

# DITAweb: Review B

---

## Table of contents

|                                   |           |
|-----------------------------------|-----------|
| Basic map elements.....           | 3         |
| <anchor>.....                     | 3         |
| <keytext>.....                    | 4         |
| <map>.....                        | 5         |
| <navref>.....                     | 6         |
| <relcell>.....                    | 7         |
| <relcolspec>.....                 | 8         |
| <relheader>.....                  | 10        |
| <relrow>.....                     | 10        |
| <reltable>.....                   | 10        |
| <topicref>.....                   | 12        |
| <topicmeta>.....                  | 13        |
| <ux-window>.....                  | 14        |
| Domain elements.....              | 16        |
| Emphasis domain elements.....     | 16        |
| <em>.....                         | 16        |
| <strong>.....                     | 16        |
| Highlighting domain elements..... | 17        |
| <b>.....                          | 17        |
| <i>.....                          | 17        |
| <line-through>.....               | 18        |
| <overline>.....                   | 18        |
| <sub>.....                        | 18        |
| <sup>.....                        | 19        |
| <tt>.....                         | 19        |
| <u>.....                          | 20        |
| Mapgroup domain elements.....     | 20        |
| <anchorref>.....                  | 20        |
| <keydef>.....                     | 22        |
| <mapref>.....                     | 23        |
| <mapresources>.....               | 25        |
| <topicgroup>.....                 | 26        |
| <topichead>.....                  | 26        |
| Index.....                        | <b>28</b> |

---

## Basic map elements

DITA maps are built from a few core elements that are used for referencing and organizing topics. The `<topicmeta>` element also is available to specify metadata for the map, for individual topics, or for groups of topics. Many elements inside `<topicmeta>` also are available inside the topic prolog.

### <anchor>

An anchor within a map is an integration point that another map can reference in order to insert its navigation into the navigation tree of the referenced map.

### Usage information

The `<anchor>` element typically is used to allow integration of run-time components. For build-time integration, you can use a `<topicref>` element to reference another map, or use the `@conref` or `@conkeyref` attribute on an element inside the map.

### Processing expectations

The mechanism by which map processors discover maps to be anchored is processor specific.

### Attributes

The following attributes are available on this element: [universal attributes](#).

For this element, the `@id` attribute is required. It is referenced by `<anchorref>` or by the `@anchorref` attribute on `<map>` elements.

### Example

This section contains examples of how the `<anchor>` element can be used.

#### Figure 1: DITA map that contains an anchor

The following code sample shows how a map creates an `<anchor>` element with an `@id` attribute set to "a1".

```
<map>
  <title>MyComponent tasks</title>
  <topicref href="start.dita" toc="yes">
    <navref mapref="othermap2.ditamap"/>
    <navref mapref="othermap3.ditamap"/>
    <anchor id="a1"/>
  </topicref>
</map>
```

The `@id` on an `<anchor>` element can be referenced by the `@anchorref` attribute on the `<map>` element in another map.

#### Figure 2: DITA map that references an anchor

In the following code sample, the DITA map references a DITA map using the `@anchorref` attribute:

```
<map anchorref="map1.ditamap#a1">
  <title>This map is can be rendered at the "a1" anchor
    in the MyComponent task map</title>
```

```
<!-- ... -->
</map>
```

## <keytext>

Key text is variable or link text that is used when resolving key references. It also specifies alternate text for images that are referenced by keys.

### Processing expectations

See [Processing key references to generate text or link text](#).

### Attributes

The following attributes are available on this element: [universal attributes](#).

### Examples

The section contains examples of how the <keytext> element can be used.

#### Figure 3: Simple example

The following code sample shows how variable text can be defined using the <keytext> element:

```
<keydef keys="company-name">
  <topicmeta>
    <keytext translate="no">Acme Widget Company</keytext>
  </topicmeta>
</keydef>
```

#### Figure 4: More complex example

The following code sample shows a variable-text definition that includes highlighting elements:

```
<keydef keys="company-name">
  <topicmeta>
    <keytext translate="no">
      <i>Super</i> Widget Squared<sup>2</sup>
    </keytext>
  </topicmeta>
</keydef>
```

#### Figure 5: Alternate text for an image

DITA implementations often reference images using keys. In such cases, the <keytext> element provides the alternate text for the image. The following code sample shows the markup for the <keytext> element:

```
<keydef keys="company-logo" href="images/logo.jpg" format="jpg">
  <topicmeta>
    <keytext>Acme Widgets logo</keytext>
  </topicmeta>
</keydef>
```

The image can be referenced by `<image keyref="company-logo"/>`. When rendered to mediums that support alternate text, the effective alternative text for the image is "Acme Widgets logo" as though a literal `<alt>` element had been a child of the `<image>`

### Figure 6: Variable text that is conditionally processed

DITA implementations often need to conditionally process product names. The following code sample shows a `<keytext>` element that contains `<ph>` elements that are conditionally processed:

```
<keydef keys="company-name">
  <topicmeta>
    <keytext translate="no">
      <ph product="cat">Acme Widgets for Cats</ph>
      <ph product="dog">Acme Widgets for Dogs</ph>
      <ph product="pig">Acme Widgets for Pigs</ph>
    </keytext>
  </topicmeta>
</keydef>
```

### Figure 7: Processing logic

The following sample shows a key definition that includes several elements within the `<topicmeta>` element:

```
<keydef href="http://www.example.com" keys="company-name" format="html" scope="external">
  <topicmeta>
    <keytext>Acme Tools</keytext>
    <navtitle>Acme Tools web site</navtitle>
    <linktitle>Acme Tools Web Portal</linktitle>
  </topicmeta>
</keydef>
```

Once processed, the effective text content of both `<ph keyref="company-name"/>` and `<xref keyref="company-name"/>` is "Acme Tools".

To set distinct text values for both the company name and the link text that is associated with the company Web site, best practices call for using two different key definitions.

## <map>

A DITA map is the mechanism for aggregating topic references and defining a context for those references. It contains references to topics, maps, and other resources; these references are organized into hierarchies, groups, and tables.

## Usage information

A map describes the relationships among a set of DITA topics. The following are types of relationships that can be described in a map:

### Hierarchical

Nested topics create a hierarchical relationship. The topic that does the nesting is the parent, and the topics that are nested are the children.

### Ordered

Child topics can be labeled as having an ordered relationship, which means they are referenced in a definite sequence.

### Family

Child topics can be labeled as having a family relationship, which means they all refer to each other.

**Comment by Kristen J Eberlein on 10 November 2021**

I moved this content from "Rendering expectations," where I do not think it belonged. I think we probably ought to be clearer about the scenarios in which titles are rendered; certainly users get confused about this. And do we cover processing expectations for submaps somewhere?

The `<title>` element can be used to provide a title for the map. In some scenarios the title is purely informational; it is present only as an aid to the author. In other scenarios, the title might be useful or even required. In a map referenced by another map, the title might be discarded as topics from the submap are aggregated into a larger publication.

## Rendering expectations

When rendering a map, processors might make use of the relationships defined in the map to create a table of contents (TOC), aggregate topics into a PDF document, or create links between topics in the output.

## Attributes

The following attributes are available on this element: [universal attributes](#), [common map attributes](#), [architectural attributes](#), `@type`, `@scope`, `@format`, and the attribute defined below.

### @anchorref

Identifies a location within another map document where this map will be anchored. Resolution of the map is deferred until the final step in the delivery of any rendered content. For example, `anchorref="map1.ditamap#a1"` allows the map with `@anchorref` to be pulled into the location of the anchor point "a1" inside `map1.ditamap` when `map1.ditamap` is rendered for delivery.

## Example

The following code sample contains six `<topicref>` elements. The `<topicref>` elements are nested and have a hierarchical relationship. The file `bats.dita` is the parent topic, and the other topics are its children. The hierarchy could be used to generate a PDF, a navigation pane in an information center, a summary of the topics, or related links between the parent topic and its children.

```
<map id="mybats">
  <title>Bats</title>
  <topicref href="bats.dita">
    <topicref href="batcaring.dita"/>
    <topicref href="batfeeding.dita"/>
    <topicref href="batsonar.dita"/>
    <topicref href="batguano.dita"/>
    <topicref href="bathistory.dita"/>
  </topicref>
</map>
```

## <navref>

A navigation reference is a reference to another map that is preserved as a transcluding link in the result deliverable, rather than resolved when the deliverable is produced. Output formats that support such linking can integrate the referenced resource when displaying the referencing map to an end user.

## Usage information

The `<navref>` element is intended as a reference to a navigation resource that can be resolved at rendering time. It enables DITA maps to be published into a help system where the referenced navigation is published independently (or might not be available at all). If available, the referenced navigation can then be resolved at rendering time within a help system.

### Comment by Kristen J Eberlein on 10 November 2021

Does the following information really belong here? It seems to be very basic map info.

In order to include another map directly without depending on the output format or help system, use a `<topicref>` element with the `@format` attribute set to `ditamap`. The effect is similar to using a `@conref` attribute. For example, the following markup represents a literal inclusion of the map `other.ditamap`:

```
<topicref href="other.ditamap" format="ditamap"/>
```

## Attributes

The following attributes are available on this element: [universal attributes](#) and the attribute defined below.

### @mapref

Specifies the URI of the map file or non-DITA resource to be referenced. It might reference a DITA map or a resource that is appropriate for a target help system. For example, it could reference an XML TOC file for use with Eclipse help.

## Example

In the following code sample, the map titled "MyComponent tasks" references the maps `othermap2.ditamap` and `othermap3.ditamap`.

```
<map>
  <title>MyComponent tasks</title>
  <navref mapref="../com.example.plugin.xml.doc/othermap1.ditamap"/>
  <navref mapref="../com.example.plugin.xml.doc/othermap2.ditamap"/>
</map>
```

## <relcell>

A cell in a relationship table is a group of one or more topic references that are related to the topic references in other cells of the same row.

## Usage information

A relationship table cell does not imply a relationship between topics or resources that are referenced in the same cell, unless the `@collection-type` attribute cell indicates that they are related.

## Attributes

The following attributes are available on this element: [universal attributes](#) and [common map attributes](#) (without `@keyscope`), `@type`, `@scope`, `@format`.

## Example

See [reltable](#) (10).

## <relcolspec>

A column specification in a relationship table column provides default attribute values for the references in that column of a relationship table.

### Usage information

You can use the <relcolspec> element to set default values for the attributes of the topics that are referenced in the column. For example, when you set the @type attribute to "concept", all <topicref> elements in the column that do not have a @type attribute specified are treated as concepts.

Adding a <topicref> element to the <relcolspec> element defines a relationship between the topic (or topics) that are contained within the <relcolspec> element and the topics that are referenced in the column of the relationship table. Note that this does not define a relationship between two cells in the same column; the only new relationship is between <topicref> targets in a <relcell> and <topicref> targets in that <relcolspec> of a column.

### Rendering expectations

When a <title> element exists inside of the <relcolspec> element, the content of the <title> element is intended to be used as the label for the related links that are defined and generated by the column. If the <title> element is not present, the labels for the related links are generated in the following ways:

- If the <relcolspec> element contains a <topicref> element that specifies a navigation title, that navigation title is used for the label.
- If the <relcolspec> element contains a <topicref> element that does not specify a navigation title but does reference a DITA topic, the label is derived from the navigation title of the referenced topic or, lacking that, the title of the topic.
- If no title is specified and no <topicref> is present in the <relcolspec>, a rendering tool might choose to generate a title for the links generated from that column.

### Processing expectations

When values are specified for attributes of <relcell> or <relrow> elements, those values override those defined for <relcolspec> attributes. Values specified for attributes of <relcolspec> elements override those defined for the <reltable> element.

### Attributes

The following attributes are available on this element: [universal attributes](#) and [common map attributes](#) (without @keyscope or @collection-type), @type, @scope, @format.

### Example

The following section contains examples of how the <relcolspec> element can be used.

#### Figure 8: Enforcing concept, type, and reference types with <relcolspec>

The following code sample shows how a <relcolspec> element can be used to define the types of topics that are referenced in a column. Three cells are defined within one row. The first cell contains one concept topic: puffins.dita. The second cell contains two task topics: puffinFeeding.dita and puffinCleaning.dita. The third cell contains a reference topic: puffinHistory.dita. Setting the



@type on each column allows (but does not require) processors to validate that the topics in each column are of the expected type.

```
<map>
  <reltable>
    <relheader>
      <relcolspec type="concept"/>
      <relcolspec type="task"/>
      <relcolspec type="reference"/>
    </relheader>
    <relrow>
      <relcell><topicref href="puffins.dita"/></relcell>
      <relcell>
        <topicref href="puffinFeeding.dita"/>
        <topicref href="puffinCleaning.dita"/>
      </relcell>
      <relcell>
        <topicref href="puffinHistory.dita"/>
      </relcell>
    </relrow>
  </reltable>
</map>
```

**Figure 9: Relationship table column headers with topics and titles**

The following code sample shows how topics and titles can be specified in a column header for relationship table column header:

```
<reltable>
  <relheader>
    <relcolspec type="task">
      <topicref href="tbs.dita">
        <topicmeta><navtitle>Troubleshooting</navtitle></topicmeta>
      </topicref>
    </relcolspec>
    <relcolspec type="reference">
      <topicref href="msg.dita">
        <topicmeta><navtitle>Messages</navtitle></topicmeta>
      </topicref>
    </relcolspec>
  </relheader>
  <relrow>
    <relcell>
      <topicref href="debug_login.dita"/>
      <topicmeta><linktitle>Debugging login errors</linktitle></topicmeta>
    </topicref>
    </relcell>
    <relcell>
      <topicref href="login_error_1.dita">
        <topicmeta><linktitle>Login not found</linktitle></topicmeta>
      </topicref>
    </relcell>
  </relrow>
  <relrow>
    <relcell>
      <topicref href="checking_access.dita">
        <topicmeta><linktitle>Checking access controls</linktitle></topicmeta>
      </topicref>
    </relcell>
    <relcell>
      <topicref href="login_error_2.dita">
        <topicmeta><linktitle>Login not allowed</linktitle></topicmeta>
      </topicref>
    </relcell>
  </relrow>
</reltable>
```

In addition to the relationships defined by the rows in the relationship table, the following relationships are now defined by the columns in the relationship table:

- `tbs.dita <--> debug_login.dita`
- `tbs.dita <--> checking_access.dita`
- `msg.dita <--> login_error_1.dita`
- `msg.dita <--> login_error_2.dita`

Ignoring the headers for a moment, the `<reltable>` here would ordinarily define a two-way relationship between `debug_login.dita` and `login_error1.dita`. This typically will be expressed as a link from each to the other. An application might render the link with a language-appropriate heading such as "Related reference", indicating that the target of the link is a reference topic.

The headers change this by specifying a new title. In the second column, the `<topicref>` specifies a title of "Messages", which should now be used together with the link to anything in that column. So, a generated link from `debug_login.dita` to `login_error1.dita` should be rendered together with the title of "Messages". How this is rendered together with the link is up to the application.

## <relheader>

A header row in a relationship table is a group of column definitions for a relationship table.

### Attributes

The following attributes are available on this element: [universal attributes](#).

### Example

See [reltable](#) (10).

## <relrow>

A row in a relationship table creates a relationship between the cells in that row, which is often expressed in output as links between the topics or resources that are referenced in those cells.

### Attributes

The following attributes are available on this element: [universal attributes](#).

### Example

See [reltable](#) (10).

## <reltable>

A relationship table is a mechanism that creates relationships among topics, based on the familiar table model of rows, columns, and cells.

### Usage information

Each column in a relationship table typically represents a specific role in a set of relationships. For example, a frequently-used type of relationship table uses the first column to contain references to task topics, while the second and third columns reference concept and reference topics. The relationship table rows define relationships between the resources referenced in different cells of the same row; in this example, each row establishes relationships between task topics and the concept and reference topics that support the tasks. When used in this manner, relationship tables can make it easy to determine where related information is missing or undefined.

Relationship tables also can be used in conjunction with hierarchies and groups to manage all the related links in an information set.

When a title is associated with a relationship table, the title typically is used as an authoring convenience and is not displayed in generated publications.

## Processing expectations

By default, the contents of a `<reltable>` element are not rendered in a table of contents; they are used only to define relationships that can be expressed as topic-to-topic links. The `<relcell>` elements can contain `<topicref>` elements, which are then related to other `<topicref>` elements in the same row (although not necessarily in the same cell).

Within a root map, the effective relationship table is the union of all relationship tables in the map hierarchy.

## Attributes

The following attributes are available on this element: [universal attributes](#), [common map attributes](#) (without `@keyscope` or `@collection-type`), `@type`, `@scope`, `@format`, and the attributes defined below.

For this element, the `@toc` attribute has a default value of "no".

### @title

Specifies an identifying title for the element.

## Example

In the following code sample, a relationship table is defined with three columns: one for "concept", one for "task", and one for "reference". Three cells are defined within each row. The first cell contains one concept topic: `about-MyDevice.dita`. The second cell contains two task topics: `setting-up-MyDevice.dita` and `operating-MyDevice.dita`. The third cell contains two reference topics: `MyDevice-settings.dita` and `MyDevice-version-info.dita`.

```
<map>
  <reltable>
    <relheader>
      <relcolspec type="concept"/>
      <relcolspec type="task"/>
      <relcolspec type="reference"/>
    </relheader>
    <relrow>
      <relcell>
        <topicref href="about-MyDevice.dita"/>
      </relcell>
      <relcell>
        <topicref href="setting-up-MyDevice.dita"/>
        <topicref href="operating-MyDevice.dita"/>
      </relcell>
      <relcell>
        <topicref href="MyDevice-settings.dita"/>
        <topicref href="MyDevice-version-info.dita"/>
      </relcell>
    </relrow>
  </reltable>
</map>
```

A graphical version of the relationship table in an editor might look like this:

| type="concept"      | type="task"   | type="reference"                                     |
|---------------------|---|--|
| about-MyDevice.dita | setting-up-MyDevice.dita<br>operating-MyDevice.dita | MyDevice-settings.dita<br>MyDevice-version-info.dita |

When rendered, links are added to topics that are in the same row, but not in the same cell. This allows simple maintenance of parallel relationships: for example, in this case, `setting-up-MyDevice.dita` and `operating-MyDevice.dita` are two tasks that require the same supporting information (concept and reference topics) but might otherwise be unrelated. When topics in the same cell are in fact related, the `@collection-type` attribute for the cell can be set to "family". If some cells or columns are intended solely as supporting information and should not link back to topics in other cells, you can set the `@linking` attribute on the `<relcell>` or `<relcolspec>` to "targetonly".

In this example, the related links would be as follows:

#### **about-MyDevice.dita**

`setting-up-MyDevice.dita`, `operating-MyDevice.dita`, `MyDevice-settings.dita`,  
`MyDevice-version-info.dita`

#### **setting-up-MyDevice.dita**

`about-MyDevice.dita`, `MyDevice-settings.dita`, `MyDevice-version-info.dita`

#### **operating-MyDevice.dita**

`about-MyDevice.dita`, `MyDevice-settings.dita`, `MyDevice-version-info.dita`

#### **MyDevice-settings.dita**

`about-MyDevice.dita`, `setting-up-MyDevice.dita`, `operating-MyDevice.dita`

#### **MyDevice-version-info.dita**

`about-MyDevice.dita`, `setting-up-MyDevice.dita`, `operating-MyDevice.dita`

Relationship tables are inherently an efficient way to manage these links. In particular, they increase the prospect for reuse among topics, because those topics do not contain context-specific links. A relationship table also makes it easy to see and manage patterns; for example, the fact that `operating-MyDevice.dita` and `setting-up-MyDevice.dita` have the same relationships to supporting information is clear from the table, but would require some comparison and counting to determine from the list summary just before this paragraph.

## **<topicref>**

A topic reference is the mechanism for referencing a topic (or another resource) from a DITA map. It can nest, which enables the expression of navigation and table-of-content hierarchies, as well as containment hierarchies and parent-child relationships.

### **Attributes**

The following attributes are available on this element: [universal attributes](#), [link-relationship attributes](#) (with a narrowed definition of `@href`, given below), [common map attributes](#), `@keys`, and `@keyref`.

#### **Comment by robander on 15 May 2021**

Most attribute sections with one-off narrowing or clarification have been converted to a paragraph – "On this element, href does xyz", while the actual definition still applies. We plan to convert this into an attribute-exception paragraph, reminding that `topicref` is meant to refer to topic-level elements, but before doing that, need to make sure the `href` topic is updated with the same info in arch spec.

#### **@href**

Points to the resource that is represented by the `<topicref>`. See for detailed information on supported values and processing implications. References to DITA content cannot be below the topic

level: that is, you cannot reference individual elements inside a topic. References to content other than DITA topics should use the `@format` attribute to identify the kind of resource being referenced.

## Example

The following code sample shows a simple map that organizes several topics about the software product "Example Tool Builder". The `<topicref>` that refers to `setup.dita` uses the `@collection-type` attribute to indicate that the order of three children topics in that section is important.

```
<map>
  <title>Example Tool Builder version 1.2.3</title>
  <topicref href="setup.dita" collection-type="sequence">
    <topicref href="prerequisites.dita"/>
    <topicref href="installing.dita"/>
    <topicref href="validating.dita"/>
  </topicref>
  <topicref href="everyday-use.dita">
    <!-- ... -->
  </topicref>
  <topicref href="troubleshooting.dita">
    <!-- ... -->
  </topicref>
</map>
```

## <topicmeta>

Topic metadata is metadata that applies to a topic based on its context in a map.

## Usage information

The metadata specified in a `<topicmeta>` element is specific to a given context within a map. If a reference to a single resource appears more than once in a map or set of maps, unique metadata can be specified in each instance. For example, when the parent `<topicref>` element results in a link, elements within the `<topicmeta>` element can be used to provide context-specific information about the link, such as link text, a short description, or a navigation title.

## Attributes

The following attributes are available on this element: [universal attributes](#).

## Example

The following example shows how the `<topicmeta>` element can contain a link title and short description:

```
<map>
  <title>Indexing elements</title>
  <topicref href="indexing.dita">
    <topicmeta>
      <linktitle>Indexing for company specialists</linktitle>
      <shortdesc>Guidelines for indexing company materials</shortdesc>
    </topicmeta>
    <!-- Additional topic references -->
  </topicref>
</map>
```

When link previews for `indexing.dita` are generated, the link title and short description provided within the `<topicmeta>` element are used.

## Related concepts

[Cascading of metadata attributes in a DITA map](#)

## Related reference

[Reconciling topic and map metadata elements](#)

## <ux-window>

A UX window specification is a collection of metadata for a window or viewport in which a user assistance topic or web page can be displayed. The window or viewport can be referenced by the <resourceid> element that is associated with a topic or <topicref> element.

## Usage information

The <ux-window> element can be used in any <topicmeta> element in a map. If more than one <ux-window> element in a map has the same @name attribute, the first window specification in document order with that @name attribute is used.

## Attributes

The following attributes are available on this element: [ID and conref attributes](#), [metadata attributes](#), @class, and the attributes defined below.

### @name (REQUIRED)

Specifies the value used to refer to this window definition.

### @top

Specifies the top position of the target help window, whether relative to the calling window or to the entire display. The value of this attribute is a real number optionally followed by a unit of measure from the set of pc, pt, px, in, cm, mm, em (picas, points, pixels, inches, centimeters, millimeters, and ems respectively). The default unit is px (pixels).

### @left

Specifies the left position of the target help window, whether relative to the calling window or to the entire display. The value of this attribute is a real number optionally followed by a unit of measure from the set of pc, pt, px, in, cm, mm, em (picas, points, pixels, inches, centimeters, millimeters, and ems respectively). The default unit is px (pixels).

### @height

Specifies the height of the window. The value of this attribute is a real number optionally followed by a unit of measure from the set of pc, pt, px, in, cm, mm, em (picas, points, pixels, inches, centimeters, millimeters, and ems respectively). The default unit is px (pixels).

### @width

Specifies the width of the window. The value of this attribute is a real number optionally followed by a unit of measure from the set of pc, pt, px, in, cm, mm, em (picas, points, pixels, inches, centimeters, millimeters, and ems respectively). The default unit is px (pixels).

### @on-top

Indicates whether the initial z-order of the target help window is on top of all windows on the desktop. Allowable values are: "yes", "no", and [-dita-use-conref-target](#). The default value is "no".

### @features

Provides a list of other window features, such as size, position, or scroll bars. Each feature name and value can not contain any blank space, and each feature name and value is separated by a comma or other delimiter character.

## @relative

Indicates whether the window dimensions are relative to the calling window or the entire target display. The default value is "no". Allowable values are:

### no

The window dimensions specified on this element are absolute positions; they are not relative to the calling window.

### yes

The window dimensions specified on this element are relative to the calling window.

### -dita-use-conref-target

See for more information.

## @full-screen

Indicates whether the window is initially displayed in a maximized state. Allowable values are "yes", "no", and [-dita-use-conref-target](#). The default value is "no".

## Examples

This section shows how the `<ux-window>` and `<resourceid>` elements work together to define and use window definitions.

### Figure 10: Using `<ux-window>` with `<resourceid>`

The following code sample shows how a window with a name of "help" is defined in the map. The window name is later referenced by the `@ux-windowref` attribute on the `<resourceid>` element.

```
<map>
  <title>Widget Help</title>
  <topicmeta>
    <ux-window id="fg23" name="help" top="10" left="20" height="400" width="500"
      features="status=yes,toolbar=no,menubar=no,location=no" relative="yes"
      full-screen="no" />
  </topicmeta>
  <topicref href="file_ops.dita" type="concept">
    <topicref href="saving.dita" type="task" />
    <topicref href="deleting.dita" type="task" />
    <topicref href="editing.dita" type="task">
      <topicmeta>
        <resourceid id="ab43" appname="ua"
          appid="5432" ux-context-string="idh_fileedit" ux-windowref="help" />
      </topicmeta>
    </topicref>
  </topicref>
</map>
```

### Figure 11: Using multiple `<ux-window>` definitions

The following code sample shows how multiple window specifications can be defined for alternate presentations, such as desktop computers and tablets:

```
<map>
  <title>Puggles Help</title>
  <topicmeta>
    <ux-window id="p76" name="ux-tablet" top="1cm" left="1cm" height="4cm" width="3cm"
      features="status=no,toolbar=no,menubar=no,location=no" relative="no"
      full-screen="no" />
    <ux-window id="p80" name="ux-desktop" top="5cm" left="10cm" height="16cm" width="12cm"
      features="status=yes,toolbar=no,menubar=no,location=yes" relative="no"
      full-screen="no" />
  </topicmeta>
  <topicref href="c_puggles_intro.dita" type="concept">
    <!-- ... -->
  </topicref>
</map>
```

## Related concepts

[Context hooks and window metadata for user assistance](#)

---

# Domain elements

A domain is a grouping of related DITA elements that can be integrated into document-type shells. The base edition of DITA includes a variety of domains for use in topics and maps.

## Emphasis domain elements

The emphasis elements are used to indicate text that has special meaning or importance, or text that needs to be distinguished from surrounding text.

### <em>

Emphasis indicates special meaning or particular importance.

### Specialization hierarchy

The <em> element is specialized from <ph>. It is defined in the emphasis-domain module.

### Attributes

The following attributes are available on this element: [universal attributes](#) and [@keyref](#).

### Example

The following code sample shows how the <em> element can be used to emphasize a phrase in a paragraph:

```
<p>A good plan once adopted and put into execution <em>should not be abandoned</em> unless it becomes clear that it can not succeed.</p>
```

### <strong>

Strong text is text that is of greater importance than the surrounding text.

### Specialization hierarchy

The <strong> element is specialized from <ph>. It is defined in the emphasis-domain module.

### Attributes

The following attributes are available on this element: [universal attributes](#) and [@keyref](#).



## Example

The following code sample shows how the `<strong>` element can be used to highlight an important detail:

```
<p>Your doctor prescribed this medicine to treat an infection.  
It is important that you <strong>take all of the medicine</strong>  
as described.</p>
```

## Highlighting domain elements

The highlighting elements are used to highlight text with styles such as bold, italic, and monospaced. These elements are intended solely for use by authors when no semantically appropriate element is available and a formatting effect is required.

### <b>

Bold text is text that is used to draw a reader's attention to a phrase without otherwise adding meaning to the content.

### Specialization hierarchy

The `<b>` element is specialized from `<ph>`. It is defined in the highlighting-domain module.

### Attributes

The following attributes are available on this element: [universal attributes](#) and [@keyref](#).

## Example

The following code sample shows a `<b>` element used to draw a reader's attention to a phrase:

```
<p>Use the bold tag <b>for visual emphasis only</b>; do not use it if another phrase-level  
element better signifies the reason for the emphasis.</p>
```

### <i>

Italic text is text that is used to emphasize the key points in printed text, or when quoting a speaker, to show which words the speaker stressed.

### Specialization hierarchy

The `<i>` element is specialized from `<ph>`. It is defined in the highlighting-domain module.

### Attributes

The following attributes are available on this element: [universal attributes](#) and [@keyref](#).

## Example

The following code sample shows an `<i>` element used to indicate the use of a foreign word:

```
<note type="tip">Take care to measure the right amount when mixing ingredients.  
A <i>laissez-faire</i> attitude to baking is a recipe for disaster.</note>
```

## <del>

A strikethrough is a typographical presentation of words with a horizontal line through their center. It can indicate that words are a mistake and not intended for inclusion, or it can be used deliberately to imply a change of thought.

### Usage information

This element is designed to enable authors to indicate a deletion or revision for rhetorical purpose; it is not intended to be used for indicating revisions.

### Specialization hierarchy

The `<del>` element is specialized from `<ph>`. It is defined in the highlighting-domain module.

### Attributes

The following attributes are available on this element: [universal attributes](#) and [@keyref](#).

### Example

The following code sample shows a `<del>` element used to indicate a rhetorical revision:

```
<p>After writing up an angry post for social media, the author  
<del>wisely reconsidered</del> decided to wait a day before sharing.</p>
```

## <u>

An overline is a horizontal line that is printed above a line of text, a mathematical symbol, or an illustration in a newspaper or journal.

### Specialization hierarchy

The `<u>` element is specialized from `<ph>`. It is defined in the highlighting-domain module.

### Attributes

The following attributes are available on this element: [universal attributes](#) and [@keyref](#).

### Example

The following code sample shows an `<u>` element used to provide the highlighting used for mathematical notation:

```
<p>Overline: <u><i>x</i></u> is the  
average value of<i>x<sub>i</sub></i></p>
```

## <sub>

A subscript is text that is printed below the line. It is frequently used in chemical and mathematical formulas.

### Specialization hierarchy

The `<sub>` element is specialized from `<ph>`. It is defined in the highlighting-domain module.

## Attributes

The following attributes are available on this element: [universal attributes](#) and [@keyref](#).

## Example

The following code sample shows how the `<sub>` element is used in a chemical formula:

```
<note>When cleaning, be sure to dilute the baking soda (NaHCO<sub>3</sub>) with water (H<sub>2</sub>O) before mixing in the vinegar (CH<sub>3</sub>COOH).</note>
```

## <sup>

A superscript is text that is printed above the line. It is frequently used in chemical and mathematical formulas.

## Specialization hierarchy

The `<sup>` element is specialized from `<ph>`. It is defined in the highlighting-domain module.

## Attributes

The following attributes are available on this element: [universal attributes](#) and [@keyref](#).

## Example

The following code sample shows a `<sup>` element used to ensure proper formatting of the exponent in the number ten to the power of five:

```
<p>The power produced by the electrohydraulic dam was 10<sup>5</sup> more than the older electric plant.</p>
```

## <tt>

Teletype text is text that is displayed on a fixed-width display such as a teletype, text-only screen, or line printer.

## Specialization hierarchy

The `<tt>` element is specialized from `<ph>`. It is defined in the highlighting-domain module.

## Attributes

The following attributes are available on this element: [universal attributes](#) and [@keyref](#).

## Example

The following code sample shows how the `<tt>` element can be used to apply monospaced highlighting:

```
<p>Make sure that the screen displays <tt>File successfully created</tt> before proceeding to the next stage of the task.</p>
```

While the example demonstrates a potential use of `<tt>`, use `<systemoutput>` if you have access to the elements in the software domain.

## <u>

An underline, also called an underscore, is a line immediately below a portion of text.

### Specialization hierarchy

The <u> element is specialized from <ph>. It is defined in the highlighting-domain module.

### Attributes

The following attributes are available on this element: [universal attributes](#) and [@keyref](#).

### Example

The following code sample shows underlining used to provide emphasis in a marketing blurb, without giving any extra meaning to the underlined phrase:

```
<p>Using our patented <u>SuperFast BitSpeed Technology</u>, our product  
will answer all of your questions only a few nanoseconds after you ask!</p>
```

## Mapgroup domain elements

The mapgroup domain elements define, group, or reference content. Many of the mapgroup elements are convenience elements; they simply provide shortcuts for an author to use existing markup.

For example, the <topichead> element enables a map to specify a heading without a reference to a topic. While a <topicref> element might accomplish the same thing by creating a title and leaving off the @href attribute, the <topichead> element makes the intent clearer and prevents the accidental inclusion of an @href attribute.

### <anchorref>

An anchor reference integrates a map branch into a point (an anchor) in the same or different DITA map. The contents of the map branch are rendered at the location of both the <anchorref> and <anchor> elements.

### Usage information

The functionality of the <anchorref> element is similar to that provided by the @anchorref attribute of the <map> element. However, instead of attaching an entire map to an anchor point, this element enables the map author to attach only the contents of a single map branch. This enables map architects to reuse a branch of content without reusing the entire map.

To prevent the content of the map branch from being rendered at the location of the <anchorref> element, set `toc="no"` on the <anchorref> element, and then set `toc="yes"` on each of its children so that they will not inherit the `toc="no"` setting.

### Rendering expectations

When possible, the content of the <anchorref> element is rendered at the referenced anchor when the map with that <anchor> element is displayed in an authoring tool.

## Processing expectations

If the rendering platform does not support runtime integration of the content at the anchor point, processors treat the `<anchorref>` element similar to a "conref push" instruction by pushing the content to the spot of the `<anchor>`. Note that many `<anchorref>` elements might push content to the same point; the order in which items are pushed is left undefined, although the order within a single `<anchorref>` element is preserved.

Metadata cascading takes place in the original authored context, because the branch of content defined with the `<anchorref>` remains independent from the referenced map. The `<anchorref>` content does not take on the cascading metadata at the `<anchor>` location. For example, if the map containing the `<anchorref>` element sets a local copyright, that copyright cascades to the `<anchorref>` element and its children; it is retained after the content is rendered at the location of the `<anchor>` element.

## Specialization hierarchy

The `<anchoref>` element is specialized from `<topicref>`. It is defined in the `mapgroup` module.

## Attributes

The following attributes are available on this element: [universal attributes](#), [link-relationship attributes](#), [common map attributes](#), `@keys`, and `@keyref`.

For this element:

- The `@href` attribute refers to an `<anchor>` element in this or another DITA map.
- The `@format` attribute has a default value of "ditamap".
- The `@type` attribute has a default value of "anchor".

## Example

The section contains an example of how the `<anchorref>` element can be used.

### Figure 12: Map that contains `<anchor>` and `<anchorref>` elements

The following code sample shows a DITA map that contains both `<anchor>` and `<anchorref>` elements:

```
<topicref href="carPrep.dita">
  <topicref href="beforePrep.dita"/>
  <anchor id="prepDetail"/>
  <topicref href="afterPrep.dita"/>
</topicref>
<!-- ... -->
<topicref href="astroTasks.dita">
  <topicref href="astroOverview.dita"/>
  <anchorref href="#prepDetail">
    <topicref href="astroChecklist.dita"/>
    <topicref href="otherPreparation.dita"/>
  </anchorref>
</topicref>
```

```
<topicref href="astroConclusion.dita"/>
</topicref>
```

### Figure 13: Effective result of evaluating the `<anchorref>` element

After the processor evaluates the contents of the `<anchorref>` element, the effective result is as follows; the content of the anchor reference is rendered both at its original location and at the location of the anchor:

```
<topicref href="carPrep.dita">
  <topicref href="beforePrep.dita"/>
  <anchor id="prepDetail"/>
  <topicref href="astroChecklist.dita"/>
  <topicref href="otherPreparation.dita"/>
  <topicref href="afterPrep.dita"/>
</topicref>
<!-- ... -->
<topicref href="astroTasks.dita">
  <topicref href="astroOverview.dita"/>
  <topicref href="astroChecklist.dita"/>
  <topicref href="otherPreparation.dita"/>
  <topicref href="astroConclusion.dita"/>
</topicref>
```

## <keydef>

A key definition provides a simple way to define a key without making the definition itself a part of rendered content.

### Usage information

The `<keydef>` element is a convenience element. It is equivalent to a `<topicref>` element that defines a key while also setting `@processing-role` to "resource-only".

Attributes defaulted on the `<keydef>` element ensure that key definitions do not appear in tables of contents, do not add extra links, and are not rendered as topics.

### Specialization hierarchy

The `<keydef>` element is specialized from `<topicref>`. It is defined in the `mapgroup-domain` module.

### Attributes

The following attributes are available on this element: [universal attributes](#), [link-relationship attributes](#), [common map attributes](#), `@keyref`, and the attributes defined below.

For this element:

- The `@keys` attribute is required.
- The `@href` attribute might be omitted when the key definition is used for variable text.
- The `@processing-role` attribute has a default value of "resource-only".

### Example

The following code sample shows several different types of key definitions:

```
<map>
  <title>Possible keys for use in the DITA specification</title>
  <!-- Key definition #1-->
  <keydef keys="dita-tc" scope="external" format="html"
    href="https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=dita">
```

```

    <topicmeta>
      <keytext>DITA Technical Committee</keytext>
    </topicmeta>
  </keydef>
  <!-- Key definition #2-->
  <keydef keys="addressing" href="dita-addressing.dita"/>
  <!-- Key definition #3-->
  <keydef keys="dita-version">
    <topicmeta>
      <keytext>2.0</keytext>
    </topicmeta>
  </keydef>
</map>

```

1. The first `<keydef>` element defines a key that links to a web page. It contains link text; it also specifies the necessary `@scope` and `@format` attributes, so that authors do not need to include them when they reference this key.
2. The second `<keydef>` element defines a key for a local DITA topic about addressing in DITA; that topic is available to resolve link text.
3. The third `<keydef>` element defines a text-only key that specifies the current DITA version number.

## <mapref>

A map reference is a mechanism for referencing a DITA map from a DITA map.

## Usage information

The `<mapref>` element is a convenience element. It is equivalent to a `<topicref>` element with the `@format` attribute set to "ditamap".

## Processing expectations

The hierarchy of the referenced map is merged into the container map at the position of the reference, and the relationship tables of the child map are added to the parent map.

### Comment by Kristen J Eberlein on 06 August 2018

Should we make normative statements about the above points?

Discussed at DITA TC meeting on 02 July 2019.

Consensus: Yes, these need to be normative statements, but this is not the correct location. We need architectural topics that address processing of map hierarchy and relationship tables. This content should be located in the same chapter as the material about subjectScheme maps; it probably should absorb content currently in Processing > Navigation > TOC.

We should plan a review that covers both architectural and element-reference topics about maps.

### Comment by Kristen J Eberlein on 05 September 2020

The terms "container map" and "parent map" here are unclear. How are they different?

## Specialization hierarchy

The `<mapref>` element is specialized from `<topicref>`. It is defined in the map-group module.

## Attributes

The following attributes are available on this element: [universal attributes](#), [link-relationship attributes](#), [common map attributes](#), [@keyref](#), and [@keys](#).

For this element, the `@format` attribute has a default value of "ditamap".

## Examples

The following section shows how `<mapref>` elements can be used to reference submaps.

### Figure 14: DITA map for all element-reference topics (`base-elements.ditamap`)

The `base-elements.ditamap` document references the `map-group-elements.ditamap`:

```
<map>
  <title>Base elements</title>
  <!-- ... -->
  <topicref href="containers/domain-elements.dita" >
    <!-- ... -->
    <mapref href="map-group-elements.ditamap" />
    <!-- ... -->
  </topicref>
  <!-- ... -->
</map>
```

### Figure 15: DITA map for map-group elements (`map-group-elements.ditamap`)

The `map-group-elements.ditamap` document contains references to the element-reference topics for the map group domain. It is constructed as a map in order to enable easy editing of the child topics.

```
<map>
  <title>Map group elements</title>
  <topicref keyref="mapgroup-d" >
    <topicref keyref="anchorref" />
    <topicref keyref="keydef" />
    <topicref keyref="mapref" />
    <topicref keyref="topicgroup" />
    <topicref keyref="topichead" />
  </topicref>
</map>
```

### Figure 16: Result after processing `base-elements.ditamap`

After processing, the `base-elements.ditamap` contains the topic references that originally were located in the submap:

```
<map>
  <title>Base elements</title>
  <!-- ... -->
  <topicref href="containers/domain-elements.dita" >
    <!-- ... -->
    <topicref keyref="mapgroup-d" >
      <topicref keyref="anchorref" />
      <topicref keyref="keydef" />
      <topicref keyref="mapref" />
      <topicref keyref="topicgroup" />
      <topicref keyref="topichead" />
    </topicref>
    <!-- ... -->
  </topicref>
  <!-- ... -->
</map>
```



## <mapresources>

The <mapresources> element serves as an intuitive location for map authors to place objects with @processing-role set to "resource-only", for example, key definitions and subject scheme maps.

### Comment by Kristen J Eberlein on 10 November 2021

OK, how about the following as an attempt to use natural language?

Map resources are objects with a @processing-role set to "resource-only", for example, key definitions and subject scheme maps. Such resources do not contribute to the navigation structure, although they might be essential for correct authoring and processing.

## Specialization hierarchy

The <mapresources> element is specialized from <topicref>. It is defined in the map group module.

## Attributes

The following attributes are available on this element: [universal attributes](#), [link-relationship attributes](#), [common map attributes](#) (excluding @chunk and @collection-type), @keys, @keyref, and the attributes defined below.

For this element, the @processing-role attribute has a default value of "resource-only".

## Examples

This section provides examples of how the <mapresources> element can be used.

### Figure 17: Specifying resource-only objects in an intuitive location in a book map

The following code sample illustrate how the <mapresources> element can group references to key definitions, subject schemes, and other resources in a bookmap:

```
<bookmap>
  <booktitle>
    <mainbooktitle>Test bookmap</mainbooktitle>
  </booktitle>
  <mapresources>
    <mapref href="key-definitions.ditamap"/>
    <mapref href="subject-scheme.ditamap" type="subjectscheme"/>
    <topicref href="cover-page.dita outputclass="cover-page"/>
  </mapresources>
  ...
</bookmap>
```

Note that this example illustrates that <mapresources> can be used to make topics available for resource-only processing. In this scenario, the company uses a processor that uses content contained in the cover-page.dita file to generate a PDF cover page.

### Figure 18: Specifying resource-only objects in a map

The following code sample shows a map that contains information for a specific model of a controller. This map is referenced in an omnibus publication that contains information for an entire family of controllers.

```
<map keyscope="model-XNP09">
  <title>Model XNP09</title>
  <mapresources>
    <keydef keys="model-illustration" href="model-XNP09.png" format="png"/>
```

```

<keydef keys="remove-cover" href="remove-cover-XNP09.png" format="png"/>
</mapresources>
<topicref href="model-overview.dita"/>
<topicref href="installing.dita"/>
<topicref href="uninstalling.dita"/>
...
</map>

```

## <topicgroup>

A topic group is a set of topic references that share common attributes and linking relationships.

### Usage information

The <topicgroup> element does not affect the navigation hierarchy of the map.

Most <titlealt> elements within the <topicmeta> element inside of a <topicgroup> have no effect on rendered publications, but can be used to hold descriptive information about the grouped <topicref> elements.

### Rendering expectations

When a map that contains a <topicgroup> element with a navigation title is used to generate publication output, processors *MUST* ignore the navigation title and *MAY* issue an error message.

### Specialization hierarchy

The <topicgroup> element is specialized from <topicref>. It is defined in the map-group domain.

### Attributes

The following attributes are available on this element: [universal attributes](#), [common map attributes](#), [@format](#), [@scope](#), and [@type](#).

### Example

In the following code sample, the <topicgroup> element specifies common attributes (@audience and @linking) that are inherited by the topic references. The navigation hierarchy is not affected.

```

<topicgroup audience="novice" linking="none">
  <topicmeta>
    <titlehint>Topics used only in "Getting started" material.</titlehint>
  </topicmeta>
  <topicref href="getting-started.dita"/>
  <topicref href="basic-concepts.dita"/>
  <topicref href="cheat-sheet-reference.dita"/>
</topicgroup>

```

## <topichead>

A topic head is a title-only entry in a DITA map.

### Rendering expectations

The content of the <titlealt> element with a @title-role of <navigation>, such as <navtitle>, appears as a heading when the map is rendered as a table of contents. In print contexts, it also appears as a heading in the rendered content.

## Processing expectations

Processors *SHOULD* generate a warning if a navigation title is not specified on a `<topichead>` element.

## Specialization hierarchy

The `<topichead>` element is specialized from `<topicref>`. It is defined in the map-group module.

## Attributes

The following attributes are available on this element: [universal attributes](#), [common map attributes](#), [@format](#), [@scope](#), and [@type](#).

## Example

In the following example, the `<topichead>` elements provide titles ("Computers" and "Books") for two groups of topics.

```
<map>
  <topichead>
    <topicmeta>
      <navtitle>Computers</navtitle>
    </topicmeta>
    <topicref href="eniac.dita"/>
    <topicref href="system360.dita"/>
    <topicref href="pdp8.dita"/>
  </topichead>
  <topichead>
    <topicmeta>
      <navtitle>Books</navtitle>
    </topicmeta>
    <topicref href="hardback.dita"/>
    <topicref href="paperback.dita"/>
  </topichead>
</map>
```

# Index

## C

- context-sensitive help [14](#)
- convenience elements
  - `<keydef>` [22](#)
  - map group domain [20](#)
  - `<mapref>` [23](#)
  - `<mapresources>` [25](#)

## D

- domains
  - emphasis [16](#)
  - highlighting [17](#)
  - mapgroup [20](#)

## E

- element groups
  - basic map [3](#)
- elements
  - basic map
    - `<anchor>` [3](#)
    - `<keytext>` [4](#)
    - `<map>` [5](#)
    - `<navref>` [6](#)
    - `<relcell>` [7](#)
    - `<relcolspec>` [8](#)
    - `<relheader>` [10](#)
    - `<relrow>` [10](#)
    - `<reltable>` [10](#)
    - `<topicmeta>` [13](#)
    - `<topicref>` [12](#)
    - `<ux-window>` [14](#)
- emphasis domain
  - `<em>` [16](#)
  - `<strong>` [16](#)
- examples
  - maps
    - audience definition [13](#)
    - relationship tables [8, 10](#)

## G

- grouping elements
  - `<topicgroup>` [26](#)

## H

- highlighting domain
  - `<b>` [17](#)

- highlighting domain (*continued*)

- `<i>` [17](#)
- `<line-through>` [18](#)
- `<overline>` [18](#)
- `<sub>` [18](#)
- `<sup>` [19](#)
- `<tt>` [19](#)
- `<u>` [20](#)

## K

- keys
  - definition
  - examples [22](#)

## M

- mapgroup domain
  - `<anchorref>` [20](#)
  - `<keydef>` [22](#)
  - `<mapref>` [23](#)
  - `<mapresources>` [25](#)
  - `<topicgroup>` [26](#)
  - `<topichead>` [26](#)
- maps
  - examples [5](#)
    - audience definition [13](#)
    - key definition [22](#)
    - relationship tables [8, 10](#)
  - metadata [13](#)
  - overview [5](#)
- messages issued by processors
  - `<navtitle>` within `<topicgroup>` [26](#)
  - `<topichead>` with no navigation title [26](#)
- metadata
  - maps [13](#)

## P

- processing expectations
  - labels for related links [8](#)

## R

- relationship tables
  - cells [7](#)
  - column definitions [8](#)
  - examples [8, 10](#)
  - headers [10](#)
  - labels for related links [8](#)
  - overview [10](#)
  - processing expectations [8](#)

relationship tables (*continued*)

rows [10](#)

rendering expectations

`<navtitle> within <topicgroup>` [26](#)

`<topicgroup>` [26](#)

## T

topics

groups [26](#)

## U

user assistance [14](#)