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

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:

Reference to International Standard	*Appropriate Australian Standard*
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	*Appropriate Australian Standard*
ISO 8632/2, Information processing systems—Computer graphics—Metafile for the storage and transfer of picture description information—Part 2: Character encoding.	No Australian equivalent.
ISO 8632/4, Information processing systems—Computer graphics Metafile for the storage and transfer of picture description information—Part 4: Clear text encoding.	No Australian equivalent.

Contents

Annexes

Figures

(PAGE 12 IN THE HARD COPY IS BLANK)

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: