OASIS Committee Note

Lightweight DITA: An Introduction Version 1.0

Committee Note 01

10 April 2018

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:
http://docs.oasis-open.org/dita/LwDITA/v1.0/cnprd02/LwDITA-v1.0-cnprd02.html (Authoritative version)
http://docs.oasis-open.org/dita/LwDITA/v1.0/cnprd02/LwDITA-v1.0-cnprd02.pdf

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
Alan Houser (arh@groupwellesley.com), Individual member

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.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) Version 1.3 Errata 01*. OASIS Approved Errata. http://docs.oasis-open.org/dita/dita/v1.3/errata01/os/dita-v1.3-errata01-os.html
- *Darwin Information Typing Architecture (DITA) Version 1.3. Part 0: Overview Plus Errata 01*. OASIS Standard incorporating Approved Errata. http://docs.oasis-open.org/dita/dita/v1.3/errata01/os/complete/part0-overview/dita-v1.3-errata01-os-part0-overview-complete.html.
- *Darwin Information Typing Architecture (DITA) Part 1: Base Edition Plus Errata 01*. OASIS Standard incorporating Approved Errata. http://docs.oasis-open.org/dita/dita/v1.3/errata01/os/complete/part1-base/dita-v1.3-errata01-os-part1-base-complete.html.
- *Darwin Information Typing Architecture (DITA) Part 2: Technical Content Edition Plus Errata 01*. OASIS Standard incorporating Approved Errata. http://docs.oasis-open.org/dita/dita/v1.3/errata01/os/complete/part2-tech-content/dita-v1.3-errata01-os-part2-tech-content-complete.html.
- *Darwin Information Typing Architecture (DITA) Part 3: All-Inclusive Edition Plus Errata 01*. OASIS Standard incorporating Approved Errata. http://docs.oasis-open.org/dita/dita/v1.3/errata01/os/complete/part3-all-inclusive/dita-v1.3-errata01-os-part3-all-inclusive-complete.html.

Abstract:

Lightweight DITA (LwDITA) is a simplified version of DITA. In comparison to DITA 1.3, LwDITA has a smaller element type and attribute set, stricter content models, and a reduced feature set. LwDITA also defines 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, Kristen James Eberlein, and Alan Houser. 10 April 2018. OASIS Committee Note 01. 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

Copyright © OASIS Open 2018. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Table of contents

1 Introduction

Lightweight DITA (LwDITA) is a simplified version of the Darwin Information Typing Architecture (DITA). In comparison to DITA 1.3, LwDITA has a smaller element type and attribute set, stricter content models, and a reduced feature set. LwDITA also defines 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

Note:

Lightweight DITA is a work in progress. This committee note outlines the current plans in order to gain design clarity and receive feedback from potential users and implementers. Please note that details might change between the publication of this committee note and the actual release of the Lightweight DITA standard.

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: https://www.slideshare.net/mpriestley/crossformat-content-with-lightweight-dita.

[LwDITA-Easy-Way]

Authoring Standards-Based Intelligent Content the Easy Way with Lightweight DITA. Authored by Carlos Evia. *Proceedings of the 35th ACM International Conference on the Design of Communication* (August 2017). https://dl.acm.org/citation.cfm?id=3121204.

[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

[LwDITA-overview]

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/.

[Markdown-Extra]

PHP Markdown Extra. https://michelf.ca/projects/php-markdown/extra/

[Markingdown-DITA]

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

[Pandoc]

Pandoc. http://www.pandoc.org/

[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.

core profile

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

custom data attributes

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

extended profile

(MDITA) The authoring profile that relies on specific Markdown variants to access DITA features such as the `@id` attribute on the root element, prolog metadata, definition lists, footnotes, and optional use of HTML element types.

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.

XDITA

The LwDITA authoring format that is based on XML.

2 Why Lightweight DITA?

Lightweight DITA is a standards-based alternative for situations in which DITA 1.3 would be too complex or for communities that 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, most are vendor-developed and proprietary. A standards-based lightweight alternative will enable the DITA community to offer a common starting point for simple DITA scenarios that remains fully compatible with DITA 1.3.

Some authoring communities have strong ties to specific formats, such as Markdown or HTML. While these 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. Lightweight DITA defines a lower-function level of interchange and mappings for HTML5 and Markdown, thus becoming the first version of DITA to be truly cross-format —allowing authoring and delivery in a mix of native formats that are 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 common ground for the functionality that is needed by the following authoring communities: learning and training, software documentation authored by subject matter experts (SMEs), and marketing content.

3 What is Lightweight DITA?

LwDITA is a proposed standard 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 exchange and publish content whose source exists in these different markup languages
- Foster the growth of new, low-cost tools and applications that support LwDITA

LwDITA is not a replacement for DITA 1.3. Organizations and teams that are already using DITA are encouraged to explore LwDITA, but they are not the primary audience for this proposed lightweight standard. Organizations and individuals that have not adopted DITA, either because XML is not a tool used in their professional communities or they are not familiar with information typing, can rely on LwDITA as their introduction to structured authoring and content reuse.

LwDITA is intended to be a conforming subset of DITA 1.3. In order to make this possible, the DITA Technical Committee will release a new multimedia domain for use with DITA 1.3.

3.1 Simplified structure

DITA 1.3 has more power (and thus complexity) than is needed in some authoring 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 element types, attributes, features, and complexity.

Conference presentations and practitioners' blogs occasionally describe DITA as an intimidating grammar 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 48 element types. 39 of the element types are defined in DITA 1.3, and the other 9 are multimedia element types that are part of a forthcoming domain intended for use with DITA 1.3.

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 others familiar with DITA can author in XML. Documents authored in the various authoring formats can be aggregated 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 mappings, for example, between DITA and JSON, AsciiDoc, or MS Word.

The XDITA and HDITA content models are designed to be functionally equivalent to 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 extensions from Markdown Extra (definition lists and footnotes), Pandoc (YAML front matter headers) and HDITA element types and attributes to overcome Markdown limitations as a language for authoring structured and reusable content.

3.3 Development of LwDITA tools and applications

The DITA Technical Committee hopes 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 components that provide increased multimedia support.

4.1 Components of the LwDITA topic

LwDITA uses a subset of the topic element types that are available in DITA 1.3.

The subset was carefully chosen to include only the most basic constructions that are needed 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 selected subset contains the following document components:

- Body
- Cross reference
- Data
- Description
- Figure
- Footnote
- Image and alternate text
- In-line formatting: Bold, italics, underline, subscript, superscript
- Lists: Definition list, 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 element types that are included in LwDITA and their availability in the authoring formats, see DITA 1.3 element types in LwDITA **(23)**.

4.2 Components of the LwDITA map

LwDITA uses a subset of the map element types that are available in DITA 1.3.

The selected subset contains the following map components:

- Data
- In-line formatting: Bold, italics, underline, superscript, subscript
- Key definition
- Link text
- Map
- Navigation title

- Phrase
- Topic metadata
- Topic reference

For a complete list of the DITA 1.3 element types that are included in LwDITA and their availability in the authoring formats, see DITA 1.3 element types 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 can appear:

- Bold
- Italics
- Phrase
- Superscript
- Subscript
- Underline

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 from most of the block contexts that are available in LwDITA.

4.4 Subset of reuse mechanisms

LwDITA offers a subset of the reuse mechanisms that are available in DITA 1.3.

Conditional processing

The only conditional processing attributes are `@props` (in XDITA) or `@data-props` (in HDITA and MDITA extended profile).

Content reference

The `@conref` (in XDITA) or `@data-conref` (in HDITA and MDITA extended profile) attribute is available on the following document components:

- 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 the MDITA core profile.

Key reference

The `@keyref` (in XDITA) or `@data-keyref` (in HDITA) or `[keyref]` (in MDITA extended profile) attribute can be used on phrase (XDITA) or span (HDITA). It is also available on links, alternative text, and data.

Linking

The URI-based and indirect key-based addressing mechanisms from DITA 1.3 are available in LwDITA.

Variable text

For variable text, such as product names, authors can use key references on phrase (XDITA) or span (HDITA).

This design simplifies the authoring experience, as there are no choices to be made. To reuse block-level content, authors will use content reference (`@conref` or `@data-conref`). For phrase-level content, authors will use key reference (`@keyref` or `@data-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 components

LwDITA adds new element types for multimedia content. These element types are compatible with HTML5; they are part of a forthcoming domain intended for use with DITA 1.3.

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 type to include multimedia content in a topic, pointing out that it corresponds to the `<object>` element type in HTML.

However, HTML5 introduced direct element types for audio and video. LwDITA updates the XML-to-HTML element type correspondence and introduces the following multimedia components, which are specialized from the DITA 1.3 `<object>` and `<param>` element types:

Audio

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

Autoplay

    Autoplay determines if audio or video content should automatically begin to play.

Controls

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

Loop

    Loop automatically returns to the start of audio or video content upon reaching its end.

Muted

    Muted indicates if the audio of a media object will be silenced or not.

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

The DITA Technical Committee is working on a multimedia domain add-on for DITA 1.3 that would include some of these element types to maintain compatibility between DITA and LwDITA.

4.6 LwDITA specialization

LwDITA follows the same specialization architecture as DITA 1.3, although there are some limitations and special rules.

Because LwDITA is a proposed standard that spans multiple authoring formats, coordination of the same specialization rules across markup languages poses some unique challenges. Not all LwDITA formats will support specialization to the same degree.

- In XDITA, authors can build new element types and attributes following the base architecture of LwDITA. Authors cannot add element types from DITA 1.3 to an XDITA specialization. For example, authors working in LwDITA cannot create a `<training-video>` element type that is specialized from the DITA 1.3 element type `<object>`. They must specialize it from the XDITA element type `<video>`.
- In HDITA, an author can express specialized relationships for any content element type using HTML5 custom data attributes.
- In MDITA, the lack of structuring tags does not allow authors to assign reliable attributes to particular content components. As a result, specialization within MDITA is limited to a subset of use cases (for example, section-level specialization) using raw HDITA syntax.

A general recommendation for LwDITA specializations is to keep in mind the *lightweight* nature of the proposed standard and avoid complicated content structures. Authors who need robust specialization for complex scenarios should use DITA 1.3.

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 element types added to support interoperability with HTML5.

5.1.1 Audience for XDITA

XDITA is designed for users who want to write DITA content but who do not want (or need) the full power of DITA.

Potential users of XDITA 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 element types, note the new `<video>` element type.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE topic PUBLIC "-//OASIS//DTD LIGHTWEIGHT DITA Topic//EN" "lw-topic.dtd">
<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>
        <video-poster value="remote-poster.jpg" />
        <media-controls />
        <media-source value="remote.mp4" />
      </video>
    </section>
  </body>
</topic>
```

XDITA topics are 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.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE map PUBLIC "-//OASIS//DTD LIGHTWEIGHT DITA Map//EN" "lw-map.dtd">
<map id="remote-main">
    <topicmeta>
        <navtitle>Remote Lighting Network</navtitle>
    </topicmeta>
    <keydef keys="product-name">
        <topicmeta>
            <linktext>Remote Network Lighting</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>
```

Note that XDITA requires a `<navtitle>` inside a `<topicmeta>` to declare a map's title. This decision eliminates the need for `<title>` as a single-purpose element in maps, and also keeps the `<topicmeta>` options parallel for maps and topics.

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 for users who want to use HTML-authoring tools to write structured content.

Potential users of HDITA 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 HTML-authoring tools
- 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
- Authors who want to write content for the Web that does not require a transformation process
- Non-English-speaking content creators who are comfortable with HTML5 semantic elements

5.2.2 Example of an HDITA topic

The following topic is authored in HDITA. The topic uses HTML5 element types and custom data attributes for content reuse and compatibility with DITA. The custom data attribute includes a content reference from a DITA topic with a disclaimer expected from all topics in this fictional scenario.

```
<!DOCTYPE html>
<html>
<head>
  <title>Installing and Setting up Remote Lighting</title>
</head>
<body>
  <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"></video>
    <p data-conref="bulbs-to-groups.dita#bulbs-to-groups/assign-disclaimer"></p>
  </article>
</body>
</html>
```

5.2.3 Example of an HDITA map

An HDITA map is authored in HTML5.

```
<!DOCTYPE html>
<html>
  <title>Remote Lighting Network</title>
<nav>
    <h1>Remote Lighting Network</h1>
    <div class="keydef">
    <span class="linktext" data-keys="product-name">Remote Lighting Network</span>
    </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>
   </html>
```

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 markup 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 for users who want to write 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 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
- Developers and writers in charge of documenting application programming interfaces (APIs) that need to share content with technical publications
- 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
- Non-English-speaking authors who want to take advantage of DITA reuse and publishing without depending on XML tags written in English

Potential users of the MDITA extended profile 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 information structures that are readily available in Markdown:

- Title
- Paragraph
- Section title
- Ordered list
- Unordered list
- Link
- Image
- Preformatted text
- Italics
- Bold

- 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.

## Suggested Steps

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

## Example

 ![Remote installation example](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. In the previous example, the generated `@id` for the topic would be `installing-and-setting-up-remote-lighting`. The same process applies to sections inside the topic. For example, the `@id` for the section titled "Suggested Steps" would be `suggested-steps`.

MDITA extended profile

The MDITA extended profile allows the following components 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.
- Extensions, adapted from diverse Markdown flavors, that enable components like definition list and footnote
- Optional raw HDITA attributes and element types. 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, a definition list, a footnote, and an HDITA element type that enables the topic to reference a video.

```
---
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.

## Important Terms

Circuit panel
:    Also known as the breaker panel, it represents the main point of distribution circuits in
a building.

Lighting Kit[^1]
```

```
:   The commercial product covered by this information materials.

[^1]:
    This document occasionally uses the term "lighting kit" (all lowercase letters) for
generic statements that do not apply exclusively to your product.

## Example

The following video demonstrates a recommended installation:

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

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 syntax 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, MDITA, or DITA 1.3 and then publish them as a unified collection. With a combination of formats that includes the MDITA extended profile, authors can even use 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`/`@data-conref` and `@keyref`/`@data-keyref` mechanisms. The example contains the following:

• An XDITA 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 reuses content from an MDITA topic
• An HDITA topic, created by a marketing specialist, that reuses content from an XDITA topic
• An extended-profile MDITA topic, created by a software developer, that reuses 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 name.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE map PUBLIC "-//OASIS//DTD LIGHTWEIGHT DITA Map//EN" "lw-map.dtd">
<map id="cross-format">
    <topicmeta>
        <navtitle>Remote Lighting Setup</navtitle>
    </topicmeta>
    <keydef keys="product-name">
        <topicmeta>
            <linktext>Remote Network Lighting</linktext>
        </topicmeta>
    </keydef>
    <topicref href="xdita-topics/bulbs-to-groups.dita" format="dita"/>
    <topicref href="hdita-topics/low-power.html" format="hdita"/>
    <topicref href="mdita-topics/basic-concepts.md" format="mdita"/>
```

```
    <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.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE topic PUBLIC "-//OASIS//DTD LIGHTWEIGHT DITA Topic//EN" "lw-topic.dtd">
<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">
<p>Your <span data-keyref="product-name"></span> 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>
<div data-class="note">
  <p id="disconnect-warning">Even in low power networks, be sure to
 disconnect all devices before performing maintenance tasks.</p>
</div>
  </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 data-conref="low-power.html#low-power/disconnect-warning"></p>
```

6 LwDITA tools

Although many of the LwDITA elements and workflows proposed in this document are still experimental, tools already exist to support organizations who want to explore using LwDITA.

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.

The Lightweight DITA subcommittee maintains a wiki page with a list of LwDITA tools and resources. The page can be accessed at https://wiki.oasis-open.org/dita/LightweightDITASubcommittee/lwditatools

Tool developers interested in having resources listed on the wiki page should email the Lightweight DITA subcommittee at dita-lightweight-dita-chair@lists.oasis-open.org

Appendix A LwDITA components

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

Appendix A.1 DITA 1.3 element types in LwDITA

This topic lists the DITA 1.3 element types 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>`	No explicit markup
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 block in the file and must be set between triple-dashed lines.
Definition description	`<dd>`	`<dd>`	See definition list
Definition list entry	`<dlentry>`	Possible with a combination of data attributes[1]	See definition list
Definition term	`<dt>`	`<dt>`	See definition list
Definition list	`<dl>`	`<dl>`	**(MDITA extended profile)** "A single-line term followed by a colon and the definition for that term," following the Markdown Extra syntax for definition list
Description	`<desc>`	`<caption>` in `<table>`; `<figcaption>` in `<figure>`; not applicable in links	Not applicable
Figure	`<fig>`	`<figure>`	Not applicable

---

[1] Although the XDITA element type `<dlentry>` cannot be mapped directly to HTML5, an author can preserve the structure and attributes of a definition list in HDITA and MDITA with custom data attributes

Component	XDITA	HDITA	MDITA
Footnote	`<fn>`	`<span class="fn">`	**(MDITA extended profile)** "A footnote is made of two things: a marker in the text that will become a superscript number; a footnote definition that will be placed in a list of footnotes at the end of the document," following the Markdown Extra syntax for footnotes
Image	`<image>`[2]	`<img>`	![alt text for an image] (images/image_name.jpg)
Italics	`<i>`	`<em>`	* or _
Key definition	`<keydef>`	`<div data-class="keydef">`	**MDITA (extended profile)** `<div data-class="keydef">` in HDITA syntax
Link text	`<linktext>`	`<span data-class="linktext">`	**MDITA (extended profile)** `<span data-class="linktext">`
List item	`<li>`	`<li>`	' -, +, or * for ul, and 0-9 and . or) for ol
Map	`<map>`	`<nav>`	See Example of an MDITA map **(19)**
Note	`<note>`	`<div data-class="note">`	Not applicable
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)** `<span>` in HDITA syntax
Preformatted text	`<pre>`	`<pre>`	Fenced code blocks (e.g. ```text```) or indented code blocks (e.g. 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

---

[2] In XDITA, `<image>` is always treated as an inline element; an `<image>` inside a `<fig>` is treated as a block element

Component	XDITA	HDITA	MDITA
Table	`<simpletable>`	`<table>`	Tables in MDITA follow the GitHub Flavored Markdown syntax. See 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>` and `<title>` for topic[3] `<h2>` for section	# or === underline for topic ## or ----- underline for section
Topic	`<topic>`	`<article>`	No explicit markup
Topic metadata	`<topicmeta>`	Not applicable	Not applicable
Topic reference	`<topicref>`	`<a href>` inside a `<li>`	[link](/URI "title") inside a list item
Underline	`<u>`	`<u>`	Not applicable
Unordered list	`<ul>`	`<ul>`	See List item

Note: HDITA is a subset of HTML5 conforming with the W3C HTML standard, and MDITA aligns with the Github Flavored Markdown specification with some extensions from Markdown Extra and Pandoc. Instances of valid HTML5 syntax and Markdown practices allowed in other flavors outside the proposed LwDITA elements can be supported by vendors at their discretion. Those syntax components would be handled differently by vendors and might not work across all LwDITA implementations.

Appendix A.2 New element types

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

Component	XDITA	HDITA
Audio	`<audio>`	`<audio>`

---

[3] In order to generate valid DITA XML and HTML5, the XDITA element type `<title>` should map to both `<title>` and `<h1>` in HDITA.

Component	XDITA	HDITA
Autoplay	`<media-autoplay>`	`@autoplay` in `<audio>` or `<video>`
Controls	`<media-controls>`	`@controls` in `<audio>` or `<video>`
Loop	`<media-loop>`	`@loop` in `<audio>` or `<video>`
Muted	`<media-muted>`	`@muted` in `<audio>` or `<video>`
Poster	`<video-poster>`	`@poster` in `<video>`
Source	`<media-source>`	`<source>`
Track	`<media-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 and how to represent them in XDITA and HDITA.

Component/Set	XDITA	HDITA
Architecture attributes		
ditaarch	`@xmlns:ditaarch`	Not applicable
DITAArchVersion	`@ditaarch:DITAArchVersion`	Not applicable
Domains	`@domains`	Not applicable
Localization attributes		
Direction	`@dir`	`@dir`
Language	`@xml:lang`	`@lang`
Translate	`@translate`	`@translate`
Data definition		
Name	`@name`	Not applicable
Value	`@value`	Not applicable
Figure display attributes		
Expanse	`@expanse`	Not applicable
Frame	`@frame`	Not applicable
Scale	`@scale`	Not applicable
Filtering attribute		
Props	`@props`	`@data-props`
Footnote control		

Component/Set	XDITA	HDITA
Callout	`@callout`	Not applicable
Image size		
Height	`@height`	`@height`
Width	`@width`	`@width`
Note type		
Type	`@type`	`@data-type`
Processing attribute		
Output class	`@outputclass`	`@class`
Reference attributes		
Link target	`@href`	`@href`
Format of target resource	`@format`	`@type`
Processing role	`@processing-role`	`@data-processing-role`
Relationship of source to target	`@scope`	`@rel`
Reuse attributes		
Identifier	`@id`	`@id`
Content reference	`@conref`	`@data-conref`
Key definition	`@keys`	`@data-keys`
Key reference	`@keyref`	`@data-keyref`
Video size		
Height	`@height`	`@height`
Width	`@width`	`@width`

Representing attributes in MDITA

With the exception of key reference, attributes are not available in the MDITA core profile. In the MDITA extended profile, you can express attributes using their HDITA representation.

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, Vasont Systems
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.
Richard Hamilton, Individual member
Nancy Harrison, Individual member
Alan Houser, Individual member
Scott Hudson, The Boeing Company
Ullakaisa Kalander, Citec
Eliot Kimber, Individual member
Tom Magliery, JustSystems Canada
Chris Nitchie, Oberon Technologies
Michael Priestley, IBM
Keith Schengili-Roberts, IXIASOFT
Dawn Stevens, Comtech Services, Inc.
Bob Thomas, Individual member
Leigh White, IXIASOFT

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

Jarno Elovirta
Roger Hadley
Kevin John
Kevin Lewis
Scott Prentice
Roger Sheen

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.
02	06 December 2016	Kristen James Eberlein	Generated working draft #1
03	29 December 2016	Kristen James Eberlein	Edits to appendix A. 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 #9.
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	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.
18	05 June 2017	Carlos Evia	Generated working draft #17 for consideration by the DITA TC.
19			Generated working draft #18. It includes the following changes:

Revision	Date	Editor	Description of changes
			• Updated Acknowledgments
20	26 June 2017	Carlos Evia	Added <footnotes> to table of elements
21	26 July 2017	Carlos Evia	Improved footnote topic and examples
22	17 August 2017	Carlos Evia	Removed proposed footnote element/ behavior and modified multimedia elements based on multimedia domain proposal
23	21 August 2017	Carlos Evia	Added specialization topic; generated working draft #19
24	19 September 2017	Carlos Evia	Incorporated subcommittee feedback; generated working draft #20
25	9 October 2017	Carlos Evia	Incorporated TC feedback. Added Alan Houser to list of editors. Generated working draft #21
26	17 October 2017	Carlos Evia	Incorporated TC feedback. Generated Committee Note 01 version
27	12 January 2018	Carlos Evia	Incorporated feedback from public review #1
28	08 February 2018	Carlos Evia	Incorporated changes for public review #2
29	27 March 2018	Carlos Evia	Generated Committee Note 01 version 1