

気象庁防災情報 XML スキーマ（公開版）検証報告

2009年5月11日 作成

2009年5月13日 更新

日本アイ・ビー・エム株式会社

ソフトウェア事業

目次

気象庁防災情報 XML スキーマ（公開版）検証報告.....	1
1 検証環境.....	2
OS等のソフトウェア環境.....	2
検証対象ソフトウェア.....	2
2 検証内容分類.....	2
XMLスキーマ検証（スキーマ読み込み時等のVALIDATIONなど）.....	2
XMLインスタンス検証（サンプルXMLインスタンス等のVALIDATIONなど）.....	2
XMLデータ処理（データバインディング、XMLDB、WEBサービス化など）.....	2
3 検証内容.....	2
XMLスキーマ検証.....	2
XMLインスタンス検証.....	2
XMLデータ処理.....	3
4 検証結果概要.....	3
XMLスキーマ検証.....	3
XMLインスタンス検証.....	3
XMLデータ処理.....	3
5 検証結果詳細.....	4
XMLスキーマ検証.....	4
XMLインスタンス検証.....	4
XMLデータ処理.....	4
6 検証結果に対するコメントなど.....	15
7 参考資料（ファイル）.....	15

1 検証環境

OS等のソフトウェア環境

以下のOSにて検証を実施しました。

- ✓ Red Hat Enterprise Linux Advanced Platform 5.3
- ✓ SUSE Linux Enterprise Server 11
- ✓ Windows Server 2003 Enterprise Edition SP2
- ✓ Windows XP Professional SP2
- ✓ Ubuntu 8.10 Server [DB2 Express-Cのみ]

検証対象ソフトウェア

以下のソフトウェアを使用して検証を実施しました。

- ✓ DB2 Database for Linux, UNIX, and Windows Version 9.5 (Fix pack 3)
- ✓ DB2 Express-C 9.5.2 for Linux (32-bit) [Ubuntu 8.10 Serverのみ]
- ✓ IBM Data Studio Developer Version 2.1
 - XML インスタンス、XML スキーマの操作及び XQuery の作成に使用
 - Web サービスの作成に使用
- ✓ WebSphere Application Server Community Edition V2.1.1
 - Web サービスの実行に使用

2 検証内容分類

XML スキーマ検証 (スキーマ読み込み時等の Validation など)

XML インスタンス検証 (サンプル XML インスタンス等の Validation など)

XML データ処理 (データバインディング、XMLDB、Web サービス化など)

3 検証内容

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しました。

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しました。その際に XML スキーマを使った Validation を実施しています。
- b) XML スキーマの any となっている部分を検証するために、サンプル XML インスタンスに新しい要素を追加したものを Validation しデータベースに登録するという検証を実施しました。

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・操作を実施しました。

- a) XML パターンに基づく索引作成
- b) 全文検索索引の作成
- c) XQuery による検索
- d) XQuery による検索 (全文索引)
- e) XQuery Update Facility による XML インスタンスの部分更新
- f) XSLT スタイルシートによる HTML への変換
- g) SOAP バインディングによる Web サービスの作成と実行
- h) HTTP POST バインディングによる Web サービスの作成と実行
- 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

4 検証結果概要

XML スキーマ検証

OK

XML インスタンス検証

OK

XML データ処理

OK

5 検証結果詳細

XML スキーマ検証

XML スキーマをデータベースに登録できることを確認しました。

【参考】 register_schema.ddl

XML インスタンス検証

a) 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ました。

【参考】 create_table.ddl、xmllist.csv、 import.ddl

b) 70_01_01(090424)kisetsu.xml に新しい要素を追加しても、同じ XML スキーマで Validation し、データベースに登録できることも確認しました。
(70_01_01(090424)kisetsu_new.xml)

【追加部分の抜粋】

<AdditionalInfo>

<ObservationAddition>

<DeviationFromNormal>-5</DeviationFromNormal>

<DeviationFromLastYear>0</DeviationFromLastYear>

<Text>テキスト部分</Text>

</ObservationAddition>

<NewElement xmlns="http://xml.kishou.go.jp/jmaxml1/addition1/">

新しい要素の追加

</NewElement>

</AdditionalInfo>

XML データ処理

a) XML パターンに基づく索引作成

下記に示すコマンドで、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ました。

```
CREATE INDEX DB2ADMIN.TITLEIDX ON DB2ADMIN.JMXTB(DOC) GENERATE KEY  
USING XMLPATTERN
```

```
'declare namespace ns1="http://xml.kishou.go.jp/jmaxml1/";
```

```
/ns1:Report/ns1:Control/ns1:Title'
```

```
AS SQL VARCHAR(512) REJECT INVALID VALUES;
```

【参考】 create_index.ddl

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ました。

【参考】 nse.bat

c) XQuery による検索

以下に示す XQuery で XML インスタンスの検索を実行し、結果が得られることを確認しました。

【XQuery】

XQUERY

```
declare namespace jmx='http://xml.kishou.go.jp/jmaxml1/';
for $doc in
db2-fn:xmlcolumn("JMXTB.DOC")/jmx:Report/jmx:Control[jmx>Title = "特殊気象報"]
return $doc
```

【結果】

```
<Control xmlns="http://xml.kishou.go.jp/jmaxml1/"
xmlns:jmx="http://xml.kishou.go.jp/jmaxml1/">
  <Title>特殊気象報</Title>
  <DateTime>2008-03-23T02:00:00Z</DateTime>
  <Type>通常</Type>
  <EditorialOffice>横浜地方気象台</EditorialOffice>
  <PublishingOffice>横浜地方気象台</PublishingOffice>
</Control>
```

(他 5 件)

【参考】 XQuery01.sql、XQuery02.sql

d) XQuery による検索（全文索引）

以下に示す全文検索を含む XQuery で XML インスタンスの検索を実行し、結果が得られることを確認しました。この検索は、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 の部分を結果として返すようになっています。

【XQuery】

XQUERY

```
declare namespace jmx='http://xml.kishou.go.jp/jmaxml1/';
declare                                     namespace
jmx_mete='http://xml.kishou.go.jp/jmaxml1/body/meteorology1/';
for $doc in db2-fn:sqlquery('SELECT DOC FROM JMXTB WHERE
CONTAINS(DOC, "("最低気温" IN SAME SENTENCE AS "平年並") & "最低
気温" IN SAME SENTENCE AS "高い")=1')
return $doc/jmx:Report/jmx_mete:Body
```

【結果】

```
<Body xmlns:jmx_eb="http://xml.kishou.go.jp/jmaxml1/elementBasis1/"
xmlns="http://xml.kishou.go.jp/jmaxml1/body/meteorology1/"
xmlns:jmx="http://xml.kishou.go.jp/jmaxml1/">
  <Notice />
  <Comment>
    <Text type="本文">予報期間 9月6日から9月12日まで
      北日本から西日本は、気圧の谷や湿った気流の影響で
      雲が広がりやすく、期間のはじめに雨の降る日がありますが、期間の中頃は高気
      圧に覆われて晴
      れる所が多いでしょう。
      沖縄・奄美は、高気圧に覆われ晴れる日が多い見込み
      ですが、期間の後半は湿った気流の影響で雲が広がりやすいでしょう。
      全国的に、最高気温は平年並か平年より高いでしょう。
      最低気温は期間の中頃に平年より低い所がありますが、そのほかは平年並か平年
      より高いでしょう。
    </Text>
  </Comment>
</Body>
(他1件)
```

【参考】XQuery04.sql

e) XQuery Update Facility による XML インスタンスの部分更新
以下に示す XQuery で XML インスタンスの部分更新を実行し、結果が得られることを確認しました。この部分更新は、バージョンに関わる 3 つの部分を変更するようになっています。

【XQuery】

XQUERY

```
declare namespace jmx='http://xml.kishou.go.jp/jmaxml1/';
declare namespace
jmx_ib='http://xml.kishou.go.jp/jmaxml1/informationBasis1/';
copy $doc :=
db2-fn:xmlcolumn("JMXTB.DOC")[jmx:Report/jmx:Control/jmx>Title = "紫外線
観測データ"]
modify (
  do replace $doc/jmx:Report/jmx_ib:Head/jmx_ib:InfoKindVersion/text()
  with "1.1"
) return $doc
```

【結果】

```
<Report xmlns="http://xml.kishou.go.jp/jmaxml1/"
xmlns:jmx="http://xml.kishou.go.jp/jmaxml1/">
  <Control>
    <Title>紫外線観測データ</Title>
    <DateTime>2008-04-08T20:26:00+09:00
    </DateTime>
    <Type>通常</Type>
    <EditorialOffice>気象庁地球環境・海洋部</EditorialOffice>
    <PublishingOffice> 気 象 庁 地 球 環 境 ・ 海 洋 部
  </PublishingOffice>
  </Control>
  <Head xmlns="http://xml.kishou.go.jp/jmaxml1/informationBasis1/">
    <Title>紫外線観測データ</Title>
    <ReportDateTime>2008-04-08T20:26:00+09:00
    </ReportDateTime>
    <TargetDateTime>2008-04-08T20:00:00+09:00
    </TargetDateTime>
    <EventID />
    <InfoStatus>発表</InfoStatus>
    <Serial />
    <InfoKind>環境気象情報</InfoKind>
```

```

        <InfoKindVersion>1.1</InfoKindVersion>
        <Headline>
            <Text />
        </Headline>
    </Head>
    .....
    </Body>
</Report>

```

【参考】XQuery03.sql

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 (HTML_table.xml、HTML_ul.xml) を作成し SQL による HTML 変換を実施し問題なく変換できることを確認しました。

【参考】HTML_table.xml、HTML_ul.xml、create_table_xsl.ddl、insert_xsl.ddl

【SQL】

```

SELECT XSLTRANSFORM(DOC USING XSL AS VARCHAR(4096))
      FROM JMXTB, XSLTB
      WHERE XSLTB.ID = ?
;

```

【XSLT 】

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
      xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
      xmlns:jmx="http://xml.kishou.go.jp/jmaxml1/"
      exclude-result-prefixes="jmx">
  <xsl:output method="html"/>
  <xsl:template match="/">
    <xsl:apply-templates select="jmx:Report/jmx:Control"/>
  </xsl:template>
  <xsl:template match="jmx:Control">
    <tr>
      <td>
        <xsl:value-of select="jmx:Title"/>
      </td>
      <td>

```

```

        <xsl:value-of select="jmx:DateTime" />
    </td>
    <td>
        <xsl:value-of select="jmx:Status" />
    </td>
    <td>
        <xsl:value-of select="jmx:PublishingOffice"
/>
    </td>
</tr>
</xsl:template>
</xsl:stylesheet>

```

【XSLT Ⅱ】

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:jmx="http://xml.kishou.go.jp/jmaxml1/"
    exclude-result-prefixes="jmx">
    <xsl:output method="html" />
    <xsl:template match="/">
        <xsl:apply-templates select="jmx:Report/jmx:Control" />
    </xsl:template>
    <xsl:template match="jmx:Control">
        <h3>
            <xsl:value-of select="jmx:Title" />
        </h3>
        <ul>
            <li>
                <xsl:value-of select="jmx:DateTime" />
            </li>
            <li>
                <xsl:value-of select="jmx:Status" />
            </li>
            <li>
                <xsl:value-of select="jmx:PublishingOffice"

```

```

/>
                </li>
            </ul>
        </xsl:template>
</xsl:stylesheet>

```

【結果 1】

```

<tr>
<td>季節観測</td><td>2008-03-23T02:00:00Z</td><td>通常</td><td>横浜地方
気象台</td>
</tr>
(他 145 件)

```

【結果 2】

```

<h3>季節観測</h3>
<ul>
<li>2008-03-23T02:00:00Z</li>
<li>通常</li>
<li>横浜地方気象台</li>
</ul>
(他 145 件)

```

g) SOAP バインディングによる Web サービスの作成と実行
以下の 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、
実行し、問題なく結果が返ることを確認しました。

【SQL/XML】

```

SELECT XMLQUERY('
    declare namespace jmx="http://xml.kishou.go.jp/jmaxml1/";
    for $doc in $DOC/jmx:Report/jmx:Control
    return $doc
') AS RET FROM JMXTB
WHERE XMLEXISTS('
    declare namespace jmx="http://xml.kishou.go.jp/jmaxml1/";
    $DOC/jmx:Report/jmx:Control[jmx:Title = $title]
' PASSING CAST (? AS VARCHAR(128)) AS "title")

```

【結果 : SOAP リクエスト】

```

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soapenv:Body>
    <q0:SqlXml xmlns:q0="urn:jma"

      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <p1>季節観測</p1>
      </q0:SqlXml>
    </soapenv:Body>
</soapenv:Envelope>

```

【結果：SOAP レスポンス】

```

<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Body>
<ns1:SqlXmlResponse xmlns:ns1="urn:jma"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="">
<row>
<RET>
<ns2:Control xmlns="http://xml.kishou.go.jp/jmaxml1/"
xmlns:jmx="http://xml.kishou.go.jp/jmaxml1/"
xmlns:ns2="http://xml.kishou.go.jp/jmaxml1/">
<ns2:Title>&#x5B63;&#x7BC0;&#x89B3;&#x6E2C;</ns2:Title>
<ns2:DateTime>2008-03-23T02:00:00Z</ns2:DateTime>
<ns2:Status>&#x901A;&#x5E38;</ns2:Status>
<ns2:EditorialOffice>&#x6A2A;&#x6D5C;&#x5730;&#x65B9;&#x6C17;&#x8C
61;&#x53F0;</ns2:EditorialOffice>
<ns2:PublishingOffice>&#x6A2A;&#x6D5C;&#x5730;&#x65B9;&#x6C17;&#x8
C61;&#x53F0;</ns2:PublishingOffice>
</ns2:Control>
</RET>

```

```
</row>
</ns1:SqlXmlResponse>
</soapenv:Body>
</soapenv:Envelope>
```

【結果】

SOAP レスポンスでは XML が文字参照で表示されているので、ブラウザで確認

```
-<RET>
  -<ns2:Control>
    <ns2:Title>季節観測</ns2:Title>
    <ns2:DateTime>2008-03-23T02:00:00Z</ns2:DateTime>
    <ns2:Status>通常</ns2:Status>
    <ns2:EditorialOffice>横浜地方気象台</ns2:EditorialOffice>
    <ns2:PublishingOffice>横浜地方気象台</ns2:PublishingOffice>
  </ns2:Control>
</RET>
```

h) HTTP バインディングによる Web サービスの作成と実行

以下の 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、実行し、問題なく結果が返ることを確認しました。

【SQL/XML】

```
SELECT XMLQUERY('
    declare namespace jmx="http://xml.kishou.go.jp/jmaxml1/";
    for $doc in $DOC/jmx:Report/jmx:Control
    return $doc
') AS RET FROM JMXTB
WHERE XMLEXISTS('
    declare namespace jmx="http://xml.kishou.go.jp/jmaxml1/";
    $DOC/jmx:Report/jmx:Control[jmx:Title = $title]
' PASSING CAST (? AS VARCHAR(128)) AS "title")
```

【結果：POST データ】

```
p1=%E5%AD%A3%E7%AF%80%E8%A6%B3%E6%B8%AC
```

【結果：XML】

```
<?xml version="1.0" encoding="UTF-8"?>
<ns1:SqlXmlResponse xmlns="" xmlns:ns1="urn:jma"
```

```

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <row>
    <RET>
      <Control xmlns="http://xml.kishou.go.jp/jmaxml1/"
xmlns:jmx="http://xml.kishou.go.jp/jmaxml1/">
        <Title>季節観測</Title>

        <DateTime>2008-03-23T02:00:00Z</DateTime>
        <Status>通常</Status>
        <EditorialOffice> 横浜地方気象台
</EditorialOffice>
        <PublishingOffice> 横浜地方気象台
</PublishingOffice>
      </Control>
    </RET>
  </row>
</ns1:SqlXmlResponse>

```

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
以下の 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、
実行し、問題なく結果が返ることを確認しました。

【SQL/XML】

```

SELECT XMLQUERY('
  declare namespace jmx="http://xml.kishou.go.jp/jmaxml1/";
  for $doc in $DOC/jmx:Report/jmx:Control
  return $doc
') AS RET FROM JMXTB
WHERE XMLEXISTS('
  declare namespace jmx="http://xml.kishou.go.jp/jmaxml1/";
  $DOC/jmx:Report/jmx:Control[jmx:Title = $title]
' PASSING CAST (? AS VARCHAR(128)) AS "title")

```

【結果：リクエスト URL】

http://localhost:8080/JmaWebService/rest/WebService/SqlXml?p1=%E5%AD%A3%E7%AF%80%E8%A6%B3%E6%B8%AC

【結果：XML】

```
<?xml version="1.0" encoding="UTF-8"?>
<ns1:SqlXmlResponse xmlns="" xmlns:ns1="urn:jma"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <row>
    <RET>
      <Control xmlns="http://xml.kishou.go.jp/jmaxml1/"
xmlns:jmx="http://xml.kishou.go.jp/jmaxml1/">
        <Title>季節観測</Title>

        <DateTime>2008-03-23T02:00:00Z</DateTime>
        <Status>通常</Status>
        <EditorialOffice> 横 浜 地 方 気 象 台
</EditorialOffice>
        <PublishingOffice> 横 浜 地 方 気 象 台
</PublishingOffice>
      </Control>
    </RET>
  </row>
</ns1:SqlXmlResponse>
```

6 検証結果に対するコメントなど

前回の報告で書かせていただいた内容と重複しますが、技術的には全く問題ないと思います。但し、大量の XML インスタンスを識別するためのユニークな ID のようなものが必要だと考えます。

7 参考資料 (ファイル)

今回の検証で作成、使用したスクリプトや XQuery を添付させていただきます。

schema/	: XML スキーマ・ディレクトリ
sample/	: サンプル XML インスタンス・ディレクトリ
create_db.ddl	: データベースの作成
register_schema.ddl	: XML スキーマの登録
create_table.ddl	: テーブル作成
xmllist.csv	: インポート用データ
import.ddl	: XML インスタンスのインポート (ロード)
create_index.ddl	: インデックス作成
nse.bat	: 全文検索用インデックス作成
XQuery01.sql	: XQuery 検索
XQuery02.sql	: XQuery 検索
XQuery03.sql	: XQuery 部分更新
XQuery04.sql	: XQuery 全文検索
create_table_xsl.ddl	: XSLT 格納用テーブル作成
insert_xsl.ddl	: XSLT 登録
HTML_table.xsl	: HTML 変換 XSLT
HTML_ul.xsl	: HTML 変換 XSLT
XSLT01.sql	: HTML 変換用 SQL
70_01_01(090424)kisetsu_new.xml	: 新しい要素を追加したサンプル XML インスタンス

以上