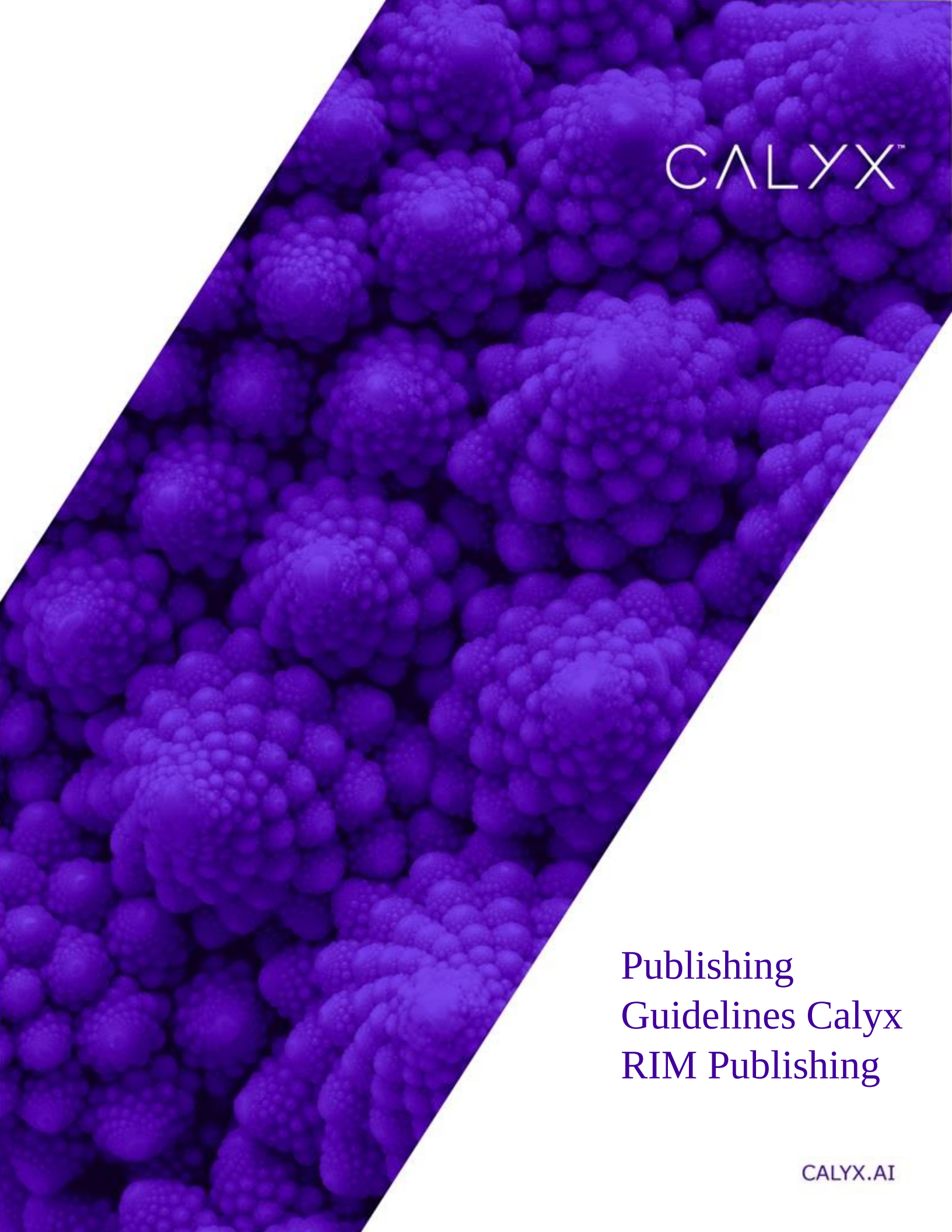

Publishing Guidelines Calyx RIM Publishing

CALYX.AI

1 Contents

2 Revision History

When Calyx releases a new version of Calyx RIM, they issue Release Notes which explain the new features and updates. The Calyx RIM Business Consulting team reviews the Release Notes against each Best Practice to determine any impact to the document:

- Impact = Release notes-documented upgrade changes this Best Practice
- No Impact = Release notes-documented upgrade changes do not affect this Best Practice

When Release Notes impact Best Practice documentation, Calyx recommends that clients review the entire Release Notes for a full understanding of all changes associated with this Best Practice documentation.

Software Version	Release/ Revision Date	Summary of Change(s)
v7.3	14-Jun-2024	Clarifying recommendations regarding Native Files and Leaf's
v7.2	4-Apr-2024	Clarify recommendations for iterative publishing
v7.2	20-Jun-2023	Update Best Practice for v7.2 – No Impact
v7.1	13-Jan-2022	Update Best Practice for v7.1 – No Impact
N/A	12-Oct-2021	Imported into Calyx template.
v7.0	25-May-2021	Update Best Practice for v7.0 – Impact
v6.2 CHF5	22-Sep-2020	First Release of Best Practice.

3 Document Purpose

The purpose of this document is to provide publishing process recommendations that will improve the usability and performance of Calyx RIM Publisher in various submission activities. Recommendations are included for major filings, such as initial applications, as well as lifecycle variations and/or minor amendments. This document provides detailed activities to avoid and provides guidance on optimizing system capabilities to reduce performance impacts.

This document does not replace any current Best Practices or the Calyx RIM Publisher Guide. To fully understand Calyx RIM Publisher and all its functionalities available, we recommend reviewing applicable Best Practices and the Calyx RIM Publisher Guide (all information provided on help.liquent.com).

We also recommend reviewing each edition of the Release Notes so that your organization can maximize the benefits of new usability enhancements.

3.1 Scope

This document provides detailed tips for:

- Any actions that should be avoided
- Maximizing system capabilities to ensure optimal performance

- Applying standard industry best practices and complying with regulations with support from Calyx Publishing experts
- Managing concurrently planned submissions

4 Business Administration Activities

Review and update OTB values as required by your organization. Please refer to the Best Practice *Data Management for Publishers* for appropriate Data Administration settings for Calyx RIM Publisher.

4.1 Publishing Settings Library

The OTB *Publishing Settings Library* settings for each *Assembly Template* come prepackaged with a set of default values. Confirm that your *Publishing Library Templates* point to the appropriate *Preview Location*, *XML Definitions* or *In-Process Rendition Location* (when using In-Process Linking).

Before starting publishing activities, Business Admins should review and update at least the following *Electronic Lifecycle Publishing (ELP)* tabs:

- Publishing Settings
- *Table of Contents (*TOC) for NeeS
- Variables
- Link Profiles

When working with the *License Module Paper Review Publishing (PRP),* be sure to set up tabs related to paper output in addition to the aforementioned:

- *Volumes*
- *Cover Pages*
- *Tabs*
- *Table of Contents (TOCs)* for Paper
- *Overlays*
- *Slip Sheets*

4.2 Default Rendition Identifier

In the Publishing Settings tab, the eDMS connector used will determine whether *Standard Rendition* or *Viewable Rendition* is the appropriate *Default Rendition Identifier*.

- If a File Share or eDMS like SharePoint or Documentum D2 is used, and the setting for ‘Use Native File’ is set to ‘No,’ be sure to select *‘Standard Rendition’* for the *‘Default Rendition Identifier’* field in the *‘Publishing Settings Library’*.

- If Veeva Vault is your eDMS, and you will use Veeva's Rendition according to your business process, then the '*Default Rendition Identifier'* should be set to *'Viewable Rendition'* in Calyx RIM. This indicates that Calyx RIM will not render the documents; instead, the document rendition will be viewable through Calyx RIM.

4.3 Leaf Handling Options

To avoid manual rework for users, consider automating Leaf handling when setting up the system.

You can use ***Leaf Attributes Mapping*** to define customized leaf and document attributes via configuration for the auto-created leafs, then use them when documents are assigned from Documentum or a File Share. Typical use cases for mapping Leaf attributes to Document Attributes include the ability to pull:

- Proper name of Output File name and cut off .docx extension
- Proper Title in case if they are stored on documents in a DMS
- Proper name for the leaf that enhances usability of the assembly tree

For ***Leaf Title Value,*** via configuration you can define a leaf title value that is taken from a source document in the propertyMappings.xml file.

When assigning and using native files, it is important to ensure that only one document is assigned per Leaf. When multiple files are assigned to a single Leaf the system will concatenate it, if a file is set to "Use Native File" then an error may occur.

4.4 Assembly Viewing: Expand Range

There is configuration in insight.var that throws confirmation on expanding some number of nodes. This minimizes the degradation of performance issue cause by expand range for large submissions.

5 General Recommendations

5.1 Web-Based Basics

This section describes basic rules and expectations for working within the assembly utilizing a web-based application like Calyx RIM Publisher.

- You can use more than one web browser session or tab to multi-task within Calyx RIM Publisher; however, you must respect all multiuser rules or it may cause performance issues and/or error messages (i.e. don't use two windows to do a second drag and drop in an assembly while you are waiting for the last drag-and-drop in that assembly to finish on the other window).
- To avoid unexpected behavior, don't open Calyx RIM Publisher in multiple windows or tabs to perform multiple actions within the same assembly. Using multiple windows or tabs to work on different Assemblies is safe.
- Avoid clicking *Refresh* multiple times to avoid prompting the system to re-process and resend redundant information to the browser. Repetitive *Refresh* clicking slows down the system.

5.2 Operations to be Considered

Users should follow these practices as they approach publishing activities and functionality:

A. Operations should complement natural processes.
   - Do not generate a Table of Contents while another user is publishing or updating the assembly structure. The output will not match the new Table of Contents.
   - Do not run queries for the submission during an update to the assembly outline or assembly content. The output will be outdated.
   - Do not delete or add content to the assembly section while publishing or updating the Table of Contents for that particular section. The data will be outdated.

B. Execute Long Running Operations during off hours.
   - Large Delete operation (100+ documents/leafs)
   - *Prepare to Publish* for any large submission
   - *Publish Request*
   - Running the Link Inspector Query

C. Wait for one job to finish before starting another one when there are multiple users working in the same assembly. Avoid Rapid Fire operations such as:
   - Requesting a new *Publish Request* while another request is still in process
   - Assigning documents while another user is performing a *Prepare to Publish (PtP)*
   - Starting one *PtP* request while another one is still in progress
   - Running *PtP* and *Publish Request* at the same time. It is acceptable to perform both steps within the *Publish Request* screen (when *PtP* checkbox is checked on the *Publish Request* screen, Calyx RIM will wait for *PtP* to complete before kicking off publishing). But performance issues arise when a user performs these functionalities concurrently.
   - Send a *Publish Request* when another user is trying to add the assembly to lifecycle

D. When multiple users are working in the same assembly, users should avoid:
   - Working in the same section or folder
   - Changing the Assembly structure within the same section or folder concurrently

E. Avoid running a *Prepare to Publish* and *Publish Request* from any root level folder

5.3 Renditions and Extractions

The rendering process creates a PDF copy of a source document, also referred to as the rendition. Rendered documents retain the content and layout of the source document while transforming it into a PDF file. Extracts from the source documents, such as bookmarks and headings, are copied to the PDF file.

- Calyx RIM creates a rendition of the document that is assigned to a Leaf.
- Calyx RIM points to the document's location within the fileshare/eDMS when the Leaf is assigned.

Calyx RIM learns about the file (# of pages, information about links and bookmarks) during the extraction process. Calyx RIM:

- Re-creates source document extracts (i.e. bookmarks and headings) in the output PDF file along with the recreation of the file.
- Copies the rendition when it publishes, creating an output PDF file based on the source documents, renditions, overlays, resolved variables, links, bookmarks, etc.

All documents within the Assembly should have Renditions and Extractions (except files that need to be submitted in their native format, such as Datasets, labeling, images, etc.).

6 Submission Creation & Publishing Recommendations

6.1 Assemblies

A. Before creating a new assembly, check whether the Updated Submitted Available assembly attribute is set to Yes. This helps to avoid creating multiple starting Assemblies. This may occur when the user forgets to check the Lifecycle box when running the *Change Submittal Status* Wizard of the Initial Assembly.

B. After the initial setup of the Assembly, publish the assembly and select 'XML only' option in the *Publish Request* view to baseline your Sequence structure and metadata so that it will publish successfully.

C. Avoid delaying the change of Submittal Status wizard on Sequences. When a Sequence is submitted to the agency and an acknowledgment is received, change the Sequence Status to *Submitted* utilizing the "*Complete Submission Status Properties*" action in the wizard. This practice is necessary when you are managing concurrent Sequences for an Application. You may use 'Updated Submitted Available' flag for any assembly which needs to be in sync with the outline of the latest submitted sequence assembly.

D. For large Assemblies, avoid using the '*Expand Range*' option on the root of the Assembly (unless you know there are only a few leafs in it) or on any folder that contains a large number of leafs. This can inundate the application server, database, and the Web page, degrading performance. It is best to navigate the tree and expand each folder independently. Refer to "Business Administration Activities – Assembly Viewing: Expand Range"

E. If it is necessary to find a specific element (folder, leaf, or document) within the assembly, typing the key words into the assembly search bar will lead you directly to the location of that element in the assembly.

F. Operations like *delete* or *drag-and-drop* may take time to process. Performing these operations many times concurrently and not waiting for one to complete before performing another operation in that folder may lead to error messages. Eventually, Calyx RIM Publisher performance may degrade.

G. Avoid clicking any button multiple times. Click it once and wait for results before clicking again.

H. Fill out all needed metadata in the initial Clinical Reports so that the data is managed in one location. If it is necessary to duplicate the study, the information will already be included in the cloned study.

I. Create additional Folders with no output folder paths to better organize information for publishers and to help with viewing Leafs.

J. When using any '*Sort'* functionality at the folder or leaf level, minimize impact on performance by only having the user who initiated the sort to work in the applicable folder.

6.2 Source Documents

Avoid blank bookmarks in source files. These are caused by extracted heading styles with no text associated to them. Blank bookmarks slow down the processing time of rendering documents and can lead to performance issues when many documents are being rendered at the same time.

- Note: This scenario can be mitigated by setting the "*Delete Blank Bookmarks*" parameter in Calyx RIM for Rendering servers associated with the eDMS. This setting pre-renders documents before they are assigned to Calyx RIM Publisher and prevents blank Bookmarks in the pre-rendered documents.

6.3 Assigning Documents

Calyx RIM Publisher allows you to assign multiple documents into an assembly at the same time. However, you must consider the following:

A. When a document is assigned, Calyx RIM has to take time to learn about the document (document name, source, # of pages, etc.). Assigning documents from eDMS takes longer than from fileshare.

B. Assigning documents in bulk can cause performance issues when documents are introduced to the system for the first time.
    - Assign documents in bulk if they have already been used in the system. This approach has less performance impact because the documents' information has already been sent to the database, i.e., the system has less actions to perform when Assigning or reassigning documents

C. The size of the document matters. Large documents with thousands of links and bookmarks take longer to assign. To minimize potential performance issues:
    - Reduce the file size in the source document
    - Clean up any scanned documents
    - When possible, reduce the number of links by creating only one link per reference per page (if the reference appears multiple times on the same page)

D. When assigning and using native files, it is important to ensure that only one document is assigned per Leaf. When multiple files are assigned to a single Leaf the system will concatenate it, if a file is set to "Use Native File" then an error may occur.

6.4 Hyperlinking

Use the *Smartlink for PDF* tool through Calyx RIM Publisher to create and edit Hyperlinks.

- With a PDF document opened when utilizing Calyx SmartLink or Calyx RIM for Viewing, avoid closing the web browser directly. Log off while a Calyx RIM Publisher session is open instead. If this step is missed, it could cause the document to be locked and prevent it from being opened by other users.
- Avoid any delays in password input when opening a document while using Calyx SmartLink for PDF. The system may time out and the document will not open.
- Avoid changing the *Rendition Identifier* type after links have been created using Calyx RIM *In-process Linking*. If the Rendition Identifier for the document is set to *Standard Rendition*, and you decide to change the Rendition Identifier type to *Viewable Rendition* (or vice versa), the links created within the document will break.
    - Note: It is best to set up the Default Rendition Identifier required in the Publishing Setting Library when the assembly is created.

6.5 Prepare to Publish

Prepare to Publish (PtP) enables the user to prepare the assembly and become submission ready for eCTD, NeeS or Paper submissions. Although this is an optional step in the publishing process, it is strongly recommended because:

- It confirms that all the files, renditions and extractions that will be necessary for a successful publish are available (or not).
- It updates data used in queries so that hidden issues can be recognized.
- It refreshes volume and page count information to ensure that the page number variables resolve correctly.

But to optimize this functionality, you should:

- Avoid checking the ‘Remove Empty Assembly Folders’ and ‘Remove Empty Assembly Leafs’ boxes during *Prepare to Publish* for a lifecycle Sequence. This functionality is designed to clean up extraneous folders and leaves from the initial Assembly. Selecting this option on a life-cycled sequence will require system time to work through the constraints, a delay which is probably unnecessary.
    - Note: Use Queries to identify empty folders and leafs. Execute deletions via Mass-Delete in the applicable Assembly Outline.
- Run *Prepare to Publish* to create Renditions and Special Sheets before running it again to refresh Publishing Readiness Data and Link Inspector Data.

6.6 Publishing

- Avoid sending multiple, concurrent publish jobs from the same Assembly. This approach could lead to Calyx RIM Publisher performance degradation.

- If only one leaf needs to be re-published, avoid republishing the entire section or publishing from the root folder. Only publish the needed leaf.
- Avoid including XML in publishing every time you publish. Only include the XML option when you need a new XML, for example to perform QC or validation activities on the xml itself.
- It is a general best practice to only select 'All Leaf Elements and XML' the first time the sequence is published. For subsequent publishing jobs, use 'Selected Leaf Elements'. Do not use the 'All Leaf Elements and XML' option if performing post-publishing work on documents. This functionality will override edits to the previously published output files on the file share. Instead, use the option of 'Selected Leaf Elements' for subsequent publishing jobs.
- As a final step execute Calyx RIM Validator to update the checksums in the final eCTD XML. Optionally, you may choose to publish the XML Only as a final step.

IMPORTANT NOTE

When performing iterative publishing, it is not recommended to publish the selected files along with the XML files each time. Including the xml during each publish can significantly affect performance as the entire XML file needs to be built, including calculating any modified file references. With larger submissions and Assemblies this can add significant time to each publish. The XML should only be generated as needed to perform QC or validation checks or to build the final xml ready for the final publish. If further publishing jobs are required, ensure to update the publishing settings accordingly as they persist from the last publishing job. When performing these publishing jobs use 'Selected Leaf Elements', then XML Only as the last step.

6.7 Lifecycle Operations

- Assign only one team member responsible for creating the next Sequence's Working Assembly. This reduces the risk that more than one *Working View* is associated with the Sequence.
- Creating a new Working Assembly for a Sequence takes time, so wait for the request to finish. If a second request is received by the system before the first is completed, multiple working views may be associated with the Sequence. The result: files appear to be moved and/or disappear from views. To resolve this issue, delete the Working Assemblies and start over.
- Be cautious when dragging and dropping a structure with leafs (e.g. from study template). You cannot remove unwanted dragged in leaves in a lifecycle working view
- Ensure metadata are identical with previous submissions (no extra spaces, caps/titles are the same, consistent name for manufacturers, etc.)
- Always append STF. Do not make the STF operation new.
- If DTD information needs to be updated in the assembly lifecycle, ensure to:
    1. Create the next working assembly first
    2. Update to the latest DTD version available in the Publishing Setting within the Publishing Settings Library (PSL)

3. Use Create eCTD wizard to add content to the working assembly

7 Wizard Windows

- Avoid clicking 'Finish' more than once, even if the screen does not seem to refresh. Repeat clicking can cause more than one job to be sent to the application server.
- After wizard screens (e.g. Importing eCTDs or Assemblies), the system may idle on the gear box screen. Because these jobs take time, the system interface may not respond back. On these occasions, you can safely open another tab and work on another task (unrelated to the task just initiated). Or simply close Internet Explorer and check back later to see if the process had completed (e.g. the Assembly you imported now displays).

8 Query & Publishing QC Tips

Prior to publishing an Assembly, publishers should use available queries to identify and address issues within the assembly (e.g. errors in the assembly, xml, document assignment, bookmarks and links, etc.).

Because these queries are considered "Long Running Operations,' you should execute them during off hours to reduce load. Off hours execution also prevents interference that can occur with user update referenced information.

The queries to be used prior to publishing include:

- **Publishing Summary Query** - Displays each component in the assembly, its types and sub-types, and indicates whether leaf options have at least one child document or any overridden attributes with renditions
- **Document List Query** - Provides a view of all placeholders and documents assigned to an assembly. It offers information retrieved from eDMS including (but not limited to): Document title, source document filepath, Assignment Status, Version number, Valid Rendition, latest version
- **Publishing Readiness Query** - This query requires Prepare to Publish functionality to be run to retrieve data. View assembly metadata and find predictive errors and warnings that may occur during publishing if an issue is not resolved.
- **Link Inspector Query** - View the status of all hyperlinks and cross references to make necessary changes prior to publishing. This query will identify links that are created with errors (potentially will not be created in the output files). Provides link status for links created with Smartlink for PDF, Smartlink for Word, native Word cross references, and internal native Adobe links

9 Publishing Large Submissions

When publishing submissions are comprised of 100s – 1000s of Leafs, there are three principles that publishers should abide by:

9.1 Principle 1

When publishing large submissions, the most important guiding principle is to avoid starting another publishing job until the previous one has either completed successfully or has failed. This principle applies to multiple types of publishing jobs:

- individual Leafs, sections, the whole Assembly
- with or without XML

9.2 Principle 2

Start by publishing small sections or documents grouped together in small batches of 15-20 Leafs. Make sure each batch is completed before the next one is started (See Principle 1). If the small batch is completed quickly, you can consider increasing the number of Leafs.

9.3 Principle 3

When publishing iteratively, it is unnecessary to publish xml every time. Including XML generation with the publish consumes a lot of resources, whether for one leaf or for the whole assembly. We recommend publishing the XML only when a submission needs to be reviewed.

9.4 Grouping Documents into one Publish Request

To publish individual Leafs (e.g those in different folders), you can use *Leaf Status* functionality to set the *Leaf Status* of individual Leafs which require publishing to *Ready for Publish*. This setting can be done on an individual level. After a successful publish, the Leaf status will automatically be set to *Published*.

9.5 Iterative Publishing

Not all documents arrive to publishers at the same time when building/assigning the assembly sections. Therefore, users should publish iteratively as sections of the Assembly are completed and documents are available.

- Publishing by sections will help system performance for both eCTD and NeeS submissions
- *Leaf Status* for published Leafs with valid renditions will change from *Planned* to *Published* following completion of the publishing job XML generation, in addition to publishing documents, consumes significant resources, whether for one Leaf or the entire Assembly because the XML provides information for every document in the submission.
- Publish with XML only when the submission is ready to be reviewed.
- Multiple concurrent requests will overwrite the same XML files multiple times
- We recommend that all other publishing be done without XML generation

10Appendix: Submission Process Flow

The Submission activity is covered under an existing Application in the appropriate Product Family.