

KETTLE_XML_MISSING_TAG_YIELDS_NULL_VALUEの件

KETTLE_XML_MISSING_TAG_YIELDS_NULL_VALUEをYにしていると、ステップ「XMLデータ取得(Get data from XML)」でXPathに関数を使えない。

Pentaho 9.4 でも Pentaho 10.2でも再現する。

音声解説

Google NotebookLMで音声解説を作ってもらいました。動画はまだ英語のみなんだけど近日中に他言語にも対応するとの事。

[Pentahoの謎パラメータ「KETTLEXMLMISSINGTAGYIELDSNULLVALUE」徹底解説「XPath関数が動かなくなる真の理由と賢い解決策](#)

発端

開発環境で作った.ktr定義を本番に近い環境でテストしたら「XMLの入力だけエラーになる。みんなで頭抱えてたところ、同僚氏が執念に近い献身で差異調査を行って件のパラメータの設定で発生する事を見つけた。いやほんとよく見つけたよ。

パラメータ整理

この事象の始まりは、データベースでの空データとnullの扱いを調整するためのパラメータ KETTLE_EMPTY_STRING_DIFFERS_FROM_NULL の追加をした事。

<https://github.com/pentaho/pentaho-kettle/commit/6d0e66de465e25f2623614e2e9971f42be3673bc>

データベース以外にXMLでも要望があったのか#3,#4の専用パラメータが新設された「KETTLE_XML_MISSING_TAG_YIELDS_NULL_VALUE」もそのうちの一つ。

#	プロパティ	説明	デフォルト	設定値等
1	KETTLE_EMPTY_STRING_DIFFERS_FROM_NULL	Pentaho全体でのnull値扱い指定	N	デフォルトはnullとemptyを区別しない Yを指定するとnullとemptyを区別する
2	KETTLE_DO_NOT_NORMALIZE_NULL_STRING_TO_EMPTY	nullの文字列をemptyに変換するか否か	N	デフォルトはnullをemptyへ変換する Yを指定するとnullをemptyへ変換しない

#	プロパティ	説明	デフォルト	設定値等
3	KETTLE_XML_EMPTY_TAG_YIELDS_EMPTY_VALUE	空タグ時、値にnullを生成するか否か	N	デフォルトはnullを生成する Yを指定するとnullを生成しない(empty)
4	KETTLE_XML_MISSING_TAG_YIELDS_NULL_VALUE	指定タグが無い場合nullを生成するか否か	N	デフォルトはemptyを生成する Yを指定するとnullを生成する

empty : ISBLANK()=Trueになる値

推理

<https://github.com/pentaho/pentaho-kettle/blob/5441e5bbc35c1e2a02b6cf36db01cb5a2e0277f2/plugins/xml/core/src/main/java/org/pentaho/di/trans/steps/getxmldata/GetXMLData.java>

KETTLE_XML_MISSING_TAG_YIELDS_NULL_VALUE=Yで事が起こるのはXPathの関数をノード(タグ)だと思って探しに行く処理になっているのではないかと。

このパラメタ有効時は、すべてノード(タグ)だと思って処理しているもしくは関数が来ることを想定していない

GitHubリポジトリのコードが現行と一致すると仮定する。

KETTLE_XML_MISSING_TAG_YIELDS_NULL_VALUE=Yの時はSelectSingleNodeメソッドを使うようだけど、ノード検索メソッドならXPath関数は当然ヒットしない。

ValueOfメソッドはXPath関数の評価まで面倒を見せてくれているんじゃないかな。

```
pentaho-kettle / plugins / xml / core / src / main / java / org / pentaho / di / trans / steps / getxmldata / GetXMLData.java ↑ Top
Code Blame 1033 lines (926 loc) · 37.8 KB · 🔍
Raw 📄 📄 📄 📄 📄 📄
720 private Object[] processPutRow( Node node ) throws KettleException {
721
722     Boolean xmlMissingTagYieldsNullValue = convertStringToBoolean(
723     Const.NVL( System.getProperty( Const.KETTLE_XML_MISSING_TAG_YIELDS_NULL_VALUE, "N" ), "N" ) );
724
725     // Handle namespaces
726     if ( meta.isNamespaceAware() ) {
727         XPath xpathField = node.createXPath( addNSPrefix( XPathValue, data.PathValue ) );
728         xpathField.setNamespaceURIs( data.NAMESPACE );
729         if ( xmlDataField.getResultType() == GetXMLDataField.RESULT_TYPE_VALUE_OF ) {
730             if ( xmlMissingTagYieldsNullValue ) {
731                 nodevalue = xpathField.selectSingleNode( node ) != null ? xpathField.valueOf( node ) : null;
732             } else {
733                 nodevalue = xpathField.valueOf( node );
734             }
735         } else {
736             Node n = xpathField.selectSingleNode( node );
737             if ( n != null ) {
738                 nodevalue = n.asXML();
739             } else {
740                 nodevalue = xmlMissingTagYieldsNullValue ? null : "";
741             }
742         }
743     } else {
744         if ( xmlDataField.getResultType() == GetXMLDataField.RESULT_TYPE_VALUE_OF ) {
745             if ( xmlMissingTagYieldsNullValue ) {
746                 nodevalue = node.selectSingleNode( XPathValue ) != null ? node.valueOf( XPathValue ) : null;
747             } else {
748                 nodevalue = node.valueOf( XPathValue );
749             }
750         } else {
751             Node n = node.selectSingleNode( XPathValue );
752             if ( n != null ) {
753                 nodevalue = n.asXML();
754             } else {
755                 nodevalue = xmlMissingTagYieldsNullValue ? null : "";
756             }
757         }
758     }
759
760     // Do trimming
```

追加検証

うるさい人が湧いたので実際にコード書いて確かめよう。

検証用のコードはこちら。

[xpathCheck.java](#)

```
package xpathCheck;

import org.dom4j.Document;
import org.dom4j.io.SAXReader;

public class XPathCheck {

    public static void main(String[] args) {

        boolean xmlMissingTagYieldsNullValue = false; //
        KETTLE_XML_MISSING_TAG_YIELDS_NULL_VALUE=N を模す デフォルト

        String filePath = "D:\\pleiades\\sampledata\\Sample.xml";
```

```
String xpathExpression = "count(/root/employees/employee)";

SAXReader rdr = new SAXReader() ;
String nodeValue;

try {
    Document doc = rdr.read(filePath);

    // KETTLE_XML_MISSING_TAG_YIELDS_NULL_VALUE 問題処理の再
    if(xmlMissingTagYieldsNullValue)
    {
        nodeValue = doc.selectSingleNode(xpathExpression)
!= null ? doc.valueOf(xpathExpression) : null;
    }
    else
    {
        nodeValue = doc.valueOf(xpathExpression);
    }

    System.out.println("XPath: " + xpathExpression);
    System.out.println("Node Value: " + nodeValue);

} catch (Exception e) {
    e.printStackTrace();
}

}
```

入力データはこちら。

[Sample.xml](#)

```
<root>

  <employees>

    <employee>
      <name>hoge</name>
      <desc>hoge</desc>
    </employee>

    <employee>
      <name>fuga</name>
      <desc>fuga</desc>
    </employee>

  </employees>
```

```
</root>
```

まずデフォルト時(Nが指定された状態)での実行結果。

```
コンソール ×
<終了> xpathCheck [Java アプリケーション] D:\pleiades\2024-12\java\17\bin\javaw.exe (2025/08/14 18:32:00 - 18:32:01 elapsed: 0:00:00.537) [pid: 30900]
XPath: count(/root/employees/employee)
Node Value: 2
```

さて次はオプション指定(Yが指定された状態)状態にする。この個所をfalse→trueに変更して実行。

```
boolean xmlMissingTagYieldsNullValue = true; //
KETTLE_XML_MISSING_TAG_YIELDS_NULL_VALUE=Y を模す オプション指定状態
```

実行結果。見た覚えのあるエラーが出てますでしょ？

```
コンソール ×
<終了> xpathCheck [Java アプリケーション] D:\pleiades\2024-12\java\17\bin\javaw.exe (2025/08/14 18:38:07 - 18:38:08 elapsed: 0:00:01.825) [pid: 5944]
org.dom4j.XPathException: Exception occurred evaluating XPath: The result of the XPath expression is not a Node. It was: 2.0 of type: java.lang.Double
    at org.dom4j.xpath.DefaultXPath.selectSingleNode(DefaultXPath.java:171)
    at org.dom4j.tree.AbstractNode.selectSingleNode(AbstractNode.java:180)
    at xpathCheck.XPathCheck.main(XPathCheck.java:24)
```

念のため、関数ではなく存在しないノードの検索にしてみましょうか。

```
String xpathExpression = "/root/employees/employee/age";
```

実行結果。修正者は関数の時もこの状態が来るとしてあのコードにしたんでしょうね。やっぱりテストケース不足です。

```
コンソール ×
<終了> xpathCheck [Java アプリケーション] D:\pleiades\2024-12\java\17\bin\javaw.exe (2025/08/14 18:56:20 - 18:56:21 elapsed: 0:00:00.639) [pid: 40944]
XPath: /root/employees/employee/age
Node Value: null
```

暫定修正

ぶっちゃけ、下手にY or Nの条件分岐なんてせずにvalueOf()の処理結果とオプション指定を見て、emptyをnullに変えりゃよかったんですよ。

xpathCheck2.java

```
package xpathCheck;

import org.dom4j.Document;
import org.dom4j.io.SAXReader;

public class XPathCheck {

    public static void main(String[] args) {

        boolean xmlMissingTagYieldsNullValue = true; //
        KETTLE_XML_MISSING_TAG_YIELDS_NULL_VALUE=Y を模す

        String filePath = "D:\\pleiades\\sampledata\\Sample.xml";
```

```
String xpathExpression = "count(/root/employees/employee)";

SAXReader rdr = new SAXReader() ;
String nodeValue;

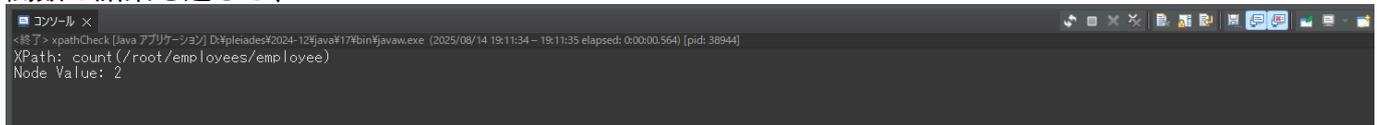
try {
    Document doc = rdr.read(filePath);

    // 暫定的な修正例
    nodeValue = doc.valueOf(xpathExpression);
    if((xmlMissingTagYieldsNullValue)&&(nodeValue.length() == 0)) nodeValue
    = null;

    System.out.println("XPath: " + xpathExpression);
    System.out.println("Node Value: " + nodeValue);

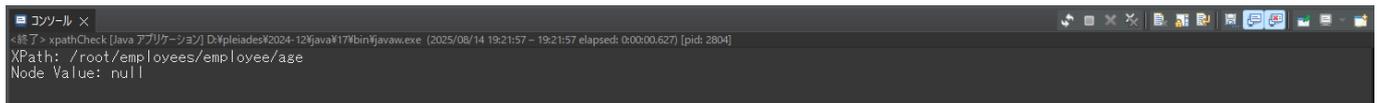
} catch (Exception e) {
    e.printStackTrace();
}
}
```

関数の結果も返るし、



存在しないノードの結果はnullになる。

```
String xpathExpression = "/root/employees/employee/age";
```



From: <https://wiki.hgotoh.jp/> - 努力したWiki

Permanent link: <https://wiki.hgotoh.jp/documents/etl/pentaho/pentaho-012>

Last update: 2025/08/19 02:05

