



Technical Note 04-21

XML Utilities

By David Adams
Technical Note 04-21

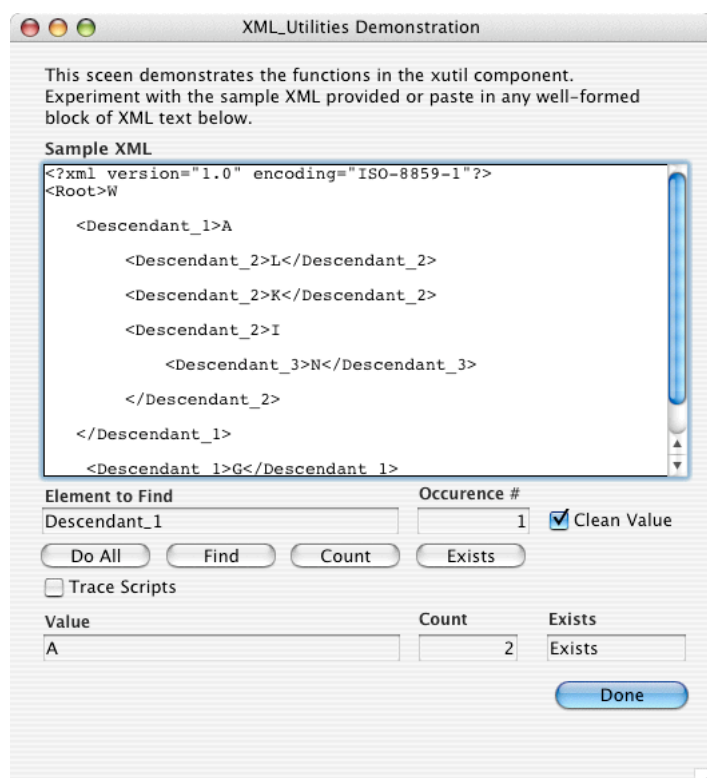
(原題: The XML_Utilities Component)

概要

4D 2003 の XML コマンドは、XML の解析やバリデート、XML ツリー内のノード移動や要素/属性の読み取りなどをサポートしています。この Tech Note では、4D のネイティブコマンドを拡張し、幾つかの便利な機能を追加するコンポーネントを紹介しています。

メソッド名	説明
xutil_CountOccurrences	特定の要素が出現する回数を返す
xutil_ElementExists	要素が存在すれば True を返す
xutil_GetValue	テキストなどの単純な要素値を返す
xutil_StripWhitespace	テキストからスペースを取り除く

XML コマンドの参考例としても活用できるように、コンポーネントだけでなくソースコードおよび例題が含まれています。

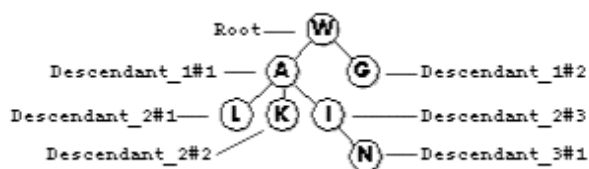


XML: 基本

- XML 要素の参照は、16 バイトの文字列で、4D の XML コマンドと共に使用されます。
 - 使用が済んだ XML ツリーは CLOSE XML コマンドでメモリを解放するようにします。
- なお XML_Uutilities コンポーネントのメソッドは CLOSE XML をコールしていません。
- 4D とは違い、XML では大文字と小文字を区別します。
 - XML_Uutilities コンポーネントのメソッドは 4D のワイルド@を文字とみなします。

4D の XML コマンドは XML を解析してツリーを構築します。XML ツリー内で移動する方法および方向には幾つかありますが、XML_Uutilities コンポーネントは上から下、左から右へノードを進むようになっています。アルゴリズムの本などで preorder traversal と呼ばれている方法です。サンプル XML とツリーの図を次に示します。

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Root>W
  <Descendant_1>A
    <Descendant_2>L</Descendant_2>
    <Descendant_2>K</Descendant_2>
    <Descendant_2>I
      <Descendant_3>N</Descendant_3>
    </Descendant_2>
  </Descendant_1>
  <Descendant_1>G</Descendant_1>
</Root>
```



XML_Uutilities コンポーネントが移動する順番およびノードの出現回数は次のとおりです。

ノード名	値	何回目?
Root	W	1
Descendant_1	A	1
Descendant_2	L	1
Descendant_2	K	2
Descendant_2	I	3
Descendant_3	N	1
Descendant_1	G	2

xutil_CountOccurrences

xutil_CountOccurrences (elementRef;targetName{;firstCall}) -> Longint

パラメータ	タイプ	説明
elementRef	文字(16)	始点となる要素参照
targetName	テキスト	探すべき要素名
firstCall	ブール	内部処理用(省略可)
count	倍長整数	要素の数

注記:

このメソッドは自らをコールすることによってすべてのノードを調べています。自らをコールすると内部的に固有のパラメータを持つメソッドコールのスタックが構築されます。ノードを調べる別の方法としては、ループの使用、配列をスタックとして利用する方法などが挙げられます。

xutil_ElementExists

xutil_ElementExists (elementRef;targetName) -> Boolean

パラメータ	タイプ	説明
elementRef	文字(16)	始点となる要素参照
targetName	テキスト	探すべき要素名
exists	ブール	要素があれば True、なければ False

xutil_GetValue

xutil_GetValue (elementRef;targetName{;index{;clean{;firstCall}}}) -> Text

パラメータ	タイプ	説明
elementRef	文字(16)	始点となる要素参照
targetName	テキスト	探すべき要素名
index	倍長整数	何番目の要素から値を読み取るかを指定
clean	ブール	空白を除去する=True
firstCall	ブール	内部処理用(省略可)
value	テキスト	合致する要素の値あるいは空の文字列

特定の要素名から値を返します。任意の引数 index を使用すれば、特定の要素の特定の出現箇所から値を読み取る事ができます。デフォルトでは 1 番目の要素が対象になります。デフォルトが False である任意の引数 clean は、値の前後にある空白を除去したい場合に使用します。

xutil_StripWhitespace

xutil_StripWhitespace (source) -> Text

パラメータ	タイプ	説明
source	テキスト	元のテキスト
result	テキスト	空白を除去されたテキスト

XML では以下の文字がホワイトスペースとして定義されています。

Tab	0x09
Line feed	0xA
Carriage return	0xD
Space	0x20

ホワイトスペースは XML を読みやすくする反面、データとしては不要な場合があります。たとえば、前述のサンプル XML の場合、Descendant_1 の最初のインスタンスの値は 16 文字長ですが、その実体は次のようなものです。

```
A<carriage_return> <carriage_return> <tab> <tab> <carriage_return> <carriage_return>
<tab> <tab> <carriage_return> <carriage_return> <tab> <tab> <carriage_return>
<carriage_return> <tab>
```

空白を除去すれば、その値は A だけとなります。

xutilStringsAreEqual

コンポーネントに含まれているおまけのメソッドです。4D の文字列コマンドは、言語設定などによって大文字と小文字を区別しなかったり、発音記号を無視したりします。次の表は 4D および XML の基準で文字列を比較した場合の結果の違いを表わしています。

文字列 A	文字列 B	A=B?(4D)	A=B?(XML)
abcde	abcde	yes	yes
abcde	abcdE	yes	no
abcde	abcdé	yes	no
abcde	ABCDE	yes	no

xutilStringsAreEqual メソッドは ASCII コードを 1 バイトずつ調べて文字列を比較します。