処理の型を指定します。

条件 [出力値] と入力 2 の値との比較条件。“入力 2[条件]出力値”で評価します。

比較値 入力 2 との比較値を指定します。

出力： True または False（出力リソースのレコード項目以外には結線できません）

説明： 現在のレコードを実行するか判定するフィルタです。入力 2 と比較値を [条件] で評価した結果、真となるレコードが実行されます。型には入力 2 の値の型を指定します。条件には次のような比較式があります。

==： 入力 2 と[比較値]が同値の場合実行されます。

!=： 入力 2 と[比較値]が異なる場合実行されます。

>： 入力 2 が[比較値]よりも大きい場合実行されます。

<： 入力 2 が[比較値]よりも小さい場合実行されます。

>=： 入力 2 が[比較値]以上の場合実行されます。

<= : 入力 2 が[比較値]以下の場合実行されます。

Is Null : 入力 2 が NULL の場合実行されます。この場合型や比較値は無視されます。

Is Error : 入力 2 が Is Error (フィルタ処理エラー) の場合実行されます。この場合型や比較値は無視されます。



リソースに RDB を選択している場合、データの絞込みとして検索条件を指定できます。これはループフィルタと機能が似ています。両方を指定した場合には、検索条件によって絞り込まれたデータに対してループフィルタが適用されます

例 : 売上金額が 2000 以上のレコードについてのみデータ処理したい場合。入力 2 に売上金額フィールドを結線し、プロパティに以下のように指定します。

型 : 売上金額のデータ型を指定

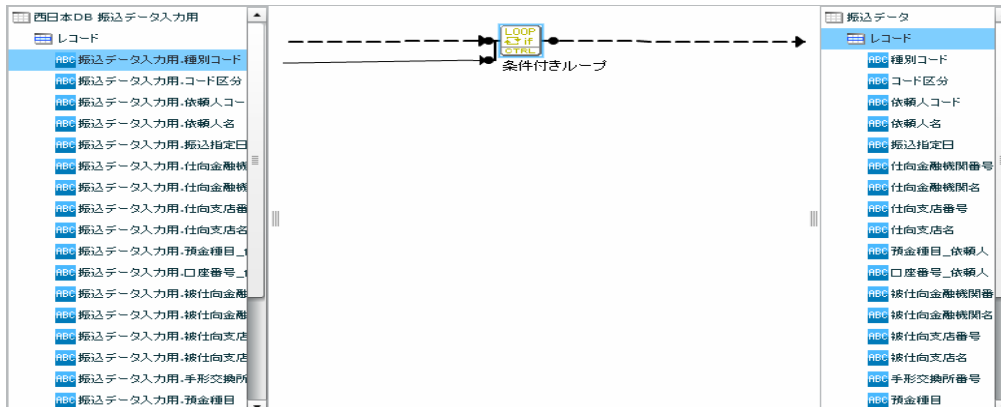
条件 : >= を指定 (売上金額(入力 2)>=2000)

比較値 : 2000 を指定

結果、売上金額が 2000 以上のレコード 2 とレコード 4 のみデータ転送処理が実行されます。

	売上日	売上金額	地域
レコード 1	2009/7/14	1000	埼玉県三郷
レコード 2	2009/7/14	2000	東京都大田区
レコード 3	2009/7/15	900	神奈川県横浜市
レコード 4	2009/7/15	2500	東京都大田区

(使用例)



入力値が null 値の場合、判断対象からはずして判断します。

付 録 B

付録 B 対応データ型

B 対応データ型

ここでは、本製品で転送可能なデータ型とその注意点を記載します。

以降の説明で×が付いているまたは、表記の無いデータ型(IMAGE/BLOB など)はサポートされません。


がついているものは、データの転送は可能ですが、テーブル作成機能にてサポートされないデータ型です。



データベース固有のデータ型をそのままフィールド名として使用すると正常に動作しない場合がありますので利用しないようにしてください。

また、フィールド名に「.」などの記号を含めた場合、DBMS 種類によっては正常に動作しない場合がありますので、利用しないようにしてください。

B-1 文字型

画面上で、のアイコンで表示されるフィールドのデータ型です。

DB2/UDB	ORACLE	MS SQL-Server	AS/400 RDB (DDS/IDDU)	AS/400 SQL Table	ACCESS 2003	MySQL	データ型の 説明
(CHAR)	(CHAR)	(CHAR)	-	(CHAR)	-	(CHAR)	短い固定長の文字列 英数カナ半角
-	-	-	(A)(*4)	-	-	-	オープン
-	-	-	(O)(*4)	-	-	-	専用
-	-	-	(J)(*4)	-	-	-	択一
-	-	-	(E)(*4)	-	-	-	ユニコード 固定文字列
(VARCHAR)	(VARCHAR2)	(VARCHAR)	-	(VARCHAR)	テキスト 型	(VARCHAR)	可変長の文字列(*1)
	(VARCHAR)						
(LONG VARCHAR)	(LONG)	(TEXT)	-	(LONG VARCHAR)	メモ型	(LONG VARCHAR)	長い可変長の文字列 (*1)
		(VARCHAR(MAX))					
-	(NCHAR2)(*2)	(NCHAR)(*2)	-	-	-	-	ユニコード 可変長文字 列
		(NCHAR(MAX))					
		(NTEXT)(*2)					
-	-	-	-	-	データリ ンク型(*3)	-	
(GRAPHIC)	-	-	(G)	(GRAPHIC)(*2)	-	-	グラフィック
(VARGRAPHIC)	-	-	-	(VARGRAPHIC)	-	-	グラフィック
		O(XML)					


注

(*1)文字型は、可変長文字列です。転送サイズは最大 32766 バイトです。

(*2)テーブル作成機能にて、NCHAR、NVARCAHR、NTEXT、GRAPHIC 型にて文字列長として設定可能なのは文字数です。作成されたフィールドの文字列長は指定した数値の2または3倍の文字列長として表示され、これらのフィールドの文字列長は変更できません。

(*3)Access のデータリンク型は、通常の文字型としてデータ転送は可能ですが、文字列の前後に#が追加されて転送されます。

B-2 数値型

画面上で、のアイコンで表示されるフィールドのデータ型です。

DB2/UDB	ORACLE	MS SQL-Server	AS/400 RDB (DDS/IDDU)	AS/400 SQL Table	ACCESS 2003	MySQL	データ型の説明
(INTEGER)	(NUMBER)	(INT)	-	(INTEGER)	長整数 型	(INTEGER)	-2147483648 から 2147483647 までの 32 ビットの符号付き整数値
-	-	-	-	-	オートナ ンバー型 (*5)	-	
(BIGINT)(*2)	(NUMBER)(*2)	(BIGINT)(*2)	-	(BIGINT)(*2)	-	(BIGINT)(*2)	-9223372036854775808 から 9223372036854775807 までの 64 ビットの符号付き整数値
(SMALLINT)	(NUMBER)	(SMALLINT)	-	(SMALLINT)		(SMALLINT)	-32768 から 32767 までの 16 ビットの符号付き整数値
-	(NUMBER)	(TINYINT)				-	0 から 255 までの符号付き または符号なしの 8 ビットの整 数値
		x (BIT)			x (YES/NO 型)		
(DECIMAL)	(NUMBER)	(DECIMAL)	-	-	-	(DECIMAL)	10 進数
		(MONEY)					
		(SMALLMONEY)					
-	(NUMBER)	(NUMERIC)	-	-	(通貨型)	-	
-	-	-	(P)	(DECIMAL)			バック 10 進(*1)
-	-	-	(S)	(NUMERIC)			ゾーン 10 進(*1)
(REAL)	(NUMBER)	(REAL)	(F)	(REAL)		(REAL)	7 桁の仮数部をサポートする 「単精度」の浮動小数点数
(FLOAT)(*3)(*4)	(FLOAT)	(FLOAT)(*3)		(FLOAT)(*3)		(FLOAT)(*3)(*4)	
(DOUBLE)	(NUMBER)	(FLOAT)(*3)	-	(DOUBLE)		(DOUBLE)	15 桁の仮数部をサポートする 「倍精度」の浮動小数点数
(FLOAT)(*3)(*4)				(FLOAT)(*3)		(FLOAT)(*3)(*4)	基本的には JDBC 型の DOUBLE と等価

注

- (*1) ゾーン 10 進 / バック 10 進は内部での表記方法であり、データベースに対してはどちらも同じ 10 進数です。
- (*2) データベースの精度は 19 桁(64 ビット)ですが、フィルタ等で処理する場合の有効桁数は 15 桁になります。
- (*3) FLOAT は n の指定により扱いが変わります。1~24 単精度、25~53 倍精度
- (*4) DB2 では、FLOAT で作成されたフィールドはその n の指定値により、テーブル作成時に REAL または DOUBLE に変換されてしまいますので FLOAT 型は存在しません。
- (*5) オートナンバー型は転送元にも指定可能です。このフィールドデータを書き込めません。

テーブル作成時の Oracle の NUMBER 型の制約について

NUMBER 型は、有効桁数 38 桁、位取り-84~127 の指定が可能ですが、テーブル作成で使用できる有効桁数と位取りの値は、以下の制約があります。ご注意ください。




有効桁数： 1~38

位取り： 0~38 (さらに、有効桁数以下であること)



DBMS、データ型により制限値は異なりますが、フィルタを利用した処理を行う場合は、有効桁数は 15 桁になりますのでご注意ください。

B-3 日付型


画面上で、のアイコンで表示されるフィールドのデータ型です。

DB2/UDB	ORACLE	MS SQL-Server	AS/400 RDB (DDS/IDDU)	AS/400 SQL Table	ACCESS 2003	データ型の説明
(DATE)	(DATE)(*1)	-	(L)	(DATE)	-	日付、月、および年

注


(*1) Oracle のバージョンにより時刻を含む場合があります。

B-4 時刻型

画面上で、のアイコンで表示されるフィールドのデータ型です。

DB2/UDB	ORACLE	MS SQL-Server	AS/400 RDB (DDS/IDDU)	AS/400 SQL Table	ACCESS 2003	データ型の説明
(TIME)	-	-	(T)	(TIME)	-	時刻、分、秒

B-5 日付時刻型

画面上で、のアイコンで表示されるフィールドのデータ型です。

DB2/UDB	ORACLE	MS SQL-Server	AS/400 RDB (DDS/IDDU)	AS/400 SQL Table	ACCESS 2003	MySQL	データ型の説明
(TIMESTAMP)	(DATE)(*6)	(DATETIME)	(Z)	(TIMESTAMP)	日付/時刻型	(TIMESTAMP)	DATE と TIME とナノ秒
	(TIMESTAMP)					(DATETIME)	
	× (TIMESTAMP with ...)						
		O(SMALLDATETIME)					

付録 C

付録 C 便利な機能

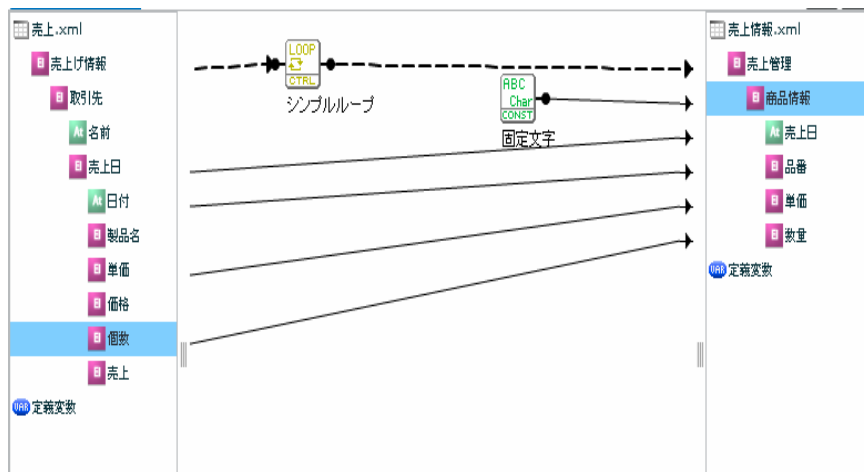
C 便利な機能

ここでは本製品を扱う中で便利な機能をご紹介します。

範囲選択

マッピングの画面上で複数のフィルターやフローを選択することができます。

1. 画面上で右クリックを押し範囲の開始地点を指定します。
2. そのまま押下しながら選択する範囲を指定します。
3. 指定した範囲内のフィルター及びフローが選択状態になります。



範囲選択状態のマッピング画面

範囲移動

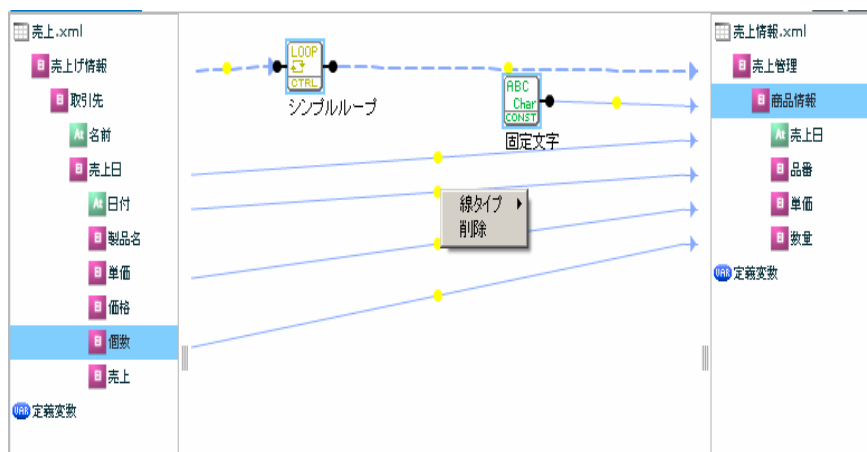
範囲選択状態のフィルタを移動したい場所にドラッグ&ドロップすることで範囲選択されたフィルタ及びアローが全て同時に移動します。

データリソースに直接つながっているアローはコピーの対象外になります。

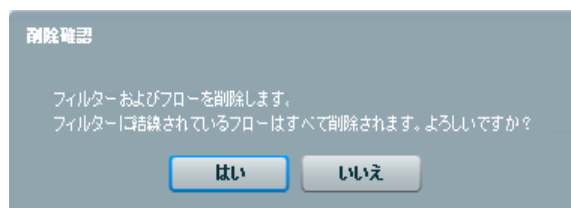
範囲削除

1. 範囲選択状態のフィルタもしくはフローの上で右クリックします。

範囲選択状態で DELETE キーを押した場合 3 へ



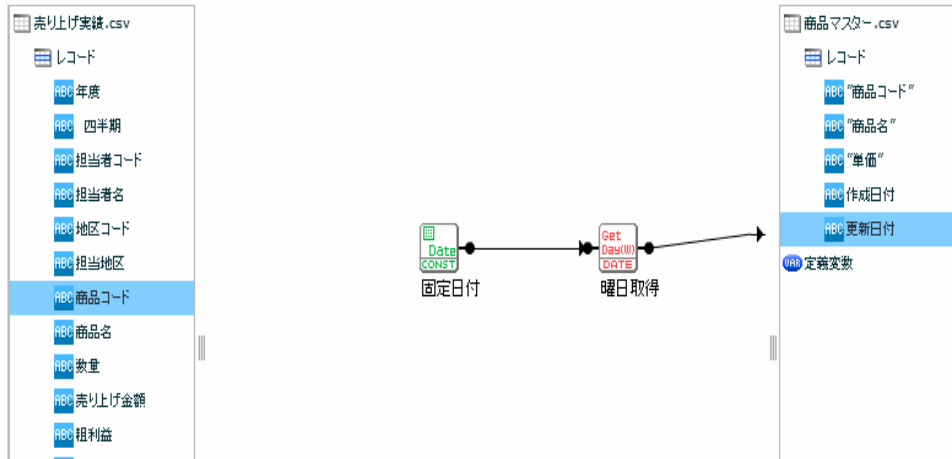
2. 削除を選択します。



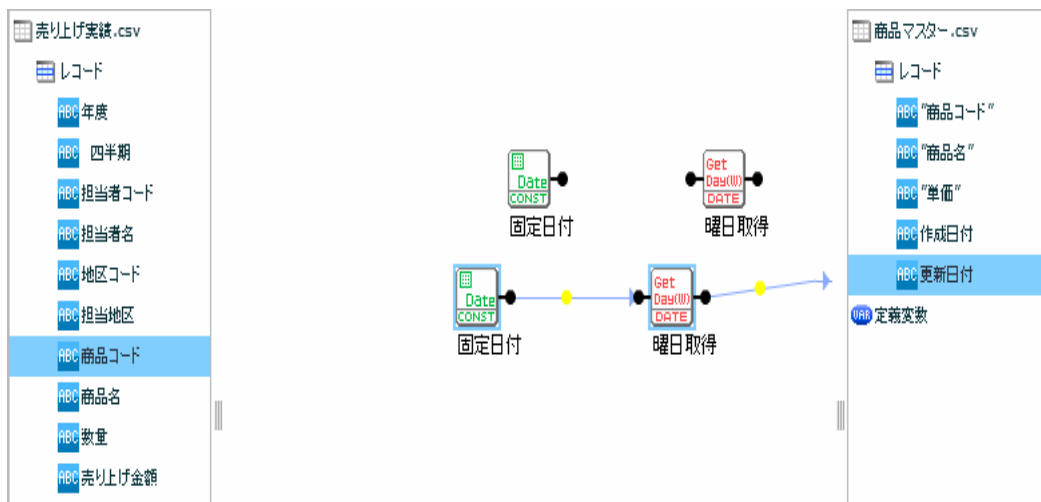
3. 「はい」を押します。 削除

範囲コピー・貼り付け

1. 範囲選択状態のフィルタもしくはフローの上で右クリックします。
又は範囲選択状態で CTRL + C を押します。 3へ
2. コピーを選択します。



3. 貼り付けたい場所で右クリックし、貼り付けを選択します。
又は CTRL + V を押します。

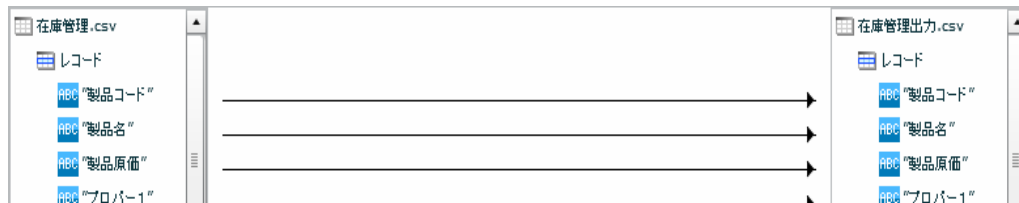


データリソースに直接つながっているアローはコピーの対象外になります。

同時スクロール

転送元、転送先のスクロールを同時に行うことができます。

1. 転送元、転送先を選択状態にします。



2. コントロールキーを押した状態で、転送元、もしくは転送先のスクロールバーをドラッグします。
転送元、転送先が同時にスクロールします。



WebReport2.0 Datamart ユーザーズマニュアル

2009年 4月 22日 第1版発行
2009年 7月 17日 第2版発行
2009年 7月 21日 第3版発行
2012年 12月 14日 第4版発行

発行 JB アドバンス・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/>

本書は著作権上の保護を受けており、本書の全部あるいは一部に関して、JB アドバンス・テクノロジー株式会社からの文書による許諾を得ず、無断で複写、複製することは禁じられています。また、本書はユーザーへ通知することなく変更される場合があります。

