

平成 21 年 12 月 28 日
気 象 庁

「気象庁防災情報 XML フォーマット」スキーマとサンプル電文の動作検証について

平成 21 年 12 月 28 日付「気象庁防災情報 XML フォーマット」スキーマとサンプル電文の修正・追加提供について、XML コンソーシアムのご協力により、各種 XML 関連ソフトウェアにおける動作状況について検証頂きました。同仕様をご利用いただく際の参考とすべく、検証結果を公開いたします。

【要旨】

各種ベンダー系、及びフリー系ソフトウェアにおいて、スキーマの読み込みとサンプル電文の妥当性検証、その他データバインディング等の検証において、良好な動作結果をご報告頂きました。

なお、一部フリー系ソフトウェアでは union 要素の処理で問題が発生しましたが、これについては回避方法も併せてご報告頂きました。

【検証内容と結果】

検証対象であるソフトウェアと動作環境、及び検証内容とその結果については別紙のとおりです。

なお、本検証結果については検証条件における動作結果であり、気象庁並びに各ソフトウェア開発元において、「気象庁防災情報 XML フォーマット」仕様の動作保証、及び完全対応を表明するものではありません。

【謝辞】

検証作業においては、XML コンソーシアムに多大なるご協力を頂きました。ここに感謝申し上げます。

各ソフトウェアによる動作検証の詳細結果について

対象ソフト	
動作環境	
検証内容	結果
*Oracle Database 11g Release 2 (11.2.0.1) for Linux x86	
1 XML スキーマ検証 全てのXMLスキーマをデータベースに登録する。	正常に完了した(OK)。
2 XML インスタンス検証 全てのサンプルXMLインスタンスをデータベース表 (バイナリXMLを使用) に格納する。また、XMLスキーマに反するXMLインスタンスをデータベース表 (バイナリXMLを使用) に格納する。以下の3パターンで動作確認。	
a) バイナリXMLは、BasicFiles と SecureFiles の両方で確認。	全てのサンプルXMLインスタンスについて、正常に完了した(OK)。 また、XMLスキーマに反するXMLインスタンスについて、エラーとなった(正常な動作:OK)。
b) SecureFiles については、圧縮 (Compress High)も確認。	全てのサンプルXMLインスタンスについて、正常に完了した(OK)。 また、XMLスキーマに反するXMLインスタンスについて、エラーとなった(正常な動作:OK)。
c)BasicFiles と SecureFiles の両方について、パーティション表へのINSERTも確認。	全てのサンプルXMLインスタンスについて、正常に完了した(OK)。 また、XMLスキーマに反するXMLインスタンスについて、エラーとなった(正常な動作:OK)。
3 XML データ処理 データベースに登録されたXMLインスタンスに対して以下の処理・操作を実施する。	
a) 特定のタグ・属性に対する索引の作成 XPathで指定した特定ノードに対してB*Tree索引およびビットマップ索引を作成できることを確認する。また、検索処理で索引が有効であることを確認する。	索引が作成できることを確認した。また、検索処理で索引が有効であることを確認した(OK)。
b) 特定のタグ以下に対する索引の作成 XPathで指定した要素以下の全てのノードに対して、タグ構造に基づく索引を作成できることを確認する (XMLIndex索引のパス・サブセットを利用)。また、検索処理で索引が有効であることを確認する。	索引が作成できることを確認した。また、検索処理で索引が有効であることを確認した(OK)。
c) 全体に対する索引の作成 以下の2通りの方法で実施し、文書全体に対して、タグ構造に基づく索引を作成できることを確認する。 また、検索処理で索引が有効であることを確認する。 1) XML索引(XMLIndex 索引) 2) 全文検索索引(Oracle Text の CONTEXT 索引 / PATH_SECTION_GROUP を利用)	索引が作成できることを確認した。また、検索処理で索引が有効であることを確認した(OK)。
d) SQL/XML関数による検索 SQL/XMLによって (extractValue を利用)、特定ノードの値を取得できることを確認する。	値が取得できることを確認した(OK)。
e) XQueryによる検索 サンプルXMLインスタンスそれぞれに対して、FLOWR構文によるXQueryを実行し(xmlQuery 関数を利用)、結果が正しく返されることを確認する。	値が取得できることを確認した(OK)。
f) XMLインスタンスの部分更新 SQL/XMLによって (updateXML を利用)、XPathで指定された特定ノードを正しく更新できることを確認する。	特定ノードを正しく更新できることを確認した(OK)。

各ソフトウェアによる動作検証の詳細結果について

・Interstage Application Server Standard-J Edition V9.2.0	
・Windows XP Professional SP2 ・Java(TM) 2 Runtime Environment, Standard Edition (1.4.2_19) ・Java(TM) 2 Runtime Environment, Standard Edition (1.5.0_17) ・Java(TM) SE Runtime Environment (1.6.0_12)	
1 XMLスキーマ検証 Fujitsu XML Processor単体により、XMLスキーマを読み込んで検証を行う。	正常に完了した(OK)。
2 XMLインスタンス検証 Fujitsu XML Processor単体により、サンプルインスタンスをXMLスキーマによる妥当性検証を実施しながらの読み込みを行う。	正常に完了した(OK)。
・Interstage Application Server Standard-J Edition V9.2.0	
・Windows Server 2003 R2 Enterprise Edition SP2 ・Java 2 Runtime Environment, Standard Edition (1.5.0_17)	
XMLデータ処理	
a)スキーマを元にWebサービス定義(WSDL)を作成、Webサービスアプリケーションおよびクライアントを作成を行う。	正常に完了した(OK)。
b)上記検証により作成したWebサービスアプリ・クライアントによりサンプルインスタンスの送受を行う。	送信・受信データの同一性が確認され問題なく転送が完了した(OK)。

各ソフトウェアによる動作検証の詳細結果について

<ul style="list-style-type: none"> •DB2 Database for Linux, UNIX, and Windows Version 9.7 (Fix pack 1) •IBM Optim Development Studio 2.2 •WebSphere Application Server Community Edition V2.1.1 	
<ul style="list-style-type: none"> •Windows XP Professional SP2 	
<p>1 XML スキーマ検証 データベースにXML スキーマを登録し、XML スキーマの妥当性を検証する。</p>	正常に完了した(OK)。
<p>2 XML インスタンス検証 a)データベースにサンプルXML インスタンスを登録し検証する。その際にXML スキーマを使ったValidation を実施する。 b) XMLスキーマのxs:any要素部分を検証するため、当該部分に新しい要素を追加したサンプルXMLインスタンスをデータベースに登録し検証する。その際にXML スキーマを使ったValidation を実施する。 具体的には、サンプルXML“70_01_01_091210_kisetsu-1.xml”の</ObservationAddition>の後に以下を追加したインスタンスにて検証する。 <NewElement xmlns="http://xml.kishou.go.jp/jmaxml1/addition1/">新しい要素の追加</NewElement></p>	<p>サンプルXML を全てXML スキーマでValidation し、データベースに登録できることを確認した(OK)。</p> <p>XML スキーマでValidation し、データベースに登録できることを確認した(OK)。</p>
<p>3 XML データ処理 データベースに登録されたXML インスタンスに対して以下の処理・操作を実施する。</p>	
a) XML パターンに基づく索引作成	指定したXML パターンに基づく索引が作成できることを確認した(OK)。
b) 全文検索索引の作成	全文検索のための索引が作成できることを確認した(OK)。
c) XQuery による検索 試験用のXQuery でXML インスタンスの検索を実行する。	結果が得られることを確認した(OK)。
d) XQuery による検索(全文索引) 1つの文に“最低気温”と“平年並”を含む、もしくは“最低気温”と“高い”を含むXML インスタンスから、Bodyの部分の結果として返すようにしたXQueryにより全文検索を含む検索を実行する。	結果が得られることを確認した(OK)。
e) XQuery Update Facility によるXML インスタンスの部分更新 XML インスタンスのバージョンに関わる3つの部分を変更する部分更新のXQueryを実行する。	結果が得られることを確認した(OK)。
f) XSLT スタイルシートによるHTML への変換 HTML に変換するためのスタイルシート(XSLT)を作成し、SQL によるHTML 変換を実施する。	問題なく変換できることを確認した(OK)。
g) SOAP バインディングによるWeb サービスの作成と実行 SQL/XMLでタイトル(title)を入力パラメータとするWeb サービス化して実行する。	問題なく結果が得られることを確認した(OK)。
h) HTTP バインディングによるWeb サービスの作成と実行 SQL/XMLでタイトル(title)を入力パラメータとするWeb サービス化して実行する。	問題なく結果が得られることを確認した(OK)。
i) REST(HTTP GET バインディング)によるWeb サービスの作成と実行 SQL/XMLでタイトル(title)を入力パラメータとするWeb サービス化して実行する。	問題なく結果が得られることを確認した(OK)。

各ソフトウェアによる動作検証の詳細結果について

・uCosminexus Application Server Standard 8(Cosminexus XML Processor 8)	
・Red Hat Enterprise Linux ES release 4 (Nahant Update 5) ・Windows XP Professional SP2 ・Java 2 Platform Standard Edition Runtime Environment Version 5.0	
XML インスタンス検証 uCosminexus Application Server Standardに含まれるCosminexus XML Processor(XML文書の解析、操作、生成を行うミドルウェア)が提供するAPI(csmjaxp.jar)を利用して検証用プログラムを作成し、以下を検証する。	
a)全サンプルXMLデータ(206ヶ)に対してXMLスキーマ検証する。	全サンプルXMLデータに対する検証結果が正常と検知された(OK)。
b)全サンプルXMLデータ(206ヶ)に対してXMLスキーマ検証する。スキーマファイルの内容を一部(異常となるよう)変更し、検証する。	全サンプルXMLデータに対する検証結果が異常と検知された(正常な動作:OK)。
・HiRDB/Single Server 08-05 ・HiRDB XML Extension 08-05	
・Windows XP Professional SP2 ・Red Hat Enterprise Linux AS release 3	
1 XMLインスタンス検証 全サンプルXMLデータをDBに格納可能か確認する。	全サンプルXMLデータをINSERT文によりDBに格納できることを確認した(OK)。
2 XMLデータ処理 全サンプルXMLデータをDBに格納し、以下の4点のXMLデータ処理が可能か確認する。	
a) XMLSERIALIZE関数、XMLEXISTS述語、XMLQUERY関数を使用した検索処理が可能か。 ①/jmx:Report/jmx:Control/jmx:Titleが”季節観測”であるXMLデータを取り出す。 ②/jmx:Report/jmx:Control/jmx:Titleの要素のみを取り出す。	検索SQLを実行できることを確認した(OK)。
b) 部分構造インデクス(XMLデータ中の特定の要素や属性の値をキー値とするインデクス)を作成可能か。 ①/jmx:Report/jmx:Control/jmx:Title に部分構造インデクスを定義する。	部分構造インデクスを定義できることを確認した(OK)。
c) 部分構造インデクスを使用した検索が可能か。	(b)で定義した部分構造インデクスを使用した検索ができることを確認した
d) 全文検索インデクスを使用した検索が可能か。	全文検索インデクスを使用した検索ができることを確認した(OK)。
・.NET Framework 3.5	
・Windows 7 Enterprise 64-bit 日本語版	
XML インスタンス検証 スキーマを使った全インスタンスの妥当性検証を行う。	正常に完了した(OK)。

各ソフトウェアによる動作検証の詳細結果について

*Visual Basic 2008 Express Edition	
*Windows XP Professional SP3	
1 XML インスタンス検証 XmlValidatingReaderにより、スキーマを使った全インスタンスの妥当性検証を行う。	全てのサンプルXMLインスタンスについて、正常に完了した(OK)。 また、XMLスキーマに反するXMLインスタンスについて、エラーとなった(正常な動作:OK)。
2 XML データ処理 スキーマをimportして独自に作成したwsdlを使って、Webサービスを呼び出すコードを自動生成する。自動生成したコードを開発環境に取り込み、コンパイルできるか確認する。	正常に完了した(OK)。
*Visual C# 2008 Express Edition	
*Windows XP Professional SP3	
XML データ処理 スキーマをimportして独自に作成したwsdlを使って、Webサービスを呼び出すコードを自動生成する。自動生成したコードを開発環境に取り込み、コンパイルできるか確認する。	正常に完了した(OK)。
*libxml2-devel-2.7.3-1.fc10.i386、(xmlstarlet-1.0.1-6.fc9.i386)	
*Fedora release 10 (Cambridge) Linux 2.6.27.5-117.fc10.i686	
XML インスタンス検証 以下のコマンドにより、スキーマを使った全インスタンスの妥当性検証を行う。 xmlstarlet val -s jmx.xsd ファイル名.xml	全てのサンプルXMLインスタンスについて、正常に完了した(OK)。 また、XMLスキーマに反するXMLインスタンスについて、エラーとなった(正常な動作:OK)。
*xerces-c-src_2_8_0、gcc (GCC) 4.3.2 20081105 (Red Hat 4.3.2-7) *xerces-c-3.0.1、gcc (GCC) 4.3.2 20081105 (Red Hat 4.3.2-7)	
*Fedora release 10 (Cambridge) Linux 2.6.27.5-117.fc10.i686	
XML インスタンス検証 xerces付属サンプルのSAX2Countを改造し、以下のコマンドによりスキーマを使った全インスタンスの妥当性検証を行う。 SAX2Count2 -v=always ファイル名.xml	全てのサンプルXMLインスタンスについて、正常に完了した(OK)。 また、XMLスキーマに反するXMLインスタンスについて、エラーとなった(正常な動作:OK)。
*Java SE Development Kit 6 (jdk1.6.0_12)	
*Windows XP Professional SP3	
1 XML インスタンス検証・データ処理 JDKのxjcコマンドを使用し、Javaのクラスを自動生成する。 xjc jmx.xsd javax.xml.bind.JAXBContextを使用して読み込み、スキーマを使って妥当性検証を行って、Javaのクラスにバインドする。	全てのサンプルXMLインスタンスについて、正常に完了した(OK)。 また、XMLスキーマに反するXMLインスタンスについて、エラーとなった(正常な動作:OK)。
2 XML インスタンス検証 javax.xml.parsers.SAXParserFactoryにより、スキーマを使った全インスタンスの妥当性検証を行う	正常に完了した(OK)。

各ソフトウェアによる動作検証の詳細結果について

*Java 2 Platform Standard Edition Development Kit 5.0 (jdk1.5.0_07)	
*Windows XP Professional SP3	
XML インスタンス検証 javax.xml.parsers.SAXParserFactoryにより、スキーマを使った全インスタンスの妥当性検証を行う	正常に完了した(OK)。
*AltovaXML 2010 (AltovaXML Version 2010)	
*Windows XP Professional SP2	
XML インスタンス検証 スキーマを使った全インスタンスの妥当性検証を行う。	正常に完了した(OK)。
*Multi Schema Validator Ver.20080213	
*Windows XP Professional SP2	
XML インスタンス検証 スキーマを使った全インスタンスの妥当性検証を行う。	正常に完了した(OK)。
*axis-1.4	
*Windows XP Professional SP3	
*Java SE Development Kit 6 (jdk1.6.0_12)	
XML データ処理 スキーマをimportして独自に作成したwsdlを使って、Webサービス(サーバー側)を呼び出すコードを自動生成する。自動生成したコードを開発環境に取り込み、コンパイルできるか確認する。 java org.apache.axis.wsdl.WSDL2Java -s jmx.wsdl	エラー発生。 NullablefloatとNullableintegerに対して自動生成されたコードがコンパイルできない。 NullablefloatとNullableinteger内のunionを、単なるString型と書き換えることで回避可能(※1)。 Axisの問題。
*axis2-1.4.1	
*Windows XP Professional SP3	
*Java SE Development Kit 6 (jdk1.6.0_12)	
1 XML データ処理 スキーマをimportして独自に作成したwsdlを使って、Webサービス(サーバー側)を呼び出すコードを自動生成する。自動生成したコードを開発環境に取り込み、コンパイルできるか確認する。 wsdl2java.bat -uri jmx.wsdl -ss	正常に完了した(OK)。
2 XML データ処理 スキーマをimportして独自に作成したwsdlを使って、Webサービス(クライアント側)を呼び出すコードを自動生成する。自動生成したコードをクラスにバインドして、SOAP呼び出しを行う。 wsdl2java.bat -uri jmx.wsdl -g -u	エラー発生。 一部のインスタンスで、実行時にバインドで失敗した。サーバ側でも同様の問題が発生する。 NullablefloatとNullableinteger内のunionを、単なるString型と書き換えることで回避可能(※1)。 Axis2の問題。 参考> https://issues.apache.org/jira/browse/AXIS2-4001

各ソフトウェアによる動作検証の詳細結果について

※1 AxisとAxis2では以下のとおりjmx_eb_nullable.xsdを書き換えることにより回避可能であることを作業員にて確認している。
なお、平成21年5月15日付で公開している検証結果における回避策については、その後の追試により、書き換え見本ではAxisの動作がされないことを確認したため、今回提示している内容にて、置き換えることにより回避可能となる(平成21年5月15日公開のスキーマについても問題なく動作することを確認)。

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:jmx_eb="http://xml.kishou.go.jp/jmaxml1/elementBasis1/"
  elementFormDefault="qualified"
  targetNamespace="http://xml.kishou.go.jp/jmaxml1/elementBasis1/">
  <xs:complexType name="nullablefloat">
    <xs:simpleContent>
      <xs:extension base="xs:string" />
    </xs:simpleContent>
  </xs:complexType>
  <xs:complexType name="nullableinteger">
    <xs:simpleContent>
      <xs:extension base="xs:string" />
    </xs:simpleContent>
  </xs:complexType>
</xs:schema>
```