BEST PRACTICE: WORKING IN THE ASSEMBLY

1 Contents

2 Revision History

When Ennov releases a new version of Ennov InSight, they issue Release Notes which explain the new features and updates. The Ennov 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, Ennov 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) (Refer to Release Notes for Full Description)
v7.3.1	28-Jun-2024	Update Best Practice for Ennov rebranding & for v7.3.1 – No Impact
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
v7.0	25-May-2021	Update Best Practice for v7.0 – Impact

3 Working in the Assembly

After the Assembly is created, you can create a shortcut to access that Assembly in the future without navigating through the Calyx RIM entities.

From the Assembly, select the Copy URL function from the right-hand button bar. This creates a shortcut to the Assembly on your clipboard. You can then save this as a shortcut on your Desktop or as an Internet Explorer favorite. When you later select this shortcut, if you are not already logged into Calyx RIM, you will be prompted to log in and taken directly to the **Assembly** page.

If you created the assembly with the template as a starting point instead of using the Wizard, you should remove any sections of the assembly that are not applicable in the current sequence, and add the structures for the study reports to be provided in the sequence using the **Study Report Wizard**. Any other structural modifications may be made, keeping in mind the following:

- The location of the leafs in the templates match the recommendations of the ICH Granularity document. Any higher-level leafs added should always be the first children under a folder prior to other sub-section folders in order to produce valid XML.
- The number attribute on the folders in the templates is used by the publishing engine to establish the XML elements and the output directory structure. Users may add their own folders for organization within the assembly, but should not complete the number attribute to prevent potential conflicts with the eCTD numbering.

Additional information regarding working within each region and module are provided below. Content is added to the assembly by right clicking and choosing **Assign Document**. This allows the user to browse all defined repositories, select content and add it wherever necessary in the Assembly. If content is added by dragging from the browse window and dropping on an assembly component, the browse dialog remains open and allows the user to work in the assembly and return to the browse dialog to select other content as needed.

The most efficient way to assign content is to expand the entire section(s) to which you want to add content, launch the browse window by right clicking and choosing **Assign Document**, and assigning content to your assembly from the top down. Users can select multiple files within the browse window and drop them into the Assembly. If dropped onto a folder, a leaf is automatically created for each document selected and the content assigned under its respective leaf, resulting in a separate output file for each document during publishing; if dropped onto a leaf, all documents are assigned under the leaf, resulting in a single output file during publishing that has rolled up all the individual documents into one. As sections are completed, it is best to collapse that portion of the assembly to reduce the need to scroll and refresh the screen.

When the assembly structure is as desired, if paper is also being produced, you should add/modify tab sheets, TOCs and any other desired paper finishing options. All TOCs, Tab Sheets, Overlays, Slip Sheets and Cover Pages are defined in the Publishing Settings. A preview of the TOCs and Tabs can be generated at any time. A particularly useful feature available at the root of the assembly is the ability to generate a preview of all tab pages into a single file. This allows you to look through all

tab sheets at once and identify any that exceed desired lengths so all necessary abbreviations can be identified at once.

3.1 Multiple References to the Same Content (Reference Leafs)

A number of example scenarios are included in version 2.6 of the STF specification that relate to inclusion of information from a single study in a different eCTD section without including the content multiple times or the result of different lifecycle operations.

If you have study content that applies to multiple sections of the eCTD, for example a controlled clinical study with PK appendices, the relevant information can be included in each section of the eCTD, such that an STF for each section will be created. Information relevant to both sections is included by including a "Reference Leaf" in one section.

The need for this "Reference Leaf" concept is also common in the Quality sections where more than one manufacturer is being used for the same product or substance. In the eCTD XML there is no difference between what designates a Reference Leaf and a traditional Leaf. It simply is a Leaf that points to the same file as another Leaf. This may be to content in the same sequence, a previous sequence or another application.

In the assembly, a Reference Leaf does not have content or produce an output file since it points to either previously submitted content or content located under another Leaf in the current assembly. To create a Reference Leaf, right-click on an existing Leaf and choose Convert to Reference Leaf. This enables you to browse through the desired Assembly (whether it is the current Assembly, a previously submitted Sequence, or a Submitted Assembly in another Application) and select the Leaf that has the desired content assigned. When published, Calyx RIM will build the appropriate relative path to the content.

As lifecycle operations are performed on Leaf elements in your assembly, you can identify any Reference Leafs that point to the same content on which you have just performed a lifecycle operation. This enables you to keep lifecycle operations in harmony.

3.2 Parallel eCTD Sequences and Assemblies

It is recommended that you follow the best practices regarding parallel eCTD sequences and parallel working assemblies in Calyx RIM.

3.2.1 Parallel eCTD Sequence

The parallel eCTD sequence model is designed around the concept that the first sequence is the initiation of the application, where other parallel sequences during the creation of the first sequence are unlikely.

3.2.2 Parallel Working Assembly

When a working assembly is created it uses the current submitted view as the basis for that working assembly. Multiple working assemblies can be created from the same submitted view.

3.3 Managing Parallel Initiating eCTD Sequences

Calyx RIM is designed to allow for management of parallel sequences to an eCTD application. However, it does require that the lifecycle of the application be established by adding the first sequence (0000) to the application lifecycle.

This model is designed around the concept that the first sequence is the initiation of the application, where other parallel sequences during the creation of the first sequence are unlikely.

There are cases where existing Regulatory Applications may be migrated to the eCTD format, and parallel sequences are being created prior to initiation of eCTD lifecycle. Within Calyx RIM, creating parallel sequences without an existing lifecycle can result in lifecycle conflicts.

The topics in this section describe a process that will avoid these lifecycle conflicts, using Standalone Assemblies as the starting point for the parallel sequences that will be merged into the full Calyx RIM hierarchy.

3.3.1 Using Standalone Assemblies

Standalone Assemblies in Calyx RIM are assemblies that have no association to a Product Family or Application. To address the needs for parallel initiating eCTD sequences, these will be used as interim assemblies until it is known which sequence will be the initiating 0000 sequence.

Prior to the inclusion of the first assembly in an eCTD lifecycle, a separate Standalone Assembly should be created for each planned sequence.

To create a Standalone Assembly in Calyx RIM:

1. In the **New** menu, choose **Standalone Assembly**. The assembly can be created using the **eCTD**
   Wizard. However, a blank assembly must exist first.
2. Select **New (Empty)** as the starting point for the Assembly and then click OK.
3. Enter a **Name** for the Assembly and set the **Assembly Type** to eCTD.
4. Set any other values as needed and click **Create**. The assembly will be created with a root element only.
5. To build the assembly structure using the eCTD Wizard, in the Wizards menu, choose Create eCTD.
6. In the Create eCTD Wizard:
   a. Select the desired **Region**.
   b. Move the appropriate **Country(ies)** to the **Selected** box.
   c. In the **Template Selection** list, select the desired Module 1 template
   d. Click **Next**.
7. Proceed through the rest of the Wizard to construct the assembly and provide the requested meta-data. Because the Standalone Assembly was created prior to executing the **eCTD Wizard**, the Publishing settings need to be applied in the newly created assembly.
8. Click **Publishing Settings** and apply the necessary publishing settings, including XML Definition Files, as follows:

a. **AU Submissions:**
    i. ich-3-2
    ii. stf-2-2
    iii. au-3-1
b. **Canada Submissions:**
    i. ich-3-2
    ii. stf-2-2
    iii. ca-2-2
c. **CH Submissions:**
    i. ich-3-2
    ii. stf-2-2
    iii. ch-1-4
d. **EU Submissions:**
    i. ich-3-2
    ii. forms-1-1 (if the eAF will be included in the application)
    iii. eu-1-x, where x is the desired version number of the EU spec
    iv. eu-3-0-1
    v. stf-2-2 (even if node-extensions are being produced, this rules file is necessary to allow the setting on the Module 4 and 5 folders that dictates how studies should be handled)
e. **GCC:**
    i. ich-3-2
    ii. stf-2-2
    iii. gc-1-5
f. **Japan Submissions:**
    i. ich-3-2
    ii. stf-2-2
    iii. jp-1-0
g. **TH:**
    i. ich-3-2
    ii. stf-2-2
    iii. th-1-0
h. **US Submissions:**
    i. ich-3-2
    ii. stf-2-2
    iii. us-3-3
i. **ZA:**
    i. ich-3-2
    ii. stf-2-2
    iii. za-1-0

9. Populate and publish the Standalone Assemblies as needed.

Since these are standalone assemblies, the system cannot add the application number and sequence numbers to the end of the path where you are publishing. You should create these folders using 'dummy' sequence numbers for each sequence in the repository and select that full path when publishing, with the Append Application and Sequence to Output location unchecked.

When it is known which assembly will be the initiating sequence for the application, it can be used to create the appropriate Sequence Assembly.

3.3.2 Creating Calyx RIM Hierarchy and Adding First Assembly

Use the **Create Submission Wizard** to create the application, event, and sequence.

To create an application, event, and sequence using the **Create Submission Wizard**:

1. In the Create Submission Summary window, click the assembly link to create an assembly.
2. Select Existing Assembly as the starting point.
3. Using the Browse button, search for or navigate to the appropriate Standalone Assembly and complete the select Existing Assembly process.
4. On the Create Assembly screen, provide a name for the Assembly and set the Electronic Output Location to the folder one level above the application number folder to allow use of the Append Application and Sequence Numbers to Output path option when publishing. This enables Calyx RIM to ensure correct STF navigation as well as future cross-sequence navigation, if necessary.
5. Click Create to create the new sequence assembly.
    a. If necessary, select the Module 1 folder and edit it to reflect the correct sequence number and submission date. At this point, any updates for this sequence should be made in this Sequence Assembly, and the Standalone Assembly used to create it should be archived or deleted.
    b. The eCTD and/or Study Report Wizard can be re-executed at any time to add additional sections.
    c. It is necessary to republish at least the XML for this new sequence assembly to get final IDs in the backbone and STF files. Content that was previously published does not need to be republished unless updates have been made. However, the 'dummy' sequence number created in the repository for any initial publishing of the Standalone Assembly should be updated to reflect the correct sequence number (0000), if necessary.
6. When the new assembly is completed, use the **Change Submittal Status Wizard** to add the assembly to the lifecycle. Select the checkbox that indicates **This submission is being submitted as a lifecycle eCTD**.

3.3.3 Creating the Second Sequence and Assembly

After the first sequence has been added to the lifecycle, the next sequence can be merged into the application.

To create the next event and sequence:

1. In the Wizards menu, choose Create Submission.
2. At the Choose Submission Type prompt, select Major Update and click Next.
3. Add the required information to the Complete the Submission Status Properties page, and click Next to create the event and a new sequence.
4. On the Create Submission Summary page, click the assembly link to create the new Working Assembly.
5. On the Create Assembly page, provide a name for the assembly.
6. No updates to the Output location should be necessary as this will persist from the original sequence.
7. In the Working Assembly, right-click on the Assembly Root or any Folder and choose Import Assembly Wizard.

3.4Managing Parallel Working Assemblies

While Calyx RIM is designed to allow multiple working sequence assemblies in an eCTD application with an established lifecycle, there are some actions that could be performed in the different sequences that could lead to redundant sections or unintended lifecycle in the submitted view.

When a working assembly is created it uses the current submitted view as the basis for that working assembly. Multiple working assemblies can be created from the same submitted view.

When one or more of those working sequence assemblies is added to a lifecycle, the other existing working assemblies are not updated with the new sequence's information. If each working assembly's lifecycle and section additions are independent of one another, there are no issues. However, if multiple working assemblies attempt to perform lifecycle operations on the same leaf from the submitted view, or if multiple working assemblies add the same new sections to the application, conflicts can arise.

Clear planning is required when working assemblies are created, in order to avoid these conflicts. The topics in this section describe some alternative actions that can be taken if such conflicts do arise.

3.4.1 Creating an Updated Working Assembly

If a planned Sequence has a working Assembly that was created from a Submitted View prior to the most current Submitted View, and you need to act on content that only exists in that updated Submitted View, you will need to synchronize the current Working View with the updates made in a Submitted View.

To synchronize a current Working View or an initial assembly with updates of the Submitted View:

1. In the Working View or an initial assembly, choose the root to synchronize the entire assembly.
2. On the **More** menu, click **Synchronize with the Submitted View**.
3. On the confirmation message, click **OK**.

The synchronization process starts. While the synchronization process is in progress, some assembly actions are unavailable.

4. To view the status of the synchronization process, use the **Job Requests** feature.

When the synchronization process is complete, the status of **the Updated Submitted Available** attribute is set to No.