Migrating to DITA 2.0

18 October 2022

This version:
https://docs.oasis-open.org/dita/migrating-dita2.0/v1.0/cn01/migrating-dita2.0-v1.0-cn01.html (Authoritative)
https://docs.oasis-open.org/dita/migrating-dita2.0/v1.0/cn01/migrating-dita2.0-v1.0-cn01.pdf

Previous version:
N/A

Latest version:
https://docs.oasis-open.org/dita/migrating-dita2.0/v1.0/migrating-dita2.0-v1.0.html (Authoritative)
https://docs.oasis-open.org/dita/migrating-dita2.0/v1.0/migrating-dita2.0-v1.0.pdf

Technical Committee:
OASIS Darwin Information Typing Architecture (DITA) TC

Chair:
Kristen James Eberlein (kris@eberleinconsulting.com), Eberlein Consulting LLC

Editors:
Zoe Lawson (mailto:zoelawson17@hotmail.com), Casenet LLC
Eliot Kimber (, ServiceNow, Inc.

Additional artifacts:
This document is part of a work product that also includes:
• ZIP file that contains the DITA source for this document. https://docs.oasis-open.org/dita/migrating-dita2.0/v1.0/cn01/migrating-dita2.0-v1.0-cn01-dita.zip

Related work:
This document is related to:
• Darwin Information Typing Architecture (DITA) Part 1: Base Edition Plus Errata 02. OASIS Standard incorporating Approved Errata. http://docs.oasis-open.org/dita/dita/v1.3/dita-v1.3-part1-base.html. This edition contains topic and map; it is designed for implementers and users who need only the most fundamental pieces of the DITA framework.
• **Darwin Information Typing Architecture (DITA) Part 2: Technical Content Edition Plus Errata 02.** OASIS Standard incorporating Approved Errata. [http://docs.oasis-open.org/dita/dita/v1.3/dita-v1.3-part2-tech-content.html](http://docs.oasis-open.org/dita/dita/v1.3/dita-v1.3-part2-tech-content.html). This edition contains the base architecture plus the technical-content specializations; it is designed for authors who use information typing and document complex applications and devices.

• **Darwin Information Typing Architecture (DITA) Part 3: All-Inclusive Edition Plus Errata 02.** OASIS Standard incorporating Approved Errata. [http://docs.oasis-open.org/dita/dita/v1.3/dita-v1.3-part3-all-inclusive.html](http://docs.oasis-open.org/dita/dita/v1.3/dita-v1.3-part3-all-inclusive.html). This edition contains the base architecture, technical content, and the learning and training specializations. It is designed for implementers who want all OASIS-approved specializations, as well as users who develop learning and training materials.

**Abstract:**
Describes the considerations and processes for migrating DITA 1.x documents, tools, and processes to use DITA 2.0.

**Status:**
This document was last revised or approved by the OASIS Darwin Information Typing Architecture (DITA) TC on the above date. The level of approval is also listed above. Check the “Latest version” location noted above for possible later revisions of this document.

TC members should send comments on this document to the TC’s email list. Others should send comments to the TC’s public comment list, after subscribing to it by following the instructions at the “Send A Comment” button on the TC’s web page at [https://www.oasis-open.org/committees/comments/index.php?wg_abbrev=dita](https://www.oasis-open.org/committees/comments/index.php?wg_abbrev=dita).

**Citation format:**
When referencing this note, the following citation format should be used:

[Migrating-DITA2.0-v1.0]

*Migrating to DITA 2.0* Version 1.0. Edited by Zoe Lawson and Eliot Kimber. 18 October 2022. Working Draft 01. [https://docs.oasis-open.org/dita/migrating-dita2.0/v1.0/cn01/migrating-dita2.0-v1.0-cn01.html](https://docs.oasis-open.org/dita/migrating-dita2.0/v1.0/cn01/migrating-dita2.0-v1.0-cn01.html).

Latest version: [https://docs.oasis-open.org/dita/migrating-dita2.0/v1.0/migrating-dita2.0-v1.0.html](https://docs.oasis-open.org/dita/migrating-dita2.0/v1.0/migrating-dita2.0-v1.0.html).
Notices

Copyright © OASIS Open 2022. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.
Table of contents

1 Introduction................................................................................................................................................ 5
   1.1 What's new in DITA 2.0......................................................................................................................5
      1.1.1 New domains.............................................................................................................................5
      1.1.2 New elements and attributes.....................................................................................................6
      1.1.3 Changed elements.................................................................................................................... 6
      1.1.4 Changed attributes.................................................................................................................... 8
      1.1.5 Features removed from DITA 2.0.............................................................................................. 9
   1.2 How to use this guide...................................................................................................................... 10
   1.3 Planning your migration.................................................................................................................10
2 Migrating DITA content.............................................................................................................................11
   2.1 Planning your source migration....................................................................................................... 12
   2.2 Analyzing Content for Migration.......................................................................................................13
      2.2.1 Recommended Tools...............................................................................................................13
      2.2.2 Analyze your source................................................................................................................13
   2.3 Migrating to DITA 2.0.......................................................................................................................13
      2.3.1 Deprecate elements and attributes..........................................................................................13
      2.3.2 Update to new elements and attributes...................................................................................13
3 Migrating DITA specializations and constraints....................................................................................... 14
   3.1 Planning your specialization and constraint migration.....................................................................14
   3.2 Before you migrate your specializations and constraints.................................................................14
   3.3 Changes to specializations..............................................................................................................14
   3.4 Changes to constraints....................................................................................................................14
   3.5 New items you may want to consider using.....................................................................................14
4 Migrating DITA transforms....................................................................................................................... 15
   4.1 Planning your transform migration...................................................................................................15
   4.2 Before you migrate your transforms................................................................................................15
   4.3 Migrate your transforms................................................................................................................15
5 Migrating DITA Tools................................................................................................................................16
   5.1 Changed elements...........................................................................................................................16
   5.2 Changed attributes..........................................................................................................................16
   5.3 Changes to Chunking.......................................................................................................................16
   5.4 Changes to indexing........................................................................................................................16
A Acknowledgments...................................................................................................................................17
B Revision history....................................................................................................................................... 18
1 Introduction

Overview of this guide.

Welcome to the fun world of migrating to DITA 2.0!

Still the same DITA you know and love, but now new and improved!

You want to take advantage of all these great new features, however DITA 2.0 is the first release that is not backwards compatible. You will need to migrate your existing source to be valid with the DITA 2.0 schema.

Comment by Zoe
Should this be divided by spec type? base vs technical content? or do we have one pass for all the things, with stuff broken out by spec?

WEK: I think covering everything as one unit is appropriate because that's the way DITA 1.x is organized.

Disposition: Unassigned

This document is divided into sections:

Migrating your DITA source
What you need to do to make your DITA 1.x documents into conforming DITA 2.0 documents.

Migrating your DITA transforms
What you need to do to make your DITA 1.x-handling transformations also handle DITA 2.0 documents.

Migrating your DITA specializations and constraints
What you need to do to make your DITA 1.x specializations and constraints into DITA 2.0 specializations and constraints.

Migrating DITA tools
Migration considerations for other DITA-aware tools, such as editors and component content management systems.

1.1 What's new in DITA 2.0
The changes in DITA 2.0 focus on removing deprecated and unused features, reducing complexity, and relaxing some constraints.

1.1.1 New domains
DITA 2.0 adds domains for alternative titles and emphasis elements (<em> and <strong>)

Alternative titles domain
DITA 2.0 generalizes the concept of alternative titles as represented by the 1.x base element types <navtitle> and <searchtitle> into the new base element type <titlealt>. The alternative titles domain uses the new <titlealt> element to define <navtitle> and <searchtitle> as specializations. It also adds the new specializations <linktitle> and <subtitle>. 
Emphasis domain

The emphasis domain defines DITA versions of the HTML `<em>` (emphasis) and `<strong>` elements. Both elements are specializations of `<ph>`. These elements have the same semantics as the HTML elements. They serve as an alternative to the elements in the highlight domain.

1.1.2 New elements and attributes

DITA 2.0 adds a few new base element types and attributes

<include>

Points to non-DITA resources that should be directly included in rendered results. Intended for use with things like code samples that are not opaque binaries or objects used by reference.

<mapresources>

New map group domain element that sets `@processing-role` to "resource-only" by default. Intended as a general container for resource-only topic references, such as key definitions and topics that contain content used only by content reference. Used in the DITA 2.0 Bookmap specialization.

<audio> and <video>

New elements for including audio and video, modeled on the HTML `<audio>` and `<video>` elements. The new base elements `<fallback>`, `<media-source>`, and `<media-track>` are used within `<audio>` and `<video>`.

@subjectrefs

The `@subjectrefs` attribute associates `<map>` and `<topicref>` elements with subjects defined in subject scheme maps. The `@subjectrefs` attribute replaces the classification domain.

1.1.3 Changed elements

A number of elements from DITA 1.x are changed in some way.

Topic <prolog> and map <topicmeta> use <titlealt>

The `<titlealts>` element is removed. Alternative titles now go in `<prolog>` (topics) or `<topicmeta>` (maps).

The `<prolog>` element now allows zero or more `<titlealt>` elements, the new specialization base for `<navtitle>`, `<searchtitle>`, and `<subtitle>`. You can use `<titlealt>` as the specialization base for additional alternative titles.

The `<topicmeta>` element now also allows zero or more `<titlealt>` elements in place of the removed `<titlealts>` element. The new `<keytext>` element may occur before `<titlealt>` in `<topicmeta>`.

<shortdesc>, <linktext>, and <searchtitle>

These elements now have the same declarations for use in maps and topics.
<hazardstatement> redesigned

- The @type attribute values now limited to "caution", "danger", "notice", and "warning" per ANSI Z535.
- The <hazardsymbol> element is now a subelement of <messagepanel>
- Within <messagepanel> the <howtoavoid> and <consequence> elements may be paired in either order.
- The <typeofhazard> element can include <hazardsymbol>.
- The <messagepanel>, <typeofhazard>, <consequence>, and <howtoavoid> elements specialize from <div> instead of <ul> and <li>.

Highlight domain elements allow @keyref

The highlight domain elements, which all specialize from <ph>, now allow @keyref.

<longquote> only allows @keyref

The <longquote> element now only allows @keyref, removing the other linking-related attributes, making <longquote> consistent with <keyword> and other elements that are links only when they specify @keyref.

indexing

- The elements <index-see> and <index-see-also> from the DITA 1.x indexing domain are now base elements.
- Removed <index-base>.
- Removed <index-sort-as>.

<imagemap>

The <imagemap> element is specialized from <div> rather than <fig>, allowing it to be included within <fig>. The <area> element is also specialized from <div> rather than <figgroup>.

<simpletable> updates

- The @colspan, @rowspan, @headers, @scope attributes are added.
- The <simpletable> element now allows <title>.
- The @specentry attribute is removed from <stentry>.

<example> in more places

Example may occur most places where block elements (i.e., <p>) may occur. Because <example> can also occur wherever <section> can occur, it may be necessary to put <example> within <p> or <fig> within <conbody>, which only allows <section> and <example> after the first <section> or <example>.

<bookmap> updates

- Modify the content model of <bookmap> to allow zero or more <ditavalref> elements after <bookmeta>
- Add a <mapresources> element to the map group domain. The <mapresources> element is:
– Specialized from `<topicref>`
– Has a content model of:
  • Zero or one `<topicmeta>` elements
  • Zero or more `%data.elements.incl;` elements
  • Zero or more `<topicref>` elements
– Has the following attributes: `@href`, `@keys`, `@keyref`, `@keyscope`, `@processing-role`, `@type`, `@cascade`, `@scope`, `@format`, `@linking`, `@toc`, `@search`, and `%univ-atts;` attributes with the same defaults as for `<keydef>` except that `@keys` is not required.
  • The `<bookmap>` element now allows zero or more `<mapresources>` elements after `<ditavalref>`
  • The `<booklists>` element allows zero or more `<amendments>`

Add `<sub>` and `<sup>` to glossary elements

Elements in `<glossentry>` that allow other inline elements now allow `<sub>` and `<sup>`.

Ability to nest steps

The `<steps>` element is now allowed within `<step>`, allowing nested steps without the need for a separate `<substeps>` element. The `<substeps>` and `<substep>` elements have been removed.

`<resourceid>` takes over for `@copy-to`

The `@copy-to` attribute is removed.

To get the effect of `@copy-to` as a way to influence the deliverable anchors associated with a topic in a given use context, use the new `@appid-role` attribute on `<resourceid>` with the value of “deliverable-anchor” to create an application-specific resource ID that contributes to the deliverable anchor constructed by the associated processor. See [xref to topic on migrating `@copy-to` to `<resourceid>`].

`@spectitle` removed from `<dl>`, `<fig>`, `<lines>`, `<note>`, `<pre>`, `<sl>`, `<ul>`, `<ol>`

The `@spectitle` attribute has been removed. To generate titles for these elements, use `@outputclass` or declare specializations that get generated titles based on their specialized element types.

Attributes removed from `<object>` and `<param>`

The following attributes are removed from `<object>`: `@declare`, `@classid`, `@classidkeyref`, `@codebase`, `@codebasekeyref`, `@archive`, `@archivekeyrefs`, `@standby`.

The `@type` and `@valuetype` attributes are removed from `<param>`.

1.1.4 Changed attributes

`@outputclass` is universal

The `@outputclass` attribute can occur on any element, including elements in DITAVAL documents.
@audience, @platform, @product, and @otherprops are specialized from @props

The conditional attributes @audience, @otherprops, @platform, and @product are now formally specialized from @props.

@chunk

The values and processing expectations for @chunk are simplified, reflecting the way most DITA users actually used chunking and simplifying the processing required by the chunking feature.

@scale

The @scale attribute has been adjusted to make it more consistent on the various elements that allow it.

1.1.5 Features removed from DITA 2.0

DITA 2.0 is as much about removing cruft and little-used features as it is about adding new things and it is these removals that require the most attention when migrating content from DITA 1.x to 2.0.

Overview of removed features

DITA 2.0 removes all elements and attributes that were marked as deprecated in DITA 1.x. This includes the @alt, @navtitle, @title, and @spectitle attributes as well as @copy-to.

DITA 2.0 removes elements and attributes that had little or no adoption.

If you use any removed element or attribute and you have content and processing that depends on it you can develop your own specializations to make that markup available in your environment.

Removed elements

The following elements have been removed:

- `<topicset>` and `<topicsetref>`
- `<anchor>` and `<anchorref>`
- `<longquoteref>`
- `<sectiondiv>` (can be replaced by `<div>`)

Changed markup reflecting removed features

The following elements and attributes were changed from DITA 1.x.

- The `<substeps>` and `<substep>` are removed. The `<steps>` element may now nest within `<step>`.
- The @copy-to attribute is removed and replaced by new features of the `<resourceid>` element.
- The classification domain is removed. The new attribute @subjectrefs for `<topicref>` and `<map>` provides for associating maps and map branches with subject scheme subjects.
- The indexing domain elements are promoted to base elements, with `<index-base>` and `<index-sort-as>` removed. The `<sort-as>` element from the utilities domain can now be used within `<indexterm>` (because `<sort-as>` is a specialization of `<data>` and `<data>` is allowed within `<indexterm>`).
- `<linktext>` is replaced by the new `<keytext>` base element.
• The @domains attribute is replaced by the @specializations attribute. The @specializations attribute declares attribute specializations. Element specializations are no longer declared within document instance markup.

Removed deprecated features

The following elements and attributes were marked as deprecated in DITA 1.x and have been removed from DITA 2.0. Most of these changes reflect the removal of attributes that held text that should be in subelements to better support localization.

- <boolean>
- <data-about>
- <indextermref>
- <itemgroup>
- @alt and @longdescref on <image>
- @collection-type="tree" from <linkpool> and <linklist>
- @collection-type <reltable> and <recolspec>
- @keyref on <navref>
- @locktitle on <topichead> and <topicgroup>
- @mapkeyref on <metadata>
- @navtitle
- @othertype and @otherjob on <audience>
- @print
- @query
- @refcols
- @role="sample" and @role="external"
- @title on <map>
- @type="internal" and @type="external" on <lq>
- @xtrf and @xtrc
- Delay conref domain

1.2 How to use this guide

This describes the audiences, who should do what, what's where.

1.3 Planning your migration

Flowcharts, cost planning, etc. goes here. This is big picture. There will be specific details in each section.

Comment by eliot.kimber
I'm not sure this is an appropriate or practical section given that the details will be very specific to a given environment.
Disposition: Unassigned
2 Migrating DITA content

These are the changes you need to make to existing content.

Migrating your maps and topics involves the following activities:

1. Determining what removed or changed DITA markup your content uses.
2. For each removed or changed feature, determining how to react: simply remove it or change it?
3. Implementing or provisioning tools and documentation to make the changes.
4. Performing the changes.

Most, if not all, DITA tools that support DITA 1.x and 2.0 will support the use of both 1.x and 2.0 content in the same environment. For example, you may decide that it makes the most sense to migrate your DITA maps to use 2.0 markup (for example, to take advantage of the new @chunk and <resourceid> features), leave your existing topics as DITA 1.x, and only create new topics as DITA 2.0 topics. In any case, you should not be forced to do an all-or-nothing migration to DITA 2.0.

If you are using local document type shells you may need to create new DITA 2.0-specific versions of your shells in order to continue using the DITA 1.x versions along with DITA 2.0.

If you are not using local shells this is a good opportunity to start doing so, creating DITA 2.0 shells to use with your new and migrated maps and topics, leaving your 1.x content to use the Technical Committee-defined document type shells.

For the vast majority of DITA content the migration from 1.x to 2.0 will likely involve the following changes:

- Converting <linktext> to <keytext> for string key definitions.
- Converting <substeps> and <substep> to <steps> and <step>.
- Removing @navtitle from topic references or converting @navtitle to <navtitle> within <topicmeta>.

For most maps, the use of @navtitle or <navtitle> on non-topic-head topicrefs is something to be avoided as it can lead to unexpected behavior when the navigation titles authors see in their map editor do not match the published titles, which will come from the referenced topics. In the early days of DITA the use of navigation titles made it easier to edit maps where the editing tool could not practically reflect topic titles in the map editor. But modern DITA editors all provide visual map editors that can reflect topic titles with appropriate performance. In that environment, specifying navigation titles on topic references simply adds complexity and opportunities for error or confusion to maps.

- Converting @alt on <image> and its specializations to <alt>. As a matter of best accessibility practice, all images should have associated alternative text. The @alt attribute has been deprecated in favor of the @alt element since DITA 1.2. Note that <image> elements that use @keyref to refer to images can get the image alternative text from the key definition.
- Replacing @copy-to with the equivalent <resourceid> elements in topic references or with some other processor-defined way to determine the deliverable anchors for specific uses of topics. The @copy-to feature was always a bad idea, however well intentioned, but it was simply too specific to a particular way of generating HTML-based outputs. The DITA 2.0 <resourceid> enhancements for influencing deliverable anchors address the shortcomings of @copy-to but they do require both updating your content and configuring or enhancing your processors to use them.

Comment by eliot.kimber
It occurs to me that a cheap and easy implementation of resourceid in OT would be to simply extend the map preprocessing to generate @copy-to for appropriate <resourceid> elements. Looking at the 3.7.2 release notes I don’t see any indication that anyone is pursuing that. Disposition: Unassigned

Depending on the specific changes you need to make, the migration can be performed using global search and replace, XQuery updates (for example, using OxygenXML’s refactoring feature or an XQuery database such as BaseX or MarkLogic), or XSLT transforms, or even manual changes made by authors as they update maps and topics in the normal course of their work.

For example, it took about four hours of effort to migrate the ServiceNow Platform Tokyo release documentation to DITA 2.0, about 40,000 topics. This time included the time required to implement an OxygenXML refactor for migrating @copy-to to <resourceid>. Except for @copy-to, all the migration was done using regular expression search and replace. This four hours of effort was the first attempt to migrate the ServiceNow Platform content and does not reflect the time needed to verify the migrated content. The work was done in OxygenXML.

Unless you are already using local document type shells and simply update those shells to use the DITA 2.0 versions of the modules they use, you will need to update the DOCTYPE declarations or grammar references in all documents to be migrated to DITA 2.0.

For many documents, the easiest way to determine what changes you need to make is to update the DOCTYPE declarations or grammar references to DITA 2.0 document type shells and then validate the documents.

2.1 Planning your source migration

To plan your source migration you first need to determine what needs to change and determine what it will take to do the changing.

To plan your migration you must answer the following questions:

- Is all the content being migrated?
  
  If your DITA environment allows DITA 1.x and 2.0 content to be processed together it may not be necessary to migrate all your content in order to start using some DITA 2.0 features. For example, it might make sense to only migrate maps so that you can take advantage of the new @chunk and <resourceid> features while leaving topics unmodified.
  
  On the other hand, migration to DITA 2.0 is usually easy enough that there’s little reason not to migrate everything in one go.

- How consistent is your content?
  
  The more consistent your content is the easier it will be to migrate it cleanly and reliably. You can use tools like the open-source BaseX XQuery database or OxygenXML’s global search to inspect your content for elements and attributes that require migration.

- How much content do you have?
  
  The sheer volume of content shouldn’t affect the time required to perform the migration because modern tools can make changes to even tens of thousands of topics quite quickly.

  A more important consideration is the time required to verify that the result of the migration is correct. This verification can be automated in several ways, including using differencing tools to inspect the changes made but for production content almost certainly human inspection of both the source and key deliverables generated from it will be required. Human inspection is always time consuming, so plan for this effort.

- How is your content managed?
If you are managing your content in a source control system like git you can use normal source control features to manage the migrated content separate from your production content, for example by creating a git branch to hold migrated content while you implement and verify your migration process and results.

If you are managing your content in a CCMS it may be necessary to export your DITA 1.x content in order to implement and test your migration before committing the result back to the CCMS. Or your CCMS may offer content migration features.

- Is tool migration required?
  
  If you are using the latest versions of Open Toolkit and your DITA-aware editor then it is likely that no additional effort is required but if you are currently using older versions of tools that predate DITA 2.0 you may need to migrate those tools before your DITA 2.0 content can be edited, managed, or published.

- What DITA 2.0 training do authors require?
  
  While DITA 2.0 is not too different from DITA 1.x there are differences that will affect authors. You should plan for developing and delivering any new or changed author training required by the migration to DITA 2.0.

For many organizations migrating from DITA 1.x to 2.0 the migration of the content itself will often be the easiest part. Coordinating the migration in the context of ongoing development and publishing processes, as well as providing appropriate author support will often be the greatest challenge in a migration effort.

2.2 Analyzing Content for Migration

To plan your content migration you must first determine what needs to be migrated.

2.2.1 Recommended Tools

The easiest way to determine what must change is to validate your DITA documents against the DITA 2.0 document types.

2.2.2 Analyze your source

Migration is much easier if you know all the specifics of what needs to happen in your source.

2.3 Migrating to DITA 2.0

Now that you have all your tools prepared and have analyzed your source, time to actually do the migration.

2.3.1 Deprecate elements and attributes

Some elements have been removed. You need to make sure your source no longer contains them.

2.3.2 Update to new elements and attributes

There are new elements you may want to start using.
3 Migrating DITA specializations and constraints

3.1 Planning your specialization and constraint migration
What do you need to analyze and sort out before you start making changes.

3.2 Before you migrate your specializations and constraints

3.3 Changes to specializations
Some definitions are updated. You may need to rework some of your specializations.

3.4 Changes to constraints
Some aspects of DITA have been refactored. To most writers there is no change, but how they're referenced has changed.

3.5 New items you may want to consider using
4 Migrating DITA transforms

You will also need to make changes to your DITA transforms.

4.1 Planning your transform migration
What you need to think about, analyze, etc. to plan your changes to your transforms.

4.2 Before you migrate your transforms
Tools to gather, things to do.

4.3 Migrate your transforms
5 Migrating DITA Tools

If you develop tools to store, publish, or work with DITA, use this section to help plan updates to your tools to make them compliant with DITA 2.0

5.1 Changed elements
New elements to support, and existing elements that have been deprecated. Focusing on the easy changes.

- include

5.2 Changed attributes
New attributes to support, and existing attributes that have been deprecated. Focusing on the easy changes.

- outputclass

5.3 Changes to Chunking

5.4 Changes to indexing
A Acknowledgments

The following individuals participated in the creation of this document and are gratefully acknowledged.

<Name, Company | Individual member>

In addition, the <name of OASIS TC> also would like to recognize the following people for their insights and support:

<Name>
<Name>
# B Revision history

The following table contains information about revisions to this document.

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Editor</th>
<th>Description of changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>18 October 2022</td>
<td>Kristen James Eberlein</td>
<td>Generated working draft #01.</td>
</tr>
</tbody>
</table>