

XML子要素同士の結合と出力例1

2024-04-15

1つのXML内の階層が違うノード要素を組み合わせる例

出力ファイルを分割する場合は[XML子要素同士の結合と出力例2](#)を参照。

希望の結果

XML内のfactoriesノードとitemsノードを結合して1行にしたい。ファイルは複数ある。

入力XML1

[report-IN1.xml](#)

```
<?xml version="1.0" encoding="UTF-8"?>
<info>
  <TITLE>これはサンプルです1</TITLE>
  <factories>
    <factoryA001><rank>A</rank><mainItem>P005</mainItem><capacity>2</capacity><note>精密加工</note></factoryA001>
    <factoryB001><rank>C</rank><mainItem>S001</mainItem><capacity>2</capacity></factoryB001>
    <factoryB010><rank>C</rank><mainItem>S020</mainItem><capacity>5</capacity><note>生産力に余力あり</note></factoryB010>
    <factoryC002><rank>A+</rank><note>ベテラン多数</note><mainItem>P005</mainItem><capacity>1</capacity></factoryC002>
    <depoD100><note>立地</note><capacity>3</capacity></depoD100>
    <depoD101><note>倉庫に問題</note><capacity>1</capacity></depoD101>
    <depoE201><note>ピックに問題</note><capacity>2</capacity></depoE201>
  </factories>
  <items>
    <item name="P001" desc="P型" type="C" />
    <item name="P005" desc="P型" type="C" />
    <item name="S001" desc="S型" type="X" />
    <item name="S020" desc="S型赤" type="X" />
    <item name="F001R" desc="F型復刻" type="J" />
  </items>
</info>
```

入力XML2

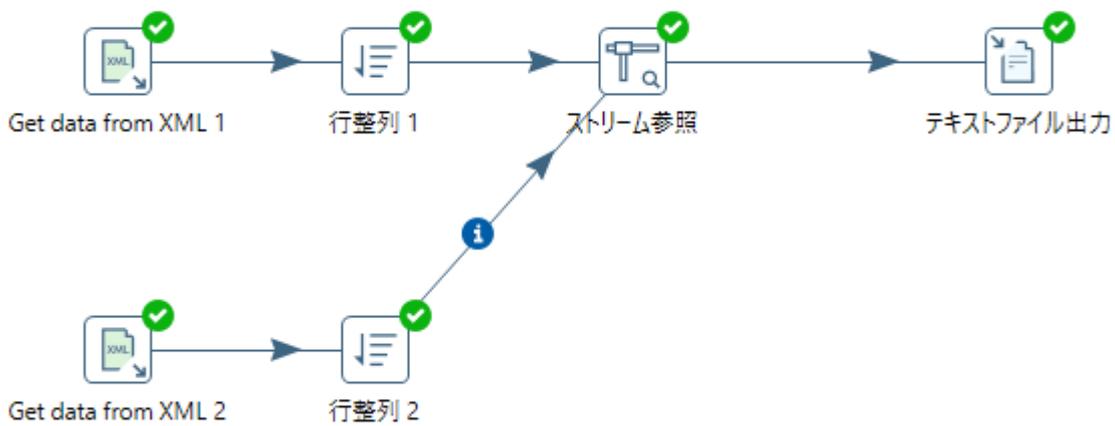
[report-IN2.xml](#)

```
<?xml version="1.0" encoding="UTF-8"?>
<info>
  <TITLE>これはサンプルです2</TITLE>
  <factories>
    <factoryF001><rank>C</rank><mainItem>P002</mainItem><capacity>2</capacity></factoryF001>
    <factoryG001><rank>C</rank><mainItem>S020</mainItem><capacity>3</capacity></factoryG001>
    <factoryG010><rank>D</rank><mainItem>S030</mainItem><capacity>2</capacity><note>設備老朽化</note></factoryG010>
    <depoA100><note>立地</note><capacity>3</capacity></depoA100>
    <depoC101><note>倉庫に問題</note><capacity>1</capacity></depoC101>
    <depoJ201><note>ピックに問題</note><capacity>2</capacity></depoJ201>
  </factories>
  <items>
    <item name="P002" desc="P型" type="C" />
    <item name="S030" desc="S型青" type="X" />
    <item name="S020" desc="S型赤" type="X" />
    <item name="J001" desc="J型" type="J" />
  </items>
</info>
```

入力XML1
出力CSV
report-OUT.csv factory,rank,mainItem,capacity,note,desc factoryF001,C,P002,2,,P型 factoryA001,A,P005,2,精密加工,P型 factoryC002,A+,P005,1,ベテラン多数,P型 factoryB001,C,S001,2,,S型 factoryB010,C,S020,5,生産力に余力あり,S型赤 factoryG001,C,S020,3,,S型赤 factoryG010,D,S030,2,設備老朽化,S型青

定義

フロー

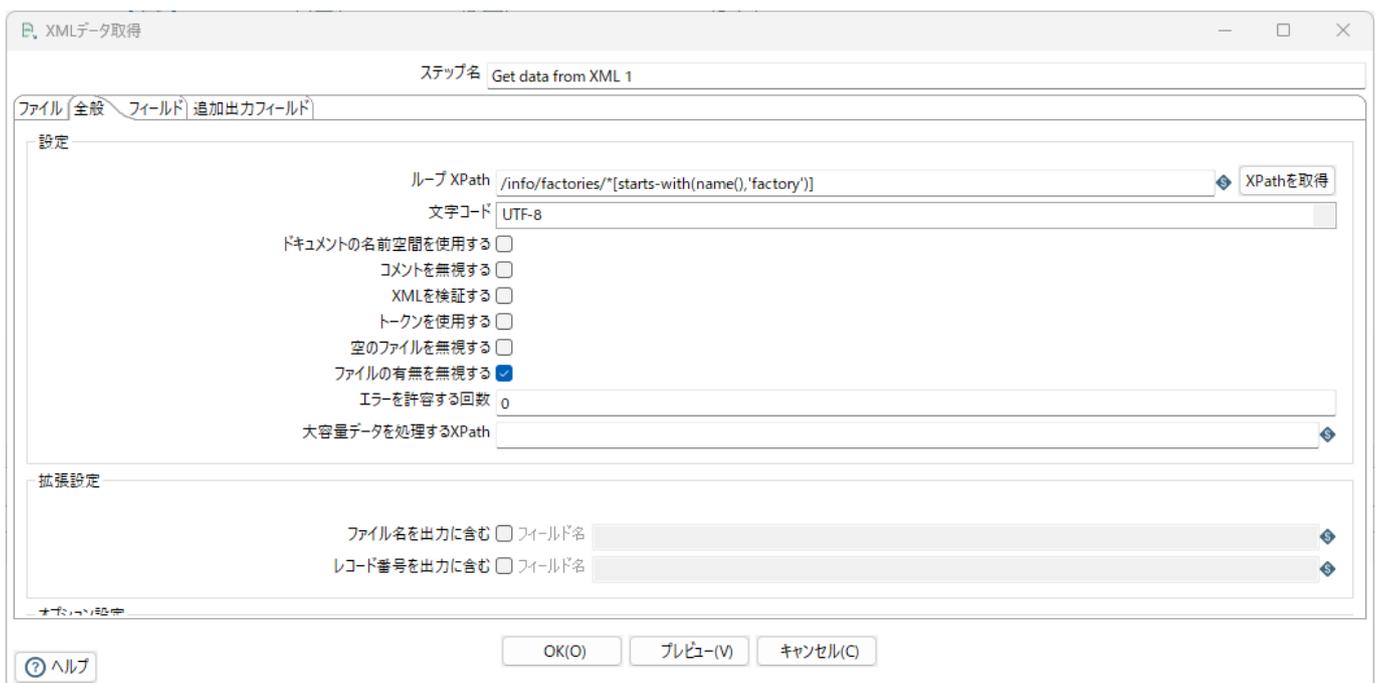


Get data from XML1

factoriesノードの読み出し部分。正規表現で所定の形式の名称を指定し、複数ファイルを読み込み対象にする。



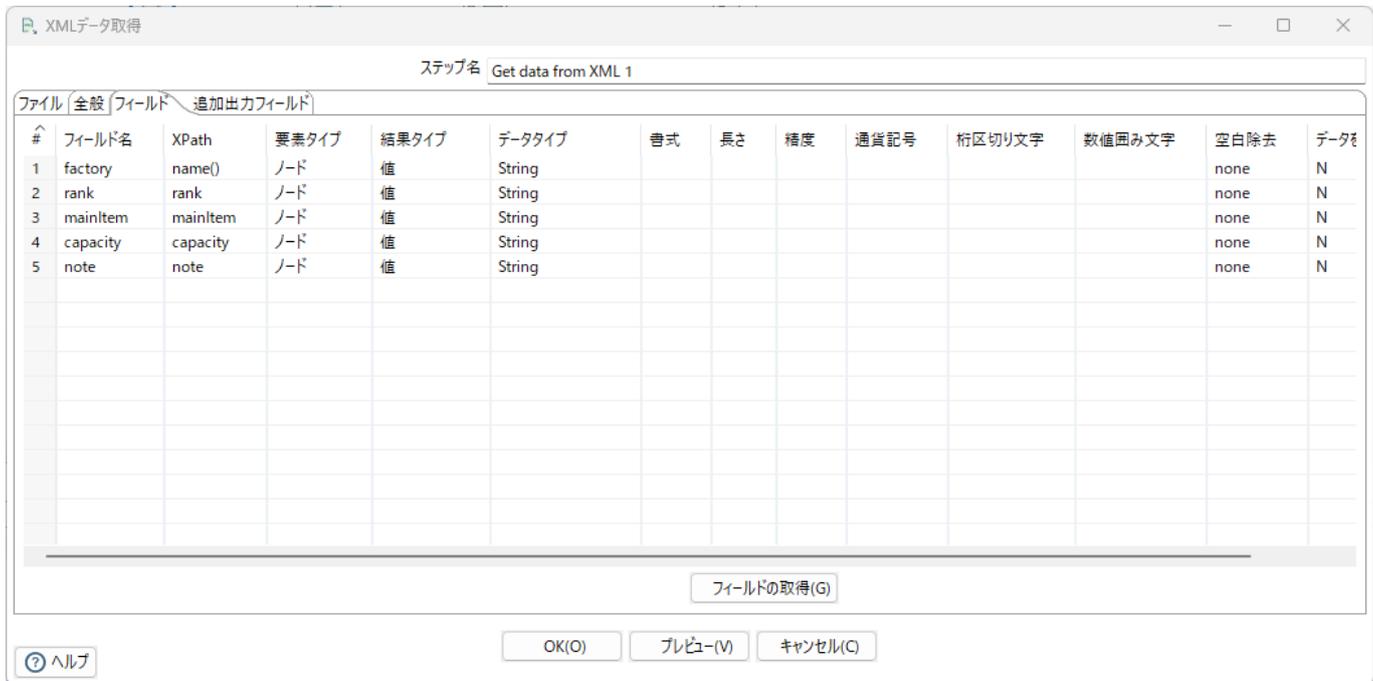
ループ XPath では、関数starts-with(),name()で処理対象ノードを指し示している。



```
/info/factories/*[starts-with(name(),'factory')]
```

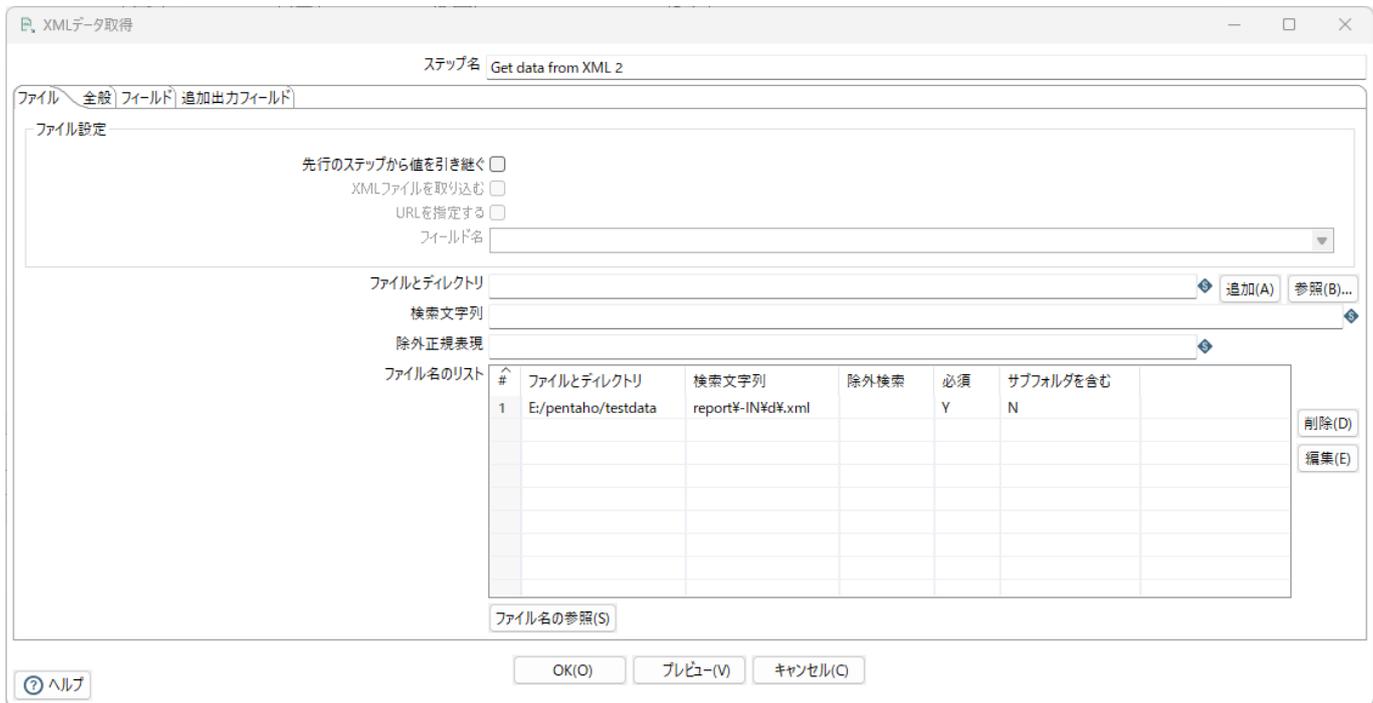
starts-with()は前方一致の検証関数、name()はノードの名前を返す関数。
この例では、名前が“factory”から始まるノードを処理対象とする指定になる。

そしてその選択されたノードから以下の情報を抜き出す。



Get data from XML2

itemsノードの読み出し部分。正規表現で所定の形式の名称を指定し、複数ファイルを読み込み対象にする。



ループXPathで “ /info/items/item” を指定する[]itemノードが処理対象になる。

XMLデータ取得

ステップ名 Get data from XML 2

ファイル (全般 | **フィールド** | 追加出力フィールド)

設定

ループ XPath /info/items/item XPathを取得

文字コード UTF-8

ドキュメントの名前空間を使用する

コメントを無視する

XMLを検証する

トークンを使用する

空のファイルを無視する

ファイルの有無を無視する

エラーを許容する回数 0

大容量データを処理するXPath

拡張設定

ファイル名を出力に含む フィールド名

レコード番号を出力に含む フィールド名

オプション設定

ファイル名を結果に含む

ヘルプ OK(O) プレビュー(V) キャンセル(C)

そしてその選択されたノードから以下の情報を抜き出す。

XMLデータ取得

ステップ名 Get data from XML 2

ファイル (全般 | **フィールド** | 追加出力フィールド)

#	フィールド名	XPath	要素タイプ	結果タイプ	データタイプ	書式	長さ	精度	通貨記号	桁区切り文字	数値固み文字	空白除去	データを代替する
1	name	@name	属性	値	String							none	N
2	desc	@desc	属性	値	None							none	N

フィールドの取得(G)

ヘルプ OK(O) プレビュー(V) キャンセル(C)

行整列 1,行整列 2

それぞれ mainItem, name でXMLからの抽出結果をソートする。ソートしないと次のストリーム参照が正しく機能しない。

ストリーム参照

行整列 1の抽出項目 mainItem をキーに行整列 2を検索する。比較する項目nameをキー項目とする。

キーに一致するものがあれば項目name, descを、一致しなければnullを付与する。

ストリームルックアップ

ステップ名

ステップ名

フィールド (参照)

#	フィールド名	フィールド名 (ルックアップ)
1	mainItem	name

フィールド (検索)

#	フィールド名 (戻り値)	フィールド名 (新規)	デフォルト	データタイプ
1	name			String
2	desc			String

メモリにキャッシュする

メモリハキャッシュされたデータを並替える

ハッシュテーブルを使用する

ヘルプ OK(O) キャンセル(C) フィールドの取得(G) ルックアップフィールドの取得(L)

テキストファイル出力

出力先をreport-OUT.csv とする。

テキストファイル出力

ステップ名

ファイル

ファイル名

出力をサブレットに渡す

親フォルダを作成

開始時にファイルを作成しない

先行のステップから値を引き継ぐ

フィールド名

拡張子

ファイル名にステップ番号を含む

ファイル名に区切り番号を含む

ファイル名に日付を含む

ファイル名に時刻を含む

日付の表示形式を指定する

日付表示形式

ファイル名を結果に含む

ヘルプ OK(O) キャンセル(C)

カンマ区切りでMS932(windows-31j)の文字コードで出力する。

テキストファイル出力

ステップ名 テキストファイル出力

ファイル [全般 フィールド]

既存のファイルに追加する

区切り文字 ,

引用符 "

フィールドを引用符で囲む

囲み込みフィックス無効化

ヘッダー

フッター

フォーマット CR+LF terminated (Windows, DOS)

圧縮形式 None

文字コード windows-31j

フィールドの右側を空白で埋める

ダンプ出力する

指定されたレコード数でファイルを分ける

終了レコードを追加する

ヘルプ OK(O) キャンセル(C)

新規に行を生成します。

項目を絞っている。

テキストファイル出力

ステップ名 テキストファイル出力

ファイル [全般 フィールド]

#	フィールド名	データタイプ	書式	長さ	精度	通貨記号	桁区切り文字	数値囲み文字	空白除去	NULL可能
1	factory	String							なし	
2	rank	String							なし	
3	mainItem	String							なし	
4	capacity	String							なし	
5	note	String							なし	
6	desc	String							なし	

ヘルプ フィールドを取得(G) 最小桁数にする(M) OK(O) キャンセル(C)

技術資料, etl, pentaho, xml, csv, xpath

From:

<https://wiki.hgotoh.jp/> - 努力したWiki

Permanent link:

<https://wiki.hgotoh.jp/documents/etl/pentaho/xml/pentaho-005>

Last update: 2025/11/20 08:53

