



# ***Qanat 2.0***

ユーザーズマニュアル

## ごあいさつ

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

(以下、「Qanat 2.0」を「本製品」と呼び、ご説明して参ります。)

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

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
著作権/ご注意 .....	i
本書の表記方法について .....	i
目次 .....	ii
<b>第1章 システムの概要 .....</b>	<b>6</b>
1.1 製品概要 .....	7
1.1.1 Qanat 2.0 とは .....	7
1.1.2 Qanat を活用するメリット .....	7
1.2 製品の構成とユーザー .....	8
1.2.1 Qanat Server とは .....	9
1.2.2 定義者機能(Flow Designer)とは .....	9
1.2.3 運用者機能(Web)とは .....	9
1.3 運用の流れ .....	10
<b>第2章 操作方法 .....</b>	<b>11</b>
2.1 ログイン/ログアウト .....	12
2.1.1 ログイン (Flow Designer) .....	12
2.1.2 ログアウト (Flow Designer) .....	13
2.1.3 ログイン (Web) .....	14
2.1.4 ログアウト (Web) .....	15
2.2 フロー定義一覧 (Flow Designer) .....	16
2.2.1 フロー定義とは .....	16
2.2.2 メニュー .....	16
2.2.3 定義一覧画面 .....	17
2.2.4 フォルダの開閉 .....	18
2.2.5 定義の新規作成 .....	20
2.2.6 フロー定義初期設定 .....	21
2.2.6.1 トリガー変数とは? .....	28
2.2.7 フロー定義の操作 .....	29
2.2.8 仕様書とは .....	30
2.2.8.1 仕様書のデザイン .....	31
2.3 実行結果の確認 (Web) .....	41
2.3.1 実行結果とは .....	41
2.3.2 実行結果の画面について .....	41
2.3.2.1 ログの詳細情報の確認 .....	43

2.3.3	メニュー	45
2.3.4	絞り込み条件	46
2.3.5	ログフォント色設定	47
2.3.6	ヘルプ画面	48
2.4	フロー定義一覧 (Web)	49
2.4.1	フロー定義一覧画面について	49
2.4.2	メニュー	50
2.4.3	絞り込み条件	50
2.4.4	設定変更	51
2.5	フロー定義画面 (Flow Designer)	56
2.5.1	フロー定義画面	56
2.5.2	メニュー	57
2.5.2.1	名前をつけて保存の概要	57
2.5.2.2	一般設定の概要	58
2.5.2.2.1	実行時データリソース切り替え設定	58
2.5.3	フロー定義領域の概要	62
2.5.3.1	コンポーネントの追加	63
2.5.3.2	範囲コンポーネントの追加	64
2.5.3.3	コンポーネントの種類	65
2.5.3.4	コンポーネントの操作	67
2.5.3.5	コンポーネントの操作の種類	68
2.5.3.6	コンポーネントの編集	69
2.5.3.7	マッパーのコピー	69
2.5.3.8	コピーしたマッパーの追加	70
2.5.3.9	分岐の追加	72
2.5.3.10	「分岐開始」での分岐削除	73
2.5.3.11	「追加分岐」での分岐削除	75
2.5.3.12	「トランザクション開始」の削除	76
2.5.3.13	分岐条件の拡大と縮小	77
2.5.3.14	分岐条件の設定	78
2.5.3.15	分岐条件で使用可能な変数	79
2.5.4	フロー変数定義	80
2.5.4.1	フロー変数とは?	80
2.5.4.2	変数設定	81
2.6	マッパー編集画面 (Flow Designer)	82
2.6.1	マッパーとは	82
2.6.2	マッパー編集画面について	82
2.6.3	共通メニュー	84
2.6.4	マッパーの設定画面	85
2.6.5	コンポーネントのタイトル	86
2.6.6	マッパーの種類	86
2.6.7	前後処理	86
2.6.8	リソース選択	90
2.6.9	RDB 選択	92

2.6.10	RDB 作成・変更	102
2.6.11	RDB を利用したときの各種メニュー	106
2.6.12	CSV ファイル選択	114
2.6.13	CSV ファイルを利用したときの各種メニュー	118
2.6.14	固定長ファイル選択	125
2.6.15	固定長ファイルを利用したときの各種メニュー	128
2.6.16	固定長ファイル転送におけるループ処理	139
2.6.17	XML ファイル選択	140
2.6.18	XML ファイルを利用したときの各種メニュー	143
2.6.19	XML ファイル転送におけるループ処理	152
2.6.20	EXCEL ファイル選択	153
2.6.21	EXCEL ファイルを利用したときの各種メニュー	158
2.6.22	EXCEL ファイルの範囲選択	173
2.6.23	EXCEL ファイル転送におけるループ処理	176
2.6.24	SALESFORCE アダプターの仕様について	177
2.6.25	SALESFORCE 選択	178
2.6.26	SALESFORCE を利用したときの各種メニュー	186
2.6.27	メール送信設定	191
2.6.28	メール送信を選択したときの各種メニュー	192
2.6.29	WebReport 選択	193
2.6.30	PDF ファイル選択	195
2.6.31	PDF ファイルの制限事項	205
2.6.32	KINTONE 選択	206
2.6.33	kintone アダプターの制限事項	222
2.6.33	Amazon S3(CSV)選択	223
2.6.34	Amazon S3(CSV)を利用したときの各種メニュー	227
2.6.35	@TOVAS 選択	233
2.6.36	@TOVAS を選択したときの各種メニュー	246
2.6.37	フィルタのプロパティ	253
2.7	スクリプト定義(Flow Designer)	254
2.7.1	スクリプト定義とは	254
2.7.2	使い方	254
2.8	QanatExecute	257
2.8.1	QanatExecute とは	257
2.8.2	使い方	257
2.9	QanatExecute.js	259
2.9.1	QanatExecute.js とは	259
2.9.2	使い方	259

## 付録 A フィルタリファレンス..... 264

A	フィルタリファレンス	265
A-1	型変換フィルタ	266
A-2	文字変換フィルタ	273
A-3	数値処理フィルタ	279

A-4	日付時刻処理フィルタ .....	283
A-5	定数出力フィルタ .....	288
A-6	その他処理フィルタ .....	293
A-7	制御フィルタ .....	307
付録 B	対応データ型 .....	309
B	対応データ型 .....	310
B-1	文字型 .....	310
B-2	数値型 .....	311
B-3	日付型 .....	312
B-4	時刻型 .....	312
B-5	日付時刻型 .....	312
付録 C	便利な機能 .....	313
C	便利な機能 .....	314
付録 D	REST API .....	316
D	REST API について .....	317
	REST API の概要 .....	317
	定義の実行 (POST) .....	321
	結果の確認 (GET) .....	322

# 第 1 章 システムの概要

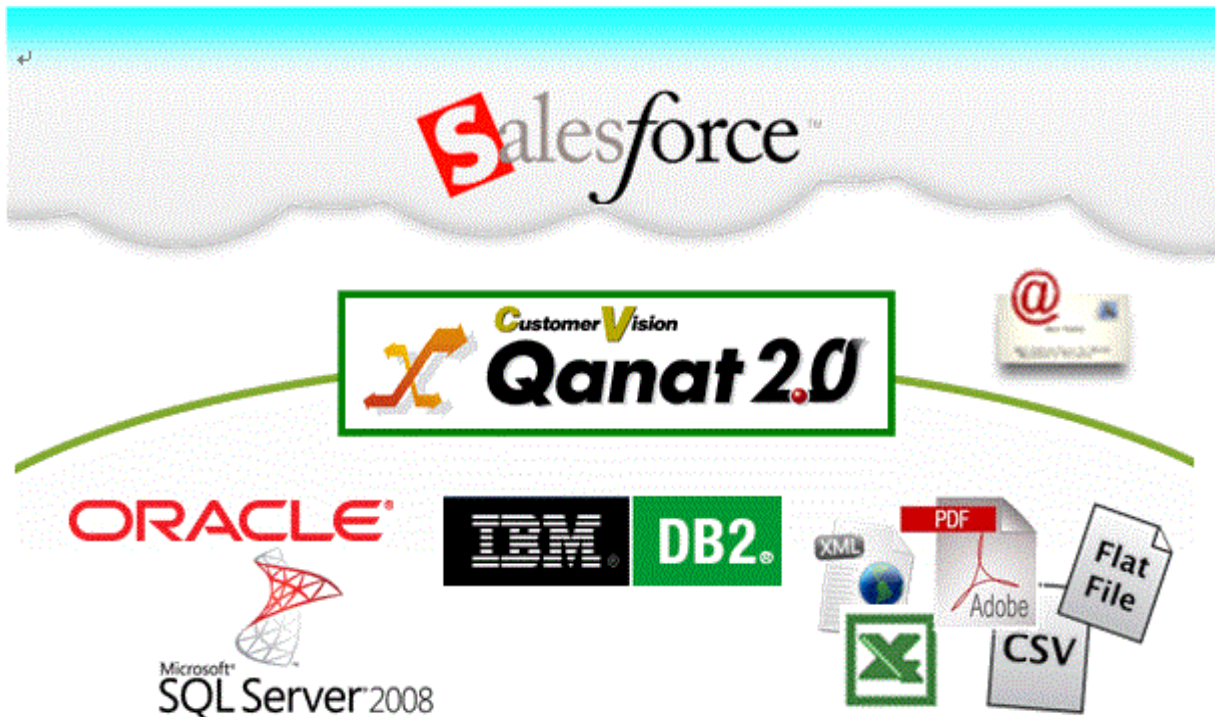
## 1.1 製品概要

本製品はシステム間の仕様の違いを吸収し、データの変換、連携を行うツールです。

Salesforce などのクラウドやデータベース、ファイル上の様々なデータをフィルタやスクリプトを使ってグラフィカルに加工し、他のクラウド、データベース、ファイルに転送することができます。

### 1.1.1 Qanat 2.0 とは

- ・様々なデータをユーザーが管理・参照しやすいよう加工し、データベース、ファイルに転送するツールです。



データ転送元とデータ転送先、変換方法を設定することで、様々なデータ転送が可能です。

### 1.1.2 Qanat を活用するメリット

- ・システム間の仕様を吸収し、グラフィカルに変換、連携ができます。
- ・変換、連携、実行の管理に必要な機能がすべてパッケージ化されています。

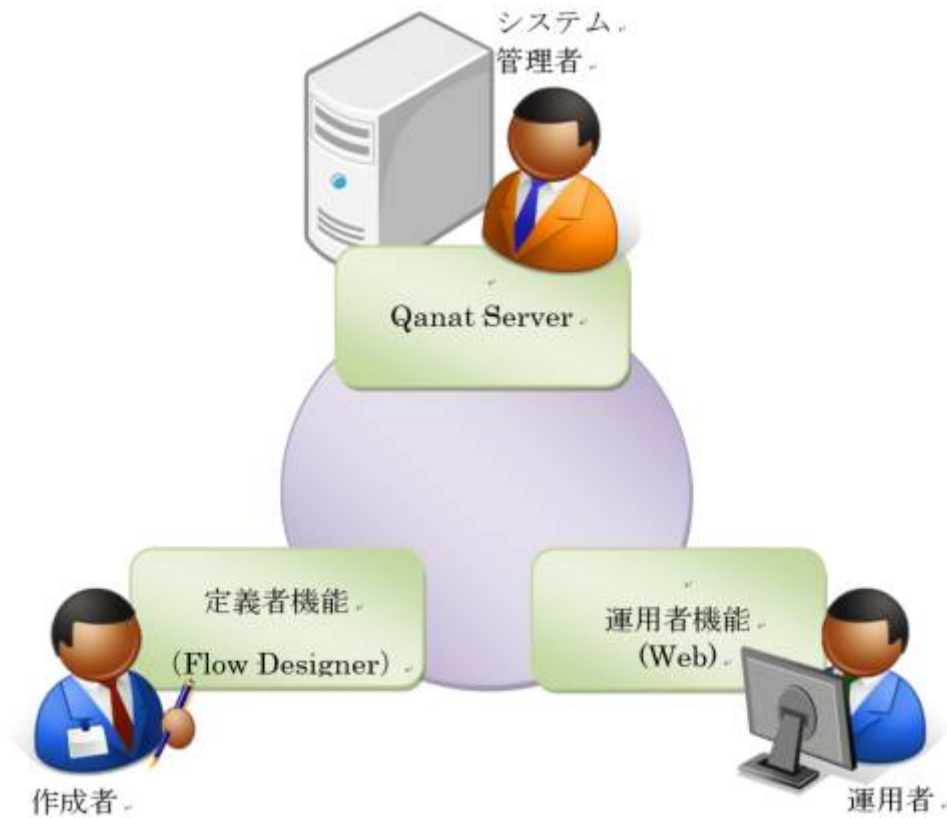


## 1.2 製品の構成とユーザー

この製品は、3つの要素で構成されています。

このマニュアルでは「定義者機能 (Flow Designer)」と「運用者機能」について説明しています。

- ・フロー定義を実行する本体である「Qanat Server」
- ・フロー定義を作成運用する為の「定義者機能 (Flow Designer)」
- ・フロー定義を運用する為の「運用者機能 (Web)」



## 1.2.1 Qanat Server とは

フロー定義が実行される本体です。導入時のシステム設定の他、必要に応じてシステム管理者が「CV 管理ツール」を使用して、下記のような設定を行います。

- ・ログイン権限の設定
- ・ユーザーの追加と削除
- ・データリソースの設定

## 1.2.2 定義者機能(Flow Designer)とは

定義作成者が使用する、Qanat Server のフロー定義を作成する為の機能です。

専用のクライアントツール(Flow Designer)を使って、Qanat Server にアクセスを行い、下記のような事を行います。

- ・データの抽出、変換、転送
- ・フローの分岐条件設定
- ・定義がどのようなタイミング、条件で実行されるのかについて設定

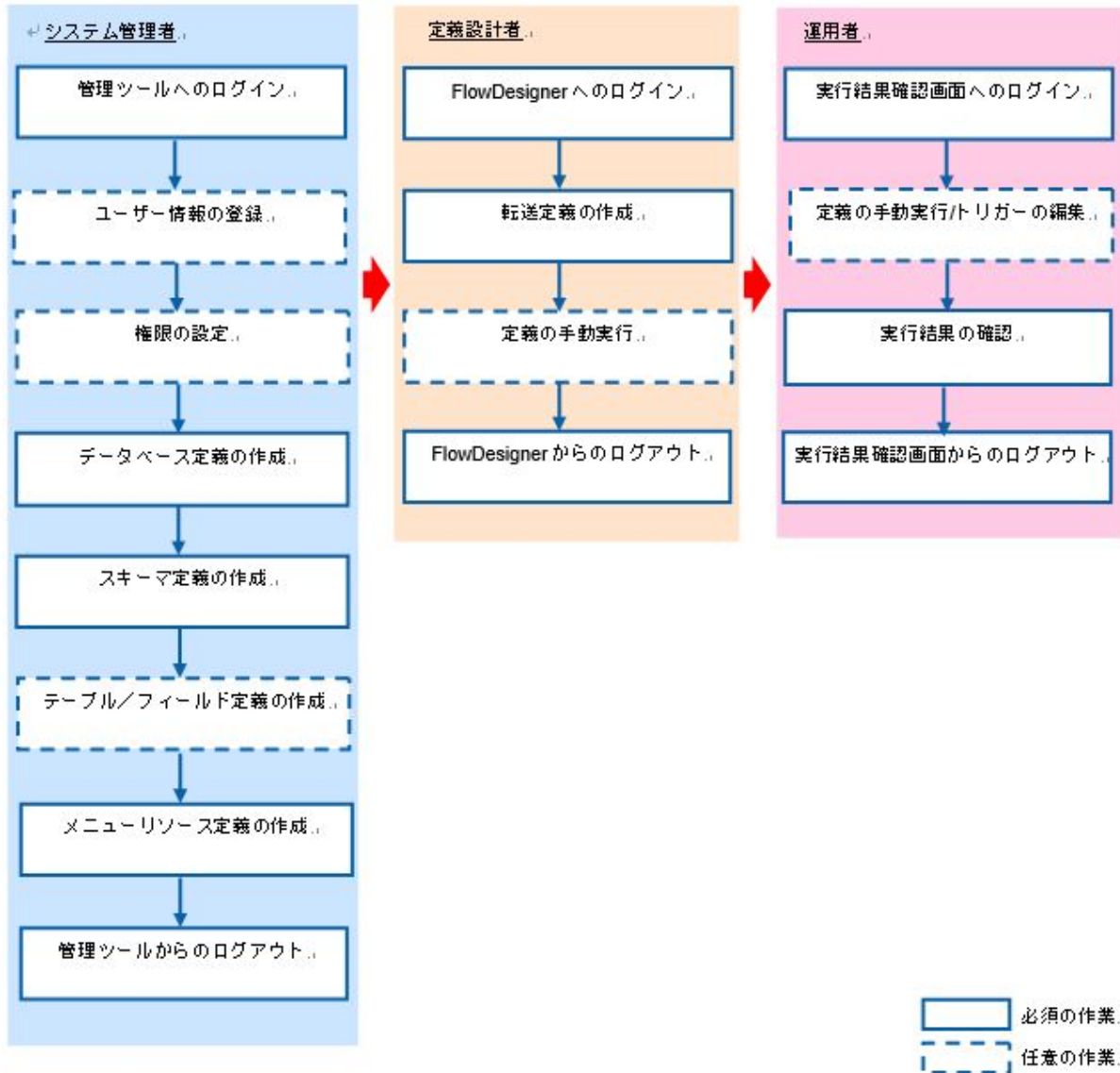
## 1.2.3 運用者機能(Web)とは

Qanat Server のフロー定義の実行状態を確認する為の機能です。定義運用者が使用します。

ブラウザから Qanat Server にアクセスを行い、下記のような事を行います。なお、運用者機能は定義者機能からも呼び出し可能です。

- ・定義の動作結果の確認
- ・動作定義の ON/OFF

## 1.3 運用の流れ



運用の流れ

## 第 2 章 操作方法

## 2.1 ログイン/ログアウト

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。

### 2.1.1 ログイン (Flow Designer)

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 [すべてのプログラム] - [Flow Designer] を選択します。
- 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。



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。

## 2.1.2 ログアウト (Flow Designer)

本製品にログインすると、フロー定義一覧画面が表示されます。

右上の赤枠のアイコンをクリックすると、ログアウト確認画面が表示され、「OK」を押すとログアウトし、プログラムは終了します。



フロー定義一覧画面



ログアウト確認画面

### 2.1.3 ログイン (Web)

本製品へのログインは、以下の手順で行います。

- 1 Web ブラウザで次の URL を実行します。

`http://[Qanat Server ホスト名]/SmartBI/QanatWeb/Manager/Login`

- 2 [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。

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。



ログイン画面

定義一覧画面が表示されます。



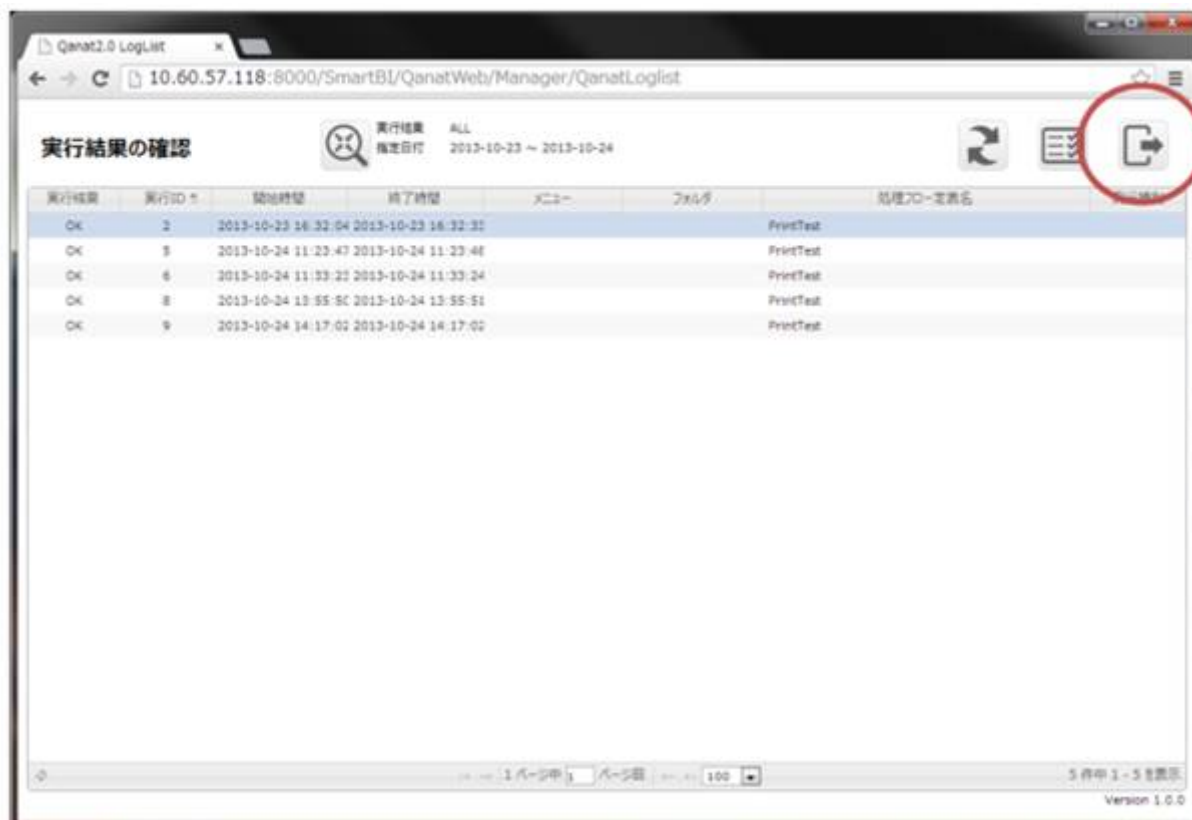
本製品にログインするためには、特別な設定が必要です。

ログインできない場合は管理者の方にご連絡ください。

## 2.1.4 ログアウト (Web)

本製品にログインすると、フロー定義一覧画面が表示されます。

右上の赤枠のアイコンをクリックすると、ログアウト確認画面が表示され「OK」を押すとログアウトし、ログイン画面に移動します。



フロー定義一覧画面



ログアウト確認画面



## 2.2 フロー定義一覧 (Flow Designer)

ここでは、定義者機能/Flow Designer のログイン後に表示される、フロー定義一覧画面の機能について説明します。

### 2.2.1 フロー定義とは

フロー定義にはマッパー定義、トランザクション等の動作を任意の順番で実行するように定義することができます。

また、フロー変数やコンポーネント変数を利用することで特定の条件で実行を分岐することができます。

### 2.2.2 メニュー

メニューには[実行結果の確認]、[ログアウト]、[ヘルプ]の3つのボタンが表示されます。



メニューバー

[実行結果の確認] ボタン

ブラウザが起動し、運用者機能の「ログイン画面」が表示されます

[ログアウト] ボタン

ログアウトします。

[ヘルプ] ボタン

ブラウザが起動し、マニュアルが表示されます。



マニュアルの参照のためには、インターネット接続が必要です。

## 2.2.3 定義一覧画面



定義一覧画面

使用可能なメニューとフォルダを表示します。

また、フォルダが開いている状態では、定義の情報を表示します。

メニュー名称	定義が含まれるメニューの名称です
フォルダ名称	定義が含まれるフォルダの名称です
定義数	フォルダに含まれる定義の数です
定義名称	作成時等で設定した定義の名称です
トリガー情報	作成時等で設定したトリガーの設定内容の要約です
作成情報	定義を作成したユーザーと、日時です
更新情報	定義を更新したユーザーと、日時です
コメント	定義に設定されたコメント（128文字まで）です
フォルダ	定義が格納されているメニュー＆フォルダ情報です

## 2.2.4 フォルダの開閉

閉じているフォルダは、下記の赤丸の[開く]ボタンをクリックすることで、開くことができます。(次ページが開いた状態です)



定義一覧画面（フォルダが閉じている状態）

開いているフォルダは、下記の赤丸の[閉じる]ボタンをクリックすることで、閉じる事ができます。(前のページが閉じた状態です)



定義一覧画面 (フォルダが開いている状態)

## 2.2.5 定義の新規作成

フォルダにある新規作成のボタン（下記の赤枠）を押すと、次ページの「フロー定義初期設定」画面が表示されます。

そのフォルダの中に新しく定義を作成することができます。



定義一覧画面

## 2.2.6 フロー定義初期設定

定義を新規に作成した際の初期設定項目が表示されます



定義一覧画面（初期状態/「手動実行」選択）

### < 共通の設定項目 >

使用可能なメニューとフォルダを表示します。

また、フォルダが開いている状態では、定義の情報を表示します。

メニュー	定義を格納するメニューの名称です
フォルダ	定義を格納するフォルダの名称です
処理フロー定義名	作成する定義の名称です
トリガー種類	この定義を実行する際のトリガーの種類を指定します トリガーの種類によって、設定項目や参照できるトリガー変数が 変化します
実行結果を通知する	「実行結果を通知する」機能を使用するかについて指定します チェックすると、設定項目が追加されます
[OK]	現在の設定内容で、定義を作成します
[キャンセル]	新規作成をキャンセルし、一覧画面に戻ります

## &lt; 「実行結果を通知する」の設定項目 &gt;

<input checked="" type="checkbox"/> 実行結果を通知する
通知条件 <input type="checkbox"/> 成功 <input type="checkbox"/> 警告 <input type="checkbox"/> エラー <input type="checkbox"/> 中止
送信先メールアドレス <input type="text"/>
詳細ログの添付 <input type="radio"/> 添付しない <input checked="" type="radio"/> 添付する

## 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です

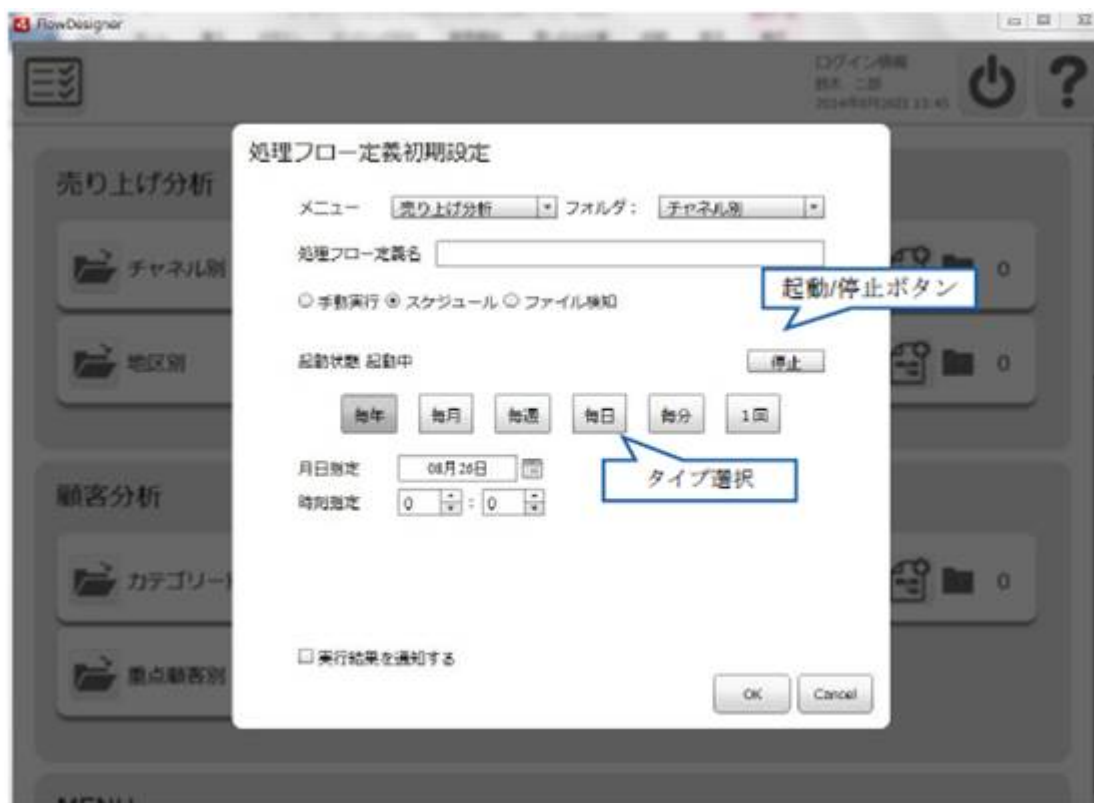
カンマで区切ることで複数のアドレスを指定できます。

## 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

## &lt;トリガー「スケジュール」の設定項目&gt;



定義一覧画面（「スケジュール」選択）

データ転送定義を特定の日時に自動的に実行できるトリガーです。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。

起動・停止ボタン

定義の起動状態を設定します

タイプ選択

スケジュールのタイプを設定します。

タイプによって設定項目が変化しますので、各タイプで必要な実行タイミングを指定します。

毎年

毎年、任意の日時に実行します

月日指定

時刻指定  :



## 毎月

毎月、任意の日時に実行します

日指定  

時刻指定   :

## 毎週

毎週、任意の曜日の任意の時間に実行します

時刻指定   :

曜日指定 日 月 火 水 木 金 土

## 毎日

毎日、任意の時間に実行します

時刻指定   :

## 毎分

特定の間隔（分単位）で実行します

分指定   ごとに実行

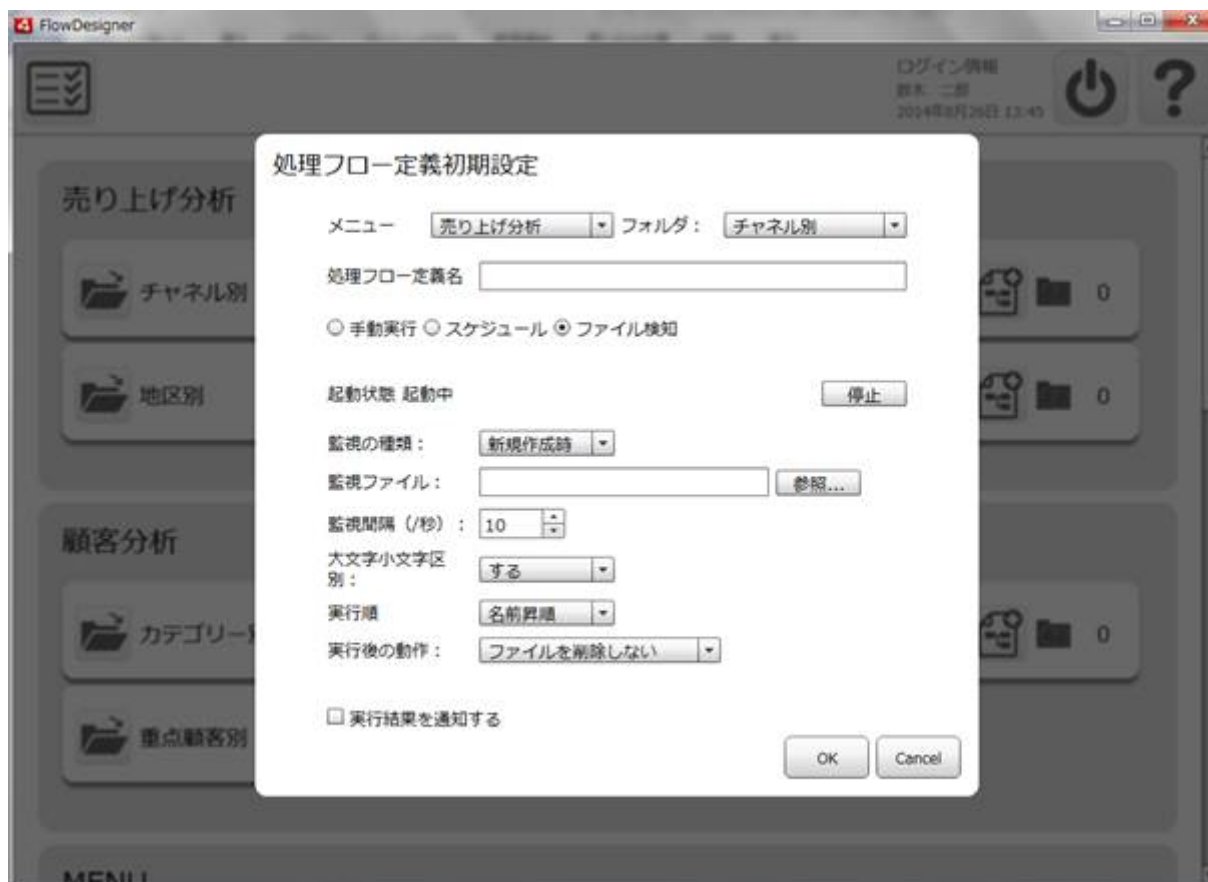
## 1回

任意の日時に一回だけ実行します

日付指定  

時刻指定   :

## &lt; トリガー「ファイル検知」の設定項目 &gt;



データ転送定義をファイル監視で自動的に実行できるトリガーです。

ファイルトリガーを利用すると、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。

#### 起動・停止ボタン

定義の起動状態を設定します。

#### 監視の種類

データ転送定義を実行するタイミングを指定します。

監視ファイルの「新規作成時」または「日付更新時」を選択できます。

#### 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。

登録されているファイルリソース及び Amazon S3 のリソースを選択できます。

ファイル名にはワイルドカードや正規表現を使用できます。

#### 監視間隔

選択されたファイルを監視する間隔を指定します。下限は 1 秒です。

### 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選択します。

### 実行順序

監視ファイル名にワイルドカードや正規表現を使用した際、合致するファイルが複数存在する場合には、合致したファイルの数だけ定義が実行されます。

定義を複数回実行する場合、一回の処理毎に使用するファイルの実行順序として「名前昇順」「名前降順」「日付昇順」「日付降順」が選択できます。

### 実行後の動作

実行後に監視ファイルに対して行う処理を設定できます。

「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。

移動先フォルダとして、ファイルリソースまたは Amazon S3 のリソースを指定できます。

監視ファイル名にワイルドカードや正規表現を使用した際、合致するファイルが複数存在する場合には、合致したファイルの数だけ定義が実行されます。

この時、全ての定義が終了する前に合致するファイルが新規作成されても、ファイルトリガーで検知することはできません。

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：売上\*.csv
- ・実行後の動作：ファイルを削除する



「売上 1.csv」「売上 2.csv」「売上 3.csv」の3ファイルが一度に新規作成され、名前降順に処理されるとします。

定義が実行されて「売上 1.csv」が削除された後に、「売上 1.csv」を新規作成した場合、「売上 3.csv」が処理されてファイルトリガーの動作が終了した時点で「売上 1.csv」が残るため、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「売上 1.csv」が残り続けて、ファイルトリガーで検知されなくなってしまう。

**一度、起動状態を「停止」に変更して「売上 1.csv」を削除し、再度「起動」させてください。**



監視間隔が長すぎる場合には、監視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設定の変更は即座に反映されません。次の監視後に反映されます。



Amazon S3のリソースを選択するためには、ライセンスの登録が必要となります。ライセンスが登録されていない場合、選択画面に表示されません。

## 2.2.6.1 トリガー変数とは？

トリガー変数とは、設定したトリガーに関する値を参照できる変数で、処理開始日やファイルトリガーで検知したファイル名を処理フロー定義内で使用することが可能です。

ファイル名や固定文字フィルタ、前後処理の値として参照することができます。また、分岐での条件設定で、分岐の条件にも使用できます。

指定には、以下の記述をします。

`&{トリガー変数名}`

読み込みファイル名として

`&{fileName}`

と指定することで、読み込みファイル名に検知したファイル名を指定することができます。

参照できるトリガー変数は設定されているトリガーによって異なります。

トリガー変数と参照できるトリガーの種類は以下となります。

トリガー変数	トリガー種類	説明
menu	手動実行	実行した定義が保存されているメニュー名です
	スケジューラー	
	ファイル検知	
folder	手動実行	実行した定義が保存されているフォルダー名です
	スケジューラー	
	ファイル検知	
startDate	手動実行	フロー定義が開始された日付です
	スケジューラー	yyyy-MM-dd で表記されます
	ファイル検知	
startDateTime	手動実行	フロー定義が開始された日時です
	スケジューラー	yyyy-MM-dd HH:mm:ss.SSS で表記されます
	ファイル検知	
resourceName	ファイル検知	ファイル検知した取得リソース名です
pathName	ファイル検知	ファイル検知したファイルのパスです
fileName	ファイル検知	ファイル検知したファイル名です

## 2.2.7 フロー定義の操作

開いているフォルダ状態で、個々のフロー定義を選択すると、フロー定義に対する操作を行う事ができます。



定義一覧画面（「スケジュール」選択）

スケジュールは指定したタイミングで定義を実行します。

### [手動実行] ボタン

この定義を実行します。

トリガーが「手動実行」の場合のみ表示されます。

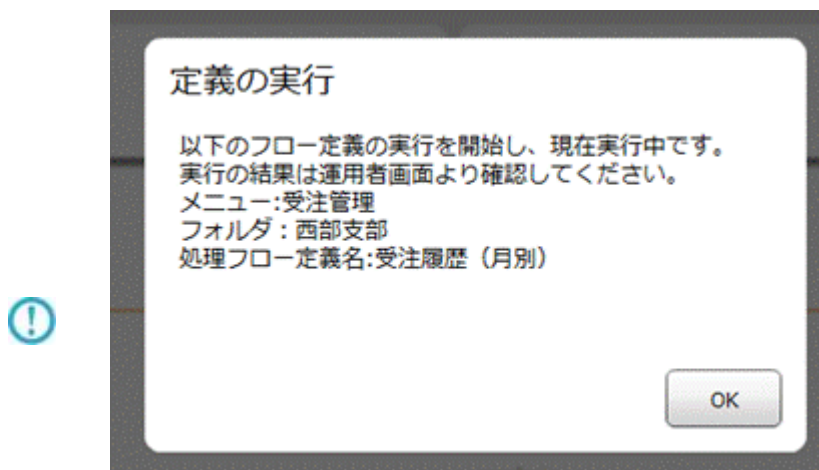
### [編集] ボタン

この定義を編集します。

### [削除] ボタン

この定義を削除します。

通常は手動実行定義を実行後に“定義の実行が正常に終了しました。”と表示されますが、処理の完了に時間がかかるフロー定義を手動実行した際、以下のダイアログが表示されることがあります。  
下記が表示された場合、処理は完了しておりませんが内部でフロー定義の実行は行われています。  
実行結果の確認については運用者画面から行ってください。



ダイアログが表示されるまでの時間を変更される場合、以下のファイル内の<value key = "REQUEST\_TIMEOUT">の値を変更する必要があります。

以下のファイル

<クライアントプログラムインストールディレクトリ>/resource/QaDef.xml

内の<value key = "REQUEST\_TIMEOUT">の値を任意の値に変更します。  
値はミリ秒で指定してください。  
デフォルトは"60000"に設定されています。

## 2.2.8 仕様書とは

各定義の仕様書を出力することができます。

フォルダにある仕様書出力のボタン（下記の赤枠）を押すと、仕様書保存画面が表示されます。



仕様書保存画面で仕様書の保存先のパスを指定し、ボタンを押すと仕様書が指定されたフォルダに出力されます。

仕様書は下記のファイル構成で出力されます。

項目名	内容
< 処理フロー定義名 > .html	仕様書の html ファイルです。
specification.css	仕様書の css ファイルです。
Flow.jpg	Flow Designer から確認できるフロー定義画面の画像です。
cXXXXXXXX.jpg	Flow Designer から確認できる各マッパーのマッパー編集画面の画像です。

### 2.2.8.1 仕様書のデザイン

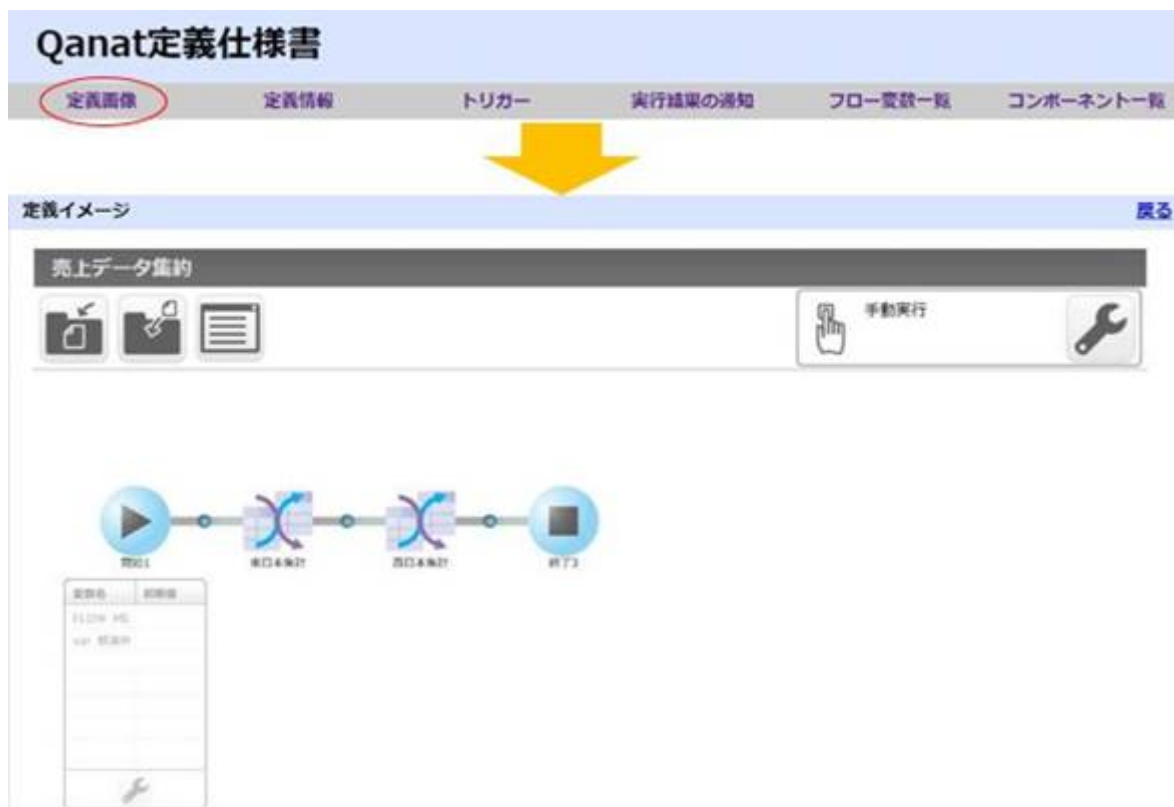
仕様書は下記のデザインで出力されます。

#### n メニュー

各設定項目のメニューが表示されます。

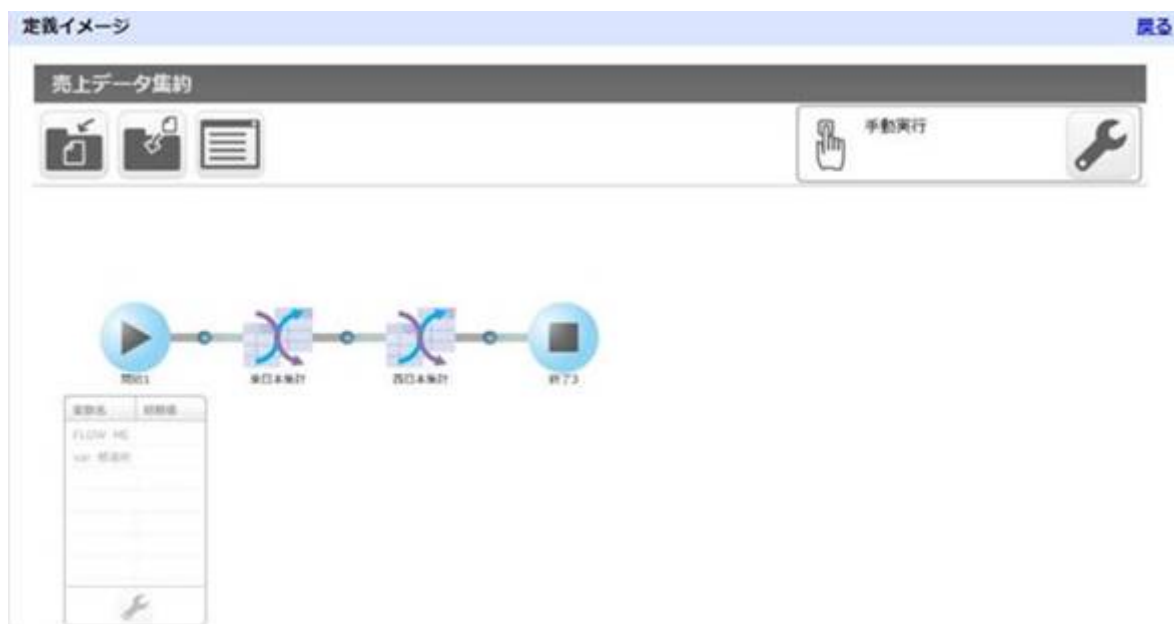
メニュー(下記の赤枠)を押すと各設定項目の表示位置へ遷移します。





#### n 定義画像

Flow Designer から確認できるフロー定義画面の画像が表示されます。戻るボタンを押すとメニューへ遷移します。



#### n 定義情報

定義の基本情報が表示されます。戻るボタンを押すとメニューへ遷移します。

定義情報		戻る
メニュー名	メニュー名	
フォルダー名	フォルダ名	
フロー定義名	定義名	
作成日時	2018-07-23 15:22:19	
作成ユーザー	cvadmin	
更新日時	2018-07-23 15:22:23	
更新ユーザー	cvadmin	
コメント	コメント	

下記の項目が表示されます。

項目名	内容
メニュー名	処理フロー定義の保存先メニュー名です。
フォルダー名	処理フロー定義の保存先フォルダー名です。
フロー定義名	処理フロー定義の定義名です。
作成日時	処理フロー定義の作成日時です。
作成ユーザー	処理フロー定義を作成したユーザー名です。
更新日時	処理フロー定義の最終更新日時です。
更新ユーザー	処理フロー定義の最終更新ユーザー名です。
コメント	設定されているコメントです。

## n トリガー

定義に設定したトリガーが表示されます。戻るボタンを押すとメニューへ遷移します。

トリガー		戻る
トリガーの種類	スケジュール	
起動状態	起動中	
スケジュール	毎週[日火木土]0時0分	

表示項目については、2.2.6 フロー定義初期設定 <トリガー「スケジュール」の設定項目> 及び <トリガー「ファイル検知」の設定項目> を参照してください。

## n 実行結果の通知

定義に設定した実行結果の通知方法が表示されます。戻るボタンを押すとメニューへ遷移します。

実行結果の通知		戻る
通知条件	成功・警告・エラー・中止	
送信先メールアドレス	qanat@jbat.co.jp	
詳細ログ	添付する	

表示項目については、2.2.6 フロー定義初期設定 <「実行結果を通知する」の設定項目> を参照してください。

## n フロー変数一覧

定義に設定したフロー変数の一覧が表示されます。戻るボタンを押すとメニューへ遷移します。

フロー変数一覧			戻る
No	変数名	初期値	
1	FLOW_MESSAGE		
2	新しい変数2	初期値	

下記の項目が表示されます。

項目名	内容
No	定義に設定したフロー変数の番号です。
変数名	定義に設定したフロー変数の名前です。
初期値	仕様書保存をした定義の定義名です。

## n コンポーネント一覧

定義に設定したコンポーネントの一覧が表示されます。

名前を押すと各コンポーネント情報へ遷移します。戻るボタンを押すとメニューへ遷移します。

コンポーネント一覧					戻る
No	名前	種類	ID	NEXT	
1	<a href="#">開始1</a>	開始	c0000001	c0000004,	
2	<a href="#">分岐開始4</a>	分岐開始	c0000004	c0000006,c0000005,	
3	<a href="#">トランザクション開始6</a>	トランザクション開始	c0000006	c0000008,	
4	<a href="#">コミット8</a>	コミット	c0000008	c0000002,	
5	<a href="#">マッパー2</a>	マッパー	c0000002	c0000009,	
6	<a href="#">ロールバック9</a>	ロールバック	c0000009	c0000007,	
7	<a href="#">トランザクション終了7</a>	トランザクション終了	c0000007	c0000005,	
8	<a href="#">分岐終了5</a>	分岐終了	c0000005	c0000003,	
9	<a href="#">終了3</a>	終了	c0000003		

下記の項目が表示されます。

項目名	内容
No	コンポーネントの番号です。
名前	コンポーネントの名前です。
種類	コンポーネントの種類です。種類には下記のものがあります。 <ul style="list-style-type: none"> <li>・開始</li> <li>・終了</li> <li>・マッパー</li> <li>・分岐開始</li> <li>・分岐終了</li> </ul>

	<ul style="list-style-type: none"> <li>・トランザクション開始</li> <li>・トランザクション終了</li> <li>・コミット</li> <li>・ロールバック</li> </ul>
ID	<p>コンポーネントの ID です。</p> <p>各コンポーネントで表示される画像名にはこの ID が利用されます。</p>
NEXT	<p>繋ぎ先のコンポーネントの ID です。カンマ区切りで複数表示されます。</p>

## n コンポーネント詳細

### l コンポーネント共通情報

コンポーネントの共通情報が表示されます。

マッパー-2 <span style="float: right;">戻る</span>	
No	5
コンポーネント名	マッパー-2
コンポーネント種類	マッパー

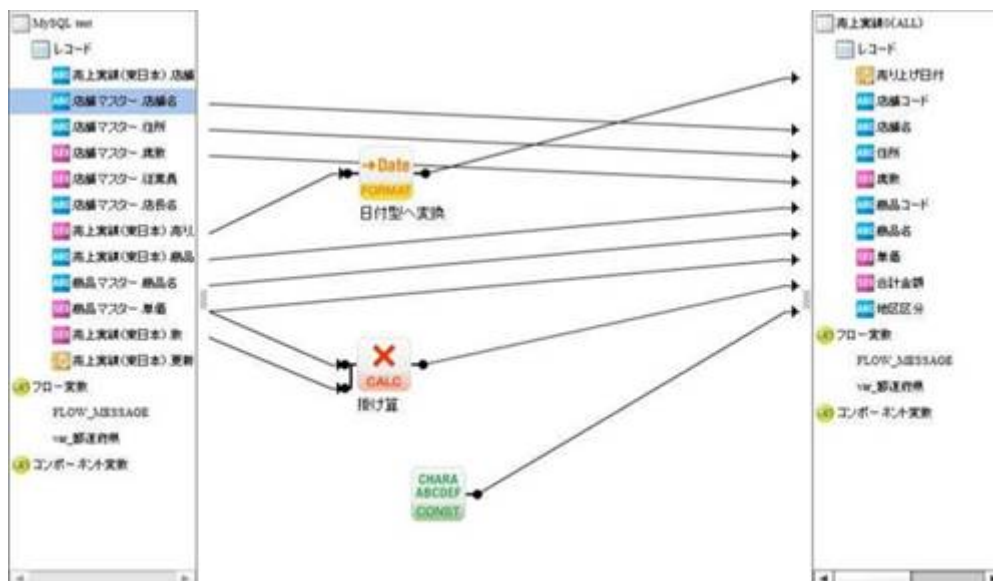
下記の項目が表示されます。

項目名	内容
No	定義に設定したコンポーネントの番号です。
コンポーネント名	定義に設定したコンポーネントの名前です。
コンポーネント種類	コンポーネントの種類です。

### l マッパー画像

Flow Designer から確認できるマッパー編集画面の画像が表示されます。

この画像は[コンポーネントの ID].jpg で保存されます。



参照しているデータリソースが存在しない場合など、正常にマッパーを開くことができない場合、マッパーの画像の生成を行うことができません。  
上記が発生した場合、仕様書の生成後、以下のメッセージが表示されます。

## ドキュメントの出力

以下のディレクトリに出力を行いました  
C:¥Users¥J31638¥Desktop¥売上データ集約

以下のマッパーのイメージ出力時にエラーが発生したため、イメージが出力されない状態で出力を行いました

- ・マッパー名：西日本集計
- ・エラー内容：リソース：test フォルダ：test ファイル：売上実績（西日本）の取得に失敗しました

閉じる

### 1 入力リソース

入力リソースに設定したアダプターの設定情報とフィールド一覧が表示されます。

入力リソースが RDB 及び kintone の場合、検索条件・並び替えを設定するとその情報も表示されます。

入力リソースが RDB の場合、テーブルのジョインを設定するとその情報も表示されます。

実行時データリソース切り替え設定が有効になっている場合、「切り替え先データリソース」「切り替え先スキーマ」が表示されます。無効になっている場合は表示されません。

## 1 出力リソース

出力リソースに設定したアダプターの設定情報とフィールド一覧が表示されます。

出力リソースが RDB の場合、結線(マッピング)を行ったフィールドのみフィールド一覧に表示されます。

実行時データリソース切り替え設定が有効になっている場合、「切り替え先データリソース」「切り替え先スキーマ」が表示されます。無効になっている場合は表示されません。

## 1 コンポーネント変数

マッパーで扱うことができるコンポーネント変数が表示されます。

### コンポーネント変数

No	変数名
1	マッパー2.result
2	マッパー2.message
3	マッパー2.readCount
4	マッパー2.writeCount
5	コミット8.result
6	コミット8.message
7	トラン開始6.result
8	トラン開始6.message
9	分岐開始4.result
10	分岐開始4.message
11	開始1.result
12	開始1.message

コンポーネント変数については、2.5.3.3 コンポーネントの種類を参照してください。

## 1 マッピング情報

マッパーに設定した結線(マッピング)情報が表示されます。

### マッピング情報

※()はDBの実フィールド名となります。

出力フィールド	入力フィールド
FLOW_MESSAGE	← No.1 固定数値
新しい変数2	← マッパー2.result
ID	← test01.ID(test01.ID)

下記の項目が表示されます。

項目名	内容
出力フィールド	出力フィールドに設定したフィールド名です。

下記の種類が表示されます。

- ・フロー変数
- ・コンポーネント変数
- ・出力リソースのフィールド

入力フィールド

入力フィールドに設定したフィールド名です。

下記の種類が表示されます。

- ・フロー変数
- ・コンポーネント変数
- ・入力リソースのフィールド
- ・フィルタ

入力フィールド又は出力フィールドに RDB のフィールドが結線(マッピング)されている場合、フィールドの実フィールド名が()内に表示されます。

入力フィールド又は出力フィールドにフロー変数が結線(マッピング)されている場合、フロー変数名を押すとフロー変数一覧へ遷移します。

#### マッピング情報

※()はDBの実フィールド名となります。

出力フィールド	入力フィールド
FLOW_MESSAGE	← No.1 固定数値
新しい変数2	← マッパー-2.result
ID	← test01.ID(test01.ID)



#### フロー変数一覧

[戻る](#)

No	変数名	初期値
1	FLOW_MESSAGE	
2	新しい変数2	初期値

入力フィールドにフィルタが結線(マッピング)されている場合、フィルタ名にカーソルが重なるとフィルタ情報が表示され、フィルタ名を押すとフィルター一覧へ遷移します。



## マッピング情報

※()はDBの実フィールド名となります。

出力フィールド	入力フィールド
FLOW_MESSAGE	← No.1 固定数値
新しい変数2	← No.1 固定数値
ID	← te

No	1
フィルタ名	固定数値
表示名	固定数値
動作説明	指定した数値を出力します。
固定出力値	123
null値出力	しない
出力先	FLOW_MESSAGE

## フィルター一覧

No	1
フィルタ名	固定数値
表示名	固定数値

## I フィルター一覧

マップーに設定した各フィルタの設定情報が表示されます。

## フィルター一覧

No	1
フィルタ名	固定数値
表示名	固定数値
動作説明	指定した数値を出力します。
固定出力値	123
null値出力	しない
出力先	FLOW_MESSAGE

下記の項目が表示されます。

項目名	内容
No	自動的に採番した番号です。
フィルタ名	フィルタの名前です。
表示名	フィルタの表示名です。
動作説明	フィルタの動作説明です。
入力元	フィルタの入力元のフィールド名です。 複数ある場合は、カンマ区切りで表示されます。
出力先	フィルタの出力先のフィールド名です。 複数ある場合は、カンマ区切りで表示されます。
フィルタ設定	上記以外はフィルタ設定です。 各フィルタ設定については、付録A フィルタリファレンスをご参照ください。

結線(マッピング)されていない又は正しくされていないフィルタについては表示されません。



## I 前処理・後処理

マッパーに設定した前処理・後処理の設定情報が表示されます。

## 前処理

サーバー	コマンド	コマンド戻り値設定	コマンド戻り値	障害発生時
Qanat2.0サーバー	echo test	成功時(他の値が戻った場合は障害)	1	スキップ

## 後処理

サーバー	コマンド	コマンド戻り値設定	コマンド戻り値	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)
Qanat2.0サーバー	echo test	成功時(他の値が戻った場合は障害)	1	行わない

表示項目については、2.6.7 前後処理をご参照ください。

サーバーに IBM サーバーを設定した場合、パスワードは暗号化されたパスワードが表示されます。

## I スクリプト

スクリプト定義が設定された場合、スクリプト内容が表示されます。

## スクリプト

内容	echo "Hello world!" echo "Hello world!" echo "Hello world!" echo "Hello world!" echo "Hello world!"
----	---

スクリプト定義については、2.7 スクリプト定義をご参照ください。

## I 分岐条件

分岐開始に設定された分岐条件が表示されます。

## 分岐条件

左項目		演算式	右項目	
項目種別	値		項目種別	値
[固定値]	0000	=	[フロー変数]	FLOW_MESSAGE

分岐条件については、2.5.3.14 分岐条件の設定をご参照ください。

## 2.3 実行結果の確認(Web)

ここでは、実行結果の確認の機能について説明します。定義の実行結果を参照できます。

定義一覧画面から移動するか、特定の URL からログインし、閲覧します。

### 2.3.1 実行結果とは

定義を手動実行した場合、トリガーから自動実行した場合の実行結果のことです。

### 2.3.2 実行結果の画面について

実行結果	実行ID	開始時間	終了時間	メニュー	フォルダ	処理フロー定義名	実行種別	ユーザー名
	85	2014-10-08 00:52	2014-10-08 00:52	受注管理	東部支部	受注集計定義手動実行テスト確認	手動実行	正塚ユカ
	84	2014-10-08 00:51	2014-10-08 00:51	受注管理	東部支部	受注集計定義手動実行テスト確認	手動実行	
	83	2014-10-08 00:51	2014-10-08 00:51	受注管理	東部支部	受注集計定義手動実行テスト確認	手動実行	
	82	2014-10-08 00:50	2014-10-08 00:50	受注管理	東部支部	受注集計定義手動実行テスト確認	手動実行	
	81	2014-10-08 00:50	2014-10-08 00:50	受注管理	東部支部	受注集計定義手動実行テスト確認	手動実行	
	80	2014-10-08 00:50	2014-10-08 00:50	受注管理	東部支部	受注集計定義手動実行テスト	手動実行	
	79	2014-10-08 00:50	2014-10-08 00:50	受注管理	東部支部	受注集計ファイル読み込み	ファイル検知	
	78	2014-10-08 00:49	2014-10-08 00:49	受注管理	東部支部	受注集計ファイル読み込み	ファイル検知	
	77	2014-10-08 00:49	2014-10-08 00:49	受注管理	東部支部	受注集計定義手動実行テスト	手動実行	
	76	2014-10-08 00:48	2014-10-08 00:48	受注管理	東部支部	受注集計ファイル読み込み	ファイル検知	
	75	2014-10-08 00:47	2014-10-08 00:47	受注管理	東部支部	受注集計定義テスト	スケジュール	
	74	2014-10-08 00:46	2014-10-08 00:46	受注管理	東部支部	受注集計定義手動実行テスト	手動実行	
	73	2014-10-08 00:46	2014-10-08 00:46	受注管理	東部支部	受注集計定義手動実行テスト	手動実行	
	72	2014-10-08 00:46	2014-10-08 00:46	受注管理	東部支部	受注集計定義手動実行テスト確認	手動実行	
	71	2014-10-08 00:46	2014-10-08 00:46	受注管理	東部支部	受注集計定義手動実行テスト確認	手動実行	
	70	2014-10-08 00:46	2014-10-08 00:46	受注管理	東部支部	受注集計定義手動実行テスト	手動実行	
	69	2014-10-08 00:46	2014-10-08 00:46	受注管理	東部支部	受注集計定義手動実行テスト確認	手動実行	
	68	2014-10-08 00:46	2014-10-08 00:46	受注管理	東部支部	受注集計定義手動実行テスト	手動実行	
	67	2014-10-08 00:46	2014-10-08 00:46	受注管理	東部支部	受注集計定義手動実行テスト確認	手動実行	
	66	2014-10-08 00:46	2014-10-08 00:46	受注管理	東部支部	受注集計定義手動実行テスト	手動実行	
	65	2014-10-08 00:45	2014-10-08 00:45	受注管理	東部支部	受注集計定義手動実行テスト	手動実行	
	64	2014-10-08 00:45	2014-10-08 00:45	受注管理	東部支部	受注集計定義手動実行テスト確認	手動実行	

実行結果の確認



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。



## 2.3.2.1 ログの詳細情報の確認

受注集計定義テスト 閉じる

実行種別 手動実行 実行ユーザー cvadmin

メニュー 受注管理

フォルダー 東部支部

データリソース切り替え設定 無効

実行結果	コンポーネントID	コンポーネント名	読込レコード数	出力レコード数	開始時間	終了時間	メッセージ
	c0000002	川崎拠点集計	42	42	2019-03-11 15:55:36	2019-03-11 15:55:36	
	c0000005	トク開始6	0	0	2019-03-11 15:55:36	2019-03-11 15:55:36	
	c0000004	横浜支店集計	1	0	2019-03-11 15:55:36	2019-03-11 15:55:36	1レコード目 エラー発生
	c0000010	0-MA*978	0	0	2019-03-11 15:55:36	2019-03-11 15:55:36	
	c0000006	トク終了7	0	0	2019-03-11 15:55:36	2019-03-11 15:55:36	
	c0000011	名古屋支店集計	42	42	2019-03-11 15:55:36	2019-03-11 15:55:36	




1 ページ中 1 ページ目 6 件中 1 - 6 を表示

実行結果の画面一覧にあるログをクリックすることで、より詳細情報を確認することができます。

実行種別	実行トリガーの種類です
実行ユーザー	定義を実行したユーザーです。
メニュー	実行した定義の格納メニューです
フォルダー	実行した定義の格納フォルダーです
データリソース切り替え設定	実行時データリソース切り替え設定の有無を表示します
実行後の動作	ファイル検知定義で検知したファイルに対する振る舞いです。

### コンポーネント一覧

実行結果

-  コンポーネントが正常に実行された場合に表示されるアイコンです。
-  実行結果一覧に表示されるログのステータスが警告の場合、エラーのコンポーネントに表示されるアイコンです。
-  実行結果一覧に表示されるログのステータスが失敗の場合、エラーのコンポーネントに表示されるアイコンです。

コンポーネント ID 実行したフロー定義内の各コンポーネントに振られる ID です。

コンポーネント名	実行したフロー定義内のコンポーネント名です。
読込レコード	コンポーネントの入力側リソースが読み込んだレコード数です。
出力レコード	コンポーネントの出力側リソースが出力したレコード数です。
開始日時	コンポーネント定義の実行が開始された日時です
終了日時	コンポーネント定義の実行が終了された日時です
メッセージ	エラー発生時に表示されるメッセージです。

スクリプト定義実行時の“出力レコード”はログの詳細情報には表示されません。

すべて “ 1 ” と表示されます。



出力レコード数
1

## 2.3.3 メニュー



[ 絞り込み ] ボタン	絞り込み検索を行うための、絞り込み条件ダイアログを開きます ボタンの右に現在の設定内容が表示されます
[ 一覧更新 ] ボタン	現在の表示内容を更新します
[ 定義一覧へ ] ボタン	定義一覧画面を表示します
[ 設定 ] ボタン	ログフォント色の設定画面を表示します
[ ヘルプ ] ボタン	ヘルプ画面を表示します
[ ログアウト ] ボタン	ログアウトし、画面を閉じます

## 2.3.4 絞り込み条件

条件を入力し、[閉じる]ボタンをクリックすると、絞り込み条件として有効になります。

実行結果	実行結果を絞り込み条件として設定します。設定できる条件はすべて/正常/警告/失敗のいずれかになります
日付指定	開始日時の範囲を絞り込み条件として設定します
検索	「実行 ID」「開始日時」「終了日時」「メニュー」「フォルダー」「処理フロー定義名」「実行種別」のいずれかの内容に対して、部分一致による絞り込みを設定します
[クリア] ボタン	絞り込み条件をすべて消去します
[閉じる] ボタン	ダイアログを閉じ、現在の設定を有効にします。

## 2.3.5 ログフォント色設定

実行結果一覧の項目にあるユーザー変数の値によって、ログのフォント色を変更することができます。



ログフォント色設定画面

追加：任意のユーザー変数を入力し、そのユーザー変数を持つログのフォント色を選択します。

編集：一覧で選択しているログフォント色設定を変更できます。

削除：一覧で選択しているログフォント色設定を削除できます。

実行結果	実行ID	開始時間	終了時間	ユーザー	フォルダ	処理フロー定義名	実行種別	ユーザー変数
成功	85	2014-10-08 00:52:21	2014-10-08 00:52:21	受注管理	業務実行部	受注集計定額手動実行テスト確認	手動実行	正常出力
成功	84	2014-10-08 00:51:07	2014-10-08 00:51:07	受注管理	業務実行部	受注集計定額手動実行テスト確認	手動実行	
成功	83	2014-10-08 00:51:00	2014-10-08 00:51:00	受注管理	業務実行部	受注集計定額手動実行テスト確認	手動実行	

ログフォント色を青色に変更した例



## 2.3.6 ヘルプ画面

[ヘルプ]ボタンをクリックするとログ監視の設定、製品のバージョンを確認することができます。



[ログ監視確認]ボタン	ログ自動削除の設定を表示します
[Qanat について]ボタン	製品のバージョンを表示します
[閉じる]ボタン	実行結果の確認画面に戻ります



製品バージョンの表示画面

## 2.4 フロー定義一覧 (Web)

### 2.4.1 フロー定義一覧画面について

ここでは、フロー定義一覧画面（運用者）の機能について説明します。

定義済みのフロー定義の起動状況を参照できます。起動状態の変更、および、「実行結果を通知する」機能の設定を変更できます。

メニュー	フォルダ	トリガー	処理フロー定義	作成日付	更新日付
受注管理	東部支部		受注集計定義テスト スケジュール 毎年3月12日 00時00分 コメント	2019-03-11 15:47:30	2019-03-12 17:11:31
Menu	FileTrigger		Test コメント	2019-03-11 14:30:07	2019-03-11 16:14:08
Menu	FileTrigger		03_商品 監視用フォルダ SMILE_kintone_files / 03_商品 / *.csv コメント	2019-03-06 15:10:44	2019-03-12 17:11:37
Menu	FileTrigger		01_売上 監視用フォルダ SMILE_kintone_files / 01_売上 / *.csv コメント	2019-03-06 15:10:43	2019-03-12 17:11:45
Menu	FileTrigger		00_ポータル 監視用フォルダ SMILE_kintone_files / 00_ポータル / *.csv コメント	2019-03-06 15:10:42	2019-03-07 17:56:42

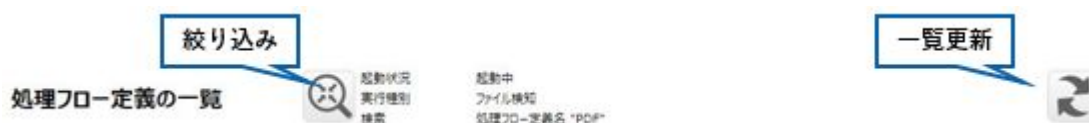
フロー定義一覧



すべての定義済み内容が一覧で表示されます。

メニュー	定義が含まれるメニューの名称です
フォルダ	定義が含まれるフォルダの名称です
[トリガー]ボタン	作成時等で設定したトリガーの設定内容の要約です
フロー定義	フロー定義名、トリガーの設定内容の要約、コメント情報です
作成日付	定義を作成した日時です
更新日付	定義を更新した日時です
起動中	起動中の定義は点滅します

## 2.4.2 メニュー



- |            |   |
|------------|---|
| [絞り込み] ボタン | 絞り込み検索を行うための、絞り込み条件ダイアログを開きます<br>ボタンの右に現在の設定内容が表示されます |
| [一覧更新] ボタン | 現在の表示内容を更新します   |

## 2.4.3 絞り込み条件

絞り込み条件 クリア

起動状況

実行種別

検索

条件を入力し、[閉じる]ボタンをクリックすると、絞り込み条件として有効になります。

- |           |   |
|-----------|---|
| 起動条件      | 起動状態を絞り込み条件として設定します。設定できる条件はすべて/起動中/停止中のいずれかとなります。            |
| 起動種別      | 起動種別を絞り込み条件として設定します。設定できる条件はすべて/手動実行/スケジュール/ファイル検知のいずれかとなります。 |
| 検索        | 「メニュー」「フォルダー」「処理フロー定義名」のいずれかの内容に対して、部分一致による絞り込みを設定します         |
| [クリア] ボタン | 絞り込み条件をすべて消去します   |
| [閉じる] ボタン | ダイアログを閉じ、現在の設定を有効にします。  |

## 2.4.4 設定変更

処理フロー定義の一覧


起動状況  
実行種別  
検索
すべて  
すべて


メニュー	フォルダ	トリガー	処理フロー定義	作成日時	更新日時
受注管理	東部支部		受注集計定義テスト スケジュール 毎年 3月 12日 00時 00分 コメント	2019-03-11 15:47:30	2019-03-12 17:11:31
Menu	FileTrigger		Test コメント	2019-03-11 14:30:07	2019-03-11 16:14:08
Menu	FileTrigger		03_商品 監視先フォルダ SMILE_kintone_Files / 03_商品 / *.csv コメント	2019-03-06 15:10:44	2019-03-12 17:11:37
Menu	FileTrigger		01_売上 監視先フォルダ SMILE_kintone_Files / 01_売上 / *.csv コメント	2019-03-06 15:10:43	2019-03-12 17:11:45

## フロー定義一覧

トリガーをマウスで選択すると、上記の赤枠のようにボタンが変化します。

クリックすると、トリガーの種類によって設定ダイアログが表示されます。

## &lt; 共通の設定項目 &gt;

## [ 閉じる ] ボタン

設定を終了し、一覧画面に戻ります

## 実行結果を通知する

「実行結果を通知する」機能を使用するかについて指定します

チェックすると、設定項目が追加されます

## ・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

## &lt;トリガー「手動」の設定項目&gt;

- [ 実行 ] ボタン      定義を実行します  
 定義名              対象の定義の定義名です

## &lt;トリガー「スケジュール」の設定項目&gt;

- 起動/停止ボタン      定義の起動状態を設定します  
 起動状況              起動状態を表示します  
 タイプ選択              スケジュールのタイプを設定します。

毎年 ... 毎年、任意の日時に実行します

月日指定  

時刻指定  

毎月 ... 毎月、任意の日時に実行します

日指定  

時刻指定  

毎週 ... 毎週、任意の曜日の任意の時間に実行します

曜日指定 日 月 火 水 木 金 土

時刻指定  

毎日 ... 毎日、任意の時間に実行します

時刻指定  

毎分 ... 特定の間隔（分単位）で実行します

分指定  分ごとに実行

1回 ... 任意の日時に一回だけ実行します

日付指定  

時刻指定

<トリガー「スケジュール」の設定項目>

起動/停止ボタン

ファイル検知の設定変更

起動状況     停止中

実行結果を通知する

起動/停止ボタン     定義の起動状態を設定します

起動状況     起動状態を表示します

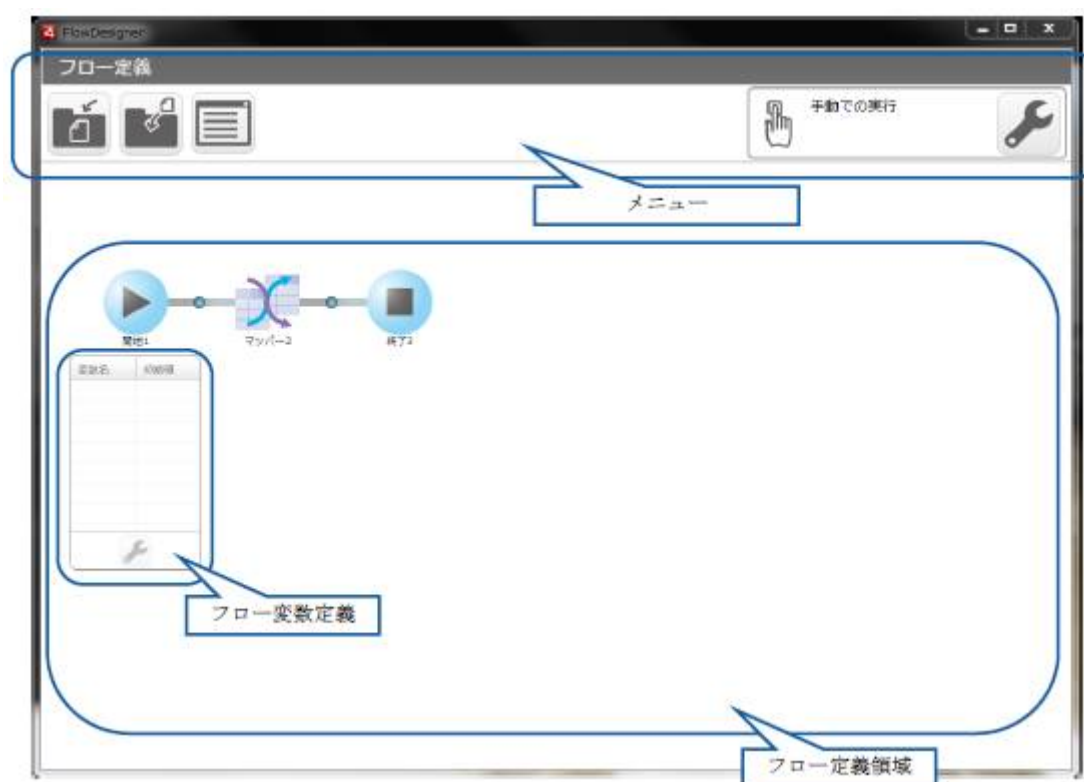


## 2.5 フロー定義画面 (Flow Designer)

ここでは、定義者機能/Flow Designer のログイン後に表示される、フロー定義一覧画面の機能について説明します。

フロー定義ではマッパー定義、トランザクション等の動作を任意の順番で実行するように定義することができます。

### 2.5.1 フロー定義画面



メニュー	様々な操作を行うメニューの領域です
フロー定義領域	実際のフローの定義が表示される領域です
フロー変数定義	フロー変数の参照や定義を行う領域です

## 2.5.2 メニュー



フロー定義名	編集中のフロー定義名です
[名前を付けて保存] ボタン	編集中のフロー定義を別名で保存します
[定義一覧画面へ移動] ボタン	定義一覧画面を Web ブラウザーで表示します
トリガー要約	トリガー内容の要約です
[一般設定] ボタン	トリガー等の設定画面を開きます

### 2.5.2.1 名前をつけて保存の概要

The dialog box '名前を付けて保存' contains the following fields and buttons:

- メニュー**: Dropdown menu with 'Menu01' selected.
- フォルダ**: Dropdown menu with 'Folder01' selected.
- 処理フロー定義名**: Text input field containing 'フロー定義'.
- コメント**: Large empty text area for comments.
- OK** and **Cancel** buttons at the bottom right.

メニュー	保存先のメニュー名です
フォルダ	保存先のフォルダ名です
処理フロー定義名	保存時の処理フロー定義名です
コメント	保存時のコメント情報です
[ OK ] ボタン	設定内容を保存して終了します
[ Cancel ] ボタン	設定内容を保存せずに終了します

実行時データリソース切り替えを有効にしている場合、保存時に以下の警告メッセージが表示され、保存することができない場合があります。  
表示された場合、一般設定画面よりリソース切り替え機能を無効にするか、リソース切り替え設定の画面から、存在するリソースへ切り替えた後、保存を実行してください。



#### 切り替えるリソースの確認

切り替え先のリソースに存在しないデータリソースまたはスキーマが指定されています。このままフロー定義を実行した場合、正常に動作しない可能性があります。そのため、一般設定画面よりリソース切り替え機能を無効にする、もしくはリソース切り替え設定の画面から、存在するリソースへ切り替えてください。

OK

## 2.5.2.2 一般設定の概要

一般設定の変更

メニュー  フォルダ:

処理フロー定義名

手動実行  スケジュール  ファイル検知

実行時データリソース切り替え設定を有効にする

実行結果を通知する

2.2.6 フロー定義初期設定 と同様の画面が表示されますが、以下の点が異なります。

- ・メニュー、フォルダの変更を行うことはできません。
- ・実行時データリソース切り替え設定の項目が表示されます。

### 2.5.2.2.1 実行時データリソース切り替え設定

実行時データリソース切り替え設定を有効とすることで、フロー定義実行時、マッパーで参照しているデータリソース/スキーマを任意に切り替えて実行を行います。



実行時に切り替えを行う設定であり、フロー定義が参照しているデータリソース/スキーマを変更する機能ではありません。  
そのため、フロー定義は変更前データリソース/スキーマが指定されている状態となります。



切り替えの対象とすることができるデータリソースはデータベースリソース、Salesforceリソースとなります。



設定時、データベースリソースのデータベース種別は判断しておりません。  
異なるデータベース種別で切り替えの設定を行った場合、実行時エラーとなります。

実行時データリソース切り替え設定はプロパティ「実行時データリソース切り替え設定を有効にする」にチェックを入れた場合に有効となります。

初期値は「チェックなし」に設定されています。

一般設定の変更

メニュー: 販売 フォルダ: kintone連携

処理フロー定義名: test

手動実行  スケジュール  ファイル検知

実行時データリソース切り替え設定を有効にする  実行結果を通知する

設定 OK Cancel



後述のデータリソースの切り替え設定を行っても、プロパティ「実行時データリソース切り替え設定を有効にする」が「チェックなし」に設定されている場合、本機能は動作しません。

データリソース/スキーマの切り替え設定は[設定]ボタンをクリックし、表示される[実行時データリソース切り替え設定]画面より行います。

実行時データリソース切り替え設定

種類	変更前データリソース	変更前スキーマ	変更後データリソース	変更後スキーマ
データベース	DB2_01_TEST	DB2ADMIN	DB2 01 TEST	DB2ADMIN
salesforce	SalesforceTest	SalesforceTest01	SalesforceTest	SalesforceTest01
データベース	MYSQL01	test	MYSQL01	test

OK Cancel

- ・ 種類 データリソースの種別を表示します。  
任意に変更することはできません。
- ・ 変更前データリソース フロー定義で参照されているデータリソースを表示します。  
任意に変更することはできません。
- ・ 変更前スキーマ フロー定義で参照されるスキーマを表示します。  
任意に変更することはできません。
- ・ 変更後データリソース 変更前データリソースに対して切り替えるデータリソースを選択します。  
初期値は変更前データリソースと同一となっています。
- ・ 変更後スキーマ 変更前スキーマに対して切り替えるスキーマを選択します。  
初期値は変更前スキーマと同一となっています。

表示されている変更前データリソース/スキーマに対して、切り替えるデータリソース/スキーマを指定します。

## 実行時データリソース切り替え設定

種類	変更前データリソース	変更前スキーマ	変更後データリソース	変更後スキーマ
データベース	DB2_01_TEST	DB2ADMIN	DB2_01_RELEASE	DB2ADMIN
salesforce	SalesforceTest	SalesforceTest01	SalesforceRelease	SalesforceRelease01
データベース	MYSQL01	test	MYSQL01	release

OK

Cancel

データリソース/スキーマの切り替え設定後、設定したデータリソースを削除、またはデータリソース/スキーマ名の変更をした場合、以下の警告が表示され、存在しないデータリソース/スキーマが赤文字で表示されます。

- ・存在しないリソースまたはスキーマが選択されています。

設定時に存在するデータリソース/スキーマ名を選択後、[OK]ボタンをクリックしてください。

エラーが出た状態でフロー定義の実行を行った場合、実行時にデータリソースが存在しないため実行エラーとなります。

## 実行時データリソース切り替え設定

種類	変更前データリソース	変更前スキーマ	変更後データリソース	変更後スキーマ
データベース	DB2_01_TEST	DB2ADMIN	DB2_01_RELEASE	
salesforce	SalesforceTest	SalesforceTest01	SalesforceRelease	SalesforceRelease01
データベース	MYSQL01	test	MYSQL01	release

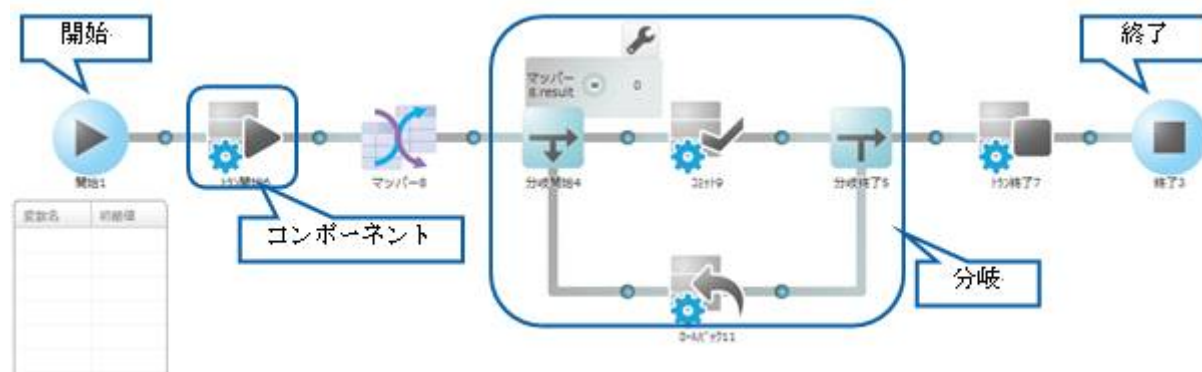
存在しないリソースまたはスキーマが選択されています

OK

Cancel



## 2.5.3 フロー定義領域の概要

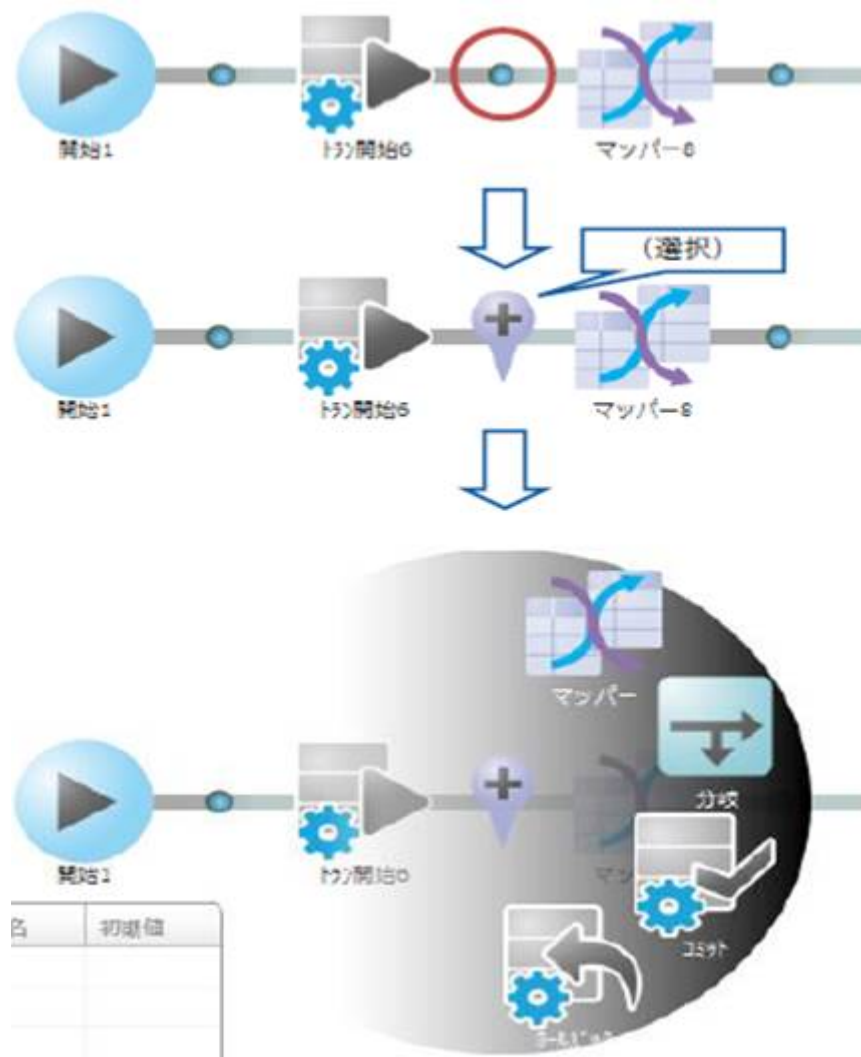


開始	フロー定義の開始地点です ここから右側に実行が進みます
終了	フロー定義の終了地点です ここまで来るとフロー定義の実行は終了となります
コンポーネント	フロー定義で実行される内容を示します
分岐	「分岐」コンポーネントを使用すると、処理の流れが分岐します

### 2.5.3.1 コンポーネントの追加

コンポーネントは、下記の赤枠内のアイコンをクリックすると表示されるパレットから行います。

パレットの内容は、押される位置によって適切なものが表示されますので、選択してください。

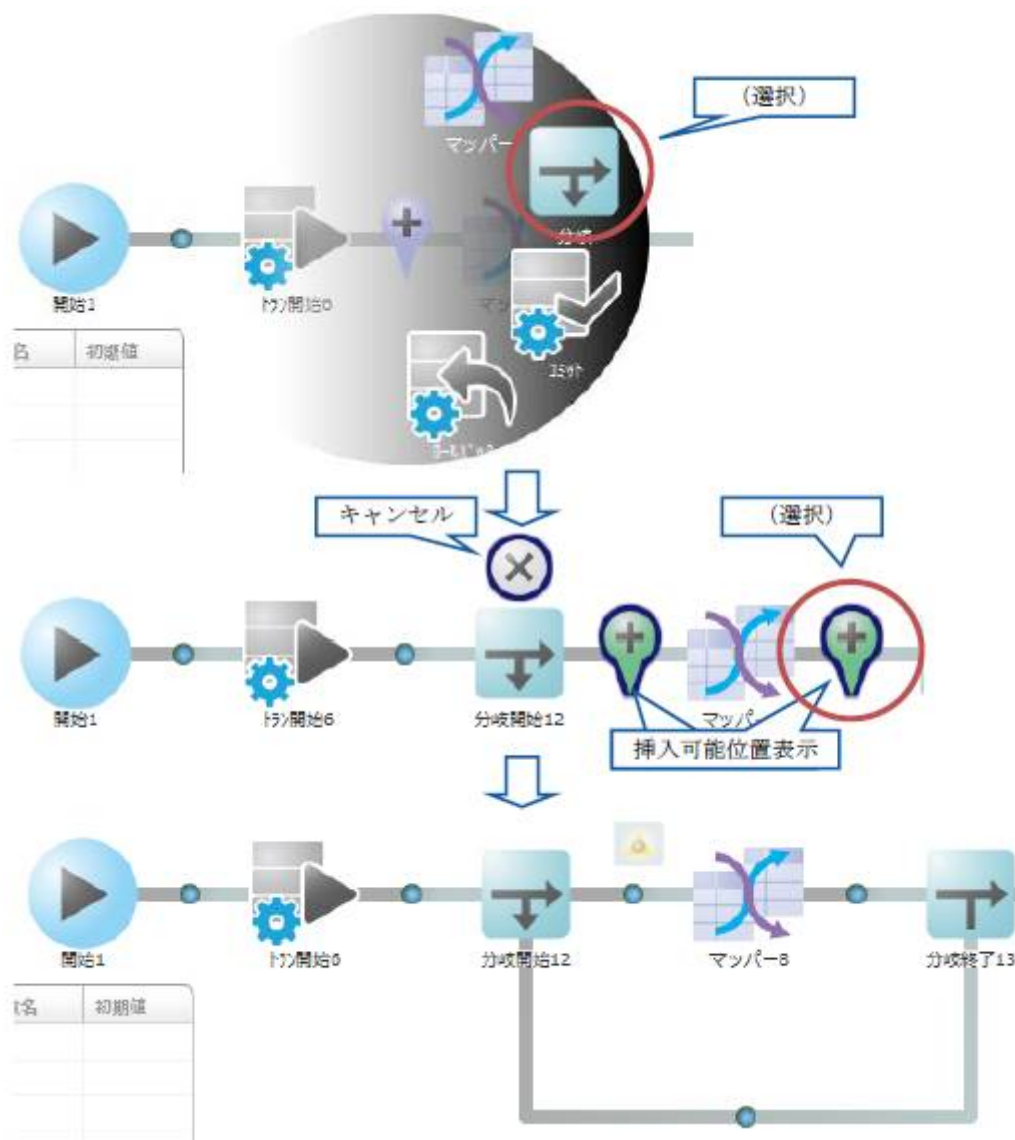




### 2.5.3.2 範囲コンポーネントの追加

「分岐」「トランザクション」のコンポーネントは、必ず開始と終了の二つのコンポーネントをセットで追加します。




そのため、1つ目(開始)のコンポーネントを追加した後、2つ目のコンポーネントの追加操作に移行します。



一つ目のコンポーネントの追加後に、[キャンセル]を押すと、一つ目のコンポーネントも削除されます。

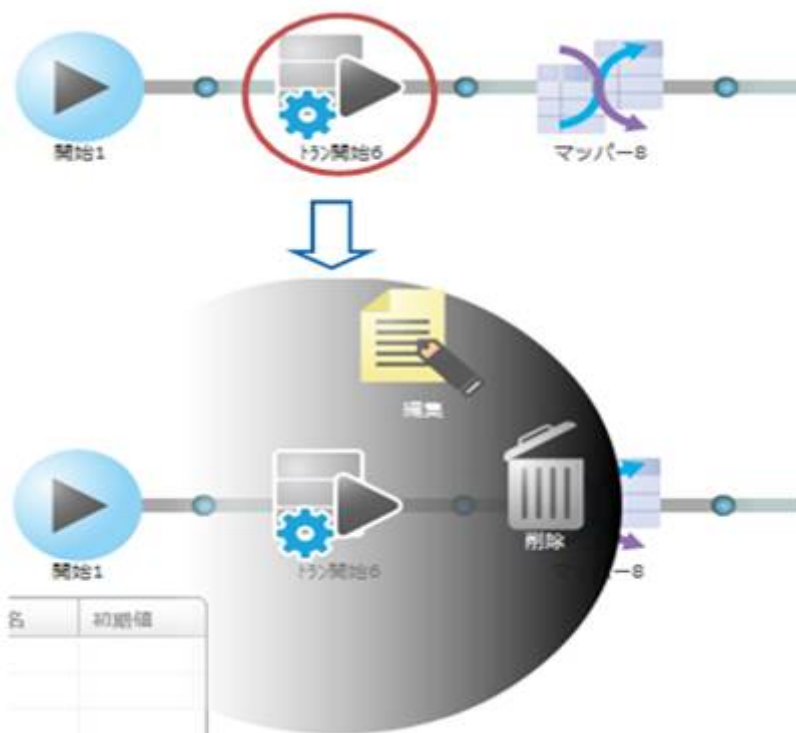
## 2.5.3.3 コンポーネントの種類

表示	名称	操作	変数	説明
	開始	編集	result message	フローの開始地点です 追加・削除することはできません
	終了	編集	result message	フローの終了地点です 追加・削除することはできません
	マッパー	編集 コピー 削除	result message readCount writeCount	データの転送等の定義を行います
	分岐開始	編集 分岐追加 削除	result message	フローの分岐の開始地点です 必ず、「分岐終了」と一緒に追加されます。分岐条件を設定することが可能です。
	追加分岐	削除	result message	追加されたフロー分岐です 追加操作は「分岐開始」から行います。分岐条件を設定することが可能です。
	分岐終了	編集	result message	フロー分岐の終了地点です 「分岐開始」と連動して追加・削除がおこなわれます。
	トランザクション開始	編集 削除	result message	トランザクションの開始地点です 必ず、「トランザクション終了」と一緒に追加されます。トランザクションとは処理の単位で、この間に行われた処理を特定のタイミングで、コミット（確定）したりロールバック（復帰/キャンセル）したりすることができます。  トランザクションは開始・終了に含まれたマッパーのうち、DBの書き込み処理に対して有効です。

	コミット	編集 削除	result message	トランザクションをコミット（確定）します  トランザクションの開始・終了の範囲でデータベースへの転送内容を確定させたい時などに使用します。トランザクション範囲にのみ挿入可能です
	ロール バック	編集 削除	result message	トランザクションをロールバック（復帰）します  トランザクションの開始・終了の範囲でデータベースへの転送内容を取りやめて元に戻したい時などに使用します。トランザクション範囲にのみ挿入可能です
	トランザ クシ ョン 終了	編集	result message	トランザクションの終了地点です  「トランザクション開始」と連動して追加・削除がおこなわれます。なお、トランザクションの終了では、トランザクションの開始・終了の範囲に、「コミット」・「ロールバック」があるかどうかによらず、コミットが行われます。





### 2.5.3.4 コンポーネントの操作

コンポーネントの操作は、コンポーネントをクリックすると表示されるパレットから行います。パレットの内容は、押される位置によって適切なものが表示されます。



コンポーネントが条件等で使用されている場合、そのコンポーネントは削除できません

## 2.5.3.5 コンポーネントの操作の種類

表示	名称	説明
	編集	コンポーネントを編集します マッパーの時のみ画面遷移を行い、その他のコンポーネントでは設定のダイアログが表示されます
	コピー	コンポーネントのコピーを行います。 「マッパー」コンポーネントでのみ表示されます。
	分岐追加	分岐範囲に分岐を追加します 「分岐開始」コンポーネントでのみ表示されます
	削除	コンポーネントの削除を行います。 <ul style="list-style-type: none"> <li>・コンポーネントの変数が利用されているなど、コンポーネントが削除できない場合、半透明の表示になります。</li> <li>・範囲コンポーネントの場合、選択したコンポーネント以外のものが同時に削除される場合があります。</li> </ul> 「分岐開始」：選択した分岐開始に対応する分岐終了と、その範囲に含まれるコンポーネントがすべて削除されます 「分岐追加」：その分岐に含まれるすべてのコンポーネントが削除されます 「トランザクション」：選択したトランザクション開始に対応するトランザクション終了と、その範囲に含まれるすべてのコミットとトランザクションがすべて削除されます。

### 2.5.3.6 コンポーネントの編集

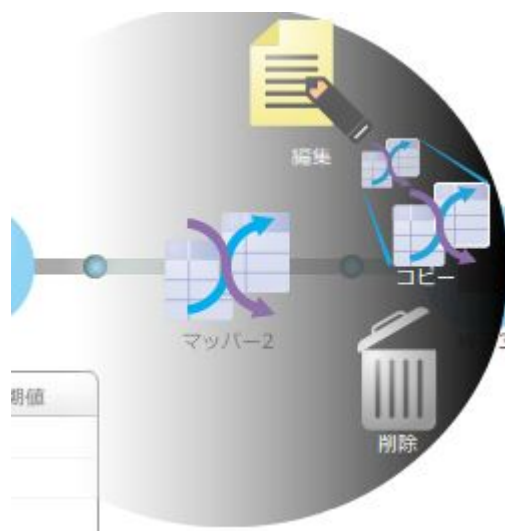
マッパーの編集時の操作については「2.6 マッパー編集画面」を参照ください。ここでは、その他のコンポーネントの編集について記載します。



- |            |   |
|------------|---|
| タイトル       | コンポーネントの下に表示されるタイトルを設定します。最大24文字まで入力可能です。 |
| [保存]ボタン    | 設定内容を保存して終了します。                           |
| [キャンセル]ボタン | 設定内容を保存せずに終了します。                          |

### 2.5.3.7 マッパーのコピー

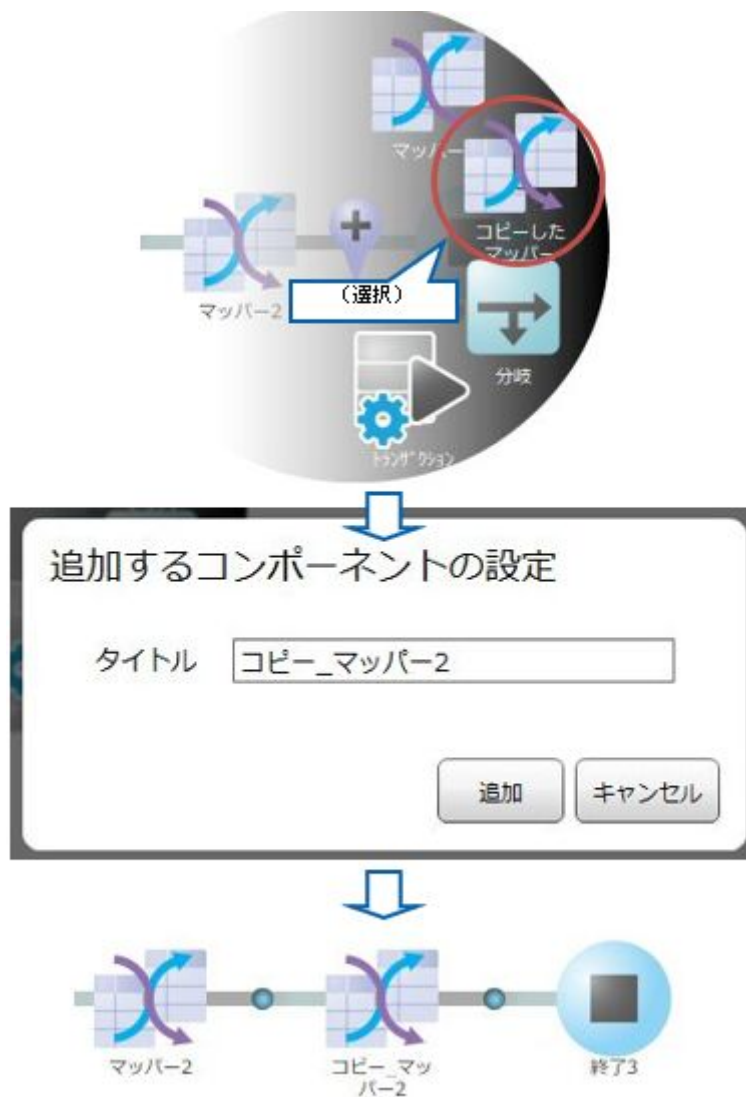
「マッパー」でのみ「コピー」を使用可能です。選択しているマッパーをコピーすることができます。コピーしたマッパーは任意の箇所にコンポーネントとして追加することが可能です。コピーしたマッパーは Flow Designer を停止するまで保持されます。



- ① 保持されるマッパーの数は1つとなります。再度コピーを行った場合、その時点で保持しているマッパーを破棄してコピーを行います。
- ① コピー後、マッパーを編集した場合、その変更内容は保持しているマッパーに影響しません。編集内容を反映させる場合、再度コピーを行ってください。
- ① マッパーのタイトルは「コピー\_<マッパーのタイトル>」でコピーが行われます。タイトルの上限值を超えた場合、マッパーのタイトルが末尾から削除された状態で保持されます。

### 2.5.3.8 コピーしたマッパーの追加

「コピー」を実行した後に使用可能です。コピーしたマッパーをコンポーネントとして追加します。コピーしたマッパーのメニュー、フォルダに関係なく、任意のフロー定義に対して行うことができます。



同名のコンポーネントが存在する場合、警告メッセージが表示され[追加]ボタンがグレーアウトします。

「タイトル」プロパティの値を変更し、[追加]ボタンをクリックしてください。



### 追加するコンポーネントの設定

タイトル

同名のコンポーネントが存在しています

結線をしているフロー変数が存在しない、またはコンポーネント変数が参照できない場合、追加後に削除した結線の内容が表示されます。

内容を確認し、必要に応じて追加したマップパーの編集を行ってください。



### 以下の結線が削除されました

フロー変数、コンポーネント変数を確認し、再度結線を行ってください

- ・ フロー変数:フロー変数B >> フロー変数:フロー変数C
- ・ フロー変数:フロー変数E >> フロー変数:フロー変数F
- ・ フロー変数:フロー変数A >> フィールド:サンプル出力フィールド1
- ・ フロー変数:フロー変数D >> フィルター:文字列を置換する処理
- ・ フィールド:支店名 >> フロー変数:フロー変数D
- ・ フィルター:全角変換の処理 >> フロー変数:フロー変数G
- ・ コンポーネント変数:マップパー2.readCount >> フロー変数:フロー変数H

表示される内容は以下のフォーマットになっています。

<入力の種類> : <入力元> >> <出力の種類> : <出力先>

<入力の種類> はフィールド、フロー変数、コンポーネント変数、フィルタのいずれかになります。

<入力元> はフィールド名、フロー変数名、コンポーネント変数名、フィルタ名のいずれかになります。

<出力の種類> はフィールド、フロー変数、フィルタのいずれかになります。

<出力先> はフィールド名、フロー変数名、フィルタ名のいずれかになります。



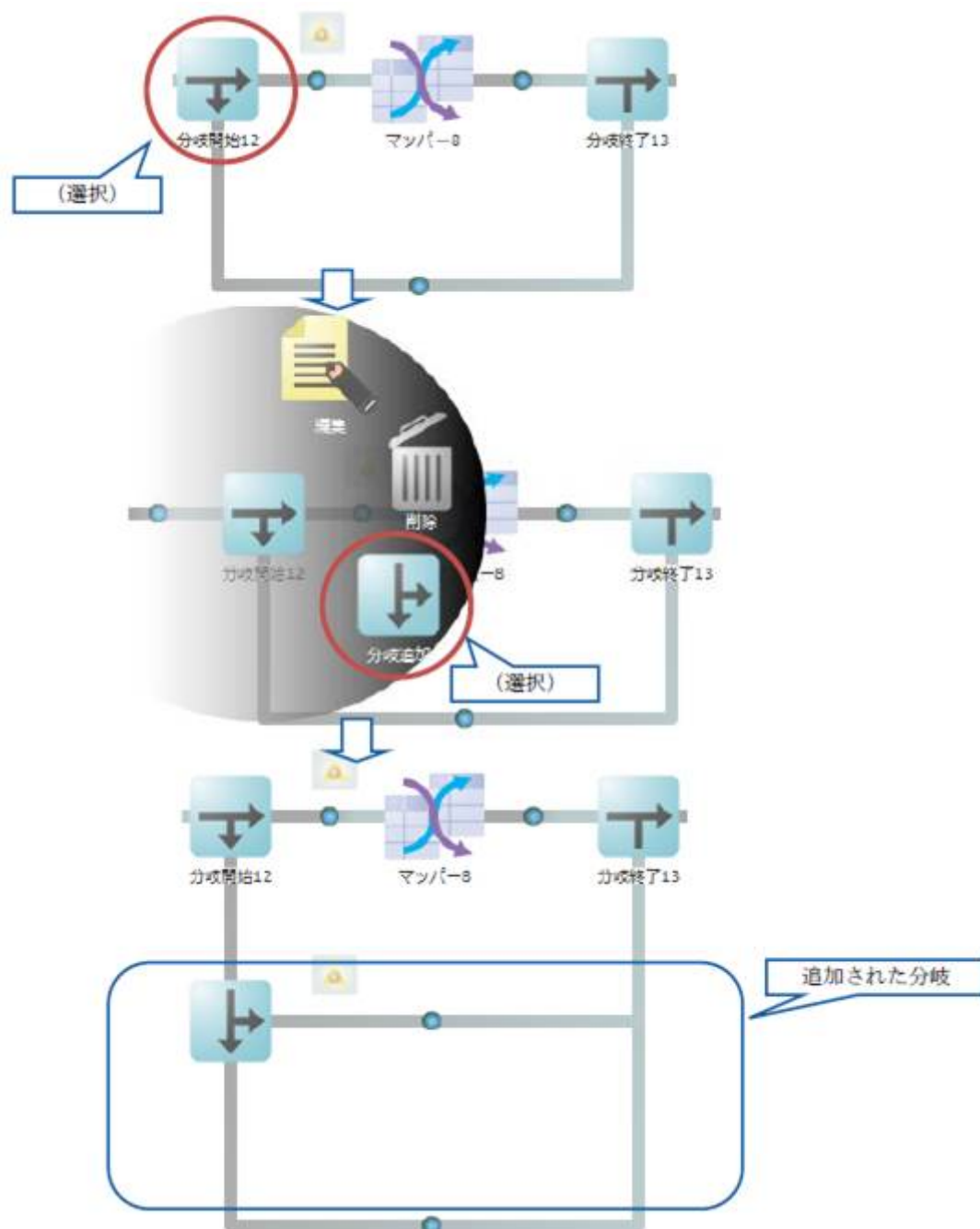
&{}表記によるトリガー変数、\${}表記によるフロー変数はそのままマップパーの追加を行います。

追加したマップパーに参照できない変数の参照を行っている箇所がないか確認をしてください。



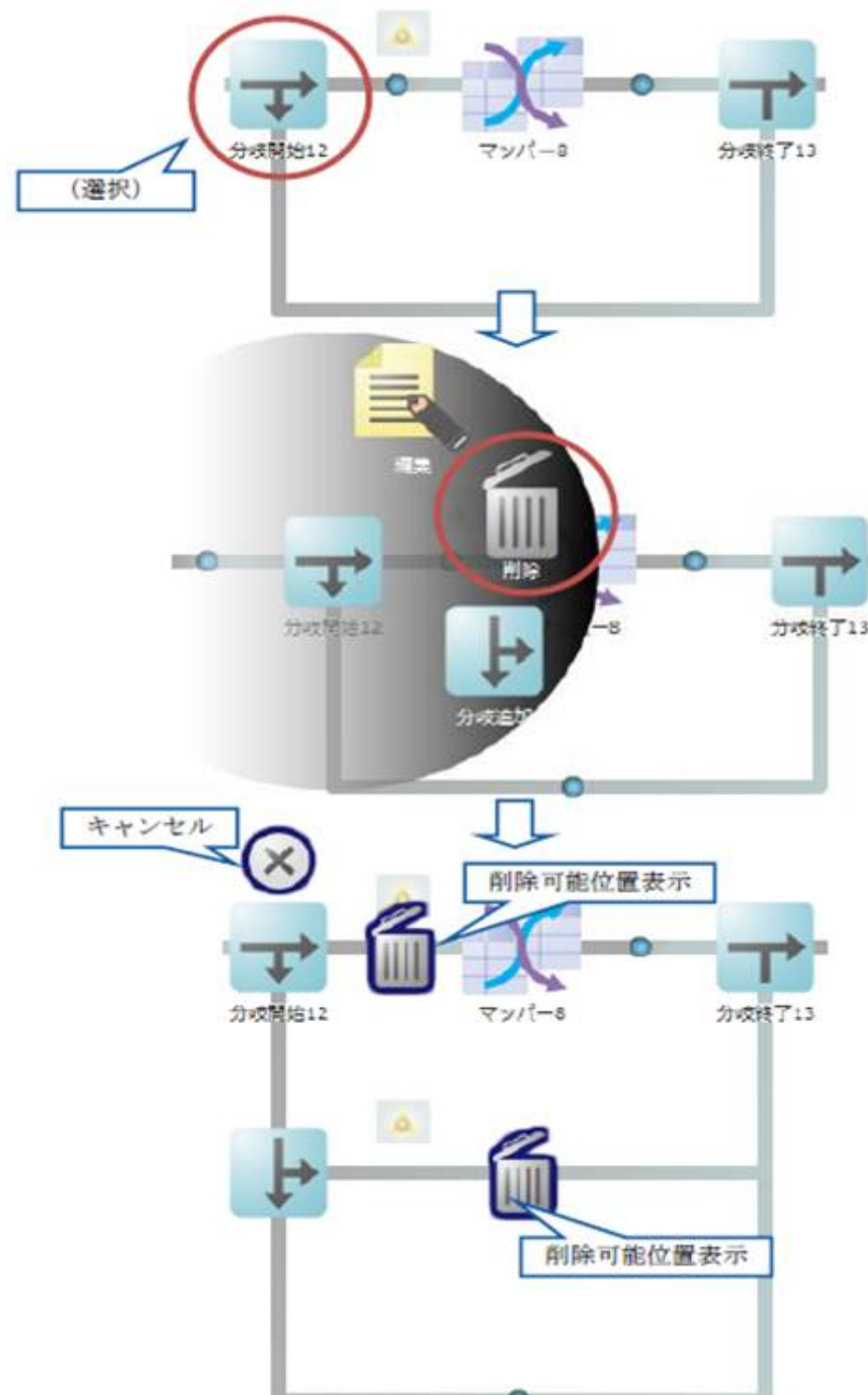
### 2.5.3.9 分岐の追加

「分岐開始」でのみ「分岐追加」を使用可能です。これを選択することでフローの分岐を追加できます。分岐は最大5つ(条件設定可能な分岐は4つ)まで設定可能で、上から順に確認し最初に条件が一致した分岐で処理が進みます。



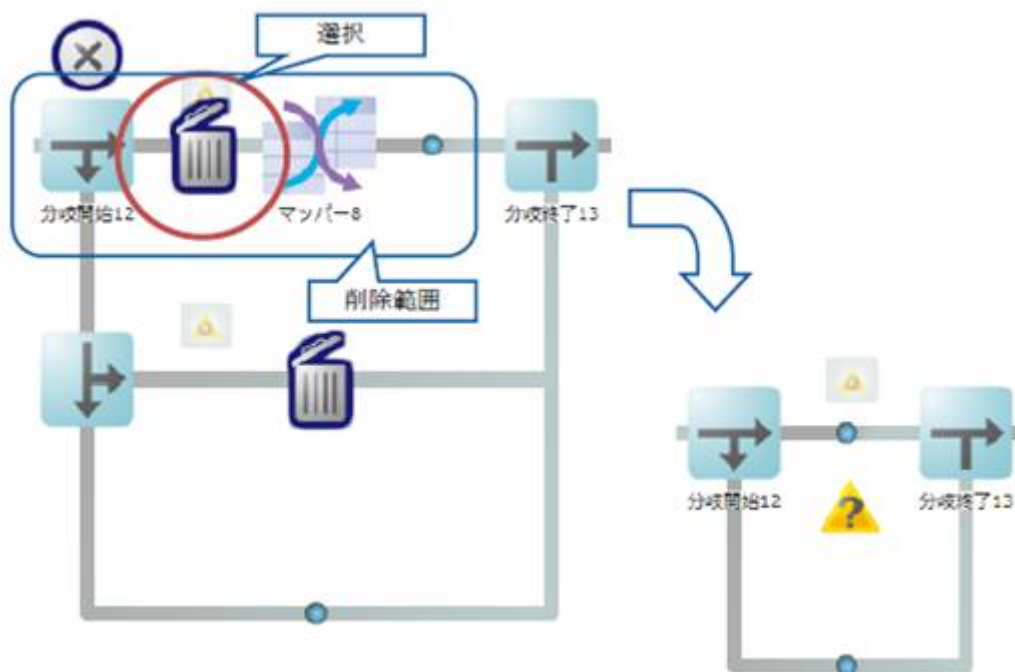
### 2.5.3.10 「分岐開始」での分岐削除

「分岐開始」で削除を実行した場合、すべての分岐の中から、どの分岐を削除するのかを選択します。



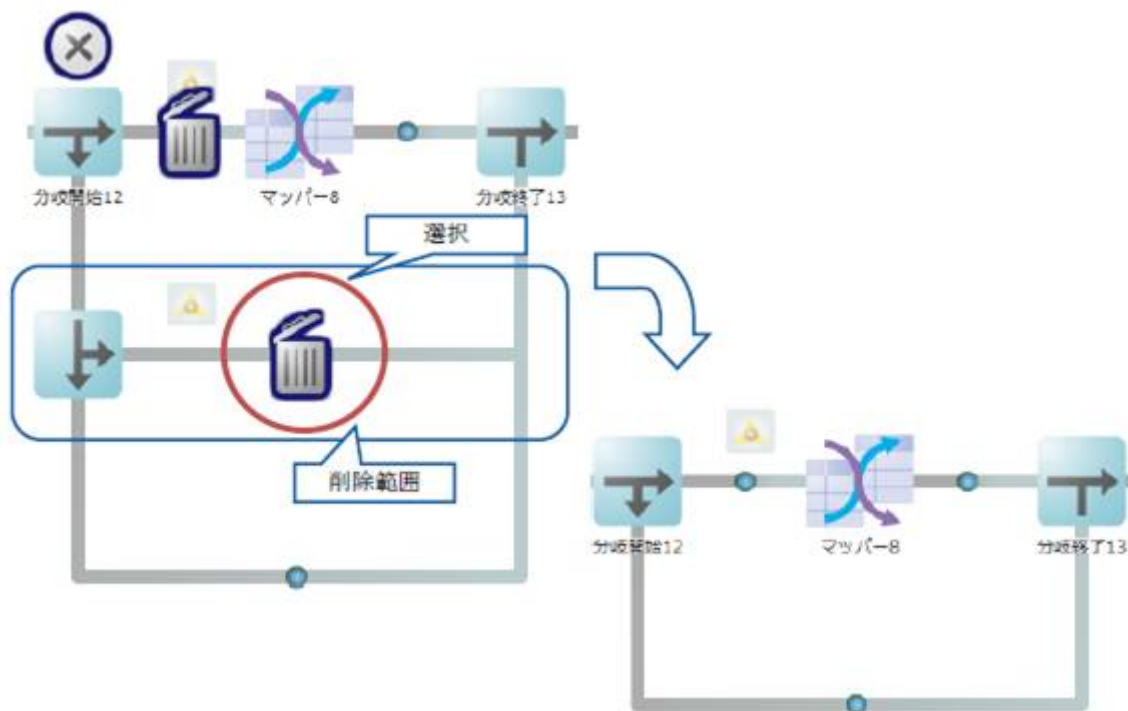
上の「削除可能位置」を選択した場合

上の列の分岐に含まれるコンポーネントが削除され、上から2つ目の分岐内容が1つ目の分岐になります。



下を選択した場合

下の列の分岐に含まれるコンポーネントが削除され、上の分岐内容はそのまま残ります



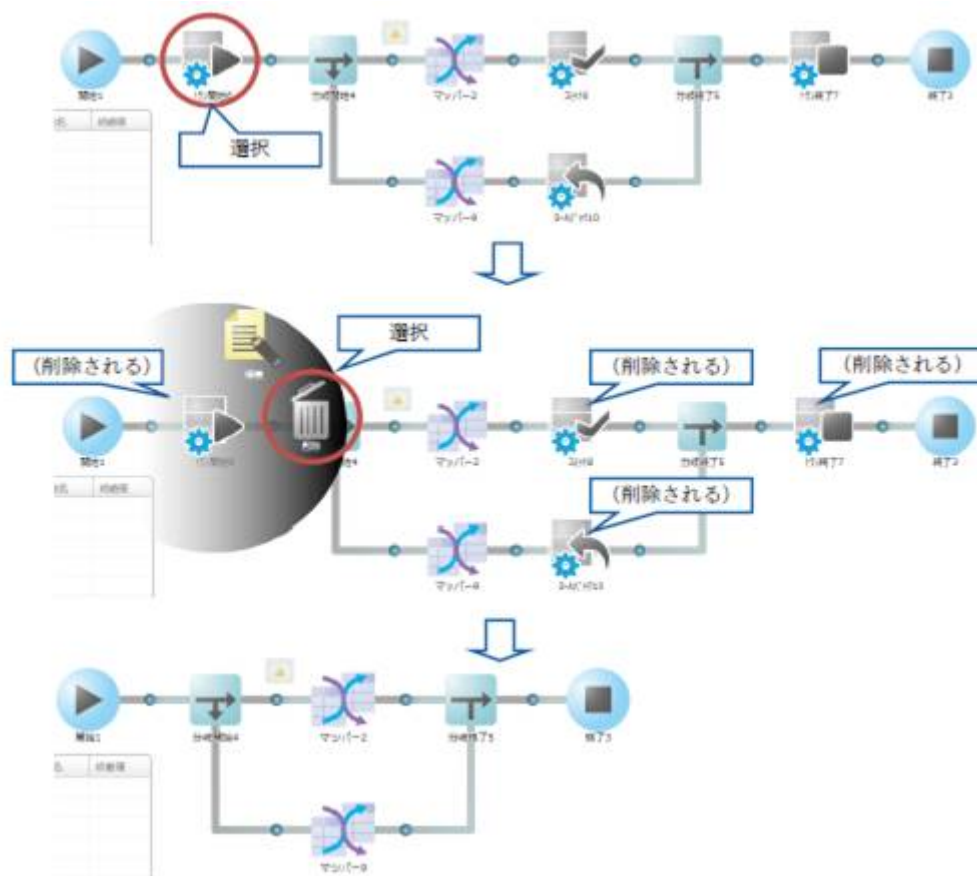
### 2.5.3.11 「追加分岐」での分岐削除

「追加分岐」で削除を実行した場合、選択した「追加分岐」に関連する分岐が削除されます



## 2.5.3.12 「トランザクション開始」の削除

「トランザクション開始」を削除した場合、対応する「トランザクション終了」と開始・終了に含まれる、すべてのコミット・ロールバックが削除されます。

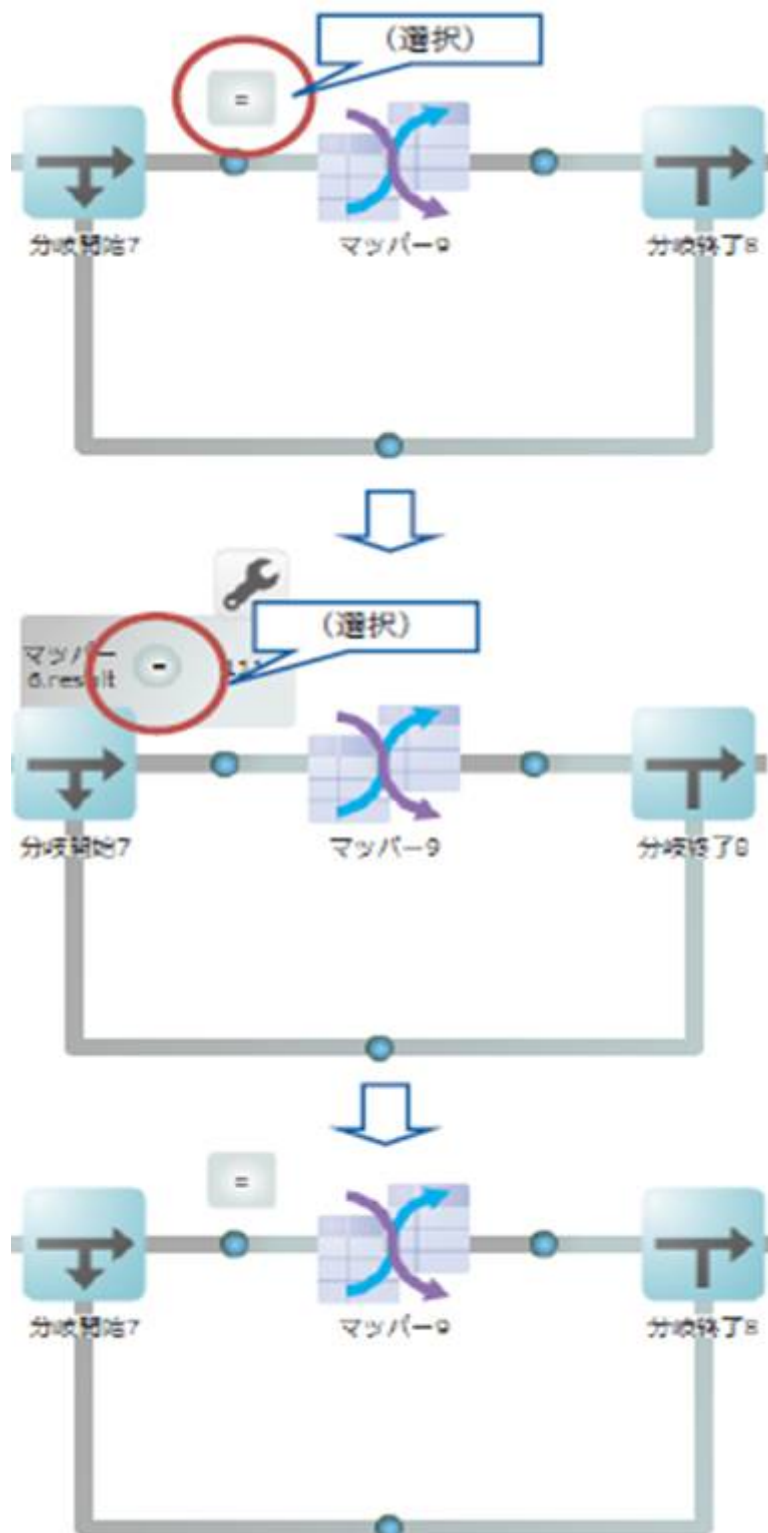


### 2.5.3.13 分岐条件の拡大と縮小

各分岐には分岐条件を設定し、その条件の時のみ、特定の処理を実行することができます。

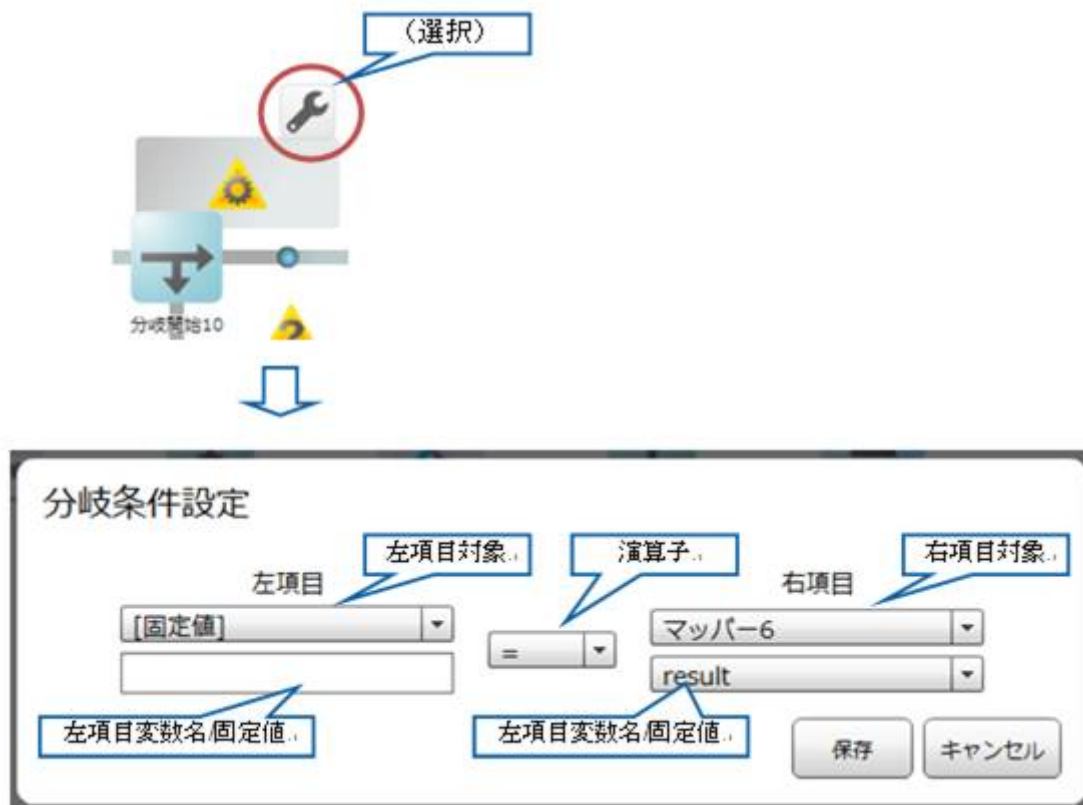
分岐条件をフロー上で確認するためには、縮小している分岐条件情報を選択して、拡大します。

元の大きさに戻すには拡大した分岐条件を再度、選択します。



### 2.5.3.14 分岐条件の設定

拡大した分岐条件の[設定]ボタンを押すと、設定画面が表示されます。



左項目対象	条件の左辺の変数の対象です
左項目変数名/固定値	条件の左辺の変数、もしくは、固定値の設定です 対象によって選択、もしくは、入力欄になります
演算子	左辺と右辺の関係を示す演算子です。 「=」「!=」「<」「>」「<=」「>=」のいずれかを設定できます。
右項目対象	条件の右辺の変数の対象です
右項目変数名/固定値	条件の右辺の変数、もしくは、固定値の設定です 対象によって選択、もしくは、入力欄になります
[保存]ボタン	設定内容を保存して終了します。
[キャンセル]ボタン	設定内容を保存せずに終了します。

## 2.5.3.15 分岐条件で使用可能な変数

種類 (対象表示名)	変数名	説明
種類 (対象表示名)	変数名	説明
固定値 ([固定値])	(なし)	特定の文字/値を、直接指定します。 他の変数と比較する為に、主に右辺に使用します。
トリガー変数 ([トリガー変数])	(設定したトリガー)	設定したトリガーで参照できるトリガー変数を指定 トリガー変数については「2.2.6.1 トリガー変数とは？」を参照ください。
フロー変数 ([フロー変数])	(定義した変数名)	フロー変数として定義を行ったものです 定義の方法については、「2.5.4 フロー変数定義」を参照ください。
コンポーネント変数 (コンポーネント名)	message	対象コンポーネント：全てのコンポーネント コンポーネントの実行時のメッセージです。コンポーネントによって内容は異なります
フロー変数 ([フロー変数])	readCount	対象コンポーネント：マッパー マッパーで読みだされた件数です
	writeCount	対象コンポーネント：マッパー マッパーで書き込まれた件数です
	Result	対象コンポーネント：全てのコンポーネント コンポーネントの実行結果です 0 : 成功 0以外 : 失敗、もしくは、エラー発生
種類 (対象表示名)	変数名	説明
固定値 ([固定値])	(なし)	特定の文字/値を、直接指定します。 他の変数と比較する為に、主に右辺に使用します。
トリガー変数 ([トリガー変数])	(設定したトリガー)	設定したトリガーで参照できるトリガー変数を指定 トリガー変数については「2.2.6.1 トリガー変数とは？」を参照ください。
フロー変数 ([フロー変数])	(定義した変数名)	フロー変数として定義を行ったものです 定義の方法については、「2.5.4 フロー変数定義」を参照ください。



また、トリガー変数、フロー変数はマッパー内の入力ファイル名、固定文字フィルタ、前後処理等で使用することが可能で、トリガー変数は`&{}`、フロー変数は`${}` で囲むことで指定が可能です。



- ・コンポーネント変数は「対象」欄では、コンポーネント名称で表示されます
- ・トリガー変数・コンポーネント変数はトリガーやコンポーネントの種類によって利用可能なものが異なります。

## 2.5.4 フロー変数定義

フロー変数を参照・定義する為の機能です。

通常はグレーアウトしていますが、マウスを合わせると内容をはっきり見ることができます。



### 2.5.4.1 フロー変数とは？

フロー変数とは、定義全体で値の受け渡しに使用することのできる変数で、

マッパー内部で変数から値を引き渡したり、格納することができ、

また、分岐での条件設定で、分岐の条件にも使用できます。

また、ファイル名、固定文字フィルタの値、前後処理で指定することも可能です。

指定には、以下の記述をします。

```
${フロー変数名}
```

上記図を例とすると、固定文字フィルタの値として

```
${変数}
```

と指定することで、固定文字フィルタから ”変数” の値 1200 が出力されます。



- ・フロー変数の値は 文字列型 として扱われます。

## 2.5.4.2 変数設定

[変数設定]ボタンを押すことによって、変数の設定を行う事ができます。編集中はフロー編集領域やメニューの操作はできません。



**変数名** フロー変数名です。クリックすることで編集できます

変数名	初期値
変数	1200

**初期値** フロー変数の初期値です。クリックすることで編集できます  
初期値の最大値は 50 文字です

変数名	初期値
変数	1200

[追加]ボタン フロー変数を新規に追加します

[削除]ボタン 選択したフロー変数を削除します  
使用中のフロー変数は削除できません

[終了]ボタン 編集を終了します

## 2.6 マッパー編集画面 (Flow Designer)

ここでは、マッパー編集画面の機能について説明します。

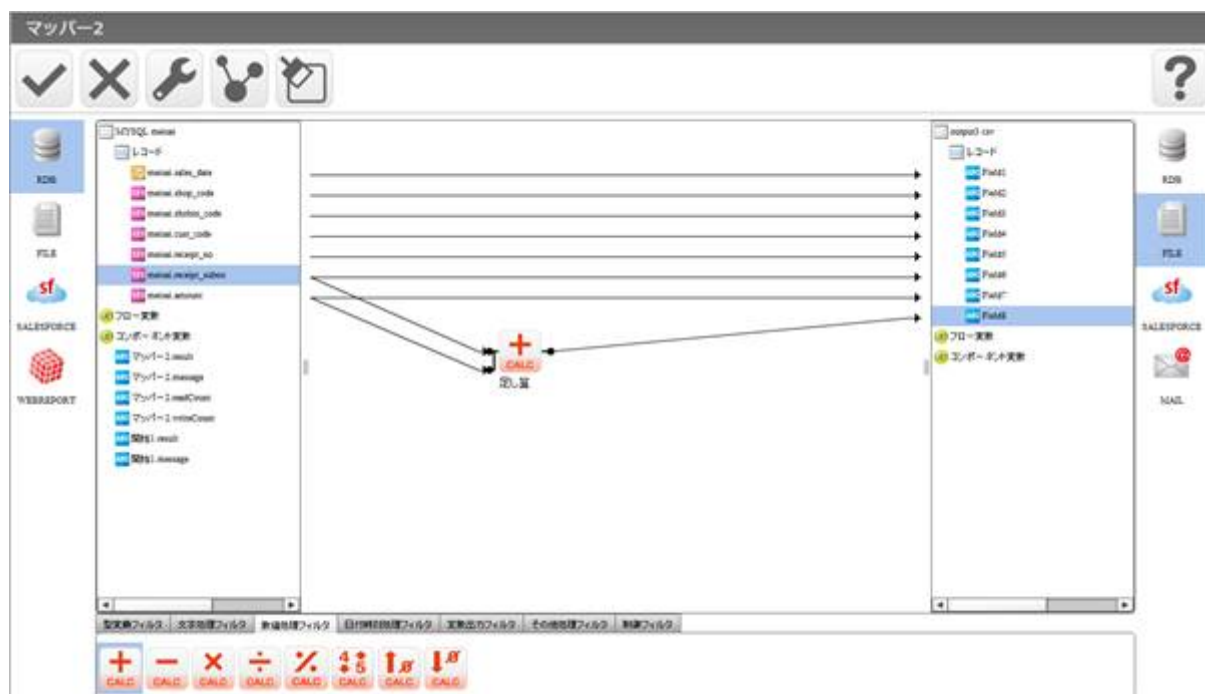
マッパー編集画面では、転送元から転送先への転送設定であるデータ転送定義である「マッパー」の作成ができます。

### 2.6.1 マッパーとは

マッパーを定義することで、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。

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。

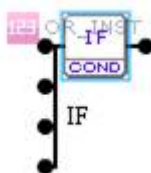
### 2.6.2 マッパー編集画面について



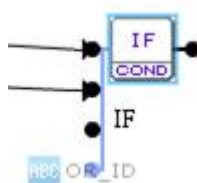
マッパー編集画面

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）のフィールドへドラッグ&ドロップすることにより、転送の設定ができ、線（フロー）が描画されます。

フィルタの入力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ドラッグ&ドロップするか、フロー接続部に転送元フィールドをドラッグ&ドロップしてください。



フィルタ上にドラッグ&ドロップ（上から順にフローが引かれます。）









フロー接続部にドラッグ&ドロップ（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。



メニュー一覧

	OK	処理フロー編集画面を表示します。 編集したマッパーの内容を保存します。
	キャンセル	処理フロー編集画面を表示します。 編集したマッパーの内容は保存されません。
	マッパーの設定	コンポーネントのタイトルを変更します。 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。
	自動結線	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。
	キャンバスのクリア	設定済の線とフィルタを一括削除します。
	ヘルプ	製品のマニュアルを表示します。

## 2.6.4 マッパーの設定画面

コンポーネントのタイトル フロー定義での表示名です

マッパーの種類 マッパーの種類です。

前後処理 マッパーを実行する前と、実行した後の処理を指定します

## 2.6.5 コンポーネントのタイトル

フロー定義上で表示されるマッパーの表示名を指定します。

## 2.6.6 マッパーの種類

マッパーの種類を指定します。

本バージョンではマッパー定義が選択された状態になっています。

## 2.6.7 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

・Qanat Server

指定したコマンドを Qanat Server を導入した OS 上で実行します。(Windows/Linux)

・入力元または出力先の IBM i (AS/400)

入力元または出力先のリソースに、本製品がサポートする IBM i (AS/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。

- ・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 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、どのような場合にどのような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。



## &lt; 後処理 &gt;

## 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

## ・マップ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。

## ・Qanat Server

指定したコマンドを、Qanat Server を導入した OS 上で実行します。(Windows/Linux)

## ・入力元または出力先の IBM i (AS/400)

入力元または出力先のリソースに、本製品がサポートする IBM i (AS/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。

## ・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作成コマンド「rmdir」を実行するコマンド

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

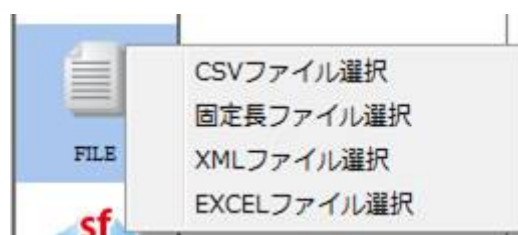


コマンド戻り値は、実行するコマンドによって異なります。事前に、どのような場合にどのような値が返されるのか把握しておく必要があります。

## 2.6.8 リソース選択



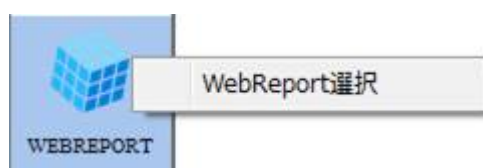
データ転送元・転送先には RDB を選択できます。データ転送元では RDB のテーブルの結合結果をリソースとして指定できます。



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・EXCEL ファイルを選択できます。



SALESFORCE では SALESFORCE のリソースを選択できます。



WEBREPORT では作成した WebReport 参照定義を選択できます。



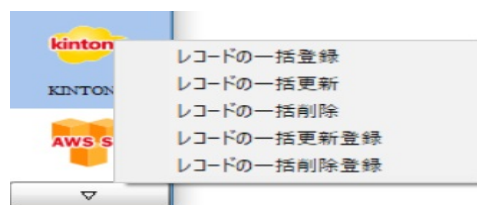
MAIL では登録したメールサーバーを指定できます。



PDF では出力する PDF ファイルを指定できます。



データ転送元の KINTONE ではレコードの一括取得を選択できます。



データ転送先の KINTONE ではレコードの一括登録・一括更新・一括削除を選択できます。



Amazon S3 では CSV ファイルを選択できます。



データ転送元の@TOVAS ではステータス取得を選択できます。



データ転送先の@TOVAS では私書箱・FAX・ファイル送信を選択できます。

## 2.6.9 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

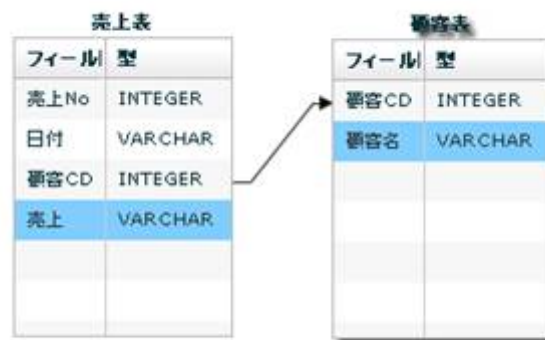
売上No	日付	顧客表 顧客CD	売上
1	2001/05/27	101	300
2	2001/05/27	101	1750
3	2001/05/27	103	5000
4	2001/06/27	104	1750
5	2001/06/27	104	2500

顧客表

顧客CD	顧客名
101	二島商店
102	姫路商事
103	大阪物産
104	神戸商店
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。

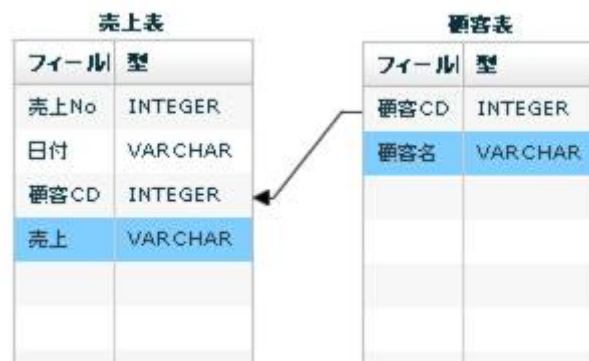


実行結果は以下の通りです。

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。

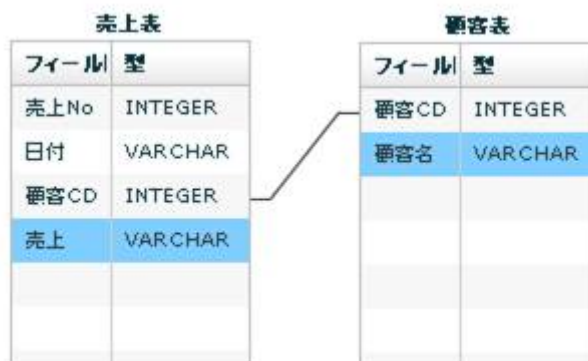


実行結果は以下の通りです。

#### 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。
カウント	選択されたフィールドの個数（行数）を返します。
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

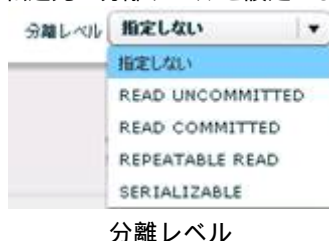
上記の出力結果

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。

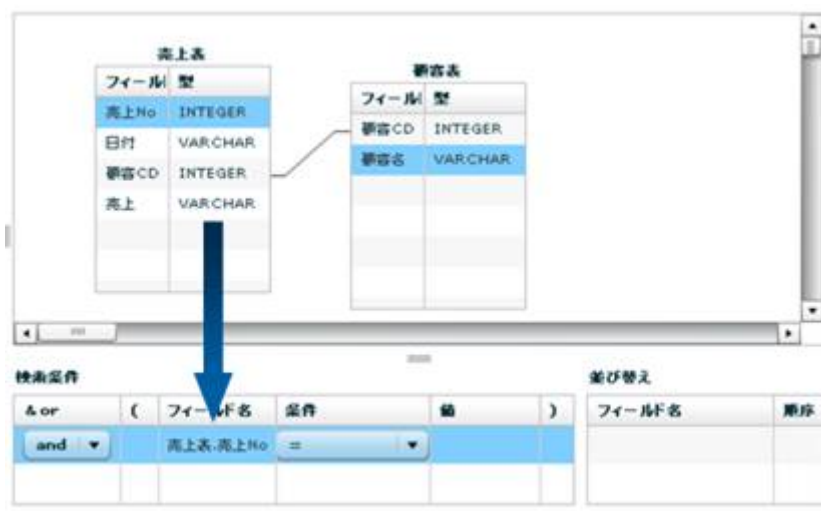


データベースにより、サポートされない分離レベルもあります。詳細は、転送元データベースの仕様を確認してください。

### 【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。



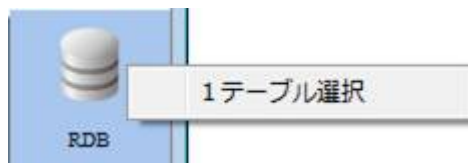
## 並び替え

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。



RDB 選択



- 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

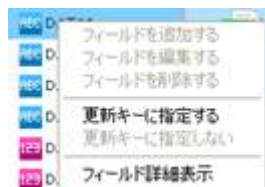
- インサート      データを追加だけします。キーの重複時にはエラーになります。
- アップデート      データを更新だけします。更新対象がない場合エラーになります。
- インサート/アップデート      データを更新キーにより、追加または更新します。
- リプレース      データを全件削除後、追加します。
- デリート      データを自動選択されるキーにより、削除します。

アップデート、INSERT/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

デフォルトの状態では、データベースからデータを取得した際に、データの前後の半角スペースは削除するよう動作をします。

例：データが ' JBAT ' だった場合は 'JBAT' と出力されます。



データ前後の半角スペースの出力を行いたい場合は Qanat Server の

qanat\_appldef.xml 内にある TRIM\_MODE を 0 (off) に設定し

サーバーの再起動を行ってください。

qanat\_appldef.xml は以下のディレクトリに導入されています。

Tomcat のインストールディレクトリ/webapps/qanat/config2/qanat/

## 2.6.10 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。



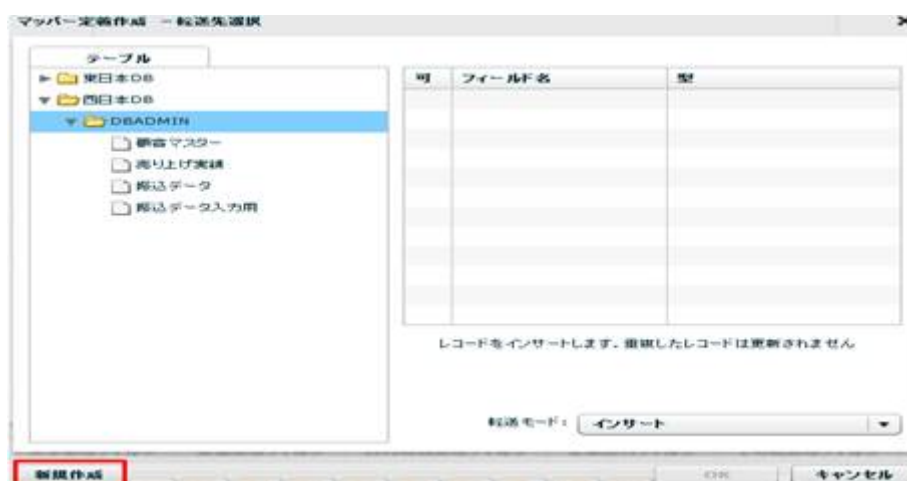
権限の章に書いてある権限がユーザーに与えられている場合のみ利用可能です。

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。

## 2.6.11 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

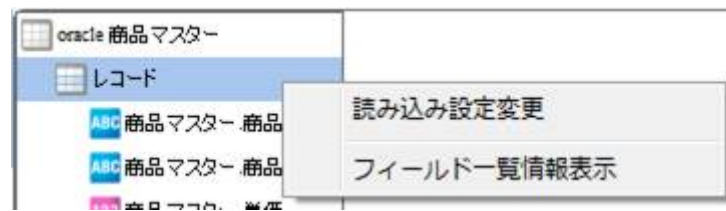


読み込み設定変更



テーブル詳細表示

## ・転送元レコード関連

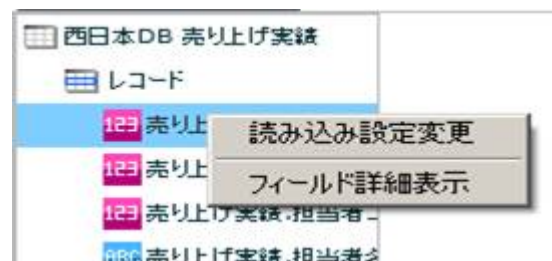


転送元レコードを選択時に右クリックで表示されるメニュー

順ポジション	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	コロイ
1	売上げ実績	年度	INTEGER	-	10	0	し/え	し/え	
2	売上げ実績	四半期	INTEGER	-	10	0	し/え	し/え	
3	売上げ実績	担当コード	INTEGER	-	10	0	し/え	し/え	
4	売上げ実績	担当名	VARCHAR	40	-	-	し/え	し/え	
5	売上げ実績	地区コード	INTEGER	-	10	0	し/え	し/え	
6	売上げ実績	地区名	VARCHAR	40	-	-	し/え	し/え	
7	売上げ実績	商品コード	INTEGER	-	10	0	し/え	し/え	
8	売上げ実績	商品名	VARCHAR	40	-	-	し/え	し/え	
9	売上げ実績	数量	INTEGER	-	10	0	し/え	し/え	
10	売上げ実績	売上げ金額	INTEGER	-	10	0	し/え	し/え	
11	売上げ実績	粗利益	INTEGER	-	10	0	し/え	し/え	

フィールド一覧情報表示

## ・転送元フィールド関連

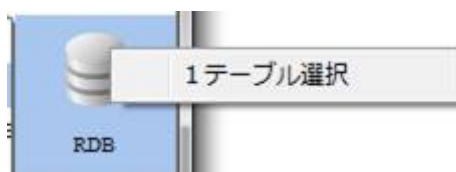


転送元フィールドを選択時に右クリックで表示されるメニュー

フィールド表示名: 売上げ実績.年度  
 フィールド実名: 売上げ実績.年度  
 型: INTEGER  
 桁数: 10      小数桁数: 0  
 主キー: し/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

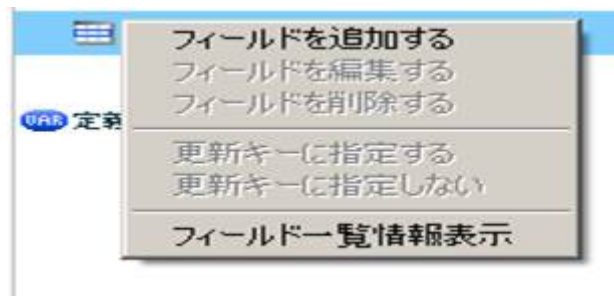


書き込み設定変更



テーブル詳細表示

## ・転送先レコード関連



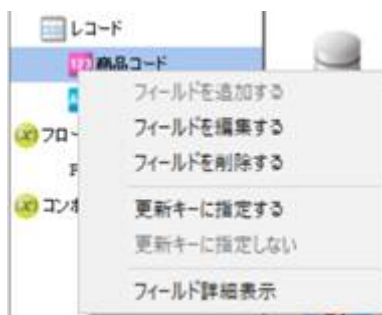
転送先レコードを選択時に右クリックで表示されるメニュー

フィールドを追加する

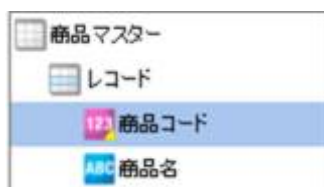
ボタソツ	フィールド名	実フィールド名	型名	文字列長	桁数	小数桁数	NULL許可	主キー	コムナ
1	年度	年度	INTEGER	-	10	0	許可	許可	
2	四半期	四半期	INTEGER	-	10	0	許可	許可	
3	得意者コード	得意者コード	INTEGER	-	10	0	許可	許可	
4	得意者名	得意者名	VARCHAR	40	-	-	許可	許可	
5	地区コード	地区コード	INTEGER	-	10	0	許可	許可	
6	得意地区	得意地区	VARCHAR	40	-	-	許可	許可	
7	商品コード	商品コード	INTEGER	-	10	0	許可	許可	
8	商品名	商品名	VARCHAR	40	-	-	許可	許可	
9	数量	数量	INTEGER	-	10	0	許可	許可	
10	売上げ金額	売上げ金額	INTEGER	-	10	0	許可	許可	
11	種利益	種利益	INTEGER	-	10	0	許可	許可	

フィールド一覧情報表示

## ・転送先フィールド関連

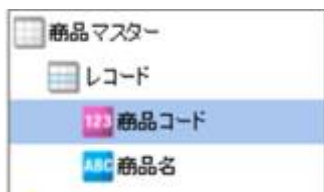


転送先フィールドを選択時に右クリックで表示されるメニュー



「更新キーに指定する」を選択した場合のフィールド表示

( フィールド名左側のアイコンの右下に、更新キーを表すマークが付加されます )



「更新キーに指定しない」を選択した場合のフィールド表示

( 更新キーを表すマークが除去されます )



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

削除 結線を削除します。

・フィルタ関連



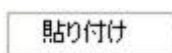
フィルタを選択時に右クリックで表示されるメニュー



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。

・何も無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のレコード関連



テーブルの新規作成時の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に作成されたフィールド関連



テーブルの新規作成時に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。
- ❗ テーブルの新規作成時に SQL 文のキーワード (key など) と同様なフィールド名を指定した場合はエラーになる場合があります。

## 2.6.12 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込みをします。

絞込みにはワイ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み込む>

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

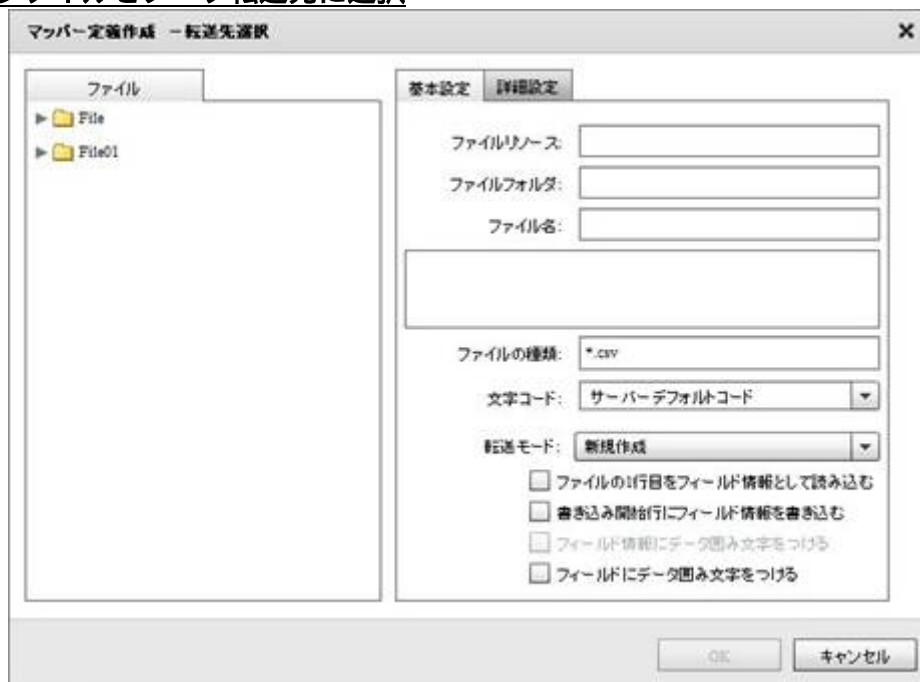
#### < データ区切り文字 >

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を選択



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。

### <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み込む>

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。

### <書き込み開始行にフィールド情報を書き込む>

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

### <フィールド情報にデータ囲み文字をつける>

出力されるフィールド情報に対するデータ囲み文字の有無を指定します。

<書き込み開始行にフィールド情報を書き込む> にチェックが入っている場合のみ指定することが可能です。

### <フィールドにデータ囲み文字をつける>

出力されるデータ部に対するデータ囲み文字の有無を指定します。

データ囲み文字の有無はフィールド単位でも指定することが可能です。

フィールドにデータ囲み文字の指定がされている場合、フィールドの設定が優先されます。

基本設定	フィールドの設定	データ囲み文字の有無
<フィールドにデータ囲み文字をつける>	クォート	
チェックあり	あり	あり
チェックあり	なし	なし
チェックあり	書き込み設定に従う	あり
チェックなし	あり	あり
チェックなし	なし	なし
チェックなし	書き込み設定に従う	なし

## &lt;転送モード&gt;

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

## &lt;データ区切り文字&gt;

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。

## &lt;データ囲み文字&gt;

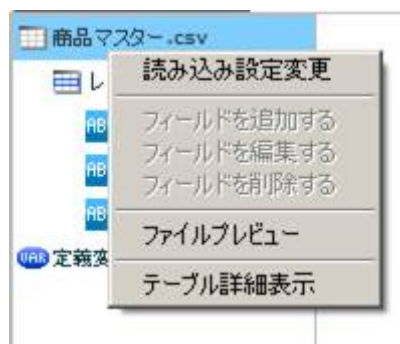
ファイル中の各データを囲む文字を指定します。囲み文字は「' / " / なし」から選択するかまたは、入力し指定します。

## &lt;改行コード&gt;

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。

## 2.6.13 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



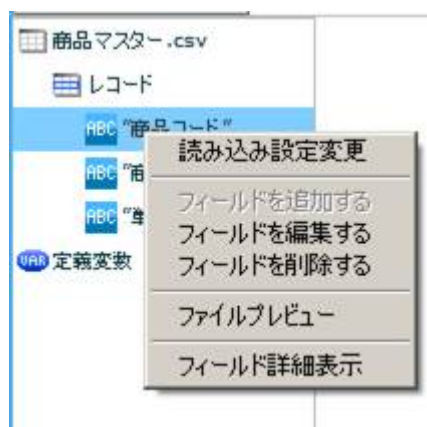
フィールドを追加



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



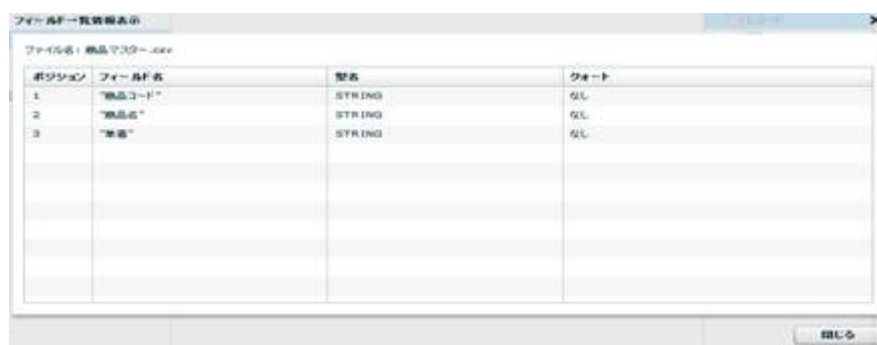
転送先レコードを選択時に右クリックで表示されるメニュー



フィールドを追加

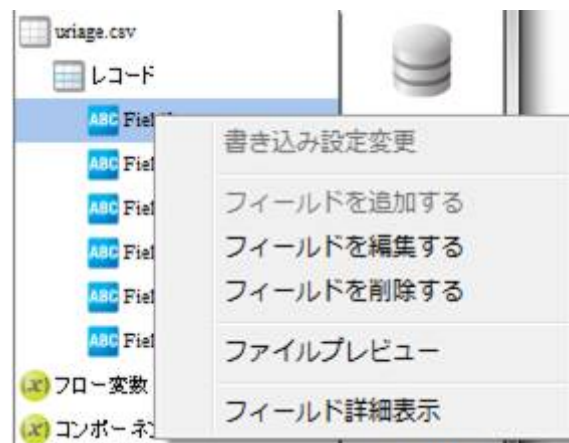


ファイルプレビュー



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



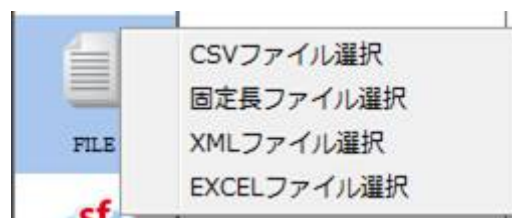
ファイルプレビュー



フィールド詳細表示

## 2.6.14 固定長ファイル選択

### 固定長ファイルをデータ転送元に選択



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>

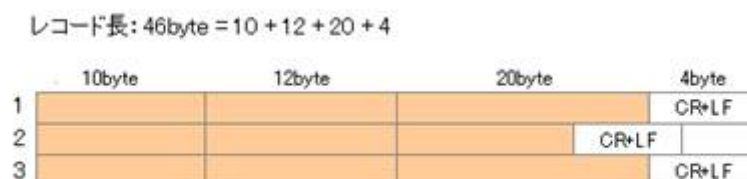
<ファイル> で指定したフォルダのうち利用して表示するファイルの絞込みをします。絞込みにはワールドカードを利用します。固定長ファイル選択の場合 “ \*.\* ” と入力されています。

## &lt; 改行コード &gt;

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。



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。

## &lt; 文字コード &gt;

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。

## &lt; レコード長 &gt;

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

## 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。定義変数を指定可能です。

<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。絞込みにはワイ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。指定した改行コードが出力レコード末尾に付加されます。転送先で表示するレコード長に改行コードの byte 数は追加されません。

<文字コード>

書き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。

<レコード長>

設定したフィールド長の総和が表示されます。ユーザーが指定することはできません。

<転送モード>

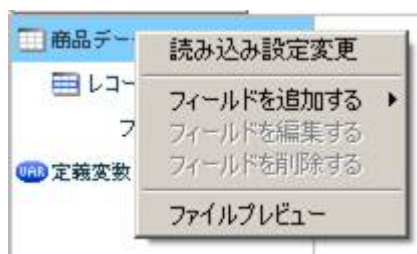
データ転送時の書き込み方法を指定します。

- ・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。
- ・追加書き込み                         : 既存データの末尾から転送データを追加で書き込みます。



## 2.6.15 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

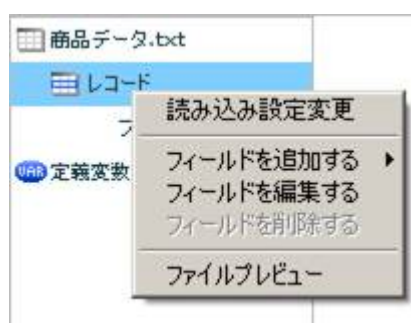
## [ファイルプレビュー]

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。

- ・レコード識別子：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が判定します。詳細は 2.6.12 をご参照ください。
- ・文字列           ：フィールドタイプとして文字型を指定します。
- ・ゾーン 10 進   ：フィールドタイプとしてゾーン 10 進数を指定します。
- ・パック 10 進   ：フィールドタイプとしてパック 10 進数を指定します。
- ・バイナリ       ：バイナリを入力データとして扱います。注意として出力ファイルのバイナリフィールドに転送する以外の処理はできません。

**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、文字コード入力が選択できます。

フィールドタイプにゾーン 10 進、パック 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)

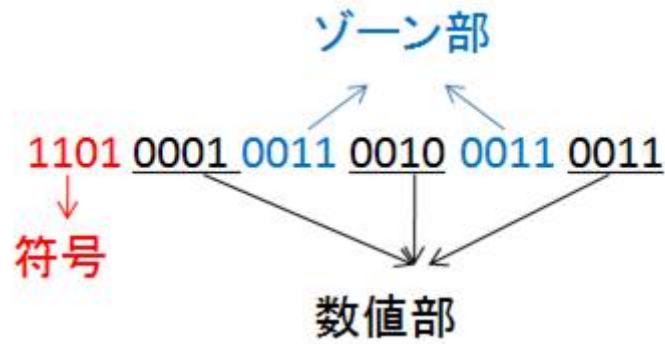
### ゾーン 10 進数について

**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。

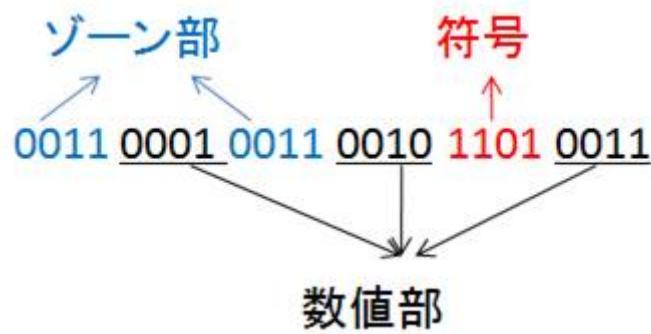
以下に -123 という数値をそれぞれの符号形式で表したバイナリ情報を説明します。



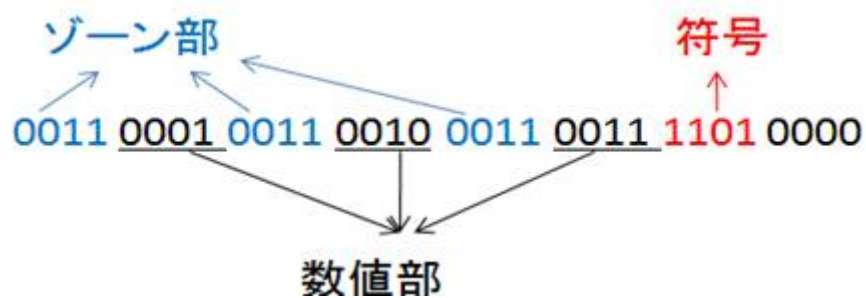
符号形式：前



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。

The dialog box '新規フィールド追加' (New Field Addition) has the following settings:

- 基本設定 (Basic Settings) tab is selected.
- フィールド名 (Field Name): 商品番号 (Product Number)
- フィールドタイプ (Field Type): 文字列 (String)
- フィールド長 (byte) (Field Length): [Empty]
- 開始位置 (byte) (Start Position): 0
- 揃え (Align):  左揃え (Left)  右揃え (Right)
- パディング文字 (Padding Character): 半角スペース (Half-width Space)

商品番号フィールド

The dialog box '新規フィールド追加' (New Field Addition) has the following settings:

- 基本設定 (Basic Settings) tab is selected.
- フィールド名 (Field Name): 商品名 (Product Name)
- フィールドタイプ (Field Type): 文字列 (String)
- フィールド長 (byte) (Field Length): 22
- 開始位置 (byte) (Start Position): 10
- 揃え (Align):  左揃え (Left)  右揃え (Right)
- パディング文字 (Padding Character): 半角スペース (Half-width Space)

商品名フィールド

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,アメリカンコーヒー,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ました。

[フィールドを編集する]

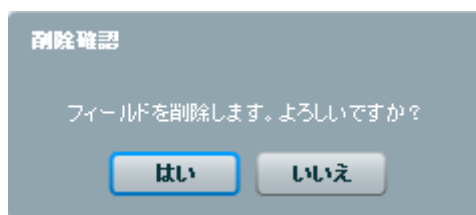
レコードパターンの名前を編集することができます。



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。

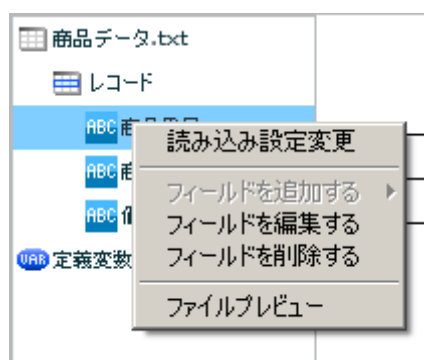


レコードパターン削除

[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



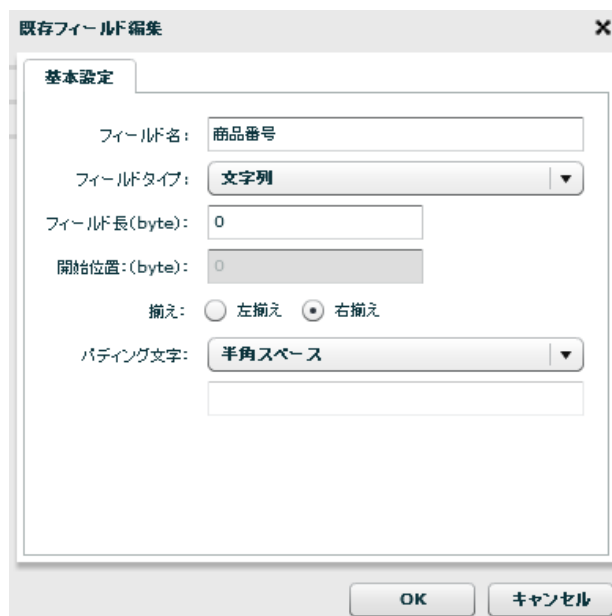
転送元フィールドを選択時に右クリックで表示されるメニュー

[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]

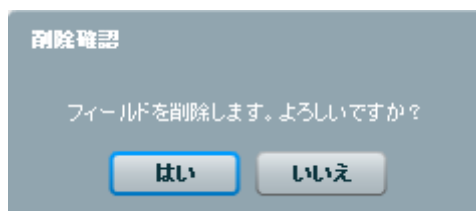
入力フィールドを編集します。



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。

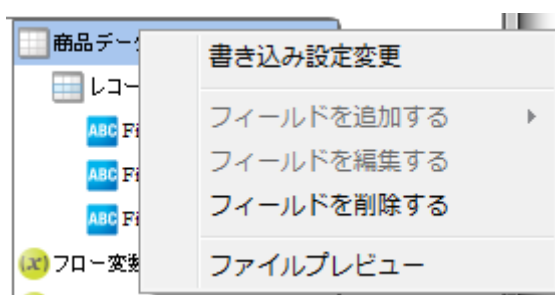


フィールド削除

## [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

ファイルリソース: ファイルリソース

ファイルフォルダ: FIXED

ファイル名: 商品データ.txt

改行コード: NONE

文字コード: サーバーデフォルトコード

レコード長(byte): 0

転送モード: 新規作成

OK キャンセル

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 商品データ.txt

101BlendCoffee 450  
102AmericanCoffee480  
103ICECoffee 480  
201Mocha 500  
202Guatemala 500  
203Columbia 500  
204Brazil 500  
205Mandarin 500  
206Kilimanjaro 550  
207Blue Mountain 630  
301Cafe au lait 550  
302ICE Cafeaulait580  
303Winner Coffee 580  
304Russia Coffee 600  
305Coffee Jelly 580  
401Lemon Tea 450

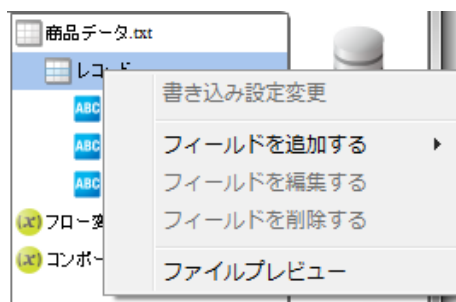
文字コード: サーバーデフォルトコード

レコード長(byte): 0

閉じる

ファイルプレビュー

## ・ 転送先レコード関連



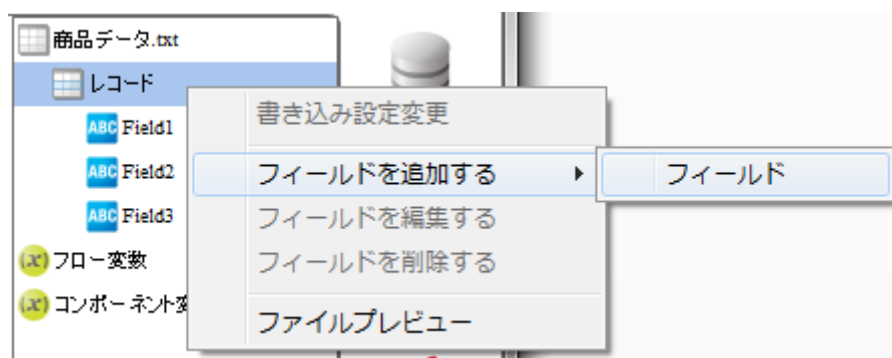
転送先レコードを選択時に右クリックで表示されるメニュー

## [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。

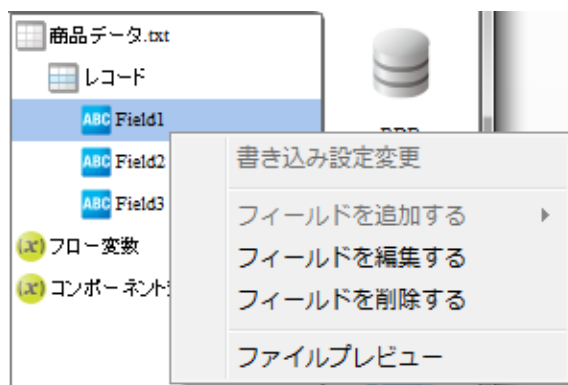


フィールド追加

## [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



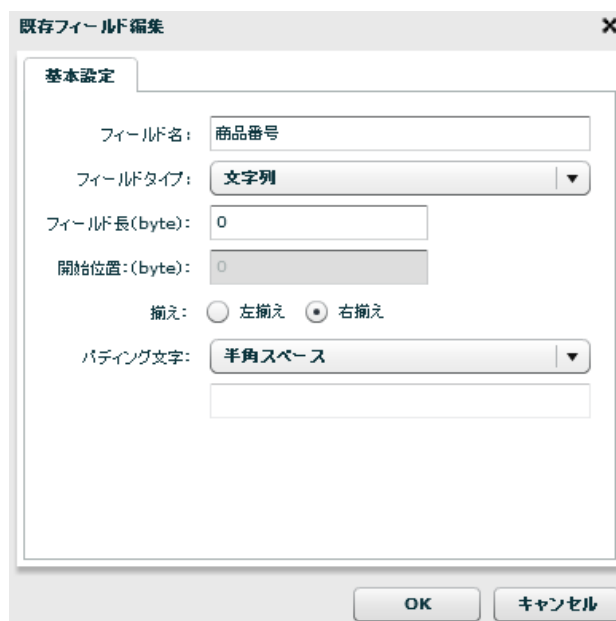
転送先フィールドを選択時に右クリックで表示されるメニュー

## [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。

## 2.6.16 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。 ）

1	23	23	23	23	23	ヘッダ (1レコード)			
2	33	ジェ化イ'ンコ	4	か'カ'ウケン	3211111111	か)ニホ'ンコ'カ	1500000	123	データ (複数レコード)
2	78	エー'キ'ンコ	2	ヨ'ル'マ'デン	12222222	か)マ'キ'ヨ'ケン	2300000	234	
2	123	フー'キ'ンコ	34	カ'マ'デン	13333333	ニ'ホ'ン'コ'カ	1800000	345	
2	33	ジェ化イ'ンコ	1	ト'ウ'ケン	11444444	サ'キ'ヨ'カ'イ	100000	456	
2	78	エー'キ'ンコ	2	ヨ'ル'マ'デン	15555555	カ'ク'ロ	10000	567	
2	123	フー'キ'ンコ	12	オ'ウ'ケン	16666666	サ'ク'ロ	10000	678	
2	123	フー'キ'ンコ	1	ト'ウ'ケン	17777777	サ'イ'ホ'キ'ヨ'カ'イ	1200000	789	
2	33	ジェ化イ'ンコ	23	カ'マ'デン	18888888	イ'ト'ク'ム	2300000	890	
2	33	ジェ化イ'ンコ	4	か'カ'ウケン	3219999999	イ'ク'カ)ニ'ホ'ン'コ'カ	100000	901	
2	78	エー'キ'ンコ	1	ト'ウ'ケン	11111111	ミ'カ'ク'デン	10000	222	
8	10.0	333000.0							トレーラ (1レコード)
9									エンド (1レコード)

レコード識別子

複数のレコードパターンをもつ固定長データ

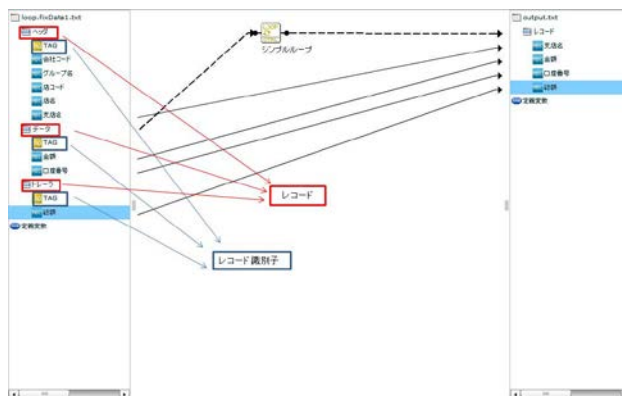
固定長データには複数レコードからなるデータにヘッダ、トレーラ、エンドなどの付加情報がのっているものがあります。

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。そのために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

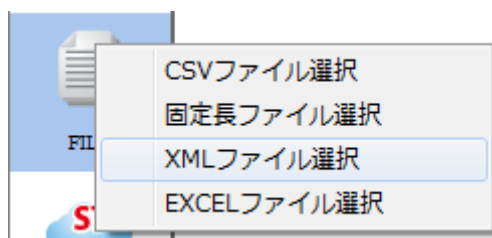
この定義で出力されるレコードは、下記のフォーマットで出力されます。

- 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

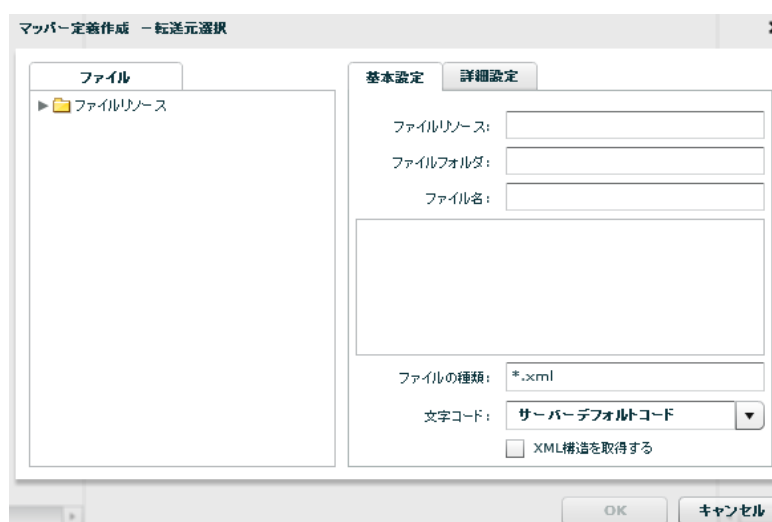
というように、ループをつないだレコードだけがループ処理を受け、ヘッダとトレーラの固有データがそれに付加され出力されます。

## 2.6.17 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り込みをします。絞り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。

#### <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

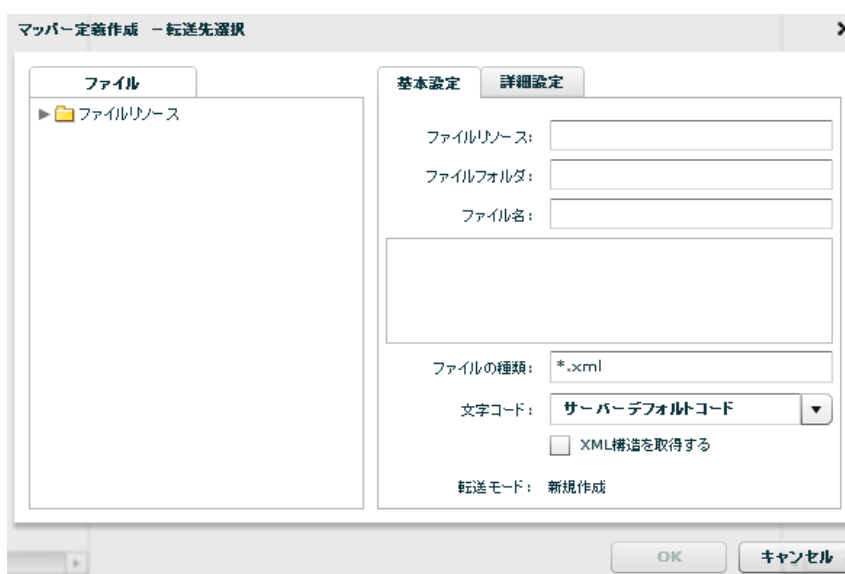


転送元 詳細設定

< 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



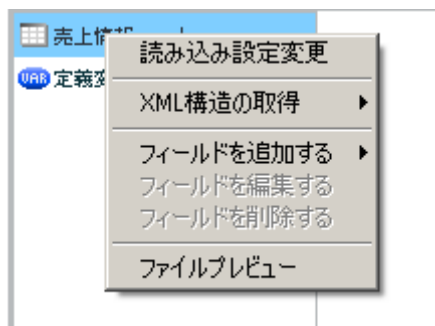
転送先 詳細設定

### < 書き込み時インデント処理を行う >

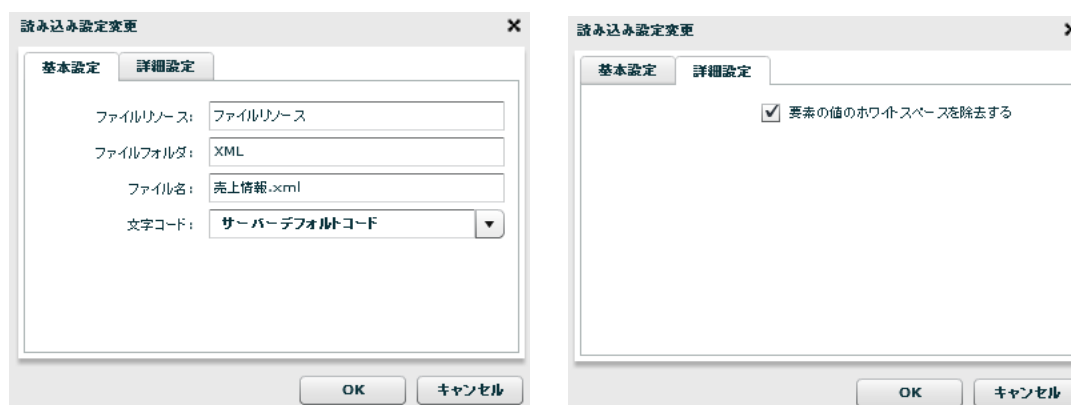
要素の先頭の字下げを行います。

## 2.6.18 XML ファイルを利用したときの各種メニュー

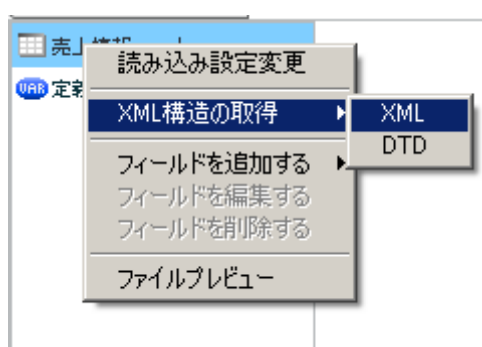
- ・ 転送元ファイル関連



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

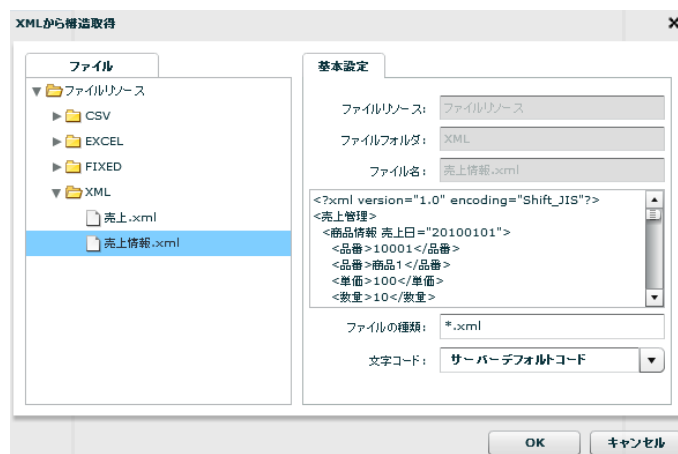


XML 構造の取得

[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]

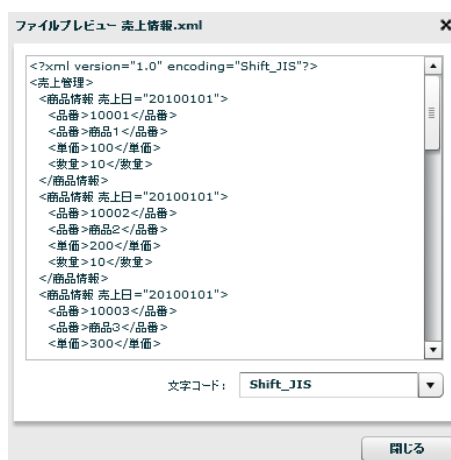
指定した DTD ファイルから XML 構造を取得します。



XML 構造取得先指定 (DTD ファイル)

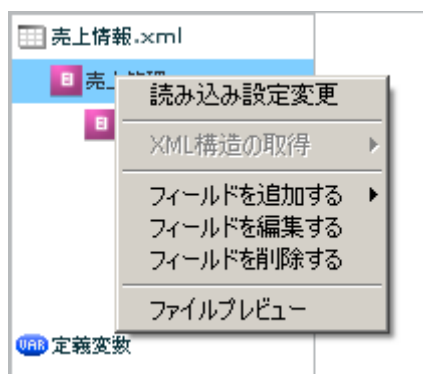
[ファイルプレビュー]

ファイルプレビューを表示します。

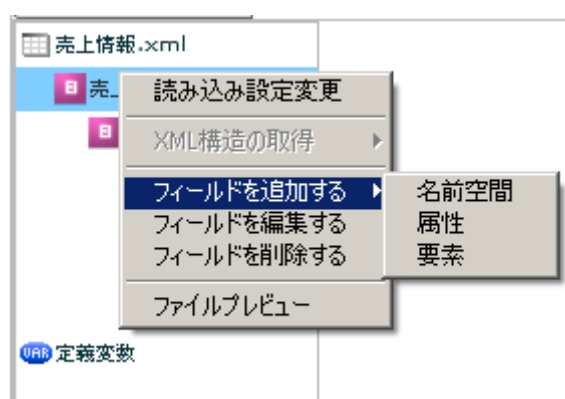


ファイルプレビュー

## ・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]

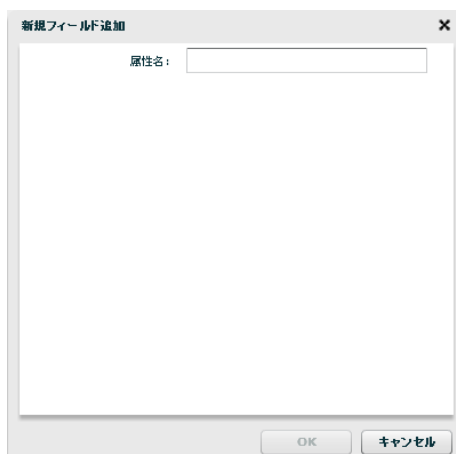
入力した接頭辞の名前空間をフィールドに追加します。



フィールド追加 名前空間

## [フィールドを追加する] [属性]

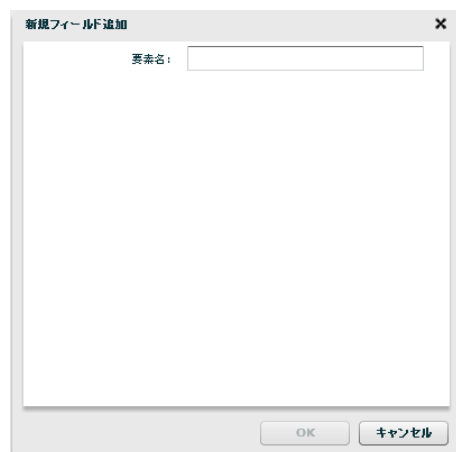
入力した名前の属性をフィールドに追加します。



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。

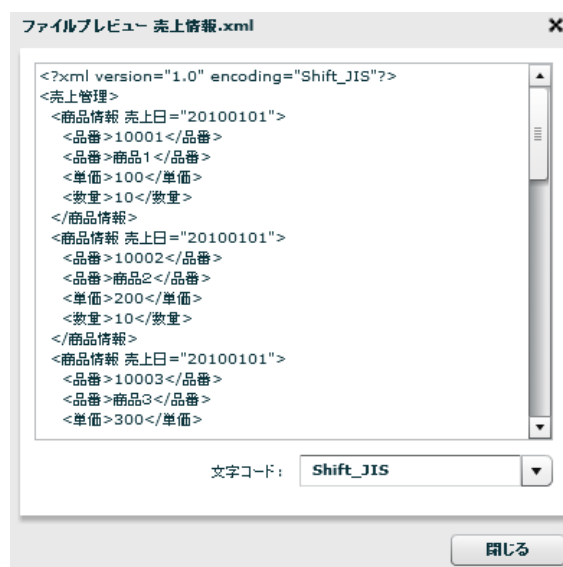


フィールド編集

## [フィールドを削除する]

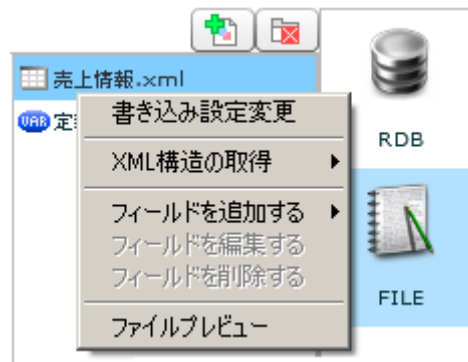
選択したフィールドを削除できます。

## [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

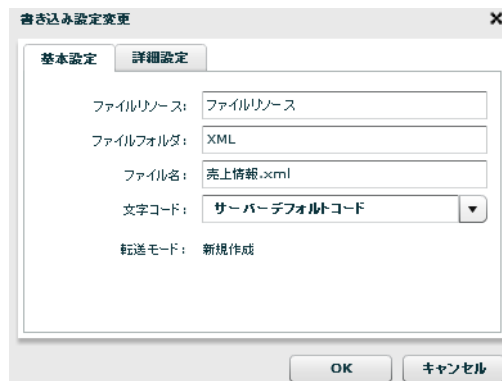


転送先ファイルを選択時に右クリックで表示されるメニュー

## [書き込み設定変更]

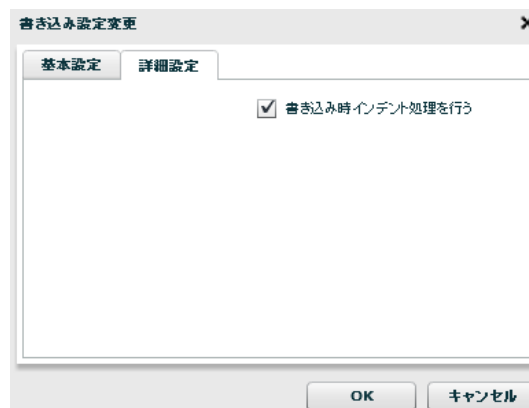
## 基本設定

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。



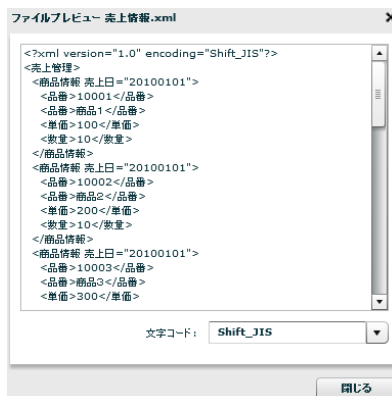
XML 構造取得先指定 (XML ファイル)

[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。

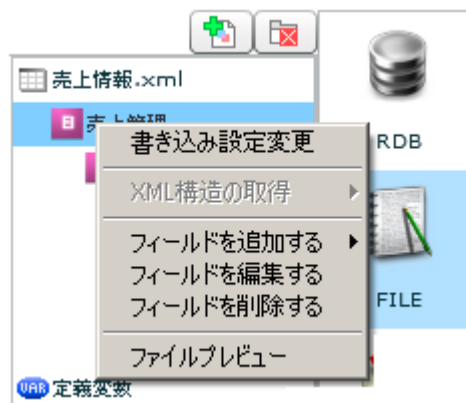
[ファイルプレビュー]

ファイルプレビューを表示します。

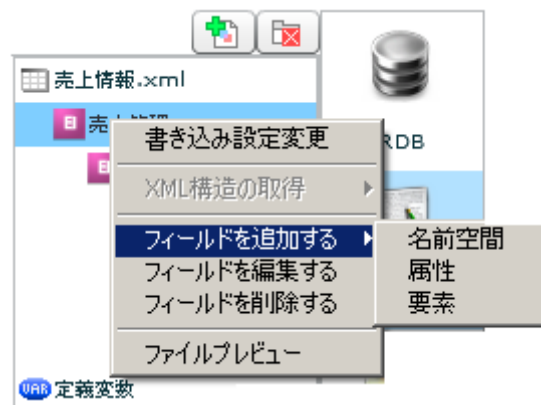


ファイルプレビュー

## ・ 転送先レコード関連



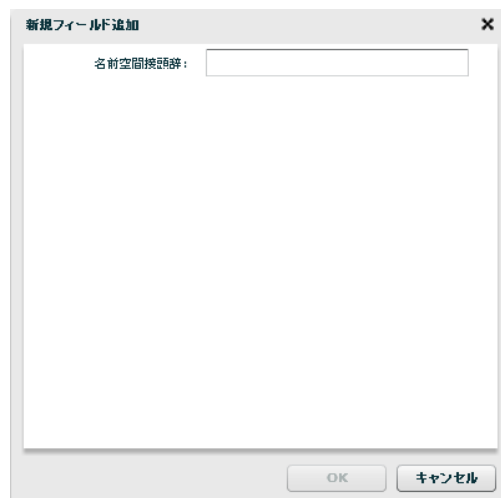
転送先レコードを選択時に右クリックで表示されるメニュー



フィールドを追加する

[フィールドを追加する] [名前空間]

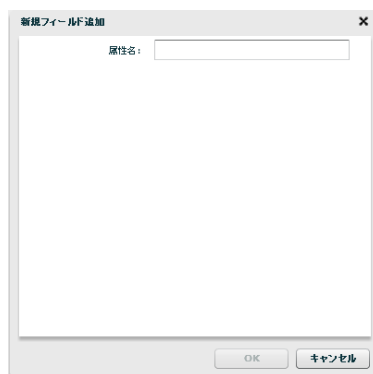
入力した接頭辞の名前空間をフィールドに追加します。



フィールド追加 名前空間

## [フィールドを追加する] [属性]

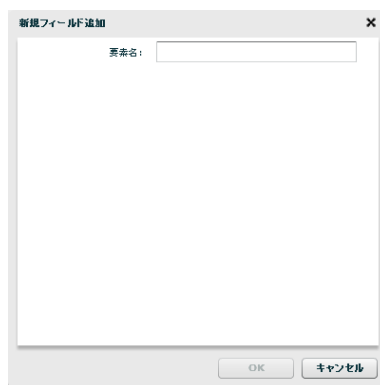
入力した名前の属性をフィールドに追加します。



フィールド追加 属性

## [フィールドを追加する] [要素]

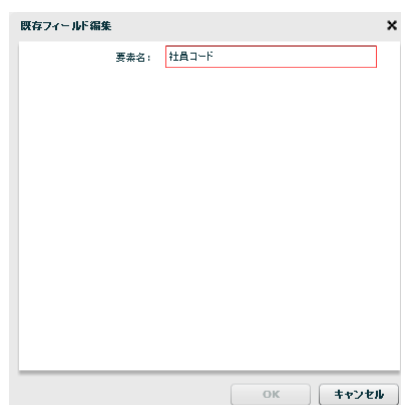
入力した名前の要素をフィールドに追加します。



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。



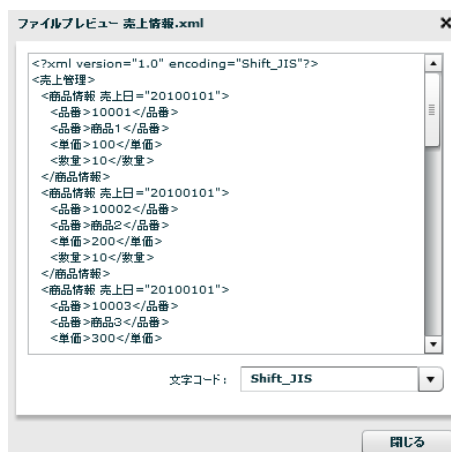
フィールド編集

## [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]



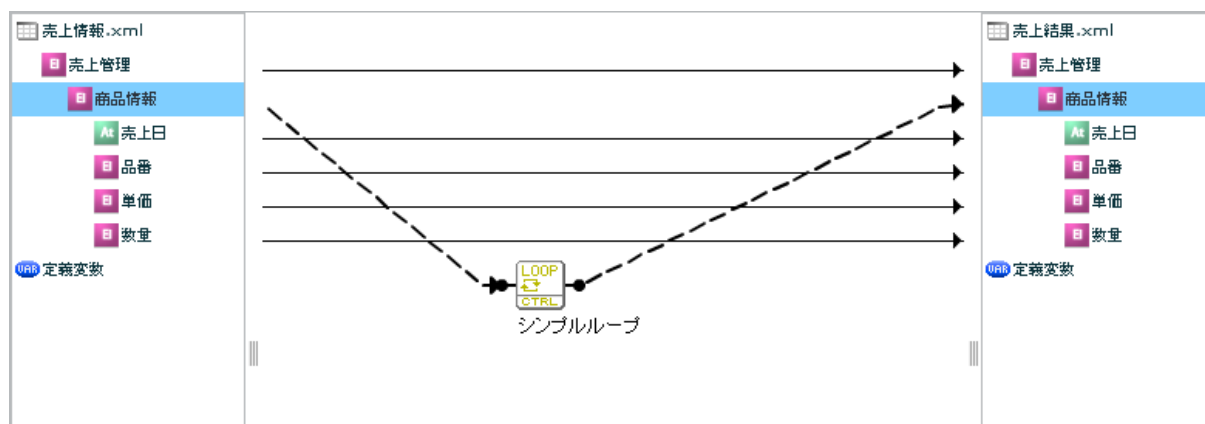
ファイルプレビュー

## 2.6.19 XML ファイル転送におけるループ処理

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。

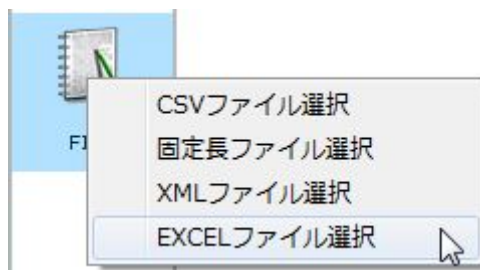
下図の場合単価フィールドがループの基準要素となります。



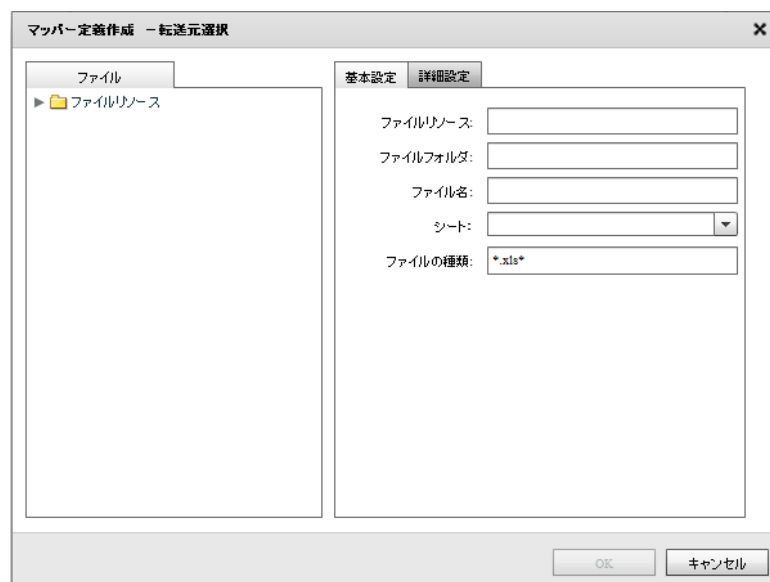
XML ファイル転送におけるループ処理

## 2.6.20 EXCEL ファイル選択

### EXCEL ファイルをデータ転送元を選択



EXCEL 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を表示します。

手動での指定や、定義変数を指定することが可能です。

#### <シート>

<ファイル> で指定したファイルのシート名をプルダウン表示します。

手動での指定や、定義変数を指定することが可能です。

#### <ファイルの種類>

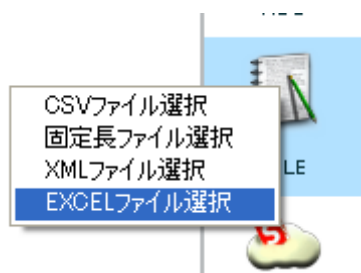
<ファイル> で指定したフォルダのうち利用して表示するファイルの絞込みをします。絞込みにはワールドカードを利用します。EXCEL ファイル選択の場合 “\*.xls\*” と入力されています。

## 詳細設定

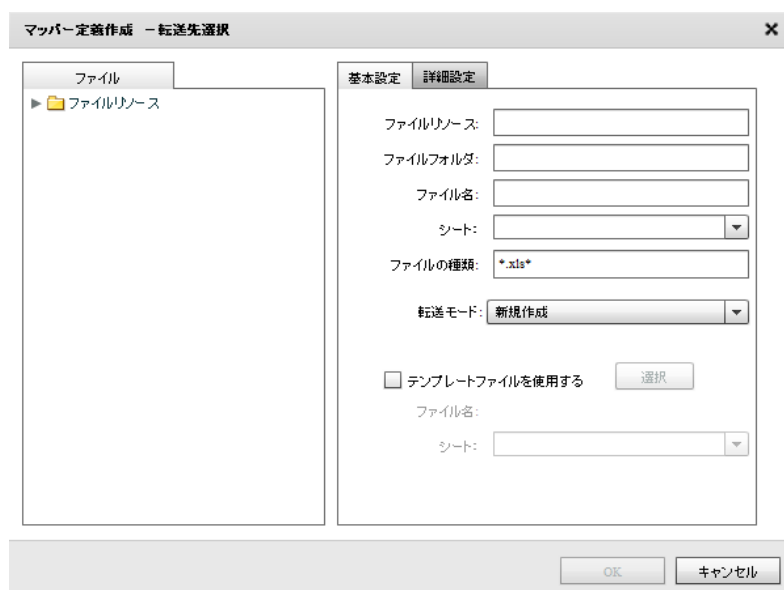
<エクセル転送モードの固有設定>

大量データ転送モードと通常転送モードを切り替えられます。大量データ転送の場合には処理速度が遅くなるためご注意ください。詳細は導入マニュアルのエクセルモード（読み込み側）をご確認ください。

## EXCEL ファイルをデータ転送先に選択



EXCEL ファイル選択



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

<ファイル名>

データ転送先のファイル名を指定します。

手動で入力することも可能です。

#### <シート>

<ファイル>で指定したファイルのシート名をプルダウン表示します。

手動で入力することも可能です。

#### <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。絞り込みにはワールドカードを利用します。EXCEL ファイル選択の場合 “ \*.xls\* ” と入力されています。

#### <転送モード>

データ転送時の書き込み方法を指定します。

- ・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。
- ・更新（ない場合新規）               既存ファイルに更新を行います。ファイルが存在しない場合、新規作成を行います。
- ・追加書き込み（ない場合新規）      既存データは更新されず、既存レコードの末尾に追加する形でデータを挿入します。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。  
  
（ただし、既存レコードの途中に空フィールドが入っていた場合追加書き込みはそこを初期位置として追加という動作になります。）

#### <テンプレートファイルを使用する>

転送モードが「新規作成」である場合、出力されるファイルに、罫線などで整形したテンプレートを指定することができます。

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1														
2		資料作成日		2012/6/13										
3		資料作成者		山田太郎										
4														
5														
6		年度	四半期	担当者コード	担当者名	地区コード	担当地区	商品コード	商品名	数量	売り上げ金額	粗利益	定価	仕入れ単価
7		2008	1	1	山田太郎	1	横浜	1	ラーメン	30	15000	5000	500	150
8		2002	2	1	山田太郎	1	横浜	2	ラーメン	10	20000	3000	2000	1000
9		2004	1	2	佐藤次郎	2	東京	3	寿司	20	30000	10000	1500	500
10		2008	4	3	鈴木三郎	3	大阪	4	たこ焼き	20	20000	5000	1000	400
11		2002	3	1	山田太郎	1	横浜	2	目玉焼き	5	10000	1500	2000	1000
12		2003	3	3	鈴木三郎	3	大阪	4	焼きそば	10	10000	2500	1000	400

テンプレートファイルを指定せず新規作成した場合の出力結果

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1														
2		資料作成日		2012/6/13										
3		資料作成者		山田太郎										
4														
5														
6		年度	四半期	担当者コード	担当者名	地区コード	担当地区	商品コード	商品名	数量	売り上げ金額	粗利益	定価	仕入れ単価
7		2003	1	1	山田太郎	1	横浜	1	ラーメン	30	15000	5000	500	150
8		2002	2	1	山田太郎	1	横浜	2	ラーメン	10	20000	3000	2000	1000
9		2004	1	2	佐藤次郎	2	東京	3	寿司	20	30000	10000	1500	500
10		2008	4	3	鈴木三郎	3	大阪	4	たこ焼き	20	20000	5000	1000	400
11		2002	3	1	山田太郎	1	横浜	2	目玉焼き	5	10000	1500	2000	1000
12		2003	3	3	鈴木三郎	3	大阪	4	焼そば	10	10000	2500	1000	400
13														

### 整形されたテンプレートファイルを指定した場合の出力結果

以下の項目は <テンプレートファイルを使用する> にチェックを入れた場合に入力することが可能になります。

#### <ファイル名>

書き込みを行う際、使用するテンプレートを指定します。[選択]ボタンをクリックすると、ファイル選択画面が表示されるので、テンプレートに使用する EXCEL ファイルを選択してください。定義変数を指定することが可能ですが、利用する際は下記の注意事項「テンプレートに定義変数を使用する際の注意」をご確認ください。

#### <シート>

<ファイル> で指定したファイルのシート名をプルダウン表示します。

手動で指定することや、定義変数を指定することが可能です。

#### テンプレートに定義変数を使用する際の注意



テンプレートファイル、及びシートは定義実行の際に 1 つのみ指定することができます。定義実行の中で、定義変数を動的に変更し、読み込みレコード毎にテンプレートファイル、およびシートを変更することはできません。データ転送実行前に設定する必要があります。タスクでタスク変数から渡す、QanatExecute で指定する、デフォルト値を指定する場合に設定されます。

#### テンプレートに使用するファイルの注意



テンプレート内に、同じファイルの別シートへの参照を行っているセルがある場合、正常に出力されないことがあります。テンプレートに使用するファイルでは別シートへの参照を行わないようにしてください。

また、xls ファイルをテンプレートとした場合、xlsx ファイルへの出力はできません。同様に xlsx ファイルをテンプレートとした場合は、xls ファイルへ出力はできません。

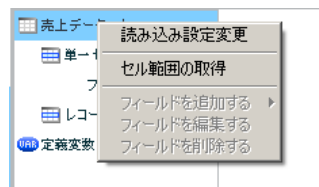
## 詳細設定

<エクセル転送モードを固有に設定する>

大量データ転送モードと通常転送モードを切り替えられます。大量データ転送の場合にはテンプレート機能、追加書き込みが利用できません。詳細は導入マニュアルのエクセルモード（書き込み側）をご確認ください。

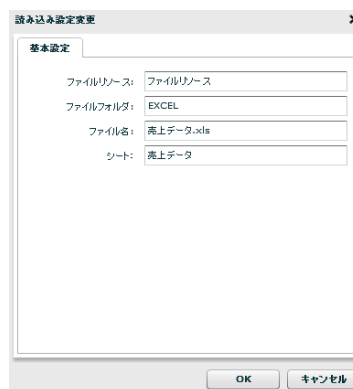
## 2.6.21 EXCEL ファイルを利用したときの各種メニュー

### ・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

#### [読み込み設定の変更]



<ファイルリソース> <ファイルフォルダ> <ファイル名>

データ転送元のファイル名を指定します。手動入力のみ可能となっています。定義変数を指定することが可能です。

<シート>

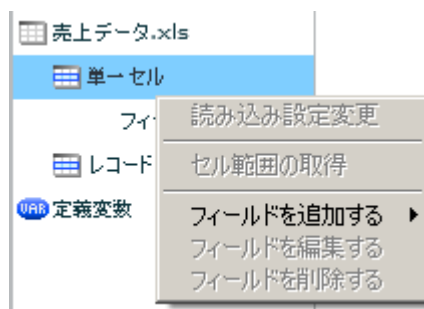
<ファイル名> で指定したファイルのシートを指定します。手動入力のみ可能となっています。定義変数を指定することが可能です。

#### [セル範囲の取得]



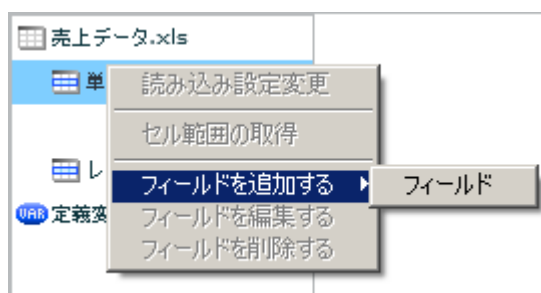
詳細は [2.6.18 EXCEL ファイルの範囲選択](#) を参照してください。

## ・転送元単一セル関連



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]



## &lt;フィールド名&gt;

データ転送元のフィールド名を指定します。この項目は必須項目となっています。単一セルのフィールド内で重複した名前は指定することができません。

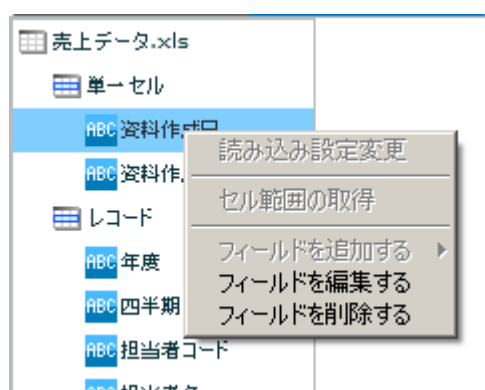
## &lt;フィールドタイプ&gt;

フィールドの型を指定します。指定できる型は 文字列/数値/日付時刻 の3種類となっています。

## &lt;セル位置&gt;

読み込みを行うセルアドレスを指定します。単一のセルを指定するため、「AB51」のような[半角アルファベット]+[半角数値]の形で指定してください。

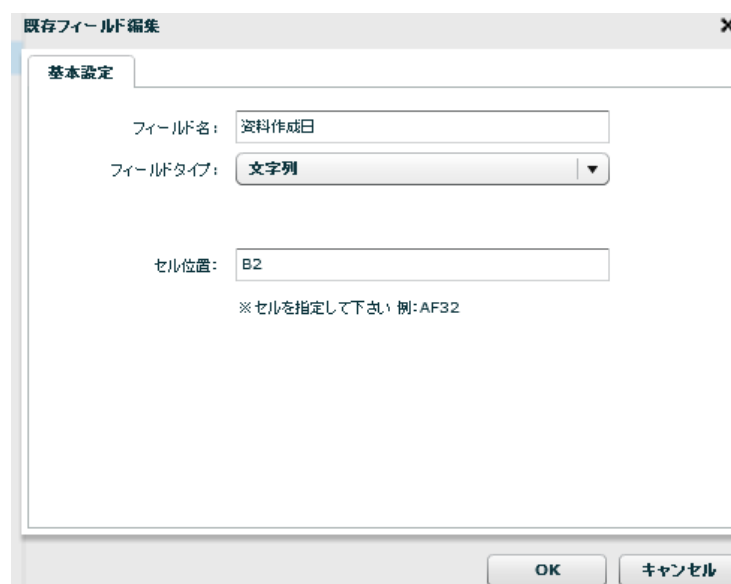
## ・転送元単一セルのフィールド関連



右クリックで表示されるメニュー



## [フィールドを編集する]



## &lt;フィールド名&gt;

データ転送元のフィールド名を指定します。この項目は必須項目となっています。単一セルのフィールド内で重複した名前は指定することができません。

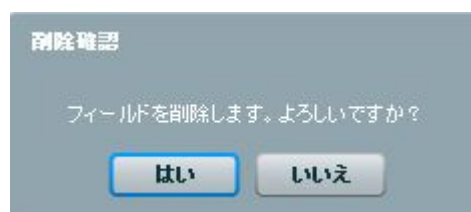
## &lt;フィールドタイプ&gt;

フィールドの型を指定します。指定できる型は 文字列/数値/日付時刻 の3種類となっています。

## &lt;セル位置&gt;

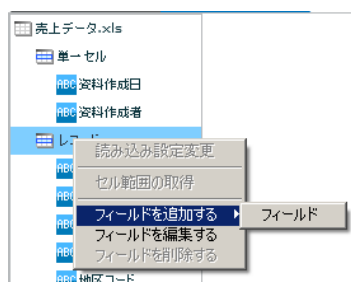
読み込みを行うセルアドレスを指定します。単一のセルを指定するため、「AB51」のような[半角アルファベット]+[半角数値]の形で指定してください。

## [フィールドを削除する]



確認のダイアログが表示されます。削除を行う場合は[はい]ボタンをクリックしてください。

## ・ 転送元レコード関連



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]

## &lt;フィールド名&gt;

データ転送元のフィールド名を指定します。この項目は必須項目となっています。レコードのフィールド内で重複した名前は指定することができません。

## &lt;フィールドタイプ&gt;

フィールドの型を指定します。指定できる型は 文字列/数値/日付時刻 の3種類となっています。

## &lt;セル位置&gt;

読み込みを行うセルアドレスを指定します。

レコードセルを指定するため、「AB」のような[半角アルファベット]、または [15]のような[半角数値]の形で指定してください。指定できる値は後述の「読み込み方向」の値に依存します。読み込み方向が「縦」に指定されていると、列をフィールドとして扱う為、[半角アルファベット]での指定になり、

「横」に指定されていると、行をフィールドとして扱う為、[半角数値]での指定となります。

## &lt; セル範囲 &gt;

レコードとして読み込むように指定されたセルの範囲を表示します。

この画面から値を変更することはできません。

変更する場合は「レコード」フィールド上で右クリックメニュー「フィールドを編集する」をクリックして行ってください。

## &lt; 読み込み方向 &gt;

レコードとして読み込むように指定されたセルの範囲を読み込む方向を指定します。デフォルトでは「縦」が指定されています。

縦 EXCEL ファイルの列をフィールドとして扱います。

横 EXCEL ファイルの行をフィールドとして扱います。

## [フィールドを編集する]



## &lt; セル範囲 &gt;

レコードとして読み込むように指定されたセルの範囲を表示します。指定は[B7:N12]のように[{開始セル}:{終了セル}]の形式で入力します。

## &lt; 読み込み方向 &gt;

レコードとして読み込むように指定されたセルの範囲を読み込む方向を指定します。デフォルトでは「縦」が指定されています。

縦 EXCEL ファイルの列をフィールドとして扱います。

横 EXCEL ファイルの行をフィールドとして扱います。

## &lt; 指定されたセル範囲を超えてもレコードが空になるまで読み込む &gt;

前述の<セル範囲>で指定した範囲以降にレコード情報が記述されていた場合の動作を指定します。

「レコードが空である」の判定は、「レコードに含まれるフィールドの値がすべて空欄である」となります。デフォルトは「いいえ」が指定されています。

いいえ 指定したセル範囲の値のみを読み取ります。

はい 指定したセル範囲以降にレコード情報が含まれている場合、レコード情報がなくなるまで読みを行います。

#### < 指定されたセル範囲内でレコードが空のとき >

前述の<セル範囲>で指定した範囲内に、空のレコードが記述されていた場合の動作を指定します。

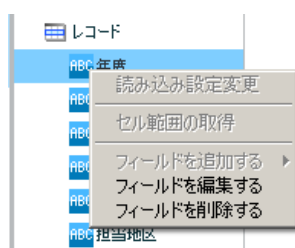
「レコードが空である」の判定は、「レコードに含まれるフィールドの値がすべて空欄である」となります。デフォルトは「読込」が指定されています。

読込 空のレコードをレコードとして扱い、処理を続行します。

スキップ 空のレコードをスキップし、次のレコードの読み込みを行います。「読込」と異なり、空のレコードは出力されません。

終了 空のレコードを読み取った時点で処理を終了します。空のレコードは出力されません。

### ・転送元レコードのフィールド関連



右クリックで表示されるメニュー

[フィールドを編集する]

既存フィールド編集

基本設定

フィールド名: 年度

フィールドタイプ: 文字列

セル位置: B

※列を指定して下さい 例: AF

セル範囲: B7:N7

読み込み方向: 縦

OK キャンセル

## &lt;フィールド名&gt;

データ転送元のフィールド名を指定します。この項目は必須項目となっています。レコードのフィールド内で重複した名前は指定することができません。

## &lt;フィールドタイプ&gt;

フィールドの型を指定します。指定できる型は 文字列/数値/日付時刻 の3種類となっています。

## &lt;セル位置&gt;

読み込みを行うセルアドレスを指定します。

レコードセルを指定するため、「AB」のような[半角アルファベット]、または [15]のような[半角数値]の形で指定してください。

指定できる値は前述の「読み込み方向」の値に依存します。読み込み方向が「縦」に指定されていると、列をフィールドとして扱う為、

[半角アルファベット]での指定になり、「横」に指定されていると、行をフィールドとして扱う為、[半角数値]での指定となります。

## &lt;セル範囲&gt;

レコードとして読み込むように指定されたセルの範囲を表示します。この画面から値を変更することはできません。変更する場合は「レコード」フィールド上で右クリックメニュー「フィールドを編集する」をクリックして行ってください。

## &lt;読み込み方向&gt;

レコードとして読み込むように指定されたセルの範囲を読み込む方向を指定します。デフォルトでは「縦」が指定されています。

縦 EXCEL ファイルの列をフィールドとして扱います。

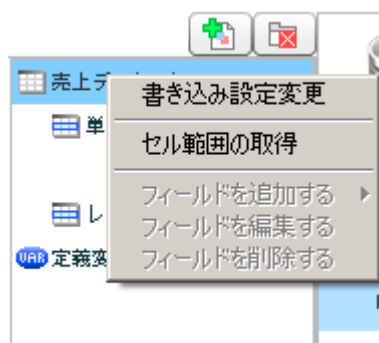
横 EXCEL ファイルの行をフィールドとして扱います。

## [フィールドを削除する]



確認のダイアログが表示されます。削除を行う場合は[はい]ボタンをクリックしてください。

## ・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

## [書き込み設定の変更]

## &lt;ファイル名&gt;

データ転送先のファイル名を指定します。手動入力のみ可能となっています。定義変数を指定することが可能です。

## &lt;シート&gt;

<ファイル名>で指定したファイルのシートを指定します。手動入力のみ可能となっています。定義変数を指定することが可能です。

## [書き込み設定の変更]

詳細設定



<エクセル転送モードを固有に設定する>

チェックを入れることで、該当のマッパー内でのみエクセル転送モードを固有に設定することができます。

- ・大量データ転送 ... エクセルモード 1 (ON) で動作
- ・通常転送 ... エクセルモード 0 (OFF) で動作

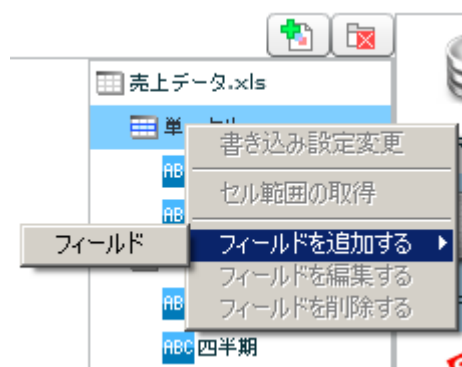
エクセルモードの説明については、「導入の手引き WinTomcat」の 10.1.8 章  
または、「導入の手引き LinTomcat」の 10.1.6 章 をご確認ください。

[セル範囲の取得]



詳細は [2.6.18 EXCEL ファイルの範囲選択](#) を参照してください。

## ・転送先単一セル関連



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]

## &lt;フィールド名&gt;

データ転送先のフィールド名を指定します。この項目は必須項目となっています。単一セルのフィールド内で重複した名前は指定することができません。

## &lt;フィールドタイプ&gt;

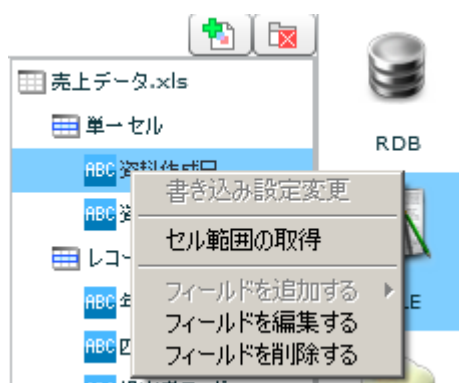
フィールドの型を指定します。指定できる型は 文字列/数値/日付時刻 の3種類となっています。セルの表示形式を指定しているものではありません。

## &lt;セル位置&gt;

読み込みを行うセルアドレスを指定します。単一のセルを指定するため、「AB51」のような[半角アルファベット]+[半角数値]の形で指定してください。

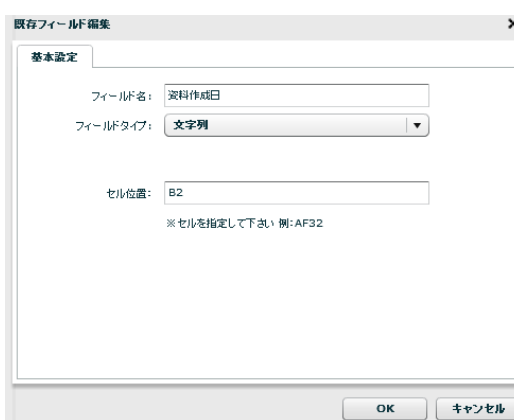


## ・転送先単一セルのフィールド関連



右クリックで表示されるメニュー

## [フィールドを編集する]



## &lt;フィールド名&gt;

データ転送先のフィールド名を指定します。この項目は必須項目となっています。単一セルのフィールド内で重複した名前は指定することができません。

## &lt;フィールドタイプ&gt;

フィールドの型を指定します。指定できる型は 文字列/数値/日付時刻 の3種類となっています。セルの表示形式を指定していません。

## &lt;セル位置&gt;

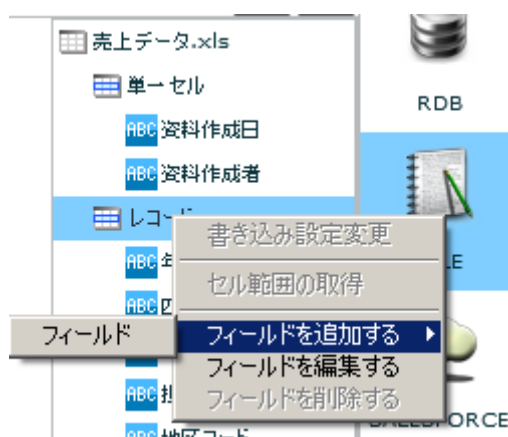
書き込みを行うセルアドレスを指定します。単一のセルを指定するため、「AB51」のような[半角アルファベット]+[半角数値]の形で指定してください。

## [フィールドを削除する]



確認のダイアログが表示されます。削除を行う場合は[はい]ボタンをクリックしてください。

## ・ 転送先レコード関連



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]

## &lt;フィールド名&gt;

データ転送先のフィールド名を指定します。この項目は必須項目となっています。レコードのフィールド内で重複した名前は指定することができません。

## &lt;フィールドタイプ&gt;

フィールドの型を指定します。指定できる型は 文字列/数値/日付時刻 の3種類となっています。

## &lt;セル位置&gt;

書き込みを行うセルアドレスを指定します。レコードセルを指定するため、「AB」のような[半角アルファベット]、または [15]のような[半角数値]の形で指定してください。

指定できる値は後述の「書き込み方向」の値に依存します。書き込み方向が「縦」に指定されていると、列をフィールドとして扱う為、

[半角アルファベット]での指定になり、「横」に指定されていると、行をフィールドとして扱う為、[半角数値]での指定となります。

## &lt;セル範囲&gt;

レコードとして書き込むように指定されたセルの範囲を表示します。この画面から値を変更することはできません。

変更する場合は「レコード」フィールド上で右クリックメニュー「フィールドを編集する」をクリックして行ってください。

## &lt;書き込み方向&gt;

レコードとして書き込むように指定されたセルの範囲の書き込み方向を指定します。デフォルトでは「縦」が指定されています。

縦 EXCEL ファイルの列をフィールドとして扱います。

横 EXCEL ファイルの行をフィールドとして扱います。

## [フィールドを編集する]



## &lt;セル範囲&gt;

レコードとして書き込むように指定されたセルの範囲を表示します。指定は[B7:N12]のように[{開始セル}:{終了セル}]の形式で入力します。

## &lt;書き込み方向&gt;

レコードとして書き込むように指定されたセルの範囲の書き込み方向を指定します。デフォルトでは「縦」が指定されています。

縦 EXCEL ファイルの列をフィールドとして扱います。

横 EXCEL ファイルの行をフィールドとして扱います。

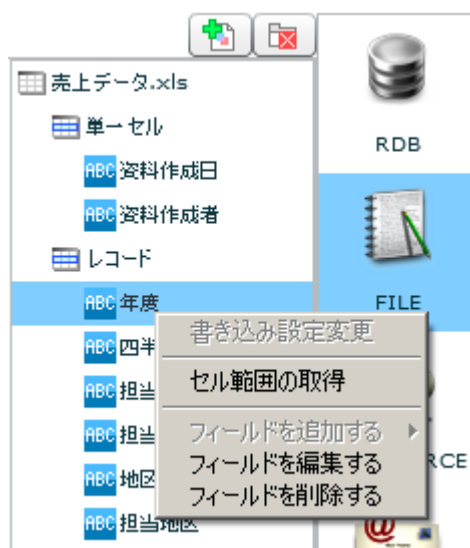
## &lt;指定されたセル範囲を超えても書き込む&gt;

前述の<セル範囲>で指定した範囲以上にレコード情報が読み込まれた場合の動作を指定します。デフォルトは「いいえ」が指定されています。

いいえ 指定したセル範囲で書き込みを行います。オーバーしたレコードは書き込みが行われません。

はい 指定したセル範囲以上にレコード情報が読み込まれた場合、レコード情報がなくなるまで書き込みを行います。

## ・転送先レコードのフィールド関連



右クリックで表示されるメニュー

[フィールドを編集する]

## &lt;フィールド名&gt;

データ転送先のフィールド名を指定します。この項目は必須項目となっています。レコードのフィールド内で重複した名前は指定することができません。

## &lt;フィールドタイプ&gt;

フィールドの型を指定します。指定できる型は 文字列/数値/日付時刻 の3種類となっています。

## &lt;セル位置&gt;

書き込みを行うセルアドレスを指定します。レコードセルを指定するため、「AB」のような[半角アルファベット]、または [15]のような[半角数値]の形で指定してください。

指定できる値は前述の「書き込み方向」の値に依存します。書き込み方向が「縦」に指定されていると、列をフィールドとして扱う為、

[半角アルファベット]での指定になり、「横」に指定されていると、行をフィールドとして扱う為、[半角数値]での指定となります。

## &lt;セル範囲&gt;

レコードとして書き込むように指定されたセルの範囲を表示します。この画面から値を変更することはできません。

変更する場合は「レコード」フィールド上で右クリックメニュー「フィールドを編集する」をクリックして行ってください。

## &lt;書き込み方向&gt;

レコードとして書き込むように指定されたセルの範囲の書き込む方向を指定します。デフォルトでは「縦」が指定されています。

縦 EXCEL ファイルの列をフィールドとして扱います。

横 EXCEL ファイルの行をフィールドとして扱います。

## [フィールドを削除する]



確認のダイアログが表示されます。削除を行う場合は[はい]ボタンをクリックしてください。

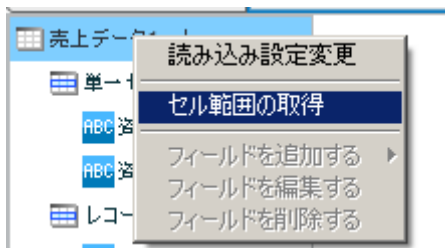
## 2.6.22 EXCEL ファイルの範囲選択

転送元、転送先に EXCEL ファイルを指定した際に、読み込み、書き込みを行うセルの指定を EXCEL ファイルを参照しながら行うことが可能です。

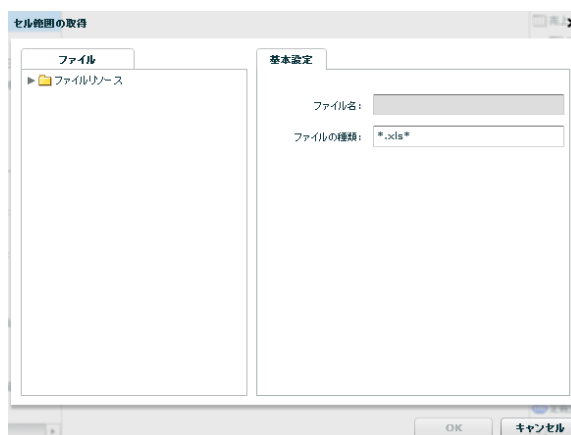
本機能を使用する際には、本製品が動作している環境に、「Microsoft Office Excel」が導入されている必要があります。

### ・セル取得までの操作

転送元または転送先上で、右クリックメニューから「セル範囲の取得」をクリックします。



EXCEL ファイルを選択する画面が表示されるので、ファイルを選択し、[OK]ボタンをクリックします。



[OK]ボタンをクリックすると、選択したファイルの本製品動作環境にダウンロードして表示します。そのため、サイズが小さいファイルを選択することを推奨します。



また、Excel を開くため FlowDesigner を管理者権限で起動しておく必要があります。

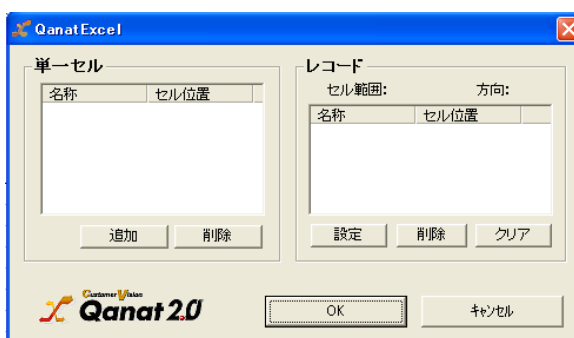
「ダウンロードが完了しました」というメッセージが表示されるので[OK]ボタンをクリックします。指定したファイルと、QanatExcel が起動します。





## ・ QanatExcel の画面について

QanatExcel は以下のように表示されます。



各項目は以下の通りです。

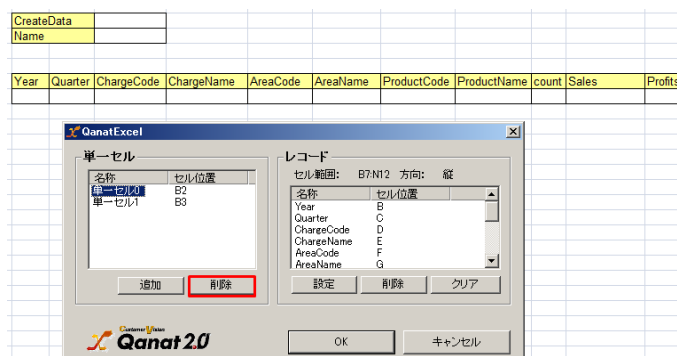
- 単一セル      単一セルとして扱うセルを表示します。  
名称はフィールド名として表示されます。名称は手動で変更することが可能です。
- レコード      レコードとして扱うセルを表示します。  
名称はフィールド名として表示されます。名称は手動で変更することが可能です。

## ・ 単一セルの取得、および削除方法

表示されている EXCEL ファイルのセルを選択状態にして、

QanatExcel の[追加]ボタンをクリックします。

名称とセル位置が追加されます。名称は手動で変更することが可能です。



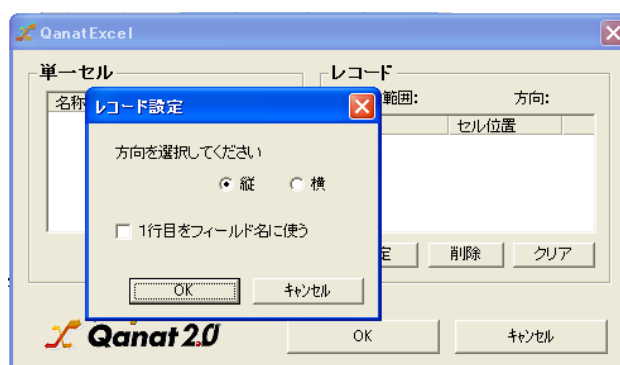
削除する場合は、削除するセルを選択状態にして、[削除]ボタンをクリックします。

## ・レコードの取得、および削除方法

表示されている EXCEL ファイルで、本製品で読み込み（書き込み）を行う範囲を選択状態にします。

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1														
2		CreateData												
3		Name												
4														
5														
6	Year	Quarter	ChargeCode	ChargeName	AreaCode	AreaName	ProductCode	ProductName	count	Sales	Profits	Price	Purchase	
7														
8														

QanatExcel の [設定] ボタンをクリックすると、「レコード設定」画面が表示されます。



各項目は以下のようにになっています。設定が正しいことを確認して、[OK] ボタンをクリックします。

**方向の選択（縦、横）** 指定した範囲の列をフィールドとするか、行をフィールドとするかを選択します。例として、B2:D7 の範囲を縦で読み込んだ場合、列 B,C,D がフィールドとして扱われます。

**1 行目をフィールド名に使う** 指定した範囲の 1 レコード目のデータをフィールド名として読み込みを行います。チェックを入れた場合、1 レコード目は読み込み対象外となります。  
例として、B2:D7 の範囲を縦で読み込んだ場合、B2,C2,D2 の値がフィールド名として使用されます。

削除する場合は、削除するセルを選択状態にして、[削除] ボタンをクリックします。  
また、[クリア] ボタンをクリックすることですべてのレコード情報を削除することが可能です。

Windows 7 では EXCEL の範囲選択を行おうとして「ファイルの読み込みエラーが発生しました」とエラー表示され、エクセルが立ち上がらない場合があります。



範囲選択に使用するエクセルファイルは以下のフォルダに一時ファイルとして作成されますので、フォルダに対してユーザーの権限を追加してください。

” インストールディレクトリ/FlowDesigner/resource/bin/

また、出力先に指定した EXCEL が開かれている（別プロセスで使用している）場合にはエラーとなります。



## 2.6.23 EXCEL ファイル転送におけるのループ処理

EXCEL ファイル転送の場合はループの基準をレコード要素として動作します。

## 2.6.24 SALESFORCE アダプターの仕様について

### 使用している API について

本製品の Salesforce アダプターは、Salesforce.com より公開されている以下の API サービスを利用し、グラフィカルな画面での処理設計およびデータの入出力を行う機能を提供しています。

SOAP API Version 37.0

- ・ query()
- ・ getUpdated()

BULK API Version 37.0

- ・ insert()
- ・ update()
- ・ upsert()
- ・ delete()

### 扱えるデータについて

本製品の Salesforce 連携アダプターは、標準オブジェクトおよびカスタムオブジェクトを対象にデータの入出力を行います。

標準オブジェクトは Salesforce に標準で含まれるデータベーステーブルです。

取引先、取引責任者、リードや商談などの種類があります。

カスタムオブジェクトは Salesforce ユーザーが任意に作成するデータベーステーブルです。存在するカスタムオブジェクトは、Salesforce 環境より設定の画面を表示し、アプリケーションの設定 カスタムオブジェクトの管理をする  
よりご確認いただけます。

( Salesforce のエディションやバージョンにより異なる可能性があります。 )

### Salesforce のサンドボックス環境について

V1.14.4 より Salesforce のサンドボックス環境に対応しました。

サンドボックス環境への接続方法については別冊 「CV 管理ツールマニュアル」の「salesforce リソース定義管理」をご確認ください。

## 2.6.25 SALESFORCE 選択

### SalesForce アダプターを使用する前に

ご利用いただく際には、以下の設定が必須となります。ご注意ください。

・ Windows の場合

`${Tomcat 導入ディレクトリ}/bin/Tomcat7w.exe` を開きます

「Java」タブの「Java Options」：の中に以下を追記します。

`-Dhttps.protocols=TLSv1.1,TLSv1.2`

上記設定をしたのち、Tomcat を再起動してください。

・ Linux の場合

`${Tomcat 導入ディレクトリ}/bin/setenv.sh` を開きます（ない場合には作成します）

JAVA\_OPTS として、以下を記述します。

`-Dhttps.protocols=TLSv1.1,TLSv1.2`

以下、設定例です。

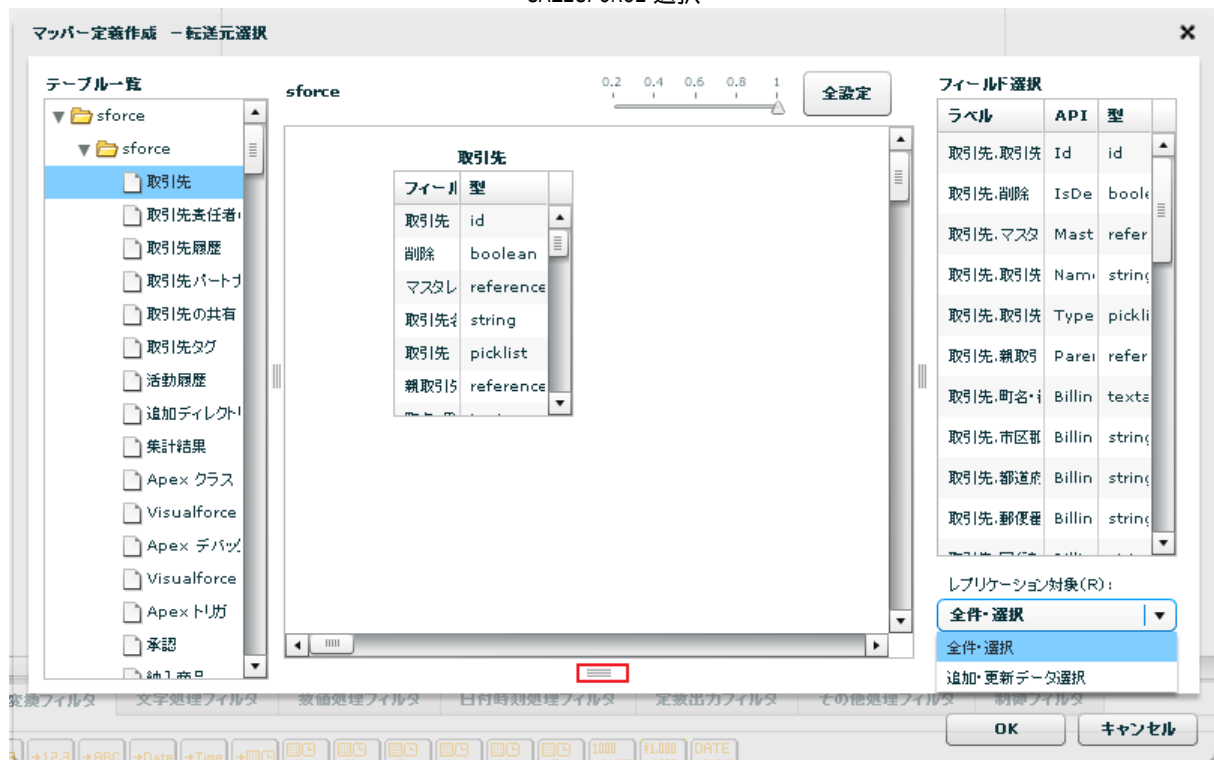
```
#!/bin/sh
JAVA_OPTS="-Dhttps.protocols=TLSv1.1,TLSv1.2"
export JAVA_OPTS
```

上記設定をしたのち、Tomcat を再起動してください。

データ転送元に SALESFORCE を選択します。



SALESFORCE 選択



転送元選択

検索条件や並び替えは赤線で囲った部分をクリックし上にもっていくと指定画面が現れます。

追加・更新データ選択時は期間指定画面が現れます。



全件選択時の転送元選択画面

【**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(カスタムオブジェクト含む)を選択します。

【**テーブル配置領域**】 フィールドを選択したいテーブルをここに配置します。

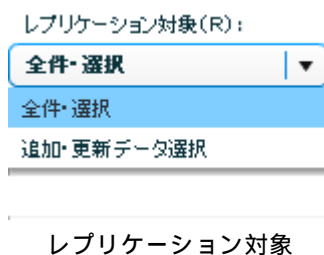
【**フィールド選択**】 テーブル一覧にて選択されたテーブルのフィールド一覧を表示します。  
フィールド選択を行うにはMapper定義上のフィールドを  
ドラッグ&ドロップや画面上の全選択ボタン押下することで追加されます。

ラベル                      Salesforce ラベル名

API                          SalesforceAPI 参照名

型                            データ型

【**レプリケーション対象**】      Salesforce データ読み取り対象を選択します。



レプリケーション対象

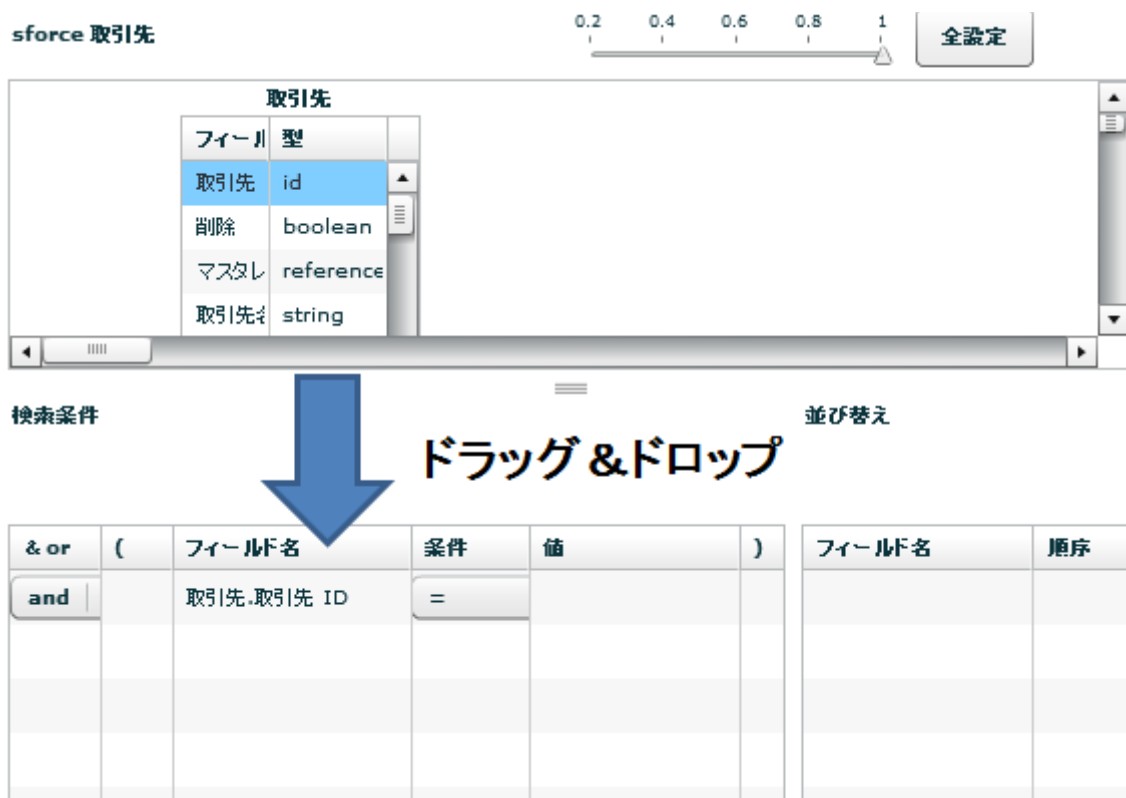
全件選択 フィールド一覧に配置したフィールドの全データ読み取りを行います。

追加・更新データ選択 指定した期間に Salesforce で追加・更新のあったデータから読み取りを行います。

( 追加・更新データ選択において読み取れる最大期間は当日から 30 日前までです。 )

**【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。



検索条件使用フィールド指定

&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。

& or	(	フィールド名	条件	値	)
and		取引先.取引先 電話	文字含む	090	
and	(	取引先.都道府県(請求先)	=	神奈川県	
or		取引先.都道府県(請求先)	=	埼玉県	)

検索条件例

上図のように検索した場合、電話番号が 090 を含みかつ都道府県が神奈川県、又は埼玉県のデータを抽出します。

**【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。

sforce 取引先 0.2 0.4 0.6 0.8 1 全設定

取引先	
フィールド	型
取引先	id
削除	boolean
マスタレ	reference

ドラッグ&ドロップ

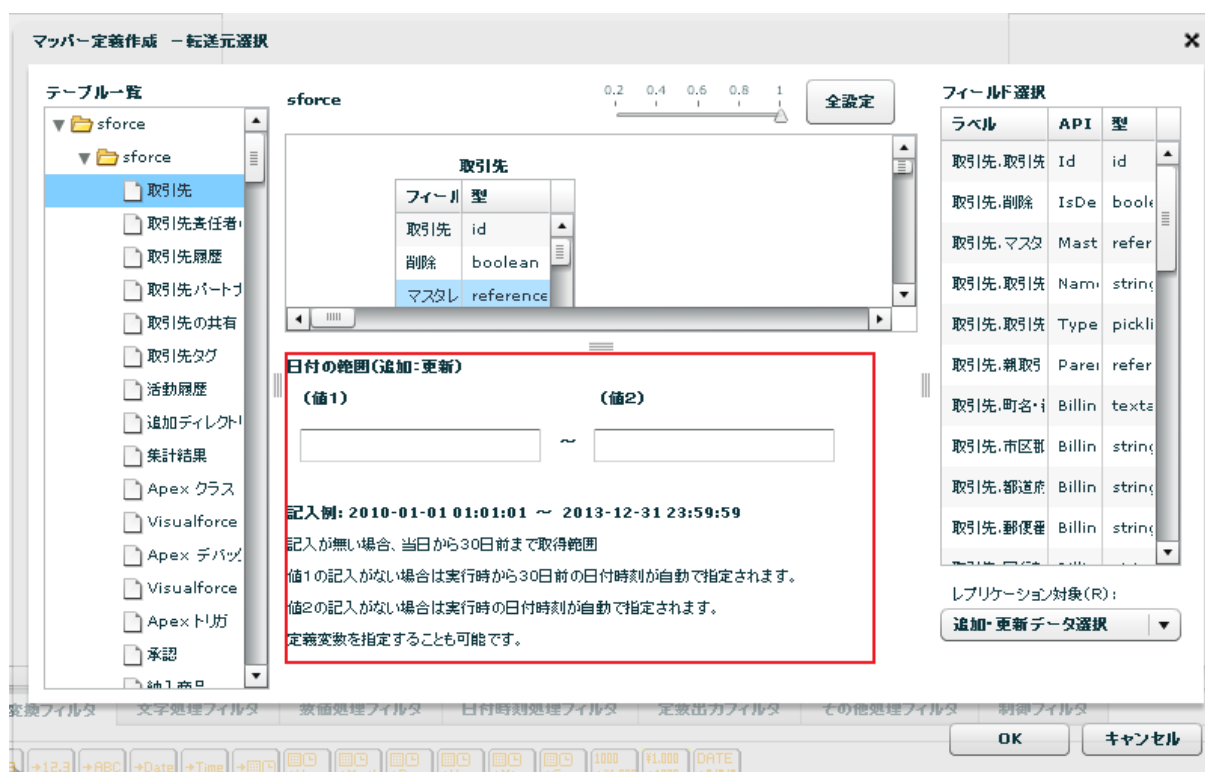
並び替え

& or	(	フィールド名	条件	値	)	フィールド名	順序
						取引先.取引先 ID	昇順 ▼

並び替えフィールド指定

フィールド名      並び替えを行うフィールド名です。

順序              昇順、降順から選択します。



追加・更新データ選択時の転送元選択画面

追加・更新データ選択時は上図のように検索条件並び替え指定領域が日付の範囲指定領域に切り替わります。

記入例にあるフォーマット (yyyy-MM-dd HH:mm:ss) で日付を値 1 ~ 2 に入力します。

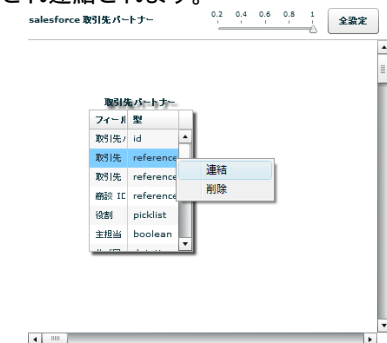
値 1 に記入がない場合、デフォルトで定義実行日の 30 日前の日付時刻が指定されます。(From 30 日前)

値 2 に記入がない場合、デフォルトで定義実行日の日付時刻が指定されます。(To 定義実行日)

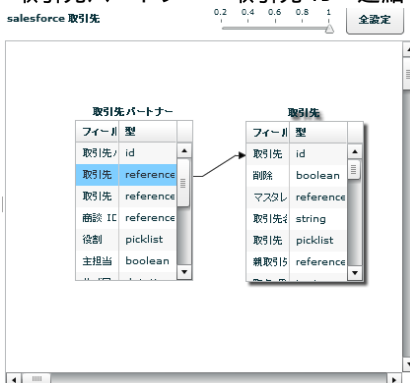


## 参照関係にある複数のオブジェクトのデータ取得

のテーブル配置領域において、型が reference のフィールドを右クリック 連結をクリックすることで参照関係にあるテーブルが表示され連結されます。



### 取引先パートナー・取引先 ID 連結



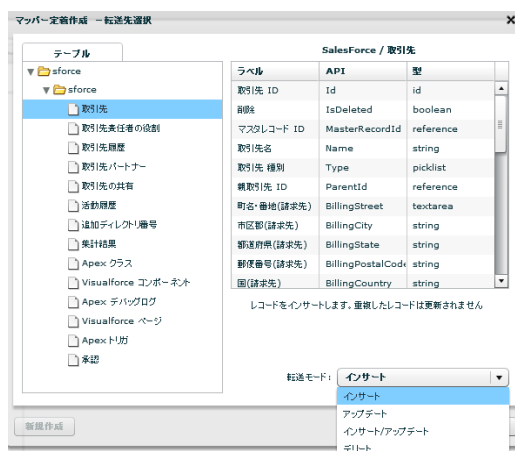
### 取引先パートナーテーブルと取引先テーブルの連結



### フィールド選択に両テーブルのフィールドをドラッグ&ドロップ

このように、参照関係をもったテーブルの場合どちらのフィールドも入力データとして選択することが可能です。

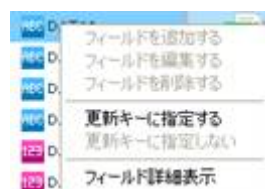
## データ転送先に SALESFORCE を選択します。



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
カラム一覧	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
ラベル	Salesforce ラベル名
API	SalesforceAPI 参照名
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。
インサート	データを追加だけします。
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	データをキーにより、追加または更新します。
デリート	データを全件削除します。

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

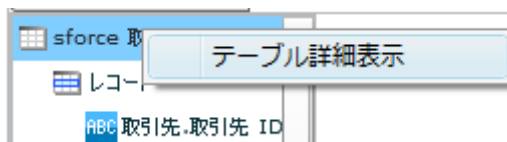
アップデートキーは、フィールドを選択状態にし、右クリックで表示されるメニュー[更新キーに指定する]にて設定可能です。



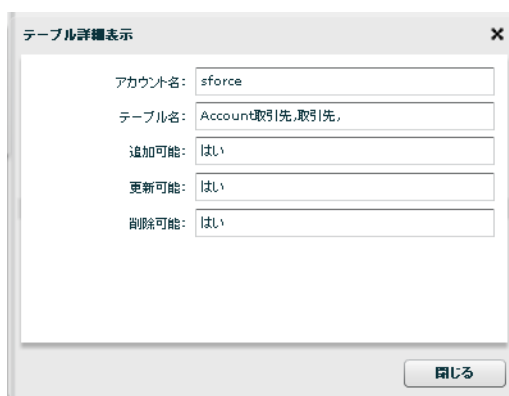
更新キー指定メニュー

## 2.6.26 SALESFORCE を利用したときの各種メニュー

### ・転送元テーブル関連

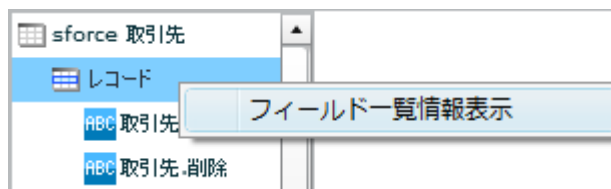


転送元テーブルを選択時に右クリックで表示されるメニュー

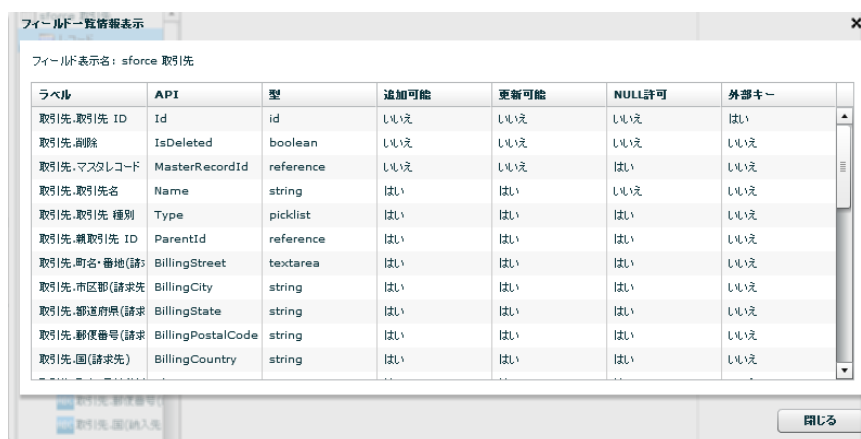


テーブル詳細表示

### ・転送元レコード関連

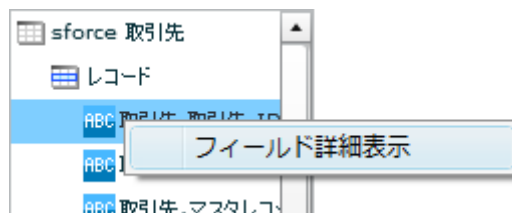


転送元レコードを選択時に右クリックで表示されるメニュー

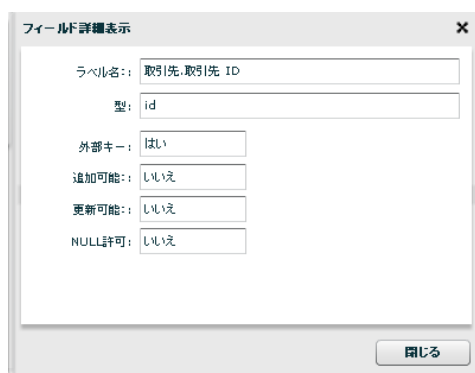


フィールド一覧情報表示

### ・ 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



フィールド詳細表示

### ・ 転送先テーブル関連

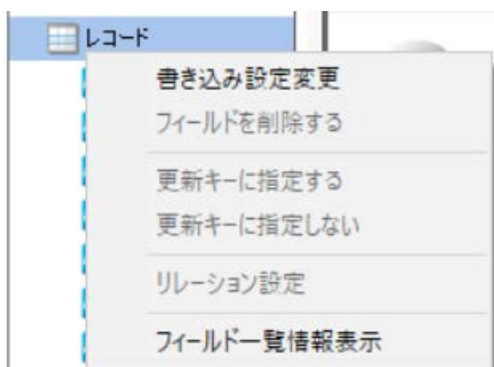


転送先テーブルを選択時に右クリックで表示されるメニュー



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

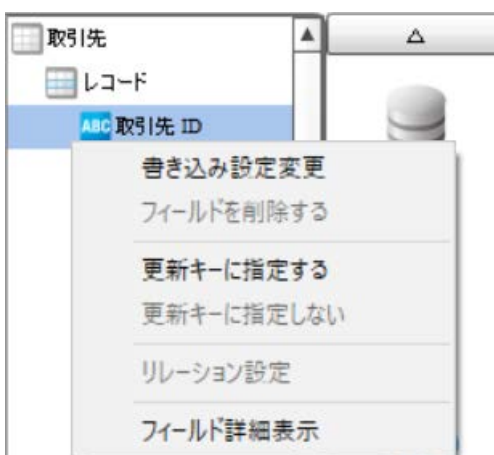
フィールド一覧情報表示

フィールド表示名: 取引先責任者の役割

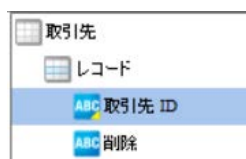
ラベル	API	型	追加可能	更新可能	NULL許可	外部キー
取引先責任者の役割 ID	Id	id	はい	はい	はい	はい
取引先 ID	AccountId	reference	はい	はい	はい	はい
取引先責任者 ID	ContactId	reference	はい	はい	はい	はい
ロール	Role	picklist	はい	はい	はい	はい
主理由	IsPrimary	boolean	はい	はい	はい	はい
作成日	CreatedDate	datetime	はい	はい	はい	はい
作成者 ID	CreatedById	reference	はい	はい	はい	はい
最終更新日	LastModifiedDate	datetime	はい	はい	はい	はい
最終更新者 ID	LastModifiedById	reference	はい	はい	はい	はい
System Modstamp	SystemModstamp	datetime	はい	はい	はい	はい
削除	IsDeleted	boolean	はい	はい	はい	はい

フィールド一覧情報表示

## ・ 転送先フィールド関連

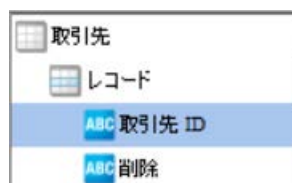


転送先フィールドを選択時に右クリックで表示されるメニュー



「更新キーに指定する」を選択した場合のフィールド表示

( フィールド名左側のアイコンの右下に、更新キーを表すマークが付加されます )



「更新キーに指定しない」を選択した場合のフィールド表示

( 更新キーを表すマークが除去されます )

フィールド詳細表示 ×

ラベル名:	<input type="text" value="取引先責任者の役割 ID"/>
API:	<input type="text" value="Id"/>
型:	<input type="text" value="id"/>
外部キー:	<input type="text" value="はい"/>
追加可能:	<input type="text" value="いいえ"/>
更新可能:	<input type="text" value="いいえ"/>
NULL許可:	<input type="text" value="いいえ"/>

フィールド詳細表示

### ・転送先フィールド関連（リレーション設定について）

本機能は 以下の条件がそろっている場合に設定が可能です。

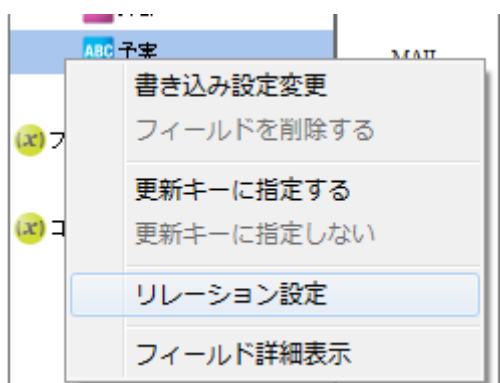
- ・転送モードが " インサート/アップデート " であること
- ・フィールド（カラム）の型が、参照型の " 参照関係 " 型であること
- ・更新可能フィールドであること（フィールド詳細表示 にて確認可能です）

こちらの機能をご利用いただきますと、SalesForce の " 外部 ID " を利用して参照関係型フィールドの更新が可能となります。

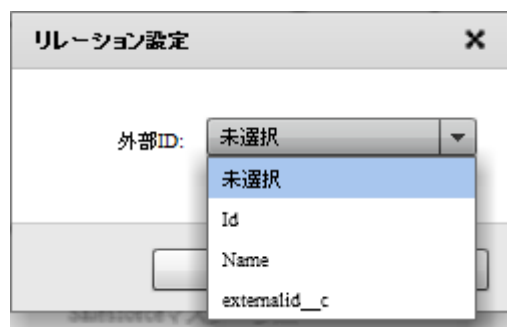
設定手順は以下となります。

書き込み先の " 参照関係 " フィールドの右クリックメニューを開きます

" リレーション設定 " を選択します



表示されるダイアログに入力する外部 ID を一覧から選択します



未選択の場合は、カスタムオブジェクト ID (Id) として扱います。

本機能をご利用いただく注意点



- ・書きこむ際（結線するデータ）の型は、指定した外部 ID と一致させる必要があります
- ・エラーが発生した場合は、次のレコードへ処理が移ります

## 2.6.27 メール送信設定

### メール送信をデータ転送先に選択



メール送信選択



メールサーバー選択

### メールサーバー

使用するメールサーバーを選択します。

### 基本設定

#### <メールサーバー>

<ファイル> で指定したファイルの名称を表示します。

#### <添付ファイルフォルダ>

メールに添付するファイルのフォルダを指定します。

選択できるフォルダは前述のファイルフォルダからとなりますので、

事前にファイルフォルダを作成する必要があります。

#### <エンコード>

メールの文字エンコーディングを指定します。

手動で入力することも可能です。

#### <件名の文字エンコードを別に指定する>

件名と本文の文字エンコーディングを別に指定する場合に使用します。

#### <件名のエンコード>

<件名の文字エンコードを別に指定する> にチェックを入れた際に入力可能となります。

指定した文字エンコーディングで件名を表示し、本文は<エンコード>で指定した

文字コードが使用されます。



## 2.6.28 メール送信を選択したときの各種メニュー

メール送信を選択した場合、添付ファイルフォルダーを指定しているかどうかで表示されるフィールドに違いが発生しますが、それ以外でレコードおよびフィールドに変更を加えることはできません。以下にフィールドの説明を記述します。



送信元	メールの送信元の指定を行うフィールドです。必須項目で、空白値、及び結線しない場合、実行時にエラーとなります。半角英数字が入力可能で、日本語などの2バイト文字を使用することはできません。
To	送信先アドレスの指定を行うフィールドです。必須項目で、カンマ区切りで複数のメールアドレスを指定することが可能です。
CC	送信メールのCC(カーボンコピー)の指定を行うフィールドです。カンマ区切りで複数のメールアドレスを指定することが可能です。
BCC	送信メールのBCC(ブラインドカーボンコピー)の指定を行うフィールドです。カンマ区切りで複数のメールアドレスを指定することが可能です。
ContentType	送信メールのコンテンツタイプを指定するフィールドです。「text/plain」などの値が指定可能です。結線しなかった場合や、空白値を指定した場合は「text/html」として動作します。
返信先	送信メールの返信先を指定するフィールドです。送信元と異なる値を設定することが可能です。カンマ区切りで複数のメールアドレスを指定することが可能です。
件名	送信メールの件名を指定するフィールドです。
本文	送信メールの本文を指定するフィールドです。
添付ファイル	送信メールに添付するファイルを指定するフィールドです。メールサーバー選択時に、「添付ファイルフォルダー」を指定した場合のみ表示されます。フィールドにはファイル名、または指定した添付ファイルフォルダーからの相対パスを指定します。ファイルパスを指定する際の区切り文字は、本製品サーバーが動作している環境によって以下のようになります。 Windows 半角スラッシュ(/) または 半角円マーク(/) Linux 半角スラッシュ(/)

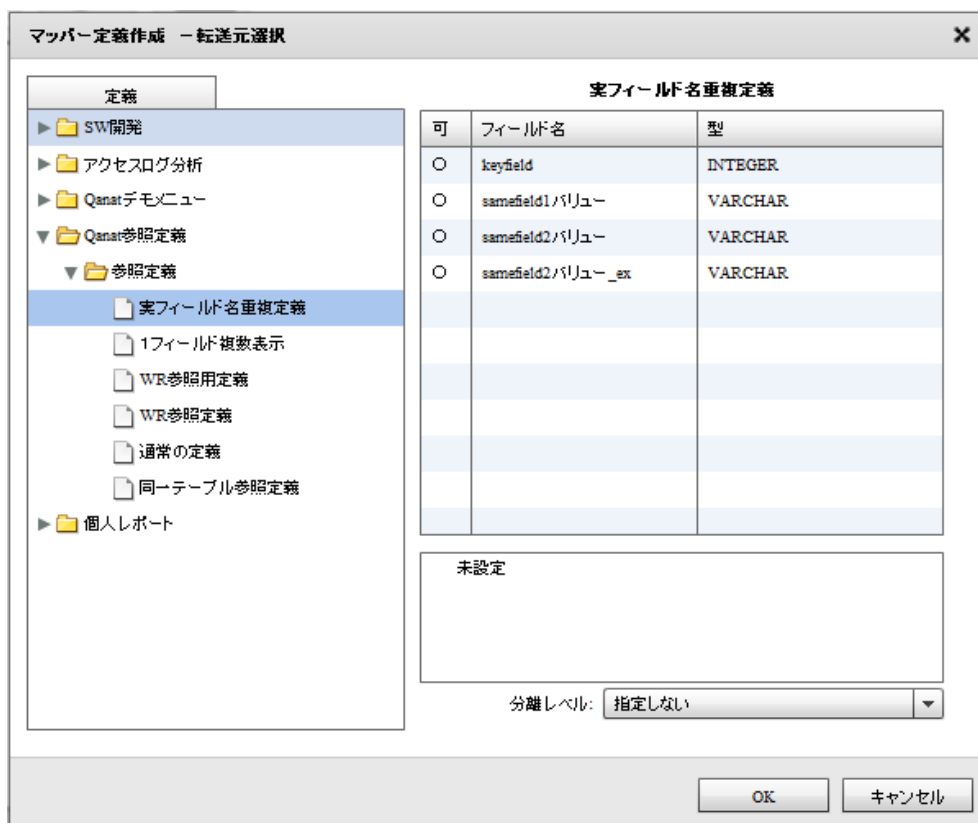
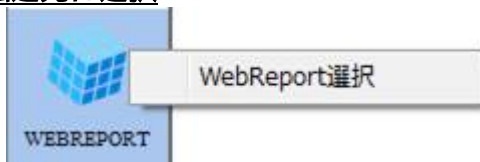
### メールアダプタの動作の注意点



メールアダプタを使用した定義は、メールサーバーにメールの送信要求を正常に行なった時点で正常終了となります。そのため、送信先が存在しないアドレスであったなどで、正常にメール送信が行なわれなかった場合も定義は正常終了します。メール送信定義を使用する場合は、メールサーバー側でエラーなどが発生していないかを確認するように運用していただくと未然にトラブルを回避することができます。

## 2.6.29 WebReport 選択

### WebReport をデータ転送元を選択



#### WebReport 定義選択

**定義一覧** 入力元に指定する WebReport 参照定義を選択します

**フィールド一覧** 定義一覧にて選択された WebReport 参照定義のフィールド情報

**可** フィールドが参照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

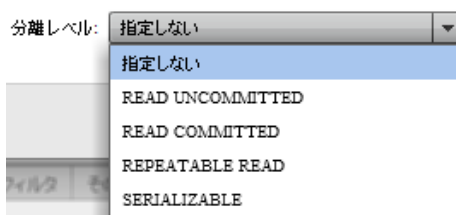
**型** データ型

**分離レベル** 転送元への分離レベルを選択します。



本製品で取扱い可能な WebReport 定義は “input 関数” が使用されていない WebReport 定義のみとなります。

Input 関数については WebReport のマニュアルをご覧ください。



転送モード

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。

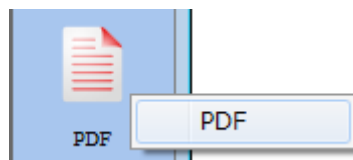
作成したい処理に適した分離レベルの設定を行ってください。



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。

## 2.6.30 PDF ファイル選択

### PDF ファイルをデータ転送先に選択



PDF ファイル選択

PDF 出力設定画面

#### <ファイルリソース>

データ転送先のファイルリソースを指定します。登録されている Qanat Server と Amazon S3 のファイルリソースが表示されます。

#### <ファイルフォルダ>

データ転送先のファイルフォルダを指定します。登録されているファイルフォルダが表示されます。

#### <ファイル>

データ転送先の PDF ファイル名を指定します。手動入力のみ可能となっています。

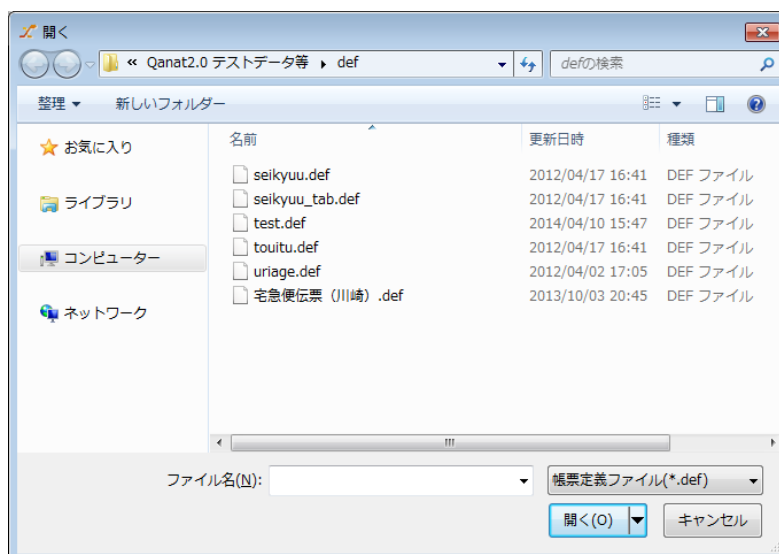
#### <帳票定義ファイル>

出力する PDF ファイルの帳票定義ファイル (def 形式ファイル) を指定します。[ファイル選択ボタン] をクリックしてファイル選択画面を表示し、使用する帳票定義ファイルを選択してください。

#### <使用するフォームの設定>

出力する PDF ファイルの出力方法を指定します。[編集]ボタンをクリックして PDF 出力設定画面を表示し、出力方法を選択してください。

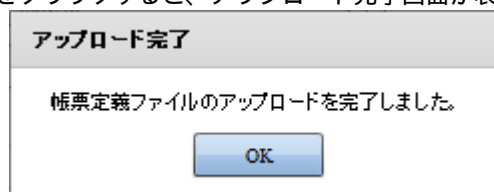
- ❗ PDF ファイルを出力するためには、帳票定義ファイルが必須となります。  
また、PDF アダプターで使用できる帳票定義ファイルは Unicode 版のみとなります。
- ❗ 帳票定義ファイルの作成には、PrintPro2.0 帳票設計が必要となります。  
作成方法に関しましては、帳票設計\_ご利用の手引き.pdf をご覧ください。
- ❗ Amazon S3 のリソースを選択するためには、ライセンスの登録が必要となります。  
ライセンスが登録されていない場合、ファイルリソースの一覧に表示されません。



ファイル選択画面

帳票定義ファイルを選択すると、[アップロード]ボタンがクリックできるようになります。

[アップロード]ボタンをクリックすると、アップロード完了画面が表示されます。

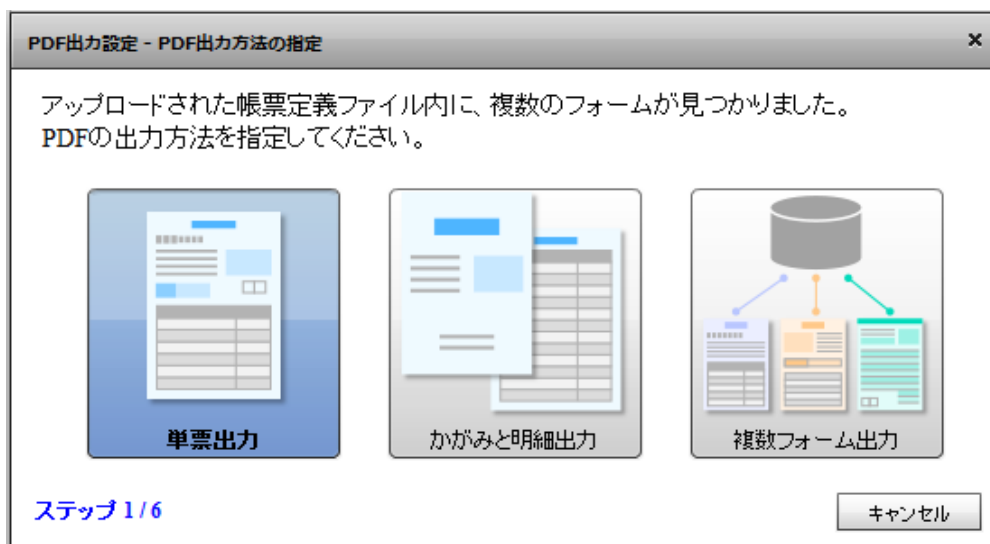


アップロード完了画面

[編集]ボタンをクリックすると、PDF 出力方法の指定画面が表示されます。

選択できる出力方法は以下になります。

- |          |                            |
|----------|----------------------------|
| 単票出力     | 指定したフォームで PDF ファイルを作成します。  |
| かがみと明細出力 | かがみをもつ PDF ファイルを作成します。     |
| 複数フォーム出力 | 複数のフォームを持つ PDF ファイルを作成します。 |



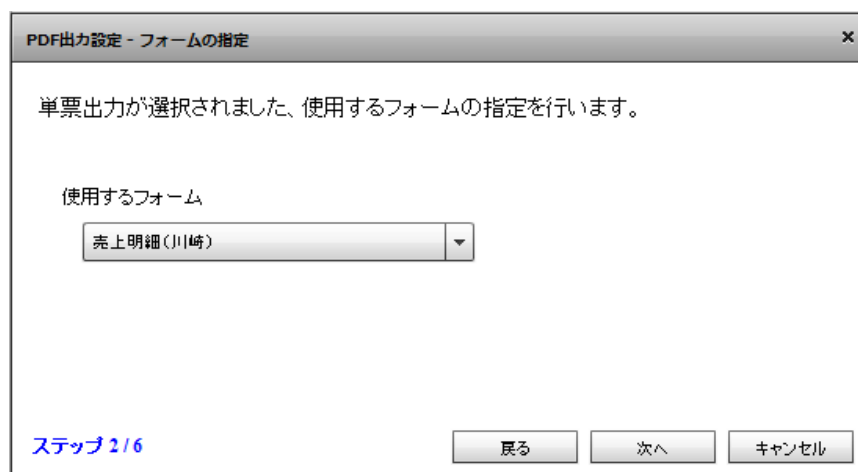
出力方法の指定

### 単票出力

使用するフォームを指定します。

指定するフォームは帳票定義ファイルに含まれるフォームから選択します。

使用するフォームを選択して[次へ]をクリックします。



使用フォームの指定

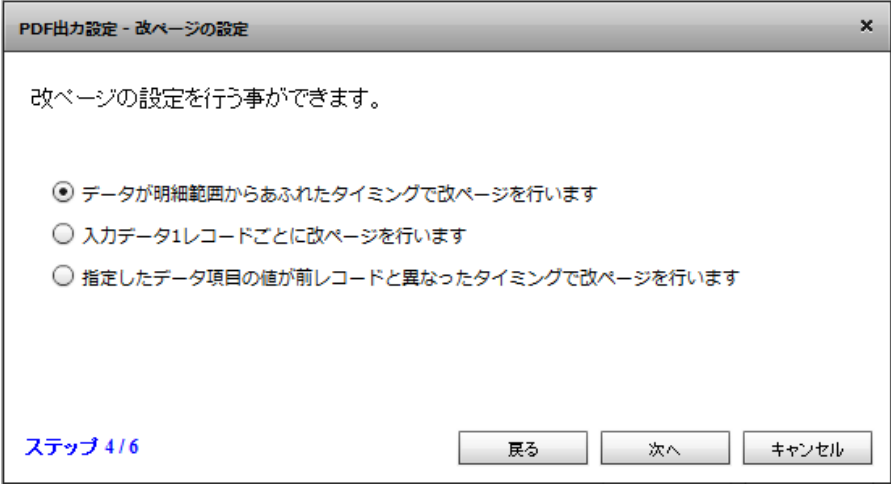
改ページ方法の設定を行います。

以下の項目が指定できます。選択した後、[次へ]ボタンをクリックします。

データが明細範囲からあふれたタイミングで改ページを行います

入力データ1レコードごとに改ページを行います

指定したデータ項目の値が前レコードと異なったタイミングで改ページを行います



PDF出力設定 - 改ページの設定

改ページの設定を行う事ができます。

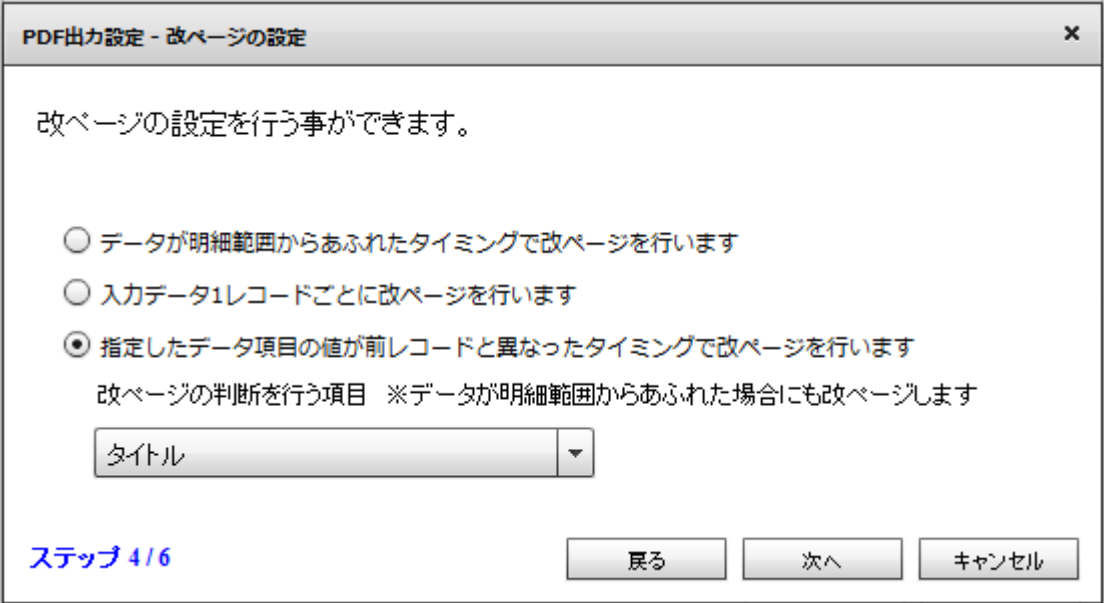
データが明細範囲からあふれたタイミングで改ページを行います  
 入力データ1レコードごとに改ページを行います  
 指定したデータ項目の値が前レコードと異なったタイミングで改ページを行います

ステップ 4/6

戻る 次へ キャンセル

## 改ページの設定

[指定したデータ項目の値が前レコード異なったタイミングで改ページを行います]を選択した場合、データ項目の指定を行います。



PDF出力設定 - 改ページの設定

改ページの設定を行う事ができます。

データが明細範囲からあふれたタイミングで改ページを行います  
 入力データ1レコードごとに改ページを行います  
 指定したデータ項目の値が前レコードと異なったタイミングで改ページを行います  
 改ページの判断を行う項目 ※データが明細範囲からあふれた場合にも改ページします

タイトル

ステップ 4/6

戻る 次へ キャンセル

## 改ページの判断を行うデータ項目の指定



指定された帳票定義ファイルの集計機能が有効になっている場合、改ページの項目の設定画面は表示されません。  
改ページの動作は帳票定義ファイルに設定された集計機能の設定に従って動作します。

フォーム設定の完了画面が表示されます。[完了]ボタンをクリックします。



PDF出力設定 - フォーム設定の完了

設定を終了します。

ステップ 6 / 6

戻る 完了 キャンセル

#### フォーム設定の完了

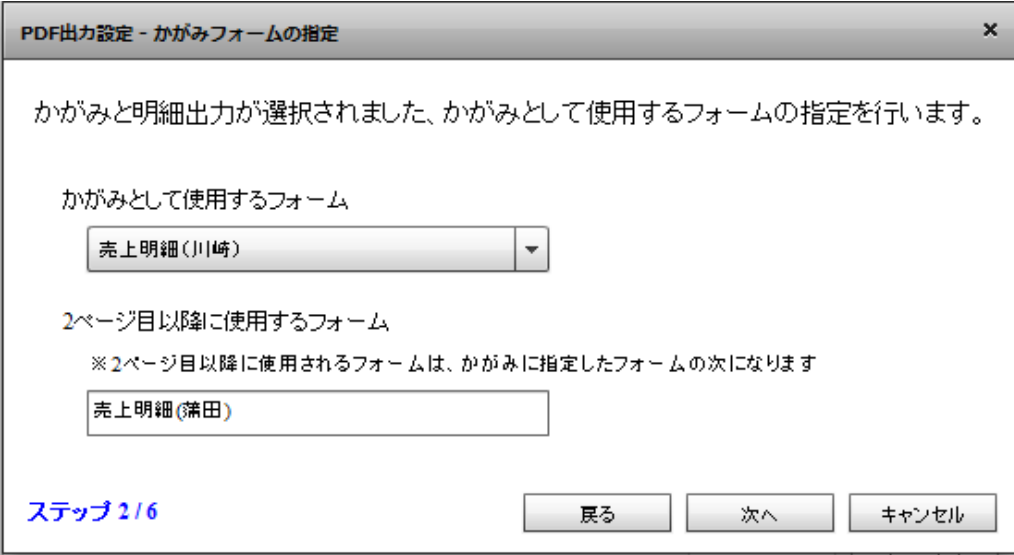
##### かがみと明細出力

かがみとして使用するフォームを選択します。

かがみフォームを指定すると、2ページ目以降に使用されるフォームが表示されます。



2ページ目以降に利用されるフォームは、かがみに指定したフォームの次のフォームとなります。任意に変更することはできません



PDF出力設定 - かがみフォームの指定

かがみと明細出力が選択されました、かがみとして使用するフォームの指定を行います。

かがみとして使用するフォーム

売上明細(川崎)

2ページ目以降に使用するフォーム

※2ページ目以降に使用されるフォームは、かがみに指定したフォームの次になります

売上明細(蒲田)

ステップ 2 / 6

戻る 次へ キャンセル

#### かがみフォームの指定

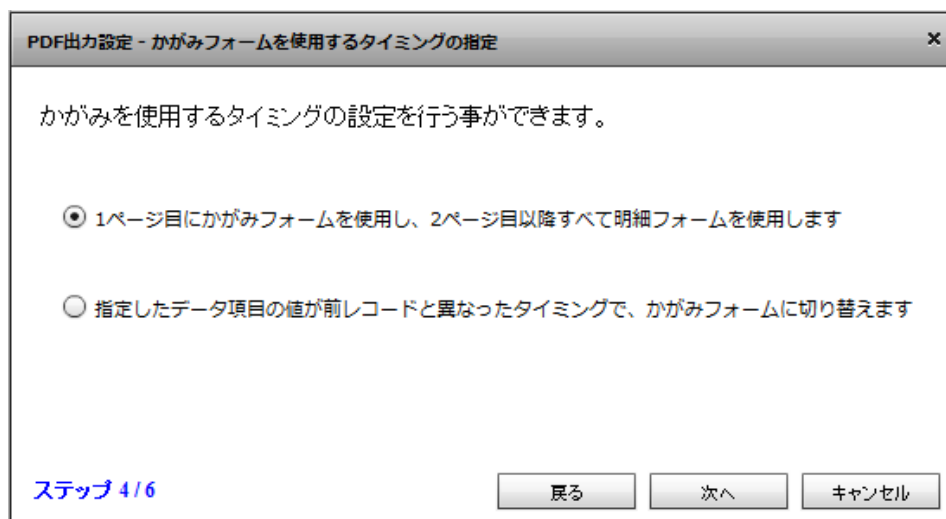
かがみを使用するタイミングを指定します。

以下の項目が指定できます。選択した後、[次へ]ボタンをクリックします。

1ページ目にかがみフォームを使用し、2ページ目以降すべて明細のフォームを使用します。

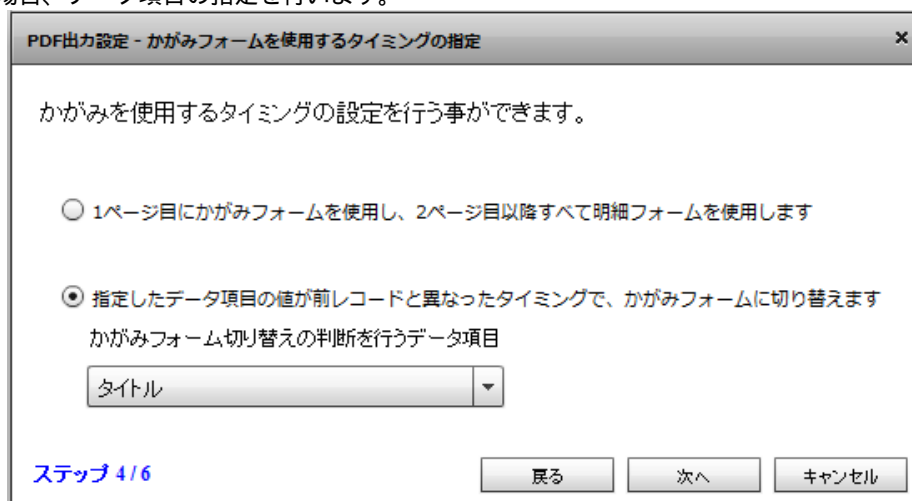
指定したデータ項目の値が前レコードと異なったタイミングで、かがみフォームに切り替えます。





#### かがみフォームを使用するタイミングの指定

[指定したデータ項目の値が前レコードと異なったタイミングで、かがみフォームに切り替えます]を指定した場合、データ項目の指定を行います。



#### フォーム切り替えの判断を行うデータ項目の指定



指定された帳票定義ファイルの集計機能が有効になっている場合、改ページの項目の設定画面は表示されません。  
改ページの動作は帳票定義ファイルに設定された集計機能の設定に従って動作します。

明細あふれ以外のタイミング(フォームの切り替え、集計機能による改ページ)で改ページが発生した場合に、連番型データの値をリセットするかを指定します。

デフォルトではチェックは入っていません。

選択した後、[次へ]ボタンをクリックします。

PDF出力設定 - 連番データの設定

明細あふれ以外のタイミングで改ページが行われる可能性があります。  
その際の連番データの動作設定を行う事ができます。

かがみフォーム切り替え時の連番型のデータの動作

連番データを1に戻します

ステップ 5 / 6

戻る 次へ キャンセル

## 連番データの設定



かがみを使用するタイミングの設定を「指定したデータ項目の値が前レコードと異なったタイミングで、かがみフォームに切り替えます」に指定した場合にのみ設定が必要となります。

フォーム設定の完了画面が表示されます。[完了]ボタンをクリックします。

PDF出力設定 - フォーム設定の完了

設定を終了します。

ステップ 6 / 6

戻る 完了 キャンセル

## フォーム設定の完了

## 複数フォーム出力

フォーム情報を取得する、データ項目の指定を行います。

指定するデータ項目は帳票定義ファイルに含まれるデータ項目から選択します。

選択した後、[次へ]ボタンをクリックします。

PDF出力設定 - 項目データの指定

複数フォーム出力が選択されました、フォームの指定に項目データを使用します。  
使用する項目データを指定してください。

使用するフォームの判断を行う入力データの項目

タイトル

ステップ 2/6

戻る 次へ キャンセル

## 項目データの指定

指定したデータ項目から、使用するフォームの情報が取得できなかった場合の動作を指定します。

以下の項目が指定できます。

選択した後、[次へ]ボタンをクリックします。

実行時エラーとします

指定したフォームを使用します。

PDF出力設定 - フォームが存在しない場合の動作設定

項目データを用いて指定したフォームが存在しない場合があります。  
その際に使用するフォームを選択してください。

実行時エラーとします

指定したフォームを使用します

ステップ 3/6

戻る 次へ キャンセル

## フォームが存在しない場合の動作設定

[指定したフォームを使用します]を指定した場合、使用するフォームを指定します。

改ページ方法の設定を行います。

以下の項目が指定できます。

データが明細範囲からあふれたタイミングで改ページを行います

入力データ1レコードごとに改ページを行います

指定したデータ項目の値が前レコードと異なったタイミングで改ページを行います

PDF出力設定 - 改ページの設定

改ページの設定を行う事ができます。

データが明細範囲からあふれたタイミングで改ページを行います  
 入力データ1レコードごとに改ページを行います  
 指定したデータ項目の値が前レコードと異なったタイミングで改ページを行います

ステップ 4/6

戻る 次へ キャンセル

#### 改ページの設定

[指定したデータ項目の値が前レコード異なったタイミングで改ページを行います]を選択した場合、データ項目の指定を行います。

PDF出力設定 - 改ページの設定

改ページの設定を行う事ができます。

データが明細範囲からあふれたタイミングで改ページを行います  
 入力データ1レコードごとに改ページを行います  
 指定したデータ項目の値が前レコードと異なったタイミングで改ページを行います  
 改ページの判断を行う項目 ※データが明細範囲からあふれた場合にも改ページします

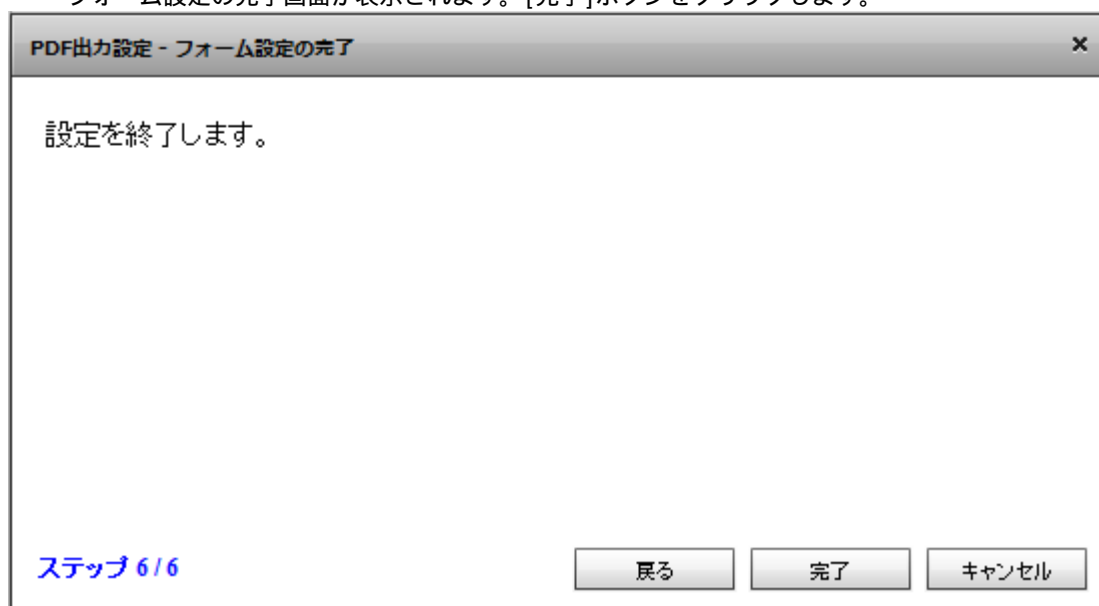
タイトル

ステップ 4/6

戻る 次へ キャンセル

#### 改ページの判断を行うデータ項目の指定

フォーム設定の完了画面が表示されます。[完了]ボタンをクリックします。



フォーム設定の完了

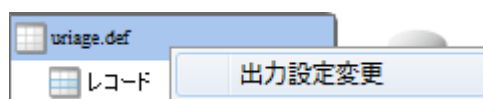
出力方法の指定を完了すると、PDF 出力設定画面の OK ボタンをクリックできるようになります。PDF

出力設定画面の OK ボタンをクリックすると、マッパー編集画面が表示されます。

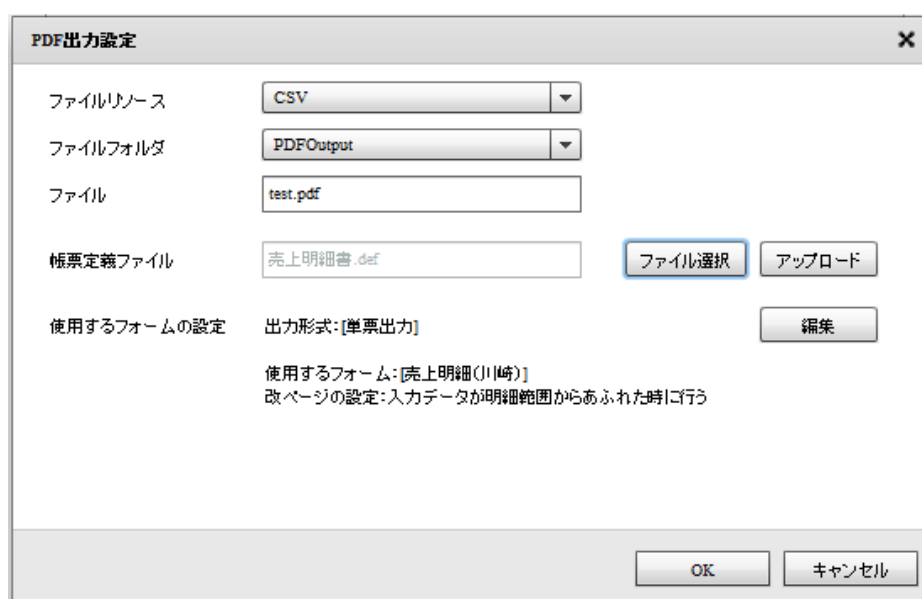


マッパー編集画面

マップターの出力側に表示されている帳票定義ファイル名を右クリックする事で、出力設定の変更を行うことができます。



帳票定義ファイル名の右クリックメニュー



出力設定の変更画面

## 2.6.31 PDF ファイルの制限事項

使用する帳票定義ファイルは、下記の点にご注意の上、作成してください。

1. 下記のフォントをご使用ください。他のフォントが指定されている場合、すべて「MSゴシック」が指定された場合と同じ動作を致します。

MS 明朝 / @MS 明朝 / MS P明朝 / MS ゴシック / @MS ゴシック  
MS Pゴシック / Arial / Courier New / Times New Roman / Symbol

2. フォントに登録がない文字は出力されません。
3. フォントは全て、サブセット埋め込みが行われます。
4. フォント設定に誤りがあった場合、IPAフォントで出力が行われます。
5. 矩形オブジェクト、角丸指定の「角を丸める大きさ」は幅のみが使用されます。
6. 塗りつぶしパターンは「グレースケール」のみ有効です。他のパターンは使用することができません。
7. 塗りつぶし矢印は、使用することができません。
8. 2次元コード PDF 417 は、使用することができません。
9. 固定イメージを貼り付けている場合、画像を出力することはできません。

## 2.6.32 KINTONE 選択

### kintone アダプターを使用する前に

ご利用いただく際には、以下の設定が必須となります。ご注意ください。

- ・ Windows の場合

`${Tomcat 導入ディレクトリ}/bin/Tomcat7w.exe` を開きます

「Java」タブの「Java Options」: の中に以下を追記します。

`-Dhttps.protocols=TLSv1.1,TLSv1.2`

上記設定をしたのち、Tomcat を再起動してください。

- ・ Linux の場合

`${Tomcat 導入ディレクトリ}/bin/setenv.sh` を開きます ( ない場合には作成します )

JAVA\_OPTS として、以下を記述します。

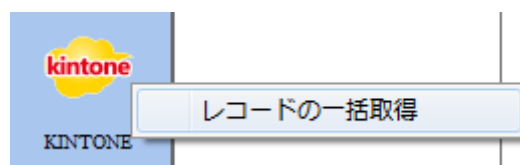
`-Dhttps.protocols=TLSv1.1,TLSv1.2`

以下、設定例です。

```
#!/bin/sh
JAVA_OPTS="-Dhttps.protocols=TLSv1.1,TLSv1.2"
export JAVA_OPTS
```

上記設定をしたのち、Tomcat を再起動してください。

データ転送元に KINTONE を選択します。



KINTONE 選択 (データ転送元)

### <レコードの一括取得>

レコードの一括取得

API【レコードの一括取得】を呼び出し、レコードを取得します。

① kintoneリソース Kintoneリソース

② kintoneApp 案件管理\_サブテーブル

③ フィールドリスト

フィールド名	フィールドタイプ	1レコード目の値
<input type="checkbox"/> 123 \$revision	__REVISION__	4
<input type="checkbox"/> 123 レコード番号	RECORD_NUMBER	2
<input type="checkbox"/> ABC 更新者	MODIFIER	Administrator
<input type="checkbox"/> ABC 作成ユーザー	CREATOR	Administrator
<input type="checkbox"/> TIME 更新日時	UPDATED_TIME	2015-07-21T06:23:00Z
<input type="checkbox"/> TIME 作成日時	CREATED_TIME	2015-04-15T01:21:00Z
<input type="checkbox"/> ABC 会社名	SINGLE_LINE_TEXT	
<input type="checkbox"/> ABC 部署名	SINGLE_LINE_TEXT	人事管理部
<input type="checkbox"/> ABC 先方担当者名	SINGLE_LINE_TEXT	佐藤二郎
<input type="checkbox"/> ABC TEL	SINGLE_LINE_TEXT	12-2345-7890

④ 全てのフィールドにチェック

⑤ 全てのフィールドのチェックを外す

⑥ 絞り込み条件の指定...

OK キャンセル

転送元選択

データ転送元では、フィールドリスト内のチェックボックスにチェックを入れる事で転送元として参照できます。

#### 【kintone リソース】

登録されている kintone リソースをリスト形式で選択します。アダプターの新規選択時に、リスト内の一番上の kintone リソースが自動的に選択されます。

#### 【kintoneApp】

登録されている kintoneApp をリスト形式で選択します。kintone リソースを選択した際に、リスト内の一番上の kintoneApp が自動的に選択されます。



### 【フィールドリスト】

選択された kintoneApp に存在するフィールドの一覧が表示されます。

kintoneApp が選択されていない場合には、何も表示されません。



フィールドタイプが FILE のフィールドは選択できません。

#### <サブテーブル>

フィールドタイプが SUBTABLE のフィールドはサブテーブルとして扱われます。

#### <サブフィールド>

サブテーブル内にフィールドが存在する場合、サブフィールドとして扱われます。

サブフィールドはサブテーブルの入れ子として表示されます。

フィールド名	フィールドタイプ	1レコード目の値
<input type="checkbox"/> ABC 小計	CALC	199999980
<input type="checkbox"/> ABC 活動履歴	SUBTABLE	[cv.qanat.kintone.database.Record@4958359]
<input type="checkbox"/> DATE 活動日	DATE	2015-04-15
<input type="checkbox"/> ABC 活動内容	DROP_DOWN	メール
<input type="checkbox"/> ABC メモ	SINGLE_LINE_TEXT	
<input type="checkbox"/> ABC 添付ファイル	FILE	[]
<input type="checkbox"/> ABC Table_2	SUBTABLE	[cv.qanat.kintone.database.Record@40d01a0d]
<input type="checkbox"/> ABC 往訪日	SINGLE_LINE_TEXT	
<input type="checkbox"/> ABC 会議内容	SINGLE_LINE_TEXT	
<input type="checkbox"/> 123 ルックアップ	NUMBER	

サブテーブル・サブフィールドの表示



サブテーブルは1つだけ選択する事ができます。サブテーブルにチェックを入れた場合、他のサブテーブルはチェック不可となります。別のサブテーブルを使用する場合には、チェックを外してください。

### 【全てのフィールドにチェック】

サブテーブルとサブフィールドを除く、選択可能な全てのフィールドがチェックされます。

### 【全てのフィールドのチェックを外す】

サブテーブルとサブフィールドを除く、選択可能な全てのフィールドのチェックが外れます。

### 【絞り込み条件の指定】

フィールドをドラッグ&ドロップし、絞り込み条件と並び替えを指定します。

絞り込み条件の指定

フィールドをドラッグ&ドロップし、絞り込み条件と並び替えを指定します。

フィールド名	フィールドタイプ	サブテーブル名
Revision	__REVISION__	
レコード番号	RECORD_NUMBER	
更新者	MODIFIER	
作成ユーザー	CREATOR	
更新日時	UPDATED_TIME	
作成日時	CREATED_TIME	
会社名	SINGLE_LINE_TEXT	
部署名	SINGLE_LINE_TEXT	

絞り込み:	(	フィールド名	条件	値	)	並び替え:	フィールド名	順序

絞り込み条件

並び替え

OK

キャンセル

絞り込み条件の指定

**<絞り込み条件>**

絞り込み条件に指定したいフィールドを、絞り込み条件の領域にドラッグ&ドロップします。  
SQL の Where 句と同じ構文を作成します。

絞り込み	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、の中から使用する検索条件を選択します。フィールドタイプによって、選択できる条件が異なります。
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。

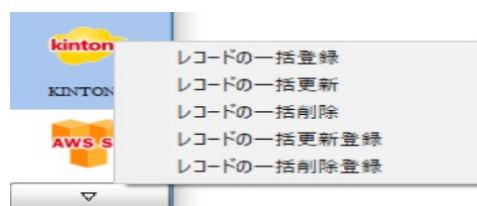
以下のフィールドタイプは、絞り込み条件として使用できません。  
\_\_REVISION\_\_ : リビジョン、CALC : 計算、RICH\_TEXT : リッチエディター、  
FILE : 添付ファイル、CATEGORY : カテゴリ

**<並び替え>**

並び替えをしたいフィールドを、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。

以下のフィールドタイプは、並び替え条件として使用できません。  
\_\_REVISION\_\_ : リビジョン、CALC : 計算、MULTI\_LINE\_TEXT : 文字列 (複数行)、  
RICH\_TEXT : リッチエディター、CHECK\_BOX : チェックボックス、  
MULTI\_SELECT : 複数選択、USER\_SELECT : ユーザー選択、  
ORGANIZATION\_SELECT : 組織選択、GROUP\_SELECT : グループ選択、  
FILE : 添付ファイル、CATEGORY : カテゴリ、STATUS\_ASSIGNEE : 作業者

データ転送先に KINTONE を選択します。



KINTONE 選択（データ転送先）

### <レコードの一括登録>

レコードの一括登録

API【レコードの一括登録】を呼び出し、レコードを登録します。

① kintoneリソース resource

② kintoneApp 案件管理

③ フィールドリスト

フィールド名	フィールドタイプ	1レコード目の値
<input type="checkbox"/> 123 \$revision	__REVISION__	
<input type="checkbox"/> 123 レコード番号	RECORD_NUMBER	
<input type="checkbox"/> ABC 更新者	MODIFIER	
<input type="checkbox"/> ABC 作成ユーザー	CREATOR	
<input type="checkbox"/> TIME UPDATE 更新日時	UPDATED_TIME	
<input type="checkbox"/> TIME UPDATE 作成日時	CREATED_TIME	
<input type="checkbox"/> ABC 会社名	SINGLE_LINE_TEXT	
<input type="checkbox"/> ABC 部署名	SINGLE_LINE_TEXT	
<input type="checkbox"/> ABC 先方担当者名	SINGLE_LINE_TEXT	
<input type="checkbox"/> ABC TEL	SINGLE_LINE_TEXT	

④ 全てのフィールドにチェック

⑤ 全てのフィールドのチェックを外す

OK キャンセル

転送先選択（一括登録）

#### 【kintone リソース】

登録されている kintone リソースをリスト形式で選択します。アダプターの新規選択時に、リスト内の一番上の kintone リソースが自動的に選択されます。

#### 【kintoneApp】

登録されている kintoneApp をリスト形式で選択します。kintone リソースを選択した際に、リスト内の一番上の kintoneApp が自動的に選択されます。

#### 【フィールドリスト】

選択された kintoneApp に存在するフィールドの一覧が表示されます。

kintoneApp が選択されていない場合には、何も表示されません。

- ❗ フィールドタイプが FILE のフィールドは選択できません。
- レコードの登録の場合、以下のフィールドタイプのフィールドは選択できません。  
 RECORD\_NUMBER : レコード番号、\_\_ID\_\_ : レコード ID、\_\_REVISION\_\_ : リビジョン  
 CREATOR : 作成者、MODIFIER : 更新者、USER\_SELECT : ユーザー選択、  
 ORGANIZATION\_SELECT : 組織選択、GROUP\_SELECT : グループ選択、  
 CATEGORY : カテゴリ、STATUS : ステータス、STATUS\_ASSIGNEE : 作業者

#### 【全てのフィールドにチェック】

サブテーブルとサブフィールドを除く、選択可能な全てのフィールドがチェックされます。

#### 【全てのフィールドのチェックを外す】

サブテーブルとサブフィールドを除く、選択可能な全てのフィールドのチェックが外れます。

### <レコードの一括更新>

レコードの一括更新とは、入力元の全てのデータを出力先の kintone のレコードデータと突き合わせて、データの更新をおこなう機能です。

#### 更新キー

本機能の利用にあたっては、レコードの突き合わせをおこなう（対象レコードを判断する）ためのフィールドを指定する必要があります。このフィールドを更新キーと呼びます。

更新キーには、フィールドタイプが RECORD\_NUMBER : レコード番号、\$id : レコード番号、「値の重複を禁止する」の設定がされた SINGLE\_LINE\_TEXT : 文字列（1行）または NUMBER : 数値 のフィールドを指定できます。マップ画面上で、フィールドの右クリックメニューから「更新キーに指定する」を選択することで更新キーとして設定できます。



更新キーに指定すると、フィールド名のアイコン部分の右下に黄色いマークが表示されます。

123 レコード番号

更新キー未指定

123 レコード番号

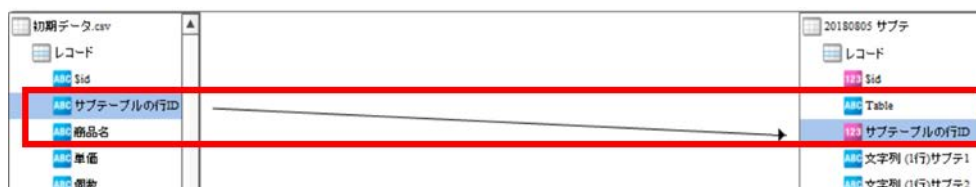
更新キーとして指定

## サブテーブルの更新キー

サブテーブルの更新キーとして以下の2種類の設定ができます。

### サブテーブルの行 ID を更新キーに設定する

マップ画面上で、サブテーブルの行 ID を結線することで更新キーとして設定できます。



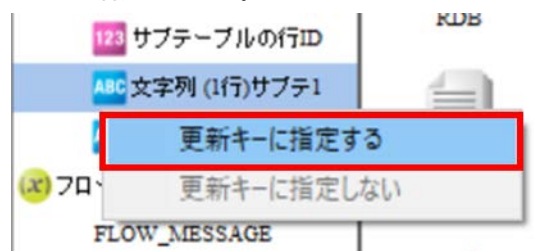
サブテーブルの更新キー設定(サブテーブルの行 ID)

### サブテーブルのフィールドを更新キーに設定する

マップ画面上で、更新キーに利用したいフィールドの右クリックメニューから

「更新キーに指定する」を選択することで更新キーとして設定できます。

更新キーには SINGLE\_LINE\_TEXT : 文字列 (1行) または NUMBER : 数値 のフィールドを更新キーとして利用できます。



サブテーブルの更新キー設定(サブテーブルのフィールド)

## レコードの一括更新 - 転送先の選択画面

レコードの一括更新

API【レコードの一括更新】を呼び出し、レコードを更新します。  
更新キーとしてレコード番号、レコードID、文字列(一行)、数値、リビジョンを指定できます。  
※アプリコードが設定されている場合、レコード番号を更新キーに指定すると実行時にエラーとなります。

① kintoneリソース resource

② kintoneApp 案件管理

③ フィールドリスト

フィールド名	フィールドタイプ	1レコード目の値
<input type="checkbox"/> Revision	__REVISION__	
<input type="checkbox"/> Sid	__ID__	
<input type="checkbox"/> ステータス	STATUS	
<input type="checkbox"/> レコード番号	RECORD_NUMBER	
<input type="checkbox"/> 更新者	MODIFIER	
<input type="checkbox"/> 作成ユーザー	CREATOR	
<input type="checkbox"/> 更新日時	UPDATED_TIME	
<input type="checkbox"/> 作成日時	CREATED_TIME	
<input type="checkbox"/> 会社名	SINGLE_LINE_TEXT	
<input type="checkbox"/> 部署名	SINGLE_LINE_TEXT	
<input type="checkbox"/> 先方担当者名	SINGLE_LINE_TEXT	
<input type="checkbox"/> TEL	SINGLE_LINE_TEXT	
<input type="checkbox"/> FAX	SINGLE_LINE_TEXT	
<input type="checkbox"/> メールアドレス	SINGLE_LINE_TEXT	

④ 全てのフィールドにチェック

⑤ 全てのフィールドのチェックを外す

⑥ 更新条件の設定...

OK キャンセル

転送先選択（一括更新）

## 【kintone リソース】

登録されている kintone リソースをリスト形式で選択します。アダプターの新規選択時に、リスト内の一番上の kintone リソースが自動的に選択されます。

## 【kintoneApp】

登録されている kintoneApp をリスト形式で選択します。kintone リソースを選択した際に、リスト内の一番上の kintoneApp が自動的に選択されます。

## 【フィールドリスト】

選択された kintoneApp に存在するフィールドの一覧が表示されます。

kintoneApp が選択されていない場合には、何も表示されません。

## 【全てのフィールドにチェック】

サブテーブルとサブフィールドを除く、選択可能な全てのフィールドがチェックされます。

## 【全てのフィールドのチェックを外す】

サブテーブルとサブフィールドを除く、選択可能な全てのフィールドのチェックが外れます。

## 【更新条件の設定】

一括更新をおこなうときに突き合わせをする更新キーの入力値に応じた更新処理の振る舞いを設定できます。

- 更新キーの値が重複するレコードが存在した場合の設定

エラー	更新キーの値が重複していた場合、更新処理がエラーになります。
最初のレコードで更新	更新キーの値が重複していた場合、重複している更新キーが含まれるレコードの内、最初のレコードで更新処理をおこないます。
最後のレコードで更新	更新キーの値が重複していた場合、重複している更新キーが含まれるレコードの内、最後のレコードで更新処理をおこないます。

- 更新キーの値が更新先アプリに存在しない場合の設定

エラー	更新キーの値が更新先アプリに存在しない場合、更新処理がエラーになります。
無視する	更新キーの値が更新先アプリに存在しない場合、そのレコードを無視して更新処理をおこないます。

- 更新キーの値が null 値もしくは空文字だった場合の設定

エラー	更新キーの値が null 値もしくは空文字だった場合、更新処理がエラーになります。
無視する	更新キーの値が null 値もしくは空文字だった場合、そのレコードを無視して更新処理をおこないます。

更新条件の設定

更新キーの値が重複するレコードが存在した場合の設定  
エラー

更新キーの値が更新先アプリに存在しない場合の設定  
エラー

更新キーの値が null 値もしくは空文字だった場合の設定  
エラー  
エラー  
無視する



フィールドタイプが FILE のフィールドは選択できません。



更新キーは1つだけ指定できます。サブテーブルも利用する場合は、サブテーブルにも更新キーを1つ指定できます。これ以上の数のフィールドに更新キー指定をおこなって結線した場合は実行時にエラーとなります。



フィールドタイプが \_\_ID\_\_ : レコード ID もしくは RECORD\_NUMBER : レコード番号のフィールドは、更新キーとしてのみ利用可能です。そのため、結線がされている場合には更新キーとして設定されていなくても、更新キーとして動作します。

以下のフィールドは更新の対象とすることはできません。



CREATOR : 作成者、CREATED\_TIME : 作成時間、MODIFIER : 更新者、  
UPDATED\_TIME : 更新時間、CALC : 計算、RICH\_TEXT : リッチエディター、  
USER\_SELECT : ユーザー選択、ORGANIZATION\_SELECT : 組織選択、  
GROUP\_SELECT : グループ選択、CATEGORY : カテゴリ、STATUS : ステータス、  
STATUS\_ASSIGNEE : 作業者



レコードの一括更新時にサブテーブルを更新対象とした際に、更新キーとして指定したサブテーブルの行 ID もしくはサブテーブルのフィールドに対して、存在しない(更新キーが一致しない)データを入力した場合には、サブテーブルの更新キー不一致等のエラーにはならず、該当レコードのサブテーブル部を無視したデータで更新処理を行います。



## <レコードの一括削除>

レコードの削除の際には、フィールドタイプが RECORD\_NUMBER : レコード番号 のフィールドが必須となっているため、あらかじめ選択済みとなっています。チェックを外す事はできません。

レコードの一括削除

API【レコードの一括削除】を呼び出し、レコードを削除します。  
削除キーとしてレコード番号、レコードID、リビジョンが指定できます。

① kintoneリソース resource

② kintoneApp 案件管理

③ フィールドリスト

フィールド名	フィールドタイプ	1レコード目の値
<input type="checkbox"/> 123 \$revision	__REVISION__	
<input checked="" type="checkbox"/> 123 レコード番号	RECORD_NUMBER	
<input type="checkbox"/> ABC 更新者	MODIFIER	
<input type="checkbox"/> ABC 作成ユーザー	CREATOR	
<input type="checkbox"/> TIME UPDATE 更新日時	UPDATED_TIME	
<input type="checkbox"/> TIME CREATE 作成日時	CREATED_TIME	
<input type="checkbox"/> ABC 会社名	SINGLE_LINE_TEXT	
<input type="checkbox"/> ABC 部署名	SINGLE_LINE_TEXT	
<input type="checkbox"/> ABC 先方担当者名	SINGLE_LINE_TEXT	
<input type="checkbox"/> ABC TEL	SINGLE_LINE_TEXT	

④ 全てのフィールドにチェック

⑤ 全てのフィールドのチェックを外す

OK キャンセル

転送先選択（一括削除）

### 【kintone リソース】

登録されている kintone リソースをリスト形式で選択します。アダプターの新規選択時に、リスト内の一番上の kintone リソースが自動的に選択されます。

### 【kintoneApp】

登録されている kintoneApp をリスト形式で選択します。kintone リソースを選択した際に、リスト内の一番上の kintoneApp が自動的に選択されます。

### 【フィールドリスト】

選択された kintoneApp に存在するフィールドの一覧が表示されます。

kintoneApp が選択されていない場合には、何も表示されません。

### 【全てのフィールドにチェック】

サブテーブルとサブフィールドを除く、選択可能な全てのフィールドがチェックされます。

### 【全てのフィールドのチェックを外す】

サブテーブルとサブフィールドを除く、選択可能な全てのフィールドのチェックが外れます。

## <レコードの一括更新登録>

レコードの一括更新登録は、入力元のデータを出力先の kintone のレコードデータと突き合わせて、データの更新をおこない、突き合わせで該当するものがないデータを登録する機能です。

### 更新キー

更新キーについては、kintone 選択の<レコードの一括更新>をご参照ください。

### レコードの一括更新登録 - 転送先の選択画面

レコードの一括更新登録

API【レコードの一括更新】と【レコードの一括登録】を呼び出します。  
更新キーに一致するレコードを更新し、一致しないレコードを登録します。  
更新キーとしてレコード番号、レコードID、文字列(1行)、数値、リビジョンを使用できます。  
※アプリコードが設定されている場合、レコード番号を更新キーに指定すると実行時にエラーとなります。

① kintoneリソース mykintone

② kintoneApp 追加機能(更新機能強化)

③ フィールドリスト

フィールド名	フィールドコード	フィールドタイプ	1レコード目の値
<input type="checkbox"/> Revision	Revision	__REVISION__	77
<input type="checkbox"/> ID	Id	__ID__	102
<input type="checkbox"/> ステータス	ステータス	STATUS	
<input checked="" type="checkbox"/> Table	Table	SUBTABLE	
<input type="checkbox"/> サブテーブルのid	subtableid*	SUBTABLE_ID	3144465
<input type="checkbox"/> 日付_サブテーブル	日付_0	DATE	2019-01-01
<input type="checkbox"/> 時刻_サブテーブル	時刻_0	TIME	00:00
<input type="checkbox"/> 日時_サブテーブル	日時_0	DATETIME	2018-12-31T15:00:00Z
<input type="checkbox"/> リンク_サブテーブル	リンク_0	LINK	https://1021
<input type="checkbox"/> 文字列 (1行)_サブテーブル	文字列_1行_0	SINGLE_LINE_TEXT	文字列サブテ1021
<input type="checkbox"/> 文字列 (複数行)_サブテーブル	文字列_複数行_0	MULTI_LINE_TEXT	文字列サブテ1021
<input type="checkbox"/> 数値_サブテーブル	数値_0	NUMBER	1021
<input type="checkbox"/> ラジオボタン_サブテーブル	ラジオボタン_0	RADIO_BUTTON	sample1

④ 全てのフィールドにチェック

⑤ 全てのフィールドのチェックを外す

⑥ 更新条件の設定...

OK キャンセル

転送選択(一括更新登録)

#### 【kintone リソース】

登録されている kintone リソースをリスト形式で選択します。アダプターの新規選択時に、リスト内の一番上の kintone リソースが自動的に選択されます。

#### 【kintoneApp】

登録されている kintoneApp をリスト形式で選択します。kintone リソースを選択した際に、リスト内の一番上の kintoneApp が自動的に選択されます。

#### 【フィールドリスト】

選択された kintoneApp に存在するフィールドの一覧が表示されます。

kintoneApp が選択されていない場合には、何も表示されません。

#### 【全てのフィールドにチェック】

サブテーブルとサブフィールドを除く、選択可能な全てのフィールドがチェックされます。

#### 【全てのフィールドのチェックを外す】

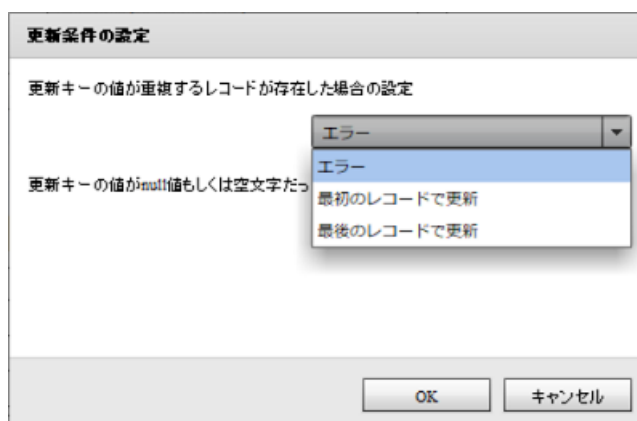
サブテーブルとサブフィールドを除く、選択可能な全てのフィールドのチェックが外れます。

### 【更新条件の設定】

一括更新をおこなうときに突き合わせをする更新キーの入力値に応じた更新処理の振る舞いを設定できます。

- 更新キーの値が重複するレコードが存在した場合の設定

エラー	更新キーの値が重複していた場合、更新処理がエラーになります。  更新キーに文字列（1行）または数値が設定されている場合、登録対象となったレコードの更新キーに重複があるとエラーになります。  (重複している更新キーの値が空文字の場合は、エラーになりません。)
最初のレコードで更新	更新キーの値が重複していた場合、重複している更新キーが含まれるレコードの内、最初のレコードで更新処理をおこないます。  更新キーに文字列（1行）または数値が設定されている場合、登録対象となったレコードの更新キーが重複しているレコードの内、最初のレコードで登録をおこないます。
最後のレコードで更新	更新キーの値が重複していた場合、重複している更新キーが含まれるレコードの内、最後のレコードで更新処理をおこないます。  更新キーに文字列（1行）または数値が設定されている場合、登録対象となったレコードの更新キーが重複しているレコードの内、最後のレコードで登録をおこないます。



- 更新キーの値が null 値もしくは空文字だった場合の設定

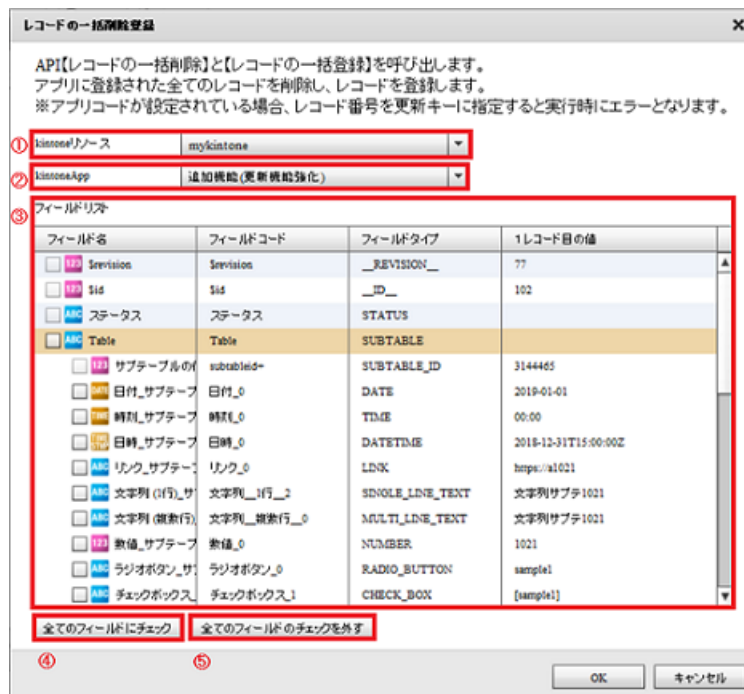
エラー	更新キーの値が null 値もしくは空文字だった場合、更新処理がエラーになります。  登録対象となったレコードの更新キーの値が null 値もしくは空文字だった場合、登録処理がエラーになります。
無視する	更新キーの値が null 値もしくは空文字だった場合、そのレコードを無視して更新処理をおこないます。

登録対象となったレコードの更新キーの値が null 値もしくは空文字だった場合、そのレコードを無視して登録処理をおこないます。

- ❗ フィールドタイプが FILE のフィールドは選択できません。
- ❗ 更新キーは1つだけ指定できます。サブテーブルも利用する場合は、サブテーブルにも更新キーを1つ指定できます。これ以上の数のフィールドに更新キー指定をおこなって結線した場合は実行時にエラーとなります。
- ❗ フィールドタイプが \_\_ID\_\_ : レコード ID もしくは RECORD\_NUMBER : レコード番号のフィールドは、更新キーとしてのみ利用可能です。そのため、結線がされている場合には更新キーとして設定されていなくても、更新キーとして動作します。
- ❗ 以下のフィールドは更新の対象とすることはできません。  
 \_\_REVISION\_\_ : リビジョン、CREATOR : 作成者、CREATED\_TIME : 作成時間、  
 MODIFIER : 更新者、UPDATED\_TIME : 更新時間、CALC : 計算、  
 RICH\_TEXT : リッチエディター、USER\_SELECT : ユーザー選択、  
 ORGANIZATION\_SELECT : 組織選択、GROUP\_SELECT : グループ選択、  
 CATEGORY : カテゴリ、STATUS : ステータス、STATUS\_ASSIGNEE : 作業者
- ❗ レコードの一括更新登録時にサブテーブルを更新対象とした際に、更新キーとして指定したサブテーブルの行 ID もしくはサブテーブルのフィールドに対して、存在しない(更新キーが一致しない)データを入力した場合には、サブテーブルの更新キー不一致等のエラーにはならず、該当レコードのサブテーブル部を無視したデータで更新処理を行います。
- ❗ 更新処理が失敗した場合、登録処理はおこなわれません。
- ❗ レコードの一括更新登録では、更新処理と登録処理をおこなうとき、事前に更新登録先のアプリに対してレコードを取得し、更新キーと突き合わせることによって更新処理をおこなうレコードと登録処理をおこなうレコードの判別をしています。そのため、以下の処理に対して API を利用します。
  - レコードを取得する  
 ([アプリに登録済みの全レコード数]/500)回 (小数点第一位を切り上げ)
  - レコードを更新する  
 ([更新処理に該当するレコード数]/100)回 (小数点第一位を切り上げ)
  - レコードを登録する  
 ([登録処理に該当するレコード数]/100)回 (小数点第一位を切り上げ)

## <レコードの一括削除登録>

レコードの一括削除登録は、出力先の kintone のレコードデータを全て削除してから、入力元のデータを登録する機能です。



転送選択(一括削除登録)

### 【kintone リソース】

登録されている kintone リソースをリスト形式で選択します。アダプターの新規選択時に、リスト内の一番上の kintone リソースが自動的に選択されます。

### 【kintoneApp】

登録されている kintoneApp をリスト形式で選択します。kintone リソースを選択した際に、リスト内の一番上の kintoneApp が自動的に選択されます。

### 【フィールドリスト】

選択された kintoneApp に存在するフィールドの一覧が表示されます。

kintoneApp が選択されていない場合には、何も表示されません。

### 【全てのフィールドにチェック】

サブテーブルとサブフィールドを除く、選択可能な全てのフィールドがチェックされます。

### 【全てのフィールドのチェックを外す】

サブテーブルとサブフィールドを除く、選択可能な全てのフィールドのチェックが外れます。

- ❗ フィールドタイプが FILE のフィールドは選択できません。  
以下のフィールドは登録の対象とすることはできません。  
RECORD\_NUMBER : レコード番号、\_\_ID\_\_ : レコード ID、\_\_REVISION\_\_ : リビジョン
- ❗ CREATOR : 作成者、MODIFIER : 更新者、USER\_SELECT : ユーザー選択、  
ORGANIZATION\_SELECT : 組織選択、GROUP\_SELECT : グループ選択、  
CATEGORY : カテゴリ、STATUS : ステータス、STATUS\_ASSIGNEE : 作業者
- ❗ 一括削除登録処理を実行中に削除登録対象のアプリのレコードを削除すると  
実行エラーになる場合があります。
- ❗ 削除処理が失敗した場合、登録処理はおこなわれません。  
  
一括削除登録では出力先の kintone アプリ内の全てのレコードの削除をおこなう  
ために事前に全てのレコードの \$id を取得し、そのデータを基にして削除処理をお  
こないます。このため、以下の処理に対して API を利用します。
- ❗ レコードを取得する  
([アプリに登録済みの全レコード数]/500)回 (小数点第一位を切り上げ)  
レコードを削除する  
([アプリに登録済みの全レコード数]/100)回 (小数点第一位を切り上げ)  
レコードを登録する  
([入力したレコード数]/100)回 (小数点第一位を切り上げ)

### 2.6.33 kintone アダプターの制限事項

kintone アダプターについて、注意書き以外に下記の点にもご注意ください。

- ・ API トークン認証には対応していません。パスワード認証 (Basic 認証含む) をご利用ください。
- ・ ゲストスペースには対応していません。

## 2.6.33 Amazon S3(CSV)選択

### Amazon S3 アダプターを使用する前に

ご利用いただく際には、以下の設定が必須となります。ご注意ください。

#### ・ Windows の場合

`${Tomcat 導入ディレクトリ}/bin/Tomcat7w.exe` を開きます

「Java」タブの「Java Options」：の中に以下を追記します。

`-Dhttps.protocols=TLSv1.1,TLSv1.2`

上記設定をしたのち、Tomcat を再起動してください。

#### ・ Linux の場合

`${Tomcat 導入ディレクトリ}/bin/setenv.sh` を開きます（ない場合には作成します）

JAVA\_OPTS として、以下を記述します。

`-Dhttps.protocols=TLSv1.1,TLSv1.2`

以下、設定例です。

```
#!/bin/sh
JAVA_OPTS="-Dhttps.protocols=TLSv1.1,TLSv1.2"
export JAVA_OPTS
```

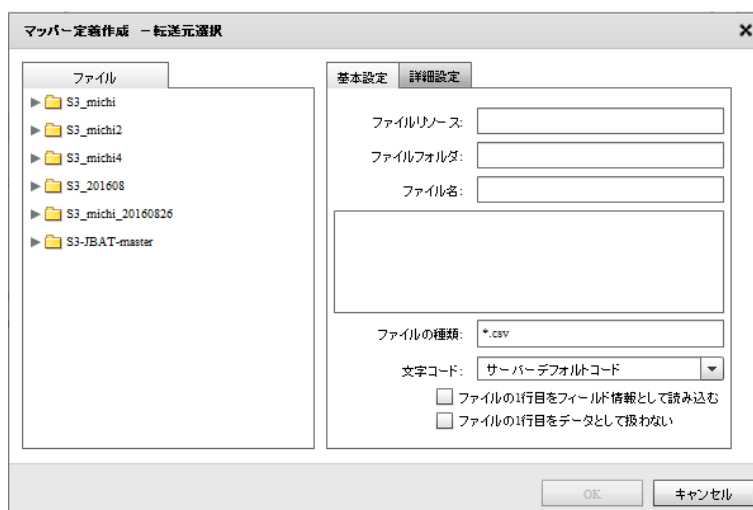
上記設定をしたのち、Tomcat を再起動してください。



## Amazon S3(CSV)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 - 基本設定

### 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。

#### <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。

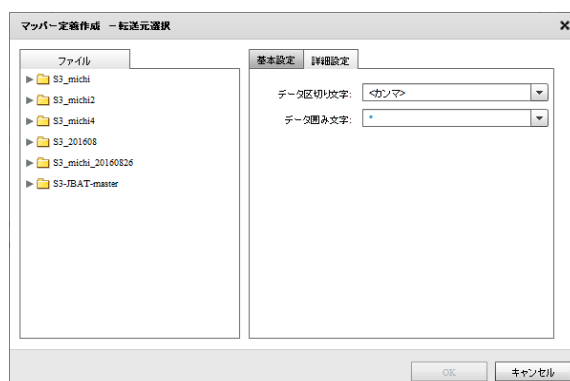
#### <ファイルの 1 行目をフィールド情報として読み込む>

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



転送元ファイル選択 - 詳細設定

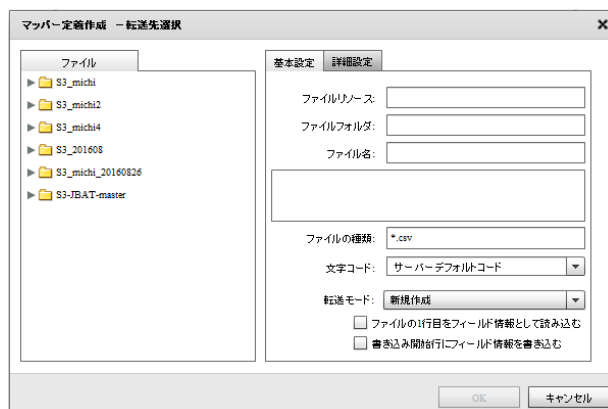
### <データ区切り文字>

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。

## Amazon S3(CSV)をデータ転送先に選択



転送先ファイル選択 - 基本設定

### 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。

## &lt;書き込み文字コード&gt;

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。

## &lt;ファイルの1行目をフィールド情報として読む&gt;

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。

## &lt;書き込み開始行にヘッダ情報を書き込む&gt;

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。

## &lt;転送モード&gt;

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



転送先ファイル選択 - 詳細設定

## &lt;データ区切り文字&gt;

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。

## &lt;データ囲み文字&gt;

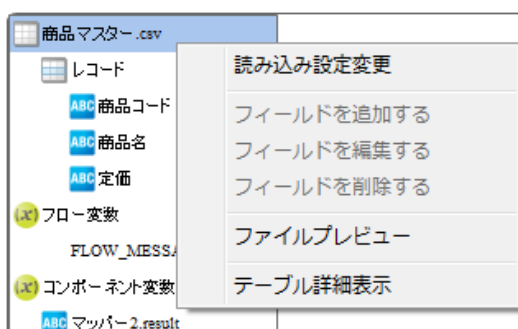
ファイル中の各データを囲む文字を指定します。囲み文字は「' / " / なし」から選択するかまたは、入力し指定します。

## &lt;改行コード&gt;

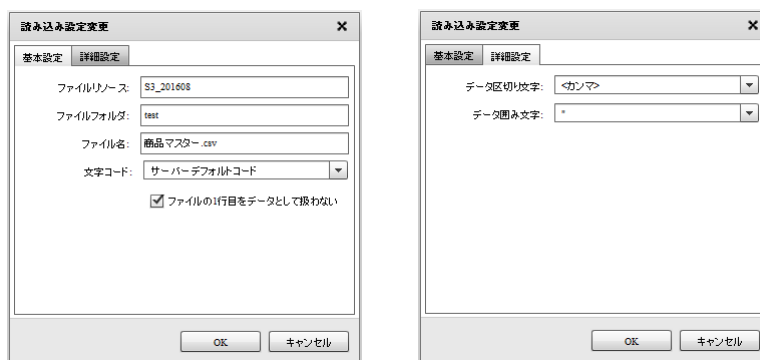
ファイル中の各データ行の改行コードを指定します。「自動(サーバー依存)/CR+LF/CR/LF」から選択します。

## 2.6.34 Amazon S3(CSV)を利用したときの各種メニュー

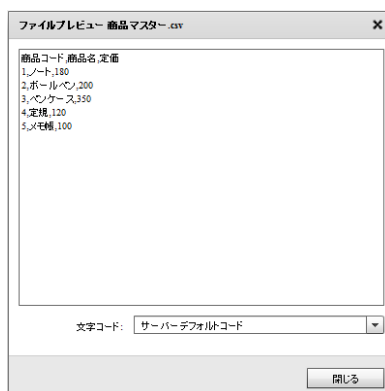
・転送元ファイル関連



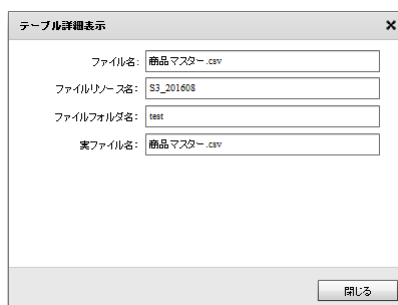
転送元ファイルを選択時に右クリックで表示されるメニュー



読み込み設定の「基本設定」タブと「詳細設定」タブ

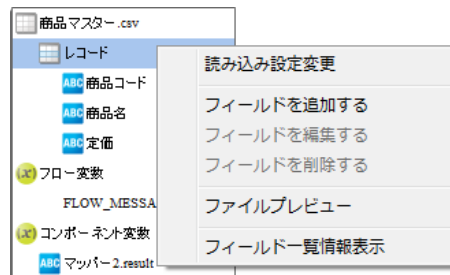


ファイルプレビュー



テーブル詳細表示

## ・ 転送元レコード関連



転送元レコード選択時に右クリックで表示されるメニュー

The '新規フィールド追加' dialog box contains the following elements:

- 表示フィールド名:
- 追加フィールド数:
- ※追加フィールド数に以上の値を指定すると、表示フィールド名に番号が付加されます。
- Buttons: OK, キャンセル

フィールドを追加

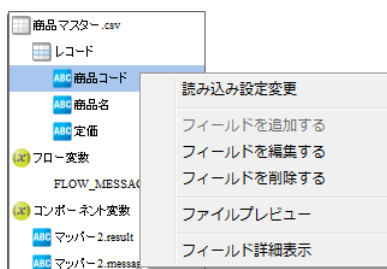
The 'フィールドー覧情報表示' dialog box displays the following table:

ポジション	フィールド名	型名	クオート
1	商品コード	STRING	なし
2	商品名	STRING	なし
3	定価	STRING	なし

Buttons: 閉じる

フィールドー覧情報

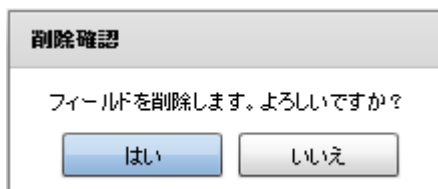
## ・転送元フィールド関連



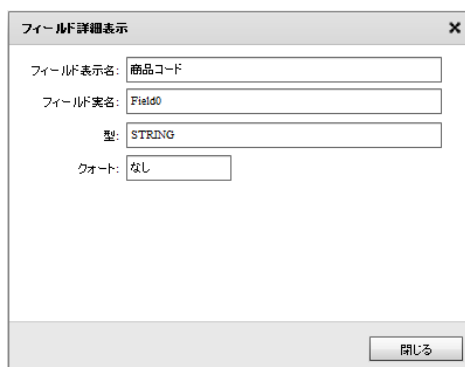
転送元フィールド選択時に右クリックで表示されるメニュー



フィールド編集

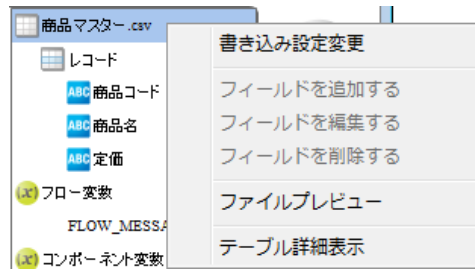


フィールド削除

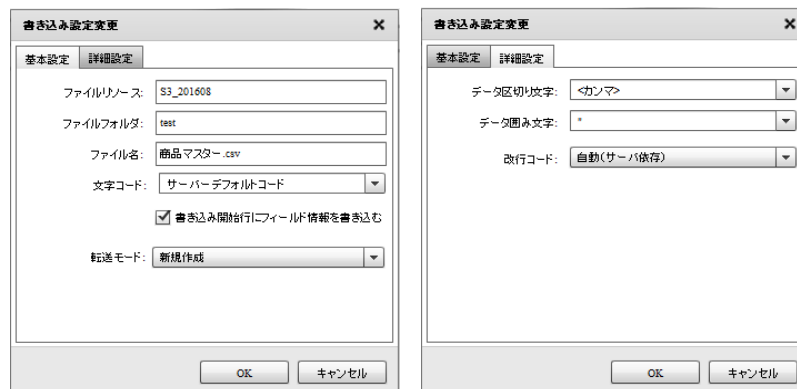


フィールド詳細

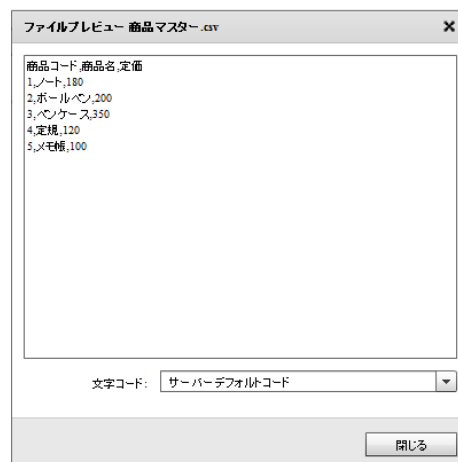
## ・転送先ファイル関連



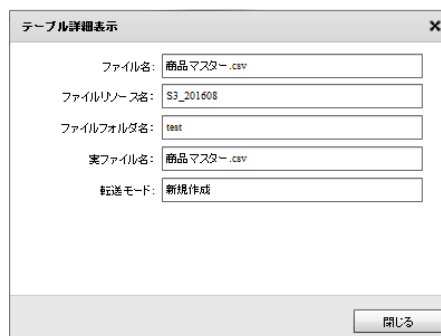
転送先ファイルを選択時に右クリックで表示されるメニュー



書き込み設定変更の「基本設定」タブと「詳細設定」タブ

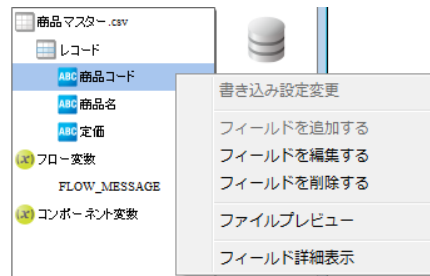


ファイルプレビュー



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

新規フィールド追加

表示フィールド名:

クォート: なし

追加フィールド数:

※追加フィールド数に以上の値を指定すると、表示フィールド名に番号が付加されます。

OK キャンセル

フィールドを追加

フィールド一覧情報表示

ファイル名: 商品マスター.csv

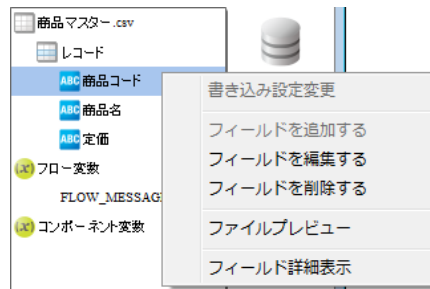
ポジション	フィールド名	型名	クォート
1	商品コード	STRING	なし
2	商品名	STRING	なし
3	定価	STRING	なし

閉じる

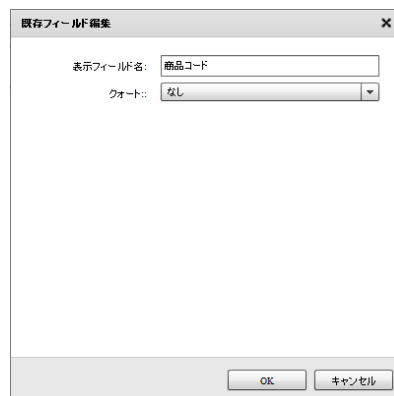
フィールド一覧情報表示



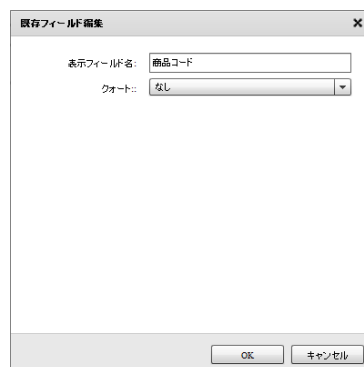
## ・ 転送先フィールド関連



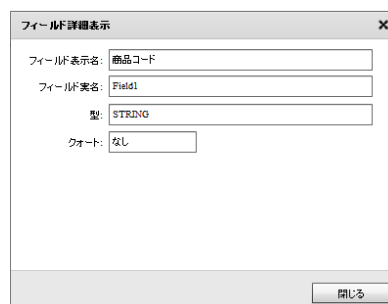
転送先フィールドを選択時に右クリックで表示されるメニュー



フィールドを編集



フィールドを削除



フィールド詳細表示

## 2.6.35 @TOVAS 選択

### @Tovas アダプターを使用する前に

ご利用いただく際には、以下の設定が必須となります。ご注意ください。

・ Windows の場合

`${Tomcat 導入ディレクトリ}/bin/Tomcat7w.exe` を開きます

「Java」タブの「Java Options」: の中に以下を追記します。

`-Dhttps.protocols=TLSv1.1,TLSv1.2`

上記設定をしたのち、Tomcat を再起動してください。

・ Linux の場合

`${Tomcat 導入ディレクトリ}/bin/setenv.sh` を開きます ( ない場合には作成します )

JAVA\_OPTS として、以下を記述します。

`-Dhttps.protocols=TLSv1.1,TLSv1.2`

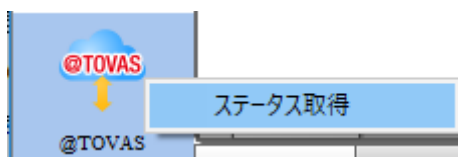
以下、設定例です。

```
#!/bin/sh
JAVA_OPTS="-Dhttps.protocols=TLSv1.1,TLSv1.2"
export JAVA_OPTS
```

上記設定をしたのち、Tomcat を再起動してください。

**データ転送元に@TOVAS を選択します。**

@Tovas をデータ転送元に指定した場合、<ステータス取得>の処理を実行することが可能です。



@TOVAS 選択 (データ転送元)

**<ステータス取得の処理内容>**

本製品から実行された私書箱/FAX/ファイル送信の送信ステータスを取得し、入力値として使用することができる機能です。

送信ステータスは「送信中」「送信完了」など、個々の送信処理が現時点でどのような状態にあるかを取得します。

ステータス取得は、送信処理時に発番される「送信 ID」を利用して送信状態を取得します。

ステータス取得の際、送信 ID の取得元として、本製品で私書箱/FAX/ファイル送信への送信処理を行った際に作成される「送信情報ファイル」を指定し、個々の送信状態を取得します。

## &lt; ステータス取得 &gt;

① ステータス取得元

② ファイル送信元:

③ 取得元ファイルパス:

④ ファイル名:

⑤ 選択

OK キャンセル

## 転送元選択

**【ステータス取得元】**

登録されている@Tovas リソースを選択します。

**【ファイル送信元】**

【ステータス取得元】で選択した@Tovas リソースを表示します。

**【取得元ファイルパス】**

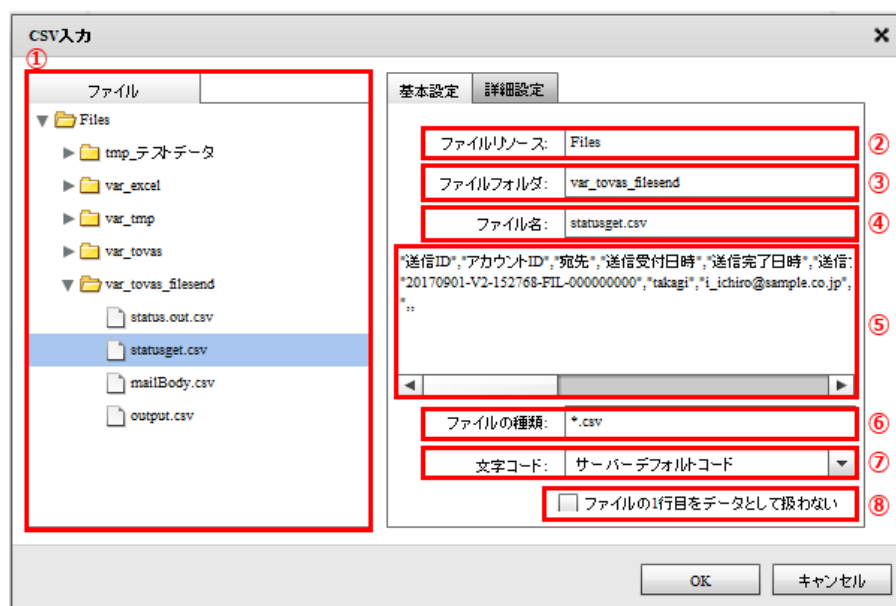
【選択】で選択されたファイルパスを表示します。

**【ファイル名】**

【選択】で選択されたファイル名を表示します。

**【選択】**

【取得元ファイルパス】【ファイル名】を選択するダイアログを表示します。



【取得元ファイルパス】【ファイル名】の選択（基本設定）

**【ファイル】**

送信 ID を取得するファイルを選択します。

**【ファイルリソース】**

【ファイル】で選択したファイルリソースを表示します。

**【ファイルフォルダ】**

【ファイル】で選択したファイルフォルダを表示します。

**【ファイル名】**

【ファイル】で選択したファイル名を表示します。

**【プレビュー表示】**

【ファイル】で選択したファイルの内容を表示します。

**【ファイルの種類】**

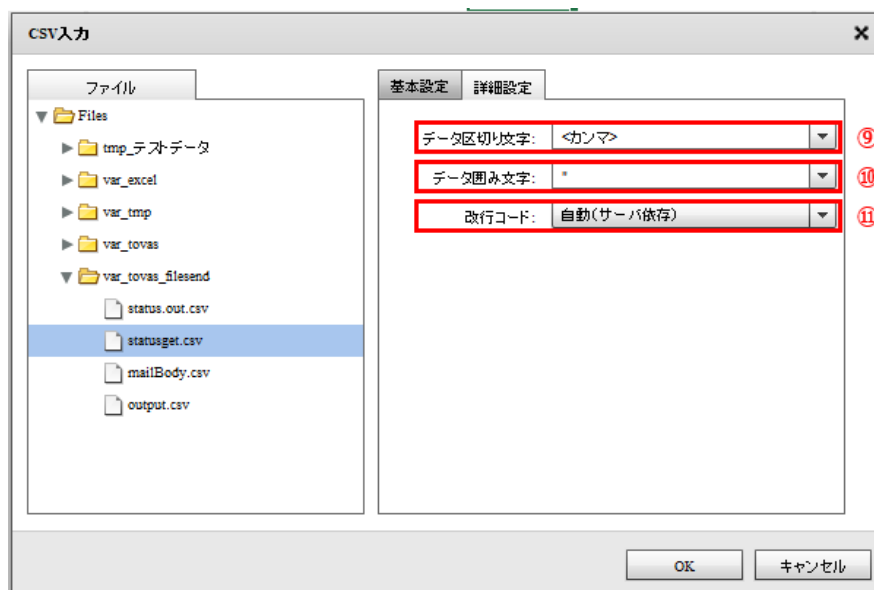
【ファイル】で選択したファイルフォルダの表示するファイルの絞込みをします。絞込みにはワールドカードを利用します。

**【文字コード】**

【ファイル】で選択したファイルを読み込む文字コードを選択します。

**【ファイルの1行目をデータとして扱わない】**

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。



【取得元ファイルパス】【ファイル名】の選択（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】

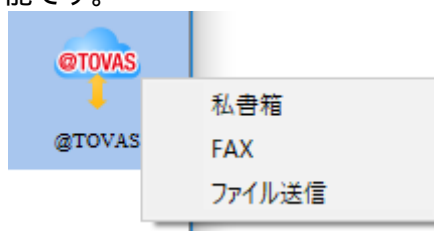
ファイル中の各データを囲む文字を指定します。基本設定タブの【プレビュー表示】を利用し、正しい囲み文字を「' / " / なし」から選択するかまたは、入力し指定します。

#### 【改行コード】

ファイル中の改行コードを指定します。正しい改行コードを「自動(サーバ依存/CR+LF/CR/LF)」から選択します。

**データ転送先に@TOVAS を選択します。**

@Tovas をデータ転送先に指定した場合、<私書箱> <FAX> <ファイル送信> の処理を実行することが可能です。



@TOVAS 選択 (データ転送先)

#### <各機能の処理内容>

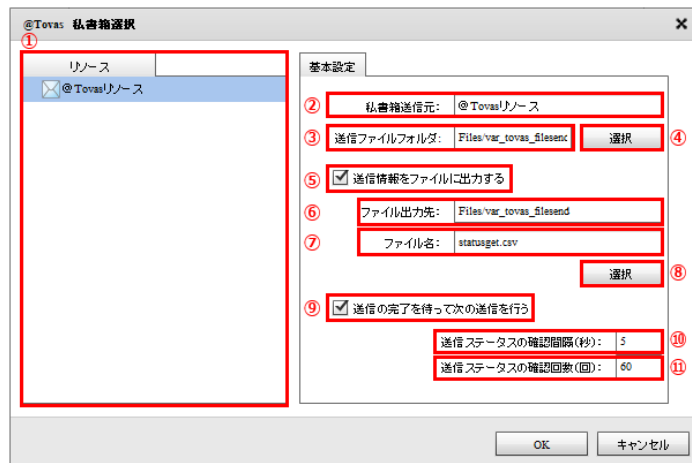
<私書箱> は指定した@Tovas 環境に私書箱送信の処理を実行する機能です。

<FAX> は FAX 送信、<ファイル送信> はファイル送信の処理を実行します。

個々のリクエストの送信完了を待って順次実行したり、送信処理のみを行い、個々の送信状況を別途取得する処理を作成することが可能です。

処理実行時の設定値として、本製品で個々の送信状況を確認するために必要な「送信情報ファイル」の出力の有無を指定することが可能です。

#### <私書箱>



転送先選択 (私書箱)

#### 【リソース】

登録されている@Tovas リソースを選択します。

#### 【私書箱送信元】

【リソース】で選択した@Tovas リソースを表示します。

#### 【送信ファイルフォルダ】

【 選択】で選択したファイルフォルダを表示します。

**【選択】**

送信するファイルが存在するファイルフォルダを選択するダイアログを表示します。

**【送信情報をファイルに出力する】**

送信情報をファイルに出力するかを選択します。

**【ファイル出力先】**

【 選択】で選択したファイルフォルダを表示します。

【送信情報をファイルに出力する】にチェックを入れた場合のみ選択することが可能です。

**【ファイル名】**

【 選択】で選択したファイル名を表示します。

【送信情報をファイルに出力する】にチェックを入れた場合のみ選択することが可能です。

**【選択】**

送信情報ファイルを出力先となるファイルフォルダを選択するダイアログを表示します。

**【送信の完了を待って次の送信を行う】**

送信時の挙動を選択します。

チェックを入れない場合

送信のリクエストを行った後、送信ステータスを確認せず次の送信リクエストを実行します。

チェックを入れた場合

送信ステータスが完了するまで次の送信リクエストは実行されません。

**【送信ステータスの確認間隔（秒）】**

送信ステータスの確認間隔を秒で指定します。初期値は「5」が指定されています。

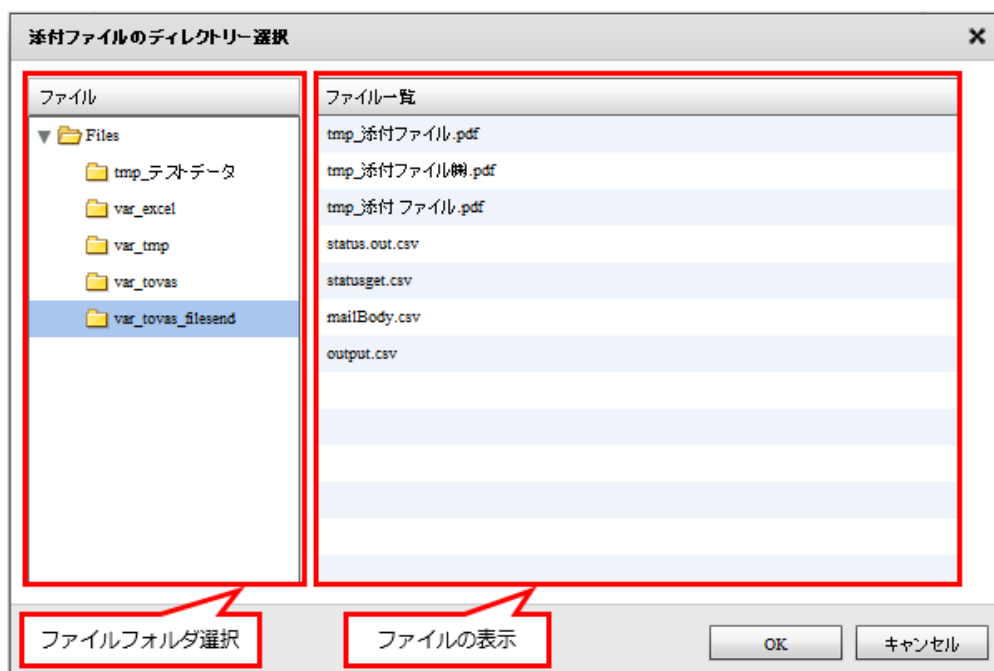
【送信の完了を待って次の送信を行う】にチェックを入れた場合のみ有効となる項目です。

**【送信ステータスの確認回数（回）】**

送信ステータスの確認回数を指定します。初期値は「60」が指定されています。

【送信の完了を待って次の送信を行う】にチェックを入れた場合のみ有効となる項目です。





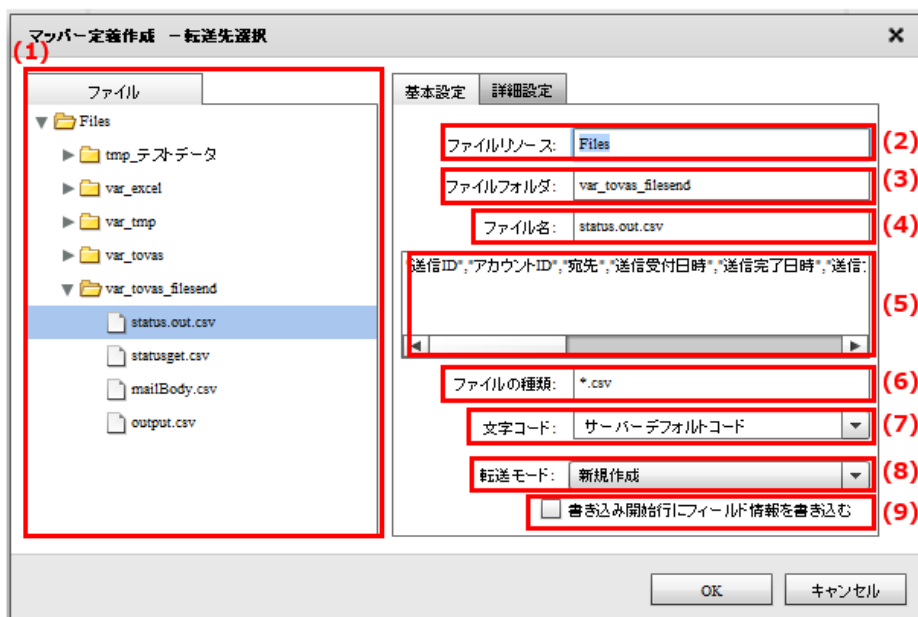
送信ファイルフォルダの選択

<ファイルフォルダ選択>

送付するファイルが存在するファイルフォルダを選択します。

<ファイルの表示>

選択したファイルフォルダに存在するファイルを表示します。



送信情報ファイルの選択（基本設定）

**(1) 【ファイル】**

送信情報を出力するファイルを選択します。

**(2) 【ファイルリソース】**

【ファイル】で選択したファイルリソースを表示します。

**(3) 【ファイルフォルダ】**

【ファイル】で選択したファイルフォルダを表示します。

**(4) 【ファイル名】**

【ファイル】で選択したファイル名を表示します。

**(5) 【プレビュー表示】**

【ファイル】で選択したファイルの内容を表示します。

**(6) 【ファイルの種類】**

【ファイル】で選択したファイルフォルダの表示するファイルの絞り込みをします。絞り込みにはワイルドカードを利用します。

**(7) 【文字コード】**

【ファイル】で選択したファイルを読み込む文字コードを選択します。

**(8) 【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。

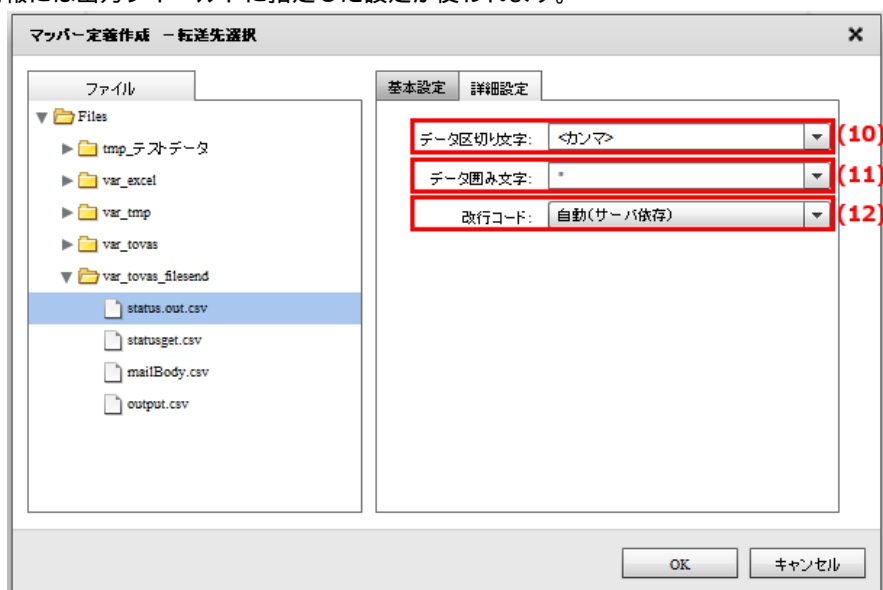
ファイルが見つからない場合は同名のファイルが作成されます。

### (9) 【書き込み開始行にフィールド情報を書き込む】

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。



送信情報ファイルの選択（詳細設定）

### (10) 【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。

### (11) 【データ囲み文字】

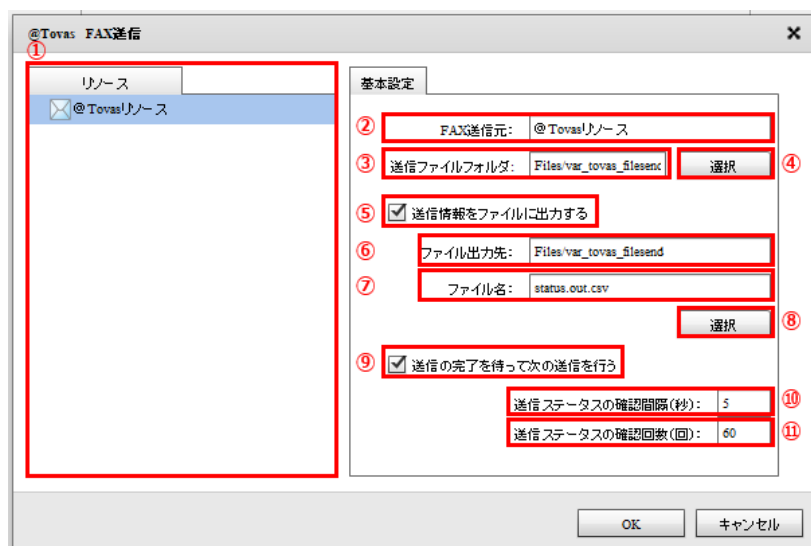
ファイル中の各データを囲む文字を指定します。基本設定タブの【プレビュー表示】を利用し正しい囲み文字を「 ' / " / なし」から選択するかまたは、入力し指定します。

### (12) 【改行コード】

ファイル中の改行コードを指定します。

正しい改行コードを「自動(サーバ依存/CR+LF/CR/LF)」から選択します。

< FAX >



転送先選択 (FAX)

**【リソース】**

登録されている@Tovas リソースを選択します。

**【FAX 送信元】**

【リソース】で選択した@Tovas リソースを表示します。

**【送信ファイルフォルダ】**

【 選択】で選択したファイルフォルダを表示します。

**【選択】**

送信するファイルが存在するファイルフォルダを選択するダイアログを表示します。

<私書箱> と同一の画面、挙動となります。

**【送信情報をファイルに出力する】**

送信情報をファイルに出力するかを選択します。

**【ファイル出力先】**

【 選択】で選択したファイルフォルダを表示します。

【送信情報をファイルに出力する】にチェックを入れた場合のみ選択することが可能です。

**【ファイル名】**

【 選択】で選択したファイル名を表示します。

【送信情報をファイルに出力する】にチェックを入れた場合のみ選択することが可能です。

**【選択】**

送信情報ファイルを出力先となるファイルフォルダを選択するダイアログを表示します。

<私書箱> と同一の画面、挙動となります。

**【送信の完了を待つ次の送信を行う】**

送信時の挙動を選択します。

チェックを入れない場合

送信のリクエストを行った後、送信ステータスを確認せず次の送信リクエストを実行します。

チェックを入れた場合

送信ステータスが完了するまで次の送信リクエストは実行されません。

**【送信ステータスの確認間隔（秒）】**

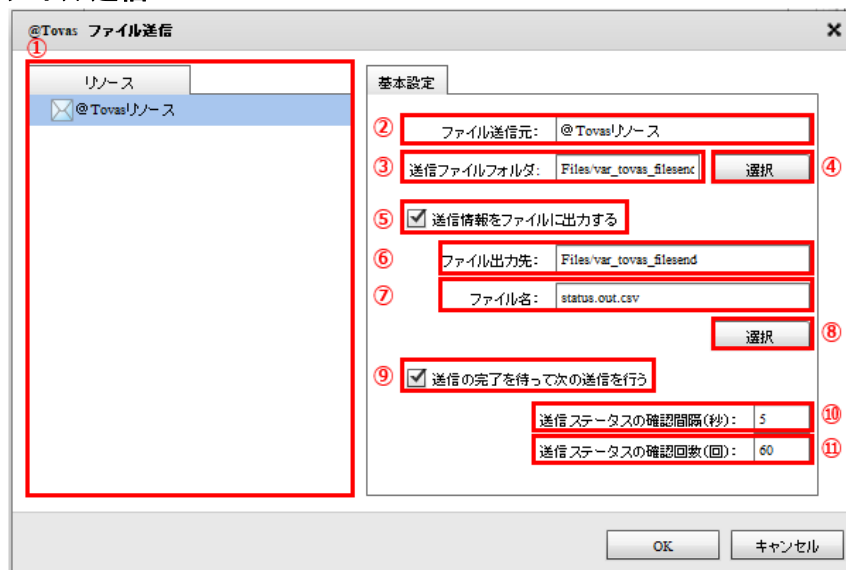
送信ステータスの確認間隔を秒で指定します。初期値は「5」が指定されています。

【送信の完了を待つ次の送信を行う】にチェックを入れた場合のみ有効となる項目です。

**【送信ステータスの確認回数（回）】**

送信ステータスの確認回数を指定します。初期値は「60」が指定されています。

【送信の完了を待つ次の送信を行う】にチェックを入れた場合のみ有効となる項目です。

**<ファイル送信>**

転送先選択（ファイル送信）

**【リソース】**

登録されている@Tovas リソースを選択します。

**【ファイル送信元】**

【リソース】で選択した@Tovas リソースを表示します。

**【送信ファイルフォルダ】**

【 選択】で選択したファイルフォルダを表示します。

**【選択】**

送信するファイルが存在するファイルフォルダを選択するダイアログを表示します。

<私書箱>と同一の画面、挙動となります。

**【送信情報をファイルに出力する】**

送信情報をファイルに出力するかを選択します。

**【ファイル出力先】**

【 選択】で選択したファイルフォルダを表示します。

【送信情報をファイルに出力する】にチェックを入れた場合のみ選択することが可能です。

**【ファイル名】**

【 選択】で選択したファイル名を表示します。

【送信情報をファイルに出力する】にチェックを入れた場合のみ選択することが可能です。

**【選択】**

送信情報ファイルを出力先となるファイルフォルダを選択するダイアログを表示します。

<私書箱>と同一の画面、挙動となります。

**【送信の完了を待って次の送信を行う】**

送信時の挙動を選択します。

チェックを入れない場合

送信のリクエストを行った後、送信ステータスを確認せず次の送信リクエストを実行します。

チェックを入れた場合

送信ステータスが完了するまで次の送信リクエストは実行されません。

**【送信ステータスの確認間隔（秒）】**

送信ステータスの確認間隔を秒で指定します。初期値は「5」が指定されています。

【送信の完了を待って次の送信を行う】にチェックを入れた場合のみ有効となる項目です。

**【送信ステータスの確認回数（回）】**

送信ステータスの確認回数を指定します。初期値は「60」が指定されています。

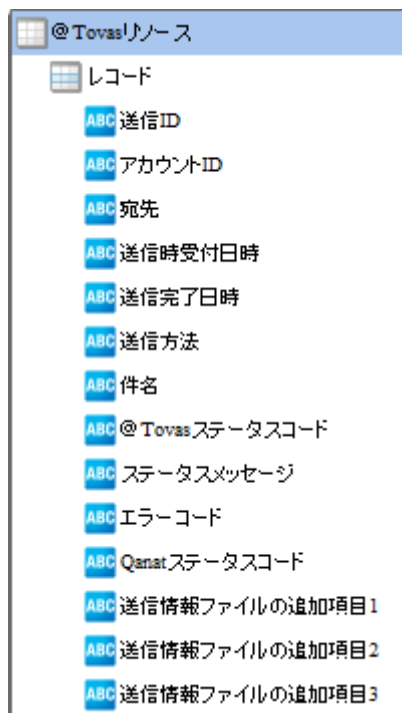
【送信の完了を待って次の送信を行う】にチェックを入れた場合のみ有効となる項目です。

## 2.6.36 @TOVAS を選択したときの各種メニュー

@TOVAS を選択した場合、選択した動作によって表示されるフィールドに違いが発生します。

以下にフィールドの説明を記述します。

### <ステータス取得>



表示フィールド（ステータス取得）

送信 ID	送信ステータスの確認を行う送信 ID を指定するフィールドです。ステータスファイルの 1 列目から取得されます。
アカウント ID	送信を行ったアカウント ID を指定するフィールドです。ステータスファイルの 2 列目から取得されます。
宛先	送信先を指定するフィールドです。ステータスファイルの 3 列目から取得されます。
送信時受付日時	@Tovas が送信処理のリクエストを受け取った日時が出力されるフィールドです。 “yyyy/MM/dd hh:mm:ss” の形式で出力されます。
送信完了日時	@Tovas が送信処理のリクエストを完了した日時が出力されるフィールドです。 “yyyy/MM/dd hh:mm:ss” の形式で出力されます。
送信方法	送信方法が出力されるフィールドです。「私書箱/FAX/ファイル送信」のいずれかの値が出力されます。
件名	件名に指定した値が出力されるフィールドです。

@Tovas ステータスコード @Tovas から発行されるステータスコードが出力されるフィールドです。以下のいずれかの値が出力されます。

値	意味
0	未送信
1	送信中
2	送信完了 または 送信エラー
3	送信キャンセル
4	送信中止

エラーコード

@Tovas から発行されるエラーコードが出力されるフィールドです。正常終了時に「0010」、異常終了の場合には「0010」以外の値が出力されます。

Qanat ステータスコード

本製品が発行するステータスコードが出力されるフィールドです。以下のいずれかの値が出力されます。

値	意味
-1	@Tovas ステータスコードの取得に失敗
0	未送信
1	送信中
2	送信完了
3	送信キャンセル
4	送信中止
5	送信エラー



@Tovas ステータスコードでは、「送信完了」と「送信エラー」が同一の値となっています。上記を判断する場合、Qanat ステータスコードで行うことが可能です。

送信情報ファイルの追加項目 1 送信情報ファイルの 12 列目で指定した値を出力するフィールドです。

送信情報ファイルの追加項目 2 送信情報ファイルの 13 列目で指定した値を出力するフィールドです。

送信情報ファイルの追加項目 3 送信情報ファイルの 14 列目で指定した値を出力するフィールドです。



送信情報ファイルの追加項目は、<ステータス取得>で取得されない任意の値を参照する際に利用するフィールドです。

<私書箱> <FAX> <ファイル送信> 実行時、任意の値をセットすることが可能です。



## &lt;私書箱&gt;

【送信情報をファイルに出力する】にチェックを入れているかどうかで表示されるフィールドに違いが発生しますが、それ以外でレコードおよびフィールドに変更を加えることは出来ません。



表示フィールド（私書箱）

To（宛先名）	@Tovas に表示される送信先宛先名を指定するフィールドです。
To（アドレス）	宛先となるアドレスを指定するフィールドです。 「,」区切りで複数アドレスを指定することが可能です。
CC（宛先名）	@Tovas に表示される CC 宛先名を指定するフィールドです。
CC（アドレス）	CC となるアドレスを指定するフィールドです。 「,」区切りで複数アドレスを指定することが可能です。
BCC（宛先名）	@Tovas に表示される BCC 宛先名を指定するフィールドです。
BCC（アドレス）	BCC となるアドレスを指定するフィールドです。 「,」区切りで複数アドレスを指定することが可能です。
送信ファイル名	送信するファイルを指定するフィールドです。 フィールドには、@Tovas リソース選択時に指定した【送信ファイルフォルダ】に存在するファイル名を指定します。
件名	送信時の件名を指定するフィールドです。
本文	送信時の本文を指定するフィールドです。
送信有効期限	現時点で無効の設定です。
再送回数	現時点で無効の設定です。
送信予定日時	送信を行う日時を指定するフィールドです。

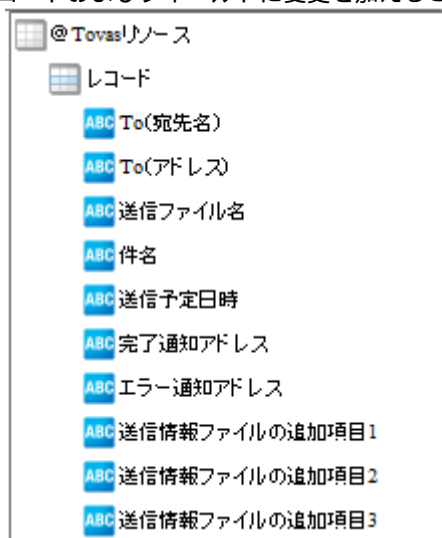
“ yyyy/MM/dd hh:mm:ss ” の形式で指定します。

指定できる日時の最大値は、リクエスト送信日時から 6 日（144 時間）後までになります。

- ダウンロード通知アドレス 送付されたファイルがダウンロードされたことを通知するアドレスを指定するフィールドです。
- エラー通知アドレス エラーが発生した際に通知を行うアドレスを指定するフィールドです。
- 送信控えアドレス 現時点で無効の設定です。
- 送信ファイルの追加項目 1 【送信情報をファイルに出力する】にチェックを入れた場合のみ表示されます。  
@Tovas から取得される値以外の、任意の値を指定するフィールドです。  
指定した値は送信情報ファイルの 12 列目に出力されます。
- 送信ファイルの追加項目 2 【送信情報をファイルに出力する】にチェックを入れた場合のみ表示されます。  
@Tovas から取得される値以外の、任意の値を指定するフィールドです。  
指定した値は送信情報ファイルの 13 列目に出力されます。
- 送信ファイルの追加項目 3 【送信情報をファイルに出力する】にチェックを入れた場合のみ表示されます。  
@Tovas から取得される値以外の、任意の値を指定するフィールドです。  
指定した値は送信情報ファイルの 14 列目に出力されます。

### < FAX >

【送信情報をファイルに出力する】にチェックを入れているかどうかで表示されるフィールドに違いが発生しますが、それ以外でレコードおよびフィールドに変更を加えることは出来ません。



表示フィールド (FAX)

- To (宛先名) @Tovas に表示される送信先宛先名を指定するフィールドです。
- To (アドレス) 宛先となるアドレスを指定するフィールドです。半角数字のみで指定します。  
「,」区切りで複数アドレスを指定することが可能です。
- 送信ファイル名 送信するファイルを指定するフィールドです。  
フィールドには、@Tovas リソース選択時に指定した【送信ファイルフォルダ】に存在するファイル名を指定します。  
送信できるファイル形式は「TIFF」「PDF」になります。
- 件名 送信時の件名を指定するフィールドです。
- 送信予定日時 送信を行う日時を指定するフィールドです。  
“ yyyy/MM/dd hh:mm:ss ” の形式で指定します。  
指定できる日時の最大値は、リクエスト送信日時から 6 日（144 時間）後までに

- 完了通知アドレス なります。  
送付されたファイルが送信されたことを通知するアドレスを指定するフィールドです。
- エラー通知アドレス エラーが発生した際に通知を行うアドレスを指定するフィールドです。
- 送信ファイルの追加項目 1 【送信情報をファイルに出力する】にチェックを入れた場合のみ表示されます。  
@Tovas から取得される値以外の、任意の値を指定するフィールドです。  
指定した値は送信情報ファイルの 12 列目に出力されます。
- 送信ファイルの追加項目 2 【送信情報をファイルに出力する】にチェックを入れた場合のみ表示されます。  
@Tovas から取得される値以外の、任意の値を指定するフィールドです。  
指定した値は送信情報ファイルの 13 列目に出力されます。
- 送信ファイルの追加項目 3 【送信情報をファイルに出力する】にチェックを入れた場合のみ表示されます。  
@Tovas から取得される値以外の、任意の値を指定するフィールドです。  
指定した値は送信情報ファイルの 14 列目に出力されます。

## &lt;ファイル送信&gt;

【送信情報をファイルに出力する】にチェックを入れているかどうかで表示されるフィールドに違いが発生しますが、それ以外でレコードおよびフィールドに変更を加えることは出来ません。



表示フィールド（ファイル送信）

To（宛先名）	@Tovas に表示される送信先宛先名を指定するフィールドです。
To（アドレス）	宛先となるアドレスを指定するフィールドです。 「,」区切りで複数アドレスを指定することが可能です。
CC（宛先名）	@Tovas に表示される CC 宛先名を指定するフィールドです。
CC（アドレス）	CC となるアドレスを指定するフィールドです。 「,」区切りで複数アドレスを指定することが可能です。
BCC（宛先名）	@Tovas に表示される BCC 宛先名を指定するフィールドです。
BCC（アドレス）	BCC となるアドレスを指定するフィールドです。 「,」区切りで複数アドレスを指定することが可能です。
送信ファイル名	送信するファイルを指定するフィールドです。 フィールドには、@Tovas リソース選択時に指定した【送信ファイルフォルダ】に存在するファイル名を指定します。
件名	送信時の件名を指定するフィールドです。
本文	送信時の本文を指定するフィールドです。
送信有効期限	送信有効期限を指定するフィールドです。 半角数字で有効日数を指定します。
再送回数	現時点で無効の設定です。
送信予定日時	送信を行う日時を指定するフィールドです。

“ yyyy/MM/dd hh:mmss ” の形式で指定します。

指定できる日時の最大値は、リクエスト送信日時から 6 日 (144 時間) 後までになります。

**ダウンロード通知アドレス** 送付されたファイルがダウンロードされたことを通知するアドレスを指定するフィールドです。

**エラー通知アドレス** エラーが発生した際に通知を行うアドレスを指定するフィールドです。

**送信控えアドレス** 送信控えを送信するアドレスを指定するフィールドです。

**送信ファイルの追加項目 1** 【送信情報をファイルに出力する】にチェックを入れた場合のみ表示されます。

@Tovas から取得される値以外の、任意の値を指定するフィールドです。

指定した値は送信情報ファイルの 12 列目に出力されます。

**送信ファイルの追加項目 2** 【送信情報をファイルに出力する】にチェックを入れた場合のみ表示されます。

@Tovas から取得される値以外の、任意の値を指定するフィールドです。

指定した値は送信情報ファイルの 13 列目に出力されます。

**送信ファイルの追加項目 3** 【送信情報をファイルに出力する】にチェックを入れた場合のみ表示されます。

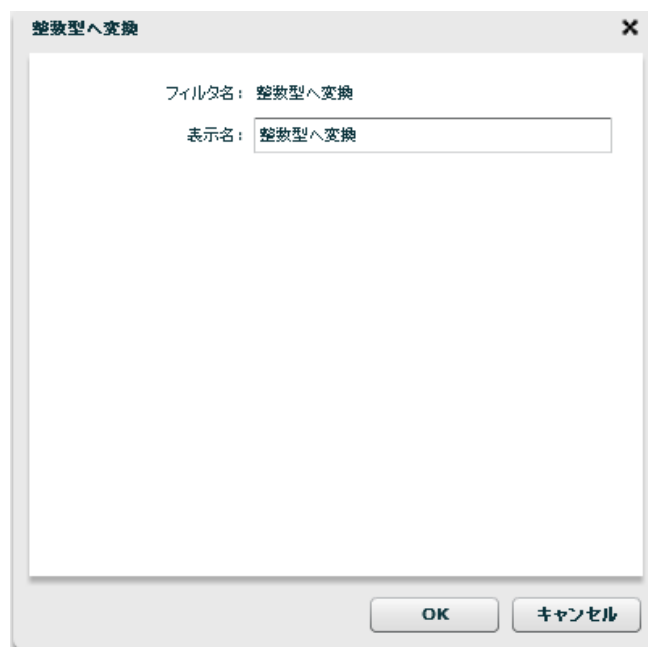
@Tovas から取得される値以外の、任意の値を指定するフィールドです。

指定した値は送信情報ファイルの 14 列目に出力されます。

## 2.6.37 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。

## 2.7 スクリプト定義(Flow Designer)

本製品で扱うことができるスクリプト定義とは、

Velocity 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。

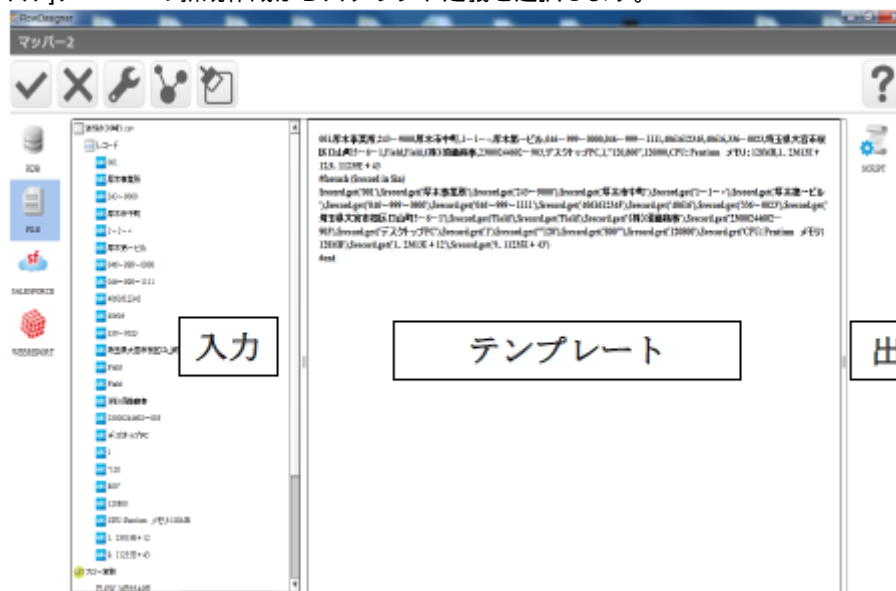
### 2.7.1 スクリプト定義とは

マッパーを定義することで、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。

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。

### 2.7.2 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。

入力リソースの全レコード

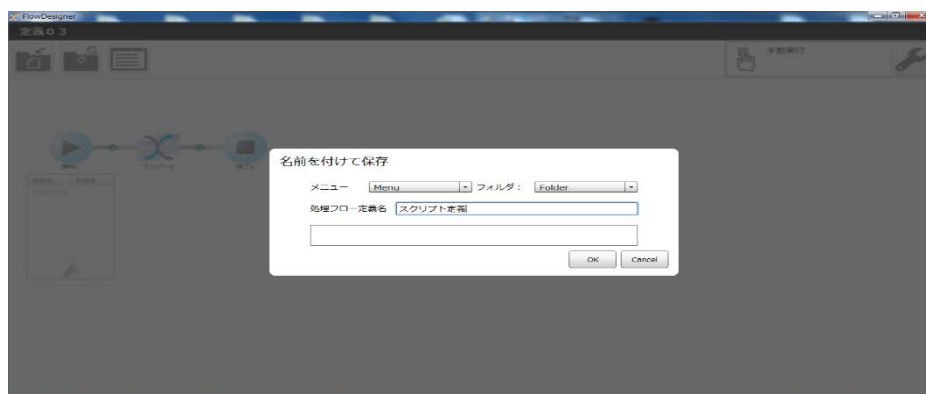
```

001,厚木事業所,243-9000,厚木市中町,1-1-*,厚木第一ビル,046-999-
0000,046-999-1111,4061612345,40616,336-0023,埼玉県大宮市桜区口山町5-6
-1,Field,Field,(株)須藤商事,23000244602-903,デスクトップ
PC,1,"120,800",120800,CPU: Pentium メモリ: 128MB,1. 23613E + 12,9. 11235E + 43
#foreach ($record in $in)
$record.get("001"),$record.get("厚木事業所"),$record.get("243-9000"),$record.get("厚
木市中町"),$record.get("1-1-*"),$record.get("厚木第一ビル"),$record.get("046-999
-0000"),$record.get("046-999-
1111"),$record.get("4061612345"),$record.get("40616"),$record.get("336-
0023"),$record.get("埼玉県大宮市桜区口山町5-6-
1"),$record.get("Field"),$record.get("Field"),$record.get("(株)須藤商事
"),$record.get("23000244602-903"),$record.get("デスクトップ
PC"),$record.get("1"),$record.get("120"),$record.get("800"),$record.get("120800"),$reco
rd.get("CPU: Pentium メモリ: 128MB"),$record.get("1. 23613E + 12"),$record.get("9.
11235E + 43")
#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

👉 手動実行



cvadmin  
2014年10月07日 02時09分

cvadmin  
2014年10月07日 02時09分

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 \$set(文字列)	変数名(文字列)の値を取得します。

スクリプト定義実行時の “出力レコード” はログの詳細情報には表示されません。

すべて “1” と表示されます。



出力レコード数

1

## 2.8 QanatExecute

### 2.8.1 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。

### 2.8.2 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。

```

#####
### 引数説明 ###
#####
### INST_PATH (QanatExecute インストールパス) *
### 例: INST_PATH=C:\QanatExecute
### JAVA_PATH (JDKのパス) *
### 例: JAVA_PATH=C:\Program Files\Java\jdk1.7.0_51
### HOST (ホスト名 or IP) *
### 例: HOST=192.168.255.255
### PORT (ポート番号) *
### 例: PORT=80
### USER (ログインユーザー名) *
### 例: USER=cvalent
### PASSWORD (ログインパスワード) *
### 例: PASSWORD=cvalent
### MENU (実行する定義のメニュー名) *
### 例: MENU=登録機
### FOLDER (実行する定義のフォルダ名) *
### 例: FOLDER=機
### DEF_NAME (実行する定義名) *
### 例: DEF_NAME=Batchファイル実行
### DEF_TYPE (実行定義タイプ) *
### 例: DEF_TYPE=File
### VARIABLE (グローバル変数) *
### 変数を指定しない場合
### VARIABLE=""
### 変数を指定する場合
### 例: VARIABLE="LOGIN" "ID"; "事業所" "J" "東日本" "東京" " " * 変数を指定する際は、(半角スペース) で区切ってください。
### 注: 変数を指定する場合は、(半角スペース) で区切ってください。
### CHANGE_LIB="" * 現在使用できない機能のため定文字のまま変更しないでください。
###
#####

@ECHO OFF
SET INST_PATH=
SET JAVA_PATH=
+
+
SET HOST=
SET PORT=
SET USER=
SET PASSWORD=
SET MENU=
SET FOLDER=
SET DEF_NAME=
SET DEF_TYPE=
SET VARIABLE=""
SET CHANGE_LIB=""
+
@ECHO ON
"%JAVA_PATH%\jre\bin\java" -jar "%INST_PATH%\QanatExecute.jar" -host %HOST% -p
+
@ECHO return code : %ERRORLEVEL%
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 qanscript:スクリプト定義  
 SET VARIABLE :   定義変数  
 SET CHANGELIB :  テーブル切替え

```
@ECHO OFF+
SET INST_PATH=C:\Users\J32204\Desktop\execute+
SET JAVA_PATH=C:\Program Files\Java\jdk1.7.0_51+
+
+
SET HOST=192.168.123.456+
SET PORT=80+
SET USER=cvadmin+
SET PASSWORD=cvadmin+
SET MENU=受注管理+
SET FOLDER=東部支部+
SET DEF_NAME=1月受注集計+
SET DEF_TYPE=qanflow+
SET VARIABLE="BREAK=20"+
SET CHANGELIB="ORDER/TEMP/DEF"="ORDER/EAST/JUNORDERDATA";+
+
+
@ECHO ON+
"%JAVA_PATH%\jre\bin\java" -jar "%INST_PATH%\QanatExecute.jar" -host %HOST%
+
@ECHO return code : %ERRORLEVEL%+
PAUSE+
←
```

#### 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。

テーブル切替えのセットを複数設定する場合、以下のように半角スペースで区切って使用してください。

例 : CHANGELIB="DB1/SCHEMA1/TAB1"="DB2/SCHEMA2/TAB2" "DB3/SCHEMA3/TAB3"="DB4/SCHEMA4/TAB4"

テーブル切替えで使用するデータソース名、スキーマ名、テーブル名は CV 管理ツールで設定した名前を指定してください。



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。

## 2.9 QanatExecute.js

### 2.9.1 QanatExecute.js とは

QanatExecute.js とは、Qanat2.0 REST API を利用して本製品のフロー定義の実行命令を呼び出すツールです。

QanatExecute.js ファイルを外部の別アプリケーション等にアップロードし、別ファイル・別アプリケーションに作成したボタンをクリックして定義の実行命令を呼び出すことで、本製品が導入されていないマシンからでも定義を実行することができます。

これにより、外部の別ファイル・別アプリケーションから本製品のフロー定義を実行することが可能になります。



Qanat2.0 REST API についての詳細は、本マニュアルの「付録 D REST API」の項目をご覧ください。

### 2.9.2 使い方

QanatExecute.js では、定義の実行命令を呼び出す QanatExecute.js ファイルを使用します。

QanatExecute.js ファイルは、後述する sample.html ファイルと併せて以下の場所に存在しています。

- ・サポートサイトにアップロードされている Qanat.zip 内
- ・V1.10.0 以降の製品メディア内のフォルダ(/Module/QanatExecute.js/js)

また、QanatExecute.js ファイルとは別に、以下2つのファイルの設定もしくは作成が必要となります。

1. Qanat サーバーが導入されたマシンの下記フォルダ内に存在する qanat\_rest.xml ファイルの設定  
<Tomcat インストールディレクトリ>/webapps/qanat/config2/qanat/



qanat\_rest.xml ファイルの設定方法は、本マニュアルの「付録 D REST API」の項目をご覧ください。

2. 定義を実行するために必要なパラメータの設定と、定義実行用のボタンを設定するファイルの作成  
サンプルとして sample.html ファイルを提供しております。

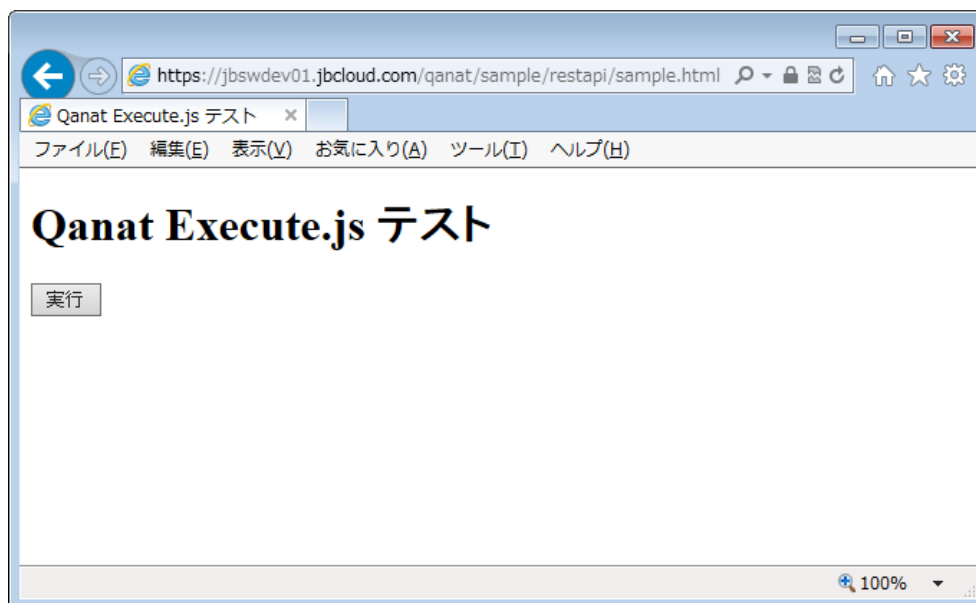
sample.html ファイルを使用する場合には、QanatExecute.js ファイルと sample.html ファイルを Qanat Server が導入されたマシンの下記フォルダにそれぞれ配置する必要があります。

QanatExecute.js : <Tomcat インストールディレクトリ>/webapps/qanat/sample/restapi/js  
sample.html : <Tomcat インストールディレクトリ>/webapps/qanat/sample/restapi

sample.html ファイルでは、画面内の「実行」ボタンをクリックすることで、sample.html ファイル内に記述されたパラメータと一致する定義を実行することができます。

sample.html を表示するには以下のアドレスにアクセスしてください。

http(s)://<Qanat Server の IP またはドメイン名>/qanat/sample/restapi/sample.html



sample.html

sample.html ファイルの内容は以下となります。

```

<!DOCTYPE html>
<html lang="ja">
<head>
  <meta charset="UTF-8">
  <title>Qanat Execute.js テスト</title>
  ① <script type="text/javascript" src="./js/QanatExecute.js"></script>
  <!--
  ② <パラメータ説明>
      必須項目
      execkey   : キー情報(qanat_rest.xmlのqanexeckeyの値)
      server    : Qanatサーバーのアドレス(IPまたはドメイン名)とポート指定
      menu      : 実行する定義が含まれるメニュー
      folder    : 実行する定義が含まれるフォルダ
      def       : 実行する定義名

      任意項目
      variables : 定義のフロー変数の初期値の上書き
      ssl       : SSL通信設定 (trueの場合にはhttps通信、falseの場合にはhttp通信)

  <パラメータ設定例>
      var execkey   = "Qanat1988";
      var server    = "server.ibat.co.jp:80"; //(※ポートの指定は任意)
      var menu      = "Menu01";
      var folder    = "Folder01";
      var def       = "販売管理DB更新";
      var variables = [{"FLOW_MESSAGE":"ABC"}, {"VAR2":"123"}];
      var ssl       = true;

  <!-->
  ③ <script type="text/javascript">
      ③-① var execkey   = "Execkey";           // キー情報
          var server    = "server.ibat.co.jp"; // サーバーアドレス
          var menu      = "Menu";             // メニュー
          var folder    = "Folder";           // フォルダ
          var def       = "Def";              // 定義名

          var variables = [];                 // フロー変数
          var ssl       = true;               // https通信設定

      ③-② function buttonClick() {
          var button = document.getElementById("QanatExecuteButton");

          // 定義終了までボタンを無効化
          button.disabled = true;

          // 定義を実行
          flowExecSSL(button, execkey, server, menu, folder, def, variables, ssl);
        }
  </script>
</head>
<body>
  <h1>Qanat Execute.js テスト</h1>
  ④ <input id="QanatExecuteButton" type="button" onclick="buttonClick()" value="実行">
</body>
</html>

```

sample.html (ファイル内容)

**【QanatExecute.js ファイルの読み込み】**

QanatExecute.js ファイルを相対パスで指定します。

**【パラメータ設定例】**

- で設定するパラメータの設定例をコメントで記述しています。

**【スクリプト部分】**

フロー定義の実行に必要なパラメータの設定と、ボタンクリック時に実行される関数を記述します。

- **【パラメータ設定部分】**

**[必須パラメータ]**

execKey : Qanat Server に配置した qanat\_rest.xml ファイルに記述されたキー情報  
sync : 定義を実行する際、終了まで待つ（非同期/同期）かどうかの設定値  
server : サーバーアドレス（IP またはドメイン名）  
menu : 定義が含まれるメニュー名  
folder : 定義が含まれるフォルダ名  
def : 定義名

**[任意パラメータ]**

variables : フロー変数(例 : [フロー変数\_文字 : "Abc", フロー変数\_数値 : 123])  
ssl : https 通信設定(true もしくは空文字 : https を使用、false : http を使用)



任意パラメータを使用しない場合には、空文字（" "）を設定してください。

- **【関数部分】**

ボタンクリック時の処理を記述します。

1 行目に記述された " buttonClick() " は、 で設定した onclick と一致する必要があります。

2 行目に記述された " QanatExecuteButton " は、 で設定した id と一致する必要があります。

**【ボタンの作成】**

画面に表示するボタンの作成を行います。

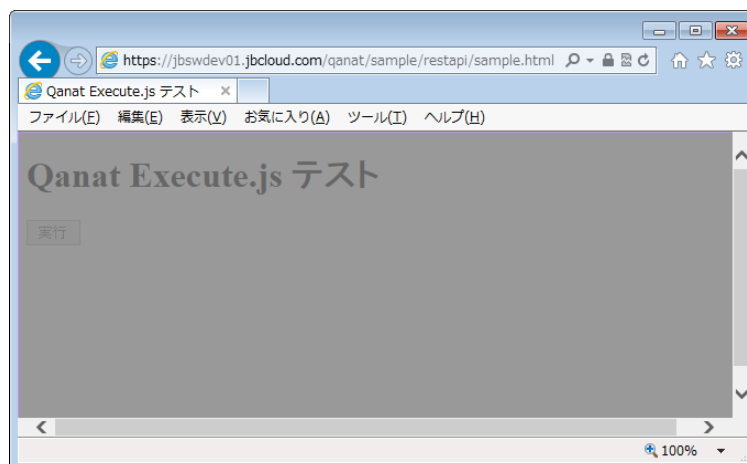
id : ボタンを一意に特定するための名称を記述します。

onclick : ボタンクリック時に呼び出される関数名を記述します。

value : ボタンの表面に表示する内容を記述します。

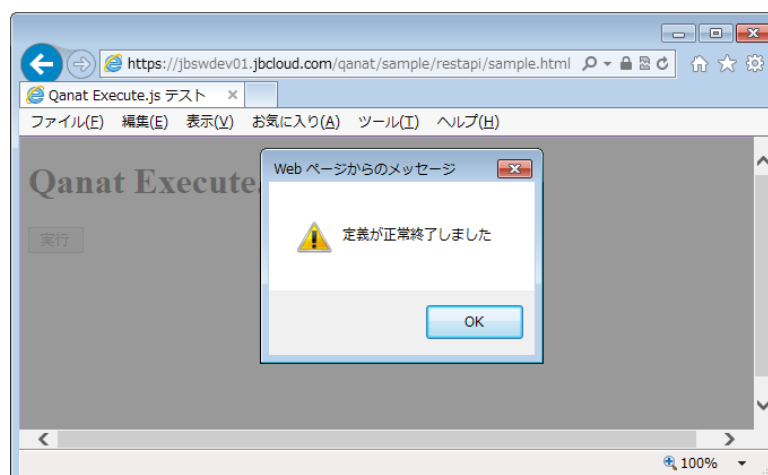
sample.html ファイルの「実行」ボタンをクリックした際の動作は以下となります。

1. 画面が暗転し、「実行」ボタンが無効になります。



「実行」ボタンクリック時の動作 1

2. 定義の実行が終了すると、アラートが表示されます。



「実行」ボタンクリック時の動作 2

3. アラートの OK ボタンを押すと、画面の暗転が元に戻り「実行」ボタンが有効になります。



「実行」ボタンクリック時の動作 3



## 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。



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。



→123  
FORMAT

### 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

→12.3  
FORMAT

### 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利用してください。



## 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
コード 1	134521	13:45:21
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
コード 1	20010807134521	2001/08/07 13:45:21
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
レコード 1	10500500	10,500,500
レコード 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>



## 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
レコード 1	/12,200	12200
レコード 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>



## 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
レコード 1	S10010	スムーズ コンパス	S10010/スムーズ コンパス
レコード 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

正規表現で指定：検索文字列を正規表現で指定するかどうかのチェックボックス

検索文字列      : 置換したい文字列

¥エスケープを使用する：置換後文字列に¥エスケープを利用するかのチェックボックス

置換後文字列    : 置換後の文字列

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す。

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

正規表現を指定した場合には、`¥r¥n` などの改行コード文字列を改行コードに置換することができます。その際には、置換後文字列に `¥エスケープ` を利用します。

例： 文字列：`¥r¥n` を 改行コード `¥r¥n` に置換する場合

正規表現で指定：チェックあり

置換前     : `¥¥r¥¥n`

¥エスケープを使用する：チェックを入れる

置換後     : `¥r¥n`

例：Unicode エスケープシーケンスの使用例

文字列：~ (文字コード：Shift-JIS) を文字コード：Windows-31j で出力する際に

文字化けさせたくない場合

正規表現で指定：チェックなし

置換前     : ~

¥エスケープを使用する：チェックを入れる

置換後     : `¥uff5e`



## 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
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	サンカクジョウギ	サンカクジョウギ

a→A  
ABC

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

A→a  
ABC

## 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： 商品名の英小文字を英大文字に変換します。

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

A1B2  
→12  
ABC

## 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
コード 1	C100000001	100000001
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
コード 1	スムーズ コンパス	スムーズコンパス
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
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。
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。

型変換フィルタ | 文字処理フィルタ | 数値処理フィルタ | 日付時刻処理フィルタ | 定数出力フィルタ | その他処理フィルタ | 制御フィルタ



### 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
レコード 1	36,012	6012
レコード 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
レコード 1	41.457	41.46
レコード 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・日付時刻の曜日を入力します。

	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。



## 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せん。



## 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せん。



## 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せん。



## 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。



## ユーザー ID

入力：引数なし

プロパティ：

表示名

出力：文字型

説明：実行時のユーザー ID を出力します。

ログインしているユーザー ID を取得します。



## 定義名称

入力：引数なし

プロパティ：

表示名

出力：文字型

説明：実行する定義名称を出力します。



## マッパー名称

入力：引数なし

プロパティ：

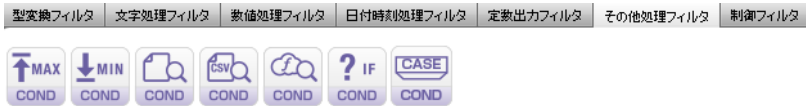
表示名

出力：文字型

説明：実行するマッパー名称を出力します。

## 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

フィルタ名: マスター参照

表示名:

参照マスター情報:

No.	キー項目
1	<input type="text"/>

出力フィールド名:

2件以上のレコードがキー項目に合致した場合の出力内容

エラーを出力     1件目を出力

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

2 件以上のレコードがキー項目に合致した場合の出力内容：

エラーとする または 見つかった 1 件目を出力する のどちらかから選択します。

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 または 1 件目出力を選択
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。



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

フィルタ名: CSVマスター参照

表示名: CSVマスター参照

参照マスター情報: 入力元が選択されていません

選択

No.	キー項目
1	

出力フィールド名: 追加 削除

2件以上のレコードがキー項目に合致した場合の出力内容

エラーを出力  1件目を出力

OK キャンセル

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

複数のデータが合致した場合 エラーにするか、1件目を出力するか選択できます

出力：文字型

2件以上のレコードがキー項目に合致した場合の出力内容：

エラーとする または 見つかった1件目を出力する のどちらかから選択します。

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 もしくは 1 件目出力を選択
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。



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。



## Salesforce マスター参照

Salesforceマスター参照

フィルタ名: Salesforceマスター参照

表示名: Salesforceマスター参照

参照マスター情報: 入力元が選択されていません

選択

No.	キー項目
1	

出力フィールド名: 追加 削除

2件以上のレコードがキー項目に合致した場合の出力内容

エラーを出力  1件目を出力

OK キャンセル

Salesforce 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 Salesforce 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

2件以上のレコードがキー項目に合致した場合の出力内容：

エラーとする または 見つかった1件目を出力する のどちらかから選択します。

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： 取引先名フィールドと取引先種別フィールドを条件として、取引先テーブルより年間売上を取得します。

	入力		出力
レコード 1	株式会社A	パートナー	1,210,000,000
レコード 2	株式会社B	小売業	520,000,000

#### 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 または 1件目出力を選択
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) 売上参照

マスターにキーが登録されていなかったレコードの年間売上フィールドには"該当無し"を出力

IF フィルタ入力前(=Salesforce マスター参照出力後)の状態

	入力	出力
レコード 1	株式会社 A	1,210,000,000
レコード 2	株式会社 B	null

IF フィルタで

1 番目 Salesforce 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	1,210,000,000	1,210,000,000
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年間売上フィールドには"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。



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。



## kintone マスター参照

kintone 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 kintone アプリ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

2 件以上のレコードがキー項目に合致した場合の出力内容：

エラーとする または 見つかった 1 件目を出力する のどちらかから選択します。

キャッシュのコントロール：

キャッシュせずに読み込みレコード毎に kintoneAPI をコールする場合チェックを入れます。

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： 都道府県フィールドと市区町村フィールドを条件として、場所別人口管理アプリより人口を取得します。

	入力		出力
レコード 1	神奈川県	横浜市	3,733,234
レコード 2	大阪府	大阪市	2,713,157

## 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 または 1 件目出力を選択
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) 売上参照

マスターにキーが登録されていなかったレコードの年間売上フィールドには"該当無し"を出力

IF フィルタ入力前(=kintone マスター参照出力後)の状態

	入力	出力
レコード 1	神奈川県	3,733,234
レコード 2	大阪府	null

IF フィルタで

1 番目 kintone 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	3,733,234	3,733,234
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年間売上フィールドには"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。



本フィルターは、kintone アダプターのライセンスがない場合にはご利用いただけません。

以下のフィールドタイプは、使用できません。

\_\_REVISION\_\_ : リビジョン、CREATOR : 作成者、MODIFIER : 更新者、  
 CALC : 計算、RICH\_TEXT : リッチエディター、CHECK\_BOX : チェックボックス、  
 MULTI\_SELECT : 複数選択、FILE : 添付ファイル、USER\_SELECT : ユーザー選択、  
 ORGANIZATION\_SELECT : 組織選択、GROUP\_SELECT : グループ選択、  
 CATEGORY : カテゴリ、STATUS\_ASSIGNEE : 作業者、  
 REFERENCE\_TABLE : 関連レコード、GROUP : グループ、LABEL : ラベル、  
 SPACER : スペース、HR : 罫線、SUBTABLE : サブテーブル



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。



## 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

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。このチェックが有りの場合、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

型変換フィルタ | 文字処理フィルタ | 数値処理フィルタ | 日付時刻処理フィルタ | 定数出力フィルタ | その他処理フィルタ | 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。



### 条件付きループ

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。

is Not Null : 入力 2 が NULL でない場合実行されます。この場合型や比較値は無視されます。

is Not Error : 入力 2 が Is Not Error (フィルタ処理エラーでない) 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

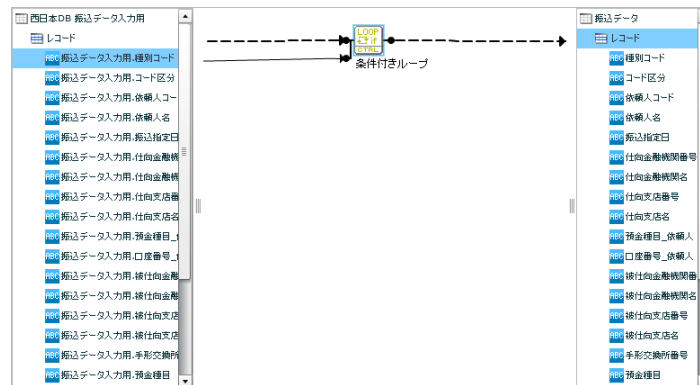
条件： >= を指定 (売上金額(入力 2)>=2000)

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)



条件の型に 日付、時刻、日付時刻 型を選択した場合には 比較値は 区切り文字( - や / 、 :) を外して入力する必要があります。

例： 日付型 2014-01-01 を比較値として扱いたい場合 比較値：20140101

時刻型 12:00:00 を比較値として扱いたい場合 比較値：120000

日付時刻型 2014-01-01 12:00:00 を比較値として扱いたい場合 比較値：20140101120000



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 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	PostgreSQL	MySQL	データ型の説明
(CHAR)	(CHAR)	(CHAR)	-	(CHAR)	(BPCHAR)	(CHAR)	短い固定長の文字列
-	-	-	(A)	-	-	-	英数カナ半角
-	-	-	(O)	-	-	-	オープン
-	-	-	(J)	-	(CHAR) (*3)	-	専用
-	-	-	(E)	-	-	-	択一
-	(NCHAR) (*2)	(NCHAR) (*2)	-	-	-	-	ユニコード固定文字列
(VARCHAR)	(VARCHAR2)	(VARCHAR)	-	(VARCHAR)	(VARCHAR)	(VARCHAR)	可変長の文字列 (*1)
	(VARCHAR)						
(LONG VARCHAR)	(LONG)	(TEXT)	-	(LONG VARCHAR)	(TEXT)	(LONG VARCHAR)	長い可変長の文字列 (*1)
		(VARCHAR(MAX))					
-	(NVARCAHR2) (*2)	(NVARCAHR) (*2)	-	-	-	-	ユニコード可変長文字列
		(NVARCAHR(MAX))					
		(NTEXT) (*2)					
-	-	-	-	-	-	-	
(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)他の DB の CHAR 型とは異なり、格納サイズが 1 バイトで固定の単一バイト内部データ型です。

## 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	PostgreSQL	MySQL	データ型の説明
(INTEGER)	(NUMBER)	(INT)	-	(INTEGER)	(INT4)	(INTEGER)	-2147483648 から 2147483647 までの 32 ビットの符号付き整数値
-	-	-	-	-	-	-	
(BIGINT)(*2)	(NUMBER)(*2)	(BIGINT)(*2)	-	(BIGINT)(*2)	(INT8)	(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)	(INT2)	(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
		× (BIT)					
(DECIMAL)	(NUMBER)	(DECIMAL)	-	-		(DECIMAL)	10 進数
		(MONEY)					
		(SMALLMONEY)					
-	(NUMBER)	(NUMERIC)	-	-	(NUMERIC)	-	
-	-	-	(P)	(DECIMAL)			バック 10 進(*1)
-	-	-	(S)	(NUMERIC)			ゾーン 10 進(*1)
(REAL)	(NUMBER)	(REAL)	(F)	(REAL)	(FLOAT4)	(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)	(FLOAT8)	(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。

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	PostgreSQL	MySQL	データ型の説明
(DATE)	(DATE) (*1)	-	(L)	(DATE)	(DATE)	(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	PostgreSQL	MySQL	データ型の説明
(TIME)	-	-	(T)	(TIME)	(TIME)	(TIME) (*2)	時刻、分、秒

注(\*2) MySQL の TIME 型は 2 桁の時分秒をサポート

## 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	PostgreSQL	MySQL	データ型の説明
(TIMESTAMP)	(DATE) (*6)	(DATETIME)	(Z)	(TIMESTAMP)	(TIMESTAMP)	(TIMESTAMP)	DATE と TIME とナノ秒
	(TIMESTAMP)					(DATETIME)	
	× (TIMESTAMP with ...)				× (TIMESTAMP with ...)		
		O(SMALLDATETIME)					

付録 C 便利な機能

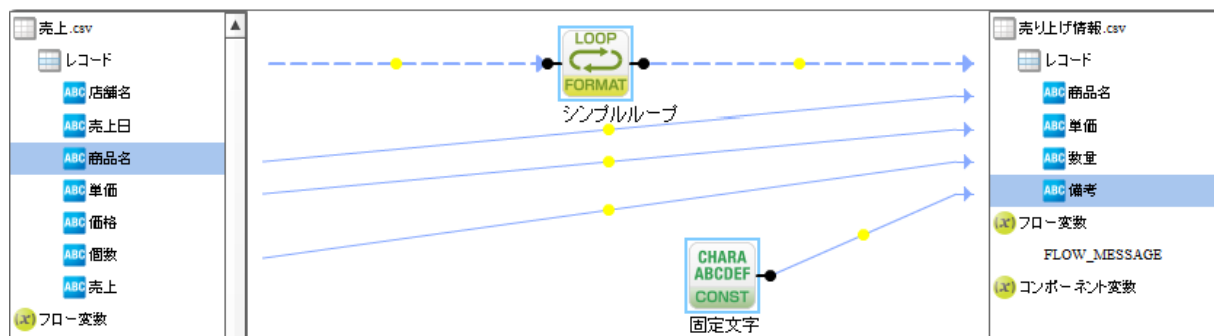
## C 便利な機能

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

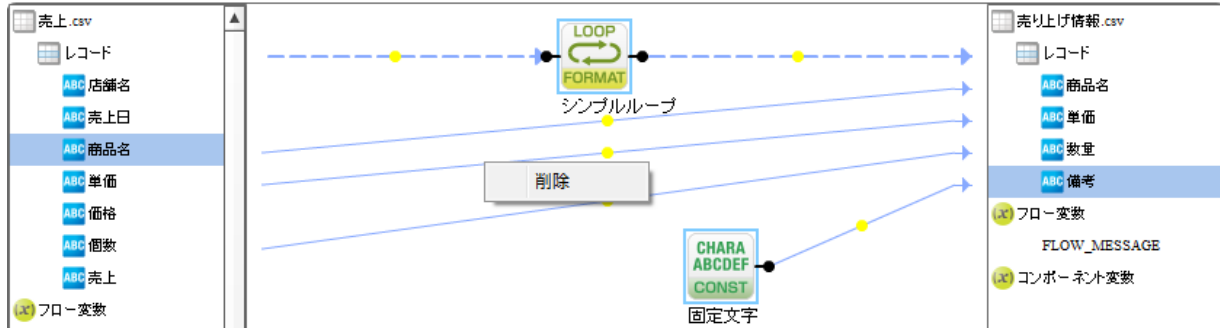
### 範囲移動

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フローが全て同時に移動します。

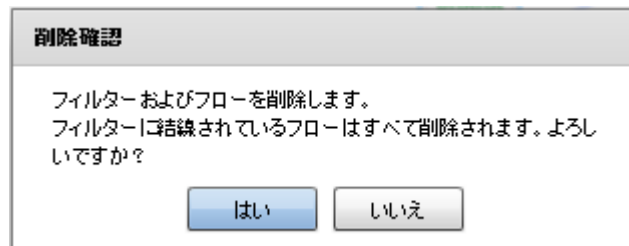
データリソースに直接つながっているフローは移動の対象外になります。

## 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。



2. 削除を選択します。



3. 「はい」を押します。 削除

付録 D REST API

# D REST API について

## REST API の概要

Qanat 2.0 では以下の API を REST API として提供しています。

V1.10.0 以降のバージョンで使用可能な API

- ・ 定義の実行 (POST)
- ・ 実行結果の確認 (GET)

## REST API の共通仕様

プロトコル : HTTPS (HTTP)

フォーマット : JSON

文字コード : UTF-8

リクエスト先 URI : `http(s)://[サーバーの FQDN]/qanat/api/[バージョン]/[サービス名]`

## レスポンス

HTTP ステータスコードが「200」であれば正常終了、それ以外はエラーです。

エラー時には、下記情報を含む JSON データをレスポンスとして受け取ります。

Response		
code	文字列	エラーコード
message	文字列	エラーメッセージ
details	配列	現在未使用

エラーレスポンスの例：

```
{
  "code": "QRA_FPE10000008"
  "message": "定義の実行許可がありません.サーバーの設定を確認してください"
  "details": []
}
```

リクエストおよびレスポンスデータの JSON フォーマットには今後フィールドやキーが追加される場合があります。

## 認証

定義の実行 (POST) を利用する際には認証設定が必要です。

認証の設定は Qanat Server の

<Tomcat のインストールディレクトリ>/webapps/qanat/config2/

ディレクトリにある "qanat\_rest.xml" を編集する必要があります。

```
<?xml version="1.0" encoding="UTF-8"?>
<restinfo>
  <!--
  *****
  MainVersion   製品のバージョン情報   ※変更不要
  *****
  -->
  <version>v1</version>
  <position>0</position>

  <!--
  *****
  Access-Control-Allow-Origin   アクセスを許可するドメイン名
                                ※必要に応じて変更してください
  *****
  -->
  <alloworigin>*</alloworigin>

  <!--
  *****
  reload   AllowDefines要素の自動読み込み設定
           1:変更を都度読み込み   0:サーバー起動時のみ読み込み
  *****
  -->
  <reload>1</reload>

  <!--
  *****
  AllowDefines   API使用を許可する定義の設定
                 許可する定義をallowdefine要素に設定してください
  *****
  REST APIの使用を許可する定義の設定例

  REST APIを利用するためのキーの値 : QANAT1988

  使用を許可する定義情報
  メニュー名 : Menu01
  フォルダ名 : Folder01
  定義名     : 販売管理DB更新

  定義を実行するユーザー
  ユーザーID : cvadmin
  パスワード : cvadmin

  ※authtypeは"default"を指定してください

  <allowdefine>
    <qanexckey>QANAT1988</qanexckey>
    <menu>Menu01</menu>
    <folder>Folder01</folder>
    <def>販売管理DB更新</def>
    <authentication>
      <authtype>default</authtype>
      <user>cvadmin</user>
      <password>cvadmin</password>
    </authentication>
  </allowdefine>

  *****
  -->
```

```

<allowdefines>
  <allowdefine>|
    <qanexeckey></qanexeckey>
    <menu></menu>
    <folder></folder>
    <def></def>
    <authentication>
      <authtype>default</authtype>
      <user></user>
      <password></password>
    </authentication>
  </allowdefine>
</allowdefines>

</rest info>

```

qanat\_rest.xml

## 認証の設定例

### フロー定義

- Menu01/Folder01/商品マスター連携
- Menu01/Folder02/販売データ連携

を定義の実行 API で使用するための設定例

```

<allowdefines>
  <allowdefine>
    <qanexeckey>QAN1234</qanexeckey>
    <menu>Menu01</menu>
    <folder>Folder01</folder>
    <def>商品マスター連携</def>
    <authentication>
      <authtype>default</authtype>
      <user>cvadmin</user>
      <password>cvadmin</password>
    </authentication>
  </allowdefine>
  <allowdefine>
    <qanexeckey>QAN5678</qanexeckey>
    <menu>Menu01</menu>
    <folder>Folder02</folder>
    <def>販売データ連携</def>
    <authentication>
      <authtype>default</authtype>
      <user>user001</user>
      <password>p@ssw0rd</password>
    </authentication>
  </allowdefine>
</allowdefines>

```

- Menu01/Folder01/商品マスター連携

は qanExecKey:QAN1234 で呼び出すことで ユーザー cvadmin で実行される

- Menu01/Folder02/販売データ連携



は qanExecKey:QAN5678 で呼び出すことで ユーザー user001 で実行される

定義の実行許可がないユーザーを登録し、API から呼び出した場合はエラーとなります。

## 定義の実行（POST）

リクエストパラメータにて指定したフロー定義が実行できます。

前提として、qanat\_rest.xml に定義の認証設定をしておく必要があります。

HTTP メソッド： POST

URI : http(s)://[サーバーのHOST]/qanat/api/[バージョン]/flow

リクエストパラメータ

Request				
1	キー	qanExecKey	文字列	実行のためのキー
2	同期/非同期	sync	trueまたはfalse	定義実行の同期/非同期選択
3	メニュー名	menu	文字列	実行定義の保存メニュー名
4	フォルダ名	folder	文字列	実行定義の保存フォルダ名
5	定義名	def	文字列	実行定義名
6	フロー変数	variables	配列	フロー変数の初期値設定

フロー変数の指定は任意となります。

また、フロー変数を使用する場合には実行するフロー定義にフロー変数を設定しておく必要があります。

フロー変数が設定してある定義に、フロー変数を渡さないで呼び出した場合

定義に設定されている初期値で動作いたします。

リクエストボディの構造

```
{
  "qanExecKey" : "( qanat_rest.xml で指定したキー )",
  "sync"       : true または false
  "flow" : {
    "menu" : "(メニュー名)",
    "folder" : "(フォルダ名)",
    "def" : "(定義名)",
    "variables" : [
      { "day" : "20160328" },
      { "month" : "3" }
    ]
  }
}
```

レスポンスパラメータ（非同期処理（sync = false）の場合）

Response				
1	実行ID	execId	数値	実行した定義の実行ID

レスポンスの構造

```
{
  "execId" : (実行番号)
}
```

レスポンスパラメータ（非同期処理（sync = true）の場合）

Response				
1	実行ID	execId	数値	実行した定義の実行ID
2	結果コード	code	数値	実行した定義の結果コード

レスポンスの構造

```
{
  "execId": (実行番号)
  "code" : (コード)
}
```

## 結果の確認（GET）

指定した実行番号で処理されたフロー定義の結果確認ができます。

HTTP メソッド： GET

URI : http(s)://[サーバーの FQDN]/qanat/api/[バージョン]/flow/status

リクエストパラメータ

Request				
1	実行番号	execId	数値	フロー定義の実行番号

「execId」のパラメータを HTTP のクエリ文字列として送信します。

クエリ文字列 : execId=100

リクエストの例 : http(s)://[サーバーの FQDN]/qanat/api/[バージョン]/flow/status?execId=100

レスポンスパラメータ

Response				
1	コード	code	数値 or 空白	0:正常 1:警告 -2:エラー 空白:実行中
2	実行結果	result	文字列	正常終了、警告、エラー、実行中のメッセージ

レスポンスボディの例

```
{
  "code" : (コード)         ・・・0 : 正常終了 1 : 警告 2 : エラー 空文字 : 実行中
  "result" : "(実行結果のメッセージ)"
}
```

Qanat 2.0 ユーザーズマニュアル

2010年 1月29日 第1版発行

2019年 04月17日 第23版発行

発行 JB アドバンスト・テクノロジー株式会社

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 アドバンスト・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。