

AMIS SERVICES BV, NIEUWEGEIN, THE NETHERLANDS

Creating Structure in Unstructured Data

DML and DDL statements used for Binary XML Securefile storage

Marco Gralike

3/4/2013

Contents

Binary XML Securefile storage – XML Schema based	2
Code used	2
Delete XML Schema if registered in the XDB Repository.....	2
Register the XML Schema for Binary XML usage	2
Create a range partitioned table with a Binary XML securefile column.....	3
Oracle Text Index	4
Creating the Unstructured XML Index	5
Adding a structured XMLIndex to the Unstructured XMLIndex	6
Creating secondary indexes on the content table columns	6
XQuery making use of the structured XMLIndex and secondary indexes	7
XQuery Full Text examples based on XMLDB- H.O.L. OpenWorld 2012.....	7
XML Schema.....	12
Mediawiki Export-0.8 XML Schema annotated for Binary XML usage	12

Binary XML Securefile storage – XML Schema based

Here some code examples that have been used to collect data for the “Creating Structure in Unstructured” presentation. The code examples given here are only one of the storage possibilities tested.

Possible storage/load alternatives are and that have been tested:

- Object Relational XMLType Table storage
- Securefile Binary XML normal column or XMLType table
- Securefile Binary XML virtual column partitioned XMLType table
- Securefile Binary XML normal range partitioned table on a unique Wikipedia Page ID column
- XML Schema based and XML Schema less variations
- SAX Loading of data while making use of client side Binary XML encoding

The test user used as the default tablespace a BIGFILE tablespace with ASSM enabled. Besides the DBA role also the XDBADMIN role was granted (and needed) and an “XMLDIR” Oracle directory was created to load the Wikipedia export-08.xsd file from a local server directory.

The XML dump file from Wikipedia can be downloaded via the information given on: http://en.wikipedia.org/wiki/Wikipedia:Database_download. For testing purposes the English full XML dump file was used called pages-articles.xml, but a smaller subset can also be downloaded via the site mentioned, or alternatives in different languages if needed.

Code used

Delete XML Schema if registered in the XDB Repository

```
EXECUTE dbms_xmlschema.deleteSchema('http://www.mediawiki.org/xml/export-0.8.xsd',4);
```

Register the XML Schema for Binary XML usage

```
BEGIN
  DBMS_XMLSCHEMA.registerSchema (
    schemaURL => 'http://www.mediawiki.org/xml/export-0.8.xsd'
  , schemadoc => bfilename('XMLDIR','export-0.8.CSX_XSD.xsd')
  , LOCAL    => TRUE    -- local
  , GENTYPES => FALSE  -- generate object types
  , GENBEAN  => FALSE  -- no java beans
  , GENTABLES => FALSE -- generate object tables
  , FORCE     => FALSE
  , ENABLEHIERARCHY => DBMS_XMLSCHEMA.ENABLE_HIERARCHY_NONE
  )
END
```

```

, OPTIONS      => DBMS_XMLSCHEMA.REGISTER_BINARYXML
, OWNER       => USER);

END;

/

```

Create a range partitioned table with a Binary XML securefile column

```

CREATE TABLE "BINARYXML_RANGE_PART_XSD"
(ID NUMBER(38),
CONTENT XMLTYPE)
XMLTYPE COLUMN content STORE AS SECUREFILE BINARY XML
XMLSCHEMA "http://www.mediawiki.org/xml/export-0.8.xsd"
ELEMENT "page"
PARTITION BY RANGE (ID)
(PARTITION RANGE_PART_01 VALUES LESS THAN (1000000) TABLESPACE "BINARY_XML"
, PARTITION RANGE_PART_02 VALUES LESS THAN (2000000) TABLESPACE "BINARY_XML"
, PARTITION RANGE_PART_03 VALUES LESS THAN (3000000) TABLESPACE "BINARY_XML"
, PARTITION RANGE_PART_04 VALUES LESS THAN (4000000) TABLESPACE "BINARY_XML"
, PARTITION RANGE_PART_05 VALUES LESS THAN (5000000) TABLESPACE "BINARY_XML"
, PARTITION RANGE_PART_06 VALUES LESS THAN (6000000) TABLESPACE "BINARY_XML"
, PARTITION RANGE_PART_07 VALUES LESS THAN (7000000) TABLESPACE "BINARY_XML"
, PARTITION RANGE_PART_08 VALUES LESS THAN (8000000) TABLESPACE "BINARY_XML"
, PARTITION RANGE_PART_09 VALUES LESS THAN (9000000) TABLESPACE "BINARY_XML"
, PARTITION RANGE_PART_10 VALUES LESS THAN (10000000) TABLESPACE "BINARY_XML"
, PARTITION RANGE_PART_11 VALUES LESS THAN (11000000) TABLESPACE "BINARY_XML"
, PARTITION RANGE_PART_12 VALUES LESS THAN (12000000) TABLESPACE "BINARY_XML"
, PARTITION RANGE_PART_13 VALUES LESS THAN (13000000) TABLESPACE "BINARY_XML"
, PARTITION RANGE_PART_MAX VALUES LESS THAN (MAXVALUE))
TABLESPACE "BINARY_XML"
PARALLEL NOLOGGING;

```

Oracle Text Index

```
DROP INDEX FT_RANGE_PART_IDX FORCE;

begin
    CTX_DDL.DROP_SECTION_GROUP('FTRI');
end;

/

begin
    CTX_DDL.DROP_PREFERENCE('STORAGE_PREFS');
end;

/

begin
    CTX_DDL.CREATE_SECTION_GROUP('FTRI','PATH_SECTION_GROUP');
    CTX_DDL.SET_SEC_GRP_ATTR('FTRI','XML_ENABLE','T');
    CTX_DDL.create_preference('STORAGE_PREFS','BASIC_STORAGE');
    CTX_DDL.set_attribute(
        'STORAGE_PREFS',
        'D_TABLE_CLAUSE',
        'LOB(DOC) STORE AS SECUREFILE (COMPRESS MEDIUM CACHE)');
    CTX_DDL.set_attribute(
        'STORAGE_PREFS',
        'I_TABLE_CLAUSE',
        'LOB(TOKEN_INFO) STORE AS SECUREFILE (NOCOMPRESS CACHE) '
    );
END;

/

CREATE INDEX FT_RANGE_PART_IDX
    ON BINARYXML_RANGE_PART_XSD(CONTENT)
    INDEXTYPE IS CTXSYS.CONTEXT
    LOCAL
    PARAMETERS(
        'storage STORAGE_PREFS
        section group FTRI'
    )
PARALLEL;
```

Creating the Unstructured XML Index

```
CREATE INDEX XMLINDEX_BINARYXML_RANGE
ON BINARYXML_RANGE_PART_XSD (CONTENT)
INDEXTYPE IS "XDB"."XMLINDEX"
  PARAMETERS ('PATHS (EXCLUDE (/page/revision/comment
                               /page/revision/text
                               )
              NAMESPACE MAPPING (xmlns="http://www.mediawiki.org/xml/export-0.8/"))
  PATH      TABLE UXI_RANGE_PATH_TABLE (TABLESPACE BINARY_XML PARALLEL)
  PIKEY     INDEX UXI_RANGE_PIKEY       (TABLESPACE BINARY_XML PARALLEL)
  PATH ID   INDEX UXI_RANGE_PATH_ID     (TABLESPACE BINARY_XML PARALLEL)
  ORDER KEY INDEX UXI_RANGE_ORDER_KEY   (TABLESPACE BINARY_XML PARALLEL)
  VALUE     INDEX UXI_RANGE_VALUE       (TABLESPACE BINARY_XML PARALLEL)
  ASYNC (SYNC ALWAYS)
  ') LOCAL (
PARTITION "RANGE_PART_01",
PARTITION "RANGE_PART_02",
PARTITION "RANGE_PART_03",
PARTITION "RANGE_PART_04",
PARTITION "RANGE_PART_05",
PARTITION "RANGE_PART_06",
PARTITION "RANGE_PART_07",
PARTITION "RANGE_PART_08",
PARTITION "RANGE_PART_09",
PARTITION "RANGE_PART_10",
PARTITION "RANGE_PART_11",
PARTITION "RANGE_PART_12",
PARTITION "RANGE_PART_13",
PARTITION "RANGE_PART_MAX")
PARALLEL ;
```

Adding a structured XMLIndex to the Unstructured XMLIndex

```
ALTER INDEX XMLINDEX_BINARYXML_RANGE
  PARAMETERS
('ADD_GROUP GROUP WIKI_GROUP_01
 XMLTABLE WIKI_GROUP_SXI_TAB_01
 XMLNAMESPACES(DEFAULT 'http://www.mediawiki.org/xml/export-0.8/' )
  , '/page'
 COLUMNS
  PAGE_ID          NUMBER(38)          PATH 'id'
  , PAGE_TITLE     VARCHAR2(4000)     PATH 'title'
  , PAGE_REV_TIMESTAMP TIMESTAMP(6) WITH TIME ZONE PATH 'revision/timestamp'
');
```

Creating secondary indexes on the content table columns

```
CREATE UNIQUE INDEX CNT_PART_PAGE_ID_UXI ON WIKI_GROUP_SXI_TAB_01 (PAGE_ID)
PARALLEL LOGGING
TABLESPACE HOTSOS;

CREATE INDEX CNT_PART_PAGE_TITLE_IX ON WIKI_GROUP_SXI_TAB_01 (PAGE_TITLE)
PARALLEL LOGGING
TABLESPACE HOTSOS;

CREATE UNIQUE INDEX CNT_PART_TITLE_ID_UIX ON WIKI_GROUP_SXI_TAB_01 (PAGE_ID,PAGE_TITLE)
PARALLEL LOGGING
TABLESPACE HOTSOS;
```

XQuery making use of the structured XMLIndex and secondary indexes

```
SELECT PAGE_ID,
       PAGE_TITLE,
       PAGE_REV_TIMESTAMP
FROM   BINARYXML_RANGE_PART_XSD t1,
       XMLTABLE (xmlnamespaces(default 'http://www.mediawiki.org/xml/export-0.8/' )
                , '/page'
                PASSING t1.content
                COLUMNS
                   PAGE_ID          NUMBER(9)          PATH 'id'
                   , PAGE_TITLE     VARCHAR2(100)      PATH 'title'
                   , PAGE_REV_TIMESTAMP TIMESTAMP(6) WITH TIME ZONE PATH 'revision/timestamp'
                )
WHERE  PAGE_TITLE='Andrea Andreani'
       AND PAGE_ID=1754
;
```

XQuery Full Text examples based on XMLDB- H.O.L. OpenWorld 2012

```
-- An XQuery Full-Text "contains text" search on a phrase with stemming.
-- The index is used since "contains text" comparisons are case insensitive.
-- Results are returned since stemming identified that sport is stem for sporting.
--
SELECT xt1.PAGE_TEXT
FROM   BINARYXML_RANGE_PART_XSD t1,
       XMLTABLE(xmlnamespaces(default 'http://www.mediawiki.org/xml/export-0.8/' ),
                '$P/page/revision/text'
                PASSING t1.content as "P"
                COLUMNS
                   PAGE_TEXT varchar2(32767) PATH '.'
                ) xt1
WHERE  XMLEExists( 'xquery version "1.0"; (: :)'
                  declare default element namespace "http://www.mediawiki.org/xml/export-0.8/"; (: :)'
                  $P/page/revision/text[. contains text {$PHRASE} using stemming]'
                PASSING
                   t1.content as "P"
```



```

        , 'oracle' as "PHRASE"
    )
AND rownum <= 10
/
--
-- An XQuery Full-Text "contains text" search on a fragment.
-- The index is used since "contains text" comparisons are case insensitive.
--
SELECT xt1.PAGE_TEXT
FROM BINARYXML_RANGE_PART_XSD t1,
XMLTABLE(xmlnamespaces(default 'http://www.mediawiki.org/xml/export-0.8/' ),
'$P/page/revision/text'
PASSING t1.content as "P"
COLUMNS
PAGE_TEXT varchar2(4000) PATH '.'
) xt1
WHERE XMLEExists( 'xquery version "1.0"; (: : )
declare default element namespace "http://www.mediawiki.org/xml/export-0.8/"; (: : )
$P/page/revision/text[. contains text {$PHRASE}]'
PASSING
t1.content as "P"
, 'oracle' as "PHRASE"
)
AND rownum <= 10
/
--
-- An XQuery Full-Text "contains text" search on a fragment using the ftand operator.
-- The index is used since "contains text" comparisons are case insensitive.
-- Results are returned since stemming identified 'centre' as a stem for 'center'
--
SELECT xt1.PAGE_TEXT
FROM BINARYXML_RANGE_PART_XSD t1,
XMLTABLE(xmlnamespaces(default 'http://www.mediawiki.org/xml/export-0.8/' ),
'$P/page/revision/text'
PASSING t1.content as "P"
COLUMNS
PAGE_TEXT varchar2(4000) PATH '.'
) xt1
WHERE XMLEExists( 'xquery version "1.0"; (: : )

```

```

        declare default element namespace "http://www.mediawiki.org/xml/export-0.8/"; (: :)
        $P/page/revision/text[. contains text {$PHRASE1} ftand {$PHRASE2} using stemming]'
        PASSING
            t1.content as "P",
            'oracle' as "PHRASE1",
            'fusion' as "PHRASE2"
        )
    AND rownum <= 10
/
--
-- An XQuery Full-Text "contains text" search on a fragment using the ftand operator.
-- The index is used since the "contains text" comparison is case insensitive.
-- The Window clause specifies that the words must appear with 2 words of each other.
--
SELECT xt1.PAGE_TEXT
FROM BINARYXML_RANGE_PART_XSD t1,
XMLTABLE(xmlnamespaces(default 'http://www.mediawiki.org/xml/export-0.8/' ),
' $P/page/revision/text'
PASSING t1.content as "P"
COLUMNS
PAGE_TEXT varchar2(4000) PATH '.'
) xt1
WHERE XMLEExists( 'xquery version "1.0"; (: :)
        declare default element namespace "http://www.mediawiki.org/xml/export-0.8/"; (: :)
        $P/page/revision/text[. contains text {$PHRASE1} ftand {$PHRASE2} using stemming window
2 words]'
        PASSING
            t1.content as "P",
            'oracle' as "PHRASE1",
            'fusion' as "PHRASE2"
        )
    AND rownum <= 10
/
--
-- An XQuery Full-Text "contains text" search on a fragment using the ftand operator.
-- The index is used since the "contains text" comparison is case insensitive.
-- The Window clause specifies that the words must appear with 6 words of each other.
-- Results are returned since the window is large enough
--

```

```

SELECT xt1.PAGE_TEXT
FROM BINARYXML_RANGE_PART_XSD t1,
XMLTABLE(xmlnamespaces(default 'http://www.mediawiki.org/xml/export-0.8/' ),
'$P/page/revision/text'
PASSING t1.content as "P"
COLUMNS
PAGE_TEXT varchar2(4000) PATH '.'
) xt1
WHERE XMLEExists( 'xquery version "1.0"; (: :)
declare default element namespace "http://www.mediawiki.org/xml/export-0.8/"; (: :)
$P/page/revision/text[. contains text {$PHRASE1} ftand {$PHRASE2} using stemming window
6 words]'
PASSING
t1.content as "P",
'oracle' as "PHRASE1",
'fusion' as "PHRASE2"
)
AND rownum <= 10
/
--
-- Window clause of 2, while searching 1.000.000 wikipedia pages
--
SELECT count(*)
FROM BINARYXML_RANGE_PART_XSD t1,
XMLTABLE(xmlnamespaces(default 'http://www.mediawiki.org/xml/export-0.8/' ),
'$P/page/revision/text'
PASSING t1.content as "P"
COLUMNS
PAGE_TEXT varchar2(4000) PATH '.'
) xt1
WHERE XMLEExists( 'xquery version "1.0"; (: :)
declare default element namespace "http://www.mediawiki.org/xml/export-0.8/"; (: :)
$P/page/revision/text[. contains text {$PHRASE1} ftand {$PHRASE2} using stemming window
2 words]'
PASSING
t1.content as "P",
'oracle' as "PHRASE1",
'ellison' as "PHRASE2"
)
/

```

```

--
-- Window clause of 6, while searching 1.000.000 wikipedia pages
--
SELECT count(*)
FROM BINARYXML_RANGE_PART_XSD t1,
XMLTABLE(xmlnamespaces(default 'http://www.mediawiki.org/xml/export-0.8/' ),
'$P/page/revision/text'
PASSING t1.content as "P"
COLUMNS
PAGE_TEXT varchar2(4000) PATH '.'
) xt1
WHERE XMLEExists( 'xquery version "1.0"; (: :)
declare default element namespace "http://www.mediawiki.org/xml/export-0.8/"; (: :)
$P/page/revision/text[. contains text {$PHRASE1} ftand {$PHRASE2} using stemming window
6 words]'
PASSING
t1.content as "P",
'oracle' as "PHRASE1",
'ellison' as "PHRASE2"
)
/

```

XML Schema

Mediawiki Export-0.8 XML Schema annotated for Binary XML usage

```
<?xml version="1.0" encoding="UTF-8" ?>
<!--
    This is an XML Schema description of the format
    output by MediaWiki's Special:Export system.

    Version 0.2 adds optional basic file upload info support,
    which is used by our OAI export/import submodule.

    Version 0.3 adds some site configuration information such
    as a list of defined namespaces.

    Version 0.4 adds per-revision delete flags, log exports,
    discussion threading data, a per-page redirect flag, and
    per-namespace capitalization.

    Version 0.5 adds byte count per revision.

    Version 0.6 adds a separate namespace tag, and resolves the
    redirect target and adds a separate sha1 tag for each revision.

    Version 0.7 adds a unique identity constraint for both page and
    revision identifiers. See also bug 4220.

    Fix type for <ns> from "positiveInteger" to "nonNegativeInteger" to allow 0
    Moves <logitem> to its right location.

    Add parentid to revision.

    Fix type for <id> within <contributor> to "nonNegativeInteger"

    Version 0.8 adds support for a <model> and a <format> tag for
    each revision. See contenthandler.txt.

    The canonical URL to the schema document is:
    http://www.mediawiki.org/xml/export-0.8.xsd
```

```

Use the namespace:
http://www.mediawiki.org/xml/export-0.8/
-->
<schema xmlns="http://www.w3.org/2001/XMLSchema"
        xmlns:mw="http://www.mediawiki.org/xml/export-0.8/"
        targetNamespace="http://www.mediawiki.org/xml/export-0.8/"
        xmlns:xdb="http://xmlns.oracle.com/xdb"
        elementFormDefault="qualified">

  <annotation>
    <documentation xml:lang="en">
      MediaWiki's page export format
    </documentation>
  </annotation>

  <!-- Need this to reference xml:lang -->
  <import namespace="http://www.w3.org/XML/1998/namespace"
          schemaLocation="http://www.w3.org/2001/xml.xsd" />

  <!-- Begin root element -->

  <element name="siteinfo" type="mw:SiteInfoType" />
  <element name="page" type="mw:PageType" />
  <element name="logitem" type="mw:LogItemType" />

  <!-- End root element -->

  <!-- Original root element -->
  <element name="mediawiki" type="mw:MediaWikiType">
    <!-- Page ID constraint, see bug 4220 -->
    <unique name="PageIDConstraint">
      <selector xpath="mw:page" />
      <field xpath="mw:id" />
    </unique>
    <!-- Revision ID constraint, see bug 4220 -->
    <unique name="RevIDConstraint">
      <selector xpath="mw:page/mw:revision" />

```

```

        <field xpath="mw:id" />

    </unique>

</element>

<complexType name="MediaWikiType">
    <sequence>
        <element name="siteinfo" type="mw:SiteInfoType"
            minOccurs="0" maxOccurs="1" />
        <element name="page" type="mw:PageType"
            minOccurs="0" maxOccurs="unbounded" />
        <element name="logitem" type="mw:LogItemType"
            minOccurs="0" maxOccurs="unbounded" />
    </sequence>
    <attribute name="version" type="string" use="required" />
    <attribute ref="xml:lang" use="required" />
</complexType>

<complexType name="SiteInfoType">
    <sequence>
        <element name="sitename" type="string" minOccurs="0" />
        <element name="base" type="anyURI" minOccurs="0" />
        <element name="generator" type="string" minOccurs="0" />
        <element name="case" type="mw:CaseType" minOccurs="0" />
        <element name="namespaces" type="mw:NamespacesType" minOccurs="0" />
    </sequence>
</complexType>

<simpleType name="CaseType">
    <restriction base="NMTOKEN">
        <!-- Cannot have two titles differing only by case of first letter. -->
        <!-- Default behavior through 1.5, $wgCapitalLinks = true -->
        <enumeration value="first-letter" />

        <!-- Complete title is case-sensitive -->
        <!-- Behavior when $wgCapitalLinks = false -->
        <enumeration value="case-sensitive" />
    </restriction>
</simpleType>

```

```

        <!-- Cannot have non-case sensitive titles eg [[FOO]] == [[Foo]] -->
        <!-- Not yet implemented as of MediaWiki 1.18 -->
        <enumeration value="case-insensitive" />
    </restriction>
</simpleType>

<simpleType name="DeletedFlagType">
    <restriction base="NMTOKEN">
        <enumeration value="deleted" />
    </restriction>
</simpleType>

<complexType name="NamespacesType">
    <sequence>
        <element name="namespace" type="mw:NamespaceType"
            minOccurs="0" maxOccurs="unbounded" />
    </sequence>
</complexType>

<complexType name="NamespaceType">
    <simpleContent>
        <extension base="string">
            <attribute name="key" type="integer" />
            <attribute name="case" type="mw:CaseType" />
        </extension>
    </simpleContent>
</complexType>

<complexType name="RedirectType">
    <simpleContent>
        <extension base="string">
            <attribute name="title" type="string" />
        </extension>
    </simpleContent>
</complexType>

<simpleType name="ContentModelType">

```



```

        <restriction base="string">
            <pattern value="[a-zA-Z][+./a-zA-Z0-9]*" />
        </restriction>
    </simpleType>

    <simpleType name="ContentFormatType">
        <restriction base="string">
            <pattern value="[a-zA-Z][+./a-zA-Z0-9]*/[a-zA-Z][+./a-zA-Z0-9]*" />
        </restriction>
    </simpleType>

    <complexType name="PageType">
        <sequence>
            <!-- Title in text form. (Using spaces, not underscores; with namespace
) -->
            <element name="title" type="string" />

            <!-- Namespace in canonical form -->
            <element name="ns" type="nonNegativeInteger" />

            <!-- optional page ID number -->
            <element name="id" type="positiveInteger" />

            <!-- flag if the current revision is a redirect -->
            <element name="redirect" type="mw:RedirectType" minOccurs="0"
maxOccurs="1" />

            <!-- comma-separated list of string tokens, if present -->
            <element name="restrictions" type="string" minOccurs="0" />

            <!-- Zero or more sets of revision or upload data -->
            <choice minOccurs="0" maxOccurs="unbounded">
                <element name="revision" type="mw:RevisionType" />
                <element name="upload" type="mw:UploadType" />
            </choice>

            <!-- Zero or One sets of discussion threading data -->

```

```

        <element name="discussionthreadinginfo" minOccurs="0" maxOccurs="1"
type="mw:DiscussionThreadingInfo" />
    </sequence>
</complexType>

<complexType name="RevisionType">
    <sequence>
        <element name="id" type="positiveInteger" />
        <element name="parentid" type="positiveInteger" minOccurs="0" />
        <element name="timestamp" type="dateTime" xdb:SQLType="TIMESTAMP WITH
TIME ZONE" />
        <element name="contributor" type="mw:ContributorType" />
        <element name="minor" minOccurs="0" maxOccurs="1" />
        <element name="comment" type="mw:CommentType" minOccurs="0"
maxOccurs="1" />
        <element name="text" type="mw:TextType" />
        <element name="sha1" type="string" />
        <element name="model" type="mw:ContentModelType" />
        <element name="format" type="mw:ContentFormatType" />
    </sequence>
</complexType>

<complexType name="LogItemType">
    <sequence>
        <element name="id" type="positiveInteger" />
        <element name="timestamp" type="dateTime" xdb:SQLType="TIMESTAMP WITH
TIME ZONE" />
        <element name="contributor" type="mw:ContributorType" />
        <element name="comment" type="mw:CommentType" minOccurs="0" />
        <element name="type" type="string" />
        <element name="action" type="string" />
        <element name="text" type="mw:LogTextType" minOccurs="0" maxOccurs="1"
/>
        <element name="logtitle" type="string" minOccurs="0" maxOccurs="1" />
        <element name="params" type="mw:LogParamsType" minOccurs="0"
maxOccurs="1" />
    </sequence>
</complexType>

<complexType name="CommentType">

```

```

        <simpleContent>
            <extension base="string">
                <!-- This allows deleted=deleted on non-empty elements, but XSD
is not omnipotent -->
                <attribute name="deleted" use="optional"
type="mw:DeletedFlagType" />
            </extension>
        </simpleContent>
    </complexType>

    <complexType name="TextType">
        <simpleContent>
            <extension base="string">
                <attribute ref="xml:space" use="optional" default="preserve" />
                <!-- This allows deleted=deleted on non-empty elements, but XSD
is not omnipotent -->
                <attribute name="deleted" use="optional"
type="mw:DeletedFlagType" />
                <!-- This isn't a good idea; we should be using "ID" instead of
"NMTOKEN" -->
                <!-- However, "NMTOKEN" is strictest definition that is both
compatible with existing -->
                <!-- usage ([0-9]+) and with the "ID" type. -->
                <attribute name="id" type="NMTOKEN" />
                <attribute name="bytes" use="optional" type="nonNegativeInteger"
/>
            </extension>
        </simpleContent>
    </complexType>

    <complexType name="LogTextType">
        <simpleContent>
            <extension base="string">
                <!-- This allows deleted=deleted on non-empty elements, but XSD
is not omnipotent -->
                <attribute name="deleted" use="optional"
type="mw:DeletedFlagType" />
            </extension>
        </simpleContent>
    </complexType>

```

```

<complexType name="LogParamsType">
    <simpleContent>
        <extension base="string">
            <attribute ref="xml:space" use="optional" default="preserve" />
        </extension>
    </simpleContent>
</complexType>

<complexType name="ContributorType">
    <sequence>
        <element name="username" type="string" minOccurs="0" />
        <element name="id" type="nonNegativeInteger" minOccurs="0" />

        <element name="ip" type="string" minOccurs="0" />
    </sequence>
    <!-- This allows deleted=deleted on non-empty elements, but XSD is not
omnipotent -->
    <attribute name="deleted" use="optional" type="mw:DeletedFlagType" />
</complexType>

<complexType name="UploadType">
    <sequence>
        <!-- Revision-style data... -->
        <element name="timestamp" type="dateTime" />
        <element name="contributor" type="mw:ContributorType" />
        <element name="comment" type="string" minOccurs="0" />

        <!-- Filename. (Using underscores, not spaces. No 'File:' namespace
marker.) -->
        <element name="filename" type="string" />

        <!-- URI at which this resource can be obtained -->
        <element name="src" type="anyURI" />

        <element name="size" type="positiveInteger" />

        <!-- TODO: add other metadata fields -->
    </sequence>

```

```
</complexType>

<!-- Discussion threading data for LiquidThreads -->
<complexType name="DiscussionThreadingInfo">
  <sequence>
    <element name="ThreadSubject" type="string" />
    <element name="ThreadParent" type="positiveInteger" />
    <element name="ThreadAncestor" type="positiveInteger" />
    <element name="ThreadPage" type="string" />
    <element name="ThreadID" type="positiveInteger" />
    <element name="ThreadAuthor" type="string" />
    <element name="ThreadEditStatus" type="string" />
    <element name="ThreadType" type="string" />
  </sequence>
</complexType>
</schema>
```