
Stage two: #164 Redesign <hazardstatement>

Redesign the hazard statement domain to better support current standards, authoring requirements, and rendering requirements.

Date and version information

This proposal contains the following information:

Date that this feature proposal was completed

10 September 2019

Champion of the proposal

Kristen James Eberlein, Eberlein Consulting LLC

Links to any previous versions of the proposal

Not applicable

Links to minutes where this proposal was discussed at stage 1 and moved to stage 2

[20 August 2019](#)

In addition, this proposal was discussed (a request for early feedback) at the following TC meetings:

- 03 September 2019
- 10 September 2019

Reviewers for Stage 2 proposal

The official stage 2 reviewers are as follows:

Scott Hudson

Dawn Stevens

In addition, the following individuals were asked for feedback on this proposal:

Jang Graat, Former individual member

Amber Swope, Individual member

Links to e-mail discussion that resulted in new versions of the proposal

Not applicable

Link to the GitHub issue

[Issue #64](#)

Original requirement or use case

In May 2018, the following TC members suggested changes to the hazard statement domain for DITA 2.0:

- Jang Graat, Individual member (no longer an OASIS member)
- Dawn Stevens, Comtech Services, Inc.
- Amber Swope, Individual member

All three TC members were motivated to improve the hazard statement domain by increasing its alignment with various safety standards; in addition, they wanted to make it easier for content developers to use and better able to support rendering requirements.

In August and September 2019, after a careful study of existing safety standards and the changes suggested by DITA TC members and the larger DITA community, I presented various options to the TC and developed consensus on the solution that is outlined in this proposal.

Use cases

This proposal will enable hazard statements authored in DITA to meet the following objectives:

Better comply with the ANSI Z535.6 standard

While the ANSI Z535.6 standard provides great flexibility for companies to implement its recommendations, it does provide strict requirements for safety alert symbols and signal words.

ANSI Z535.6 defines a *signal word* as a “word that calls attention to a safety message or messages or a property damage message or messages, and designates a degree or level of hazard seriousness.” The standard only permits four signal words: DANGER, WARNING, CAUTION, and NOTICE.

The @type attribute on the <hazardstatement> element specifies the signal word. For DITA 1.2 and 1.3, the values of the @type attribute are identical to those permitted on <note>. That allows a lot of values that are not germane for hazard statements and so can introduce authoring dissonance.

Improve the semantic nature of the specialization base

Currently, <hazardstatement> is specialized from , and its child elements are specialized from . This specialization hierarchy does not make sense.

Enable hazard images to be associated with <messagepanel> and its child elements

Currently, the hazard images are part of the content model for the <hazardstatement> element. This presents a problem when the <hazardstatement> element contains multiple <messagepanel> elements; there is no way to determine which hazard images are associated with specific message panels – nor can it be determined *what* the hazard image represents: <typeofhazard>, <consequence>, or <howtoavoid>.

Simplify the authoring experience

Currently, the fixed order of the child elements within <messagepanel> forces some authors to author content in a different order than it will be rendered and read by consumers. This can be counter intuitive.

Also, often authors must use a value for the @outputclass attribute in order to indicate how a list in the <howtoavoid> element should be rendered. This makes the authoring process harder than it needs to be.

Simplify the work of CSS and stylesheet developers

Many implementations must use transformations, CSS, and stylesheets to swap the order of the <consequence> and <howtoavoid> elements. This can be burdensome to small implementations, and it can easily be avoided by relaxing the content model of <messagepanel>.

Currently, many implementations rely on CSS and style sheets to render the simple list allowed within <howtoavoid> as either an ordered or unordered list. Allowing and within <howtoavoid> will eliminate this extra work.

New terminology

Not applicable

Proposed solution

- Restrict the values of the @type attribute on <hazardstatement> to danger, caution, warning, notice, and -dita-use-conref-target; require the @type attribute.
- Improve the element-reference topic for <hazardstatement> to include definitions of the values for the @type attribute that match those in the ANSI Z535.6 standard. The following definition descriptions are taken verbatim from ANSI Z535.6:

@danger

Indicates a hazardous situation that, if not avoided, will result in death or serious injury. This signal word is to be limited to the most extreme situations.

@warning

Indicates a hazardous situation that, if not avoided, could result in death or serious injury.

@caution

Indicates a hazardous situation that, if not avoided, could result in minor or moderate injury.

@notice

Indicates information considered important, but not hazard-related (e.g. messages relating to property damage).

Note that the DITA spec will not use these definitions verbatim. They will be edited to match our style guidelines.

- Expand the content model of <messagepanel> to enable greater flexibility in the order of the child elements. While <typeofhazard> will remain required and the first child of <messagepanel>, it now can be followed by either of the following:
 - <consequence> (zero or more), <howtoavoid> (one or more)
 - <howtoavoid> (one or more), <consequence> (zero or more)
- Expand the content model of <howtoavoid> to permit and , in addition to <sl>
- Change the specialization base of <messagepanel>, <typeofhazard>, <consequence>, and <howtoavoid> to <div>
- Add <hazardsymbol> to the content models of <messagepanel>, <typeofhazard>, <consequence>, and <howtoavoid>
- Remove <hazardsymbol> from the content model of <hazardstatement>

Benefits

This proposal addresses the following questions:

Who will benefit from this feature?

The following will benefit from this redesign:

- All DITA implementations that use the hazard statement domain
- DITA implementations that do not use the hazard statement domain due to concerns that the hazard statement domain is inadequate, difficult to use, or difficult to render
- Companies whose documentation requires hazard statements but who have not migrated to DITA due to concerns that the hazard statement domain is inadequate, difficult to use, or difficult to render

What is the expected benefit?

A hazard statement domain that is easier to use and easier to render

How many people probably will make use of this feature?

Unknown

How much of a positive impact is expected for the users who will make use of the feature?
Significant

Technical requirements

This proposal involves the following changes:

Refactoring an element

- `<hazardstatement>`
 - Remove several values for the `@type` attribute
 - Make the `@type` attribute required
 - Remove `<hazardsymbol>` from the content model
- `<messagepanel>`
 - Modify content model to enable better flexibility regarding the order of elements
 - Add `<hazardsymbol>` (zero or more) to the content model
 - Change specialization base to `<div>`
- `<typeofhazard>`
 - Add `<hazardsymbol>` (zero or more) to the content model
 - Change specialization base to `<div>`
- `<consequence>`
 - Add `<hazardsymbol>` (zero or more) to the content model
 - Change specialization base to `<div>`
- `<howtoavoid>`
 - Add `` and `` to the content model
 - Add `<hazardsymbol>` (zero or more) to the content model
 - Change specialization base to `<div>`

Processing impact

Not applicable

Overall usability

Improved usability for current and future users.

Backwards compatibility

This proposal addresses the following questions:

Was this change previously announced in an earlier version of DITA?

No

Removing a document type that was shipped in DITA 1.3?

No

Removing a domain that was shipped in DITA 1.3?

No

Removing a domain from a document type shell was shipped in DITA 1.3?

No

Removing or renaming an element that was shipped in DITA 1.3?

No

Removing or renaming an attribute that was shipped in DITA 1.3?

No

Changing the meaning of an element or attribute in a way that would disallow existing usage?

Yes – Removing values for the @type attribute on <hazardstatement> that were allowed in DITA 1.2 and 1.3.

Changing a content model by removing something that was previously allowed, or by requiring something that was not?

Yes – Removing <hazardsymbol> from the content model of <hazardstatement>

Yes – Making the @type attribute required on @hazardstatement

Changing specialization ancestry?

Yes – As follows:

Element	DITA 1.3	DITA 2.0
<consequence>	" + topic/li hazard-d/consequence "	" + topic/div hazard-d/consequence "
<howtoavoid>	" + topic/li hazard-d/howtoavoid "	" + topic/div hazard-d/howtoavoid "
<messagepanel>	" + topic/ul hazard-d/messagepanel "	" + topic/div hazard-d/messagepanel "
<typeofhazard>	" + topic/li hazard-d/typeofhazard "	" + topic/div hazard-d/typeofhazard "

Removing or replacing a processing feature that was defined in DITA 1.3?

No

Are element or attribute groups being renamed or shuffled?

No

Migration plan

This proposal addresses the following questions:

Might any existing documents need to be migrated?

Yes, implementations will need to rework <hazardstatement> elements to relocate <hazardsymbol> elements.

Yes, if existing DITA topics use values of @type on <hazardstatement> that are removed.

Yes, if existing DITA topics contain <hazardstatement> elements that do not set the @type attribute.

Might any existing processors or implementations need to change their expectations?

Authoring tools might need to change the CSS or XSLT that is used to render hazard statements.

Implementations might need to adjust their stylesheets for rendering output.

Might any existing specialization or constraint modules need to be migrated?

(Unlikely) If an implementation has specialized elements from the hazard statement domain, it might need to modify their specialization modules.

If no migration need is anticipated, why not?

Not applicable

Costs

This proposal has a (time and effort) impact on the following groups:

Maintainers of the grammar files

- (DTD) Edits to `hazardstatementDomain.mod` and `hazardstatementDomain.ent`
- (RNG) Edits to `hazardstatementDomain.rng`

Editors of the DITA specification

The changes to the DITA specification will be minor to medium:

- Editorial changes to the `<hazardstatement>` topic to specify values for `@type`
- Updates to the "Specialization hierarchy" sections for the `<messagepanel>`, `<typeofhazard>`, `<consequence>`, and `<howtoavoid>` topics
- Changes to the "Example" sections in the `<hazardstatement>`, `<messagepanel>`, `<typeofhazard>`, `<consequence>`, and `<howtoavoid>` topics
- Updates to the "Usage information" section in the `<messagepanel>` topic, to explain meaning of `<hazardsymbol>` as direct child of `<messagepanel>`

The entire topic collection for the hazard statement domain is overdue for an edit, but that should be handled separately from this proposal. We should consider providing more information about formatting expectations for elements in the hazard statement domain.

No changes to the information architecture or specification terminology will be required.

Vendors of tools

Authoring tools that render the `<hazardstatement>` element will need to modify the CSS or XSLT that renders the element.

Applications that style the `<hazardstatement>` element in output formats will need to modify their stylesheets.

DITA community-at-large

The changes to the hazard statement domain should not add to a perception of DITA complexity. The changes should be simple and intuitive for end users to understand.

Producing migration instructions or tools

Migration instructions will be straight forward.

Examples

This section provides examples of the new markup.

Figure 1: Loosening content model of `<messagepanel>` and `<howtoavoid>`

The following code sample illustrates the following:

- Increased flexibility for the content model of `<messagepanel>`
- Use of an unordered list within the `<howtoavoid>` element

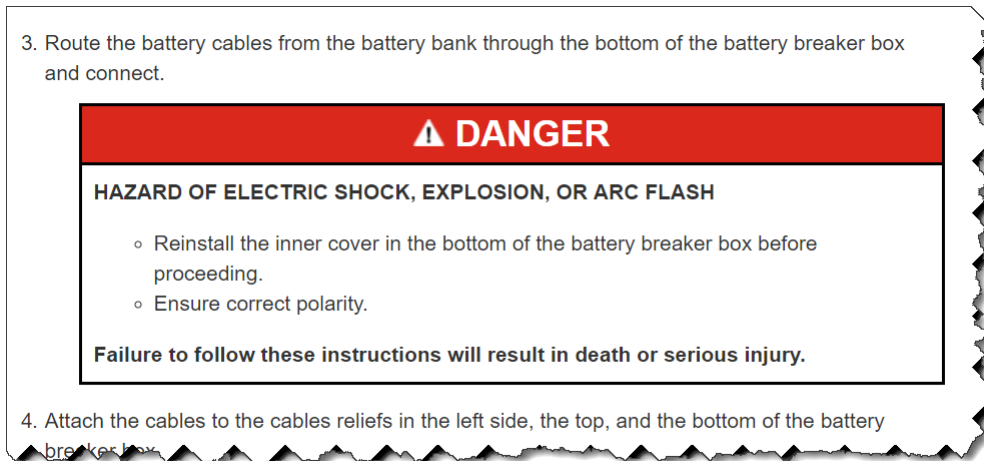
```
<hazardstatement type="danger">
  <messagepanel>
    <typeofhazard>HAZARD OF ELECTRIC SHOCK, EXPLOSION, OR ARC FLASH</typeofhazard>
    <howtoavoid>
      <ul>
        <li>Reinstall the inner cover in the bottom of the battery breaker box
          before proceeding.
        </li>
      </ul>
    </howtoavoid>
  </messagepanel>
</hazardstatement>
```

```

    <li>Ensure correct polarity.</li>
  </ul>
</howtoavoid>
<consequence>Failure to follow these instructions will result in death or serious
  injury.
</consequence>
</messagepanel>
</hazardstatement>

```

This markup might be rendered as the following:



This is an actual screen capture from the Schneider Electric installation instructions for [Gallery XV Battery Breaker Box](#), taken in August 2019.

Note that the company currently must override the HTML transformation to get the desired rendering; the changes that we are implementing for DITA 2.0 will make that override unnecessary.

Figure 2: Adding <hazardsymbol> to <typeofhazard>, <consequence>, and <howtoavoid>

The following code sample illustrates how a company could use a <hazardstatement> element to generate what ANSI Z535.6 calls a "grouped safety message":

```

<hazardstatement type="warning">
  <messagepanel>
    <typeofhazard>
      <hazardsymbol keyref="electric-shock-hazard"/>
      ELECTRIC SHOCK HAZARD</typeofhazard>
      <consequence>The equipment must be grounded. Improper grounding, setup, or usage of
        the system can cause electric shock
      </consequence>
      <howtoavoid>
        <hazardsymbol keyref="ground-power-source"/>
        <ul>
          <li>Turn off and disconnect power at main switch before disconnecting any
            cables or before servicing or installing any equipment.</li>
          <li>Connect only to grounded power sources.</li>
          <li>All electric wiring must be done by a qualified electrician and comply
            with all local codes and regulations.</li>
        </ul>
      </howtoavoid>
    </messagepanel>
    ...
  <messagepanel>
    <typeofhazard>
      <hazardsymbol keyref="burn-hazard"/>
      BURN HAZARD</typeofhazard>
      <consequence>Electric surfaces and fluid that's heated can become very hot during
        operation.</consequence>
    </messagepanel>
  </hazardstatement>

```

```

<howtoavoid>
  To avoid burns:
  <ul>
    <li>Do not touch hot fluid or equipment.</li>
  </ul>
</howtoavoid>
</messagepanel>
</hazardstatement>

```

This markup might be rendered as the following:








 WARNING	
 	<p>ELECTRIC SHOCK HAZARD</p> <p>This equipment must be grounded. Improper grounding, setup, or usage of the system can cause electric shock.</p> <ul style="list-style-type: none"> • Turn off and disconnect power at main switch before disconnecting any cables and before servicing or installing equipment. • Connect only to grounded power source. • All electrical wiring must be done by a qualified electrician and comply with all local codes and regulations.
 	<p>TOXIC FLUID OR FUMES HAZARD</p> <p>Toxic fluids or fumes can cause serious injury or death if splashed in the eyes or on skin, inhaled or swallowed.</p> <ul style="list-style-type: none"> • Read Safety Data Sheet (SDS) for handling instructions and to know the specific hazards of the fluids you are using, including the effects of long-term exposure. • When spraying, servicing equipment, or when in the work area, always keep work area well ventilated and always wear appropriate personal protective equipment. See Personal Protective Equipment warnings in this manual. • Store hazardous fluid in approved containers, and dispose of it according to applicable guidelines.
	<p>PERSONAL PROTECTIVE EQUIPMENT</p> <p>Always wear appropriate personal protective equipment and cover all skin when spraying, servicing equipment, or when in the work area. Protective equipment helps prevent serious injury, including long-term exposure; inhalation of toxic fumes, mists or vapors; allergic reaction; burns; eye injury and hearing loss. This protective equipment includes but is not limited to:</p> <ul style="list-style-type: none"> • A properly fitting respirator, which may include a supplied-air respirator, chemically impermeable gloves, protective clothing and foot coverings as recommended by the fluid manufacturer and local regulatory authority. • Protective eyewear and hearing protection.
	<p>BURN HAZARD</p> <p>Equipment surfaces and fluid that's heated can become very hot during operation. To avoid severe burns:</p> <ul style="list-style-type: none"> • Do not touch hot fluid or equipment.

Figure 3: Adding <hazardsymbol> to <messagepanel>

The following code sample illustrates how <messagepanel> will be able to contain <hazardsymbol>. This will be useful for companies as they migrate from the old to new content models.

```

<hazardstatement type="warning">
  <messagepanel>
    <typeofhazard>GENERAL HAZARDS</typeofhazard>
    <consequence>Overriding or defeating the interlocks will expose personnel to
      hazardous conditions.</consequence>
    <howtoavoid>DO NOT override or defeat the interlocks unless specifically directed to
      do so in the procedures. When directed to override an interlock, follow all
      safety procedures and apply HEI (Lockout/Tagout) procedures as necessary.
    </howtoavoid>
    <hazardsymbol keyref="general-warning"/>
  </messagepanel>
</hazardstatement>

```