

DITA 1.3 proposed feature #13102, Stage 2

Contents

DITA 1.3 proposed feature #13102, Stage 2.....3

DITA 1.3 proposed feature #13102, Stage 2

Define a release management domain for DITA 1.3 that enables content workers to log comments and metadata at the topic level when changes are made. The model for this domain comes from bookmap's bookchangehistory, with modifications.

Date and version information

- The latest revision date for this proposed feature document is 2013 August 20.
- Tom Cihak is the champion of this proposal.
- The proposal was initially accepted for inclusion into DITA 1.3 on 2012 Feb 21: [Minutes from 2012 Feb 21](#).

Original requirement or use case

As described in Seth Park's [email](#), organizations and content authors need to help customers locate significant changes in revisions of large documents. These content providers would benefit from the ability to record release notes at the topic level, rather than in separate topics or in CMS metadata.

Change information must be subject to the same applicability methods that affect other content. That is, change information must include @props attributes, which presupposes the use of elements in the DITA namespace.

This use case has a complication, however, that makes the use of select attributes, while required, insufficient by themselves: in some organizations, release notes belong in one and only one document. In other words, once the release note has appeared in a release document, it should never appear in another one.

This requirement makes using dates as a selection mechanism highly desirable, if not mandatory.

Use cases

Medical Device Industry Requirements

- product group responsible for the change -- which product is this
- regulatory relevance -- ISO, FDA, a specific IEEE requirement
- functional area requesting the change -- engineering, support
- localization required -- yes/no
- function responsible for validation
- identify who made the change (secure ID)
- time and date of the change to the second
- nature of the change -- draft comment, including if change is valid for each reuse of the content (metadata change)
- resolution of the change: accepted, rejected, and so on
- some way of marking the exact location of the change in the content of the topic

Proposed solution

Release notes, stored with date/time information in the changed topics, may be aggregated by an independent process to compile a release notes report based on information supplied in the change history, including—but not limited to—date and time.

Here is the recommended scenario:

Based on externally supplied parameters, date/time criteria are established. Though normally the last release date is used, the user is free to choose any date; thus, the system would work just as well for a cumulative scheme as for release-by-release aggregation.

The external process combs through the topics contained in the map, selecting those changehistory items that meet the chosen criteria. These items are aggregated into a DITA topic that the user may retrieve; the topic is then available for

editing or further processing. When the DITA topic is included in the map for publication, filtering will ensure that only those release items appropriate for the given map are included.

Benefits

- Content creators will benefit; they will find it easier to record release notes if they don't have to record them in a separate topic. Accuracy and compliance will improve as a result.
- Organizations will benefit from increased accuracy and detail. Their documents will be better and their customers happier. It will be easier for them to comply with any regulatory or other outside requirements for change tracking.
- Having release notes available in DITA metadata is itself a benefit. Any organization is free to process and use the release notes as it sees fit, and an important class of information is provided a stable location, free from the vagaries of content management systems and local practice.
- For topic based displays such as html, having release notes in the topic will facilitate a display such as is common in wiki pages, where the topic history is included in a separate tab accessible at any time.

Technical requirements

DTD and Schema modifications

Topic or map specialization

Release management domain

Domain

This domain is to be included in the DITA topic prolog.

Elements

- The proposed new elements are shown in the figures at the end of this section.
- No processing is requested at this time.
- The new elements will only be permissible in the topic prolog.
- The elements are translatable.

Attributes

None

Processing impact

- An external process, before or after publication, can collect release notes from the topics contained in a map. With time, it may be desirable to incorporate this collection into publication processing; however, this is not a requirement.
- There should be no impact on other processing features, since the elements of this domain will most likely be extracted and compiled into a separate report.
- There are no known edge cases.

Overall usability

- The domain adds some additional complexity to the DITA topic, although it is optional, and similar to the existing bookchangehistory domain.
- Adding release notes to the topic prolog may make the prolog much longer.

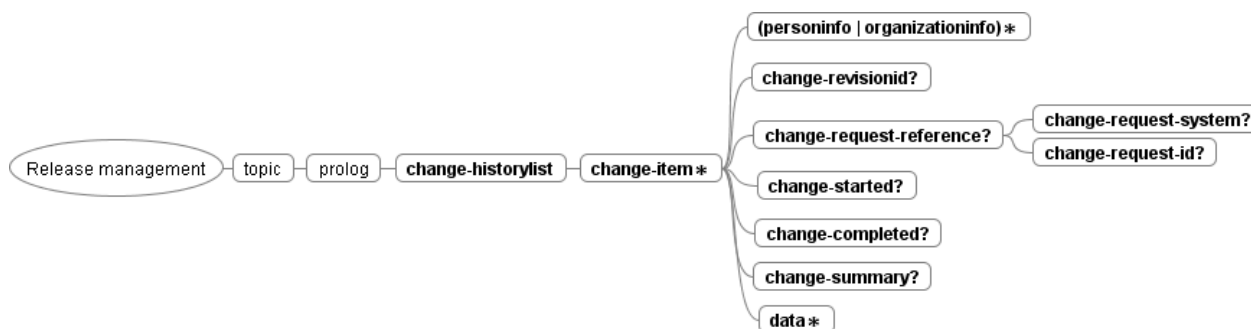


Figure 1: Release Management Elements

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- =====
      DITA Release Management Metadata Domain

      Defines element types for capturing change details within
      topics or maps.

      Depends on the XNAL domain.

      DITA 1.3

      Copyright (c) 2013 OASIS Open

      ===== -->

<!-- ===== -->
<!--                               Release Management DOMAIN ENTITIES -->
<!-- ===== -->

<!ENTITY % relmgmt-d-metadata
  "change-historylist
  "
>

<!ENTITY relmgmt-d-att
  "(topic relmgmt-d+xnal-d)"
>

<!-- ===== End Highlight Domain Entities ===== -->

```

Figure 2: releaseManagementDomain.ent

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- =====
      DITA Release Management Metadata Domain

      Defines element types for capturing change details within
      topics or maps.

      Depends on the XNAL domain.

      DITA 1.3

      Copyright (c) 2013 OASIS Open

      ===== -->

<!ENTITY % change-historylist           "change-historylist" >
<!ENTITY % change-item                 "change-item" >
<!ENTITY % change-revisionid           "change-revisionid" >
<!ENTITY % change-request-reference     "change-request-reference" >
<!ENTITY % change-started              "change-started" >
<!ENTITY % change-completed            "change-completed" >
<!ENTITY % change-summary              "change-summary" >
<!ENTITY % change-request-system        "change-request-system" >
<!ENTITY % change-request-id           "change-request-id" >

<!-- Long Name: Change History List -->

<!ENTITY % changehistory.data.atts

```

```

        '%univ-atts;
            datatype
                CDATA
                #IMPLIED
            outputclass
                CDATA
                #IMPLIED
        '>
<!ENTITY % change-historylist.content
    "(%change-item;)*"
>
<!ENTITY % change-historylist.attributes
    '%changehistory.data.atts;
    name
        CDATA
        "change-historylist"
    ,
>
<!ELEMENT change-historylist %change-historylist.content; >
<!ATTLIST change-historylist %change-historylist.attributes;>

<!-- Long Name: Change Item

    An individual release note.

-->
<!ENTITY % change-item.content
    "(%personinfo;|
        %organizationinfo;)*,
        %change-revisionid;?,
        %change-request-reference;?,
        %change-started;?,
        %change-completed;,
        %change-summary;?,
        %data;*)"
>
<!ENTITY % change-item.attributes
    '%changehistory.data.atts;
    name
        CDATA
        "change-item"
    ,
>
<!ELEMENT change-item %change-item.content; >
<!ATTLIST change-item %change-item.attributes; >

<!-- Long Name: Revision ID

    Specifies the revision ID to which the change applies.
    Revision IDs are normally specified within publication
    maps (e.g., Bookmap <revisionid> element).

-->
<!ENTITY % change-revisionid.content
    "(%data.cnt;)*"
>
<!ENTITY % change-revisionid.attributes
    '%changehistory.data.atts;
    name
        CDATA
        "change-revisionid"

```

```

'
>
<!ELEMENT change-revisionid %change-revisionid.content; >
<!ATTLIST change-revisionid %change-revisionid.attributes; >

<!-- Long Name: Change Request Reference

        Provides traceability to an external change
        request or other ticketing system;
-->
<!ENTITY % change-request-reference.content
        "(%change-request-system;?,
        %change-request-id;?)"
>
<!ENTITY % change-request-reference.attributes
        '%changehistory.data.atts;
        name
        CDATA
        "change-request-reference"
'
>
<!ELEMENT change-request-reference %change-request-reference.content; >
<!ATTLIST change-request-reference %change-request-reference.attributes; >

<!-- Long Name; Change Request System

        Some description of or identifier for the information system that
        manages or serves the referenced change request, for example,
        an issue tracking system.
-->
<!ENTITY % change-request-system.content
        "(%data.cnt;)*"
>
<!ENTITY % change-request-system.attributes
        '%changehistory.data.atts;
        name
        CDATA
        "change-request-system"
'
>
<!ELEMENT change-request-system %change-request-system.content; >
<!ATTLIST change-request-system %change-request-system.attributes; >

<!-- Long Name: Change request ID

        The identifier of the change request, such as an issue
        ID or ticket number.

-->
<!ENTITY % change-request-id.content
        "(%data.cnt;)*"
>
<!ENTITY % change-request-id.attributes
        '%changehistory.data.atts;
        name
        CDATA
        "change-request-id"
'
>
<!ELEMENT change-request-id %change-request-id.content; >
<!ATTLIST change-request-id %change-request-id.attributes; >

<!-- Long Name: Change started date

```

```

-->
<!ENTITY % change-started.content
  "(#PCDATA |
   text)*"
>
<!ENTITY % change-started.attributes
  '%changehistory.data.atts;
  name
    CDATA
      "change-started"
  ,
>
<!ELEMENT change-started %change-started.content; >
<!ATTLIST change-started %change-started.attributes; >

<!-- Long Name: Change completed date -->
<!ENTITY % change-completed.content
  "(#PCDATA |
   text)*"
>
<!ENTITY % change-completed.attributes
  '%changehistory.data.atts;
  name
    CDATA
      "change-completed"
  ,
>
<!ELEMENT change-completed %change-completed.content; >
<!ATTLIST change-completed %change-completed.attributes; >

<!-- Long Name: Change Summary

      The portion of the release note that will/may appear in a document

-->
<!ENTITY % change-summary.content
  "(%data.cnt;)*"
>
<!ENTITY % change-summary.attributes
  '%changehistory.data.atts;
  name
    CDATA
      "change-summary"
  ,
>
<!ELEMENT change-summary %change-summary.content; >
<!ATTLIST change-summary %change-summary.attributes; >

<!-- ===== -->
<!-- SPECIALIZATION ATTRIBUTE DECLARATIONS -->
<!-- ===== -->

<!ATTLIST change-historylist      %global-atts;
      class CDATA "- topic/metadata rm-d/change-historylist ">
<!ATTLIST change-item             %global-atts;
      class CDATA "- topic/data rm-d/change-item ">
<!ATTLIST change-revisionid       %global-atts;
      class CDATA "- topic/data rm-d/change-revisionid ">
<!ATTLIST change-request-reference %global-atts;
      class CDATA "- topic/data rm-d/change-request-reference ">
<!ATTLIST change-started          %global-atts;
      class CDATA "- topic/data rm-d/change-started ">
<!ATTLIST change-completed        %global-atts;

```



```

        class CDATA "- topic/data rm-d/change-completed ">
<!ATTLIST change-summary          %global-atts;
        class CDATA "- topic/data rm-d/change-summary ">
<!ATTLIST change-request-system    %global-atts;
        class CDATA "- topic/data rm-d/change-request-system ">
<!ATTLIST change-request-id        %global-atts;
        class CDATA "- topic/data rm-d/change-request-id ">

<!-- ===== End of Release Management domain ===== -->

```

Figure 3: releaseManagementDomain.mod

Costs

- Maintainers of the DTDs and XSDs will have to add a new domain; however, these additions are very similar to the existing bookmap's bookchangehistory domain, but simpler.
- Editors of the DITA specification will need to add topics for change-historylist and its children and edit one (prolog). Elements personinfo and organizationinfo are incorporated from the xnal domain.
- No changes are required to the DITA information architecture.
- XML editors, component content management systems, and processors, should be able to rely on DTDs to support the new elements, assuming the prolog is visible. CMS vendors should have no problems. No processing is being requested.
- For the DITA community-at-large, these elements should be easy to understand (comparable to the existing bookchangehistory domain); in any case, they are optional and can be easily ignored by those with no need for them.

Outline the impact (time and effort) of the feature on the following groups:

Examples

Example 1

This figure shows three simple release notes added to a single topic. This topic is used in documentation for two products, A and B.

```

<prolog>
...
  <changehistory-list>
    <change-item product="productA productB">
      <personinfo>Tom Cihak</personinfo>
      <change-completed>2012-12-21</change-completed>
      <change-summary>Made change 1 to both products</change-summary>
      <data>Details of change 1</data>
    </change-item>
    <change-item product="productA">
      <personinfo>Tom Cihak</personinfo>
      <change-completed>2013-03-19</change-completed>
      <change-summary>Made change 2 to product A</change-summary>
      <data>Details of change 2</data>
    </change-item>
    <change-item product="productA productB">
      <personinfo>Tom Cihak</personinfo>
      <change-completed>2013-03-20</change-completed>
      <change-summary>Made change 3 to both products</change-summary>
      <data>Details of change 3</data>
    </change-item>
  </changehistory-list>
...
</prolog>

```

Figure 4: Excerpt from prolog of topic myTopic

To illustrate the desirability of date filtering, in this scenario product A's manual was updated on 1 Feb 2013, while product B's manual hasn't been revised since 10 Nov 2012. Then, on 22 Mar 2013, both manuals are being revised.

Here is what these documents should contain:

Table 1: Excerpt from product A's release notes, 22 Mar 2013, last issued 1 Feb 2013

Topic: myTopic	Made change 2 to product A
Topic: myTopic	Made change 3 to both products

Table 2: Excerpt from product B's release notes, 22 Mar 2013, last issued 10 Nov 2012

Topic: myTopic	Made change 1 to both products
Topic: myTopic	Made change 3 to both products

Note that change 1 already appeared in the 10 Nov 2012 release notes of product A. Therefore, it must *not* appear in the 22 Mar 2013 release notes, or it will alert the customer to something that hasn't changed (again).

Achieving this goal with select attributes alone would come at the cost of additional attribute values for not only each product, but for each revision of each product—which would become unmanageable when instead of two products there were 35, and instead of one topic 120,000.

Example 2

Example from the medical device industry. This example illustrates that the proposal meets the requirement for down-to-the second date/time recording.

```
<prolog>
...
  <change-historylist>
    <change-item>
      <personinfo>Joe Smith</personinfo>
      <change-completed>2013-03-19T13:30:22.25-05:00</change-completed>
      <change-summary>Made change 1</change-summary>
      <data href="#changel1site">Details of change 1</data>
    </change-item>
    <change-item>
      <personinfo>Bob Smith</personinfo>
      <organizationinfo>NAMDM</organizationinfo>
      <change-completed>1360652700</change-completed>
      <change-summary>Made change 2</change-summary>
      <data href="#change2site">Details of change 2</data>
    </change-item>
  </change-historylist>
...
</prolog>
```

Figure 5: Medical Device Industry Example