# Lightweight DITA: An Introduction

## Working Draft 16

## 02 June 2017

### Specification URIs

**This version:**
http://docs.oasis-open.org/dita/LwDITA/v1.0/cn01/LwDITA-v1.0-cn01.html (Authoritative)
http://docs.oasis-open.org/dita/LwDITA/v1.0/cn01/LwDITA-v1.0-cn01.pdf

**Previous version:**
Not applicable.

**Latest version:**
http://docs.oasis-open.org/dita/LwDITA/v1.0/LwDITA-v1.0.html (Authoritative)
http://docs.oasis-open.org/dita/LwDITA/v1.0/LwDITA-v1.0.pdf

**Technical Committee:**
OASIS Darwin Information Typing Architecture (DITA) TC

**Chair:**
Kristen James Eberlein (kris@eberleinconsulting.com), Eberlein Consulting LLC

**Editors:**
Carlos Evia (cevia@vt.edu), Virginia Tech
Kristen James Eberlein (kris@eberleinconsulting.com), Eberlein Consulting LLC

**Additional artifacts:**
This document is part of a work product that also includes:

- ZIP file that contains the DITA source for this document. http://docs.oasis-open.org/dita/LwDITA/v1.0/cn01/LwDITA-v1.0-cn01-DITA-source.zip
- ZIP files that contains the grammar files for Lightweight DITA. *http://docs.oasis-open.org/dita/LwDITA/v1.0/cn01/LwDITA-v1.0-cn01-grammars.zip*
- ZIP file that contains a sample LwDITA document. *http://docs.oasis-open.org/dita/LwDITA/v1.0/cn01/LwDITA-v1.0-cn01-samples.zip*

**Related work:**
This document is related to:

- Darwin Information Typing Architecture (DITA) Part 0: Overview. http://docs.oasis-open.org/dita/dita/v1.3/dita-v1.3-part0-overview.html.

- Darwin Information Typing Architecture (DITA) Part 1: Base Edition. http://docs.oasis-open.org/dita/dita/v1.3/dita-v1.3-part1-base.html. This edition contains topic and map; it is designed for implementers and users who need only the most fundamental pieces of the DITA framework.
- Darwin Information Typing Architecture (DITA) Part 2: Technical Content Edition. http://docs.oasis-open.org/dita/dita/v1.3/dita-v1.3-part2-tech-content.html. This edition contains the base architecture plus the technical-content specializations; it is designed for authors who use information typing and document complex applications and devices.
- Darwin Information Typing Architecture (DITA) Part 3: All-Inclusive Edition. http://docs.oasis-open.org/dita/dita/v1.3/dita-v1.3-part3-all-inclusive.html. This edition contains the base architecture, technical content, and the learning and training specializations. It is designed for implementers who want all OASIS-approved specializations, as well as users who develop learning and training materials.

### Abstract:
Lightweight DITA (LwDITA) is a simplified version of DITA that is designed to ease adoption and implementation of DITA. In comparison to DITA 1.3, LwDITA has a limited element and attribute set, stricter content models, and fewer features. LwDITA also provides mappings between XML, HTML5, and Markdown, enabling authoring, collaboration, and publishing across different markup languages.

### Status:
This document was last revised or approved by the OASIS Darwin Information Typing Architecture (DITA) TC on the above date. The level of approval is also listed above. Check the "Latest version" location noted above for possible later revisions of this document.

TC members should send comments on this document to the TC's email list. Others should send comments to the TC's public comment list, after subscribing to it by following the instructions at the "Send A Comment" button on the TC's web page at https://www.oasis-open.org/committees/comments/index.php?wg_abbrev=dita.

### Citation format:
When referencing this note, the following citation format should be used:

[LwDITA-intro-v1.0]
*Lightweight DITA: An Introduction* Version 1.0. Edited by Carlos Evia and Kristen James Eberlein. 02 June 2017. Working Draft 16. http://docs.oasis-open.org/dita/LwDITA/v1.0/cn01/LwDITA-v1.0-cn01.html. Latest version: http://docs.oasis-open.org/dita/LwDITA/v1.0/LwDITA-v1.0.html.

## Notices

# Table of contents

# 1 Introduction

Lightweight DITA (LwDITA) is a simplified version of the Darwin Information Typing Architecture (DITA). It is designed to ease adoption and implementation of DITA. In comparison to DITA 1.3, LwDITA has a limited element and attribute set, stricter content models, and fewer features. LwDITA also provides mappings between XML, HTML5, and Markdown, enabling authoring, collaboration, and publishing across different markup languages.

This committee note covers the following points:

- Rationale for LwDITA
- Design of LwDITA
- Authoring formats in LwDITA
- Potential audiences for LwDITA
- Cross-format authoring and publishing
- Current LwDITA tools

## 1.1 References

The following are references to external documents or resources that readers of this document might find useful.

**[GFM]**

*GitHub Flavored Markdown Spec*. *https://github.github.com/gfm/*.

**[HTML5]**

*HTML5 W3C Recommendation*. Edited by Ian Hickson, Robin Berjon, Steve Faulkner, Travis Leithead, Erika Doyle Navara, Edward O'Connor, and Silvia Pfeiffer. 28 October 2014. *http://www.w3.org/TR/2014/REC-html5-20141028/*. Latest version: *http://www.w3.org/TR/html5/*.

**[LwDITA-cross-format-content]**

*Cross-format content with Lightweight DITA*. Session by Michael Priestley, Jenifer Schlotfeldt, and Carlos Evia. Session at CMS/DITA North America 2016. Latest version: *http://www.slideshare.net/mpriestley/crossformat-content-with-lightweight-dita?qid=802e7d40-7bbf-42ba-8aca-a446cdb78ce5&v=&b=&from_search=6*.

**[LwDITA-pre/overview]**

*Lightweight DITA: A pre/overview*. Session by Michael Priestley at CMS/DITA North America 2016. Latest version: *http://www.slideshare.net/mpriestley/lightweight-dita-a-preoverview?qid=b2aade0d-6c48-4ca7-b572-f69ff3f9467f&v=&b=&from_search=3*

**[LwDITA]**

*Overview of Lightweight DITA*. Blog post by Michael Priestley. 11 April 2014. Latest version: *http://dita.xml.org/blog/overview-of-lightweight-dita-xdita-and-hdita*

**[LwDITA-IXIASOFT]**

*Lightweight DITA: What Is It and Can I Use It in the IXIASOFT DITA CMS?* Authored by Leigh W. White. 28 November 2016. Latest version: *http://www.ixiasoft.com/en/news-and-events/blog/2016/lightweight-dita-what-it-and-can-i-use-it-ixiasoft-dita-cms/*.

**[Markingdown-DITA]**

*Marking Down DITA*. Authored by Roger Fienhold Sheen. 30 April 2015. Latest version: *http://infotexture.net/2015/04/dita-ot-markdown-plugin/*.

**[Structured-Authoring-wo-XML]**

*Structured Authoring without XML: Evaluating Lightweight DITA for Technical Documentation* Authored by Carlos Evia and Michael Priestley. *Technical Communication*, volume 63, number 1 (February 2016): 23-37.*http://www.ingentaconnect.com/contentone/stc/tc/2016/00000063/00000001/art00004*.

**[YAML]**

*YAML Specification Index*. Edited by Oren Ben-Kiki, Clark Evans, Ingy döt Net. 29 September 2009. *http://yaml.org/spec/*. Latest version: *http://yaml.org/spec/1.2/spec.html*.

## 1.2 Terminology

This section provides information about terminology and how it is used in this committee note.

**ATX headers**

(MDITA) One or two hash (#) marks at the beginning of a line of text. One hash mark indicates a topic title, and two hash marks indicates a section title.

**core profile**

(MDITA) The authoring profile that aligns with the specification for GitHub Flavored Markdown.

**custom data attributes**

Custom attributes, such as `@data-hd-conref`, that are used in HDITA and the extended profile of MDITA in order to use such DITA features as conref and keyref.

**document type**

A type of DITA topic or map that is designed for a specific purpose.

**extended profile**

(MDITA) The authoring profile that relies on specific MarkDown variants in order to enable use of such DITA features as the `@id` attribute on the root element, prolog metadata, and optional use of HTML elements.

**HDITA**

The LwDITA authoring format that is based on HTML5.

**MDITA**

The LwDITA authoring format that is based on Markdown.

**slug**

A URL-friendly version of a topic title.

**specialization**

The process of creating a new DITA element or attribute from an existing element or attribute. The new element or attribute inherits characteristics of the element or attribute from which it was specialized, which reduces design work and enables the reuse of existing transformations.

**XDITA**

The LwDITA authoring format that is based on XML.

# 2 Why Lightweight DITA?

DITA 1.3 is seen as too complex for some scenarios. In addition, some communities do not use XML as an authoring platform.

DITA 1.3 is a mature architecture with a deep set of advanced features. This maturity can be intimidating for those considering adoption, especially for simple scenarios. While simplified versions of DITA exist, they are vendor-developed and proprietary. A standards-based lightweight entry point will enable the DITA community to offer a common starting point for simple DITA scenarios that remains fully compatible with DITA 1.3 solutions.

Some authoring communities have strong ties to specific authoring formats, such as Markdown. While these alternative formats do not have the same expressiveness as XML, they bring with them a set of tools and practices that can be a natural fit with a DITA ecosystem, if we can define a lower-function level of interchange. LwDITA can provide this mapping; it can become the first version of DITA to be truly cross-format - allowing authoring and delivery in a mix of native formats all mapped to a common semantic standard.

The Lightweight DITA subcommittee began work by identifying key authoring communities that were interested in the benefits that LwDITA could provide; it then identified scenarios including cross-format authoring and reuse. LwDITA represents the common denominator for the functionality that is needed by the following authoring communities: learning and training, SME-authored software documentation, and marketing content.

# 3 What is Lightweight DITA?

LwDITA is a proposed specification for expressing simplified DITA documents in XML, HTML5, and Markdown.

The core goals of LwDITA are the following:

- Provide a simpler DITA experience
- Provide mappings between XML, HTML5, and Markdown that enable individuals to:
  - Author content in the format of *their* choice
  - Easily collaborate and publish content across these different markup languages
- Foster the growth of new, low-cost tools and applications that support LwDITA

## 3.1 Simplified structure

DITA 1.3 has more power (and thus complexity) than is needed in some situations. LwDITA provides a simpler alternative.

While LwDITA supports core features in the DITA standard – semantic tagging, topic orientation, content reuse, conditional processing, and specialization – LwDITA deliberately limits itself to generic structures that are highly applicable across many industries. This results in a much smaller standard in terms of elements, attributes, features, and complexity.

Conference presentations and practitioners' blogs occasionally describe DITA as an intimidating language with too many document and element types. In the base edition, DITA 1.3 has three document types and 189 element types. In contrast, LwDITA has two document types and 40 elements. 33 of the elements are defined in DITA 1.3, and the other seven are multimedia elements that are expected to be part of DITA 2.0.

This pragmatic design has benefits for both small and large projects, as well as new and existing DITA implementations. Compared to DITA 1.3, the learning curve for LwDITA will be shorter, and implementing LwDITA might involve less change management and, as a result, lower costs.

## 3.2 Support for non-XML formats

LwDITA adds support for structured authoring in HTML5 and Markdown.

New forms of non-XML structured authoring have gained popularity. Authors use the extended semantic markup of HTML5 to create structured documents for the Web. Many in industry and academia have adopted plain text languages like Markdown.

In its initial release, LwDITA has three authoring formats:

**XDITA**

An XML-based variant

**HDITA**

An HTML5-based variant

**MDITA**

A Markdown-based variant

These authoring formats will enable and enhance collaboration across divisional silos. Engineers can author in Markdown, marketing writers can author in HTML5, and technical writers and other familiar with DITA can author in XML. Documents authored in the various authoring formats can be aggregated together and published as a single document collection. They also can easily integrate into DITA 1.3 collections.

These three authoring formats do not represent a final version of LwDITA. In the future, based on community interest and development resources, LwDITA might add additional authoring formats, for example, mappings between DITA and JSON, AsciiDoc, or MS Word.

HDITA and XDITA are designed to be fully compatible with each other, while MDITA is a compatible subset. XDITA and HDITA conform with the OASIS DITA and W3C HTML5 standards, respectively. In its core profile, MDITA aligns with the GitHub Flavored Markdown specification. In its extended profile, MDITA can incorporate YAML front matter headers and HDITA elements and attributes to overcome Markdown limitations as a language for authoring structured and reusable content.

## 3.3 Development of LwDITA tools and applications

We hope that LwDITA will make it easier for companies to develop inexpensive tools for authoring, aggregating, and publishing LwDITA content.

DITA 1.3, with its many elements and advanced features, makes it difficult for companies to implement new authoring and publishing systems. In contrast, the simplified and predictable structure of LwDITA ought to remove many of the barriers that stand in the way of the development of new tools, both commercial and open-source.

# 4 Lightweight DITA design

LwDITA is designed to have a smaller element set, a stricter content model, and fewer reuse mechanisms than DITA 1.3. However, LwDITA also includes new elements and attributes that provide increased multimedia support.

## 4.1 Elements in the LwDITA topic

LwDITA is designed to use a subset of the topic elements that are available in DITA 1.3.

The subset was carefully chosen to include the most basic constructions that are needed in order to structure information effectively. The Lightweight DITA subcommittee considered the needs of diverse industries and sectors (including education, engineering, healthcare, and marketing) when selecting topic elements for LwDITA.

The elements selected represent the following types of information:

- Body
- Cross reference
- Data
- Description
- Figure
- Footnote
- Image and alternate text
- In-line formatting: Bold, italics, underline, super script, subscript
- Lists
  - Definition list
  - List item
  - Ordered list
  - Unordered list
- Note
- Paragraph
- Phrase
- Prolog
- Preformatted text
- Section
- Short description
- Table
- Title
- Topic

For a complete list of the DITA 1.3 elements that are included in LwDITA and their availability in the authoring formats, see *DITA 1.3 elements in LwDITA* **(23)**.

## 4.2 Elements in the LwDITA map

LwDITA is designed to use a subset of the map elements that are available in DITA 1.3.

The elements selected represent the following types of information:

- Data
- In-line formatting: Bold, italics, underline, super script, subscript
- Key definition
- Link text
- Map
- Navigation title
- Phrase
- Topic metadata
- Topic reference

For a complete list of the DITA 1.3 elements that are included in LwDITA and their availability in the authoring formats, see *DITA 1.3 elements in LwDITA* **(23)**.

## 4.3 Stricter content model

LwDITA has a much stricter content model than DITA 1.3 This ensures a predictable markup structure in topics that simplifies reuse, transformations, style sheet logic, and tools development.

This strict content model minimizes authoring decisions by presenting limited choices for elements and attributes. This model, however, depends on a few strict rules. For example, in XDITA and HDITA, with a few exceptions, all text must be within paragraph elements. Exceptions are the description, short description, and title elements. Within paragraphs, the following inline elements can appear:

- Bold
- Italics
- Phrase
- Superscript
- Subscript
- Underline (only available in XDITA)

In DITA 1.3, the following markup is valid:

```
<section>Compatible light bulbs include the following:
 <ul>
  <li>Compact Fluorescent</li>
  <li>Light Emitting Diode</li>
 </ul>
</section>
```

In contrast, in XDITA the following markup must be used:

```
<section>
 <p>Compatible light bulbs include the following:</p>
 <ul>
  <li>
```

```
     <p>Compact Fluorescent</p>
    </li>
    <li>
     <p>Light Emitting Diode</p>
    </li>
   </ul>
  </section>
```

Note that all text is wrapped in `<p>` elements. This restriction of mixed content in block elements simplifies tool development for processing LwDITA content, and it also enables easier content reuse, as authors can conref paragraphs into most of the block elements that are available in LwDITA.

## 4.4 Subset of reuse mechanisms

LwDITA has a smaller set of reuse mechanisms than DITA 1.3.

**Conditional processing**

The only conditional processing attribute is the `@props` attribute.

**Content reference**

The `@conref` attribute is available on the following elements:

- Audio
- Definition description
- Definition list
- Definition list entry
- Definition term
- Footnote
- List item
- Note
- Ordered list
- Paragraph
- Preformatted text
- Section
- Simple table
- Simple table entry
- Simple table header
- Simple table row
- Unordered list
- Video

The content reference mechanism is not available in MDITA.

**Key reference**

The `@keyref` attribute is available only on the phrase or span element.

**Variable text**

For variable text, such as product names, authors can use `@keyref` on phrase or span.

This design simplifies the DITA authoring experience, as there are no choices to be made. To reuse block-level content, authors will use `@conref`. For phrase-level content, authors will use `@keyref`.

For a complete list of the DITA 1.3 attributes that are included in LwDITA, see *DITA 1.3 attributes in LwDITA* **(26)**.

## 4.5 New multimedia elements

LwDITA adds new elements for multimedia content. These elements are designed for compatibility with HTML5.

For years, authors have used different approaches to embed multimedia content in DITA-based deliverables for the Web. The DITA 1.3 specification recommends the `<object>` element to include multimedia content in a topic, pointing out that it corresponds to the `<object>` element in HTML. However, one of HTML5 key features was the introduction of direct elements for audio and video. LwDITA updates the XML-to-HTML element correspondence and introduces the following multimedia elements, which are specialized from the DITA 1.3 `<object>` element:

**Audio**

Audio is a link to sound to be included in the content.

**Controls**

Controls enable user interfaces for video playback and volume in Web-aimed transformations.

**Fallback**

Fallback is content, or a link, presented as alternative to media resources of audio or video.

**Poster**

Poster is a link to an image or static video frame.

**Source**

Source is a link to media resources of audio or video content.

**Track**

Track is a link time-based text data relevant to audio or video content.

**Video**

Video is a link to an audiovisual product to be included in the content.

These multimedia elements are not available in the MDITA core profile; they must be expressed in raw HDITA syntax as part of the MDITA extended profile.

## 4.6 Modified footnote element

LwDITA includes a modified element for footnote. It has a required `@id` attribute to force the *use-by-reference* model that is available in DITA 1.3.

The LwDITA `<fn>` element has the following characteristics:

- Block element
- Available in all block contexts

# 5 LwDITA authoring formats

LwDITA offers three authoring formats: XDITA, HDITA, and MDITA.

## 5.1 XDITA

XDITA is the authoring format of LwDITA that uses XML to structure information. XDITA is a subset of DITA, with new multimedia elements added to support interoperability with HTML5.

### 5.1.1 Audience for XDITA

XDITA is designed to be used by individuals who want to author DITA content but who do not want (or need) the full power of DITA.

Potential users of XDITA might include the following:

- Information developers who use an XML editor but who want a smaller set of elements and attributes with which to work
- Departments who want to reduce the cost of developing and maintaining style sheets
- Content developers who want their DITA content to be subsumed by a product documentation set that is based on Markdown or HTML5

### 5.1.2 Example of an XDITA topic

The following topic is authored in XDITA. In addition to basic DITA elements, note the new `<video>` element that is highlighted in bold.

```
<topic id="install-and-setup">
  <title>Installing and Setting up Remote Lighting</title>
  <shortdesc>Installation of your lighting kit includes installing the light bulbs into
  light fixtures, preparing the remote control, and programming lighting groups.
  </shortdesc>
  <prolog>
    <data name="author" value="Kevin Lewis"/>
  </prolog>
  <body>
   <section>
    <title>Steps</title>
    <ul>
      <li><p>Install light bulbs.</p></li>
      <li><p>Prepare remote control.</p></li>
      <li><p>Program lighting groups.</p></li>
    </ul>
  </section>
  <section>
    <title>Example</title>
    <p>The following video demonstrates a recommended installation:</p>
  <video>
      <controls />
      <source value="remote.mp4" />
    </video>
  </section>
  </body>
</topic>
```

XDITA topics are designed to be fully compatible with DITA topics. An author can work on an XDITA topic and keep it in a collection of LwDITA topics, but that same topic will also be compatible with maps and topics authored in DITA 1.3.

### 5.1.3 Example of an XDITA map

The following map is authored in XDITA.

```
<map id="remote-main">
    <topicmeta>
     <navtitle>Remote Lighting Network</navtitle>
    </topicmeta>
    <keydef keys="product-name">
      <topicmeta>
        <linktext><ph>Remote Network Lighting</ph></linktext>
      </topicmeta>
    </keydef>
    <topicref href="introduction.dita">
     <topicmeta>
      <navtitle>Introduction</navtitle>
     </topicmeta>
    </topicref>
    <topicref href="alternatives.dita">
     <topicmeta>
      <navtitle>Alternative lighting setups</navtitle>
     </topicmeta>
     <topicref href="low-power.dita">
      <topicmeta>
       <navtitle>Low power installation</navtitle>
      </topicmeta>
     </topicref>
     <topicref href="high-power.dita">
      <topicmeta>
       <navtitle>High power installation</navtitle>
      </topicmeta>
     </topicref>
    </topicref>
</map>
```

## 5.2 HDITA

HDITA is the authoring format of LwDITA that uses HTML5 to structure information. It also uses custom data attributes to provide interoperability with DITA.

### 5.2.1 Audience for HDITA

HDITA is designed to be used by individuals who want to author structured content using tools that are designed for HTML authoring.

Potential users of HDITA might include the following:

- Marketing writers who want to contribute to DITA-based product documentation without using an XML editor
- Software developers who want to contribute to documentation using tools for authoring HTML content
- Teachers and trainers who want to create course content for a Web site or learning management system (LMS)
- Bloggers and content strategists who want to be able to create and edit content using mobile devices

### 5.2.2 Example of an HDITA topic

The following topic is authored in HDITA. The topic uses HTML5 elements and custom data attributes for content reuse and compatibility with DITA. The custom data attribute highlighted

in bold includes a content reference from a DITA topic with a disclaimer expected from all topics in this fictional scenario.

```
<meta name="author" content="Kevin Lewis">
<article id="install-and-setup">
<h1>Installing and Setting up Remote Lighting</h1>
  <p>Installation of your lighting kit includes installing the light bulbs into light
 fixtures, preparing the remote control, and programming lighting groups.</p>
    <h2>Steps</h2>
      <ul>
        <li><p>Install light bulbs.</p></li>
        <li><p>Prepare remote control.</p></li>
        <li><p>Program lighting groups.</p></li>
      </ul>
      <h2>Example</h2>
      <p>The following video demonstrates a recommended installation:</p>
      <video src="remote.mp4" controls poster="remote.png" />
   <p data-hd-conref="bulbs-to-groups.dita#bulbs-to-groups/assign-disclaimer" />
  </article>
```

### 5.2.3 Example of an HDITA map

An HDITA map is authored in HTML5.

```
<nav>
    <h1>Remote Lighting Network</h1>
    <div data-hd-class="keydef">
    <var data-hd-class="linktext">Remote Lighting Network</var>
    </div>
    <ul>
     <li><p><a href="introduction.html">Introduction</a><p></li>
     <li><p><a href="alternatives.html">Alternative lighting setups</a></p>
      <ul>
       <li><p><a href="low-power.html">Low power installation</a></p></li>
       <li><p><a href="high-power.html">High power installation</a></p></li>
       </ul>
      </li>
     </ul>
   </nav>
```

## 5.3 MDITA

MDITA is the authoring format of LwDITA that uses Markdown to structure information.

LwDITA includes two profiles for authoring MDITA topics:

**Core profile**

Aligns with the GitHub Flavored Markdown spec and includes elements common to *most* Markdown flavors.

**Extended profile**

Relies upon features only available in specific flavors of Markdown to enable a more consistent DITA-like experience.

### 5.3.1 Audience for MDITA

MDITA is designed to be used by individuals who want to author structured content with the minimum of overhead, but who also want to take advantage of the reuse mechanisms associated with the DITA standard and the multi-channel publishing afforded by standard DITA tooling.

Potential users of the MDITA core profile might include the following:

- Software developers who want to contribute to DITA-based product documentation without using an XML editor
- Software developers who want to contribute to product documentation using the tools and markup of their choice
- Individuals authoring content using a platform, such as a mobile device, that does not support an XML editor
- Individuals authoring content quickly that must be later refactored as structured content

Potential users of the MDITA extended profile might include the following:

- Content curators who receive occasional contributions from developers written in Markdown
- Technical editors who need to incorporate Markdown files in DITA or XDITA topic collections
- Content developers familiar with DITA or XDITA who want to use Markdown as an authoring language on devices that do not support XML editors

### 5.3.2 Examples of MDITA topics

An MDITA topic is authored in Markdown. MDITA topics can be created using either core or extended profiles.

#### *MDITA core profile*

The MDITA core profile contains simple, straight-forward information structures that are readily available in Markdown:

- Title
- Paragraph
- Section title
- Unordered list
- Table
- Code block

The MDITA core profile aligns with the GitHub Flavored Markdown Spec. The following example shows an MDITA core-profile topic:

```
# Installing and Setting up Remote Lighting

Installation of your lighting kit includes installing the light bulbs into light
 fixtures, preparing the remote control, and programming lighting groups.

## Steps

  1. Install light bulbs.
  2. Prepare remote control.
  3. Program lighting groups.

## Example

 ! [Image](remote.png)
```

In an MDITA topic, the required topic `@id` attribute is generated with a slug version of the topic title, following a process similar to the WordPress URL creation for posts.

### *MDITA extended Profile*

The MDITA extended profile acknowledges the limitations of Markdown as a language for structuring content. This profile allows the following elements to enhance interoperability with other LwDITA authoring formats and DITA 1.3:

- An optional YAML front matter header. 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.
- Optional raw HDITA attributes and elements. Although MDITA allows for this kind of syntax extension, its validation will depend on specific implementations.

The following example shows an MDITA extended-profile topic with a YAML header indicating its `@id` and author, and an HDITA element that enables the topic to reference a video (indicated in bold text.

```
---
id: install-and-setup
author: Kevin Lewis
---

# Installing and Setting up Remote Lighting

Installation of your lighting kit includes installing the light bulbs into light
 fixtures, preparing the remote control, and programming lighting groups.

Before you attempt to install your lighting kit, please turn off the power in your
 electrical circuit panel,

## Steps

  1. Install light bulbs.
  2. Prepare remote control.
  3. Program lighting groups.

## Example

The following video demonstrates a recommended installation:

<video src="remote.mp4" controls poster="remote.png" />
```

MDITA topics are designed as a compatible subset of XDITA and HDITA topics.

## 5.3.3 Example of an MDITA map

An MDITA map is authored in Markdown. The following example uses MDITA core-profile code to produce a map with a title, and an unordered list (itself containing a nested, unordered list) of titles for topics and their associated file names.

```
# Remote Lighting Network

   - [Introduction](introduction.md)
   - [Alternative lighting setups](alternatives.md)
       - [Low power installation](low-power.md)
       - [High power installation](high-power.md)
```

## 5.4 Authoring cross-format content with LwDITA

LwDITA enables cross-format content sharing. Authors can create topics in XDITA, HDITA, or extended-profile MDITA and then publish them as a unified collection that uses content referencing and key referencing.

In the following example, a team that develops content for a lighting product shares topics authored in the LwDITA authoring formats. The team even takes advantage of the conref and keyref mechanisms. The example contains the following:

- A DITA map that references topics authored in XDITA, HDITA, MDITA, and DITA 1.3. It also contains a key definition for the product name.
- An XDITA topic, created by a technical writer, that conrefs content from an MDITA topic
- An HDITA topic, created by a marketing specialist, that conrefs content from an XDITA topic
- An extended-profile MDITA topic, created by a software developer, that conrefs content from an HDITA topic

Each of the LwDITA topics use a key reference to refer to the product name.

### 5.4.1 Cross-format example: XDITA map

The following XDITA map links to topics authored in the three formats of LwDITA and DITA 1.3. It also provides a key for the product's name.

```
<map>
<topicmeta>
  <navtitle>Remote Lighting Setup</navtitle>
</topicmeta>

   <keydef keys="product-name">
      <topicmeta>
        <linktext><ph>Remote Network Lighting</ph></linktext>
      </topicmeta>
    </keydef>

  <topicref href="xdita-topics/bulbs-to-groups.dita" format="dita"/>
  <topicref href="hdita-topics/low-power.html" format="html"/>
  <topicref href="mdita-topics/basic-concepts.md" format="markdown"/>
  <topicref href="external/dita-topics/contact-info.dita" format="dita"/>
</map>
```

### 5.4.2 Cross-format example: XDITA topic

The following XDITA topic contains a key reference to a product name and a content reference to a paragraph from an MDITA topic.

```
<topic id="bulbs-to-groups">
   <title>Programming Light Bulbs to a Lighting Group</title>
   <shortdesc>You can program one or more light bulbs to a lighting group to operate
 that group with your remote control.</shortdesc>
   <body>
   <section id="context">
   <p>Your <ph keyref="product-name"/> remote control can manage up to 250 network
 light bulbs on the same lighting network. When you add a light bulb to the network,
 you can program it to one or more lighting groups.</p>
   <p id="assign-disclaimer">You must assign a light bulb to at least one lighting
 group to operate that light bulb.</p>
   </section>
   <section id="steps">
    <ol>
    <li><p conref="basic-concepts.md#basic-concepts/power-off" /></li>
    <li><p>Remove any existing light bulb from the light fixture.</p></li>
```

```
    <li><p>Install the network light bulb into the light fixture as you would any
 standard light bulb.</p></li>
    <li><p>Turn power to the light fixture on.</p></li>
   </ol>
   </section>
 </body>
 </topic>
```

### 5.4.3 Cross-format example: HDITA topic

The following HDITA topic contains a key reference to a product name and a content reference to a paragraph from an XDITA topic.

```
<!DOCTYPE html>
<html>
  <title>Low-Power Networking</title>
  <article id="low-power">
    <h1>Low-Power Networking</h1>
    <p>Your <span data-hd-keyref="product-name" /> operates at a low level of
 networking power but can successfully connect at long distances because they can send
 information from light bulb to light bulb.</p>
<p data-hd-conref="bulbs-to-groups.dita#bulbs-to-groups/assign-disclaimer" />
    <p id="disconnect-warning" data-hd-class="note">Even in low power networks, be sure
 to disconnect all devices before performing maintenance tasks.</p>
  </article>
</html>
```

### 5.4.4 Cross-format example: MDITA topic

The following MDITA extended-profile topic contains a key reference to a product name and a content reference to a paragraph from an HDITA topic.

```
---
id: basic-concepts
---
You can network LED light bulbs together to operate wirelessly from a remote control
 using the RemotaLux app.

# Basic Concepts of Network Lighting

Network light bulbs from your [product-name] work with your light fixtures the same way
 as standard light bulbs. They are different, however, in a couple of ways:

   - The lighting element in the light bulb uses energy-efficient LED technology.
   - The light bulb includes wireless technology that allows the light bulb to connect
 to a network and be managed remotely using the RemotaLux app.

<p id="power-off">Make sure power to the fixture where you are installing the light
 bulb is turned OFF.</p>

<p conref="low-power.html#low-power/disconnect-warning" />
```

# 6 Lightweight DITA tools

Several tools already exist to support organizations who want to explore using LwDITA.

**Markdown DITA-OT plug-in developed and maintained by Jarno Elovirta**

This plug-in contains the following features:

- A custom Markdown parser that enables the use of MDITA with DITA-OT
- A DITA-OT transformation type that generates Markdown from DITA source files

The plug-in is available at *https://github.com/jelovirt/dita-ot-markdown*. It is licensed for use under the Apache License 2.0 and can be used with DITA-OT, version 2.1 and later.

**oXygen XML Editor, versions 18.1 and later**

Oxygen XML Editor includes features that enable content developers to integrate Markdown documents in a DITA project. The integration between the Markdown editor and DITA includes actions to export or convert Markdown documents to DITA topics; a preview of how topics will look after conversion is also available. In addition, the DITA Maps Manager view also enables content developer to reference Markdown topics in a DITA map. Oxygen XML Editor also includes support and authoring templates for topic and map in XDITA by incorporating the DTD files produced by the Lightweight DITA subcommittee.

The DITA Technical Committee expects that the release of Lightweight DITA as an OASIS standard will lead to a rapid increase in the number of commercial and open-source tools that provide support for LwDITA.

# Appendix A LwDITA elements and attributes

This section lists the elements and attributes that are available in LwDITA.

## Appendix A.1 DITA 1.3 elements in LwDITA

This topic lists the DITA 1.3 elements that are available in LwDITA. It also lists how to represent them in XDITA, HDITA, and MDITA.

| Component | XDITA | HDITA | MDITA |
|---|---|---|---|
| Alternate text | `<alt>` | Attribute on `<img>` | [text] |
| Body | `<body>` | `<body>` | Not applicable |
| Bold | `<b>` | `<strong>` | ** or __ |
| Cross reference | `<xref>` | `<a href>` | [link](/URI "title") |
| Data | `<data>` | `<meta>` | **(MDITA extended profile)**Any variables declared in a YAML front matter header. The front matter must be the first thing in the file and must be set between triple-dashed lines. |
| Definition description | `<dd>` | `<dd>` | **(MDITA extended profile)**`<dd>` in HDITA syntax |
| Definition list entry | `<dlentry>` | Not applicable | Not applicable |
| Definition term | `<dt>` | `<dt>` | **(MDITA extended profile)**`<dt>` in HDITA syntax |
| Definition list | `<dl>` | `<dl>` | **(MDITA extended profile)**`<dl>` in HDITA syntax |
| Description | `<desc>` | `<caption>` in `<table>`; `<figcaption>` in `<figure>`; not applicable in links | Not applicable |
| Figure | `<fig>` | `<figure>` | Not applicable |

| Component | XDITA | HDITA | MDITA |
|---|---|---|---|
| Footnote | `<fn>` | `<div data-hd-class="fn">` | **(MDITA extended profile)**`<div data-hd-class="fn">` |
| Image | `<image>` | `<img>` | ![alt text for an image](images/image_name.jpg) |
| Italics | `<i>` | `<em>` | * or _ |
| Key definition | `<keydef>` | `<div data-hd-class="keydef">` | **MDITA (extended profile)** `<div data-hd-class="keydef">` in HDITA syntax |
| Link text | `<linktext>` | `<var data-hd-class="linktext">` | **MDITA (extended profile)** `<var data-hd-class="linktext">` |
| List item | `<li>` | `<li>` | ' -, +, or * for ul, and 0-9 and . or ) for ol |
| Map | `<map>` | `<nav>` | Not applicable |
| Note | `<note>` | `<div data-hd-class="note">` | **(MDITA extended profile)**`<div data-hd-class="note">`in HDITA syntax |
| Ordered list | `<ol>` | `<ol>` | See list item |
| Paragraph | `<p>` | `<p>` | Two carriage returns |
| Navigation title | `<navtitle>` | Not applicable | Not applicable |
| Phrase | `<ph>` | `<span>` | **(MDITA extended profile)**`<ph>` in XDITA syntax |
| Preformatted text | `<pre>` | `<pre>` | ```text``` |
| Prolog | `<prolog>` | `<meta>` inside `<head>` | Provided in YAML header |
| Section | `<section>` | `<section>` | ## or ----- underline |
| Short description | `<shortdesc>` | Implied in first paragraph | Implied in first paragraph |
| Table | `<simpletable>` | `<table>` | Tables in MDITA follow the GitHub Flavored Markdown syntax. See |

| Component | XDITA | HDITA | MDITA |
|---|---|---|---|
| | | | section 4.10 of the GFM spec |
| Simple table entry | `<stentry>` | `<th>` for headers and `<td>` for normal entries | See Table |
| Simple table header | `<sthead>` | `<tr>` | See Table |
| Simple table row | `<strow>` | `<tr>` | See Table |
| Subscript | `<sub>` | `<sub>` | **(MDITA extended profile)**`<sub>` in HDITA syntax |
| Superscript | `<sup>` | `<sup>` | **(MDITA extended profile)**`<sup>` in HDITA syntax |
| Title | `<title>` | `<h1>` for topic<br>`<h2>` for section | # or === underline for topic<br>## or ----- underline for section |
| Topic | `<topic>` | `<article>` | Not applicable |
| Topic metadata | `<topicmeta>` | Not applicable | Not applicable |
| Topic reference | `<topicref>` | `<a href>` inside a `<li>` | [link](/URI "title") inside a list item |
| Underline | `<u>` | Not applicable | Not applicable |
| Unordered list | `<ul>` | `<ul>` | See List item |

## Appendix A.2 New elements

This topic lists the new XML elements that are part of LwDITA and how to represent them in XDITA and HDITA. These new elements are not available in the MDITA core profile and, if needed, can be represented with their raw HDITA equivalents as part of the MDITA extended profile.

| Component | XDITA | HDITA |
|---|---|---|
| Audio | `<audio>` | `<audio>` |
| Controls | `<controls>` | `@controls` in `<audio>` or `<video>` |
| Fallback | `<fallback>` | `<source>` |
| Poster | `<poster>` | `@poster` in `<video>` |
| Source | `<source>` | `<source>` |

| Component | XDITA | HDITA |
|---|---|---|
| Track | `<track>` | `@track` in `<audio>` or `<video>` |
| Video | `<video>` | `<video>` |

## Appendix A.3 DITA 1.3 attributes in LwDITA

This topic lists the DITA 1.3 attributes that are available in LwDITA. It also lists how to represent them in XDITA and HDITA. With the exception of key reference, these attributes are not available in the MDITA core profile and, if needed, can be represented with their raw HDITA equivalents as part of the MDITA extended profile.

| Component | XDITA | HDITA |
|---|---|---|
| Content reference | `@conref` | `@data-hd-conref` |
| Direction | `@dir` | `@dir` |
| Expanse | `@expanse` | Not applicable |
| Frame | `@frame` | Not applicable |
| Identifier | `@id` | `@id` |
| Importance | `@importance` | `@data-hd-importance` |
| Key reference | `@keyref` | `@data-hd-keyref` |
| Language | `@xml:lang` | `@lang` |
| Output class | `@outputclass` | `@class` |
| Props | `@props` | `@data-hd-props` |
| Scale | `@scale` | Not applicable |
| Translate | `@translate` | `@translate` |
| Type | `@type` | `@data-hd-type` |

### *Reuse attribute in MDITA*

In an MDITA core-profile topic, a key reference is represented using the GitHub Flavored Markdown syntax for shortcut reference links: `[key-value]`. There is no equivalent for content reference in the MDITA core profile.

# Appendix B Acknowledgments

The following individuals participated in the creation of this document and are gratefully acknowledged.

Robert D. Anderson, IBM
Jan Benedictus, Fonto Group BV
Deb Bissantz, Healthwise
Stan Doherty, Individual member
Kristen James Eberlein, Eberlein Consulting LLC
Carlos Evia, Virginia Tech
Mark Giffin, Individual member
Tim Grantham, Precision Content Authoring Solutions Inc.
Nancy Harrison, Individual member
Alan Houser, Individual member
Ullakaisa Kalander, Citec
Eliot Kimber, Individual member
Michael Priestley, IBM
Keith Schengili-Roberts, IXIASOFT

In addition, the OASIS DITA Technical Committee also would like to recognize the following people for their insights and support:

Jarno Elovirta
Kevin Lewis

# Appendix C Revision history

The following table contains information about revisions to this document.

| Revision | Date | Editor | Description of changes |
|---|---|---|---|
| 01 | 5 November 2016 | Carlos Evia | Created stub files for working draft. |
| 02 | 06 December 2016 | Kristen James Eberlein | Generated working draft 01 |
| 03 | 29 December 2016 | Kristen James Eberlein | • Edits to appendix A<br>• Generated working draft #2 |
| 04 | 23 January 2017 | Kristen James Eberlein | Generated working draft #3 |
| 05 | 30 January 2017 | Carlos Evia | Generated working draft #4 |
| 06 | 2 February 2017 | Carlos Evia | Generated working draft #5 |
| 07 | 6 February 2017 | Carlos Evia | Generated working draft #6 |
| 08 | 16 February 2017 | Carlos Evia | Generated working draft #7 |
| 09 | 21 February 2017 | Carlos Evia | Generated working draft #8 |
| 10 | 8 March 2017 | Carlos Evia | Separated MDITA in core and extended profiles. Generated Working Draft 09 |
| 11 | 20 March 2017 | Carlos Evia | Generated working draft #10 |
| 12 | 8 May 2017 | Carlos Evia | Incorporated feedback from internal SC review and generated Working Draft 11 |
| 13 | 25 May 2017 | Carlos Evia/K. Eberlein | Made content and editorial changes after call with K. Eberlein. Generated working draft 12. |
| 14 | 25 May 2017 | Kristen James Eberlein | High-level edit to enforce consistent terminology and usage. Generated working draft #13. |
| 15 | 29 May 2017 | Kristen James Eberlein | Substantial rework of 2 "Why Lightweight DITA and its children." Generated Working Draft 14. |
| 16 | 29 May 2017 | Kristen James Eberlein | Incorporated material from Michael Priestley and generated Working Draft #15. |
| 17 | 02 June 2017 | Kristen James Eberlein | Edited footnote topic. Generated working draft #16. |

| Revision | Date | Editor | Description of changes |
|---|---|---|---|
| 18 | 05 June 2017 | Carlos Evia | Generated working draft #17 to consideration by the DITA TC. |