User Guide: Authoring Structured Documentation in DITA Using Oxygen XML

Authored by: Seethalaxmi Palkumar

Published on: March, 2025

Contents

1.
Introduction

This user guide takes you through the process of getting started with structured documentation using DITA in Oxygen XML Editor. It covers the essentials, from setting up the editor to creating and managing structured content. With a focus on usability, this guide helps users navigate the interface, understand XML fundamentals, and work with DITA. Let's begin with an overview of structured documentation.

What is Structured Documentation?

Structured documentation is a systematic approach to content creation that organizes information using predefined rules and structures. Unlike traditional word processing, structured documentation adheres to specific frameworks that ensure consistency, modularity, and reusability. The logical structure of the document allows readers to become familiar and navigate through its pages without difficulty. Some examples of structured documentation include technical manuals, user guides, API documentation, and software documentation.

While there are many frameworks and standards that support structured documents, such as DocBook, AsciiDoc, reStructuredText, etc., this guide will explore DITA, an XML-based standard designed for technical documentation.

Why Use DITA for Technical Writing?

Darwin Information Typing Architecture (DITA) is an XML-based standard designed for authoring structured content. It offers:

1. Modular Content: Breaks information into reusable components.
2. Consistency: Standardized structure ensures uniform documents.
3. Multichannel Publishing: Easily converts to PDF, HTML, EPUB, and more.
4. Content Reuse: Avoids duplication by referencing existing content and simplifying updates.
5. Scalability: Manages large documentation projects efficiently.
6. Easy Maintenance: Updating one topic reflects everywhere it’s used.
7. Customization: Can be tailored for various types of documentation.
8. Collaboration: Works with CMS and version control for team workflows.
9. Compliance: Supports structured, audit-ready documentation.

DITA simplifies both the process for writers and the experience for readers, ensuring clear, accessible, and well-organized documentation.

Oxygen XML Editor

Oxygen XML Editor is a widely adopted tool for XML editing, authoring, and development. It provides a user-friendly interface with advanced features designed to enhance productivity. Supporting XML-based formats like DITA, DocBook, and TEI, it provides all the capabilities needed for DITA authoring while also offering advanced XML development and publishing functionalities, including real-time validation and error detection.

Who is This Guide For?

- Writers using Oxygen XML for the first time.
- Beginner technical writers looking to learn about DITA.
- Anyone trying to understand structural documentation.

Let’s get started by installing Oxygen XML Editor.

2.

Setting Up Oxygen XML for DITA

Installing and Launching Oxygen XML Editor

1. Visit the official Oxygen XML website.
2. Choose the appropriate version for your operating system (Windows, macOS, Linux, Eclipse, or Others).
3. Click on **Download** to get the installer for the selected version.
4. Run the installer and follow the on-screen instructions, including selecting the preferred language and accepting the license agreement.
5. Activate the license by entering the provided key or selecting a trial version if you're evaluating the software.
6. Copy and paste the **evaluation license key** sent to your email to activate the application.
7. Launch Oxygen XML Editor to begin using it.

Setting Up a DITA Project

1. Open Oxygen XML Editor.
2. Set the project type to **DITA** by selecting it from the top-right corner.
3. Create a new project:
   - Go to **File > New**.
   - Select **Project** as the file template.
   - Define the project name and location, then click **Create**.
4. Create a DITA Map within the project:
   - Go to **File > New** again.
   - This time, select **Map** as the file template.
   - Enter a name for the Map and click **Create**.

Figure 1. The project type must be set to DITA.

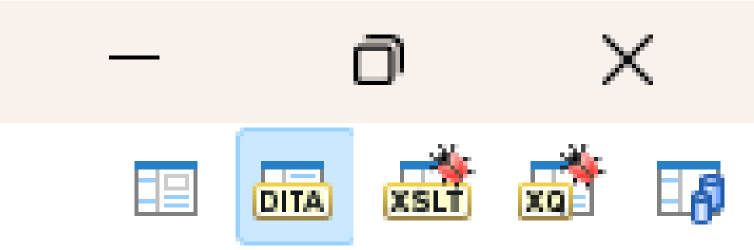
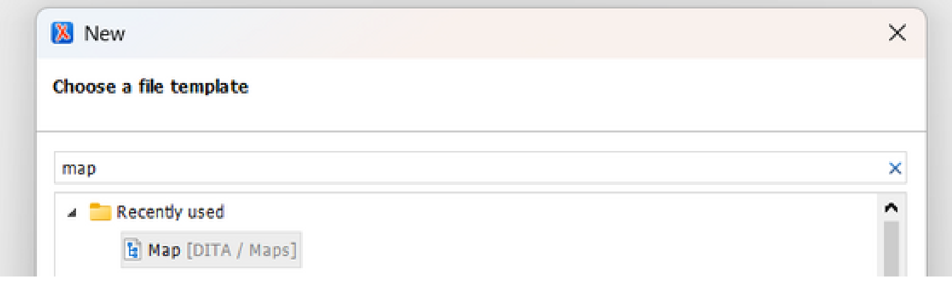


Figure 2. Select Map from the file template to create a DITA Map.



New

Choose a file template

map

Recently used

Map [DITA / Maps]

How to Manage DITA Maps?

A DITA map is a hierarchical structure that defines how topics are linked and displayed in output formats. It can be efficiently managed using the DITA Maps Manager, a key feature in Oxygen XML Editor that helps structure and control your documentation.

To open it, go to the Project tab on the left, right-click on a DITA map, select **Open With**, and choose **DITA Maps Manager**.

The DITA Maps Manager primarily allows you to:

- Add new or existing topics to the map.
- Drag and rearrange topics to modify their order and nesting levels.
- Preview content in various supported formats.

3.

Creating a New DITA Document

Difference between Concept, Task, and Reference Topics

To create a DITA document, it is important to understand the difference between the three types of topics.

Concept Topics	Task Topics	Reference Topics
• Concept topics provide background information, definitions, or explanations. • They're used to help readers understand the underlying ideas or context of a subject.	• Task topics offer step-by-step instructions for performing procedures or actions. • They're ideal for guides that explain how to complete specific tasks.	• Reference topics include detailed technical information, specifications, or supporting data. • They provide readers with essential reference material for further clarification.

Creating a New DITA Topic

1. Open the DITA map in the DITA Maps Manager.
2. Go to **File > New**, and select the appropriate DITA topic template (refer to the table above for Concept, Task, or Reference).
3. Enter a title and save the topic in the project directory with the .dita extension (e.g., my-topic.dita).
4. The new file automatically opens in the editor.
5. Write your content using predefined DITA XML elements, such as <title>, <shortdesc>, <p>, and <section>, to maintain structured formatting.
6. Use **Author Mode** for a visual editing experience or **Text Mode** for direct XML editing.
7. Save your work regularly to prevent data loss.

How to Append a Topic to a DITA Map?

1. Right-click on the DITA Map in the DITA Maps Manager.
2. Select **Append Child**.
3. If the topic was recently edited, choose **Reference to the currently edited file.**
4. Otherwise, click **Import** to browse and select the required topic file.

What Is XML Structure and Its Elements in a DITA File?

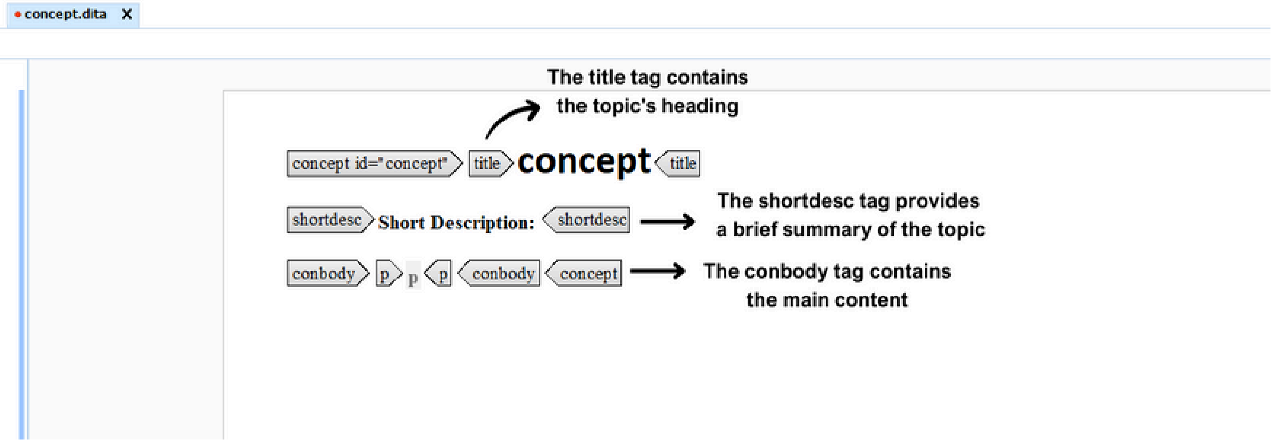
A DITA file follows a structured XML format, ensuring consistency and reusability across documentation. It begins with an XML declaration, followed by a root <topic> element that defines the document type.

Each <topic> element (or its specializations like <concept>, <task>, and <reference>) must have a unique id attribute to enable proper referencing and reuse within a DITA map.

In Oxygen XML Editor, when you create a new DITA topic, the file automatically includes:

1. Title: Specifies the topic's heading (<title>).
2. Topic body elements: Contain the main content, including paragraphs (<p>), lists (<ul>/<ol>), and other structural elements. The body element varies by topic type:
   - Concept topics use <conbody>
   - Task topics use <taskbody>
   - Reference topics use <refbody>

Figure 3. Basic structure of a concept topic with its elements.



The title tag contains the topic's heading

The shortdesc tag provides a brief summary of the topic

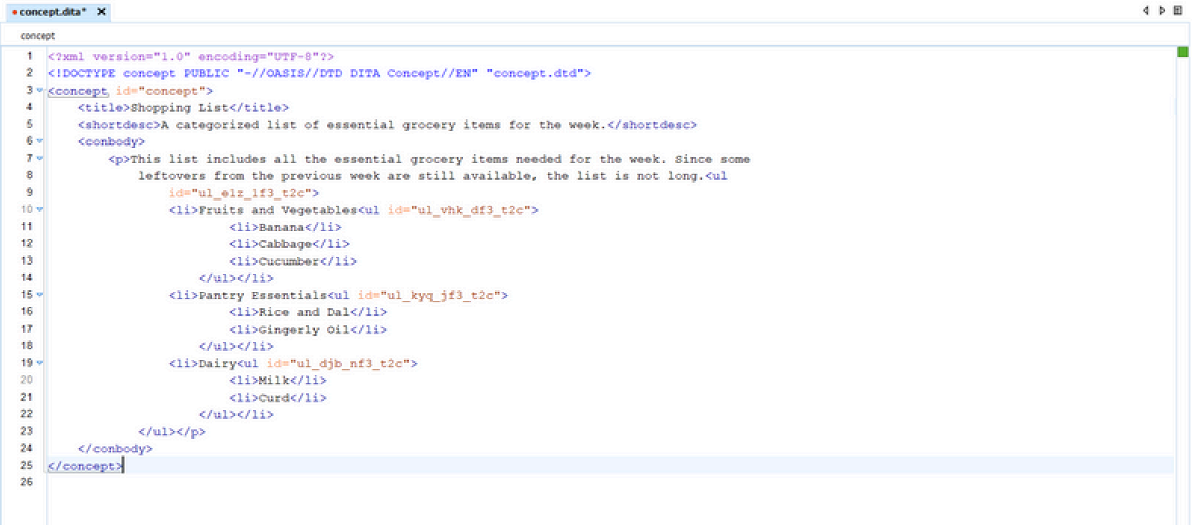
The conbody tag contains the main content

Text Mode vs. Author Mode

Text mode allows for direct XML editing, where content is written in raw XML format. Users manually insert tags, attributes, and elements, offering greater control over the document's structure. It is preferred by users who need precise XML customization for troubleshooting errors or applying batch changes.

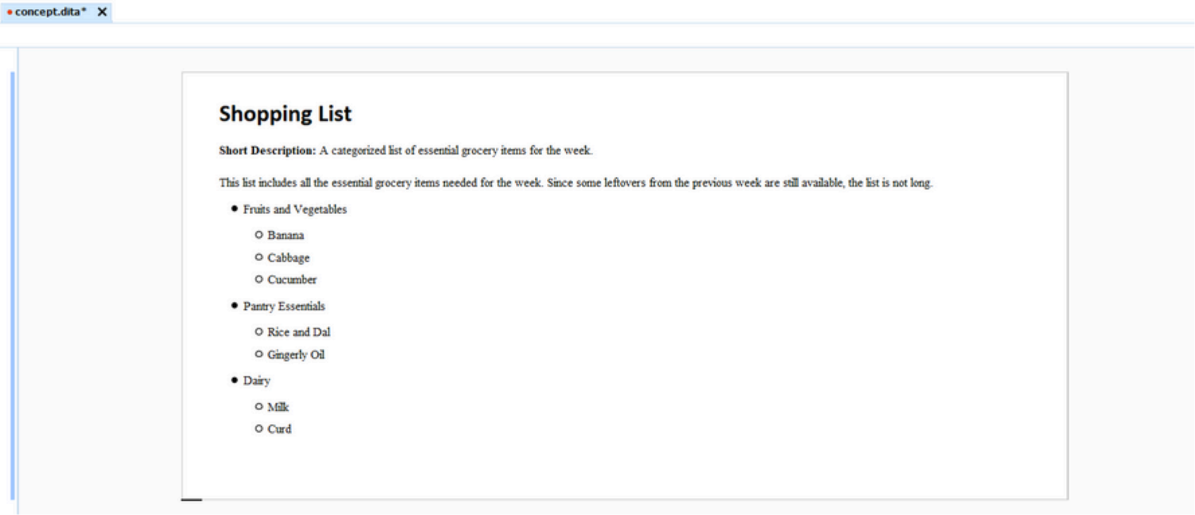
Author mode provides a structured WYSIWYG interface, making it easier to create and edit content without manually handling XML syntax. Instead of viewing raw tags, users interact with formatted text, similar to a word processor, while still maintaining semantic structure and metadata integrity.

Figure 4. Example of text mode.



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE concept PUBLIC "-//OASIS//DTD DITA Concept//EN" "concept.dtd">
<concept id="concept">
    <title>Shopping List</title>
    <shortdesc>A categorized list of essential grocery items for the week.</shortdesc>
    <conbody>
        <p>This list includes all the essential grocery items needed for the week. Since some
            leftovers from the previous week are still available, the list is not long.<ul
                id="ul_elz_lf3_t2c">
                <li>Fruits and Vegetables<ul id="ul_vhk_df3_t2c">
                        <li>Banana</li>
                        <li>Cabbage</li>
                        <li>Cucumber</li>
                    </ul></li>
                <li>Pantry Essentials<ul id="ul_kyq_jf3_t2c">
                        <li>Rice and Dal</li>
                        <li>Gingerly Oil</li>
                    </ul></li>
                <li>Dairy<ul id="ul_djb_nf3_t2c">
                        <li>Milk</li>
                        <li>Curd</li>
                    </ul></li>
            </ul></p>
    </conbody>
</concept>
```

Figure 5. The same example in text mode rendered in author mode.



Shopping List

Short Description: A categorized list of essential grocery items for the week.

This list includes all the essential grocery items needed for the week. Since some leftovers from the previous week are still available, the list is not long.

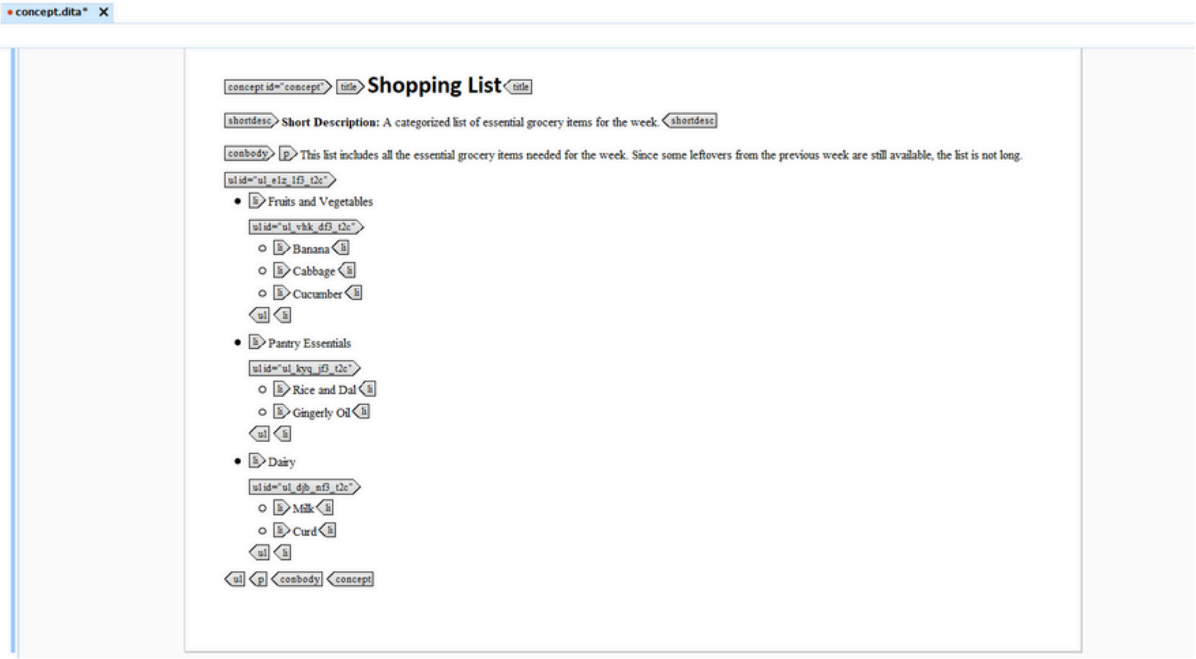
- Fruits and Vegetables
  - Banana
  - Cabbage
  - Cucumber
- Pantry Essentials
  - Rice and Dal
  - Gingerly Oil
- Dairy
  - Milk
  - Curd

4.
Structuring Content in DITA

The following list contains the key elements that organize content into structured format:

1. **Titles**: Every DITA topic requires a <title> element to define the heading.
2. **Paragraphs**: The <p> element structures the main content into readable text blocks.
3. **Tables**: Tables (<table>) present structured data clearly using <row> (for rows) and <entry> (for columns).
4. **Lists**: DITA supports three types of lists.
    - Unordered Lists (<ul>): For non-sequential items.
    - Ordered Lists (<ol>): For step-by-step instructions.
    - Description Lists (<dl>): For defining terms (<dt>) and their descriptions (<dd>).

Figure 6. Structure of a topic with key elements such as title, short description, conbody, and nested lists.



```
• concept.dita*  ✕

concept id="concept"  title  Shopping List  title
shortdesc  Short Description: A categorized list of essential grocery items for the week.  shortdesc
conbody  p  This list includes all the essential grocery items needed for the week. Since some leftovers from the previous week are still available, the list is not long.
ul id="ul_e1z_1f3_t2c"
  • li  Fruits and Vegetables
    ul id="ul_vhk_df3_t2c"
      ○ li  Banana  li
      ○ li  Cabbage  li
      ○ li  Cucumber  li
    ul  li
  • li  Pantry Essentials
    ul id="ul_kyq_jf3_t2c"
      ○ li  Rice and Dal  li
      ○ li  Gingerly Oil  li
    ul  li
  • li  Dairy
    ul id="ul_djb_nf3_t2c"
      ○ li  Milk  li
      ○ li  Curd  li
    ul  li
ul  p  conbody  concept
```

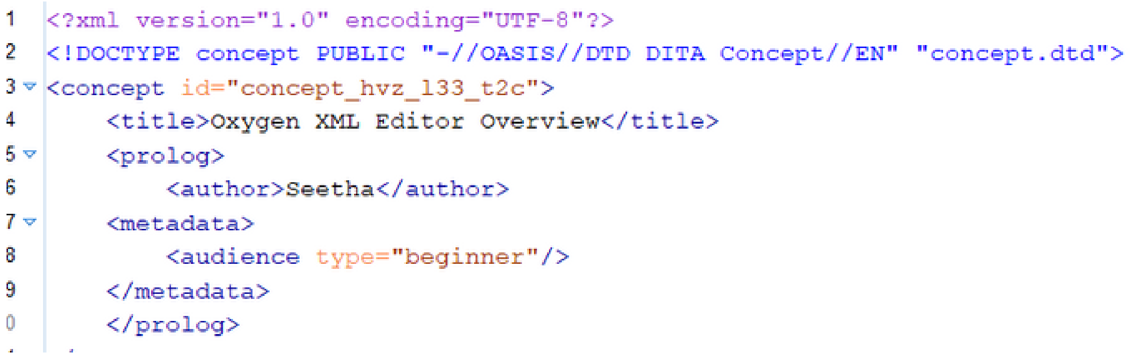
Metadata and Attributes

Metadata elements in DITA provide additional information about a topic, enhancing searchability, personalization, and version control. Metadata elements are placed inside <metadata>, which is located in the <prolog> section of a topic.

Here are some commonly used metadata elements:

Element	Use Case	Example
<author>	Specifies the topic's author.	<author>Seetha</author>
<audience>	Defines the target audience.	<audience type="beginner"/>
<prodinfo>	Provides product and version details.	<prodinfo product="Oxygen XML Editor" version="25.1"/>
<keywords>	Lists important terms for searchability.	<keywords><keyword>Metadata</keyword><keyword>DITA</keyword></keywords>
<category>	Classifies the topic into a category.	<category>Technical Writing</category>
<permissions>	Specifies content access permissions.	<permissions view="internal"/>
<resourceid>	Assigns an external identifier to the topic.	<resourceid id="concept-3"/>

Figure 7. Metadata elements nested within the prolog section.



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE concept PUBLIC "-//OASIS//DTD DITA Concept//EN" "concept.dtd">
<concept id="concept_hvz_l33_t2c">
    <title>Oxygen XML Editor Overview</title>
    <prolog>
        <author>Seetha</author>
    <metadata>
        <audience type="beginner"/>
    </metadata>
    </prolog>
```

Attributes in DITA help filter, categorize, and manage content dynamically. They allow writers to tailor content based on factors like audience, product version, or platform. This ensures that users only see the most relevant information for their specific needs.

Here are some commonly used attributes:

Element	Use Case	Example
audience	Targets specific user groups.	<p audience="beginner">This is an introduction to DITA.</p>
platform	Filters content based on operating systems or devices.	<p platform="windows">Click the Start menu.</p>
product	Associates content with a specific product or version.	<p product="OxygenXML">Use the DITA Map Editor.</p>
props	Custom attribute for defining content variations.	<p props="trial-user">Upgrade to the full version.</p>
importance	Specifies priority or emphasis of content.	<p importance="high">Read this before proceeding.</p>

5.
Managing and Reusing Content in DITA

DITA allows writers to effectively maintain consistency across all documents through its content reuse capabilities. This section introduces three key reuse mechanisms:

1. Content References (conref): Reuse content across multiple topics to maintain consistency.
2. Conditional Text: Filter and customize content based on audience, product versions, or platforms.
3. Cross-references (xref) and Key References (keyref): Enhance navigation and streamline content management.

Why Reuse Content?

Content reuse minimizes duplication and ensures that documentation remains consistent and easy to update. When identical content (a set of instructions, warnings, or notes) appears in multiple locations, referencing the original content instead of duplicating it makes maintenance more efficient. Any updates made to the source content automatically reflect across all instances where it is reused.

1. Reusing Content with <conref>

Content references (conref) allow the reuse of content across multiple DITA topics. Instead of duplicating text, conref pulls content from a source topic and displays it wherever referenced, eliminating the need for manual updates across multiple topics.
Steps to Use <conref>:

1. Identify the source topic and ensure it has a unique id attribute.
2. Insert a <conref> attribute in the target topic to reference the source content.
   Example: <note conref="source_topic.dita#id-attribute"/>
   Replace source_topic with the actual topic filename and id-attribute with the ID of the element you want to reference.
3. Validate the document by clicking the Validate icon (✓) in the toolbar at the top.

2. Implementing Conditional Text for Different Outputs

Conditional attributes filter content based on the audience, platform, or product version. The type of condition applied depends on the attribute used, ensuring users see only the information relevant to their category.
For example:
<note audience="admin">Ensure you have administrator privileges before proceeding.</note>
<note audience="user">Contact your administrator if you encounter permission issues.</note>

To make conditional text work, profiling needs to be set up, followed by generating outputs. This process ensures the correct content is displayed for each condition. As this guide is for beginners, profiling and output generation will be covered separately.

3. Reusing Content with Key References <keyref> & Static Links <href>

The keyref attribute enables content reuse without hardcoding file paths. Instead of linking directly to a topic, it references a key defined in the DITA map. This makes keyref ideal for content that may change in the future, such as product names or document titles.

The href attribute creates a static link to a specific file, similar to a traditional hyperlink. Use href only for permanent links that do not require reuse across multiple documents.

In the following example, keyref links to a reusable key, while href directly references a file:

```
<xref keyref="installation_guide"/>
<xref href="setup.dita"/>
```

A Comparative Table of Conrefs, Hrefs, and Keyrefs

Method	Use case	Examples
conref	Reusing structured content across topics	Warnings, standardized notes, troubleshooting steps
keyref	Dynamic linking for reusable text	Company names, product versions, document titles
href	Directly linking to specific topics or files	Installation guides, external websites

6.

Publishing the Document

After writing the content and adding topics to the DITA map, the next step is to validate and publish the document. Begin by clicking the Validate icon (✓) in the toolbar to ensure there are no errors.

Publishing a DITA document involves converting it into different output formats, such as PDF, HTML, WebHelp, and EPUB. This process is handled through Oxygen XML Editor's transformation scenarios, which automate the conversion using the DITA Open Toolkit (DITA-OT).

DITA supports multiple output formats based on project requirements. The most commonly used formats include:

- **PDF**: Used for printable documentation, user manuals, and offline reference materials.
- **HTML5**: Suitable for web-based documentation, online help systems, and mobile-friendly content.

To generate these outputs, follow these steps:

1. Load the DITA map for the project.
2. In the DITA Maps Manager, select the **main DITA map**.
3. Click the **Configure Transformation Scenario** button (wrench icon) in the toolbar.
4. A list of available transformation types will open. Choose PDF via DITA-OT, HTML5 via DITA-OT, or another format as needed.
5. Click **Edit** to adjust parameters, such as:
    - Styling (using XSL-FO or CSS for formatting).
    - Title and footer customization.
    - TOC (Table of Contents) inclusion.
6. Click **Apply** to start the transformation process.
7. Once completed, check the output folder for the generated file.

For custom transformations, Oxygen XML allows users to create a new transformation scenario by clicking New and modifying the parameters.

7. Conclusion

This guide has covered the fundamental concepts of authoring structured documentation in DITA using Oxygen XML. While there are more advanced features to explore, the topics covered here provide a solid foundation for getting started with the tool. We have discussed structuring topics, reusing content, applying metadata and attributes, linking and navigation, and publishing documentation.

To ensure efficiency and maintainability in your documentation, follow these best practices:

- Use well-defined topic types to ensure clarity and logical organization.
- Leverage content reuse through conrefs and keyrefs to maintain consistency and reduce redundancy.
- Apply metadata and attributes to improve searchability, filtering, and content personalization.
- Establish clear linking and navigation using DITA Maps, xrefs, and keyrefs.
- Implement conditional processing to generate tailored content for different audiences and outputs.
- Validate content regularly using Oxygen XML’s built-in tools to detect and resolve errors.
- Optimize publishing workflows by utilizing predefined transformations for various output formats.

This guide concludes here, providing you with the essentials to begin creating your first document. Continue exploring DITA’s capabilities to refine your workflow and enhance content quality.