

Australian Standard[®]

**INFORMATION PROCESSING—
TEXT AND OFFICE SYSTEMS—
STANDARD GENERALIZED
MARKUP LANGUAGE (SGML)**

This Australian Standard was prepared by Committee IS/1, Information Processing Systems. It was approved on behalf of the Council of the Standards Association of Australia on 27 October 1987 and published on 1 December 1987.

The following interests are represented on Committee IS/1:

Australian Association of Permanent Building Societies:
Australian Banker's Association
Australian Bureau of Statistics
Australian Computer Equipment Manufacturers Association
Australian Computer Society
Australian Computer Users Association
Australian Computing Services Association
Australian Information Industry Association
CSIRO
CSIRONET
Department of Defence
Department of Industry, Technology and Commerce
Life Insurance Federation of Australia
Public Service Board, N.S.W.
Telecom Australia
Universities and Colleges

Review of Australian Standards. To keep abreast of progress in industry, Australian Standards are subject to periodic review and are kept up to date by the issue of amendments or new editions as necessary. It is important therefore that Standards users ensure that they are in possession of the latest edition, and any amendments thereto.

Full details of all Australian Standards and related publications will be found in the Standards Australia Catalogue of Publications; this information is supplemented each month by the magazine 'The Australian Standard', which subscribing members receive, and which gives details of new publications, new editions and amendments, and of withdrawn Standards.

Suggestions for improvements to Australian Standards, addressed to the head office of Standards Australia, are welcomed. Notification of any inaccuracy or ambiguity found in an Australian Standard should be made without delay in order that the matter may be investigated and appropriate action taken.

Australian Standard[®]

**INFORMATION PROCESSING—
TEXT AND OFFICE SYSTEMS—
STANDARD GENERALIZED
MARKUP LANGUAGE (SGML)**

First published as AS 3514 1987

Incorporating:
Amdt 1—1996

PUBLISHED BY STANDARDS AUSTRALIA
(STANDARDS ASSOCIATION OF AUSTRALIA)
1 THE CRESCENT, HOMEBUSH, NSW 2140

ISBN 0 7262 4786 3

PREFACE

This Standard was prepared by the Association's Committee on Information Processing Systems. It is identical with and has been reproduced from International Standard ISO 8879—1986; drawn up by ISO TC 97, Information Processing Systems.

This Standard:

- specifies an abstract syntax known as the Standard Generalized Markup Language (SGML). The language expresses the description of a document's structure and other attributes, as well as other information that makes the markup interpretable.
- specifies a reference concrete syntax that binds the abstract syntax to specific characters and numeric values, and criteria for defining variant concrete syntaxes.
- defines conforming documents in terms of their use of components of the language.
- defines conforming systems in terms of their ability to process conforming documents and to recognize markup errors in them.
- specifies how data not defined by this Standard (such as images, graphics, or formatted text) can be included in a conforming document.

The Standard is one of a series of Open Systems Interconnection (OSI) Standards which are currently under development or in the course of publication. Since OSI Standards are developmental, there may be some minor difficulties encountered in their implementation. For this reason, SAA will be providing a limited interpretation service to coordinate and disseminate information concerning difficulties which are identified in using this Standard.

For the purpose of this Australian Standard, the text of the ISO Standard given herein should be modified as follows:

- (a) *Terminology.* The words 'Australian Standard' should replace the words 'International Standard' wherever they appear.
- (b) *Cross-references.* The references to International Standards should be replaced by references to Australian Standards as follows:

<i>Reference to International Standard</i>	<i>Appropriate Australian Standard</i>
ISO 639, Codes for the representation of names of languages.	No Australian equivalent.
ISO 646, Information processing—7-bit coded character set for information interchange.	AS 1776, Information processing—7-bit coded character set for information interchange.
ISO 9069, Information processing SGML support facilities—SGML Document Interchange Format (SDIF).	No Australian equivalent.
ISO 9070, Information processing SGML support facilities—Registration procedures for public text.	No Australian equivalent.
ISO 2022, Information processing—ISO 7-bit and 8-bit coded character sets—Code extension techniques.	AS 1953, Information processing—ISO 7-bit and 8-bit coded character sets—Code extension techniques.
ISO 3166, Codes for the representation of names of countries.	AS 2632, Codes for the representation of names of countries.
ISO 4873, Information processing—ISO 8-bit code for information interchange Structure and rules for implementation.	No Australian equivalent.
ISO 6937, Information processing—Coded character sets for text communication.	AS 2793 series, Information processing—Coded character sets for text communication.

Reference to International Standard

ISO 8632/2, Information processing systems—Computer graphics—Metafile for the storage and transfer of picture description information—Part 2: Character encoding.

ISO 8632/4, Information processing systems—Computer graphics—Metafile for the storage and transfer of picture description information—Part 4: Clear text encoding.

Appropriate Australian Standard

No Australian equivalent.

No Australian equivalent.

© Copyright — STANDARDS AUSTRALIA

Users of Standards are reminded that copyright subsists in all Standards Australia publications and software. Except where the Copyright Act allows and except where provided for below no publications or software produced by Standards Australia may be reproduced, stored in a retrieval system in any form or transmitted by any means without prior permission in writing from Standards Australia. Permission may be conditional on an appropriate royalty payment. Requests for permission and information on commercial software royalties should be directed to the head office of Standards Australia.

Standards Australia will permit up to 10 percent of the technical content pages of a Standard to be copied for use exclusively in-house by purchasers of the Standard without payment of a royalty or advice to Standards Australia.

Standards Australia will also permit the inclusion of its copyright material in computer software programs for no royalty payment provided such programs are used exclusively in-house by the creators of the programs.

Care should be taken to ensure that material used is from the current edition of the Standard and that it is updated whenever the Standard is amended or revised. The number and date of the Standard should therefore be clearly identified.

The use of material in print form or in computer software programs to be used commercially, with or without payment, or in commercial contracts is subject to the payment of a royalty. This policy may be varied by Standards Australia at any time.

Contents

0	Introduction	13
0.1	Background	13
0.2	Objectives	14
0.3	Organization	15
1	Scope	16
2	Field of application	16
3	References	17
4	Definitions	17
5	Notation	32
5.1	Syntactic Tokens	33
5.2	Ordering and Selection Symbols	33
6	Entity Structure	33
6.1	SGML Document	33
6.2	SGML Entities	33
6.2.1	S Separator	34
6.2.2	Entity End	34
6.2.3	Implied SGML Declaration	34
6.3	Non-SGML Data Entity	34
7	Element Structure	34
7.1	Prolog	34
7.2	Document Element	34
7.2.1	Limits	34
7.3	Element	34a
7.3.1	Omitted Tag Minimization	34a
7.3.1.1	Start-tag Omission	34a
7.3.1.2	End-tag Omission	35
7.3.2	Data Tag Minimization	35
7.3.3	Quantities	35
7.4	Start-tag	35
7.4.1	Minimization	35
7.4.1.1	Empty Start-tag	35
7.4.1.2	Unclosed Start-tag	36
7.4.1.3	NET-enabling Start-tag	36
7.4.2	Quantities	36
7.5	End-tag	36
7.5.1	Minimization	36
7.5.1.1	Empty End-tag	36
7.5.1.2	Unclosed End-tag	36
7.5.1.3	Null End-tag	36
7.6	Content	36
7.6.1	Record Boundaries	37
7.7	Document Type Specification	37
7.8	Generic Identifier (GI) Specification	38
7.8.1	Rank Feature	38
7.8.1.1	Full Generic Identifier	38
7.8.1.2	Rank Stem	38
7.9	Attribute Specification List	38

7.9.1	Minimization	38
7.9.1.1	Omitted Attribute Specification	38
7.9.1.2	Omitted Attribute Name	38
7.9.2	Quantities	38
7.9.3	Attribute Value Specification	38
7.9.3.1	Minimization	39
7.9.4	Attribute Value	39
7.9.4.1	Syntactic Requirements	39
7.9.4.2	Fixed Attribute	39
7.9.4.3	General Entity Name	39
7.9.4.4	Notation	39
7.9.4.5	Quantities	39
8	Processing Instruction	39
8.1	Quantities	40
9	Common Constructs	40
9.1	Replaceable Character Data	40
9.2	Character Data	40
9.2.1	SGML Character	40
9.2.2	Function Character	40
9.3	Name	40
9.3.1	Quantities	40
9.4	Entity References	40
9.4.1	Quantities	40
9.4.2	Limits	40
9.4.3	Obfuscatory Entity References	41
9.4.4	Named Entity Reference	41
9.4.4.1	Applicable Entity Declaration	41
9.4.5	Reference End	41a
9.4.6	Short Reference	41a
9.4.6.1	Equivalent Reference String	41a
9.5	Character Reference	42
9.6	Delimiter Recognition	42
9.6.1	Recognition Modes	42
9.6.2	Contextual Constraints	43
9.6.3	Order of Recognition	44
9.6.4	Delimiters Starting with the Same Character	44
9.6.5	Short References with Blank Sequences	44
9.6.5.1	Quantities	44
9.6.6	Name Characters	44
9.7	Markup Suppression	44
9.8	Capacity	44
10	Markup Declarations: General	46
10.1	Parts of Declarations	46
10.1.1	Parameter Separator	46
10.1.2	Parameter Literal	46
10.1.2.1	Quantities	46
10.1.3	Group	46
10.1.3.1	Quantities	46
10.1.4	Declaration Separator	46
10.1.5	Associated Element Type	47
10.1.6	External Identifier	47
10.1.6.1	Quantities	47
10.1.6.2	Capacities	47
10.1.7	Minimum Literal	47
10.1.7.1	Quantities	47
10.2	Formal Public Identifier	47
10.2.1	Owner Identifier	47
10.2.1.1	ISO Owner Identifier	47
10.2.1.2	Registered Owner Identifier	47

10.2.1.3	Unregistered Owner Identifier	48
10.2.2	Text Identifier	48
10.2.2.1	Public Text Class	48
10.2.2.2	Public Text Description	48
10.2.2.3	Public Text Language	48
10.2.2.4	Public Text Designating Sequence	48
10.2.2.5	Public Text Display Version	49
10.3	Comment Declaration	49
10.4	Marked Section Declaration	49
10.4.1	Quantities	49
10.4.2	Status Keyword Specification	49
10.5	Entity Declaration	50
10.5.1	Entity Name	50
10.5.1.1	Quantities	50
10.5.1.2	Capacities	50
10.5.2	Entity Text	50
10.5.3	Data Text	50
10.5.4	Bracketed Text	51
10.5.4.1	Quantities	51
10.5.5	External Entity Specification	51
11	Markup Declarations: Document Type Definition	51
11.1	Document Type Declaration	51
11.2	Element Declaration	52
11.2.1	Element Type	52
11.2.1.1	Ranked Element	52
11.2.1.2	Quantities	52
11.2.2	Omitted Tag Minimization	52
11.2.3	Declared Content	52
11.2.4	Content Model	52
11.2.4.1	Connector	53
11.2.4.2	Occurrence Indicator	53
11.2.4.3	Ambiguous Content Model	53
11.2.4.4	Data Tag Group	53
11.2.4.5	Quantities	54
11.2.5	Exceptions	54
11.2.5.1	Inclusions	54
11.2.5.2	Exclusions	54
11.3	Attribute Definition List Declaration	54
11.3.1	Quantities	54
11.3.2	Attribute Name	54
11.3.3	Declared Value	55
11.3.4	Default Value	55
11.3.4.1	Quantities	55
11.3.4.2	Capacities	55
11.4	Notation Declaration	55
11.4.1	Data Attributes	55
11.4.1.1	Associated Notation Name	55a
11.4.1.2	Data Attribute Specification	55a
11.5	Short Reference Mapping Declaration	56
11.6	Short Reference Use Declaration	56
11.6.1	Use in Document Type Declaration	56
11.6.2	Use in Document Instance	56
11.6.3	Current Map	56
12	Markup Declarations: Link Process Definition	56
12.1	Link Type Declaration	56
12.1.1	Simple Link Specification	57
12.1.1.1	Limits	57
12.1.2	Implicit Link Specification	57
12.1.3	Explicit Link Specification	57
12.1.3.1	Limits	57
12.1.4	Link Type Declaration Subset	57

12.1.4.1	Parameter Entities	57
12.1.4.2	Link Attributes	57
12.1.4.3	Simple Link	57
12.2	Link Set Declaration	57
12.2.1	Source Element Specification	58
12.2.2	Result Element Specification	58
12.2.3	ID Link Set Declaration	58
12.3	Link Set Use Declaration	58
12.4	Current Link Set	58a
13	SGML Declaration	59
13.1	Document Character Set	59
13.1.1	Character Set Description	59
13.1.1.1	Base Character Set	59
13.1.1.2	Described Character Set Portion	59
13.1.2	Non-SGML Character Identification	60
13.2	Capacity Set	60
13.3	Concrete Syntax Scope	60
13.4	Concrete Syntax	61
13.4.1	Public Concrete Syntax	61
13.4.2	Shunned Character Number Identification	61
13.4.3	Syntax-reference Character Set	62
13.4.4	Function Character Identification	62
13.4.5	Naming Rules	62
13.4.6	Delimiter Set	63
13.4.6.1	General Delimiters	63
13.4.6.2	Short Reference Delimiters	63
13.4.7	Reserved Name Use	63
13.4.8	Quantity Set	63
13.5	Feature Use	64
13.5.1	Markup Minimization Features	64
13.5.2	Link Type Features	64
13.5.3	Other Features	64
13.6	Application-specific Information	65
14	Reference and Core Concrete Syntaxes	65
15	Conformance	65
15.1	Conforming SGML Document	65
15.1.1	Basic SGML Document	65
15.1.2	Minimal SGML Document	65
15.1.3	Variant Conforming SGML Document	65
15.2	Conforming SGML Application	65
15.2.1	Application Conventions	65
15.2.2	Conformance of Documents	65
15.2.3	Conformance of Documentation	65
15.3	Conforming SGML System	65
15.3.1	Conformance of Documentation	66
15.3.2	Conformance to System Declaration	66
15.3.3	Support for Reference Concrete Syntax	66
15.3.4	Support for Reference Capacity Set	67
15.3.5	Consistency of Parsing	67
15.3.6	Application Conventions	67
15.4	Validating SGML Parser	68
15.4.1	Error Recognition	68
15.4.2	Identification of SGML Messages	68
15.4.3	Content of SGML Messages	68
15.5	Documentation Requirements	68
15.5.1	Standard Identification	68
15.5.2	Identification of SGML Constructs	68
15.5.3	Terminology	69
15.5.4	Variant Concrete Syntax	69

15.6	System Declaration	69
15.6.1	Concrete Syntaxes Supported	69
15.6.1.1	Concrete Syntax Changes	69
15.6.1.2	Character Set Translation	69
15.6.2	Validation Services	70
15.6.3	SDIF Support	70

Annexes

A	Introduction to Generalized Markup	71
A.1	The Markup Process	71
A.2	Descriptive Markup	72
A.3	Rigorous Markup	74
A.4	Conclusion	76
A.5	Acknowledgments	77
A.6	Bibliography	77
B	Basic Concepts	78
B.1	Documents Document Type Definitions and Procedures	78
B.1.1	Documents	78
B.1.2	Document Type Definitions	78
B.1.3	Procedures	79
B.2	Markup	79
B.3	Distinguishing Markup from Text	80
B.3.1	Descriptive Markup Tags	80
B.3.2	Other Markup	81
B.3.3	Record Boundaries	81
B.3.3.1	Record Boundaries in Data	82
B.3.3.2	Record Boundaries in Markup	82
B.4	Document Structure	82
B.4.1	Document Type Definitions	82
B.4.2	Element Declarations	83
B.4.2.1	Content Models	83
B.4.2.2	Connectors and Occurrence Indicators	83
B.4.2.3	Entity References in Models	84
B.4.2.4	Name Groups	84
B.4.2.5	Data Characters	85
B.4.2.6	Empty Content	85
B.4.2.7	Non-SGML Data	85
B.4.2.8	Summary of Model Delimiters	86
B.5	Attributes	86
B.5.1	Specifying Attributes	86
B.5.1.1	Names	86
B.5.1.2	Attribute Values	87
B.5.2	Declaring Attributes	87
B.5.2.1	Attribute Definition Syntax	87
B.5.2.2	Complex Attribute Values	88
B.5.2.3	Name Token Groups	89
B.5.2.4	Changing Default Values	89
B.6	Entities	90
B.6.1	Entity Reference Syntax	90
B.6.2	Declaring Entities	90
B.6.2.1	Processing Instructions	91
B.6.2.2	Entities with Entity References	91
B.6.2.3	External Entities	91
B.6.2.4	Public Entities	92
B.7	Characters	92

B.7.1	Character Classification	92
B.7.2	Character References	93
B.7.3	Using Delimiter Characters as Data	94
B.8	Marked Sections	95
B.8.1	Ignoring a Marked Section	95
B.8.2	Versions of a Single Document	96
B.8.3	Unparsable Sections	96
B.8.4	Temporary Sections	97
B.8.5	Keyword Specification	97
B.8.6	Defining a Marked Section as an Entity	97
B.9	Unique Identifier Attributes	98
B.10	Content Reference Attributes	98
B.11	Content Model Exceptions	99
B.11.1	Included Elements	99
B.11.2	Excluded Elements	99
B.12	Document Type Declaration	100
B.13	Data Content	100
B.13.1	Data Content Representations	101
B.13.1.1	Character Data (PCDATA, CDATA, and RCDATA)	101
B.13.1.2	Non-SGML Data (NDATA)	101
B.13.2	Data Content Notations	102
B.13.2.1	Notations for Character Data	102
B.13.2.2	Notations for Non-SGML Data	103
B.13.2.3	Specifying Data Content Notations	103
B.14	Customizing	104
B.14.1	The SGML Declaration	104
B.14.1.1	Optional Features	104
B.14.1.2	Variant Concrete Syntax	104
B.14.2	Impact of Customization	104
B.15	Conformance	105
C	Additional Concepts	106
C.1	Markup Minimization Features	106
C.1.1	SHORTTAG: Tags With Omitted Markup	106
C.1.1.1	Unclosed Short Tags	107
C.1.1.2	Empty Tags	107
C.1.1.3	Attribute Minimization	107
C.1.2	OMITTAG: Tags May be Omitted	108
C.1.2.1	Tag Omission Concepts	109
C.1.2.2	Specifying Minimization	109
C.1.2.3	End-tag Omission: Intruding Start-tag	110
C.1.2.4	End-tag Omission: End-tag of Containing Element	110
C.1.2.5	Start-tag Omission: Contextually Required Element	111
C.1.2.6	Combination with Short Tag Minimization	111
C.1.2.7	Markup Minimization Considerations	111
C.1.3	SHORTREF: Short Reference Delimiters May Replace Complete Entity References	112
C.1.3.1	Typewriter Keyboarding: Generalized WYSIWYG	112
C.1.3.2	Typewriter Keyboarding Example: Defining a Short Reference Map	112
C.1.3.3	Typewriter Keyboarding Example: Activating a Short Reference Map	113
C.1.3.4	Tabular Matter Example	114
C.1.3.5	Special Requirements	115
C.1.4	DATATAG: Data May Also be a Tag	115
C.1.5	RANK: Ranks May be Omitted from Tags	118
C.2	LINK Features: SIMPLE, IMPLICIT, and EXPLICIT	119
C.2.1	Link Process Definitions	120
C.3	Other Features	120

C.3.1	CONCUR: Document Instances May Occur Concurrently	120
C.3.2	SUBDOC: Nested Subdocument Entities May Occur	121
C.3.3	FORMAL: Public Identifiers are Formal	121
D	Public Text	122
D.1	Element Sets	122
D.1.1	Common Element Types	122
D.1.2	Pro Forma Element Types	122
D.2	Data Content Notations	122
D.3	Variant Concrete Syntaxes	123
D.3.1	Multicode Concrete Syntaxes	123
D.4	Entity Sets	123
D.4.1	General Considerations	124
D.4.1.1	Format of Declarations	124
D.4.1.2	Corresponding Display Entity Sets	125
D.4.1.3	Entity Names	125
D.4.1.4	Organization of Entity Sets	126
D.4.2	Alphabetic Characters	126
D.4.2.1	Latin	126
D.4.2.2	Greek Alphabetic Characters	129
D.4.2.3	Cyrillic Alphabetic Characters	131
D.4.3	General Use	133
D.4.3.1	Numeric and Special Graphic Characters	133
D.4.3.2	Diacritical Mark Characters	135
D.4.3.3	Publishing Characters	135
D.4.3.4	Box and Line Drawing Characters	137
D.4.4	Technical Use	138
D.4.4.1	General	138
D.4.4.2	Greek Symbols	140
D.4.4.3	Alternative Greek Symbols	141
D.4.5	Additional Mathematical Symbols	142
D.4.5.1	Ordinary Symbols	142
D.4.5.2	Binary and Large Operators	142
D.4.5.3	Relations	143
D.4.5.4	Negated Relations	145
D.4.5.5	Arrow Relations	146
D.4.5.6	Opening and Closing Delimiters	147
E	Application Examples	148
E.1	Document Type Definition	148
E.2	Computer Graphics Metafile	152
E.3	Device-Independent Code Extension	152
E.3.1	Code Extension Facilities	152
E.3.1.1	Avoiding False Delimiter Recognition	153
E.3.1.2	Eliminating Device and Code Dependencies	155
F	Implementation Considerations	157
F.1	A Model of SGML Parsing	157
F.1.1	Physical Input	157
F.1.1.1	Entities	157
F.1.1.2	Record Boundaries	157
F.1.2	Recognition Modes	157
F.1.3	Markup Minimization	158
F.1.4	Translation	159
F.1.5	Commend Language Analogy	159

F.2	Initialization	159
F.2.1	Initial Procedure Mapping	159
F.2.2	Link Process Specification	159
F.2.3	Concurrent Document Instances	159
F.3	Dynamic Procedure Mapping	160
F.4	Error Handling	160
G	Conformance Classification and Certification	161
G.1	Classification Code	161
G.1.1	Feature Code	161
G.1.2	Validation Code	162
G.1.3	Syntax Code	163
G.2	Certification Considerations	163
H	Theoretical Basis for the SGML Content Model	164
H.1	Model Group Notation	164
H.2	Application of Automata Theory	164
H.3	Divergence from Automata Theory	165
I	Nonconforming Variations	166
I.1	Fixed-length Generic Identifiers	166
I.2	Single Delimiter	166

Figures

1	Character Classes: Abstract Syntax	41
2	Character Classes: Concrete Syntax	42
3	Reference Delimiter Set: General	43
4	Reference Delimiter Set: Short References	45
5	Reference Capacity Set	61
6	Reference Quantity Set	64
7	Reference Concrete Syntax	66
8	Typical SGML Declaration for Basic SGML Document	67
9	Element Markup	81
10	Start-tag with 2 Attributes	87
11	Multicode Basic Concrete Syntax	124
12	Graphics Metafile Attributes (1 of 2): Encoding and View	153
13	Graphics Metafile Attributes (2 of 2): Size and Rotation	154
14	Function Characters for Device-Independent Multicode Concrete Syntaxes	155
15	FSV Conformance Classification	162

Information Processing — Text and Office Systems — Standard Generalized Markup Language (SGML)

O Introduction

This International Standard specifies a language for document representation referred to as the “Standard Generalized Markup Language” (SGML). SGML can be used for publishing in its broadest definition, ranging from single medium conventional publishing to multi-media data base publishing. SGML can also be used in office document processing when the benefits of human readability and interchange with publishing systems are required.

0.1 Background

A document can be viewed in the abstract as a structure of various types of element. An author organizes a book into chapters that contain paragraphs, for example, and figures that contain figure captions. An editor organizes a magazine into articles that contain paragraphs that contain words, and so on.

Processors treat these elements in different ways. A formatting program might print headings in a prominent type face, leave space between paragraphs, and otherwise visually convey the structure and other attributes to the reader. An information retrieval system would perhaps assign extra significance to words in a heading when creating its dictionary.

Although this connection between a document's attributes and its processing now seems obvious, it tended to be obscured by early text processing methods. In the days before automated typesetting, an editor would “mark up” a manuscript with the specific processing instructions that would create the desired format when executed by a compositor. Any connection between the instructions and the document's structure was purely in the editor's head.

Early computerized systems continued this approach by adding the process-specific “markup” to the machine-readable document file. The markup still consisted of specific processing instructions, but now they were in the language of a formatting program, rather than a human compositor. The file could not easily be used for a different purpose, or on a different computer system, without changing all the markup.

As users became more sophisticated, and as text processors became more powerful, approaches were developed that alleviated this problem. “Macro calls” (or “format calls”) were used to identify points in the document where processing was to occur. The actual processing instructions were kept outside of the document, in “procedures” (or “macro definitions” or “stored formats”), where they could more easily be changed.

While the macro calls could be placed anywhere in a document, users began to recognize that most were placed at the start or end of document elements. It was natural, therefore, to choose names for such macros that were “generic identifiers” of the element types, rather than names that suggested particular processing (for example, “heading” rather than “format-17”), and so the practice of “generic coding” (or “generalized tagging”) began.

Generic coding was a major step towards marking automated text processing systems reflect the natural relationship between document attributes and processing. The advent of “generalized markup languages” in the early 1970's carried this trend further by providing a formal language base for generic coding. A generalized markup language observes two main principles: