

Stage three

Contents

Stage three: #647 Replace classification domain with a new attribute.....	3
The @subjectrefs attribute.....	10
Example: A subject scheme map used to define taxonomic subjects.....	12
Example: How attributes cascade from one map to another	14
Index.....	16

Stage three: #647 Replace classification domain with a new attribute

Replace classification domain with a new attribute: @subjectrefs

Champion

Kristen James Eberlein, Eberlein Consulting LLC

Tracking information: Stage two

Event	Date	Links
Initial suggestion	E-mail, 09 December 2021	Minutes, 04 January 2022
Stage 1 proposal accepted	11 January 2022	Minutes, 11 January 2022 GitHub issue
Stage 2 proposal submitted to TC for early feedback (not applicable to all proposals)	Not applicable	
Stage 2 proposal submitted to reviewers	E-mail, 07 March 2022	Carsten Brennecke, SAP Gershon Joseph, Precision Content I've also requested feedback from the following members of the larger DITA community: <ul style="list-style-type: none"> • Peyton Bentley, Kaplan • Joe Gelb, Zoomin Software • Pam Noreault, Ellucian • Shane Taylor, Cengage Group
Stage 2 proposal submitted to TC	E-mail, 17 March 2022 E-mail, 29 March 2022	
Stage 2 proposal discussed by TC	22 March 2022 29 March 2022	Minutes, 22 March 2022 Minutes, 29 March 2022
Stage 2 proposal approved by TC	05 April 2022	Minutes, 05 April 2022
Stage 2 proposal reopened	24 May 2022 The proposal was reopened to allow @subjectrefs on <map>.	
Revised stage 2 proposal submitted to reviewer		Not applicable
Revised stage 2 proposal submitted to TC	E-mail, 26 May 2022 E-mail, 29 May 2022	

Event	Date	Links
Revised stage 2 proposal approved	31 May 2022	Add link to "Minutes, 31 May 2022"

Tracking information: Stage three

Event	Date	Links or notes
Stage 3 proposal submitted to reviewers	E-mail, 04 June 2022	Robert Anderson, Oracle Carsten Brennecke, SAP Gershon Joseph, Precision Content Eliot Kimber, ServiceNow
Stage 3 proposal submitted to TC	<xref to e-mail sent to list>	
Stage 3 proposal discussed	<Date>	<xref to meeting minutes where discussed>
Stage 3 proposal approved	<Date>	<xref to meeting minutes where discussed>

Approved technical requirements

Remove the classification domain and the classification map. Add a new attribute, @subjectrefs, to the <topicref> element and specializations of <topicref>, as well as <map> and specializations of <map>. This includes both elements in the base and the technical content edition.



Note: The @subjectrefs attribute has no meaning if it is specified on a key definition^[1] that does not reference a resource.

¹Eliot Kimber: Why is this key definition--I think this applies to any topicref that does not specify @"href" and does not have any link text.

Kristen James Eberlein: Because that's what we approved at the stage two level. And we would want @"subjectrefs to have meaning if it is applied to a <topicgroup> elements -- that use case eliminates the wording that you suggest. FYI, this wording was explicitly discussed at the TC meeting on 29 March; you were present and agreed to it. I do know that we don't always catch everything at first ... Check out the minutes to the TC meetings referenced in the "Tracking information" section. Do you want to reopen the stage two proposal so that we can modify the "Proposed solution"? I've had to reopen several stage 2 proposals that I championed for this sort of thing: * #36: Remove deprecated items * #316: Diagnostics element * This proposal * Probably more Maybe we should allow the stage three proposals to state "This is what was approved at stage two, and here is a revised version for stage three.

Eliot Kimber: If our process doesn't allow for changes in the Stage 3 proposal, I think that's a process flaw--the Stage 3 proposal is where we see the final language in its specification context. I like a "we changed this from Stage 2" annotation as a requirement--that makes the change fully transparent and traceable.

Kristen James Eberlein: I've sent an e-mail to the TC today (13 June 2022) about this.

Eliot Kimber: To handle the topicgroup with descendants case then the statement would to be something like A topic reference that does not reference a resource and has no descendant topic references.

The @subjectrefs attribute will have the following characteristics:

- It will cascade.
- It will take multiple values, separated by white space. The values are one or more keys, defined in a subject scheme map.

While the DITA TC does not want to introduce any processing expectations for @subjectrefs in the DITA 2.0 time frame, it is possible that we will introduce them in the future.

Dependencies or interrelated proposals

None.

Removed grammar files

The following grammar files need to be removed. In addition, the catalog files that currently reference the following files will need to be modified.

Base

- doctypes/dtd/subjectScheme/classifyDomain.ent
- doctypes/dtd/subjectScheme/classifyDomain.mod
- doctypes/rng/subjectScheme/classifyDomain.rng

Technical content

- doctypes/dtd/classificationMap directory
- doctypes/rng/classificationMap directory

The classification domain and the document-type shell for a base classification map will be loaded into the GitHub repository for specializations that are removed from the DITA standard.

Modified grammar files

This proposal will require modifications to the following files:

Base edition

- dtd/base/map.mod
- dtd/base/mapGroup.mod
- rng/base/map.rng
- rng/base/mapGroupDomain.rng

Technical content edition

- dtd/technicalContent/glossrefDomain.mod
- rng/technicalContent/glossrefDomain.mod

No changes are required to the bookmap grammar files, since the relevant attribute definitions reference the topicref attribute entities that are defined in the base grammar files.

Robert D Anderson: I think I'm starting to favor removing this note entirely - if we are not defining "meaning" for any other case, we do not need to define "absence of meaning" either. That can come later, if/when we define the actual meaning for the attributes.

Kristen James Eberlein: Hopefully the TC will agree today (14 June 2022) that we do not need to reopen the stage two proposal. I also think that removing the note is the simplest and best approach.

In the content below, the following conventions are used:

- Bold is used to indicate code to be added, for example, **addition**.
- Line-through and red text is used to indicate code to be removed, for example, ~~removal~~.
- Ellipses (...) indicate where code is snipped for brevity.

```

<!ENTITY % topicref-atts
    "...
        subjectrefs
            CDATA
                #IMPLIED"
>
...
<!ENTITY % topicref-atts-without-format
    "...
        subjectrefs
            CDATA
                #IMPLIED"
>

```

Figure 1: Changes to map.mod

```

<!ENTITY % keydef.attributes
    "...
        subjectrefs
            CDATA
                #IMPLIED
                %univ-atts;"
>

```

Figure 2: Changes to mapGroup.mod

```

<div>
  <a:documentation>COMMON ATTRIBUTE SETS</a:documentation>

  <define name="topicref-atts">
    ...
    <optional>
      <attribute name="subjectrefs"/>
    </optional>
  </define>
  ...
  <define name="topicref-atts-without-format">
    ...
    <optional>
      <attribute name="subjectrefs"/>
    </optional>
  </define>

```

Figure 3: Changes to mapMod.rng

```

<div>
  <a:documentation>Key Definition</a:documentation>
  ...
  <define name="keydef.attributes">
    ...
    <optional>
      <attribute name="subjectrefs"/>
    </optional>
    <ref name="univ-atts"/>
  </define>

```

```
</define>
```

Figure 4: Changes to mapGroupDomain.rng

```
<!ENTITY % glossref.attributes
    "...
    subjectrefs
        CDATA
        #IMPLIED
    %univ-atts;"
>
```

Figure 5: Changes to glossrefDomain.mod

```
<div>
  <a:documentation>ELEMENT TYPE DECLARATIONS</a:documentation>
  <div>
    <a:documentation> LONG NAME: Glossary Reference </a:documentation>
    ...
    <define name="glossref.attributes">
      ...
      </optional>[2]
      <optional>
        <attribute name="subjectrefs"/>
      </optional>
      <ref name="univ-atts"/>
    </define>
```

Figure 6: Changes to glossrefDomain.rng

Modified terminology

Not applicable

Modified specification documentation

The following topics need to be removed or modified:

Topics and maps to be removed

- archSpec/base/classification-maps.dita
- langRef/classification-domain-elements.ditamap
- langRef/containers/classify-d.dita
- ~~<filepath>langRef/base/subjectCell.dita</filepath>[3]~~
- langRef/base/subjectref.dita
- langRef/base/subjectref.dita^[4]

²Gershon Joseph:

The end tag for the preceding optional element should not be marked in bold, since it's not new. It's part of the current, existing spec code.

Kristen James Eberlein:

I've corrected this in the DITA source. Thanks for noticing this.

³Kristen James Eberlein: [Deletion]

⁴Gershon Joseph: This file is listed twice.

- langRef/base/topicapply.dita
 - langRef/base/topicCell.dita
 - langRef/base/topicsubject.dita
 - langRef/base/topicSubjectHeader.dita
 - langRef/base/topicSubjectRow.dita
 - [langref/base/topicSubjectTable](#)
- [5]

Architectural topics and maps to be modified

File	Modification
cascading-in-a-ditamap.dita	Add @subjectrefs to the list of attributes that cascade.
cascading-of-attributes-from-map-to-map.dita	Add @subjectrefs to the list of attributes that cascade from map to map.
subjectSchema.dita	Remove the following sentence: "In conjunction with the classification domain, subject definitions can be used for retrieval and traversal of the content at run time when used with information viewing applications that provide such functionality."
subject-scheme-maps.ditamap	Remove reference to classification-maps.dita. Add references to new topics: <ul style="list-style-type: none"> • The subjectrefs attribute on page 10 • Example: A subject scheme map used to define taxonomic subjects on page 12
example-cascade-map-to-map-attributes.dita	Modify the content as shown in Example: How attributes cascade from one map to another on page 14. Modified content is marked with rev="proposal-647" and highlighted with

Kristen James Eberlein: Good catch. I checked this list against the topics in the spec, and also noticed that it did not include topicSubjectTable.

⁵Kristen James Eberlein: [Insertion]

File	Modification
	blue in the PDF for this proposal.

Element-reference topics and maps to be modified

The following attribute definition for `@subjectrefs` will need to be added to the appropriate attribute topic:

Specifies one or more subject keys that are defined in a subject scheme map. Multiple values are separated by white space.^[6]

Base edition

The following topics will need to include the `@subjectrefs` attribute:

- `<keydef>`
- `<map>`
- `<mapref>`
- `<topichead>`
- `<topicgroup>`
- `<topicref>`

This might happen automatically depending on what attribute "grouping" the `@subjectrefs` attribute is part of.

⁶Kristen James Eberlein: Should be "Specifies one or more keys that are defined by subject definitions in a subject scheme map." I think we want to avoid introducing new terminology such as "subject keys".

Robert D Anderson: Yes, it should

Eliot Kimber: I agree. Actually, now that I read this again, I think "that are defined in a subject scheme map" is too strong because it implies a requirement if they are not (either because the key is defined by not in a subject scheme map or because the key is not defined in any map). I think "expect to be defined in subject scheme maps". "in a subject scheme map" could be read as expecting all the referenced keys to be in a single map.

Kristen James Eberlein: Hmm ... We could change the wording to one of the following in order to eliminate your concern that we are implying that the subject definitions must be in a single subject scheme map: * "Specifies one or more keys that are defined by subject definitions in subject scheme maps" * Specifies one or more keys that are defined by subject definitions in one or more subject scheme maps" And, of course, we can state (do we really need to do so?) that a `@subjectrefs` attribute that references a key NOT defined by a subject definition in a subject scheme map is meaningless." I don't think we need to be concerned about the meaning of a `@subjectrefs` attribute referencing a key that is not defined. Surely that is covered by standard rules around key resolution and processing.

Robert D Anderson: I like a simple addition of something like "that are each defined" rather than the extra words of "one or more keys from one or more subject scheme maps"

Technical content edition The following topics will need to include the `@subjectrefs` attribute:

- `<glossref>`
- All bookmap elements that are defined with the `topicref-atts` entity

This might happen automatically depending on what attribute "grouping" the `@subjectrefs` attribute is part of.

Topics to be added

The following topics are to be added to the "DITA map processing" chapter, in the "Subject scheme maps and their usage" section:

- [The subjectrefs attribute](#) on page 10
- [Example: A subject scheme map used to define taxonomic subjects](#) on page 12

Migration plans for backwards incompatibilities

Implementations that use the classification domain and classification map will need to do one of the following, if they want to move to DITA 2.0:

- Download the classification domain from the GitHub repository and integrate it into the relevant document-type shells. If the implementation currently uses the OOB OASIS-provided document-type shell for the classification map, they will need to move to using a custom document-type shell.
- Replace map markup that uses the classification domain with the new `@subjectrefs` attribute, as well as modifying any processing that is based on the classification domain.

The `@subjectrefs` attribute

The `@subjectrefs` attribute specifies one or more keys that are defined by a subject definition in a subject scheme map. Multiple values are separated by white space.

The `@subjectrefs` attribute cascades. It can be used on a `<topicref>` element to associate the referenced topic^[8] with a subject defined in a subject scheme map.^[7]

⁸Eliot Kimber: `c/topic/resource/` I don't see a reason to limit the association scope, especially since we are not defining any processing implications for the association.

Kristen James Eberlein: This is handled by your previous comment; no changes to the source are required.

⁷Eliot Kimber: Change to: When specified on a topic reference, associates the referenced resource with subjects defined in subject scheme maps. This is more active, removes the limitation to topics, and reflects the potential for multiple values on `@subjectrefs`

Kristen James Eberlein: Good thinking. I've changed the sentence to read as follows, which reflects a slight editorial change from your suggestion: "When specified on a topic reference, the

The @subjectrefs attribute has no meaning if it is specified on a key definition that does not reference a resource.^[9]

[10]

@subjectrefs attribute associates the referenced resource with subjects that are defined in subject scheme maps."

⁹Eliot Kimber: I don't think this statement is necessary because either it's not true (if we're not defining any processing expectations we can't say what is and isn't meaningful) or because it's trivially obvious. But in either case it has no actionable effect relative to our processing expectations (because there aren't any). If this statement is still necessary it should refer to topic references, not key definitions, since being a key definition has no bearing on the use or non-use of @subjectrefs--unspecialized topicrefs can specify @subjectref and not reference any resource.

Kristen James Eberlein: Thinking about this ... 1. Perhaps we need to back track from our original stated position of not introducing "any processing expectations for @"subjectrefs in the DITA 2.0 time frame." Maybe we need to introduce a limited set of processing expectations, maybe we need to introduce a full set of processing expectations. Note: Changes about the wording that I quoted require reopening the stage two proposal. 2. We explicitly called out key definitions, since TC members had concerns around that. We do not want to state that @subjectrefs does not have a meaning when it is specified on a topic reference that does not specify a resource, since that would rule out the useful use case of specifying @subjectrefs on the <topicgroup> element. (That might be something that we want to show explicitly in an example ...)

Eliot Kimber: See my comment that was against the stage two proposal. By adding the qualification "and that has no descendant topic references" addresses the topic group concern. But yes, I am explicitly suggesting that we need to have processing expectations specifically for the allowed (but not required or even encouraged) resolution of key names specified in @subjectrefs. I don't think we need anything beyond that.

¹⁰Eliot Kimber: Per my mail to the list, this topic currently says nothing about what processors or required, allowed, or encouraged to do with the keys specified on @subjectrefs. Without that it's ambiguous as to whether treating them as normal key references is or isn't required and what the processor behavior should or may be when a subject keyref cannot be resolved. While we don't want to specify processing expectations with respect to what it *means* to have a subject associated with a topicref I think we have to specify what the key resolution expectations and requirements are since address processing is independent of the meaning applied to a resolved subject reference. If we say that the value of @subjectrefs is zero or more key names then I think we are obligated to say what the key resolution requirements are. Given that, I think there should be a processing expectations section something like: Processing Expectations Processors MAY attempt to resolve the key names in @subjectref using the same key resolution rules as for @keyref. Processors MAY report unresolvable subject references as warnings or errors. Beyond the option of attempting to resolve key references in @subjectrefs, there are no processing expectations for either the values of @subjectrefs or any subjects addressed by @subjectrefs.

Kristen James Eberlein: If we are going to make a normative statement about our expectations for @subjectrefs and key resolution, then MAY is too weak. That effectively would make it impossible to specify ANY additional processing expectations for @subjectrefs in future 2.x releases.

Eliot Kimber: I don't think I agree about MAY limiting our options. Because subject scheme maps are already defined to be useful outside the context of being used by direct reference there's an implicit expectation that the specification of a subject's key name need never be treated as a literal key reference but simply utterance of a name of a subject resolved in some non-DITA-defined way.

Example: A subject scheme map used to define taxonomic subjects

A subject scheme map can be used to define taxonomic subjects. Once defined, the subjects can be referenced by applying^[11] a @subjectrefs attribute to^[12] a <topicref> element.

The following subject scheme map defines a set of subjects that are used to classify content:

```
<subjectScheme[14]>
```

That is, it's already the case that subject schemes can be used as configuration files by processors in a way that is suggested but not required by the original subject scheme specification. Given that, it would inappropriate to ever make subject key resolution a MUST. So if we decided in a later update to impose some meaning to @subjectrefs we could go from MAY to SHOULD for the resolution of key names in @subjectrefs but we could never go from SHOULD to MUST. We could hedge our bets by starting at SHOULD but I don't see how that makes a material difference in what users will actually experience in products: either products will resolve keys in @subjectrefs or they won't and both choices are conforming if we say MAY or SHOULD. At the end of the day what processors actually do will be driven by user requirements (or what contributors step up to implement). In the future any processing expectations we want to add about the *meaning* of associated subjects will have to be MAY or SHOULD, both for the reasons I've outlined and because anything in this realm is processor-specific so can only ever be a processing or rendering suggestion. So the only question for future meanings can be MAY or SHOULD, not SHOULD or MUST. Note that a separate standard for the *application* of subject schemes to DITA content could impose MUST requirements because it would be a separate specification that you choose to adopt or not adopt, but DITA abdicated that ability at the start with the way subject scheme processing was originally defined.

Kristen James Eberlein: We do not include "Processing expectations" sections in architectural topics, although of course we can include normative statements in architectural topics. We've stated that the value of @subjectrefs is one or more keys. The spec has clearly defined expectations for key resolution, all of which apply here. Therefore, I do not think we need to make any statements about @subjectrefs and key resolution.

¹¹Eliot Kimber: using

¹²Eliot Kimber: on

¹⁴Eliot Kimber:

I just realized in the course of putting together the examples below that <subjectScheme> and <subjectdef> do not allow the @keyscope attribute.

I think that's a bug and we should allow it in 2.0 so that a subjectScheme map that is intended to be used as a normal submap can choose to always be in a scope. Likewise, any subjectdef should be able to establish a key scope for itself and its subjects.

Kristen James Eberlein:

This was an explicit decision made in the DITA 1.3 time frame. Robert Anderson and Chris Nitchie felt strongly about this, and I deferred to their concerns. (I had previously run into difficulties with using subject scheme to define controlled values that would have been eliminated if I could have used key scopes ...)

```

<subjectdef keys="content-types">
  <subjectdef keys="conceptual-material"/>
  <subjectdef keys="reference"/>
  <subjectdef keys="tutorial"/>
</subjectdef>
<subjectdef keys="operating-systems">

```

Nonetheless, <subjectScheme> and <subjectdef> do not allow @keyscope, and so this proposal cannot include examples of markup that would allow @keyscope in such places.

We cannot treat the fact that <subjectScheme> and <subjectdef> do not allow the @keyscope attribute as a bug, since that was an explicit decision on part of the TC.

Allowing @keyscope on these elements would require a new DITA 2.0 proposal. Send an e-mail to the TC if that is something that you want to champion.

Eliot Kimber:

If it was a considered decision to disallow keyscope on subjectScheme I guess we're stuck with that.

However, it is still the case that references to subjectSchemes may specify @keyscope so the case has to be addressed at least through the sort of examples I provided.

If a subjectScheme map is never referenced from another map using DITA-defined map referencing facilities then it may or not be in an imposed key scope because the implications in that case are entirely processor specific (for example, I could imagine a runtime parameter of "key scope for referenced subject schemes" or a way to associate a key scope with configured subject schemes in some tool-specific configuration mechanism).

But if subject scheme maps are referenced using DITA-defined map reference features then they may be associated with a key scope in the context of the referencing map if if the subject scheme map itself cannot define its default key scope.

This also highlights what we already know, that using keys for subject schemes as definitions of subjects is problematic at best.

But as long subject scheme maps are specializations of <map> they must participate in normal map processing and semantics and that includes key scopes.

DITA users can certainly choose to treat subjects scheme maps as totally separate things that are never literally included in other maps, but only used as standalone definitions of subjects used in some processor-specific way. But that is not the only way to use them.

Kristen James Eberlein:

Eliot, I don't want to include a second example topic if the only reason is to address the use of key scopes.

My rationale:

- * We want our example topic here in this chapter of the spec to only cover the most common usage.
- * If we want an example tht addresses key scopes and @"subjectref, that example should be located in the "DITA addressing" chapter".

```

<subjectdef keys="linux"/>
<subjectdef keys="macosx"/>
<subjectdef keys="windows"/>
</subjectdef>
<subjectdef keys="user-tasks">
  <subjectdef keys="administering"/>
  <subjectdef keys="developing"/>
  <subjectdef keys="installing"/>
  <subjectdef keys="troubleshooting"/>
</subjectdef>
</subjectScheme>

```

The keys assigned to the subject definitions can be referenced by specifying the @subjectrefs attribute on topic references in a navigation map:

```

<map>
<title>User assistance for the Acme Widget</title>
<!-- ... -->
<topicref keyref="install-overview" subjectrefs="installing">
  <topicref keyref="install-linux"/>
  <topicref keyref="install-macosx"/>
  <topicref keyref="install-windows"/>
  <topicref keyref="install-troubleshooting" subjectrefs="troubleshooting"/>
</topicref>
<!-- ... -->
</map>

```

Because the @subjectrefs attribute cascades, the effective value of the above markup is the same as the following markup:

```

<map>
<title>User assistance for the Acme Widget</title>
<!-- ... -->
<topicref keyref="install-overview" subjectrefs="installing">
  <topicref keyref="install-linux" subjectrefs="installing"/>
  <topicref keyref="install-macosx" subjectrefs="installing"/>
  <topicref keyref="install-windows" subjectrefs="installing"/>
  <topicref keyref="install-troubleshooting" subjectrefs="installing
troubleshooting"/>
</topicref>
<!-- ... -->
</map>

```

Example: How attributes cascade from one map to another

In this scenario, attributes in one map cascade to a nested map.

Assume the following references in test.ditamap:

```

<map>
  <topicref href="a.ditamap" format="ditamap" toc="no"/>
  <mapref href="b.ditamap" audience="developer"/>
  <mapref href="c.ditamap#branch2" platform="myPlatform"/>

```

```
<mapref> [15]href="d.ditamap" subjectrefs="puzzles"/>
</map>
```

- The map `a.ditamap` is treated as if `toc="no"` is specified on the root `<map>` element. This means that the topics that are referenced by `a.ditamap` do not appear in the navigation generated by `test.ditamap`, except for branches within the map that explicitly set `toc="yes"`.
- The map `b.ditamap` is treated as if `audience="developer"` is set on the root `<map>` element. If the `@audience` attribute is already set on the root `<map>` element within `b.ditamap`, the value `developer` is added to any existing values.
- The element with `id="branch2"` within the map `c.ditamap` is treated as if `platform="myPlatform"` is specified on that element. If the `@platform` attribute is already specified on the element with `id="branch"`, the value `myPlatform` is added to existing values.
- The map `d.ditamap` is treated as if `subjectrefs="puzzles"` is set on the root `<map>` element. If the `@subjectrefs` attribute is already set on the root `<map>` element within `d.ditamap`, the value `puzzles` is added to any existing values.

¹⁵Robert D Anderson:

Typo, extra greater-than symbol

Kristen James Eberlein:

I've made that change in the proposal AND the source files for the branch.

Index

C

- cascading metadata
 - example
 - cascading attributes between maps [14](#)

E

- examples
 - cascading metadata
 - cascading attributes between maps [14](#)
 - subject scheme maps
 - defining taxonomic subjects [12](#)

S

- @subjectrefs
 - overview [10](#)