Publication: Review M: DITAVAL elements (00815392-DC_1)

Topic: prop (DA00509468)

Paragraph-level comments

The `<prop>` element in a DITAVAL document specifies filtering or flagging actions that occur when rendering. The actions target conditional-processing attributes: `@props` or specializations of `@props`, such as `@audience`, `@deliveryTarget`, `@otherprops`, `@platform`, and `@product`.

Annotation	Reviewer	Status	Type	Date	Topic version
If we say "such as", we should list one or two, not all of them. I suggest reducing this list to at most 3 items. My comments to the attribute values made below apply to the revprop element as well. I'm not repeating them there.	gjoseph	updated	comment	16/3/2022 13:42:34	
Changed to read "The actions target the @props attribute or specializations of @props, including @audience, @deliveryTarget, @otherprops, @platform, and @product." keberlein Marking this comment **COMPLETED.**	keberlein	updated	comment	17/3/2022 13:51:00	

The `<prop>` element in a DITAVAL document specifies filtering or flagging actions that occur when rendering. The actions target conditional-processing attributes: `@props` or specializations of `@props`, such as `@audience`, `@deliveryTarget`, `@otherprops`, `@platform`, and `@product`.

Annotation	Reviewer	Status	Type	Date	Topic version
Do we explain in the definitions of @audience, @deliveryTarget, @otherprops, @platform, and @product that they are specializations of @props? It seems to be mentioned elsewhere, not not clearly in the definition of the attribute. (I know we already reviewed them, but I'm looking at it from a different perspective.) If someone makes a new specialization of @prop, does it automagically work with <prop>?	zlawson	updated	comment	21/3/2022 01:34:27	
A couple of points here:	keberlein	updated	comment	21/3/2022 16:31:28	

- We have not yet reviewed the definitions of the conditional-processing attributes. They are grouped as "metadata attributes" in the (larger) "universal attributes" group.
- Yes, we clearly state (and you can see this in the draft DITA 2.0 specification PDF that all these attributes are specializations of @props. (And that is new with DITA 2.0; previously these attributes were just part of the base.)
- If an implementation specializes @props, whether it "automagically" works with a DITAVAL document depends on the processor. If the processor is specialization-aware, yes, it will automatically work.

Marking this comment as **CLOSED**, as it requires no changes to the source.

The following table lists the functions that the `<prop>` element in a DITAVAL document performs:

Annotation	Reviewer	Status	Type	Date	Topic version
The following table lists the functions that the element in a DITAVAL file ~~document~~ perform s:	gjoseph	updated	change	16/3/2022 13:43:17	
No -- We use the phrase "DITAVAL document," not "DITAVAL file". Marking this comment as **CLOSED.**	keberlein	updated	comment	16/3/2022 22:54:08	

A `<prop>` element that specifies both an `@att` and `@val` attribute

Annotation	Reviewer	Status	Type	Date	Topic version
A element that specifies both an @att and a @val attribute	gjoseph	updated	change	16/3/2022 13:44:54	
Done. Marking this comment as **COMPLETED.**	keberlein	updated	comment	16/3/2022 22:57:49	

Sets a default action (exclude, flag, include, or pass through) for all conditional-processing attributes not explicitly specified in the DITAVAL document

Annotation	Reviewer	Status	Type	Date	Topic version
	gjoseph	updated	comment	16/3/2022 13:52:51	
Why do we list the actions here? If you want to list them, list them on the first row, not the last row.					

Annotation	Reviewer	Status	Type	Date	Topic version
I find the order of this topic rather confusing. It's impossible to understand the results listed in this table without first understanding the @att, @val, and @action attributes, which are only described significantly further down in the topic. Re your 1st point, I moved the actions to the first row. Re your 2nd point: This is a reference topic. It has a set order of sections. It is NOT intended to provide a user with a tutorial about prop elements. Marking this comment **COMPLETED.**	keberlein	updated comment		16/3/2022 23:17:08	

Rendering expectations

Annotation	Reviewer	Status	Type	Date	Topic version
Shouldn't the rev in the paragraphs below be prop? rev isn't an element in DITAVAL files.	dstevens	updated comment		15/3/2022 15:56:38	
Good catch; corrected. Marking as **COMPLETED.**	keberlein	updated comment		15/3/2022 20:39:55	

For the `@color` and `@backcolor` attributes on `<rev>` and `<revprop>`, processors *SHOULD* support the following values:

- The color names listed under the heading "<color>" in the XSL version 1.1 specification
- The associated hex code

Annotation	Reviewer	Status	Type	Date	Topic version
I found this very confusing, since 1) <rev> isn't in ditaval at all, and 2) <revprop> isn't mentioned till later in the document. If, as Dawn says, <rev> should be <prop>, then it would make a bit more sense. this is about <prop> and <revprop>, then it needs to be altered. Assuming that this shows up twice in the PDF because it appears in both the <prop> and <revprop>.	nharrison	updated comment		21/3/2022 19:08:05	
Already changed rev to prop here, as indicated in my response to Dawn. Marking this comment **CLOSED.**	keberlein	updated comment		21/3/2022 19:09:50	

- The color names listed under the heading "<color>" in the XSL version 1.1 specification

Annotation	Reviewer	Status	Type	Date	Topic

Do we still want to be limited to the XSL 1.1 specification?	zlawson	updated comment	21/3/2022 01:55:57
Hmm ... I added the link text to the xref element. The URL might still be what was first specified for DITA 1.1 in 2007. **@Robert?** Do we want to make a change here? The latest release of XSLT is XSLT 3.0 from 2017, but I'm not sure about XSL specifically. We potentially could: • Update the cross reference to point to a newer version of XSL spec (although I don't know if there have been changes for the color element.) • Leave the current cross reference as-is • Simply list the colors listed in the XSL 1.1 spec: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow ---- Marking this as **REFERRED,** and sending e-mail to the TC asking for a volunteer to do the research on this item.	keberlein	updated comment	21/3/2022 17:56:17

If two or more DITAVAL properties apply `@outputclass` flags to the same element, processors treat the flagged element as if each value was specified for the `@outputclass` attribute. The order of the DITAVAL-provided tokens is undefined.

Annotation	**Reviewer**	**Status**	**Type**	**Date**	**Topic version**
We refer to "DITAVAL properties" in this and other paragraphs, yet the last sentence refers to "DITAVAL tokens". I suggest we use either "tokens" or "properties" globally and not use the other one in the spec.	gjoseph	updated comment		16/3/2022 13:58:36	
We mean two different things here: • "DITAVAL properties" refer to the prop elements in a DITAVAL document. Each prop element specifies a ptoperty. • The "DITAVAL-provided tokens" refers to the tokens specified by the @outputclass attribute on a prop element. In general, the nouns "properties" and "tokens" are NOT interchangeable.	keberlein	updated comment		16/3/2022 23:02:15	

I have updated the wording to be more precise. Marking this comment as **COMPLETED.**

If two or more DITAVAL properties apply `@outputclass` flags to the same element, processors treat the flagged element as if each value was specified for the `@outputclass` attribute. The order of the DITAVAL-provided tokens is undefined.

Annotation	Reviewer	Status	Type	Date	Topic version
Do we want to mention <style-conflict> somewhere in here? Maybe that should be in Rendering expectations?	zlawson	updated	comment	21/3/2022 01:38:23	
@Zoe, why would we want to mention <style-conflict> in a "Rendering expectations" section in the props topic? I think that might belong in a conceptual topic that covered what rules can be specified in a DITAVAL document, but I don't think it belongs here in the props topic. Marking this comment as **CLOSED.**	keberlein	updated	comment	21/3/2022 16:55:03	

If two or more DITAVAL properties apply `@outputclass` flags to the same element, processors treat the flagged element as if each value was specified for the `@outputclass` attribute. The order of the DITAVAL-provided tokens is undefined.

Annotation	Reviewer	Status	Type	Date	Topic version
but is it the case that even if the order or DITAVAL-based is undefined, they'"re all processed as if they're all specified before any non-DITAVAL-specified tokens? Might make sense to clarify that	nharrison	updated	comment	21/3/2022 19:12:51	
I wondered if we should clarify that. **@Robert?**	keberlein	updated	comment	21/3/2022 20:24:05	
I've added a new example to explicitly cover this in the processing section for conditional processing.	randerson	updated	comment	21/3/2022 22:33:16	
Marking this comment as **ACCEPTED**, since I think we need to wait until the review of architectural content about conditional processing before resolving it.	keberlein	updated	comment	22/3/2022 09:57:05	

Specifies the action to be taken. The following values are supported:

exclude

Indicates that the content is excluded from the output, if all values for the specified attribute are excluded.

flag

Indicates that the content is included in the output and flagged, if the content has not been excluded.

include

Indicates that the content is included in the output. This is the default behavior, unless otherwise set.

passthrough

Indicates that the content is included in the output and that the attribute value is preserved. This enables further processing by a runtime engine. The attribute value is preserved in the syntax that is required by the runtime engine.

Annotation	Reviewer	Status	Type	Date	Topic version
I think this topic should really have a specific related link to Conditional processing to better explain the "if all values for the specified attribute are excluded" I think there needs to be an addition "if the content has not been excluded by other attributes" or something.	zlawson	updated comment		21/3/2022 01:42:46	
I think we need to do a better job of explaining what we are trying to get at with the following phrasing: • "exclude": "... if all values for the specified attribute are excluded." • "flag": "... if the content has not been excluded." • "include": "... unless otherwise set." This will be in the content of the architectural topics about conditional processing. Marking this comment as **ACCEPTED**, since I think we need to wait on the review of the architectural topics before adjusting any wording here or adding related links.	keberlein	updated comment		21/3/2022 17:17:15	

Specifies the conditional-processing attribute that is targeted. The value is the literal attribute name or the name of a group within one of those attributes, with the group name specified using the generalized attribute syntax. If the @att attribute is absent, then the <prop> element declares a default behavior for anything not explicitly specified in the DITAVAL document.

Annotation	Reviewer	Status	Type	Date	Topic version
What is generalized attribute syntax? The only mention in the current spec is this paragraph. I searched for generalized attribute. There's a description of grouping syntax, but it's not overly clear. (at least ot at quarter to ten pm)	zlawson	updated comment		21/3/2022 01:47:23	

Again, I think we need to wait on the review of the architectural topics on conditional processing before we adjust any wording here.

Marking this comment as **ACCEPTED.**

| | keberlein | updated comment | 21/3/2022 17:19:14 |

(If the `@action` attribute is set to flag) Specifies the background color for flagged text. Colors can be entered by name or code. When images are flagged, the background color is rendered as a thick border. If the `@action` attribute is not set to flag, this attribute is ignored.

Annotation	Reviewer	Status	Type	Date	Topic version
(~~If~~ When the @action attribute is set to flag ~~,)~~ s ~~S~~pecifies the background color for flagged text. Colors can be entered by name or code. When images are flagged, the background color is rendered as a thick border. If the @action attribute is not set to flag, this attribute is ignored.	gjoseph		updated	change	16/3/2022 14:18:51
Marking this comment as **CLOSED,** since this point is being handled by another comment.	keberlein		updated comment	19/3/2022 21:48:06

(If the `@action` attribute is set to flag) Specifies the background color for flagged text. Colors can be entered by name or code. When images are flagged, the background color is rendered as a thick border. If the `@action` attribute is not set to flag, this attribute is ignored.

Annotation	Reviewer	Status	Type	Date	Topic version
Global comment whenever we're mentioning color: When we state by name, do we need to indicate which list of color names? I think named colors have grown over the years. Are there differences beten XLS-FO and CSS? Instead of code, do we want to say "hex code"?	zlawson		updated comment	21/3/2022 01:52:39
A couple of points:	keberlein		updated comment	21/3/2022 17:04:49

- Yes, we should state "hex code" rather than just code. I've made that change in the source.
- We have a normative statement about WHAT colors processors SHOULD support for @backcolor and @color. It appears in the "Rendering expectations" section for both the prop and revprop topics. Although we do not state it explicitly, I think it is implicit that processors can support other colors if they want to. **@Robert,** do you agree? Do we need to consider making a normative statement about that?

- The normative statement states that the colors that processors should SUPPORT are outline in the XSL spec. You've made another comment elsewhere in the review asking whether we should reference a more recent version of the XSL specification than version 1.1 ...
- Differences between XSL-FO and CSS is not relevant here.

Based on conversation with Robert, clarified the wording in the normative statement about @backcolor and @color to make it even clearer that processors can support other colors that the subset specified in the XSL 1.1 spec.

Marking this comment **COMPLETED.**

(If the `@action` attribute is set to flag) Specifies the color for flagged text. Colors can be entered by name or code. When images are flagged, the color is rendered as a thin border. If the `@action` attribute is not set to flag, this attribute is ignored.

Annotation	Reviewer	Status	Type	Date	Topic version
(~~If~~ When the @action attribute is set to flag ,) s ~~S~~pecifies the color for flagged text. Colors can be entered by name or code. When images are flagged, the color is rendered as a thin border. If the @action attribute is not set to flag, this attribute is ignored.	gjoseph	updated	change	16/3/2022 14:20:07	
Marking this comment as **CLOSED,** since this point is being handled by another comment.	keberlein	updated	comment	19/3/2022 21:48:39	

(If the `@action` attribute is set to flag) Specifies a value for the `@outputclass` attribute. The flagged element is treated as if the specified `@outputclass` value was specified on that element.

Annotation	Reviewer	Status	Type	Date	Topic version
Should this description also contain the sentence "If the @action attribute is not set to flag this attribute is ignored"? as all others include? I personally think the parenthetical remark is enough and the sentence could be removed from the others, but I would advocate consistency either way.	dstevens	updated	comment	15/3/2022 16:10:04	
Added the missing sentence. I'd be happy to remove it ("If the @action attribute is not set to flag, this attribute is ignored.") from all the attribute definitions -- **@Robert?**	keberlein	updated	comment	15/3/2022 20:44:36	

Setting the status of this comment to **REFERRED.** I'll be sending an e-mail to the TC about this point AND Gershon's comment that we should remove the parenthetical statement that begins each .of these attribute definitions | keberlein | updated comment | 19/3/2022 21:44:45

(If the @action attribute is set to flag) Specifies a value for the @outputclass attribute. The flagged element is treated as if the specified @outputclass value was specified on that element.

Annotation	Reviewer	Status	Type	Date	Topic version
When (If the @action attribute is set to flag ,) S specifies a value for the @outputclass attribute. The flagged element is treated as if the specified @outputclass value was specified on that element. If the @action attribute is not set to flag, this attribute is ignored.	gjoseph		updated change	16/3/2022 14:22:50	
Setting the status of this comment to **REFERRED**. I'll be sending an e-mail to the TC about this point AND Dawn's related suggestion that we remove the stock sentence about "If the @action attribute is not set to flag this attribute is ignored".	keberlein		updated comment	19/3/2022 21:46:40	

(If the @action attribute is set to flag) Specifies the formatting to use for flagged text. This attribute can contain multiple space-delimited tokens. If the @action attribute is not set to flag this attribute is ignored.

Annotation	Reviewer	Status	Type	Date	Topic version
When (If the @action attribute is set to flag ,) s Specifies the formatting to use for flagged text. This attribute can contain multiple space-delimited tokens. If the @action attribute is not set to flag , this attribute is ignored.	gjoseph		updated change	16/3/2022 14:23:39	
Marking this comment as **CLOSED,** since this point is being handled by another comment.	keberlein		updated comment	19/3/2022 21:49:15	

(If the @action attribute is set to flag) Specifies the formatting to use for flagged text. This attribute can contain multiple space-delimited tokens. If the @action attribute is not set to flag this attribute is ignored.

Annotation	Reviewer	Status	Type	Date	Topic version
Why don't we list the supported tokens here?	zlawson		updated comment	21/3/2022 01:37:11	

When I was reviewing, I was actually thinking I could put any CSS into here if I were rendering HTML, but I now think that's wrong. I think we need to be explicit about what's allowed here. I'm not sure I'd go back to look at the Rendering expectations to get the list of tokens.

It's kind of hard to list the supported tokens here. Why? We specify a list of tokens that we state processors SHOULD support -- bold, double-underline, italic, overline, and underline -- but we also state that processors MAY support proprietary tokens. I think a list of supported tokens would be appropriate for an implementation's *User Guide*, but it is not something that we can put here in the spec. Marking this comment **CLOSED**.	keberlein updated comment	21/3/2022 17:09:23	

- Sets a default action of exclude. With the exception of the other conditions specified in the DITAVAL document, the content of any element that specifies a conditional-processing attribute is excluded from the rendered output.

Annotation	Reviewer	Status	Type	Date	Topic version
Worth being explicit that if there are no @props related attributes, the content is included? Maybe better to say "Sets a default action to exclude all conditions"? My initial read is that absolutely everything is excluded except @otherprops and @product=base-product.	zlawson	new	comment	21/3/2022 01:50:21	
We do state clearly here that that content excluded is "the content of any element **that specifies a conditional-processing attribute**" (emphasis added.) I think that the existing phrasing is more precise than what you suggest. **@Robert?** I've added the following elaboration to the end of the example: "When a DITA map is processed using the above DITAVAL document, the following DITA elements are excluded: • Any element for which the @audience, @deliveryTarget, @platform, and @props attributes (or specializations of @props) specify a non-null value. • Any element for which the @product attribute specifies a value that is not equal to base-product. All other content is included."	keberlein		updated comment	21/3/2022 17:29:15	

> **@Zoe**, does this clarify things for you?

Topic: style-conflict (DA00508674)

Paragraph-level comments

In the case of conflicts between flagging methods that are specified for elements at different levels of the containment hierarchy, the flagging method specified for the element at the lowest level of the hierarchy applies. For example, if the `<section>` element is to be flagged with green text and a `<p>` element is to be flagged with red text, a paragraph within a section should be rendered with red text.

Annotation	Reviewer	Status	Type	Date	Topic version
Is this Usage Information basically saying "in this scenario, <style-conflict> doesn't apply"?	zlawson	updated	comment	21/3/2022 02:05:14	
No -- What this section is trying to state is the following: • DITA elements nest. Because of this, you might have a section element with otherprops="section", and that section might contain a p element with otherprops="p". • You could have a DITAVAL document that specifies two properties: 1) otherprops="section" should be flagged with green text, and 2) otherprops="p" should be flagged with red text. This does not create a style conflict. • Style conflicts apply when there are conflicting flagging methods applied to the SAME element. **@Robert,** do I have this correct? If so, we might want to be more explicit about what a style conflict is. This could be handled by updating the shortdesc for this topic. I have updated the wording about thisto be more precise and more accurately reflect the scenario that I described above. ---- Confirmed with Robert that my interpretation is correct. Updated the shortdesc in this topic, and moved the content about "when glagging methods are specified for elements at different levels of the containment hierarchy" to the architectural topic about flagging. Marking this comment **COMPLETED.**	keberlein	updated	comment	21/3/2022 18:32:12	
My understanding is the same as what @Kris says - I stumbled over this the first time I read it as well, I think	randerson	updated	comment	21/3/2022 22:30:08	

updating wording is the best resolution here.

Any element that specifies a value of dita lwdita (or lwdita dita) is rendered with a light-yellow background color.

Annotation	Reviewer	Status	Type	Date	Topic version
Any element that specifies a value of "dita lwdita " (or "lwdita dita ") is rendered with a light-yellow background color.	gjoseph	updated	change	16/3/2022 15:00:11	
Done. Marking this comment **COMPLETED.**	keberlein	updated	comment	16/3/2022 23:09:00	

Topic: revprop (DA00509116)

Paragraph-level comments

The @rev attribute is not designed to be used for version control.

Annotation	Reviewer	Status	Type	Date	Topic version
or conditions? (depending on what terminology we use). Worth stating explicitly here that @rev is only flagging, not include/exclude?	zlawson	updated	comment	21/3/2022 01:57:26	
A couple of points: • We state explicitly in the shortdesc for this topic that @rev can be used only for flagging, so I do not think we need to restate that here in the "Usage information" section. • I've updated the "Usage information" content here to read: "Neither the revprop element or the @rev attribute are designed to be used for version control. Marking this comment **COMPLETED.**	keberlein	updated	comment	21/3/2022 18:10:17	

Rendering expectations

Annotation	Reviewer	Status	Type	Date	Topic version
Similar to my comment on <prop>, do we want to mention <style-conflict> somewhere in here?	zlawson	updated	comment	21/3/2022 01:58:13	

No. Marking this comment **CLOSED.**	keberlein	updated comment		21/3/2022 18:17:02	

`@action` (REQUIRED)

Annotation	Reviewer	Status	Type	Date	Topic version
similar comments as for <prop> Need to link to wherever we explain how multiple values of tokens interact.	zlawson	updated comment		21/3/2022 01:59:39	
Marking this comment **CLOSED.** Any changes that are needed to the source will be tracked through the ACCEPTED comment on the prop topic.	keberlein	updated comment		21/3/2022 18:18:21	

(If the `@action` attribute is set to flag) Specifies a color, style, or character to be used for rendering a change bar. If the `@action` attribute is not set to flag, this attribute is ignored.

Annotation	Reviewer	Status	Type	Date	Topic version
When ~~(If~~ the @action attribute is set to flag ~~,)~~ s ~~S~~pecifies a color, style, or character to be used for rendering a change bar. If the @action attribute is not set to flag, this attribute is ignored.	gjoseph	updated	change	16/3/2022 14:42:28	
Marking this comment as **CLOSED**, since this point is being handled by another comment.	keberlein	updated comment		19/3/2022 21:50:31	

(If the `@action` attribute is set to flag) Specifies a color, style, or character to be used for rendering a change bar. If the `@action` attribute is not set to flag, this attribute is ignored.

Annotation	Reviewer	Status	Type	Date	Topic version
The prop element does not have the @changebar attribute. Is this by design or an omission in the spec, dtd, or both?	gjoseph	updated comment		16/3/2022 14:44:20	
The @changebar attribute is only specified on the <revprop> element. It is only used for revision flagging Marking this comment **CLOSED.**	keberlein	updated comment		16/3/2022 23:04:18	

(If the `@action` attribute is set to flag) Specifies a color, style, or character to be used for rendering a change bar. If the `@action` attribute is not set to flag, this attribute is ignored.

Annotation	Reviewer	Status	Type	Date	Topic version
Dumb question - do we need to define what a changebar is?	zlawson	updated	comment	21/3/2022 02:02:00	
No -- It's a standard publishing convention. Marking this comment **CLOSED.**	keberlein	updated	comment	21/3/2022 18:14:45	

`@outputclass`

Annotation	Reviewer	Status	Type	Date	Topic version
Assuming this is single sources from prop, then my previous comment will already be applied here. But just in case, should there be an explicit sentence about the attribute being ignored as is on backcolor, changebar, color, and style.	dstevens	updated	comment	15/3/2022 16:45:57	
Yes, the attribute definition is reused, so this is already handled. Marking this comment as **CLOSED.**	keberlein	updated	comment	15/3/2022 20:53:46	

(If the `@action` attribute is set to flag) Specifies the formatting to use for flagged text. This attribute can contain multiple space-delimited tokens. If the `@action` attribute is not set to flag this attribute is ignored.

Annotation	Reviewer	Status	Type	Date	Topic version
same comment as for <prop>	zlawson	updated	comment	21/3/2022 02:00:05	
Marking this comment **CLOSED.** This content is single-sourced.	keberlein	updated	comment	21/3/2022 18:19:24	

Topic: val (DA00509513)

Paragraph-level comments

This section contains examples of DITAVAL documents and how they can be used.

Annotation	Reviewer	Status	Type	Date	Topic version
Along the lines of Gershon's earlier comments, we probably	nharrison	updated	comment	21/3/2022 19:32:26	

Annotation	Reviewer	Status	Type	Date
want to use 'DITAVAL files' instead of 'DITAVAL documents' here. Even after decades in the industry, I still think of a 'document' as something read and acted upon by humans, not processors. No -- we cannot call them "files". A DITAVAL might be any of the following: • A file on the file system • A set of rules in memory • Another way of storing conditional-processing information that can be expressed using DITAVAL syntax Marking this comment as **REJECTED**.	keberlein	updated	comment	21/3/2022 20:17:31

The following code sample shows a DITAVAL document that includes content, flags certain content, flags certain revisions, and provides a background color for when there are style conflicts:

Annotation	Reviewer	Status	Type	Date	Topic version
includes certain content?	zlawson	updated	comment	21/3/2022 02:07:05	
The DITAVAL document here in the example effectively includes ALL content. Yes, it sets a default policy for elements that specify audience="everybody", but by default ALL content is included. So, maybe not the best example ... I've updated the example to exclude audience="internal-test" and updated the intro to the code block. Marking this comment **COMPLETED.**	keberlein	updated	comment	21/3/2022 18:57:42	

• Elements marked with `rev="1.2"`are flagged with the default revision flags, which are implementation dependent.

Annotation	Reviewer	Status	Type	Date	Topic version
missing space after "1.2"	dstevens	updated	comment	15/3/2022 16:51:42	
Corrected. Marking as **COMPLETED.**	keberlein	updated	comment	15/3/2022 20:38:02	

• Elements marked with `rev="1.2"`are flagged with the default revision flags, which are implementation dependent.

Annotation	Reviewer	Status	Type	Date	Topic

Annotation	Reviewer	Status		Date
				version
Do we ever talk about "default revision flags" anywhere? Should there be something about that in <revprop>?	zlawson	updated comment		21/3/2022 02:08:31
The "Rendering expectations" section of the mentions that "processors can provide default alternate text to indicate the start and end point of the flagged content." I suspect that is what this example implies. **@Robert**, is that the case? Or do we need to cover an expectation that processors will provide some sort of default revision flags? --- I've added the following content to an archSpec topic: "Processors can implement default behaviors for flagging. This might include alternate text to indicate the start and end points of revised content or stylistic formatting when no specific flagging behavior is specified." Marking this comment as **COMPLETED.**	keberlein	updated comment		21/3/2022 15:42:59
I think this is trying to convey that an implementation can provide its own default flags. For example, if you say to flag rev="1.2" but don't say how, a processor can do its own thing to flag it (start/end images or similar). We should clarify this in the arch spec when talking about ditaval (basically saying that a processor can do it, but don't mandate anything).	randerson	updated comment		21/3/2022 22:26:46

Topic: alt-text (DA00508762)

Paragraph-level comments

The `<alt-text>` element in a DITAVAL document specifies alternate text for an image that is used to flag content. If an image is not specified, the text is used to mark the flagged content.

Annotation	Reviewer	Status	Type	Date	Topic version
The element in a DITAVAL file ~~document~~ specifies alternate text for an image that is used to flag content. If an image is not specified, the text is used to mark the flagged content.	gjoseph	updated	change	16/3/2022 13:14:44	
No -- We use the phrase "DITAVAL document," not "DITAVAL file".	keberlein	updated	comment	16/3/2022 22:45:49	

Marking this comment as **CLOSED.**

The `<alt-text>` element in a DITAVAL document specifies alternate text for an image that is used to flag content. If an image is not specified, the text is used to mark the flagged content.

Annotation	Reviewer	Status	Type	Date	Topic version
The parent topic uses "DITAVAL file" and this topic uses "DITAVAL document". We should use one of these terms consistently throughout the spec and not use the other in the spec. My personal preference is to use "DITAVAL file".	gjoseph	updated	comment	16/3/2022 13:15:48	
The convention is to use the phrase "DITAVAL document". Marking this comment **CLOSED.**	keberlein	updated	comment	16/3/2022 22:46:19	

The `<alt-text>` element in a DITAVAL document specifies alternate text for an image that is used to flag content. If an image is not specified, the text is used to mark the flagged content.

Annotation	Reviewer	Status	Type	Date	Topic version
I'm with Gershon on preferring a consistent usage, and on preferring 'file' to 'document' for that usage, but we do need to start the sentence with either "The <alt-test> element...." or "This element" for this description, and similar changes to all the following topics where he's made this change.	nharrison	updated	comment	21/3/2022 18:56:07	
We cannot use the term DITAVAL file. The word "file" implies something concrete that might not be present. The DITAVAL could be "virtual" or syntax that is passed somehow into the processing chain. Marking this comment **CLOSED.**	keberlein	updated	comment	21/3/2022 19:05:35	

If no alternate text is specified, processors can provide default alternate text to indicate the start and end point of the flagged content.

Annotation	Reviewer	Status	Type	Date	Topic version
Should this be normative? (It probably shouldn't be, but for some reason my brain questioned it this review.)	zlawson	updated	comment	21/3/2022 01:21:51	
@Robert, will you explain to Zoe why we did not use a normative "MAY" here?	keberlein	updated	comment	21/3/2022 15:24:56	

Update based on spec editors' call with Robert on 21 March 2022:

Basically, we try to be very careful about when we use normative statement, especially "MAY" ones. The "MAY" statements are pretty nebulous ... We tend to save them for when we think people might really question when the spec permits them to do something.

Here, that really isn't an issue.

Marking this comment **CLOSED.**

The following code sample shows a DITAVAL document that is used to render icons before content that is specific to particular audiences. The `<alt-text>` element provides alternate text for the icons:

Annotation	Reviewer	Status	Type	Date	Topic version
The following code sample shows a DITAVAL file ~~document~~ that is used to render icons before content that is specific to particular audiences. The element provides alternate text for the icons:	gjoseph	updated	change	16/3/2022 13:21:33	
No -- We use the phrase "DITAVAL document," not "DITAVAL file". Marking this comment as **CLOSED.**	keberlein	updated	comment	16/3/2022 22:46:40	

```
<val>
   <prop action="flag" att="audience" val="novice">
   <startflag imageref="novice-icon.gif">

   <alt-text>Novice</alt-text>
   </startflag>
   </prop>
   <prop action="flag" att="audience" val="expert">
   <startflag imageref="expert-icon.gif">

   <alt-text>Expert</alt-text>
   </startflag>
   </val>
```

Annotation	Reviewer	Status	Type	Date	Topic version
Is it worth having an example that includes an image of rendered output? Maybe not specifically here, but somewhere in the ditaval examples, or maybe in the flagging section?	zlawson	updated	comment	21/3/2022 01:24:31	

Is it worth having a simple example of using {{ for <startflag>/<alt-text> and }} for <endflag>/<alt-text>?

Annotation	Reviewer	Status	Type	Date
I honestly do not know if we need an example of rendered content or not. **@Robert?** I don't think we want to have an example of specifying "{{" and "}}" for alternate text. That's not alternate text, just formatting, and so it would NOT be accessible to folks using screen readers. --- Based on spec editors' call on 22 March 2022. We need to leave this open for a while. There currently is an example of rendered content in one of the archSpec topics, although we don't know if it is necessary there. Marking this as **ACCEPTED**, and we can track whether or not not to have such an example as we progress with the next review of the conditional processing content.	keberlein	updated	comment	21/3/2022 18:22:36

Topic: endflag (DA00508812)

Paragraph-level comments

The `<endflag>` element in a DITAVAL document specifies information that identifies the end of flagged content. The information can be an image, alternate text, or both.

Annotation	Reviewer	Status	Type	Date	Topic version
The element in a DITAVAL file ~~document~~ specifies information that identifies the end of flagged content. The information can be an image, alternate text, or both.	gjoseph	updated	change	16/3/2022 13:23:03	
No -- We use the phrase "DITAVAL document," not "DITAVAL file". Marking this comment as **CLOSED.**	keberlein	updated	comment	16/3/2022 22:47:26	

Specifies a URI reference to the image, using the same syntax as the `@href` attribute. See The href attribute for information on supported values and processing implications.

Annotation	Reviewer	Status	Type	Date	Topic version
In a recent call, I thought I heard that all elements with @href	nharrison	updated	comment	21/3/2022	

| | | | | | 19:23:36 |

should also have the other 3 linking @s; if this is meant to be used like @href, should it also have those other 3 @s?. of course, this would also apply to <startflag>

Confirmed with Robert that having @format, @scope, and @type on all elements that specify @href ONLY applies to DITA elements. DITAVAL document elements are different beasts :)	keberlein	updated comment	21/3/2022 20:20:26

Marking this comment **CLOSED.**

```
<val>
   <prop action="flag" att="audience" val="administrator">
   <startflag>
   <alt-text>Administrator content</alt-text>
   </startflag>
   <endflag>
   <alt-text>End of administrator content</alt-text>
   </endflag>
   </prop>
   </val>
```

Annotation	Reviewer	Status	Type	Date	Topic version
I think we should bold the start and end tags of the endflag element. Remove the hyphen from the start-flag and end-flag elements in the prose that introduces the element. They should be startflag and endflag, not start-flag and end-flag (based on the text in this topic, I did NOT check the DTD...)	gjoseph		updated comment	16/3/2022 13:26:39	
Corrected the names of the elements. Marking this comment as **COMPLETED.**	keberlein		updated comment	16/3/2022 22:51:44	

Topic: DITAVAL elements (DA00509121)

Topic-level comments

Annotation	Reviewer	Status	Type	Date	Topic version
Wondering whether this is the place to talk about global versus branch DITAVALs? At a minimum, it might be good to mention that all the conventions here are equally applicable to global and branch filtering.	sdoherty		updated comment	15/3/2022 13:56:38	

I'm glad you brought this up, and it's certainly something that we need to address here. I do want to wait to add new verbiage, however, until after we have reviewed the following: • Conditional processing (archSpec) topics • Branch filtering (archSpec) topics • DITAVAL-reference domain topics So, marking this as **ACCEPTED** and logging it as a work item on the Wiki page at https://wiki.oasis-open.org/dita/DITA2.0-review-work-items	keberlein	updated comment	19/3/2022 21:24:08	
I think this is the wrong place for my comment. It probably should be a part of the Conditional Processing (profiling) section we haven't reviewed yet, but I don't want to forget it. Some processors are now allowing for multiple ditavals. Do we want to cover that scenario somewhere? Add that information when discussing error conditions?	zlawson	updated comment	21/3/2022 01:03:42	
Re "some processors are now allowing for multiple ditavals" -- are you referring to branch filtering and use of the DITAVAL reference domain? If so, we'll consider if we need to adjust the wording of this topic AFTER we've reviewed the branch filtering and DITAVAL reference-domain topics. Stan Doherty already made a comment raising that point, which I marked as **ACCEPTED**. If you are NOT referring to branch filtering and the use of the DITAVAL reference domain, are you referring to DITA-OT enabling the use of multiple DITAVAL documents, as outlined in https://www.dita-ot.org/2.5/release-notes/index.html#highlights__2637? If so, I think that's a similar issue that we should consider after reviewing the architectural content about conditional processing. Marking this comment as **ACCEPTED.**	keberlein	updated comment	21/3/2022 15:23:24	

Paragraph-level comments

DITAVAL elements

Annotation	Reviewer	Status	Type	Date	Topic version
I'm looking for a place that explicitly indicates processing order going from top to bottom, which might seem obvious, but a lot of my clients choose to exclude all attributes and then add only the ones they want to include, and they are	dstevens	updated comment		15/3/2022 16:05:39	

sometimes confused by seemingly conflicting elements without the understanding that order matters.

@**Dawn**, Robert and I really are not sure what you mean here. Can you be more explicit about what your clients try to do and where they run into trouble? Thanks.	keberlein	updated comment	15/3/2022 21:06:31
I'll try. Ultimately the examples address the point, especially in the val topic, where first everything is excluded and then individual items are added back in. My clients first don't understand why it wouldn't conflict -- one line says exclude and another says include, so they need to understand that the commands are done sequentially, first things are turned off then something is turned on. Second they don't necessarily understand why they might want to do this -- so you aren't explicitly turning off every item so each time you add another value you add an another exclude. Does that help? Maybe it's not really necessary because as I said the example shows it.	dstevens	updated comment	15/3/2022 21:21:02

	keberlein	updated comment	19/3/2022 21:37:31

Thanks, @**Dawn.** This does clarify your original comment. I was tripped up by your mention of order, which really does not apply to how processors handle the properties specified in a DITAVAL document.

What I am taking away from your 2nd comment is that your customers are confused by the breadth of what can be specified in a DITAVAL document:

- Set a default policy for ALL conditional processing attributes
- Set a default policy for a specific conditional processing attribute
- Set an action for a specific conditional-processing attribute/value pair

And also confused by the practice of combining specifying default policies AND (seemingly conflicting) specifying actions for specific conditional-processing attribute/value pairs.

I hope that this is clearer in the revised element-reference topics, and that it will be even clearer when we have revised architectural topics about conditional processing.

I'm marking this comment as **ACCEPTED**, since I think we need to wait until after the review of the architectural topics to assess whether we need additional revisions to clarify this area of potential confusion.

A conditional processing profile (DITAVAL file) identifies the attribute values that are conditionally processed before rendering. The profile has an extension of .ditaval.

Annotation	Reviewer	Status	Type	Date	Topic version
I find that some people don't equate flags as "conditional processing" but only the inclusion/exclusion. Would it be helpful to explicitly indicate the ditaval file is also used for flagging attributes?	dstevens	updated	comment	15/3/2022 15:53:10	
I think you've identified an important point; I agree that people think of conditional processing as filtering, not as filtering and flagging. I think we have two choices here, then: 1. Stress that conditional processing = filtering + flagging 2. Change the terminology that we use. I think we'd need to avoid the term "conditional processing profile" ... #2 might be a radical change that could be difficult to implement, but I do think that for most folks the phrase "conditional processing" = filtering/inclusion or exclusion. There are many users in the DITA community who have no idea that DITA supports flagging, especially folks who learned DITA using the SDL CCMS ... Marking this comment as **REFERRED**, and planning to send an e-mail to the TC to address this.	keberlein	updated	comment	15/3/2022 20:52:17	

A conditional processing profile (DITAVAL file) identifies the attribute values that are conditionally processed before rendering. The profile has an extension of .ditaval.

Annotation	Reviewer	Status	Type	Date	Topic version
A conditional processing profile (DITAVAL) file) identifies the attribute values that are conditionally processed before rendering. The DITAVAL profile file has an extension of .ditaval.	gjoseph	updated	change	16/3/2022 13:06:32	
No -- we need to refer to "DITAVAL documents," not "DITAVAL files. Marking this comment **CLOSED.**	keberlein	updated	comment	16/3/2022 22:44:20	

Topic: startflag (DA00508991)

Paragraph-level comments

Specifies a URI reference to the image, using the same syntax as the `@href` attribute. See The href attribute for information on supported values and processing implications.

Annotation	Reviewer	Status	Type	Date	Topic version
I recommend cautioning the reader that there are processor-specific issues here. DITA-OT has had difficulty with processing these images across folders.	sdoherty	updated	comment	15/3/2022 13:53:21	
I'm nervous about that sort of caution in the specification. The spec is declaring what the proper syntax is. If DITA-OT does not handle it properly, that's a bug in DITA-OT (and saying here 'processors might not work right' almost gives it a free pass to never fix that).	randerson	updated	comment	15/3/2022 15:12:08	
Totally agree with Robert here. @Stan, your comment would be appropriate for an implementation's User Guide, but not the spec. Marking this comment **CLOSED.**	keberlein	updated	comment	15/3/2022 20:55:23	

Specifies a URI reference to the image, using the same syntax as the `@href` attribute. See The href attribute for information on supported values and processing implications.

Annotation	Reviewer	Status	Type	Date	Topic version
what about the other 3 linking @s (see my question wrt <endflag>	nharrison	updated	comment	21/3/2022 19:24:17	
Marking this comment as **CLOSED,** since we can track any necessary changes with the comment that you made in the endflag topic.	keberlein	updated	comment	21/3/2022 20:21:51	

The following code sample shows a DITAVAL document that is used to render icons before content that is specific to a particular operating system. The `<startflag>` elements specify the icons, and the `<alt-text>` elements provides alternate text.

Annotation	Reviewer	Status	Type	Date	Topic version
	gjoseph	updated	change	16/3/2022	

The following code sample shows a DITAVAL document that is used to render icons before content that is specific to a particular operating system. The elements specify the icons, and the elements specify the ~~provides~~ alternate text.

Done. Marking this comment **COMPLETED.**

keberlein updated comment

16/3/2022
23:05:33