6

THE LwDITA TOPIC COMPONENTS

In the previous chapter I analyzed the LwDITA document type of map, which works as a collection and organization mechanism for content units. Those content units are represented in the document type of topic. Although in Chapter 3 I introduced the main content components available in LwDITA, here I focus on how to represent each of those topic components in the authoring formats of XDITA, HDITA, and MDITA. Keep in mind that LwDITA is a work in progress and some elements in this chapter might change between the publication of this book and the actual release of the Lightweight DITA standard.

The examples included in this chapter continue the scenario of Chef Pedro's content requirements for the promotional *Sensei Sushi* brochure. The sample files are available for download from the *Creating Intelligent Content with Lightweight DITA* GitHub repository (https://github.com/carlosevia/lwdita-book).

I have organized this chapter according to the following classification of LwDITA content components:

- Basic topic components
- Table components
- Highlight components
- Metadata components
- Multimedia components.

In the last section of this chapter I highlight one of the most promising LwDITA features: the ability to create coherent and consistent user deliverables from cross-format sources developed by diverse groups of content creators.

Basic topic components

Topic

XDITA

The essential unit of content in LwDITA is a topic, which is represented in XDITA with the XML tag **<topic>** . The generic topic is the only document type available for creating content in the initial release of the LwDITA specification. For Chef Pedro's content needs, a very simple topic would look as the example in Figure 6.1.

You can type the code from Figure 6.1 in any text editor, but working in a LwDITA-aware application like Oxygen XML Editor brings benefits like auto-completion of tags and templates, which can reduce spelling errors and typos. The first line of code in the example is the XML declaration, which announces to the computer processor that this is an XML file and should be treated as such. The second line indicates that this is not just any XML file, and that it must be validated against the document type definition (DTD) of *lw-topic.dtd*, which is a grammar file that includes the XDITA syntax rules and was created by the Lightweight DITA subcommittee at OASIS. Another benefit of using an LwDITA-aware app to create content is that it will probably include the grammar files and provide validation as you type. This second line of code enforces the XDITA authoring format. Without it, the topic would just be a regular XML file (like the first marinara sauce recipe from Chapter 1), and would need its own stylesheets and schemas to produce deliverables for end users.

In XDITA, the attribute for a unique identifier at the topic level (the **id** portion of the **<topic id="franchise-intro">** line) is required. Without it, the topic will not validate as standards-based XDITA content. I am a fan of using the same text for the topic identifier attribute and the file name: e.g., the **@id** of this topic is "franchise-intro," and I would save it as *franchise-intro.dita* (XDITA topics can be saved with the file extensions *.xml* or *.dita*). The next line of code in the

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE topic PUBLIC "-//OASIS//DTD LIGHTWEIGHT DITA Topic//EN" "lw-topic.dtd">
<topic id="franchise-intro">
  <!-- Some content will go here -->
</topic>
```

FIGURE 6.1 Minimal XDITA topic. The opening and closing XML tags for topic (in bold font) create an environment for content. In this example, the only content is a placeholder XML comment (marked by the characters <!-- and -->).

FIGURE 6.2 Validation error reported by Oxygen XML Editor. Oxygen validates the XDITA topics as the authors develop them. In this case, the "red squiggly of doom" error tells the authors that the topic is not valid and must be fixed.

example is an XML comment (enclosed within the characters <**!**-- and -->) that I am using as a temporary placeholder for actual content.

At this point, an author new to structured content should resist the urge to preview the topic in a deliverable. The "format free" characteristic of intelligent content (Rockley et al., 2015) depends on the separation of content and presentation; after all, this sample topic can be transformed to different deliverables with a variety of formats and stylesheets. The concept of "what you see is what you get" (WYSIWYG) from many web developing editors does not translate to topic-based authoring in XML because a topic is just a component in a larger collection of chunks that need to be processed to create deliverables. Furthermore, this sample topic stub is not yet valid XDITA. A LwDITA-aware tool like Oxygen XML would inform the author that the topic is incomplete. Some of my students call the Oxygen validation error notifier the "red squiggly of doom" (Figure 6.2).

Before I fix the XDITA topic to remove the error messages, I will present the mapping of the XDITA element of topic to the HDITA and MDITA authoring formats.

HDITA

The HTML5 semantic tag of **<article>** provides a content environment similar to the **<topic>** component from XDITA. In HDITA, **<article>** has a required **@id**, that could have the same value as the file's name. In the following example, the **@id** attribute on **<article>** is "*franchise-intro*", and the file can be saved as *franchise-intro.html* (Figure 6.3). The first line of code in the file is establishing that the HDITA topic should be validated as HTML.

You can create this HDITA topic in any text editor, a more advanced HTML editor, or a LwDITA-aware tool like Oxygen XML. The validation process of an HTML5 file as an HDITA topic will be application-specific, as HTML does not have a customizable document type definition or schema syntax.

```
<!DOCTYPE html>
<article id="franchise-intro">
  <!-- Some content will go here -->
</article>
```

FIGURE 6.3 Minimal HDITA topic. This topic needs the HTML5 element of **<article>** to create an environment for content. This file, however, is not valid HTML5. Just like the XDITA example in Figure 6.1, the HDITA topic requires a **<title>** element, which I will add in the following section.

MDITA

In MDITA core profile, any structure saved as Markdown format can work as a basic topic. Figure 6.4 reproduces the essential topic with a placeholder comment for content. In Markdown, a title is not required, so this one-line example can work as a valid basic (but useless for practical reasons) topic, even if its content is just a commented annotation. You can create this file in any text editor or a LwDITA-aware tool, and save it as *franchise-intro.md*.

In XDITA and MDITA, a topic requires an **@id**, which is provided as an attribute of the components **<topic>** and **<article>**, respectively. In MDITA, the required **@id** should be automatically generated when the topic is saved, through a "slug"-generation process. Therefore, the **@id** for *franchise-intro.md*, if needed for future topic linking and reuse, will be *franchise-intro*. In MDITA extended profile, authors can specify an **@id** for the topic that overrides the "slugify" process. The assigned **@id** should be included in an optional YAML[1] front matter header. In *Lightweight DITA: An Introduction*, we wrote the following about the optional YAML header for MDITA topics:

> This YAML header can supply a direct value for the @id attribute that is required on the root element of a DITA topic; it can also include prolog metadata about who authored the DITA topic. If included in a topic, the YAML front matter header must be the first thing in the MDITA file and must be set between triple-dashed lines.
>
> (Evia et al., 2018, p.19)

Figure 6.5 shows the *franchise.intro.md* topic with an assigned **@id** of *sensei-intro* that would override the **@id** generated with the file's name.

```
<!-- Some content will go here -->
```

FIGURE 6.4 Minimal MDITA topic. This topic does not need a body element. It does not need a title either, and it could only include a placeholder comment for future content.

```
---
id: sensei-intro
---

<!-- Some content will go here -->
```

FIGURE 6.5 Sample MDITA extended profile topic with an assigned **@id** in a YAML header. The assigned **@id** will be attached to the topic even if the authors rename the source file (with the default "slugify" process of MDITA core profile, renaming a file will change its automatically-generated **@id**).

The validation of Markdown files as MDITA topics will also be application-specific. The Lightweight DITA subcommittee at OASIS will release recommendations and standards for that validation, but will not produce tools or files to enforce it, leaving the implementation to interested vendors and developers.

Title

XDITA

The "red squiggly of doom" from Figure 6.2 indicates that, according to the XDITA grammar rules, a topic cannot be empty. A valid XDITA topic needs, at least, a title, which is represented by the XML element **<title>**. The new version of the basic XDITA topic should look as Figure 6.6.

The title component in this revised example includes the XML tags that represent the **<title>** element and the actual content for human users that appears inside the tags. The content embraced by the XML tags requires proper capitalization and correct spelling according to any style guide developed or adopted by Chef Pedro and his team. With this new title component, the topic stub now can clear validation warnings in a LwDITA-aware tool like Oxygen XML Editor, and it can be seen in *author* mode, which *hides* the XML code and allows editing in an environment similar to a WYSIWYG editor (Figure 6.7).

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE topic PUBLIC "-//OASIS//DTD LIGHTWEIGHT DITA Topic//EN" "lw-topic.dtd">
<topic id="franchise-intro">
  <title>An innovative, attractive, and out of the ordinary concept</title>
</topic>
```

FIGURE 6.6 Minimal XDITA topic with a title component. The file will now clear validation in LwDITA-aware apps and also shows actual content for human readers.

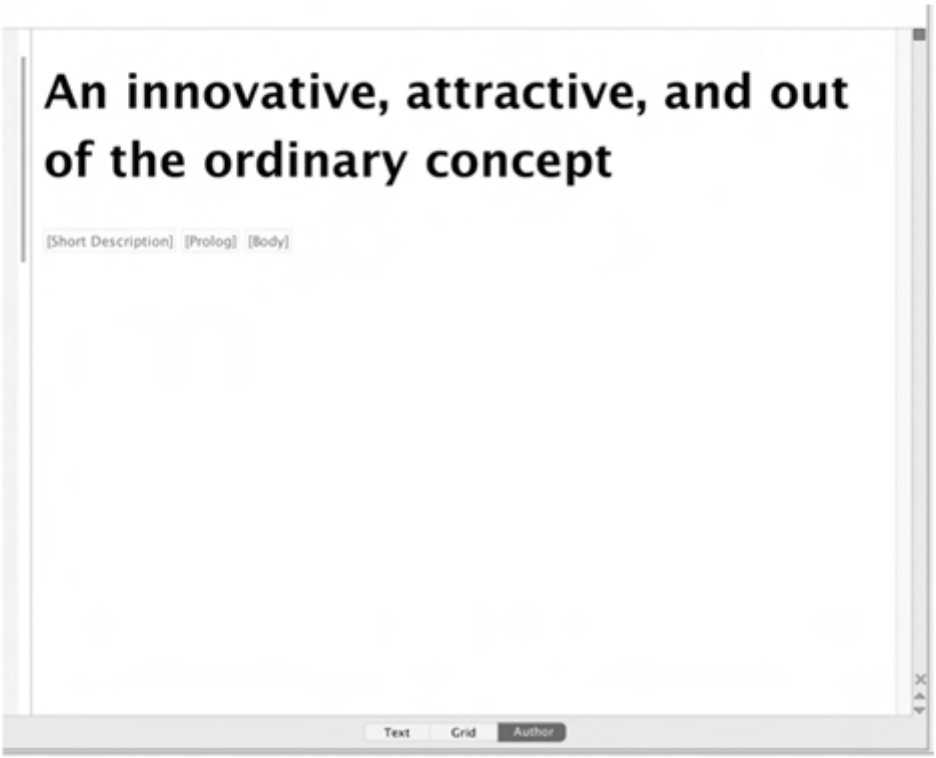


FIGURE 6.7 Author mode in Oxygen XML Editor. This mode provides a quick preview of a topic, similar to a WYSIWYG editor. With its title element, the sample XDITA topic now cleared Oxygen error messages.

HDITA

According to the W3C recommendation for HTML5, the "document's title is often different from its first heading, since the first heading does not have to stand alone when taken out of context." (W3C, 2017). Therefore, whereas a valid HTML5 file requires a **<title>** element, this component won't perform as the topic's first heading. As a result, HDITA topics need both a title (**<title>**) element and a heading 1 (**<h1>**) element to signal the document's first heading. Redundantly, both elements should contain the same information in order to comply with syntax rules from HTML5 and LwDITA. LwDITA-aware tools could automate the process of generating a **<h1>** from a **<title>** and reduce repetition for human authors. Figure 6.8 shows the franchise introduction topic with a title and a heading 1.

MDITA

The level-one heading that provides a title for an MDITA topic can be expressed, according to the GitHub Flavored Markdown (GFM) spec, both with an *atx*

```
<!DOCTYPE html>
<title>An innovative, attractive, and out of the ordinary concept</title>
<article id="franchise-intro">
  <h1>An innovative, attractive, and out of the ordinary concept</h1>
</article>
```

FIGURE 6.8 Sample HDITA topic with the required title component. In order to be valid as an HDITA topic and HTML5 file at the same time, the topic requires a title and a heading 1, which should hold the same content.

("the true structured text format")[2] or *setext* (Structure Enhanced Text)[3] heading. Figure 6.9 shows the franchise introductory topic in MDITA core profile with an *atx* heading for its title.

In Figure 6.10, an MDITA extended profile with an assigned **@id** of "*franchise-intro*" uses a *setext* heading for its title component.

Short description

XDITA

An optional topic component that I have addressed in previous chapters is the multipurpose short description (**<shortdesc>** in XDITA), which performs a staging move (see Table 2 in Chapter 2) before a topic goes into detail about a specific subject. Figure 6.11 shows a short description in the introductory XDITA topic for the *Sensei Sushi* franchise brochure.

```
# An innovative, attractive, and out of the ordinary concept

<!-- Some content will go here -->
```

FIGURE 6.9 Sample MDITA core profile topic with a title component. The title is marked by a heading 1 in *atx* syntax.

```
---
id: franchise-intro
---

An innovative, attractive, and out of the ordinary concept
==========================================================

<!-- Some content will go here -->
```

FIGURE 6.10 Sample MDITA extended profile topic with a title component. The title is marked by a heading 1 in *setext* syntax.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE topic PUBLIC "-//OASIS//DTD LIGHTWEIGHT DITA Topic//EN" "lw-topic.dtd">
<topic id="franchise-intro">
  <title>An innovative, attractive, and out of the ordinary concept</title>
  <shortdesc>Are you interested in investing with us? Welcome to our franchise
information package.</shortdesc>
</topic>
```

FIGURE 6.11 XDITA topic with a short description component. The opening and closing tags of **<shortdesc>** hold content that the authors can use as a staging rhetorical move in the topic.

HDITA and MDITA

In HDITA and MDITA topics, the optional short description is implied in the first paragraph of content. Figure 6.12 shows the introductory topic with a short description paragraph in HDITA.

Figure 6.13 presents the same topic in MDITA extended profile, which includes the YAML header with an assigned **@id**, an *atx* heading for the title, and the short description implied in the first paragraph.

```
<!DOCTYPE html>
<title>An innovative, attractive, and out of the ordinary concept</title>
<article id="franchise-intro">
  <h1>An innovative, attractive, and out of the ordinary concept</h1>
  <p>Are you interested in investing with us? Welcome to our franchise
information package.</p>
</article>
```

FIGURE 6.12 HDITA topic with a short description component. The short description defaults to the first paragraph in an HTML file. If authors need more structure and their workflows require identifying a first paragraph as a short description, they should use the XDITA syntax instead.

```
---
id: franchise-intro
---

# An innovative, attractive, and out of the ordinary concept

Are you interested in investing with us? Welcome to our franchise information
package.
```

FIGURE 6.13 MDITA topic with a short description component. The short description is implied in the first paragraph of a Markdown file.

Body and Paragraph

The previous examples did not have much content but are all valid LwDITA topics. Maybe Chef Pedro only needs a brief blurb for this information. If so, the topics could end here and join a content collection to produce deliverables for end users. Nevertheless, despite the potential uses of such a content blurb, the component that contains the majority of conventions for content is the body – represented in XDITA and HDITA by the tag **<body>**. The following section introduces additional topic components that can be included inside the body environment.

Inside the body component, the most common unit for small pieces of content is a paragraph (**<p>**). The example in Figure 6.14 adds a body component, with a paragraph, to the franchise introduction topic in XDITA:

The HDITA version of the paragraph, seen in Figure 6.15, looks pretty similar to its XDITA counterpart.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE topic PUBLIC "-//OASIS//DTD LIGHTWEIGHT DITA Topic//EN" "lw-topic.dtd">
<topic id="franchise-intro">
  <title>An innovative, attractive, and out of the ordinary concept</title>
  <shortdesc>Are you interested in investing with us? Welcome to our franchise
  information package.</shortdesc>
  <body>
    <p>We offer more than 30 exclusive creations of original rolls, from the
California roll to sushi with BBQ chicken or grilled steak.</p>
  </body>
</topic>
```

FIGURE 6.14 XDITA topic with a body component. In XDITA, the body environment can hold several topic components. The primary component inside a body is a paragraph, which should be represented by opening and closing **<p>** tags.

```
<!DOCTYPE html>
<title>An innovative, attractive, and out of the ordinary concept</title>
<body>
  <article id="franchise-intro">
    <h1>An innovative, attractive, and out of the ordinary concept</h1>
    <p>Are you interested in investing with us? Welcome to our franchise
information package.</p>
    <p>We offer more than 30 exclusive creations of original rolls, from the
California roll to sushi with BBQ chicken or grilled steak.</p>
  </article>
</body>
```

FIGURE 6.15 HDITA topic with paragraph components. In HDITA, a paragraph is also represented by opening and closing **<p>** tags, and paragraphs should be inside a body environment.

The GFM spec defines paragraph as a "sequence of non-blank lines" (GitHub Flavored Markdown Spec, 2017). Thus, an MDITA paragraph does not require any tags, as seen in Figure 6.16.

Lists

The body of a LwDITA topic can also contain topical conventions for lists. LwDITA includes components for representing the following types of lists:

- Definition lists
- Ordered lists
- Unordered lists.

In XDITA, the definition list (**<dl>**) is a content environment that includes a definition list entry (**<dlentry>**), a definition term (**<dt>**), and a definition description (**<dd>**). In this example (Figure 6.17), Chef Pedro and his team use a definition list to describe the terms that apply to franchise owners of *Sensei Sushi*.

The previous example shows the "minimize mixed content" rule in action. Whereas in DITA 1.3 a definition description can contain character data or additional XML elements, in XDITA it must include a paragraph (**<p>**) housing any text. Without a doubt, the definition list is one of the most verbose XDITA environments. The tags required for declaring the list, its entries, terms, and descriptions do not look extremely lightweight. In cases like this one, an LwDITA-aware editor can help with tag autocompletion and authorial guidance.

The W3C recommendation for HTML explains the **<dl>** element as a "description list of zero or more term-description groups. Each term-description group consists of one or more terms (represented by <dt> elements), and one or more descriptions (represented by <dd> elements)" (W3C, 2017). Thus, in HDITA the definition list environment looks as shown in Figure 6.18. Note that the "minimize elements that allow mixed content" also applies to HDITA topics.

```
---
id: franchise-intro
---

# An innovative, attractive, and out of the ordinary concept

Are you interested in investing with us? Welcome to our franchise information
package.

We offer more than 30 exclusive creations of original rolls, from the California
roll to sushi with BBQ chicken or grilled steak.
```

FIGURE 6.16 MDITA topic with paragraph components. In MDITA, a paragraph is a sequence of non-blank lines and should be visually separated from other content components by a return or enter.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE topic PUBLIC "-//OASIS//DTD LIGHTWEIGHT DITA Topic//EN" "lw-topic.dtd">
<topic id="franchise-terms">
  <title>Profit, fun, and flavor under the same brand</title>
  <body>
    <dl>
      <dlentry>
        <dt>Initial investment:</dt>
        <dd><p>$700 (includes initial franchise fee)</p></dd>
      </dlentry>
      <dlentry>
        <dt>Franchise fee:</dt>
        <dd><p>$200</p></dd>
      </dlentry>
    </dl>
  </body>
</topic>
```

FIGURE 6.17 Definition list environment in XDITA. The component includes one or more definition list entries, each with a definition term and a definition description.

There is no direct equivalent for a definition list in MDITA core profile, although some Markdown flavors do have a way to represent this component. If an author needs to include a definition list in MDITA, the recommendation is to take advantage of the MDITA extended profile and express the definition list with a raw code block of HDITA syntax. The following example (Figure 6.19) shows the definition list in MDITA extended profile with an HDITA code snippet.

```
<!DOCTYPE html>
<title>Profit, fun, and flavor under the same brand</title>
<body>
  <article id="franchise-terms">
    <h1>Profit, fun, and flavor under the same brand</h1>
    <dl>
      <dt>Initial investment:</dt>
      <dd>
        <p>$700 (includes initial franchise fee)</p>
      </dd>
      <dt>Franchise fee:</dt>
      <dd>
        <p>$200</p>
      </dd>
    </dl>
  </article>
</body>
```

FIGURE 6.18 Definition list environment in HDITA. Unlike its counterpart in XDITA, this definition list does not need the holding definition entry environment to contain pairs of definition term and definition description.

```
---
id: franchise-terms
---

# Profit, fun, and flavor under the same brand

<dl>
     <dt>Initial investment:</dt>
     <dd>
       <p>$700 (includes initial franchise fee)</p>
     </dd>
     <dt>Franchise fee:</dt>
     <dd>
       <p>$200</p>
     </dd>
</dl>
```

FIGURE 6.19 Definition list environment in an MDITA extended profile topic. The definition list element does not exist in GitHub Flavored Markdown syntax. Thus, if authors need a definition list in MDITA, they should express it with a raw HDITA code block.

In XDITA and HDITA, the ordered list (**<ol>**) is a much simpler environment than the definition list, and it only requires one or more instances of a list item (**<li>**). Figure 6.20 includes an ordered list in XDITA giving steps that a potential investor should go through to obtain the franchise license.

The "minimize mixed content" rule in LwDITA wraps content from the list items in paragraphs. Authors coming to XDITA from HTML5 might find that unusual, but this can simplify content reuse in more advanced situations.

The ordered list would look as follows (Figure 6.21) in HDITA:

The unordered list (**<ul>**) is a very similar component. Figure 6.22 shows an unordered, or bulleted, list with features of the *Sensei Sushi* franchise in an XDITA topic.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE topic PUBLIC "-//OASIS//DTD LIGHTWEIGHT DITA Topic//EN" "lw-topic.dtd">
<topic id="franchise-plan">
  <title>Make a plan! Start your future today!</title>
  <body>
    <ol>
      <li><p>Contact one of our franchise advisors</p></li>
      <li><p>Pick a location for your restaurant</p></li>
      <li><p>Follow our franchise guide</p></li>
    </ol>
  </body>
</topic>
```

FIGURE 6.20 Ordered list environment in XDITA. The "minimize mixed content" rule forces authors to wrap the content of each list item in paragraphs.

```
<!DOCTYPE html>
<title>Make a plan! Start your future today!</title>
<body>
  <article id="franchise-plan">
    <h1>Make a plan! Start your future today!</h1>
    <ol>
      <li>
        <p>Contact one of our franchise advisors</p>
      </li>
      <li>
        <p>Pick a location for your restaurant</p>
      </li>
      <li>
        <p>Follow our franchise guide</p>
      </li>
    </ol>
  </article>
</body>
```

FIGURE 6.21 Ordered list environment in HDITA. The component follows the same structure as its counterpart in XDITA.

Figure 6.23 shows the same topic in HDITA syntax.

To express ordered and unordered lists in MDITA, you should follow the recommendations from the GFM spec, which specify that "a list is an ordered list if its constituent list items begin with ordered list markers, and a bullet list if its constituent list items begin with bullet list markers" (GitHub Flavored Markdown Spec, 2017). The spec adds that "an ordered list marker is a sequence of 1–9 arabic (*sic*) digits (0–9), followed by either a . character or a) character." (GitHub Flavored Markdown Spec, 2017) Figure 6.24 shows the list for potential investors in MDITA syntax.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE topic PUBLIC "-//OASIS//DTD LIGHTWEIGHT DITA Topic//EN" "lw-topic.dtd">
<topic id="franchise-offer">
  <title>What we offer</title>
  <body>
    <ul>
      <li><p>"Know-how" license</p></li>
      <li><p>Warranty of exclusive territory</p></li>
      <li><p>Initial training</p></li>
      <li><p>Support through online, email, and telephone channels</p></li>
    </ul>
  </body>
</topic>
```

FIGURE 6.22 Unordered list environment in XDITA. Each list item will create, when the topic is processed, a bulleted entry.

```
<!DOCTYPE html>
<title>What we offer</title>
<body>
  <article id="franchise-offer">
    <h1>What we offer</h1>
    <ul>
      <li>
        <p>"Know-how" license</p>
      </li>
      <li>
        <p>Warranty of exclusive territory</p>
      </li>
      <li>
        <p>Initial training</p>
      </li>
      <li>
        <p>Support through online, email, and telephone channels</p>
      </li>
    </ul>
  </article>
</body>
```

FIGURE 6.23 Unordered list environment in HDITA. The component follows the same structure as its counterpart in XDITA.

Unordered list markers can be a -, +, or * character, as shown in Figure 6.25 of an MDITA topic with a list of what the *Sensei Sushi* franchise package offers.

Image and Figure

XDITA

In XDITA, non-textual topical conventions start with an image (**<image>**). The image component is always treated as an inline element: its appearance will not begin on a new line in publications generated from topics that contain it. The

```
---
id: franchise-plan
---

# Make a plan! Start your future today!

1.  Contact one of our franchise advisors
2.  Pick a location for your restaurant
3.  Follow our franchise guide
```

FIGURE 6.24 Ordered list environment in MDITA. When the topic is processed, the list items will be automatically numbered in a logical sequence established by the number assigned to the first list item.

```
---
id: franchise-offer
---

# What we offer

-   "Know-how" license
-   Warranty of exclusive territory
-   Initial training
-   Support through online, email, and telephone channels
```

FIGURE 6.25 Unordered list environment in MDITA. The GFM spec allows the use of the characters -, +, or * as unordered list item markers.

figure (<**fig**>) environment provides a wrap for the image. A figure is treated as a block element in XDITA. Figure 6.26 includes both an inline image and a block figure.

Notice that the images, both in inline and block mode, can include a component for alternate text (<**alt**>), which provides a behind-the-scenes textual description for human users and machine processors that cannot perceive images or require additional information. The figure environment includes a description (<**desc**>) that functions as a caption or label for the image. To put the previous example in context, the *author* mode in Oxygen XML can provide a simple "preview" of the topic and its visual elements (Figure 6.27).

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE topic PUBLIC "-//OASIS//DTD LIGHTWEIGHT DITA Topic//EN" "lw-topic.dtd">
<topic id="franchise-intro">
  <title>An innovative, attractive, and out of the ordinary concept</title>
  <shortdesc>Are you interested in investing with us? Welcome to our franchise
information package.</shortdesc>
  <body>
    <p>We offer <image href="images/plus-sign.jpg"><alt>Icon for a plus sign
</alt></image>than 30 exclusive creations for original rolls, from the
California roll to sushi with BBQ chicken or grilled steak.</p>
    <fig>
      <desc>This is love!</desc>
      <image href="images/sushi-love.jpg">
        <alt>Sensei Sushi logo</alt>
      </image>
    </fig>
  </body>
</topic>
```

FIGURE 6.26 Image and figure components in an XDITA topic. Both contain alternate text for human or machine readers that cannot process a visual element.

FIGURE 6.27 Preview of the XDITA topic with image and figure environments in Oxygen XML *author* mode. The image is in line with its containing paragraph, whereas the figure is its own block environment.

HDITA

In HDITA, the inline image is represented with the HTML tag **<img>**, which requires the attribute of **@src** to specify the source file for the image. The block image should be included in the **<figure>** environment, and it can include a title (**<title>**) element to provide a description. The **@alt** attribute works just like it does in XDITA syntax. The HDITA topic in Figure 6.28 includes the inline image for the plus sign and the block figure with the *Sensei Sushi* logo.

You can save the HDITA topic from the previous example as *franchise-intro.html* and open in it a web browser to see the difference between an inline and block image. Figure 6.29 shows a browser view of the HDITA topic.

MDITA

In MDITA, the syntax to represent an inline image includes the label for alternate text and the link to the image file in the following structure:

```
<!DOCTYPE html>
<title>An innovative, attractive, and out of the ordinary concept</title>
<body>
  <article id="franchise-intro">
    <h1>An innovative, attractive, and out of the ordinary concept</h1>
    <p>Are you interested in investing with us? Welcome to our franchise
information package.</p>
    <p>We offer <img src="images/plus-sign.jpg" alt="Icon for a plus sign" />
than 30 exclusive creations for original rolls, from the California roll to
sushi with BBQ chicken or grilled steak.</p>
    <figure>
      <figcaption>This is love!</figcaption>
      <img src="images/sushi-love.jpg" alt="Sensei Sushi logo" />
    </figure>
  </article>
</body>
```

FIGURE 6.28 Image and figure components in an HDITA topic. Both need the HTML5 tag of **<img>** with the attribute of **@src** for the source file of an image.

```
![Alternate text](path-to-image-file.jpg)
```

The image can become a block element if it is treated as a paragraph (i.e., it is separated from other elements with a return or enter), and it can include an optional title or description in the following structure:

```
![Alternate text](path-to-image-file.jpg "Optional title")
```

Figure 6.30 shows the franchise introductory topic in MDITA syntax with the inline image for the plus sign and the block figure, with a title, for the *Sensei Sushi* logo.

FIGURE 6.29 Preview of the HDITA topic with image and figure environments in a web browser. The image is in line with its containing paragraph, whereas the figure is its own block environment.

```
---
id: franchise-intro
---

# An innovative, attractive, and out of the ordinary concept

Are you interested in investing with us? Welcome to our franchise
information package.

We offer ![Icon for a plus sign](images/plus-sign.jpg) than 30
exclusive creations for original rolls, from the California roll to
sushi with BBQ chicken or grilled steak.

![Sensei Sushi logo](images/sushi-love.jpg "This is love!")
```

FIGURE 6.30 Image and figure components in an MDITA topic. The exclamation point before the link text notifies processors that an image element is included in the topic.

Cross-reference

XDITA

A cross-reference (**<xref>**) component in XDITA allows the creation of links to internal and external resources. A topic can include cross-references to other topics in the same collection or to external sources like a PDF file or a website. In Figure 6.31, the topic with the bulleted list of franchise features now includes a hyper reference (**@href**) to an external website hosting the *Sensei Sushi* knowledge base for technical support.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE topic PUBLIC "-//OASIS//DTD LIGHTWEIGHT DITA Topic//EN" "lw-topic.dtd">

<topic id="franchise-offer">
  <title>What we offer</title>
  <body>
    <ul>
      <li><p>"Know-how" license</p></li>
      <li><p>Warranty of territory exclusivity</p></li>
      <li><p>Initial training</p></li>
      <li><p>Support through online, email, and telephone channels</p></li>
      <li><p>Access to our <xref href="http://senseisushico.com/kb"
 scope="external" format="html">knowledge base</xref></p></li>
    </ul>
  </body>
</topic>
```

FIGURE 6.31 XDITA topic with a cross-reference to an external website. The **<xref>** tags embrace the phrase "knowledge base," which will become active and clickable on deliverables produced from this topic.

The attributes for **@scope** and **@format** inside the cross-reference are optional. The scope attribute, if used, must have a value of "external", "local", or "peer".

HDITA

In HTML5, a cross-reference is represented with a combination of the anchor (**<a>**) element and the hyper reference (**@href**) attribute. This interactive content structure can also include attributes for relationship (**@rel**) and type (**@type**) that correspond to the DITA/XDITA attributes of scope (**@scope**) and format (**@format**), respectively. Figure 6.32 features an HDITA topic with the cross-reference component linking to the external HTML files that contain the *Sensei Sushi* knowledge base.

MDITA

In MDITA, a cross-reference is represented with a Markdown link structure that includes a link text and link destination in the following format:

```
[link text](link destination).
```

Figure 6.33 shows an MDITA version of the franchise offer topic with the cross-reference link to the external *Sensei Sushi* knowledge base.

```
<!DOCTYPE html>
<title>What we offer</title>
<body>
  <article id="franchise-offer">
    <h1>What we offer</h1>
    <ul>
      <li>
        <p>"Know-how" license</p>
      </li>
      <li>
        <p>Warranty of territory exclusivity</p>
      </li>
      <li>
        <p>Initial training</p>
      </li>
      <li>
        <p>Support through online, email, and telephone channels</p>
      </li>
      <li>
        <p>Access to our <a href="http://senseisushico.com/kb" rel="external"
type="text/html">knowledge base</a></p>
      </li>
    </ul>
  </article>
</body>
```

FIGURE 6.32 HDITA topic with a cross-reference to an external website. Compared to the XDITA example, the **<a>** tags replace the **<xref>** to embrace the phrase "knowledge base."

```
---
id: franchise-offer
---

# What we offer

-   "Know-how" license

-   Warranty of territory exclusivity

-   Initial training

-   Support through online, email, and telephone channels

-   Access to our [knowledge base](http://senseisushico.com/kb)
```

FIGURE 6.33 MDITA topic with a cross-reference to an external website. The GFM structure of link text + link destination does not require additional attributes for scope or type of link.

Footnote

XDITA

The cross-reference component also enables the inclusion of footnotes (**<fn>**) in XDITA topics. In Figure 6.34, the list of franchise terms includes a cross-reference to a footnote providing information about the initial fee paid by franchise owners.

The footnote component is composed of two elements: the cross-reference that calls it, and the actual footnote content. In the previous example, the cross-reference is calling a footnote with the **@id** value of "*initial-fee*" located in the topic with the **@id** value of "*franchise-terms*". The actual footnote contains a paragraph with text.

HDITA and MDITA

A cross-reference also provides the foundation for footnotes in HDITA. The HTML5 division element (** **) with the custom data attribute of **@data-class="fn"** creates a section for the footnote content. The footnote division should have a unique **@id**. In the body of the text, where the footnote is called from, an internal cross-reference in the form of **<a href="#footnote-id">** should embrace the text you want to make clickable for the footnote. In Figure 6.35, the clickable text is a number 1 with the superscript (<sup>) format. I will talk more about superscript text in the section about highlighting components.

In MDITA extended profile, a footnote can be represented with an HDITA code block following the structure from Figure 6.35. There is no equivalent for the footnote component in MDITA core profile.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE topic PUBLIC "-//OASIS//DTD LIGHTWEIGHT DITA Topic//EN" "lw-topic.dtd">
<topic id="franchise-terms">
  <title>Profits, fun, and flavor under the same brand</title>
  <body>
    <dl>
      <dlentry>
        <dt>Initial investment:</dt>
        <dd>
          <p>$700<xref href="#franchise-terms/initial-fee"/></p>
        </dd>
      </dlentry>
      <dlentry>
        <dt>Franchise fee:</dt>
        <dd><p>$200</p></dd>
      </dlentry>
    </dl>
    <fn id="initial-fee">
      <p>The initial investment price includes the first franchise fee payment
      </p>
    </fn>
  </body>
</topic>
```

FIGURE 6.34 Footnote in an XDITA topic. The cross-reference component links to an internal section in the topic. The link requires the **@id** of the topic (preceded by a # character) and the **@id** of the footnote element (preceded by a / character).

The Lightweight DITA subcommittee at OASIS expects that, as more developers start adopting the proposed standard, software applications will automate processes such as the insertion of footnotes.

Note

XDITA

The topical component for note (**<note>**) in XDITA gives authors a tool to make block-level content stand out from regular paragraphs. The attribute for note type (**@type**) is optional, but if used it must take a value from the following: "caution", "warning", "danger", "trouble", "notice," or "note". Example 32 has a note, with a "notice" type attribute, replacing the footnote from Figure 6.36.

HDITA and MDITA

In HDITA, the note component is expressed with an HTML division (** **) environment with the custom data attribute of **@data-class="note"**. The attribute for note type (**@data-type**) is optional, but if used it must take a value from the following: "caution", "warning", "danger", "trouble", "notice," or "note".

```
<!DOCTYPE html>
<title>Profits, fun, and flavor under the same brand</title>
<body>
  <article id="franchise-terms">
    <h1>Profits, fun, and flavor under the same brand</h1>
    <dl>
      <dt>Initial investment:</dt>
      <dd>
        <p>$700<a href="#initial-fee"><sup>1</sup></a></p>
      </dd>
      <dt>Franchise fee:</dt>
      <dd>
        <p>$200</p>
      </dd>
    </dl>

    <div id="initial-fee" data-class="fn" data-type="notice">
          <p>The initial investment price includes the first franchise
fee payment</p>
    </div>
  </article>
</body>
```

FIGURE 6.35 Footnote in an HDITA topic. The cross-reference component links to an internal division in the topic. HDITA footnote references need a clickable text component (a superscript number 1 in this example).

Figure 6.37 includes the HDITA version of the franchise terms topic with the "notice" note explaining the initial franchise payment.

The content component for note is not available in MDITA. If needed, you can include a note in an MDITA extended profile topic with an HDITA code block using the syntax of `<div data-class="note">` seen in figure 6.37.

Phrase

XDITA

If you need to make inline-level text stand out, a phrase (**<ph>**) component can do that inside a paragraph. In Figure 6.38, the first list item includes the term

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE topic PUBLIC "-//OASIS//DTD LIGHTWEIGHT DITA Topic//EN"
"lw-topic.dtd">
<topic id="franchise-terms">
  <title>Profits, fun, and flavor under the same brand</title>
  <body>
    <dl>
      <dlentry>
        <dt>Initial investment:</dt>
        <dd>
          <p>$700</p>
          <note type="notice">
            <p>The initial investment price includes the first franchise fee
payment</p>
          </note>
        </dd>
      </dlentry>
      <dlentry>
        <dt>Franchise fee:</dt>
        <dd><p>$200</p></dd>
      </dlentry>
    </dl>
  </body>
</topic>
```

FIGURE 6.36 Note environment in an XDITA topic. The note includes a paragraph with content that will stand out from the topic's other elements in deliverables produced from this source.

```
<!DOCTYPE html>
<title>Profits, fun, and flavor under the same brand</title>
<body>
  <article id="franchise-terms">
    <h1>Profits, fun, and flavor under the same brand</h1>
    <dl>
      <dt>Initial investment:</dt>
      <dd>
        <p>$700</p>
        <div data-class="note" data-type="notice">
          <p>The initial investment price includes the first franchise fee
             payment</p>
        </div>
      </dd>
      <dt>Franchise fee:</dt>
      <dd>
        <p>$200</p>
      </dd>
    </dl>
  </article>
</body>
```

FIGURE 6.37 Note environment in an HDITA topic. The note is created in HTML5 with a division element and custom data attributes to specify its class of "note" and an optional type.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE topic PUBLIC "-//OASIS//DTD LIGHTWEIGHT DITA Topic//EN"
"lw-topic.dtd">
<topic id="franchise-offer">
  <title>What we offer</title>
  <body>
    <ul>
      <li><p><ph translate="no">Know-how</ph> license</p></li>
      <li><p>Warranty of territory exclusivity</p></li>
      <li><p>Initial training</p></li>
      <li><p>Support through online, email, and telephone channels</p></li>
    </ul>
  </body>
</topic>
```

FIGURE 6.38 XDITA topic with a phrase component. The **@translate** attribute has a value of "no", which indicates to machine processors and human editors that the marked phrase should not be translated.

"know-how." As Chef Pedro and his team prepared to translate materials for the *Sensei Sushi* operations in Mexico, they realized that "know-how" does not need to be translated for audiences reading Mexican Spanish. In business-speak, "know-how" means the same for their audiences in English and Spanish. Thus, the author can flag the phrase-level content with a **<ph>** and an attribute for **@translate** with the value of "*no*". This flag will tell human authors who can see the code and machine processors that this specific phrase should not be translated from the source language.

HDITA and MDITA

The inline component to flag a phrase should be expressed in HDITA with the HTML5 span (****) element. Figure 6.39 shows the "know-how" phrase with a do not translate attribute in an HDITA topic.

MDITA

Although there is no direct equivalent for phrase or span in Markdown, you can flag phrase-level content in MDITA extended profile using an HDITA code snippet. Figure 6.40 includes the HDITA **** environment in an MDITA extended profile topic to select the phrase that should not be translated.

Preformatted Text

XDITA and HDITA

In a publishing environment using DITA or LwDITA, the *Sensei Sushi* staff has little to no control over the appearance of text in user deliverables. In the

```
<!DOCTYPE html>
<title>What we offer</title>
<body>
  <article id="franchise-offer">
    <h1>What we offer</h1>
    <ul>
      <li>
        <p><span translate="no">"Know-how"</span> license</p>
      </li>
      <li>
        <p>Warranty of territory exclusivity</p>
      </li>
      <li>
        <p>Initial training</p>
      </li>
      <li>
        <p>Support through online, email, and telephone channels</p>
      </li>
    </ul>
  </article>
</body>
```

FIGURE 6.39 HDITA topic with a phrase component. The phrase is expressed with the HTML5 **** element.

process of transformation, the content sources will be linked to stylesheets and templates with rules that most likely will be outside of a casual author's control. However, in some cases authors might want to control the look of some textual elements regardless of transformation and deliverable rules. Figure 6.41 shows how the preformatted text component (**<pre>**) in XDITA preserves the stanza or set of lines in the promise creed of *Sensei Sushi*. Because the text is wrapped in the preformatted environment, the lines will appear separated in any deliverable and won't be affected by external templates or stylesheets.

```
---
id: franchise-offer
---

# What we offer

-   <span translate="no">"Know how"</span> license

-   Warranty of territory exclusivity

-   Initial training

-   Support through online, email, and telephone channels
```

FIGURE 6.40 MDITA topic with a phrase component. If you are working in MDITA and need to mark a specific phrase, you should use a raw HDITA code snippet and create an HTML5 span.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE topic PUBLIC "-//OASIS//DTD LIGHTWEIGHT DITA Topic//EN"
"lw-topic.dtd">
<topic id="sensei-promise">
  <title>The Sensei Sushi Promise</title>
  <body>
    <pre>
      Sensei Sushi cares about tradition
      Sensei Sushi cares about the customer
      Sensei Sushi cares about fun.
     </pre>
  </body>
</topic>
```

FIGURE 6.41 Preformatted text in XDITA. The **<pre>** environment preserves line breaks as determined by the author.

Figure 6.42 shows a preview of the preformatted text in Oxygen XML's *author* mode.

The **<pre>** component performs the same function in HDITA topics, as shown in Figure 6.43.

A quick view of the HDITA topic from Figure 6.43 on a web browser shows the preformatted text with the line structure set in the source code (Figure 6.44).

MDITA

You can express the LwDITA content component of preformatted text in MDITA via indented or fenced code blocks in Markdown. The GFM spec describes indented code blocks as "composed of one or more indented chunks separated by blank lines. An indented chunk is a sequence of non-blank lines, each indented four or more spaces" (GitHub Flavored Markdown Spec, 2017). The spec also explains that a fenced code block begins with a code fence, indented no more than three spaces, and then defines code fence as "a sequence of at least three consecutive backtick characters (`) or tildes (~). (Tildes and backticks cannot be mixed.)" (GitHub Flavored Markdown Spec, 2017).

Figure 6.45 includes an MDITA topic with the *Sensei Sushi* promise preformatted via the indented code blocks method.

Section

XDITA and HDITA

The section (**<section>**) component gives authors a resource to create content separations inside a topic. Each section can contain its own title under the same parent topic. In this example, the legal team will add content to a section labeled "Terms and conditions" inside the franchise offer topic. Figure 6.46 shows the "Terms and conditions" section in XDITA, where the section's title is provided by a **<title>** tag immediately placed after the opening **<section>** tag.

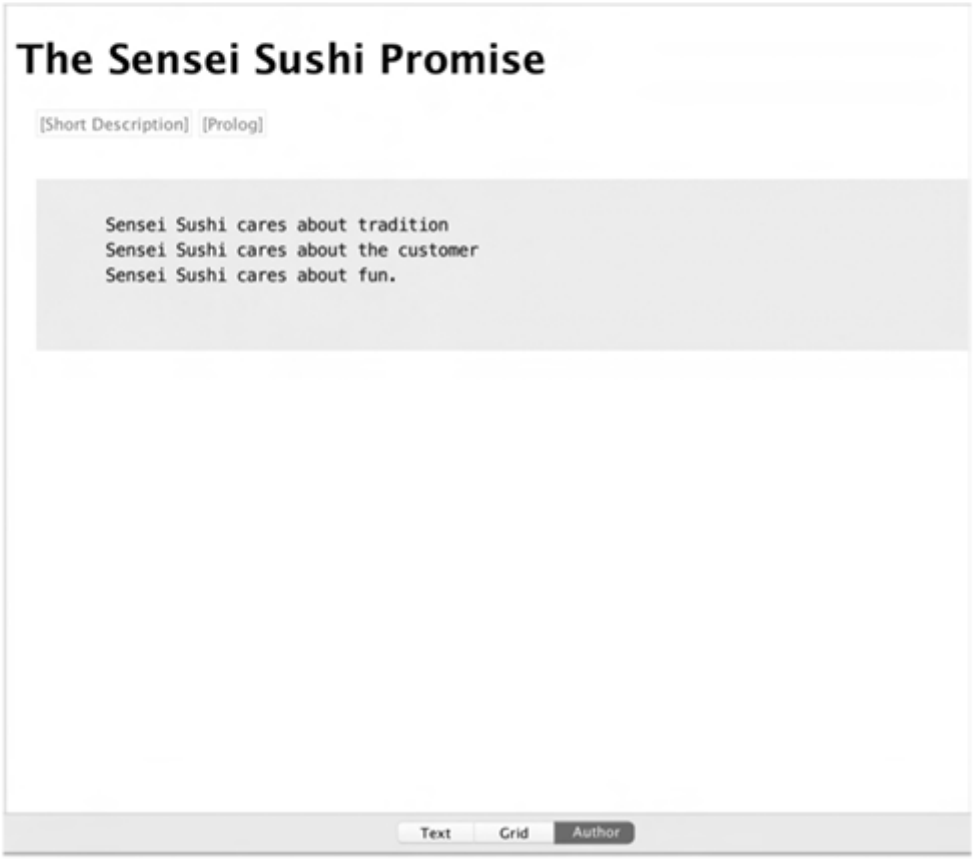


FIGURE 6.42 Preview of an XDITA topic with a preformatted text environment in Oxygen XML's *author* mode. The line breaks inside the **<pre>** tags appear as the author wrote them.

```
<!DOCTYPE html>
<title>The Sensei Sushi Promise</title>
<body>
  <article id="sensei-promise">
    <h1>The Sensei Sushi Promise</h1>
    <pre>
      Sensei Sushi cares about tradition
      Sensei Sushi cares about the customer
      Sensei Sushi cares about fun.
    </pre>
  </article>
</body>
```

FIGURE 6.43 Preformatted text in HDITA. The syntax is identical to its counterpart in XDITA.

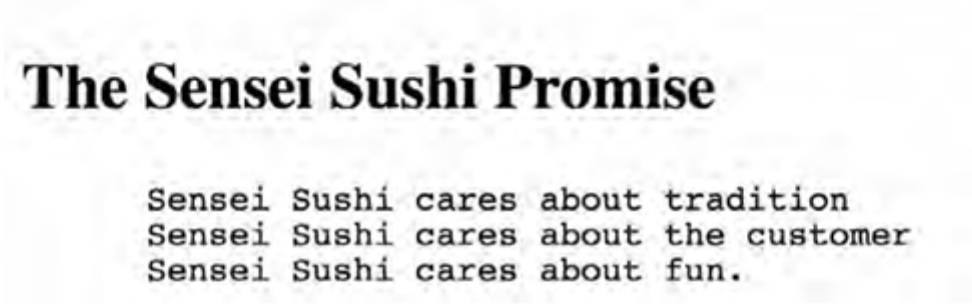


FIGURE 6.44 Web browser view of the preformatted text example in HDITA. The browser respects the line breaks established in the topic.

Figure 6.47 reproduces the franchise offer topic in HDITA syntax. In HDITA, the section component is also represented with the tag **<section>**. However, the section's title should be included in an HTML heading 2 (**<h2>**) element.

MDITA

If you are working in MDITA, you should use a Markdown heading 2 to create a content division similar to a section in XDITA and HDITA. You can represent the heading 2 with an *atx* (**##**) or *setext* (with underline from the character -) header. Figure 6.48 shows how a level-two *atx* heading can create a section in an MDITA topic.

If you need to make a direct link to a specific section inside an MDITA topic, the **@id** attribute for the section will be automatically generated as a slug.

```
---
id: sensei-promise
---

# The Sensei Sushi Promise


          Sensei Sushi cares about tradition

          Sensei Sushi cares about the customer

          Sensei Sushi cares about fun.
```

FIGURE 6.45 Preformatted text in MDITA. According to the GFM spec, preformatted lines can be indented or fenced by backtick characters or tides. In this example, the text is preformatted with indents.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE topic PUBLIC "-//OASIS//DTD LIGHTWEIGHT DITA Topic//EN"
"lw-topic.dtd">
<topic id="franchise-offer">
  <title>What we offer</title>
  <body>
    <ul>
      <li><p><ph translate="no">"Know-how"</ph> license</p></li>
      <li><p>Warranty of territory exclusivity</p></li>
      <li><p>Initial training</p></li>
      <li><p>Support through online, email, and telephone channels</p></li>
    </ul>
    <section id="terms">
      <title>Terms and conditions</title>
      <!--  Some legal content here -->
    </section>
  </body>
</topic>
```

FIGURE 6.46 Example of a section component in XDITA. The section has its own **@id** and **<title>**, functioning as a separate (but related) environment inside a larger topic.

Therefore, the **@id** for the "Terms and conditions" section from Figure 6.48 would be "*terms-and-conditions*".

Table components

XDITA

In DITA 1.3, you can represent a tabular environment in more than one way. Probably the most common is through the CALS[4] table format, which also comes with a collection of tags and options that make it a rather heavy

```
<!DOCTYPE html>
<title>What we offer</title>
<body>
  <article id="franchise-offer">
    <h1>What we offer</h1>
    <ul>
      <li>"Know-how" license</li>
      <li>Warranty of exclusive territory</li>
      <li>Initial training</li>
      <li>Support through online, email, and telephone channels</li>
    </ul>
    <section id="terms">
      <h2>Terms and conditions</h2>
      <!--  Some legal content here -->
    </section>
  </article>
</body>
```

FIGURE 6.47 Section component in HDITA. The title for the section should be provided in a heading 2 following HTML5 syntax.

```
---
id: franchise-offer
---

# What we offer

-   "Know-how" license
-   Warranty of exclusive territory
-   Initial training
-   Support through online, email, and telephone channels

## Terms and conditions

<!--  Some legal content here -->
```

FIGURE 6.48 Section component in MDITA. The section environment opens with a heading 2, which in this example is represented by `##` in *atx* syntax.

environment for casual content contributors. In XDITA, the only allowed table model is a simple table (**<simpletable>**), which also exists in DITA 1.3. As its name implies, simple table is "simpler" than the CALS model and contains the following components:

- Table header (**<sthead>**)
- Table row (**<strow>**)
- Table entry (**<stentry>**).

Figure 6.49 includes a simple table that presents the ingredients and quantities needed to prepare the "Fancy Roll" – a signature *Sensei Sushi* product.

Figure 6.50 shows a preview of the simple table environment in Oxygen's *author* mode.

HDITA

In HDITA, you should represent the components of a LwDITA table with the following HTML5 elements:

- Table (**<table>**)
- Table header (**<th>**)
- Table row (**<tr>**)
- Table entry (**<td>**).

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE topic PUBLIC "-//OASIS//DTD LIGHTWEIGHT DITA Topic//EN"
"lw-topic.dtd">
<topic id="fancy-roll">
  <title>Fancy Roll</title>
  <body>
    <simpletable>
      <sthead>
        <stentry>
          <p>Ingredient</p>
        </stentry>
        <stentry>
          <p>Amount</p>
        </stentry>
        <stentry>
          <p>Unit shipped</p>
        </stentry>
      </sthead>
      <strow>
        <stentry>
          <p>Gohan rice</p>
        </stentry>
        <stentry>
          <p>140 gms.</p>
        </stentry>
        <stentry>
          <p>14 kgs.</p>
        </stentry>
      </strow>
      <strow>
        <stentry>
          <p>Soya paper sheet</p>
        </stentry>
        <stentry>
          <p>1 pc.</p>
        </stentry>
        <stentry>
          <p>10 pcs.</p>
        </stentry>
      </strow>
    </simpletable>
  </body>
</topic>
```

FIGURE 6.49 Table inside an XDITA topic. To keep the inevitably verbose nature of a tabular environment in XML, XDITA uses the **<simpletable>** format instead of the more elaborate CALS table.

Figure 6.51 shows the HDITA code for the topic including a table with the ingredients, amounts, and units shipped for franchise owners interested in preparing the "Fancy Roll."

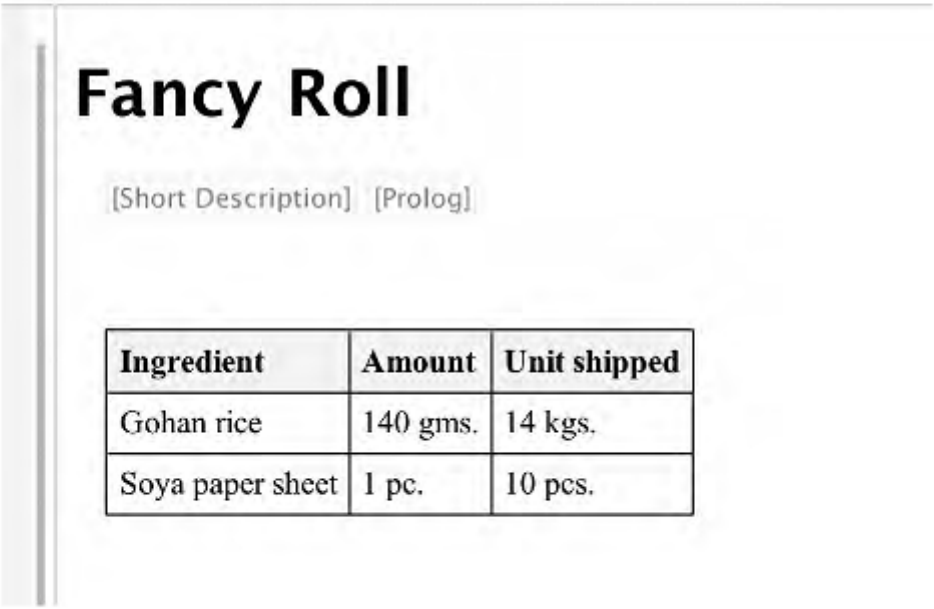


FIGURE 6.50 Preview of an XDITA table with the ingredients and shipped units for the Fancy Roll in Oxygen's *author* mode.

MDITA

The MDITA table component is based on the tabular extension included in GitHub Flavored Markdown. The table should contain a header row, a delimiter row, and zero or more rows with entries. Table entries inside a row should be separated by pipes (|), and the delimiter should contain hyphens (-), "and optionally, a leading or trailing colon (:), or both, to indicate left, right, or center alignment respectively" (GitHub Flavored Markdown Spec, 2017).

Figure 6.52 shows the topic with the "Fancy Roll" table in MDITA format.

Highlighting Components (Available in Both Topic and Map)

In LwDITA, authors have the following components to highlight sections of a text:

- Bold
- Italics
- Subscript
- Superscript
- Underline.

```
<!DOCTYPE html>
<title>Fancy Roll</title>
<body>
  <article id="fancy-roll">
    <h1>Fancy Roll</h1>
    <table>
      <tr>
        <th>
          <p>Ingredient</p>
        </th>
        <th>
          <p>Amount</p>
        </th>
        <th>
          <p>Unit shipped</p>
        </th>
      </tr>
      <tr>
        <td>
          <p>Gohan rice</p>
        </td>
        <td>
          <p>140 gms.</p>
        </td>
        <td>
          <p>14 kgs.</p>
        </td>
      </tr>
      <tr>
        <td>
          <p>Soya paper sheet</p>
        </td>
        <td>
          <p>1 pc.</p>
        </td>
        <td>
          <p>10 pcs.</p>
        </td>
      </tr>
    </table>
  </article>
</body>
```

FIGURE 6.51 Table inside an HDITA topic. Deliverable-specific formatting should be added when the topic is processed by a LwDITA-aware software tool.

```
---
id: fancy-roll
---

# Fancy Roll

| Ingredient        | Amount   | Unit shipped |
| ----------------- | -------- | ------------ |
| Gohan rice        | 140 gms. | 14 kgs.      |
| Soya paper sheet  | 1 pc.    | 10 pcs.      |
```

FIGURE 6.52 Table inside an MDITA topic. MDITA takes advantage of the table extension in GitHub Flavored Markdown to represent tabular content.

In this section, I present those components with their corresponding mappings in the three initial LwDITA authoring formats and a brief example.

XDITA

- Bold (****): <li> Warranty of **** exclusive territory ****
- Italics (****): * We offer more than 30 exclusive creations of original rolls, from the* **** *California* **** *roll* to sushi with BBQ chicken or grilled steak.
- Subscript (**<sub>**): One serving of sushi rice requires 2 cups of H**_{**2**}**O.
- Superscript (**<sup>**): Franchised restaurants should be located in areas of at least 200 ft**^{**2**}**.
- Underline (**<u>**): Franchise owners **<u>**must**</u>** pay their fees on the first week of each calendar month.

HDITA

- Bold (****): <li> Warranty of ****exclusive territory **** .
- Italics (****): We offer more than 30 exclusive creations of original rolls, from the ****California roll**** to sushi with BBQ chicken or grilled steak.
- Subscript (**<sub>**): One serving of sushi rice requires 2 cups of H**_{**2**}**O .
- Superscript (**<sup>**): Franchised restaurants should be located in areas of at least 200 ft**^{**2**}**.
- Underline (**<u>** [5]): Franchise owners **<u>**must**</u>** pay their fees on the first week of each calendar month.

MDITA

- Bold (text wrapped with ** or with __): Warranty of **exclusive territory** or Warranty of __exclusive territory__.
- Italics (text wrapped with * or with _): a) We offer more than 30 exclusive creations of original rolls, from the *California roll* to sushi with BBQ chicken or grilled steak Or b) We offer more than 30 exclusive creations of original rolls, from the _California roll_ to sushi with BBQ chicken or grilled steak.
- Subscript (**<sub>**): One serving of sushi rice requires 2 cups of H**_{**2**}**O .
- Superscript (**<sup>**): Franchised restaurants should be located in areas of at least 200 ft**^{**2**}**.
- Underline (**<u>** in an HDITA code snippet): Franchise owners **<u>**must**</u>** pay their fees on the first week of each calendar month.

Metadata Components

The semantically rich nature of intelligent content described by Rockley et al. (2015; p. 5) depends on solid metadata capabilities. In LwDITA, those capabilities are represented in components that do not necessarily appear in end-user deliverables, but give human content managers and machine processors information for computing and automation.

XDITA

At the topic level, metadata components are housed inside the prolog (**<prolog>**) environment and represented with the data (**<data>**) component, which depends on pairs of the **@name** and **@value** attributes. Those attributes can provide information like the language of a topic, critical dates for a topic (creation, last revision, expiration, etc.), and much more. In Figure 6.53, a data component provides information about the topic's author. In deliverables produced by LwDITA-aware tools, readers most likely will not see Victoria's name as the author, but the information will be there for human supervisors tracking writers' contributions and machine processors organizing content by author.

HDITA

Topic-level metadata should be included in a **<meta>** tag inside a **<head>** environment. The metadata entries should be expressed in pairs of name (**@name**) and value (**@content**), according to the W3C recommendation for HTML5 metadata[6].

Figure 6.54 places the author metadata on an HDITA topic, which labels the file as written by Victoria Fernando.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE topic PUBLIC "-//OASIS//DTD LIGHTWEIGHT DITA Topic//EN"
"lw-topic.dtd">
<topic id="franchise-intro">
  <title>An innovative, attractive, and out of the ordinary concept</title>
  <shortdesc>Are you interested in investing with us? Welcome to our franchise
information package.</shortdesc>
  <prolog>
    <data name="author" value="Victoria Fernando"/>
  </prolog>
  <body>
    <p>We offer more than 30 exclusive creations for original rolls, from the
California roll to sushi with BBQ chicken or grilled steak.</p>
 </body>
</topic>
```

FIGURE 6.53 Metadata component to identify the author of an XDITA topic. Readers of the Sensei Sushi brochure most likely won't know who created the introductory topic, but the information is embedded in the topic for human authors and machine processors with access to the XDITA source.

MDITA

So far in this chapter, I have only used the YAML header in MDITA extended profile to assign an **@id** to some topics. The YAML header can also house metadata pairs in the form of *name: value*. Figure 6.55 uses the YAML header to embed metadata that identifies Victoria Fernando as author of the topic.

Multimedia Components

LwDITA includes multimedia components that did not make it to the original release of the DITA 1.3 standard. Those components originated in LwDITA but will be included in a DITA 1.3 addendum and in the planned DITA 2.0 standard.

```
<!DOCTYPE html>
<head>
  <title>An innovative, attractive, and out of the ordinary concept</title>
  <meta name="author" content="Victoria Fernando">
</head>
<body>
  <article id="franchise-intro">
    <h1>An innovative, attractive, and out of the ordinary concept</h1>
    <p>Are you interested in investing with us? Welcome to our franchise
information package.</p>
    <p>We offer more than 30 exclusive creations for original rolls, from the
California roll to sushi with BBQ chicken or grilled steak.</p>
  </article>
</body>
```

FIGURE 6.54 Metadata component to identify the author of an HDITA topic. The metadata element is the only HDITA component that requires an HTML5 **<head>** element.

```
---
id: franchise-intro
author: Victoria Fernando
---

# An innovative, attractive, and out of the ordinary concept

Are you interested in investing with us? Welcome to our franchise
information package.

We offer more than 30 exclusive creations for original rolls, from the
California roll to sushi with BBQ chicken or grilled steak.
```

FIGURE 6.55 Metadata component to identify the author of an MDITA topic. The metadata information, provided in an optional YAML header, is only available in the MDITA extended profile.

The LwDITA multimedia components are organized around the audio (<**audio**>) and video (<**video**>) elements.

Audio in XDITA

In XDITA, audio can include the following components:

- Description (<**desc**>; previously mentioned in the figure environment)
- Controls (<**media-controls**>)
- Autoplay (<**media-autoplay**>)
- Loop (<**media-loop**>)
- Muted (<**media-muted**>)
- Source (<**media-source**>)
- Track (<**media-track**>).

Figure 6.56 shows an XDITA topic with an audio element that features the *Sensei Sushi* jingle. This audio-enhanced topic won't make it to the printed brochure for potential franchise owners, but can be included on the company's website.

The <**media-source**> component has an attribute of **@value**, which provides the path for the actual audio file that includes the recorded jingle.

Audio in HDITA and MDITA

In HDITA and MDITA extended profile, audio can include the following components:

- Description (**@title**)
- Controls (**@controls**)
- Autoplay (**@autoplay**)
- Loop (**@loop**)
- Muted (**@muted**)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE topic PUBLIC "-//OASIS//DTD LIGHTWEIGHT DITA Topic//EN"
"lw-topic.dtd">
<topic id="sensei-jingle">
  <title>The Sensei Sushi Jingle</title>
  <body>
    <audio>
      <desc>Recording of the Sensei Sushi jingle</desc>
      <media-controls value="true"/>
      <media-autoplay value="false"/>
      <media-loop value="true"/>
      <media-muted value="true"/>
      <media-source value="sensei-audio.mp3"/>
      <media-track value="sensei-audio.vtt" type="captions"/>
    </audio>
  </body>
</topic>
```

FIGURE 6.56 Audio content environment in XDITA. The components for source and track should be present; all others are optional.

- Source (**<source>**)
- Track (**<track>**).

Figure 6.57 shows an HDITA topic with the audio environment. The same environment, in an HTML5 code block, can be used in an MDITA extended profile topic if needed.

Video in XDITA

In XDITA, video can contain the following components:

- Description (**<desc>**; previously mentioned in the figure environment)
- Poster (**<video-poster>**)

```
<!DOCTYPE html>
<title>The Sensei Sushi Jingle</title>
<body>
  <article id="sensei-jingle">
    <h1>The Sensei Sushi Jingle</h1>
<audio title="Recording of the Sensei Sushi jingle" controls autoplay
loop muted>
  <source src="sensei-audio.mp3"/>
  <track src="sensei-audio.vtt" kind="captions"/>
</audio>
  </article>
</body>
```

FIGURE 6.57 Audio content environment in HDITA. In HTML5, many media components represented with elements in XDITA are expressed with attributes instead. Pay attention to the element-to -attribute correspondence in some multimedia components.

- Controls (**<media-controls>**)
- Autoplay (**<media-autoplay>**)
- Loop (**<media-loop>**)
- Muted (**<media-muted>**)
- Source (**<media-source>**)
- Track (**<media-track>**).

In Figure 6.58, the topic features a video component about the *Sensei Sushi* promise.

Video in HDITA and MDITA

In HDITA and MDITA extended profile, video can include the following components:

- Description (**@title**)
- Poster (**@poster**)
- Controls (**@controls**)
- Autoplay (**@autoplay**)
- Loop (**@loop**)
- Muted (**@muted**)
- Source (**<source>**)
- Track (**<track>**).

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE topic PUBLIC "-//OASIS//DTD LIGHTWEIGHT DITA Topic//EN"
"lw-topic.dtd">
<topic id="sensei-promise">
  <title>The Sensei Sushi Promise</title>
  <body>
    <video>
      <desc>Video about the Sensei Sushi promise</desc>
      <video-poster value="sensei-video.jpg"/>
      <media-controls value="true"/>
      <media-autoplay value="false"/>
      <media-loop value="true"/>
      <media-muted value="true"/>
      <media-source value="sensei-video.mp4"/>
      <media-track value="sensei-video.vtt" type="captions"/>
    </video>
  </body>
</topic>
```

FIGURE 6.58 Video content environment in XDITA. The main difference from the audio-enhanced example is the **<video-poster>** component, which provides an optional image for transformations, such as the PDF brochure, that cannot display a video.

Figure 6.59 shows an HDITA topic with the video environment. The same environment, in an HTML5 code block, can be used in an MDITA extended profile topic if needed.

The examples included in this chapter have featured topics for the *Sensei Sushi* brochure created in parallel XDITA, HDITA, and MDITA versions. Ignoring the experimental nature of this scenario, you might wonder now who would need different authoring formats to create the same file (e.g., why would Chef Pedro need XDITA, HDITA, and MDITA versions of the franchise offer topic?). The answer is one of the strongest selling points of LwDITA: the proposed standard allows authors to produce content in their preferred authoring format (XML, HTML5, or Markdown) with the benefit that those formats will be compatible with each other. If the promotional brochure were a real deliverable, Chef Pedro would not need three versions of the same topic. However, the legal team could author the franchise terms topic in HDITA, the kitchen manager could write the sample recipe in XDITA, and the marketing editor could create the introductory topic in MDITA. Those files, produced in different languages, would live together in LwDITA maps and produce cross-format deliverables, and that is the focus of the next section.

All Together Now: Cross-format Authoring in LwDITA

Charlotte Robidoux's article "Rhetorically Structured Content: Developing a Collaborative Single-Sourcing Curriculum" has become a timeless call to embrace structured authoring as a unifier of academia and industry practices in technical communication. Robidoux, director of digital content at Cognizant, authored that article when she was a content strategist and publications manager at Hewlett-Packard. Reporting on the practices of content developers at HP, Robidoux wrote the following about the existence of content silos:

```
<!DOCTYPE html>
<title>The Sensei Sushi Promise</title>
<body>
  <article id="sensei-promise">
    <h1>The Sensei Sushi Promise</h1>
    <video title="Video about the Sensei Sushi promise" controls autoplay
loop muted poster="sensei-video.jpg">
  <source src="sensei-video.mp4"/>
  <track src="sensei-video.vtt" kind="captions"/>
</video>
  </article>
</body>
```

FIGURE 6.59 Video content environment in HDITA. In HTML5 syntax, many of the video components that are represented with XML elements in XDITA are attributes of the **<video>** tag.

> We also observed how common it is for writers to adopt their own favorite practices when working on books independently. Working in "silos," as Rockley (2002) indicated, gave rise to inconsistency, which was difficult for editors to overcome across many thousands of pages of product documentation. Yet, when we systematically implemented structured writing guidelines from the ground up within a document set, as suggested by Ament (2002), we began to see clarity and improved consistency.
>
> (Robidoux, 2008, p.121)

Robidoux cited the first edition of Ann Rockley's *Managing Enterprise Content: A Unified Content Strategy*, which in its second edition still contains a strong warning about the development or content silos or, as Rockley described the problem, the silo trap. According to Rockley, silos are equivalent to "plaque in your arteries, inhibiting the blood flow to your vital organs. If silos hinder the flow of information, the organization is unable to function effectively or respond rapidly to threats and opportunities" (Rockley & Cooper, 2012, p.6).

Even the most "intelligent" content written in properly-structured DITA XML cannot communicate by itself with related files that were created in HTML or Markdown. The Lightweight DITA subcommittee at OASIS posits that content silos could finally communicate in a common language regardless of file format if they adopt the proposed LwDITA standard.

The Cross-format Sensei Sushi Brochure

I retake the scenario of Chef Pedro and the *Sensei Sushi* content needs in this section. As the business grows, so do its communication problems. The *Sensei Sushi* content could be easily siloed according to different teams' practices and preferred authoring languages. Let's revisit the *Sensei Sushi* content silos,

- The legal team wrote the franchise terms topic in HTML
- The kitchen manager had help from a technical writing intern who authored some recipes and procedures in XML
- The marketing editor had been creating introductory information in Markdown.

In a desktop publishing tool or word processor, you would have to copy and paste from one format to the other, making the source files instantly obsolete and taking the content out of the authors' preferred working environments. Using the DITA 1.3 standard, you could incorporate those files as external resources; i.e., a DITA map could reference the diverse authoring formats and produce a deliverable. However, each section would have its own formatting and there would be no consistency in presentation. Furthermore, in DITA

XML you cannot reuse content from, say, an HTML file in an XML topic. With LwDITA, the map could be processed to generate consistent deliverables that, regardless of authoring format, look and feel the same to end users. And, as we will see in Chapter 8, the cross-format capabilities of LwDITA allow seamless content reuse at the topic, section, paragraph, or even phrase level among XDITA, HDITA, and MDITA content files.

This means that the *Sensei Sushi* brochure designer can use a LwDITA map to aggregate and collect pieces of content, while the legal, kitchen, and marketing teams create their content in whatever LwDITA (or DITA XML) format they prefer. All those topics would live on the same intelligent content repository and produce deliverables that, to an end user, would look like coherent units without differences in style or format.

For this example, let's bring back three sample topics from earlier in this chapter. We will use the following files:

- The franchise introductory topic created in MDITA (Figure 6.16) by the marketing team (saved as *franchise-intro.md*)
- The franchise terms topic created in HDITA (Figure 6.37) by the legal team (saved as *franchise-terms.html*)
- The sample recipe for the "Fancy Roll" created in XDITA (Figure 6.49) by the kitchen staff and the technical writing intern (saved as *fancy-roll.dita*).

A map created in XDITA (Figure 6.60) could reference those topics as follows:

The brochure developer can save this map as *crossformat-brochure.ditamap*. The **@format** attribute inside each topic reference (**<topicref>**) instructs the processor on how to treat each file linked from the map. To produce a PDF deliverable from this map, the one-line command for the DITA-OT would be similar to the one we used to process an XDITA map in Chapter 5 (Figure 6.61).

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE map PUBLIC "-//OASIS//DTD LIGHTWEIGHT DITA Map//EN" "lw-map.dtd">
<map id="sensei-brochure">
  <topicmeta>
    <navtitle>Sensei Sushi Franchise Opportunity</navtitle>
  </topicmeta>
  <topicref href="franchise-intro.md" format="mdita"/>
  <topicref href="franchise-terms.html" format="hdita"/>
  <topicref href="fancy-roll.dita" format="xdita" />
    </map>
```

FIGURE 6.60 XDITA map for the cross-format version of the *Sensei Sushi* brochure. The topic reference components include attributes for format (in bold font), which take the value ("xdita", "hdita", or "mdita") corresponding to each file's authoring language.

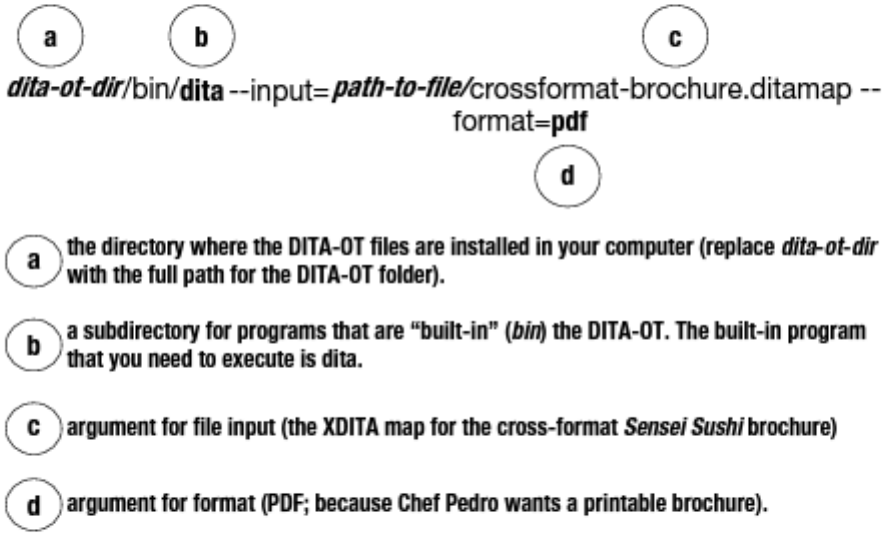


FIGURE 6.61 Command for the DITA-OT to generate a PDF deliverable from the cross-format *Sensei Sushi* XDITA map.

Figure 6.62 displays the HDITA version of the cross-format map to produce the *Sensei Sushi* introductory brochure.

The HDITA map can be saved as *crossformat-brochure.html*. Figure 6.63 shows the DITA-OT command to produce a PDF deliverable from the HDITA map.

Lastly, Figure 6.64 shows the MDITA version of the cross-format map for the *Sensei Sushi* brochure.

```
<!DOCTYPE html>
<title>Sensei Sushi Franchise Opportunity</title>
<nav id="crossformat-brochure">
  <h1>Sensei Sushi Franchise Opportunity</h1>
  <ul>
    <li>
      <p><a href="franchise-intro.md" type="text/markdown">Introduction</a>
        <p>
    </li>
    <li>
      <p><a href="franchise-terms.html" type="text/html">Legal terms</a></p>
    </li>
    <li>
      <p><a href="fancy-roll.dita" type="text/xml">Sample recipe</a></p>
    </li>
  </ul>
</nav>
```

FIGURE 6.62 HDITA map for the cross-format version of the *Sensei Sushi* brochure. To make the file valid HTML5, the attribute for format in each topic reference is **@type**, and the values are "text/markdown" for MDITA, "text/html" for HDITA, and "text/xml" for XDITA.

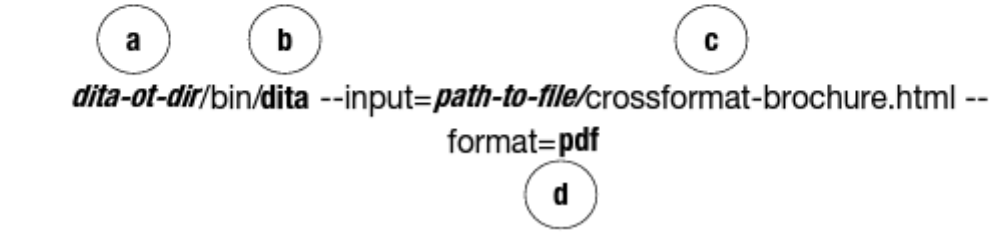


FIGURE 6.63 Command for the DITA-OT to generate a PDF deliverable from the cross-format *Sensei Sushi* HDITA map.

Figure 6.65 shows the DITA-OT command to generate a PDF deliverable from the MDITA map in Figure 6.64.

A LwDITA-aware software application like Oxygen XML Editor can also process the XDITA, HDITA, and MDITA versions of the cross-format map. The "Apply Transformation Scenario(s)" dialog box in Oxygen would present the user with options to produce deliverables from each of these LwDITA maps. Figure 6.66 shows the table of contents of a PDF built in Oxygen XML from the XDITA version of this cross-format map.

```
# Sensei Sushi Franchise Opportunity

-   [Introduction](franchise-intro.md)

-   [Legal terms](franchise-terms.html)

-   [Sample recipe](fancy-roll.dita)
```

FIGURE 6.64 MDITA map for the cross-format version of the *Sensei Sushi* brochure. In Markdown syntax there is no need for attributes specifying file types or formats.

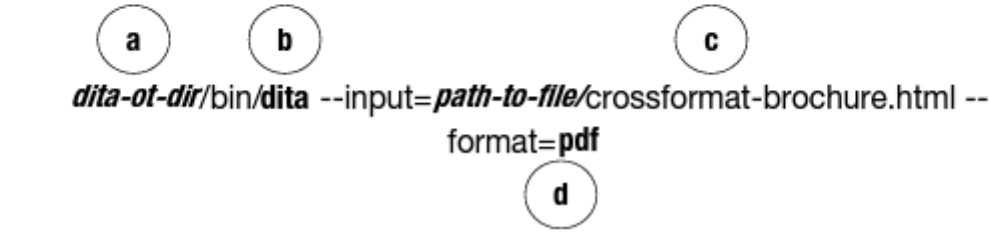


FIGURE 6.63 Command for the DITA-OT to generate a PDF deliverable from the cross-format *Sensei Sushi* HDITA map.

Figure 6.65 shows the DITA-OT command to generate a PDF deliverable from the MDITA map in Figure 6.64.

A LwDITA-aware software application like Oxygen XML Editor can also process the XDITA, HDITA, and MDITA versions of the cross-format map. The "Apply Transformation Scenario(s)" dialog box in Oxygen would present the user with options to produce deliverables from each of these LwDITA maps. Figure 6.66 shows the table of contents of a PDF built in Oxygen XML from the XDITA version of this cross-format map.

```
# Sensei Sushi Franchise Opportunity

-   [Introduction](franchise-intro.md)

-   [Legal terms](franchise-terms.html)

-   [Sample recipe](fancy-roll.dita)
```

FIGURE 6.64 MDITA map for the cross-format version of the *Sensei Sushi* brochure. In Markdown syntax there is no need for attributes specifying file types or formats.

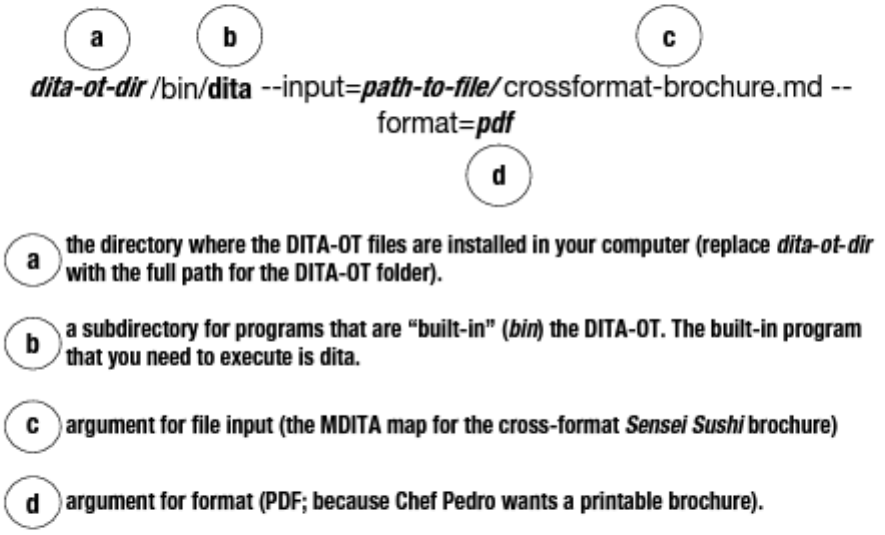
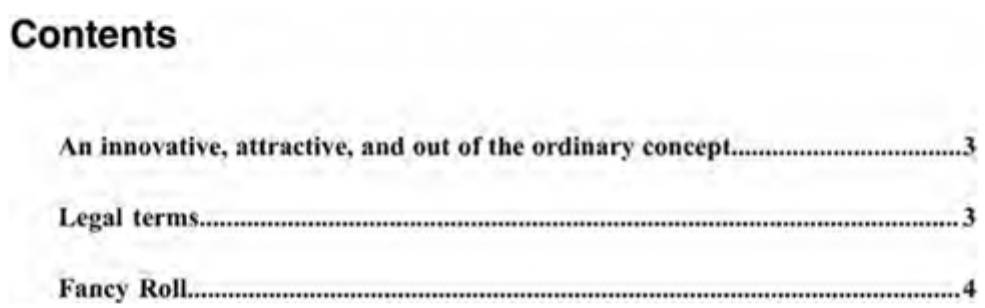


FIGURE 6.65 Command for the DITA-OT to generate a PDF deliverable from the cross-format *Sensei Sushi* MDITA map.

So, Is My Content Intelligent Now?

This chapter introduced you to the main components available in LwDITA to structure and present information. Becoming familiar with these content structures and how to represent them in the initial LwDITA authoring formats is an important step in preparing intelligent content deliverables. However, LwDITA alone will not make any content intelligent. In order to take advantage of the LwDITA proposed standard and make content stand out from the millions of dead pages created in word processors and desktop publishing tools, we need to


Contents

An innovative, attractive, and out of the ordinary concept....3

Legal terms....3

Fancy Roll....4

FIGURE 6.66 Table of contents of a PDF deliverable produced from the XDITA version of the cross-format map. End users will not know which topic was created in which authoring format. The effect of content silos is minimized when the final product presents each topic with the same format and appearance.

go back to a concept I discussed in Chapter 1 and prepare to combine content structured with LwDITA with principles of computational thinking.

In the following chapter, I revisit the concept of abstraction as a starting point of computational thinking. I also set the foundations used in Chapter 8 to examine an author's role in intelligent content workflows, expanding a series of abstractions (Evia et al., 2015) based on thinking processes recommended for technical writers. Then, I will connect those abstractions and models to LwDITA.

Notes

1 http://yaml.org/
2 https://github.github.com/gfm/#atx-heading
3 https://github.github.com/gfm/#setext-heading
4 Continuous Acquisition and Lifecycle Support
5 Although the W3C recommendation for HTML5 treats the <u> element as "unarticulated" (https://www.w3.org/TR/html5/textlevel-semantics.html#the-u-element), HDITA gives it a semantic meaning of underline to map the <u> component usage in DITA and XDITA
6 https://www.w3.org/TR/html50/document-metadata.html#attr-meta-name

References

Evia, C., Sharp, M. R., & Perez-Quiñones, M. A. (2015). Teaching structured authoring and DITA through rhetorical and computational thinking. *IEEE Transactions on Professional Communication*, 58(3), 328–343.

Evia, C., Eberlein, K., & Houser, A. (Eds.) (2018). *Lightweight DITA: An introduction*. Version 1.0. OASIS.

GitHub Flavored Markdown Spec. (2017, August 1). Retrieved from https://github.github.com/gfm/

Robidoux, C. (2008). Rhetorically structured content: Developing a collaborative single-sourcing curriculum, *Technical Communication Quarterly*, 17(1), 110–135.

Rockley, A. (2002). *Managing enterprise content: A unified content strategy*. Indianapolis, IN: New Riders.

Rockley, A. & Cooper, C. (2012). *Managing enterprise content: A unified content strategy*. (2nd ed.). Berkeley, CA: New Riders.

Rockley, A., Cooper, C., & Abel, S. (2015). *Intelligent Content: A Primer*. Laguna Hills, CA: XML Press.

W3C. (2017, December 14). HTML 5.2: W3C recommendation. Retrieved from https://www.w3.org/TR/html52