

Best Practice Guide: Math in PDF

Version 1.0



L^AT_EX LWG

Copyright © 2026 PDF Association or its licensors.

This work is licensed under the Creative Commons Attribution 4.0 International License.

To view a copy of this license, visit
<http://creativecommons.org/licenses/by/4.0/>

or send a letter to
Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

PDF Association
Friedenstr. 2A · 16321 Bernau bei Berlin · Germany

E-mail: copyright@pdfa.org
Web: www.pdfa.org

Published in Germany and the United States of America

Vendors own their respective copyrights wherever they are mentioned. The information contained in this document is provided only as general information, which may be incomplete or outdated. Users of this document are responsible for independently verifying any and all information. Any mention of companies or products does not imply endorsement or support of any of the mentioned information, services, products, or providers.

Overview

The Best Practice Guide provides recommendations for creating STEM documents and other documents with complex technical and mathematical content using PDF files conforming to PDF 1.7 and PDF 2.0. This guidance includes the proper ways to embed mathematical formulas and discusses potential limitations when using PDF 1.7.

This document is non-normative and does not modify any requirements in existing ISO standards for PDF.

PDF 2.0 is strongly recommended because:

- PDF 2.0 includes MathML as one of three standardized namespaces, while PDF 1.7 does not specify the use of MathML.
- PDF 2.0 incorporates the use of Associated Files (AF), which provide a standardized mechanism for embedding MathML and/or \LaTeX representations of formulas as a supplement to their representation in PDF. PDF's AF feature was introduced for PDF 1.7 by PDF/A-3, but is not used by accessibility standards prior to PDF/UA-2, which is based on PDF 2.0.

In order to fully benefit from Tagged PDF and the accessibility features of PDF, documents with mathematical content are strongly recommended to comply with one or both of the following standards:

- PDF/UA-2
- WTPDF

PDF/UA-1 is not recommended, as it does not include provisions for tagging mathematical content and only supports an inferior method; describing the math using the **Alt** property.

In this document the term "mathematical formulas or expressions" refers to any structure that can be represented in MathML syntax. PDF 2.0 normatively references (see [errata to ISO 32000-2, 14.8.6.3](#) the MathML Core specification to define the MathML namespace.

In most contexts it is recommended to restrict the use of MathML to the subset of MathML defined by MathML Core, as this is the subset of MathML that is implemented in web browsers. The use of additional attributes which can be mapped to MathML4 attributes is encouraged, as explained below.

Non-mathematical formulas not representable with MathML are out of scope of this document.

This document focuses primarily on PDF authoring applications, such as \LaTeX . It also addresses PDF processors that do not use the above standards and therefore may have different behavior, or when processing PDF data that does not conform to these standards, but are nevertheless recommended by this document.

This Best Practice Guide is published as PDF 2.0 document and complies with PDF/UA-2 and Well-Tagged PDF for Accessibility and Reuse, as well as PDF/A-4f long term archival standard. It embeds CSS as an associated file for better MathML rendering in the Derived HTML.

References

- [1] L^AT_EX Tagging Project, The L^AT_EX Tagged PDF repository, <https://latex3.github.io/tagging-project>
- [2] ISO 14289-2, Document management applications – Electronic document file format enhancement for accessibility – Part 2: Use of ISO 32000-2 (PDF/UA-2), <https://pdfa.org/iso-14289-2-pdfua-2>
- [3] ISO 32000-2, Document management – Portable document format – Part 2: PDF 2.0, <https://pdfa.org/resource/iso-32000-2>
- [4] Well-Tagged PDF (WTPDF), Using Tagged PDF for Accessibility and Reuse in PDF 2.0, <https://pdfa.org/wtpdf>
- [5] ISO 14289-1, Document management applications – Electronic document file format enhancement for accessibility – Part 1: Use of ISO 32000-1 (PDF/UA-1), <https://pdfa.org/resource/iso-14289-pdfua>
- [6] ISO/TS 32005, Document management – Portable Document Format – PDF 1.7 and 2.0 structure namespace inclusion in ISO 32000-2, <https://pdfa.org/resource/iso-32005>
- [7] MathML Core, W3C Candidate Recommendation, World Wide Web Consortium (W3C), <https://www.w3.org/TR/mathml-core/>
- [8] Deriving HTML from PDF, <https://pdfa.org/resource/deriving-html-from-pdf>
- [9] Errata corrections for ISO 32000-2:2020 (PDF 2.0), <https://pdf-issues.pdfa.org/32000-2-2020>

Notation

The document uses the same notation as ISO 32000-2, 4.1 when referencing PDF syntax. Specifically, the names of keys in PDF dictionaries and other predefined PDF names are written in bold font; values of dictionary keys are written in italic font. All XML tags, attributes and L^AT_EX commands are styled as code. Examples of assistive technology (AT) reading are styled as code with an italic font.

How Math is included into PDF

As per ISO 14289-2, 8.2.5.29.1, Mathematical formulas are required to be included in accessible PDF files using MathML syntax tagged with a **Formula** structure element. The MathML is represented either as **math** structure elements in the MathML namespace or as a MathML file in XML syntax associated with a **Formula** structure element.

Math as a child of a Formula tag

As required by ISO 32000-2, 14.8.6.3 and ISO 14289-2, 8.2.5.29.1, MathML **math** structure elements are either a direct child of a **Formula**, or (where MathML permits this) a child (direct or indirect) of another structure element defined in MathML namespace. In the latter case, it is not required to enclose the inner **math** structure element with a parent **Formula** tag.

When a **Formula** structure element contains **math** as a direct child, it is strongly recommended that the **Placement** attribute of **Formula** tag matches the **display** attribute of the **math** tag in XML syntax.

The following structure elements may be contained by structure elements defined in the MathML namespace:

- Structure elements from the MathML namespace. The containment rules are defined in this case by the MathML specification
- A limited number of PDF tags as defined in [the section *Inclusion of PDF structure elements into MathML* below](#).

Use of Associated files

A **Formula** structure element may have associated MathML as an embedded file. The embedded file is required to include an **AFRelationship** entry with value *Supplement* and a MIME type of *application/mathml+xml* (`application#2Fmathml+xml` when encoded as a PDF name) as the value of the **Subtype** entry (see ISO 14289-2, 8.2.5.29.1). It is strongly recommended that the associated file contains an equivalent representation to that which would have been used had a **math** structure element been used inside the **Formula**.

Use of the Alt property

A **Formula** structure element containing a mathematical expression (either as a child of a tag or as an associated file) may additionally contain an **Alt** entry that provides an alternative text description of the formula. Such an alternative description can be used by PDF processors that do not support MathML syntax to provide a human readable description of the formula, similar to alternative text for images.

Precedence

If a **Formula** structure element contains both the representation of a MathML expression as an associated file and a **math** structure element as a direct child of **Formula**, it is strongly recommended that the **math** structure element takes precedence.

If both MathML and an **Alt** entry are provided for the **Formula** structure element, it is strongly recommended that by default PDF processors provide the user with the MathML representation of the formula rather than the value of the **Alt** entry.

Example

The example of representing Euler's formula in PDF

$$e^{i\phi} = \cos(\phi) + i \sin(\phi)$$

uses both the **math** structure element as a direct child of a **Formula** tag and the file `mathml-1.xml` in MathML XML syntax associated with the same **Formula** tag. It also contains an **Alt** entry. The content of the **math** structure element takes precedence.

```
Formula ID="ID.001" Alt="Euler's formula" AF="mathml-1.xml"
  math display="block"
    msup
      mi e
      mrow
        mi i
        mi φ
    mo =
    ... (12 MathML structure elements omitted)
```

Representation of PDF Structure Elements as XML

ISO 32000-2 does not provide any guidance on the mapping between XML elements and corresponding PDF structure elements. The suggested correspondence is shown in [Table 1](#).

Table 1: Correspondence between PDF and XML syntax for MathML

| PDF Structure | XML |
|--|---|
| Structure element in a Namespace | XML element in the same namespace as specified by the NS entry |
| Lang Property | <code>lang</code> attribute (no namespace) |
| Alt Property | <code>alt</code> attribute (no namespace) |
| T Property | <code>title</code> attribute (no namespace) |
| ID Property | <code>id</code> attribute (no namespace) |
| Attribute with Owner <i>NSO</i> with same NS object as the NS object specified for the parent element | attribute (no namespace) eg <code>lspace=2pt</code> |
| Attribute with Owner <i>NSO</i> with NS object specifying the Namespace URI (so to represent an attribute in the element's namespace a second NS object would be needed) | attribute (in Namespace) eg <code>xlink:href="..."</code> |
| Text extracted from a marked content region after applying toUnicode translation | Element text content |

Inclusion of PDF structure elements into MathML

MathML (both [MathML3](#) and [MathML4](#)) permits the use of [Phrasing HTML5 tags](#) in any leaf nodes of the MathML structure tree. It is permitted to use a limited subset of PDF tags as children of structure elements defined in the MathML namespace.

[Table 2](#) specifies a set of PDF structure elements permitted as children of the MathML namespace's **mtext** structure element in PDF, together with their HTML5 equivalents. The latter are strongly recommended to be used if the same MathML syntax is expressed via MathML contained in an Associated File (AF).

Table 2: Permitted standard PDF structure elements inside MathML

| PDF structure element | HTML5 |
|------------------------|-----------------------------------|
| Reference, Link | a tag |
| Strong | strong tag |
| Code | code tag |
| Em | em tag |
| Span | span tag |
| Lbl | property intent=":equation-label" |

As some processors cannot handle PDF structure elements (such as the **Lbl** structure element) as a child of a MathML structure element, it is recommended to structure the equation with a label as a MathML **mtable** structure element and represent the label as a MathML structure element **mtd** with Attribute **intent** equal to *:equation-label*.

Example equation labels and references

The following \LaTeX document

```
\DocumentMetadata{
  lang      = en,
  pdfversion = 2.0, pdfstandard = {ua-2, a-4},
  tagging-setup={math/setup=mathml-SE}
}
\documentclass{article}
\usepackage{unicode-math}
\usepackage{hyperref}

\begin{document}
Euler formula:
\begin{equation}\label{eqn1}
  e^{i\phi} = \cos(\phi) + i\sin(\phi)
\end{equation}

Hence
\begin{equation}
  e^{i\pi} = \text{(use \eqref{eqn1})} = -1
\end{equation}
\end{document}
```

compiles into the following visual representation:

Euler formula:

$$e^{i\phi} = \cos(\phi) + i \sin(\phi) \quad (1)$$

Hence

$$e^{i\pi} = (\text{use (1)}) = -1 \quad (2)$$

There are two equivalent ways to represent this structure in PDF:

1. Using MathML structure elements with embedded structure elements from the standard PDF 2.0 namespace directly in the structure tree. The pseudo-code below represents the corresponding PDF structure tree:

```
latex:text-unit % role mapped to Part
  latex:text % role mapped to P
  Formula ID="ID.007"
  math
    mtable intent=":system-of-equations"
      mtr
        mtd intent=":equation-label"
          mtext
            Lbl (1)
        mtd
          ... (MathML structure elements omitted)

latex:text-unit % role mapped to Part
  latex:text % role mapped to P
  Formula ID="ID.008"
  math
    mtable intent=":system-of-equations"
      mtr
        mtd intent=":equation-label"
          mtext
            Lbl (2)
        mtd
          ... (MathML structure elements omitted)
      mtext
        (see
          Reference /Ref="ID.007"
            (1)
          OBJR <Link annotation>
        )
      ... (MathML structure elements omitted)
```

2. Embedding complete MathML representations of the formulas as Associated Files of type *application/mathml+xml* associated with the respective **Formula** structure element. In this case PDF 2.0 structure elements contained within the **mtext** structure element are mapped to their equivalent HTML tags.

The following pseudo code demonstrates the PDF structure tree in this case

```

latex:text-unit % role mapped to Part
  latex:text % role mapped to P
    Formula ID="ID.007" AF="mathml-1.xml"

latex:text-unit % role mapped to Part
  latex:text % role mapped to P
    Formula ID="ID.008" AF="mathml-2.xml"

```

where the code of the two associated MathML files looks like:

mathml-1.xml

```

<math display="block" xmlns="http://www.w3.org/1998/Math/MathML">
<table displaystyle="true" intent=":system-of-equations">
  <mtr>
    <mtd intent=":equation-label">
      <mtext>(1)</mtext>
    </mtd>
    <mtd>
      <msup>
        <mi>e</mi>
        <mrow>
          <mi>i</mi>
          <mi>φ</mi>
        </mrow>
      </msup>
      <mo>=</mo>
      <mi mathvariant="normal">cos</mi>
      <mo>( </mo>
      <mi>φ</mi>
      <mo>)</mo>
      <mo>+</mo>
      <mi>i</mi>
      <mi mathvariant="normal">sin</mi>
      <mo>( </mo>
      <mi>φ</mi>
      <mo>)</mo>
      <mo>.</mo>
    </mtd>
  </mtr>
</table>
</math>

```

mathml-2.xml

```
<math display="block" xmlns="http://www.w3.org/1998/Math/MathML">
  <mtable displaystyle="true" intent=":system-of-equations">
    <mtr>
      <mtd intent=":equation-label">
        <mtext>(2)</mtext>
      </mtd>
      <mtd>
        <msup>
          <mi>e</mi>
          <mrow>
            <mi>i</mi>
            <mi>π</mi>
          </mrow>
        </msup>
        <mo>=</mo>
        <mtext>(use <a href="#ID.007">(1)</a>)</mtext>
        <mo>=</mo>
        <mo>-</mo>
        <mn>1</mn>
        <mo>.</mo>
      </mtd>
    </mtr>
  </mtable>
</math>
```

Note that in this case the a tag from mathml-2.xml references the **Formula** structure element with ID value *ID.007* from the PDF structure tree.

Example of multiline display formula

The L^AT_EX data below contains a two line formula aligned by the = sign.

```
\DocumentMetadata{
  lang          = en,
  pdfversion    = 2.0, pdfstandard = {ua-2, a-4},
  tagging-setup={math/setup=mathml-SE}
}
\documentclass{article}
\usepackage{unicode-math}

\begin{document}
\setcounter{equation}{4}

\begin{align}
  f(x)&=x^2\\
  f'(x)&=2x
\end{align}
\end{document}
```

The above L^AT_EX source has the following visual representation

$$f(x) = x^2 \tag{5}$$

$$f'(x) = 2x \tag{6}$$

The corresponding structure tree in PDF is represented using a two row table with three columns; one column for the equation label and two columns for two aligned formula parts.

```
latex:text-unit % role mapped to Part
  Formula ID="ID.001"
  math
    mtable intent=":system-of-equations"
      mtr
        mtd intent=":equation-label"
          mtext
            Lbl (5)
        mtd
          ... (MathML structure elements representing f(x))
        mtd
          ... (MathML structure elements representing =x^2)
      mtr
        mtd intent=":equation-label"
          mtext
            Lbl (6)
        mtd
          ... (MathML structure elements representing f'(x))
        mtd
          ... (MathML structure elements representing 2x)
```

Example of nested math

The following L^AT_EX file demonstrates the use of math inside math.

```
\DocumentMetadata{
  lang      = en,
  pdfversion = 2.0, pdfstandard = {ua-2, a-4},
  tagging-setup={math/setup=mathml-SE}
}
\documentclass{article}
\usepackage{unicode-math}

\begin{document}
\[ P_{r - j} =
\begin{cases}
0 & \& \text{if } r - j \text{ is odd,} \\
r! \, \backslash, (-1)^{(r - j)/2} & \& \text{if } r - j \text{ is even.}
\end{cases}
\end{cases}
\]
\end{document}
```

It is compiled into the visual representation

$$P_{r-j} = \begin{cases} 0 & \text{if } r - j \text{ is odd,} \\ r! (-1)^{(r-j)/2} & \text{if } r - j \text{ is even.} \end{cases}$$

The corresponding structure tree includes **math** structure elements contained within an **mtext** structure element.

```
latex:text-unit % role mapped to Part
  Formula ID="ID.001"
  math
  msub
  ... (MathML structure elements representing P_{r-j})
  mo
  =
  mrow
  mo
  {
  mrow
  mspace
  mtable
  mtr
  mtd
  mn
  0
  mtd
  mspace
  mtd
  mtext
  if
  math
  mi
  i
  mo
  -
  mi
  j
  is odd
  mtr
  ... (MathML structure elements representing
  the second case)
  mspace
```

Use of Artifacts and ActualText

Some mathematical operators use special graphics in their visual representation. These are expressed semantically in the MathML namespace via the corresponding tags. Typical examples are

msqrt and **mfrac** for (square) roots and mathematical fractions, respectively. It is strongly recommended that these special graphics are marked as artifacts in PDF syntax by means of marked content sequences in the page stream.

For example, the following formula

$$\sqrt{a + b}$$

is tagged in PDF as

```
math
  msqrt
    mi a
    mo +
    mi b
```

and is drawn in page content as

```
/Artifact
BMC
  ... graphics representing square root symbol
EMC
/mi <</MCID 1 >>
BDC
  ... positioning operators omitted
BT (a) Tj ET
EMC
/mo <</MCID 2 >>
BDC
  ... positioning operators omitted
BT (+) Tj ET
EMC
/mo <</MCID 3 >>
BDC
  ... positioning operators omitted
BT (b) Tj ET
EMC
```

In other cases the visual representation of operators is drawn in PDF page content as a combination of line art and/or several glyphs, but is represented in MathML as one or several Unicode characters. It is strongly recommended that such operators are represented in page content as **Span** marked content sequences with the **ActualText** entry representing the Unicode value of the operator.

For example, the matrix

$$\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

is tagged in PDF as

```
math
  mo (
  mtable
    mtr
      mtd
        mn 1
    mtr
      mtd
        mn 2
    mtr
      mtd
        mn 3
  mo )
```

and is drawn in page content as

```
/mo <</MCID 1 /ActualText (\() >>
BDC
  ... graphics representing the left bracket
EMC
/mn <</MCID 2 >>
BDC
  ... positioning operators omitted
BT (1) Tj ET
EMC
/mn <</MCID 3 >>
BDC
  ... positioning operators omitted
BT (2) Tj ET
EMC
/mn <</MCID 4 >>
BDC
  ... positioning operators omitted
BT (3) Tj ET
EMC
/mo <</MCID 5 /ActualText (\)) >>
BDC
  ... graphics representing the right bracket
EMC
```

Use of intent attributes

To improve the generation of accessible readings, the MathML used to tag mathematics may be enhanced by the use of the [intent attribute](#) added in MathML4.

Previous examples in this document included this attribute to improve the readings of aligned equations, distinguishing them from other alignments such as matrices.

Another possible use is to improve the reading of specific notations. Consider \bar{x}

```
<mover><mi>x</mi><mo>_</mo></mover>
```

This may be read as x bar when AT has no information about the intended meaning and just “reads the notation”.

The reading may be improved by changing the markup to

```
<mover intent="mean($x)">
  <mi arg="x">x</mi>
  <mo>_</mo>
</mover>
```

or

```
<mover intent="complex-conjugate($x)">
  <mi arg="x">x</mi>
  <mo>_</mo>
</mover>
```

which would produce (English) readings of *mean of x* or *complex conjugate of x*, respectively.

Both of these intent concepts are [core concepts](#) which means that they may be read in any appropriate language used by the AT. Other concept names that do not appear in the core list may be used, but typically they will be read as-is without specific translation.

Sometimes it is inappropriate for AT to make any guess regarding “common” mathematical notation. For example, x^2 by default would be read as *x squared*. Changing this to x^2 with `<msup intent=":literal"><mi>x</mi><mn>2</mn></msup>` AT produces a reading such as *x super 2 end super* which does not imply a mathematical meaning to the superscript.

It may be expected that authoring systems defining commands for mathematical operators will supply suitable intents automatically. In \LaTeX , packages defining new commands may automatically add intent, or authors can add specific intents using the standard `\MathMLintent` and `\MathMLarg` commands.