

**An OASIS DITA Adoption Technical Committee Publication**

## **DITA 1.3 Feature Article: DITA 1.3 from A-Z**

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

Date: 25 Aug 2017

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



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 © 2017 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	3 Mar 2015	Keith Schengili-Roberts	Initial draft
Second Draft	17 Mar 2015	JoAnn Hackos	Copyedit pass
Third Draft	5 Jun 2015	Keith Schengili-Roberts	Update against latest DITA 1.3 draft
Fourth Draft	15 Jun 2015	Keith Schengili-Roberts	Edits based on JoAnn Hackos' follow-up comments.

Revision	Date	Author	Summary
Fifth Draft	9 Jul 2015	Keith Schengili-Roberts	Edits and content added based on additional comments.
Sixth Draft	20 Jul 2015	Keith Schengili-Roberts	Edits based on Stan Doherty's additional comments.
Seventh Draft	16 Dec 2015	Keith Schengili-Roberts	Extensive edits and additions based on feedback from the OASIS DITA Technical Committee.
Eighth Draft	6 Jun 2016	Keith Schengili-Roberts	Used DITA 1.3 XML Mention domain in content. Also an additional edit round based on further feedback from OASIS DITA Technical Committee members.
Ninth Draft	7 Apr 2017	Keith Schengili-Roberts	Used DITA 1.3 XML Mention domain in content. Incorporated additional edits based on feedback from Kris Eberlein and Robert Anderson.
Tenth Draft	25 Aug 2017	Keith Schengili-Roberts	Incorporating additional edits from Nancy Harrison, fixed typos.

## Table of Contents

DITA 1.3 from A-Z.....	5
Accessibility Markup for Tables.....	5
Branch Filtering.....	5
Cascade Attribute.....	6
Context-Sensitive Help.....	7
deliveryTarget.....	7
Grouping Things Using div.....	7
Highlight Domain.....	8
Inline Elements for Describing Markup Language.....	8
Key Scopes.....	8
Learning and Training Specialization.....	8
MathML and SVG Now Natively Supported.....	9
RELAX NG.....	9
Release Management.....	9
Same Topic References.....	9
Sorting Element.....	9
Table and Cell Rotation Now Possible.....	10
Task for Troubleshooting.....	10
Troubleshooting Note Type.....	10
Troubleshooting Topic.....	10
Other Updates to Individual DITA 1.3 Elements and Attributes.....	10

## DITA 1.3 from A-Z

This article describes the significant additions and changes appearing in the DITA 1.3 specification. It is designed to provide a broad overview of the features of the specification rather than providing in-depth information on each subject.

The information in this article is based upon the latest draft version of the All-Inclusive Edition of the draft DITA 1.3 specification, which contains all of the OASIS-approved specializations for DITA, including the Learning & Training specialization. In a working DITA 1.3 environment, you may not need or have access to all of the domains whose elements are described here. For example, if your firm does not use the Learning & Training specialization, the elements described in this article belonging to that domain may not be available for use. Similarly, an Information Architect can use constraints to restrict other DITA 1.3 elements from use.

The DITA 1.3 specification document is intended primarily for technology implementers, such as developers at software firms building tools that will work with the standard. The rest of us—Content Creators, Information Architects, Technical Documentation Managers, etc.—need a plain-language overview of what DITA 1.3 offers in terms of new features to expect and to plan for. This article is intended for the latter group.

## Accessibility Markup for Tables

---

For those using Braille or spoken-word browsers, table-based content can be hard to understand, as it may not be easy to keep track of what the column or row headings are for the cells in the table. While table markup has been fully accessible since DITA 1.0, there are some tools that may not insert those elements automatically or correctly, and there are some situations where table header usage may be atypical, such as when a single header cell or header column is present in the middle of a table. In these types of cases, content creators now have additional tools at their disposal to ensure that header information is properly associated with cell content for accessible browsers.

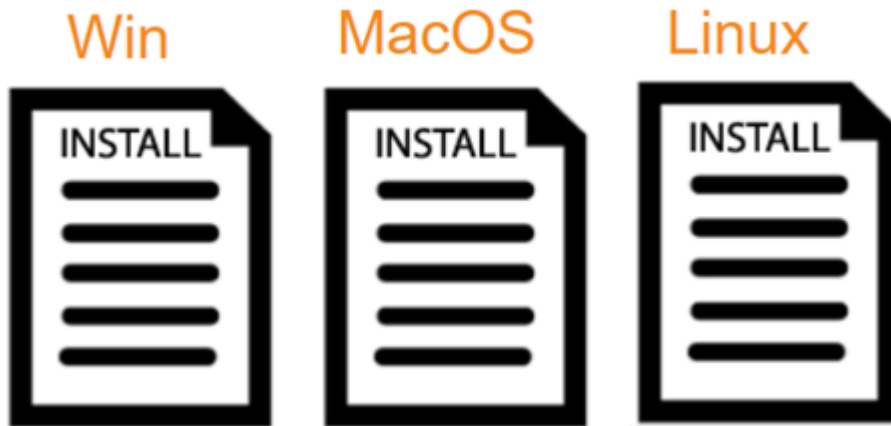
Content creators can now explicitly indicate the id attribute value of the header or column that should be associated for an individual cell entry. For example, a row with three cells might be read aloud by a browser as "123, 7:00, 7:30". Without context, the intent of this information is unintelligible. The table accessibility elements allow for the manual association of a description for the header or column to provide more context, so a spoken-word browser could now say something like the following instead: "Flight 123, Boarding 7:00, Expected Departure 7:30".

## Branch Filtering

---

The DITA 1.2 ditaval element allows content creators to set output conditions for their content. The new DITA 1.3 element ditavalref allows you to set unique conditions on a topicref (and its descendants) within a map, allowing for publications that reflect different conditions for different branches of the map. Branch filtering can be applied broadly, ranging from filtered material within a single paragraph in a topic, or to an entire branch of topics. ditavalref extends topic-level reuse by allowing a topic used multiple times in the map to reflect different filtering conditions and thus produce different output for each place the topic is used.

Under DITA 1.2 it is possible to publish the map showing all of the "Windows" settings using a specific ditaval set for that operating system, and again using a separate "MacOS" ditaval, and once more using a ditaval set to "Linux", as depicted in the following illustration:



**Figure 1: Producing separate publications from one map using DITA 1.2 ditaval**

When using the DITA 1.3 ditavalref element it is now more easily possible to combine all of this information so that the operating system content appears in sequence within a single document by using the same-map, separate-condition feature of branch filtering. Using the previous example of the software manual, using ditavalref a single publication can be outputted so that where the content for each of these operating systems appear in sequence within the same publication, which is illustrated in the following image:



**Figure 2: Producing a single document with sequentially-filtered content using DITA 1.3 ditavalref**

This same sequencing within a single publication could also be accomplished under DITA 1.2, but only by using longer and more verbose DITA code. DITA 1.3 branch filtering allows for greater flexibility and more concise tagging.

## Cascade Attribute

---

In DITA 1.2, metadata values were always additive by default. If you have a map that has a metadata value attached to it (for example platform="Windows"), and then you have a topicref that brings in another metadata value (such as platform="MacOS") that topicref would inherit the metadata value from the map, even if that was not the intention.

The new cascade attribute in DITA 1.3 allows content creators to disable this default behavior. If the map element that specifies platform="Windows" also specifies cascade="nomerge", the default cascading behavior for any descendant topicrefs is turned off. So, if a specific topicref is now set to platform="MacOS" it no longer inherits the "Windows" metadata value from the map, which was the previous default under DITA 1.2. The cascade attribute provides content creators with more flexibility for running processing routines based on metadata.

## Context-Sensitive Help

---

DITA support for various forms of Context-Sensitive Help (CSH) systems has been enhanced, allowing content creators to set the attributes for a specific callback ID and its context. Positional information for a particular target display can be set using the new ux-window element and then referenced by an attribute or resourceid within a topic, or by the topicref. Content creators can also establish the priority setting of a CSH instruction that could, for example, ensure that the ux-window attribute value specified within a topic overrides a more general setting for that same CSH system referenced at the map level. Help practitioners can now encode callback IDs and target window definitions for multiple target Help environments within the same set of DITA source files.

## deliveryTarget

---

The print attribute has always had two practical limitations:

- The limited set of values (yes, no, or printonly) was too limiting to address the increasingly diverse range of available output options
- The processor determined the filtering behavior. For example, with print="no" a processor would likely exclude a topic from PDF and include it for XHTML, but its choice for EPUB might not be what is expected.

In DITA 1.3 the print attribute has been deprecated in favor of deliveryTarget, allowing for any conceivable output type or device to be targeted. deliveryTarget accepts one or more CDATA values, opening things up so that future output formats yet to be devised can be referenced.

deliveryTarget acts as a conditional processing attribute, so it can be set on a map or to a topicref to indicate which types of output format are to be used (or should not be used, depending on what the output conditions are). And whereas print could only be used in maps, deliveryTarget can be used in maps or topics.

For example, <topicref deliveryTarget="html5 epub3 kindle" href="sample.dita"> has three values set for deliveryTarget. A conditional profile can then be set to determine whether sample.dita is included or excluded when the map it is contained within is rendered.

Information architects can use subjectScheme maps to control the values for the deliveryTarget attribute. These controlled values can populate selection lists in the authoring environment; they also can be used to validate the values used for the deliveryTarget attribute at publishing time.

Using deliveryTarget does *not* remove the need to create specific transforms for each output type.

## Grouping Things Using div

---

Content creators can use the new block-level div element to create arbitrary groupings of content within a topic. It works in the same way as its HTML equivalent, defining an arbitrary block section. In HTML div is often used to group content which can then be formatted using CSS. In DITA its primary purpose is to be used for organizing content that can then be referenced by a conref or a conkeyref. It can also be used by Information Architects as a convenient block-level base for specialization.

## Highlight Domain

---

The highlighting domain of elements are designed to indicate text that should be rendered with a particular typographic effect. This domain includes the elements bold (b), italic (i), and underline (u) among others. DITA 1.3 adds two new elements to this domain: line-through and overline. The line-through element indicates text that may no longer be relevant but needs to be displayed. Text marked up with overline is used in electrical engineering documentation to indicate active-low logic elements, and in a mathematical context to indicate that certain symbols belong together.

## Inline Elements for Describing Markup Language

---

The new XML mention domain contains several inline elements created to describe XML markup:

- elements, using `xmlelement`
- attributes, using `xmlatt`
- namespace names, using `xmlnsname`
- processing instructions, using `xmlpi`
- numeric character entities, using `numcharref`
- parameter entities, using `parameterentity`
- text entities, using `textentity`.

For example, if you want to describe information about an XML element, you will no longer have to use entity characters for angle brackets (lt for <, and gt for >), but can instead wrap a pair of `xmlelement` elements around it and the output processor will apply the appropriate formatting.

## Key Scopes

---

DITA 1.3 brings with it the ability to "scope" keys using the new `keyscope` attribute, meaning that different key values can be "scoped" to specific locations within a map. If you have a topic that is being reused, content creators can specify different key bindings for the same key name depending upon where in a publication the topic is referenced. Key scopes help to maximize the reuse possibilities for keys in topics. It is now possible to use one key in a topic that reflects different values depending on the defined scopes of the `topicrefs` that include the topic.

Scoped keys also allow for cross-deliverable linking. In DITA 1.3 you can define a map as being "peer" to another, so that the key values belonging to one map can be used by the other and vice-versa. Scoped keys allow key references in one map to be resolved based on a peer map, which can be used to generate cross references into the other map or publication.

With key scopes it is now easier to do the following:

- Combine multiple standalone publications with overlapping key names into a single, omnibus publication. Simply set the `keyscope` attribute on the references to each map, preventing their keys from interfering with each other.
- Create a "mail merge" publication. Create a single map with key scope-defining wrappers around each instance of the template topic and the appropriate key definitions.
- Create multi-product documentation, where each section contains some common key names. Put key scopes around the parts of the map that pertain to different products.
- Use keys to create cross-references to content contained within a peer map.

## Learning and Training Specialization

---

In DITA 1.2, content creators were restricted only to phrase-level elements in the question statements, answer options, and feedback components of questions (interactions). This meant that there was no possibility for adding figures, lists, tables,



etc. For DITA 1.3 the new learning2 domain allows paragraphs and other block-level elements within question statements, answer options, and feedback, providing greater flexibility for content creators.

Content creators can additionally aggregate learning content, supporting files, and assessments into the new learningObjectMap element, and similarly, the learningGroupMap element can be used to collect together learning objects and group elements.

## MathML and SVG Now Natively Supported

---

Before DITA 1.3, anyone using MathML (a specification for writing equations in XML) had to insert mathematical notations by integrating the MathML standard into customized versions of their DITA DTD or Schema. MathML is now defined as a standard DITA 1.3 domain, so it can be used natively, using either the mathml element for containing MathML code directly within a topic, or mathmlref to reference the URI of a file containing MathML markup.

Similarly, inserting Scalable Vector Graphics (SVG) images—another, separate XML standard used for creating vector-based illustrations—is also available as a DITA 1.3 domain and can be used natively. Previously anyone wanting to use SVG images had to use the image element to point to them, and support for this graphic type was spotty with some output processors. Content creators wanting to insert SVG images into their topics can now use the svg-container element to hold native SVG code, or use svgreg to reference the URI of a SVG image file.

## RELAX NG

---

The XML rules for previous versions of DITA were defined using Document-Type Definitions (DTDs) or W3C's XML Schema formats. With DITA 1.3, the rules are now defined using RELAX NG (REgular LAnguage for XML Next Generation). Ideally, programs used for writing or processing DITA 1.3 content will be able to validate DITA topics based on this standard. RELAX NG makes defining DITA document type shells and specializations easier because its syntax is better suited to the task of defining DITA grammars. DITA 1.3 will continue to be made available in DTD and Schema formats, but the RELAX NG version is the normative version from which the others are derived.

## Release Management

---

The new release management domain in DITA 1.3 provides markup element that can be used to capture release notes information within maps and topics. Summary information on significant changes to a product can now be added to the prolog of a topic or within the topicmeta of a map. If information is added as topics or maps are authored, content creators will no longer have to rely on memory alone to construct release notes. The date and time information embedded in the release management content allows processors to programmatically select only those comments that are pertinent to a given release, while still keeping information about all successive significant changes over the development of a product. Additional elements are provided to reference bug-tracking systems, identify change request details, and provide other external release information.

## Same Topic References

---

In DITA 1.3, it is now possible to link to an ID located elsewhere within the same topic by simply using a period in the URI after the hash character instead of having to include the topic ID. If a content creator wanted to link to a sample figure contained elsewhere in a topic, the link would look like this: `<xref="#./subsection" type="fig">more information on this subject is available</xref>`.

## Sorting Element

---

The new sort-as element is designed to help processors to sort content where the base method for sorting is not obvious, such as for words written in Chinese/Japanese/Korean characters where there is no inherent alphabetical order. A content

creator can now add the phonetic sound of a word and have the processor sort things like glossary entries, titles, or an index based on the spelled-out sound. For example, the Chinese word for "hello" could be sorted in a glossary by its sound ("ni hao") as: `<glossterm><sort-as value="ni hao"/>你好</glossterm>`, placing alongside other Chinese words starting with an "n" sound.

## Table and Cell Rotation Now Possible

---

Two attributes have been added to DITA 1.3 to aid content creators who need more control over the print formatting of tables. `table` can now take the attribute `orient` that instructs processors to print tables in landscape mode. Individual cell entries can take the new `rotate` attribute that can turn content 90 degrees counterclockwise.

## Task for Troubleshooting

---

The new `tasktroubleshooting` element has been added to the `task` topic type to assist readers who may encounter a common problem when attempting a task. This optional element appears at the conclusion of a task, along with `result`, `example`, and `postreq`. `tasktroubleshooting` must appear after a `result` and before an `example` and/or `postreq`.

Task topics can also take the optional troubleshooting-related element `steptroubleshooting`. `steptroubleshooting` can be used after `cmd` or `stepresult` at the end of a particular step for situations when a user is likely to encounter a problem when performing that portion of a task.

## Troubleshooting Note Type

---

A new attribute value is available for the `note` element: `trouble`. This lets authors create notes that are expressly designed to inform readers of a potential problem and how to resolve it. For example, `<note type="troubleshooting">Use an ice-scraper to remove ice and snow that has accumulated on the windshield.</note>`

## Troubleshooting Topic

---

DITA 1.3 adds a new topic type: `troubleshooting`. Its main structural elements (excluding elements common to other topic types) consist of the following:

- `condition` – describing a symptom that a user may encounter
- `troubleSolution` – one or more of these, describing a possible resolution to the issue
- `cause` – an optional follow-up for `condition` or `troubleSolution` describing a possible reason for the problem
- `remedy` – an optional, step-wise description of how to fix the problem, which can also include the optional `responsibleParty` element indicating who is expected to follow the remedy's steps

The new `troubleshooting` topic is designed to provide better structure for content based on the long-established troubleshooting pattern, instead of trying to shoe-horn it into another topic type. It also enables content creators to focus on addressing and solving specific problems a user may encounter, and provides an opportunity for labeling troubleshooting content as such, making it easier for users to find. It also provides an ideal launching point for those wanting to specialize other flavors of troubleshooting topics.

## Other Updates to Individual DITA 1.3 Elements and Attributes

---

- The `cite` element can be added to content within a title, allowing for citation references within a heading.
- The `data` element can now hold more comprehensive information, providing the option of adding citations, links, or short quotations to its content. `data` and the `data-about` elements can also be used as child elements of any list type, such as ordered or unordered lists.

- For many elements, the attribute values for format and scope—commonly used in links—can now be inferred based on the value of href. For example, this means that links to a file with a ".html" extension will automatically be treated as links to an HTML file, without the need to set format.
- Use of the draft-comment element has been expanded and now can be used in more places.
- The inline text element can now be used just about anywhere, and provides a flexible alternative to using ph when content creators want to mark up phrases for reuse.
- Content creators can now add the ph, sub, sup elements (and any other specializations of ph) to individual indexterm elements.
- Keyrefs can now be used with the object and param elements.
- The prop and revprop elements can now take additional style values.
- Titles can now take the rev (revision) attribute.
- You can now add an xref link within a shortdesc.
- Filtering attribute values can now be placed into a parenthetical group whose individual values can be selected against.