

An OASIS DITA Adoption Technical Committee Publication

DITA 1.3 Feature Article: Don't (XML) Mention It: Tagging XML Constructs in DITA 1.3

Author: Keith Schengili-Roberts
On behalf of the DITA Adoption Technical Committee

Date: 18 Jul 2016

This is a Non-Standards Track Work Product and is not subject to the patent provisions of the OASIS IPR Policy.



Notices

OASIS (Organization for the Advancement of Structured Information Standards) is a not-for-profit, international consortium that drives the development, convergence, and adoption of e-business standards. Members themselves set the OASIS technical agenda, using a lightweight, open process expressly designed to promote industry consensus and unite disparate efforts. The consortium produces open standards for Web services, security, e-business, and standardization efforts in the public sector and for application-specific markets. OASIS was founded in 1993. More information can be found on the OASIS website at <http://www.oasis-open.org>.

The OASIS DITA Adoption Technical Committee members collaborate to provide expertise and resources to educate the marketplace on the value of the DITA OASIS standard. By raising awareness of the benefits offered by DITA, the DITA Adoption Technical Committee expects the demand for, and availability of, DITA conforming products and services to increase, resulting in a greater choice of tools and platforms and an expanded DITA community of users, suppliers, and consultants.

DISCLAIMER: All examples presented in this article were produced using one or more tools chosen at the author's discretion and in no way reflect endorsement of the tools by the OASIS DITA Adoption Technical Committee.

This white paper was produced and approved by the OASIS DITA Adoption Technical Committee as a Committee Draft. It has not been reviewed and/or approved by the OASIS membership at-large.

Copyright © 2016 OASIS. All rights reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website. This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English. The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns. This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Document History

Revision	Date	Author	Summary
First Draft	13 Jun 2016	Keith Schengili-Roberts	Initial draft
Second Draft	17 Jun 2016	Keith Schengili-Roberts	Incorporating initial review comments and those from a DITA Adoption Committee member.
Third Draft	6 Jul 2016	Keith Schengili-Roberts	Incorporating additional review comments and suggestions from DITA Technical Committee members.

Revision	Date	Author	Summary
Fourth Draft	15 Jul 2016	Keith Schengili-Roberts	Incorporating final review comments and suggestions from DITA Technical Committee members.
Final	18 Jul 2016	Keith Schengili-Roberts	Approved

Table of Contents

Don't (XML) Mention It: Tagging XML Constructs in DITA 1.3.....	5
Introduction to the XML Mention Domain.....	5
Writing tags with <code><xmlement></code> and attributes with <code><xmlatt></code>	5
Referencing numeric characters and text entities with <code><numcharref></code> and <code><textentity></code>	5
Describing parameter entities with <code><parameterentity></code>	6
Referencing processing instructions using the <code><xmplpi></code> element.....	6
Using <code><xmlnsname></code> to describe namespaces.....	6
When and how to use <code><markupname></code>	7
Conclusion.....	7

Don't (XML) Mention It: Tagging XML Constructs in DITA 1.3

Introduction to the XML Mention Domain

A minor frustration with DITA was an inability to easily describe XML constructs such as element and attribute names. If you have ever found yourself using the angle bracket entities and wishing that there was another, more elegant way to describe your XML code, then look no further than the DITA 1.3 XML Mention domain of elements available in DITA 1.3. This domain now allows content authors to mark up XML content for XML elements and attributes, numeric character references, text entities, parameter entities, processing instructions, and namespace names. Two key advantages of this approach are that you no longer have to "escape" inline XML references and you have the means to more effectively search for content that is semantically described by the XML Mention elements. So no more looking through every angle bracket reference, percentage symbol, ampersand, or other characters typically used to format XML content in the hope that you will eventually find what you are looking for. This white paper focuses on how the DITA 1.3 XML Mention elements can be used.

Writing tags with `<xmlelement>` and attributes with `<xmlatt>`

If you have ever written an internal DITA style guide detailing how to use various DITA elements and attributes, you will definitely appreciate the new `<xmlelement>` and `<xmlatt>` elements.

For example, let's say you are writing an internal DITA style guide and want to indicate that your writers should avoid using the non-semantic bold and italic elements in their DITA content. Instead of using angle bracket entities to describe what you mean, you can use `<xmlelement>`. At output, the latest version of the DITA Open Toolkit (DITA-OT) uses a different font and wraps angle brackets automatically around the described element. For example:

```
<p>Please avoid using the <xmlelement>b</xmlelement> (bold) and <xmlelement>i</xmlelement> (italic) tags in your content. Here at ACME we use semantic markup whenever possible.</p>
```

Similarly, if you want to indicate in your DITA style guide which note attributes to use and not to use, you can indicate attribute names using `<xmlatt>`. At output, the default behavior of the DITA-OT is to display the attribute value using a different font and preceded by the "@" character. The following example shows how `<xmlatt>` might be used in practice:

```
<p>When using the <xmlelement>note</xmlelement> element in ACME documentation, use the following <xmlatt>type</xmlatt> values: "caution" to inform users when they need to take care before proceeding with a task, "warning" to indicate a potentially hazardous situation for the user, and "danger" for scenarios that might involve injury or death. ACME documentation also uses the "tip" and "important" <xmlatt>type</xmlatt> attribute values for <xmlelement>note</xmlelement>.</p>
```

Referencing numeric characters and text entities with `<numcharref>` and `<textentity>`

Prior to DITA 1.3, if you wanted to show a numeric character reference in your document, you had to "escape" the leading ampersand so that it didn't actually render the character itself. If, for example, you wanted to show a reference to the Greek capital letter sigma (used as a summation operator in mathematics) inline in your content, you had to write it as `Σ` (decimal base) or `Σ` (hexadecimal base) so that it would render as "`Σ`" and "`Σ`" respectively, and not as the character (" Σ ") itself. The new `<numcharref>` element in DITA 1.3 makes this action easier, eliminating the need for explicitly referencing the leading ampersand and octothorpe and trailing semicolon characters that are inserted at output by the DITA-OT. The following sample code demonstrates how `<numcharref>` can be used:

<p>To include the summation operator symbol, simply type <numcharref>931</numcharref> or <numcharref>x3A3</numcharref> in the text field.</p>

Similarly, the new <textentity> element in DITA 1.3 allows you to show text entities the same way. By default, at output the DITA-OT adds a leading ampersand and a trailing semi-colon to the text entity it encloses, just as with <numcharref>. The following example shows how this element can be used:

<p>Thanks to DITA 1.3, you no longer have to wrap the less-than (<textentity>lt</textentity>) or greater-than (<textentity>gt</textentity>) text entities around elements to describe them in your DITA content. You can simply use <xmlelement>xmlelement</xmlelement> instead.</p>

Describing parameter entities with <parameterentity>

Parameter entities are used to define a collection of elements, attributes, and their values in XML. If you have ever looked at the .mod or .ent files of complex DTD, you have seen parameter entities. As an example, the parameter used to define the paragraph element in DITA is called "%p.content;". If you are trying to explain aspects of a DTD, you may need the new <parameterentity> element. By default the latest version of the DITA-OT wraps a leading percentage symbol and trailing semi-colon around the parameter entity. The following is an example of how it can be used in practice:

<p>For example, the parameter entity used to define the paragraph element in DITA is called <parameterentity>p.content</parameterentity>.</p>

Referencing processing instructions using the <xmippi> element

In XML, processing instructions are used to carry information on how a processor should work with a defined instruction. An XML processing instruction typically contains a name followed by an optional value. While processing instructions are not part of the DITA standard, some authoring tools enable their use within DITA. Additionally, processing instructions are used in other types of XML, such as DocBook or XSL. An example of a processing instruction in XSL looks like this: <?xml-stYLESHEET type="text/xsl" href="style.xml"?>, where <?xml-stYLESHEET?> is the name of the processing instruction and the remainder is a value that describes how the processor should interpret and work with it. To describe processing instruction names in DITA 1.3, use the <xmippi> element. At output, the default behavior of the DITA-OT is to display its contents using a different font. The following example shows how it can be used:

<p>In DocBook processing instructions that begin with <xmippi>dbfo</xmippi> can be used for formatting operations. An example of how this can be used to set the width for a two-and-a-half inch sidebar would look like this: <codeph><?dbfo sidebar-width="2.5in"?></codeph>.</p>

Using <xmllnsname> to describe namespaces

In XML, a namespace is a mechanism designed to avoid clashes between elements or attributes from more than one XML vocabulary. For example, HTML and DocBook both have a "table" element just like DITA does, and if you wanted to be clear as to which table element you are referencing when you are using more than one XML instance of table, you would do so by declaring the namespace that this table belongs to. The following example shows how it might be used in practice:

<p>In MathML, the <xmlelement>m:math</xmlelement> is the root element. The "m:" prefix is bound to the MathML 2.0 namespace: <xmllnsname>http://www.w3.org/TR/2003/REC-MathML2-20031021/</xmllnsname>.</p>

By default, at output the DITA-OT formats anything contained within <xmllnsname> in a different font.

When and how to use <markupname>

While not part of the XML Mention domain, it's worth including the DITA 1.3 <markupname> element is relevant here because all of the other elements within the XML Mention are specialized from it. In general, it is recommended that the other, more specific XML Mention elements be used whenever possible or that <markupname> be used for describing XML markup that is not covered by the other elements in that domain. If a given DITA document type does not include the XML Mention domain but does include the markup name domain, <markupname> can act as a generic descriptor for XML markup. Here's an example of it in practice:

```
<p>In commonElementsMod.rng, the <markupname>basic.ph</markupname> pattern provides a choice of several inline elements.</p>
```

There is no guidance in the DITA 1.3 specification for formatting <markupname> in output. It semantically describes XML markup but does not prescribe how it should appear. By default, the DITA-OT formats content contained within <markupname> in a different font.

Conclusion

The addition of the XML Mention domain makes it much easier for content creators to mark up mentions of the names of XML constructs within their documentation without having to resort to using awkward entities.

For more information on the XML Mention domain or anything else relating to DITA 1.3, head to the OASIS website [for the official specification](#).

Thanks goes out to Jason Owen, Leigh White, Bob Thomas, Eliot Kimber, Tom Magliery, JoAnn Hackos, Nancy Harrison and Stan Doherty for their editorial assistance with this article.