Table of contents

1 DITA 2.0 proposed content

1.1 Indexes

Processors can generate an index from `<indexterm>` elements.

1.1.1 Index elements

The content of the `<indexterm>` elements provides the text for the entries in the generated index. `<indexterm>` elements can be nested to create secondary and tertiary index entries.

The following elements can contain information that processors use to generate indexes.

<indexterm>
   (Unless combined with `<index-see>`) Instructs a processor to generate an index entry with a page reference. The `@start` and `@end` attributes on `<index-see>` can specifiy index ranges.

<index-see>
   Instructs a processor to generate a redirection to another index entry. The redirection typically has a page reference while the index entry does not.

<index-see-also>
   Instructs a processor to generate a redirection to another index entry. Both the index entry and the redirection typically have page references.

> **Comment by Kristen J Eberlein on 21 July 2019**
>
> Can we remove the word "typically" from the above statements?

How the index elements are combined, the location of `<indexterm>` elements, and the hierarchy of the DITA maps all effect how the index elements are processed and the resulting generated index entries.

1.1.2 Location of <indexterm> elements

`<indexterm>` elements can occur in topic bodies, topic prologs, and DITA maps.

The location of the `<indexterm>` element has an effect on where the `<indexterm>` points to and thus the page references that are rendered in the generated index.

Topic bodies
   An `<indexterm>` element that is located in a topic (and not the topic prolog) is a point reference to the location where the `<indexterm>` element occurs.

Topic prologs or DITA maps
   An `<indexterm>` element that is located in a topic prolog or as a child of a `<topicref>` element is a point reference to the title of the topic.

1.1.3 Index page references

Typically, an `<indexterm>` element instructs a processor to generate an index entry with a page reference.

The nesting of `<indexterm>` elements and the presence of redirection elements has an effect on page references in the generated index:

- An `<indexterm>` element that does not contain child `<indexterm>` elements (or an `<index-see>` element) contributes a page number to the generated index.

- An `<indexterm>` element that contains child `<indexterm>` elements contributes to the hierarchy of the multi-level index entry that is generated. Only the final nested `<indexterm>` element contributes a page number to the generated index.
- If an `<indexterm>` element also contains one or more `<index-see>` elements, only the `<index-see>` elements contribute page numbers to the generated index.
- If an `<indexterm>` element also contains one or more `<index-see-also>` elements, both the `<indexterm>` and `<index-see>` elements contribute page numbers to the generated index.

Comment by Kristen J Eberlein on 08 July 2019

The following RFC-2119 statement is new for DITA 2.0. Thoughts?

(For processors that support indexing) If multiple `<indexterm>` elements exist that contain identical content, a processor **SHOULD** generate only a single index entry, although all locations associated with the `<indexterm>` element contribute page references.

1.1.4 Index redirection

The `<index-see>` and `<index-see-also>` elements enable redirection to other index entries within the generated index.

The `<index-see>` element refers to an index entry that the reader should use *instead of* the current one, whereas the `<index-see-also>` element refers to an index entry that the reader should use *in addition to* the current one.

In order to prevent index entries that are both page references and redirections, it is an error if the following conditions occur:

- An `<indexterm>` element contains an `<index-see>` element, and the publication contains one or more `<indexterm>` elements with identical textual content.

  For example, topics referenced by the master map include the following markup:

  ```
  <!-- Topic A -->
  <indexterm>memory stick
    <index-see>USB drive</index-see>
  </indexterm>

  <!-- Topic B -->
  <indexterm>memory stick</indexterm>
  ```

- An `<indexterm>` element contains an `<index-see>` element and an `<index-see-also>` element.

  For example, a topic contains the following `<indexterm>` element:

  ```
  <indexterm>
    memory stick
      <index-see>USB drive</index-see>
      <index-see-also>flash stick</index-see-also>
  </indexterm>
  ```

Comment by Kristen J Eberlein on 07 July 2019

Do we want to state what processors should do for the first error condition listed above? Issue an error messsage?

Do we want to use RFC-2119 terminology?

A processor **MAY** give an error message when it encounters both `<index-see>` and `<index-see-also>` elements within the same `<indexterm>` element. Processors **MAY** recover from this error condition by treating the `<index-see>` element as an `<index-see-also>` element.

1.1.5 Index ranges

Authors can use the `@start` and `@end` attributes on `<indexterm>` elements to index extended discussions. Processors generate index entries that range over several pages.

The start of an index range is indicated by an `<indexterm>` with a `@start` attribute.

The end of a range is indicated by whichever of the following occurs first:

- An `<indexterm>` element with an `@end` attribute with a value that matches the `@start` attribute on the `<indexterm>` element that begins the range
- The applicable scope boundary

The applicable scope boundary depends on the location of the start element:

Topic body
    End of the topic body.

Topic prolog
    End of the topic cluster. This includes the topic that contains the start element and any child topics.

DITA map
    Either the end of the topic cluster (the topic that contains the `<indexterm>` element with the `@start` attribute and any child topics) or the end of the DITA map, whichever occurs first.

Processors that support index ranges **SHOULD** do the following:

- Match `@start` and `@end` attributes by a character-by-character comparison with all characters significant and no case folding occurring

  > **Comment by Kristen J Eberlein on 09 July 2019**
  >
  > What does "all character significant and no case folding occurring" mean?

- Ignore `@start` and `@end` attributes if they occur on an `<indexterm>` element that has child `<indexterm>` elements
- When index ranges with the same identifier overlap, the widest range applies, and end ranges are matched with start ranges by last-in-first-out. In other words, the ranges are interpreted as nested rather than overlapping with the highest-level container taking precedence over narrower contained ranges.

  > **Comment by Kristen J Eberlein on 09 July 2019**
  >
  > I'm not sure I understand this. Can we make it clearer?

- Handle an end-of-range `<indexterm>` that is nested within one or more `<indexterm>` elements. The end-of-range `<indexterm>` element should have no content of its own; if it contains content, that content is ignored.
- Ignore unmatched index range end indications.

1.1.6 Examples of indexing

This section contains examples and scenarios that illustrate the use and processing of indexing elements.

1.1.6.1 Example: Nested <indexterm> elements

This example contains a multilevel `<indexterm>` element.

```
<indexterm>cheese
  <indexterm>sheeps milk
    <indexterm>pecorino</indexterm>
  </indexterm>
  <indexterm>goats milk
    <indexterm>chevre</indexterm>
  </indexterm>
</indexterm>
```

The previous code is equivalent to the following:

```
<indexterm>cheese
  <indexterm>sheeps milk
    <indexterm>pecorino</indexterm>
  </indexterm>
</indexterm>
<indexterm>cheese
  <indexterm>goats milk
    <indexterm>chevre</indexterm>
  </indexterm>
</indexterm>
```

A processor will generate the following index entries:

- A primary entry for "cheese"
- Secondary entries for "goats milk" and "sheeps milk"
- Tertiary entries for "chevre" and "pecorino"

1.1.6.2 Example: Index ranges

There are three locations that can declare a index range: the body of a topic, the prolog of a topic, and a map.

Index range within a single topic

In the following example, the index range begins at the start of the second paragraph and continues to the beginning of the last paragraph. If the matching end range markup was not included, the index range would end at the end of the body element.

```
<topic id="accounting">
  <title>Accounting regulations</title>
  <body>
    <p>Be ethical in your accounting.</p>
    <p><indexterm start="acctrules">rules</indexterm>Remember to do all of the
following: ...</p>
    <!-- ...pages worth of rules... -->
    <p><indexterm end="acctrules"/>Failure to comply will get you audited.</p>
  </body>
  <!-- Potential sub-topics -->
</topic>
```

Index range that starts within a topic prolog

In the following example, the range begins with the start of the topic title; it includes the entire topic and any subtopics. The range ends within the same prolog, regardless of whether `<indexterm end="acct"/>` is specified in the prolog.

```
<topic id="accounting">
  <title>Accounting regulations</title>
```

```
   <prolog>
     <metadata>
       <keywords>
         <indexterm start="acct">accounting</indexterm>
       </keywords>
     </metadata>
   </prolog>
   <!-- ... -->
</topic>
```

Now assume that the topic in the previous sample is named `acct.dita`. Ranges defined in a prolog cover sub-topics, including those nested based on a map; in the following example, this means that the range covers all of `acct.dita`, as well as the child topics `procedures.dita` and `forms.dita`:

```
<topicref href="acct.dita">
  <topicref href="procedures.dita"/>
  <topicref href="forms.dita"/>
</topicref>
```

Index range specified within a map

In the final example, the index range is specified in a map.

```
<topicref href="acct.dita">
  <topicmeta>
    <keywords>
      <indexterm start="acct">accounting</indexterm>
      <indexterm end="acct"/>
      <indexterm start="govt">government forms</indexterm>
    </keywords>
  </topicmeta>
  <!-- Nested topicref elements -->
</topicref>
<topicref href="taxfiling.dita">
  <topicmeta>
    <keywords>
      <indexterm end="govt"/>
    </keywords>
  </topicmeta>
</topicref>
```

The index range for "accounting" begins with the start of the first topic title in `acct.dita`; it includes that topic as well as any subtopics.

The index range for "government forms" begins with the start of the first topic title in `acct.dita`, and continues until the end of the last element in the file `taxfiling.dita`. If the end range for "govt" was not specified, the range would continue to the end of the map.

1.2 Indexing group elements

DITA provides elements that enable processors to generate an index.

1.2.1 <index-see>

An `<index-see>` element directs the reader to an index entry that the reader should reference instead of the current one.

Usage information

There can be multiple `<index-see>` elements within an `<indexterm>` element.

> **Comment by Kristen J Eberlein on 07 July 2019**

Processing expectations

Processors should ignore an `<index-see>` element if its parent `<indexterm>` element contains any `<indexterm>` children.

Attributes

The following attributes are available on this element: and `@keyref`.

Examples

The following code sample shows how an `<index-see>` element is used to refer readers to the preferred term:

```
<indexterm>Carassius auratus
  <index-see>goldfish</index-see>
</indexterm>
```

This markup typically will generate an index entry without a page reference. It might look like the following:

carassius auratus, *see* goldfish

Figure 1: Use of an <index-see> element

The following code sample shows how an `<index-see>` is used to redirect to a more complex index entry:

```
<indexterm>feeding goldfish
   <index-see>goldfish<indexterm>feeding</indexterm></index-see>
</indexterm>
```

Figure 2: Use of an <index-see> element to redirect to a multi-level index entry

1.2.2 <index-see-also>

An `<index-see-also>` element directs the reader to an index entry that the reader should reference in addition to the current one.

Usage information

There can be multiple `<index-see-also>` elements within a single `<indexterm>` element.

Processing expectations

Processors should ignore an `<index-see-also>` element if its parent `<indexterm>` element contains any `<indexterm>` children.

Attributes

The following attributes are available on this element: and `@keyref`.

Examples

The following code sample shows the use of an `<index-see-also>` element to generate a "see also" reference to the index entry for "goldfish".

```
<indexterm>carp
   <index-see-also>goldfish</index-see-also>
</indexterm>
```

This markup generates a primary index entry for "carp" and secondary index entry that instructs the reader to "see also goldfish".

> **Comment by Kristen J Eberlein on 21 July 2019**
>
> Is it accurate to refer to the generated see-also entry as a secondary index entry?

Figure 3: Use of an <index-see-also> element

The following code sample shows the use of an `<index-see-also>` element to redirect to a multilevel `<indexterm>` element:

```
<indexterm>feeding
   <index-see-also>goldfish <indexterm>feeding</indexterm></index-see-also>
</indexterm>
```

This code sample assumes that the following index entry exists within the DITA source:

```
<indexterm>goldfish
  <indexterm>feeding</indexterm>
</indexterm>
```

This markup generates index entries such as the following:

feeding 348
see also goldfish, feeding
...

goldfish
feeding 370

Figure 4: Use of an <index-see-also> element to redirect to a multilevel index entry

1.2.3 <indexterm>

The content of an `<indexterm>` element is used to produce an index entry in a generated index. Nested `<indexterm>` elements create multi-level indexes.

Rendering expectations

The content of `@indexterm` entries is not rendered in the flow of body text; it is rendered only as part of a generated index.

Attributes

The following attributes are available on this element: , `@keyref`, and the attributes defined below.

@start
    Specifies the start of an index range

@end
    Specifies the end of an index range. The end of an index range is indicated by whichever comes first of the following:

- Applicable scope boundary
- An `<indexterm>` element with an `@end` attribute with a value that matches that `@start` attribute that began the index range

Examples

When index entries are placed in the body of a topic, they serve as point references to their location in the topic.

In the following code sample, the `<indexterm>` element provides a point reference to the beginning of the paragraph.

```
<p><indexterm>databases</indexterm>Databases are used to ...</p>
```

Figure 5: Index reference to a point within in a topic

When index entries are located within the `<prolog>` element in a topic or the `<topicmeta>` element in a DITA map, they serve as point references to the start of the topic title.

In the following code sample, the `<indexterm>` element provides a reference to the topic as a whole; the generated index entry is associated with the start of the `<title>` element.

```
<topic id="about-databases">
  <title>About databases</title>
  <prolog>
    <metadata>
      <keywords>
        <indexterm>databases</indexterm>
      </keywords>
    </metadata>
  </prolog>
  <body>
    <!-- content... -->
```

```
    </body>
</topic>
```

Figure 6: Index entries within topic prologues or DITA maps

The effect is the same as if the `<indexterm>` element had been located in the map:

```
<map>
  <topicref href="aboutdatabases.dita">
    <topicmeta>
      <keywords>
        <indexterm>databases</indexterm>
      </keywords>
    </topicmeta>
  </topicref>
  <!-- ... -->
</map>
```

A simple index range will look something like this:

```
<indexterm start="cheese">cheese</indexterm>
<!-- ... additional content -->
<indexterm end="cheese"/>
```

The previous combination of terms will generate a top-level index term for "cheese" that covers a series of pages, such as:

- Cheese 18-24

Figure 7: A simple index range

Specifying a range for nested terms is similar. In this sample, the range is specified for the tertiary index entry "pecorino":

```
<indexterm>cheese
  <indexterm>sheeps milk
    <indexterm start="level-3-pecorino">pecorino</indexterm>
  </indexterm>
</indexterm>
 <!-- ... additional content ... -->
<indexterm end="level-3-pecorino"/>
```

The generated index for that range would look something like this:

- cheese

    - sheeps milk

        - pecorino 18-24

Figure 8: A more complex index range

---

2 DITA 1.3 content

2.1 Indexes

An index can be generated from index entries that occur in topic bodies, topic prologs, or DITA maps.

The specialized indexing domain also provides elements to enable additional indexing function, such as "See" and "See also".

For more information, see Indexing group elements (12).

2.2 Indexing group elements

The indexing domain provides elements for use with indexing. The elements allow authors to define "See" and "See also" references, and to override the default sort order for a term.

Indexing domain elements typically work with the `<indexterm>` and `<index-base>` elements; `<index-base>` is grouped with elements that are typically useful only in specialization contexts.

2.2.1 <index-see>

An `<index-see>` element within an `<indexterm>` element redirects the reader to another index entry that the reader should reference instead of the current one.

The `<index-see>` and `<index-see-also>` elements allow a form of redirection to another index entry within the generated index. The `<index-see>` element refers to an index entry that the reader should use *instead of* the current one, whereas the `<index-see-also>` element refers to an index entry that the reader should use *in addition to* the current one.

Processors should ignore `<index-see>` and `<index-see-also>` elements if their parent `<indexterm>` element contains any `<indexterm>` children.

Because an `<index-see>` indicates a redirection to use instead of the current entry, it is an error if, for any `<index-see>`, there is also an `<index-see-also>` or an `<indexterm>` for the same index entry (that is, another entry with an identical sort key). For example, if an `<indexterm>` element with the content "Memory stick" also includes `<index-see>USB drive</index-see>`, it is an error if there is also an `<indexterm>` with the contents "Memory stick". This is to prevent index entries that are both a redirect and a page reference, such as:

* Memory stick 42, 106
* See USB drive

An implementation **MAY** give an error message when it encounters this condition, and **MAY** recover from this error condition by treating the `<index-see>` as an `<index-see-also>`.

There can be multiple `<index-see>` elements for a single index entry.

Inheritance

+ topic/index-base indexing-d/index-see

Example

The following example illustrates the use of an `<index-see>` redirection element within an `<indexterm>`:

```
<indexterm>Carassius auratus
    <index-see>Goldfish</index-see>
</indexterm>
```

This will typically generate an index entry without a page reference:

- Carassius auratus, *see* Goldfish

The following example illustrates the use of an `<index-see>` redirection element to a more complex (multilevel) `<indexterm>`:

```
<indexterm>Feeding goldfish
    <index-see>Goldfish <indexterm>feeding</indexterm></index-see>
</indexterm>
```

This is part of the indexing markup that might generate index entries such as:

- Feeding goldfish
    - *see* Goldfish feeding
- Goldfish
    - feeding, 56
    - flushing, 128, 345

The following example illustrates using a specialization of `<ph>` within `<index-see>`:

```
<indexterm>Einstein's mass and energy equation
   <index-see>E=mc<sup>2</sup></index-see>
</indexterm>
```

Attributes

The following attributes are available on this element: Universal attribute group and `@keyref`.

2.2.2 <index-see-also>

An `<index-see-also>` element within an `<indexterm>` redirects the reader to another index entry that the reader should reference in addition to the current one.

The `<index-see>` and `<index-see-also>` elements allow a form of redirection to another index entry within the generated index. The `<index-see>` element refers to an index entry that the reader should use *instead of* the current one, whereas the `<index-see-also>` element refers to an index entry that the reader should use *in addition to* the current one.

Processors should ignore `<index-see>` and `<index-see-also>` elements if their parent `<indexterm>` element contains any `<indexterm>` children.

In addition to its "see also" redirection, an `<index-see-also>` functions as a pointwise index term, thereby typically generating a page reference as well as the "see also" indication.

It is not an error for there to be multiple `<index-see-also>` elements for a single index entry.

Inheritance

+ topic/index-base indexing-d/index-see-also

Example

The following example illustrates the use of an `<index-see-also>` redirection element within an `<indexterm>`:

```
<indexterm>Carp
    <index-see-also>Goldfish</index-see-also>
</indexterm>
```

This will typically generate a page reference to "Carp" and a redirection:

- Carp, 56
    - *see also* Goldfish

The following example illustrates the use of an `<index-see-also>` redirection element to a more complex (multilevel) `<indexterm>`:

```
<indexterm>Feeding
    <index-see-also>Goldfish <indexterm>feeding</indexterm></index-see-also>
</indexterm>
```

This is part of the indexing markup that might generate index entries such as:

- Feeding, 348
    - *see also* Goldfish feeding
- Goldfish
    - feeding, 56
    - flushing, 128, 345

The following example illustrates using a specialization of `<ph>` within `<index-see-also>`:

```
<indexterm>µ = E<sub>0</sub>/V<sup>2</sup>
  <index-see-also>E=mc<sup>2</sup></index-see-also>
</indexterm>
```

Attributes

The following attributes are available on this element: Universal attribute group and `@keyref`.

2.2.3 <indexterm>

The content of an `<indexterm>` element is used to produce an index entry in a generated index. You can nest `<indexterm>` elements to create multi-level indexes. The content is not output as part of the topic content, only as part of an index.

An `<indexterm>` element without the `@start` or `@end` attribute specified is interpreted as a point reference that contributes the number of the current page to an index entry; the content of the index entry is the content of the `<indexterm>` element. All `<indexterm>` elements with the same content are "merged" to form a single index entry in the resulting index, and all contributed page numbers are included in that index entry.

In the case of nested `<indexterm>` elements, each `<indexterm>` with no `<indexterm>` children (the "leaves") contributes a page number to the generated index; the ancestral `<indexterm>` elements for each leaf `<indexterm>` provide the higher levels for the multilevel entry.

An `<indexterm>` that occurs in a topic `<prolog>` is interpreted as a point reference to the title of the topic. Likewise, in a DITA map, an `<indexterm>` that occurs in `<topicmeta>` inside of a `<topicref>` is interpreted as a point reference to the title of the referenced topic.

It is an error if an `<indexterm>` containing no `<indexterm>` children contains both an `<index-see>` and an `<index-see-also>`. (Note: `<index-see>` and `<index-see-also>` elements within `<indexterm>` elements that do contain `<indexterm>` children are ignored.) In the case of this error condition, an implementation **MAY** give an error message, and might recover by treating all such `<index-see>` elements as `<index-see-also>` elements.

> **Note** The `<index-see>` and `<index-see-also>` elements are domain specializations of the `<index-base>` element, and are discussed in detail with the indexing domain.

The `@start` and `@end` attribute on `<indexterm>` can be used in cases where one wants to index an extended discussion that might continue over a number of pages. The start of a range is indicated by an `<indexterm>` with a `@start` attribute. The end of a range is indicated with an `<indexterm>` with an

@end attribute whose value matches that of the @start attribute on the start-of-range <indexterm>. Such markup contributes to the generated index a page range covering all pages in the index range.

The end-of-range <indexterm> should have no content of its own; if it contains content, that content is ignored. There is no reason for the end-of-range <indexterm> to have any <indexterm> ancestors; however, an implementation should be able to handle an end-of-range <indexterm> that is nested within one or more <indexterm> elements.

The @start and @end attributes are defined as CDATA, although it is a best practice that the values should not contain any whitespace characters (such as a space or tab) or control characters. Matching of @start and @end attributes is done as a character-by-character comparison with all characters significant and no case folding occurring. The @start and @end attributes are ignored if they occur on an <indexterm> element that has child <indexterm> elements.

Index range indications can occur in the <topicmeta> of a <topicref> at the map level, in the prolog of a topic, or in the body of a topic, and are interpreted as follows (see Figure 11: Index ranges (16) for samples):

- In a map, the start range points to the start of the topic title of the topic being referenced by its containing <topicref>. The end range points to the end of the final child contained by the topic being referenced by its containing <topicref>, or to the end of the final topic referenced by the current map (whichever comes first). When a start and end range occur in the same <topicmeta>, the range applies to the containing <topicref> and its children.
- In the prolog of a topic, the start range points to the start of the containing topic's title. The range ends with a matching index range end in the same prolog, regardless of whether the end range is specified. The range applies to the containing topic and all its children including child relationships defined in a map.
- In the body of a topic, the range starts where the start <indexterm> occurs and ends at the matching index range end indication within the same body, or at the end of the body, whichever comes first. Such an index range does not span sub-topics of the topic.

When index ranges with the same identifier overlap, the widest range applies, and end ranges are matched with start ranges by last-in-first-out. In other words, the ranges are interpreted as nested rather than overlapping with the highest-level container taking precedence over narrower contained ranges.

As defined above, there is no such thing as an index range start that isn't terminated by either a matching end or some maximum scope. There can, however, be unmatched index range end indications; these are ignored.

Inheritance

- topic/indexterm

Example

- The following <indexterm> is a point reference to a specific paragraph within a topic:

```
<p><indexterm>databases</indexterm>Databases are used to ...</p>
```

- The following <indexterm> is a point reference to the start of the title of the concept:

```
<concept id="db">
  <title>About databases</title>
  <prolog>
    <metadata>
      <keywords><indexterm>databases</indexterm></keywords>
    </metadata>
  </prolog>
```

```
  <body><!-- content... --></body>
</concept>
```

- The following `<indexterm>` is a point reference to the start of the title of `aboutdatabases.dita`:

```
<topicref href="aboutdatabases.dita">
  <topicmeta>
    <keywords><indexterm>databases</indexterm></keywords>
  </topicmeta>
  <!-- other topicref elements -->
</topicref>
```

Figure 9: Single point index terms

The following sample represents three levels of index markup:

```
<indexterm>cheese
  <indexterm>sheeps milk
    <indexterm>pecorino</indexterm>
  </indexterm>
  <indexterm>goats milk
    <indexterm>chevre</indexterm>
  </indexterm>
</indexterm>
```

The previous sample is equivalent to the following sample:

```
<indexterm>cheese
  <indexterm>sheeps milk
    <indexterm>pecorino</indexterm>
  </indexterm>
</indexterm>
<indexterm>cheese
  <indexterm>goats milk
    <indexterm>chevre</indexterm>
  </indexterm>
</indexterm>
```

In each case, a generated index would include something like the this:

- cheese
    - goats milk
        - chevre 14
    - sheeps milk
        - pecorino 14

Figure 10: Nested index terms

A simple index range will look something like this:

```
<indexterm start="cheese">Cheese</indexterm>
<!-- ... additional content -->
<indexterm end="cheese"/>
```

The previous combination of terms will generate a top-level index term for "Cheese" that covers a series of pages, such as:

- Cheese 18-24

Specifying a range for nested terms is similar. In this sample, the range is specified for the tertiary index entry "pecorino":

```
<indexterm>cheese
  <indexterm>sheeps milk
    <indexterm start="level-3-pecorino">pecorino</indexterm>
  </indexterm>
</indexterm>
 <!-- ... additional content ... -->
<indexterm end="level-3-pecorino"/>
```

The generated index for that range would look something like this:

- cheese
    - sheeps milk
        - pecorino 18-24

There are three locations that can declare a range - the body of a topic, the prolog of a topic, and a map.

- In the following example, the range begins at the start of the second paragraph, and continues to the last paragraph. If the matching end range was not included, the range would end at the end of the body element.

  ```
  <topic id="accounting">
    <title>Accounting regulations</title>
    <body>
      <p>Be ethical in your accounting.</p>
      <p><indexterm start="acctrules">Rules</indexterm>Remember to do all of the
  following: ...</p>
      <!-- ...pages worth of rules... -->
      <p><indexterm end="acctrules"/>Failure to comply will get you audited.</p>
    </body>
    <!-- Potential sub-topics -->
  </topic>
  ```

- In the following example, the range begins with the start of the topic's title, and covers the entire topic and any sub-topics. The range ends within the same prolog, regardless of whether `<indexterm end="acct"/>` is specified in the prolog.

  ```
  <topic id="accounting">
    <title>Accounting regulations</title>
    <prolog>
      <metadata>
        <keywords><indexterm start="acct">Accounting</indexterm></keywords>
      </metadata>
    </prolog>
    <!-- Body and sub-topics -->
  </topic>
  ```

- Now assume that the topic in the previous sample is named `acct.dita`. Ranges defined in a prolog cover sub-topics, including those nested based on a map; in the following example, this means that the range covers all of `acct.dita`, as well as `procedures.dita` and `forms.dita`:

  ```
  <topicref href="acct.dita">
    <topicref href="procedures.dita"/>
    <topicref href="forms.dita"/>
  </topicref>
  ```

- In the final example, the range is specified in a map. The index range for "Accounting" begins with the start of the first topic title in `acct.dita`, and covers that file as well as any sub-topics. The index range for "Government forms" begins with the start of the first topic title in `acct.dita`, and

continues until the end of the last element in the file `taxfiling.dita`. If the end range for "govt" was not specified, the range would continue to the end of the map.

```
<topicref href="acct.dita">
  <topicmeta>
    <keywords>
      <indexterm start="acct">Accounting</indexterm>
      <indexterm end="acct"/>
      <indexterm start="govt">Government forms</indexterm>
    </keywords>
  </topicmeta>
  <!-- Nested topicref elements -->
</topicref>
<topicref href="taxfiling.dita">
  <topicmeta>
    <keywords>
      <indexterm end="govt"/>
    </keywords>
  </topicmeta>
</topicref>
```

Figure 11: Index ranges

```
<p>Einstein's most famous equation
E=mc<sup>2</sup><indexterm>E=mc<sup>2</sup></indexterm>
expresses the relationship between mass and energy.</p>
```

All the elements in the highlighting domain are specializations of `<ph>`.

Figure 12: Index term with <ph> or <ph> specializations

Attributes

The following attributes are available on this element: Universal attribute group, `@keyref`, and the attributes defined below.

@start
> Specifies that an index entry is positioned at the beginning of a range. The value matches the `@end` attribute on another `<indexterm>`.

@end
> Specifies that an index entry is positioned at the end of a range; value matches the `@start` attribute on another `<indexterm>`.

Index

C

Carassius auratus, *See* goldfish

E

elements
    indexing group
        `<index-see>` 7
        `<index-see-also>` 8
        `<indexterm>` 10
examples
    indexes 5–7

I

`<index-see>` 7
`<index-see-also>` 8
indexes
    elements 3
    examples 5–7
    location of elements 3
    page references 3
    ranges 5
    redirection 4
`<indexterm>` 10

P

processing expectations
    index ranges 5