

zipcloud 郵便番号データ
ダウンロード & CSVデータ
取り込み
(Magic xpi 4.13版)



OUTPERFORM THE FUTURE™

目次: zipcloud郵便番号データダウンロード&CSVデータ取り込み

第1章 zipcloud郵便番号データダウンロード

- 1.1 zipcloudの郵便番号データ(加工済バージョン)
- 1.2 REST Client : zipcloudのホームページのHTMLの取得
 - 1.2.1 リソースの設定
- 1.3 ダウンロードURLの取得
- 1.3 ダウンロードファイル名の取得
- 1.4 更新日付の取得
- 1.5 REST Client : 郵便番号データ(加工済バージョン)zipファイルのダウンロード
- 1.6 zipファイルの解凍
- 1.7 flgファイル作成
- 1.8 ファイルコピー(csvファイルとflgファイル)
- 1.9 トリガーについて

第2章 CSVデータ取り込み

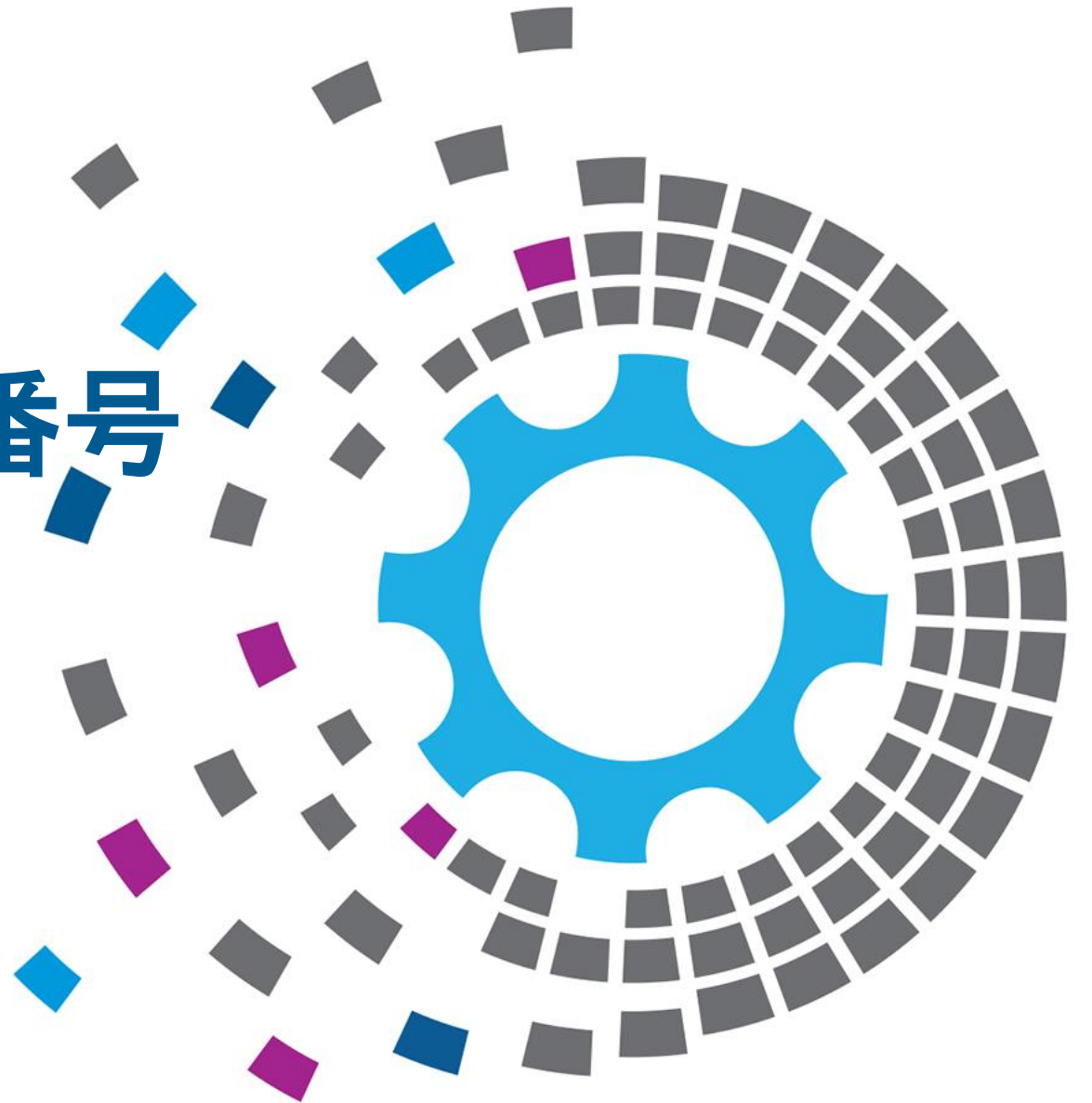
- 2.1 フロー「CSVデータ取り込みパターン1」
- 2.2 フロー「CSVデータ取り込みパターン2」

補足



OUTPERFORM THE FUTURE™


第1章 zipcloud 郵便番号 データダウンロード



OUTPERFORM THE FUTURE™

1.1 zipcloudの郵便番号データ(加工済バージョン)

- 加工済みバージョンのデータは、日本郵便のデータを元に独自の加工(「以下に掲載がない場合」を取り除いたり、不空数行にまたがって住所がセットされているのを1行にまとめるなど)を行っており、すぐに展開して利用することができる。(ちなみに、2023年6月より、日本郵便でも1データ1行、UTF-8エンコーディングの形式のCSVデータのダウンロードが開始された)
- このサンプルプロジェクトでは、zipcloudの加工済バージョンの郵便番号データ(x-ken-all.csv)の最新版をダウンロードし、データベースの郵便番号用のマスタテーブルに取り込む処理のシナリオを、実装したものである。



The screenshot shows a web browser window with the URL zipcloud.ibsnet.co.jp. The page displays download options for postal code data. A table shows a file named 'x-ken-all202306.zip' with a size of 1.7MB (1,734,573 bytes) and 127,399 rows. Below this, there is a section titled '郵便番号データ(加工済バージョン)とは' (What is the processed version of postal code data?). It explains that the data is processed from publicly available data on the Japan Post website, removing certain entries. A table below shows the processing rules for missing data.

	郵便番号	都道府県名	市区町村名	町域名
加工前	6000000	北海道	札幌市中央区	以下に掲載がない場合



OUTPERFORM THE FUTURE™

1.2 zipcloudのホームページのHTMLの取得

- zipcloudのホームページのHTMLを取得する。
- このHTMLには、郵便番号データ(加工済バージョン)のダウンロードURL、ダウンロードファイル名、更新日などの情報があり、それらの情報を取得していく
- 次ページにて、まずリソースの設定について解説する

The screenshot displays the '設定' (Settings) window of the Magic REST Client. The left sidebar shows a tree view with 'リソース' (Resources) expanded to 'REST Client_Zipcloud' (MGXPI413). The main area shows a table of resource settings:

#	名	タイプ	フォーマット	値
1	サーバURL	文字	2500	http://zipcloud.ibsnet.co.jp
2	セキュリティ	文字	30	None
3	クライアント証明書の要求	文字	1	No

Below the table, there is a text box with the following text: 'リソース >> REST Client >> REST Client_Zipcloud (REST Clientを使用すると、OAuth認証などのRestful APIの機能を利用できます。)'

At the bottom of the main window, there are buttons for 'アクセストークン' and 'パス' (highlighted with a red box). A curved arrow points from the 'パス' button to a secondary 'パス' dialog box.

The secondary 'パス' dialog box has a table for path segments:

#	パス名
1	/
2	/zipcodedata/download

Below this table is a 'Parameters' section with tabs for 'クエリパラメータ', 'ヘッダパラメータ', 'フォームパラメータ - リクエスト', and 'フォームパラメータ - レスポンス'. The 'クエリパラメータ' tab is active, showing a table with columns for '#', 'パラメータ', and 'デフォルト値'. At the bottom of the dialog are buttons for 'パラメータ', '削除', 'コピー', and 'キャンセル'.



OUTPERFORM THE FUTURE™

1.2.1 リソースの設定

- REST Clientのzipcloud用のリソース: REST Client_zipcloud
- 「REST Client_zipcloud」リソースの主なパラメータ

設定

リソース

REST Client_zipcloud

Database_zipcloud

サービス
カレンダー
プロジェクト環境
一般環境

#	名	タイプ	フォーマット	値
1	サーバURL	文字	2500	http://zipcloud.ibsnet.co.jp
2	セキュリティ	文字	30	None
3	クライアント証明書 の要求	文字	1	No

①: zipcloudのホームページのURL (http://zipcloud.ibsnet.co.jp)

リソース >> REST Client >> REST Client_zipcloud (REST Clientを使用すると、Oauth認証、multipart/form-data、send/receive、添付ファイルなどのRestful APIの機能を利用できます。)

アクセストークン **パス** [パス]ボタンをクリック ⇒ 次ページに続く

追加 削除 ロード 適用 OK キャンセル



OUTPERFORM THE FUTURE™

1.2.1 リソースの設定 - パス①

- REST Client_zipcloudのパス:「/」

①: zipcloudのホームページ直下のパス

クエリパラメータ、ヘッダパラメータ、フォームパラメータ (2つ) の各タブには、パラメータは設定していない

#	パス名
1	/
2	/zipcodedata/download

#	パラメータ	デフォルト値
---	-------	--------

#	Name	Part Type	Content-Type	Encoding
---	------	-----------	--------------	----------

#	パス名	Part Type	Content-Type	エンコーディング
---	-----	-----------	--------------	----------



OUTPERFORM THE FUTURE™

1.2.1 リソースの設定 - パス②

- REST Client_zipcloudのパス②:「/zipcodedata/download」

②: zipcloudのダウンロードURLのベースのパス

クエリパラメータに「di」を設定している値は、加工済みバージョンのダウンロード用のコード

ヘッダパラメータ、フォームパラメータ(2つ)の各タブには、パラメータは設定していない

#	パス名
1	/
2	/zipcodedata/download

#	パラメータ	デフォルト値
1	di	1688112638794

#	Name	Default Value
---	------	---------------

#	パス名	Part Type	Content-Type	エンコーディング
---	-----	-----------	--------------	----------



1.2.1 リソースの設定 - パス② の補足説明

- REST Client_zipcloudのパス②:「/zipcodedata/download」の補足説明

HTMLソース表示画面:加工済バージョンの箇所のHTMLソースのスクリーンショット

拡大

ipcodedata/download?di= 3 / 3

ipcodedata/download?di= 3 / 3

オレンジ色の箇所がダウンロードURLで、di=の後ろの数値がダウンロードURLごとに異なる。
~di=までのURL文字列は、このページに3カ所あり、3つ目が加工済バージョンのダウンロードURLである

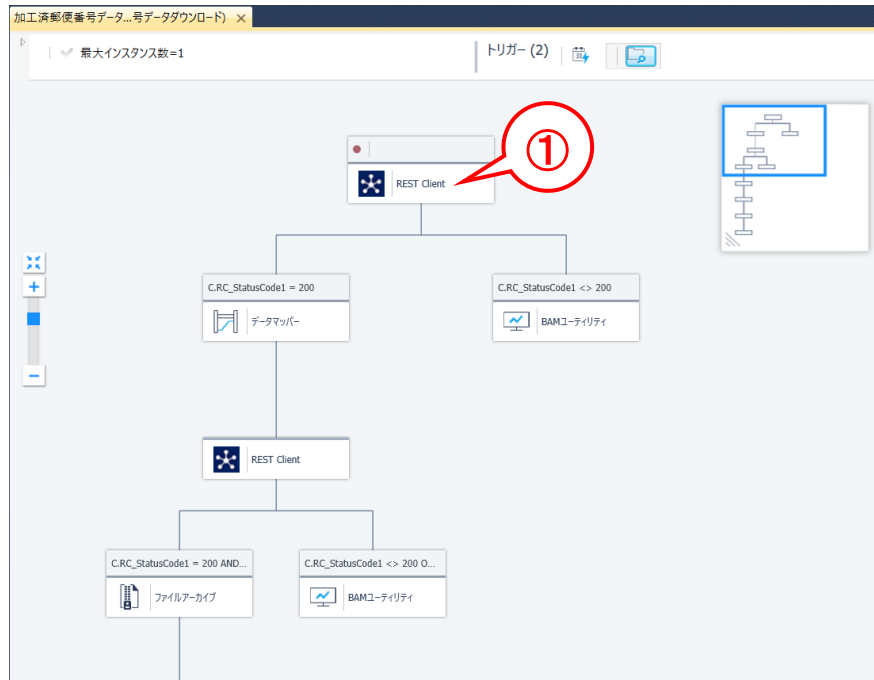
```
<div class="dlEntry">
  <div class="dlTitle">全国一括データ（加工済バージョン）</div>

  <div class="dlDetail">
    <table cellspacing="0">
      <tr>
        <th>ファイル名</th>
        <td class="fnColumn">x-ken-all202306.zip</td>
        <th>公開日</th>
        <td>2023年6月30日</td>
      </tr>
      <tr>
        <th>サイズ</th>
        <td>1.7MB (1,734,573バイト) </td>
        <th>行数</th>
        <td>127,399</td>
      </tr>
      <tr>
        <th>MD5</th>
        <td colspan="3">b037ba76e2b7f3</td>
      </tr>
    </table>
  </div>

  <div class="dlButton">
    <a href="/zipcodedata/download?di=1688112638794" rel="nofollow" onclick="javascript:
pageTracker._trackPageview('/zipcodedata/download/x-all/1688112638794');"></a>
```

1.2 zipcloudのホームページのHTMLを取得する

- フロー「加工済郵便番号データzipのダウンロード」の第一ステップであるREST Clientアダプタ①の解説



① : zipcloudのホームページのHTMLを取得するREST Clientのステップ
REST Clientのリソースは「Rest Client_zipcloud」を選択

プロパティ

REST Client ステップ特性

完了を待つ	No
インタフェース	XML
アドバンスド	
セーブポイント	No
コールロジックフロー	No
ロギング	
ロギングスコープ	No
ステップロギングオプション	前
メッセージ	
Blob	
ファイル拡張子	
設定	
リソース名	REST Client_zipcloud

続く



OUTPERFORM THE FUTURE™

1.2 zipcloudのホームページのHTMLを取得する(続き)

- REST Clientアダプタ①の解説(続き)
- 「REST Client_zipcloud」リソースのパス①を使用する。

• REST Clientアダプタ①の設定画面

• zipcloudのホームページHTMLソース

REST Client 構成

番号を振ったパラメータの説明は次ページに続く

接続
リソース名: REST Client_Zipcloud

オペレーション
パス: / ①

オペレーション: Get ②

セキュリティ: No

リクエスト Content-Type: text/html ③

レスポンス Content-Type: text/html ④

要求エンコーディング: UTF-8

出力
結果保存: Variable ⑤ F.zipcloudダウンロードページ_HTML_Bib

ヘッダフィルタ:

ヘッダ: C.RC_HeaderBlob1

ステータスコード: C.RC_StatusCode1

メッセージ: C.RC_Message1

成功: Variable C.RC_IsSuccess

パラメータ スキーマ更新 OK キャンセル

```
4  
5  
6 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
7 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
8 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ja" lang="ja">  
9 <head>  
10 <meta http-equiv="content-type" content="text/html; charset=UTF-8" />  
11 <title>郵便番号データのダウンロード - zipcloud</title>  
12 <meta name="description" content="株式会社アイビスが運営する郵便番号データ配信サービス「zipcloud」。Google App Engine 上で稼働しています。"/>  
13 <meta name="keywords" content="郵便番号データ, ダウンロード, ZIP形式, 加工済バージョン, 更新通知, 郵便番号検索API, zipcloud" />  
14 <link rel="SHORTCUT ICON" href="/favicon.ico" />  
15 <link rel="stylesheet" type="text/css" href="/css/common.css?v=20150825" />  
16 <script type="text/javascript" src="/js/jquery-1.3.2.min.js"></script>  
17 <script type="text/javascript" src="/js/jquery.curvycorners.packed.js">  
18 </script>  
19 <script type="text/javascript" src="/js/rollover.js"></script>  
20 <script type="text/javascript">  
21 $(document).ready(function() {  
22     $('#SideBox').corner({  
23         tl: {radius: 8},  
24         tr: {radius: 8},  
25         bl: {radius: 8},  
26         br: {radius: 8}  
27     });  
28 </script>  
29 </head>
```

エンコーディングを合わせる

1.2 zipcloudのホームページのHTMLを取得する(続き)

- 設定画面のパラメータ詳細:

①: REST Client_zipcloudリソースに登録したパスをコンボボックスから選択する
ここでは「/」を選択

②: 「Get」メソッドを選択

リクエストContentは空白のまま

③: レスポンスContentは「text/html」を横の[...]のボタンをクリックして表示される画面から選択

④: HTMLのエンコーディング(charsetの箇所)と一致するエンコーディングを選択

⑤: 取得したHTMLを保存する変数を選択



OUTPERFORM THE FUTURE™

1.2 zipcloudのホームページのHTMLを取得する

REST Clientアダプタ①の設定画面(続き)

REST Client 構成

接続
リソース名: REST Client_zipcloud

オペレーション
パス: /
オペレーション: Get
セキュリティ: No
リクエスト Content Type:
レスポンス Content Type: text/html
要求エンコーディング: UTF-8 Charset:
出力
結果保存: Variable F.zipcloudホームページ_HTML_Blib
ヘッダフィルタ:
ヘッダ: C.RC_HeaderBlob1
ステータスコード: C.RC_StatusCode1
メッセージ: C.RC_Message1
成功: Variable C.RC_IsSuccess

パラメータ スキーマ更新 OK キャンセル

- ① : レスポンスヘッダの保存先変数 (Blobタイプ) を選択
- ② : レスポンスのステータスコードの保存先変数 (Numericタイプ) を選択
- ③ : レスポンスメッセージの保存先変数 (Alphaタイプ) を選択
- ④ : リクエストの成否の保存先の指定 (ここでは変数 (変数) を選択) Logicalタイプ変数を選択

これらのパラメータにはコンテキスト変数を用意している

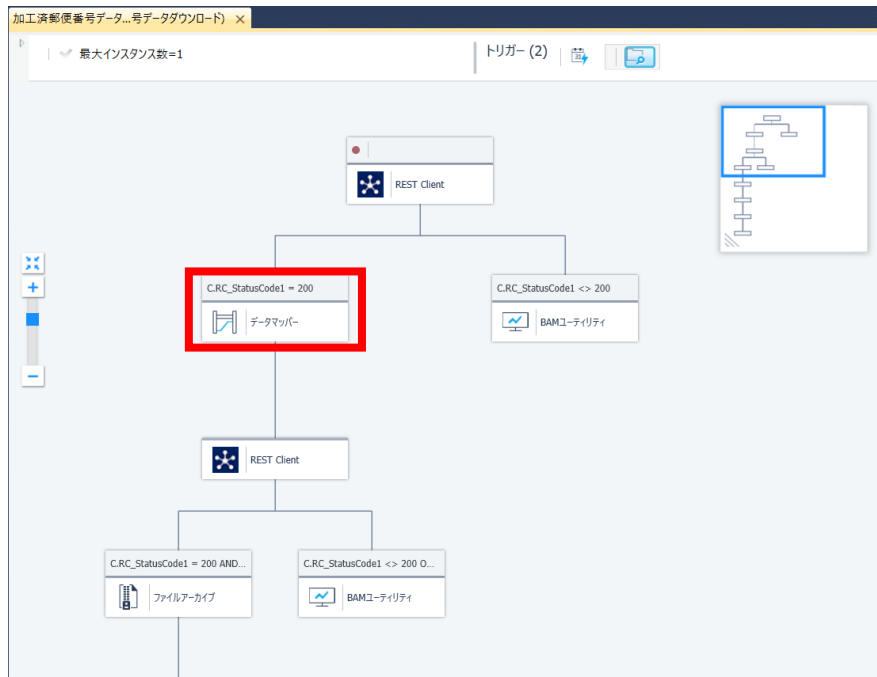
#	名	説明	タイプ	長さ	デフォルト値
22	C.RC_HeaderBlob1		Blob		
23	C.RC_StatusCode1		Numeric	3.0	
24	C.RC_Message1		Alpha	10000	
25	C.RC_IsSuccess		Logical	1	



OUTPERFORM THE FUTURE™

1.3 ダウンロードURLの取得

- zipcloudのホームページのHTMLから、最新の加工済みバージョンの郵便番号データをダウンロードするための、ダウンロードURLを取得する
- 「1.2.1 リソースの設定 - パス②の補足説明」にて、郵便番号データ（加工済みバージョン）のダウンロードURLは、「zipcodedata/download?di=」の文字列の3番目のもの（2023年6月末のは「zipcodedata/download?di=1688112638794」である）
- つまり、3番目の「zipcodedata/download?di=」を含む文字列（37文字）を取得すれば良い



データマッピングステップの条件の補足説明：
条件には「**C.RC_StatusCode1 = 200**」と設定

これは、REST Clientアダプタの戻り値のステータスコード（HTTPレスポンスステータスコード）に設定している変数「C.RC_StatusCode1」の値が正常終了を示す「200」であることを判定し、Trueであればこのステップが実行される

分岐しているもう一方のBAMユーティリティアダプタのステップの条件はステータスコードが200以外の場合に実行される



OUTPERFORM THE FUTURE™

1.3 ダウンロードURLの取得(続き)

- Magic xpiにはHTMLパーサは備わっていない
- そのため、HTMLを1行ずつ読み込み、行に指定する文字列を含むかどうかを調べて、目的の文字列を取得し、必要な文字を抽出する方法で対処している
- 使用するアダプタはデータマッパーで、送り元にはFlatFileを配置している
- FlatFileはhtmlの1行の長さを扱えるよう文字数を指定する。(1行の最大文字数)

データマッパー (加工済...-タzipのダウンロード) x 加工済郵便番号データ...号データダウンロード*

送り元 検索 Apply

FlatFile_1

Record (3)

Row (2)

フラットファイル

ファイルの例

タイプ ポジション 列区切文字 レコード区切り文字

#	名	データタイプ	フォーマット	From	長さ
1	Row	Alpha	1000	1	1000

プロパティ

FlatFile_1 フラットファイルスキーマ

スキーマ情報

ID 1

一般

送り元タイプ Variable

データ送り元エンコーディング UTF-8

変数 Ezipcloudホームページ_HTML_Bib

タイプ ポジション

列区切文字

レコード区切り文字

区切文字をデータに含める No

詳細 (コレクション)

名 FlatFile_1



1.3 ダウンロードURLの取得(続き)

- データマッパーの送り先にはVariables (変数) を配置
- この後の章で説明する、「F.ダウンロードファイル名」「F.更新日」の合計3つの情報を保存する変数をデータマッパーの送り先に配置している。
- ここでは、「F.ダウンロードURL」について解説する

Variables3の
プロパティ

ID	変数
3	F.ダウンロードURL

#	選択	変数名	タイプ
5	✓	F.ダウンロードURL	Alpha
6		F.ダウンロードファイル名	Alpha



1.3 ダウンロードURLの取得(続き)

- 「F.ダウンロードURL」のプロパティの計算値欄：

`MID (Src.S1/Record/Row , InStr (Src.S1/Record/Row , 'zipcodedata/download?di') , 37)`

MID関数で文字列を指定した位置から指定した文字数分抽出している。

InStr関数で「zipcodedata/download?di」を含む文字の開始位置を取得し（InStr関数）、そこから37文字の文字列を抽出

送り先	検索
fx	▶ X Variable_2
fx	▼ X Variable_3
fx ?	▼ Instance
fx	α FダウンロードURL
fx	▶ X Variable_4

プロパティ	F.ダウンロードURL Simple項目
▼ その他	
▼ 一般	
内容のエンコーディング	Unicode
内部データタイプ	Alpha
計算値	MID (Src.S1/Record/Row , InStr (Src.S1/Record/Row , 'zipcodedata/download?di') , 37)
外部データタイプ	Alpha
書式	200
Nullable	False
常にカスタム書式を使用	No
説明	
パス	Instance/FダウンロードURL

Destination/Instance/FダウンロードURL : 計算値 [Alpha]

MID (Src.S1/Record/Row , InStr (Src.S1/Record/Row , 'zipcodedata/download?di') , 37)



1.3 ダウンロードURLの取得(続き)

- 「Variables3」 の下の 「Instance」 フォルダのプロパティ：
マルチアップデートに「F.カウンタ」を設定。式は「F.カウンタ + 1」
条件は、「InStr (Src.S1/Record/Row , 'zipcodedata/download?di=') > 0」
(Src.S1/Record/Rowに文字列zipcodedata/download?di=を含む場合) F.カウンタをカウントアップする (zipcloudのホームページのhtmlにはこの文字列が3か所ある)

「Instance」フォルダのプロパティ

マルチアップデートのコレクション詳細

「F.カウンタ」の条件の式エディタ画面

#	変数名	式	条件
1	F.カウンタ	F.カウンタ + 1	InStr (Src.S1/Record/Row , 'zipcodedata/download?di=') > 0

Destination/Instance : 条件 [Logical]

F.カウンタ = 3 AND InStr (Src.S1/Record/Row , 'zipcodedata/download?di=') > 0



1.3 ダウンロードURLの取得(続き)

- 「Variables3」 の下の 「Instance」 フォルダのプロパティ :

条件欄に設定している計算式は下記の通り

$F.カウント = 3 \text{ AND } \text{InStr} (\text{Src.S1/Record/Row} , 'zipcodedata/download?di=') > 0$

Src.S1/Record/Rowに文字列zipcodedata/download?di=を含む場合でかつF.カウントが3の場合 (3か所目である) に条件が成立し、「Variable3」の「Instance」の下の「F.ダウンロードURL」が実行される (計算値欄の計算結果がF.ダウンロードURLにセットされる)

The screenshot shows the Magic software interface. On the left, a tree view shows the project structure with 'Instance' under 'Variable_3' highlighted. A red box highlights the 'Instance' folder, and a red arrow points to the 'プロパティ' (Properties) window. The 'プロパティ' window shows the '条件' (Condition) tab with the formula $F.カウント = 3 \text{ AND } \text{InStr} (\text{Src.S1/Record/Row} , 'zipcodedata/download?di=') > 0$ entered. A red box highlights this formula in the '条件' tab. Below the 'プロパティ' window, the '条件の式エディタ画面' (Formula Editor) is shown with the same formula entered in a text field. A red box highlights the formula in the editor.

※条件欄よりも先にマルチアップデートの方が計算処理される (補足のh)



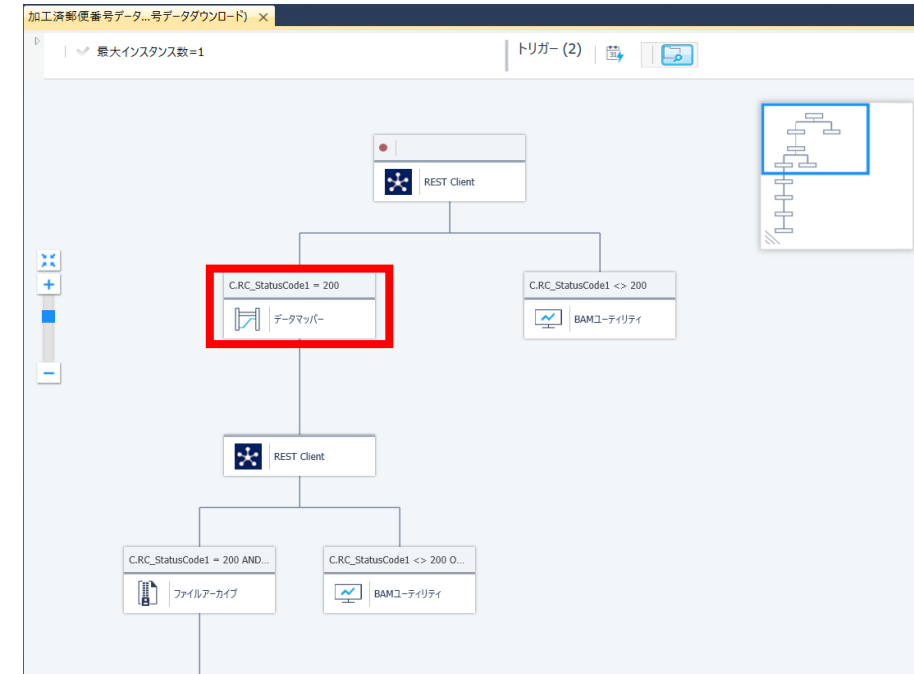
OUTPERFORM THE FUTURE™

1.3ダウンロードファイル名の取得

- ダウンロードするファイル名は、「x-ken-all<年月数値文字>.zip」である。

```
155 <div class="dlEntry">
156 <div class="dlTitle">全国一括データ（加工済バージョン）</div>
157
158
159
160 <div class="dlTable">
161 <table border="1">
162 <thead>
163 <tr>
164 <th>ファイル名</th>
165 <th>公開日</th>
166 <th>サイズ</th>
167 <th>行数</th>
168 <th>MD5</th>
169 </thead>
170 <tbody>
171 <tr>
172 <td class="fnColumn">x-ken-all202306.zip</td>
173 <td>2023年6月30日</td>
174 <td>1.7MB (1,734,573バイト)</td>
175 <td>127,399</td>
176 <td colspan="3">b037ba76e2b7f30cdb88344efc27dd43</td>
177 </tr>
178 </tbody>
179 </table>
180 </div>
181 <div class="dlButton">
182 <a href="/zipcodedata/download?di=1688112638794" rel="nofollow" onclick="javascript:
pageTracker._trackPageview('/zipcodedata/download/x-all/1688112638794');"></a>
```

「x-ken-all」を探して、それを含む19文字分抽出する



OUTPERFORM THE FUTURE™

1.3ダウンロードファイル名の取得(続き)

- データマッパーの送り先「Variables2」

The screenshot illustrates the configuration of a data mapper's destination. On the left, a '送り先' (Destination) list shows 'Variable_2' selected. A red box highlights this selection, with an arrow pointing to the 'Variables2のプロパティ' (Properties of Variables2) window. This window shows the 'Variable_2 変数スキーマ' (Variable_2 Variable Schema) with the 'F.ダウンロードファイル名' (F. Download File Name) property highlighted. Below it, the '送り先特性' (Destination Characteristics) window shows a table of variables, with the 'F.ダウンロードファイル名' variable selected (checked).

Variables2のプロパティ

Variable_2 変数スキーマ

スキーマ情報	値
ID	2
変数名	F.ダウンロードファイル名
説明	Variable_2

送り先特性

変数の選択画面

#	選択	変数名	タイプ
5		F.ダウンロードURL	Alpha
6	✓	F.ダウンロードファイル名	Alpha
7		F.ダウンロードパス名	...



1.3ダウンロードファイル名の取得(続き)

- データマッパーの送り先「Variables2」 – 「Instance」のプロパティの条件欄
条件式は下記の通り

`InStr (Src.S1/Record/Row , 'x-ken-all') > 0`

(Src.S1/Record/Row に「x-ken-all」が含まれる場合に条件が成立する

⇒ 「F.ダウンロードファイル名」が実行される)

The screenshot shows the Magic Software interface. On the left, a tree view shows the 'Instance' property selected under 'Variable_2'. A blue arrow points from this selection to the 'Instanceのプロパティ' (Instance Properties) window on the right. In this window, the 'Condition' field is highlighted with a blue box and contains the formula `InStr (Src.S1/Record/Row , 'x-ken-`. Below this, the 'Condition Formula Editor' (条件の式エディタ画面) is open, showing the full formula `InStr (Src.S1/Record/Row , 'x-ken-all') > 0` in a red box. A blue arrow points from the 'Condition' field to the editor.

※このInstanceのプロパティのマルチアップデートは設定していない。



1.3ダウンロードファイル名の取得(続き)

- 「F.ダウンロードファイル名」のプロパティの計算値欄：
計算式は下記の通り

`MID (Src.S1/Record/Row , InStr (Src.S1/Record/Row , 'x-ken-all') , 19)`

Src.S1/Record/Row の「x-ken-all」を含む開始位置から19文字抽出する

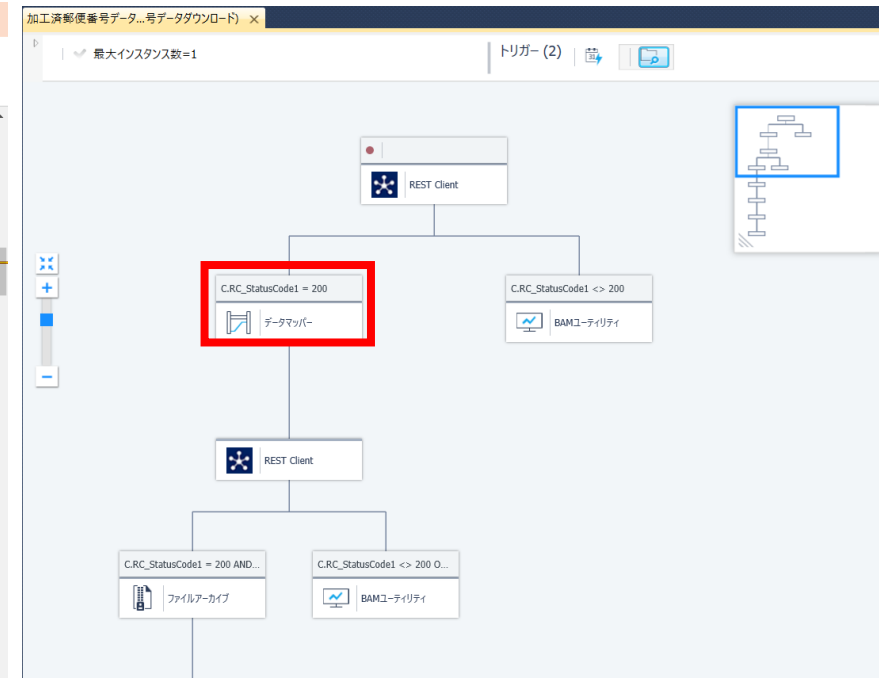
The screenshot shows the software's property editor for 'F.ダウンロードファイル名'. The '計算値' (Calculation) field contains the formula: `MID (Src.S1/Record/Row , InStr (Src.S1/Record/Row , 'x-ken-all') , 19)`. A red box highlights the formula, and a red arrow points from the 'F.ダウンロードファイル名' property in the left pane to the formula field. Below the property editor, a '計算値の式エディタ画面' (Calculation Formula Editor) window is open, showing the same formula in a text field. A red box highlights the formula in the editor, and a red arrow points from the formula field in the property editor to the editor window.



1.4 更新日付の取得

- HTMLのソースの中から「全国一括データ (20)」という文字列のある行を探して取得し、そこから「日付部分(年月日)」を抽出する

```
75 <span class= attention >  
76 ※解凍後のCSVファイルの仕様については、<a href="http://www.post.japanpost.jp/zipcode/dl/readme.html" target="_blank">日  
本郵便のWebサイト</a>をご確認ください。<br />  
77 ※差分データは、1つの圧縮ファイル中に「新規追加データ」と「廃止データ」を含んでいます。<br />  
78 ※公開しているデータは、<strong>「読み仮名の促音・拗音を小書きで表記するもの」</strong>になります。  
79 </span>  
80 </p>  
81 </div class="dlEntry">  
82 <div class="dlTitle">全国一括データ (2023年6月30日更新分)</div>  
83 </div class="dlDetail">  
84 <table cellpadding="0">  
85 <tr>  
86 <th>ファイル名</th>  
87 <td class="fnColumn">ken-all1202306.zip</td>  
88 <th>公開日</th>  
89 <td>2023年6月30日</td>  
90 </tr>  
91 <tr>  
92 <th>サイズ</th>  
93 <td>1.7MB (1,757,558バイト) </td>  
94 <th>行数</th>  
95 <td>124,578</td>  
96 </tr>  
97 <tr>  
98 <th>MD5</th>  
99 <td colspan="3">c2cadad2b5fa02a7a1277c61bc906087</td>  
100 </tr>  
101 </table>  
102 </div class="dlDetail">  
103 </div class="dlDetail">
```



OUTPERFORM THE FUTURE™

1.4 更新日付の取得(続き)

- zipcloudのホームページのHTMLにおいて、「全国一括データ(20)」を含む行は下記の行だけである

<div class="dlTitle">全国一括データ(2023年6月30日更新分)

</div>

20xx年xx月xx日の部分を抽出するには？

StrToken関数を活用する

StrToken (Src.S1/Record/Row , 2 , '(') ⇒ F.更新日抽出1



StrToken (F.更新日抽出1 , 1 , '更新')



F.更新日抽出2 ← 2023年6月30日

F,更新日抽出1およびF.更新日抽出2の計算処理(マルチアップデート)には、条件が設定されていることに留意



1.4 更新日付の取得(続き)

- データマッパーの送り先「Variables4」の「Instance」のプロパティのマルチアップデート

The screenshot shows the Magic software interface. On the left, a tree view shows 'Variable_4' with an 'Instance' property. A red box highlights 'Instance' and an arrow points to the 'プロパティ' (Properties) window. The 'プロパティ' window shows 'Instance Complex項目' with a 'マルチアップデート' (Multi-update) property set to '(コレクション)'. A red box highlights this property and an arrow points to the 'マルチアップデートの画面' (Multi-update screen). The 'マルチアップデートの画面' is a table with columns '#', '変数名' (Variable Name), '式' (Expression), and '条件' (Condition). Two rows are highlighted with red boxes: row 1 with 'F.更新日抽出 1' and row 2 with 'F.更新日抽出 2'. Arrows from these rows point to their respective formula editors. The formula editor for 'F.更新日抽出 1' shows the formula: `StrToken (Src.S1/Record/Row , 1 , '更新')`. The formula editor for 'F.更新日抽出 2' shows the formula: `StrToken (Src.S1/Record/Row , 2 , ' (')`. The '条件' (Condition) column contains the formula: `InStr (Src.S1/Record/Row , '全国一括データ (20') > 0`. A red box highlights this formula and an arrow points to the 'マルチアップデートの条件欄の式はすべて同じ' (Multi-update condition column formulas are all the same) text.

「Instsnce」のプロパティ

プロパティ

Instance Complex項目

その他

一般

条件

マルチアップデート

InStr (Src.S1/Record/Row , '全国一括データ (20') > 0 (コレクション)

マルチアップデートの画面

#	変数名	式	条件
1	F.更新日抽出 1	StrToken (Src.S1/Record/Row , 1 , '更新')	InStr (Src.S1/Record/Row , '全国一括データ (20') > 0
2	F.更新日抽出 2	StrToken (F.更新日抽出 1 , 1 , ' (')	InStr (Src.S1/Record/Row , '全国一括データ (20') > 0
3	F.更新日_月日部分抽出	StrToken (F.更新日抽出 2 , 1 , ' (')	InStr (Src.S1/Record/Row , '全国一括データ (20') > 0
4	F.月部分数値文字列抽出	StrToken (F.更新日_月日部分抽出 , 1 , ' (')	InStr (Src.S1/Record/Row , '全国一括データ (20') > 0
5	F.月部分数値文字列抽出	IF (Len (Trim (F.月部分数値文字列抽出) > 0 , ' (')	InStr (Src.S1/Record/Row , '全国一括データ (20') > 0
6	F.日部分数値文字列抽出	Left (StrToken (F.更新日_月日部分抽出 , 1 , ' (')	InStr (Src.S1/Record/Row , '全国一括データ (20') > 0

F.更新日抽出 1 の式欄の式エディタ画面

Src.S4/Instance : Instance - 式 [Alpha]

StrToken (Src.S1/Record/Row , 1 , '更新')

F.更新日抽出 1 の式欄の式エディタ画面

Src.S4/Instance : Instance - 式 [Alpha]

StrToken (Src.S1/Record/Row , 2 , ' (')

マルチアップデートの条件欄の式はすべて同じ

Src.S4/Instance : Instance - 条件 [Logical]

InStr (Src.S1/Record/Row , '全国一括データ (20') > 0



OUTPERFORM THE FUTURE™

1.4 更新日付の取得(続き)

- データマッパーの送り先「Variables4」の「Instance」のプロパティのマルチアップデート(続き)
その他、更新日の月日部分や月部分の数値文字列、日部分の数値文字列の抽出などを行っている。

#	変数名	式	条件
1	F.更新日抽出 1	StrToken (Src.S1/Reco	InStr (Src.S1/Record/F
2	F.更新日抽出 2	StrToken (F.更新日抽出 1, 1	InStr (Src.S1/Record/Row, '
3	F.更新日 月日部分抽出	StrToken (F.更新日抽出 2	InStr (Src.S1/Record/Row, '
4	F.月部分数値文字列抽出	StrToken (F.更新日_月日部	InStr (Src.S1/Record/Row, '
5	F.月部分数値文字列抽出	IF (Len (Trim (F.月部分数値	InStr (Src.S1/Record/Row, '
6	F.日部分数値文字列抽出	Left (StrToken (F.更新日 月	InStr (Src.S1/Record/Row, '

マルチアップデートの条件欄の式はすべて同じ

```
InStr (Src.S1/Record/Row, '全国一括データ (20') > 0
```

F.更新日抽出 1 の式欄の式エディタ画面

```
StrToken (F.更新日抽出 2, 2, '年')
```

F.月部分数値文字列抽出の式欄の式エディタ画面

```
StrToken (F.更新日_月日部分抽出, 1, '月')
```

F.月部分数値文字列抽出 (2つ目) の式欄の式エディタ画面

```
IF (Len (Trim (F.月部分数値文字列抽出))=1, '0' & Trim (F.月部分数値文字列抽出), F.月部分数値文字列抽出)
```

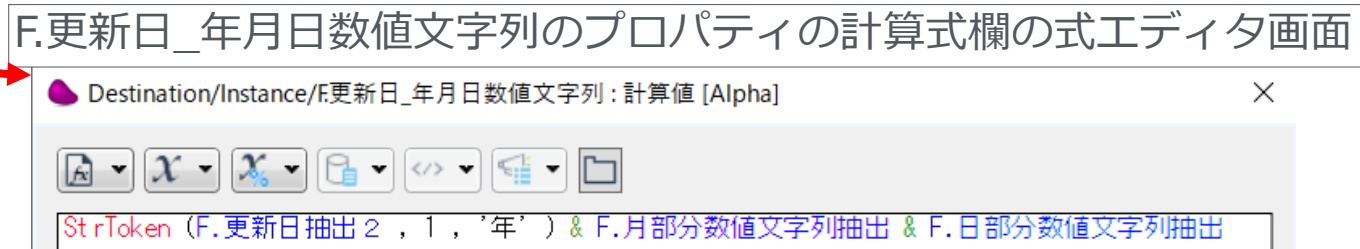
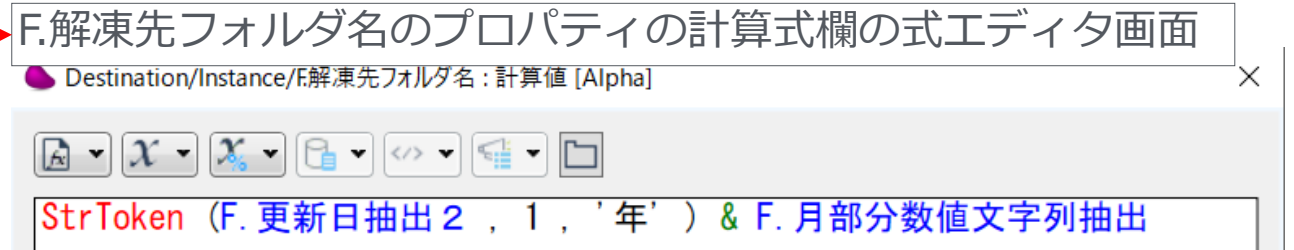
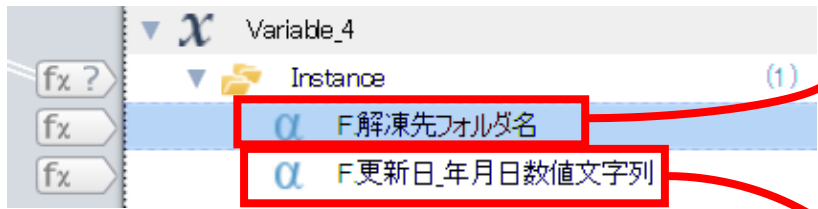
F.日部分数値文字列抽出の式欄の式エディタ画面

```
Left (StrToken (F.更新日_月日部分抽出, 2, '月'), 2)
```



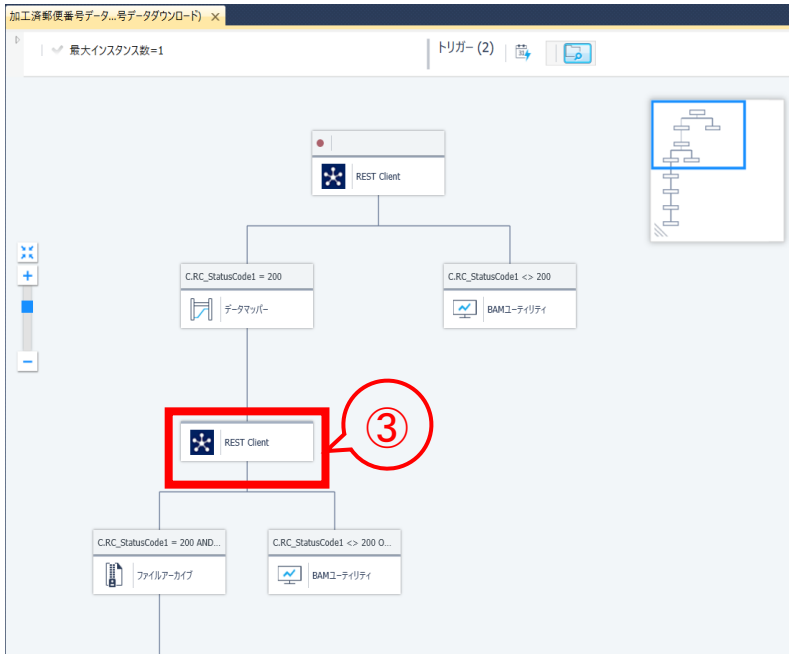
1.4 更新日付の取得(続き)

- 「Variables4」には、「F.解凍先フォルダ名」「F.更新日_月日数値文字列」「F.更新日_年月日数値文字列」を選択していて、これらの変数のプロパティの計算値に計算式を設定している



1.5 REST Client : 郵便番号データ(加工済バージョン)zipファイルのダウンロード

- REST Client zipcloud用リソースのパス②を利用する



③REST Clientステップの設定画面

REST Client 構成

接続
リソース名: REST Client_zipcloud

オペレーション
パス: **/zipcodedata/download** (A)

オペレーション: **Get** (B)

セキュリティ: No

リクエスト Content Type: [] ...

レスポンス Content Type: [] ...

要求エンコーディング: **Binary** (C) Charset: []

出力
結果保存: **File** (D) EnvVal ('currentprojectdir') & []

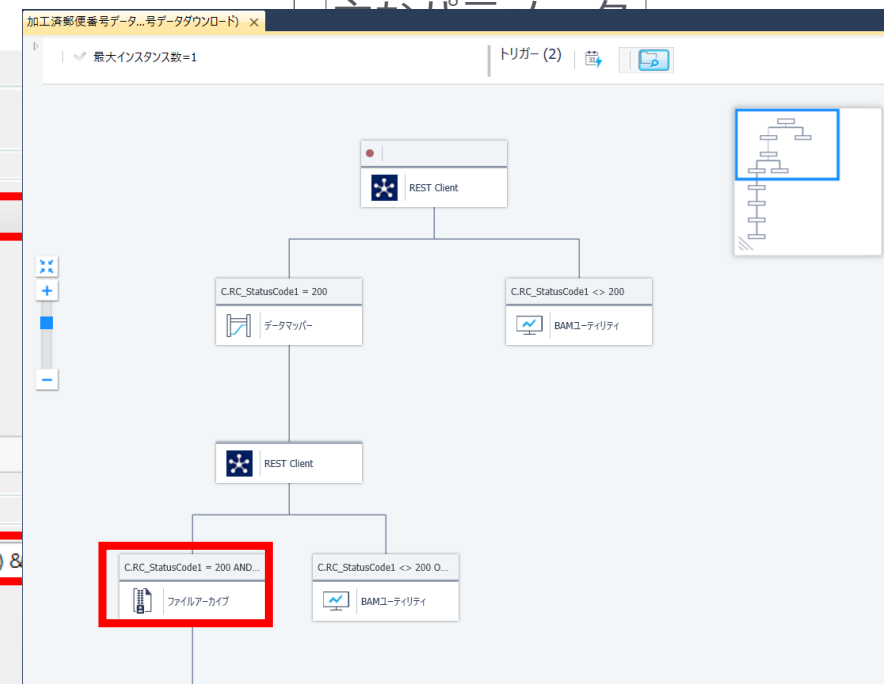
ヘッダ: []

ヘッダ: [] C.RC_HeaderBlob1

ステータスコード: [] C.RC_StatusCode1

メッセージ: [] C.RC_Message1

成功: [] Variable C.RC_IsSucc...



REST Client : Configuration - 結果保存 [Alpha]

EnvVal ('currentprojectdir') & 'Zipcloud_DL%' & F.ダウンロードファイル名

[パラメータ]をクリック

次ページに続く

パラメータ

スキーマ更新

OK

キャンセル



OUTPERFORM THE FUTURE™

1.5 REST Client : 郵便番号データ(加工済バージョン)zipファイルのダウンロード

- REST Client ステップの設定画面(続き)
パラメータ画面

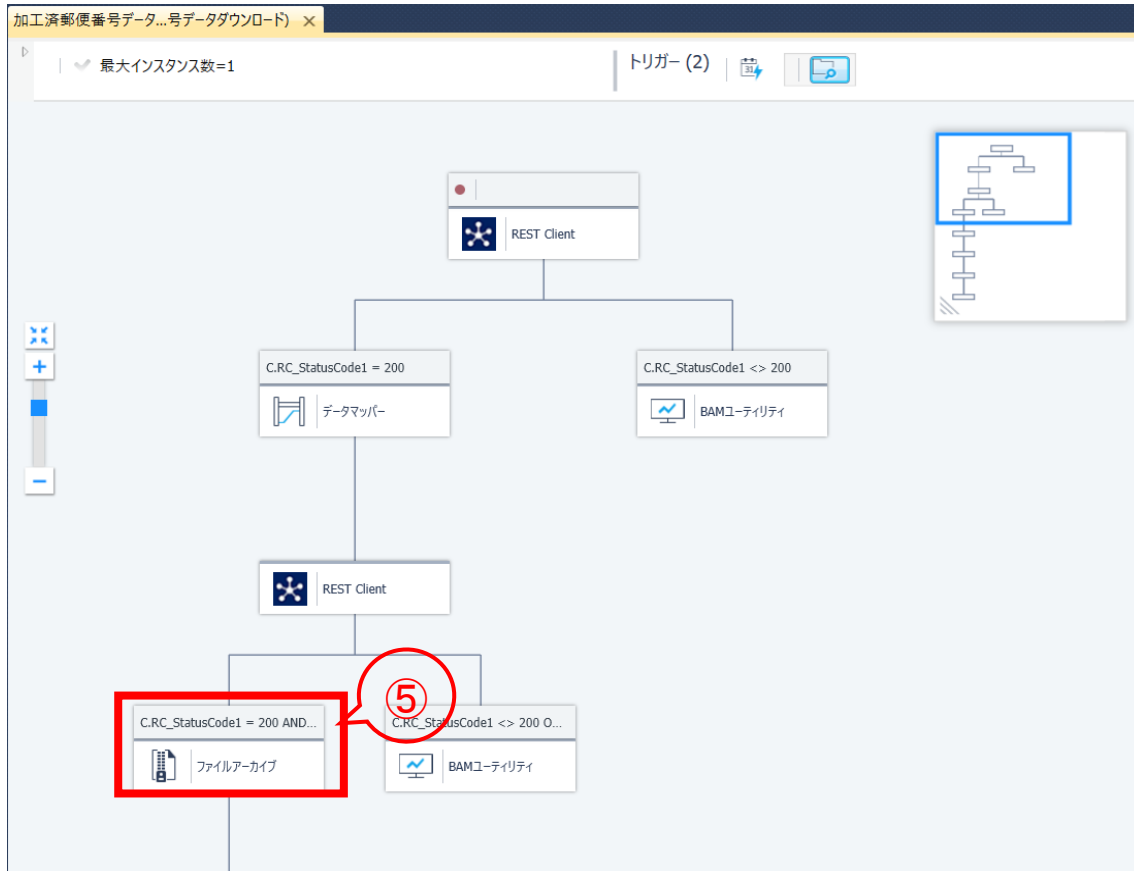
#	パラメータ	値
1	di	Right (F.ダウンロードURL, 13)

diパラメータの値欄に、F.ダウンロードURLの数値の部分抽出した数値文字列を設定



1.6 zipファイルの解凍

- 1.5 でダウンロードに成功したzipファイルを解凍する
ファイルアーカイブアダプタを利用する



⑤ファイルアーカイブアダプタステップ
「Extract」メソッドを使用して、zipファイルを解凍する



1.6 zipファイルの解凍(続き)

- ファイルアーカイバアダプタステップのダイレクトアクセスメソッドの画面

#	名	条件
1	Extract	

パラメータ名	タイプ	数	出力	値
抽出送り元ファイル	Alpha	260	In	EnvVal('currentprojectdir') & 'Zipcloud_DL'
抽出場所	Alpha	260	In	EnvVal('currentprojectdir') & 'Zipcloud_DL' &
暗号化パスワード	Alpha			
エラーコード	Numeric		Out	

主な設定・パラメータ

- ① : 「Extract」メソッドを選択
- パラメータ :
- ② : 解凍元のzipファイルのパスを設定
- ③ : 解凍先のフォルダパスを設定

②の値の式エディタ画面

式の内容 : EnvVal('currentprojectdir') & 'Zipcloud_DL' & F.ダウンロードファイル名

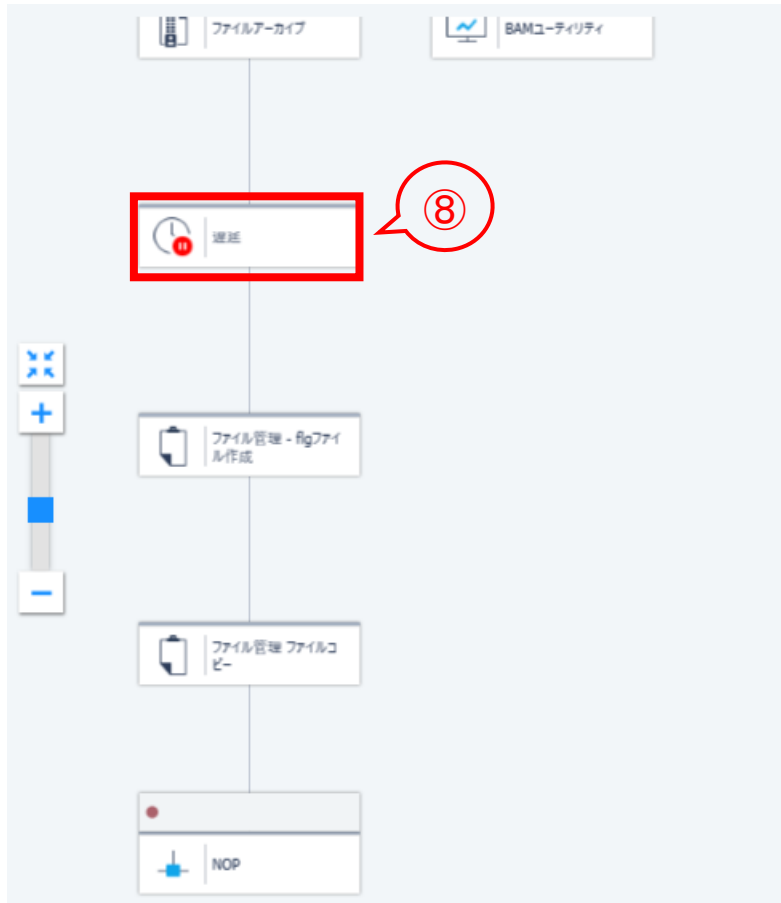
③の値の式エディタ画面

式の内容 : EnvVal('currentprojectdir') & 'Zipcloud_DL' & F.解凍先フォルダ名



1.7 遅延(一時待機)

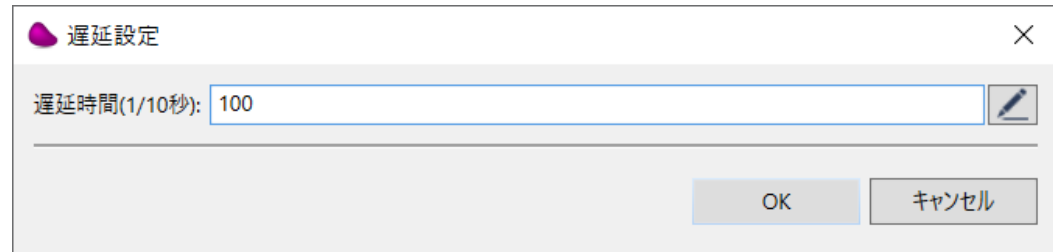
- zipファイルの解凍に時間を要した場合を想定して、一時待機するために、遅延アダプタを配置している



⑧遅延アダプタステップ

指定する遅延時間の単位は、10分の1秒
100と指定すると、その10分の1の10秒待機される

遅延時間の設定画面



1.7 flgファイルの作成

- zipファイルの解凍が完了した後、CSVデータ取り込み処理のフローを実行したい
CSVデータ取り込み処理のフローは2パターン用意しており、Directory Scannerトリガーを配置しているため、監視フォルダに監視対象ファイル(フィルタ条件に合致するファイル)を検出すれば、フローが開始されるようになっている。
フローの監視フォルダは異なるが、フィルタ条件の設定値は2パターンのフロー共に同じx-ken-all*.flgである
この拡張子flgファイルを作成する
ファイルの中身はテキストで、郵便番号データの更新日付を保存する
作成したflgファイルは、先にCSVデータファイルを移動してから、移動する
これはファイルの移動の順序に意図があるため(詳細は次章のトリガーの説明を参考)



1.7 flgファイルの作成(続き)

- ファイル管理アダプタのCreate Fileメソッドを使用する

パラメータ名	タイプ	書式	入出力	値
対象ファイル名	Alpha	240	In	EnvVal ('currentprojectdir') & 'Zipcloud_DL¥' & F. 解凍先フォルダ名 & '¥x-ken-all.flg'
エンコーディング	Numeric	1	In	Ansi
データ	Blob		In	F.更新日_年月日数値文字列
エラーコード	Numeric	12	Out	

EnvVal ('currentprojectdir') & 'Zipcloud_DL¥' & F. 解凍先フォルダ名 & '¥x-ken-all.flg'

①対象ファイル：①の欄の式は、式エディタ画面を参照

EnvVal ('currentprojectdir') & 'Zipcloud_DL¥' & F. 解凍先フォルダ名 & '¥x-ken-all.flg'

②エンコーディング：Ansi

③データ：F.更新日_年月日数値文字列



1.8 ファイルコピー（csvファイルとflgファイル）

- フロー「CSVデータ取り込み（パターン1・パターン2）」のDirectory Scannerトリガーによる監視フォルダに、フィルタ条件に合致するファイルをコピーすることで、データ取り込み処理の開始させることを目的とする
- zipファイルを解凍して展開されたx-ken-all.csvファイルと、作成したflgファイルを子の順序で、CSVデータ取り込み処理を行うフローの監視フォルダにコピーする
- CSVデータ取り込み処理のフローは2パターン用意しており、それぞれ監視フォルダのパスが異なるので、移動先のパスを指定することにより、どちらのパターンのフローでCSVデータ取り込み処理を行うのか、切り替えることができるようにしている
- 監視フォルダパスについての詳細は、次章の説明（2.1.2 / 2.2.2）をご覧ください



OUTPERFORM THE FUTURE™

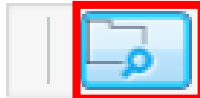
1.9 トリガーについて

- フロー「加工済郵便番号データzipのダウンロード」のトリガーには、スケジュールユーティリティトリガーとDirectory Scannerトリガーの2つのトリガーを配置した
- 基本のトリガーはスケジュールユーティリティトリガーで、毎月1日の3時にダウンロードする
- スケジュールユーティリティトリガーがうまく機能しなかったり、ダウンロードに失敗したた場合の対策として、手動でフローを開始する仕組みを設けるため、Directory Scannerトリガーを配置した
⇒ Directory Scannerトリガーの監視フォルダに検知対象のファイルを配置することで、フローを開始させることができる



1.9 トリガーについて(続き)

トリガー (2)



スケジュールユーティリティ

#	名称
1	Scheduler1

スケジュールサービス選択: Scheduler_1

スケジュールタイプ: 月次

休日用カレンダー: None

イベント時間を定義

<input checked="" type="checkbox"/> 1月	<input checked="" type="checkbox"/> 4月	<input checked="" type="checkbox"/> 7月	<input checked="" type="checkbox"/> 10月
<input checked="" type="checkbox"/> 2月	<input checked="" type="checkbox"/> 5月	<input checked="" type="checkbox"/> 8月	<input checked="" type="checkbox"/> 11月
<input checked="" type="checkbox"/> 3月	<input checked="" type="checkbox"/> 6月	<input checked="" type="checkbox"/> 9月	<input checked="" type="checkbox"/> 12月

日付を選択: ... 1

実行時刻: ... 03:00

コンポーネント設定: ディレクトリスキャナ

#	送り元	ディレクトリ	サ.	フィルタ	アクション	送り先ディレクトリ
1	LAN	EnvVal ('currentprojectdir') & 'DL_DSTTrigger'	<input type="checkbox"/>	DL_Trigger*.txt	Move	EnvVal ('currentprojectdir') & 'DL_DSTTrigger\done'



OUTPERFORM THE FUTURE™

第2章 CSVデータ取り込み



OUTPERFORM THE FUTURE™

2.1 フロー「CSVデータ取り込みパターン1」

- 郵便番号データCSVファイルを元に、郵便番号マスタテーブルにデータを挿入する処理のフロー
- 郵便番号データCSVファイルのデータ件数はとても多い(12万件超)
- CSVファイル(12万件超)のファイルをそのまま使用してデータ挿入処理を行うフローについて解説する

前提:

- DBMS: Microsoft SQL Serverを利用している
- CreateTable文のファイルは用意済
- 先に[zipcloud]データベースを作成しておく必要がある
Microsoft SQL Management Studioを使用して、
[zipcloud]データベースを作成する(補足資料b)参照)



OUTPERFORM THE FUTURE™

2.1.1 フロー「CSVデータ取り込みパターン1」リソース

■ 設定画面のリソース「Database_zipcloud」

#	名	タイプ	フォーマット	値
1	DBMS	文字	30	Microsoft SQL Server
2	データベース名	文字	30	%DB_zipcloud%
3	サーバ	文字	30	%SV_zipcloud%
4	ユーザ	文字	100	%User_zipcloud%
5	パスワード	文字	100	*****
6	開始オーナー	文字	30	*****

① : DBMSの種類をコンボボックスから選択

このプロジェクトではデータベース「zipcloud」のDBMSである「Microsoft SQL Server」を選択

② : データベース名を設定

環境変数を使用する場合は、値欄の右端の[...]ボタン (A) をクリックして表示される環境変数のリストから選択すると、環境変数名の前後に「%」で括られた文字列が値欄にセットされる

③ : データベースサーバ名を設定

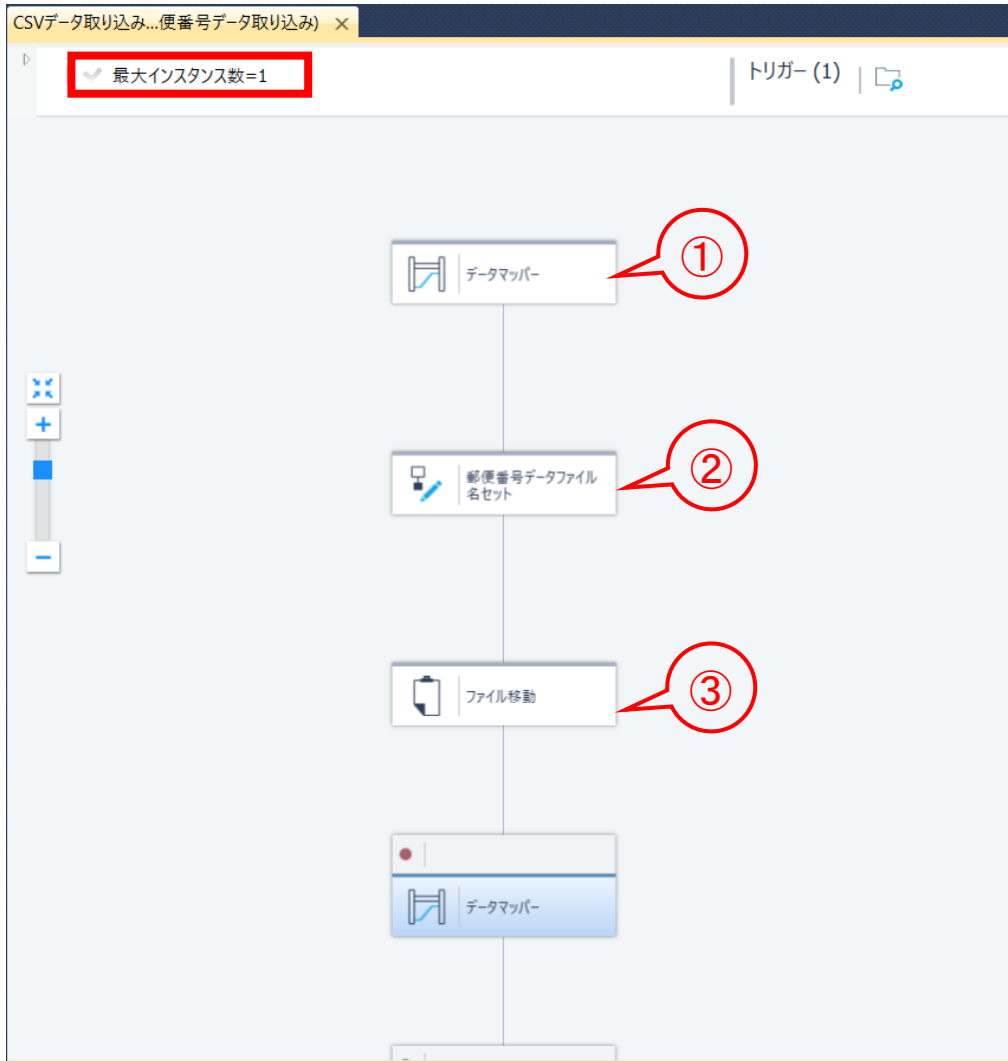
④ : データベースに接続するユーザを設定

⑤ : データベースに接続するユーザのパスワードを設定 (※補足c)に注意点の説明あり)



2.1.1 フロー「CSVデータ取り込みパターン1」フロー図

・ フロー図



トリガー：Directory Scanner（詳細は以降のページにて解説）

ステップ①：「データマッパー」アダプタ
Directory Scannerトリガーのフィルタ条件に設定しているflgファイルに含まれる郵便番号データの更新日を取得する

ステップ②：「Flow Data」アダプタ
Flgファイルを検出して取得した送り元/送り先ファイルパスの拡張子「flg」を「csv」に置換して、郵便番号データCSVファイルパス（移動元/移動先）を変数にセットする

ステップ③：「データマッパー」アダプタ
郵便番号CSVデータファイル⇒郵便番号マスタテーブルにデータを取り込む処理

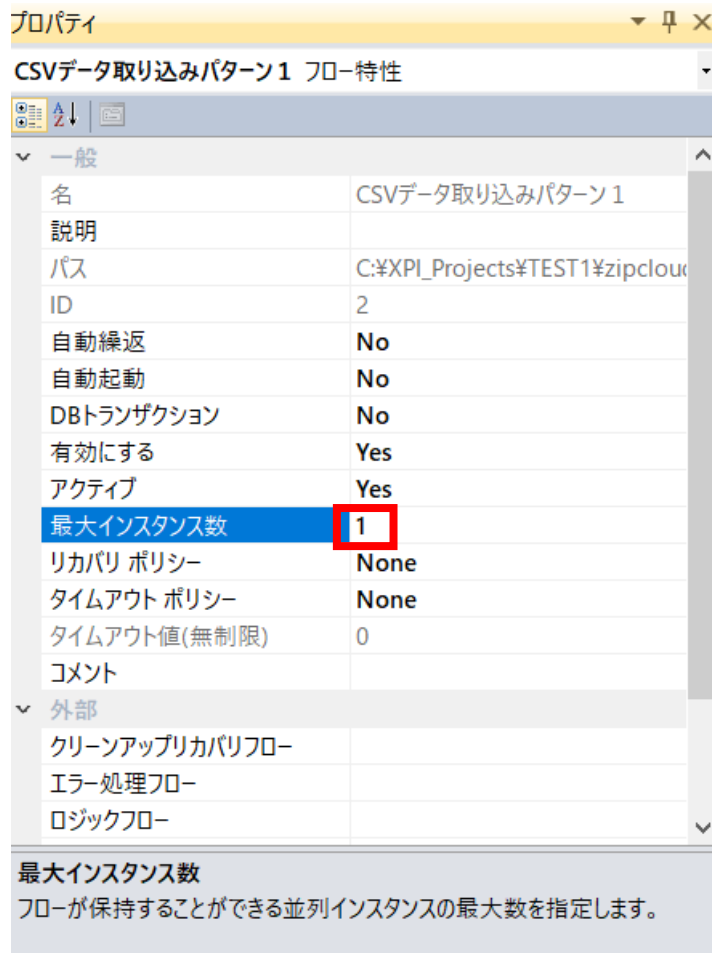
次ページでフローのプロパティの解説をする



OUTPERFORM THE FUTURE™

2.1.1 フロー「CSVデータ取り込みパターン1」フロー(続き)

・ フローのプロパティ



The screenshot shows a 'Properties' dialog box for a flow named 'CSVデータ取り込みパターン1'. The 'General' tab is selected, and the 'Maximum Instance Count' (最大インスタンス数) is set to 1, which is highlighted with a red box. Other properties include Name, Description, Path, ID, and various execution options.

プロパティ	
CSVデータ取り込みパターン1 フロー-特性	
一般	
名	CSVデータ取り込みパターン1
説明	
パス	C:\XPI_Projects\TEST1\zipcloud
ID	2
自動繰返	No
自動起動	No
DBトランザクション	No
有効にする	Yes
アクティブ	Yes
最大インスタンス数	1
リカバリ ポリシー	None
タイムアウト ポリシー	None
タイムアウト値(無制限)	0
コメント	
外部	
クリーンアップリカバリフロー	
エラー処理フロー	
ロジックフロー	

最大インスタンス数
フローが保持することができる並列インスタンスの最大数を指定します。

最大インスタンス数：1

最大インスタンス数は、同時並行実行可能なインスタンス数（スレッド数）を指定することが出来るパラメータ

これを「1」に設定することで、同時並行実行可能なスレッド数の上限が1となり、複数並行実行出来ないよう、制限を掛けることが出来る。

つまり、「1」を設定することで、このフローは一つしか実行出来なくなる。トリガー条件を満たす状態が同時並行で発生しても、1つだけが実行され、その処理が終了またはアボートした後実行される



2.1.2 フロー「CSVデータ取り込みパターン1」のトリガー

- トリガーについて (Directory Scanner)

監視フォルダパス: EnvVal ('currentprojectdir') & 'Input_YubinCSV¥Pattern1'

移動先フォルダパス: EnvVal ('currentprojectdir') & 'Input_YubinCSV¥Pattern1¥done'

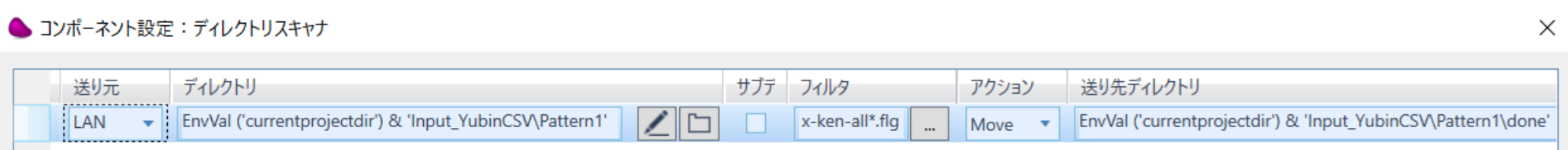
フィルタ: x-ken-all*.flg

※補足

x-ken-all.csvファイルの件数が多くファイルサイズが大きいことから、ファイルの移動に時間を要することを考慮して、フィルタでは拡張子「flg」ファイル（ファイルサイズが小さい数バイト）を検出するようにして工夫している

x-ken-all*.csvファイルの監視フォルダへのコピーあるいは移動が完了した後に、フィルタ条件のx-ken-all*.flgファイル（サイズが小さいファイル）を移動させることで、サイズの大きいファイルのコピーあるいは移動の途中でファイルが検知されてしまうことにより「Move」アクションがエラーとなる（ファイルロックにより移動が失敗する）ことを防止するための策である。

これはファイルの処理順序と、サイズが小さいファイルのコピーあるいは移動には時間が掛からないので、サイズが小さいファイルを検出するよう、フィルタ条件を指定している工夫がポイントとなる。



OUTPERFORM THE FUTURE™

2.1.2 フロー「CSVデータ取り込みパターン1」のトリガー(続き)

- Directory Scannerトリガー [ログと戻値詳細]ボタンをクリックして表示される設定画面
設定内容は下記の通り

ログと戻値詳細

ログ詳細

ログアクション

タイプ: テキスト

Log file path: %log%zipcloudProject1_Input_YubinCSV_Folder_triggerlog.txt

戻値詳細

取得ファイル内容保存先: ...

送り先ファイル名保存先: ... F.郵便番号データflgパス_送り先

送り元ファイル名保存先: ... F.郵便番号データflgパス_送り元

引数XMLの場所を選択: ...

General

FTP転送モード: Binary

OK 取消

- ① : 「ログアクション」にチェックする
- ② : タイプ「テキスト」
ログファイルパスを指定する
他のプロジェクトやフローと異なるよう、ユニークな名前にすることを推奨する
- ③ : Directory Scannerトリガーで検知したファイルパスを保持するための変数を選択する
- ④ : Directory Scannerトリガーで検知し、移動する先のファイルパスを保持するための変数を選択する



2.1.3 「郵便番号データファイル名セット」ステップ

- ①「郵便番号データファイル名セット」ステップは、Flow Dataアダプタステップ

このステップでは、Directory Scannerトリガーで検出したファイルパスおよび、移動先のファイルパスの拡張子「flg」を「csv」に置換している

#	アクション	タイプ	動的	名	データタイプ	Encoding	1.	更新式	条件
1	Update	Flow		F.郵便番号データCSVファイルパス_送り元	Alpha	Ansi		RepStr (F.郵便番号データflgパス_送り元, '.flg', '.csv')	
2	Update	Flow		F.郵便番号データCSVファイルパス_送り先	Alpha	Ansi		RepStr (F.郵便番号データflgパス_送り先, '.flg', '.csv')	

- ① : 送り元 (監視フォルダ) の検出ファイルパスの拡張子「flg」を「csv」に置換処理
② : 送り先 (移動先) ファイルパスの拡張子「flg」を「csv」に置換処理



2.1.4「ファイル移動」ステップ

- ②「ファイル移動」ステップは、ファイル管理アダプタのステップ
郵便番号データCSVファイルを移動する

①のステップ「郵便番号データファイル名セット」で郵便番号CSVデータファイルのパス(移動前・移動先)をセットした変数を利用してファイルを移動する

ファイル管理アダプタでは、ファイルの移動は「Rename File」メソッドを利用することで実現できる



ダイレクトアクセスメソッド: ファイル管理

#	名	条件
1	Rename File	

パラメータ名	タイプ	書式	入/出力	値
ファイル名From	Alpha	260	In	F.郵便番号データCSVファイルパス_送り元
ファイル名To	Alpha	260	In	F.郵便番号データCSVファイルパス_送り先
エラーコード	Numeric	12	Out	



メソッド、パラメータの設定詳細については次ページで解説

2.1.4 「ファイル移動」ステップ(続き)

- ②「ファイル移動」ステップはファイル管理アダプタを利用している
設定画面

ダイレクトアクセスメソッド: ファイル管理

#	名	条件
1	Rename File.....	

パラメータ名	タイプ	書式	入出力	値
ファイル名From	Alpha	260	In	F.郵便番号データCSVファイルパス_送り元
ファイル名To	Alpha	260	In	F.郵便番号データCSVファイルパス_送り先
エラーコード	Numeric	12	Out	

① : 「Rename File」メソッドを選択

Ⓐ : ファイル元のパスを指定 (変数「F.有印番号データCSVファイルパス_送り元」を設定)

Ⓑ : 移動先のパスを指定 (変数「F.有印番号データCSVファイルパス_送り先」を設定)



2.1.5 「データマッパー」ステップ

- ③ 「データマッパー」ステップにて、CSV→郵便番号マスタテーブルへの挿入処理

プロパティ	
データマッパー ステップ特性	
▼ General	
ステップ名	データマッパー
説明	郵便番号CSVデータ⇒郵便番号マスタ
ステップID	7
コンポーネントID	11
コンポーネントタイプ	データマッパー
条件	
処理モード	リニア
完了を待つ	No
▼ アドバンスド	
セーブポイント	No
コールロジックフロー	No
▼ マッパー	
エラー動作	Exit
DBトランザクション	zipcloud
DBトランザクションレベル	Mapper

処理モード：リニア

エラー動作：Exit

DBトランザクション：zipcloud

郵便番号マスタテーブルのデータベースリソース名

DBトランザクションレベル：Mapper

マッパー全体がひとつのトランザクションとなる

エラー動作で「Exit」が選択されている場合は、データベースでエラーとなると、それまでの挿入処理がすべてロールバックされ、ステップが終了する

「データマッパー
(郵便番号CSVデータ⇒郵便番号マスタ取込) ステップ」
のプロパティ

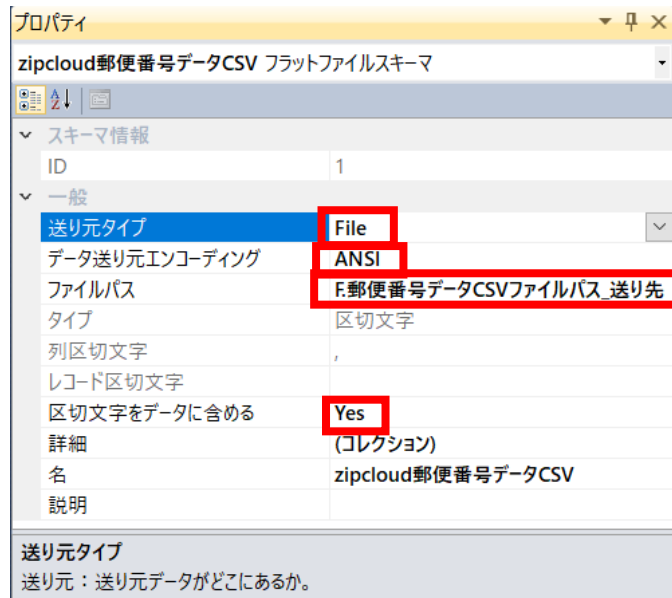


OUTPERFORM THE FUTURE™

2.1.5 「データマッパー」ステップ(続き)

- ③ 「データマッパー」ステップにて、CSV→郵便番号マスタテーブルへの挿入処理
「データマッパー」ステップの設定：

送り先：FlatFileのプロパティ



送り元タイプ：File

データ送り元エンコーディング：ANSI (SJIS)

ファイルパス：変数 (F.郵便番号データCSVファイルパス_送り先) ※1

区切り文字をデータに含める：Yes (※2)

詳細：(コレクション)

(※2：フィールド定義の設定の詳細は次ページで解説)

※1

②「ファイル移動」ステップにて移動した先のファイルパスを指定するため、「F.郵便番号データCSVファイルパス_送り先」変数で正しい

※2

郵便番号CSVデータのデータの一部に、データの前後に" (ダブルクォーテーション) 記号が含まれている。

Yesを選択することで、区切り文字がデータ内に含まれる場合の対応を行ってくれる上、取込時に"記号を取り除く処理を自動的に行ってくれる。今回は特に後者の処理("記号を取り除く処理)のためにYesを選択した

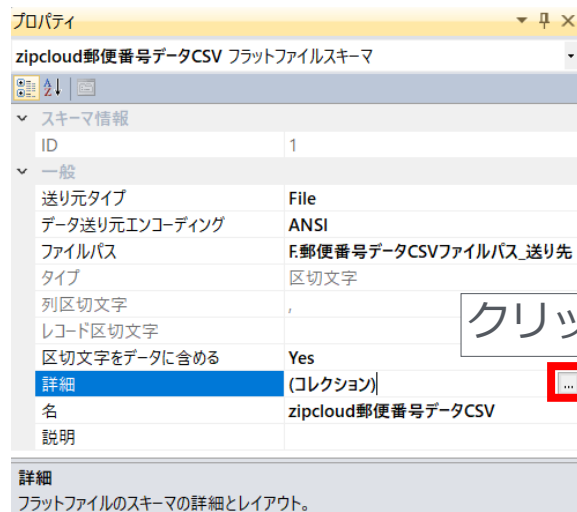


2.1.5 「データマッパー」ステップ(続き)

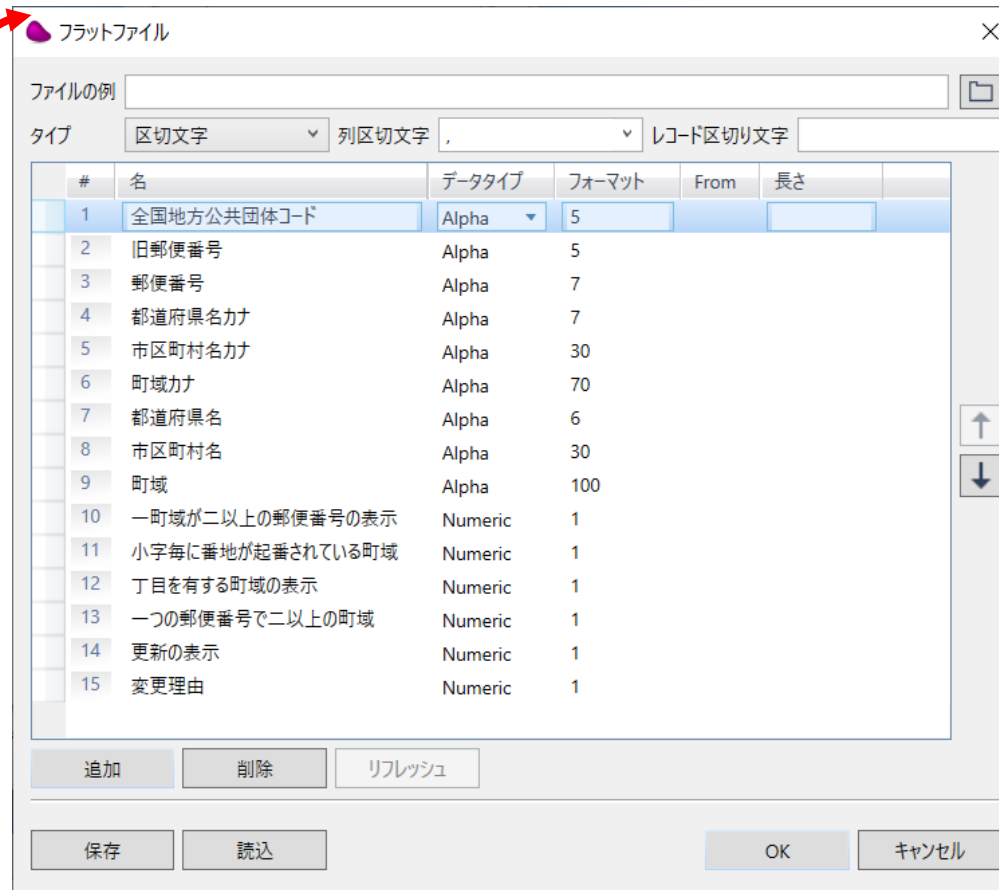
- ③ 「データマッパー」ステップにて、CSV→郵便番号マスタテーブルへの挿入処理

「データマッパー」ステップの設定：

送り元：FlatFileのプロパティ



クリック



「名」、「データタイプ」、「フォーマット」を設定する

データ送り元エンコーディングが「ANSI」を選択しているため、「フォーマット」の値はバイト数の扱いとなる

ちなみに、データ送り元エンコーディングが「Unicode」や「UTF-8」の場合は文字数の扱いとなる

フィールドの名称「名」欄では、全角数字、全角英字、全角記号、半角カナを使用しないこと

動作不良に至ることが確認されている

新規でフィールド名を設定する場合のTipsについて、補足に説明がある



OUTPERFORM THE FUTURE™

2.1.6 「NOP」アダプタステップ (Tips)

- フローの最後の「NOP」アダプタステップは、デバッグ時のブレイクポイントを設定するために設けている

デバッグ実行時に、フローを最後まで実行させてしまうと、もう一度フローを実行し直ための操作が必要となるので、最後のステップに「NOP」を配置してブレイクポイントを設定しておくことで、そこでブレイクされる(一時停止される)ので、コンテキストビューアを表示して、各種変数の現在値を確認することが出来るようになる、ちょっとしたTips(便利な使い方)である

「NOP」アダプタステップは「No OPeration」の略。このステップ自体は何も処理をしない
なお、このステップ実行後、C.Sys.ErrorCodeは「0」がセットされる(常に正常終了)

「NOP」アダプタはフローの先頭で分岐したい場合にも活用することが出来る(分岐のルートステップ)

ちなみにブレイクポイントはデバッグ実行時に機能し、そのステップの(実行)条件を満たすと実行する前に一時停止される



2.2フロー「CSVデータ取り込みパターン2」

- 2.1のフロー「CSVデータ取り込みパターン1」は、件数が多いCSVデータファイルひとつで取込処理を実行するフローでした

この場合、取込処理に時間を要するのは自明である

ここでは、CSVデータファイルを4つに分割してデータ取り込み処理を4つ同時並行実行させることで、取込処理全体に要する時間軽減が期待できる

ひとつのファイルで取込処理を行った場合
の処理時間のイメージ

4つのファイルに分割して4つ同時並行実行した場合
の処理時間のイメージ

処理
時間

処理
時間

処理
時間

処理
時間

処理
時間

同時並行実行時はスレッドライ
センスも実行した数分消費され
る

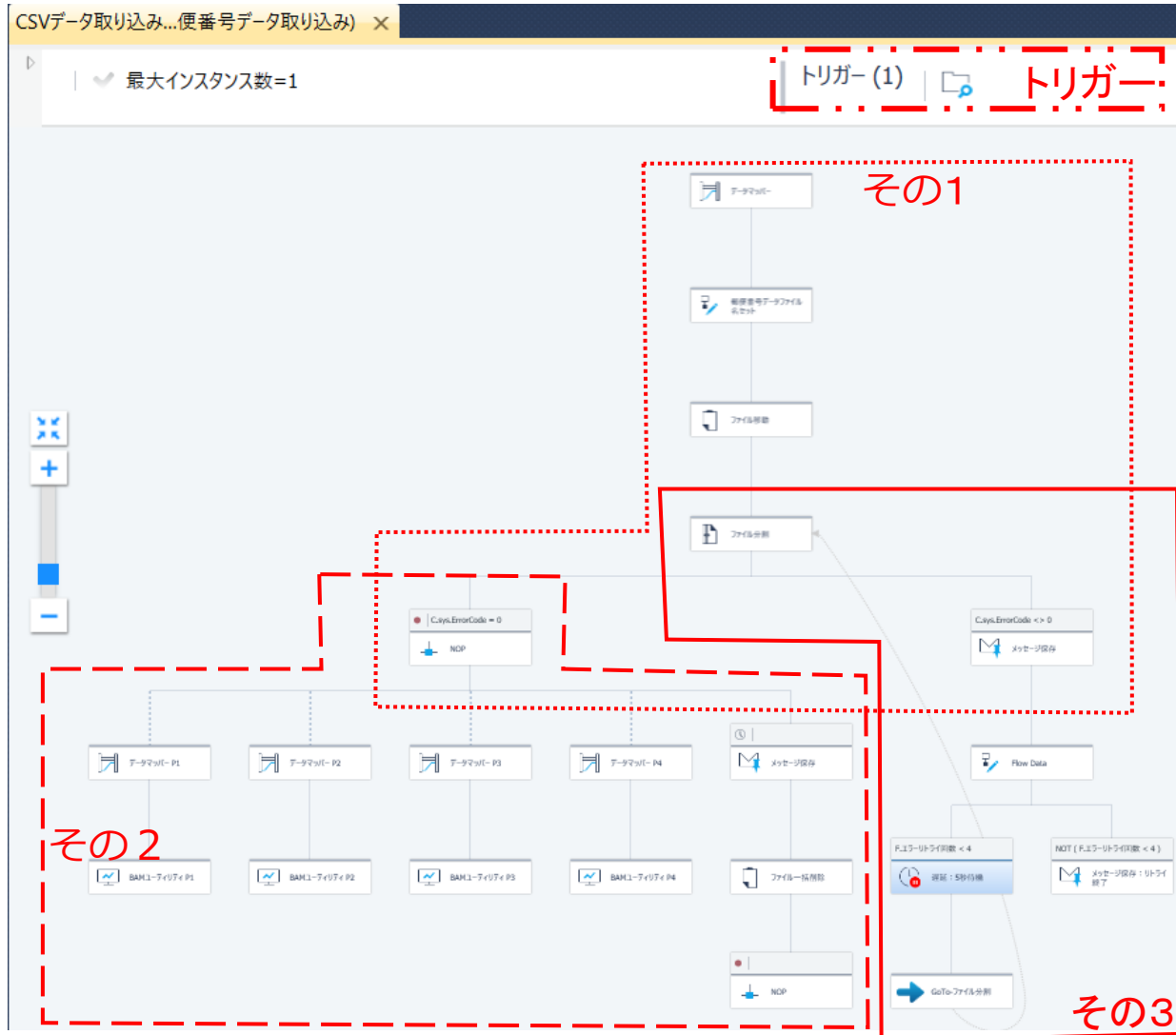
※同時並行実行の動作確認は、デバッグ実行ではなくサーバ実行で行うようにしてください
デバッガでは、ブレイクポイントを設定して実行する場合は単体のフローデ
バッグ操作が基本となっている
並行実行するステップ以降でブレイクポイントを設定して複数のフローの実
行をブレイクさせた後、デバッガで個々のフローの実行コントロールができ
ない



OUTPERFORM THE FUTURE™

2.2.1フロー「CSVデータ取り込みパターン2」全体フロー図


- 全体フロー図



OUTPERFORM THE FUTURE™

2.2.2 フロー「CSVデータ取り込みパターン2」フロー図トリガー

- トリガーについて (Directory Scanner)

トリガー (1) | 

監視フォルダパス: EnvVal ('currentprojectdir') & 'Input_YubinCSV¥**Pattern2**'

移動先フォルダパス: EnvVal ('currentprojectdir') & 'Input_YubinCSV¥**Pattern2**¥done'

フィルタ: x-ken-all*.flg

上記のパスの**太字**の箇所がパターン1と異なる

コンポーネント設定: ディレクトリスキャナ

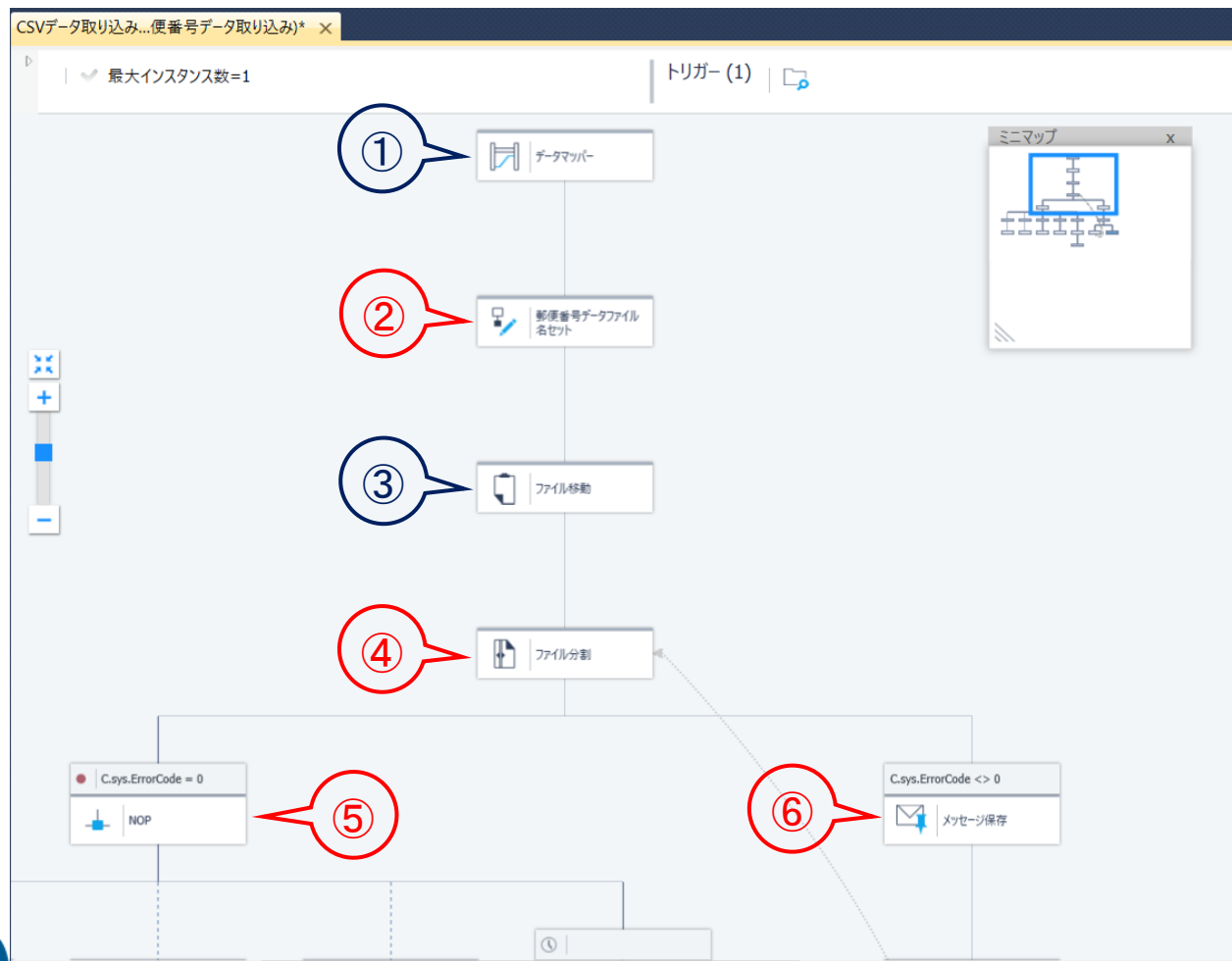
送り元	ディレクトリ	サブテ	フィルタ	アクション	送り先ディレクトリ
LAN	EnvVal ('currentprojectdir') & 'Input_YubinCSV¥Pattern1'	<input type="checkbox"/>	x-ken-all*.flg	Move	EnvVal ('currentprojectdir') & 'Input_YubinCSV¥Pattern1¥done'



OUTPERFORM THE FUTURE™

2.2.3 フロー「CSVデータ取り込みパターン2」フロー図その1

・ フロー図 その1



ステップ①③の処理の内容は、フロー「CSVデータ取り込みパターン1」と基本的に同じ

トリガーは、監視フォルダパスが異なる

②は、ファイル分割のための接頭辞をセットする処理を追加

④以降のステップが異なる

④：「ファイル分割」ステップ
ファイル分割アダプタを使用
行数指定により、約12万件の郵便番号データを4つに分割（以降のページで解説）

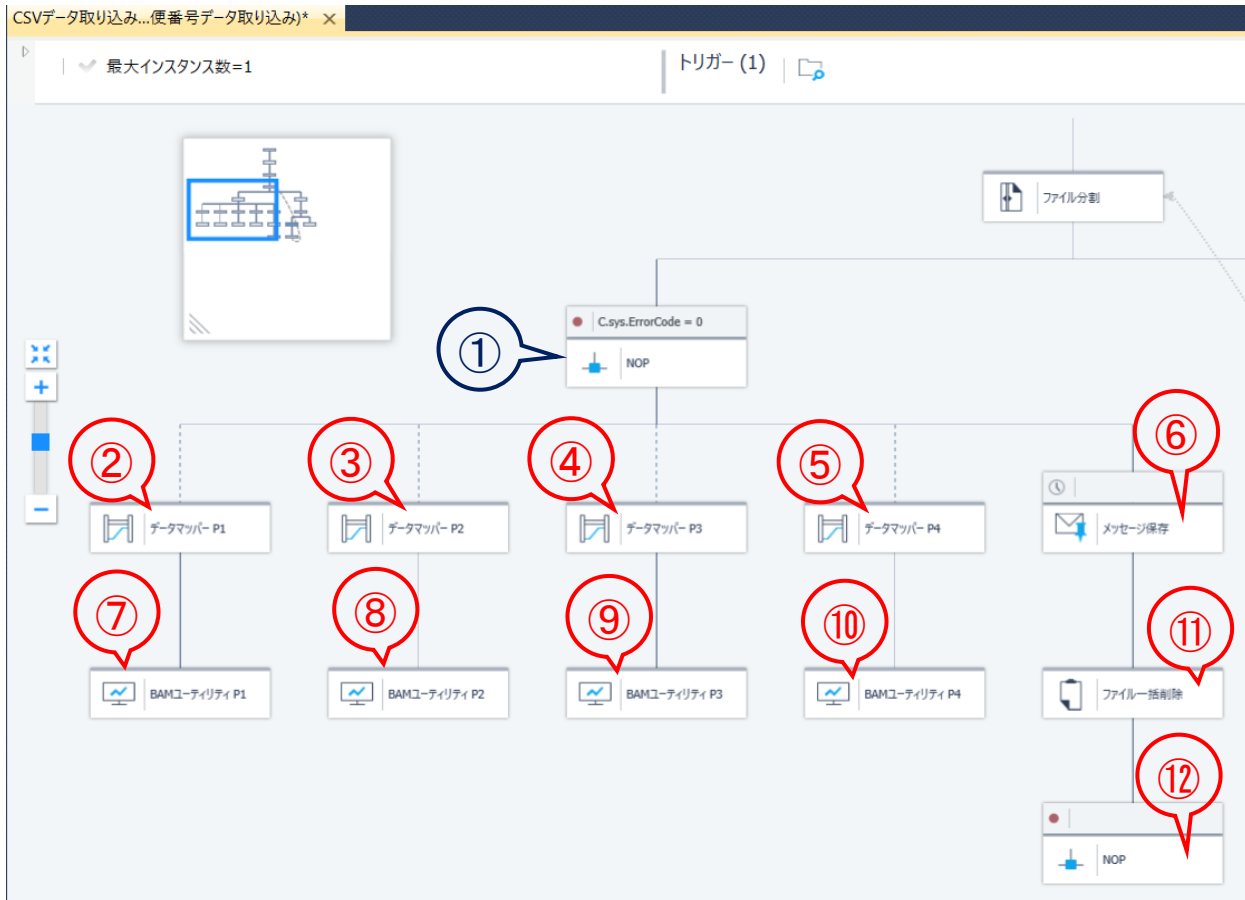
⑤：CSVデータ取り込みの並行実行の分岐のルートステップ（以降のページで詳細説明）

⑥：「ファイル分割」処理が失敗した場合のリトライ処理のルートステップ（以降のページで詳細説明）



2.2.4 フロー「CSVデータ取り込みパターン2」フロー図その2

■ フロー図 その2 (CSVデータ取り込み 並行実行処理の部分)

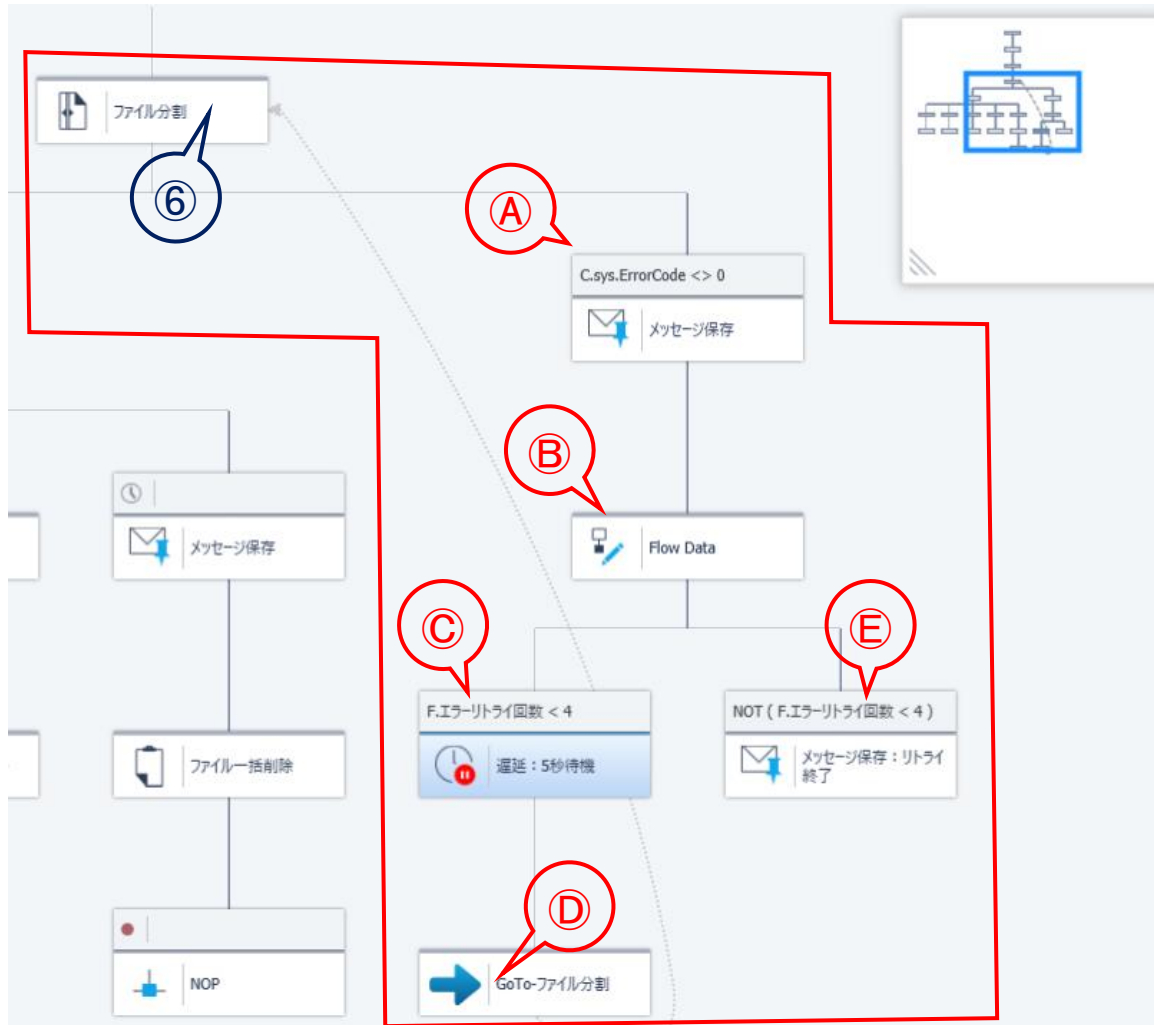


- ① : 「NOP」アダプタ
分岐ルートステップとして配置
条件 : C.Sys.ErrorCode = 0 (正常終了時)
- ②～⑤ : 「データマッパー」アダプタ
CSVデータ取込処理の「パラレル」モードのステップ。
4つの分割ファイルをそれぞれ使用している
- ⑥ : 「メッセージ保存」アダプタ
「リニア」モードのステップで「完了を待つ」をYesに
設定している
アクティビティログに指定したメッセージを出力する
(「完了を待つ」の詳細は2.2.8で解説)
- ⑦～⑩ : 「BAMユーティリティ」アダプタ
アクティビティログに指定したメッセージを出力する
(BAMユーティリティ用の確認タブ画面を利用することが出来る)
- ⑪ : 「ファイル管理」アダプタ
分割ファイルを一括削除するBATを実行
- ⑫ : 「NOP」アダプタ
デバッグ実行時のブレイクポイント設定するために配置



2.2.5 フロー「CSVデータ取り込みパターン2」フロー図その3

■ フロー図その3 ファイル分割ステップ以下のリトライ処理の箇所



Ⓐ : アクティビティログにメッセージを出力する
条件 : C.Sys.ErrorCode <> 0
C.Sys.ErrorCodeが0でない場合はエラーと判定されるので、エラー時はリトライ処理に遷移させる

Ⓑ : F.エラーリトライ回数をカウントアップする

Ⓒ : 5秒待機する
条件 : F.エラーリトライ回数 < 4 (4未満)

Ⓓ : ファイル分割ステップに遷移する (リトライする) GoToステップで「ファイル分割」に移動

Ⓔ : アクティビティログにリトライ回数の条件 (3回) に達したため処理を終了した旨のメッセージを出力してフローを終了する
条件 : NOT (F.エラーリトライ回数 < 4) (4未満の逆、4以上)



2.2.6「ファイル分割」ステップ④(フロー図その1のステップ④)

- 郵便番号CSVデータファイルの件数が12万超あるため、4つのファイルに分割する目的で配置して「ファイル」分割アダプタのステップ

変数「F.I_行数」(Numericタイプ)で設定している数値(現在の設定値(デフォルト値):35000)で12万超のデータを4つのファイルに分割している



2.2.6 「ファイル分割」ステップ④(フロー図その1のステップ④)(続き)

■ 設定画面

ダイレクトアクセスメソッド: ファイル分割

パラメータ名	タイプ	書式	入/出力	値
送り元タイプ	Numeric	1	In	ファイル
送り元ファイル名	Alpha	260	In	F.郵便番号データCSVファイルパス_送り先
送り先ディレクトリ	Alpha	260	In	EnvVal ('currentprojectdir') & 'Input_YubinCSV'
行数	Numeric	12	In	F.I_行数
送り先ファイル接頭辞	Alpha	30	In	F.ファイル分割接頭辞
上書き	Numeric	1	In	Yes
エンコーディング	Alpha	100	In	'Shift-JIS'
分割ファイル数	Numeric	12	Out	F.O_分割数
ファイルリストを返す	Blob		Out	F.O_分割ファイルリスト
エラーコード	Numeric	12	Out	

送り元タイプ : ファイル
送り元ファイル名 : F.郵便番号データCSVファイルパス_送り先 (※1)
送り先ディレクトリ : EnvVal ('currentprojectdir') & 'Input_YubinCSV¥Pattern2¥done¥split'
行数 : F.I_行数
送り先ファイル接頭辞 : F.ファイル分割接頭辞
上書き : Yes
エンコーディング : Shift_JIS
分割ファイル数 : F.O_分割数
ファイルリストを返す : F.O_分割ファイルリスト

※1 「ファイル移動」ステップの移動先のファイルパスを指定するので「_送り先」が名前にあるこの変数名で正しい

送り元タイプ : ファイル or BLOB
送り元ファイル名 : ファイル分割する元ファイルのファイルパスを指定する
送り先ディレクトリ : ファイル分割先のフォルダパスを指定する
行数 : 「Split By Lines」メソッドではこのパラメータで指定した行数で割ってファイルを分割する
送り先ファイル接頭辞 : ファイル分割したファイル名に付加する接頭辞を指定する
上書き : Yes or No
エンコーディング : 送り元ファイルのエンコーディングに合わせる
分割ファイル数 : 行数で指定した値で割った結果の分割されたファイル数の保存先の変数を選択する
ファイルリストを返す : 分割されたファイルの一覧を保存する変数を選択する



2.2.7 「NOP」ステップ(フロー図その2の①ステップ)

- CSVデータ取り込み処理の並行実行の分岐のルートステップとして配置

条件 : C.Sys.ErrorCode = 0

C.Sys.ErrorCodeは上のステップが実行された際にセットされ、値が0の場合は正常に終了したことを示す

この下に、CSVデータ取り込み処理(データマッパー)を4つ(処理モードはパラレル)とメッセージ保存を1つ(処理モードはリニア)を配置している



OUTPERFORM THE FUTURE™

2.2.8 「データマッパー Px」ステップ (フロー図その2の②～⑤のステップ)

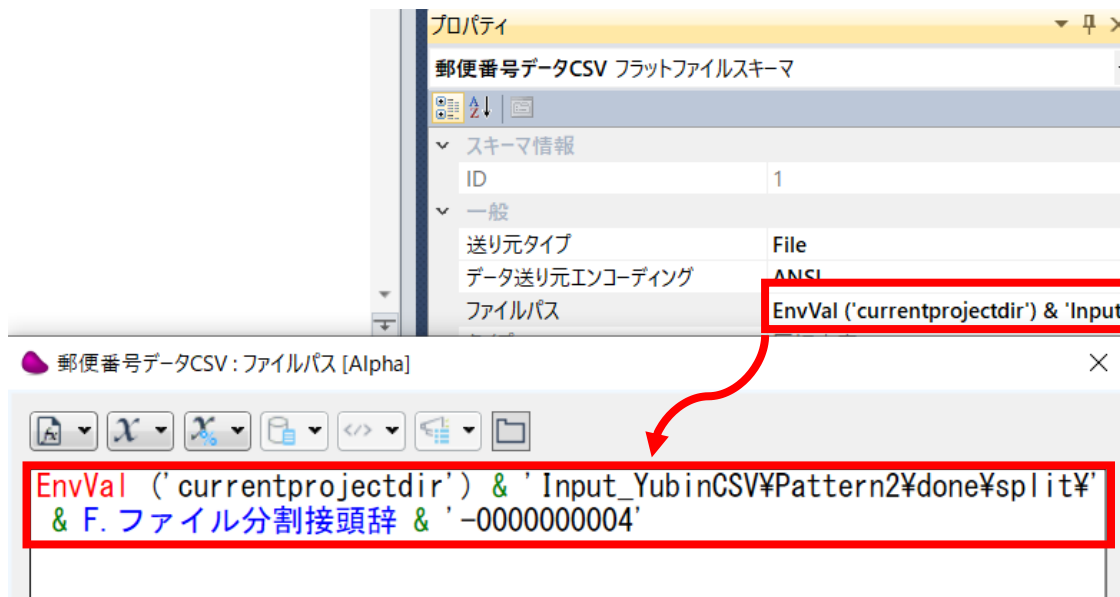
- CSVデータ取り込み処理の平行ステップ

名前の末尾にP1-P4を付加

各ステップのプロパティの処理モードは「平行」を選択

これにより、このフローとは異なる別のスレッドで並行実行させることができる。

他の平行ステップの「データマッパー Px」との違いは、送り元のFlatFileのプロパティのファイルパスが異なる(4分割されたファイル名を個別に指定)



プロパティ	
データマッパー P4 ステップ特性	
ステップ名	データマッパー P4
説明	
ステップID	19
コンポーネントID	11
コンポーネントタイプ	データマッパー
条件	
処理モード	平行
完了を待つ	No
アドバンスド	
セーブポイント	No
コールロジックフロー	No
マッパー	
エラー動作	Exit
DBトランザクション	Database_zipcloud
DBトランザクションレベル	Mapper



2.2.9 「メッセージ保存」ステップ (フロー図その2の⑥ステップ)

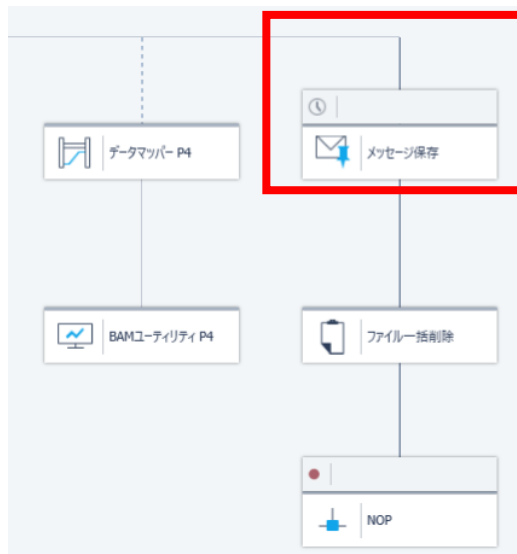
- 並行実行している4つの「データマッパー」が完了するまで処理を待つよう、このステップのプロパティの「完了を待つ」にYesを選択している

このパラメータは処理モードが「リニア」の場合のみ選択が可能である

「完了を待つ」というパラメータ名であるが、正常に終了するか、あるいはエラーでアボートする場合のどちらも含まれる

つまり、この「完了を待つ」がYesの「メッセージ保存」ステップは、同一レベルにある「平行」モードのステップがすべて完了するまで待機してから最後に実行される

最後に実行される「リニア」モードのこのステップが実行されてアクティビティログにメッセージが出力されることで、「平行」ステップがすべて完了したことを確認できることになる



処理モードがリニアのステップのステップ間の縦の線は実線で、平行モードは破線で表示される

「完了を待つ」=Yesのステップの左上には時計のアイコンが付加される

プロパティ	
メッセージ保存 ステップ特性	
General	
ステップ名	メッセージ保存
説明	
ステップID	12
コンポーネントID	6
コンポーネントタイプ	メッセージ保存
条件	
処理モード	リニア
完了を待つ	Yes
アドバンスド	
セーブポイント	No
コールロジックフロー	No



2.2.10 「BAMユーティリティ」ステップ (フロー図その2の⑦～⑩のステップ)

- 「パラレル」モードの「データマッパー」ステップの下の「リニア」モードの「BAMユーティリティ」アダプタのステップ

「データマッパー」ステップを並行実行するスレッドと同一のスレッドで実行される (BAMユーティリティステップの処理モードが「リニア」であるため)

「BAMユーティリティ」アダプタは、アクティビティログに指定したメッセージを出力する

「メッセージ保存」アダプタとの違いは、設定出来るパラメータが多いこととBAMユーティリティのメッセージを表示する専用のタブ画面がMagicモニタに用意されていること

BAM設定

メッセージ: 郵便番号データ取り込みデータマッパー P4: ' & IF (C.sys.LastErrorCode = 0, '成功', '失敗')

BLOB:

拡張子:

カテゴリ: 'ファイル分割 - CSV取り込み'

優先順位:

ステータス:

ユーザ キー1:

ユーザ キー2:

OK キャンセル

プロパティ

BAMユーティリティ P4 ステップ特性

General	
ステップ名	BAMユーティリティ P4
説明	
ステップID	25
コンポーネントID	16
コンポーネントタイプ	BAMユーティリティ
条件	
処理モード	リニア
完了を待つ	No
アドバンスド	
セーブポイント	No
コールドジョックフロー	No

2.2.11 「ファイル一括削除」ステップ (フロー図その2の⑪のステップ)

- 「ファイル管理」アダプタのステップ

Command Line メソッドを使用し、ファイル一括削除のバッチを使用して、分割したファイルを一括削除する

ファイル一括削除のバッチは、プロジェクトフォルダ配下のBATフォルダに配置

(BAT¥delete.bat)



```
delete.bat - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
@del /Q %1
1行、1列 100% Windows (CRLF) UTF-8
```

※ファイル管理アダプタの「Delete File」「Copy File」メソッドではパラメータにワイルドカード「*」指定は出来ないため、delete.batを用意した。



2.2.11 「ファイル一括削除」ステップ (フロー図その2の⑪のステップ) (続き)

■ 「ファイル一括削除」ステップの設定

ダイレクトアクセスメソッド: ファイル管理

#	名	条件
1	Command Line	

パラメータ名	タイプ	書式	入/出力	値
実行ファイル名	Alpha	260	In	EnvVal ('currentprojectdir') & 'BAT\delete
コマンドラインパラメータ	Alpha	1000	In	EnvVal ('currentprojectdir') & 'Input_YubinCSV
タイムアウト	Numeric	12	In	
ログファイル場所	Alpha	260	In	EnvVal ('log') & 'delete_bat.log'
Return Code	Numeric	12	Out	
エラーコード	Numeric	12	Out	

実行ファイル名 : 用意した「delete.bat」ファイルパスを指定

コマンドラインパラメータ : batの引数を指定

タイムアウト : 未設定 (0を指定したのと同義 : タイムアウトしない)

ログファイル場所 : コマンド (bat) を実行した際のエラーの出力先ファイルパスを指定

実行してエラーが発生した場合、ここに指定したファイルから、実行時のエラーメッセージを確認できる

ReturnCode : タイムアウトが未設定または0を設定した場合に機能するパラメータで、Numericタイプ変数を設定すると、実行後その変数にコマンドの戻り値がセットされる

コマンドの戻り値 (ReturnCode) を取得したい場合に利用する。ただし、タイムアウトに値を設定すると利用できない。タイムアウトとReturnCodeはどちらか一方だけ利用できると理解しておくこと



OUTPERFORM THE FUTURE™

2.2.12 「NOP」ステップ(フロー図その2の⑫のステップ)

- デバッグ実行時のブレイクポイント設定するために配置した「NOP」アダプタ
デバッグ実行時、フローの最終ステップでブレイクさせたい場合のために配置している
「NOP」アダプタ自体は何も処理をしない

デバッグ実行時にブレイクポイントを設定していない時やデバッグ実行でなく通常の実行時には実行されるが、何も処理はしなく、このステップ実行後エラーシステム変数である C.Sys.ErrorCode には「0」(正常終了)がセットされる

※このステップの処理モードが「リニア」になっているので、他のパラレルに設定しているステップのフロー内のステップでブレイクポイントを設定していな状態であれば、ここにブレイクポイントを設定してブレイクできることを確認済み



2.2.14 「メッセージ保存」ステップ①A (フロー図その3のステップ①A)

- アクティビティログに指定したメッセージを出力する「メッセージ保存」アダプタのステップ
条件 : C.Sys.ErrorCode <> 0 (<>記号はNot Equal を示す記号
エラーシステム変数 C.sys.ErrorCodeの値が「0」でないと判定された場合にこのステップに遷移される
「ファイル分割」ステップでエラーが発生した場合にこのステップが実行されるので、そのことを示すメッセージをアクティビティログに出力している



2.2.15 「Flow Data」ステップ③(フロー図その3の③のステップ)

- 「F.エラーリトライ回数」変数をカウントアップする「Flow Data」アダプタのステップ
エラー発生時のリトライ回数をカウントし、回数の上限值を管理することを目的とする
このフローでは、リトライ回数の上限值(ロジックで固定)を3回とし、以降のステップの条件で、その上限値を使用して判定している。
「F.エラーリトライ回数」の値が上限値未満であれば、「ファイル分割」ステップを繰り返すイメージ



2.2.16 「遅延」ステップ③(フロー図その3の③のステップ)

- 指定した時間待機させるために配置した「遅延」アダプタのステップ

指定する時間の単位は10分の1秒である

例えば、「50」という値を指定すると5秒待機する

「ファイル分割」ステップが失敗した場合に、即リトライするのではなくて、一定時間待機させてからリトライさせる目的で配置している



2.2.17 「Goto」(フロー図その3のD)

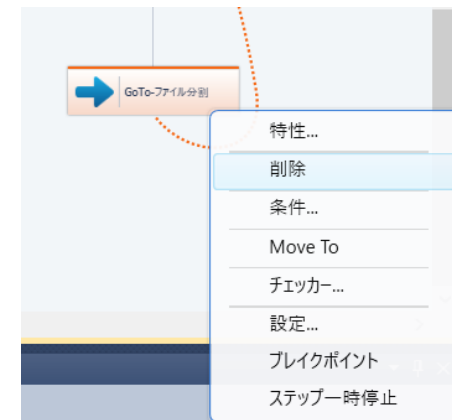
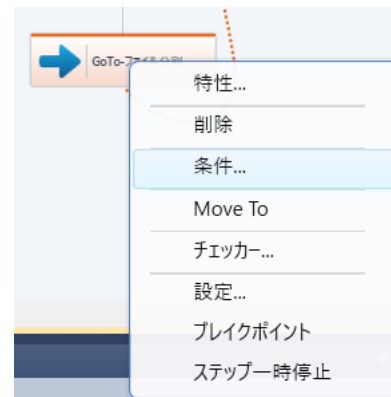
- 指定したステップに遷移させるために「Goto」を利用することが出来る

条件を指定しなければ、無条件に遷移し、条件を指定すればその条件を満たす場合遷移する
「Goto」を利用するには、ステップを選択して右クリックすると表示されるメニューから「Goto」をクリックし、マウスマーカーの形が変わり「Select Goto Destination」と表示された後、遷移したいステップをクリックする

選択したステップの下に「Goto」が表示され、クリックしたステップに向けて破線が引かれる

「Goto」に条件を設定するには、「Goto」を右クリックして表示されるメニューから「条件」をクリックすると表示される式エディタ画面にて、条件式を記述すればよい

「Goto」を右クリックして表示されるメニューには、いろいろなメニューがあり、「Goto」の削除もこのメニューから「削除」を選択すれば良い



2.2.18「メッセージ保存:リトライ終了」ステップ ⑤(フロー図その3の⑤ステップ)

- F.エラーリトライ回数が上限値(4回)に達したことを、アクティビティログにメッセージ出力させる目的で配置したステップ

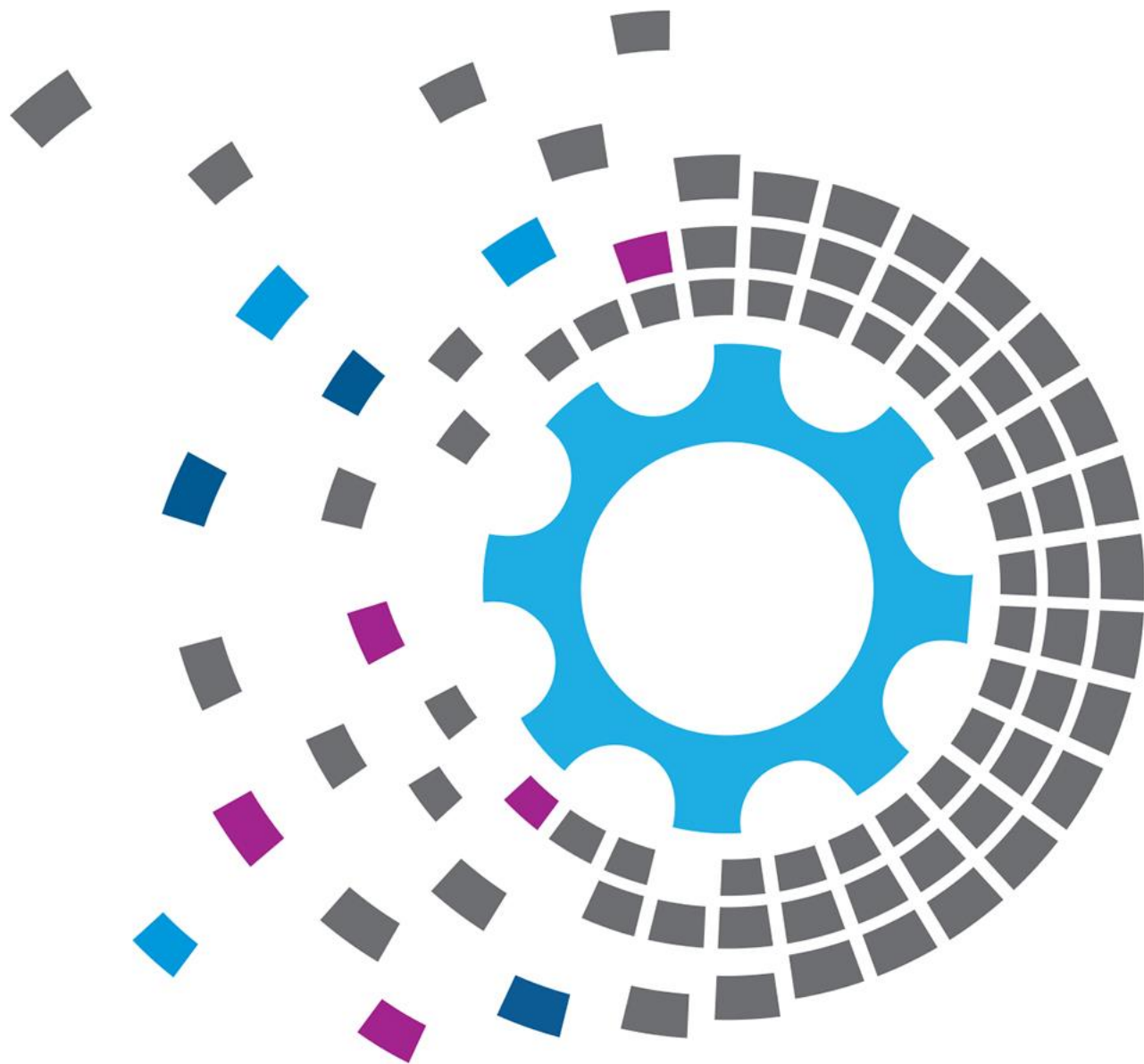
条件: NOT (F.エラーリトライ回数 < 4)

(F.エラーリトライ回数 < 4)の否定、つまりF.エラーリトライ回数 < 4 の判定結果がFalseの場合にTrueと判定される

このステップの処理自体は必須ではないが、F.エラーリトライ回数が上限(3回を超える)に達したことを明確に示すために、上限値を超えた旨を示すための処理のステップを配置することをお勧めしたい



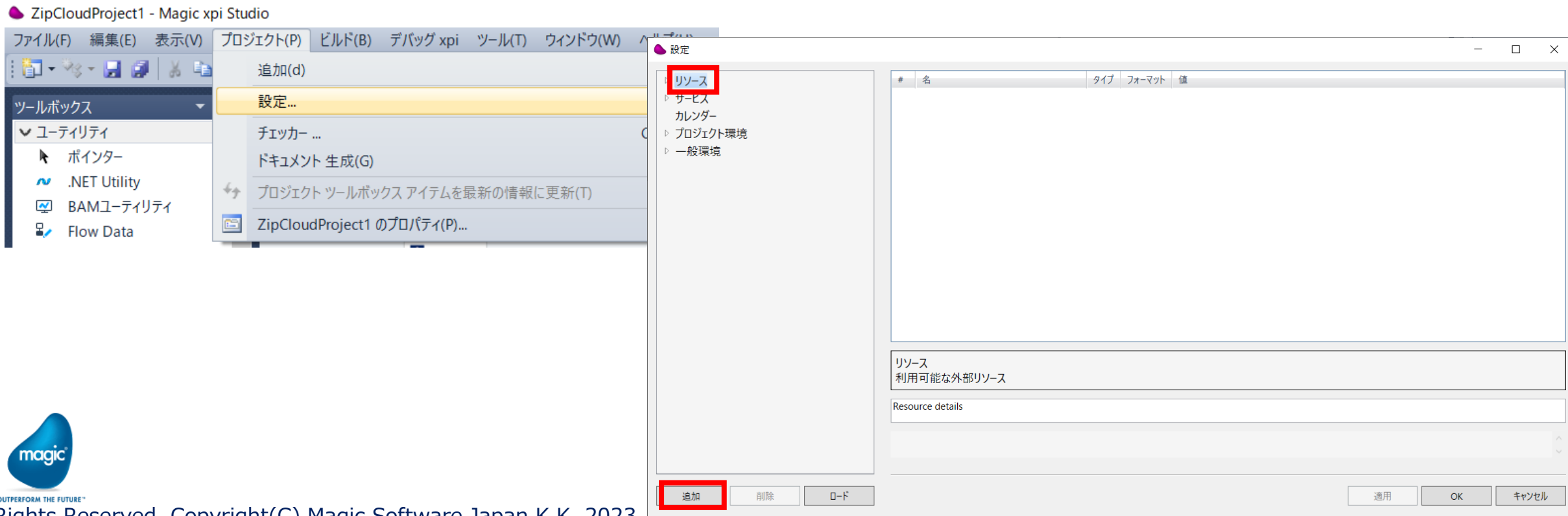
補足



OUTPERFORM THE FUTURE™

a) リソースの新規登録

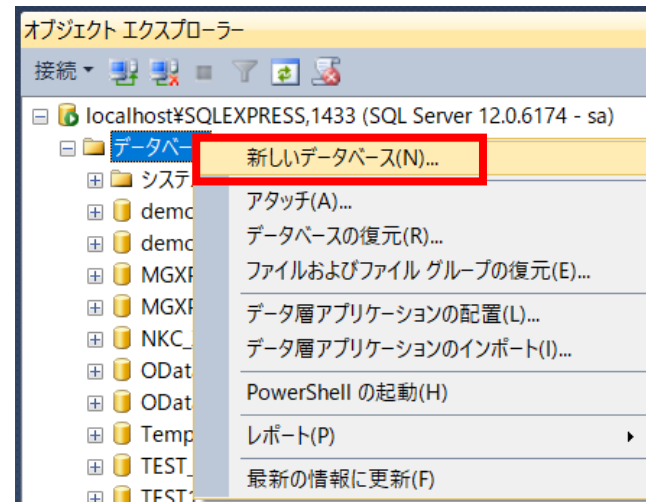
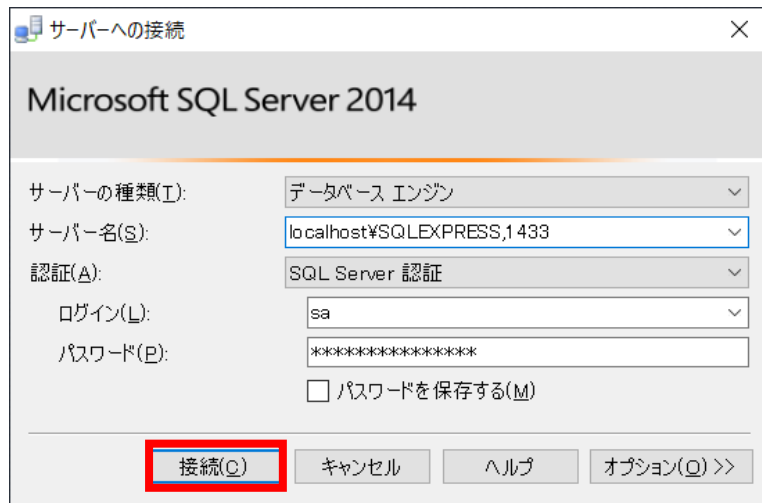
- Magic xpiスタジオでプロジェクトを作成するか、既存のプロジェクトを開いた状態にする。
- メニューのプロジェクトー設定をクリックし、設定画面を開く。
- 設定画面の左ペインのツリーの上部のリソースにカーソルがある状態のとき、設定画面左下部の[追加]ボタンをクリックすると、新たなリソースが作成できる。



OUTPERFORM THE FUTURE™

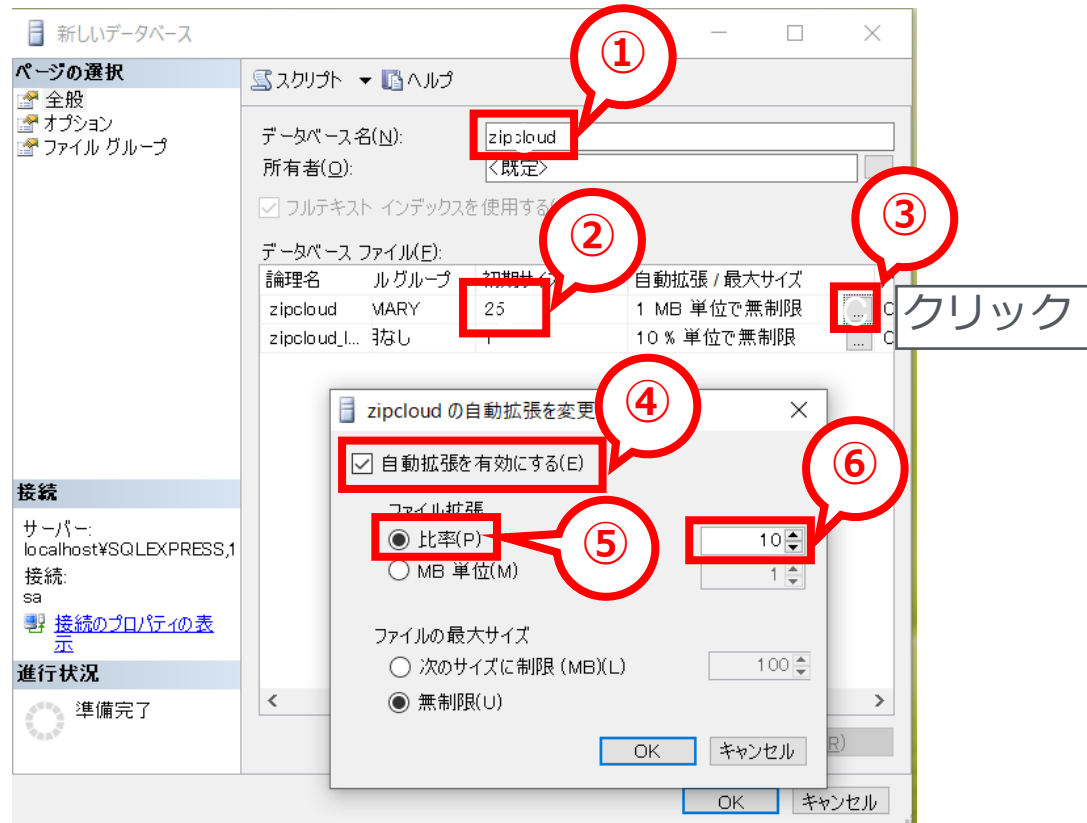
b) データベースの作成手順

- SQL Server Management Studio (SSMS) を使用して、データベースを作成する手順について説明する
- SSMSを起動し、内部DB用のデータベースサーバであるSQL Serverに接続する
- オブジェクト エクスプローラーペインのツリーを展開し、データベースを右クリックして「新しいデータベース(N)」をクリックする



b) データベースの作成手順(続き)

- 「新しいデータベース」の設定画面が開く
- 設定内容は下記の通り



- ① : データベース名は「zipcloud」とする
- ② : 初期サイズを25MBに変更する
- ③ : 自動拡張の設定画面を表示するため、[...]ボタンをクリックする
zipcloudの自動拡張を変更する画面表示される
- ④ : 自動拡張を有効にするにチェックされていることを確認する
- ⑤ : ファイル拡張は比率を選択する
- ⑥ : 比率は10 (%)

[OK]ボタンをクリックして順に画面を閉じ、データベースを作成する



c) リソースのパラメータでユーザー環境変数を使用する場合の注意点

- ユーザー環境変数を登録する場所と値に設定する文字

設定画面において、プロジェクト環境と一般環境の2カ所にユーザー環境変数の設定箇所が存在し、それぞれ保存先のファイルが異なる。

(プロジェクト環境:プロジェクトのifs.ini 一般環境:magic.ini)

また、値に記号を含む場合は、一般環境のユーザー環境変数を利用すること

(プロジェクト環境びユーザー環境変数の場合、値に記号があると、実行時にうまく利用できないことがある。上手くいかない場合はユーザー環境変数を使用するのを止めて、値欄に直接文字列を設定して確認するようにしてみるなどの対応を行ってください)

- 使用するユーザー環境変数の名前

データベースリソースのパラメータでユーザー環境変数を使用する場合、リソース名を含むユーザー環境変数名を使用しないようにしてください。

例えば、データベースリソース名がzipcloudの場合、zipcloud_DBやzipcloud_USERやzipcloud_SERVERなどのユーザー環境変数をデータベース名、サーバ、ユーザなどのパラメータで使用すると、実行時に正常に利用できないことが確認されています。ユーザー環境変数名にリソース名の使用を避けるなどの対応を検討ください



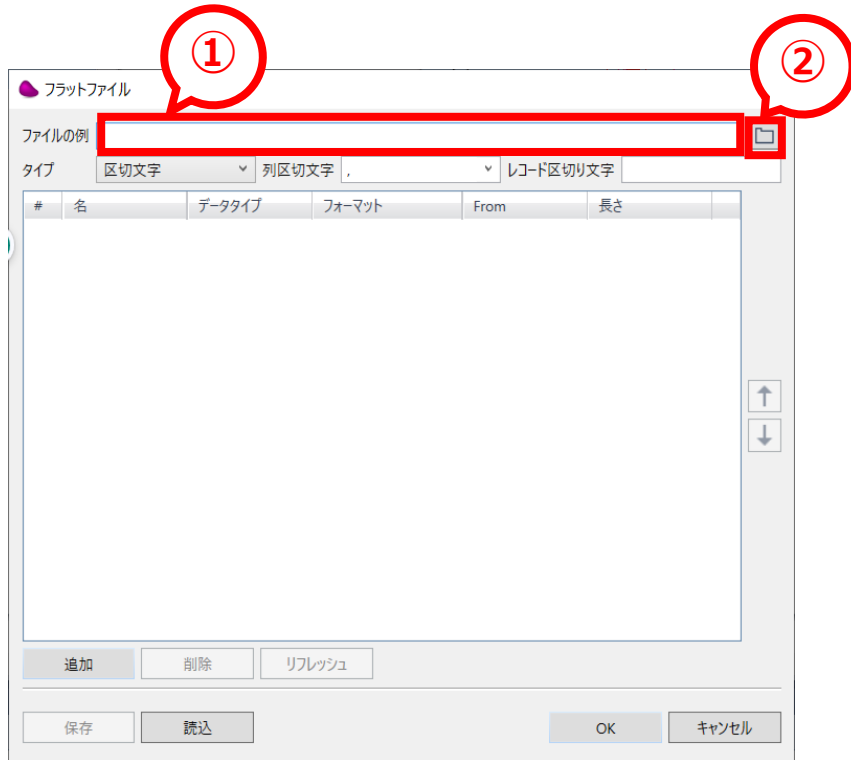
d) 郵便番号マスタテーブル作成

- SQL Server ManagementStudioを起動して、内部DB用に使用しているデータベースサーバに接続する
- プロジェクト「zipcloudProject1」のフォルダの下「DB_zipcloud_create_table_sql」の下にある「Create_ziptable_zipcloud.sql」をSQL Server ManagementStudioにドラッグ & ドロップして開き、そのSQL文(CreateTable文)を実行して、郵便番号マスタテーブルを作成する
- ツリーペインのツリーのデータベース「zipcloud」を展開し、テーブルに郵便マスタテーブル「ziptable_zipcloud」が作成されていることを確認する



f) FlatFileの詳細(フィールド定義)でのTips その1

- FlatFileの詳細(フィールド定義)でフィールド名を容易に設定する方法
指定した区切り文字を使用してフィールド名だけを入力したCSVファイルを用意する



①：ファイルの例欄に、用意したフィールド名を設定したCSVファイルのパスを設定する

②：右横のフォルダアイコンをクリックすると、ファイル選択ダイアログが開く。
そのダイアログ画面で、用意したCSVファイルがあるフォルダを開き、CSVファイルを選択すると、①の欄にそのファイルのパスをセットすることが出来る

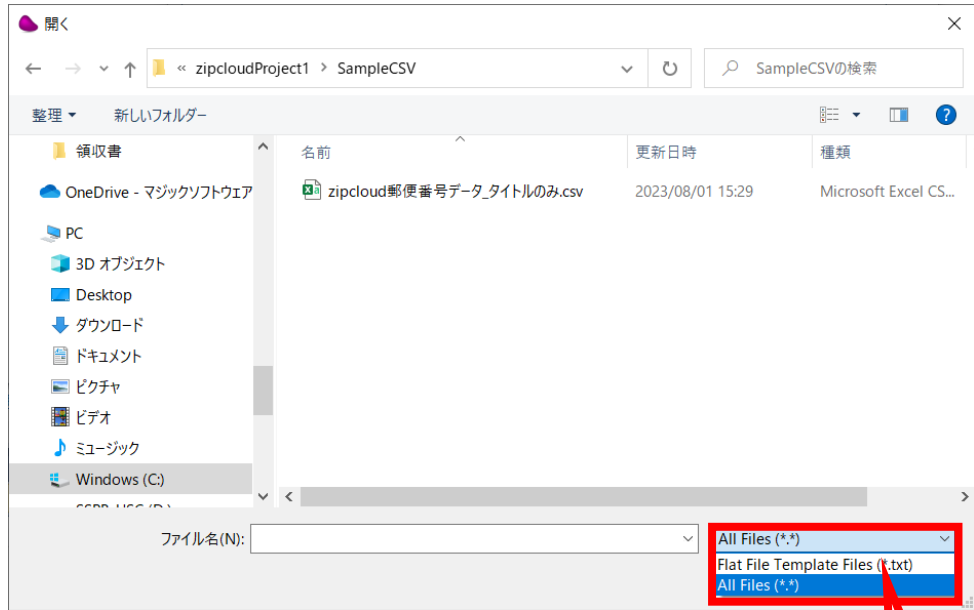


f) FlatFileの詳細(フィールド定義)でのTips その1(続き)

- FlatFileの詳細(フィールド定義)でフィールド名を容易に設定する方法(続き)

ファイルの例欄の右横のフォルダアイコンをクリックして表示されるファイル選択ダイアログ

注意点:



①: ファイル名欄右横のコンボボックスはデフォルトは「Flat File Template Files (*.txt)」になっているそのため、ファイル拡張子が「.txt」のものしか表示されない。

このコンボボックスから、「All Files (*.*)」を選択すると、すべてのファイルが表示されるようになるので、拡張子「.CSV」ファイルを選択する場合は、「All Files (*.*)」に変更してください。

ちなみに、タイトルのみのCSVファイルを用意する際に予めファイル拡張子を「.txt)」にしておいても結構です。

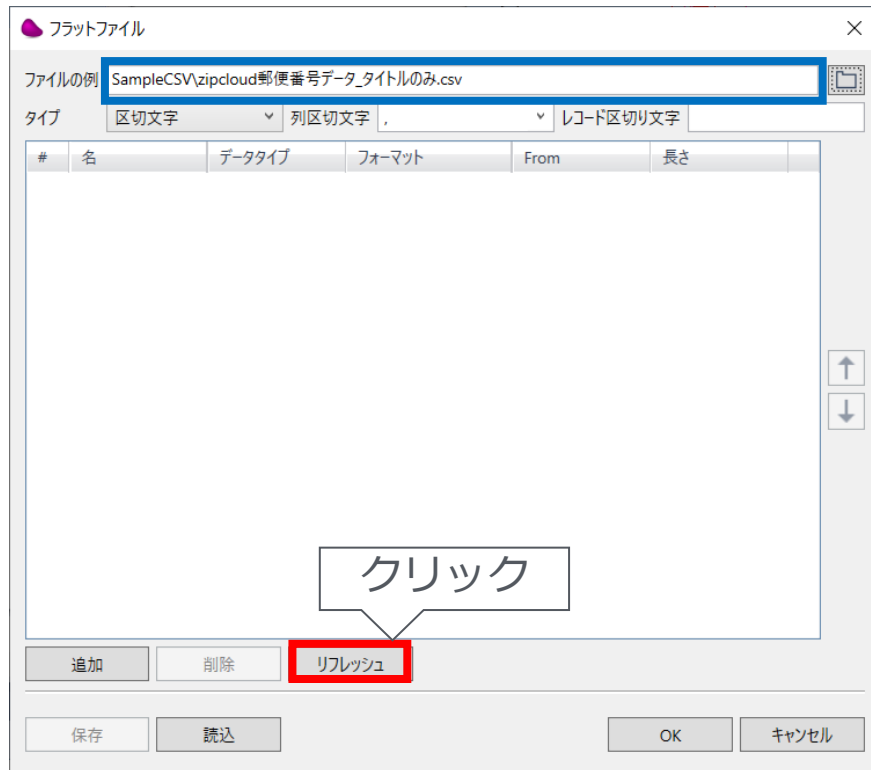


OUTPERFORM THE FUTURE™

f) FlatFileの詳細(フィールド定義)でのTips その1(続き)

- FlatFileの詳細(フィールド定義)でフィールド名を容易に設定する方法(続き)

ファイルの例欄にタイトルのみ設定されているCSV形式のファイルパスが設定されている状態



ファイル名欄にCSV形式のフォーマットでタイトルがセットされているファイルのパスがセットされている状態になっていれば、画面下部の[リフレッシュ]ボタンをクリックできるようになる

この[リフレッシュ]ボタンをクリックすると、タイトルの文字列が、画面中央のフィールド定義の箇所に取り込んでくれる(次ページの画面)



f) FlatFileの詳細(フィールド定義)でのTips その1(続き)

- FlatFileの詳細(フィールド定義)でフィールド名を容易に設定する方法(続き)

[リフレッシュ]ボタンをクリックした後、フィールド定義の箇所の名に、各フィールドのタイトルが取り込まれている

#	名	データタイプ	フォーマット	From	長さ
1	方公共団体コード	Alpha	30		
2	旧郵便番号	Alpha	30		
3	郵便番号	Alpha	30		
4	都道府県名カナ	Alpha	30		
5	市区町村名カナ	Alpha	30		
6	町域カナ	Alpha	30		
7	都道府県名	Alpha	30		
8	市区町村名	Alpha	30		
9	町域	Alpha	30		
10	一町域が二以上の	Alpha	30		
11	小字毎に番地が延	Alpha	30		
12	丁目を有する町域	Alpha	30		
13	一つの郵便番号で	Alpha	30		
14	更新の表示	Alpha	30		
15	変更理由	Alpha	30		

[リフレッシュ]ボタンをクリックした後すると、タイトルの文字列が、画面中央のフィールド定義の箇所に取り込まれる
このとき文字化けとか、上手く取り込まれない場合は、Flat Fileのプロパティのエンコーディングの設定値と実際のタイトル行がセットされているCSVファイルのエンコーディングが一致していることを確認し、異なる場合は適宜調整する
また、この画面のタイプが区切り文字で列区切り文字がCSVファイルの形式と一致していることを確認し、異なる場合は適宜調整する

データタイプ欄は「Alpha」が選択され、フォーマット欄は30が固定で自動的にセットされる
これらは、実際に使用するCSVデータの各フィールドの最大長やデータ形式に合わせて個別に設定する



f) FlatFileの詳細(フィールド定義)でのTips その2

- FlatFileの詳細(フィールド定義)を行ったもの情報を利用する方法

既存のフィールド定義の画面を開き、画面左下の[保存]ボタンをクリックしてFlatFileスキーマ定義をファイルに保存する。

後で分かるように、確認しやすい名前を付けることをお勧めする

設定したい詳細(フィールド定義)の画面を開き、画面左下の[読込]ボタンをクリックして、利用したいFlatFileスキーマ定義ファイルを選択し、[OK]ボタンをクリックする

定義情報が読み込まれて反映される

※[読込]ボタンと[保存]ボタンを間違えないようにしてください。

操作を誤ると、既存のスキーマ定義ファイルを上書き保存してしまうと取り返しがつかなくなります。

(スキーマ定義ファイルのバックアップを常に行っておくことはとても重要です)



f) FlatFileの詳細(フィールド定義)でのTips その2

- FlatFileの詳細(フィールド定義)を行ったもの情報を利用する方法

※注意

[保存]ボタンで保存したFlat File スキーマ定義ファイルを編集して[読込]を行うことは止めてください

Flat File スキーマ定義ファイルのフォーマットについては公開されていません

独自に編集したスキーマ定義ファイルを読み込んで利用することで、予期しない動作に繋がる恐れがある



OUTPERFORM THE FUTURE™

g) StrToken関数

- 例えば、カンマで区切られている文字列の各フィールドのデータを取得したいような場合に、この関数を利用することで用意に取得することが出来る。

区切り文字を含む文字列の例: 0600000,北海道,札幌市中央区

カンマで区切られているこの文字列の第一フィールド(0600000)の文字列を取得するには、次のようにStrToken関数を使う

```
StrToken ( '0600000,北海道,札幌市中央区' , 1 , ',' )
```

StrToken関数の構文は、StrToken ([文字列(Alphaタイプ) , [トークンインデックス (Numericタイプ)], [区切り文字 (Alphaタイプ)])

戻り値: 指定したトークンインデックスの文字列 (Alphaタイプ)

第二フィールドの「北海道」を取得したい場合は、トークンインデックスに「2」を指定すれば良い

この関数は区切り文字を境にして文字を抽出してくれるので、開始位置や文字数(バイト数)を指定しづらい場合この関数を活用できると重宝する

2023年7月20日や2023年10月10日の7月の「7」や10月の「10」を取得したい場合だと取得したい文字数が異なるが、StrToken (StrToken (F.日付 , 1 , '月') , 2 , '年')で取得できる(※「F.日付」変数に「2023年7月20日」や「2023年10月10日」などの文字列が保存されているとする)

(月を区切り文字としその第一フィールド「2023年7」または「2023年10」を取得し、年を区切り文字としその第二フィールド「7」または「10」が取得される)



h) データマッパーのマルチアップデート

- データマッパーの送り先ノードのフォルダアイコンのプロパティのマルチアップデートは、条件欄よりも先に実行されることを理解しておくことはとても重要
- マルチアップデートに設定している各行の条件欄に条件式を設定していれば、その条件を満たした場合に実行され、条件欄を設定していなければレコードが読み込まれる度に実行される

