A large, stylized graphic in the background features a light blue face with two circular eyes and a pen nib for a mouth. The pen nib is light green and points downwards. The entire graphic is enclosed within a light green circular arc.

# **DITA specialization**

## **Extensibility and standards compliance**

Sarah O'Keefe  
Chief Executive Officer, Scriptorium

Gretyl Kinsey  
Technical Consultant, Scriptorium

October 23, 2017

## The curse of standards

Open standards are great—until you realize that the standard is almost-but-not-quite perfect for your requirements. Then you face ugly choices: do you compromise by forcing your elliptical content into the round standard, or customize the standard to make it just right? Some companies join the standards-making body, and eventually get their requirements added to the standard.

All of these approaches are problematic:

- *Compromise* and use the standard means that you don't get all of the features you need for your content requirements.
- *Customize* the standard means that you break the standard. Instead of having content that conforms to a standard, you now have custom content "based on the standard." Over time, your company's specific implementation of the standard will drift farther and farther from the specification.
- *Change* the standard means hours of committee meetings, and, usually, a lengthy process to get the changes you need.

## DITA specialization: What makes it different

*Specialization*<sup>1</sup> is a technique for modifying the basic DITA structure. When you specialize, you can add and remove elements and attributes. A specialized structure is still valid DITA, which means you can process it through the DITA Open Toolkit and in DITA-compliant editors.<sup>2</sup>

This ability to extend the standard without breaking default processing sets DITA apart from other standards.

### Create new elements

When you create a new element, you must use an existing element as a starting point. The content model of the new element can be the same as the original element, or it can use a subset of the original element's content model. You cannot introduce new elements into the content model unless those elements are specialized from the originally available elements.

*Table 1: Specialized content model must fit inside original content model*

Original content model	Proposed specialization	Valid?
a or b or c	a	Yes
	a then b then c	Yes
	a and d	<b>No</b>
	a and d ( <i>d specialized from a or b or c</i> )	Yes
	b and c	Yes

You can only make the specialized new element available in places where the original element would have been valid.

### Create new attributes

The process of creating new attributes is similar. You base new attributes on existing attributes, and they can only be available in locations where the parent attribute is available.

### Fallback processing

After creating new elements, you modify your output processing to account for those elements. If you do not set up processing for the new elements, the DITA Open Toolkit uses *fallback processing*. That is, it processes your specialized elements using the processing available for the parent element. If you specialize an xyz element from the p element, the Open Toolkit first looks for processing unique to the xyz element. If that is not present, it then looks for p element processing.

<sup>1</sup> <https://docs.oasis-open.org/dita/v1.0/archspec/ditaspecialization.html>

<sup>2</sup> Note that some software applications claim to provide DITA support but only support the out-of-the-box DITA elements. If you intend to specialize, make sure that your tools can handle a specialized content model.

As a result, you can specialize and be confident that your specialization will not break a *DITA-compliant toolchain*. But be aware that some DITA publishing tools do not support fallback processing because they process based on element names only. To support fallback processing, the publishing tool must understand the inheritance relationship between specialized elements and the parent elements.

## Specialization and generalization

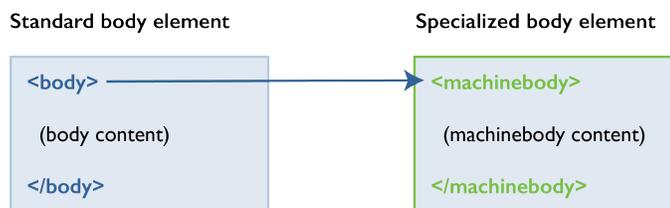
*Generalization* is the process of reverting specialized content back to the default elements. If you specialize xyz from p, then generalization would replace all xyz elements with p elements. If you need to share your specialized content with another organization, you have two choices. You can share your files, along with the specialization itself, so that the other organization can understand and use your specialized content. Or you can generalize your content back to baseline DITA, and share that content. The first option gives you better semantics and is useful for close collaboration, but it requires the receiving organization to configure their systems to support your specialization. The second option lets you send your files without worrying about the receiving organization's DITA configuration.

## Examples

Here are a few examples of DITA specializations.

### Element specialization

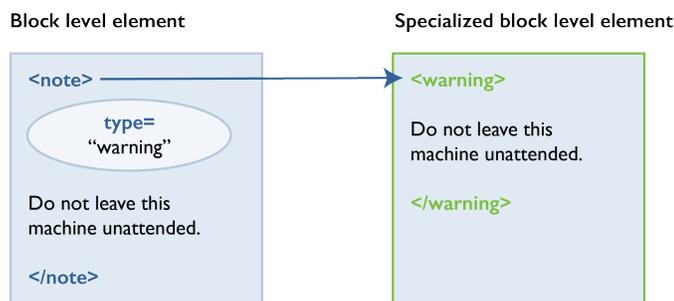
When you specialize elements, you create new elements from existing elements. You can specialize entire topic structures, or you can create new elements. The reference, task, and concept topics in DITA are in fact specializations of the base topic element. The following example shows a specialization of the body element to machinebody.



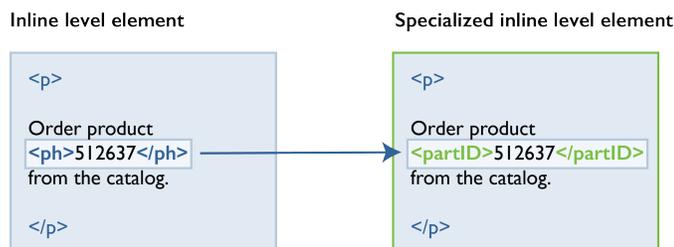
At the block level, a common specialization is adding warning, caution, and other types of notes. By default, you can create a warning as a note with a type attribute:

```
<note type="warning">This is a warning.</note>
```

But if you want to create specific elements for warnings, cautions, and so on, you need to specialize and change the content model.



You can also specialize inline elements. The phrase element (ph) is a general-purpose element. To identify parts of a machine in your content, you could start with a specialization of ph to create a partID element.



Attribute specialization is likely a better choice for this issue. More on that in the next section.

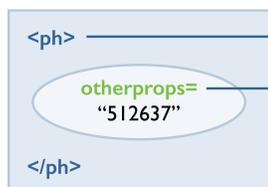
## Attribute specialization

In addition to creating new elements, you can specialize to create new attributes. For the part number example, you might consider setting up the part number as an attribute rather than as text inside the element. The use of the attribute could make it easier to parse the part numbers and then do clever things like reach into parts databases for more information about the part.

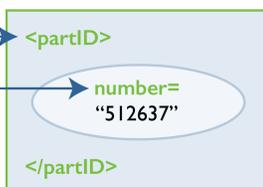
So our typical part reference looks like this:

```
<partID number="512637">steering wheel</partID>
```

Standard DITA attribute



Specialized DITA attribute



## man pages

A man page is command-line help in UNIX-based systems. For example, here is the beginning of the man page for the ls command:

```
LS(1)                                BSD General Commands Manual
LS(1)

NAME
    ls -- list directory contents

SYNOPSIS
    ls [-ABCFGHLOPRSTUW@abcdefghijklmnopqrstuwxl] [file ...]

DESCRIPTION
    For each operand that names a file of a type other than directory, ls
    displays its name as well as any requested, associated information.
    For
    each operand that names a file of type directory, ls displays the names
    of files contained within that directory, as well as any requested,
    associated information.

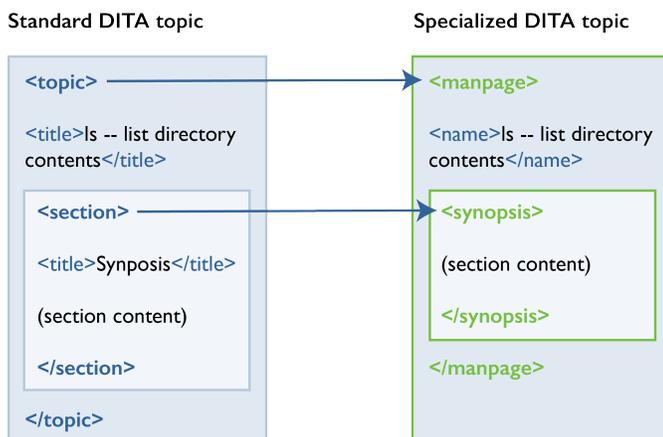
    If no operands are given, the contents of the current directory are
    displayed.  If more than one operand is given, non-directory operands
    are displayed first; directory and non-directory operands are sorted
    separately and in lexicographical order.

    The following options are available:
```

You could create content in regular DITA that results in man page output:

```
<topic>
<title>ls -- list directory contents</title>
<section>
<title>SYNOPSIS</title>
<p>    ls [-ABCFGHLOPRSTUW@abcdefghijklmnopqrstuwxl] [file ...]</p>
</section>
```

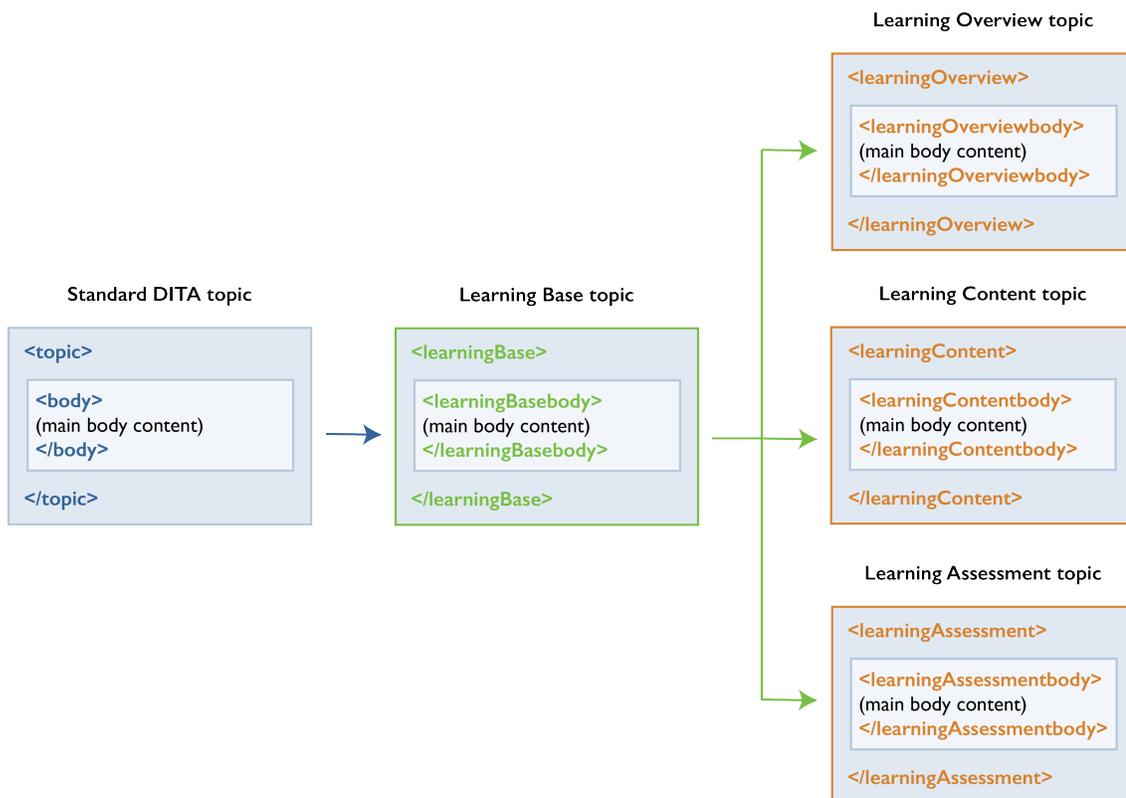
But a specialization is a better approach here. We can specialize section elements to create elements that match each section of the man page, like synopsis, description, and so on.



When you process the DITA content, you can then create man page format as output.

## Learning content

A specialization for Learning and Training is included with DITA 1.3. Here you can see a very complex specialization, where you have new topic types for different types of learning content.



Inside the each of the learning topic types, you will find additional specialized elements. The learning assessment topic, for example, lets you create a variety of questions, including multiple choice, true/false, and matching questions.

```
<learningAssessment id="assessment_k1d_qc2_2t">
  <title>section</title>
  <prolog>
    <author href="http://www.scriptorium.com" scope="external"
format="html">Simon Bate,
    Scriptorium</author>
    <critdates>
      <created date="2015-08-28"/>
    </critdates>
  </prolog>
  <learningAssessmentbody>
    <lcInteraction>
      <lcTrueFalse id="link8">
        <lcQuestion>A &lt;section> element can contain other &lt;section>
elements.</lcQuestion>
        <lcAnswerOptionGroup id="lcAnswerOptionGroup_fnf_zc2_2t">
          <lcAnswerOption>
            <lcAnswerContent>False</lcAnswerContent>
            <lcCorrectResponse/>
          </lcAnswerOption>
          <lcAnswerOption>
            <lcAnswerContent>True</lcAnswerContent>
          </lcAnswerOption>
        </lcAnswerOptionGroup>
        <lcFeedbackIncorrect>DITA does not allow nesting of &lt;section>
elements.
        </lcFeedbackIncorrect>
      </lcTrueFalse>
    </lcInteraction>
  </learningAssessmentbody>
</learningAssessment>
```

## The dark side of specialization: cost

Specialization results in standards-compliant DITA content, but it increases implementation and maintenance costs. Here are just a few examples:

- *Output stylesheets.* Specialized elements, by default, are processed in the same way as the parent element that they are derived from. If you want to handle their formatting differently, you need to make changes in your stylesheets.
- *Authoring tools.* You may want to modify how the authoring tools display your specialized elements to content creators.
- *Localization workflow.* Your localization tools will need additional configuration to account for specialized elements.
- *Content exchange.* If you share content with another organization that doesn't use your specialization, you need to generalize back to the default DITA elements before sending content to them.

Each time you create a specialized element, you have to account for these and other issues. You have to decide whether the increased complexity adds enough value to be worthwhile.

---

## Alternatives

To minimize the amount of specialization, consider these options:

- *The outputclass attribute.* You can use @outputclass to identify elements that require special formatting, such as a pull quote or special positioning instructions for a graphic. Using outputclass doesn't require a modification of the content model, but you have to account for outputclass in your stylesheet transform.
- *The <data> element.* You can use the <data> element for information that doesn't fit into regular DITA elements.
- *Constraints.* DITA constraints provide a lightweight option for eliminating unneeded elements.
- *Lower your standards.* Instead of trying to get an exact fit in DITA for your content, consider using an existing element that is good enough for your purposes.

## Summary

DITA specialization offers a unique way to adapt the DITA standard to your requirements while remaining in compliance with the standard. At Scriptorium, we find that nearly all organizations need custom metadata via attribute specialization to account for unique requirements. Our experience with specialization at the element and topic level is more varied. Some of our customers do little or no element specialization; others create new topic types and many new elements.

Any DITA implementation should begin with a thorough assessment of the organization's content model, specialization requirements, and constraint possibilities.

## About the authors

*Sarah O'Keefe* is the chocoholic founder and CEO of Scriptorium Publishing and a content strategy consultant.

Scriptorium offers content strategy services to help customers solve business problems and achieve strategic goals. Since founding the organization in 1997, Sarah has worked with numerous organizations to assess content issues, develop solutions, and implement new technologies that unlock content value. She is the coauthor of [Content Strategy 101: Transform Technical Content into a Business Asset](#).

An experienced public speaker, Sarah is in demand at conferences worldwide. Sarah is fluent in German, reads voraciously, and enjoys swimming, kayaking, and other water sports along with knitting and college basketball. She has strong aversions to raw tomatoes, eggplant, and checked baggage.

You can follow Sarah on Twitter at [@sarahokeefe](#).

*Gretyl Kinsey*, Technical Consultant, has been with Scriptorium since 2011. She has led complex content strategy projects for customers in education, life sciences, and high tech.

She is an experienced interviewer and researcher who also develops business cases and writes recommendation reports. She has worked on every stage of implementation from analysis to content modeling to follow-on support. With an interest in tech comm tools and technologies and a passion for learning, Gretyl frequently writes course content and creates demo videos for [LearningDITA.com](#). She has given presentations on DITA, content strategy, and other industry topics.

Gretyl graduated from UNC-Chapel Hill with a BA from the School of Journalism and Mass Communication, where she specialized in multimedia production. Outside of work, she enjoys writing and performing music, drawing comics, cosplaying, swimming, and trying out dessert recipes.

You can follow Gretyl on Twitter at [@gretylkinsey](#).

