



Adobe FrameMaker

Getting Started with LwDITA



September 2022

Contents

- Getting Started with Lightweight DITA in Adobe FrameMaker 1**

- The ini file 2**

- LwDITA First-Time Setup 3**
 - Extraction Guidelines 3
 - .ini Setup 3
 - Environment Setup 3
 - Application File 3

- Elements and Attributes 4**
 - Elements 4
 - Attributes 5
 - DITAVAL 6
 - Graphics 7
 - Lists 8
 - Unordered list (ul) 8
 - Ordered list (ol) 8
 - Definition list (dl) 8
 - Nesting lists 9
 - Maps 10
 - Note 10
 - Section 11
 - Shortdesc 11
 - Simple Table 12
 - Topicref 16
 - Cross-references 16
 - Section-level xref content 16

Appearance of xref content	16
--------------------------------------	----

Getting Started with Lightweight DITA in Adobe FrameMaker

Lightweight DITA 1.0 (LwDITA_1.0) is a simplified version of DITA supported by FrameMaker.

We include full support for the XML environment with a clean, simple interface, albeit with a more restrictive set of elements than DITA 1.3 provides. Our goal is to allow authors to collaborate and publish content using a lighter and faster DITA environment.

The supporting documents include this Getting Started Guide as well as multiple files and folders to guide you in learning more about using LwDITA_1.0.

The ini file

Add these lines into the C:\Program Files\Adobe\Adobe FrameMaker 2022\fm\init\ditafm\ditafm.ini file.

They are broken out here to make it very easy to select required core content.

```
[LW_DITA_Applications]
_SectionType=APP_MAPPING_DITA_1_3
topic=          LwDITA_1.0_topic          |TOPIC
map=            LwDITA_1.0_map            |MAP
```

LwDITA First-Time Setup

Instructions for installing LwDITA.

Extraction Guidelines

After receiving and unzipping the LwDITA file, ensure that the **LwDITA_1.0** folder is located in *C:\Program Files\Adobe\Adobe FrameMaker 2022\Structure\xml*; move it to this location if necessary.

.ini Setup

Before using LwDita, the *ditafm.ini* file must be edited to include the following:

NOTICE: Type or enter the following code right before the [BUILD FILE] section in *ditafm.ini*

```
[LW_DITA_Applications]
_SectionType=APP_MAPPING_DITA_1_3
topic=          LwDITA_1.0_topic          |TOPIC
map=            LwDITA_1.0_map            |MAP
```

Environment Setup

- 1) From **Structure** menu, select **DITA > DITA Options...** to open the *DITA Options* dialog.
- 2) In the *DITA Options* dialog, select **LW_DITA_Applications** from the *Use Application Mapping* dropdown.
- 3) Click **Save**.

Application File

The application file is a collection of shortcuts to specific required files. Ensure that any variables in the file are correctly updated.

- 1) From **Structure** menu, select **Application Definition > Edit Global Application Definitions** to open the *structapps.fm* document.
- 2) Open the *structapps_DITA_LW_Only.fm* file.
- 3) Select the two **XMLApplication** elements.
- 4) Copy the applications to the *structapps.fm* document.
- 5) Click **Save** and re-read the applications into FrameMaker.

Elements and Attributes

LwDITA supports a large range of element and attribute combinations, which combine to create XML-based content that is richly formatted by FrameMaker.

While not all elements are listed below, the most common elements are. The list is generally in either the order or frequency of use. While not a complete list, it should get you started with LwDITA. The information provided here is lightly based on a far more comprehensive list found on the [OASIS LwDITA site](#).

Elements

The following is a partial list of elements and includes some of the most commonly used ones.

Element name and overview	Notes
<p>topic</p> <p>Content is generally contained in a topic. It's a file (like this document) in which all your content is written, and then stored as XML content.</p>	<p>Each topic is likely used in one or more map and inserted much like a section in a chapter or a chapter in a book.</p>
<p>title</p> <p>Introduces something with a paragraph of text that is often uniquely formatted.</p>	<p>Commonly used in a topic, table, figure, or section. The format for these can be contextually assigned based on the parent element. That is, a title in a table may read Table 1, but if used in a figure may read <i>Figure 1</i>. Format is secondary to the function of the title.</p> <p>The title may also be used to generate a table of contents, a list of tables, a list of figures, or other lists.</p> <p>A cross-reference (for example, "See Table 1 on page 23") often is to the parent of the title, but the text of the title displays.</p>
<p>shortdesc</p> <p>A quick and brief sentence that enhances the title; often used as the <i>hover text</i> on a link, or summarized into a list of topics when linked pages are built between primary and secondary materials.</p>	<p>Generally, try not to repeat the title in the shortdesc; instead use it to add value. If the title is Chair and Table you could add a sentence such as:</p> <ul style="list-style-type: none"> • The Chair can Table a motion. • Ensure the Chair is close enough to the Table for ideal ergonomics. <p>Each of these adds value to the title and provides some context about the topic.</p>

Element name and overview	Notes
<p>body Contains the core content (aside from the title, shortdesc, and prolog information) of the topic.</p>	<p>In the body you can add paragraphs, tables, images, sections, lists, and other content that is common to documentation.</p>
<p>p Paragraph is usually the most common element used and appears in many places. Within some elements, such as body, the p must be inserted in order to add content.</p>	<p>The paragraph can be inserted in the body, table cells, list items, and generally is used to type longer strings of text. The format of the element is contextual. That is, a paragraph can be in a list item and appear as a bullet (if in an unordered list), or can appear numbered (if in an ordered list), or can appear with a heading like Note, Caution, or Warning automatically added.</p>

Attributes

Attributes provide information about elements. For example, a Note element may have a type attribute. This allows the Note to be further defined as being a notice, tip, caution, or warning type.

The following is a partial list of attributes and includes some of the most commonly used ones.

Attribute name and overview	Notes
<p>dir Defines the direction of content.</p>	<p>Content can be defined as running left to right or right to left. Formatting can be assigned based on this.</p>
<p>xml:lang Defines the language of the content.</p>	<p>By defining the language, specific spell check and grammar rules may be assigned, punctuation or format rules assigned, or other properties given to content.</p>
<p>translate Defines if content should be translated or not.</p>	<p>If set to a value of yes or no, this allows translators to know if content should be translated. For example, a code sample may contain an instruction in English for a programmer, but in German for a dialog. Defining part of it to translate allows for the German to later be translated to, say, Spanish, but to leave the code components in English.</p>

Attribute name and overview	Notes
<p>outputclass Provides a format instruction to the element.</p>	<p>Whenever possible this should be assigned automatically by software. For example, tables may use outputclass to automatically assign table format. This very table has an outputclass in the source DITA which drives the format of the table when it is ingested into a template that understands the value of the attribute. Based on format rules of the templates that display or publish content the appearance of a range of materials can be broadly customized with the use of this attribute.</p>

DITAVAL

Conditional processing profile used to identify which values are to be used for conditional processing during a particular output, build, or some other purpose.

This conditional processing profile is identified with the extension `.ditaval`, and contains several elements:

Element	Description
val	root element; can contain a style-conflict element followed by prop or revprop elements
prop	can contain startflag / endflag elements
revprop	can contain startflag / endflag elements
startflag	can contain alt-text elements
endflag	can contain alt-text elements

NOTE: Conditional processing code should provide a report of any attribute values encountered in content that do not have an explicit action associated with them.

NOTICE: The following content is representative of a sample DITA val file.

```
<?xml version="1.0" encoding="UTF-8"?>
<val>
<prop action="exclude" att="product" val="ProductB" />
<prop action="exclude" att="audience" val="EndUser"/>
</val>
```

This val support publishing the ProductA for Admin users by suppressing two things:

- 1) Trigger an action to exclude anything where the attribute named *product* is set to *ProductB*.
- 2) As above for the attributes named *audience* when set to *EndUser*.

Graphics

Graphics can visually represent information to quickly show complex ideas.

Imagine trying to describe a company logo. You may need to define dimensions, file format, and placement. Defining the shapes, angles, colors, and other properties can become difficult to do.

Instead, a graphic can simply be referenced and inserted into a document. To do so the graphic must be imported and referenced. The code for this is relatively simple even for a complex image and, at a minimum, may be encoded as:

```
<image href="filename.ext"/>
```

This code defines an *element*, *attribute*, and *value*. This information is transformed by software for on-screen or print display. When shown in a tool that works with DITA, the image may appear as seen in the following samples, and might be inline, below a line, or even a part of a more complex figure with support for a title if required.

A sample image in a paragraph may be coded as: `<p><image href="images/logo.png"/></p>` which may display the image as seen here:



A sample image in a figure may be coded as: `<fig><title>Company Logo</title><image href="images/logo.png"/></fig>` which may display the figure, title, and image as seen here:

Figure 1: Adobe Technical Communication Suite Logo



Since both of the images reference the same logo, a single change to the image would update the representation in both instances. However, in the second instance the title remains the same. Generally, images can be updated as needed and the most current graphic always displays.

NOTE: It is generally suggested that standard image formats such as a jpg, png, gif, or svg is used when working with DITA to simplify publishing. However, source images in formats such as Adobe Photoshop or Adobe Illustrator should be maintained to enable editing of the images at any time.

Much like keeping DITA content but delivering a PDF, web content, or even a mobile app is common practice, so to is keeping the source graphics.

Lists

Content can be grouped into three list types, each of which has a unique purpose and format.

Unordered list (**ul**)

A list in which the order does not matter.

- Each item in a **ul** is considered equal.
- The reading order does not impact the understanding of the content.
- The visual representation of a **ul** may have bullets, dashes, or other symbols, but not numbers, alphabetical, or roman numeral initial characters as these are more indicative of an order.

Ordered list (**ol**)

A list in which the order matters. This type of list may represent the steps of a task or show a relationship between items where one is of a higher priority than another.

- 1) In order, the **ol** contains the root ordered list element.
- 2) Each **ol** must contain at least one list element (**li**) to represent the list item.
- 3) Each **li** must contain at least one element, most often a **p** element, to represent the paragraph of content that is typed into the numbered item.

Definition list (**dl**)

This is a single term, formatted to be followed by a colon and the definition for that term. For example, “Structured Content: definition of structured content”.

dl

Definition list is the starting point for a list of terms and definitions. It contains one or more **dlentry**.

dlentry

Definition list entry is a single entry for each definition in the list. One **dlentry** should be created for each term and definition combination.

dt

Definition term is the short initialism or acronym that is being defined.

dd

Definition description is the longer phrase that expands on the idea of the term and explains it to the reader.

Nesting lists

While DITA technically allows unlimited nesting, it is suggested nesting be limited to prevent excessively complex information from being too deeply nested and losing the clarity of a message.

- Each item in a **ul** is considered equal.
 - The reading order does not impact the understanding of the content.
 - The visual representation of a **ul** may have bullets, dashes, or other symbols, but not numbers, alphabetical, or roman numeral initial characters as these are more indicative of an order.
 - Each nested item in a **ul** is considered equal.
 - The reading order does not impact the understanding of the content.
 - The visual representation of a **ul** may have bullets, dashes, or other symbols, but not numbers, alphabetical, or roman numeral initial characters as these are more indicative of an order.
- 1) In order, the **ol** contains the root ordered list element.
 - 2) Each **ol** must contain at least one list element (**li**) to represent the list item.
 - 3) Each **li** must contain at least one element, most often a **p** element, to represent the paragraph of content that is typed into the numbered item.
 - a) In order, the nested **ol** contains the root ordered list element.
 - b) Each **ol** must contain at least one list element (**li**) to represent the list item.
 - c) Each **li** must contain at least one element, most often a **p** element, to represent the paragraph of content that is typed into the numbered item.
- Each item in a **ul** is considered equal.
 - The reading order does not impact the understanding of the content.
 - The visual representation of a **ul** may have bullets, dashes, or other symbols, but not numbers, alphabetical, or roman numeral initial characters as these are more indicative of an order.
 - a) In order, the nested **ol** contains the root ordered list element.
 - b) Each **ol** must contain at least one list element (**li**) to represent the list item.
 - c) Each **li** must contain at least one element, most often a **p** element, to represent the paragraph of content that is typed into the numbered item.
- 1) In order, the **ol** contains the root ordered list element.
 - 2) Each **ol** must contain at least one list element (**li**) to represent the list item.
 - 3) Each **li** must contain at least one element, most often a **p** element, to represent the paragraph of content that is typed into the numbered item.

- Each nested item in a **ul** is considered equal.
- The reading order does not impact the understanding of the content.
- The visual representation of a **ul** may have bullets, dashes, or other symbols, but not numbers, alphabetical, or roman numeral initial characters as these are more indicative of an order.
- Each item in a **ul** is considered equal.
- The reading order does not impact the understanding of the content.
- The visual representation of a **ul** may have bullets, dashes, or other symbols, but not numbers, alphabetical, or roman numeral initial characters as these are more indicative of an order.
 - Each nested item in a **ul** is considered equal.
 - The reading order does not impact the understanding of the content.
 - The visual representation of a **ul** may have bullets, dashes, or other symbols, but not numbers, alphabetical, or roman numeral initial characters as these are more indicative of an order.
 - It is **strongly** suggested that lists not exceed 2 levels of depth in your content.
 - Nesting lists too deeply risks alienating readers by drilling down too low into the specifics. It's better to work with clear topics instead and plan content.
 - Some software may configure lists nested too deeply with a unique format to flag it for review.

Maps

Documents that organize topics and other resources into structured collections of information.

Maps specify hierarchy and relationships among topics and provide the context in which keys are defined and resolved. Map documents should have the .ditamap file extension.

Maps use <topicref> elements to reference specific topics, maps, and non-DITA resources such as HTML or TXT files. The <topicref> elements can be nested or grouped to create relationships that exist among those topics. Because DITA maps provide this context for topics, topics themselves can be relatively context-free; they can be used and reused in multiple contexts.

In many cases a map, at publishing, will create linking structures between parent and child content automatically as well. Plan and manage maps well!

Note

Notes can be used to highlight important and/or tangential information.

The <note> element differentiates information from the main body of text in order to expand on or call attention to a specific point. It's a great feature to use in order to grab the reader's attention and ensure they see the information you want to highlight from or about your main body of text.

This is a normal body of text, containing information about the <note> element in LwDITA. It explains what the <note> element is used for and why it's a great feature to utilize in your authoring. If you want to draw the reader's attention to a specific piece of information, contain it in a note.

NOTE: This element designates information you want the reader to see, making it stand out from the other text on the page.

For example, the note and its content may be formatted uniquely in the template to display with a different standard from other paragraphs. This helps the note draw attention to it and make the reader take notice.

CAUTION: Notes have attributes that can drive their appearance. For example, setting the *type* attribute to *caution* may cause a note to look more like this content.

WARNING: Notes have attributes that can drive their appearance. For example, setting the *type* attribute to *warning* may cause a note to look more like this content.

DANGER: Notes have attributes that can drive their appearance. For example, setting the *type* attribute to *danger* may cause a note to look more like this content.

Section

A section is a component that provides a way to organize information within a topic.

The <section> element organizes related information within a topic. For example, a topic about the United States may have <section> elements for California, New York, and Florida. A topic about Colors may have <section> elements for Red, Blue, and Green.

Sections relate to each other as peers within a topic and cannot be nested; they may have separate titles, but this is not required.

Generally, use a section to organize information that may not be considered involved or complex enough to be a stand-alone topic. If you feel that sections need their own subordinate sections, consider your map and rework your content into topics.

Shortdesc

Located between the title and body, shortdesc provides a brief description of the topic.

The short description <shortdesc> element represents the purpose or theme of a topic and is also intended to be used as a link to preview and for searching. The <shortdesc> can also be used in a DITA map.

While inclusion of the <shortdesc> element is not mandated by DITA or the tools, it is recommended that topics contain this element. In cases where a topic only contains one paragraph, it is preferable to include this text in the <shortdesc> and leave the body empty.

Simple Table

This element inserts a very basic table structure which allows content in cells (using the <p> element). The following is a simple table. In some tools, the outputclass can be used to drive the format. This table has the outputclass set to RuledPrimaryShadeHead.

Table component	Format instructions
The body of the table contains a collection of stentry elements.	An stentry is formatted based on the parent. If in the body, it appears as seen here. If in the header, it has format as seen in the first row of the simpletable seen here.
stentry elements have a range of optional elements that can be inserted	Content such as a paragraph, list (including ordered, unordered, and definition), preformatted content, or note can be added.
stentry content is based on the same styles as content outside a table	The core difference is that the content in these cells is usually a bit smaller than the content outside a table. However, lists, paragraphs, notes, and all other content should look generally the same.
p elements	The p element is likely the most common content within the stentry. It is formatted like the traditional body text, but a point size smaller.
ul and ol elements	<p>Both list types are supported. Content is also reduce by one point size.</p> <ul style="list-style-type: none"> • unordered list content has the same bullet style • no numbering is used <ol style="list-style-type: none"> 1) Ordered list content has the same number style 2) Continuation of the numbering displays an incremental increase <p>dt</p> <p>Also reduces the size of the paragraph within the dd element</p> <p>dd</p> <p>The dd element requires a p element and that is already reduced in size within the table</p>

The following is a simple table. In some tools the outputclass can be used to drive the format. This table has the outputclass set to RuledSecondaryShadeHead.

Table component	Format instructions
The body of the table contains a collection of stentry elements.	An stentry is formatted based on the parent. If in the body it appears as seen here. If in the header it has format as seen in the first row of the simpleteable seen here.
stentry elements have a range of optional elements that can be inserted	Content such as a paragraph, list (including ordered, unordered, and definition), preformatted content, or note can be added.
stentry content is based on the same styles as content outside a table	The core difference is that the content in these cells is usually a bit smaller than the content outside a table. However, lists, paragraphs, notes, and all other content should look generally the same.
p elements	The p element is likely the most common content within the stentry. It is formatted like the traditional body text, but a point size smaller.
ul and ol elements	<p>Both list types are supported. Content is also reduced by one point size.</p> <ul style="list-style-type: none"> • unordered list content has the same bullet style • no numbering is used <ol style="list-style-type: none"> 1) Ordered list content has the same number style 2) Continuation of the numbering displays an incremental increase <p>dt</p> <p>Also reduces the size of the paragraph within the dd element</p> <p>dd</p> <p>The dd element requires a p element and that is already reduced in size within the table</p>

The following is a simple table. In some tools the outputclass can be used to drive the format. This table has the outputclass set to FormatA.

Table component	Format instructions
The body of the table contains a collection of stentry elements.	An stentry is formatted based on the parent. If in the body it appears as seen here. If in the header it has format as seen in the first row of the simpletable seen here.
stentry elements have a range of optional elements that can be inserted	Content such as a paragraph, list (including ordered, unordered, and definition), preformatted content, or note can be added.
stentry content is based on the same styles as content outside a table	The core difference is that the content in these cells is usually a bit smaller than the content outside a table. However, lists, paragraphs, notes, and all other content should look generally the same.
p elements	The p element is likely the most common content within the stentry. It is formatted like the traditional body text, but a point size smaller.
ul and ol elements	<p>Both list types are supported. Content is also reduced by one point size.</p> <ul style="list-style-type: none"> • unordered list content has the same bullet style • no numbering is used <ol style="list-style-type: none"> 1) Ordered list content has the same number style 2) Continuation of the numbering displays an incremental increase <p>dt</p> <p>Also reduces the size of the paragraph within the dd element</p> <p>dd</p> <p>The dd element requires a p element and that is already reduced in size within the table</p>

The following is a simple table. In some tools the `outputclass` can be used to drive the format. This table has the `outputclass` defaulting to no value. Based on your software this can have a default format defined.

Table component	Format instructions
The body of the table contains a collection of stentry elements.	An stentry is formatted based on the parent. If in the body it appears as seen here. If in the header it has format as seen in the first row of the simpletable seen here.
stentry elements have a range of optional elements that can be inserted	Content such as a paragraph, list (including ordered, unordered, and definition), preformatted content, or note can be added.
stentry content is based on the same styles as content outside a table	The core difference is that the content in these cells is usually a bit smaller than the content outside a table. However, lists, paragraphs, notes, and all other content should look generally the same.
p elements	The p element is likely the most common content within the stentry. It is formatted like the traditional body text, but a point size smaller.
ul and ol elements	<p>Both list types are supported. Content is also reduced by one point size.</p> <ul style="list-style-type: none"> • unordered list content has the same bullet style • no numbering is used <ol style="list-style-type: none"> 1) Ordered list content has the same number style 2) Continuation of the numbering displays an incremental increase <p>dt</p> <p>Also reduces the size of the paragraph within the dd element</p> <p>dd</p> <p>The dd element requires a p element and that is already reduced in size within the table</p>

Topicref

Identifies a topic (such as a concept, task or reference) or other resource

A <topicref> element can contain other <topicref> elements, allowing you to express navigation or table-of-contents hierarchies, as well as implying relationships between a containing (parent) <topicref> and its children.

Cross-references

A cross-reference might point to another location within a topic, between topics, or to external links such as a web page.

The cross-reference is also referred to as an xref element. Generally, an xref has a format associated with it. This is often stored as part of the outputclass attribute.

A sample xref in a paragraph may be coded to include the following:

- `<p>Refer to <xref href="#id_UniqueTopicID" /></p>` which may display as [Cross-references](#). Note that this xref uses the topic's id attribute to create a unique identifier.
- Links may also point to websites. [OASIS LwDITA site](#).

Section-level xref content

When adding any xref, do not link to the title. Instead, link to the parent topic or section that contains the title.

A sample xref in a paragraph may be coded to include the following:

- `<p>Refer to <xref href="#id_UniqueTopicID/id_UniqueSectionID" /></p>` which may display as [Section-level xref content](#). Note that this xref uses both the topic's and the section's id attributes to create a unique identifier.

Appearance of xref content

Generally the appearance of a link is likely in a unique format. Often it is displayed as blue text when published, but link formats can be any mix of character level properties.