Review O: Accessibility

Copyright © OASIS Open . All Rights Reserved.

Table of contents

1 Accessibility 3
1.1 Handling accessibility in content and in processors 3
1.2 Accessible content 5
1.3 Accessible tables 6
1.4 Examples of DITA markup for accessibility 8
1.4.1 Example: Alternate text for an image 8
1.4.2 Example: Alternate text for an image map 9
1.4.3 Example: Fallback information for multimedia content 10
1.4.4 Example: Complex table with accessibility markup 10
1.4.5 Example: Complex table with some manually-specified accessibility markup 12
1.4.6 Example: Complex table with manual accessibility markup 14
A Aggregated RFC-2119 statements 16

Index 17

Copyright © OASIS Open . All Rights Reserved.

1 Accessibility

DITA has markup and features that enable producing output that is accessible by all audiences.

1.1 Handling accessibility in content and in processors

Accessibility requirements vary depending on how content is rendered. Making content accessible is work that involves both content authors and the processors that render DITA content.

One of the best known standards for accessible content is the set of Web Content Accessibility Guidelines (WCAG) from W3C. While content formats and content authors might have unique or additional accessibility needs, the rules outlined in the WCAG provide a reference point for considering how to create accessible content in DITA.

Comment by Stan Doherty on 20 June 2022

Recommend "the following rules" over "the rules"

Comment by Bob Johnson on 14 June 2022

No alternative to WCAG is available. It is the accessibility standard. I recommend revising this text to "The foundation for accessible content is the Web Content Accessibility Guidelines (WCAG).

[Regarding the inline link] Two points here.

First, accessibility guidance recommends against inline links. Inline links are distracting for users on screen readers, and are also problematic for users with cognitive disabilities such as ADD and executive function disorder. I recommend moving the link to the end of the topic.

Second, the currently effective versions of WCAG are 2.0 and 2.1. They are cumulative. WCAG 2.1 extends the 2.0 standard

WCAG 2.2 is scheduled for release in September 2022. It should be a recommended standard by the time this content is released. WCAG 2.2 is cumulative with 2.0 and 2.1. I recommend updating the text to this effect. My text above " the currently effective versions of WCAG are 2.0 and 2.1. They are cumulative. WCAG 2.1 extends the 2.0 standard." could be the basis of this text.

I recommend links to the 2.0, 2.1, and 2.2 standards formatted at the end of the topic using a linklist element. I recommend using "Web Content Accessibility Guidelines 2.0" and so forth as the link text. .

[Kris Eberlein, 17 June 2022]: Great info, Bob. We will incorporate the changes that you suggested. Can you point me to guidelines that recommend against inline links? I'd like to be able to share them with clients who are link crazy.

[Kris Eberlein, 20 June 2022]: I've made the wording changes that you suggested. I also added a relationship table link to WCAG.

Disposition: Completed

The guidelines fall into several categories:

Comment by Stan Doherty on 20 June 2022

Copyright © OASIS Open . All Rights Reserved.

Do we care about SVG edge cases such as:

```
<svgcontainer>
 <svgref href="foo.svg"/>
</svgcontainer>
```

Should we recommend that authors reference SVG images via the <img> element?

General content guidelines

Many accessibility guidelines and best practices apply to all content. Such guidelines are generally outside the scope of this specification.

For example, a guideline might recommend against multiple levels of nested unordered lists, because such lists are difficult to navigate with a screen reader. As a general content standard, DITA cannot prohibit such nesting. However, implementations can prevent such nesting through business processes or rule-based processing such as Schematron.

Another common accessibility recommendation is to avoid flashing or flickering video content. The DITA `<video>` element is a general mechanism for including video, and the content of that video is outside the scope of this specification.

Markup guidelines

Other accessibility guidelines require use of specific DITA markup. Such guidelines are addressed in this specification.

Comment by Stan Doherty on 20 June 2022

Recommend "require the use of" instead of "require use of".

For example, a requirement that images specify alternate text requires use of the `<alt>` element within an image. However, a guideline that the alternate text be *meaningful* is not something that can be enforced by DITA markup.

Comment by Stan Doherty on 20 June 2022

Recommend - "For example, requiring that images specify alternate text with the <alt> element within the <image> element."

Guidelines that require enablement by DITA processors

Some accessibility guidelines require processors to take advantage of specific DITA markup.

Authors can use specific markup to enable accessible output. For example, by specifying a header row in a table, an author can define a header for every cell in the table body. However, to make the relationship between the table cell and header cell specific in a rendered format like HTML, the processor must make those relationships explicit in the output

Comment by Bob Johnson on 14 June 2022

[Re the 2nd sentence, change "a header row" to] "header rows and header columns"

[Robert Anderson 20 June] I don't think that is necessary - one of the goals I had with the original "for example" was highlighting that by simply adding a table header row, you can create a header for every cell in the body, which is correct. You can also add a header column but that is not necessary as part of the example?

Copyright © OASIS Open . All Rights Reserved.

Processor requirements outside the scope of DITA markup

Processors have many other accessibility concerns that are outside the scope of this specification.

For example, WCAG has a requirement for contrast ratios when rendering text content.

> **Comment by Robert Johnson on 20220618**
> Guidance regarding contrast ratios applies to any substantive content, not just text. So, for example, if the content includes a spacer that serves only to support the look and feel of the content and has no meaningful value, contrast ratio is not important. But if the content includes a graphic of bar chart or pie chart that is meaningful to the content, the contrast ratio is important.
>
> [Robert Anderson 20 June] Will change "text content" to "substantive content"
>
> [Kris Eberlein, 20 June]: Done. Marking disposition as "Completed".
>
> **Disposition: Completed**

That requirement is unrelated to the source content. Such requirements apply to rendering mechanisms such as the CSS that is used to style DITA content in a browser.

As another example, a DITA processor might generate automated headings or include characters in output, such as:

- A section heading for an element specialized from `<section>`, such as "Requirements" for the `<prereq>` element in a task topic
- The greater-than character (>) that is typically used between phrases that are part of a menu navigation

It is up to the processor to use correct rendering for these cases, such as heading markup and accessible text alternatives for character displays such as the menu separator.

1.2 Accessible content

DITA provides elements and attributes that are designed to make content accessible.

Many common types of content are not accessible to all readers. For example, an image cannot be rendered by a screen reader, and a video cannot be rendered in many formats. DITA includes markup features that are designed to convey alternate versions of such content.

Alternate text for images

Alternate text is a textual description of an image. Systems often render the alternate text when the reader is using assistive technology or the image cannot be rendered.

The `<alt>` element is available inside of images as a way to specify alternate text.

Alternate text for areas of image maps

Within an image map, each defined area of the image can specify a cross reference. Whether the cross reference actually specifies a URI reference, the text within that cross reference is intended to function as alternate or hover text.

> **Comment by dstevens73 on 14 June 2022**
>
> Suggest: Whether or not the cross reference specifies a URI reference.

Long descriptions for media

A long description reference is a reference to a textual description of a graphic or object. This is typically used to provide an extended description when the graphic or object is too complicated to describe with alternate text.

Copyright © OASIS Open . All Rights Reserved.

Processors can handle the reference in the following ways:

- Render the graphic or object as a link
- Make the extended description available to accessibility tools such as a screen readers

> **Comment by Gershon Joseph**
> 19 June 2022: … such as screen readers (delete "a")
> **Disposition: Completed**

While DITA provides the markup to enable these accessibility features, it is up to DITA processors to render output that uses the markup properly. For example, when a processor generates HTML5, alternate text must be specified using the `@alt` attribute on the `<img>` element.

1.3 Accessible tables

The complexity of table rendering requires authors and processors to be aware of several table-specific elements and attributes if they want to ensure that tables are accessible.

DITA topics support two types of tables: complex table and simple table.

The `<table>` element uses the OASIS Exchange Table Model, a simplification of the CALS table model. The complex table provides a wide variety of controls over the display properties of the data and even the table structure itself.

The `<simpletable>` element is structurally less complex than the `<table>` element and so is an easier base for specialization. It reflects a content model that

> **Comment by Eliot Kimber**
> c/this close to the HTML table/is close to the HTML table model/
>
> [Kris Eberlein, 20 June 2022] Thanks for catching this. I've corrected this in the source branch.
>
> **Disposition: Completed**

this close to the HTML table. The `<simpletable>` element does not provide much control over formatting, although it permits titles and row and column spanning.

The following list provides information about table features that have an effect on table accessibility. Note that some features are applicable only to the complex tables that are produced by the `<table>` element.

> **Comment by Zoë Lawson on 20 June 2022**
>
> Are these really the only two tables? I'm thinking of choicetable, maybe something in reference topics (properties?), etc. They could be all tables that will be outside of base, so this may be out of scope. But should we mention that any tables specialized off of these two base tables will have similar accessiblity things?
>
> [Kris Eberlein, 21 June 2022]: You are correct; both `<choicetable>` and `<properties>` are specializations of `<table>` – and they are in technical content, not the base. We don't need to mention them here, because we mention early in the spec that when we refer to an element type, we refer also to specializations from that element type.

Captions
Both table models allow for a caption to be provided by using the `<title>` element.

Cell headers
(Complex table only) When entries within a table function as headers, but do not fall into the categories of column or row headers, the `@id` and `@headers` attributes on `<table>` cells can be

Copyright © OASIS Open . All Rights Reserved.

used. Specifying the `@id` attribute on the cell that functions as a header, and setting the `@headers` attribute to that ID value on table cell for which it acts as a header serves to relate table cells to headers.

> **Comment by Gershon Joseph**
> 19 June 2022: on "the" cell that functions... (add missing"the")

Column headers

> **Comment by Eliot Kimber**
> I think the column headers and row headers sections as written here confuse row headers (header cells that apply to a row) with column headers and header rows (cells and rows that provide headers for columns). I made specific comments below but reading the whole thing I think it just needs to be rewritten so that all references to @rowheader are in the row headers section and stuff relevant to column headers is in the column headers section.

Both table models provide support for column headers:

Complex table

(First column) The `@rowheader` attribute can be set to "firstcol" to indicate that the first column is a header. Alternatively, the `@scope` attribute in the first entry in the first row

> **Comment by Gershon Joseph**
> 19 June 2022: to (delete this "to")

can be set to "col" to indicate that it is a header for the entire column.

> **Comment by Eliot Kimber**
> @rowheader is for row headers, not column headers, so it's in the wrong place. From the current 2.0 draft:
>
> firstcol
>
> Indicates that entries in the first column of the table are row headers.

(Other columns) The `<colspec>` element can define which columns function as headers. For that case, set the `@rowheader` attribute to "headers" on the column or columns that function as headers. Alternatively, set the `@scope` attribute on each relevant entry in the first row to "col", indicating that the entry is a header for the entire column.

> **Comment by Eliot Kimber**
> Again, this reference to @rowheaders is for row headers, not column headers, so should be in the row headers section.

Simple table

The `@keycol` attribute can be set to the number of the column that functions as a header.

Row headers

Both table models provide support for a row header:

Complex table

The `<thead>` element can provide one or more header rows.

Simple table

The `<sthead>` element can

Copyright © OASIS Open . All Rights Reserved.

Comment by Gershon Joseph
19 June 2022: be used to (delete preceding "be used to" to be consistent with the previous thead description)

provide a single row header.

Summaries

(Complex table only) While the `@summary` attribute on tables is deprecated in HTML5, the `<desc>` element within a `<table>` can be used to store a summary. Since the content of the `<desc>` element is typically rendered as part of the content flow when used within `<table>`, processors might need special configuration to support this usage.

1.4 Examples of DITA markup for accessibility

This section contains examples of how DITA markup facilitates accessibility.

Comment by Kristen J Eberlein on 09 June 2022

A couple of comments:

- This section does not include examples of simple tables and accessibility. Do we need an example here? Can we simply point to the examples in the `<simpletable>` topic?
- The example topics about `<table>` were relocated from the element-reference `<table>` topic. They were reviewed as part of that content, but might need some slight tweak now that they are part of the "Accessibility" topic.

  [Dawn Stevens, 14 June 2022] My opinion on this is that we should have the same approach for table and simpletable. either point to examples in the other topics or have examples here.

  [Kris Eberlein, 15 June 2022] OK, sounds as if we should add a topic for "Example: Simple table" and cover accessibility markup that is applicable to simple table. We have two examples in the `<simpletable>` element-reference topic (simple table and simple table with row and column spanning), but neither example explicitly talks about accessibility.

  [Bob Johnson, 18 June 2022] I agree with taking a consistent approach to the two table options.

  [Robert Anderson, 20 June 2022] Agreed - need a volunteer to write up simpletable example topics for this section.

  [Kris Eberlein, 20 June 2022]: Marked as "Accepted" and added to Wiki page for volunteer tasks.

Disposition: Accepted

1.4.1 Example: Alternate text for an image

In this scenario, an image of a ticketing workflow also provides alternate text that describes the image.

The following code sample references an image `workflow.png`. Alternate text describes the workflow that is shown in the image.

Comment by Robert Johnson on 20220618
This example alt text is rather long. The first sentence is enough to communicate the meaning of the image. Text longer than one or two sentences should generally be moved to the running text. It is aceptable in that case to reference the running text in the alt text. If I saw this example, I would recommend moving everything after the first sentence to the running text, where it would likely be

Copyright © OASIS Open . All Rights Reserved.

helpful for all users. I would then add something like, "See the text after the image for details about the workflow."

[Kris Eberlein, 18 June 2022] I completely agree with about best practices for alternate text. But this is not an example of a full topic, just the markup for an `<image>` element. And given the context of an `<image>` element alone, without the context of the full topic, I think we need a full description of the image and what it shows.

Should we rework this topic to include markup for an entire topic? I don't think so, since the purpose of this topic is to talk about the image element and accessibility considerations, not demonstrate best practices for alternate text.

Now, if you can think of a way to handle both objectives concurrently ...

[Eliot Kimber, 20 June 2022]: I think I agree with Robert J.: The example text would be fine as something like "Workflow diagram showing a ticketing workflow. The workflow is described in the text."

That makes the point about alt text without being unusually long or verbose. Another principle of accessible content is don't put essential information only in a graphic, so emphasizing that the information is in the text is also reflective of good accessibility practice.

[Robert Anderson, 20 June 2022] I'm fine with removing most of this text as suggested - I like Eliot's suggested version, leaving this example as a simple example of the image / alt text markup.

[Kris Eberlein, 20 June 2022]: Done. Marking disposition as "Completed".

Disposition: Completed

```
<image href="workflow.png">
  <alt>The image is a workflow diagram that shows a ticketing workflow.
       The initial status is "Open". From there, the ticket can move to
       "Closed" or "In progress". A ticket that is "In progress" can move
       to "Done" or "Blocked". A "Blocked" ticket can move back to "In
       Progress" or directly to "Done". There are no steps beyond "Closed"
       or "Done".</alt>
</image>
```

1.4.2 Example: Alternate text for an image map

Comment by Kristen J Eberlein on 09 June 2022

I think we need a realistic example here in this section which focuses on accessibility. The `<imagemap>` topic is in the "Utilities domain," which has not yet been reviewed. The example there is old and not very good; it dates back to DITA 1.0.

Here are some possible ideas for what to showcase:

- A simple pie chart of application users by industry sectors
- A collage of photos of a company's executive team
- A diagram of a sampler quilt, with various blocks linked to pages that explain the specific blocks

[Bob Johnson 20220618] I had a quick look at the existing example, and I agree that it leaves much to be desired. The example alt text looks like placeholders someone expected to update later. I like the pie chart example. I can collaborate on developing the alt text for that example.

[Kris Eberlein, 20 June 2022]: Changing the disposition of this comment to "Accepted". We'll add developing content for this topic to our list of volunteer tasks. We'll go with the pie chart example unless determining the coordinates for pie slices is too difficult.

Copyright © OASIS Open . All Rights Reserved.

Disposition: Accepted

1.4.3 Example: Fallback information for multimedia content

In this scenario, fallback content is provided for systems that cannot display multimedia content.

The referenced video provides an image as fallback. If a system does not support video, it will display the image `video-not-available.png`, which specifies its own alternate text.

```
<video height="300px"
                loop="false"
                muted="false"
                poster="demo1-video-poster"
                width="400px">
  <desc>A video that illustrates how to conduct a system health scan.</desc>
  <fallback>
    <image href="video-not-available.png">
      <alt>This video cannot be displayed.</alt>
    </image>
  </fallback>
  <media-source href="video.mp4" format="video/mp4"/>
</video>
```

1.4.4 Example: Complex table with accessibility markup

In the following code sample, the table uses the `<thead>` element to identify header rows and the `@rowheader` attribute to identify a header column

Comment by Eliot Kimber
identify a row header column

Or maybe even more clearly, the `@rowheader` attribute to identify a column where each entry serves a header for the entry's row.

. These header relationships can be used to automatically create renderings of the table in other formats, such as HTML, that can be navigated using a screen reader or other assistive technology.

```
<table frame="all" rowheader="firstcol">
  <title>Sample of automated table accessibility</title>
  <desc>Names are listed in the column c1. Points are listed in both data columns, with
        expected points in column c2 and actual points in column c3.</desc>
  <tgroup cols="3">
    <colspec colname="c1"/>
    <colspec colname="c2"/>
    <colspec colname="c3"/>
    <thead>
      <row>
        <entry morerows="1">Name</entry>
        <entry namest="c2" nameend="c3">Points</entry>
      </row>
      <row>
        <entry>Expected</entry>
        <entry>Actual</entry>
      </row>
    </thead>
    <tbody>
      <row>
        <entry>Mark</entry>
        <entry>10,000</entry>
        <entry>11,123.45</entry>
      </row>
      <row>
        <entry>Peter</entry>
        <entry>9,000</entry>
        <entry>11,012.34</entry>
```


Copyright © OASIS Open . All Rights Reserved.

```
      </row>
      <row>
        <entry>Cindy</entry>
        <entry>10,000</entry>
        <entry>10,987.64</entry>
      </row>
    </tbody>
  </tgroup>
</table>
```

In this code sample, navigation information for assistive technology is derived from two sources:

- The `<thead>` element contains two rows, and indicates that each entry in those header rows is a header cell for that column. This means that each body cell can be associated with the header cell or cells above the column. For example, in the second body row, the entry "Peter" can be

  Comment by Eliot Kimber
  I think these can be a bit more direct: the entry Peter is associated with the header Name. That is what the markup mean and no processing that associated Peter with a different header would be correct.

  associated with the header "Name"; similarly, the entry "9,000" can be associated with the headers "Expected" and "Points".
- The `@rowheader` attribute that is specified on `<table>` indicates that the first column plays a role as a

  Comment by Eliot Kimber
  insert "row "

  header. This means that each body cell in columns two and three can be

  Comment by Eliot Kimber
  c/can be/is/

  associated with the header cell in column one. For example, in the second body row, the entry "9,000" can be associated with the header "Peter".

  Comment by Eliot Kimber
  I find this description of what a row header does confusing. I think it might be clearer to say "This means that the header cell in column is associated with the other body cells in the same row." The columns are really relevant to the rowheader semantic and the mention of the columns initially confused me.

As a result of these two sets of headers, a rendering of the table can associate the entry "9,000" with three headers: "Peter", "Expected", and "Points", thus making it fully navigable by a screen reader or other assistive technology.

The output might be rendered in the following way:

Name	Points	
	Expected	Actual
Mark	10,000	11,123.45
Peter	9,000	11,012.34
Cindy	10,000	10,987.64

Copyright © OASIS Open . All Rights Reserved.

Comment by Eliot Kimber
Might be useful to say how a typical screen reader would use these headers: when the user navigates to the cell containing "9,000" a screen reader can report the headers "Points", "Expected" and "Peter" as the headers for that cell.

[Robert Anderson, 20 June 2022] Isn't this described just above the image? Sort of like with our alt-text example, it seems like adding it here would just duplicate the same association described in the text

1.4.5 Example: Complex table with some manually-specified accessibility markup

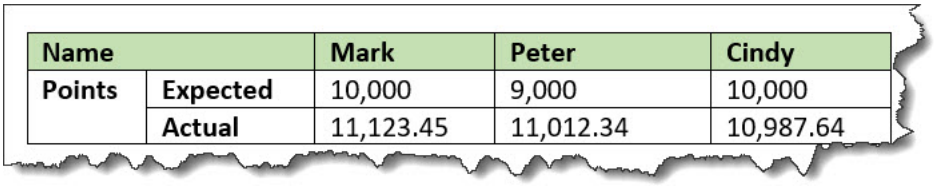
In some complex tables, the `<thead>` element and `@rowheader` attribute might not be enough to support all accessibility needs. Assume that the table

Comment by dstevens73 on 14 June 2022

Generally, I teach to avoid directional words; the reason for which is illustrated in this topic-by-topic review, where I could access this topic first and there is no "above". and even in print, the example might be on a previous page, not "above". Best to cross reference?

[Kris Eberlein, 14 June 2022]: This content was originally authored as part of a single element-reference topic, and the examples developed on each other. Thanks for catching this, which we certainly need to fix now that these examples are separate topics.

above is flipped so that the names are listed across the top row, instead of in the first column, as shown in the following screen capture:



Name		Mark	Peter	Cindy
Points	Expected	10,000	9,000	10,000
	Actual	11,123.45	11,012.34	10,987.64

Comment by Robert Johnson
I understand the motivation for this capability to mimic the support available in HTML, but in practice, I recommend against merging cells as in this example. A user on a screenreader builds a mental model of the table. Screenreaders do not have the ability to announce that a cell is merged. If such a capability existed, I suspect it would hinder understanding of the table and its contents. Merged cells are also problematic for users navigating by keyboard. In this example, keyboard navigation would default to the "Expected" row, which might not be the intention of the user. If that user also has a congitive disability such as ADD, the need to navigate to the correct row would likely be distracting and hinder access to the content. A user on a screenreader typically navigates using the keyboard, and the user agent behavior would hinder the development of the mental model. Merged cells within a table (in other owrds not in a first or last column or row) are even more problematic.

This is another case, similar the discussions in the "Handling accessibility" topic that practice is outside the scope of the spec. I would like to see a discussion somewhere that while this support exists, using it might not be good practice.

If I were reviewing a topic that included this example, I would recommend a single header column with the text "Expected points" and "Actual points" respectively. That would be easier to navigate, and more meaningful to a user on a screenreader.

Copyright © OASIS Open . All Rights Reserved.

[Kris Eberlein, 20 June 2022] Regarding your second paragraph. I hope that Tahzoo will join OASIS and so enable you to join the TC. For years, I've wanted the DITA TC to issue a committee note about "DITA and accessibility." In a committee note, we could provide information about best practices in a way that we cannot do in the specification.

[Robert Anderson, 20 June 2022] this is really intended to be a "if you have to make this kind of table, here is how it would work" example - definitely not mean to be an example of a common / best practice table. It's really intended to illustrate how to make use of accessibility attributes for the rare case that the more common approach does not fit. I think we should probably make that clearer at the start, so we don't appear to be endorsing this layout.

Here, the `@rowheader` attribute cannot be used, because it is only able to specify the first column as a header column. In this case, the `@scope` attribute can be used to indicate that entries in the first and second columns function as headers for the entire row (or row group, in the case of a cell that spans more than one row).

The following code sample demonstrates the use of the `@scope` attribute to facilitate navigation of these rows by a screen reader or other assistive technology. Note that the `<thead>` element is still used to imply a header relationship with the names at the top of each column.

```
<table frame="all">
 <title>Sample with two header columns</title>
 <tgroup cols="5">
  <colspec colname="c1"/>
  <colspec colname="c2"/>
  <colspec colname="c3"/>
  <colspec colname="c4"/>
  <colspec colname="c5"/>
  <thead>
   <row>
    <entry namest="c1" nameend="c2">Name</entry>
    <entry>Mark</entry>
    <entry>Peter</entry>
    <entry>Cindy</entry>
   </row>
  </thead>
  <tbody>
   <row>
    <entry morerows="1" scope="rowgroup"><b>Points</b></entry>
    <entry scope="row"><b>Expected</b></entry>
    <entry>10,000</entry>
    <entry>9,000</entry>
    <entry>10,000</entry>
   </row>
   <row>
    <entry scope="row"><b>Actual</b></entry>
    <entry>11,123.45</entry>
    <entry>11,012.34</entry>
    <entry>10,987.64</entry>
   </row>
  </tbody>
 </tgroup>
</table>
```

Comment by Robert Johnson
I think it would be helpful to also illustrate the result HTML. Not an image of the rendered table, but the result HTML code.

[Kris Eberlein, 18 June 2022]: We added the screen capture because spec reviewers were having difficulties visualizing how the table would be rendered.

Robert, Anderson, can you address the issues that would be involved if we included how the DITA markup might be rendered in HTML output?

Copyright © OASIS Open . All Rights Reserved.

[Gershon Joseph, 19 June 2022] I think both sample HTML code and an image of the resulting Web rendition would be most helpful to spec readers.

Eliot Kimber, 20 June 2022

1 to Gershon and Robert J. I have had many discussions along the lines of "HTML generated from DITA is generated with the necessary accessibility attributes by Open Toolkit." Showing an example of that might reduce the need for people to take my word for it.

I'll also mention that at GAO we had many complex tables that needed to use `@scope` in the way shown here but we found it impractical to author it given that the original tables were being converted automatically from Word tables without any regard for accessibility concerns. We identified the need for some kind of accessibility markup editor tool that could be used by an accessibility specialist to make adding the markup practical but that work was not funded during the time I was there.

[Robert Anderson, 20 June 2022] Given all of the feedback, I think it makes sense to add "common HTML mapping" (that is, not authoritative, but possible/likely) for each of the table examples.

1.4.6 Example: Complex table with manual accessibility markup

In extremely complex tables, such as those with a single header cell in the middle of the table, fine-grained accessibility controls are available to explicitly associate any content cell with any header cell. This might also be useful for cases where processors do not support the implied accessibility relationships described

Comment by dstevens73 on 14 June 2022

see comment in previous example about the word "above"

[Kris Eberlein, 14 June 2022] This content was originally authored as part of a single element-reference topic, and the examples developed on each other. Thanks for catching this, which we certainly need to fix now that these examples are separate topics.

above.

In the following sample, header cells are identified using the `@id` attribute, which is referenced using the `@headers` attribute on appropriate content cells. This makes all header relationships in the table explicit. Note that this sample ignores the `@scope` attribute, which could be used to exercise manual control without setting as many attribute values; it also ignores the fact that `<thead>` creates a header relationship even when the `@id` and `@headers` attributes are not used.

```
<table frame="all">
  <title>Sample with fully manual accessibility control</title>
<desc>Names are listed in the column c1. Points are listed in both data columns, with
expected points in column c2 and actual points in column c3.</desc>
  <tgroup cols="3">
    <colspec colname="c1"/>
    <colspec colname="c2"/>
    <colspec colname="c3"/>
    <thead>
      <row>
        <entry morerows="1"> </entry>
        <entry namest="c2" nameend="c3" id="pts">Points</entry>
      </row>
      <row>
        <entry id="exp" headers="pts">Expected</entry>
        <entry id="act" headers="pts">Actual</entry>
      </row>
    </thead>
    <tbody>
```


Copyright © OASIS Open . All Rights Reserved.

```
      <row>
        <entry id="name1">Mark</entry>
        <entry headers="name1 exp pts">10,000</entry>
        <entry headers="name1 act pts">11,123.45</entry>
      </row>
      <row>
        <entry id="name2">Peter</entry>
        <entry headers="name2 exp pts">9,000</entry>
        <entry headers="name2 act pts">11,012.34</entry>
      </row>
      <row>
        <entry id="name3">Cindy</entry>
        <entry headers="name3 exp pts">10,000</entry>
        <entry headers="name3 act pts">10,987.64</entry>
      </row>
     </tbody>
   </tgroup>
</table>
```

The output might be rendered in the following way:

Comment by Robert Johnson
An image of the rendered table in this case is not very informative. The rendering does not show the HTML code that the screenreader, for example, would be interpreting to announce to the user. The rendering would be the same regardless of the existence of the row and columns headers.

I think it would be more helpful to show the HTML code that would result from the markup above, which the screenreader would be interpreting for the user.

If someone were really ambitious, adding a recording of a screenreader announcing the table as it was navigated might be informative. The recording should include both audio and video, so the user can see the table being navigated and hear the screenreader interpreting the table.

[Kris Eberlein, 18 June 2022] We added the screen capture because spec reviewers were having difficulties visualizing how the table would be rendered.

Robert, Anderson, can you address the issues that would be involved if we included how the DITA markup might be rendered in HTML output?

My initial reaction is that a recording of a screen reader announcing the table as navigated would be awesome, but I might not be thinking of all relevant issues. Robert, your thoughts?

[Eliot Kimber, 20 June 2022]: Tony Morales taught me how to use the screen reader in macOS. I could take a stab at recording a screen reader navigation of the table. I'd be curious if our Zoom recording transcription maker would transcribe the screen reading.

	Points	
	Expected	Actual
Mark	10,000	11,123.45
Peter	9,000	11,012.34
Cindy	10,000	10,987.64

Copyright © OASIS Open . All Rights Reserved.

A Aggregated RFC-2119 statements

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

Copyright © OASIS Open . All Rights Reserved.

Index

A

accessibility
DITA markup 5
tables 6

Copyright © OASIS Open . All Rights Reserved.