

Review I: Equation, MathML, and SVG domains

Table of contents

1 Equation domain.....	3
1.1 <equation-block>.....	3
1.2 <equation-figure>.....	4
1.3 <equation-inline>.....	5
1.4 <equation-number>.....	6
2 MathML domain.....	8
2.1 <mathml>.....	8
2.2 <mathmlref>.....	9
3 SVG domain.....	12
3.1 <svg-container>.....	12
3.2 <svgref>.....	13
A Aggregated RFC-2119 statements.....	16
Index.....	17

1 Equation domain

The **equation domain** includes elements that authors can use to identify, number, and format equations within a document. This domain can be used independently of the MathML domain.

1.1 <equation-block>

The <equation-block> element represents an equation that is presented as a separate block within a text flow or an <equation-figure>.

Usage information

When an <equation-block> element has multiple direct child elements, each child represents an alternative form of the equation.

Rendering expectations

Block equations can be numbered.

Processing expectations

When there are multiple forms of an equation, processors can choose the form or forms that they render. For example, if there is both an image and MathML markup, an HTML-generating processor could generate both the image reference and the MathML with appropriate HTML @class or @id values to enable dynamic rendering **that is** based on browser capability.

Specialization hierarchy

The <equation-block> element is specialized from <div>. It is defined in the equation domain module.

Attributes

The following attributes are available on this element: [universal attributes](#).

Example

The following code sample shows how an <equation-block> element can include two alternative forms of the same equation:

```
<equation-block>
  <!-- Imaged-based equation -->
  <image keyref="equation-image-01">
    <alt>a squared plus b squared.</alt>
  </image>
  <!-- MathML-based equation -->
  <mathml>
    <math>
      <m:semantics>
        <m:mrow>
          <m:msqrt>
            <m:mrow>
              <m:msup><m:mi>a</m:mi><m:mn>2</m:mn></m:msup>
              <m:mo>+</m:mo>
              <m:msup><m:mi>b</m:mi><m:mn>2</m:mn></m:msup>
            </m:mrow>
          </m:msqrt>
        </m:mrow>
      </m:semantics>
    </math>
  </mathml>
```

```
        </m:mrow>
      </m:msqrt>
    </m:mrow>
  </m:semantics>
</m:math>
</mathml>
</equation-block>
```

1.2 <equation-figure>

The <equation-figure> element is a container for equations and their supporting information.

Usage information

Equation figures can have titles, descriptions, figure groups, and all other figure components. The direct children of <equation-figure> can be the equation content itself (for example, <mathml> or an image reference), or it can be one or more <equation-block> elements, along with other elements allowed within figures.

When an <equation-figure> element has multiple direct child <mathml>, <equation-block>, , or <pre> elements, each child represents an alternative form of the equation.

When the intent is to combine equations and commentary within an <equation-figure>, use child <equation-block> elements to contain the equations and so clearly distinguish them from the commentary.

Rendering expectations

Equation figures can be numbered. Either standard figure numbering can be used, or <equation-number> elements can be placed within <equation-block> elements.

Processing expectations

When there are multiple forms of an equation, processors can choose the form or forms that they render. For example, if there is both an image and MathML markup, an HTML-generating processor could generate both the image reference and the MathML with appropriate HTML @class or @id values to enable dynamic rendering that is based on browser capability.

Specialization hierarchy

The <equation-figure> element is specialized from <fig>. It is defined in the equation domain module.

Attributes

The following attributes are available on this element: [display attributes](#) and [universal attributes](#).

Example

The following code sample shows how an <equation-figure> element can contain both MathML content and commentary. The MathML content is contained with a nested <equation-block> element, and it is followed by commentary that is contained in a nested <p> element.

```
<equation-figure>
  <title>An equation with commentary</title>
  <equation-block>
    <mathml>
```

```
<m:math display='block'>
  <m:semantics>
    <m:mrow>
      <m:mfrac>
        <m:mrow><m:mi>n</m:mi><m:mo>!</m:mo></m:mrow>
        <m:mrow><m:mi>r</m:mi><m:mo>!</m:mo>
          <m:mrow>
            <m:mo>(</m:mo>
            <m:mrow><m:mi>n</m:mi><m:mo>#x2212;</m:mo><m:mi>r</m:mi></m:mrow>
            <m:mo>)</m:mo>
          </m:mrow>
          <m:mo>!</m:mo>
        </m:mrow>
      </m:mfrac>
    </m:mrow>
  </m:semantics>
</m:math>
</mathml>
</equation-block>
<p>Where
  <equation-inline><mathml><m:math><m:mi>r</m:mi></m:math></mathml></equation-inline>
  is greater than 1.</p>
</equation-figure>
```

1.3 <equation-inline>

The <equation-inline> element represents an equation that is presented inline within a paragraph or similar context.

Usage information

Inline equations are not intended to be numbered.

When an <equation-inline> element has multiple direct child elements, each child represents an alternative form of the equation.

Processing expectations

When there are multiple forms of an equation, processors can choose the form or forms that they render. For example, if there is both an image and MathML markup, an HTML-generating processor could generate both the image reference and the MathML with appropriate HTML @class or @id values to enable dynamic rendering **that is** based on browser capability.

Specialization hierarchy

The <equation-inline> element is specialized from <ph>. It is defined in the equation domain module.

Attributes

The following attributes are available on this element: [universal attributes](#) and [@keyref](#).

Examples

This section contains examples of how the `<equation-inline>` element can be used.

Figure 1: An inline equation

The following code sample shows how a paragraph can contain an `<equation-inline>` element that holds MathML markup:

```
<p>Consider the following equation:  
<equation-inline>  
  <mathml>  
    <math display='inline'>  
      <m:semantics>  
        <m:mrow>  
          <m:msqrt>  
            <m:mrow>  
              <m:msup><m:mi>a</m:mi><m:mn>2</m:mn></m:msup>  
              <m:mo>+</m:mo>  
              <m:msup><m:mi>b</m:mi><m:mn>2</m:mn></m:msup>  
            </m:mrow>  
          </m:msqrt>  
        </m:mrow>  
      </m:semantics>  
    </m:math>  
  </mathml>  
</equation-inline>  
It is simple arithmetic that school children understand.</p>
```

Figure 2: An inline equation that is image-based

The following code sample shows how the `<equation-inline>` element can contain an image:

```
<p>The Pythagorean Theorem describes the relationship among the three sides of a  
right triangle. In any right triangle, the sum of the areas of the squares formed on the  
legs of the triangle equals the area of the square formed on the hypotenuse:  
<equation-inline>  
  <image keyref="equation-image-01">  
    <alt>a squared plus b squared.</alt>  
  </image>  
</equation-inline>  
</p>
```

1.4 `<equation-number>`

The `<equation-number>` element indicates that a block equation should be numbered. It optionally specifies the number to use for the block equation.

Usage information

In normal usage, a block equation has a single number. However, the `<equation-number>` element can occur multiple times within the `<equation-block>` element. This enables the use of numbers with different (and exclusive) conditional properties.

When the `<equation-figure>` element contains content, the content of the element should be the number value without any **rendering-specific** punctuation, for example, "3.2a" rather than "(3.2a)".

Rendering expectations

001 (16)

In this context, white-space content is considered equivalent to empty content. When the `<equation-number>` element has empty content, the equation number **SHOULD** be generated. When the `<equation-number>` element is not empty,

the content **SHOULD** be used as the equation number. Processors **MAY** add punctuation or decoration to the number.

The details of equation numbering and number presentation are processor-specific. A common practice is to present the equation number to the right of the equation, centered vertically within the vertical extent of the block equation.

Specialization hierarchy

The `<equation-number>` element is specialized from `<ph>`. It is defined in the equation domain module.

Attributes

The following attributes are available on this element: [universal attributes](#).

Examples

This section contains examples of how the `<equation-number>` element can be used:

Figure 3: An equation where the number will be generated

The following code sample shows how an `<equation-number>` element can be used to indicate to a processor that an equation number should be auto-generated:

```
<equation-block id="eq-001">
  <equation-number/>
  <image keyref="equation-image-01">
    <alt>a squared plus b squared.</alt>
  </image>
</equation-block>
```

Figure 4: An equation where the equation number is explicitly specified

The following code sample shows how an `<equation-number>` element can specify the value for an equation number:

```
<equation-block id="eq-3.2a">
  <equation-number>3.2a</equation-number>
  <image keyref="equation-image-01">
    <alt>a squared plus b squared.</alt>
  </image>
</equation-block>
```

2 MathML domain

The MathML domain elements enable the use of embedded or referenced MathML markup. Referenced MathML markup must be stored in separate, non-DITA XML documents. MathML is a W3C standard.

When MathML elements are embedded in DITA documents that are validated using DTDs, the MathML elements must use a namespace prefix in order to avoid conflict with the DITA-defined elements of the same name. Documents validated using RELAX NG can default the MathML namespace on the MathML `<math>` element. MathML elements that are referenced using the `<mathmlref>` element do not need to have a namespace prefix, because they are parsed separately from the DITA documents that refer to them.

By default, the MathML domain is configured to use the namespace prefix "m" for the MathML elements.

Related information

[Mathematical Markup Language \(MathML\), Version 3.0](#)

2.1 <mathml>

The `<mathml>` element contains MathML markup or other content that contributes to a semantic equation.

Usage information

The `<mathml>` element can contain MathML elements, references to MathML elements stored in separate, non-DITA documents, or `<data>` elements.

The `<mathml>` element is not intended to represent a semantic equation, only content that contributes to a semantic equation. Use the equation domain elements or their equivalent to represent equations semantically, for example, to enable numbering of equations.

The MathML markup must have a root element of `<math>` within the MathML namespace: <http://www.w3.org/1998/Math/MathML>.

Specialization hierarchy

The `<mathml>` element is specialized from `<foreign>`. It is defined in the MathML domain module.

Attributes

The following attributes are available on this element: [universal attributes](#).

Example

The following code sample shows how to use a `<mathml>` element to include MathML content:

```
<equation-block>
  <mathml>
    <m:math>
      <m:semantics>
        <m:mrow>
          <m:msqrt>
            <m:mrow>
              <m:msup>
```



```

    <m:mi>a</m:mi>
    <m:mn>2</m:mn></m:msup>
    <m:mo>+</m:mo>
    <m:msup>
      <m:mi>b</m:mi>
      <m:mn>2</m:mn>
    </m:msup>
  </m:mrow>
</m:msqrt>
</m:mrow>
</m:semantics>
</m:math>
</mathml>
</equation-block>

```

2.2 <mathmlref>

The <mathmlref> element references a non-DITA XML document that contains MathML markup.

Usage information

The <mathmlref> element enables the use MathML markup by reference. The reference must be to a MathML <math> element. The reference can be one of the following:

- A URI that addresses an XML document. The XML document has a MathML <math> element as the root element.
- A URI that addresses an XML document and contains a fragment identifier that is the XML ID of a <math> element within the document.

The reference can be direct, using the @href attribute, or indirect, using the @keyref attribute. For indirect referencing, only the key name should be specified. The ID of the <math> element must be specified as part of the value for the @href attribute on the key definition.

For example, to refer to the <math> element with the @id of "math-fragment-02" within a larger document using a key reference, you would define the key in the following way:

```
<keydef keys="math-fragment-0002" href="mathml/mathml-library.xml#math-fragment-02"/>
```

You reference this key by using just the key name:

```
<mathref keyref="math-fragment-0002"/>
```

Processing expectations

002 (16) Processors **SHOULD** process the MathML as though the <m:math> element occurs directly in the content of the containing <mathml> element.

Specialization hierarchy

The <mathmlref> element is specialized from <include>. It is defined in the MathML domain module.

Attributes

The following attributes are available on this element: [inclusion attributes](#), [universal attributes](#), @format, @href, @keyref, and @scope.

For this element:

- The @format attribute has a default value of "mml".
- The @href attribute is a reference to a MathML document or <mathml> element. If the <mathml> element is the root element of the referenced resource, then no fragment identifier is required. Otherwise, a fragment identifier must be specified, where the fragment identifier is the XML ID of the <mathml> element.
- The @parse attribute has a default value of "xml".

Examples

This section contains examples of how the <mathmlref> element can be used.

Figure 5: Referencing a MathML <math> root element

The following code sample shows how a <mathmlref> element can be used to reference a MathML <math> element that is the root element of its containing document:

```
<equation-block>
  <mathml>
    <mathmlref href="../mathml-source/mathml-root-mathml.mml"/>
  </mathml>
</equation-block>
```

The mathml-root-mathml.mml file contains the following content. Note that the <math> element sets the MathML namespace as the default namespace, so there are no namespace prefixes on the MathML markup.

```
<?xml version="1.0" encoding="UTF-8"?>
<math xmlns="http://www.w3.org/1998/Math/MathML" xmlns:xlink="http://www.w3.org/1999/xlink">
  <mstyle displaystyle="false" scriptlevel="0">
    <mrow>
      <mfrac>
        <mrow>
          <mi mathcolor="gray">sin</mi>
          <mo rspace="verythinmathspace"/>
          <mi>θ</mi>
        </mrow>
        <mi>π</mi>
      </mfrac>
    </mrow>
  </mstyle>
</math>
```

Figure 6: Referencing a specific <math> element within a document

The following code sample shows how a <mathmlref> element can reference a specific <math> element in a containing XML file:

```
<equation-block>
  <mathml>
    <mathmlref href="../mathml-source/mathml-equation-library.xml#mathfrag-02"/>
  </mathml>
</equation-block>
```

The mathml-equation-library.xml file contains the following content:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <part>
    <math id="timeinday" xmlns="http://www.w3.org/1998/Math/MathML">
      <mi>x</mi>
    </math>
    <math id="mathfrag-02" xmlns="http://www.w3.org/1998/Math/MathML">
      <math>
```

```
<mrow>
  <mi>y</mi>
  <mo>=</mo>
  <mn>5</mn>
  <mi>x</mi>
  <mo>+</mo>
  <mn>2</mn>
</mrow>
</math>
</part>
<!-- ... -->
</root>
```

3 SVG domain

The SVG domain elements enable **the use of embedded or referenced SVG markup. Referenced SVG markup must be stored in separate, non-DITA XML documents.** SVG is a W3C standard.

For SVG markup that is stored directly in DITA documents that are validated using DTDs, the SVG elements must use a namespace prefix in order to avoid conflict with DITA-defined elements of the same name. Documents validated using RELAX NG can default the SVG namespace on the SVG `<svg>` element. SVG elements that are referenced using the `<svgref>` element do not need to have a namespace prefix, because they are parsed separately from the DITA documents that refer to them. By default, the SVG domain is configured to use the namespace prefix "svg" for the SVG elements.

Related information

[Scalable Vector Graphics \(SVG\) 1.1 \(Second Edition\)](#)

3.1 `<svg-container>`

The `<svg-container>` element stores content that contributes to a scalable vector graphic (SVG).

Usage information

The `<svg-container>` element can contain SVG elements, references to SVG elements that are stored in separate, non-DITA documents, or `<data>` elements.

The SVG markup must have a root element of `<svg>` with the SVG namespace: "http://www.w3.org/2000/svg".

Specialization hierarchy

The `<svg-container>` is specialized from `<foreign>`. It is defined in the SVG domain module.

Attributes

The following attributes are available on this element: [universal attributes](#).

Example

The following code sample shows how `<svg-container>` elements can be used in a DITA topic. It is used to generate both inline SVG markup and a titled figure that contains SVG markup:

```
<topic id="svg-test-topic-01">
  <title>SVG Domain Test: Namespace Prefixed SVG Elements</title>
  <body>
    <!-- SVG inline -->
    <svg-container>
      <svg:svg width="100" height="100">
        <svg:defs>
          <svg:filter id="f1" x="0" y="0">
            <svg:feGaussianBlur in="SourceGraphic" stdDeviation="15"/>
          </svg:filter>
        </svg:defs>
        <svg:rect width="90" height="90" stroke="green" stroke-width="3" fill="yellow"
          filter="url(#f1)"/>
      </svg:svg>
    </svg-container>
    <!-- ... -->
  </body>
</topic>
```

```

<fig>
  <title>Figure with SVG container</title>
  <svg-container>
    <svg:svg width="4in" height="6in" version="1.1">
      <svg:circle cx="150" cy="200" r="100" fill="url(#grad_blue)"/>
      <svg:rect x="70" y="320" height="40" width="80" fill="aqua"/>
      <svg:text x="90" y="350" font-size="30" fill="green">Go</svg:text>
    </svg:svg>
  </svg-container>
</fig>
</body>
</topic>

```

3.2 <svgref>

The <svgref> element references a non-DITA XML document that contains scalable vector graphic (SVG) markup.

Usage information

The <svgref> element enables the use of SVG markup by reference. The reference must be to a SVG <svg> element **that is stored in a separate, non-DITA XML document**. The reference can be one of the following:

- A URI that addresses an XML document which has a SVG <svg> element as the root element
- A URI that addresses an XML document and contains a fragment identifier that is the XML ID of a <svg> element within the document

The reference can be direct, using the @href attribute, or indirect, using the @keyref attribute. For indirect referencing, only the key name should be specified. The ID of the <svg> element must be specified as part of the value for the @href attribute on the key definition.

For example, to refer to the <svg> element with the @id of "svg-fragment-02" within a larger document using a key reference, you would define the key in the following way:

```
<keydef keys="svg-fragment-0002" href="svg/svg-library.xml#svg-fragment-02"/>
```

You reference this key by using just the key name:

```

<svg-container>
  <svgref keyref="svg-fragment-0002"/>
</svg-container>

```

Processing information

003 (16) Processors **SHOULD** process the SVG as though the <svg> element occurs directly in the content of the containing <svg-container> element.

Specialization hierarchy

The <svgref> is specialized from <include>. It is defined in the SVG domain module.

Attributes

The following attributes are available on this element: [inclusion attributes](#), [link-relationship attributes](#), [universal attributes](#), and [@keyref](#).

For this element:

- The @format attribute has a default value of "svg".
- The @href attribute is a reference to an SVG document or SVG element. If the <svg> element is the root element of the referenced resource, then no fragment identifier is required. Otherwise, a fragment identifier must be specified, where the fragment identifier is the XML ID of the <svg> element.
- The @parse attribute has a default value of "xml".

Examples

This section contains examples of how the <svgref> element can be used.

Figure 7: Referencing an SVG that is a root element

The following code sample shows how an <svgref> element can be used to reference an <svg> element that is the root element of its containing document:

```
<fig>
  <title>Figure with an SVG container</title>
  <svg-container>
    <svgref href="media/svg/svg-graphic-01.xml"/>
  </svg-container>
</fig>
```

The `svg-graphic-01.xml` file contains the following content. Note that the <svg> element sets the SVG namespace as the default namespace, so there are no namespace prefixes on the SVG markup.

```
<?xml version="1.0" encoding="UTF-8"?>
<svg xmlns="http://www.w3.org/2000/svg" width="100" height="100">
  <defs>
    <filter id="f1" x="0" y="0">
      <feGaussianBlur in="SourceGraphic" stdDeviation="15"/>
    </filter>
  </defs>
  <rect width="90" height="90" stroke="green" stroke-width="3" fill="yellow"
    filter="url(#f1)"/>
</svg>
```

Figure 8: Referencing a specific SVG within a document

The following code sample shows an <svgref> element can be used to reference a specific <svg> element in a containing XML file:

```
<fig>
  <title>Figure with SVG container</title>
  <svg-container>
    <svgref href="media/svg/svg-library.xml#frag-0001" />
  </svg-container>
</fig>
```

The `svg-library.xml` file contains the following content:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <part>
    <svg id="frag-0001" xmlns="http://www.w3.org/2000/svg" width="100" height="100">
      <defs>
        <filter id="f1" x="0" y="0">
          <feGaussianBlur in="SourceGraphic" stdDeviation="15"/>
        </filter>
      </defs>
      <rect width="90" height="90" stroke="green" stroke-width="3" fill="yellow"
        filter="url(#f1)"/>
    </svg>
    <!-- ... -->
```

```
</part>  
</root>
```

A Aggregated RFC-2119 statements

This appendix contains all the normative statements from the DITA for Technical Content 2.0 specification. They are aggregated here for convenience in this non-normative appendix.

Item	Conformance statement
001 (6)	In this context, white-space content is considered equivalent to empty content. When the <code><equation-number></code> element has empty content, the equation number SHOULD be generated. When the <code><equation-number></code> element is not empty, the content SHOULD be used as the equation number. Processors MAY add punctuation or decoration to the number.
002 (9)	Processors SHOULD process the MathML as though the <code><m:math></code> element occurs directly in the content of the containing <code><mathml></code> element.
003 (13)	Processors SHOULD process the SVG as though the <code><svg></code> element occurs directly in the content of the containing <code><svg-container></code> element.

Index

D

domains elements
 SVG domain
 overview [12](#)

E

equation domain [3](#)
 equation-block [3](#)
 equation-figure [4](#)
 equation-inline [5](#)
 equation-number [6](#)
 overview [3](#)
equation-block [3](#)
equation-figure [4](#)
equation-inline [5](#)
equation-number [6](#)

M

MathML domain [8](#)
 `<mathml>` [8](#)
 `<mathmlref>` [9](#)
 overview [8](#)
`<mathmlref>` [9](#)

S

SVG domain [12](#)
 svg-container [12](#)
 `<svgref>` [13](#)
svg-container [12](#)
`<svgref>` [13](#)