

Publication: Review E: Table elements (00812677-DC_1)

Topic: entry (DA00508546)

Paragraph-level comments

If the @rotate attribute is not specified, the contents of the cell are not rotated. In situations where a stylesheet or other formatting mechanism specifies table cell orientation, the @rotate attribute can be ignored.

Annotation	Reviewer	Status	Type	Date	Topic version
It's not clear to me why @rotate is singled out here. If the stylesheet specifies separators or alignment, their corresponding attributes can also be ignored, correct?	dstevens	updated	comment	14/12/2021 15:01:14	
That's a good point. Like the @orient attribute, @rotate was introduced in DITA 1.2. The DITA 1.2 spec contained the following in the description of the @rotate attribute: "If this attribute is not specified, no rotation occurs . In situations where a stylesheet or other formatting mechanism specifies table cell orientation, the @rotate attribute can be ignored."	keberlein	updated	comment	15/12/2021 16:01:58	
It looks like this information was simply removed into a "Processing expectations" section for DITA 2.0.					
I think we need to consider whether the spec should make a broader statement about the fact that stylesheets might override stuff set in certain presentational attributes. And is that rendering or processing? Laughter.					
As with the table orient attribute ... another option is that we just delete the "rotate can be ignored if your stylesheets say otherwise" bit. It's a rendering instruction, and the spec cannot force a tool to respect it. We're basically saying "If you have a local style for tables / entries, you can use that local style for tables / entries."	randerson	updated	comment	16/12/2021 14:27:13	
Really good call! Removing this content.	keberlein	updated	comment	16/12/2021 17:14:43	
Marking this comment COMPLETED .					

If the @rotate attribute is not specified, the contents of the cell are not rotated. In situations where a stylesheet or other formatting mechanism specifies table cell orientation, the @rotate attribute can be ignored.

Annotation	Reviewer	Status	Type	Date	Topic version
The spec should state positive outcomes, not negative ones. I think the first sentence should be re-written as: If the @rotate attribute is specified, the contents of the cell are rotated.	gjoseph	updated	comment	14/12/2021 17:56:42	
@Gershon, I am going to close this comment because Dawn has a similiar one, with broader scope and implications for what the spec should state about stylesheets overriding what is set in some table attributes. It's difficult to track overlapping comments. Marking this comment CLOSED .	keberlein	updated	comment	15/12/2021 16:07:55	

The following attributes are available on this element: universal attributes (without the Metadata attribute group), @base , @rev , @align , @char , @charoff , @colsep , @rowsep , and @valign , and the attributes defined below.

Annotation	Reviewer	Status	Type	Date	Topic version
I suppose this would be a global comment, but it just occurred to me here, probably because of the duplicate @headers attribute -- perhaps the attributes should be alphabetized for easy reference?	dsteven	updated	comment	14/12/2021 15:03:08	
The attributes SHOULD be alphabeticized. I've corrected this list, and removed the duplicate entry for @headers. Marking this comment COMPLETED .	keberlein	updated	comment	14/12/2021 19:20:13	

Specifies one or more <entry> headers that apply to the current entry. The @headers attribute contains an unordered set of unique, space-separated tokens, each of which is an ID reference of an entry from the same table.

Annotation	Reviewer	Status	Type	Date	Topic version
I have honestly never heard of this attribute. I don't understand from this description what its purpose is.	dsteven	updated	comment	14/12/2021 15:04:26	
@headers was added to entry in DITA 1.3, as part of Scott Hudson's proposal to improve accessibility for some edge	keberlein	updated	comment	15/12/2021 16:31:22	

cases/complex tables. See <https://www.oasis-open.org/apps/org/workgroup/dita/download.php/51457/1-3proposal-62.html>

I also don't really understand the purpose of this attribute from its description ... I'll go look at the proposal and also the examples in this topic.

My expectation is that this would be *extremely* rarely used - I have never worked with a table that requires its use. The examples in the table topic right now give an extended example of a table where it is used. But in most cases, you are much, much better off relying on a few attributes and a processor to handle accessibility for you.

This is really intended for really edge-case tables, such as those that might have a header cell in the middle of the table, or for cases where an application is unable to infer accessibility from the markup. In that case, you can set @headers on every table entry, and explicitly link it to each relevant header cell in the table. I personally consider this 1) very difficult to do by hand, and 2) very error prone when adding rows/columns in the future, so would only use this in those edge-case scenarios.

randerson updated comment 15/12/2021 23:21:45

So, what do we what to do with the description of the @headers attribute? @Dawn, @Robert -- please make some suggestions.

keberlein updated comment 16/12/2021 13:52:24

I don't think there is much we can do here -- I don't think we can or should put enough info in the attribute description that it would clarify every usage. We could add information to the "Usage" section that explains it in detail, or even better, refer to an arch spec section on accessibility that lays it out in detail. Otherwise we're veering back into the problematic DITA 1.2 / DITA 1.3 areas where you have too much conceptual / processing detail hidden in the definition of the attribute.

randerson updated comment 16/12/2021 14:31:09

I think at most we should just tweak the wording, like, rather than "Specifies one or more entry headers that apply to the current entry", perhaps "Specifies which entries in the current table provide headers for this cell".

keberlein updated comment 20/12/2021 11:44:56

Implemented Robert's secon suggestion for a specific wording tweak. Will open another

comment to track the possibility of adding an architectural topic about accessibility.

Marking this comment **COMPLETED**.

Specifies one or more <entry> headers that apply to the current entry. The @headers attribute contains an unordered set of unique, space-separated tokens, each of which is an ID reference of an entry from the same table.

Annotation	Reviewer	Status	Type	Date	Topic version
This attribute is showing up twice, but my comments just attach to this one.	dstevens	updated	comment	14/12/2021 15:06:43	
No changes required, making this comment CLOSED .	keberlein	updated	comment	14/12/2021 19:25:22	
I suspect that when I tried to cut and paste as part of alphabetizing, I copy/pasted instead...	randerson	updated	comment	16/12/2021 14:27:47	

Specifies the column name in which an entry is found. The value is a reference to the @colname attribute on the <colspec> element.

Annotation	Reviewer	Status	Type	Date	Topic version
So, why do I use this? Is it inherited? What happens if it doesn't match the <colspec>?	zlawson	updated	comment	20/12/2021 20:34:58	
I don't know. @Kris is it worth trying to remove this attribute? The only time I've ever seen it used, someone had set it, and then the table broke when it was copy/pasted or a column was added.	randerson	updated	comment	21/12/2021 03:07:07	
@Robert, I have never used it or seen it used. Do you understand what the use cases for this attribute are? Seems like we need to understand that before considering removing the attribute.	keberlein	updated	comment	21/12/2021 14:06:00	
I have definitely seen it added by editors. It is a way to be explicit about what column your entry is in ... I think maybe, really stretching here, in text view it might make it easier to picture your table because it's explicit.	randerson	updated	comment	22/12/2021 18:53:19	
After asking if it should be removed, I'm actually rethinking that -- I do remember seeing it added by at					

least one editor, which makes me think that editors might add it by default with their reused CALS support, and removing the attribute might break that editor support.

We definitely do **NOT** want to do things that would make existing tables in DITA source invalid, without having a truly compelling use case -- and we do not have one here.

keberlein updated comment 27/12/2021 14:35:06

No change required, so marking this comment **CLOSED**.

Specifies the last logical column that is included in a horizontal span. The value is a reference to the @colname attribute on the <colspec> element.

Annotation	Reviewer	Status	Type	Date	Topic version
<p>I realize I'm asking lots of questions that might be all answered by "go read the other table spec", and I generally avoid spanning tables because that just makes everything too complicated, so I don't know how a lot of these attribute really work. So what might be second nature is alien to me.</p>					
<p>If you specify @namest, do you have to specify @nameend?</p>	zlawson	updated	comment	20/12/2021 20:42:08	
<p>Do we need to mention that you use @namest and/or @morerows in the 'first' entry? I guess that's implicit, but sometimes when looking at tables, I might be working backwards. e.g. if I'm in Excel, I will grab cell 3C and then 3B, and I know "merge" will work. That may not happen with the equivalent action with DITA...</p>					
<p>Zoe, it's alien to me also. I usually use simple tables, and if I am spanning ANYTHING, I'm using helper features in an editor.</p>					
<p>We cover this in the spec, because the audience for the spec is implementors, who might be just crazy enough to build a new table rendering engine. I'm going to have to punt most of your questions over to Robert.</p>	keberlein	updated	comment	20/12/2021 21:40:06	
<p>I'm not sure that we need to get this explicit. As Kris noted, the audience here is implementors working with table models, and any implementor building on this likely already has a CALS-like module of some sort.</p>	randerson	updated	comment	21/12/2021 03:10:58	
<p>The namest/nameend attributes are only meaningful in pairs. It does not make any sense to say where a column</p>					

ends without saying where it starts, or the reverse. I think the most I'd want to say is something like "works together with [the other]" -- this is not worth a normative rule like "MUST be specified with..."

For morerows - there is no end, it's just a number of rows spanned, so the issue does not come up.

No changes to the spec required. Marking this comment **CLOSED**.

keberlein updated comment 21/12/2021 14:09:36

Specifies that the current entry is a header for other table entries. The following values are valid:

row

The current entry is a header for all cells in the row.

col

The current entry is a header for all cells in the column.

rowgroup

The current entry is a header for all cells in the rows that are spanned by this entry.

colgroup

The current entry is a header for all cells in the columns that are spanned by this entry.

-dita-use-conref-target

See for more information.

Annotation	Reviewer	Status	Type	Date	Topic version
Alphabeticize the values for @scope.					

Done.	keberlein	updated comment		20/12/2021 13:30:44	
Marking this comment COMPLETED .					

Specifies that the current entry is a header for other table entries. The following values are valid:

row

The current entry is a header for all cells in the row.

col

The current entry is a header for all cells in the column.

rowgroup

The current entry is a header for all cells in the rows that are spanned by this entry.

colgroup

The current entry is a header for all cells in the columns that are spanned by this entry.

-dita-use-conref-target

See for more information.

Annotation	Reviewer	Status	Type	Date	Topic version
Do we need to talk about how these might interact with	zlawson	updated comment		20/12/2021 20:37:45	

entries in <thead>? Do I have to set these if I'm in a <thead>

If I am using rowgroup or colgroup, does that mean I have to set @namest and @nameend?

My take is that a thead implies "colgroup" for every entry in the heading. The updated examples are intended to illustrate that these values are not needed when thead is used, but we should explicitly state that. I'd say that thead implies a processing default of "colgroup" but we need to run that by the TC. Adding a draft comment to clarify this.

randerson updated comment 21/12/2021 03:05:50

Making this comment **COMPLETED**, since Robert added the draft comment.

keberlein updated comment 27/12/2021 14:59:56

Topic: table (DA00509284)

Topic-level comments

Annotation	Reviewer	Status	Type	Date	Topic version
I hate to suggest it, because it makes my head hurt, but do we need an example showing how all the complex table display options might work, specifically @colsep and @rowsep? Or is there some architectural topic about those display attributes? Are there normative statements about them? Do we tell folks that there's no guarantee that your display attributes will be followed in the output?	zlawson	updated	comment	20/12/2021 21:41:08	
I just read about @char and @charoff...and I need help with those :-). Although that could be as simple as "this is how you can align on the decimal point for a column of numbers"					
I really don't want an example of every one. Reasoning: * We are trying to limit examples to the most common cases of an element. I don't want an exception for table that requires an entire example for every display related attribute. * Most of these, at this point, are very rarely used, if at all. (I've never seen char/charoff used.) * I think processing support for these already varies considerably, based on how well the apps support CALS, so I'm hesitant to add examples that say how things work	randerson	updated	comment	21/12/2021 03:31:55	

... when it's possible applications have never handled them and never will.

* We should not have normative statements about these; it's one of those "This is how it works" things, so it's redundant to say "This means applications do X so applications MUST do X"

To be clear ... I'm not saying "nobody supports this" (which would immediately make me wonder if we should include them). I'm just saying, I don't *know* if anyone uses them, but I know applications that don't support them and don't have issues. So I'm leery of removing them, but also leery of making the support more obviously demanding.

keberlein updated comment 21/12/2021
12:57:06

I'm with Robert on this one. Several points here:

We do not want the examples in the element-reference topics to demonstrate EVERY use of attributes or processing. We want to show at least one clear, common usage of the element (or cross reference to such an example). We only include multiple examples if we need to demonstrate a couple of markup scenarios. I think simpletable is a good example of what is appropriate to cover in the "Examples" section; we have three examples, each of which shows something important:

- The most common use of simple table: Table with a row header
- A more elaborate use of simple table: Table with a title and some spans
- A simple table that uses @keycol: an attribute unique to simpletable and DITA

Do we need to be more explicit about the display attributes and how they work? I don't think so. I think we need to provide a clear explanation of the semantic of the attributes -- although I feel a little funny about talking about the "semantics" of a display attribute. A couple of reasons behind my thinking here:

- Most often, company stylesheets handle how tables are rendered. In some cases, this might include overriding **almost all** display attributes so that the company can ensure a standard rendering for tables. (Information developers can do some crazy, scary things with this markup.)
- Processors (and remember this category includes authoring tools) have varying levels of support for how they handle the display attributes. I don't want to go anywhere near laying out expectations

for what processors should do with display attributes.

- Most of the table attributes come from CALS tables and later the OASIS spec that Norm Walsh authored. Any implementor worth their salt should look there if they are implementing table support.

Do we want to make normative statements about display attributes? Most definitely NOT.

Should we warn readers that display attributes might be overridden by processors? Maybe. If we do this, it should be a simple, single crisp statement.
@Robert, you thoughts on this?

Robert and I discussed this at a spec editors' call last week. After mulling this over, we decided NOT to include such a statement. We just could not figure out where to locate it! If table was not grouped together with simple tables, then we could put it in the container topic, but the spec currently is not grouped that way.

keberlein updated comment 27/12/2021 14:40:15

Marking this comment **CLOSED**.

Paragraph-level comments

An optional `<title>` inside the `<table>` element provides a caption to describe the table. In addition, the optional `<desc>` element enables a table description that is parallel with a figure description.

Annotation	Reviewer	Status	Type	Date	Topic version
Regarding this sentence: In addition, the optional <code><desc></code> element enables a table description that is parallel with a figure description.					
I don't think the comparison with a figure description offers any value to the reader. The purpose of this optional element is to hold a description of the table that contains additional information not found in the table title. Renderers may use this description to describe the table upon user request or when the table cannot be rendered. (Consider assistive readers here, the audio readout often reads the description instead of the table itself, or before the table itself.)	gjoseph	updated comment		15/12/2021 08:55:47	
I agree that the comparison with figure is not useful. However, do remember that desc is not the equivalent of alt; the contents of desc within table ARE intended to be	keberlein	updated comment		15/12/2021 15:17:47	

rendered in the content flow. In fact, we have a normative statement to this effects in the "Rendering expectations" section of the desc topic.

Changed the second sentence to simply read "In addition, the optional desc element enables a table description."

I wonder whether we need to mention in this (table) topic that when used, the content of the optional desc and title elements are typically rendered as part of the content flow.

@Robert and @Gershon?

I'm not sure. Strictly speaking, that's about rendering <desc>. But yeah I assume it will be missed if we don't mention it here.

Implementations I've worked with always display the desc by default as part of the content flow, visible in the browser / without a screen reader. That said, I've also helped with customizations that explicitly used this as the screen reader summary of the table. I think either approach is valid (which is why the spec makes this a SHOULD and not a MUST), but must be used consistently or your content breaks -- which is why there is a normative rule pushing one consistent default.

randerson updated comment 15/12/2021 23:15:13

@Robert, so what do we want to do here?

- Add a "Rendering expectations" section, and mention that the contents of the optional desc and title are typically rendered as part of the content flow?
- Do we want to make any mention of implementations using the content of desc as a screen reader summary of the table? Or is that more appropriate for mention in the desc topic?
- Bring this item to the TC for discussion?

keberlein updated comment 16/12/2021 13:57:49

I would bring it to the TC because this confuses so many people, but with a suggestion that we resolve it by adding a rendering section that reminds that the description is typically part of the document flow. I'd probably favor noting that implementations have been known to use this as table-summary metadata, but I'm not sure how to say that with spec language and not totally upend the "SHOULD be rendered" normative rule from the desc topic.

randerson updated comment 16/12/2021 14:40:31

Marking this comment as **REFERRED**. I've

keberlein updated comment 20/12/2021

In DITA tables, in place of the @expanse attribute used by other DITA elements, the @pgwide attribute is used in order to conform to the OASIS Exchange Table Model. The @pgwide attribute has a similar semantic (1=page width; 0=resize to galley or column).

Annotation	Reviewer	Status	Type	Date	Topic version
I think the last sentence needs to be revised. When I read "the @pgwide attribute has a similar semantic", I thought that meant that @expanse also used 0 and 1. Or that it matched whatever @pgwide is in the OASIS Exchange Table Model.	zlawson	updated	comment	20/12/2021 21:19:11	
I don't think we need to compare attributes. I'd rather just say what the two values are and leave it at that.					
I removed the 2nd sentence of the paragraph. @pgwide is describe clearly in the "Attributes" section.	keberlein	updated	comment	20/12/2021 21:46:51	
Marking this comment COMPLETED.					

Note:

The @scale attribute represents a stylistic markup property that is currently maintained in tables for legacy purposes. External stylesheets should enable less dependency on this attribute. Use the @scale attribute judiciously.

Annotation	Reviewer	Status	Type	Date	Topic version
Should we REMOVE the @scale attribute from the DITA 2.0 table model? Based on this note, I think we should.	gjoseph	updated	comment	15/12/2021 09:00:27	
I don't think so, just because I've worked with content that uses it to fit tables on a PDF page, and dropping it with a comment that says "just use CSS" leaves those people with nothing. The only way to scale a table in PDF at that point is to write custom rendering for this fairly basic feature.					
I think the note comes from expectaions around HTML -- and it was added in DITA 1.0, unchanged since. The thought was "In HTML this is better controlled by CSS so we should do the same". But PDF lives on and I'd rather not pull out a useful feature with no clear replacement. Better to remove the note.	randerson	updated	comment	15/12/2021 23:17:04	
Removed the note. Marking this comment	keberlein	updated	comment	16/12/2021 13:48:49	

COMPLETED.

In situations where a stylesheet or other formatting mechanism specifies table orientation based on criteria other than the value of `@orient`, or for non-paginated outputs, the `@orient` attribute can be ignored.

Annotation	Reviewer	Status	Type	Date	Topic version
<p>As with rotate, I don't think i understand why this specific attribute is called out as being able to be ignored?</p>	dstevens	updated	comment	14/12/2021 15:26:05	
<p>I had to do some research. We introduced <code>@orient</code> in DITA 1.2, I think. (This attribute did not appear in the DITA 1.1 spec.)</p> <p>In the 1.2 spec, the description of the <code>@orient</code> attribute contained the following sentence: " In situations where a stylesheet or other formatting mechanism specifies table orientation based on other criteria, or for non-paginated outputs, the <code>@orient</code> attribute can be ignored."</p> <p>So, it looks as if we simply moved the 1.2 content (slightly tweaked for precision) into a "Processing expectations" section.</p>					
<p>My questions:</p> <ol style="list-style-type: none">1. Does this information belong with the description of the <code>@orient</code> attribute or in a section?2. If it belongs in a section, is the information about rendering or processing?					
<p>I think the intent of the content was simply to communicate that "The <code>@orient</code> attribute is intended for print-based display. Stylesheets might implement rules that specify table orientation, and if they do, the <code>@orient</code> attribute is ignored."</p>					
<p>I moved it to processing in my last edit before this review, so I can be blamed for this...</p>	randerson	updated	comment	15/12/2021 23:07:50	
<p>My reasoning was that this is an explicit instruction to processors on when they can ignore an attribute value that otherwise has a specific meaning. As such, it should not be hidden in the definition of the attribute -- we've been working to take <i>*all*</i> processing details out of the langref attribute definitions, so that they are highlighted like this.</p>					
<p>I do think this is a closer call than many though, so if someone wants to argue that it should be restored to the</p>					

attribute or moved to a "Rendering" section, please do so...

@Robert, I think we should consider handling this as part of a larger statement that stylesheets can/might override any number of attributes set on tables. See Dawn's comment about @rotate on the entry element. What do you think?

keberlein updated comment 16/12/2021 14:01:06

Similar to the comment on @rotate on entry, we're simply removing this content.

keberlein updated comment 16/12/2021 17:24:45

Marking this comment **COMPLETED**.

The following attributes are available on this element: universal attributes , @colsep , @frame , @rowsep , @rowheader , @scale , and the attributes defined below.

Annotation	Reviewer	Status	Type	Date	Topic version
Should complex table attributes be in this list?	zlawson	updated	comment	20/12/2021 21:25:39	
No ... complex table atts is a group of "All the ugly CALS related attributes", but they're available in an odd mix of places. The table element uses colsep, rowsep, and rowheader (but not the others, like valign).	randerson	updated	comment	21/12/2021 03:36:18	
Markjng this comment as COMPLETED (since Robert added the draft comment.)	keberlein	updated	comment	27/12/2021 14:55:37	

Specifies the orientation of the table in page-based outputs. This attribute is primarily useful for print-oriented display. The following values are valid:

port

The same orientation as the text flow.

land

90 degrees counterclockwise from the text flow.

-dita-use-conref-target

See for more information.

Annotation	Reviewer	Status	Type	Date	Topic version
The descriptions of port and land are rather odd. Initially I thought of port as in port versus starboard and land versus sea! Also, it's not dependant on the flow of the text. It's based on the orientation of the actual PAGE.	gjoseph	updated	comment	15/12/2021 09:16:08	

So, I suggest the following text instead:

port

Portrait page orientation. The page is oriented with its long side vertical and its short side horizontal.

land

Landscape page orientation. The page is oriented with its long side horizontal and its short side vertical.

Good suggestion; done.

keberlein updated comment 15/12/2021
16:49:53

Marked as **COMPLETED**.

Example

Annotation	Reviewer	Status	Type	Date	Topic version
Example s	gjoseph	updated	change	15/12/2021 09:17:51	
Done. Marked as COMPLETED .	keberlein	updated	comment	15/12/2021 16:50:37	

This section contains example of how the <table> element can be used.

Annotation	Reviewer	Status	Type	Date	Topic version
This section contains example s of how the element can be used.	dstevens	updated	change	14/12/2021 15:28:33	
Corrected. Marking this comment COMPLETED .	keberlein	updated	comment	14/12/2021 19:40:08	

The following code sample shows a table that is used to provide reference information about animals and gestation:

Annotation	Reviewer	Status	Type	Date	Topic version
It's not technically wrong, but in building this table, I would title the second column Gestation (in months), and then leave the word months out of every entry.	dstevens	updated	comment	14/12/2021 15:30:08	

```

<table>
  <tgroup cols="2">
    <colspec colwidth="121*" />
    <colspec colwidth="76*" />
    <thead>
      <row>
        <entry colname="COLSPEC0" valign="top">Animal</entry>
        <entry colname="COLSPEC1" valign="top">Gestation</entry>
      </row>
    </thead>
    <tbody>
      <row>
        <entry>Elephant (African and Asian)</entry>
        <entry>19-22 months</entry>
      </row>
      <row>
        <entry>Giraffe</entry>
        <entry>15 months</entry>
      </row>
      <row>
        <entry>Rhinoceros</entry>
        <entry>14-16 months</entry>
      </row>
      <row>
        <entry>Hippopotamus</entry>
        <entry>7 1/2 months</entry>
      </row>
    </tbody>
  </tgroup>
</table>

```

Annotation	Reviewer	Status	Type	Date	Topic version
Why are you putting the @colname in the <entry>, not the <colspec>? Is there an advantage?	zlawson	updated comment		20/12/2021 21:27:55	
Also, right now the attributes section doesn't list @valign. I think that gets solved if you add a link to the complex table attributes.					
This example is of the table element, which does not allow valign.					
The use of colname here is an error - removing. The current example is invalid (you can't put colname on an entry unless that same name is on a colspec.)	randerson	updated comment		21/12/2021 03:37:47	
Robert has updated the code sample in the topic. Marking this comment COMPLETED .	keberlein	updated comment		21/12/2021 15:00:40	

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

Annotation	Reviewer	Status	Type	Date	Topic version
There isn't a code example here that shows the <code>@rowheader</code> .	dstevens	updated	comment	14/12/2021 15:31:47	
Apologies, @Dawn; I screwed things up last night. I'm committing an updated topic to GitHub, and will be sending out a revised PDF.	keberlein	updated	comment	14/12/2021 20:30:45	
Marking this comment COMPLETED .					
In addition, I would expect the output to show the header column also formatted in bold, or some background shading, to illustrate the use of the <code>@rowheader</code> .	gjoseph	updated	comment	15/12/2021 09:30:42	
Added Gershon's comment to a draft comment in the topic, for consideration when screen captures are added.	keberlein	updated	comment	20/12/2021 12:31:46	
Marking this comment COMPLETED .					

In this case 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 `@scope` 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.

Annotation	Reviewer	Status	Type	Date	Topic version
In this case the <code>@rowheader</code> attribute cannot be used, because it is only able to specify the first column as a header column. In this case, the <code>@scope</code> 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 <code>@scope</code> to facilitate navigation of these rows by a screen reader or other assistive technology. ⚠ Note that the element is still used to imply a header relationship with the names at the top of each column.	gjoseph	updated	change	15/12/2021 09:42:34	
Done. Marked as COMPLETED .	keberlein	updated	comment	15/12/2021 16:52:32	

In this case 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 `@scope` 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.

Annotation	Reviewer	Status	Type	Date	Topic version
Can we get this information about why you use <code>@rowheader</code> vs <code>@scope</code> into the definitions of the attributes?	zlawson	updated	comment	20/12/2021 21:31:49	
With <code>@rowheader</code> , is that limited to one column, or specifically the first column?					
rowheader when used on <code><table></code> is limited to just the first column.					
when used on colspec it can work on any column. But ... that's difficult to use, particularly in a graphical table editor.	randerson	updated	comment	21/12/2021 03:34:44	
I think which to use is probably a best practice, but I'm not certain. <code>@Kris</code> , this makes me wonder if we should be scrapping <code>@rowheader</code> ...					
We discussed removing <code>@rowheader</code> on the spec editors' call last week, and agreed to leave the table model as-is.	keberlein	updated	comment	27/12/2021 14:44:41	
Marking this comment CLOSED .					

```
<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>

```

Annotation	Reviewer	Status	Type	Date	Topic version
<p>This example makes no sense to me to get the output shown. Name is written to span two columns, not two rows like the output. Points is written to span two rows, not columns like the output. But I've never used the scope attribute. So I thought I might be missing something, so I copied the code and processed it with the default PDF output and did not get the shown output.</p>	dstevens	updated	comment	14/12/2021 15:46:25	
<p>If it's right that this shown output is what is expected, why would anyone write it that way anyway?</p>					
<p>You're right, this is an error in the example - I think the code block and the actual sample table flipped col span / row span attributes. Regardless the "It might look like this" table doesn't use the same markup as the code sample.</p>	randerson	updated	comment	15/12/2021 23:11:27	
<p>Done; marked as COMPLETED.</p>	keberlein	updated	comment	16/12/2021 14:06:08	

Topic: stentry (DA00508735)

Topic-level comments

Annotation	Reviewer	Status	Type	Date	Topic version
<p>I reiterate my comments on &lt;entry> for @colspan, @rowspan, and @scope.</p>	zlawson	updated	comment	20/12/2021 21:08:47	

I also echo Dawn's comments on @headers from <entry>, but will wait for the discussion that's already started. Potentially link to an architectural topic and/or a really good example?

I'm not sure what to do about colspan/rowspan -- the attributes on table are namest/nameend for column spanning, which are gross and come from the old model, while simpletable is just colspan -- the number of columns to span.

Row spanning also differs a little, but in this case both versions are just a number -- CALS / complex table is morerows=2 (this spans 2 more rows plus the current), while simpletable uses the HTML approach of rowspan (total number of rows spanned by this entry, so rowspan="2" means this spans a total of two rows).

randerson updated comment 21/12/2021 03:24:51

Marking **Closed** based on the previous comment.

Note that in working on the entry topic, I've added a draft comment to the @scope attribute definition to remind us about addressing your comments on that attribute. So this one is sort of closed for colspan/rowspan, and accepted for scope.

randerson updated comment 22/12/2021 19:12:53

Paragraph-level comments

Specifies the number of rows that a cells is to span inside a simple table.

Annotation	Reviewer	Status	Type	Date	Topic version
Specifies the number of rows that a cell s is to span inside a simple table.	dstevens	updated	change	14/12/2021 15:20:39	
Good catch; corrected.	keberlein	updated	comment	14/12/2021 19:37:08	
Marking this comment COMPLETED .					

Topic: colspec (DA00508850)

Paragraph-level comments

A column specifications provides information about a single column in a table that is based on the OASIS Exchange Table Model. The information might include a column name and number, cell content alignment, or a column width.

Annotation	Reviewer	Status	Type	Date	Topic version
A column specification (singular, not plural) Do you need the "a" before column width?	zlawson	updated	comment	20/12/2021 20:26:14	
Done. Thanks for catching this. Marking this comment COMPLETED .	keberlein	updated	comment	20/12/2021 21:36:53	

Specifies the number of a column in the table, counting from the first logical column to the last column.

Annotation	Reviewer	Status	Type	Date	Topic version
the column, not 'a column', since the attribute is for a specific column? I'm not entirely sure what "counting from the first logical column to the last column" is specifically trying to say. Do I start with 0 or 1? Do I have to have the columns numbered in order? Or could I use 1, 3, 5? Why do I want to use this? (huh, on reading the spec, it doesn't really do anything super useful. " The value of colnum is not useful to identify a column in an entry , so serves no functional purpose other than a consistency check on the order of the colspec s.") Or should we just link to the other spec?	zlawson	updated	comment	20/12/2021 20:31:12	
I don't want to link to the OASIS exchange spec, because - as you noted - this attribute does not do much, and going off to read that spec will leave you thinking "I read this whole other spec and it still doesn't do much". Agreed on "a column" => "the column" First logical column is, I believe, a standard-ish way of avoiding saying "left" vs "right" -- column 1 is the first/left column in an English topic and the first/right column in an Arabic one. I had never considered the 1/3/5 approach. So I would say "Don't do this". @Kris is this something we need to go into detail on?	randerson	updated	comment	21/12/2021 02:57:46	

Suggestion to handle the 0 or 1 question: change the wording to

Specifies the number of the column in the table, where 1 equals the first logical column.

randerson updated comment 22/12/2021 19:18:35

I don't think we need to address the 1/3/5 question just because that issue has never come up in 16+ years and it seems unlikely to with future implementations.

Made the change. Marking this comment **CLOSED**.

keberlein updated comment 27/12/2021 14:51:08

Specifies the column width.

Annotation	Reviewer	Status	Type	Date	Topic version
Should we add some information about valid values for this attribute? Also refer to the units that can be specified, relative versus absolute width. Otherwise perhaps just link to the OASIS Exchange table model topic that describes how to use this attribute in that spec.	gjoseph	updated	comment	14/12/2021 17:53:01	
Here is the URL: https://www.oasis-open.org/specs/tm9901.html#AEN530 The document is " XML Exchange Table Model Document Type Definition," Organization for the Advancement of Structured Information Standards (OASIS) Technical Memorandum TR 9901:1999 The relevant text: "Either proportional measure of the form number*, e.g., "5*" for 5 times the proportion, or "*" (which is equivalent to "1*"); fixed measure, e.g., 2pt for 2 point, 3pi for 3 pica. (Mixed measure, e.g., 2*+3pt, while allowed in the full CALS table model, is not supported in this Exchange model.) Coefficients are positive integers or fixed point numbers; for fixed point numbers, a leading (possibly 0) integer part is required, and implementations should support at least 2 decimal places. A value of "" [the null string] is treated as a proportional measure of "1*". The fixed unit values are case insensitive. The standard list of allowed unit values is "pt" (points), "cm" (centimeters), "mm" (millimeters), "pi" (picas), and "in" (inches). The default fixed unit should be interpreted as "pt" if neither a proportion nor a fixed unit is specified.	keberlein	updated	comment	15/12/2021 19:07:10	

Declared value

CDATA

Default

IMPLIED (means assume a proportional measure of "1*")

@Robert, Gershon -- What do you want to do with this?

It probably makes sense to be explicit, though I'll admit I feel ambivalent about this. We've never been explicit, and it then raises the same question we had about the width attribute - do we have to explain each abbreviation, do we have an exhaustive list that rules out other measurements, etc.

randerson updated comment 16/12/2021
14:23:58

I added more complete information, including a cross reference to the authoritative document. I also added a draft comment stating that I was ambivalent about providing such exhaustive info, and that perhaps it should be scaled back.

keberlein updated comment 20/12/2021
12:51:05

Marking this comment **ACCEPTED**.

Topic: tbody (DA00509457)

Paragraph-level comments

A table body is a group of rows in a table that is based on the OASIS Exchange Table Model.

Annotation	Reviewer	Status	Type	Date	Topic version
Can we use "collection" instead of group of rows? Just to make sure I keep it separate from tgroup?					
Or, is it better to say that <tbody> is the wrapper for collection of rows that make up a table? I think you can only have one <tbody> per <tgroup> (I tried to figure it out from the other spec, but got lost)	zlawson	updated	comment	20/12/2021 21:38:10	
Do we want to differentiate from <thead> in some way?					
Sure, we can call it a "collection".	keberlein	updated	comment	27/12/2021 14:27:05	

Made the change, and am marking this comment as **COMPLETED**.

Topic: tgroup (DA00509017)

Paragraph-level comments

A table group is a grouping of a table header and table body, based on the OASIS Exchange Table Model.

Annotation	Reviewer	Status	Type	Date	Topic version
Wrapper instead of grouping? Just to avoid reiteration?					
Can you have multiple tgroups in a DITA table? Would it be useful?	zlawson	updated	comment	20/12/2021 21:45:46	
Sure, we can call it "a wrapper that contains ..."					
Yes, you can have multiple <tgroup> elements, but just DON'T do it.	keberlein	updated	comment	22/12/2021 18:53:37	
Made the change, and marking this comment as COMPLETED .					

Topic: Table elements (DA00513551)

Topic-level comments

Annotation	Reviewer	Status	Type	Date	Topic version
Add a architectural topic about accessibility. Such a topic could include:					
<ul style="list-style-type: none">• Info about @headers• The complex examples about accessibility currently in the table topic	keberlein	updated	comment	20/12/2021 11:47:32	
Marking this comment as ACCEPTED .					
Do we want to provide a link to the OASIS Exchange Table Model? https://www.oasis-open.org/specs/tm9901.html	zlawson	updated	comment	20/12/2021 20:23:56	
I don't think it's necessary here, just because following a link to it should not give you any information that is	randerson	updated	comment	21/12/2021 02:54:29	

necessary for understanding tables in the DITA context. We do list it in the non-normative references in the spec: <http://docs.oasis-open.org/dita/dita/v1.3/errata02/os/complete/part3-all-inclusive/introduction/non-normative-references.html>

Following up with a clearer explanation of my reasoning:

Our stuff is based on the OASIS Exchange Model, but you should not have to go read that spec to understand how to use the element. Even though this element is based on an external model, it is a DITA element and the DITA spec should be complete in describing it. So, I'd rather not have a lot of links to the external spec, outside of the one in our non-normative references.

randerson updated comment 22/12/2021 19:08:50

Marking as **Closed**

Paragraph-level comments

The <simpletable> element is structurally less complex than the OASIS table, and so is an easier base for specialization. The <simpletable> element does not provide as much control over formatting. It is useful for describing lists of data with regular headings, such as telephone directory listings, display adapter configuration data, or API properties.

Annotation	Reviewer	Status	Type	Date	Topic version
API properties seems an odd thing to include in this list solely because of the existence of the properties table. I know most of my clients choose to just use regular tables instead of properties tables, but for the spec, it seems we might want to avoid suggesting that a simple table be used instead of a properties table.	dstevens	updated	comment	14/12/2021 14:51:39	
Changed to read "such as telephone directory listings or configuration settings."	keberlein	updated	comment	14/12/2021 19:46:09	
Marking this comment CLOSED .					

The <simpletable> element is structurally less complex than the OASIS table, and so is an easier base for specialization. The <simpletable> element does not provide as much control over formatting. It is useful for describing lists of data with regular headings, such as telephone directory listings, display adapter configuration data, or API properties.

Annotation	Reviewer	Status	Type	Date	Topic version
	sdoherly	updated	change	16/12/2021	

The element is structurally less complex than the OASIS table and so is an easier base for specialization. The element does not provide as much control over formatting. It is useful for describing lists of data with regular headings, such as telephone directory listings, display adapter configuration data, or API properties.

21:30:41

Removing the comma, but the wording has already been significantly tweaked.

keberlein updated comment 20/12/2021 11:49:14

Marking this comment **COMPLETED**.

Topic: simpletable (DA00509060)

Paragraph-level comments

The <simpletable> is designed for closer compatibility with HTML5 tables. It can be used for basic tabular data such as phone directory listings or parts lists. It can contain a title and allows column and row spanning. The @keycol attribute indicates which column represents the "key" or term-like column of the structure.

Annotation	Reviewer	Status	Type	Date	Topic version
"closer" -- than what? Incomplete comparison. Should explicitly state closer than <table>	dstevens	updated	comment	14/12/2021 15:09:49	
Changed to "close compatibility". Marking this comment COMPLETED .	keberlein	updated	comment	14/12/2021 19:27:02	

The <simpletable> is designed for closer compatibility with HTML5 tables. It can be used for basic tabular data such as phone directory listings or parts lists. It can contain a title and allows column and row spanning. The @keycol attribute indicates which column represents the "key" or term-like column of the structure.

Annotation	Reviewer	Status	Type	Date	Topic version
For some reason, it strikes me odd that we're singling out @keycol here. I'm also bothered that this description of @keycol seems to be adding semantic information, but the attribute description is just "it make a vertical header". Are there accessibility things included with @keycol? Or is it just "this makes it a header"? (And I know this isn't a user's guide, but how do I do something similar with a <table>? I'm also facinated that	zlawson	updated	comment	20/12/2021 20:59:31	

in some comment you mentioned that you always use `<simpletable>`, whereas I always use `<table>`, and I'm curious as to why we came to those decisions...and would that information be useful in an architectural topic?)

We single out `@keycol` because it is a unique-to-simpletable attribute.

keberlein updated comment 20/12/2021 21:41:35

They are entirely different models, and yes, `keycol` is unique to `simpletable` / is not based on an external standard.

The (somewhat) equivalent on the complex table would be `-- rowheader="firstcol"` is functionally the same as `keycol="1"`, but `keycol` can select any column and `rowheader` cannot. The other somewhat comparison is that `keycol="x"` is equivalent to setting `scope="col"` on column X.

randerson updated comment 21/12/2021 03:14:48

Basically complex tables are a lot more complex and it makes the markup uglier, while simpletables came up with `keycol` as a much simpler approach to the most common case.

No, we do not want to have an architectural topic that goes into the considerations between opting between simple tables and CALS table. That would be appropriate for a DITA about DITA or a company's internal guide to using DITA.

Re the other points that you raised:

- I do agree that we should make a pass over this topic and ensure that our descriptions of `@keycol` are clear and crisp; I'll take that on.
- **@Robert**, are they accessibility considerations with `@keycol` that we would bring up in a topics (or collection of topics) about accessibility?

keberlein updated comment 21/12/2021 13:33:08

Marking this comment **ACCEPTED**.

The `<simpletable>` element can be used as the base for specialized structures, such as the property and choice tables that are available in the Technical Content edition.

Annotation	Reviewer	Status	Type	Date	Topic version
The <code><table></code> usage section links to <code><simpletable></code> with a "if you're just doing something simple, go use simpletable".	zlawson	updated comment		20/12/2021 21:14:04	

Should we link to <table> for here? If you're doing something complex (such as?) go see <table>?

I'm ... not sure. Mostly because like Kris, I tend to steer people to simpletable to make content simpler, and now that this table has spanning I'm not sure of a good case where I'd say "this table is too complex, go use CALS"

randerson updated comment 21/12/2021 03:20:56

I think we do not need a cross reference to table.
Rationale:

- It's easy for someone to use the "DITA elements, A to Z" topic, and we do not want to clutter the spec with unnecessary cross references.
- The cross reference in the table topic is here because many people are not aware of simple table. I think everyone is aware of table!

keberlein updated comment 21/12/2021 13:43:44

Marking this comment **CLOSED**.

Examples

Annotation	Reviewer Status	Type	Date	Topic version
The example sucks. I've used simpletable with @keycol often over the years, although I've most often used it in two-column tables where I don't use a header, sort of as a clearly tabular representation of a definition list.				
I think we should replace the example. -----				
Replaced the code sample with that of a simple table with columns for "Menu items," "Calories," and "Price". The @keycol attribute is set for the first column.				
Marking this comment as COMPLETED .				
	keberlein updated comment		15/12/2021 15:41:08	

Draft comment: robander Dec 13 2021

I feel like the P / not P example is not very useful, as anyone without a grasp of boolean logic will not have any idea what this is trying to illustrate. Maybe we should use something like a parts list, which we already mention in Usage information?

Annotation	Reviewer Status	Type	Date	Topic version
	dstevens updated comment		14/12/2021	

I agree. It's not a particularly useful example even if you understand boolean logic -- because if you understand it, you would never need hte table.

15:19:45

I changed it to a simple table that contains food items and prices.

keberlein updated comment 14/12/2021 19:33:50

Marking this comment **COMPLETED**.

Example with column and row spanning

The following code sample shows a simple table that tracks meals. It has a title and column and row spans.

```
<simpletable>
  <title>Food log for Wednesday</title>
  <thead>
    <stentry>Meal</stentry>
    <stentry>Food</stentry>
  </thead>
  <strow>
    <stentry colspan="2">Fasting period</stentry>
  </strow>
  <strow>
    <stentry>Lunch</stentry>
    <stentry rowspan="2">Pasta</stentry>
  </strow>
  <strow>
    <stentry>Dinner</stentry>
  </strow>
</simpletable>
```

Annotation	Reviewer	Status	Type	Date	Topic version
Change title of the figure to "Example of a simple table with column and row spanning".	keberlein	updated	comment	15/12/2021 15:50:11	
Done, and marked as COMPLETED .					

The following code sample shows a simple table that tracks meals. It has a title and column and row spans.

Annotation	Reviewer	Status	Type	Date	Topic version
My little brain that avoids spanning as much as possible because it's hard could really, really use a screen shot of sample rendering.					
I can't visualize what this spanned table would look like, (or why I'd merge those cells together but that's more like the tone conversation).	zlawson	updated	comment	20/12/2021 21:02:51	

Agreed, Kris is adding screen caps

randerson updated comment 21/12/2021 03:19:19

Screen captures added. Marking this comment **COMPLETED.**

keberlein updated comment 21/12/2021 14:55:24

Example using @keycol

The following code sample shows how the @keycol attribute can be used. The value of the @keycol attribute specifies that the first column is the header column. This indicates that items in the first column are headers for the row. Rendering of the header column is left up to the implementation.

```

<simpletable keycol="1">
  <sthead>
    <stentry>Term</stentry>
    <stentry>Categorization</stentry>
    <stentry>Definition</stentry>
  </sthead>
  <strow>
    <stentry>Widget</stentry>
    <stentry>noun</stentry>
    <stentry>Thing that is used for something</stentry>
  </strow>
  <strow>
    <stentry>Frustration</stentry>
    <stentry>noun</stentry>
    <stentry>What you feel when you drop the widget</stentry>
  </strow>
</simpletable>

```

Annotation	Reviewer	Status	Type	Date	Topic version
Do keycol and rowheader do essentially the same thing on the two different table types? Since rowheader is explicitly mentioned to be not part of the standard on which the table is based, is there any reason we couldn't be using the same thing in the two tables?	dstevens	updated	comment	14/12/2021 16:39:23	
My comments:					
<ul style="list-style-type: none"> • @keycol has been around since the beginning; it's in heavy use. • @rowheader is part of the CALS (now OASIS Open Exchange Model) table. I'm no table expert; in the very rare situations where I use table (instead of simpletable), I rely on my editor to assist me in doing spans. 	keberlein	updated	comment	14/12/2021 19:57:29	
So, I don't think we want to consider making any changes here. @Robert, your take?					
	randerson	updated	comment	16/12/2021	

At this point I'd leave it alone, just because it would be another backwards incompatible change to markup that we know is in use, and it's quite late for that. (In my head I'm contrasting that with the "Remove xyz element" stuff under discussion, which is also backwards incompatible but with elements we think are virtually unused.)

The rowheader attribute doesn't come from CALS / OASIS Exchange, it's added by DITA explicitly for accessibility purposes. It is also added to colspec, which makes this trickier. It can say "The first column is a header" on tables, "The first column is nothing special" on tables, or it can say "This column is a header" when used on colspec.

Keycol on simpletables has the same goal of specifying a header, but simply (because simpletable...) does so by specifying the single column.

I'm sure there are ways we could reconcile this, but given that these table models already differ so much, I don't think it's worth it.

Marking this comment **CLOSED**.

keberlein updated comment 16/12/2021
17:28:32

The following code sample shows how the @keycol attribute can be used. The value of the @keycol attribute specifies that the first column is the header column. This indicates that items in the first column are headers for the row. Rendering of the header column is left up to the implementation.

Annotation	Reviewer	Status	Type	Date	Topic version
I realize examples can be hard to write. But this example gives me the opportunity to ask is there any concern that examples throughout the spec are all over the place in terms of tone. Some are definitely more real than others, some are generic, some specific, some funny, etc. In this particular case, it doesn't seem like coming up with a more applicable example would be hard.	dstevens	updated	comment	14/12/2021 15:18:39	
@Dawn (or others coming into this review), do you want to suggest a policy for examples? So far, we've only agreed on the following points: <ul style="list-style-type: none"> • Examples should not only relate to software and hardware • Examples should be realistic, if possible And I agree that this particular example sucks and recommended replacing it. I've added a separate comment	keberlein	updated	comment	14/12/2021 19:36:06	

so we can track that work item separately from this discussion of what examples should be like.

Sent e-mail to the TC. Marking this comment as **REFERRED.**

The following code sample shows how the @keycol attribute can be used. The value of the @keycol attribute specifies that the first column is the header column. This indicates that items in the first column are headers for the row. Rendering of the header column is left up to the implementation.

Annotation	Reviewer	Status	Type	Date	Topic version
Referring to the spec terminology discussion I would say that rendering of the header column is implementation dependent. I think, we should be clear and consistent in the wording.	fwegmann	updated	comment	20/12/2021 17:03:10	
Agreed. (Worth noting, this is a non-normative example, so terminology doesn't have to be quite as exact ... but no reason to differ I think)	randerson	updated	comment	21/12/2021 03:15:29	
I agree that we need to be more precise about our wording.					
Frank, you brought some important points out in your e-mail to the TC. However, Robert and I need to discuss them, and bring our recommendation to the TC. This requires selecting the phrases that will use, as well as drafting definitions for them. This work requires some thought, as it overlaps with OASIS rules for normative statements and conformance clauses.	keberlein	updated	comment	21/12/2021 13:37:35	
Marking this comment ACCEPTED.					

The following code sample shows how the @keycol attribute can be used. The value of the @keycol attribute specifies that the first column is the header column. This indicates that items in the first column are headers for the row. Rendering of the header column is left up to the implementation.

Annotation	Reviewer	Status	Type	Date	Topic version
and here's a completely different definition of @keycol than in the usage description.	zlawson	updated	comment	20/12/2021 21:05:21	
I know we don't want to be completely redundant, but consistency is good. And I'm very sensitive to descriptions that imply semantics, since that's so important with DITA.					

I'd quibble that this is not another definition, it's trying to use more tutorial-type approach in a non-normative example ... but yes, consistency would be helpful.

randerson updated comment 21/12/2021 03:18:31

Marking this comment as **CLOSED**, since another comment was accepted that covers crisping up the language around @keycol.

keberlein updated comment 21/12/2021 13:39:36

Topic: sthead (DA00509054)

Topic-level comments

Annotation	Reviewer	Status	Type	Date	Topic version
Do we need to discuss how this does (or does not) interact with the @scope of <stentry>?	zlawson	updated	comment	20/12/2021 21:09:59	
Included by thead and sthead in my draft comment that this should be explicitly covered - but need TC discussion of how	randerson	updated	comment	21/12/2021 03:26:08	
Marking this comment as REFERRED . @Robert, please make sure that you send an e-mail to to TC about the table things that you want the TC to consider.	keberlein	updated	comment	27/12/2021 15:05:58	

Paragraph-level comments

Topic: thead (DA00508593)

Topic-level comments

Annotation	Reviewer	Status	Type	Date	Topic version
In the <table> topic, there's some stuff about how <thead> helps with accessibility etc. Should some of that information go here?	zlawson	updated	comment	20/12/2021 21:46:36	
I think we're crying out for an accessibility section in the spec, and this should probably have a related link to it. @Kris does that make sense?	randerson	updated	comment	21/12/2021 03:40:13	

I've added a draft comment to the topic, stating that we should add a link (type unspecified) to the material (to be added) about accessibility.

keberlein updated comment 21/12/2021
13:56:10

Marking this comment **COMPLETED**.

Paragraph-level comments