XML Documentation for Adobe Experience Manager
Installation and Configuration Guide

August 21, 2018
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Migrate existing content</td>
<td>20</td>
</tr>
<tr>
<td>Upload existing DITA content</td>
<td>20</td>
</tr>
<tr>
<td>Use a WebDAV tool</td>
<td>20</td>
</tr>
<tr>
<td>Use FrameMaker</td>
<td>20</td>
</tr>
<tr>
<td>Use curl commands</td>
<td>22</td>
</tr>
<tr>
<td>Migrate non-DITA content</td>
<td>23</td>
</tr>
<tr>
<td>Migrate Microsoft Word documents</td>
<td>24</td>
</tr>
<tr>
<td>Migrate XHTML documents</td>
<td>25</td>
</tr>
<tr>
<td>Migrate DocBooks</td>
<td>26</td>
</tr>
<tr>
<td>Migrate unstructured FrameMaker documents</td>
<td>27</td>
</tr>
<tr>
<td>Migrate any other structured document</td>
<td>29</td>
</tr>
<tr>
<td>Configure templates and conditional tags</td>
<td>31</td>
</tr>
<tr>
<td>Configure custom DITA topic template</td>
<td>31</td>
</tr>
<tr>
<td>Configure custom DITA map template</td>
<td>32</td>
</tr>
<tr>
<td>Customize Web Editor</td>
<td>33</td>
</tr>
<tr>
<td>Customize toolbar</td>
<td>33</td>
</tr>
<tr>
<td>Add a feature in the toolbar</td>
<td>33</td>
</tr>
<tr>
<td>Remove a feature from the toolbar</td>
<td>35</td>
</tr>
<tr>
<td>Customize WYSIWYG view</td>
<td>36</td>
</tr>
<tr>
<td>Configure enterprise or product-specific profiles</td>
<td>37</td>
</tr>
<tr>
<td>Configure global profile</td>
<td>38</td>
</tr>
<tr>
<td>Create and configure a folder-specific profile</td>
<td>38</td>
</tr>
<tr>
<td>Configure conditional attributes</td>
<td>39</td>
</tr>
<tr>
<td>Configure authoring templates</td>
<td>40</td>
</tr>
<tr>
<td>Create custom authoring template</td>
<td>42</td>
</tr>
<tr>
<td>Version management</td>
<td>44</td>
</tr>
<tr>
<td>Configure file auto-checkout for Web Editor</td>
<td>44</td>
</tr>
<tr>
<td>Integrate desktop-based XML editors</td>
<td>45</td>
</tr>
<tr>
<td>Enable file editing in FrameMaker from the Web Editor</td>
<td>45</td>
</tr>
<tr>
<td>Configure output generation settings</td>
<td>46</td>
</tr>
<tr>
<td>Configure output presets</td>
<td>46</td>
</tr>
<tr>
<td>Apply preset changes</td>
<td>47</td>
</tr>
<tr>
<td>Configure FrameMaker Publishing Server</td>
<td>47</td>
</tr>
<tr>
<td>Configure blended publishing within an existing AEM Site</td>
<td>48</td>
</tr>
<tr>
<td>Customize AEM Site output design template</td>
<td>50</td>
</tr>
<tr>
<td>Customize design template for generating output</td>
<td>51</td>
</tr>
<tr>
<td>Customize DITA element mapping with AEM components</td>
<td>53</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>elementmapping.xml structure</td>
<td>53</td>
</tr>
<tr>
<td>DITA element schema</td>
<td>56</td>
</tr>
<tr>
<td>Customize DITA map console</td>
<td>58</td>
</tr>
<tr>
<td>Handle image rendition during output generation</td>
<td>58</td>
</tr>
<tr>
<td>Change the recently generated outputs list limit</td>
<td>59</td>
</tr>
<tr>
<td>Output generation performance optimization</td>
<td>59</td>
</tr>
<tr>
<td>Customize and customize workflows</td>
<td>60</td>
</tr>
<tr>
<td>Customize review workflow</td>
<td>60</td>
</tr>
<tr>
<td>Customize email templates</td>
<td>62</td>
</tr>
<tr>
<td>Customize post-output generation workflow</td>
<td>62</td>
</tr>
<tr>
<td>Configure post-processing XML workflow launcher</td>
<td>64</td>
</tr>
<tr>
<td>Translate content</td>
<td>66</td>
</tr>
<tr>
<td>Enable component-based translation workflow</td>
<td>66</td>
</tr>
<tr>
<td>Customize search form</td>
<td>67</td>
</tr>
<tr>
<td>Appendix</td>
<td>68</td>
</tr>
<tr>
<td>Custom templates</td>
<td>68</td>
</tr>
<tr>
<td>Content migration and upload</td>
<td>68</td>
</tr>
<tr>
<td>Workflow offloading</td>
<td>69</td>
</tr>
<tr>
<td>DITA-OT Profiles configuration</td>
<td>69</td>
</tr>
<tr>
<td>Using custom DITA-OT</td>
<td>70</td>
</tr>
<tr>
<td>Export custom DITA profile as package</td>
<td>71</td>
</tr>
<tr>
<td>Output history</td>
<td>71</td>
</tr>
<tr>
<td>PDF generation</td>
<td>72</td>
</tr>
<tr>
<td>Using post generation workflows</td>
<td>72</td>
</tr>
<tr>
<td>Unstructured publishing</td>
<td>72</td>
</tr>
<tr>
<td>Baseline</td>
<td>72</td>
</tr>
<tr>
<td>AEM Site publishing</td>
<td>73</td>
</tr>
<tr>
<td>Other recommendations</td>
<td>75</td>
</tr>
<tr>
<td>Permissions</td>
<td>75</td>
</tr>
<tr>
<td>Translation</td>
<td>75</td>
</tr>
<tr>
<td>Logging</td>
<td>76</td>
</tr>
<tr>
<td>Unicode support</td>
<td>76</td>
</tr>
<tr>
<td>DTD</td>
<td>76</td>
</tr>
<tr>
<td>Overridden AEM features</td>
<td>76</td>
</tr>
<tr>
<td>FrameMaker integration</td>
<td>77</td>
</tr>
</tbody>
</table>
About this guide

XML Documentation for Adobe Experience Manager (referred to as XML Documentation solution later in this guide) is a powerful, enterprise-grade component content management solution (CCMS). It enables native DITA support in Adobe Experience Manager, empowering AEM to handle DITA-based content creation and delivery. It empowers authors to create content using any offline DITA authoring tool, such as Adobe FrameMaker or an easy-to-use built-in Web Editor.

This guide provides the instructions to download, install, and configure XML Documentation solution. In this guide, you will find detailed instructions to set up XML Documentation solution as per your organizational authoring and publishing needs.

This guide is intended for the following type of audiences:

- Administrators, who would install and manage XML Documentation solution on Adobe Experience Manager.
- Publishers, who would run the publishing task to generate output in various formats.

Content structure

The information in this guide is organized as follows:

- **About this guide**: This topic provides an introduction to this guide, intended audience, and how the information is organized in this guide.
- **Download and install**: This topic describes how to download, install, or upgrade XML Documentation solution.
- **User administration and security**: This topic describes the core concept of users and authentication in AEM and the default user groups created by XML Documentation solution.
- **Use custom DITA-OT and DITA specialization**: This topic explains how to configure custom DITA-OT plug-ins and use DITA specialization.
- **Configure document states**: This topic explains how to configure custom states for your DITA documents.
- **Migrate existing content**: This topic describes how to on-board your existing content on AEM repository.
- **Configure templates and conditional tags**: This topic describes how to configure topic and map templates to meet your authoring needs. Also, learn about tagging system in AEM and how to configure tags to work with XML Documentation solution.
- **Customize Web Editor**: This topic explains the various customizations that you can make in the Web Editor to enhance its functionality.
- **Configure enterprise or product-specific profiles**: This topic explains the process of creating folder profiles and giving permissions to specific users.
- **Version management**: This topic describes how to configure automatic file checkout for files that are opened for editing in the Web Editor.
• **Integrate desktop-based XML editors**: This topic describes the settings you need to enable editing documents in an external XML editor.

• **Configure output generation settings**: This topic describes the various configurations that you can make to customize the default output generation process.

• **Configure and customize workflows**: This topic describes the various configurations to customize the default workflows shipped in XML Documentation solution.

• **Translate content**: This topic provides links to the relevant Help articles in AEM documentation to help you understand and configure the translation framework. Also, learn how to enable component-based translation workflow.

• **Customize search form**: This topic describes how to configure the search form to add filtering options for DITA type documents.

• **Appendix**: The appendix contains information about the best practices for working with or setting up features in XML Documentation solution.

### Additional resources

Following is a list of other helpful resources of XML Documentation solution:

• User Guide for Adobe Experience Manager 6.4 and 6.3, 6.2, and 6.1

• API Reference Guide

• Quick Start Guide for Adobe Experience Manager 6.4 and 6.3, 6.2, and 6.1
Download and install

The XML Documentation solution is made available through Adobe Licensing Website (LWS). You can download the XML Documentation solution from your LWS account and install it on all Adobe Experience Manager (AEM) instances in your setup. Typically, your authoring instance and production instance of AEM will be hosted on different servers. You will have to install the XML Documentation solution on all instances of AEM that you intend to use.

Before you begin the download and installation process, you must ensure that your system meets the technical requirements to install the XML Documentation solution.

Technical requirements

Adobe Experience Manager
- Version 6.3 Service Pack 1
- Version 6.2 Service Pack 1, Cumulative Fix Pack 10
- Version 6.1 Service Pack 2, Cumulative Fix Pack 13

Operating systems
- Microsoft Windows Server 2012 R2
- Red Hat Linux 7 and 6
- Ubuntu

Java Development Kit
- Oracle SE 8 JRE 1.8.x
- Oracle SE 7 JRE 1.7.x

Web browser
- Google Chrome (recommended)

Install Adobe Experience Manager

XML Documentation solution is a plug-in that installs on top of Adobe Experience Manager. Installing AEM requires understanding of some basic AEM concepts and recommended deployment scenarios. The following links will help you get started with AEM installation:
- Basic AEM Concepts in AEM 6.4, 6.3, 6.2, and 6.1
- Recommended AEM Deployments in AEM 6.4, 6.3, 6.2, and 6.1

Once you have identified the deployment strategy that works best for your organization, perform the installation process as described in the Getting Started in AEM 6.4, 6.3, 6.2, and 6.1.
Download and install XML Documentation solution for the first time

Perform the following steps to download and install the XML Documentation solution for the first time on a computer:

**IMPORTANT:** As XML Documentation solution works in conjunction with AEM’s native functionalities, ensure that you have released the lock on all files before starting the installation process.

1) Download the XML Documentation solution from your LWS account.
2) Log into your AEM instance and navigate to the CRX Package Manager. The default URL to access the package manager is:
   
   http://<server name>:<port>/crx/packmgr/index.jsp
   
   *The Package Manager manages the packages on your local AEM installation. For more information about working with the Package Manager, see How to Work With Packages in AEM 6.4, 6.3, 6.2, and 6.1.*

3) To upload the XML Documentation solution, click **Upload Package**.
4) In the Upload Package dialog, navigate to the XML Documentation solution file that you downloaded in Step 1 and click **OK**.
   
   *The package is uploaded to your AEM instance.*
5) To install the package, click **Install**.

![](https://example.com/package.png)

6) In the Install Package dialog, click **Install**.

7) To get started with the XML Documentation solution, click the Home button in the upper-left corner of the CRX Package Manager.

**NOTE:** Perform the installation procedure on all instances of AEM servers in your setup.

### Verify XML Documentation solution installation

Once you have installed XML Documentation solution, you need to verify whether the installation was successful or not. Perform the following steps to verify the installation process:

1) Log into your AEM instance and navigate to the AEM Web Console Bundles page. The default URL to access the bundles page is:
   
   `http://<server name>:<port>/system/console/bundles`

   *A list of bundles is shown.*

2) Filter the list of bundles by entering `fmdita` in the filtering text box and press **Enter**.

   *The list of bundles is filtered to show the bundles installed by XML Documentation solution. If the installation was successful, then all installed bundles will have a **Status of Active**.*

   *If any of the bundles does not have an **Active** status, then check the AEM logs to troubleshoot the installation issue.*

### Upgrade XML Documentation solution

**IMPORTANT:** As XML Documentation solution works in conjunction with AEM’s native functionalities, ensure that you have released the lock on all files before starting the upgrade process.

If you are using XML Documentation solution version 2.2 or later, then you can easily upgrade to the latest version of XML Documentation solution. With the upgrade feature, you don’t have to uninstall the...
previous version XML Documentation solution. The procedure to upgrade your XML Documentation solution is similar to the fresh install. You need to download the XML Documentation solution package, upload it on AEM, and install it to upgrade your older instance of XML Documentation solution.
User administration and security

To access and configure features in XML Documentation solution, you need to create users. These users can then be assigned permissions or roles to access all or specific features in XML Documentation solution. Learn how to configure and maintain user authorization and also understand the theory behind how authentication and authorization works in AEM.

The following topics will help you understand the user administration and security related concepts and features:

- Users and Groups in AEM 6.4, 6.3, 6.2, and 6.1
- Permissions in AEM in 6.4, 6.3, 6.2, and 6.1
- Managing Users and Groups in AEM 6.4, 6.3, 6.2, and 6.1
- Managing Permissions in AEM 6.4, 6.3, 6.2, and 6.1

User groups created by XML Documentation solution

XML Documentation solution provides three out-of-the-box groups (or roles) to manage different tasks in a DITA project. These roles are - Authors, Reviewers, and Publishers. Depending upon the role a user is associated with, they are allowed to perform specific tasks. For example, publishing task can be performed only by a publisher, but not by an author or a reviewer. Similarly, an author can create a new topic, and a reviewer can only review a topic.

By default, there are user roles created by AEM, such as Owners, Editors, and Observers. XML Documentation solution maps the Owners role with Publishers and Editors role with Authors. This way all users who are a part of Owners or Editors role in AEM, they get all privileges available to the Publishers and Authors role in XML Documentation solution.

**TIP:** See Permissions for best practices around setting user permissions.

The following table lists various tasks and the roles that can perform those tasks:

<table>
<thead>
<tr>
<th>Task</th>
<th>Authors or Editors</th>
<th>Reviewers</th>
<th>Publishers or Owners</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create DITA Topic¹</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Create DITA Map</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Map Collections</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Create Review Task</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Review Topic²</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Key Resolution</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
</tbody>
</table>
## CHAPTER 3 USER ADMINISTRATION AND SECURITY

<table>
<thead>
<tr>
<th>Task</th>
<th>Authors or Editors</th>
<th>Reviewers</th>
<th>Publishers or Owners</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open in FrameMaker</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Check-out/Check-in</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Edit Topic</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Move Topic</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Edit Topic Properties</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Copy</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Delete</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Share</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Document state</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create/edit document state profile</td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Change document state[^3]</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Features available in DITA map console (Output Presets tab)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Generate</td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Edit</td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Duplicate</td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Create</td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Delete Preset</td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Features available in DITA map console (Outputs tab)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>View generated output</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Features available in DITA map console (Topics tab)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create Review Task</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Edit</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Features available in DITA map console (Baselines tab)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create</td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Edit</td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Duplicate</td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
</tbody>
</table>
The following list contains some recommendations and points related to user groups and corresponding permissions:

- If you want a user to start the translation or a review workflow, ensure that the user is a member of the `publishers` and `projects-administrators group`, with read, create, delete, and modify permissions available on the source and target language folders.

- If you create a project, you are the owner of the project with Publisher permissions. For other users in a project to be able to see team members, create tasks, or create workflows, they must have read access on `/home/users` and `/home/groups` nodes. One way of giving read access on `/home/users` and `/home/groups` nodes is by giving read access to the `projects-users` group.

- Reviewers can access and add review comments on a topic under review from the Project console or from inbox notification link. Also, this access is only available till the time the review task is open.

- By default, Publisher is granted access and permissions on the following folders in DAM:
  - `/content/fmdita` → Read and Write
  - `/content/dam/outputs` → Read and Write
  - `/content/output/sites` → Read and Write

You must give explicit read and write permissions to your publisher if you are using any other location apart from the default publishing locations mentioned above.

- All users under Authors, Reviewers, and Publishers roles have read access on all content in DAM.

- Authors and Publishers have create, update, read, and delete permissions on content in their project only.

- If you want to give admin rights to any user, you can do so by giving them access through AEM standard groups like administrators, projects-administrators, or OSGI configuration (Felix console).

---

### Additional notes on user groups

1. If a user is not a member of the project, but is a member of AEM Authors or Publishers group, then such users must be given explicit write permission on the project folder to be able to create or edit document. In addition, user can create topics or maps only in the projects where they are the member of that project.

2. If Authors and Publishers are invited for a review.

3. Depending on the rights given to the user in the document state profile.
• If an asset is not a part of any project, then all roles will only have read permission on such asset. The administrator will have to give explicit permissions (create, update, or delete) to the desired role.

• If an asset is a part of two projects, then the access rights on such asset is governed by the roles’ permissions.

• If a user is assigned different roles in two projects containing same asset, then that user gets combined permissions from different roles. For example:
  – User X is Author in Project-A
  – User X is Reviewer in Project-B
  – Both projects use the same asset

  In this scenario, user X will have all the privileges of Authors and Reviewers on the shared asset in both projects.

• To give a user rights to change a document state, make sure that you add the user in the state transition section of the document state profile.
Use custom DITA-OT and DITA specialization

The DITA Open Toolkit (DITA-OT) is a set of Java-based, open source tools that provide processing for DITA maps and topic content. XML Documentation solution allows you to easily import and use custom DITA-OT plug-ins. Once imported, XML Documentation solution can be configured to use the custom DITA-OT plug-in to generate output in any format. At the time of generating the output, simply select the DITA-OT option, and the XML Documentation solution uses the custom DITA-OT plug-in to generate the required output.

**NOTE:** XML Documentation solution is shipped with DITA-OT version 2.5.2. However, XML Documentation solution supports DITA-OT version 1.7 and above. For the complete list of DITA-OT versions, see DITA-OT versions.

**TIP:** See DITA-OT Profiles configuration and Using custom DITA-OT for best practices around using custom DITA-OT plug-ins.

Use custom DITA-OT plug-ins

There are two ways to use custom DITA-OT plug-in for publishing. First method is to upload the custom DITA-OT plug-in into AEM repository. The other method is to save the custom DITA-OT plug-in on your server, create a Profile and provide the location of the custom DITA-OT plug-in in your Profile.

By default, XML Documentation solution comes with a pre-configured Profile that contains the configurations for the default templates to use for editing and publishing content. You can create custom profiles with custom templates to be used while editing documents and custom DITA-OT plug-ins to publish content.

The default DITA-OT package available with XML Documentation solution comes with Apache FOP XSL-FO processor, which does not support rendering of MathML equations. If you are using MathML equations in your content, then ensure that you have integrated a MathML rendering engine plug-in for Apache FOP or use a different XSL-FO processor.

**IMPORTANT:** If you have upgraded XML Documentation solution from version 2.2 to 2.5.1 or 2.6, then all changes made through the configuration manager are automatically picked and stored in the Default Profile.

Perform the following steps to upload custom DITA-OT plug-in into AEM repository:

1) Log into AEM and open the CRXDE Lite mode.
2) Download the DITA-OT.ZIP file.
The location of the DITA-OT.ZIP file is /etc/fmdita/dita_resources/DITA-OT.zip.

3) Extract the contents of the zip file on your server.

4) Use a DITA-OT plug-in integrator mechanism to integrate the new version of DITA-OT with your custom DITA-OT plug-in.

   **NOTE:** For more information about manually integrating plug-ins, see the *Manually installing plug-ins* topic in DITA-OT documentation.

5) Create the ZIP file again keeping the same name (DITA-OT.ZIP) and the folder structure.

6) Upload the updated ZIP file back into the AEM repository.

   **NOTE:** It is recommended not to overwrite the default DITA-OT package. You should upload your custom DITA-OT package containing your plug-in at some other location under the apps folder.

7) Open the default DITA Profile for editing and save it (without making any updates) for the changes to take effect.

Perform the following steps to create a new profile and configure it to use custom DITA-OT plug-in stored on your server:

1) Store the custom DITA-OT plug-in on your server.

   **NOTE:** The folder structure for storing the custom DITA-OT plug-in should be: \<parent-folder>\DITA-OT.

2) Click on the Adobe Experience Manager link at the top and choose **Tools**.

3) Select **XML Documentation** from the list of tools.

4) Click on the **DITA Profiles** tile.

   **NOTE:** The Default Profile information is displayed in the Profiles page. If you have upgraded XML Documentation solution from version 2.2 to 2.5.1 or 2.6, then all changes made through the configuration manager are automatically picked and stored in the Default Profile.
5) You can choose to edit the Default Profile, create a new profile, or duplicate settings from the Default Profile to create a new profile.

**NOTE:** You can update the Default profile, but you cannot delete it. However, all new profiles that you create can be edited and deleted.

6) Configure the following properties to use the custom DITA-OT plug-in:

<table>
<thead>
<tr>
<th>Property name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Profile Properties</strong></td>
<td></td>
</tr>
<tr>
<td>Profile Name</td>
<td>Provide a unique name for this profile.</td>
</tr>
<tr>
<td>Reuse Output</td>
<td><em>(Optional)</em> If your profile is based on an existing profile, then select this option. Selecting this option ensures that XML Documentation solution does not extract the contents of DITA-OT package again and reuses the existing DITA-OT package.</td>
</tr>
<tr>
<td>Profile Extract Path</td>
<td><em>(Optional)</em> Specify the path where DITA-OT is kept on disk. By default, XML Documentation solution bundles a DITA-OT package in its repository and it is extracted on the disk at this path. <strong>NOTE:</strong> You can define this path using any existing system variable or property. See description the DITA-OT Environment Variables property for more information.</td>
</tr>
<tr>
<td>Assigned Path</td>
<td><em>(Optional)</em> Specify the path in your content repository for which this profile is applicable. You can specify multiple locations.</td>
</tr>
<tr>
<td><strong>DITA-OT Properties</strong></td>
<td></td>
</tr>
<tr>
<td>DITA-OT Timeout</td>
<td><em>(Optional)</em> Specify the time (in seconds) for which the XML Documentation solution waits for a response from the DITA-OT plug-in. If no response if received in the specified time, XML Documentation solution terminates the publishing task and the task is flagged as failed. Also, the failure logs are made available in the output generation log file. Default value: 300 seconds (5 minutes)</td>
</tr>
<tr>
<td>DITA-OT PDF Arguments</td>
<td><em>(Optional)</em> Specify the custom command-line arguments that are processed by the custom DITA-OT plug-in for generating the PDF output.</td>
</tr>
<tr>
<td>DITA-OT AEM Arguments</td>
<td><em>(Optional)</em> Specify the custom command-line arguments that are processed by the custom DITA-OT plug-in for generating the AEM Site output.</td>
</tr>
<tr>
<td>DITA-OT Build XML</td>
<td><em>(Optional)</em> Specify the path of the custom Ant build script bundled with the customized DITA-OT plug-in. This path is relative to the DITA-OT directory on your file system.</td>
</tr>
<tr>
<td>Property name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DITA-OT Ant Script Folder</td>
<td><em>(Optional)</em> Specify the path of the DITA-OT Ant script folder. This path is relative to the DITA-OT directory on your file system.</td>
</tr>
</tbody>
</table>
| DITA-OT Environment Variables       | *(Optional)* Specify environment variables to pass on to the DITA-OT process. By default, the XML Documentation solution adds four variables - ANT_OPTS, ANT_HOME, PATH, and CLASSPATH. You can reuse any of the existing system environment variables or properties for building new environment variables. For example, if you have JAVA_HOME system variable defined in your system and you want to define a new environment variable called JAVA_BIN that is built using JAVA_HOME. Then, you can add the definition of JAVA_BIN as:  
   JAVA_BIN= ${JAVA_HOME}/bin  
   **NOTE:** You can also use Java system properties to build environment variables. For example, if AEM start script defines a Java system property java.io.tmpdir to a temporary directory, you can use this property to define new variable as:  
   ${java.io.tmpdir}/fmdita/dita_ot.  
   **IMPORTANT:** To reuse any existing system variable or property, it must be enclosed within ${}. |
| Overwrite DITA-OT Output            | *(Optional)* If this option is selected, then the DITA-OT package from the AEM repository will replace the one on the disk. This is done on the activation of the ConfigManager. If you want to specify the path of a DITA-OT package that is available on your server, then deselect this option. |
| AEM DITA-OT Zip Path/Local DITA-OT Directory Path | Depending on your selection in the Overwrite DITA-OT Output, specify the complete path where the custom DITA-OT.zip file is stored. This could be the path in your AEM repository or local system. |
| DITA-OT Plug-in Path                | Path of the custom plug-in. This plug-in is integrated automatically with the main DITA-OT package.                                           |
| Integrate Catalogs                  | *(Optional)* Path of the custom DTD and XSD catalog.xml files in the AEM repository. This should be provided only when the catalogs are missing from the DITA-OT package. These catalogs are automatically integrated with the main DITA-OT as a plug-in. |
| Add System ID Catalog               | *(Optional)* Select this option only if there are missing Public ID entries in the catalog or if the DITA files use only the System IDs that are relative to the server path from where they are uploaded. |
INTEGRATE DITA SPECIALIZATION

CHAPTER 4 USE CUSTOM DITA-OT AND DITA SPECIALIZATION

NOTE: The XML Documentation solution installer creates two environment variables that you can use to specify the path of the custom DITA-OT plug-in files. These environment variables are: DITAOT_DIR, which contains the path of the DITA-OT directory on the file system; and DITA-MAP_DIR, which contains the path where the DITA map content is extracted on the file system.

7) Click Done to save the profile.

NOTE: You can export the custom DITA profile as a package and upload on the other XML Documentation solution instances to save time. For more information, see Export custom DITA profile as package.

Integrate DITA specialization

DITA specialization is the process of creating new DITA structures by adding new elements or removing existing elements. To create a new DITA element, you can take an existing DITA element as the base and modify it as per your authoring requirements. In essence, DITA specialization allows you to create customized information models that meet your business requirements while retaining the benefits of the existing DITA architecture.

You can use the Profile feature to store custom DITA specialization settings. These settings can then be used at the time of authoring and publishing custom DITA content. XML Documentation solution allows you to use Public ID and System ID in your custom DTDs/XSDs.

NOTE: XML Documentation solution Web Editor does not have support for XSDs.

Perform the following steps to create a new profile and configure it to use specialized DTDs and XSDs in your XML Documentation solution:

1) Create a specialization folder on your local machine that contains the specialized DTDs and XSDs.

2) Specify the DTD details in the catalog.xml file that must also be included in the specialization folder.

   NOTE: In case of DITA 1.3, the default location for DTD catalog.xml file in the AEM repository is: /etc/fmdita/dita_resources/DITA-1.3/dtd/catalog.xml.

3) Specify the XSD details in the catalog.xml file that must also be included in the specialization folder.

   NOTE: In case of DITA 1.3, the default location for XSD catalog.xml file in the AEM repository is: /etc/fmdita/dita_resources/DITA-1.3/xsd/catalog.xml.
4) Upload the folder to the following location:
   `/etc/fmdita/dita_resources`

5) Click on the Adobe Experience Manager link at the top and choose **Tools**.

6) Select **XML Documentation** from the list of tools.

7) Click on the **DITA Profiles** tile.
   
   **NOTE:** The Default Profile information is displayed in the Profiles page. If you have upgraded XML Documentation solution from version 2.2 to 2.5.1 or 2.6, then all changes made through the configuration manager are automatically picked and stored in the Default Profile.

8) You can choose to edit the Default Profile, create a new profile, or duplicate settings from the Default Profile to create a new profile.
   
   **NOTE:** You cannot delete the Default Profile. However, all new profiles that you create can be edited and deleted.

9) In the **Schema > Catalog** settings, specify the path of the custom DTD and XSD `catalog.xml` files in your AEM repository.

10) Select the **Add System ID Catalog** option.
    
    **NOTE:** Select this option only if there are missing Public ID entries in the catalog or if the DITA files use only the System IDs that are relative to the local file path from where they are uploaded.

    For more information about other properties on the Profiles page, see the properties table in Step 6 of the Use custom DITA-OT plug-ins section.

11) Click **Done** to save the profile.

**NOTE:** You can export the custom DITA profile as a package and upload on the other XML Documentation solution instances to save time. For more information, see *Export custom DITA profile as package*. 
Configure document states

XML Documentation solution lets you define the document states for your DITA topics according to your organization’s requirements. You can define the different states of your document from start to the end. For example, the first state can be draft and it can move to review, approved, translated, and finally to published.

You can also define who can move the document from one state to another state. For example, an author can create a document and the default state of the document can be draft. When the author sends the document for review he can change the document state to in review. The reviewer can change the document state to either approved or to draft again based on the review process. If the document is approved, the publisher can change the document state to translated or published depending on the workflow.

**NOTE:** If a user belongs to the *administrators* group, the user can change a document’s state from any state irrespective of the document state change transition defined in the system.

Create a document state

To create a document state, perform the following steps:

1) *(In AEM 6.4, 6.3, and 6.2)* Click on the Adobe Experience Manager link at the top and choose **Tools**.

   *(In AEM 6.1)* Click on the Experience Manager link, and then click **Tools**.

2) Select XML Documentation from the list of tools.

3) Click on the Document States tile. Assets States page is displayed. By default, the page shows a default profile.

4) Click **Create Profile** and enter the following details:
   – Enter the name for the profile in the Profile field.
   – Specify the path where you want to apply the new profile.
   – Specify the states of the document in the **Allowed States** under **States**. The default document state are Draft, Edit, In-Review, Approved, and Done.

   *Click Add button to add a document state.*

   *Click Delete icon to delete a document state.*

   **NOTE:** Do not delete a document state if documents are still in that state. If you delete a document state, you won’t be able to change the document state of such documents unless you belong to the *administrator* user group.

   – Specify the start state of the document in the **Start State**.
   – Specify the end state of the document in the **End State**.
   – Specify the state transition of the document in **From** and **To** under **State Transition**.

   *Specify the users and user groups who can change the document state in Groups.*

   *Click Add button to add a state transition.*

   **Click Delete icon to delete a state transition.**
NOTE: Do not delete a state transition if documents are still in From state. If you delete a state transition, you won’t be able to change the document state of such documents unless you belong to the administrator user group.

5) Click Done.

Create a copy of a document state profile

Depending on your requirement, you can create a copy of an existing document state profile. You can use the copy as a base for creating another document profile.

To create a copy of a document state profile, perform the following steps:

1) (In AEM 6.4, 6.3, and 6.2) Click on the Adobe Experience Manager link at the top and choose Tools.
   (In AEM 6.1) Click on the Experience Manager link, and then click Tools.
2) Select XML Documentation from the list of tools.
3) Click on the Document States tile.
   Assets States page is displayed.
4) Select the document state profile that you want to duplicate and click Duplicate Profile.
5) Make required changes and click Done.

Delete a document state or state transition

NOTE: Do not delete a document state or state transition if documents are still in the state or in state transition. If you delete a state or state transition, you won’t be able to change the document state of such documents unless you belong to the administrator user group.

Perform the following steps to delete a document state or state transition from a document state profile:

1) (In AEM 6.4, 6.3, and 6.2) Click on the Adobe Experience Manager link at the top and choose Tools.
   (In AEM 6.1) Click on the Experience Manager link, and then click Tools.
2) Select XML Documentation from the list of tools.
3) Click on the Document States tile.
   Assets States page is displayed.
4) Select the document state profile from where you want to delete the document state and click Edit Profile.
5) Delete the document state or state transition and click Done.

Delete a document state profile

To delete a document state profile, perform the following steps:

1) (In AEM 6.4, 6.3, and 6.2) Click on the Adobe Experience Manager link at the top and choose Tools.
   (In AEM 6.1) Click on the Experience Manager link, and then click Tools.
AUTOMATE THE DOCUMENT STATE CHANGE

CHAPTER 5 CONFIGURE DOCUMENT STATES

2) Select XML Documentation from the list of tools.
3) Click on the Document States tile.
   *Assets States page is displayed.*
4) Select the document state profile that you want to delete and click **Delete Profile**.

**Automate the document state change**

If you do not want to manually change the document states, you can create a workflow and automate the document state change.

Perform the following steps to automate the document state change:

1) Open the workflow page from the AEM server URL.
   `<AEM_Server_URL>:<port>/workflow`
2) Open a workflow from the workflow page. For example, Review Topic.
3) Select **Process Step** from the **Workflow** section of the AEM dialog and drag-drop on the workflow.

4) Double-click the process and open the **Step Properties** dialog.
5) Enter the following details in the **Process** tab of the dialog and click OK:
   – Select **Set Document State for any DAM asset** from the Process drop-down.
   – Select the Handler Advance check-box.
   – Enter the name of the document state in the **Arguments** text box.
   **NOTE:** Make sure that you enter the correct document state in the Argument text box. If you enter a wrong name, the document will be set to the wrong document state.

6) Click **Save** to save the workflow.
Migrate existing content

XML Documentation solution allows you to convert a variety of structured and unstructured documents into DITA format. This topic covers the information about uploading your DITA content in AEM repository and converting non-DITA content to DITA format.

**TIP:** See *Content migration and upload* for best practices around using migrating and uploading existing content.

Upload existing DITA content

Most likely you would have a repository of existing DITA content that you would like to use with the XML Documentation solution. For such existing content, you can use any of the following approaches to bulk upload your content into AEM repository.

Use a WebDAV tool

If you are authoring your topics and maps in any other DITA editor, you can use any WebDAV tool to upload your files. The procedure given in this section uses WinSCP as the WebDAV tool to upload content.

Perform the following steps to use WinSCP to upload files:

1) Download and install WinSCP on your computer.
2) Launch the WinSCP app.
   
   *The Login dialog appears.*

3) On the Login dialog, specify a New Site setting by choosing *WebDAV* as the *File Protocol* and providing other connection details such as:
   - the URL where you AEM server is hosted,
   - the port number (default is 4502), and
   - the user name and password to access your AEM server.

4) Click *Login*.

*On a successful connection, you will see the contents of AEM Assets in the WinSCP user interface. You can easily browse, create, update, or delete content using the WinSCP file explorer.*

Use FrameMaker

Adobe FrameMaker comes with a powerful AEM connector that allows you to easily upload your existing DITA and other FrameMaker documents (.book and .fm) into AEM. You can use various file upload functionalities such as uploading a single file, uploading a complete folder with or without dependencies (like content references, cross-references, and graphics).
Perform the following steps to use FrameMaker’s AEM Connector to upload content:


2) Open the **Connection Manager** dialog.

3) Enter the following details to connect to the AEM repository:
   - **Name**: Enter a descriptive name to identify the connection to your AEM server.
   - **Server**: Enter the URL and port number of your AEM server.
   - **User Name/Password**: Enter the user name and password to access AEM server.

4) Click **Connect**.
Once the connection is successfully established, the Assets from AEM repository are displayed in the Repository Manager window.

Right-clicking on any file or folder allows you to perform related operations. For example, if you right-click on a folder, you get options to upload a file, upload file with dependencies, upload an entire folder and so on.

Use curl commands

You can also use curl commands to create a folder in DAM, upload files, and add metadata on the uploaded content.

Create a folder

Run the following command to create a folder in AEM repository:

```
curl --user <username>:<password> --data jcr:primaryType=sling:Folder "<server folder path>"
```

Specify the following parameters to create a folder:

- `<username>:<password>`: Specify the user name and password to access the AEM repository. This user must have the folder creation privileges.
- `jcr:primaryType=sling:Folder`: Specify this parameter as is to create a folder type resource.
- `<server folder path>`: Complete folder path including the name of the new folder that you want to create in AEM repository. For example, if you specify the path as
http://192.168.1.1:4502/content/dam/projects/xml-documentation-solution, then the folder xml-documentation-solution is created within the projects folder in DAM.

Upload a file

Run the following command to upload a file in AEM repository:

curl --user <username>:<password> -T "<local file path>" "<server folder path>"

Specify the following parameters to upload a file:

- <username>:<password>: Specify the user name and password to access the AEM repository. This user must have write privileges on the server folder path.
- local file path: Complete file path on your local system that you want to upload.
- <server folder path>: Complete folder path on the AEM server where you want to upload the file.

Add metadata

Run the following command to add metadata on a file:

curl --user <username>:<password> -F<attribute name>=<value> <metadata node path>

Specify the following parameters to add metadata information:

- <username>:<password>: Specify the user name and password to access the AEM repository. This user must have write privileges on the metadata node path.
- -F<attribute name>=<value>: The <attribute name> is the name of the metadata attribute, such as audience and the <value> could be internal. You can specify multiple attribute name value pairs separated by space.
- <metadata node path>: Complete folder path including the file name and its metadata node. For example, if you specify the path as http://192.168.1.1:4502/content/dam/projects/xml-documentation-solution/intro.xml/jcr:content/metadata, then the specified metadata information is set on intro.xml file.

Migrate non-DITA content

This section guides you through the migration process to get non-DITA documents from the following sources:

- Microsoft Word
- XHTML
- DocBook
- Unstructured FrameMaker
- Any other structured document
Migrate Microsoft Word documents

XML Documentation solution allows you to migrate your existing Word documents (.docx) into DITA topic type documents. You need to specify the input and output folder locations along with other parameters and the document gets converted into DITA document. Depending on the content, you could have a .dita file and a .ditamap file.

To be able to convert a Word document successfully, your document should be well-structured. For example, your document should have a Title, followed by Heading 1, Heading 2, and so on. Each of the headings should have some content in it. If your document is not well-structured, the process might not work as expected.

By default, XML Documentation solution uses the Word-to-DITA (Word2DITA) transformation framework. This transformation depends on the style-to-tag mapping configuration file. To be able to use the Word2DITA transformation successfully, you must consider the following guidelines for preparing your Word document for conversion:

**NOTE:** If you make any changes in the default style-to-tag mapping configuration file, then you must update and use the guidelines confirming to your updated style mapping.

- Ensure that your document starts with a Title; this Title is mapped to the DITA map title. Also, the Title must be followed by some regular content.
- After the Title, there should be Heading 1, Heading 2, and so on. Each Heading must have some content in it. The Headings are converted into new Concept type topics. The hierarchy of the generated topics is as per the Heading levels in the document, for example, Heading 1 will precede Heading 2, and Heading 2 will precede Heading 3 content.
- The document must have at least one Heading type content.
- Ensure that you do not have any grouped images. In case you have grouped images in your document, ungroup all such images.
- Remove all headers and footers
- “List Bullet” and “List Bullet 2” are converted to 1st-level and 2nd-level unordered lists respectively.
- “List Bullet” and “List Number2” are converted to 1st-level and 2nd-level ordered lists respectively.
- All hyper-links are converted to XREF.
- The transformation does not recognize any inline styles and treats such content as normal text. If you want to retain the inline styles, convert them to a Word style. For example, converting bold text to a Word style would typically involve the following steps:
  a) Open the Find and Replace dialog (Ctrl+h).
  b) Bring the cursor in the Find What text box, and click More > Format > Font.
  c) In the Find Font dialog, select the Bold style from the Font Style list and click OK.
  d) Bring the cursor in the Replace With text box, and click More > Format > Style.
  e) Select the style with which you want to replace bold formatted text and click OK.
  f) Click Replace All to replace all bold text with the chosen style.
Perform the following steps to convert your existing Word documents into DITA topic type document:

1) Log into AEM and open the CRXDE Lite mode.
2) Navigate to the default configuration file available at the following location:
   /libs/fmdita/config/w2d_io.xml
3) Create an overlay node of the config folder within the apps node.
4) Navigate to the configuration file available in the apps node:
   /apps/fmdita/config/w2d_io.xml
   The w2d_io.xml file contains the following configurable parameters:
   - In the inputDir element, specify the location of the input folder wherein your source Word documents are available. For example, if your Word documents are stored in a folder named wordtodita in projects folder, then specify the location as:
     /content/dam/projects/wordtodita/
   - In the outputDir element, specify the location of the output folder or keep the default output location to save the converted DITA document. If the specified output folder does not exist on DAM, then the conversion workflow creates the output folder.
   - For the createRev element, specify whether a new version of the converted DITA topic is to be created (true) or not (false).
   - In the s2tMap element, specify the location of the map file that contains mappings for Word document styles to DITA elements. The default mapping is stored in the file located at:
     /etc/fmdita/word2dita/word-builtin-styles-style2tagmap.xml
     NOTE: For more information about the structure of word-builtin-styles-style2tagmap.xml file and how you can customize it, see Style to Tag Mapping in DITA For Publishers User Guide.
5) Save the w2d_io.xml file.
6) After configuring the required parameters in the w2d_io.xml file, log into AEM and open the Assets console.
7) Navigate to the input folder location (wordtodita).
8) Upload the source Word documents into this folder. For information on uploading content on DAM, see Upload existing DITA content.

Using the <config> </config> block, you can define one or multiple blocks of configurations for conversion. The conversion workflow gets executed and the final output in the form of a DITA topic is saved in the location specified in the outputDir element.

Migrate XHTML documents

XML Documentation solution allows you to convert your existing XHTML documents into DITA topic type documents. You need to specify the input and output folder locations along with other parameters and the documents get converted into DITA documents. There is a one-to-one mapping between the HTML files and the resultant DITA files, which means that there is a .dita file created for each .html file in the input directory.
CHAPTER 6  MIGRATE EXISTING CONTENT

To be able to convert a XHTML document successfully, your document should be well-structured. For example, your document should have a Title, followed by Heading 1, Heading 2, and so on. Each of the headings should have some content in it. If your document is not well-structured, the process might not work as expected.

To convert your existing XHTML document into DITA topic, perform the following steps:

1) Log into AEM and open the CRXDE Lite mode.
2) Navigate to the default configuration file available at the following location:
   /libs/fmdita/config/h2d_io.xml
3) Create an overlay node of the config folder within the apps node.
4) Navigate to the configuration file available in the apps node:
   /apps/fmdita/config/h2d_io.xml

The h2d_io.xml file contains the following configurable parameters:
- In the inputDir element, specify the location of your input folder wherein your source XHTML documents are available. For example, if your XHTML documents are stored in a folder named xhtmltodita in projects folder, then specify the location as:
  /content/dam/projects/xhtmltodita/
- In the outputDir element, specify the location of your output folder or keep the default output location. If the specified output folder does not exist on DAM, then the conversion workflow creates the output folder.
- For the createRev element, specify whether a new version of the converted DITA topic is to be created (true) or not (false).

5) Save the h2d_io.xml file.
6) After configuring the required parameters in the w2d_io.xml file, log into AEM and open the Assets console.
7) Navigate to the input folder location (xhtmltodita).
8) Upload the source XHTML documents into this folder. For information on uploading content on DAM, see Upload existing DITA content.

Using the <config> </config> block, you can define one or multiple blocks of configurations for conversion. The conversion workflow gets executed and the final output in the form of a DITA topics is saved in the location specified in the outputDir element.

Migrate DocBooks

XML Documentation solution allows you to convert your existing DocBook documents into DITA documents. You need to specify the input and output folder locations, the files that need to be processed, and whether a new version of the document is required or not. Depending on the content, you could have a .dita file and a .ditamap file.

To convert your existing DocBook documents into DITA format, perform the following steps:

1) Log into AEM and open the CRXDE Lite mode.
2) Navigate to the default configuration file available at the following location:
CHAPTER 6

MIGRATE EXISTING CONTENT

/ports/fmdita/config/doc2dita_io.xml

3) **Create an overlay node of the config folder within the apps node.**

4) **Navigate to the configuration file available in the apps node:**

   /apps/fmdita/config/doc2dita_io.xml

   **The doc2dita_io.xml file contains the following configurable parameters:**

   - **In the inputDir element**, specify the location of your input folder wherein your source DocBook documents are available. For example, if your DocBook documents are stored in a folder named docbooktodita in projects folder, then specify the location as:

     /content/dam/projects/docbooktodita/

   - **In the outputDir element**, specify the location of your output folder or keep the default output location. If the specified output folder does not exist on DAM, then the conversion workflow creates the output folder.

   - **In the inpExt element**, specify the file extensions that need to be picked up for conversion. By default, the XML and DocBook file extensions are picked up for conversion. If your files are saved in any other extension, you can specify those file extensions in this parameter. If you have multiple file extensions, then specify those file extensions separated by a comma.

   - **In the assetExt element**, specify the file extensions of the image files that need to be picked up for processing. By default, the JPG, GIF, EPS, and PNG file extensions are picked up for processing. If your files are saved in any other extension, you can specify those file extensions in this parameter. If you have multiple file extensions, then specify those file extensions separated by a comma.

   - **For the createRev element**, specify whether a new version of the converted DITA topic is to be created (true) or not (false).

5) **Save the doc2dita_io.xml file.**

6) **After configuring the required parameters in the doc2dita_io.xml file, log into AEM and open the Assets console.**

7) **Navigate to the input folder location (docbooktodita).**

8) **Upload the source DocBook documents into this folder. For information on uploading content on DAM, see Upload existing DITA content.**

   Using the <config> </config> block, you can define one or multiple blocks of configurations for conversion. The conversion workflow gets executed and the final output in the form of a DITA topic is saved in the location specified in the outputDir element.

**Migrate unstructured FrameMaker documents**

XML Documentation solution allows you to convert your existing unstructured FrameMaker (.fm and .book) documents into DITA documents. The first step is to create style mappings using FrameMaker and save those settings in a .sts file. Next, if you are using custom DITA, then you can map your custom elements with the source FrameMaker formats in the ditaElems.xml file. For example, if you have created a custom element named impnote to handle all important notes, then you can define this custom element in the ditaElems.xml file. Once this custom element is defined, XML Documentation
solution would not raise an error while converting FrameMaker document containing `impnote` element.

Also, if you want to specify some additional attributes with your custom or valid DITA element, you can define those in the `style2attrMap.xml` file. For example, you can specify the `type` attribute with the value of `important` to be passed on with the `impnote` element. This additional information can be specified in the `style2attrMap.xml` file.

In addition to specifying

To convert your existing unstructured FrameMaker documents into DITA format, perform the following steps:

1) Create style mappings in FrameMaker and save those settings in a `.sts` file.
2) Log into AEM and open the CRXDE Lite mode.
3) If you have custom DITA elements, define those in the `ditaElems.xml` file available at the following location:
   `/libs/fmdita/config/ditaElems.xml`
4) Create an overlay node of the `config` folder within the `apps` node.
5) Navigate to the configuration file available in the `apps` node:
   `/apps/fmdita/config/ditaElems.xml`
   
   The `ditaElems.xml` file contains a single configurable parameter:
   - In the `elem` parameter, specify the name of the custom element that you want to use in your converted DITA documents. This element would be passed on as is in the generated DITA documents.
6) If you want to specify additional attributes, define those in the `style2attrMap.xml` file available at the following location:
   `/libs/fmdita/config/style2attrMap.xml`
7) Create an overlay node of the `config` folder within the `apps` node.
8) Navigate to the configuration file available in the `apps` node:
   `/apps/fmdita/config/style2attrMap.xml`
   
   The `style2attrMap.xml` file contains the following configurable parameters:
   - In the `fmStyle` parameter, specify the source format used in the FrameMaker document that you want to map.
   - In the `ditaAttr` element, specify the DITA attribute that you want to map with the source format.
   - In the `ditaVal` element, specify the value for the mapped attribute. If you don’t have any value, you can leave this entry blank.
9) Save the `style2attrMap.xml` file.
10) After configuring the required parameters in the `style2attrMap.xml` file, log into AEM and open the Assets console.
11) Navigate to and click on the FrameMaker document that you want to convert.

   *The DITA map console appears showing the list of Output Presets available to generate output.*
12) Select DITA output format and configure the required parameters.

**NOTE:** You must use the same settings file (.sts) that you created in FrameMaker. Also, specify the Settings Name and Destination Path.

13) Click the **Generate** icon to start the output generation process.

Using the `<attrMap> </attrMap>` block, you can define one or multiple blocks of configurations for conversion. Depending on the content, you could have a `.dita` file and a `.ditamap` file as the converted files.

**Migrate any other structured document**

XML Documentation solution allows you to convert your existing structured documents into valid DITA documents. You need to specify the input and output folder locations, the location of your transformation file, the extension with which the final output is saved, and whether a new version of the document is required or not.

To convert your existing structured documents into DITA format, perform the following steps:

1) Log into AEM and open the CRXDE Lite mode.
2) Navigate to the default configuration file available at the following location:
   ```/libs/fmdita/config/XSLConfig.xml```
3) Create an overlay node of the `config` folder within the `apps` node.
4) Navigate to the configuration file available in the `apps` node:
   ```/apps/fmdita/config/XSLConfig.xml```

The **XSLConfig.xml** file contains the following configurable parameters:

- In the `inputDir` element, specify the location of your input folder wherein your source structured documents are available. For example, if your structured documents are stored in a folder named `xsltodita` in `projects` folder, then specify the location as:
  ```/content/dam/projects/xsltodita/```
- In the `outputDir` element, specify the location of your output folder or keep the default output location. If the specified output folder does not exist on DAM, then the conversion workflow creates the output folder.
- In the `xslFolder` element, specify the location of the folder where the XSL transformation files are stored.
- In the `xslPath` element, specify the location of the primary `.XSL` file that is used to initiate the conversion process.
- In the `outputExt` element, specify the file extensions of the final output file that is created from the transformation stream.
- For the `createRev` element, specify whether a new version of the converted DITA topic is to be created (`true`) or not (`false`).

5) Save the `XSLConfig.xml` file.
6) After configuring the required parameters in the `XSLConfig.xml` file, log into AEM and open the Assets console.
7) Navigate to the input folder location (xsltodita).
8) Upload the source structured documents into this folder. For information on uploading content on DAM, see *Upload existing DITA content*.

Using the `<config> </config>` block, you can define one or multiple blocks of configurations for conversion. The conversion workflow gets executed and the final output in the form of a DITA topic is saved in the location specified in the `outputDir` element.
Configure templates and conditional tags

XML Documentation solution comes with topic and DITA map templates. You can create your own custom template and share the same with your authors. This topic covers the process of using custom topic and map templates. In addition to the custom templates, your authors would also need tags to organize their content. Learn about the tagging system and how your authors can use these tags in their topics, maps, and other project assets.

Configure custom DITA topic template

The XML Documentation solution comes with four out-of-the-box DITA topic templates, which are:

- Topic
- Task
- Concept
- Reference

You can create topics based on these templates through the Web Editor. Or, you can define your own topic templates that can then be used to create new topics from the Web Editor.

Perform the following steps to add your custom topic templates:

1) Log into AEM and open the CRXDE Lite mode.
2) Navigate to the following location:
   `/libs/fmdita/xmleditor/templates`
   *XML Documentation solution stores the default templates in the above location.*
3) Create an overlay node of the `templates` folder within the `apps` node.
4) You can add your template files in the `apps` node. For example, the location of the templates folder in `apps` node will be:
   `/apps/fmdita/xmleditor/templates`
5) Add the following properties in your topic template file:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>jcr:description</code></td>
<td>A description for your topic template file. This description appears below the template title.</td>
</tr>
<tr>
<td><code>jcr:title</code></td>
<td>A title for your topic template file.</td>
</tr>
<tr>
<td><code>ranking</code></td>
<td>A number specifying the template ranking on the blueprint page. Lower the number, higher is the ranking of the template. For example, the ranking of reference type document is 400 and topic type document is 100; with this ranking the topic file is listed before the reference file.</td>
</tr>
</tbody>
</table>
Next time you create a new topic, your template shows up in the Blueprint page. For more information about creating a DITA topic, see the XML Documentation for Adobe Experience Manager User Guide.

**NOTE:** You can also copy and paste any existing template node and replace the XML content of that template with your custom content. XML content of every template is located in the "jcr:content/renditions/original" node under the template node. The content can be directly edited as text in the CRXDE Lite mode.

**TIP:** See *Custom templates* for best practices around using custom topic templates.

---

**Configure custom DITA map template**

The XML Documentation solution comes with two out-of-the-box map templates - DITA map and Bookmap. You can create maps based on these templates through the Map Editor. Or, you can define your own map templates that can then be used to create new map from the Map Editor.

Perform the following steps to add your custom map templates:

1) Log into AEM and open the CRXDE Lite mode.
2) Navigate to the following location:
   `/libs/fmdita/ditamapeditor/core/templates`
   XML Documentation solution stores the default map templates in the above location.
3) Create an overlay node of the templates folder within the apps node.
4) You can add your template files in the apps node. For example, the location of the templates folder in apps node will be:
   `/apps/fmdita/ditamapeditor/core/templates`
5) Add the following properties in your map template file:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>jcr:description</td>
<td>A description for your map template file. This description appears below the template title.</td>
</tr>
<tr>
<td>jcr:title</td>
<td>A title for your map template file.</td>
</tr>
<tr>
<td>ranking</td>
<td>A number specifying the template ranking on the blueprint page. Lower the number, higher is the ranking of the template. For example, the ranking of bookmark file is 400 and map file is 300; with this ranking the map file is listed before the bookmark file.</td>
</tr>
</tbody>
</table>

Next time you create a new map, your template shows up in the Blueprint page. For more information about creating a DITA map, see the XML Documentation for Adobe Experience Manager User Guide.

**TIP:** See *Custom templates* for best practices around using custom map templates.
Customize Web Editor

XML Documentation solution comes bundled with a powerful Web Editor that allows your authors to create and edit DITA documents. You can customize the Web Editor’s toolbar to add or remove any feature that you can access from the toolbar. Also, you can change the look-and-feel of the document when it is opened in the Web Editor.

Customize toolbar

By default, the Web Editor is shipped with the most common editorial features required by any DITA editor. Features such as inserting elements of type list (numbered or bulleted), cross-reference, content reference, table, paragraph, and character formatting are available in the editor. In addition to these basic elements, you can customize the Web Editor to insert elements that are used in your authoring environment.

There are two ways of customizing the Web Editor’s toolbar:

- Add a new functionality to the toolbar
- Remove any existing functionality from the toolbar

Add a feature in the toolbar

Adding a functionality to the Web Editor involves two primary tasks - adding an icon for the feature in the `ui_config.json` file and adding the background functionality in JavaScript.

Perform the following steps to add a feature to the Web Editor’s toolbar:

1) Log into AEM and open the CRXDE Lite mode.
2) Navigate to the default configuration file available at the following location:
   `/etc/designs/fmdita/clientlibs/xmleditor/ui_config.json`
3) Create a copy of the default configuration file at the following location:
   `/apps/fmdita/xmleditor/ui_config.json`
4) Navigate to and open the `ui_config.json` file in the `apps` node for editing.
5) In the `ui_config.json` file, add the definition of the new feature in the toolbars section. Typically, you can create a new toolbar button group and add one or more toolbar buttons to it. Or, you can add a new toolbar button within an existing toolbar group. The following details are required to create a new toolbar group:

   **type**
   
   Specify `blockGroup` as the `type` value. This value indicates that you are creating a block group that would contain one or more toolbar groups.

   **extraclass**
   
   Name of the class or classes separated with space.
items

Specify the definition of all groups in the toolbar. Each group can contain one or multiple toolbar icons. To define icons within a toolbar group, you need to again define the type attribute within the items, and set its value to buttonGroup. Specify one or more class names in the extraclass property. Specify the feature name in the label property. The following snippet from the ui_config.json file shows the definition for the main toolbar block, followed by the buttonGroup definition:

```
"toolbar": {
  "type": "blockGroup",
  "extraclass": "toolbar operations",
  "items": [
    {
      "type": "buttonGroup",
      "extraclass": "left-controls",
      "label": "Left Controls",
      "items": [
        {
          "type": "buttonGroup",
          "extraclass": "left-controls",
          "label": "Left Controls",
          "items": [

Within the items collection, you need to specify the definition for one or more toolbar icons.

You need to define the following properties to add a toolbar icon:

type

Specify button as the type value. This value indicates that you are adding a toolbar button.

icon

Specify the name of the Coral icon that you want to use in the toolbar.

variant

Specify quiet as the variant value.

title

Specify the tooltip for the icon.

on-click

Specify the command name defined for the feature in the JavaScript file. If your command requires input parameters, then specify the command name as:

"on-click": {"name": "AUTHOR_INSERT_ELEMENT", "args": "simpletable"}

show or hide

If you are defining the show property, then specify the modes in which the icon is displayed. Possible values are - @isAuthorMode, @isSourceMode, @isPreviewMode, true (display in all modes), or false (hide in all modes).
In place of `show`, you can also define the `hide` property. The possible values are same as in `show` property with the only difference that the icon is not displayed for the specified mode.

*The following example shows XML Documentation solution version number when the user clicks on the Show Version icon in the toolbar.*

*Add the following code to a JavaScript file:*

```javascript
$(document).ready(setTimeout(function(){
  fmxml.commandHandler.subscribe({
    key:'user.alert',
    next: function()
    {
      alert("XML Documentation solution version x.x")
    }
  }), 1000))
```

*Add the feature in the ui_config.json file as:*

```
"type": "button",
"icon": "alert","variant": "quiet","title": "About XML Documentation solution","show": "true","on-click": "user.alert"
```

6) Create a `clientlib` folder and add your JavaScript into this folder.

7) Update the categories property of the `clientlib` folder by assigning it the value of `apps.fmdita.xml_editor.page_overrides`.

8) Save the `ui_config.json` file and reload the Web Editor.

**Remove a feature from the toolbar**

At times you might not want to give all features currently available in the Web Editor, in that case you can remove the unwanted feature from the Web Editor’s toolbar.

Perform the following steps to remove any unwanted feature from the toolbar:

1) Log into AEM and open the CRXDE Lite mode.

2) Navigate to the default configuration file available at the following location:.

   `/etc/designs/fmdita/clientlibs/xmleditor/ui_config.json`

3) Create a copy of the default configuration file at the following location:

   `/apps/fmdita/xmleditor/ui_config.json`

4) Navigate to and open the `ui_config.json` file in the `apps` node for editing.

*The `ui_config.json` file has three sections:*

**toolbars**

This section contains the definition of all features available in the editor's toolbar such as Insert/Remove Numbered List, (file) Close, Save, Comments and more.

**shortcuts**

This section contains the definition of keyboard shortcuts assigned to a particular feature in the editor.

**templates**

This section contains the predefined structure of DITA elements that you can use in your document. By default, the templates section contains template definitions for a paragraph, simple table, table,
and body elements. You can create a template definition for any element by adding a valid XML structure for the desired element.

5) From the toolbars section, remove the entry of the feature that you do not want to expose to your users.

6) Save the `ui_config.json` file and reload the Web Editor.

**Customize WYSIWYG view**

In addition to customizing the functionality exposed by the Web Editor, you can also customize the look-and-feel of your document when it is previewed or opened for editing in the Web Editor.

Perform the following steps to customize the rendition of DITA document while previewing or editing it in the Web Editor:

1) Log into AEM and open the CRXDE Lite mode.

2) Create a new node, say `web-editor-custom-css`, of type `cq:ClientLibraryFolder`. This node is used to store the customization files.

3) Add the following property to your node (`web-editor-custom-css`):

<table>
<thead>
<tr>
<th>Property name</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>categories</td>
<td>String</td>
<td>Specify the value as: <code>apps.fmdita.xml_editor.contentOverrides</code></td>
</tr>
</tbody>
</table>

4) Create a new file named `css.txt`.

5) Create a CSS file, say `overrides.css`, which will store the customized element definitions.

**NOTE:** You can customize any element’s definition by using its name and adding custom display attributes to it. For example, to customize a list element use `.li` and add display attributes to it.

6) Enter the name of the file you created in Step 5 (`overrides.css`) in `css.txt`.

7) In the `overrides.css` file, provide your custom settings to display DITA content. In the following example, the topic’s title has been customized to display in red color and text marked as bold in a paragraph appears in green color.

```
.title{
  color:red
}
.p .b{
  color:green
}
```

8) Save all changes and open any topic in Preview or edit mode to see the changes.
CHAPTER 9 CONFIGURE ENTERPRISE OR PRODUCT-SPECIFIC PROFILES

Configure enterprise or product-specific profiles

In an enterprise, different groups or products may use different authoring templates, output templates, and conditional attribute profiles (subject schemes). Configuring these only at an enterprise level can make the author's experience difficult, as they will see templates or profiles that are not relevant to them.

XML Documentation solution allows you to configure authoring templates, output templates, and conditional attribute at an enterprise (global) level as well as at a folder level. This way, you can segregate the configurations for different departments or products in your enterprise.

Also, you can delegate the folder-specific configuration to a department or product administrators to decentralize the administration.

Using the Folder Profiles tile in the XML Documentation settings, you can configure setting under the following tabs:

- **General**: The general tab is available when you are configuring project or product level settings. You can configure settings such as the folder path on which the settings will be applicable and users who will have administrative rights to create or update configurations.

- **Conditional Attributes**: Use this tab to configure conditional attributes at global or enterprise level. A conditional attribute is a combination of the attribute name and value, and you can also define a label for it. The conditional attributes that you define here are available to all users across projects.

- **Authoring Template**: Use this tab to configure the templates that your authors will use to create DITA content. There are six out-of-the-box DITA topic templates that you can use:
  - Reference
  - Task
  - DITAVAL
  - Concept
  - Topic
  - Troubleshooting

  **NOTE:** You also have a Blank DITA template in the list, which can be used to create custom templates. You can use it as the base, make modification to it, and save it with a different name. Once it is added to the global authoring templates set, it becomes available for authoring.

- **Output Preset**: Similar to Authoring Templates, there are five pre-configured output presets:
  - AEM Site
  - PDF
  - HTML5
  - EPUB
  - Custom

  **Publishers have access to these out-of-the-box output presets to publish content. You can make changes to the default output presets or create new presets as per your publishing needs. Once configured, the publishing presets then become available to all publishers in the system.**
Similar to the global profile, you can create folder-specific profiles. In a folder-specific profile, you can define the folder on which the setting will be applicable. These settings include the conditional attributes, authoring templates, and output presets. The conditional presets and authoring templates are then made available to authors who work in the configured folder. Similarly, publishers will have access to the configured output presets defined within the configured folder.

A folder-specific profile overrides the settings configured in the global profile. In other words, if a folder has a specific folder profile, then it will show the authoring templates, output templates, and conditional attribute values configured in its corresponding folder profile. It will not show the values configured in the global profile.

The following sections will walk you through the process of configuring global profile and folder-specific profiles.

Configure global profile

Perform the following steps to configure the global profile:

1) Log into Adobe Experience Manager as an administrator.
2) **(In AEM 6.4, 6.3, and 6.2)** Click on the Adobe Experience Manager link at the top and choose Tools.
   **(In AEM 6.1)** Click on the Experience Manager link, and then click Tools.
3) Select XML Documentation from the list of tools and click the Folder Profiles.
   *For the first time the Folder Profiles page is shown with only the Global Profile tile.*
4) Click on the Global Profile tile.
5) To configure conditional attributes, see Configure conditional attributes.
6) To configure Authoring Template, see Configure authoring templates.
7) To configure output presets, see Configure output presets.
8) After making all required updates, save and close the Global Profile.

Create and configure a folder-specific profile

Perform the following steps to configure a folder-specific profile:

1) Log into Adobe Experience Manager as an administrator.
2) **(In AEM 6.4, 6.3, and 6.2)** Click on the Adobe Experience Manager link at the top and choose Tools.
   **(In AEM 6.1)** Click on the Experience Manager link, and then click Tools.
3) Select XML Documentation from the list of tools and click the Folder Profiles.
   *Folder Profiles page is shown with the default Global Profile.*
4) Click Create.
5) Enter the following details in the Create Folder Profile dialog:
   - Name of the folder profile.
   - Path of the folder where the profile will be applicable.
NOTE: You cannot apply multiple folder profiles on a folder. Ensure that the folder you are selecting here, does not have any other profile applied to it.

6) Click **Create**.

A new tile with the name of the folder profile is created in the Folder Profiles page

7) Click on the folder profile tile to edit.

A General tab with the folder profile's name and configured folder information is shown.

8) Click **Edit** to add users who will have access to modify the folder profile.

**NOTE:** Users that you add here will have the administrative rights to update the conditional attributes, authoring template, and output presets configured for this folder-specific profile.

9) To add a user, select a user from the **Admin Users** drop-down and click **Add**.

**NOTE:** You can add multiple users to the folder profile from the drop-down.

10) After adding all required users to the folder profile, click **Save**.

11) Click **Save**.

Now you are ready to configure the conditional attributes, authoring templates, and output presets.

### Configure conditional attributes

Perform the following steps to configure global or folder-specific conditional attributes:

1) Log into Adobe Experience Manager as an administrator or the user having administrative rights on a folder-specific profile.

2) *(In AEM 6.4, 6.3, and 6.2)* Click on the Adobe Experience Manager link at the top and choose **Tools**.

*(In AEM 6.1)* Click on the Experience Manager link, and then click **Tools**.

3) Select **XML Documentation** from the list of tools and click the profile that you want to configure.

**NOTE:** You can choose to configure the Global Profile or a folder-specific profile.

4) Click on the **Conditional Attributes** tab.

5) Click **Edit**.

6) Click **Add**.

7) Enter the **Name**, **Value**, and a **Label** for the conditional attribute.

*You can save a profile with only the attribute name. However, an attribute can only be used when it has a value specified to it. If you specify both - value and label for an attribute, the Web Editor shows only the value of the attribute. The label is shown to the publishing administrator at the time of creating conditional preset.*

*The following screenshot shows the definition for the platform attribute with value of unix and a label of Red Hat Linux.*
CONFIGURE ENTERPRISE OR PRODUCT-SPECIFIC PROFILES

8) If you want to add more values for the same attribute, click the + icon and enter additional value and label.
9) If you want to add more attributes, click Add.
10) Click Save.

**NOTE:** If you are using custom attributes (other than the default attributes provided in DITA specification) to conditionalize content, then you must explicitly define those in the /libs/fmdita/config/condAttrList.xml file. For making any customizations, create an overlay of the config node in the apps folder and update the condAttrList.xml file in the apps folder.

Configure authoring templates

XML Documentation solution comes with seven out-of-the-box authoring templates. You can choose to have only a few templates available to your authors. In case you use a custom template, the same can be configured and made available for authoring. You can customize the authoring templates from the Authoring Template tab in the Folder Profile.

1) Log into Adobe Experience Manager as an administrator or the user having administrative rights on a folder-specific profile.

2)  
   (In AEM 6.4, 6.3, and 6.2) Click on the Adobe Experience Manager link at the top and choose Tools.
   (In AEM 6.1) Click on the Experience Manager link, and then click Tools.

3) Select XML Documentation from the list of tools and click the profile that you want to configure.
   **NOTE:** You can choose to configure the Global Profile or a folder-specific profile.

4) Click on the Authoring Template tab.

5) Click Edit.
   
   You get the options to add authoring templates by searching from the default location or browsing for it.
NOTE: By default, all templates are stored in the /content/dam/topic-templates folder. You can create templates by navigating to the default template location. For more information about creating custom templates, see Create custom template.

6) Do one of the following to select a template to add:
   – Choose **Search or Type** and enter or select the name of a template from the drop-down list. The drop-down list consists of all default templates and any new template that you have created.

   – Click **Browse** and select a template from DAM.

7) Click **Add**.
The selected template is added to the Templates list.

NOTE: In AEM 6.4 and 6.3 you can change the order of templates by dragging and dropping them at the desired position in the list. The position of templates controls the order in which they show in the Blueprint page in the topic creation workflow.

8) Click **Save**.

In case you have configured the authoring templates on a folder-specific profile, the configured templates get associated with the configured folder. All projects created under the configured folder will have access to only those templates that are configured under the folder-specific profile.

**Create custom authoring template**

XML Documentation solution provides an easy way of creating authoring templates. As a system administrator, you can use the Web Editor to create authoring template from scratch. XML Documentation solution stores the default templates in /content/dam/topic-templates folder. Any new template that you create is added to the default template store. You can then add the new template in the global profile or assign it to a specific folder using the folder-specific profile.

Perform the following steps to create a custom authoring template:

1) Log into Adobe Experience Manager as an administrator.
2) In the Asset console, navigate to the /content/dam/topic-templates location.
3) Click **Create > DITA Template**.
4) On the Blueprint page, select the type of the DITA topic template that you want to create.
   
   NOTE: You can use the Blank template to start from scratch. The Blank template does not have any structure or elements in it.
5) Click **Next**.
6) On the new template Properties page, enter a **Title**, **Name**, and **Description** for the template.
   
   NOTE: The name is automatically suggested based on the Title of your template. If you want to manually specify the name, then ensure that the Name does not contain any spaces, apostrophe, or braces and ends with .dita.
7) *(Optional)* Click the **Add a Thumbnail** button to browser for and select a thumbnail to associate with your template.

8) Click **Create**. The Topic Created message appears.

   *You can choose to open the template for editing in the Web Editor, or save the template file in the default template store location. Once the template is created, you can use the Web Editor to customize the template as per your authoring needs. Once a template is in place, ensure that you associate it either with a global or folder-specific profile.*

**Using an existing authoring template**

You can use an existing authoring template by uploading it in the default templates folder:

```
/content/dam/topic-templates
```

However, to use a custom authoring template, ensure that you disable the WCMMode in your template. This can be done by including the following piece of code in your template such that it loads before any XML Documentation solution components:

```%
<%com.day.cq.wcm.api.WCMMode.DISABLED.toRequest(request);%>
```

This ensures that your template is not rendered with any additional WCM related markup and appears consistent across author and publish instances.

**Upload an existing template using HTTP API**

You can also use the AEM HTTP API to upload an existing template. For details about APIs available to work with AEM Assets, see **Asset HTTP API** documentation in AEM 6.4, 6.3, 6.2, or 6.1. Use the information under **Create an Asset** section to upload your topic template within the default templates folder:

```
/content/dam/topic-templates
```

After uploading the template, use the CRXDE Lite view to access your template file properties and create or configure the following properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>templateDescription</td>
<td>String</td>
<td>A descriptive name for the template. For example, Template for creating user guides.</td>
</tr>
<tr>
<td>templateTitle</td>
<td>String</td>
<td>A title that will appear in the new template blueprint page.</td>
</tr>
<tr>
<td>fmdita-dita-template</td>
<td>Boolean</td>
<td>Set the value as <code>true</code> to create this topic file as a template.</td>
</tr>
<tr>
<td>fmdita-newfile</td>
<td>Boolean</td>
<td>Set the value as <code>true</code>.</td>
</tr>
</tbody>
</table>

**NOTE:** If you want to add a thumbnail for the topic template, it should be uploaded as a dam asset parallel to `jcr:content`. 
Version management

Versioning is an important aspect of any content management system. It allows you to create a snapshot of your digital asset at a specific point in time. With a version of a digital asset in place, you can restore the required version of the asset and update it. Typically, for creating a version of any asset, you would check out and check in the required asset. XML Documentation solution comes with a feature of automatically checking out a file (DITA topic or map) whenever that file is opened in the editor.

Configure file auto-checkout for Web Editor

The XML Documentation solution Web Editor allows you to create and update DITA topics. You can configure the Web Editor to automatically give exclusive write permission on any document that is opened for editing. This ensures that no other writer accidentally overwrites a topic that is opened for editing by another writer. Once a topic is opened for editing, an author can check-in the file by at the time of closing the file.

By default, the file auto-checkout feature is disabled. Perform the following steps to enable file auto-checkout feature:

1) Open the Adobe Experience Manager Web Console Configuration page.  
   The default URL to access the configuration page is:  
   http://<server name>:<port>/system/console/configMgr
2) Search for and click on the com.adobe.fmdita.xmleditor.config.XmlEditorConfig bundle.
3) Select the Auto-Checkout Configuration option.
4) Click Save.

NOTE: Irrespective of whether you turn on or off this feature, the file Check Out and Check In options are always available in a topic preview.
Integrate desktop-based XML editors

There are a lot of XML editors available on the market, and you could be using one already. Adobe FrameMaker is one of the most powerful XML editor, which comes with AEM connector. Using the AEM connector in FrameMaker, you can easily connect with AEM repository, check-out and check-in files, and edit files directly in FrameMaker. You can also configure XML Documentation solution to launch FrameMaker from the Web Editor. Once you have the file opened in FrameMaker, you can edit and check the file back into AEM repository.

Enable file editing in FrameMaker from the Web Editor

You can use FrameMaker or any other DITA editor to create and update DITA content. However, if your organization uses FrameMaker as DITA editor, then you can give your users an option to open DITA documents in FrameMaker from AEM.

By default, your users do not see the Open in FrameMaker button on the AEM toolbar. Perform the following steps to add this button on the AEM toolbar:

1) Open the Adobe Experience Manager Web Console Configuration page. The default URL to access the configuration page is:
   http://<server name>:<port>/system/console/configMgr
2) Search for and click on the com.adobe.fmdita.xmleditor.config.XmlEditorConfig bundle.
3) Select the Show Open in FrameMaker Button option.
4) Click Save.
Configure output generation settings

XML Documentation solution comes with a lot of configuration options for you to customize the output generation process. You can configure the default output presets that can be made available to all publishers in your organization or you can customize your AEM site output template. This topic covers all configurations and customizations that would help you set up your output generation process.

Configure output presets

In a typical enterprise setup, different output templates could be used for different products or user guides. Also, there could be some common output generation processes that should be used by all publishers and a set of specific output generation processes for a specific group of publishers or projects.

XML Documentation solution allows the administrator to create output presets with specific settings that can then be used by all or a specific set of publishers to generate output. For example, the administrator can create one output preset to generate user guides that is common across all publishers. And, another one to create the programming user manuals that is specific to a set of publishers. Both of these presets can be configured to use different output templates. In this example, the common publishing preset for generating the user guide can be configured at the global level. And, the output preset for generating programming user manual can be configured at a folder-level.

Once the default output presets have been created in the system, all DITA maps created after that will use the default presets to generate output. However, all existing DITA maps would continue to use the output presets that were earlier configured with them. If you want to apply the new output preset on all existing DITA maps, then you need to run the Apply preset changes workflow.

In addition to the presets configured at the global or enterprise level, publisher would still have the rights to create more output presets. However, those presets are tied to the DITA map for which they are created. For more details about creating regular output presets for a DITA map, see Create, edit, duplicate, or remove an output preset in XML Documentation for Adobe Experience Manager User Guide.

Perform the following steps to configure global or folder-specific output presets:

1) Log into Adobe Experience Manager as an administrator or the user having administrative rights on a folder-specific profile.

2) (In AEM 6.4, 6.3, and 6.2) Click on the Adobe Experience Manager link at the top and choose Tools. (In AEM 6.1) Click on the Experience Manager link, and then click Tools.

3) Select XML Documentation from the list of tools and click the profile that you want to configure. NOTE: You can choose to configure the Global Profile or a folder-specific profile.

4) Click on the Output Presets tab.

A list of out-of-the-box output presets is displayed, which includes AEM Site, PDF, HTML5, EPUB, and CUSTOM.

5) Do one of the following to create or edit an output preset:

   – Click Create to create a new output preset from scratch.
Click Duplicate to create a copy of the selected output preset. You can make changes to the duplicate preset and save it. Click Edit to open the selected preset’s configuration for editing. For information about output preset settings, see Understanding the output presets in XML Documentation for Adobe Experience Manager User Guide.

6) Click Save to save the preset settings.

All DITA maps created or uploaded after this will have the new or updated output preset.

NOTE: You can also configure the presets using the presets.json file which is available in the /libs/fmdita/config folder. For making any customizations, create an overlay of the config node in the apps folder and update the presets.json file in the apps folder.

Apply preset changes

A new output preset created at the global level is made available to all new DITA maps that you create going forward. Similarly, if a new output preset is created at a folder-level, then that preset is made available to all maps that will be created in the configured folder. By default, a new output preset is not made available to any existing DITA map.

If you have updated and existing output preset, or you want to make a new output preset available to existing DITA maps, then perform the following steps:
1) Select the output preset that you want to apply to existing DITA maps.
2) Click Apply Preset Changes.
3) In the Apply Preset Changes dialog, you can choose from:
   - Selecting Overwrite Existing Preset option: If you select this option, then any updates that you made in the existing output presets will overwrite settings in all existing DITA maps where that preset is used. However, doing so will result in loss of any existing conditional preset and baseline information associated with the map.
   - Not selecting Overwrite Existing Preset option: If you do not select this option, then any updates that you made in the existing output presets will not impact the existing DITA maps. However, newly created DITA map will get the updated output preset.
4) Click OK to apply changes from the selected output presets on all existing DITA maps.

Configure FrameMaker Publishing Server

You can use FrameMaker Publishing Server (FMPS) to generate output for your DITA content. Configuring FMPS will allow you to generate output in multiple formats supported by FMPS.

NOTE: To generate output using FMPS, you need to have the FMPS server setup. For installation and configuration details, see the FrameMaker Publishing Server User Guide.

To configure your XML Documentation solution to use FMPS, update the following properties of the com.adobe.fmdita.config.ConfigManager bundle in the Web Console.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FrameMaker Publishing Server Login Domain</td>
<td>Specify the domain name or the workgroup name on which the FrameMaker Publishing Server is hosted.</td>
</tr>
<tr>
<td>FrameMaker Publishing Server Username and Password</td>
<td>Specify the user name and password to access the FrameMaker Publishing Server.</td>
</tr>
<tr>
<td>FMPS Timeout</td>
<td><em>(Optional)</em> Specify the time (in seconds) for which the XML Documentation solution waits for a response from the FrameMaker Publishing Server. If no response is received in the specified time, XML Documentation solution terminates the publishing task and the task is flagged as failed. Default value: 300 seconds (5 minutes)*</td>
</tr>
<tr>
<td>External AEM URL</td>
<td><em>(Optional)</em> The AEM URL where the FrameMaker Publishing Server will place the generated output files. For example, <code>http://&lt;server-name&gt;:4502</code>.</td>
</tr>
<tr>
<td>AEM Admin Username and Password</td>
<td><em>(Optional)</em> The user name and password for an administrator of your AEM setup. This will be used by FrameMaker Publishing Server to communicate with AEM.</td>
</tr>
</tbody>
</table>

**Configure blended publishing within an existing AEM Site**

If you have an AEM Site that contains DITA content, you can configure your AEM Site output to publish DITA content to a predefined location within your site. For example, in the following screenshot of an AEM Site page, the `ditacontent` node is reserved to store DITA content:
CHAPTER 12 CONFIGURE OUTPUT GENERATION SETTINGS

The remaining nodes in the page are authored directly from the AEM Site editor. Configuring the publish setting to publish DITA content to a predefined location ensures that none of your existing non-DITA content gets modified by the XML Documentation solution publishing process.

You need to perform the following configurations on your existing site to allow publishing of DITA content to a predefined node:

- Configure your site's template properties
- Add nodes in your site to publish DITA content

Perform the following steps to configure your existing site's template properties:

1) Log into AEM and open the CRXDE Lite mode.
2) Navigate to your site's template configuration node. For example, the XML Documentation solution stores the default template configurations in the following node:

   /libs/fmdita/config/templates/default

   **NOTE:** Do not make any customizations in the default configuration files available in the `libs` node. You must create an overlay of the `libs` node in the `apps` node and update the required files in the `apps` node only.

3) Add the following properties:

<table>
<thead>
<tr>
<th>Property name</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>topicContentNode</td>
<td>String</td>
<td>Specify the node name where you would like to publish the DITA content. For example, the default node where XML Documentation solution publishes DITA content is: <code>jcr:content/contentnode</code></td>
</tr>
</tbody>
</table>
The following screenshot shows the properties added in the default template node of XML Documentation solution:

![Screenshot of CRXDE Lite with properties added](image)

Next time when you publish any DITA content using your site's template configurations, the content gets published into the nodes specified in the `topicContentNode` and `topicHeadNode` properties.

However, for existing sites, you must manually add the `topicContentNode` and `topicHeadNode` nodes.

Perform the following steps to add the required nodes to your existing site:

1) Log into AEM and open the CRXDE Lite mode.
2) Locate `jcr:content` within your site node.
3) Add `topicContentNode` and `topicHeadNode` nodes with the same name that you specified in the site's template configurations.

### Customize AEM Site output design template

The XML Documentation solution supports creating outputs in following formats:

- AEM Site
- PDF
- HTML5
- EPUB
- Custom output through DITA-OT
For the AEM Site output, you can assign different design templates with different output tasks. These design templates can render the DITA content in different layouts. For example, you could specify different design templates for internal and external audiences.

You can also use customized DITA Open Toolkit (DITA-OT) plug-ins with the XML Documentation solution. You can upload these custom DITA-OT plug-ins to generate PDF output in a specific way.

**TIP:** See *AEM Site publishing* for best practices around creating AEM Site output.

### Customize design template for generating output

The XML Documentation solution uses a set of predefined design templates to generate AEM Site output. You can customize the XML Documentation solution design templates to generate the output that conforms to your corporate branding. A design template is a collection of various styles (CSS), scripts (both server- and client-side), resources (images, logos, and other assets), and JCR nodes that tie all these resources together. A design template can be as simple as a single server-side script with just a couple of JCR nodes, or a complex combination of styles, resources, and JCR nodes. Design templates are used by XML Documentation solution’s publishing subsystem while generating AEM Site output and they control the structure, look and feel of the generated output.

There is no restriction as to where the design template resources should be located on the server, but they are usually logically organized as per their function. For example, the default template has all its JavaScript and CSS files stored under `/etc/designs/fmdita/clientlibs/siteoutput/default` folder. Wherever these files are located, they are linked together by a collection of JCR nodes. Together, these JCR nodes and the files constitute the whole design template.

The default design template shipped with the XML Documentation solution allows you to customize the landing, topic, and search page components. You can make a copy of the default design and the corresponding reference templates and specify different components to generate the desired output.

Perform the following steps to specify your own design template to use for AEM Site output generation:

1. Log into AEM and open the CRXDE Lite mode.
2. Navigate to the default design template node. The location of the default design template node is:
NOTE: Make a copy of the default design templates from the `libs` folder to the `apps` folder and make changes in the `apps` folder. You must also make changes in the templates referenced from the default template node. The referenced templates are placed under `/libs/fmdita/templates/default/cqtemplates` node. Make a copy of the referenced templates in the `apps` folder before making any changes.

3) Click the `default` component in the `templates` node to access its properties.

The XML Documentation solution design template properties are described in the following table.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>landingPageTemplate,</td>
<td>Specify the <code>cq:Template</code> node for these corresponding pages (landing, search, and topic). By default the <code>cq:Template</code> node for these pages can be found in <code>/libs/fmdita/templates/default/cqtemplates</code> node. This node defines the structure and properties of the landing, search, and topic pages.</td>
</tr>
<tr>
<td>searchPageTemplate,</td>
<td></td>
</tr>
<tr>
<td>topicPageTemplate</td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>A descriptive name of your design template.</td>
</tr>
<tr>
<td>topicContentNode</td>
<td>Specify the location of the node that will contain the DITA content in a topic page. Path is relative to the topic page.</td>
</tr>
<tr>
<td>topicHeadNode</td>
<td>Specify the location of the node that will contain the head values (or metadata) derived from the DITA content. Path is relative to topic page.</td>
</tr>
</tbody>
</table>
After creating a custom design template node, you must update the Design option in the AEM Site output presets to use the custom design template node.

For more information, see Creating your First Adobe Experience Manager 6.3 website and The Basics of developing your own website on AEM.

**Customize DITA element mapping with AEM components**

DITA elements in the XML Documentation solution are mapped to their corresponding AEM components. The XML Documentation solution uses this mapping in workflows such as publishing and review to convert DITA element to a corresponding AEM component. The mapping is defined in the `elementmapping.xml` file, which can be accessed from the CRXDE Lite mode. Access the following URL in the CRXDE Lite mode:

/\[libs/fmdita/config/elementmapping.xml\]

**NOTE:** Do not make any customizations in the default configuration files available in the `libs` node. You must create an overlay of the `libs` node in the `apps` node and update the required files in the `apps` node only.

You may use the predefined DITA element mappings, or you can map DITA elements to your custom AEM components. To use your custom AEM components, you need to understand the structure of the `elementmapping.xml` file.

**elementmapping.xml structure**

A high-level overview of the `elementmapping.xml` structure is explained below:

1) Every DITA element is first searched for a corresponding component mapping based on the element name. For example:

```
<ditaelement>
   <name>substeps</name>
   <class>- topic/ol task/substeps</class>
   <componentpath>dita/components/ditaolist</componentpath>
   <type>COMPOSITE</type>
   <target>para</target>
</ditaelement>
```
In the above example, all substeps DITA elements are rendered using the dita/components/ditaolist component.

2) If a DITA element does not find a match based on the name, then a match on the basis of the class is done. For example:

   <ditaelement>
     <name>topic</name>
     <class>- topic/topic</class>
     <componentpath>fmdita/components/dita/topic</componentpath>
     <type>COMPOSITE</type>
     <target>para</target>
     <attributemap>
       <attribute from="id" to="id" />
     </attributemap>
   </ditaelement>

   In the above example, if there is no mapping defined for the task element, then the task element is mapped to the above component because task is inherited from the topic component.

3) When an element has a corresponding component mapping, then further processing of its child elements is determined by type. For example:

   <ditaelement>
     <name>title</name>
     <class>- topic/title</class>
     <componentpath>foundation/components/title</componentpath>
     <type>STANDALONE</type>
     <target>para</target>
     <textprop>jcr:title</textprop>
   </ditaelement>

   type takes the following values:
   – COMPOSITE: element to component mapping continues for child elements as well.
   – STANDALONE: child elements of the current element are not mapped further.

   In the above example, if the <title> element has any child elements, they will not be mapped to any other component. The component for <title> element is responsible for rendering all child elements inside the <title> element.

4) If there are multiple components mapped to a single DITA element, then the best match for the element is selected. To select the best match component, domain and structural specialization of DITA elements is considered.

   If there are DITA elements with domain specialization and a component is mapped for domain specialization, then that component is given high priority.

   Similarly, if there are DITA elements with structural specialization and a component is mapped for structural specialization, then that component is given high priority.

5) You can use <attributemap> in element mapping to map attribute values to the corresponding node properties.
6) `textprop` can be used for serializing the text content of a DITA element to a node property. In addition, it can be used multiple times in an element tag to serialize the text content at multiple locations in published hierarchy. You can also customize the location and name of the target property. For example:

```xml
<ditaelement>
  <name>title</name>
  <class>- topic/title</class>
  <componentpath>foundation/components/title</componentpath>
  <type>STANDALONE</type>
  <target>para</target>
  <textprop>jcr:title</textprop>
</ditaelement>
```

The above element mapping specifies that the text content of `<title>` element will be saved as value of a property named `jcr:title` on the output node.

7) `xmlprop` can be used for serializing the entire XML for a given element to a node property. The component can then read this node property and do custom rendering. For example:

```xml
<ditaelement>
  <name>svg-container</name>
  <class>+ topic/foreign svg-d/svg-container</class>
  <componentpath>fmdita/components/dita/svg</componentpath>
  <type>STANDALONE</type>
  <target>para</target>
  <xmlprop>data</xmlprop>
</ditaelement>
```

The above element mapping specifies that the entire XML markup for element `<svg-container>` will be saved as value of a property named `data` on the output node.

8) There is a special attribute mapping to handle path resolution in output generation process. For example:

```xml
<attributemap>
  <attribute from="href" to="fileReference" ispath="true" rel="source" />
  <attribute from="height" to="height" />
  <attribute from="width" to="width" />
</attributemap>
```

For the above `attributemap`, the `href` attribute in your DITA element will be mapped to a node property named `fileReference`. Now since `ispath` is set to `true`, the output generation process resolves this path and then sets it in `fileReference` node property.

How this resolution happens is determined on the basis of value of the `rel` attribute in attribute mapping.

- If `rel=source`, then value of `href` is resolved with respect to the DITA source file that is currently being processed. The value of `href` is resolved and placed in the value of `fileReference` property.
– If rel=target, then value of href is resolved with respect to the root publish location. The value of href is resolved and placed in the value of fileReference property.

*If you do not want any pre-processing or resolution to happen on path attributes, then you need not specify the ispath attribute. The value is copied as is and the component can do the required resolution.*

### DITA element schema

Following is an example of the DITA element schema in `elementmapping.xml` file:

```xml
<ditaelement>
  <name>element_name</name>
  <class>element_class</class>
  <componentpath>fmdita/components/dita/component_name</componentpath>
  <type>COMPOSITE|STANDALONE</type>
  <attributeprop>propname_a</attributeprop>
  <textprop>propname_t</textprop>
  <xmlprop>propname_x</xmlprop>
  <xpath>xpath expression string</xpath>
  <target>head|para</target>
  <wrapelement>div</wrapelement>
  <wrapclass>class_name</wrapclass>
  <attributemap>
    <attribute from="attrname" to="propname" ispath="true|false" rel="source|target" />
  </attributemap>
  <skip>true|false</skip>
</ditaelement>
```

The following table describes the elements in the DITA element schema:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;ditaelement&gt;</code></td>
<td>The top level node for each mapping element.</td>
</tr>
</tbody>
</table>
| `<class>`    | The class attribute of the target DITA element for which you are writing the component. For example, the class attribute for the DITA topic is: 
  - topic/topic |
| `<componentpath>` | The CRXDE path of the mapped AEM component.                                 |
| `<type>`     | Possible values:                                                            |
  - **COMPOSITE**: Process child elements as well                                   |
  - **STANDALONE**: Skips processing of child elements                             |
## Chapter 12 Configure Output Generation Settings

### Additional notes
- If you plan to override the default element mapping, it is recommended that you do not make the changes in the default `elementmapping.xml` file. You should create a new mapping XML file and place the file at another location, preferably inside custom `apps` folder that you create.
- If you are planning to override some (and not all) of the element mappings, you do not have to replicate the entire `elementmapping.xml` file. You need to create a new XML mapping file and define only the elements that you are overriding.
- After you have created the XML file in the custom location, update the Override Element Mapping setting in the `com.adobe.fmdita.config.ConfigManager` bundle.

### Element Description

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;attributeprop&gt;</code></td>
<td>Used for mapping serialized DITA attributes and values to AEM nodes as property. For example, if you have <code>&lt;note type=&quot;Caution&quot;&gt;</code> element and the component that is mapped for this element has <code>&lt;attributeprop&gt;attr_t&lt;/attributeprop&gt;</code>, then the node's attribute and value is serialized to <code>attr_t</code> property of the corresponding AEM node (<code>attr_t-&gt;type=&quot;caution&quot;</code>).</td>
</tr>
<tr>
<td><code>&lt;textprop&gt;propname_t&lt;/textprop&gt;</code></td>
<td>Save the <code>getTextContent()</code> output to property defined by <code>propname_t</code>.</td>
</tr>
<tr>
<td><code>&lt;xmlprop&gt;propname_x&lt;/xmlprop&gt;</code></td>
<td>Save serialized XML of this node to property defined by <code>propname_x</code>.</td>
</tr>
<tr>
<td><code>&lt;xpath&gt;</code></td>
<td>If XPath element is provided in the element mapping, then along with element name and class the XPath condition should also be satisfied for the component mapping to be used.</td>
</tr>
<tr>
<td><code>&lt;target&gt;</code></td>
<td>Place for the DITA element in the <code>crx</code> repository at specified location. Possible values:</td>
</tr>
<tr>
<td></td>
<td>• <strong>head</strong>: Under the head node</td>
</tr>
<tr>
<td></td>
<td>• <strong>text</strong>: Under the paragraph node</td>
</tr>
<tr>
<td><code>&lt;wrapelement&gt;</code></td>
<td>The HTML element to wrap the contents within.</td>
</tr>
<tr>
<td><code>&lt;wrapclass&gt;</code></td>
<td>The element value to the property <code>wrapclass</code>.</td>
</tr>
<tr>
<td><code>&lt;attributemap&gt;</code></td>
<td>Container node containing one or more <code>&lt;attribute&gt;</code> nodes.</td>
</tr>
<tr>
<td><code>&lt;attribute</code></td>
<td>Maps the DITA attributes to AEM properties:</td>
</tr>
<tr>
<td><code>from=&quot;attrname&quot;</code></td>
<td>• <strong>from</strong>: DITA attribute name</td>
</tr>
<tr>
<td><code>to=&quot;propname&quot;</code></td>
<td>• <strong>to</strong>: AEM component property name</td>
</tr>
<tr>
<td>`ispath=&quot;true</td>
<td>false&quot;`</td>
</tr>
<tr>
<td>`rel=&quot;source</td>
<td>target&quot;`</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE</strong>: If <code>attrname</code> starts with <code>%</code>, then map <code>attrname</code> minus '%' to <code>prop 'propname'</code>.</td>
</tr>
</tbody>
</table>
Customize DITA map console

XML Documentation solution gives you the flexibility of extending the capabilities of the DITA map console. For example, if you have a set of reports that are different from what is available in XML Documentation solution, you can add such reports to the map console. To customize the map console, you need to create an AEM Client Library (or ClientLib) that will contain the code to perform the functionality that you need.

**NOTE:** Direct modifications to page components is not recommended, as it will get overwritten by new releases of the product.

XML Documentation solution provides the `apps.fmdita.dashboard-extn` category for customizing map console. Whenever the map console is loaded, the functionality created under the `apps.fmdita.dashboard-extn` category gets executed and loaded.

**NOTE:** For more information about creating AEM Client Library, see Using Client-Side Libraries.

Handle image rendition during output generation

AEM comes with a set of default workflows and media handles to process assets. In AEM, there are pre-defined workflows to handle asset processing for the most common MIME types. Typically, for every image that you upload, AEM creates multiple renditions of the same in binary format. These renditions may be of different size, with a different resolution, with an added watermark, or some other changed characteristic. For more information about how AEM handles assets, see Processing Assets Using Media Handlers and Workflows in AEM 6.4, 6.3, 6.2, and 6.1.

XML Documentation solution allows you to configure which image rendition to use at the time of generating output for your documents. For example, you can choose from one of the default image rendition or create one and use the same to publish your documents. Image rendition mapping for publishing your documents is stored in the `/libs/fmdita/config/renditionmap.xml` file. A snippet of `renditionmap.xml` file is as follows:

```xml
<renditionmap>    <mapelement>        <mimetype>image/png</mimetype>        <rendition output="AEMSITE">cq5dam.web.1280.1280.jpeg</rendition>        <rendition output="PDF">original</rendition>        <rendition output="HTML5">cq5dam.web.1280.1280.jpeg</rendition>        <rendition output="EPUB">cq5dam.web.1280.1280.jpeg</rendition>        <rendition output="CUSTOM">cq5dam.web.1280.1280.jpeg</rendition>    </mapelement>
...</n</renditionmap>
```

The `mimetype` element specifies the MIME type of the file format. The `rendition output` element specifies the type of output format and the name of rendition (for example, `cq5dam.web.1280.1280.jpeg`) that should be used for publishing the specified output. You can specify the image renditions to use for all supported output formats - AEMSITE, PDF, HTML5, EPUB, and CUSTOM.
If the specified rendition is not present, then XML Documentation solution publishing process first looks for the web rendition of the given image. If even the web rendition is not found, then the original rendition of the image is used.

**NOTE:** These image renditions control only the output generation. An image's web rendition is used when you open a document for preview or review.

### Change the recently generated outputs list limit

You can change the maximum number of generated outputs that are displayed in the Outputs tab for a DITA map. By default, a list of last 25 generated outputs is shown. To change the number of outputs to display in the list, update the **Outputs List Limit** setting in the `com.adobe.outputs.limit` setting in the `com.adobe.fmdita.config.ConfigManager` bundle.

**TIP:** See *Output history* for best practices around working with output history.

### Output generation performance optimization

XML Documentation solution allows you to configure the output generation processes pool size that controls the number of output generation processes that run concurrently. By default, the process pool size is set to number of processing cores available in your system plus one. You might want to change this value to 1 in case you want sequential publishing. In this case, the first publishing task gets executed and the next publishing task is stored in the publishing queue.

To change the output generation processing pool size, update the **Generation Pool Size** setting in the `com.adobe.fmdita.publish.manager.PublishThreadManagerImpl` bundle.

**RELATED LINKS:**
Configure and customize workflows

Workflows enable you to automate Adobe Experience Manager (AEM) activities. A workflow consists of a series of steps that are executed in a specific order. You can define a distinct activity to execute on each step. For example, you can send an email notification to all reviewers in a group when a topic review is created. Or, send a notification to the publisher when an output generation task completes.

For more information about workflows in AEM, see:
- Improving the performance of workflows that use significant server resources: Concurrent Workflow Processing in AEM 6.4, 6.3, 6.2, and 6.1.

The sections in this topic will walk you through various customizations that you can make in the default workflows shipped in XML Documentation solution.

**TIP:** See *Workflow offloading* for best practices around offloading workflow.

Customize review workflow

Every organization’s content authoring team works in a specific way to meet their business requirements. In some organizations there is a dedicated editor, whereas some other organization could have automated editorial review system in place. For example, in an organization a typical authoring and publishing workflow could include tasks like - whenever an author is done with authoring content, it automatically goes to the reviewers, and when the review is complete it goes to the publisher for generating the final output. In AEM, activities that you do on your content and assets can be combined in the form of a process and mapped to an AEM workflow. For more information about workflows in AEM, see Administering Workflows in AEM 6.4, 6.3, 6.2, and 6.1.

XML Documentation solution allows you to customize the default review workflow. You can use the following four custom review-related processes with your other authoring or publishing workflows.

- **Create Review:** This process prepares the metadata required to create a review task. For example, it will assign review permission to the reviewers, set the status of the topics to under review, set the review timelines, and more. Out of the four processes, this is the only mandatory process that must be included in your custom workflow. In your workflow, you may choose to include or exclude the other three processes.

- **Assign Review Task:** This process creates the review task and sends the task notification to the initiator and reviewers.

- **Send Review Email:** This process sends the review email to the initiator and reviewers.

- **Schedule Job to Close Review:** This process ensures that the review process completes on reaching the deadline.
When you are creating a custom review workflow, the first task is to set the required metadata required by the Create Review process. To do so, you can create an ECMA script. A sample of the ECMA script that assigns the metadata is given below:

```javascript
var workflowdata=workItem.getWorkflowData();
workflowdata.getMetaDataMap().put("initiator","admin");
workflowdata.getMetaDataMap().put("operation","AEM_REVIEW");
workflowdata.getMetaDataMap().put("orgTopics","/content/dam/xml-solution/review.xml");
workflowdata.getMetaDataMap().put("payloadJson","{"base":="/content/dam/xml-solution\"","asset":[/"content/dam/xml-solution/review.xml\"],"referrer":"\""\"}"");
workflowdata.getMetaDataMap().put("deadline","2017-06-27T13:19:00.000+05:30");
workflowdata.getMetaDataMap().put("title","Review through custom workflow");
workflowdata.getMetaDataMap().put("description","Initiate this review process using the AEM workflow");
workflowdata.getMetaDataMap().put("assignee","user-one","user-two");
workflowdata.getMetaDataMap().put("status","1");
workflowdata.getMetaDataMap().put("projectPath","/content/projects/review");
workflowdata.getMetaDataMap().put("startTime",System.currentTimeMillis());
```

You can create this script in the `/etc/workflows/scripts` node. The following table describes the properties being assigned by this ECMA script:

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>initiator</td>
<td>String</td>
<td>User ID of the user initiating the review task.</td>
</tr>
<tr>
<td>operation</td>
<td>String</td>
<td>A static value set as AEM_REVIEW.</td>
</tr>
<tr>
<td>orgTopics</td>
<td>String</td>
<td>Path of the topics being shared for review. Specify multiple topics separated by comma.</td>
</tr>
<tr>
<td>payloadJson</td>
<td>JSON object</td>
<td>Specify the following values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• base: path of the parent folder containing the topic sent for review.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• asset: path of the topic sent for review.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• referrer: leave it blank.</td>
</tr>
<tr>
<td>deadline</td>
<td>String</td>
<td>Specify the time in <code>yyyy-MM-dd'T'HH:mm:ss.SSSXXX</code> format.</td>
</tr>
<tr>
<td>title</td>
<td>String</td>
<td>Enter a title for the review task.</td>
</tr>
</tbody>
</table>
Once you have created the script, call it before calling the Create Review process in your workflow. Then, depending on your requirements, you can call the other review workflow processes.

**Customize email templates**

A number of the XML Documentation solution workflows make use of email notifications. For example, if you initiate a review task, an email notification is sent to the reviewers. However, to ensure that the email notification is sent, you have to enable this functionality in AEM. To enable email notification in AEM, see the article Configuring Email Notification in AEM documentation.

The XML Documentation solution contains a set of email templates that you can customize. Perform the following steps to customize these templates:

1) Log into AEM and open the CRXDE Lite mode.
2) In the Navigator tab, go to the following location: 
   `/libs/fmdita/mail`
   
   **NOTE:** Do not make any customizations in the default configuration files available in the `libs` node. You must create an overlay of the `libs` node in the `apps` node and update the required files in the `apps` node only.

3) The mail folder contains the following customizable templates:

<table>
<thead>
<tr>
<th>Template Filename</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>closereview.html</td>
<td>This email template is used when a review task is closed.</td>
</tr>
<tr>
<td>createreview.html</td>
<td>This email template is used when a new review task is created.</td>
</tr>
<tr>
<td>reviewapproval.css</td>
<td>This CSS file contains the styling of email templates.</td>
</tr>
</tbody>
</table>

**Customize post-output generation workflow**

XML Documentation solution gives you the flexibility to specify a post-output generation workflow. You can perform some post-processing tasks on the output that gets generated using the XML Documentation solution. For example, you might want to apply some CQ tags on the generated AEM Site output.
set certain properties on the PDF output, or you might want to send an email to a set of users once the output is generated.

You can create a new workflow model to use as a post-output generation workflow. When a post-output generation workflow is triggered, the output generation workflow shares contextual information through the workflow metadata map, which you can use to perform processing on the generated output. The following table describes the contextual information shared as metadata:

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>outputName</td>
<td>String</td>
<td>Name of the output preset used to generate the output.</td>
</tr>
<tr>
<td>generatedPath</td>
<td>String</td>
<td>Path in DAM where the generated output is stored.</td>
</tr>
<tr>
<td>outputType</td>
<td>com.adobe.fmd.ita.output.OutputType</td>
<td>Type of the output preset.</td>
</tr>
<tr>
<td>outputTitle</td>
<td>String</td>
<td>Title of the output preset.</td>
</tr>
<tr>
<td>outputHistoryPath</td>
<td>String</td>
<td>Repository path of the history node.</td>
</tr>
<tr>
<td>isSuccess</td>
<td>Boolean</td>
<td>A flag depicting the final status of the output generation process - success or failure.</td>
</tr>
<tr>
<td>logPath</td>
<td>String</td>
<td>Path in DAM where the output generation logs are saved.</td>
</tr>
<tr>
<td>generatedTime</td>
<td>Long</td>
<td>Time at which the output generation process was triggered.</td>
</tr>
<tr>
<td>initiator</td>
<td>String</td>
<td>The user ID of the user who triggered the output generation workflow.</td>
</tr>
</tbody>
</table>

To make use of the output generation metadata, you can create an ECMA script or an OSGi bundle. A sample of the ECMA script that uses the metadata is given below:

**NOTE: You can create this script in the `/etc/workflows/scripts` node.**

```javascript
var session = workflowSession.getSession(); // Obtain session object to read/write the repository.
var payload = workItem.getWorkflowData().getPayload().toString(); // Get the workflow payload (the ditamap file on which the generation was triggered)
var metadata = workItem.getWorkflowData().getMetaDataMap(); // Get the workflow metadata object
var generatedPath = metadata.get("generatedPath"); // supplied by XML Documentation solution
var username = metadata.get("initiator"); // supplied by XML Documentation solution
```
var successful = metadata.get("isSuccess"); // supplied by XML Documentation solution
var title = metadata.get("outputTitle"); // supplied by XML Documentation solution
var subject = "Output Generation Finished";
var message = "Generation of output " + title + " just finished " +
(successful ? "successfully. " : "unsuccessfully. ");
message += "It was triggered by " + username;
if (successful) {
message += "<br/><br/>The path to the generated output is " +
generatedPath;
}
/*
MailerAPI.sendMail("dl-docs-authors", subject, message);
*/

Once you have created the script, call the custom script in your workflow. Then, depending on your requirements, you can call the other workflow processes. Once you have designed your custom workflow, call the Finalize Post Generation as the last step in your workflow process. The Finalize Post Generation step ensures that the status of the output generation task gets updated to Finished on completion of the output generation process. After creating a custom post-output generation workflow, you can configure it with any of your output generation presets. Select the required workflow in the Run Post Generation Workflow property of the required preset. When you run an output generation task using the configured output preset, the task status (in the Output tab) changes to Post-Processing.

Configure post-processing XML workflow launcher

XML Documentation solution creates a bunch of workflows that allow you to work with DITA content in AEM. For example, there are workflows that get executed when you upload DITA content or update existing content. These workflows parse DITA documents and perform various tasks such as setting the metadata, adding default output presets to new DITA maps, and other related tasks. XML Documentation solution also adds workflow launchers that trigger the post-processing workflow on various events like content upload and modification.

You can change the default behavior of the workflow execution as per your processing needs. To do so, in the AEM Workflow Launcher, select the Launcher tab, and find a list of all launchers under the Post
Process XML category. You can enable or disable the required launchers on the basis of their Event Type or Globbing pattern.

Once you have changed the activation status to Enabled or Disabled, then turn off the Enable Post Processing Workflow Launchers setting in the `com.adobe.fmdita.config.ConfigManager` bundle. Once you disable this setting, every time you update any settings in the ConfigManager, the activation status of the Post Process XML launchers is not changed.
Enable component-based translation workflow

Automate the translation of page content, assets, and user-generated content to create and maintain multilingual websites. To automate translation workflows, you integrate translation service providers with AEM and create projects for translating content into multiple languages. AEM supports human and machine translation workflows.

- **Human translation**: Content is sent to your translation provider and translated by professional translators. When complete, the translated content is returned and imported into AEM. When your translation provider is integrated with AEM, content is automatically sent between AEM and the translation provider.
- **Machine translation**: The machine translation service immediately translates your content.

Translating content involves the following steps:

2. Associate the pages of your language master (in AEM 6.4, 6.3, 6.2, and 6.1) with the translation service and framework configurations.
3. Identify the type of content (in AEM 6.4, 6.3, 6.2, and 6.1) to translate.
4. Prepare the content for translation (in AEM 6.4, 6.3, 6.2, and 6.1) by authoring the language master and creating the root pages of language copies.
5. Create translation projects (in AEM 6.4, 6.3, 6.2, and 6.1) to gather the content to translate and to prepare the translation process.
6. Use the translation projects to manage the content translation process (in AEM 6.4, 6.3, 6.2, and 6.1).

If your translation service provider does not provide a connector to integration with AEM, AEM supports the manual extraction and re-insertion of translation content in XML format.

**TIP**: See *Translation* for best practices around translating content.

Enable component-based translation workflow

If the connector for the translation vendor does not support DITA content, then the component-based translation workflow needs to be enabled. Once enabled, the translatable content is sent as asset metadata. However, the connector needs to support asset metadata translation for this workflow to work.

Perform the following steps to enable component-based DITA translation workflow:

1. Open the Adobe Experience Manager Web Console Configuration page.
   
   *The default URL to access the configuration page is:*
   
   http://<server name>:<port>/system/console/configMgr


3. Select the **Component-Based DITA Translation Workflow** option and click **Save**.
Customize search form

With XML Documentation solution, you add the capability of handling XML content in AEM. You can create different types of DITA documents - topic, concept, reference, task, and map. However, the default asset searching capability in AEM doesn’t give you an option to filter assets based on the DITA file types. You can customize the default Search Forms in AEM to add the capability to filter assets based on the DITA file types. See Configuring Search Forms in AEM 6.4, 6.3, 6.2, and 6.1 documentation for more details about how to configure Search Forms.

While configuring the new search predicate, you must configure the following parameters:

**Field Label**

The label that will appear as the collapsible header or as the field label of the predicate. You can provide a value as DITA File Type.

**Property Name**

The property to be searched on. For DITA documents, the property that contains the information about document type can be found under `<any_topic_node>/jcr:content/metadata/dita_class`.

**Option Path**

Provide the content node path where you have defined the predicates for DITA document types. An example of the content node path is: `/app/dam/content/predicates/typetemplates`.

Within the given node there must be a definition for each DITA document type, as described in the following table:

<table>
<thead>
<tr>
<th>jcr:title</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Map</td>
<td>- map/map</td>
</tr>
<tr>
<td>Topic</td>
<td>- topic/topic</td>
</tr>
<tr>
<td>Task</td>
<td>- topic/topic task/task</td>
</tr>
<tr>
<td>Concept</td>
<td>- topic/topic concept/concept</td>
</tr>
<tr>
<td>Reference</td>
<td>- topic/topic reference/reference</td>
</tr>
</tbody>
</table>

**IMPORTANT:** Ensure that the value for each predicate is as defined in the `dita_class` attribute of the respective document type. You should copy the values (including trailing spaces) and paste them in the **value** field.
Appendix

This appendix provides best practices for working with XML Documentation for Adobe Experience Manager. Following these best practices will help you set up, organize DITA content, publish content, and develop processes around content creation, management, and publishing.

The information in this topic is intended for the following type of audiences:

- Publishers, who would run the publishing task to generate output in various formats
- Administrators, who would install and manage XML Documentation solution on Adobe Experience Manager

Custom templates

- Custom topic templates can be added in /apps/fmdita/xmleditor/templates. Custom DITA map templates can be added in /apps/fmdita/ditamapeditor/core/templates.
- It is recommended to set the jcr:title and jcr:description properties on the template that will help in describing the template and to easily identify the template in the DITA map or topic creation page.
- The order of the templates in the Create Map or Create Topic page is determined by the ranking property set on the template. Templates are shown in ascending order of the ranking property.
- It is recommended that all custom templates have the property jcr:mimeType = "application/xml" set on <customtemplate>/jcr:content/renditions/original/jcr:content. This ensures that the new topics and maps are indexed properly.

Content migration and upload

Consider the following best practices while migrating content to AEM from third-party DITA authoring tools, unstructured content from FrameMaker, or directly uploading DITA content to AEM.

- After uploading content, it is recommended to let the post-processing workflows complete before triggering publish or any other heavyweight task.
- It is important not to disable the post-processing workflows or their launchers.
- When you upload an asset in AEM, AEM launches a workflow to extract metadata or create renditions. Therefore, when you migrate a large amount of content, do it in batches of 500 with 2-5 minutes of interval in-between. The interval between the workflows reduces the load on the system.
- Post-processing workflows also create the default set of output presets on DITA maps. Also, automatic links fixing also relies on the post-processing workflow.
- When a DITA map without any output presets is added to a map collection, it appears blank on the page.
• Set Document State workflow sets the document state of new DITA map or topic content as Draft just after uploading or creating the content. This also happens if you are copying or moving DITA content. Disabling it causes document state of new files not set to the Draft status immediately.

• When moving, copying, or renaming DITA content, the internal DITA references are adjusted only if the movement is triggered through the Move/Copy option in the DAM UI. If you move, copy, or rename DITA content using the JCR API, or through CRXDE, or any other API, then you must explicitly fix any references to or from the moved or copied content.

• When same content is uploaded to same location on AEM assets via WebDAV or FrameMaker, then the content gets automatically versioned. This can lead to repository bloat in cases where users don’t wish to keep older versions. AEM Assets' version purging should be configured in such cases.

• DITA map creation in FrameMaker to DITA conversion works only for FrameMaker Publishing Server with FrameMaker (2017 release) Update 1 and later builds.

• Use in-band metadata wherever metadata is defined as inline attribute-value pairs.

NOTE: Elements metadata node gets created only till XML Documentation solution build 2.1. This feature will be removed from future builds. Users upgrading to newer builds from 2.1 or later will have to use a script to get rid of existing elements metadata.

Workflow offloading

XML Documentation solution post-process and publish workflows do not support the traditional AEM model of workflow off-loading using sling job queues. However, load sharing can still be achieved by having separate instances for asset upload, authoring, and production instances.

• Asset upload instance: is where all assets are uploaded and post-processed. Replication agents can be used to replicate processed assets (even DITA content) to the authoring instance.

• Authoring instance: is where all authoring takes place. Replication agents can be configured to replicate reviewed and approved content to the production instance to be published.

• Production instance: is where the publishing takes place. Depending on requirements this can be replaced with a publishing instance which takes care of publishing the DITA content and replication agents replicate the published pages to a different server which then acts as production.

DITA-OT Profiles configuration

The following guidelines are for DITA-OT, which can be configured using the DITA Profiles (Adobe Experience Manager link > XML Documentation > Profiles) or the Configuration Manager.

Timeout

The DITA-OT timeout value defines how long XML Documentation solution waits for a DITA-OT child process to complete before considering it in deadlock. If you are publishing large documents, you may need to increase this value. You can tell if XML Documentation solution is timing out while waiting for DITA-OT by looking at the output generation log file. For more information on accessing
the log files, see the View and check the log file section in the XML Documentation for Adobe Experience Manager User Guide. This property can be configured in the Profile.

**Memory requirements**

The amount of memory required by DITA-OT depends on a lot of factors like the size of content, nested references, number of files, and more. If you observe publishing failures with the output generation logs containing an `OutOfMemoryException` in DITA-OT stack trace, you will need to increase the amount of memory allocated to DITA-OT. For more details on how to change memory allocation from DITA-OT command-line prompt, see Increasing Java memory allocation. This property can be configured in the Profile.

**Generation Pool Size**

By default, the **Generation Pool Size** is set to the number of processors available to the Java Virtual Machine +1. This setting is available in the `com.adobe.fmdita.publish.manager.PublishThreadManagerImpl` component in the Configuration Manager.

**NOTE:** The **Generation Pool Size** configuration only controls how many concurrent external DITA-OT processes are launched and not the actual number of threads taken up by those DITA-OT processes. External from XML Documentation solution’s perspective includes both DITA-OT and FrameMaker Publishing Server tasks. For example, if thread pool configuration is set to 3, then at any given time a maximum of 3 concurrent DITA-OT processes will be active. Each of these DITA-OT processes can launch any number of further threads.

**Using custom DITA-OT**

XML Documentation solution provides an option to use custom DITA-OT package in place of the default package shipped with XML Documentation solution. Consider the following points while using a custom DITA-OT package:

- XML Documentation solution supports DITA 1.2 and 1.3 document types. Specialized DITA is also supported, however it is recommended to use the base DITA specifications. Use specialized DITA only if your content needs are not met with the base DITA document types.

- The default DITA-OT package shipped with XML Documentation solution contains a plug-in to generate EPUB output. This plug-in is not available in the DITA-OT package available for download from the official DITA-OT website. If you are using a custom DITA-OT package, ensure that it contains the plug-in to generate EPUB output, else the EPUB output generation will fail.

- The PDF output preset uses the `pdfx` transformation that is supported by the default DITA-OT package. When using custom DITA-OT package, you must specify the transformation that should be used to generate the PDF output. Also, this transformation has to be supported by your custom DITA-OT plug-in.
Export custom DITA profile as package

You can export your custom DITA profiles as a package and then import it to the other instances of XML Documentation solution. This saves time when you have multiple instances of XML Documentation solution and want to have same DITA profile on all of them.

Perform the following steps to use your custom DITA profile on other XML Documentation solution instances:

1) Customize your DITA profile and save it.
2) Do the following to create a package:
   a) Open CRX package manager and click **Create package**.
   b) Enter package name and click **OK**.
   c) Click **Edit** to edit the package.
   d) On the **Edit Package** dialog, select **Filters > Add filter**.
   e) Select a profile from the root path `/content/fmdita/profiles/` and click **Done**.
      **NOTE:** You can check the name of the profile from the profile properties.
   f) Click **Save** to save your changes.
   g) Click **Build** to create the package.
3) After your package is created, click **Download** to download the package to your local machine.
4) Import the downloaded package to the other instances of XLM Documentation solution using the package manager.

For more information on how to work with a package, see Packaging Adobe Experience Manager 6 applications in AEM documentation.

Output history

- Every output generation information is tracked as an entry under the following repository node: `/content/fmdita/metadata/outputHistory`
- If you observe that the output history tab in the DITA map console is empty even after you have generated multiple outputs, it could mean that your output history node is corrupted. You can clean the output history for a particular DITA map file by deleting its output history node.
  **Note:** Deleting the output history node for a DITA map deletes its complete output history.
- Output history nodes are named on the UUIDs of their corresponding DITA map files. You can find the output history node for a particular DITA map by performing the following steps:
  a) Using CRXDE, navigate to the DITA map file node.
  b) Find the `jcr:uuid` property on the DITA map node and copy its value.
c) Navigate to
/content/fmdita/metadata/outputHistory/<value_you_copied_in_previous_step>.

d) The selected node is the output history node for the concerned DITA map.

- The output history nodes also store the output generation logs. Currently there is no automated way to clear old output history. However, user can write custom scripts to delete old output history nodes by checking jcr:created property of logs node.

**PDF generation**

When creating PDF output and storing them in DAM, you should take care that asset update workflow triggers a sub-asset creation for the generated PDFs. These sub-assets take up a lot of disk space and the sub-asset creation workflow also takes up a lot of CPU. In most of the cases, you won’t have a need of these sub-assets and should preferably disable these.

**Using post generation workflows**

For any custom post-generation workflow, you must put in the **Finalize Post Generation** step as the last step. Otherwise, the output generation will not be properly tracked in output history.

**Unstructured publishing**

- The custom `.sts` file for HTML publishing should be of the same FrameMaker version as that of the FrameMaker Publishing Server.
- The `joboptions.xml` file located at `/libs/fmdita/config/` should not be empty. This file specifies distiller job options to be used for PDF creation of unstructured files.

**Baseline**

- Do not remove a baseline that is used in any output preset. Trying to generate output using a deleted baseline would result in a failure.
- If different versions of an asset are referenced in different topics and if that referenced file is picked automatically, the latest version of the referenced file will be used while publishing. Make sure that same version for such asset is referenced everywhere to avoid undesirable results in published outputs.
- A DITA map containing baseline should not be moved to some other location. Doing so results in loss of baseline information from the DITA map.
AEM Site publishing

Page naming
Till XML Documentation solution version 2.0, all generated site page names were in lowercase. Starting from XML Documentation solution version 2.1, site page names have same case as the corresponding DITA filename. So anyone upgrading from 2.0 to a later version might need to modify external links to their site pages to take care of case changes.

Activation to publish instance
When AEM Site output is generated with Delete and Create option selected for the Existing Output Pages setting, then the previously generated output is first deleted and then pages are re-created. If replication agent is configured on author instance to automatically replicate all changes to AEM site pages to publish instance, then it is possible that by the time pages get re-created, page deletion event is propagated on the publish instance by the replication agent. In that case, the AEM site pages will be marked deactivated on the publish instance.

AEM Site page creation logic
If you are re-publishing some content after changing the Design Path setting to a different template, it is recommended to either publish to a different destination or select the Delete and Create setting to avoid inconsistent output.

AEM Site templates
You can customize the site template to control the structure of the created pages and also the existence of the default search and landing pages, see Customize AEM Site output design template for more information.

Concurrent publishing of the same DITA map
This should be done only at different destination paths. If there are two or more publishing workflows writing to a common destination path, some or all of them might fail and the consistency of the published content is not guaranteed.

Guidelines on creating and overwriting output
When you publish your output in AEM Site format, then you have an option to manage existing pages in the destination. When you select Overwrite Content option in the Existing Output Pages setting, then for any existing page in the destination, XML Documentation solution recreates its contentnode and headnode. Then the new content is written in the newly created nodes. This option does not create any new revision of page.

The Delete and Create option in the Existing Output Pages setting deletes any existing pages in the destination path and then publishes the new content.

Sub-node publishing
XML Documentation solution supports non-destructive publishing, so that you can combine the published DITA content with content created through other means in a page. For this to work, it is important that your DITA map structure mirrors your site structure and the topic filenames match the page names (case-sensitive since version 2.1). Once you have your site structure and the corresponding DITA map hierarchy ready, you can configure the AEM Site template for this purpose:
• Use the `topicContentNode` and `topicHeadNode` properties of your AEM Site template node to specify the node paths where you want DITA content to be published.

• During publishing, XML Documentation solution cleans these nodes of any existing content and publishes new content in them without touching any other nodes in the page structure.

• You must select the **Overwrite Content** option in the **Existing Output Pages** setting in AEM Site preset so that XML Documentation solution does not delete and recreate the whole page.

• For more details about the Site template node and its properties, see *Customize AEM Site output design template*. 
Other recommendations

This topic contains information about various other tasks and workflows that you can optimize.

Permissions

To give a user permissions to use the XML Documentation solution’s all capabilities and generate output in a specific format, you can add that user to the Publishers group created by XML Documentation solution. This ensures that the user has full access to all XML Documentation solution publishing functionalities.

In addition to above permissions, read/write access to specific DAM folders can be configured for this user as per your requirements from the useradmin interface.

Translation

- All translatable content should be placed under a valid self-contained language folder (language copy root) following the naming convention appropriate to the AEM version. For example, if there is a folder for keeping German content, then there should be no content reference from German folder to any other language folder.
- Component-based translation can be used even when the translation vendor does not support XML/DITA translation. However, asset metadata translation needs to be supported by the translation vendor.
- Any common content should be stored outside of any language folder. This ensures that references from language copies to the shared content is relative. This helps in maintaining language copies and ensures that no patching of references is required when creating or updating language copies.
- If additional attributes of XML tags need to be translated in component-based translation, they can be specified in /etc/workflow/models/translation/dita_rules.xml. All attribute names need to be prefixed with -dita_.
- When component-based translation is enabled, references to assets in the source DITA file are automatically changed to point to the same asset under the destination language folder. For this to work, the referenced asset should be under a valid language folder and the language folder must have the same language code as the source language.
- The translation tab in the DITA map dashboard of a DITA map shows all the assets reachable from this DITA map including any non-DITA assets like images or videos.
- When translating non-DITA assets, it is necessary to make sure that the asset properties that need to be translated are present in /etc/workflow/models/translation/translation_rules.xml.
- When translating assets using the translation tab in the DITA map dashboard, it is imperative to accept or reject the translated content using the Accept or Reject buttons in the translation project.
This is applicable to assets that are being translated for the very first time (when the asset with the destination language does not exist).

Logging

Custom Logger
- Preferably create a custom logger for \texttt{fmdita} specific logs (com.adobe.fmdita) so they can be analyzed easily.
- For information on how to create a custom logger, see Logging in AEM 6.4, 6.3, 6.2, and 6.1 documentation.

Unicode support

- To enable support for Unicode characters, open the Configuration Manager and search for \textit{Apache Sling Request Parameter Handling} configuration and set the value of \textit{Default Parameter Encoding} to UTF-8.

DTD

- After specifying a Custom DTD in the DITA Profiles, it is imperative to clear the browser cache to make sure that the old DTDs are not used.
- It is recommended that the \texttt{catalog.xml} of the Custom DTD contain the Public IDs of all files in the DTD so as to avoid a DTD path resolution error. If this is not possible, it is recommended that the topic at the very least specifies the correct System ID. If the System ID is relative to the local file path from where the file is uploaded, it is recommended to select the Add System ID Catalog option at the time of creating or updating the Profile.

Overridden AEM features

- In AEM 6.2, the default inbox and notification features are customized to support the notifications sent during the review workflow. If you make any further changes in the inbox and notification features, you might lose the existing customizations.
- The move, copy & paste, and delete operations for AEM Assets have been overridden. The customized content for the move feature is placed at /apps/dam/gui/content/assets/moveassetwizard, for copy & paste at /apps/dam/gui/content/assets/jcr:content/actions/primary/pasteasset2, and for delete feature at /apps/dam/gui/content/assets/jcr:content/header/items/deletedita location. If you make any further changes in the move, copy & paste, and delete features, you might lose the existing customizations.
FrameMaker integration

- Users need to have read permission on root ("/") node of AEM repository to successfully connect to AEM repository from FrameMaker.
- The file check-in behavior in FrameMaker (2017 release) Update 1 is different from the file check-in experience in the XML web editor. In FrameMaker, when a checked out file is checked back in, the changes are not saved in the latest persisted version, but they are saved in the Latest version that does not have a version number associated with it. This behavior will be changes in the upcoming update for FrameMaker.