

# XML子要素同士の結合と出力例3

2024-04-16

1つのXML内の階層が違うノード要素を組み合わせる例

フロー全体の動きは[XML子要素同士の結合と出力例1](#)を参照。

## 希望の結果

XML内のfactoryノードの抽出条件をもっと厳しくしたい。

Get data from XML 1で行っている以下の指定は「ノード名が“factory”から始まる」の意なので、これを「ノード名が“factory”から始まり英大文字1桁+数字3桁かつ最後の数字が1のノード」を対象にしたい。

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

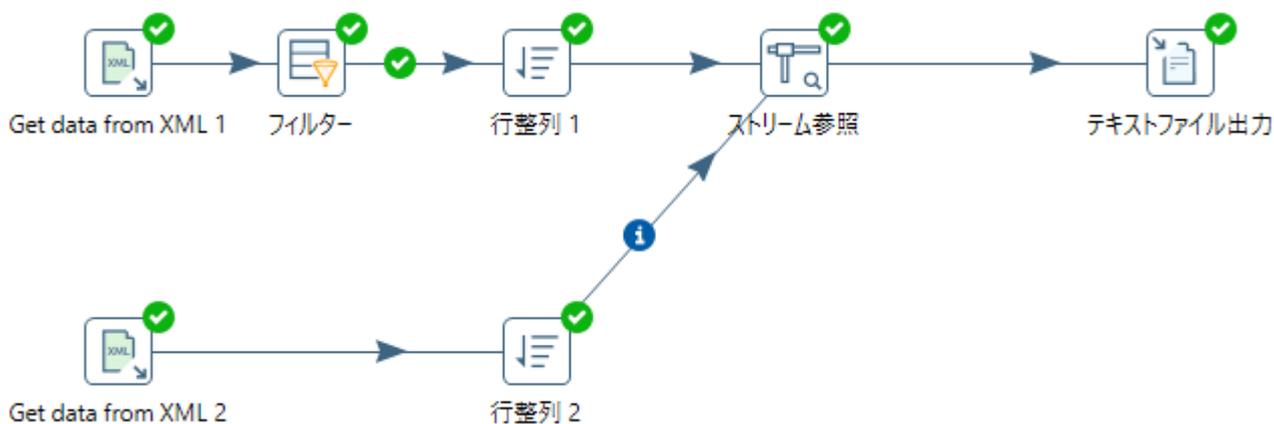
PentahoのXPathは正規表現を受け付けてくれないようなので、別の方法でフィルタする必要がある。

<b>入力XML1</b>
<pre>report-IN1.xml &lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;info&gt;   &lt;TITLE&gt;これはサンプルです1&lt;/TITLE&gt;   &lt;factories&gt;     &lt;factoryA001&gt;&lt;rank&gt;A&lt;/rank&gt;&lt;mainItem&gt;P005&lt;/mainItem&gt;&lt;capacity&gt;2&lt;/capacity&gt;&lt;note&gt;精密加工&lt;/note&gt;&lt;/factoryA001&gt;     &lt;factoryB001&gt;&lt;rank&gt;C&lt;/rank&gt;&lt;mainItem&gt;S001&lt;/mainItem&gt;&lt;capacity&gt;2&lt;/capacity&gt;&lt;/factoryB001&gt;     &lt;factoryB010&gt;&lt;rank&gt;C&lt;/rank&gt;&lt;mainItem&gt;S020&lt;/mainItem&gt;&lt;capacity&gt;5&lt;/capacity&gt;&lt;note&gt;生産力に余力あり&lt;/note&gt;&lt;/factoryB010&gt;     &lt;factoryC002&gt;&lt;rank&gt;A+&lt;/rank&gt;&lt;note&gt;ベテラン多数&lt;/note&gt;&lt;mainItem&gt;P005&lt;/mainItem&gt;&lt;capacity&gt;1&lt;/capacity&gt;&lt;/factoryC002&gt;     &lt;depoD100&gt;&lt;note&gt;立地&lt;/note&gt;&lt;capacity&gt;3&lt;/capacity&gt;&lt;/depoD100&gt;     &lt;depoD101&gt;&lt;note&gt;倉庫に問題&lt;/note&gt;&lt;capacity&gt;1&lt;/capacity&gt;&lt;/depoD101&gt;     &lt;depoE201&gt;&lt;note&gt;ピックに問題&lt;/note&gt;&lt;capacity&gt;2&lt;/capacity&gt;&lt;/depoE201&gt;   &lt;/factories&gt;   &lt;items&gt;     &lt;item name="P001" desc="P型" type="C" /&gt;     &lt;item name="P005" desc="P型" type="C" /&gt;     &lt;item name="S001" desc="S型" type="X" /&gt;     &lt;item name="S020" desc="S型赤" type="X" /&gt;     &lt;item name="F001R" desc="F型復刻" type="J" /&gt;   &lt;/items&gt; &lt;/info&gt;</pre>
<b>入力XML2</b>

<p><b>入力XML1</b></p> <p><a href="#">report-IN2.xml</a></p> <pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;info&gt;   &lt;TITLE&gt;これはサンプルです2&lt;/TITLE&gt;   &lt;factories&gt;     &lt;factoryF001&gt;&lt;rank&gt;C&lt;/rank&gt;&lt;mainItem&gt;P002&lt;/mainItem&gt;&lt;capacity&gt;2&lt;/capacity&gt;&lt;/factoryF001&gt;     &lt;factoryG001&gt;&lt;rank&gt;C&lt;/rank&gt;&lt;mainItem&gt;S020&lt;/mainItem&gt;&lt;capacity&gt;3&lt;/capacity&gt;&lt;/factoryG001&gt;     &lt;factoryG010&gt;&lt;rank&gt;D&lt;/rank&gt;&lt;mainItem&gt;S030&lt;/mainItem&gt;&lt;capacity&gt;2&lt;/capacity&gt;&lt;note&gt;設備老朽化&lt;/note&gt;&lt;/factoryG010&gt;     &lt;depoA100&gt;&lt;note&gt;立地&lt;/note&gt;&lt;capacity&gt;3&lt;/capacity&gt;&lt;/depoA100&gt;     &lt;depoC101&gt;&lt;note&gt;倉庫に問題&lt;/note&gt;&lt;capacity&gt;1&lt;/capacity&gt;&lt;/depoC101&gt;     &lt;depoJ201&gt;&lt;note&gt;ピックに問題&lt;/note&gt;&lt;capacity&gt;2&lt;/capacity&gt;&lt;/depoJ201&gt;   &lt;/factories&gt;   &lt;items&gt;     &lt;item name="P002" desc="P型" type="C" /&gt;     &lt;item name="S030" desc="S型青" type="X" /&gt;     &lt;item name="S020" desc="S型赤" type="X" /&gt;     &lt;item name="J001" desc="J型" type="J" /&gt;   &lt;/items&gt; &lt;/info&gt;</pre>
<p><b>出力CSV</b></p> <p><a href="#">report-OUT.csv</a></p> <pre>factory,rank,mainItem,capacity,note,desc factoryF001,C,P002,2,,P型 factoryA001,A,P005,2,精密加工,P型 factoryB001,C,S001,2,,S型 factoryG001,C,S020,3,,S型赤</pre>

## 定義

## フロー



XML子要素同士<sup>1</sup>の結合と出力例1との差分について説明する。

## フィルター

Get data from XML 1でフィルターしきれなかった内容を補完する。  
 “factory”で始まる名前が項目factoryに格納されてくるので、改めて正規表現で名称を指定し、合致しないレコードを除外する。

フィルター

ステップ名 フィルター

条件式が真 (True) のときに実行する処理 行整列 1

条件式が偽 (False) のときに実行する処

フィルタ条件

factory REGEXP factory[A-Z]\d\d1 (String)

ヘルプ OK(O) キャンセル(C)

この例の意味するところは「項目factoryの値が “ factoryXnn1”の形式[Xは英大文字でnnは00～99であれば処理する」の意となる。

[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-007>

Last update: **2025/11/20 08:53**

