

An OASIS DITA Adoption Technical Committee Publication

DITA 1.3 Feature Article: User Assistance Enhancements in DITA 1.3

Author: Chris Goolsby
On behalf of the DITA Adoption Technical Committee

Date: 1 August 2016

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 © 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	18 June 2016	Goolsby	Initial draft
Final Draft	1 August 2016	Goolsby	Final draft

Table of Contents

User Assistance Enhancements in DITA 1.3.....	4
Introduction.....	4
Enhanced <resourceid> Metadata for Maps and Topics.....	4
Target Display Information.....	6
Conclusion.....	8

User Assistance Enhancements in DITA 1.3

Introduction

DITA 1.3 provides enhanced support for context-sensitive help and other types of user assistance (UA). Previously, the only support for help was in the `<resourceid>` element in a topic's `<prolog>` or a map's `<bookmeta>` and `<topicmeta>` elements. Attributes on `<resourceid>` enabled you to provide a help ID and the name of an application associated with that ID. While you could have any number of `<resourceid>` elements, this support did not cover many other aspects of context-sensitive help systems.

For example, there was also no way to resolve conflicts between `<resourceid>` information defined in a map `<topicref>` element and `<resourceid>` information defined in a topic `<prolog>`. The 1.3 release corrects this and enables you to provide much more help-related information directly in your DITA source.

The following UA enhancements are available in the 1.3 release:

- Enhanced `<resourceid>` metadata for maps and topics

New `<resourceid>` attributes enable you to specify any number of help targets in the same DITA source. A help target is a topic or other content to be displayed when help is called from a particular place in the user interface. You can identify the target based on a context ID, a string, or both at the same time. You can also specify how to resolve any conflicts between the metadata in a map and a topic.

- Target display information

The new `<ux-window>` element for use in maps enables you to provide very specific information about how to display the window containing the help target. This information is referenced from the new `<resourceid>` `@ux-windowref` attribute.

While these enhancements provide additional support for traditional tri-pane help systems, they also give you great flexibility to support and manage new types of help systems being developed both now and in the future. For example, you can now support the following use cases:

- One set of DITA source supports multiple help applications. Each application might have its own way to handle and process help metadata, but you can support them all in your source.
- One set of DITA source supports multiple product contexts. Again, each context can have its own help metadata defined in the source.
- One set of DITA source supports multiple display environments. For each display environment, you can provide clear instructions on how to position your help window and what features should be available. Whether tri-pane help, embedded user assistance, wiki help, or whatever new help systems the industry spawns, you can provide the needed display information right in your source.

Note that support for these UA enhancements is dependent on your DITA processor. Check to see whether your processor has been updated to support the new markup before using it.

Enhanced `<resourceid>` Metadata for Maps and Topics

With the DITA 1.3 UA enhancements, you can easily support multiple help applications and platforms. The new mapping metadata supports the many different ways that an application might call a help topic. It also enables you to resolve any conflicts between the metadata in a topic and its map. This flexibility enables you to provide the right guidance whether your help is called from an iPhone, an Android phone, a desktop program, or any other device or application.

The `<resourceid>` element has the following new attributes:

- `@appid` – Relates the topic to the specified ID value.

Note that the @id attribute has been deprecated in 1.3. You should use this new attribute instead.

- @ux-context-string – Relates the topic to the specified context-string.
- @ux-source-priority – Specifies how to resolve conflicts between UA callbacks defined in a map or in a topic.

You can use the following values with this attribute:

- topic-and-map – Use the IDs from both the topic and the topic reference.
- topic-only – Use the IDs from the topic and ignore any IDs in the topic reference.
- map-only – Use the IDs from the topic reference and ignore any IDs in the topic.
- map-takes-priority – Use the IDs from the topic reference if present, otherwise use the IDs from the topic.
- topic-takes-priority – Use the IDs from the topic if present, otherwise use the IDs from the topic reference.

You can now specify callback help IDs using either the @@appid or @ux-context-string attributes. If a context-sensitive help system uses name-value pairs to define the ID, you can use both attributes to define the ID.

These new attributes also support the various ways that context-sensitive help systems map topics to user interface elements. A system could use one of the following mapping relationships:

- one-to-one – Where a user interface element is associated with a specific topic
- one-to-many – Where a user interface element is associated with a number of topics
- many-to-one – Where many user interface elements are associated with the same topic

The mapping is often implemented using a key, which could be an ID or a name-value pair. For example, a menu item might have a help ID of 236, and the associated UA topic has a matching ID of 236. In a many-to-one relationship, each user interface element might have a different ID but still resolve to the same topic. You accomplish this type of mapping through an “alias,” enabling one topic to have many IDs.

In a single-source environment, a topic might be used in different context-sensitive help systems for different products. For example, a topic explaining a **Print** dialog box might be used in several help systems. This raises the possibility of mapping conflicts, such as when the dialog box help ID in one application is 876 but in a different application the dialog box ID is 395. In this case, 876 might be the ID for a completely different user interface element. The @ux-source-priority attribute provides a way to avoid these potential conflicts by enabling you to specify which ID should be used when conflicting IDs are specified in both the map and the topic.

For Example

The following example shows how you can use these new attributes in a DITA map:

```
<map>
...
<topicref href="dialog-1.dita">
  <topicmeta>
    <resourceid
      appname="ios7a"
      <b>appid="iphone"</b>
      <b>ux-content-string="callback_4437"</b>
      <b>ux-source-priority="map-takes-priority"</b>/>
    <resourceid
      appname="kitkat"
      <b>appid="droid"</b>
      <b>ux-content-string="id#4500"</b>
      <b>ux-source-priority="map-takes-priority"</b>/>
    </topicmeta>
  </topicref>
...
</map>
```

In this case, the <resourceid> metadata supports two outputs: one for Apple and one for Android. You can add any number of additional targets for the referenced topic.

The topic might also have `<resourceid>` metadata defined in the `<prolog>` element:

```
<topic>
...
<prolog>
  <resourceid
    appname="kitkat"
    <b>appid="droid"</b>
    <b>ux-content-string="id#4501"</b>/>
  </prolog>
...
</topic>
```

Note that the `@ux-context-string` value defined in the topic is different from the one defined in the map. However, since the map metadata has `ux-source-priority="map-takes-priority"`, the help system can easily resolve this conflict in favor of the map's metadata.

Target Display Information

In addition to supporting help topic mapping from multiple help systems and platforms, the DITA 1.3 UA enhancements enable you to provide display information for the help windows on those systems and platforms. You can now put different target display information in your DITA source for every help platform you support. The display metadata is directly associated with the mapping metadata providing a full set of help metadata for each platform.

DITA maps now contain the new `<ux-window>` element to provide help window metadata. The `<ux-window>` element is contained in the `<topicmeta>` element. You can include any number of `<ux-window>` elements in the map enabling you to define window information for any number of help targets.

The `<ux-window>` metadata is referenced from a `<resourceid>` element. The `<resourceid>` has a new attribute `@ux-windowref` that provides the association to the matching `<ux-window>` element.

The `<ux-window>` element has the following attributes:

- `@name` – Identifies the `<ux-window>` element.

Use this attribute to provide a unique identifying name for this set of help window information. The value you provide in this attribute is the one used in the `<resourceid>` element's `@ux-windowref` attribute to associate this metadata with a specific `<resourceid>`.

This attribute is required.

- `@top` – Provides the top position of the target help window.

This position is relative either to the calling window or the entire display, determined by the value of the `@@relative` attribute. The value of the `@top` attribute is a real number with an optional unit of measure from the following set:

- `pc` – picas
- `pt` – points
- `px` – pixels
- `in` – inches
- `cm` – centimeters
- `mm` – millimeters
- `em` – ems

The default value is pixels.

- `@left` – Provides the left position of the target help window.

This position is also relative either to the calling window or the entire display and uses the same units of measure as the `@top` attribute.

- `@height` – Provides the height of the target help window.

This attribute uses the same units of measure as the @top attribute.

- @width – Provides the width of the target help window.

This attribute uses the same units of measure as the @top attribute.

- @on-top – Determines whether the help window should appear on top of all other windows on the display.

Allowed values are “yes”, “no”, and “-dita-use-conref-target”. The default is “no”.

- @relative – Determines whether the help window position is relative to the calling window or the entire target display.

The default is “no” indicating that the help window position is relative to the entire display. Set this attribute to “yes” to indicate the help window position is relative to the calling window. You can also set the value to “-dita-use-conref-target”.

- @full-screen – Determines whether the help window is initially displayed full screen.

Allowed values are “yes”, “no”, and “-dita-use-conref-target”. The default is “no”.

- @features – Provides a list of other features for the help window.

This attribute does not have a set list of values. Instead, you can use this attribute to provide any number of other help feature names and values that are supported by your help system. For example, you could indicate whether the help window should display a menu bar or provide some sort of status information for the window. Whatever other help window metadata is needed for your help system can be provided in this attribute.

Each feature name and value must not contain any blank spaces. Each feature must be separated by a comma or other delimiter character.

Note that the target display metadata is similar to that in the JavaScript window.open method’s specs parameter. You can consult this parameter for potential help window features to use in the @features attribute.

For Example

The following example builds on the example from the mapping metadata topic by providing target display metadata for the <resourceid> elements in the DITA map:

```
<map>
...
<b><ux-window
  name="apple"
  top="10"
  left="20"
  height="400"
  width="500"
  relative="yes"
  full-screen="no"
  features="toolbar=no,menubar=yes,status=yes"/>
<ux-window
  name="android"
  top="15"
  left="15"
  height="500"
  width="600"
  relative="yes"
  full-screen="no"
  features="toolbar=yes,menubar=no,status=no"/></b>
...
<topicref href="dialog-1.dita">
  <topicmeta>
    <resourceid
      appname="ios7a"
      appid="iphone"
      ux-content-string="callback_4437"
      ux-source-priority="map-takes-priority"
      <b>ux-windowref="apple"</b>/>
    <resourceid
      appname="kitkat"
      appid="droid"
```



```
ux-content-string="id#4500"  
ux-source-priority="map-takes-priority"  
<b>ux-windowref="android"</b></>  
</topicmeta>  
</topicref>  
...  
</map>
```

Note that both of the <resourceid> elements now have associated <ux-window> elements providing the target window information for these help callbacks.

Conclusion

DITA now enables you to include any user assistance metadata you need directly in your document source. You no longer need to rely on external Help Authoring Tools (HATs) to provide this support. DITA also now enables you to reuse your source for any number of help systems and ensures that the right callback and help window display information is available for whatever type of platform you need to support.