

Getting an XML Element's Full Path

By David Adams

Technical Note 07-03

Abstract

このテクニカルノートでは、4th DimensionのDOMコマンドを使用して、XMLノードのフルパスを構築する方法を示します。またサンプルデータベースには実装済みのコードが含まれています。

Overview

XMLは、システム間のデータ交換や、プログラム設定およびその他データの格納フォーマットなど、データフォーマットのスキームとして、たいへん身近なものとなりました。XMLを使用する際の難関のひとつは、情報を階層的に格納し体系化するという、XMLの最大の機能の中にあります。例として、以下の短いXMLを見てみましょう：

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<contacts>
  <contact>
    <name>Dan West</name>
    <business>
      <phone>
        <area_code>04</area_code>
        <number>0035-9110</number>
      </phone>
    </business>
    <home>
      <phone>
        <area_code>02</area_code>
        <number>6493-3250</number>
      </phone>
    </home>
  </contact>
</contacts>
```

この例では、以下の要素が2回ずつ登場しています：

```
<phone>
<area_code>
<number>
```

これらの要素は<business>と<home>要素の子要素として登場します。XMLドキュメントを処理する際、要素名が繰り返し現れるのは普通のことです。このような場合、ノードへのフルパスを取得することで、それらを区別することができるようになります。例えば<area_code>への2つのインスタンスへのパスは、以下のようになります：

```
/contacts/contact/business/phone/area_code/
/contacts/contact/home/phone/area_code/
```

このテクニカルノートでは、ノードへのフルパスを取得する方法を説明しています。またコードが実装されたサンプルデータベースを含んでいます。

Review: Native DOM Commands

4th Dimensionにはすでに、パスを使用するDOM (Document Object Model) コマンドが含まれています。例えば上に示した2つの<area_code>要素にアクセスすることは簡単で、以下のように行います:

```
C_STRING($16;$businessPhone_xmlref)
C_STRING($16;$homePhone_xmlref)

$businessPhone_xmlref:=DOM Find XML element($root_xmlref;
"/contacts/contact/business/phone/area_code/")

$homePhone_xmlref:=DOM Find XML element($root_xmlref;
"/contacts/contact/home/phone/area_code/")
```

DOM Find XML element 関数は簡単・有効ですが、事前にフルパスが分かっていないと使用することができません。このコマンドはほとんどのXML処理で使用することができますが、すべてではありません。例えば:

■ XMLは一般に木構造を上から下にたどっていく方法で検査されます。このテクニックは既存のテクニカルノートやサンプルデータベース、および*The 4D Web Services Companion*で使用されています。この方法を使用した場合、フルパスを最初に知ることができません。代わりに、コードは順番にそれぞれのノードを処理していきます。**DOM Find XML element**関数は、木構造のトップから下る方向で処理を行います。対して一般の処理は、ノードから登る方向で木構造をたどります。

■ **DOM Find XML element**関数のパスシンタックスは、すべてのマッチパターンを表現することはできません。例えば、この関数は/phone/area_code/で終了するパスといった条件を使用することができません。ドキュメントルートやフルパスがどうかにかかわらず、ノードや木構造の連続した一部分に基づいて処理を行うと便利であったり、それが必要な場合があります。

幸いなことに、他のコマンドを使用して、簡単にフルパスを生成する関数を記述することができます。

Note XML パス、要素名、および属性名は大文字小文字を区別します。このような場合の文字列比較については*4D Technical Note 05-41, Case-Sensitive Operations in 4th Dimension*を参照してください。

The Routines

DOM_GetFullPathルーチンのコードとエラーハンドラを以下に示します。

DOM_GetFullPath

以下のコードは、カレントノードのフルパスを返します。無効なノードやヌルを渡すと、空の文字が返されます。

```
C_TEXT($0;$fullPath_text)
C_STRING(16;$1;$noderef) `<-- The calling routine must pass in this parameter.

$noderef:=$1
$fullPath_text:=""

` Store existing error/error handling state.
If (Undefined(Error))
    Error:=0
End if

C_LONGINT($previousValueOfErrorVariable_I)
$previousValueOfErrorVariable_I:=Error

C_STRING(31;$previousErrorMethodName_s)
$previousErrorMethodName_s:=Method called on error

ON ERR CALL("DOM_ErrorTrappingRoutine")

C_TEXT($name_text)
$name_text:=""
DOM GET XML ELEMENT NAME($noderef;$name_text) ` Default to starting element name.

While (OK=1)
    Case of
        ¥ ($name_text="#document") ` The #document node is a synthetic node above the tree.
            OK:=0 ` Get out of routine.

        ¥ ($name_text="") ` Bad reference.
            OK:=0 ` Get out of routine.

        Else ` Valid name.
            $fullPath_text:= "/" + $name_text + $fullPath_text
            OK:=1 ` Continue processing.
            $noderef:=DOM Get parent XML element($noderef;$name_text) ` Try to get a parent.

    End case
End while

` Restore previous error/error handling state.
Error:=$previousValueOfErrorVariable_I ` Restore original error value.
ON ERR CALL($previousErrorMethodName_s) ` Restore original error handler.

If ($fullPath_text# "") ` A path was built.
    $fullPath_text:=$fullPath_text + "/" ` Add final slash.
End if

` Return result.
$0:=$fullPath_text
```

DOM_ErrorTrappingRoutine

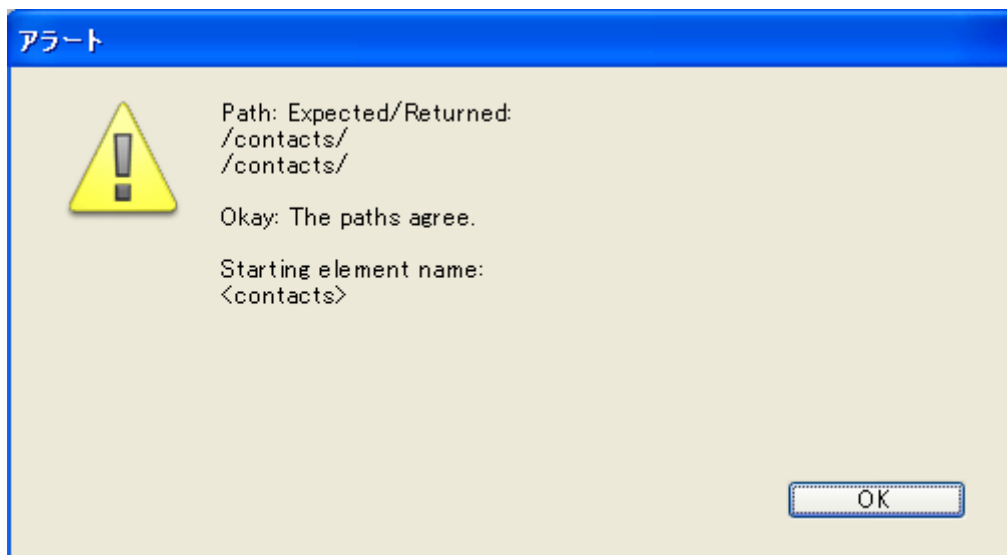
*DOM_ErrorTrappingRoutine*は*DOM_GetFullPath*によってインストールされ、無効なノードを参照することによって生成されるエラーをトラップします。エラーハンドラは以下のようなシンプルなコードです:

```
DOM_Error:=Error
```

The Sample Database

サンプルデータベースには、先に説明したコードと、単純なテストルーチン

*Test_GetFullPath*が含まれています。テストコードは有効なノードや無効なノードを含むさまざまな条件で*DOM_GetFullPath*を実行するようデザインされています。テストごとに、コードは結果を以下のようなダイアログに表示します:



Summary

4th DimensionのDOM XMLコマンドには、特定のXMLノードへのフルパスを返す関数が含まれていません。しかしながら、この関数を、**DOM Get parent XML element** コマンドを使用して簡単に実装することができます。サンプルデータベースには *OM_GetFullPath* という名前の、フルパスを返すメソッドが含まれています。