Action Plan Background: XML Document Type Definition 1.0 Files

Author: Andrea Goethals, FCLA
Release Date: 07/10/03

Change History:
02/01/2008 Added section 2.2 “Internal Technical Metadata”.

Preface:

Document Type Definition (DTD) files exist for both SGML and XML. There are enough significant differences between them to treat them as separate file formats. This report only applies to XML DTDs; a separate report will cover SGML DTDs if needed. In the FCLA Digital Archive (FDA), the terms 'XML DTDs' and 'SGML DTDs' will be used to distinguish between them. For example, a DTD for HTML documents is an SGML DTD; a DTD for XHTML documents is an XML DTD.

XML DTDs can be internal or external to XML documents. This report only applies to external DTDs. Internal DTDs are covered in the Action Plan Background Report for XML 1.0 (available on the FDA website).

There is no W3C Recommendation pertaining only to XML DTDs, therefore there is no real versioning of XML DTD file formats. The description of XML DTD files (aka 'DTD external subsets') is included in the W3C XML 1.0 Recommendation [Bray et. al., 2000]. For convenience in the FDA, an XML DTD will be labeled with the version of the W3C XML Recommendation that it is described in. For instance, a file in the XML DTD format described in the XML 1.0 Recommendation will be recorded in the FDA as an XML DTD 1.0 file.

In some cases registered MIME type/subtypes are not useful to digital archives. There are two registered MIME types for XML external parsed entities (text/xml-external-parsed-entity and application/xml-external-parsed-entity), but these MIME type/subtypes will not be used in the archive for several reasons. Given a text-based file, there is no way that a computerized program can determine if it is this MIME type because XML external parsed entities can be plain text, DTDs, XML or any other text-based format. XML external parsed entities are only recognizable as that format by following a chain of references starting from an XML document or from an XML schema. Instead, the archive program will label the file using its actual file format (plain text, DTD, XML, etc.). This means that what the archive calls a DTD could be called an external parsed entity file in some cases. This possibility exists for other text-based formats as well.

1 General Description

1.1 Format Name: XML Document Type Definition (XML DTD)

1.2 Version: 1.0
See the *Preface* in this report for a discussion of the version.

1.3 MIME media type name: application

1.4 MIME subtype: xml-dtd

1.5 Short Description: A textual document that describes the grammar for a particular set of XML documents.
1.6 Common Extensions: .dtd, .ent (for files containing only entity declarations)

1.7 Color depth: Not applicable

1.8 Color Space: Not applicable

1.9 Compression: Not applicable
1.10 Progressive Display: Not applicable

1.11 Animation: Not applicable

1.12 Magic number(s):

[Murata et.al., 2001] says that DTD files can be identified in the same way as XML documents. That is, there is no surefire identification method, but UTF-8 encoded documents usually start with an XML declaration ("<?xml"), and UTF-16 encoded documents usually start with the BOM (Byte Order Mark) followed by an XML declaration. See the XML 1.0 Action Plan Background Report, available on the FCLA Digital Archive website, for further details.

In practice, this is rarely followed for DTDs. The XML 1.0 Recommendation says that the XML declaration is optional for external DTDs, and only implies using the BOM with external DTDs that use UTF-16 encoding. The W3C's DTD for XML Schema at http://www.w3.org/2001/XMLSchema.dtd does not include the XML declaration. An examination of several books and websites on DTDs shows that the usual practice for external DTDs is to just include what would be found between the '[' and ']' characters of an XML document's Document Type Declaration that has an internal DTD. Figures 1 and 2 illustrate this.

To be able to recognize a DTD as a DTD, such as the one shown in Figure 2, an automated program would have to rely on a combination of the file extension and checking for correct markup syntax. It must not rely on nor preclude the existence of an XML declaration or the BOM. Without an XML declaration or the BOM, however, the character encoding of a DTD becomes more difficult to determine. According to the XML 1.0 Recommendation, XML processors should be able to read UTF-8 and UTF-16 encoded external entities, and external DTDs are a special type of external entity. Without an XML declaration specifying a particular encoding, and without the BOM for UTF-16 encoding, DTDs should be assumed to be in UTF-8 encoding. The XML 1.0 Recommendation states that it is a 'fatal error' for documents not to be in UTF-8 encoding if they do not specify that they are in a different encoding nor provide the BOM for UTF-16 encoding. 'Smart' DTD processors could be made to recognize valid UTF-bytes for strings like 'ELEMENT', 'ATTLIST', etc, to make sure that the document really is in UTF-8

encoding. Note that if the file is really in ASCII encoding it will still be compatible with UTF-8 encoding.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE greeting [
     <!ELEMENT greeting (#PCDATA)>
]>
<greeting>Hello, world!</greeting>
```

Figure 1: A simple XML document, hello.xml, with an internal DTD.

```
<!ELEMENT greeting (#PCDATA)>
```

Figure 2: The typical contents of an external DTD created for hello.xml. Another common practice is to begin the file with comments.

2 Content and Features

2.1 Essential and Distinguishing Characteristics

DTD files are text files that describe the grammar of a particular set of XML documents. To associate a DTD file with an XML document, the XML document refers to the external DTD in a Document Type Declaration (see glossary). The DTD can be used by an XML processor to 'validate' the XML document, which means to check that the XML document complies with the constraints declared in the DTD. A DTD can be composed of element, attribute, entity and notation declarations, parameter-entity references, general entity references, processing instructions and comments.

2.2 Internal Technical Metadata

XML DTD only extract text-related technical metadata. Please refer to the Plain Text Action Plan Background document for the list of extracted technical metadata.

3 Usefulness

3.1 Version Duration: Same as XML 1.0: 5 years, 5 months and continuing

3.2 History of Prior Versions Duration:

This is the first 'version' of XML DTD. As noted in the *Preface* of this report, XML DTDs do not really have versions as there are no stand-alone specifications for them. As a matter of convenience in the FDA, XML DTD versions will parallel the XML versions as long as the W3C describes them together in the same document.

3.3 Expected Newer Versions:

XML 1.1 is currently a W3C Candidate Recommendation. No changes are made to XML DTDs in the 1.1 document.

3.4 Existence of Publicly Available Complete Specifications:

The W3C XML 1.0 Recommendation is freely downloadable from the W3C website at http://www.w3.org/TR/REC-xml.

3.5 Specifications-controlling Body:

The World Wide Web Consortium (W3C), which is composed of a 'Team' (about 60 researchers and engineers, most of whom work at the US MIT, the French ERCIM headquarters, or the Japanese Keio University), 'Members' (408 corporations and institutions), a Technical Architecture Group (TAG), an Advisory Board, and a director (Tim Berners-Lee).

3.6 Related Legal Issues:

Same as for XML 1.0 - see section *3.6 Related Legal Issues* in the "Action Plan Background: XML 1.0" on the FCLA Digital Archive website.

3.7 Application and Platform Support:

There are many commercial and free XML DTD software tools, available on all major operating systems. Websites listing DTD software include:

- "DTD Software", O'Reilly XML.com website, http://www.xml.com/pub/rg/DTD_Software
- "Publicly Available Software for SGML/XML/DSSSL", OASIS Cover pages website, http://xml.coverpages.org/publicSW.html
- "Free XML tools and software", by Lars Marius Garshol, http://www.garshol.priv.no/download/xmltools/
- "DTD/Schema Editors and Tools", PerfectXML website, http://www.perfectxml.com/soft.asp?cat=11

3.8 Limitations:

Like XML documents, XML DTD files can link to external files. Unlike XML documents, all of these links can be automatically recognized by a computer program, because the XML DTD format is not extensible. There are three locations in an XML DTD that a link to an external file can be found: within a general entity declaration, within a parameter entity declaration, and within a notation declaration.

Within a general entity declaration. These declarations have the syntax, where externalID is the URI of an external reference:

```
<!ENTITY entityName SYSTEM externalID> or
<!ENTITY entityName PUBLIC publicID externalID>
```

Within a parameter entity declaration. These declarations have the syntax, where externalID is the URI of an external reference:

```
<!ENTITY % entityName SYSTEM externalID> or
<!ENTITY % entityName PUBLIC publicID externalID>
```

Within a notation declaration. These declarations have the syntax, where externalID is the URI of an external reference:

```
<!NOTATION notationName SYSTEM externalID> or
<!NOTATION notationName PUBLIC publicID externalID> or
<!NOTATION notationName PUBLIC publicID>
```

Notice that using the third notation declaration syntax, the URI of the file can not be known unless the public identifier publicID can be used to generate the URI of the file.

Although the externalID is supposed to be a valid URI according to the recommendation, in practice externalID is sometimes used to specify a program or mime type. This means that a computer program has to expect that the externalID might not point to a valid URI. Some examples of this use include:

```
<!NOTATION pl SYSTEM "/usr/bin/perl">
<!NOTATION PCL SYSTEM “PCL Language”>
<!NOTATION gif SYSTEM "image/gif">
<!NOTATION GIF87a PUBLIC “-//CompuServe//NOTATION Graphics
Interchange Format 87a//EN”>
```

3.9 Perceived Popularity:

When the first edition of the XML 1.0 Recommendation was published in February 1998, DTDs were the only schema that could be used to validate XML documents. A little over three years later, May 2001, the W3C published the XML Schema 1.0 Recommendation, which describes an alternative to XML DTDs. XML Schema has some advantages over XML DTDs. XML Schema is written in XML, so some of the same tools used for XML documents can be used for XML Schema. XML Schema has better data typing, supports namespaces and allows inheritance for elements and attributes [Lee & Chu, 2000]. However, XML DTDs have some advantages over XML Schema. Despite its non-XML syntax, it is compact, well understood, less complex and has more tool support than W3C XML Schema. An examination of the xml-dev mailing list at xml.org shows that a frequently criticized feature of the W3C XML Schema is the PSVI (Post Schema Validation Infoset). As a result of validation by a W3C XML Schema, an augmented infoset (the PSVI) with default values and data types is produced. The central criticism of PSVI seems to be that the augmented infoset should not be so tightly coupled with validation. 'Thin clients' may want the validation but not the extra information. Some applications may want information like default values but may not want to validate.

In a 2002 study of 190,417 XML documents on 19,254 separate websites, [Mignet et.al, 2003] found that 48% of the XML documents used external DTDs, while 9% used W3C XML Schemas. A year has passed since the study and it is not known whether W3C XML Schemas have grown in popularity relative to XML DTDs.

While it had been assumed that W3C XML Schema would eventually overtake DTDs in popularity, there are other XML schema languages contending for that role. [van der Vlist, 2002] groups XML schema languages into three categories: rule based languages (XSLT, Schematron), grammar based languages (Relax NG) and object oriented languages (W3C XML Schema). [Lee & Chu, 2000], [van der Vlist, 2001], and [Walsh, 2002] describe multiple XML schema

languages. There are several XML validators that work with multiple XML schema languages including Sun's Multi-Schema XML Validator (MSV) and James Clark's Jing.

There is an ISO/IEC effort currently underway called 'Document Schema Definition Languages', (DSDL), to "create a framework within which multiple validation tasks of different types can be applied to an XML document in order to achieve more complete validation results than just the application of a single technology" [Hohman, 2002]. There are very few DSDL documents currently available to the general public, but the documents that are available show that the framework will at least include RELAX NG, Schematron and XML DTDs. The DSDL public mailing list reveals that they are working on a new specification for XML DTDs that adds to XML DTDs some of the functionality that they have been lacking - data types and namespaces. This effort along with the trend of XML processors supporting multiple XML schema languages indicates that the field of XML schema languages is widening rather than narrowing.

4 Related Formats

4.1 Specification Variations: N/A

4.2 Other Specifications used by this Format: N/A

5 Summary and Conclusions

XML DTDs are popular despite their strange syntax and design limitations. While the W3C published the XML Schema Recommendation in May 2001, it is still not evident that they will replace XML DTDs in popularity. The field is still too young to determine.

Like XML documents, XML DTDs can link to external files. If technically and legally possible, care should be taken by a digital archive to preserve these external files along with the DTD. In addition, if the name of the linked-to file changes as a result of action taken by the archive, the name of the linked-to file will have to be changed within the DTD.

6 References

[Allen, 2001] Allen, Mark. "Document Type Declaration Reference", Version 0.5.0, February 11, 2001.
http://www.comptechdoc.org/independent/web/dtd/

[Bray et. al., 2000] Bray, Tim; Jean Paoli, C. M. Sperberg-McQueen, and Eve Maler. "Extensible Markup Language (XML) 1.0 (Second Edition)", W3C Recommendation; October 6, 2000.
http://www.w3.org/TR/2000/REC-xml-20001006

[Clark, 1997] Clark, James. "A Comparison of SGML and XML", W3C Note, December 15, 1997.
http://www.w3.org/TR/NOTE-sgml-xml.html

[Cowan, 2002] Cowan, John. “Extensible Markup Language (XML) 1.1”, W3C Candidate Recommendation. October 15, 2002.
http://www.w3.org/TR/2002/CR-xml11-20021015/

[Hohman, 2002] Hohman, G. Ken. “Homepage for documents Schema Definition Languages”, June 10, 2002.
http://www.dsdl.org/

[Holzner, 2001] Holzner, Steven. “Inside XML”, New Riders Publishing, Indianapolis, IN; 2001.

[Kennedy, 2002] Kennedy, Diane. “From SGML to XML; From DTD to Schema; Making the Choice”, March 2003.
http://www.idealliance.org/papers/xml02/dx_xml02/papers/03-01-03/03-01-03.html#comparison

[Lee & Chu, 2000] Lee, Dongwon; and Wesley W. Chu. “Comparative Analysis of Six XML Schema Languages”, ACM SIGMOD Record 29(3), September 2000.
http://www.cobase.cs.ucla.edu/tech-docs/dongwon/ucla-200008.html

[Mignet et.al., 2003] Mignet, Laurent; Denilson Barbosa, and Pierangelo Veltri. “The XML Web: A First Study”, WWW2003; Budapest, Hungary; May 20-24, 2003.
http://www.cs.toronto.edu/~mignet/Publications/www2003.pdf

[Murata et.al., 2001] Murata, M.; S. St. Laurent, and D. Kohn. “XML Media Types”, Request for Comments: 3023, The Internet Corporation for Assigned Names and Numbers (IANA), January 2001.
http://www.rfc-editor.org/rfc/rfc3023.txt

[St. Laurent, 1999] St. Laurent, Simon. “XML: A Primer”, Second Edition, M&T Books, Foster City, CA; 1999.

[van der Vlist, 2001] van der Vlist, Eric. “Comparing XML Schema Languages”, O'Reilly XML.com website, December 12, 2001.
http://www.xml.com/pub/a/2001/12/12/schemacompare.html

[van der Vlist, 2002] van der Vlist, Eric. “XML schema languages tutorial”, XML Europe 2002 Conference & Exposition, Barcelona, Spain; May 20-23 2002.

[Walsh, 1998] Walsh, Norman. “Converting an SGML DTD to a XML”, O'Reilly XML.com website, July 8, 1998.
http://www.xml.com/pub/a/98/07/dtd/index.html

[Walsh, 2002] Walsh, Norman. “XML Schema Languages: Why and How to Validate Your XML Documents”, Features section, Linux Magazine, February 2002.

http://www.linux-mag.com/2002-02/xml_01.html

7 Glossary of Terms

Attribute declaration - Found within DTDs; used to describe one or more valid attributes for a particular element. The general syntax of an attribute declaration is:

```
<!ATTLIST ELEMENT_NAME
     ATTRIBUTE_NAME TYPE DEFAULT_VALUE
     ATTRIBUTE_NAME TYPE DEFAULT_VALUE
     ATTRIBUTE_NAME TYPE DEFAULT_VALUE
     ...
     ATTRIBUTE_NAME TYPE DEFAULT_VALUE>
```

Document type declaration - Found within XML documents; used to declare a DTD internal subset, or to refer to the URI of an DTD external subset, or to do both. The URI can point to the same file system as the XML document or may be a URL. The general syntax of a document type declaration is:

```
<!DOCTYPE ROOT_ELEMENT_NAME ( [DTD] |
                              “SYSTEM” URI |
                              “SYSTEM” URI [DTD] |
                              “PUBLIC” IDENTIFIER URL |
                              “PUBLIC” IDENTIFIER URL [DTD])
```

Element declaration - Found within DTDs; used to describe valid grammar for elements that can be used by XML documents. The general syntax of an element declaration is: `<!ELEMENT NAME CONTENT_MODEL>`, where 'NAME' is the element name and 'CONTENT_MODEL' can be 'ANY', 'EMPTY', child elements or mixed content.

General entity declaration - Found within DTDs; used to associate names with definitions so that the definitions can be referred to using the name within an XML document or a DTD in the form of an entity reference. The general syntax for an entity declaration is: `<!ENTITY NAME DEFINITION>`
The definition can be text, or a URI of an external file containing the text.

Entity reference (or general entity reference) - Can be used within XML documents or DTDs; refers to the contents of a named entity; uses the '&' and ';' characters as delimiters.

Parameter entity declaration - Found within external DTDs or within an XML document's internal DTD; used to associate names with definitions so that the definitions can be referred to using the name elsewhere within the DTD in the form of a parameter entity reference. The general syntax for a parameter entity declaration is:
`<!ENTITY % NAME DEFINITION>`
The definition can be text or a URI of an external file containing the text.

Parameter entity reference - Can be used within the declarations of external DTDs or in limited positions within an XML document's internal DTD; uses the '%' and ';' characters as delimiters; is a reference to the value of a parameter entity declaration declared earlier in the DTD.