



Published 1988-07-01

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION

Information processing — Text and office systems — Standard Generalized Markup Language (SGML) AMENDMENT 1

*Traitement de l'information — Systèmes bureautiques — Langage standard généralisé de balisage (SGML)
AMENDEMENT 1*

Amendment 1 to International Standard ISO 8879:1986 was prepared by Technical Committee ISO/TC 97, *Information processing systems*.

1 Introduction

This amendment enhances the technical content of ISO 8879. The purpose of these enhancements is to improve the expression of the design of SGML, not to change that design. Typographic errors and omissions are also corrected, clarifying notes are provided for unclear areas, and the following ambiguities are resolved (principal clauses affected are in parentheses):

- Entities may be parsed with respect to the base document type as well as active document and link types (4.2, 4.3, 7.1, 12.1.1)
- Data attributes (notation parameters) have the same syntax as other attributes (10.5.5, 11.3, 11.4)
- Link syntax allows element nesting (12.1.4, 12.2, 12.3) and general entities (9.4.4, 10.4.2, 12.1.4.1)
- Potential error conditions may be reported (15.4)
- Conditions under which start-tag omission is ambiguous are clarified (7.3.1.1)
- SHORTTAG and DATATAG have priority over OMITTAG (7.3.2, 7.4.1.1)
- Forms of mixed content in which treatment of separator characters could cause anomalous results are clarified (7.6.1) and warned against (11.2.4)
- A tokenized attribute value has no extraneous spaces (7.9.3, 7.9.4)
- CDATA and SCDATA (like NDATA) are data entities (4.7.5.1, 7.9.4.3, 10.5.3, 10.5.5)
- General entity name attribute, like other tokenized attributes, has a list form (7.9.4, 11.3.3)
- System declaration enhanced and clarified (15.6)
- General rules for delimiter recognition were modified to avoid conflict with existing exceptions (9.6, 9.6.1, Figure 3, 9.6.2, 9.6.6, 10.5.1, 11.2.2)

2 Changes to ISO 8879-1986

Changes are listed in order by page number and clause or sub-clause number. Page numbers are prefixed by the word "Page"; clause and sub-clause numbers are not prefixed. All notes are part of the corrected text, not part of the instructions for correction. Text appearing after a colon (:) is unquoted literal text to be inserted or deleted, as indicated.

UDC 681.3.06 : 655.53

Ref. No. ISO 8879:1986/A1:1988 (E)

Descriptors : data processing, text processing, information interchange, documentation, documents, logical structure, programming (computers), artificial languages, programming languages.

© International Organization for Standardization, 1988 •

Printed in Switzerland

Price based on 15 pages

Page 4

- 2 Change "formatted" to: imaged

Page 5

- 3 After the em dash in the reference to ISO 646 insert: ISO

- 4.2 Replace definition with:

A document type that the system has identified as being active.

NOTE — An SGML entity is parsed with respect to its active document types, if any, or if not, with respect to its base document type and any active link types.

- 4.3 Replace definition with:

A link process that the system has identified as being active.

- 4.8.1 Insert new definition after 4.8:

4.8.1 associated notation (name): A notation name associated with the subject of a markup declaration by its *associated notation name* parameter.

Page 8

- 4.67 Change "is its default" to: becomes its default

- 4.69 In the first sentence, change "link type definition" to: by a link process definition

- 4.72.1 Insert new definition after 4.72:

4.72.1 data attribute: An attribute of the data conforming to a particular data content notation.

NOTE — In most cases, the value of the data attributes must be known before the data can be interpreted in accordance with the notation.

Page 9

- 4.75 Delete both occurrences of: non-SGML

- 4.75.1 Insert new definition after 4.75:

4.75.1 data entity: An entity that was declared to be data and therefore is not parsed when referenced.

NOTE — There are three kinds: character data entity, specific character data entity, and non-SGML data entity.

Page 12

- 4.143 Change "is" to: can be

- 4.156 Renumber (and reorder) these sub-clauses as follows: 4.156 becomes 4.158, 4.158 becomes 4.157, and 4.157 becomes 4.156.

- 4.159.1 Insert new definition after 4.159:

4.159.1 initial link set: The link set that is current at the start of the document instance.

Page 15

- 4.225 Change "is" to: can be

Page 17

- 4.267 In the definition, insert before the colon: a semantic error (such as a generic identifier that does not identify an element type) or

- 4.267 In item b), change "group's" to: token's

- 4.267 Re-letter items e) and f), respectively, to f) and g), and insert new item e): an otherwise allowable omission of a tag that creates an ambiguity;

- 4.269 Replace period at end of note with: (see 5.1).

Page 21

- 5.2 Insert new paragraph before last paragraph:

Successive instances of a syntactic token are deemed to be repetitions of a repeatable token, where permissible, rather than instances of multiple tokens.

- 6.1 In [1], insert before "non-SGML": *character data entity* | *specific character data entity* |
 6.2 In [2], move the first "s*," to immediately after the equals sign and delete the second.
 6.2 In [3], delete: s*
 6.2 In item a), change "appropriate delimiter action is performed" to: ensuing markup is parsed (see item d, below)

Page 22

- 6.2 After item c), add new item d):

Within markup, a character is tested to see if it is part of a delimiter string, a separator, delimited literal data, or a token (see 9.6.6).

- 6.3 Change heading to: Data Entities

- 6.3 Insert new productions before [6]:

[5.1] character data entity = SGML character*, Ee

[5.2] specific character data entity
 = SGML character*, Ee

- 6.3 In the note, delete: non-SGML

- 7.1 Replace last sentence of first paragraph with: An SGML entity is parsed with respect to its base document type and any active link types unless one or more active document types are identified, in which case it is parsed only with respect to them.

- 7.1 Add to last paragraph:

There can be an active document type only if "CONCUR YES" is specified on the *SGML declaration*. The base document type can be active only if at least one other document type is active.

NOTE — Parsing with respect to the base document type alone is accomplished by not identifying an active document type.

There can be an active link type for an entity if it has at least one link process definition. The possibility of simultaneous active link types depends on the class of link type, as follows:

simple More than one simple link type can be active, in addition to any implicit and explicit links.

implicit Only one can be active.

explicit More than one can be active only if they form a chain of processes in which the source document type of the first is the base document type, the source document type of the second is the result document type of the first, etc. The last link type in the chain can be an implicit link.

- 7.2 In [10], replace the expression with: *base document element, other prolog**

- 7.2.1 Replace "definitions" with: declarations

- 7.3.1 Replace first paragraph with:

A tag can be omitted only as provided in this sub-sub-clause, and only if the omission would not create an ambiguity, and if "OMITTAG YES" is specified on the *SGML declaration*.

Page 23

- 7.3.1.1 Insert new paragraph after list:

It is ambiguous to omit the start-tag of an element whose content begins with a short reference string whose mapping is changed by the element's associated short reference map.

- 7.3.1.2 Delete: for a *document element*, or

- 7.3.1.2 Re-letter items a) and b), respectively, to b) and c), and insert new item a): by the end of an *SGML document entity* or *SGML subdocument entity*;

7.3.2 In the first sentence, change "the *end-tag* can be omitted for" to: data can serve as the *end-tag* of

7.3.2 Add new paragraph:

An element that has a data tag is not treated as having an omitted *end-tag*.

7.4.1.1 In [16], delete: *document type specification*, s*.

Page 24

7.4.1.1 Insert note after list:

NOTE — A *generic identifier specification* is implied for an *empty start-tag* prior to determining whether any tags were omitted before it.

7.4.1.3 In [18], delete: *document type specification*,

7.4.2 Change "resolution of references" to: interpretation of attribute value literals

7.5.1.1 In [21], delete: s*.

7.6 In [27], change "*link type*" to: *link set*

Page 25

7.6.1 Add note after second paragraph (after the list):

NOTE — The determination that an **RE** is data is made during recognition of markup. Markup recognition, which includes recognition of omitted tags, occurs before the determination of whether an **RE** can be ignored according to the above rules. This sequence produces intuitive results in the normal case, where data can occur anywhere in the *content* of an element, as in:

(quote | #PCDATA)*

In content models where the occurrence of data is restricted, however, as in

(x, #PCDATA)

the situations described in items a) and b) would be treated as errors during markup recognition. The use of such content models is normally unnecessary and therefore not normally recommended (see 11.2.4).

7.7 In the last paragraph, delete: or "EXPLICIT YES"

7.7.1 Delete sub-sub-clause.

7.7.2 Delete sub-sub-clause.

Page 26

7.8 Insert after "valid": In a *generic identifier specification*

7.8.1.1 After "specified for", insert: the start-tag of

7.9.3 Add to paragraph:

NOTE — Interpretation of an *attribute value literal* occurs as though the attribute were *character data*, regardless of its actual declared value.

An attribute value other than *character data* is tokenized by replacing a sequence of **SPACE** characters with a single **SPACE** character and ignoring leading or trailing **SPACE** characters.

NOTE — Tokenization is performed without regard to the original literal; for example, whether CDATA or SDATA entities were used is irrelevant.

Page 27

7.9.3.1 Replace paragraph with:

An *attribute value specification* can be an *attribute value* (that is, not an *attribute value literal*) only if it contains nothing but name characters and either:

- a) it occurs in an *attribute definition list*; or
- b) "SHORTTAG YES" is specified on the *SGML declaration*.

7.9.4 In [35], change "*general entity name*" to: *general entity name* | *general entity name list*

- 7.9.4 Insert new production after [35]:
 [35.1] general entity name list = *name list*
- 7.9.4 In [39], [40], [42], and [43], replace "s+" with: **SPACE**
- 7.9.4 Change "fourteen" to: fifteen
- 7.9.4.3 Replace sub-clause with:

The value of a *general entity name* attribute, and of each *name* in a *general entity name list*, must be the *name* of a data entity or an SGML subdocument entity. Such a name must be declared as though it occurred in a *general entity reference* with no *name group* specified (see 9.4.4.1).

- 7.9.4.5 In item a), change "for an" to: for a *general entity name list*,
- 7.9.4.5 Add new paragraph after last paragraph:

In a single *attribute specification list*, the total number of names in *general entity name* and *general entity name list* attribute values, whether defaulted or specified, cannot exceed the "GRPCNT" quantity.

Page 28

- 9.1 Add new paragraph and note after last paragraph:

A reference to a non-SGML data entity, PI entity, or SGML subdocument entity, is prohibited in *replaceable character data*. A reference to a CDATA entity or SDATA entity is permitted.

NOTE — An effect of this sub-clause is to require that an element or marked section that is declared to be *replaceable character data* must start and end in the same entity.

- 9.3 Add note:

NOTE — A *number* or *number token* is not a quantity but a character string, like a *name* or *name token*: therefore, "01" and "1", for example, are not equivalent.

Page 29

- 9.4.3 In item "b" of the note, change "must (and other elements should)" to: , or of a marked section that is ignored, must (and the content of other elements and marked sections should)
- 9.4.3 In item "c" of the note, replace everything up to the right parenthesis with: For other elements and marked sections, the *start-tag* and *end-tag* (or *marked section start* and *marked section end*
- 9.4.4 In [59] and [60], replace "*document type specification*" with: *name group*?
- 9.4.4 Replace paragraph with new text:

A *name group* can be specified only if "CONCUR YES", "SIMPLE YES", "IMPLICIT YES", or "EXPLICIT YES" is specified on the *SGML declaration*, and the reference does not occur within a *start-tag* or *prolog*.

NOTE — This requirement does not prohibit specifying a *name group* for a *parameter entity reference* within a *marked section start* in a document instance.

If a *name group* is specified, the entity reference is ignored unless a *name* in the *name group* is that of an active document type or link type.

9.4.4.1 Applicable Entity Declaration

NOTE — The following requirements apply to entity names used in the value of general entity name and general entity name list attributes, as well as to named entity references.

Before an entity *name* can be used, it must be declared by an *entity declaration* in all applicable DTDs, except that a general entity *name* that is undeclared in a particular DTD is treated as though it had been declared for the default entity if that DTD has one.

The applicable DTDs depend on the context in which the entity *name* occurs, as follows:

- a) Within a start-tag, all DTDs applicable to the tag.

NOTE — That is, either the base DTD or those of active document types named in the tag's *document type specification*.

- b) Elsewhere in a document instance, the DTDs of active document types or the source DTDs of active link types specified in a named entity reference's *name group*, or, for any name that is undeclared in such a DTD, or if no *name group* is specified, the base DTD.

NOTE — An effect of this requirement is that an entity declared in the base document type can be referenced in an instance of any document type in which no entity with the same name was defined, and in which, for a general entity, no default entity was defined.

- c) Within a DTD, the same DTD.
- d) Within a *result attribute specification*, the result DTD.
- e) Elsewhere in an LPD, the source DTD.

NOTE — An LPD is parsed as though entity declarations within it had occurred in its source DTD.

NOTES

- 1 An effect of these requirements is that a default entity declared in the base document type will be referenced by an undeclared general entity name in an instance of any document type that does not itself have a default entity.
- 2 If an entity declaration specifies a data content notation, the notation must be declared in the same DTD as the entity.

9.4.5 In the paragraph after the note, replace "*name character*" with: character that could occur in the reference

Page 30

9.5 In the last paragraph before the notes, insert before the period: in the context in which the replacement occurs.

9.6 Change the initial "A" to: Except as otherwise provided in this International Standard, a

9.6 Change "*translation-reference character set*" to: *syntax-reference character set*.

9.6 Add note and new paragraph:

NOTE — A named character reference can be used to enter the first character of a delimiter string or delimiter-in-context, but only the first character.

All characters of a delimiter string or delimiter-in-context (see 9.6.2) must occur in the same entity.

9.6.1 For "CON", add note at end of Meaning:

NOTE — Most delimiters will not be recognized when the *content* is *character data* or *replaceable character data*.

9.6.1 In "CXT" Meaning, change "DS" to: DSM

Page 31

Figure 3 Except for "DSC", change all "DS" in the Mode column to: DSM

Figure 3 For "DSC", add Constraint: ENT

Figure 3 For "ETAGO", add an additional Mode: TAG

Figure 3 For "MINUS", add Constraint: EX

Figure 3 For "MINUS", change Description of Role to: Exclusion

Figure 3 For "PERO", add an additional Mode: LIT

Figure 3 For "PLUS", add Constraint: EX

9.6.1 In the list, change "DS" to "DSM" and insert new entry before it:

DS Recognized in a declaration subset.

9.6.1 In the Meaning of "DSM" (originally "DS"), change "and" to: or

9.6.1 In the Meaning of "MD", delete the parenthesized phrase.

9.6.1 Add note after list:

NOTE — Recognition modes can nest. For example, when a markup declaration begins, the recognition mode becomes “MD”. If a **GRPO** occurs in the declaration, the mode will become “GRP”, and an **MDC** (for example) will not be recognizable. When a **GRPC** occurs, the “GRP” mode ends and the recognition mode again becomes “MD”.

9.6.2 In the Meaning of “NMS”, insert before “is specified”: , “SIMPLE YES”, “IMPLICIT YES”, or “EXPLICIT YES”,

9.6.2 Replace second paragraph with:

Other contextual constraints are:

9.6.2 Add to end of second list:

EX In “MD” mode, recognized only at the start of a delimiter-in-context in which it is followed by **grpo**; in “GRP” mode, no constraints.

ENT Recognized only in the same entity as the corresponding **dso**.

Page 32

9.6.6 In the first paragraph, replace each occurrence of “*name start character*” with: *name character*

Page 34

10.1.1 In the second sentence of the first paragraph, change “intervening” to: surrounding or intervening

Page 35

10.1.5 Add paragraph and note:

A *generic identifier* can be a valid *associated element type* whether or not it is specified as an *element type* in the document type definition.

NOTE — This provision makes it easier to use public definitions that name a large set of generic identifiers in conjunction with a more restrictive document type that does not permit all of them.

10.1.6 Add sentence to 1st paragraph: The generated *system identifier* must be the same for all uses of the *external identifier*, except in the case of a default entity, where it must be the same for all references with a given undeclared entity name.

10.1.6.2 Replace the words “whether specified or generated” with: added to that of its *public identifier* component

10.1.7 Replace period at end of paragraph with: , except that such a sequence is ignored if it occurs at the start or end of the *minimum literal*.

Page 36

10.2.2.2 Insert new production after [87]:

[87.1] ISO text description = *minimum data*

10.2.2.3 Move “two-character” before “language code”.

10.2.2.3 In the second sentence, change “must” to: should

10.2.2.4 In [89] and in the paragraph that follows it, change “*minimum literal*” to: *minimum data*

10.2.2.4 Change both occurrences of “must” to: should

10.2.2.4 Add to note 1:

The full set of 128 characters, which is not registered, should be designated by:

ESC 2/5 4/0

Page 37

10.2.2.4 In note 2, change “it uniquely identifies” to: it should uniquely identify

10.4.2 In [97], replace “*qualified status keyword* |” with a left parenthesis and insert a second right parenthesis before the first asterisk.

10.4.2 Delete productions [98] and [99].

- 10.4.2 In the definition for "IGNORE", replace "marked section declaration" with "*marked section start* or *marked section end*" and delete: , but its status keyword specification is ignored
- 10.4.2 Delete the two paragraphs that begin with "A *qualified status keyword*".

Page 38

- 10.4.2 Replace the first paragraph beginning "If the effective" with the following paragraph and note:
- If the effective status is "CDATA" or "RCDATA", a *marked section declaration* is terminated by the first *marked section end*.

NOTE — A nested *marked section declaration* within the CDATA or RCDATA marked section in the same entity could not occur, as the markup would not be recognized.

- 10.5.1 Insert paragraph after the definition of "DEFAULT":

The *pero* in production 104 is recognized as a delimiter in this context without a contextual constraint.

- 10.5.1.1 Replace "one character" with: the number of characters in a *pero* delimiter string
- 10.5.1.2 Delete the second sentence of the second paragraph.
- 10.5.2 Add note after second paragraph:

NOTE — An entity whose text includes a *parameter literal* is considered a single entity even if the literal contained entity references that were resolved when the literal was interpreted.

- 10.5.3 For "PI", delete "data" from the definition.
- 10.5.3 In note 1, change *character data* to: a data character
- 10.5.3 Change "*translation-reference character set*" to: syntax-reference character set
- 10.5.3 Add note 3:

No data content notation applies to an internal data entity that is the subject of an explicit content reference. To specify a notation, the entity must be declared as an external data entity.

Page 39

- 10.5.5 In [109], replace "NDATA" with: ("CDATA" | "NDATA" | "SDATA")
- 10.5.5 In [109], insert before the right parenthesis: , *data attribute specification?*
- 10.5.5 Add new items to list:

CDATA means the entity is a *character data entity*.

SDATA means the entity is a *specific character data entity*.

- 10.5.5 Add the text of the note to the preceding paragraph and delete the note.
- 10.5.5 In the penultimate paragraph, delete both occurrences of "non-SGML" in the first sentence, and replace the second sentence with: The referenced entities, and any data entities referenced within a nested structure of data entities, should be declared in the same document type definitions as the original data entity.

- 11.1 Add new paragraph:

An *Ee* or *parameter entity reference* cannot occur in a *document type declaration*, except within the *document type declaration subset*.

Page 40

- 11.2.1 In the first paragraph, change "whether directly or indirectly" to: and it cannot be the same as a *rank stem*, or as the *generic identifier* that is derived by appending the *rank suffix* of a *ranked element* to its *rank stem*.
- 11.2.2 Insert paragraph after the definition of "*minus*":

The *minus* in productions 123 and 124 is recognized as a delimiter in this context without a contextual constraint.

- 11.2.2 In last paragraph, insert after "has": a content reference attribute or
- 11.2.4 Replace the definition of "ANY" with:

means the *content model* is an optional and repeatable **or** group whose members are “#PCDATA” and all of the GIs specified as element types in the same document type definition.

Page 41

11.2.4 In first paragraph, insert note after first sentence:

NOTE — It is recommended that “#PCDATA” be used only when data characters are to be permitted anywhere in the *content* of the element; that is, in a *content model* where it is the sole token, or where **or** is the only connector used in any *model group*.

This recommendation is made because separator characters, which are recognized as separators in *element content*, are treated as data in *mixed content*. In content models where this recommendation is not followed, such as

(x, #PCDATA)

an **RE** occurring before the start-tag of “x” would (because it is data) imply the start of an “x” element (because an “x” is contextually required). The actual start-tag for “x” would then be treated as an error (because a second “x” is not permitted).

An equivalent for a non-recommended content model can normally be obtained by replacing “#PCDATA” with the GI of an element whose content is “#PCDATA” and both of whose tags can be omitted.

This recommendation should not be construed as deprecating the use of the data tag feature (see the note in 11.2.4.4), but care should be taken with separator characters.

11.2.4 In first paragraph, change “In either case, the” to: The

11.2.4 Add new paragraph and note:

A *generic identifier* can be a valid *element token* whether or not it is specified as an *element type* in the document type definition.

NOTE — This provision makes it easier to use public definitions that name a large set of generic identifiers in conjunction with a more restrictive document type that does not permit all of them.

11.2.4.1 Add new paragraph:

A *model group* with a single *content token* is regarded as a **seq** group.

11.2.4.3 Change “look-ahead.” to: looking ahead in the document instance. The priority rules stated earlier in 11.2.4 are not considered in determining whether a content model is ambiguous.

11.2.4.4 Add to the end of [133]: , *occurrence indicator*?

Page 42

11.2.4.5 In the second paragraph, insert “content” before “tokens” and change “model group” to: content model

11.3 In [141], change “associated element type” to: (*associated element type* | *associated notation name*)

11.3 Change “associated element type” to: individual associated element type

11.3.1 Add: NOTE -- Default values should not be considered in applying this rule.

11.3.2 Add paragraph and note:

Within a document type definition, the same *attribute name* should be used for all attributes having a *declared value* of “ID”.

NOTE — Use of a common name for all ID attributes emphasizes the requirement that an ID value must be unique within a document instance.

Page 43

11.3.3 In [145], change “ENTITY” to: “ENTITY” | “ENTITIES”

11.3.3 Add to list after “ENTITY”:

ENTITIES means the *attribute value* is a *general entity name list*.

11.3.4 In the note, change “name” to: name, general entity name list,

11.4 Delete the note.

Page 44

11.4 At end of sub-clause, add new sub-sub-clause:

11.4.1 Data Attributes

NOTE — Data attributes are defined for a data content notation. Values can be specified for them on the entity declarations for data entities conforming to the notation.

The *declared value* of a data attribute cannot be “ENTITY”, “ENTITIES”, “ID”, “IDREF”, “IDREFS”, or “NOTATION”.

“CURRENT” or “CONREF” cannot be specified for a data attribute.

11.4.1.1 Associated Notation Name

[149.1] associated notation name = *rni*, “NOTATION”, *ps* +, (*notation name* | *name group*)

where

NOTATION means the attributes being defined are data attributes.

Each *name* in the *name group* must be a *notation name* defined in the same document type definition in which this attribute definition list occurs, and not specified for any other attribute definition list. It need not have been defined prior to this declaration, but must be defined prior to a reference to an entity for which it was declared to be the notation.

11.4.1.2 Data Attribute Specification

[149.2] data attribute specification = *ps* +, *dso*, *attribute specification list*, *s**, *dsc*

The validity of a data *attribute specification list* is determined by the attribute definition list associated with the data content notation. The attribute definition list must have been declared prior to this declaration.

The *data attribute specification* must be omitted if its *attribute specification list* is empty.

12.1 Before the first left parenthesis in [154], insert: *ps* +,

12.1 Add new paragraph:

An *Ee* or *parameter entity reference* cannot occur in a *link type declaration*, except within the *link type declaration subset*.

Page 45

12.1.1 Before the second *rni* in [156], insert: *ps* +,

12.1.1 Insert new sub-sub-sub-clause after 12.1.1:

12.1.1.1 Limits

The number of simple link processes that can be active simultaneously cannot exceed the quantity specified for “SIMPLE” on the *link type features* parameter of the *SGML declaration*.

12.1.2 Before the *rni* in [157], insert: *ps* +,

12.1.2 In the last paragraph, change “name” to: . or another document type that is the last result document type in a chain of processes

12.1.3 Before the *result document type name* in [158], insert: *ps* +,

12.1.4 In [161], replace the expression with: (*link attribute set* | *link set declaration*)*, *ID link set declaration?*, (*link attribute set* | *link set declaration*)*

12.1.4.1 In the heading, delete: Parameter

12.1.4.1 Delete first sentence.

12.1.4.1 In the second sentence, change “end of the source document type declaration subset” to: start of the source document type declaration subset, but after the entity declarations of any preceding active link type declarations.

12.1.4.2 At the start of the last paragraph, insert: “CURRENT” or

12.2 In [163], replace the parenthesized expression and its plus suffix with: (*ps* +, *link rule*) +

12.2 After [163], add:

- 12.2 [163.1] link rule = *source element specification* | *explicit link rule*
Add to end of [164]:

| (*rni*, "INITIAL")

where

- 12.2 **INITIAL** identifies the link set that is current when the document instance begins.
12.2 In the first sentence, change "definition" to: declaration
12.2 After first paragraph, insert new paragraphs:

"#INITIAL" must be specified for exactly one *link set declaration* in a *link type declaration sub-set*.

If an implicit link is specified, a *link rule* must be a *source element specification*; if an explicit link is specified, it must be an *explicit link rule*.

Page 46

- 12.2 Delete the paragraph and note.

- 12.2.1 In [165], insert after the existing comma: (*ps* +, *rni*, "USELINK", *ps* +, (*link set name* | (*rni*, "EMPTY")))?, (*ps* +, *rni*, "POSTLINK", *ps* +, *link set specification*)?,

- 12.2.1 In [166], change *ps** to: *s**

- 12.2.1 After [166] add:

where

USELINK means that the specified link set will become the current link set when an element of an associated type becomes the current element.

NOTE — The "#USELINK" parameter acts like a link set use declaration that occurs immediately after the start-tag; that is, it affects the subelements of an element. It has no effect if an element is empty.

POSTLINK means that the specified link set will become the current link set when an element of an associated type ceases to be the current element.

NOTE — The "#POSTLINK" parameter acts like a link set use declaration that occurs immediately after the end-tag (or after the start-tag if the element is empty and has no end-tag); that is, it affects succeeding elements within the same containing element.

- 12.2.1 Replace first paragraph with new paragraphs and note:

The specified link set must be defined by a *link set declaration* in the same *link type declaration*.

An element type can be an *associated element type* in only one link rule in a link set, unless in each such link rule there is a *link attribute specification*.

NOTE — An application must be able to determine from the link attributes which rule applies to a given instance of the element type. For example, the application could define a "usage" attribute whose value is an expression that tests source attribute values and the state of processing; the rule would apply if the expression were true.

- 12.2.1 After the first sentence of the existing second paragraph, add: The attribute definition list must have been declared prior to this declaration.

- 12.2.2 Change the heading to: **12.2.2 Explicit Link Rule**

- 12.2.2 Insert new production before [167]:

[166.1] explicit link rule = (*source element specification*, *ps* +, *result element specification*)
| (*source element specification*, *ps* +, *rni*, "IMPLIED") | (*rni*, "IMPLIED", *ps* +, *result element specification*)

- 12.2.2 In [167], replace the expression with: *generic identifier*, *result attribute specification*?

- 12.2.2 In [168], change *ps** to: *s**

- 12.2.2 In the definition of "IMPLIED", change "*result element*" to: source or result element

- 12.2.2 Replace first paragraph with new paragraph:

12.2.3 “#IMPLIED” can be linked to a given result element type only once in a link set.
After 12.2.2 add:

12.2.3 ID Link Set Declaration

[168.1] ID link set declaration = *mdo*, “IDLINK”, (*ps* +, *name*, *ps* +, *link rule*) +, *ps**, *mdc*

The *name* should be the unique identifier of a source element. The corresponding *link rule* will apply to that element regardless of the current link set. The *associated element type* of the *link rule* must be that of the identified source element.

A *name* can be associated with only one *link rule* in an ID link set, unless in each such *link rule* there is a *link attribute specification*.

12.3 In [169], replace the parenthesized expression with: *link type name*

12.3 At end of [170] add: | (*rni*, “RESTORE”)

12.3 Insert new item at end of list:

RESTORE means the link set is that associated with the current element or, if there is none, the link set that was current when the current element began.

12.3.1 Delete the sub-sub-clause.

12.3.2 Delete the heading and the first sentence.

12.3.2 Add note after first paragraph:

NOTE — As an aid to revision of the document, a declaration such as

<!USELINK #RESTORE linktype>

should be specified at the point where the link set activated by a *link set use declaration* is no longer required.

12.3.2 Add new sentence to last paragraph:

If the *link type name* is not an active link type, the declaration is ignored.

12.3.3 Renumber the heading to: 12.4

12.3.3 In the last sentence, change “empty” to: initial

Page 47

13 In [171], replace “ISO 8879-1986” with: *minimum literal*

13 Insert new paragraph after [171]:

The *minimum data* of the *minimum literal* must be “ISO 8879:1986”.

13.1.1 In [173], replace the expression with: *base character set*, *ps* +, *described character set portion*, (*ps* +, *base character set*, *ps* +, *described character set portion*)*

Page 49

Figure 5 For “ATTCAP”, replace the period with: , plus NAMELEN for each occurrence (whether or not the attribute is specified) in a link set declaration of an element type associated with the definition, or, in an entity declaration, of a notation name associated with the definition.

Figure 5 For “ATTCHCAP”, replace the description with:

Character of normalized length of an attribute value defined as a default value, or explicitly specified (not defaulted) in a link set declaration or data attribute specification.

Figure 5 For “GRPCAP”, replace description with: Content token at any level of a content model (a data tag group is three tokens).

Figure 5 For “IDCAP”, delete the parenthesized phrase.

Figure 5 For “MAPCAP”, delete the parenthesized phrase and replace the period with: , plus, for each map declared, NAMELEN for each short reference delimiter in the concrete syntax (whether or not the delimiter is specified in the map).

13.4.1 In the second sentence of the first paragraph, change the first two occurrences of “character” to: character other than a **Digit**, **LC Letter**, or **UC Letter**

13.4.2 In the definition of CONTROLS, change “document” to: system