flow_dita_webhelp

sectionDef = "coverSection" pageDef="Cover"

sectionDef = "coverSection" pageDef="Blank"

Contents

sectionDef = "contentsSection" pageDef="Contents"

sectionDef = "contentsSection" pageDef="Contents2"

List of Figures

List of Tables

1 Technical writing (WEG?)

1.1 Structured authoring

1.1.1 Definition

Structured authoring is a way of writing in which one or more authors use:

- a common, limited set of components (“information building blocks”) set by an information model.
- standards, rules and guidelines to structure the text.

1.1.2 Result

Structured authoring seperates the content from the layout. It structures content through information models and layout through templates and stylesheets. These output any content in different layout styles.

1.1.3 Purpose

- Maximize consistency
  - content
  - structure
- Re-use
  - Single sourcing [p. 13] (content ↔ context)
  - Multi-channel publishing [p. 14] (write content once, publish to multiple outputs)
- Collaboration with other authors

1.1.4 Use

You can apply structured authoring to all kinds of texts. It is not limited to the type of software application you use. You can therefore use structured authoring in:

- word processing programs such as Microsoft Word
- desk top publishing software such as Adobe InDesign
- document processors such as Adobe FrameMaker
- help authoring tools such as MadCap Flare
- XML programs such as Oxygen XML Author or JustSystems XMetaL
- wikis such as Confluence or MediaWiki

1.1.5 What is a topic

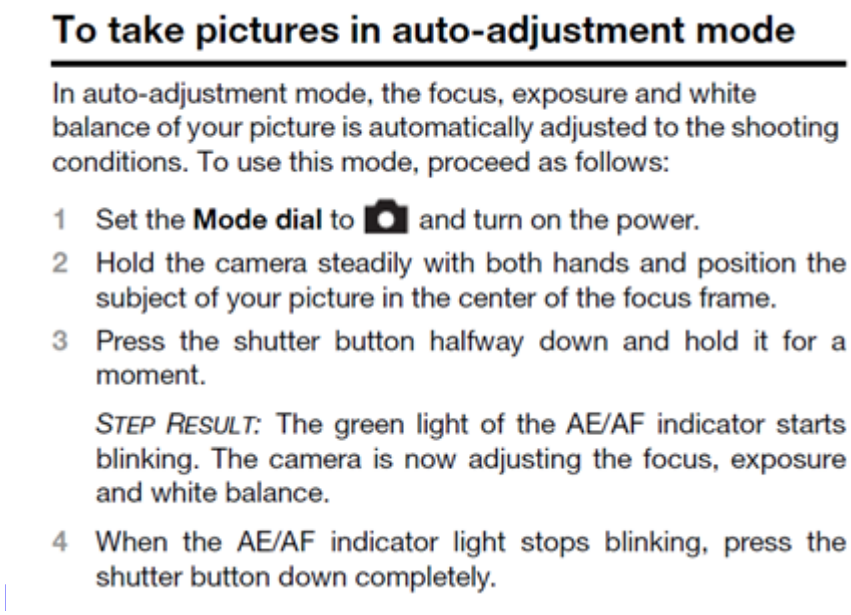
In DITA, all content is chunked into topics. Topics are individual units of information, organized around a single subject or answering a single question. Each topic makes sense on its own and can be authored as a stand-alone piece of content. A topic is considered the smallest information unit and multiple topics can be combined to create bigger information units (like chapters and volumes).

A topic is traditionally composed of a title and some content. That content is built up of structural components, such as the following:

- Paragraphs
- Tables
- Lists (bulleted lists, numbered steps …)
- Illustrations
- Notes ("Warning!", "Attention!", "Tip" …)

All the structural components of which your content consists are predefined in an information model . Technical writers working on the same project use this information model to ensure that all topics (that make up the final information product) are structured in the same manner.

The following topic answers the question “How do I take pictures in auto-adjustment mode?” It is composed of a title, some introductory sentences and a list of numbered steps.



To take pictures in auto-adjustment mode

In auto-adjustment mode, the focus, exposure and white balance of your picture is automatically adjusted to the shooting conditions. To use this mode, proceed as follows:

1 Set the **Mode dial** to and turn on the power.

2 Hold the camera steadily with both hands and position the subject of your picture in the center of the focus frame.

3 Press the shutter button halfway down and hold it for a moment.

STEP RESULT: The green light of the AE/AF indicator starts blinking. The camera is now adjusting the focus, exposure and white balance.

4 When the AE/AF indicator light stops blinking, press the shutter button down completely.

1.1.6 What is an information model

An information model is a formal description of the structure and the structural components of which your information product consists.

structure	the combination of the components’ order and hierarchy
structural component	title, warning, table, illustration, (un)ordered list …
information product	this can be a “document” such as a manual or a work instruction, but it can be an online help as well, or an intranet system or data base.

An information model defines rules on how to structure the content. These rules need to be logical and make sense, so writers can easily learn and apply them. Some writing applications can validate these rules to enforce consistency. Validation greatly facilitates the writer’s task.

- DITA
- DocBook
- Information Mapping
- templates in authoring tools

1.1.7 Best practices of structured authoring

Every technical writer should frame the list below (or stick it to the door of the lavatory) to be reminded daily of how to write good documentation.

Audience-oriented authoring

Know who you are writing for and keep your readers in mind when putting text to paper. Continually ask yourself: is this piece of information relevant for my readers?

You might want to work with layers of complexity through conditionalized text.

- beginners vs. advanced users
- end users vs. administrators

Audience-oriented writing is also called user-oriented writing or profiling.

Coherence

Make sure to group information that belongs together, both on a micro level (page) as on a macro level (the entire “publication” such as the manual or the intranet).

Consistency

Limit one term to one concept (terminological consistency) but also be consistent with punctuation, layout and structure. This is where information models, with their templates and XML schemes, prove their worth.

Illustrations and animations

Add illustrations and animations wherever you can, but:

- Avoid screenshots:
  - you have to check them for every new release.
  - they often do not provide added value.
- Avoid text in illustrations:
  - they ask more time to update.
  - generally the text cannot be translated.

Minimalism

Keep it simple. Avoid walls of text[1].

---

1 Source: 16 quick tips for better work instructions, by Dozuki

Only provide the information your readers need to grasp the concept or perform the task

Cut into redundant information. Do not crowd your page with unnecessary information or too many details. Your reader will lose interest or even get lost.

Only provide the information once

Do not offer a thousand ways to reach the same solution: one is enough. Avoid repeating information in several places in the document, if possible. Links are a useful solution here.

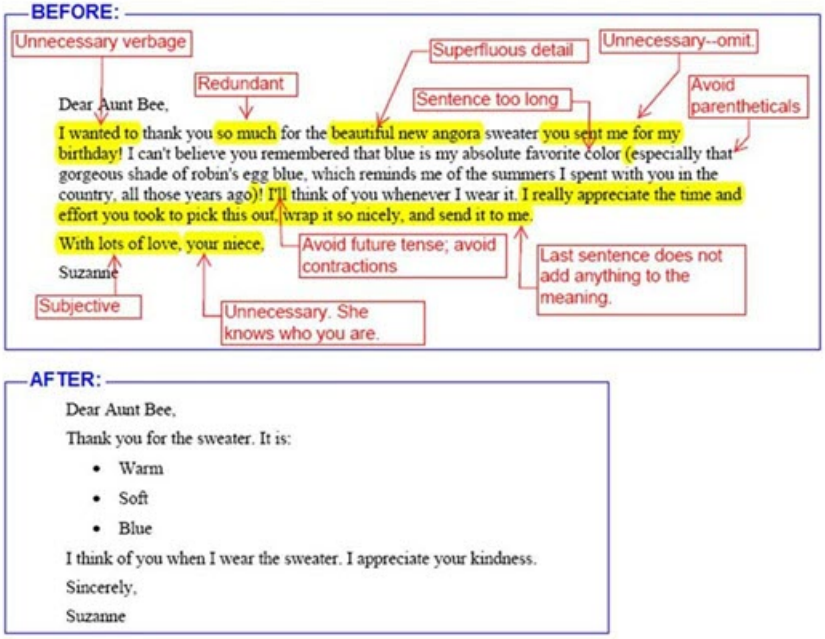
Some advantages

- Information scanning improves.
- Users remember more.
- Smaller costs for DTP work, translations and printing.

Do not exaggerate

Minimalism walks a thin line between leaving out the unnecessary and keeping the essentials. Check that you have not cut out so much that there are gaps in your explanation or procedure. This is very important in work instructions: gaps can create safety risks.

How a career in technical communication ruined me as a letter writer



1

1 Source: unknown

Segmentation ("Chunking")

Split or "chunk" long blocks of text into smaller information units to facilitate access to the information.

Use:

- information types
- titles and subtitles
- tables
- lists

Structure

Structure the information building blocks (sections, topics, chapters …) logically, consistently and coherently to create a tree structure that is accessible from different points of view.

"Scannable" titles

Start with short but meaningful titles that express the quintessence of the message. Avoid vague titles, as readers use the titles to scan the document to quickly find the information that is relevant to them.

1.2 Five copywriting errors (WEG?)

You can find the full article here.

In summary:

1.2.1 1st error: Writing inwardly

The writer focuses on his own experience with the product, not on what the product can offer to the reader.

1.2.2 2nd error: Burying the lead

To keep the attention of the reader, it is important to sell your message succinctly and put your most important points first.

1.2.3 3rd error: Mediocre meta material

Make intelligent use of meta information which will be picked up by search engines.

1.2.4 4th error: Saying too much

Less is more.

1.2.5 5th error: Weak or no calls to action

Make clear which actions the reader can take.

2 DITA

2.1 What is DITA?

2.1.1 Definition

DITA is an open standard to structure documentation and is therefore often used in technical writing.

It stands for Darwin Information Typing Architecture.

Darwin

A set of information models that can evolve and be specialized to account for the client's particular authoring and publishing needs.

Information Typing

Content is organized in chunks of information called "topics" and topics are categorized by the type of information they contain:

- task
- concept
- reference
- (or other types of information)

See also: What is an information type [p. 7]

Architecture

A set of standardized, open-source, well-documented information models for various types of content

2.1.2 History

DITA was originally developed by IBM but since 2005 it is defined and maintained by OASIS, a non-profit consortium comprised of members from the technology industry, users and influencers.

2.1.3 Features

- Re-use
  - Single sourcing [p. 13] (content ↔ context)
  - Multi-channel publishing [p. 14] (write content once, publish to multiple outputs)
- Collaboration with other authors
- Minimalism [p. 3]
- Structured authoring [p. 1]
- Intelligent content
- Conditional content
- Writing for translation

DITA and structured authoring are closely related. The first three DITA features are also structured authoring features.

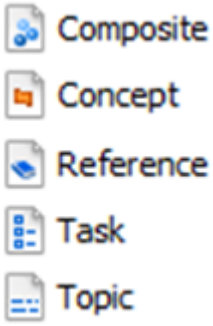
2.1.4 References

As DITA is an open standard, you can consult the DITA specifications for free on the internet:

- DITA 1.3: http://docs.oasis-open.org/dita/dita/v1.3/errata01/os/complete/part0-overview/dita-v1.3-errata01-os-part0-overview-complete.html
- DITA 1.2: http://docs.oasis-open.org/dita/v1.2/os/spec/DITA1.2-spec.html

2.2 What is an information type

DITA distinguishes different types of topics, each intended for a particular type of information. These topic types are therefore also called information types. A well-written topic includes only one type of information.



The three core information types in DITA are the following:

Information type	Description
Task	Task topics include the steps of a particular task or procedure. They answer the question “How to…?”.
Concept	Concept topics include background information that users must know (e.g. definitions, examples…) before executing a particular task or procedure. They answer the question “What is…?” or "What does a certain term mean?".
Reference	Reference topics provide quick access to reference information that is relevant to the subject at hand. Reference information is often presented as a list or a table (e.g. an overview of error codes, a bibliography…).

Generic topics (**topic**) and **composite** topics should only be used for specific purposes:

Information type	Description
Topic	Generic topics are designed as a starting point for specialization. We use them, however, to create title-only topics (stub topics) when composing a DITA map.
Composite	Composite topics can include task, concept and reference elements. You can only use these topics to create conref libraries, not to create topics with mixed content.

Technical writers split up texts depending on the kind of information they want to present to the reader. By using information types to filter and structure the information, the reader will no longer have to spend energy on this when reading the text.

The components in an information type and their structure and hierarchy are defined in the information model the writer uses.

2.3 What is a DITA map

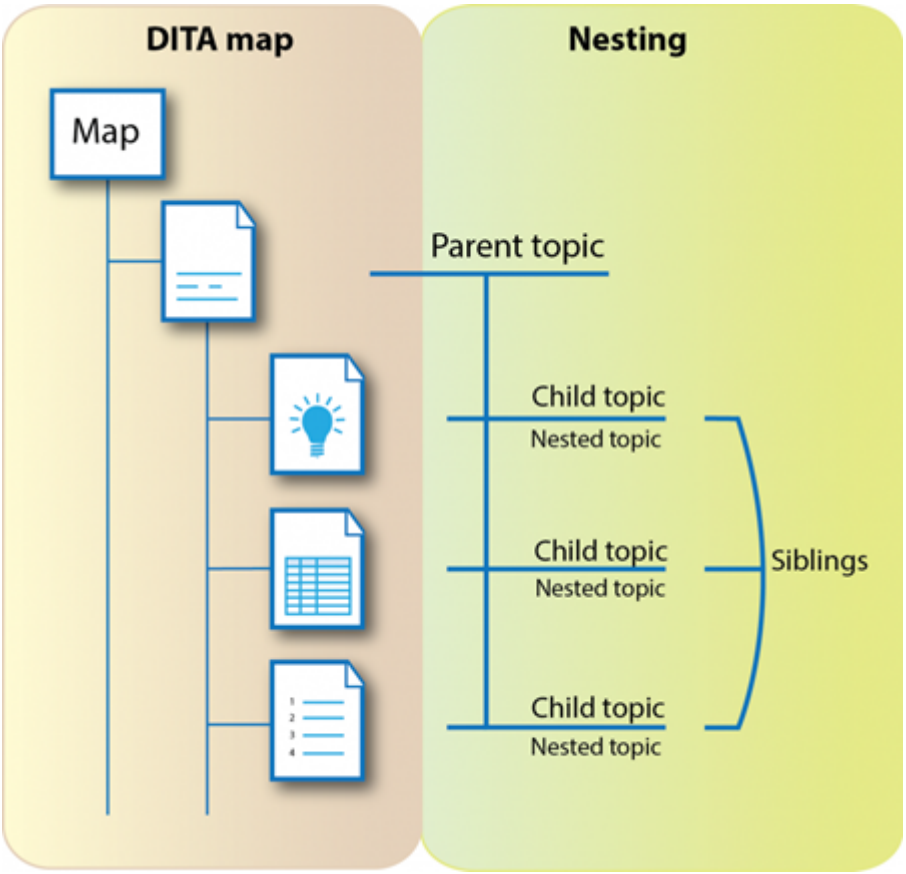
A DITA map is a collection of topics assembled into a sequence and hierarchy.

2.3.1 Description

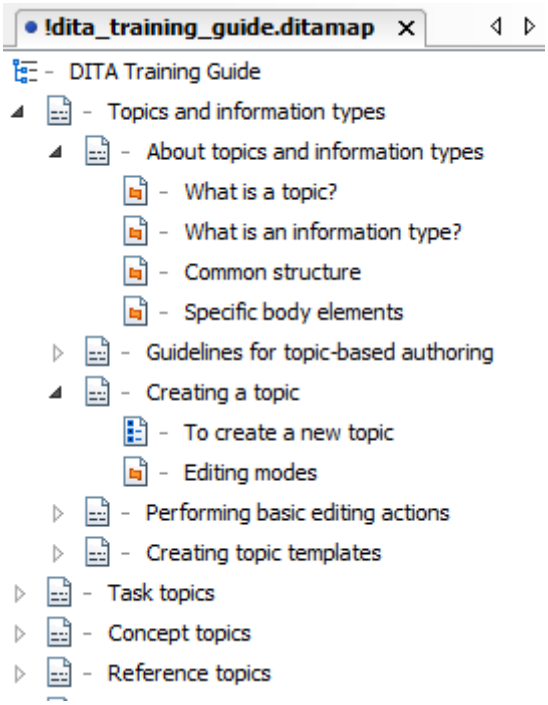
DITA maps are files with the extension *.ditamap*. Unlike topics, DITA maps shouldn't contain any content of their own. Instead, they contain references to the content and provide a structure for these references. In this sense, you can best compare a DITA map to an outline that describes the hierarchy and the order of its topics.

2.3.2 Hierarchy and nesting

DITA maps contain the structure of your topic references. This structure resembles a table of contents and helps your reader locate the required information. To create a meaningful structure, you can group relevant topics together and nest them under a topic grouping or a . The head topic is referred to as the 'parent topic' and the nested topics underneath it as its 'child topics'. The relationship between the child topics is defined as 'siblings'.



Here you can see the table-of-contents-like structure of a DITA map created by the nested topics:



2.3.3 Purposes and advantages of DITA maps

As a DITA map defines the relationships between its topics, it forms the basis for navigating and linking content. A well-organized DITA map allows you to quickly find the topics you need.

Main purposes

Concretely, you use a DITA map structure to:

- Collect topics you want to appear in your output.
- Organize the topics in a manageable structure that is easy to navigate - both for you and your reader.
- Define the relationship between topics and correctly link them.

Easy collection of topics and simple creation of topic structure

In essence, the topics you write are loose files that you can gather in a DITA map. You enter a topic into your DITA map by adding a reference to that topic - a link. As such, when you delete a topic from the list, you do not delete the actual file but only its reference within the map. This makes adding, reorganizing or deleting topics an easy task. For instance, when a particular topic is no longer useful for the product you write about (the new model no longer carries a particular function), you can delete its reference from the map. When later the function is relevant again, you can just re-add the topic reference, as you never deleted the actual topic file in the first place.

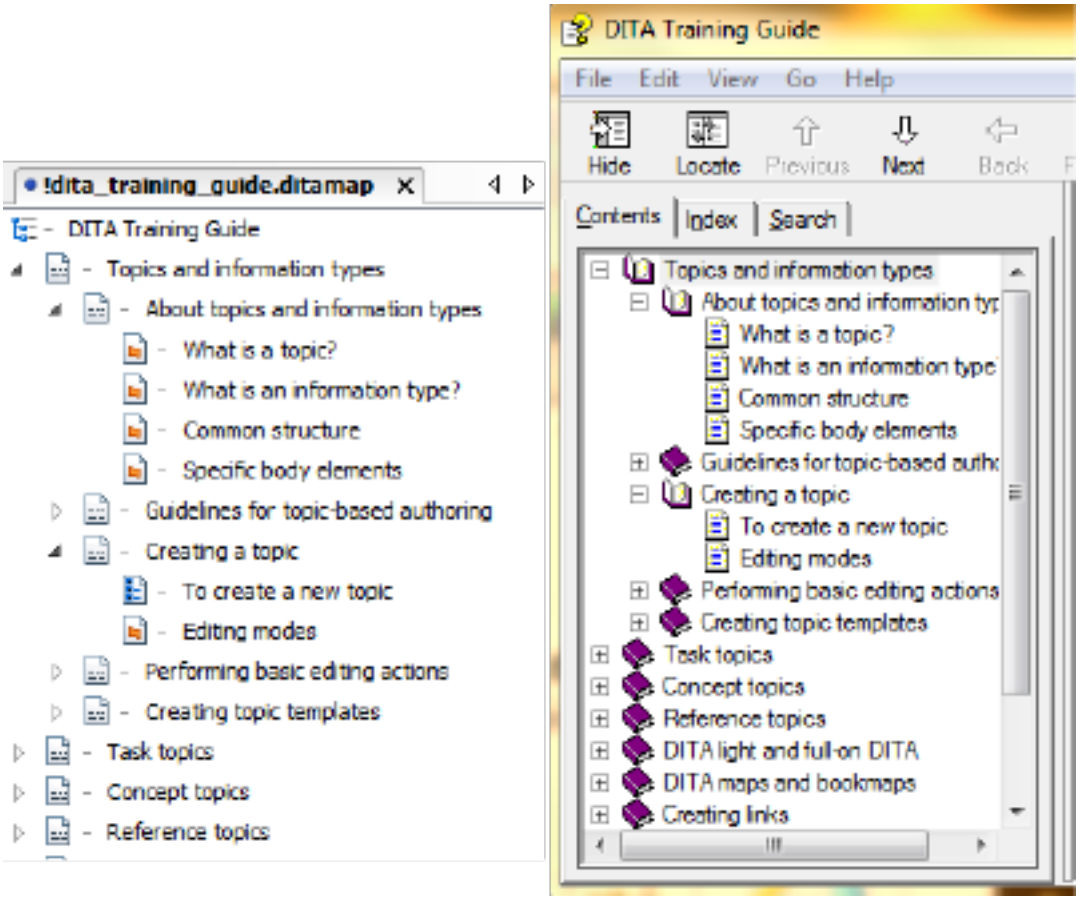
The newest model of a photo camera no longer supports the option to add color filters while you take pictures, so you remove the topicrefs about color filters from the DITA structure. The new manual for the camera no longer contains these topics, but you keep the original topic files in case a future camera model does have filters.

DITA hierarchy in output

In the DITA map, you organize your topics in a usable structure. When you generate output, this structure is used to organize your content and your table of contents. Nothing actually changes on the topic-level of your

content. You do not need to start numbering or changing the titles of topics to reflect their position within the DITA map. This structure is automatically generated and numbered (if applicable) while output is being generated. Of course, how exactly it appears in the output depends entirely on the type of format you are generating and (if applicable) on the template you use.

The table of contents of this webhelp file follows the DITA map structure of the original document:



Additionally, when you later want to change the order of your topics, promote or demote them, the structure will automatically change when you generate output once more. This makes structuring and re-using topics an easy task where you do not need to manually change every title according to its new position, nor adjust the table of contents afterwards.

Relationship table

The DITA map is a hierarchy of links to your topics. If you add a relationship table to the DITA map, you can also create links that go beyond the hierarchical structure. As such, most of your linking is centralized and easy to navigate in one file. This makes it easier to glance over the entire structure and establish which relationships exist and how they might need to be adjusted.

2.4 What is a bookmap

2.4.1 Definition

A bookmap is a specialization of a DITA map and is meant to be used when you intend to publish your documents in a book format. It contains structure and metadata, but no content. It is especially designed to create output in book-style formation and therefore includes the following major structures:

- title
- bookmeta (owners, authors, publishing data,…)
- front matter (acknowledgements, definitions, abbrevations,…)
- chapters and parts
- appendix topics
- back matter (index, notices, lists,…)
- relationship tables

2.4.2 Purpose of a bookmap

A bookmap is best suited for when you specifically want to create output in the traditional form of a book. Its template allows you to organize your content in a book format by adding, among others, chapters, parts and appendixes.

Note: *Bookmaps are not suited for output formats other than traditional book formats. If you are planning different types of output, you should consider using a regular DITA map and including book metadata by using the elements and attributes in the* ***‹topicmeta›*** *element.*

You use DITA to publish a research paper, so you write your information in a bookmap and add the necessary chapters. You can now make use of, for instance, the frontmatter to add a cover and the backmatter for glossaries.

2.5 Learning DITA (WEG?)

2.5.1 DITA for kids (WEG?)

DITA for kids is a project teachers can make use of in class. The project itself is finished, but the code is open-source and can still be edited.

Teachers can download and edit Wikipedia articles with the open-source tool. All content in the articles (including images) can be reused and edited in the DITA-editor. The teacher can then group the new topics in themes and publish everything to a package. This package can then be shared with other PCs.

The full explanation can be found on http://dita.xml.org/node/2400.

Don’t forget to take a look at the YouTube video:

2.5.2 Free e-learning course LearningDITA

Scriptorium, one of our partners in the US, has launched the LearningDITA website.

It is a free and basic e-learning course for anyone who wants to learn DITA. You learn about the DITA specification and the different topic types in DITA, and also how to make bookmaps.

The course itself is written in DITA, making use of the DITA-specialization for e-learning.

More info on: https://learningdita.com/about/

3 Automation of documentation processes

The automation of documentation processes has been extremely beneficial to the productivity of many companies, saving time and cutting costs in multiple areas and at multiple stages of the document life cycle.

The concepts and methods described in the folowing topics have been invaluable in the development of automated documentation processes:

3.1 Advantages of re-using content

Content re-use is one of the most important advantages of using DITA for authoring: it allows you to create content once and then re-use it wherever you need it again. Of course, if you want to take advantage of this feature, you have to write your topics in such a way that they still make sense in different contexts. In other words, your topics need to be self-contained snippets of content. When that is the case, you can enjoy the following merits of content re-use.

3.1.1 Efficiency

- By re-using content, you only have to write a certain topic once and you can re-use it in other locations. This means you free up time to create more content.
- You reduce your maintenance time and costs as you only need to adjust your original topic once and all locations where the topic is used will automatically be updated accordingly. This way you save time as you do not need to search through your entire document for the other snippets of text that also need to be changed.
- You reduce translation costs as the topic only needs to be translated once.

3.1.2 Consistensy and accuracy

By re-using content, you ensure consistency everywhere as the re-used content is the same in all places. Similarly, updated content is going to be the same everywhere so you can be sure the correct changes are applied to every relevant topic. This means you also reduce human error that could be caused by different writers updating all information.

3.1.3 Risk management

By re-using content, you reduce the risks that come with last-minute changes as your content is easier to update; you ensure that all instances are updated at once. If you need to update every individual instance of an element, there is a much higher risk that one or two mistakes will slip through.

3.2 Single sourcing

Writing technique by which you write content once but reuse it in several places.

Reusing content instead of copying it has the advantage that, in case of changes, you only need to modify the text once and it changes automaticly in all places where the text occurs. This is at the same time its biggest disadvantage: you always need to check if the change applies to all places where the text occurs.

sectionDef = "firstSection" pageDef="Right"

You can reuse several levels of information units, from ‹ph› elements to entire topics.

3.3 Multi-channel publishing

Multi-channel publishing is publishing documentation in different output formats, such as webhelp or pdf.

One of the great advantages of DITA content is its ability to be published in many different kinds of output formats. Its characteristic re-use strategies along with conditional processing were designed with efficient content-managing and publishing in mind. Essentially, one single source can become many different types of documents.

3.3.1 Structure vs. lay-out

DITA is an ideal writing tool as it helps you organize your content into specific elements embedded within a larger, universal structure. These structuring elements have nothing to do with a lay-out, instead they are fixed compartments in which you store your documentation. This also means that when your documentation is fully written, the DITA file cannot be published and immediately used as is.

The advantage of this universal (and free, open source) structure is that it can easily be opened in other programs more suited for publishing, such as publishing tools or DITA processors (Oxygen, XMetaL), which will do the transformation for you. They transform your base DITA files into a publishable file of your choosing, and in doing so, apply a new lay-out on top of the structure. Lay-out, in this sense, is merely a layer that makes the structure more visible and accessible to the reader.

It is important to understand that the underlying structure is not changed when the DITA files are transformed. Your content does not change; it is merely put in a new outfit more suited to your needs.

In summary:

DITA structure	Lay-out
Focus on content	Focus on appearance
Fixed in unchanging elements	Options change from program to program
Will remain the same in different programs	Might change if a document is opened in a different program
Adheres to strict DITA rules	Adheres to the rules of the individual programs and to the user's preferences
Allows the user to focus on content	Can distract the user from writing content

3.3.2 Publishing DITA content

During publishing, the DITA structure is used as the skeleton on top of which a skin of lay-out is applied. This publishing process involves a number of steps that interpret DITA maps, resolve conref elements and cross-references, apply formatting numbering and labels, and map DITA elements to output format elements.

During this process, the content is given a new look according to the requirements of the output format. Since normal computers usually have much larger screens and are differently used than tablets and smartphones, the output also needs to be presented differently on these different devices. Each device type has its own conventions and requirements, such as button size, font size, resolution etc.

3.4 Why use metadata

Metadata is of essential importance when organizing information. By correctly creating and applying metadata you can make it easier to search through information which is relevant to your company or organization. This way, you can add digital identifying criteria to your documents, which help you organize your resources in a digital information architecture.

3.4.1 What is metadata?

Metadata is data that provides information about other data. That information can contain a lot of things:

- Content
- Facts
- Records
- Tools
- Data
- ...

3.4.2 What types of metadata are there?

Metadata of documents are not set in stone. As the owner of a document, you can add and delete metadata as you wish. But it is essential, when saving documents in an organization or company, that there is one administrator who created well-thought-out rules.

Some examples of common metadata include:

- Author or creator of the file or document
- Time and date of creation
- File size

These examples occur frequently when applying metadata. But you can add different types of metadata to files if you think that is necessary.

Example: If you're working in Photoshop, Lightroom or other graphic editors, it is not uncommon to add metadata that apply to the size of the picture or the location where the picture was taken.

- How large is the picture?
- Which colors are being used?
- Where has the picture been taken?

But you can even go further: Which mood does this picture provoke? The possibilities with metadata are infinite.

3.4.3 International standards for metadata

The possibilities may be endless, but they are bound by international standards. The most important of these are the American National Standards Institute (ANSI) and the International Organization for Standardization (ISO). The most exalted metadata standard is ISO/IEC 11179 Metadata Registries (MDR).

3.4.4 Sources

- http://oxforddictionaries.com/
- http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=39438

4 XML editors

Since DITA is an open standard, it can of course be edited from any text editor such as Notepad++ or other text editing tools. However, most editing tools don't have support for the specific DITA standard. For this you need a DITA authoring tool. The following tools are used within our company:

4.1 Technical editors

These editors give you the ability to 'hide' the tags while writing, allowing you to focus on the content while still writing structured XML documents.

- Adobe FrameMaker in Structured mode. In structured mode, you can edit DITA XML documents right from the familiar FrameMaker interface.
- Oxygen XML [p. 16] is an editor especially for DITA. It consists of a few separate packages, with an ever-growing range of products:
  - Oxygen XML Editor is the complete package, meant for people that do technical writing as well as developing DITA Schemas and transformations.
  - Oxygen XML Author is meant for technical writers. With this package you get a more user-friendly interface and most complicated DITA options are well-hidden in the interface.
  - Oxygen XML Web Author builds on the Oxygen XML Author version, allowing online collaboration between different teams and automatic cloud storage.
  - Oxygen XML Developer is meant for documentation architects and designers. It is a tool you can use to determine how the documentation in the team should be structured and what the output should look like.
-

4.2 User-friendly editors

DITA and XML both have a reputation of being complicated, with their tags and attributes. Recently, however, more and more user-friendly editors are being developed.

Some examples:

- Fonto Editor, made in The Netherlands, can be useful for the following scenario:
  a Subject Matter Experts (SME's) write reasonable structured topics
  b Technical writer improves structure within the Fonto Editor or an alternative DITA-editor
- Paligo, made in Sweden

4.3 Oxygen XML

Oxygen is an XML editor (developed by Syncro Soft) that provides a suite of XML authoring and development tools. It is designed to accommodate a large number of users, ranging from beginners to XML experts. It is available on multiple platforms, all major operating systems, and as a standalone application or an Eclipse plug-in. You can use Oxygen XML Editor in conjunction with all XML-based technologies and it includes a variety of tools for creating, editing, and publishing XML documents.[1]

---

1 Source: https://www.oxygenxml.com/

4.3.1 Guidelines and tips for writing in Oxygen XML

Guidelines for chunking content

Include enough information

Although a topic should only answer a single question, it should still contain enough information to make sense on its own. Which info the topic has to contain depends on the information type:

Information type	Has to include enough info to
Task topic	Begin and complete a particular task
Concept topic	Grasp a particular concept
Reference topic	Look up and understand particular information

Note: *Topics often include related content. A particular task topic can, for example, use concepts that are described in separate concept topics. As a result, it is impossible to make each topic completely autonomous. You can solve that problem by adding related links to your topic (using the* ***‹related-links›*** *element); the body part of your topic remains discrete, while you still guide the user to the related information.*

Mind Miller's Law (7±2)

According to Miller's law, an average person can only hold 7±2 objects in working memory. You should keep in mind this law when chunking content into topics. For instance, if you are writing a task topic and notice that it contains over ten steps, try to reduce the number of steps. You can do this by:

- grouping short steps.

  Note: *For example, the following steps:*

  a Choose **File**
  b Choose **New**
  c Choose **Topic**

  can be grouped into: Choose ***File*** *>* ***New*** *>* ***Topic****.*
- splitting your task into two or more subtasks.

  Note: *You can then include an overview topic that describes the entire process and refers to the different steps (subtasks) of that process.*

The same goes for other lists. If you have a list of twenty safety instructions, for example, you can group the related instructions in separate sections. You can then also include a specific title for each group of safety instructions, to better guide the users.

Include enough information

The first challenge in topic-based authoring is chunking your content into discrete topics. It can be hard to determine the ideal length of a topic: it should be short enough to answer a single question yet still include enough information to stand on its own. Additionally, you should also keep in mind Miller's law when chunking your content.

Guidelines for writing topic titles

Once you have decided which information you will include in a particular topic, you can choose an appropriate title for that topic. Well-chosen titles are crucial to the scannability of your document: they help users to quickly find the info they are looking for.

Reflect the content

A good topic title reflects the content of the topic. Because most topics answer a specific question, you can use (a derivative of) that question as your title. For example:

Information type	Question	Title
Task topic	How do I insert batteries in my camera?	How to insert batteries in your camera
Concept topic	What is white balance?	What is white balance
Reference topic	What do these battery icons mean?	Overview of battery icons

If you have a hard time finding a title that covers the content of a topic, that topic probably addresses more than one subject or includes several information types. You then have to split the topic into two or more smaller topics, so you can label them appropriately.

Be concise and specific

Concise topic titles are the best way to quickly guide the reader to the correct information. When you have written a title, always ask yourself if you cannot remove any words.

Although it might seem fun, avoid using puns, tropes and figures of speech. Technical documentation is not the place to give your literary aspirations free course.

However, you should also take care not to become too concise: your topic titles should still be specific. Vague topic titles are meaningless and do not speak to the user. When you have written a title, you should ask yourself if you do not have to add a word or two. Instead of using “Introduction” or “Overview” as a topic title, for example, specify what you are introducing or which information there is in the overview (“Introducing DITA”/“Getting started with DITA”, “Overview of DITA information types”).

Use consistent syntax

Use the same syntax for titles of similar topics or - if possible - for all topics of the same information type. Like that, the user immediately knows which kind of information he will find in the topic and can decide whether or not he has to read it.

“How to insert batteries in your camera”, “Inserting batteries in your camera”, “To insert batteries in your camera” or “Insert batteries in your camera”, for example, are all valid titles for a task topic. If you choose one formulation and use it for all your task topics, users can immediately distinguish task topics from concept and reference topics.

Guidelines for creating DITA maps and bookmaps

Mind Miller's Law (7±2)

It is useful to take some time to structure your information in your DITA map. A well-organized structure can go a long way in helping your reader find and understand information. As such, it is useful to keep in mind the following guidelines.

As much as it applies to creating sections within topics, Miller’s law (an average person can only hold 7±2 objects in working memory) applies to creating structure in your DITA map. As such, it is advised to create a parent topic with around seven child topics.

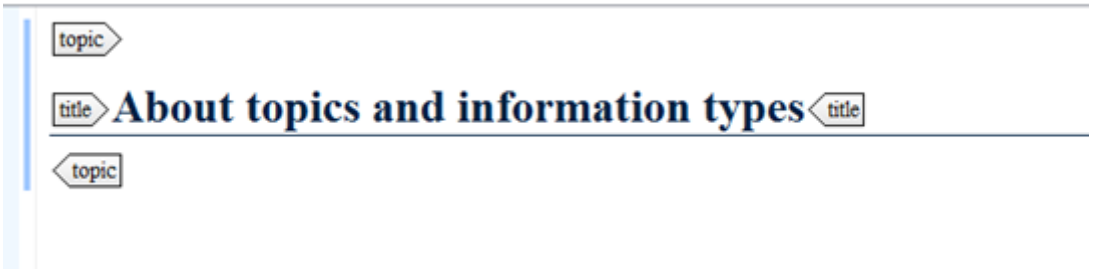
Additionally, the human mind can process about four concepts at a time, which means that when you start nesting topics, you should take care to create no more than four generations of topic families. The more you start nesting, the higher the chance the reader might start to lose track and forget how everything fits together.

Use stub topics

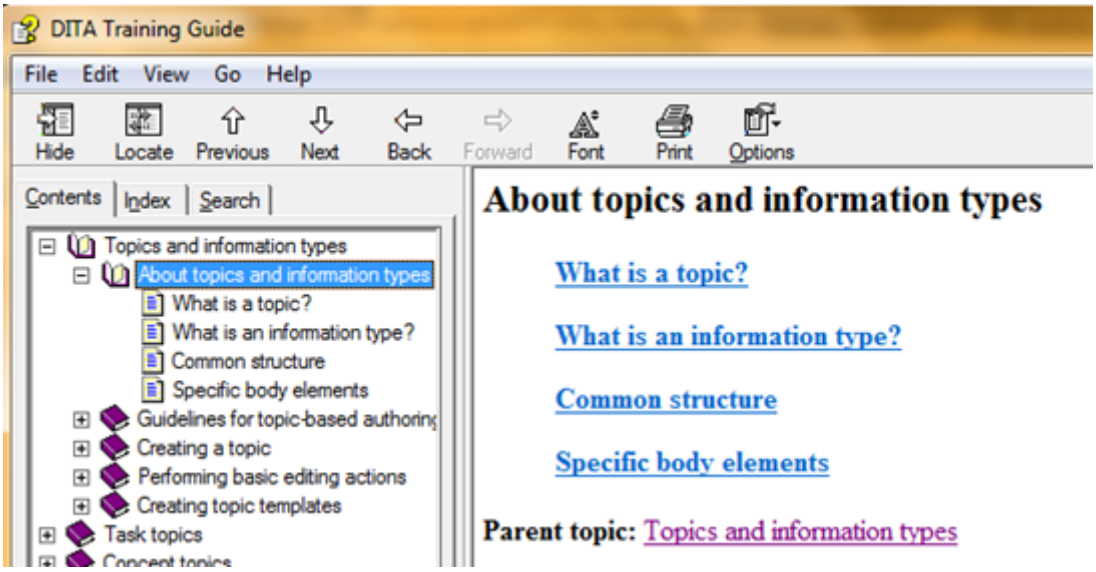
When you are organizing your topics into groups, avoid using topic headings (‹**topichead**› elements). Instead, use stub or title topics: these are general topics that contain only a title.

This type of topic is automatically filled in with links to its child topics in output. If you want it to display content instead, you can use the *@chunk* attribute on this stub topic and enter the value *to-content*. This way all information from the child topics will be added to the stub topic in the output.

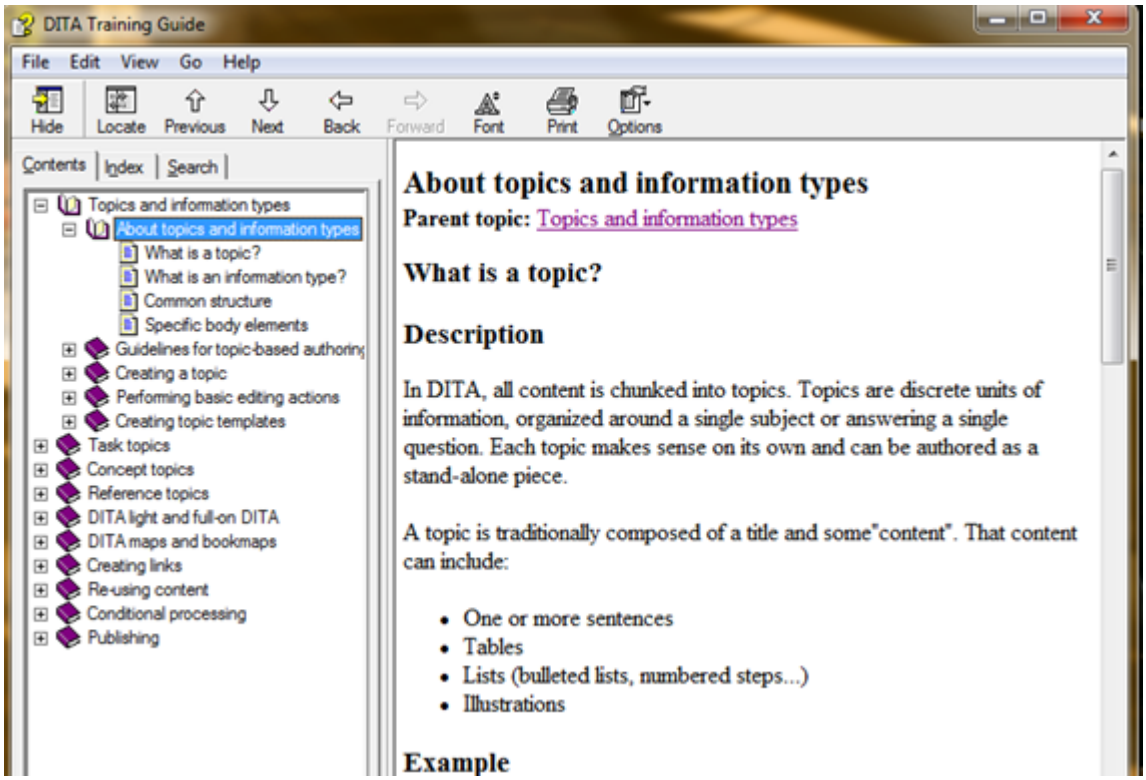
A stub topic as it appears in DITA:



The same stub topic as it appears in webhelp. It only includes links to its child topics:



The same stub topic with the *@chunk* attribute and the value *to-content* added to it. The stub topic is automatically filled in with the information from the child topics:



Nest DITA-maps

To further enhance your structure, you can also use DITA maps nested within other DITA maps. This can help you organize your content more clearly by separating complex collections of topics into a number of smaller, simpler collections.

You can, for instance:

- organize your topics in separate DITA maps which represent different chapters. This way you enhance your structure to make it easier to navigate.
- re-use sets of topics. It is far easier to re-use a whole DITA map than it is to re-use every topic individually.
- divide your work load between colleagues. If everyone works in their designated DITA map, you decrease the risks of different writers working on the same file.

Important:

Avoid nesting bookmaps within DITA maps as the structure will cause confusion for both processing software and documentation authors. Bookmaps should only be used at the top level.

Guidelines for creating links

When it comes to creating links between topics, you should take into account the following guidelines:

- Do not overuse inline links [p. 21]
- Think about the phrasing and positioning of inline links [p. 21]
- Do not overuse relationship tables [p. 21]
- Add collection-type links [p. 21]

Do not overuse inline links

It is easy to overuse inline links in your text, making it confusing to the reader. It is important to use them only when necessary and keep in mind the following guidelines:

Avoid inline links to tables and figures in a topic

Keep your topics short and self-contained. These links can become difficult to maintain when you update your topics, for instance by deleting a table or moving it to a different topic. They can also make it difficult to re-use topics. Instead, you can consider re-using the table or the figure somewhere else in the form of a content reference (conref).

Create inline links to repeated steps

You might use phrases such as 'Repeat step 2 to add each topic to the DITA map'. In this case you will need to update this phrase every time you add a step before 'step 2'. You can solve this by adding an **‹xref›** element that links to the relevant step. In doing so, every time the numbering changes in the steps, the **‹xref›** element will ensure the text changes automatically.

Create inline links to high-level tasks

If you have a manual that contains a task flow with too many task topics, the user might start to lose track of the big picture. As these tasks might be spread throughout your DITA map, you might not be able to take advantage of the default linking system. In this case, you can use inline links to show users a roadmap of the task flow they will have to follow. This way, they can plan for tasks they will need to complete in the future.

Think about the phrasing and positioning of inline links

For all their advantages, inline links are not without their flaws. Tests conducted by Jared Spool support the view that links outside of the text are more effective than inline, embedded links. Spool's study concluded that:

- Links are less usable when embedded in the text.
- Longer links are more effective than shorter ones.
- People scan for target words - the scent of information.

Do not overuse relationship tables

Relationship tables are useful in linking content that does not get connected by the hierarchical linking system. Still, it is important not to start linking too much. What applies for inline linking, applies for relationship tables as well. Too many will only confuse the user.

A good general rule to keep in mind is that hierarchical linking is nearly error-proof and low maintenance. As such, it is a good system to rely on. Only start adding relationship tables if you need relevant topics of information to be linked for the user's convenience. When you do, follow these guidelines:

- Add links for sibling topics if you set the *@linking* attribute to *none* in the parent topic but want to have related links for a select few of the child topics.
- Add links between topics that would not otherwise be there in the hierarchy of links.
- Add links to topics outside of the hierarchy in the current DITA map.
- Keep the number of related links in each topic to fewer than five when possible.

Add collection-type links

A *@collection-type* attribute defines a relationship among nested topics and affects the way in which automatic links are generated in the output of your DITA map.

Elements to which you can apply the @collection-type attribute

You create a collection of links by setting the value of the *@collection-type* attribute, which defines how the nested topics are related. You can apply the *@collection-type* attribute to:

- **‹topicref›** elements in a DITA map.

Note: *These have to be the parent* ***‹topicref›*** *elements in the hierarchy.*

- ***‹topicgroup›*** elements in a relationship table.
- columns in relationship tables.

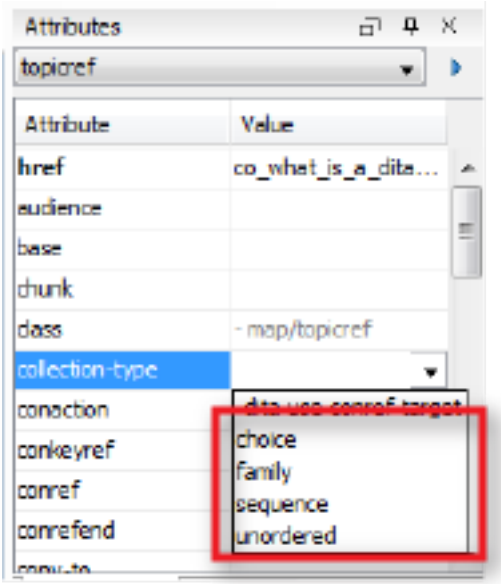
Note:

Although @collection-type attributes are valid in relationship table ***‹relcell›*** *elements, some don't make a lot of sense in that context.*

For example, if you set a @collection-type attribute in a relationship table cell to sequence, it will result in a ***Next Topic*** *link in the output linking to the next topic referenced in the cell, rather than to the next topic in the TOC sequence (as may logically be expected by the reader).*

Possible values of the @collection-type attribute

You can set the values of the *@collection-type* attribute to one of the four following values:



Value of *@collection-type* attribute	Description
choice	Not commonly used, but it is intended for situations where the reader needs to select one child topic to proceed. This may be useful when the output document is an interactive decision-support application. Most processors treat choice in the same way as unordered.
family	Links are generated from parent to children, from children to parent, and from sibling to sibling.
sequence	Links are generated from parent to children, from children to parent, and from child to previous sibling (if applicable) and next sibling (if applicable). It creates a numbered sequence.
unordered	Links are generated from parent to children, and from children to parent. These links can be completed in any order.

The following table can help you decide which type is most appropriate for your situation:

Question	Value
Should the nested task topics be completed in a particular order?	*sequence*
Do the topics represent choices?	*choice* or *family*
Can the task topics be done in any order?	*unordered* or *family*
Are the topics closely related and need links between sibling topics?	*family*

Different levels of content re-use

Re-using DITA maps

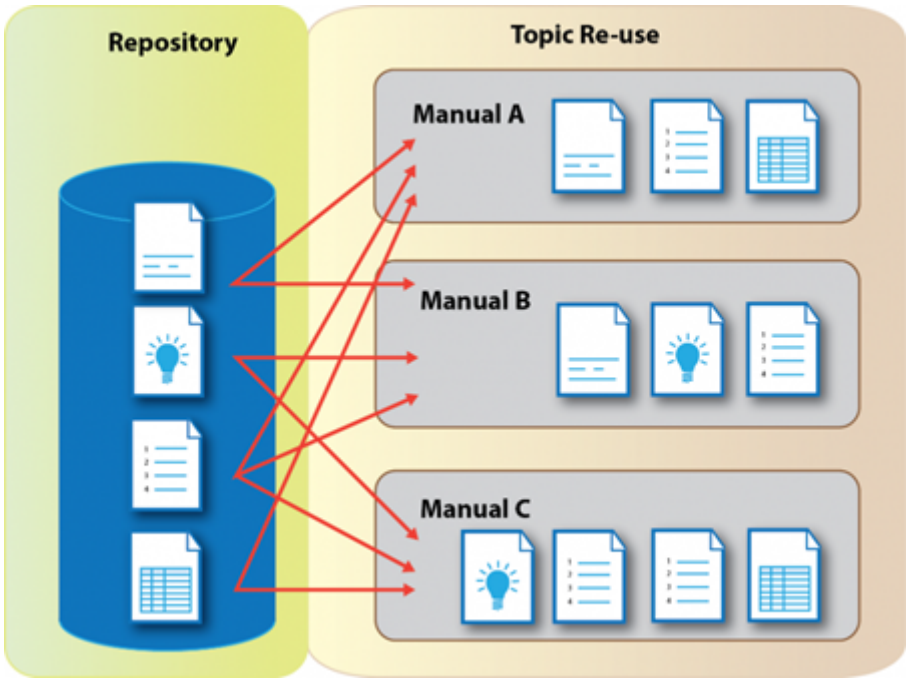
You can re-use entire DITA maps in your structure. To create re-usable DITA maps, create and organize topics into submaps that document common features or user tasks.

Example: Even if you have only one user guide, you can create one DITA map for installation topics, one DITA map for security topics, and one DITA map for configuration topics.

Re-using topics

Re-using topics is a very good way of making efficient use of your content. When you write your content according to the best-practice suggestions, your topics will be clear and self-contained, meaning they can easily be re-used elsewhere and still make sense.

The main way you can re-use a topic is by inserting it into multiple DITA maps. This way you essentially have a single repository of topics, and you link the ones you need in the relevant DITA map(s).



There are types of topics that respond very well to being re-used like this, for instance:

- Topics that contain standard text, such as legal information.
- Topics with a recurring information set, such as a procedure that must be done in multiple task flows or scenarios.
- Topics that need to appear in different sets of information. For example, when you need to add the same topic to a help system and to a PDF file for a user guide.

Tip:

If you have a similar topic but with content that needs to be customized, you can create a DITA template instead. Writers can re-use that template to make similarly structured topics but with unique content.

Note:

Using the same topic several times in the same DITA map can create problems when generating output (all links to the topic will point to the first instance of the topic).

*If you need the same topic to appear several times in the same map, you should use the @copy-to attribute. You enter a new filename in the @copy-to attribute of the ‹**topicref**› element. In this way, you create duplicate topic references in the DITA map without actually creating additional topics in the source files. You can then use the ‹**linktext**› and ‹**shortdesc**› elements in the ‹**topicmeta**› element of the topic reference to provide a unique title and short description to help you distinguish between the two in link previews.*

Using content references (conrefs)

Content references (conrefs) are generally the smallest type of content re-use as you only reference a single element of your content. However, the size of the element may vary greatly from one word to an entire section.

You cannot conref loose bits of content; the text must always be embedded within its element tags. This is because text on its own cannot have an ID to reference, only elements can hold an ID. This also means you are bound by the DITA structure rules and need to re-use an element within its appropriate content. You cannot, for instance, conref a ‹**step**› element (used only in task topics) in a concept topic.

Re-using elements

On the smallest scale, you can reference a single word or phrase. This should be done with care, however, as conreffing these can lead to issues such as ungrammatical sentences. To avoid any issues, it is safest to stick to fixed nouns, names and phrases that are unlikely to be modified by grammar, such as:

Commonly re-used elements	Examples
‹**cite**›	Titles of books
‹**filepath**›	File, directory or path names
‹**keyword**›	Product names, model numbers
‹**menucascade**› and ‹**uicontrol**›	Interface buttons, menus, check boxes, tabs
‹**wintitle**›	Names of windows, panels, pages or dialog boxes

On a larger scale, commonly referenced elements include:

- lists
- steps
- tables
- sentences
- paragraphs

Re-using content from non-DITA sources

You can re-use existing content such as product codes and external sources by automatically pulling it into your DITA topics. For example, instead of writing a programming reference guide from scratch, you might pull comments from the software code and use those comments in your DITA topics.

Because DITA is a semantic tagging language, it can easily map non-DITA content to the proper DITA elements and insert the content in the correct location in your DITA topics.

Glossaries through tooltips

Glossaries versus tooltips

If your documentation contains a lot of technical terms or acronyms, you can support your reader with a glossary. Glossaries are usually limited to static lists, added as a chapter at the end of a manual.

However, Oxygen's WebHelp outputs offer a solution whereby users are no longer forced to leave the page they are reading: glossaries through tooltips. Tooltips are one-term glossary pop-ups that appear on screen if you hover your mouse over the term or acronym. If you click the term, the glossary item opens as a separate topic.

Creating tooltips in Oxygen's online help output

Proceed as follows:

1 Create a glossterm topic [p. 25].

2 Add the glossary term to the glossary map [p. 25].

3 Reference the glossary term in a topic [p. 26].

Create a glossterm topic

1 Select ***File*** > ***New***.

2 In the **New** window, select ***Framework templates*** > ***DITA*** > ***topic*** > ***Learning*** > ***Glossentry [DITA / topic]***.

3 Click **Create**.

4 Fill out the **‹glossterm›** and the **‹glossdef›** elements.

5 Save the glossentry topic in your project.

Note: *You can give this topic any name you like, but for clarity's sake, we suggest you start its name with glos_.*

Add the glossary term to the glossary map

Before you begin:
Your publication needs to contain a glossary map. If this is not the case yet, add a ditamap to your publication.

Note: *You can give this ditamap any name you like, but for clarity's sake, we suggest you call it _Glossary.*

1 Open the *_Glossary* ditamap in the **DITA Maps Manager**.

2 Right-click the *_Glossary* ditamap and select ***Append child*** > ***Glossary Reference...***.

3 Select your newly created glossentry topic.

4 Click **Insert and close**.
Results: Oxygen adds the topic to the *_Glossary* ditamap as a **‹glossref›** element but gives an error message that the **‹glossref ›** element needs a key.

5 Open the publication or the *_Glossary* ditamap in the **Author** window.

6 Select the newly added ‹**glossref**› element.

7 Fill out its *@keys* attribute in the **Attributes** window.
Results: You can give this *@keys* attribute any name you like, but for clarity's sake, we suggest you give it the same name as its ‹**glossterm**› or title.

8 Save your changes.

Reference the glossary term in a topic

1 In a topic, select the words for which you wish to show a tooltip.

2 Surround them with a ‹**term**› element.

3 In the **Attributes** window, open the *@keyref* attribute list by clicking the downward arrow in the *@keyref* field.
Results: It lists all possible glossentries for your publication.

4 Select the appropriate value for the *@keyref* attribute.

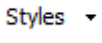
5 Save your changes.

Displaying topic titles instead of file names in a DITAMAP file

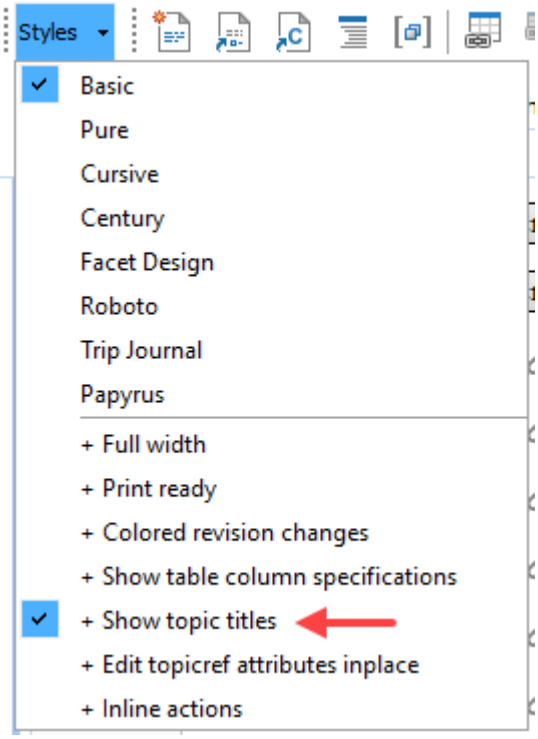
If you open a ditamap (or bookmap) in the **Author** mode in Oxygen, you will instantly notice that all the ‹**topicref**› elements are by default displayed according to their file name.

*To change the file names of ‹**topicref**› elements in a .DITAMAP file to their corresponding topic titles, proceed as follows:*

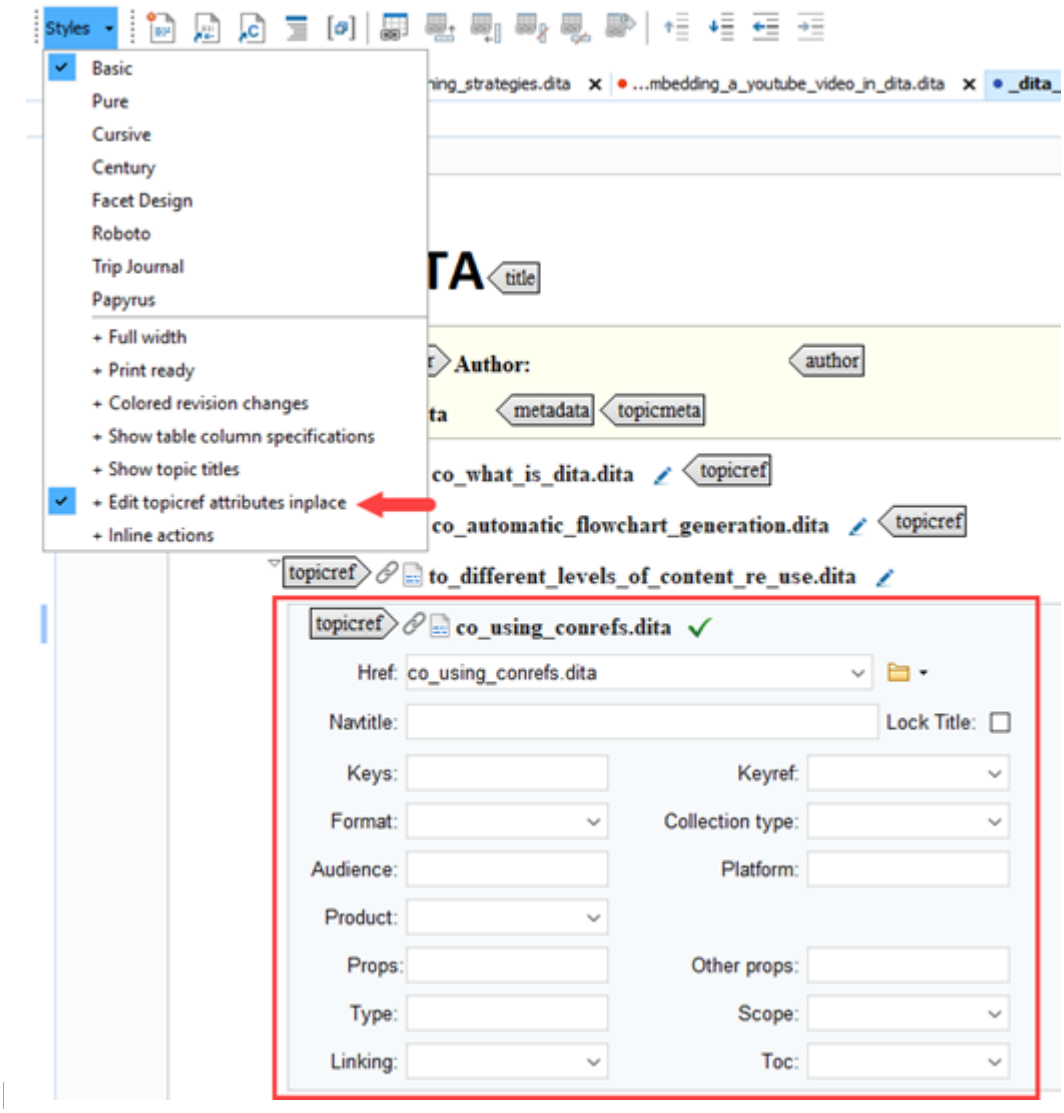
1 Select the **Styles** button in the toolbar:



2 In the selection menu that appears, select the option **+ Show topic titles**.



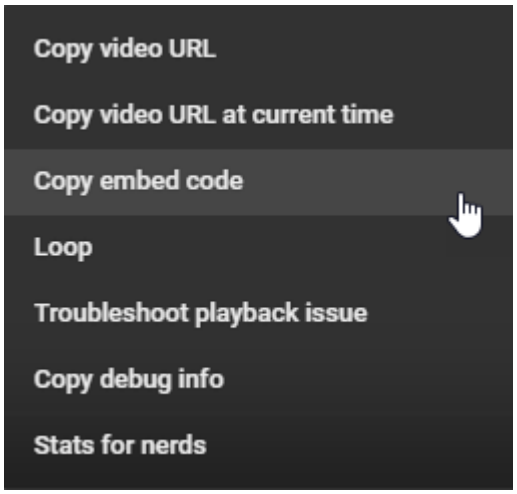
Tip: *Another useful option in the **Styles** selection menu is **+ Edit topicref attributes inplace** which allows you to adjust the attributes specific to each topic within the ditamap in which these topics are referenced.*



Embedding a YouTube video in a DITA topic

Proceed as follows to succesfully embed a YouTube video into a DITA topic:

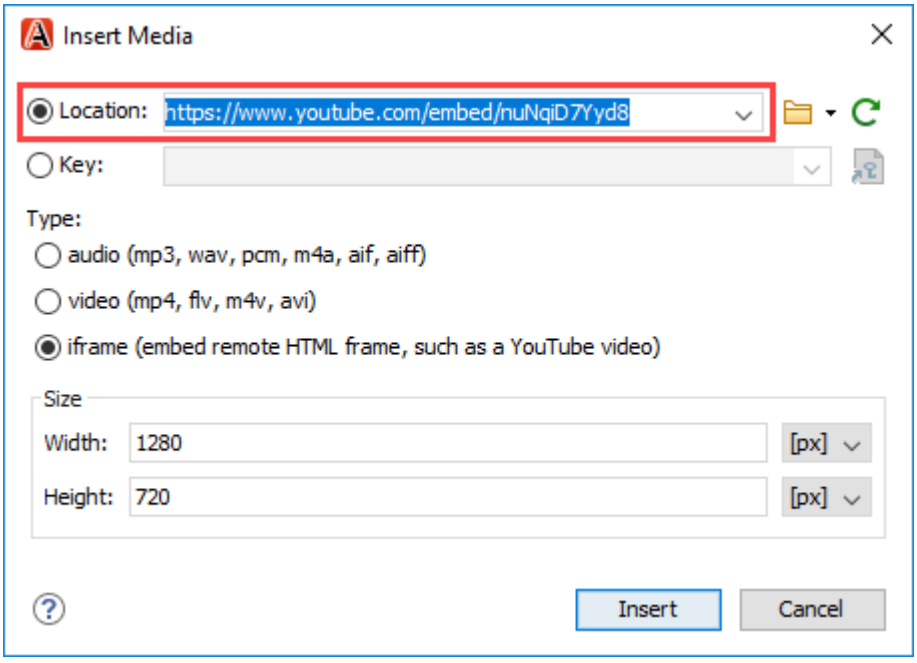
1. Find the video that you want to include in your DITA topic on Youtube and right-click it.
2. Select **Copy embed code** in the selection menu that appears.



3 Select

in the toolbar of Oxygen XML Editor/Author.
Results: The **Insert Media** window opens.

4 Paste the embed code into the field next to **Location**.
Results: The embed code will automatically convert to a URL.



5 Make sure to check the radio button next to **iframe (embed remote HTML frame, such as a YouTube video)**.

6 Select **Insert**.

Adding a Dutch dictionary to Oxygen's Spell Check

If you use OxygenXML to write Dutch texts, you can add a Dutch dictionary to the spell checker.

Proceed as follows to install the Dutch dictionary in Oxygen:

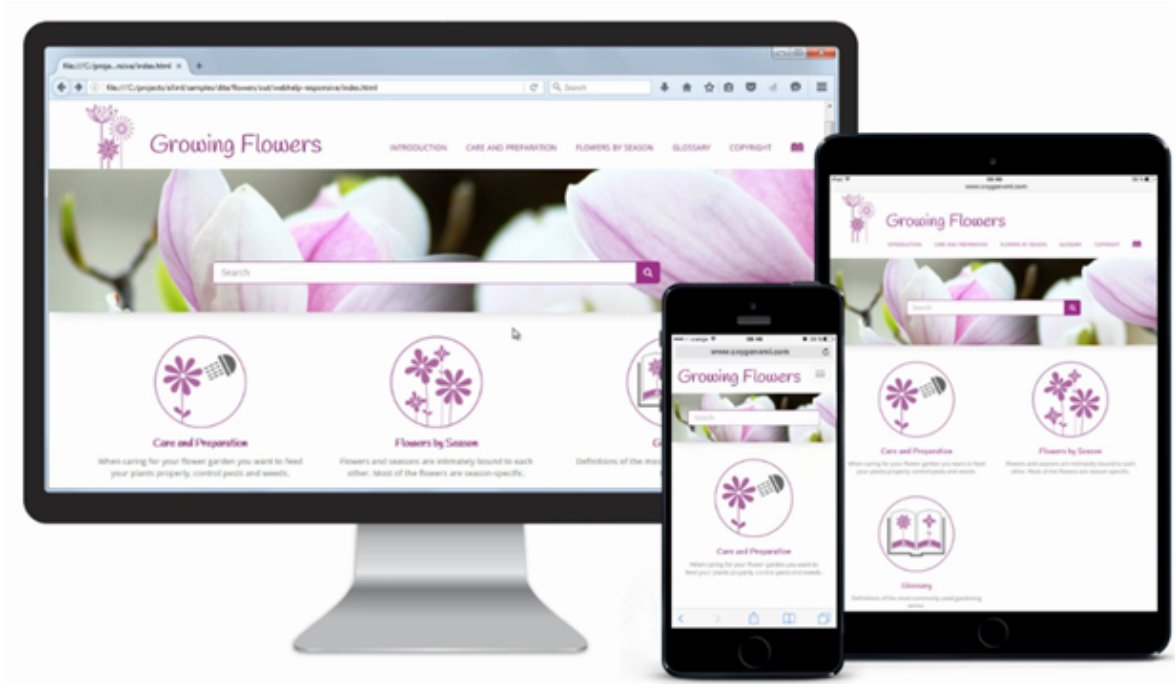
1 Locate the following file: *S: \ temp \ hunspell_NL*

2. Copy both files to *C: \ Program Files \ Oxygen XML Author 20 \ dicts \ spell*.
3. Restart OxygenXML.
4. Make sure the language of your topics is set to NL, so for example: <task xml:lang="nl">.
5. Do one of the following:

If you prefer ...	select ...
Automatic spell check	***Options*** > ***Preferences*** > ***Spell Check*** > ***Automatic spell check***
Manual spell check	***Edit*** > ***Check Spelling or Edit*** > ***Check Spelling in Files***

Responsive online help

One of the outputs you can select in Oxygen is **DITA Map Webhelp Responsive**. The name derives from the system's ability to adapt to any device or screen size to provide an optimal viewing and interaction experience.

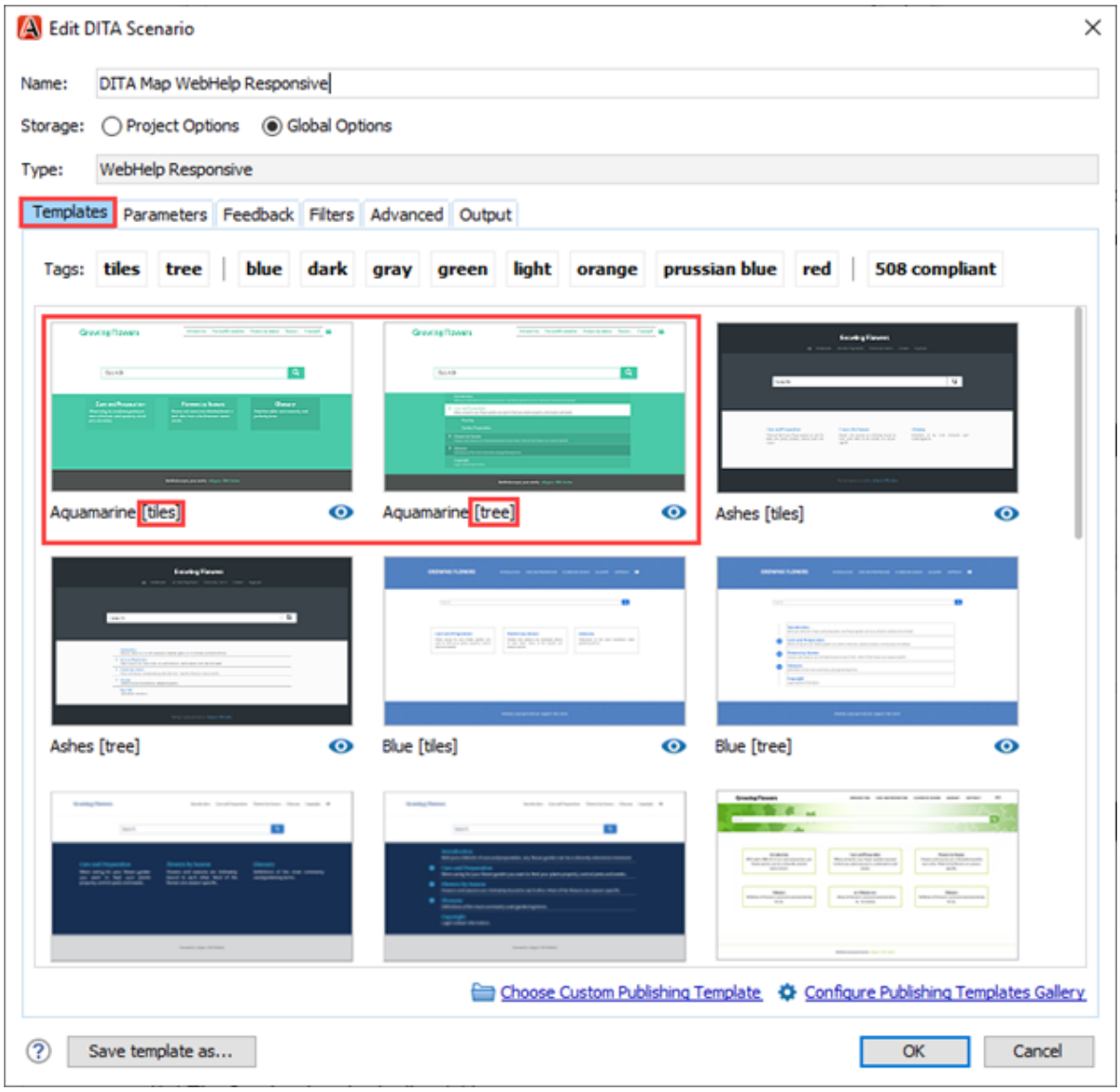


There are a variety of different templates from which you can choose (in the **Edit DITA scenario** window) to create your responsive online help.

Note: *You also have the possibility to use your own custom publishing template.*

The templates exist in two versions:

- The first-level topics in tiles.
- The first- and second-level topics as a tree.



For more information, see https://www.oxygenxml.com/doc/versions/18/ug-editor/topics/webhelp-responsive-description.html.

Selecting DITA Map Webhelp Responsive as output format

Proceed as follows:

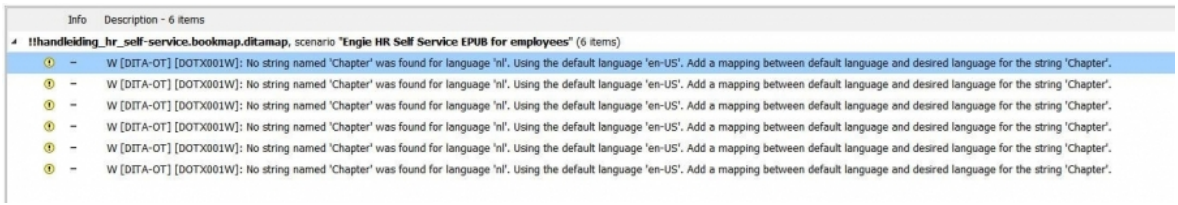
1. Click

2. Select **DITA Map Webhelp Responsive**, with or without **feedback**.
3. Click **Duplicate**.
4. Give your copy a name.
5. In tab **Templates**, select your skin.
6. Adapt the other tabs where necessary.
7. Click **OK**.

Oxygen Webhelp and EPUB transformation error

Condition

When publishing a project using a transformation scenario, you might see an error message like this:



Cause

It is possible that the language you are writing in, is not defined in OxygenXML. In that case, words such as Chapter, Search and Part might not be translated when creating output.

Remedy

In order to have a completely localized webhelp or EPUB file, follow this procedure:

1. Make sure that all the topics in your project have the correct *@xml:lang* attribute.
2. Look for the *strings-[lang1]-[lang2].xml* file in *DITA-OT-DIR/plugins/oxygen-webhelp-responsive/resources/localization* folder.
   Example: The Canadian French file would be: *strings-fr-ca.xml*.
3. If the *strings-[lang1]-[lang2].xml* file does not exist, use the following procedure to create one:

a. Copy the *.../localization/strings-en-us.xml* file. Change the name and open it.
b. Add all the labels found in the *DITA-OT_DIR/xsl/common/strings-en-us.xml* file and paste them into your file.
c. Translate all the labels from the above language file. Labels are stored in XML elements that have the following format: <str name="Label name">Caption</str>.
d. Add the new language to the *strings.xml* file in the *localization* folder by adding <lang xml:lang="nl-nl" filename="strings-[lang1]-[lang2].xml"/>.

4. Run Oxygen as administrator and run the predefined transformation scenario called **Run DITA OT Integrator** by executing it from the **Apply Transformation Scenario(s)** dialog box.

Note: *If the integrator is not visible, select the* ***Show all scenarios*** *action that is available in the* ***Settings*** *drop-down menu.*

5. Edit the **WebHelp transformation scenario** and set the *args.default.language* parameter to the code of the language you want to localize.
   Example: *fr-ca* for Canadian French
6. Run the transformation scenario to produce the WebHelp output.

Detecting and fixing errors with Quick Fixes

Oxygen XML Author 17 now has *quick fixes*. Quick fixes can help you quickly check if content adheres to the so-called *business rules*. These are rules for writing and structuring content, much like the rules you can find in style guides.

Quick fixes could work as follows:

1. Someone (for instance, an Information Architect) writes business rules.

   Note: *For example: "A semicolon cannot be used at the end of a list."*
2. An XML-specialist turns the business rule into a *Schematron pattern*.

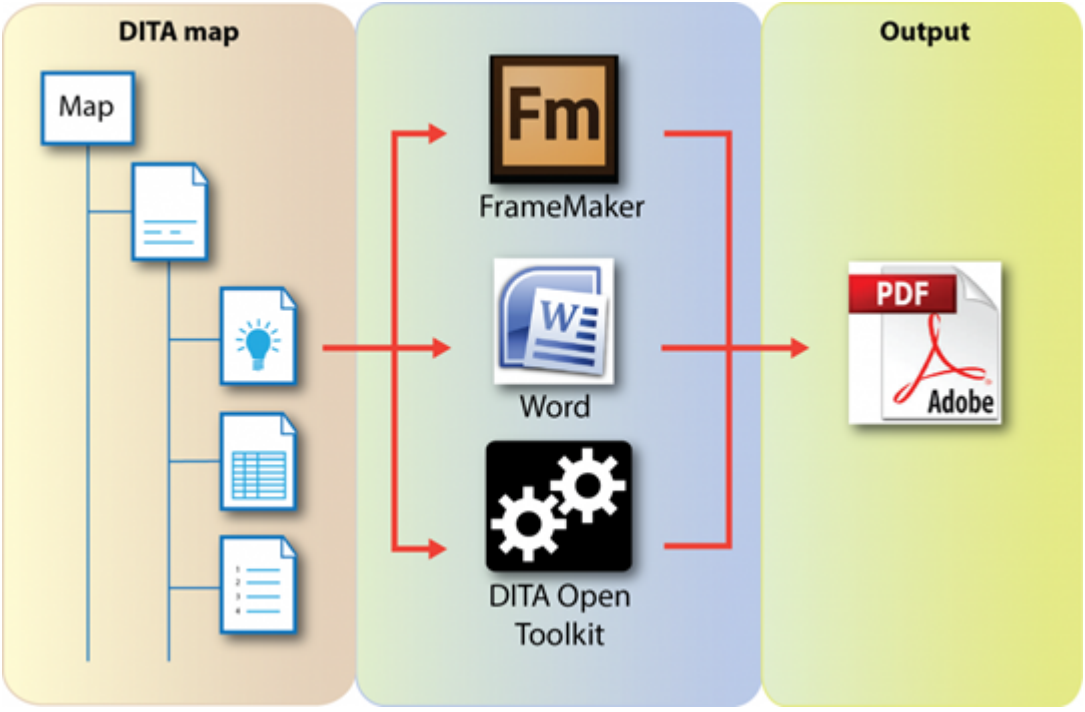
3 This schematron pattern can be used to automatically detect instances in which a technical writer has not adhered to (one of) the predefined business rules. The technical writer can then quickly find excerpts that don't match the business rules and instantly correct them by clicking on the light bulb that appears before the problem in question.

More info: http://blog.oxygenxml.com/2015/05/schematron-checks-to-help-technical.html

4.3.2 Publishing DITA content in Oxygen XML

Two main publishing strategies

There are several ways of publishing your documentation:



These ways can be grouped into two main methods for publishing documentation:

1. The first method is by using the DITA OT (Open Toolkit) that is embedded in many DITA editors, such as Oxygen XML Editor. This makes the publishing process easier as you can apply the type of transformation you want directly to your files, using the same program in which you've written them. Your processor will then run the files through the toolkit and generate the output you want.

   While using the OT is easy at first, it may require a lot of work and understanding of the processes involved to achieve a proper result. More often than not, the output from the OT gives your text a very basic lay-out. When you write and publish for different clients and different products, you might need varying lay-outs. Therefore, you might need to adjust the process manually to achieve the desired result.
2. A second method that avoids this issue is DITA2PDF . Essentially, it involves using an intermediary program in which you can adjust the template of your lay-out accordingly before you publish to PDF. While it might still require some searching and puzzling, intermediary programs such as FrameMaker and Word tend to be more user-friendly and allow you to apply your desired lay-out template in one go.

Publishing your DITA content with DITA OT

With the help of Oxygen, you can write one file and publish it to multiple output formats, such as:

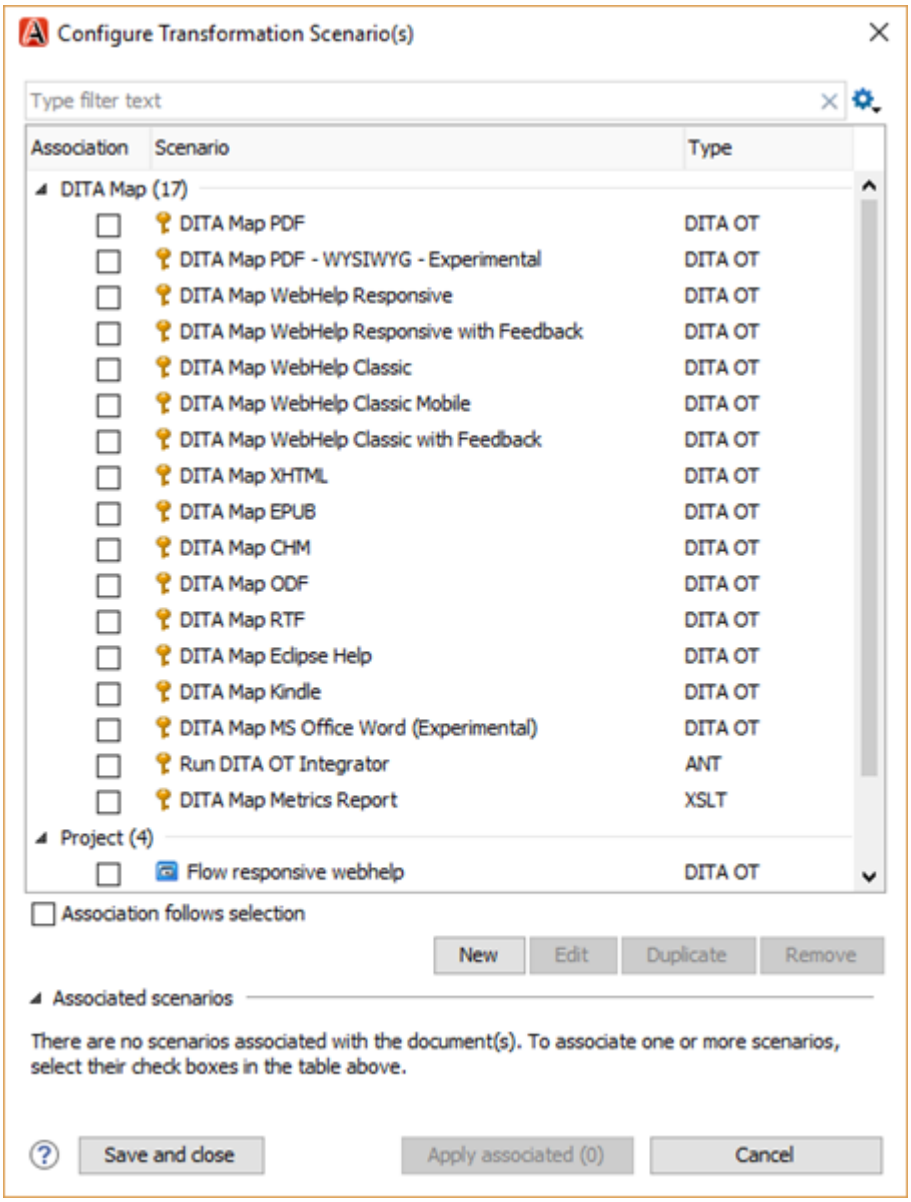
- PDF
- DOCX
- EPUB
- Online help
- etc.

To make outputs in different formats:

1. Click

   Results: The **Configure transformation scenario(s)** window opens.



2 Select all transformations you want to associate with the Oxygen project and click **Apply associated**.
Results: The chosen transformations will be applied to the current project.

Publishing DITA files to PDF with DITA2PDF

DITA2PDF is a method of publishing DITA files to the PDF output format. Briefly summarized, you write texts in Oxygen and you publish them via FrameMaker:

1 Write your content in Oxygen.

2 Install the ditafmx-plugin for FrameMaker.

3 Use the plugin to import your Oxygen files into FrameMaker.
Results: The files will be transformed into FrameMaker files and a template (including layout) will be applied to the files.

4 If necessary, make some last-minute modifications to the text.

Note: *Things to watch out for:*

- Adjust the images to the right resolution.
- Resize the tables.
- Check if any spaces have disappeared (this can happen next to inline images).
- Check if all conditional text is correct.
- etc.

5 Publish the document to PDF.

Conditional processing

In the topics listed in the table below, you will find an answer to the following questions:

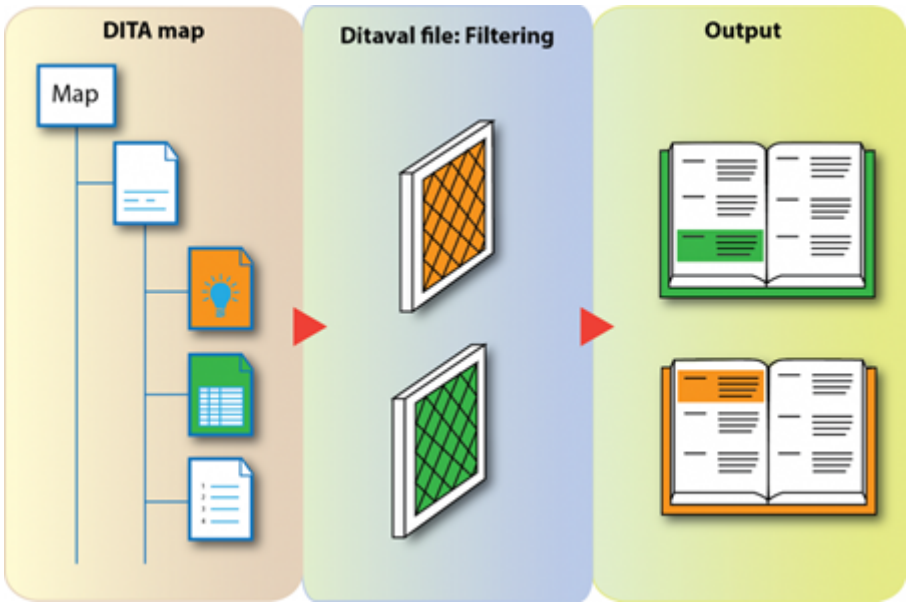
Question:	Topic:
What is conditional processing?	What is conditional processing [p. 35]
What is a DITAVAL file?	What is a DITAVAL file [p. 36]
How does conditional processing work?	Using DITAVAL files in DITA [p. 38]
Why use conditional processing?	Purpose and advantages of conditional processing [p. 38]

What is conditional processing

Definition

Conditional processing is a method through which content can be included, excluded or highlighted in the output, allowing you to create different outputs from a single source file.

In essence, conditional processing is a form of content re-use . However, rather than using one item of information (through a conref) in different files, you contain everything in one file and filter out what you don't need. It is at its most useful when you write nearly identical information for very similar products, so you do not need to write and maintain separate but highly similar manuals. Conditional processing allows you to write just one manual which contains everything, and then highlights or filters out the necessary information during the publishing process.



Highlighting and filtering content is done by applying attributes to the information that differs per product/ model. These attributes can then be used to:

- Highlight the content as only being valid for a certain version or model (for example: information that only applies for Windows or Mac).
- Filter out that content when output is being generated.

What is a DITAVAL file

A DITAVAL file (or a "conditional processing profile"[1]) is used to filter your DITA content depending on the type of:

- audience that your content is addressing: e.g. a novice audience, an expert audience ...
- product that your content is about: e.g. IPhone X, IPhone 11, IPhone 11 Pro ...
- platform that your audience is working with: e.g. iOS 12.4, iOS 13 ...
- ...

Example of a DITAVAL file in Oxygen XML

Within the DITAVAL file, you establish rules to include, exclude or flag DITA content that has been attributed a particular value.

This is what a DITAVAL file might look like:

- in the **text** view of Oxygen XML:

```
<val>
   <prop action="exclude" att="audience" val="novice"/>
   <prop action="flag" att="audience" val="general"/>
</val>
```

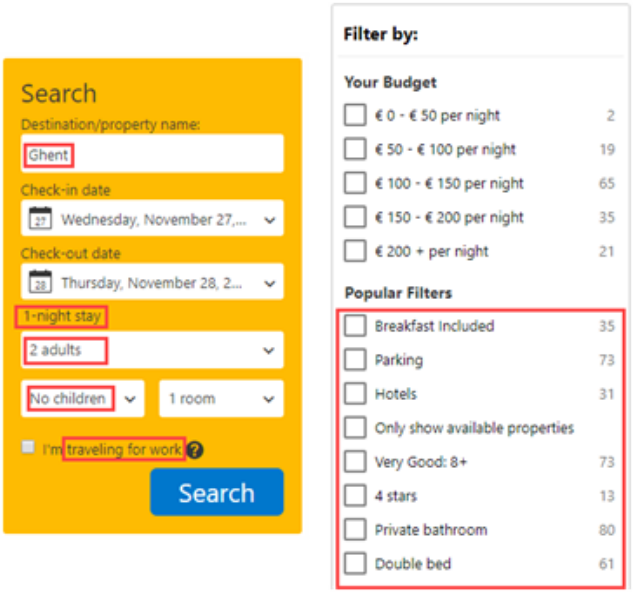
- in the **Author** view of Oxygen XML:

The *@action* attribute determines what should happen with the various attribute values of the *@audience* attribute. In this case, the DITAVAL file will exclude all content, that is applicable to a novice audience, from the published output (in other words: all topics or elements that are attributed the attribute value *@novice* for the *@audience* attribute are exluded from the publication.). On the other hand, all topics or elements that are connected to the *@general* audience are not only included, but also highlighted in the publication.

Four possible actions of the DITAVAL file

A DITAVAL file defines four different conditional processing actions in an *@action* attribute:

1 https://www.oxygenxml.com/dita/1.3/specs/langRef/containers/ditaval-elements.html

Value of *@action* attribute	Description
Include	This action will include the content in the output.
Exclude	This action will exclude (filter) the content from the output.
Flag	This action will include and highlight the content in the output.
Passthrough	This action will: • include the content in the output • preserve the attribute value as part of the output stream for further processing within the (dynamic) output. In other words, the conditional processing (or filtering) does not take place during the publishing phase of the DITA content. Instead, the various types of content and their corresponding attribute values are passed on to the output itself, which is where the filtering will take place. ***Note:*** *Take for example a search engine with filtering options. A customer of Booking.com will be able to efficiently search through the many possible accomodations that the site offers by using "filters," which are essentiallly attribute values that have been attributed to the various types of content:* 

1

Applying the DITAVAL file to your DITA map

1 Source: https://www.webworks.com/Documentation/Reverb/index.html#page/Preparing%20and%20Publishing%20Content/Preparing%20DITA%20Files.3.07.htm

Using DITAVAL files in DITA

Conditional processing means you apply attribute values to certain elements of your text. These values are then entered into a separate DITAVAL (DITA values) file in which you write rules that specify whether elements with those values need to be included, excluded (filtering) or flagged (highlighting). While generating output, the rules in this DITAVAL file are applied to the end-result.

Below, the three main steps in conditional processing are described.

Planning an attribute scheme

Before worrying about applying conditions, you should first establish a scheme and define exactly what you need to create conditions for. This way you can plan out which attributes you need and which values to assign to each of them. To do this, define your:

1. Audience: who will use your product and what specific information do they need?
2. Platform: which operating systems, platforms and other environments are used with your product?
3. Product: which models/versions are you planning and how do they differ?

⚠ ***Attention:***

Each of the items you define should receive a unique value in an attribute. Do not re-use value names in different attributes as this will lead to confusion for your writers.

Example: You want to teach your users how to save their files using Microsoft Word, but you know there are several version of Word on the market. As your documents only deal with Word 2003 and Word 2007, you plan your conditional attributes and values accordingly:

Purpose	Attribute	Value
Only for Word 2003	*@platform*	2003
Only for Word 2007	*@platform*	2007

Applying attributes

Once you have specified which values you need and in which circumstances you will need them, you can start applying them as you write. Applying them while writing is far easier than having to hunt through your documentation afterwards to find the elements that are product-, audience- or platform-specific.

You can apply conditional attribute values in:

Element	Purpose
Topics	Include, exclude or flag specific elements
DITA maps	Include or exclude entire topics

Note: *It is recommended to apply attributes to the topic references in the DITA map, and not to the* ***‹topic›*** *element within the topic in question.*

Purpose and advantages of conditional processing

Purpose

When you write documentation for a certain series of products, several products can share a lot of the same information. For instance, the task describing how to install an appliance can largely be shared between different models or versions of that appliance. At the same time, each model might have its own unique requirements for the installation procedure.

Rather than having to rewrite the whole procedure for each model, you can include all information in one topic or map and apply conditional attributes to the model-specific information. When you publish this information later on, you can use these attributes to either highlight relevant information per model or filter out unwanted information.

Advantages

Concretely, conditional processing allows you to:

- Create different outputs from a single topic by filtering out unnecessary information for each output type.

  Note: *This way you only have to maintain one core set of topics. This method automatically ensures that all generated versions of the core set of topics contain the same (up-to-date) shared content.*
- Highlight specific content.

  Note: *For instance: information that is only relevant to a specific version of a program.*
- Improve how writers find specific content.

  Note: *Writers can find topics in their content management systems by searching for information about the topics, such as feature numbers or product names.*

4.3.3 Oxygen XML software updates

Syncro Soft regularly releases new versions of its various software applications (including Oxygen XML Editor, Oxygen XML Author, …), often with a wide range of new features and productivity improvements.

- New in Oxygen 21.1 [p. 40]
- New in Oxygen 21.0 [p. 39]
- New in Oxygen 19.0 [p. 39]

New in Oxygen 19.0

Text: https://www.oxygenxml.com/xml_editor/whatisnew19.html

If you do not have time to view the entire video, use this list with time stamps to view the topic you are interested in.

- [0:30] Create (multiple) new topics at once and automatically add them to your DITA map
- [1:00] Convert topics to a different information type
- [1:23] DITA ‹**keydef**› elements presented in a separate pane
- [1:45] Autocompletion in the webhelp search
- [2:01] Search for phrases in the webhelp by putting the text in quotes
- [2:24] Oxygen Chemistry (XML + CSS = PDF) makes it easier to style and customize your PDF output
- [3:07] Multimedia support: easily insert audio clips and video into your topics at the click of a button

New in Oxygen 21.0

New in Oxygen 21.0

Text: https://www.oxygenxml.com/xml_editor/whatisnew21.0.html

If you do not have time to view the entire video, use this list with time stamps to view the topic you are interested in.

DITA Authoring Improvements

- [0:24] Search for keys in the Dita Reusable Components view with the improved filtering options.
- [0:42] When inserting an image, edit the attributes and reference of an image by using the *@keyref* attribute.
- [0:51] See a preview of the image that you are referencing using the *@keyref* attribute in the in-place attributes editor or the **Attributes** view.
- [1:04] Specify conditional processing attributes using grouped values, when a particular attribute applies to multiple specialized subcategories.
- [1:20] Oxygen now includes version 3.2.1 of the DITA OT publishing engine.

Web Author Enhancements

- [1:28] Use the **DITA Keys** toolbar button in Web Author to quickly find a particular key and insert a reference to it in your document.
- [1:48] Integrate Web Author with file repository services such as Alfresco, Perforce Helix etc.
- [2:03] New document template in Web Author to create markdown documents.
- [2:20] Design improvements for working with tables in Web Author.

CSS-based PDF Publishing Features

- [2:50] Add an edit link for each topic in the PDF output.
- [3:05] Display tracked changes and comments in the PDF output.

JSON New Features and Improvements

- [3:26]**Outline** view was redesigned for JSON documents.
- [3:38] A more robust internal model for JSON editing support.
- [3:52] Addition of intelligent JSON schema-driven content completion assistant.
- [4:10] Support for using XSLT processing to transform JSON document to various formats.

XSLT & XQuery Improvements

- [4:29] Content completion offers proposals for options and funtions.
- [4:44] New XPath 3.1 functions in the content completion of XSLT documents.
- [4:54] New XPath 3.1 proposals in content completion of XQuery documents.
- [5:06] Use a Saxon configuration file in the **Compile XSL Stylesheet for Saxon** tool.

New in Oxygen 21.1

Version 21.1 of the Oxygen XML Editor comes with a range of new features and a lot of new improvements.

Some examples include:

- the introduction of HTML5 support, with specialized text-based editing and validation support
- a new **Resource Hierarchy/Dependencies** view
- the ability to quickly find similar topics
- the possibility of adding a mini-TOC at the beginning of each chapter
- the inclusion of Oxygen PDF Chemistry as the default PDF publishing engine
- ...

To see a complete overview of everything new in Oxygen XML 21.1, visit https://www.oxygenxml.com/xml_editor/whatisnew21.1.html.

5 DITA Open Toolkit (OT)

5.1 What is the DITA Open Toolkit (OT)

The DITA Open Toolkit (OT) is a free publishing engine that uses a Java-based software tool for automating software build processes. The OT allows you to publish your documentation into many varying output formats. It does this by interpreting your DITA maps, resolving *@conref* attributes and cross-references, applying formatting numbering and labels, and mapping DITA elements to output format elements.

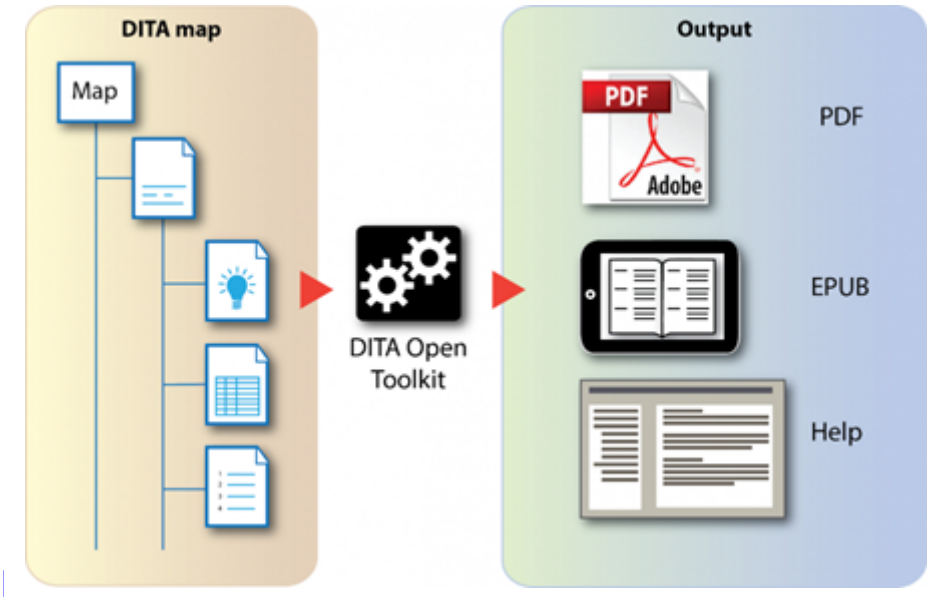
As it is an open-source use of the DITA standard, it does not produce a ready-to-use product which delivers high-quality output 'out of the box'. Instead, it should be considered as a good starting point for publishing, but usually more effort is required to create a finished product.

As the DITA OT is an open-source engine, it is often integrated into DITA editors such as Oxygen, XMetaL etc. As such, it allows you to generate output in the same program in which you wrote your documentation.

5.2 What does the DITA Open Toolkit (OT) do

5.2.1 What it does

The main goal of the DITA OT is transforming DITA content (maps and topics) into output formats such as PDF, XHTML, Help formats… This is often done by using a DITA editor such as Oxygen which has already integrated the OT into its programming.



All transformations require a base XML document (like what you see in the Oxygen Text view). This document is then transformed according to an XSL stylesheet, which contains your information and describes how it needs to be displayed. To this end, an XSL consists of three parts:

- XSL-T (XSL Transformations): a language for transforming XML documents
- XML-Path (Path Language): an expression language used by XSL-T to access or refer parts of an XML document
- XSL-FO (XSL Formatting Objects): an XML vocabulary for specifying formatting semantics

5.2.2 What it doesn't do

As the DITA OT is a free, open-source engine, it will not instantly give you a usable end product. More specifically, when you generate a PDF from the OT, the PDF will display a very basic lay-out of low to medium quality. In order to change this into high-quality material, you need to customize the OT as well as engage in some XSL-FO development work.

5.3 Webcast: Localization and the DITA Open Toolkit

This webcast discusses the use of the DITA Open Toolkit during the localization process:

6 DITAToo DITA CMS

6.1 What is DITAToo DITA CMS

DITAToo is a DITA content management system (DITA CMS) developed by Intuillion[1].

DITAToo consists of three major components:

1. DITAToo Content Repository: a database where you can store all your DITA content.
2. DITAToo Admin: an application, used to perform administrative tasks, such as:
   - creating user accounts
   - specifying access permissions for each user
   - specify translation languages
   - ...
3. DITAToo Author: an application that you can use to remotely connect to the DITAToo Content Repository and work on your DITA content.

DITAToo also allows you to automatically convert Word files to DITA files and it facilitates tasks like publishing DITA content, assembling maps, translation management …

To learn more about DITAToo and its various different features, you can visit http://www.ditatoo.com/products/overview/.

6.2 Automatic flowchart generation

With DITAToo, you can automatically generate flowcharts based on the DITA troubleshooting topics you have written:

If you are interested in other possibilities on content automation with respect to DITAToo DITA CMS, see also http://www.ditatoo.com/products/automated-content-assembly/.

TransforMagic from Intuillion

Info: http://intuillion.com/products/content-automation/videos/

1 http://www.ditatoo.com/

sectionDef = "firstSection" pageDef="Left"