Technical Content Development Guide

Contents

Document History

Version	Updates	Date
A	Initial document version.	October 27, 2020
B	Updated links.	October 30, 2020
C	Added About the Technical Content Development Guide on page 5.	February 24, 2021

About the Technical Content Development Guide

Learn the standards Jorsek applies when developing technical content.

With many writers contributing to the content library at Jorsek, it's important to have a well-defined content development guide. Having a unique content development guide promotes consistency and reduces questions about style, structure, and process.

As the makers of a DITA Component Content Management System (CCMS), we are especially aware of the benefits that structured authoring provides. We also understand that moving to a structured authoring environment requires a huge shift in processes and strategies. Not only that, but the DITA specification includes hundreds of elements and attributes that are available to use in topics and maps. Having a content development guide is not only critical to creating consistent documentation but also to the effective use of DITA and easyDITA features.

We've made our Technical Content Development Guide public in order to provide our customers, and DITA users in general, with examples of considerations for use in their own content development guides. The guidelines in this guide were made with our content needs in mind. We hope our guide can help you create a content development guide that fits your organization's needs.

> ⊙ **TIP:** You can download the DITA source of this guide, upload it to your easyDITA instance, and modify accordingly.

Jorsek Style Guide

Maintaining a style guide promotes consistency and reduces questions about style.

> ⊘ REMEMBER: We develop the Jorsek Style Guide with our content needs in mind. We recommend developing your own style guide that aligns with your organization goals.

Acronyms

An acronym is the shortened form of a word or phrase.

Guidelines

- If the acronym is specific to an industry or software and a reader may not recognize it, write it out.
- For every topic, write out the full term, followed by the acronym in parentheses. As you continue writing, you can use the abbreviation in place of the full term.

Example

Each document needs a corresponding Document Type Configuration (DTC). The DTC tells easyDITA what documents to associate with the configuration.

Capitalization

Limit capitalization to proper nouns.

Guidelines

- Capitalize names of people, places, organizations, and interfaces.
- Capitalize names the way they were intended to be capitalized, regardless of their location in the sentence. You can start a sentence with a lowercase proper noun.
- Capitalize topic titles, figure titles, table titles, and section titles.

Examples

easyDITA can be integrated with oXygen.

The **Dashboard** interface enables you to...

Diction

Preferring some terms over the others makes consistent documentation.

General Terms

Preferred	Non-preferred	Example
shows, appears	displays	The element appears.
enable	allow	The feature enables you to do something.
file name	filename	Enter a file name.
image	picture, photo	Select an image.
screenshot	screen capture, screen image	Upload a screenshot.
want	wish, desired	Select the option that you want to enable.

Windows and Panes

Term	Usage	Example
window	large interface areas that contain buttons, fields, and a close icon.	In the **Bulk Change** window...
dialog	small interface areas, usually for errors or confirmations	In the **Insert** dialog...
pane	distinct interface areas grouping related items typically placed on the right or left	In the **Activity** pane...

Selectable Areas

Term	Usage	Example
drop-down menu	selectable list items in a menu	From the the **More** drop-down menu, select...

Term	Usage	Example
check box	selectable boxes	Select the Owner check box...
field	area where you can enter text strings	In the Keys field, enter...

Actions

Term	Usage	Example
click	interactions with buttons, icons, or links	In the content library, click the Open icon.
right-click	opening a contextual drop-down menu	In the content library, right-click a file and...
select	interactions with a drop-down menu	In the More drop-down menu, select...
select and clear	interactions with a checkbox	Select the check boxes next to the files you want to rename.
enter	typing text into with a text area	Enter a title for your file.
navigate	find a file or folder	Navigate to the image you want to upload.
place	cursor location	Place your cursor where you want to insert a table.

File Extensions

File extensions use different style guidelines depending on the specificity of what you are referencing.

Guidelines

- If you refer to a file type, use all uppercase

  For example, write "Upload a PDF."

- If you refer to a specific file, use lowercase for the file extension

  For example, write "Include `image.png` in the folder."

Keyboard Key Names

Introduce keyboard key names in the correct DITA elements.

Stylistic Guidelines

- Use abbreviated keyboard key forms
- Capitalize the first letter of a keyboard key name

  > ⓘ NOTE: Some publishing scenarios may automatically capitalize entire keyboard key names.

- Prefer ENTER over RETURN

DITA Guidelines

- To write about a single keyboard key, use the shortcut element nested in the UI control element
- To write about keyboard shortcuts, use the he shortcut element nested in the UI control element, and menu cascade element

Examples

ALT

CMD

To copy the selected text, press CMD → C .

Numbers

Using specific guidelines for numbers and digits makes scanning for information easier.

Date and Time

- Use a 12-hour time format and include the 12-hour period and timezone. The timezone should be abbreviated and in all uppercase.

  For example, "The meeting is at 1:00 pm EST."

- Use a lowercase acronym after the numeric time.

  For example, "The revisions were made on 01-31-2030."

- Use a MM-DD-YYYY format for dates or write the date out in full.

  For example, "The meeting is on January 31, 2030."

Digits

- Spell out numbers one through nine

  For example: "Select two options."

- Use numerals for numbers 10+

  For example, "Select 12 files in the file listing."

- Use a comma in numbers of four digits or more

  For example, "There are over 5,000 combinations."

Punctuation

Jorsek follows The Chicago Manual of Style, with some notable punctuation style considerations outlined for reference.

Bullets

- Use a bullet list when organizing three or more items into a list makes more sense.

- Capitalize the first letter of each list item.

- Use parallel structure when writing a bullet list. For example, start each list item with a verb.

- Do not add a period to the end of any list item, even if the list items are complete sentences.

  The only exception is when a list item includes more than one sentence. In this situation, you must add an end punctuation to all sentences in the list item and all list items in the bullet list.

Example

The interface enables you to:

- View activity

- Filter events

- Search keywords

Commas

The Oxford comma is the final comma in a series of three or more items. Using the Oxford comma helps to reduce ambiguity in writing.

- In a series of three or more items, use a comma before "and" and "or"

- Use a comma to connect two independent but connected clauses into a single sentence

- Do not use a comma to connect two independent thoughts

Example

To reduce the ambiguity and make it clear that the dedication is to the writer's parents, to Leroy, and to Joanne, we use the Oxford comma:

"I dedicate this book to my parents, Leroy, and Joanne."

In the following example, we don't know if the book is dedicated to the writer's parents, Leroy and Joanne, or to the writer's parents, as well as Leroy and Joanne.

"I dedicate this book to my parents, Leroy and Joanne."

Periods

- Always use a period at the end of a complete sentence. The only exception to this rule is

bullet lists.

▪ Place periods inside quotation marks at the end of a sentence, unless the parenthetical content is nested in another sentence. In that case, the period should go outside of the quotation marks.

Voice

We want our content to be simple, objective, and professional. However, we also want to maintain a conversational and engaging tone.

Guidelines

▪ Use active voice

For example, write "We recommend that you do not edit the underlying code." instead of ~~"It is recommended that you do not to edit the underlying code."~~

▪ Write in the second person narration

For example, write "You can use the **History** tab to restore a file to the previous state."

▪ Keep it simple and direct. Be conservative in adjective and adverb usage, and use simple, common words.

For example, write "Now..." instead of ~~"Once you have done that..."~~

Jorsek Information Model

The Jorsek Information Model contains guidelines for structuring content at Jorsek.

> ⚠ REMEMBER: We develop the Jorsek Information Model with our content needs in mind. We recommend developing your own information model that aligns with your organization goals.

Topic Types

Each topic type has a specific structure.

concept topic

Information that helps users understand information about an idea or feature.

task topic

Procedural information that helps users accomplish a goal.

process topic

High-level procedural information that groups task topics.

reference topic

Reference information, typically organized into tables, that users can refer to when completing a task.

FAQ topic

Answers to frequently asked questions asked by users.

troubleshooting topic

Troubleshooting steps that help users find a solution to software issues.

glossary entry topic

Term definitions and terminology information that define terms unfamiliar to new users.

Concept Topic

Concept topics contain the essential and complete information needed to understand a concept.

Concept topics answer the following question: "What is *x*?"

Title Guidelines

- Use title case
- Use nouns or noun phrases

Examples: "Account Summary", "Projects Interface"

Style Guidelines

- Convey the essential information about a concept
- Provide only conceptual information, not reference or procedural information
- Focus only on one particular idea

Considerations

Ask yourself the following questions when creating concept topics:

- Is there relevant background information a reader needs?
- Are there technical notions for understanding *x* that need definition?
- What are *x*'s most important features or characteristics?
- Would an example help clarify what *x* is?

Structure

concept topics can contain the following elements:

- concept element
  1. title element
  2. abstract element
     - short description element
  3. prolog element
  4. concept body element
     - section element
     - paragraph element
     - example element
     - definition list element

- ▪ unordered list element
- ▪ ordered list element

Elements

title element

Entitles a topic, a section, or a container element.

abstract element

Provides introductory content that would be unfit for a short description element.

short description element

Illustrates the topic purpose in two or three sentences (no more than 50 words). short description elements can provide content for link previews and search engines.

prolog element

Contains topic metadata. Can contain multiple resource ID elements that you can use to implement contextual helps into applications.

concept body element

The main element of a concept topic.

section element

Contains relevant content.

paragraph element

Encapsulates a single idea in a paragraph.

example element

Illustrates or supports a topic by providing relevant examples.

definition list element

Contains a list of terms and their definitions.

unordered list element

Contains a list of items whose order is not important.

ordered list element

Contains a list of items whose order is important.

Concept Topic Example: Toaster Classic

The Toaster Classic is the one-of-a-kind, classic machine that's designed to ensure that each slice toasts evenly to your preferred level.

Have perfect slices of delight with your favorite jam or butter. Or if you want to toast something that's frozen, the Defrost feature thaws it before the toaster starts toasting. The Toaster Classic even includes preset settings for:

- Toast
- Waffles
- Bagels
- English Muffin

Task Topic

Use task topics to provide step-by-step procedures for completing a task.

Task topics answer the following question: "How do I do *x*?"

Title Guidelines

- Use title case
- Begin with an infinitive

Examples: "Create an Assignment", "Reassign a Project"

Style Guidelines

- Communicate each step clearly.
- Only include information relevant to completing a step.
- State what the expected result is in the step result element for all steps. The last step in a task topic does not require a step result element because the task result is included in the result element.
    - Use the conditional processing attribute "userTesting" for every step result element and result element that you apply.
    - Remove the conditional processing attribute only when the task or step result is non-obvious. For example, the task result for branching operations or moving a file to the Trash are non-obvious.
- Only include images or screenshots for steps where the interface changes as a result of the step, and it's relevant to illustrate this change to our users. Otherwise, avoid using images and screenshots due to high maintenance costs.
    - To provide a screenshot of the interface during the step, use the information element.
    - To provide a screenshot of the interface with completed fields or selections, use the step example element.
    - To provide a screenshot of the interface after completing the step, use the step result element.

Considerations

Ask yourself the following questions when creating task topics:

- What information is required to complete each step in the task?
- What background information does a user need to get started on the task?
- Would images or examples help clarify how to complete the procedure?
- What is the expected result of completing the procedure?

Structure

task topics can contain the following elements:

- task element
    1. title element
    2. abstract element
        - short description element
    3. prolog element
    4. task body element
        a. prerequisite element
        b. context element
        c. steps element
            - step element
                i. command element
                ii. information element
                iii. step example element
                iv. choices element
                v. substeps element
                vi. step result element
        d. result element
        e. postrequisite element

Elements

title element

Entitles a topic, a section, or a container element.

abstract element

Provides introductory content that would be unfit for a short description element.

short description element

Illustrates the topic purpose in two or three sentences (no more than 50 words). short description elements can provide content for link previews and search engines.

prolog element

Contains topic metadata. Can contain multiple resource ID elements that you can use to implement contextual helps into applications.

task body element

The main element of a task topic or a process topic.

prerequisite element

Describes the requirements that must be met before beginning a procedure.

context element

Provides brief background information for a procedure.

steps element

Contains step elements needed to complete a procedure.

step element

Contains a command element that describes an action and other elements that help you perform the action. To ensure optimal readability of your procedures, limit the number of step elements to 10 per one task topic.

command element

Describes an action that must be completed. We recommend using a single, unambiguous, and imperative sentence in each command element.

information element

Clarifies a step or provides additional information about a step. You can use the information element to add a figure

element, code block element, or note element.

step example element

Provides an example on how to perform a step.

substeps element

Contains substep elements required to complete a step.

choices element

Provides different options to complete a given step. Each choice element encapsulates a single option.

step result element

Describes the expected step outcome.

result element

Describes the expected task topic outcome.

postrequisite element

Describes requirements that a user must meet after completing the procedure.

Task Topic Example: Toasting Bread

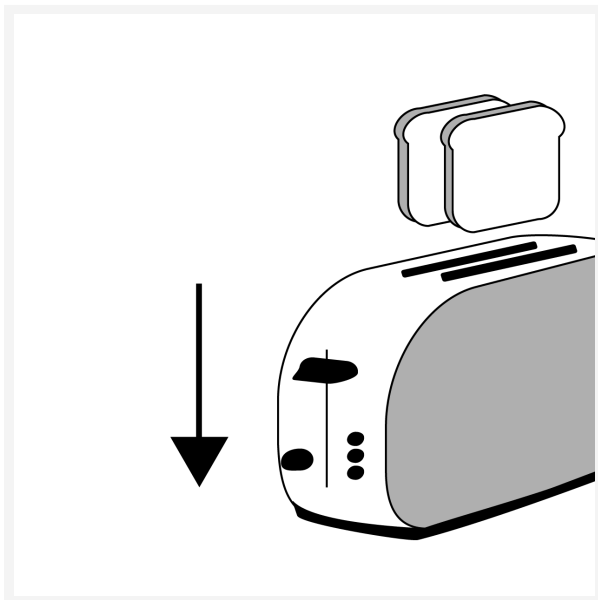Use the dials and knobs to toast your bread to your preferred level.

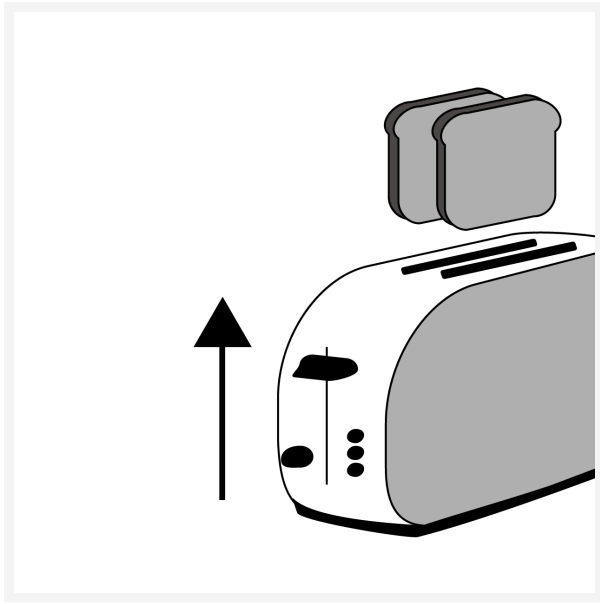## Before you begin

Obtain the following items:

- A plugged in Toaster Classic
- Two slices of bread
- A plate
- Butter or Jam (optional)

Procedure

1. Turn the knob to the desired browning level.

2. Press down the lever to start toasting.

   Result: Your bread slices and lever automatically lift up once the toasting cycle is complete.

Result

Your toast is ready for you to spread your favorite jam, butter, or whatever you like on it!

Process Topic

Use process topics to organize multiple task topics into a cohesive set of instructions.

Process topics answer the following question: "How do I do *x*?"

Title Guidelines

- Use title case
- Begin with an infinitive

Examples: "Set Up a Virtual Machine", "Develop a Course"

Style Guidelines

- Depending on the information architecture of your documentation, you may use the following approaches to develop process topics:
    - Collection-type approach. See Collection-Type Approach to Processes on page 26.
    - Cross-reference approach. See Cross-References in Processes on page 27.
- Differentiate process topics from task topics by assigning a dedicated attribute to their task element root tags.

> ⓘ **TIP:** For example, you can set the `process` value to the outputclass attribute. Then, you can set the publishing plugin to render the process topics in a desired way

Structure

process topics can contain the following elements:

- task element
    1. title element
    2. abstract element
        - short description element
    3. prolog element
    4. task body element
        a. prerequisite element
        b. context element
        c. steps element
            - step element
                i. command element
                ii. information element

       iii. step example element

       iv. choices element

       v. substeps element

       vi. step result element

    d. result element

    e. postrequisite element

Elements

title element

Entitles a topic, a section, or a container element.

abstract element

Provides introductory content that would be unfit for a short description element.

short description element

Illustrates the topic purpose in two or three sentences (no more than 50 words). short description elements can provide content for link previews and search engines.

prolog element

Contains topic metadata. Can contain multiple resource ID elements that you can use to implement contextual helps into applications.

task body element

The main element of a task topic or a process topic.

prerequisite element

Describes the requirements that must be met before beginning a procedure.

context element

Provides brief background information for a procedure.

steps element

Contains step elements needed to complete a procedure.

step element

Contains a command element that describes an action and other elements that help you perform the action. To ensure optimal readability of your procedures, limit the number of step elements to 10 per one task topic.

command element

Describes an action that must be completed. We recommend using a single, unambiguous, and imperative sentence in each command element.

information element

Clarifies a step or provides additional information about a step. You can use the information element to add a figure element, code block element, or note element.

step example element

Provides an example on how to perform a step.

substeps element

Contains substep elements required to complete a step.

choices element

Provides different options to complete a given step. Each choice element encapsulates a single option.

step result element

Describes the expected step outcome.

result element

Describes the expected task topic outcome.

postrequisite element

Describes requirements that a user must meet after completing the procedure.

Collection-Type Approach to Processes

Processes created by adjusting the collection-type attribute are easy to create and maintain.

> ☆ **IMPORTANT:** Linking created by adjusting the collection-type attribute is compatible only with the DITA Open Toolkit publishing scenarios with the args.rellinks parameter set to all.

Considerations

Keep in mind the following advantages of process topics using the collection-type attribute:

- Easy to create and maintain. If the number or order of procedures appended to a process topic change, the process topic list updates automatically.
- Reflects the topics' hierarchy in a map.

Keep in mind the following limitations of process topics using the collection-type attribute value:

- You cannot manually add additional information to the process topic steps, as they are generated automatically.
- You cannot append the same procedure under multiple process topics in the same map.
- Your DITA Open Toolkit publishing scenarios may require configuration to render the process topic steps as expected.

Example

Rendered Process

Process in the Table of Contents

Cross-References in Processes

Processes that use cross-references are flexible but may become hard to maintain.

Considerations

Keep in mind the following advantages of process topics that use cross-references:

- You can add additional information to the process list. For example, you can add note elements or mark process step elements as optional.

- You can link to the same task topic (procedure) from multiple process topics in a map.

- Compatible with a variety of publishing engines and publishing scenarios.

Keep in mind the following limitations of process topics that use cross-references:

- The process topic steps are not generated automatically so you need to add them manually.

- Cross-referenced content can be hard to maintain. The process topic steps do not update automatically when the number or order of task topics (procedures) changes.

- Topic hierarchy may not always be reflected, as the task topics (procedures) do not need to be appended under the process topic.

Example

Process Example

Reference Topic

Use reference topics to provide specification information that supports conceptual information and task completion.

Reference topics provide detailed, quickly accessed data, most often in tables.

Title Guidelines

- Use title case
- Use nouns or noun phrases
- Include the referenced subject, interface, etc.

Examples: "DITA Open Toolkit Errors", "Element Properties"

Style Guidelines

- Organize content into scannable structures (tables, bullet lists, code blocks, etc.)
- Provide only reference information, not conceptual or procedural information

Considerations

Ask yourself the following questions when creating reference topics:

- Is there any reference information to support concept or task topics?
- How can I organize this reference content so it can be easily scanned?

Structure

reference topics can contain the following elements:

- reference element
  1. title element
  2. abstract element
     - short description element
  3. prolog element
  4. reference body element
     - section element
     - table element
     - example element

Elements

title element

Entitles a topic, a section, or a container element.

abstract element

Provides introductory content that would be unfit for a short description element.

short description element

Illustrates the topic purpose in two or three sentences (no more than 50 words). short description elements can provide content for link previews and search engines.

prolog element

Contains topic metadata. Can contain

multiple resource ID elements that you can use to implement contextual helps into applications.

reference body element

The main element of a reference topic.

section element

Contains relevant content.

table element

Contains scannable reference material.

example element

Illustrates or supports a topic by providing relevant examples.

Reference Topic Example: Toaster Classic Specifications

The Toaster Classic includes many capabilities. The specifications of these capabilities are listed.

Specification	Description
Capacity (slots)	2
Output per hour	30
Loading (kW)	2.2
Dimensions (cm)	36 x 21 x 22
Weight (kg)	4.25
Slot width (mm)	28
Materials	Stainless steel ends, brass body

FAQ Topic

FAQ topics provide answers to frequently asked questions in a clear and concise question-answer format.

Title Guidelines

- Use sentence case
- Write in question format, including ending punctuation

Examples: "What is the maximum upload size?", "How do I change the owner of multiple files?"

Style Guidelines

- Use short, direct titles
- Be concise with your answers

Considerations

Ask yourself the following questions when creating FAQ topics:

- Is this a common question or problem?
- Is a short answer sufficient for this question or would it be better answered in the user guide or a tutorial?

Structure

FAQ topics can contain the following elements:

- FAQ element
    1. question element
    2. prolog element
    3. answer element

Elements

question element
Contains a frequently asked question.

prolog element
Contains topic metadata. Can contain multiple resource ID elements that you can use to implement contextual helps into applications.

answer element
Provides an answer to the question.

Troubleshooting Topic

Troubleshooting topics describe problems and provide possible remedies.

Title Guidelines

Keep these guidelines in mind when naming troubleshooting topics:

- Use title case
- Use nouns or noun phrases
- Include the issue or error

Examples: "DITA Open Toolkit Error DX931118", "Flickering Screen"

Style Guidelines

Keep these guidelines in mind when creating troubleshooting topics:

- Document common or obscure problems and their remedies
- Include all possible remedies for the problem

Considerations

Ask yourself the following question when creating troubleshooting topics:

- Is this problem reproducible?
- Do I have all the required information for the situation?
- Is there any additional information required for the remedies?

Structure

troubleshooting topics can contain the following elements:

- troubleshooting element
    1. title element
    2. abstract element
        - short description element
    3. prolog element
    4. troublebody element
        a. condition element
        b. solution element
            i. cause element
            ii. remedy element

Elements

title element

Entitles a topic, a section, or a container element.

abstract element

Provides introductory content that would be unfit for a short description element.

short description element

Illustrates the topic purpose in two or three sentences (no more than 50 words). short description elements can provide content for link previews and search engines.

prolog element

Contains topic metadata. Can contain multiple resource ID elements that you can use to implement contextual helps into applications.

troublebody element

The main element of a troubleshooting topic.

condition element

Illustrates the state that the troubleshooting topic should fix.

solution element

Contains a cause element and a remedy element.

cause element

Describes the source problem.

remedy element

Contains steps that provide a potential solution for the problem.

Troubleshooting Topic Example: DITA-OT Error DOTX017E

An `<xref>` error using the DITA Open Toolkit.

Condition

Publishing using the DITA Open Toolkit.

Cause

Found a link or cross reference with an empty `<href>` attribute (href="").

Remedy

1. Navigate to the file listed with the error.

2. Remove the empty `<href>` attribute or provide a value.

Glossary Entry Topics

Use glossary entry topics define a term and its terminology information.
You can use glossary entry topics to build glossaries to ensure a consistent vocabulary.

Title Guidelines

- Use lowercase
- Use the word you're defining

Examples: "branch", "release"

Guidelines

- Define one term per glossary entry topic
- Define terms concisely

Considerations

Ask yourself the following questions when creating glossary entry topics:

- Is this a common term?
- Is there relevant terminology information an end user needs?

Structure

glossary entry topics can contain the following elements:

- glossary entry element
    1. glossary term element
    2. glossary definition element
    3. prolog element
    4. glossary body element
        a. glossary part of speech element
        b. glossary status element
        c. glossary property element
        d. glossary surface form element
        e. glossary usage element
        f. glossary scope note element
        g. glossary symbol element
        h. glossary alt element

        ▪ glossary acronym element

    i. glossary alt element

        ▪ glossary synonym element

    j. glossary alt element

        ▪ glossary short form element

    k. glossary alt element

        i.   glossary abbreviation element

        ii. glossary status element

        iii. glossary property element

        iv. glossary usage element

        v. glossary alternate for element

Elements

glossary term element

Provides a term you want to define.

glossary definition element

Provides a definition for a term.

prolog element

Contains topic metadata. Can contain multiple resource ID elements that you can use to implement contextual helps into applications.

glossary body element

The main element of a glossary entry topic.

glossary part of speech element

Specifies the part of speech for the preferred and alternate glossary terms. By default, all glossary terms should be nouns.

glossary status element

Identifies the usage status of the preferred and alternate glossary terms. You control the status by assigning an appropriate

value to the value attribute of the glossary status element.

glossary property element

Specifies additional characteristics of the preferred and alternate glossary terms such as the gender of a noun.

glossary surface form element

Combines multiple glossary term forms. It is useful if you want to introduce a term for the first time in a given topic. For example, you may want to introduce a full tern with a corresponding acronym like "Virtual Machine (VM)" and then use the acronym "VM" only.

glossary usage element

Provides information about when to use the glossary term.

glossary scope note element

Specifies the contexts in which the glossary terms are applicable. For example,

a glossary scope note element may specify that given glossary terms are applicable for MacOS devices.

glossary symbol element

Specifies an image associated with a glossary term.

glossary alt element

Contains alternatives for a glossary term.

glossary acronym element

Provides an acronym for a glossary term.

glossary synonym element

Provides a synonym for a glossary term.

glossary short form element

Provides a shortened version of a glossary term.

glossary abbreviation element

Provides an abbreviation of a glossary term.

glossary alternate for element

Indicates a variant term relationship to another variant term in addition to the preferred term.

Glossary Entry Example: UI Control Element

The UI control element indicates clickable interface elements like buttons, fields, or menu items.

DITA Elements

We use a limited number of the available DITA elements.

Block Elements

Each block element starts with a new line.

Abstract and Short Description Elements

A short description element provides a short summary about a topic. An abstract element extends the short description.

Consistent use of short descriptions and abstracts enables you to quickly navigate topics.

DITA Guidelines

- Do not make short descriptions longer than two sentences
- If you need a more robust introduction to a topic that does not fit into the short description element, combine an abstract element with an short description element
- An abstract element can contain block elements that are unavailable in short description elements

Style Guidelines

- Do not begin short descriptions with "Learn how...", "This topic...", etc.
- Do not restate the title
- Be descriptive

Short Description Example

The Gray and Color templates are complete and modular styles that you can leverage to quickly create your own PDF Generator templates.

Abstract and Short Description Example

You can customize the PDF Generator templates by embedding or importing the CSS code into the code editor.

> ⊘ **TIP:** We recommend extending the Gray and Color templates with custom CSS rather than creating a template from scratch.

Choice Table Element

Use the choice table element to provide different options for performing a step. The option descriptions may contain multiple actions.

Guidelines

- Add a choice table element if there are multiple options to perform a step.

- Any option in the choice table element should require more than one action.

  If every option require only one action, use the choices element. See Choices Element on page 39.

- In the step element introduce the choice table element.

  For example, enter `Save the document by doing any of the following:`

- Start each option with "To" to ensure that the options are understood as not obligatory.

  For example, enter `To save the document by using the toolbar,`

- If the option description includes two or more actions, add an introductory text and put the actions in the ordered list element.

  For example, enter `do the following:`

- The choice table element headers ("Option" and "Description") are generated automatically in the output. Don't include the choice option heading element or the choice description heading element.

Example

The following example shows a choice table element that gives you three different options to open a map in the **source editor**.

Choice Table Example

1. Open a map in the source editor by doing any of the following:	
Option	**Description**
To open the source editor directly from the Content Library,	in the Content Library, right- click a map and select Open in Source Editor.
To open the source editor from the map editor,	do the following: a. In the Content Library, hover over a map and click the Dock button. b. In the map editor pane, right-click the map and select Edit Element XML .
To open the source editor from the topic editor,	do the following: a. In the Content Library, hover over a map and click the Open icon or double-click a map. b. In the left pane, right-click the map name and select Edit Element XML .

Choices Element

You use the choices element to provide different options to perform a step. The options cannot contain more than one action.

Guidelines

- Add a choices element if there are multiple options to perform a step.

- Each option in the choices element should require only one action.

  If any option require more than one action, use the choice table element. See Choice Table Element on page 38.

- In the step element introduce the choices element.

  For example, enter `Save the document by doing any of the following:`

- Start each option with "To" to ensure that the options are understood as not obligatory.

  For example, enter

  `To save the document by using your keyboard, press CMD→F.`

Example

The following example shows a choices element that gives you two different options to save your edits.

Choices Example

```
5. Save your edits by doing any of the following:
     ○ To save the file and preview the changes, click Save.
     ○ To save the file and close the Edit Source window, click Save & Close.
```

Code Block Element

The code block element contains long portions of code. This element is helpful for providing sample code in steps or as reference material.

Guidelines

- Use the code block element to contain two or more lines of code that must be set off from the main flow of text
- Limit the use of the code block element to within task topics and reference topics
- Indent and space code correctly

> ⊙ **TIP:** You can add snippets of code within the main flow of text using the code phrase element. For more information, see Code Phrase Element on page 61.

Example

```
<!DOCTYPE html> <html> <body> <h1>My First Heading</h1> <p>My first
paragraph.</p> </body>
```

Definition List Element

A definition list element contains a list of terms with corresponding definitions.

Guidelines

- Use definition lists to explain terms that may not be known to your end users
- Insert an introductory paragraph before each definition list
- Use parallel constructions in each definition description

Example

The SSG interface enables you to upload configuration files that determine the look and feel of the published websites.

Scenario name

Determine the name of the SSG publishing scenario that appears in the publishing interface.

Scenario Logo

Determine the site logo.

Custom Theme

Determine the style of the entire site, including the leading colors and global typography settings.

Custom Global CSS

Determine the styling of rendered DITA elements in the central content area.

Custom Zip File

Determine the style and contents of components like footer or home page.

Figure and Image Elements

Use the figure element to insert images and associated elements into your topics. Images add value to your content or convey complex ideas.

Figure Element

The figure element can nest the following elements:

- title element
- image element
    - alt element

Guidelines

- If you use figures in a task topic to explain a step, nest it inside another element first.

  For example, insert a figure in the information element, step result element, or step example element.

- Nest the title element inside the figure element only if it adds value.

  For example, there is no need to add titles if you use figure to explain a step.

- Nest the alt element inside every image element to make your content more accessible.

  For example, the image description that you provide in the alt element can be processed by screen readers.

Example

Topic Editor
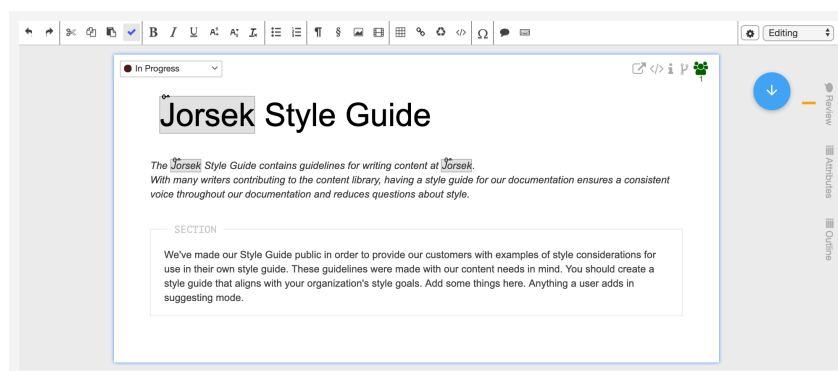

```
<fig>
<title>Topic Editor</title>
<image width="420"
href="../../_Media/information_type_concept_ex.png" id="image_lhq_pdt_53b">
<alt> A sample content topic opened in the <ph conkeyref="vars/TE"/>.</alt>
</image>
</fig>
```

Title Element Guidelines

- Insert title element only if it adds value to your content
- Be descriptive but keep the title short and simple
- Use title case

Image Element Guidelines

- Assume a file naming convention. For example, `<category>_<object_illustrated>.<extension>` (`pdf_generator_editor_left_pane.png`).
- For static images, use the PNG format so that the images can be used in different screen sizes and for a variety of publishing outputs like HTML, PDF, etc.
- For animated images, use the GIF format. For more information, see Animated Images on page 44.
- If you want to annotate an image, set the annotation color to `#FF0000` (red).
- If you want to insert an inline icon, do the following:
  - If possible, resize the icons to 16x16 px
  - Use the image element alone (do not encapsulate it in the figure element)
  - Set the image element placement attribute to the inline value

Inline Icon Example.
Inline icons can be easier to understand than icon names. In the following example, the "**Configure Database Sources** icon" text was replaced with an actual icon "⚙☰".

2. In the Data Source Explorer pane, click ⚙☰ .

Alt Element Guidelines

The alt element provides alternative text for an image. This element is critical to make your content accessible. For example, the alt element content can be processed by screen readers.

- Do not use directional language, such as "above", "next to", "beside", etc.
- You may need to provide some contextual information in the image description
- Describe the relevant parts of the image in the same way you would verbally describe the image to someone
- In some cases, along with the image description, explain why an image was included

Animated Images

You insert GIFs to your topics by using image elements.

Guidelines

- Treat GIFs as additional and optional aids to your textual content. Your documentation should be understandable without any graphical aids.

- Include GIFs only into online deliverables. If you plan to publish a given asset both as an online and printed deliverable, include a static image equivalent into printed deliverables.

> ⓘ **TIP:** Assign the `@audience="print"` attribute to the static image and the `@audience="online"` attribute to the animated image. Use an appropriate DITAVAL when publishing your asset.

Static and Animated Images Profiled for Different Deliverables
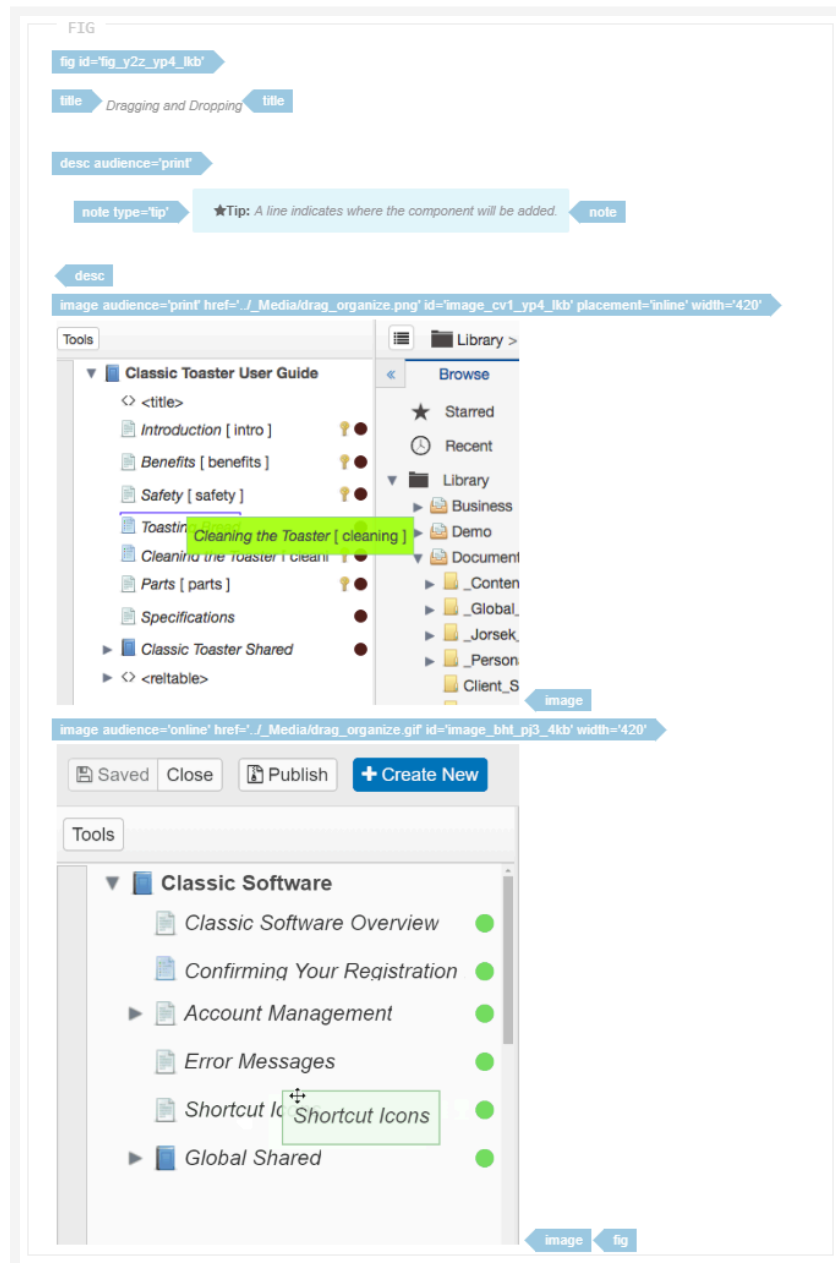
- Crop your GIFs so that they only show the relevant part of the interface that you capture.
- If you capture a web-application, use the zoom feature built-in your web-browser as much as possible.

Requirements

You can use the software of your choice to capture GIFs. However, ensure that your captures meet the following requirements.
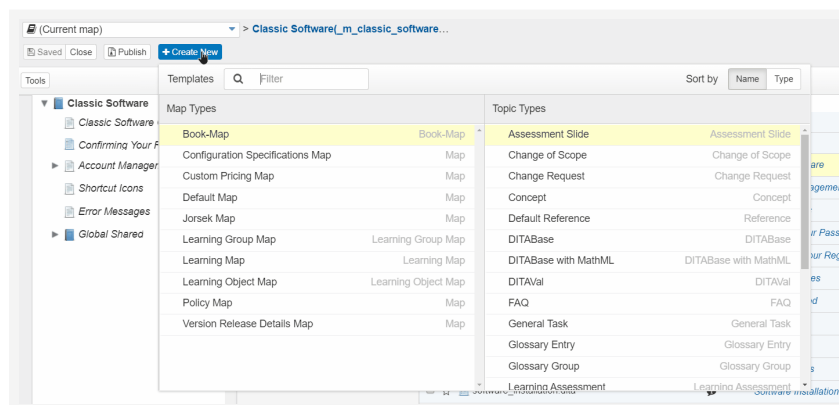
GIF Requirements

Category	Requirement
Frame rate	30 Frames Per Second (FPS)
Transition type	Fade
Transition length	10 frames
Effects (for example, highlighted mouse clicks)	No

Example

Animated images are useful if you want to demonstrate interface operations like dragging and dropping.

Dragging and Dropping

Footnote Element

Use the footnote element to refer to additional information from the main body of the document.

Guidelines

- Depending on the output, footnotes can render differently. Once you publish your content, always check if the footnotes are rendered as expected.

- Use numerical footnote annotations with textual content. They update automatically as you add new footnotes based on the their position in the topic.

- Use symbolic footnote annotations with numerical content. You need to manually update symbolic annotations as you add new footnotes.

> ⓘ **REMEMBER:** Unlike symbolic annotations, numerical annotations can be understood as a `<numerical_data>` to the power of `<your_numerical_annotation>`.
>
> - Correctly annotated numerical content: 0.181193^{*}
>
> - Incorrectly annotated numerical content: 0.181193^{2}

Examples

The following examples show footnote elements in different outputs.

Numerical Footnotes in the PDF Output

Footnotes Example

□ JORSEK

Footnotes Example

Paragraph A[1]

Paragraph B[2]

Paragraph C[3]

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur urna ante, cursus sit amet commodo quis, feugiat elementum lorem. Quisque tempor, nulla dapibus venenatis aliquet, leo ligula tincidunt nisl, vitae auctor eros leo ut magna. Nam eleifend suscipit sem non lacinia. Vivamus eget vehicula turpis. Aliquam erat volutpat. Aliquam venenatis mollis nulla, quis faucibus nisi ornare quis. Praesent eget ligula cursus, vestibulum dui sed, congue dolor. Nullam erat tortor, lacinia sed nunc euismod, lacinia elementum massa. Donec consectetur sollicitudin pellentesque. Nullam quis dolor et leo ullamcorper rhoncus faucibus vel lacus. Donec id velit elit. Curabitur at tincidunt augue. Nullam at vehicula est, faucibus tincidunt felis. Nam non enim neque. Fusce porttitor aliquet dolor sed facilisis. Nunc vitae justo ac dui sagittis elementum.

[1] Paragraph A footnote content
[2] Paragraph B footnote content
[3] Paragraph C footnote content

1

Symbolic Footnotes in the HTML5 Output

Footnotes Example

Paragraph 1[*]

Paragraph 2[**]

Paragraph 3[***]

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur urna ante, cursus sit amet commodo quis, feugiat elementum lorem. Quisque tempor, nulla dapibus venenatis aliquet, leo ligula tincidunt nisl, vitae auctor eros leo ut magna. Nam eleifend suscipit sem non lacinia. Vivamus eget vehicula turpis. Aliquam erat volutpat. Aliquam venenatis mollis nulla, quis faucibus nisi ornare quis. Praesent eget ligula cursus, vestibulum dui sed, congue dolor. Nullam erat tortor, lacinia sed nunc euismod, lacinia elementum massa. Donec consectetur sollicitudin pellentesque. Nullam quis dolor et leo ullamcorper rhoncus faucibus vel lacus. Donec id velit elit. Curabitur at tincidunt augue. Nullam at vehicula est, faucibus tincidunt felis. Nam non enim neque. Fusce porttitor aliquet dolor sed facilisis. Nunc vitae justo ac dui sagittis elementum.

[*] Paragraph 1 footnote content
[**] Paragraph 2 footnote content
[***] Paragraph 3 footnote content

Note Element

Notes draw an end user's attention to important information.

Guidelines

- Use the note element to contain information that is more important than what occurs in the body text
- Add a note element near, typically directly in or below, the element it relates to

Note Types

Default Note

Provides general additional information that needs to be differentiated from the body text.

Tip

Provides helpful information that can make a process easier.

Important

Provides information that is more important to the user than a general note.

Remember

Reminds users to not forget important information.

Caution

Warns users when there is potential for a negative outcome.

Examples

> ⓘ NOTE: Available options may vary based on your configuration.

> ⓘ TIP: To select a range of check boxes, check the first box in the range and then while holding SHIFT, select the last box in the range.

Paragraph Element

A paragraph element is a block of text that contains a single idea.

Guidelines

- Limit paragraphs to one main idea
- Aim for paragraphs to be between two to five sentences
- Vary sentence lengths
- Avoid transitional words like therefore, first, etc. to improve reusability
- Break up long blocks of text with figures, lists, or other visual elements to help readers easily scan material

Examples

Metadata are tags you apply to your files to classify them. These custom metadata tags must be manually created and assigned to your files, but once they've been created and assigned, you can use them in powerful ways to find and analyze content in your repository.

There are two types of metadata: Taxonomy Metadata and Label Metadata. You use both types of metadata the same way, though they differ in the ability to create new metadata tags on the fly.

Related Links Element

The related-links element enables you to quickly add links to related information at the end of a topic.

Guidelines

Keep these guidelines in mind when using the related-links element:

- If you link to topics in the **content library**, links added in the related-links element resolve to topic titles.

- If you publish to a print-friendly output (for example, PDF), ensure that the topic that you link to is included in the same map as the topic that you link from.

- If you link to external resources, for example websites, you must add a link text manually. Remember to make it generic to avoid maintenance issues.

- The related links are usually displayed at the end of the topic. Some web-like outputs (for example, webhelps) may display them in a separate navigation pane.

Structure

- topic body
    - topic elements
    - related-links element
        - link element (any number)
            - linktext element
            - desc element
        - linklist element (any number)
            - title element
            - desc element
            - linklist element (any number)
            - link element (any number)
                - linktext element
                - desc element
        - linkpool element (any number)
            - title element
            - desc element
            - linkpool element (any number)
            - link element (any number)

- linktext element
- desc element

Elements

related-links element

A container for links to the resources related to the topic.

link element

A link to a:

- DITA or non-DITA file in your **content library**
- A website or file hosted on an external server

linktext element

Enables you to provide a custom link text.

desc element

Describes an element.

linklist element

A container that enables you to arrange a group of link elements.

linkpool element

A container that contains a group of link elements with common characteristics like audience or source.

title element

The title of a given element.

Examples

Related Links.

The following example from the Jorsek Portal shows that:

- "Reference A" links to a DITA topic in the **content library**.

- "Sample_Word_Document.docx" links to a Microsoft Word document in the **content library**. When you click it, the file downloads.

- "DITA Open Toolkit Homepage" links to https://dita-ot.org/. The link has custom link text, and a description that is visible as a tooltip when you hover over the link.

Resource ID Element

The resource ID element enables you to assign unique IDs to topics so you can use them in contextual helps.

> ⊙ TIP: Contextual helps are created by mapping the documentation IDs into applications.

Guidelines

- Each resource ID element must be unique in a map
- Assign meaningful IDs to make it easier to map them into applications
- If you want to map a topic into multiple places in the application or multiple applications, you can add several unique resource ID elements to a topic

Section Element

The section element is an organizational subset of information that is related directly to a single topic.

Guidelines

- Limit section use to cases where the topic has multiple components that require explanation
- If there are multiple sections in a topic, add a title element to each section to make scanning the information easier
- Section titles should be title case

Section Divider Element

You can use sectiondiv elements to group content within a section element.

Guidelines

- A sectiondiv element cannot contain a title element. If you want to introduce a hierarchy below the section element level, consider splitting content into multiple topics instead.

- You can nest a sectiondiv element within a sectiondiv element. This is useful if you want to develop structured information within section elements.

- You can use sectiondiv elements to reuse a group of elements from a section element.

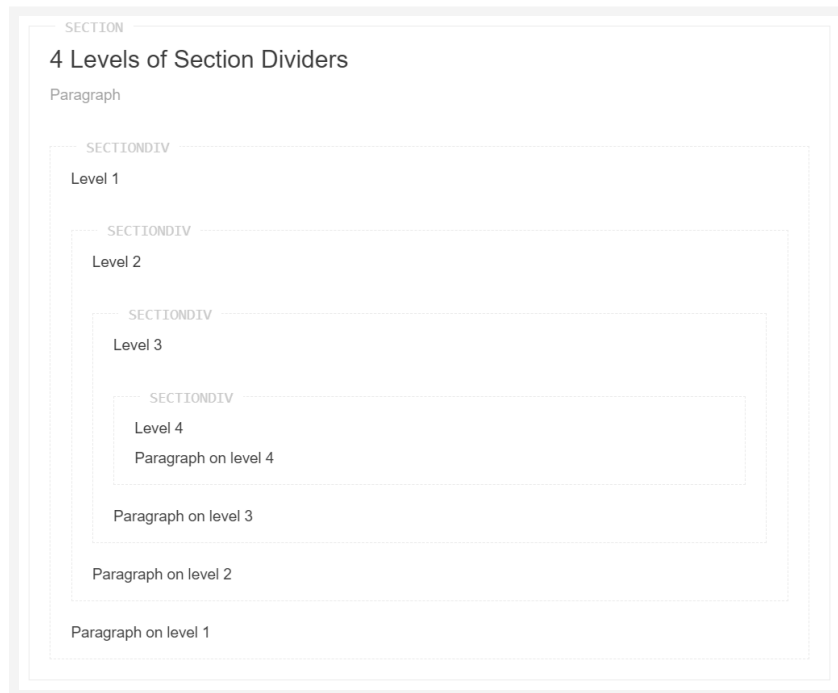- Most publishing scenarios do not style the sectiondiv element in the output.

  > ⓘ TIP: You can easily style sectiondiv elements by configuring a PDF Generator scenario.

Example

The following example shows a section element that contains four levels of sectiondiv elements:

Nested Section Dividers

```
SECTION

4 Levels of Section Dividers
Paragraph

    SECTIONDIV

    Level 1

        SECTIONDIV

        Level 2

            SECTIONDIV

            Level 3

                SECTIONDIV

                Level 4
                Paragraph on level 4

            Paragraph on level 3

        Paragraph on level 2

    Paragraph on level 1
```

```
<section>
    <title>4 Levels of Section Dividers</title>
    <sectiondiv>
        Level 1
        <sectiondiv>
            Level 2
            <sectiondiv>
                Level 3
                <sectiondiv>
                    Level 4
                    <p>Paragraph on level 4</p>
                </sectiondiv>
                <p>Paragraph on level 3</p>
            </sectiondiv>
            <p>Paragraph on level 2</p>
        </sectiondiv>
        <p>Paragraph on level 1</p>
    </sectiondiv>
</section>
```

Nested Sections Dividers with no Styling Applied.

The following example shows that nested sectiondiv elements are not styled in any way in the PDF output by default.

4 Levels of Section Dividers

Level 1
Level 2
Level 3
Level 4

Paragraph on level 4
Paragraph on level 3
Paragraph on level 2
Paragraph on level 1

Nested Section Dividers with Styling Applied.

The following example shows styled sectiondiv elements published by using a custom PDF Generator scenario.

4 Levels of Section Dividers

Level 1
　　Level 2
　　　　Level 3
　　　　　　Level 4
　　　　　　Paragraph on level 4
　　　　Paragraph on level 3
　　Paragraph on level 2
Paragraph on level 1

Table Element

Organize content into tables to create easily scannable content for end users.

Guidelines

- Use the table element so you can create complex tables with joined cells. Do not use the simple table element.

- Use column headings.

- Use title case for all column headings.

- Use consistent grammatical structure for cell content.

- Provide a description of the table's content directly before the table element itself.

Publishing Considerations

Our publishing plugins handle table styling, so you don't need to worry about:

- Row heights

- Column widths

- Border styling

Example

Option	Description
Insert	Add an element.
Remove	Remove an element.
Append	Add an element as a child of the selected element.

Unordered and Ordered List Elements

Lists enable you to present information in an engaging and easily scannable way.

Guidelines

- Use parallel structure when writing a bullet list. For example, start each bullet point with a verb.

- Capitalize the first letter of each list item.

- Include all common information from the list items in the introductory sentence to avoid repetition.

- Do not use an ordered list to document procedures within a concept or reference topic. Instead, use a task topic.

- Do not add a period to the end of any list item, even if the list item is a complete sentence.

  The only exception to this rule is when a list item uses more than one sentence. In this situation, you must add an end punctuation to all sentences in the list item and all items in the bullet list must also include end punctuation.

Unordered List Element

When you need to list three or more items in no particular order, use the unordered list element so the reader can quickly scan the bullet list items. Using an unordered list element prevents long lists of items from being embedded in a sentence, which is more difficult to read.

Ordered List Element

The ordered list element provides some structure so that the list items are numbered. Only use the ordered list element in a concept topic to explain a process.

> ⓘ NOTE: If there is a procedure to follow, it's typically best to use a task topic instead.

Examples

Modify the following properties:

- Owner

- Status

- Date

- Metadata

Creating and using Taxonomy Metadata requires you to:

1. Create a taxonomy of terms

2. Assign taxonomy to a metadata category

3. Assign metadata to files

Inline Elements

Inline elements do not start with separate lines.

Code Phrase Element

The code phrase element contains a snippet of code that should be used within the main text.

Guidelines

- Include snippets of code that are shorter than one line
- Include any markup that's necessary, such as `<codeph>`
- Do not use the code phrase element if you are writing out, for example, "code phrase"

Example

The `<template-type="">` identifies the template type.

Cross-Reference Element

Use the cross-reference when you want to link to another topic, a file, or an external site.

Guidelines

- Do not create cross-references to maps. The publishing engines may render such links incorrectly.
- Do not place links in the middle of a sentence. Instead, place them at the end of the sentence as a supplemental reference.

File Path Element

The file path element contains the location of a file, a file name, or a directory.

Guidelines

- Include the file extension when referring to a file name

Examples

Move `example.png` to a new location.

You can access the files at `../Documentation/User_Guide`

Menu Cascade Element

The menu cascade element contains a series of selections related to interface components such as menus or buttons.

Guidelines

- Include series of two or more selections
- Enter the name of each clickable interface element in a separate UI control element

Example

In the **topic editor,** click **Link→Insert Xref**.

Object Element

The object element contains links to videos that are rendered as embedded videos in the output.

Guidelines

- Embed the URL to the video in the `@data` attribute of the object element
- Ensure to use the value `video` in the `@type` attribute of the object element

Phrase Element

You can use phrase elements to mark up parts of content (for example, a part of a paragraph) for conditional processing, reuse, or custom rendering in the output.

Guidelines

- You can reuse variables through phrase elements and conkeyref attributes.

- You can reuse a subset of content contained in an element by wrapping some content in a phrase element.

- You can profile your content by using phrase elements and conditional processing attributes.

- To achieve specific processing or formatting on a phrase level in the output, you can pass additional information to the publishing engine by using phrase elements and outputclass attributes.

> ⓘ NOTE: Depending on the publishing scenario that you use, you may be able to pass different outputclass attribute values to the publishing engine. The DITA outputclass attribute becomes a `class` attribute in most output formats. For more information, contact your easyDITA Administrator or Customer Success Manager.

Examples

Variable Example.

The following example shows a product name variable inserted in the short description element.
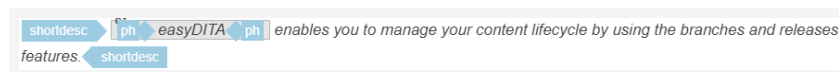
Reused Phrase Element.

The following example shows a reused phrase element in the result element.

> ⓘ NOTE: The result element is not directly reused on purpose. This strategy enables you to keep multiple results in a single result element in a warehouse topic.

Profiled Phrase Element.

The following example shows a paragraph that includes two phrase elements, one profiled for "Product A" and one profiled for "Product B". Each profile can be included or excluded from a publication.
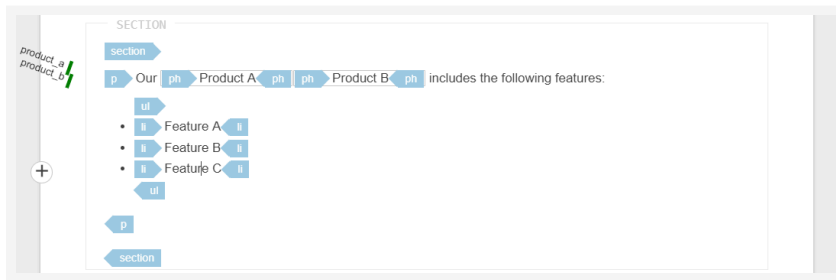
Shortcut Element

The shortcut element contains keyboard keys and keyboard shortcuts.

Guidelines

- Follow our guidelines for referring to keyboard keys. See Keyboard Key Names on page 9.
- Use the menu cascade element to contain keyboard shortcuts containing multiple keyboard keys.

Examples

Press CMD .

Press CMD → SHIFT → V .

System Output Element

A system output element indicates any output from the computer.

Guidelines

- Do not follow the system output element elements with any punctuation marks
- System output elements are useful in result elements and step result elements

Example

The concurrent jobs counter shows 99

Q Element

Use the quote element to ensure that consistent quotation marks are used throughout the documentation.

Guidelines

- Do not enter quotation marks manually
- Encapsulate the text that you want to quote in the quote element
- Ensure that the quote elements render properly in the output

Example

In the following example, the text that precedes the figure refers to the particular parts of the figure. The quoted words are encapsulated in the quote elements.

UI Control Element

The UI control element contains user interface components, such as buttons or menu options.

Guidelines

- Each UI control element should only contain one user interface component, including:
    - Buttons
    - Menu options
    - Text fields
    - Check boxes
- Use the menu cascade element when referring to a series of components in the user interface. For more information, see Menu Cascade Element on page 62.

Examples

Click OK.

Clear the Recreate check box.

User Input Element

The user input element contains text that an end user should input into a field, typically a text box.

Guidelines

- Limit the use of the user input element to task topics and reference topics
- Do not follow the user input element elements with any punctuation marks

Examples

In the Date field, enter `01-31-2018`

Window Title Element

The window title element contains names of windows, dialogs, or panes.

Guidelines

- Use the window title element when using the names of windows, dialogs, or panes.
- Include the interface component type along with the interface name. This means if you're referring to a window, it would be the name of the window plus the word "window".

Examples

In the **Properties** window, select...

In the **Rename** dialog, select...

Variable Name Element

A variable name element enables you to indicate a part of an user input element, file path element, or system output element that may vary.

> ⑦ **TIP:** If needed, you can also use variable name elements in code phrase elements and code block elements.

Guidelines

- Do not enter any brackets or quotation marks inside variable name elements, they are added automatically by our publishing engines
- Try to use concise names for variables
- Explain even the most obvious variables

Example

The following example shows a task step with the `<site_name>` variable.

1. Enter `gatsby new "<site_name>" https://gitlab.com/Jorsek/portal/gatsby-starter-easydita.git`

   Where `<site_name>` is the name of the directory in which you want to develop your site.

Maps

Maps can include topics and submaps.

Map Guidelines

- Use the Jorsek map template to create new maps. This ensures that our reusable content and variables are included in each deliverable
- Use general, though clear, titles for maps
- Consider creating submaps to organize content that relates to a single, main topic so that the submap can be reused in other maps

Map Structure

- Jorsek map
    - title element (required)
    - topicref element (optional, any number)
    - mapref element (optional, any number)
    - topichead element, topicmeta element, and navtitle element (optional, any number)
        - mapref element (optional, any number)
    - reltable element (optional)

Map Elements

topicref element
Includes a topic in a map.

mapref element
Includes a map in a map.

topichead element
topicmeta element
navtitle element
Specify navigational headings for maps.

reltable element
Includes a relationship table in a map.

## Map Attributes

`@processing-role="resource-only"`

Apply the `@resource-only` attribute value to a resource so it does not show in the final output, but is used to resolve references. This attribute is typically applied to warehouse topics or warehouse maps.

Maps without Topic Hierarchy

"Flat maps" contain topics that are organized without any hierarchy.

Map without Topic Hierarchy - Example


```
<map id="_m_faqs" title="FAQs">
        <topicref href="how_do_i_create_a_topic.dita" />
        <topicref href="how_do_i_create_a_map.dita" />
</map>
```
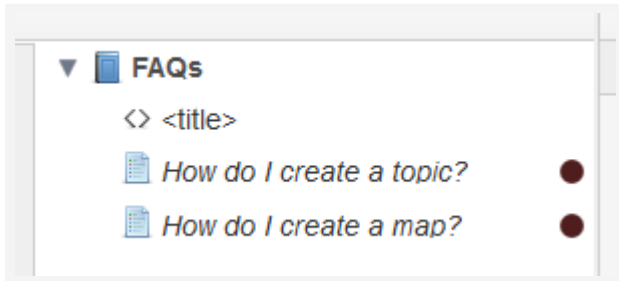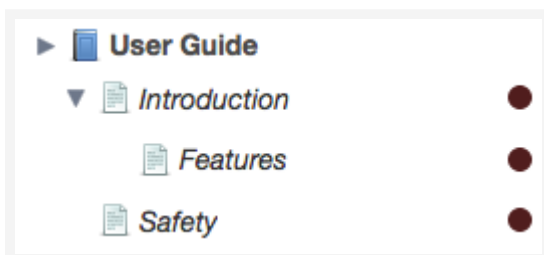
Maps with Topic Hierarchy

A map with a topic hierarchy contains topics with a parent or child relationship.

> ⓘ TIP: Topic hierarchies may be harder to reuse than map hierarchies. Map hierarchies can be easily added to new maps, while topic hierarchies have to be rebuilt each time.

Map with Topic Hierarchy - Example


```
<map id="_m_user_guide" title="User Guide">
        <topicref href="introduction.dita">
                <topicref href="features.dita"/>
        </topicref>
        <topicref href="safety.dita"/>
        ...
</map>
```

Maps with Map Hierarchy

A map with a map hierarchy is created by nesting maps within each other.

Map with Map Hierarchy - Example


```
<map id="_m_user_guide" title="User Guide">
        <mapref href="_m_topic_editor.ditamap" />
        <mapref href="_m_map_editor.ditamap" />
</map>
```

Considerations

Map titles do not display as headings. When published, there is no visual hierarchy created to distinguish between map levels so it looks flat. To resolve this, you can add navigational headings for each map title.

Jorsek Strategies

Establishing content development strategies promotes consistency and reduces questions about the development process.

> ⏀ **REMEMBER:** We develop the Jorsek Strategies with our content needs in mind. We recommend developing your own strategies that align with your organization goals.

Publishing

We use the PDF Generator and content portal publishing engines.

Guidelines

- Use the "jorsek_pdf" publishing scenario

  > ⓘ **NOTE:** Should there be anything wrong with the "jorsek_pdf" publishing scenario, fall back to the "DITA-OT 2.5.4 Jorsek PDF" scenario.

- Apply the appropriate customer-facing DITAVAL.

  > ⓘ **NOTE:** For general PDF publishes, use the "customer_facing_print" DITAVAL.

Release Details

We document changes for every major release of our product.

Guidelines

- Use the "Generate Release Notes from JIRA Issues" report to generate release notes from Jira.
- Group release notes together into topics based on the interface area they affect.
- In each topic, include "New" and "Fixed" sections.
- Organize release notes by using the unordered list element.
- Begin each release note with a past tense verb. For example, `Fixed locking issues that prevented locks from releasing.`
- Include the related Jira story information in each entry.
- Conditionalize the Jira story information with the `@audience="internal"` attribute.
- Include screenshots or GIFs to illustrate new features or enhancements.

File Statuses

We use the following file statuses.

New

A resource that was created recently.

Obsolete

No longer needed or relevant.

Needs Reevaluation

Out of date, incorrect, or incomplete.

In Progress

In development or planned to be in development.

In Review

Being reviewed.

Approved

Finalized, can be published.

Ready to Merge

Approved and needs to be merged with a production branch.

Writing for Reuse Conventions

We develop content in a way that enables us to maximize reuse.

Use	Avoid
"In the **following** example"	"In the example **below**"
"In the **previous** example"	"In the example **above**"
Nouns	Personal, demonstrative, and possessive pronouns