

Getting an XML Element's Depth

By David Adams

Technical Note 07-01

Abstract

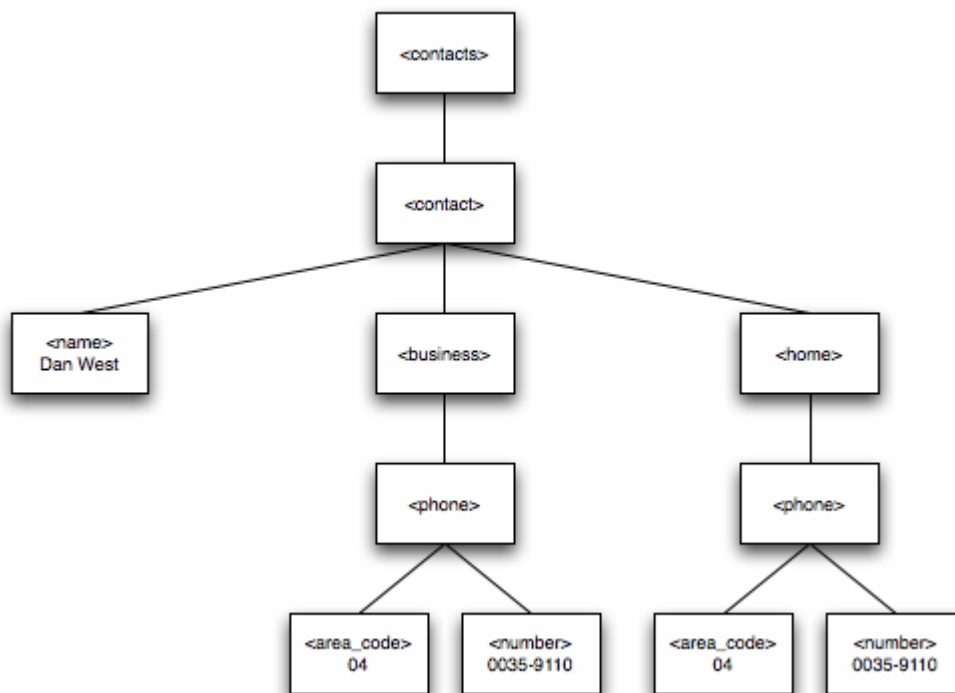
4th DimensionのDOMコマンドにはノードのレベルを返すコマンドが用意されていません。しかしこの機能は**DOM Get parent XML element** コマンドを使用して簡単に実装することができます。このテクニカルノートでは必要なコードを説明し、また実装済みのサンプルデータベースを提供します。

Overview

DOM (Document Object Model) コマンドを使用してXMLドキュメントを読み込むと、XMLは木構造として扱われます。例えば以下の短い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>
```

木構造として解釈されると、ノードは以下に示すようにリンクされます：



木構造内のそれぞれのノードはレベルを持ちます。例えば<contacts>要素は1で、<contact>要素は2、<business>要素は3です。XMLの木構造やノードを処理する際、ノードのレベルを知っていると便利な場合があります。しかし4th DimensionのDOMコマンドにはノードのレベルを返す関数が含まれていません。幸いなことにこの機能は**DOM Get parent XML element**コマンドを使用して容易に追加することが可能です。このテクニカルノートでは必要なコードを説明し、また実装済みのサンプルデータベースを提供します。

Uses for Retrieving the Current Node's Level

DOMによるXML要素のレベルを取得することは、そのレベル深度により特別な動作を行う必要がある場合に必要となります。例として、XMLの木構造を階層リストにコピーするメソッドでは、深度毎にリスト項目のフォーマットを設定することができます。以下の例ではレベル2と4のノードにボールド書体を設定し、項目を括弧で囲っています：

```
contacts
├── [contact]
│   ├── name
│   ├── business
│   │   ├── [phone]
│   │   │   ├── area_code
│   │   │   └── number
│   └── home
│       ├── [phone]
│       │   ├── area_code
│       │   └── number
```

XMLの構造や必要な処理により、他にもノードのレベル取得が必要な場合があります:

- ・ 埋め込む値を保持するために新しいレコードを追加する場合。例えば[Contact]や[Phone]レコードを<contact>や<phone>要素に埋め込む場合など。
- ・ レベルに基づき要素を無視したい場合。上の例ではレベル1の<contacts>要素を無視したい場合など。
- ・ XMLのテキスト表示をフォーマットする場合。例えば要素のレベルを示すためにTabを挿入する場合など。

The Routines

DOM_GetNodeLevelコードとエラーハンドラは以下のようになります。

DOM_GetNodeLevel

```
C_LONGINT($0;$nodeLevel_count)
C_STRING(16;$1;$noderef)

$noderef:=$1
$nodeLevel_count:=0

` 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($currentNodeName_text)
$currentNodeName_text:=""

` Default to starting element name.
DOM GET XML ELEMENT NAME($noderef;$currentNodeName_text)
While (OK=1)
    Case of
        ¥ ($currentNodeName_text="")
            OK:=0 ` We're done scanning.

        ¥ ($currentNodeName_text="#document")
            ` An artificial node above the tree. We're not treating this as a valid ancestor.
            OK:=0 ` We're done scanning.
    Else
        $nodeLevel_count:=$nodeLevel_count+1 ` Save this value as it may be the last parent.
        ` Try to get another parent.
        $noderef:=DOM Get parent XML element($noderef;$currentNodeName_text)
    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.
```

```
$0:=$nodeLevel_count
```

DOM_ErrorTrappingRoutine

DOM_ErrorTrappingRoutine は**DOM_GetNodeLevel** によってインストールされ、無効なノード読み込みを行おうとしたときのエラーをトラップします。エラーハンドラは以下の一行が記述された簡単なものです:

```
OM_Error:=Error
```

The Sample Database

サンプルデータベースには上に示したコードと簡単なテストルーチン

Test_GetNodeLevelが含まれています。テストコードは**DOM_GetNodeLevel**をさまざまな条件で試すようデザインされていて、正当なノードや不正なノード、**#document** ノードなどをテストします。それぞれのテストにおいて、テスト結果をダイアログに表示します:



Summary

DOMを処理する際、XML要素の深度レベルを知っていると便利な場合があります。このテクニカルノートではDOM Get parent XML elementコマンドを使用してこの動作を実装する方法を示しました。またこのメソッドは添付のサンプルデータベースに実装されています。