Bijlage 28
Comparison of SGML and XML

James Clark
E-mail: jjc@jclark.com

abstract
This document provides a detailed comparison of SGML (ISO 8879) and XML.

status of this document
This document is a NOTE made available by the W3 Consortium for discussion only. This indicates no endorsement of its content, nor that the Consortium has, is, or will be allocating any resources to the issues addressed by the NOTE. Errors or omissions in this document should be reported to the author.

1 Differences Between XML and SGML

XML allows only documents that use the SGML declaration in this note. This declares all the following SGML features as NO:

- DATATAG
- OMITTAG
- RANK
- LINK (SIMPLE, IMPLICIT and EXPLICIT)
- CONCUR
- SUBDOC
- FORMAL

Note that it differs from the reference concrete syntax in a number of ways:

- It also declares no short reference delimiters; it follows that SHORTREF and USEMAP declarations cannot occur in XML
- The PIC (processing instruction close) delimiter is ?>
- Quantities and capacities are effectively unlimited
- Names are case sensitive (NAMECASE GENERAL is NO)
- Underscore and colon are allowed in names
- Names can use Unicode characters and are not restricted to ASCII

The following constructs which are permitted in SGML when SHORTTAG is YES are not allowed in XML:

- Unclosed start-tags
- Unclosed end-tags
- Empty start-tags
- Empty end-tags
- Attribute values in attribute specifications entered directly rather than as literals
- Attribute specifications that omit the attribute name

NET delimiters can be used only to close an empty element. In SGML without the Web SGML Adaptations Annex, the NET delimiter is declared as />. With this approach, XML is not allowing null end-tags and is allowing net-enabling start-tags only for elements with no end-tag. In SGML with the Web SGML Adaptations Annex, there is a separate NESTC (net-enabling start tag close) delimiter. This allows the XML <e/> syntax to be handled as a combination of a net-enabling start-tag <e/ and a null end-tag >. With this approach, XML is allowing a net-enabling start-tag only when immediately followed by a null end-tag.

XML imposes the following restrictions not in SGML:

- Entity references
  - Entity references must be closed with a REFC delimiter
  - References to external data entities in content are not allowed
  - General entity references in content are required to be synchronous
  - External entity references in attribute values are not allowed
  - Parameter entity references are allowed in the internal subset only within a declaration separator (that is, at a point where a markup declaration could occur)
- Character references
  - Character references must be closed with a REFC delimiter
  - Named character references are not allowed
  - Numeric character references to non-SGML characters are not allowed
- Entity declarations
  - A #DEFAULT entity cannot be declared
  - External SDATA entities are not allowed

World Wide Web Consortium Note 15-December-1997, NOTE-sgml-xml-971215, available from http://www.w3.org/TR/NOTE-sgml-xml-971215.
Section 3, 'SGML declaration for XML', is not included here; see the Web document.

  - External CDATA entities are not allowed
  - Internal SDATA entities are not allowed
  - Internal CDATA entities are not allowed
  - PI entities are not allowed
  - Bracketed text entities are not allowed
  - External identifiers must include a system identifier
  - Attributes cannot be specified for an entity
  - The replacement text of general text entities and external parameter entities is required to be well-formed
  - An ampersand in a parameter literal must be followed by a syntactically valid entity reference or numeric character reference
- Attribute definition list declarations
  - Associated element type in attribute definition list declarations cannot be a name group
  - Attributes cannot be declared for a notation
  - CURRENT attributes are not allowed
  - Content reference attributes are not allowed
  - NUTOKEN(S) declared values are not allowed
  - NUMBER(S) declared values are not allowed
  - NAME(S) declared values are not allowed
  - A name token group must use the or connector
  - Attribute values specified as defaults in attribute definition list declarations must be literals (SGML allows them not to be even when SHORTTAG is NO)
- Element type declarations
  - Associated element type in element type declaration cannot be a name group
  - In an element declaration, a generic identifier cannot be specified as a rank stem and rank suffix (SGML allows this even when the RANK feature is NO)
  - Minimization parameters in element declarations are not allowed
  - RCDATA declared content are not allowed
  - CDATA declared content are not allowed
  - Content models cannot use the and connector
  - Content models for mixed content have a restricted form
  - Inclusions are not allowed
  - Exclusions are not allowed
- Comments
  - A parameter separator cannot contain comments; this means that markup declarations (other than comment declarations) cannot contain comments
  - Empty comment declarations (<!> in the reference concrete syntax) are not allowed
  - A comment declaration cannot contain more than one comment
  - In a comment declaration, an S separator is not allowed before the final MDC
- Processing instructions
  - Processing instructions must start with a name (the PI target)
  - A processing instruction whose PI target is xml can only occur at the beginning of a external entity and must be an XML declaration if it occurs in the document entity, and otherwise an text declaration
  - A PI target must not match [Xx][Mm][Ll] unless it is xml
- Marked sections
  - In marked section declarations, TEMP status keyword is not allowed
  - RCDATA marked sections are not allowed
  - INCLUDE/IGNORE marked sections are not allowed in the document instance
  - In a marked section declaration, a status keyword specification that contains no status keywords is not allowed
  - In a marked section declaration, a status keyword specification cannot contain more than one status keyword
  - Marked sections are not allowed in the internal subset
  - Parameter separators are not allowed in status keyword specifications in the document instance; in particular, parameter entity references are not allowed
- Other
  - Names beginning with [Xx][Mm][Ll] are reserved
  - The SGML declaration must be implied and cannot be explicitly present in the document entity
  - When < and & occur as data, they must be entered as < and &
  - A parameter separator required by the formal syntax must always be present and cannot be omitted when it is adjacent to a delimiter

XML predefines the semantics of the attributes xml:space and xml:lang. It also reserves all attribute, element type and notation names beginning with [Xx][Mm][Ll].

XML requires that an SGML parser use an entity manager that behaves as follows:

- Lines are terminated by newline (Unicode code #X000A) rather than being delimited by RS and RE as with a typical SGML entity manager
- System identifiers are treated as URLs
- The entity manager must support entities encoded in UTF-16 and UTF-8, and must be able automatically to detect which encoding an entity uses based on the presence of the byte order mark
- The entity manager should be able to recognize the encoding declaration in the XML declaration and

encoding PI and use it to determine the encoding of entity

XML imposes requirements on the information that a parser must make available to an application.

XML depends on the following changes to SGML made by Web SGML Adaptations Annex:

- HCRO delimiter (for hex numeric character references); for XML this is &#x
- EMPTYNRM feature that allows elements declared EMPTY to have end-tags
- NESTC delimiter
- Duplicate enumerated attribute tokens are allowed
- Relaxation of rules on use of parameter entity references inside groups
- Multiple ATTLIST declarations for a single element type
- ATTLIST declarations which don't declare any attributes
- KEEPRSRE feature that turns off SGML's rules for ignoring RSs and REs
- Fully-tagged SGML documents; a document that is fully-tagged but not type-valid is a conforming SGML document; this makes all XML documents, including those that are well-formed but not valid, conforming SGML documents
- Predefined data character entities in the SGML declaration (for lt, amp and so on)
- Unlimited capacities and quantities

The Web SGML Adaptations Annex also enables some XML restrictions to be enforced in SGML:

- SHORTTAG is unbundled, so the SGML declaration can allow attribute defaulting and NET without allowing other SHORTTAG constructs
- The SGML declaration can assert that a document is integrally stored, which disallows improperly nested entity references in content

2 Transforming SGML to XML

For most restrictions in XML that go beyond SGML, it is possible to transform an SGML document automatically into a document that meets the restrictions, and is equivalent in the sense that it has the same ESIS. There are a number of restrictions for which this is not the case:

External SDATA entities, external CDATA entities These could be transformed into NDATA entities.

Subdocument entities These could be converted into NDATA entities with a notation that indicates that they are SGML or XML.

References to external data entities in content These could be transformed into an empty element with an attribute whose declared value is ENTITY.

Data attributes Since an external data entity can only be used in an ENTITY or ENTITIES attribute on an element, these could be transformed into other attributes on the element.

Internal SDATA entities References could be transformed into numeric character references to the appropriate Unicode character; if used in an entity or entities attribute, the entity will have to be made external.

Internal CDATA entities If used in an ENTITY or ENTITIES attribute, the entity will have to be made external (references to CDATA entities are not part of ESIS).

PI entities If they contain ?>, they cannot be converted into an XML PI. It could be an application convention that entity references are replaced in PIs. Also if they do not start with a name, they cannot be converted into a well-formed XML PI.

names An SGML document can have a concrete syntax which allows characters in names that XML does not allow in names.