INTERNATIONAL STANDARD

ISO/IEC 40314

First edition
2016-03-01

Information technology — Mathematical Markup Language (MathML) Version 3.0 2nd Edition

Technologies de l'information — Langage de marquage mathématique (MathML) Version 3.0 2e édition

Reference number
ISO/IEC 40314:2016(E)

© ISO/IEC 2016



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2016, Published in Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

FOREWORD

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT), see the following URL: Foreword — Supplementary information.

ISO/IEC 40314 was prepared by W3C and was adopted, under the PAS procedure, by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, in parallel with its approval by national bodies of ISO and IEC.

© ISO/IEC 2016 – All rights reserved

Contents

1 Introduction **9**
1.1 Mathematics and its Notation . . . 9
1.2 Origins and Goals . . . 10
1.2.1 Design Goals of MathML . . . 10
1.3 Overview . . . 11
1.4 A First Example . . . 11

2 MathML Fundamentals **14**
2.1 MathML Syntax and Grammar . . . 14
2.1.1 General Considerations . . . 14
2.1.2 MathML and Namespaces . . . 14
2.1.3 Children versus Arguments . . . 15
2.1.4 MathML and Rendering . . . 15
2.1.5 MathML Attribute Values . . . 15
2.1.6 Attributes Shared by all MathML Elements . . . 20
2.1.7 Collapsing Whitespace in Input . . . 21
2.2 The Top-Level `<math>` Element . . . 22
2.2.1 Attributes . . . 22
2.2.2 Deprecated Attributes . . . 24
2.3 Conformance . . . 24
2.3.1 MathML Conformance . . . 24
2.3.2 Handling of Errors . . . 27
2.3.3 Attributes for unspecified data . . . 27

3 Presentation Markup **28**
3.1 Introduction . . . 28
3.1.1 What Presentation Elements Represent . . . 28
3.1.2 Terminology Used In This Chapter . . . 29
3.1.3 Required Arguments . . . 30
3.1.4 Elements with Special Behaviors . . . 31
3.1.5 Directionality . . . 32
3.1.6 Displaystyle and Scriptlevel . . . 33
3.1.7 Linebreaking of Expressions . . . 34
3.1.8 Warning about fine-tuning of presentation . . . 35
3.1.9 Summary of Presentation Elements . . . 37
3.1.10 Mathematics style attributes common to presentation elements . . . 38
3.2 Token Elements . . . 38
3.2.1 Token Element Content Characters, `<mglyph/>` . . . 39
3.2.2 Mathematics style attributes common to token elements . . . 41
3.2.3 Identifier `<mi>` . . . 45

© ISO/IEC 2016 – All rights reserved

3.2.4 Number <mn> . . . 46
3.2.5 Operator, Fence, Separator or Accent <mo> . . . 47
3.2.6 Text <mtext> . . . 60
3.2.7 Space <mspace/> . . . 62
3.2.8 String Literal <ms> . . . 64
3.3 General Layout Schemata . . . 64
3.3.1 Horizontally Group Sub-Expressions <mrow> . . . 64
3.3.2 Fractions <mfrac> . . . 67
3.3.3 Radicals <msqrt>, <mroot> . . . 69
3.3.4 Style Change <mstyle> . . . 69
3.3.5 Error Message <merror> . . . 72
3.3.6 Adjust Space Around Content <mpadded> . . . 73
3.3.7 Making Sub-Expressions Invisible <mphantom> . . . 78
3.3.8 Expression Inside Pair of Fences <mfenced> . . . 80
3.3.9 Enclose Expression Inside Notation <menclose> . . . 83
3.4 Script and Limit Schemata . . . 85
3.4.1 Subscript <msub> . . . 86
3.4.2 Superscript <msup> . . . 87
3.4.3 Subscript-superscript Pair <msubsup> . . . 87
3.4.4 Underscript <munder> . . . 88
3.4.5 Overscript <mover> . . . 89
3.4.6 Underscript-overscript Pair <munderover> . . . 91
3.4.7 Prescripts and Tensor Indices <mmultiscripts>, <mprescripts/>, <none/> 93
3.5 Tabular Math . . . 95
3.5.1 Table or Matrix <mtable> . . . 95
3.5.2 Row in Table or Matrix <mtr> . . . 99
3.5.3 Labeled Row in Table or Matrix <mlabeledtr> . . . 99
3.5.4 Entry in Table or Matrix <mtd> . . . 101
3.5.5 Alignment Markers <malignmark/>, <malignmark/> . . . 101
3.6 Elementary Math . . . 110
3.6.1 Stacks of Characters <mstack> . . . 111
3.6.2 Long Division <mlongdiv> . . . 113
3.6.3 Group Rows with Similiar Positions <msgroup> . . . 114
3.6.4 Rows in Elementary Math <msrow> . . . 115
3.6.5 Carries, Borrows, and Crossouts <mscarries> . . . 115
3.6.6 A Single Carry <mscarry> . . . 116
3.6.7 Horizontal Line <msline/> . . . 117
3.6.8 Elementary Math Examples . . . 118
3.7 Enlivening Expressions . . . 124
3.7.1 Bind Action to Sub-Expression <maction> . . . 124
3.8 Semantics and Presentation . . . 126

4 Content Markup 127
4.1 Introduction . . . 127
4.1.1 The Intent of Content Markup . . . 127
4.1.2 The Structure and Scope of Content MathML Expressions . . . 128
4.1.3 Strict Content MathML . . . 128
4.1.4 Content Dictionaries . . . 129
4.1.5 Content MathML Concepts . . . 130
4.2 Content MathML Elements Encoding Expression Structure . . . 131

© ISO/IEC 2016 – All rights reserved

4.2.1 Numbers <cn> . . . 132
4.2.2 Content Identifiers <ci> . . . 138
4.2.3 Content Symbols <csymbol> . . . 140
4.2.4 String Literals <cs> . . . 142
4.2.5 Function Application <apply> . . . 143
4.2.6 Bindings and Bound Variables <bind> and <bvar> . . . 146
4.2.7 Structure Sharing <share> . . . 148
4.2.8 Attribution via semantics . . . 150
4.2.9 Error Markup <cerror> . . . 151
4.2.10 Encoded Bytes <cbytes> . . . 152
4.3 Content MathML for Specific Structures . . . 152
4.3.1 Container Markup . . . 153
4.3.2 Bindings with <apply> . . . 154
4.3.3 Qualifiers . . . 156
4.3.4 Operator Classes . . . 162
4.3.5 Non-strict Attributes . . . 169
4.4 Content MathML for Specific Operators and Constants . . . 170
4.4.1 Functions and Inverses . . . 170
4.4.2 Arithmetic, Algebra and Logic . . . 180
4.4.3 Relations . . . 200
4.4.4 Calculus and Vector Calculus . . . 205
4.4.5 Theory of Sets . . . 224
4.4.6 Sequences and Series . . . 233
4.4.7 Elementary classical functions . . . 243
4.4.8 Statistics . . . 247
4.4.9 Linear Algebra . . . 253
4.4.10 Constant and Symbol Elements . . . 260
4.5 Deprecated Content Elements . . . 268
4.5.1 Declare <declare> . . . 268
4.5.2 Relation <reln> . . . 268
4.5.3 Relation <fn> . . . 268
4.6 The Strict Content MathML Transformation . . . 268

5 Mixing Markup Languages for Mathematical Expressions 272
5.1 Annotation Framework . . . 272
5.1.1 Annotation elements . . . 272
5.1.2 Annotation keys . . . 273
5.1.3 Alternate representations . . . 274
5.1.4 Content equivalents . . . 275
5.1.5 Annotation references . . . 276
5.2 Elements for Semantic Annotations . . . 276
5.2.1 The <semantics> element . . . 276
5.2.2 The <annotation> element . . . 277
5.2.3 The <annotation-xml> element . . . 278
5.3 Combining Presentation and Content Markup . . . 281
5.3.1 Presentation Markup in Content Markup . . . 281
5.3.2 Content Markup in Presentation Markup . . . 282
5.4 Parallel Markup . . . 282
5.4.1 Top-level Parallel Markup . . . 282
5.4.2 Parallel Markup via Cross-References . . . 283

© ISO/IEC 2016 – All rights reserved

6 Interactions with the Host Environment 286
6.1 Introduction . . . 286
6.2 Invoking MathML Processors . . . 286
6.2.1 Recognizing MathML in XML . . . 286
6.2.2 Recognizing MathML in HTML . . . 287
6.2.3 Resource Types for MathML Documents . . . 287
6.2.4 Names of MathML Encodings . . . 287
6.3 Transferring MathML . . . 288
6.3.1 Basic Transfer Flavor Names and Contents . . . 288
6.3.2 Recommended Behaviors when Transferring . . . 289
6.3.3 Discussion . . . 289
6.3.4 Examples . . . 290
6.4 Combining MathML and Other Formats . . . 292
6.4.1 Mixing MathML and XHTML . . . 294
6.4.2 Mixing MathML and non-XML contexts . . . 294
6.4.3 Mixing MathML and HTML . . . 294
6.4.4 Linking . . . 295
6.4.5 MathML and Graphical Markup . . . 296
6.5 Using CSS with MathML . . . 297
6.5.1 Order of processing attributes versus style sheets . . . 298

7 Characters, Entities and Fonts 299
7.1 Introduction . . . 299
7.2 Unicode Character Data . . . 299
7.3 Entity Declarations . . . 300
7.4 Special Characters Not in Unicode . . . 300
7.5 Mathematical Alphanumeric Symbols . . . 300
7.6 Non-Marking Characters . . . 303
7.7 Anomalous Mathematical Characters . . . 303
7.7.1 Keyboard Characters . . . 303
7.7.2 Pseudo-scripts . . . 304
7.7.3 Combining Characters . . . 306

A Parsing MathML 308
A.1 Use of MathML as Well-Formed XML . . . 308
A.2 Using the RelaxNG Schema for MathML3 . . . 308
A.2.1 Full MathML . . . 309
A.2.2 Elements Common to Presentation and Content MathML . . . 309
A.2.3 The Grammar for Presentation MathML . . . 311
A.2.4 The Grammar for Strict Content MathML3 . . . 323
A.2.5 The Grammar for Content MathML . . . 325
A.2.6 MathML as a module in a RelaxNG Schema . . . 332
A.3 Using the MathML DTD . . . 333
A.3.1 Document Validation Issues . . . 333
A.3.2 Attribute values in the MathML DTD . . . 333
A.3.3 DOCTYPE declaration for MathML . . . 334
A.4 Using the MathML XML Schema . . . 334
A.4.1 Associating the MathML schema with MathML fragments . . . 334
A.5 Parsing MathML in XHTML . . . 334
A.6 Parsing MathML in HTML . . . 334

© ISO/IEC 2016 – All rights reserved

B Media Types Registrations **335**
B.1 Selection of Media Types for MathML Instances . . . 335
B.2 Media type for Generic MathML . . . 336
B.3 Media type for Presentation MathML . . . 337
B.4 Media type for Content MathML . . . 338

C Operator Dictionary (Non-Normative) **340**
C.1 Indexing of the operator dictionary . . . 340
C.2 Format of operator dictionary entries . . . 340
C.3 Notes on `lspace` and `rspace` attributes . . . 341
C.4 Operator dictionary entries . . . 341

D Glossary (Non-Normative) **379**

E Working Group Membership and Acknowledgments (Non-Normative) **383**
E.1 The Math Working Group Membership . . . 383
E.2 Acknowledgments . . . 386

F Changes (Non-Normative) **387**
F.1 Changes between MathML 3.0 First Edition and Second Edition . . . 387
F.2 Changes between MathML 2.0 Second Edition and MathML 3.0 . . . 390

G Normative References **391**

H References (Non-Normative) **393**

I Index (Non-Normative) **395**
I.1 MathML Elements . . . 395
I.2 MathML Attributes . . . 400

© ISO/IEC 2016 – All rights reserved

Chapter 1

Introduction

1.1 Mathematics and its Notation

A distinguishing feature of mathematics is the use of a complex and highly evolved system of two-dimensional symbolic notation. As J. R. Pierce writes in his book on communication theory, mathematics and its notation should not be viewed as one and the same thing [Pierce1961]. Mathematical ideas can exist independently of the notation that represents them. However, the relation between meaning and notation is subtle, and part of the power of mathematics to describe and analyze derives from its ability to represent and manipulate ideas in symbolic form. The challenge before a Mathematical Markup Language (MathML) in enabling mathematics on the World Wide Web is to capture both notation and content (that is, its meaning) in such a way that documents can utilize the highly evolved notation of written and printed mathematics as well as the new potential for interconnectivity in electronic media.

Mathematical notation evolves constantly as people continue to innovate in ways of approaching and expressing ideas. Even the common notation of arithmetic has gone through an amazing variety of styles, including many defunct ones advocated by leading mathematical figures of their day [Cajori1928]. Modern mathematical notation is the product of centuries of refinement, and the notational conventions for high-quality typesetting are quite complicated and subtle. For example, variables and letters which stand for numbers are usually typeset today in a special mathematical italic font subtly distinct from the usual text italic; this seems to have been introduced in Europe in the late sixteenth century. Spacing around symbols for operations such as +, -, × and / is slightly different from that of text, to reflect conventions about operator precedence that have evolved over centuries. Entire books have been devoted to the conventions of mathematical typesetting, from the alignment of superscripts and subscripts, to rules for choosing parenthesis sizes, and on to specialized notational practices for subfields of mathematics. The manuals describing the nuances of present-day computer typesetting and composition systems can run to hundreds of pages.

Notational conventions in mathematics, and in printed text in general, guide the eye and make printed expressions much easier to read and understand. Though we usually take them for granted, we, as modern readers, rely on numerous conventions such as paragraphs, capital letters, font families and cases, and even the device of decimal-like numbering of sections such as is used in this document. Such notational conventions are perhaps even more important for electronic media, where one must contend with the difficulties of on-screen reading. Appropriate standards coupled with computers enable a broadening of access to mathematics beyond the world of print. The markup methods for mathematics in use just before the Web rose to prominence importantly included TEX (also written `TeX`) [Knuth1986] and approaches based on SGML ([AAP-math], [Poppelier1992] and [ISO-12083]).

It is remarkable how widespread the current conventions of mathematical notation have become. The general two-dimensional layout, and most of the same symbols, are used in all modern mathematical communications, whether the participants are, say, European, writing left-to-right, or Middle-Eastern,

© ISO/IEC 2016 – All rights reserved

writing right-to-left. Of course, conventions for the symbols used, particularly those naming functions and variables, may tend to favor a local language and script. The largest variation from the most common is a form used in some Arabic-speaking communities which lays out the entire mathematical notation from right-to-left, roughly in mirror image of the European tradition.

However, there is more to putting mathematics on the Web than merely finding ways of displaying traditional mathematical notation in a Web browser. The Web represents a fundamental change in the underlying metaphor for knowledge storage, a change in which *interconnection* plays a central role. It has become important to find ways of communicating mathematics which facilitate automatic processing, searching and indexing, and reuse in other mathematical applications and contexts. With this advance in communication technology, there is an opportunity to expand our ability to represent, encode, and ultimately to communicate our mathematical insights and understanding with each other. We believe that MathML as specified below is an important step in developing mathematics on the Web.

1.2 Origins and Goals

1.2.1 Design Goals of MathML

MathML has been designed from the beginning with the following ultimate goals in mind.

MathML should ideally:

- Encode mathematical material suitable for all educational and scientific communication.
- Encode both mathematical notation and mathematical meaning.
- Facilitate conversion to and from other mathematical formats, both presentational and semantic. Output formats should include:
  - graphical displays
  - speech synthesizers
  - input for computer algebra systems
  - other mathematics typesetting languages, such as TEX
  - plain text displays, e.g. VT100 emulators
  - international print media, including braille

  It is recognized that conversion to and from other notational systems or media may entail loss of information in the process.
- Allow the passing of information intended for specific renderers and applications.
- Support efficient browsing of lengthy expressions.
- Provide for extensibility.
- Be well suited to templates and other common techniques for editing formulas.
- Be legible to humans, and simple for software to generate and process.

No matter how successfully MathML achieves its goals as a markup language, it is clear that MathML is useful only if it is implemented well. The W3C Math Working Group has identified a short list of additional implementation goals. These goals attempt to describe concisely the minimal functionality MathML rendering and processing software should try to provide.

- MathML expressions in HTML (and XHTML) pages should render properly in popular Web browsers, in accordance with reader and author viewing preferences, and at the highest quality possible given the capabilities of the platform.
- HTML (and XHTML) documents containing MathML expressions should print properly and at high-quality printer resolutions.
- MathML expressions in Web pages should be able to react to user gestures, such those as with a mouse, and to coordinate communication with other applications through the browser.

© ISO/IEC 2016 – All rights reserved

- Mathematical expression editors and converters should be developed to facilitate the creation of Web pages containing MathML expressions.

The extent to which these goals are ultimately met depends on the cooperation and support of browser vendors and other developers. The W3C Math Working Group has continued to work with other working groups of the W3C, and outside the W3C, to ensure that the needs of the scientific community will be met. MathML 2 and its implementations showed considerable progress in this area over the situation that obtained at the time of the MathML 1.0 Recommendation (April 1998) [MathML1]. MathML3 and the developing Web are expected to allow much more.

1.3 Overview

MathML is a markup language for describing mathematics. It is usually expressed in XML syntax, although HTML and other syntaxes are possible. A special aspect of MathML is that there are two main strains of markup: Presentation markup, discussed in Chapter 3, is used to display mathematical expressions; and Content markup, discussed in Chapter 4, is used to convey mathematical meaning. Content markup is specified in particular detail. This specification makes use of an XML format called Content Dictionaries This format has been developed by the OpenMath Society, [OpenMath2004] with the dictionaries being used by this specification involving joint development by the OpenMath Society and the W3C Math Working Group.

Fundamentals common to both strains of markup are covered in Chapter 2, while the means for combining these strains, as well as external markup, into single MathML objects are discussed in Chapter 5. How MathML interacts with applications is covered in Chapter 6. Finally, a discussion of special symbols, and issues regarding characters, entities and fonts, is given in Chapter 7.

1.4 A First Example

The quadratic formula provides a simple but instructive illustration of MathML markup.

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

MathML offers two flavors of markup of this formula. The first is the style which emphasizes the actual presentation of a formula, the two-dimensional layout in which the symbols are arranged. An example of this type is given just below. The second flavor emphasizes the mathematical content and an example of it follows the first one.

```
<mrow>
  <mi>x</mi>
  <mo>=</mo>
  <mfrac>
    <mrow>
      <mrow>
        <mo>-</mo>
        <mi>b</mi>
      </mrow>
      <mo>&PlusMinus;</mo>
      <msqrt>
```

© ISO/IEC 2016 – All rights reserved