

**DITA 1.3 proposed feature 13092**

# Contents

**DITA 1.3 proposed feature 13092..... 3**

# DITA 1.3 proposed feature 13092

---

Allow `<ph>` within `<indexterm>`.

## Date and version information

Include the following information:

- Date completed: 6 May 2013
- Champion: Eliot Kimber
- Email: <https://lists.oasis-open.org/archives/dita/201108/msg00010.html>

## Original requirement

Without `<ph>` it is impossible to represent things like superscripts and subscripts in index entries (because `<sub>` and `<sup>` are specializations of `<ph>`). There is no reason to disallow `<ph>` within `indexterm`.

## Use cases

Use cases include:

- Subscripts and subscripts within index entries
- Any other typographic or semantic markup specialized from `<ph>` that might be need to be used in an index entry.

## Benefits

This change ensures that the content model of `<indexterm>` is not over constrained, allowing representation of any phrase-level typographic or semantic distinction that might be needed.

## Costs

- Maintainers of the DTDs and XSDs:
  - Add `<ph>` the content model of `<indexterm>`, `<index-see>`, and `<index-see-also>`.
- Editors of the DITA specification:
  - How many new topics will be required?
    - No new topics
  - How many existing topics will need to be edited?
    - The generated content model description for `<indexterm>`, `<index-see>`, and `<index-see-also>` will reflect the addition of `<xref>`.
  - Will the feature require substantial changes to the information architecture of the DITA specification?
    - The feature does not represent a substantial change to the architecture.
- Vendors of tools:
  - XML editors will need to allow editing of the updated content models.
  - Processors may need to provide additional rules for handling `<ph>` within index terms. In particular, they cannot treat index terms as simple strings. (But note that `<indexterm>` already allowed `<keyword>` and `<term>`, so there has always been this requirement for index entries.)
- DITA community-at-large. Will this feature add to the perception that DITA is becoming too complex?
  - This change should not add to the perception of complexity for authors.

## Technical requirements

### DTD declarations

Modify `commonElements.mod` to add "%ph;" to the content model for `<indexterm>`:

```
<!--                                LONG NAME: Index Term                                -->
<!ENTITY % indexterm.content
        "(%words.cnt; |
         %ph; |
         %indexterm; |
         %index-base;)*"
>
```

Modify `indexingDomain.dom` to add "%ph;" to the content models for `<index-see>` and `<index-see-also>`:

```
<!--                                LONG NAME: Index See                                -->
<!ENTITY % index-see.content
        "(%words.cnt; |
         %ph; |
         %indexterm;)*"
>
<!ENTITY % index-see.attributes
        "keyref
         CDATA
         #IMPLIED
         %univ-atts;"
>
<!ELEMENT index-see      %index-see.content;>
<!ATTLIST index-see      %index-see.attributes;>

<!--                                LONG NAME: Index See Also                            -->
<!ENTITY % index-see-also.content
        "(%words.cnt; |
         %ph; |
         %indexterm;)*"
>
```

### RNG declarations

Modify `commonElements.mod.rng` to add "ph" to the content model for `<indexterm>`:

```
...
<define name="indexterm.content">
  <zeroOrMore>
    <choice>
      <ref name="words.cnt"/>
      <ref name="ph"/>
      <ref name="indexterm"/>
      <ref name="index-base"/>
    </choice>
  </zeroOrMore>
</define>
...
```

Modify `indexingDomain.mod.rng` to add `<ph>` to the content models for `<index-see>` and `<index-see-also>`:

```
...
<define name="index-see.content">
  <zeroOrMore>
    <choice>
      <ref name="words.cnt"/>
```

```

        <ref name="ph"/>
      <ref name="indexterm"/>
    </choice>
  </zeroOrMore>
</define>
...
<define name="index-see-also.content">
  <zeroOrMore>
    <choice>
      <ref name="words.cnt"/>
      <ref name="ph"/>
      <ref name="indexterm"/>
    </choice>
  </zeroOrMore>
</define>
...

```

### XSD Declarations

Modify commonElementMod.xsd to add <ph> to the content model for <indexterm>:

```

...
<xs:group name="indexterm.content">
  <xs:sequence>
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:group ref="words.cnt" minOccurs="0"/>
      <xs:group ref="ph" minOccurs="0"/>
      <xs:group ref="indexterm" minOccurs="0"/>
      <xs:group ref="index-base" minOccurs="0"/>
    </xs:choice>
  </xs:sequence>
</xs:group>
...

```

Modify indexingDomain.xsd to add <ph> to the content models for <index-see> and <index-see-also>:

```

...
<xs:group name="index-see.content">
  <xs:sequence>
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:group ref="words.cnt"/>
      <xs:group ref="ph" minOccurs="0"/>
      <xs:group ref="indexterm"/>
    </xs:choice>
  </xs:sequence>
</xs:group>
...
<xs:group name="index-sort-also.content">
  <xs:sequence>
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:group ref="words.cnt"/>
      <xs:group ref="ph" minOccurs="0"/>
      <xs:group ref="indexterm"/>
    </xs:choice>
  </xs:sequence>
</xs:group>
...

```

## Examples

Index term with various typographic elements:

```
<p>Many people consider Euler's identity,
<i>e</i><sup><i>#x03c0</i></i><i>i</i></sup>+1=0,
<indexterm><i>e</i><sup><i>#x03c0</i></i><i>i</i></sup>+1=0 <index-see>Euler's
identity</index-see> </indexterm> <indexterm>Euler's identity</indexterm>
<indexterm>2.71828
  <index-see><i>e</i> (Euler's number)</index-see>
</indexterm>
to be the most beautiful equation in mathematics.</p>
```