

Using Pinnacle 21 Enterprise™ for define.xml Creation: Tips and Tricks from a CRO Perspective

Frank Menius, Covance, Cary, USA

ABSTRACT

Creating a define.xml 2.0 document that is compliant and ready for regulatory review can be time consuming and fraught with hazards. Pinnacle 21 Enterprise is a great tool that can be used to speed up this process and ensure compliance with regulatory bodies, but how can a preparer make sure they are getting the most out of the product? What options are available that are not being taken advantage of? What are some common issues when using P21 Enterprise and what are the ways to work around them? This paper will detail an overview of define.xml creation with Pinnacle 21 Enterprise by highlighting tips, tricks, and work arounds.

WHAT IS THE DEFINE.XML

According to the Clinical Data Interchange Standards Consortium (CDISC) the define.xml includes metadata that describe any tabular dataset structure. The submission of define.xml is required by FDA and the PMDA “to inform regulators which datasets, variables, controlled terms, and other specified metadata where used.” (www.cdisc.org/standards/data-exchange/define-xml) The define.xml is one of the most important parts of any submission, as any data described by it cannot be properly understood without the define.xml giving them context.

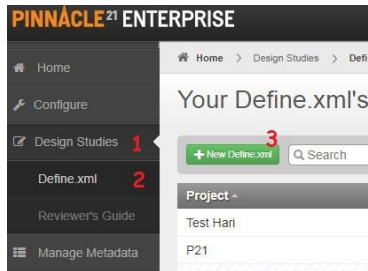
As previously stated, the creation of a proper define.xml can be very time consuming and labor intensive. During preparation the assumption must be made that the reviewers at the regulatory agencies have no knowledge of programming study specifics or the programming language used. Therefore, every effort needs to be made to communicate the information in the define.xml in the clearest most concise language possible.

P21 Enterprise offers a range of tools to make define.xml creation more streamlined, while leveraging the same quality and standards checks used by the FDA to review the submission. When used to its full potential P21 Enterprise can provide a real time check on the health and validity of the study data.

WHERE TO START

Note: This paper assumes proper system and user setup, as well as the proper creation of Project, Study, and Data Package folders, in order to focus on the define.xml creation process.

The first step in creating a define.xml using P21 Enterprise is to initialize the define creation module. This is performed from the home screen by selecting **Design Studies > Define.xml** from the left navigation bar and then clicking the green **+New Define.xml** button from the top.



The user will be presented with a screen offering six creation options. These options establish where P21 Enterprise will obtain the metadata for the define.xml creation.

OPTIONS INTERNAL TO P21 ENTERPRISE:

Start from Scratch will provide the user with a blank module to manually complete. This option will be the most labor intensive and be prone to error, so this is not a recommended starting place. If an organization has no prior validated data or specification from which to initiate a define, it is recommended to export a blank specification from P21 Enterprise and use the “Import Excel Specification” option described below.

Copy from Standard allows the creation of a define based on an organizations standard metadata at the beginning of a study. This method may not be that viable for Clinical Research Organizations (CROs) with multiple clients and standards. However, if CRO’s have ongoing partnerships with particular companies it may be advantageous to create a standard for each partner to expedite future work.

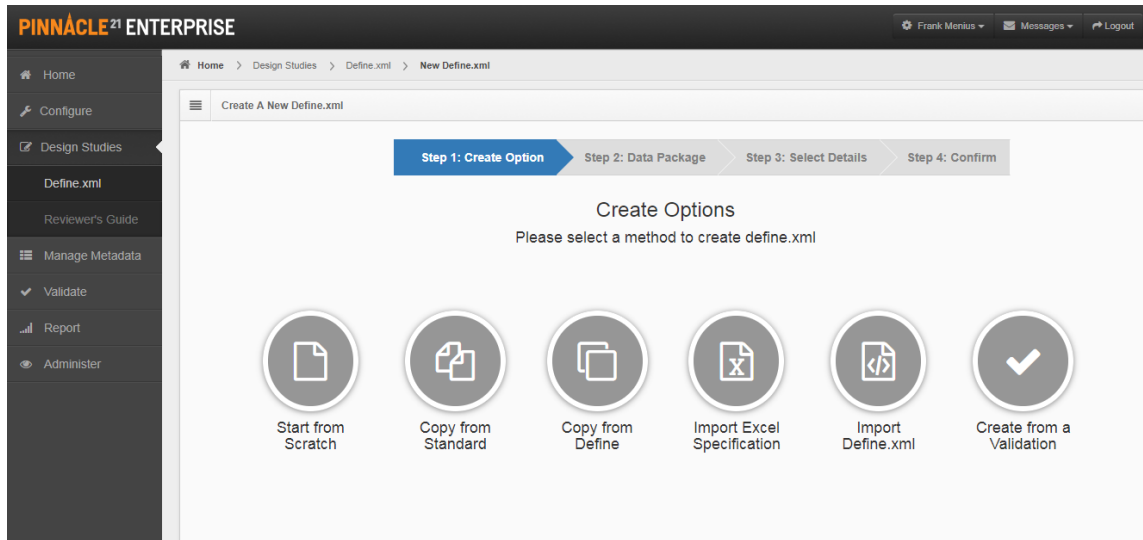
Copy from Define allows for the copy of another study’s define.xml metadata already the user’s system as a template for the new define.xml.

OPTIONS FROM SOURCES EXTERNAL TO P21 ENTERPRISE:

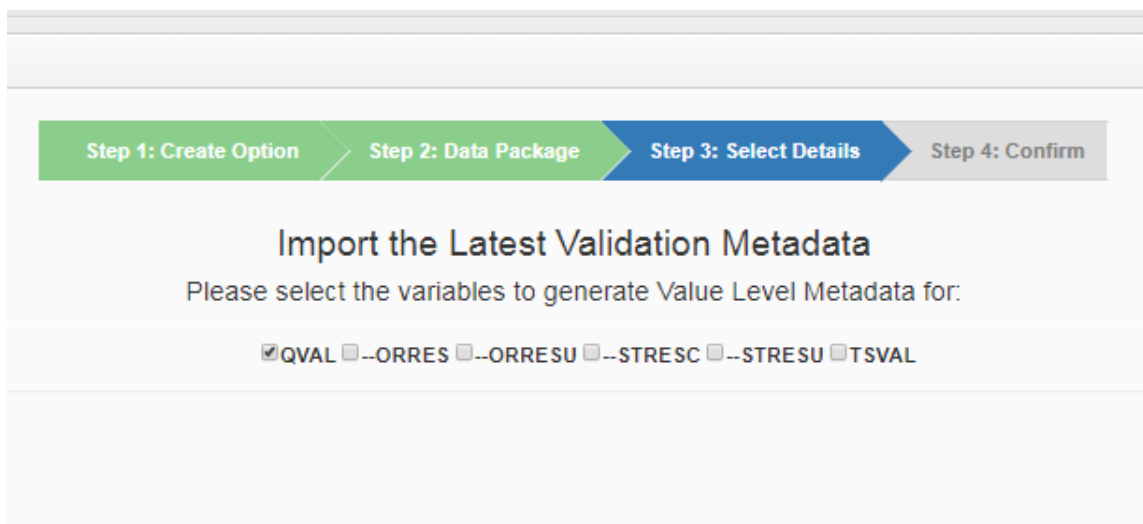
Import Excel Specification allows the user to import the metadata from a standardized specification existing outside the system in Microsoft Excel format. If the specifications are written in define ready language this can be a massive time saver. This can also be used as a bug fixing technique (to be address later).

Import Define.xml allows the import of metadata from an already created define.xml 1.0 or 2.0. This method can be useful when study documents are inherited from other entities and need to be edited or incorporated into a larger document.

Create from Validation utilizes transport files (“.xpt” files) previously validated using P21 Enterprise. This is my recommended method and will benefit from the full spectrum of functions, capabilities, and quality checks that P21 Enterprise offers.



Once the selection is made the wizard walks the user through selecting the correct Data Package and the necessary files location(s). Depending on the selection the user may also need to select the variables for which Value Level Metadata (VLM) need to be specified. VLM are specifications for the derivation of values within subgroups of a column. For the Study Data Tabulation Model (SDTM) data packages is expected VLM are provided for QVAL, at a minimum. Additionally, VLM are also recommended for –ORRES columns. For the Analysis Dataset Model (ADaM) data packages, VLM are expected to be provided for AVAL derivations. The selection of other options is possible but should be discussed with study leads as it will create additional cross-synchronization checks and corrections between the Origin of the VLM versus the Origin of the entire Variable within the system. Additional VLM may be added later as needed, but it is important to remember that VLM should be used in a way that it is useful – it should answer questions rather than create noise or worse, more questions.



EDITING THE DEFINE

Once the define module has been initialized, the next step is to populate protocol-specific the information for the define. From the home screen, the user will navigate to **Design Studies > Define.xml** and select

the appropriate study with a define from the **Your Define.xml's** table. The selection is made by clicking the blue hyperlink under **Data Package**. This link is not placed logically to anyone not familiar with P21 Enterprise, it is in the middle column.

Your Define.xml's

+ New Define.xml Search

| Project ^ | Study ^ | Data Package ^ | Standard ^ | Last M |
|----------------|------------------|---------------------------------------|-------------|--------|
| P21 Validation | P21 E - PQ | SDTM-IG 3.2 | SDTM-IG 3.2 | 2017-1 |
| P21 Validation | P21 E - PQ | SDTM Define from spec | SDTM-IG 3.2 | 2017-C |
| P21 Validation | P21 E - PQ | SDTM Define 1.0 | SDTM-IG 3.2 | 2017-C |
| P21 Validation | P21 E - PQ | ADaM-IG 1.0 | ADaM-IG 1.0 | 2017-C |
| Training1 | training study 2 | define from spec test | SDTM-IG 3.2 | 2017-C |
| Training1 | training study 2 | test define 1.0 | SDTM-IG 3.2 | 2017-C |

Depending on the selections used to create the define module, when the module is opened a great deal of information may have been automatically imported from prior metadata into the define. It is up to the user to verify the imported information is correct and provide any which was not imported.

THE PROPERTIES TAB

The define module will always open on the **Datasets** tab, but I feel it is important to initially verify the information on the **Properties** tab prior to moving forward. On the **Properties** tab confirm the accuracy of all fields, as these will affect and inform additional system checks. Some specific items to check include:

- Was Protocol name imported?
- Does it reflect the proper Standard and Controlled Terminology?
- Are all dictionaries included?
- Are all the documents included (acrf.pdf, csdrg.pdf, adrg.pdf, computational_methods.pdf, etc.)?
- Does the document href (filename) match the standards for the submission?

Note: In older instances of P21 Enterprise the Study Data Reviewer's Guide (SDRG) is automatically named "reviewersguide.pdf" when it should be named "csdrg.pdf" to be submission compliant.

Define: P21 > P21-OQ > SDTM

+ Import Export Copy Compare

Properties Datasets Variables Value Level Codelists Terms Methods Comments Issues 301 History

Define Properties

Name: P21-OQ

Description: SAMPLE STUDY DEVELOPED TO DEMONSTRATE A SAMPLE SUBMISSION

Protocol: Protocol?

Standard: SDTM-IG 3.2

SDTM CT: 2017-06-30

Language: English

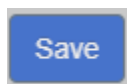
Dictionaries

| ID | Name | Data Type | Dictionary | Version |
|---------|----------------------------------|-----------|------------|------------|
| MedDRA | Adverse Event Dictionary | text | MedDRA | 20.0 |
| NDF-RT | Pharmacological Class Dictionary | text | NDF-RT | 2017-06-14 |
| SNOMED | Trial Indication Dictionary | text | SNOMED | 2017-03-01 |
| WHODrug | WHO Drug Dictionary | text | WHODrug | 2017-01-01 |

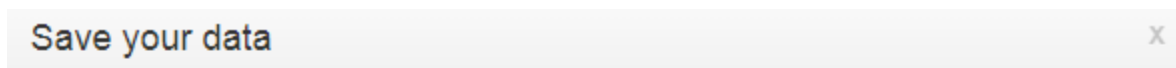
Documents

| ID | Title | Href |
|----------------|-----------------------------|--------------------------|
| scrf | Annotated Case Report Form | scrf.pdf |
| reviewersguide | Study Data Reviewer's Guide | CSDRG_reviewersguide.pdf |

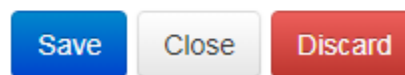
In P21 Enterprise, any field that can be manually changed will alert the user in real-time upon field completion with non-expected values/issues. Changes should be made and saved by clicking the blue **Save** button on the bottom right.



TIP: It is important to save before clicking on another tab in the system to prevent the loss of work. P21 Enterprise cannot hold work from multiple tabs between saves. Saving also starts the system of cross references and validation checks throughout the define module. It is good practice to hit save after every single field is updated in the document. Still, it is likely the user will be presented at some point with a “Save your data” warning or an error. Loss of work may be prevented by copying what was most recently added into the define module and pasting it into Microsoft’s Excel or a text editor temporarily. It is then possible to remove the entry from the define module and hit save in P21 Enterprise. At this point P21 Enterprise will save properly and the text that was copied over can then be pasted back into the define module. Once **Save** has been pressed a second time the user may continue as usual.



You must save changes before continuing in another table!



THE DATASETS TAB

Similar to the **Properties** tab, the **Datasets** tab will be auto-populated unless starting the define.xml completely from scratch. However, it should be stressed the information is not always accurate due to a multitude of reasons (e.g. items may not have been updated as frequently as needed in the source metadata). Ensure to check all columns especially the **Structure** and **Key Variables** columns. Make any needed changes and save.

TIP: Sometimes studies will add or remove datasets and/or variables. If this happens it is relatively easy to manually update in the define module. In any of the tables a row can be inserted above, inserted below, or removed by right clicking the row in question.

| Dataset | Description | Class |
|---------|--------------------------------------|-----------------|
| AE | Adverse Events | EVENTS |
| CM | Concomitant Medications | INTERVENTIONS |
| DA | Drug Account | FINDINGS |
| DM | Demographic | SPECIAL PURPOSE |
| DS | Disposition | EVENTS |
| EG | ECG Test Results | FINDINGS |
| EX | Exposure | INTERVENTIONS |
| IE | Inclusion/Exclusion Criteria Not Met | FINDINGS |

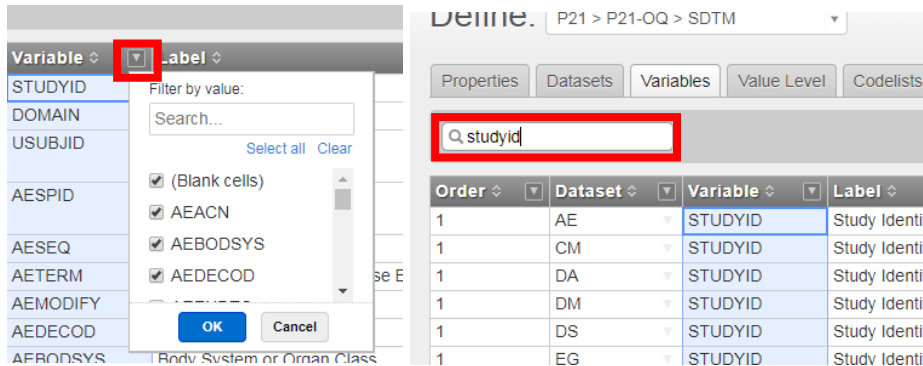
THE VARIABLES TAB

Most of the define creation work will be performed on the **Variables** tab. Here as with the other tabs the majority of the information has been auto-populated based on the selection at the beginning of the define build process. It will be the user's primary responsibility to confirm or provide the Origin for each variable as well as links to any relevant Comments, Methods, or Codelists.

Every variable must have an Origin. P21 will alert the user with an issues flag if an origin is not provided. There is one important exception. Variables with multiple value level origins **do not** get an origin on the **Variable** tab. In those cases, the origin is populated on the **Value Level** tab and the **Variable** tab origin cell is left blank

The possible origins values consist of **CRF**, **Derived**, **Assigned**, **Protocol**, **eDT**, or **Predecessor**. These values can be typed into the Origin column or selected from a dropdown list. The selection of **CRF** will require a page number, which is hyperlinked to the acrf.pdf in a later process. The selection of **Derived** will require a link be provided in the Method column. The selection of **Predecessor** will require the predecessor value be typed into the Predecessor column. The selection of **Assigned**, while not required by the system, should be accompanied by a link in the Comment column. **Protocol** and **eDT** have no additional requirements.

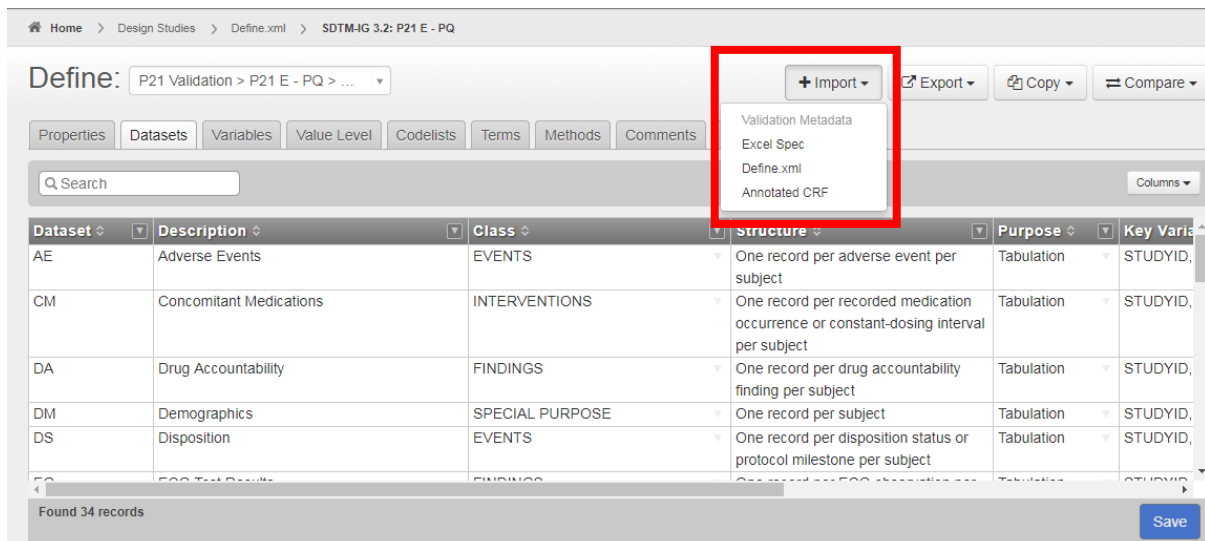
TIP: If multiple domains contain the same variable with the same or similar origin you can filter the page by any column by clicking the down arrow in the column header and then setting the filter. Similarly, the search bar can be used to filter for certain words. A word of caution when using filters - sometimes if a filter is used when adding or deleting records, hitting save will cause the program to freeze. The changes will be saved but the define module will need to be exited and reentered.



ANNOTATED CRF

The first step on the **Variables** tab should be to import the Annotated Case Report Form (aCRF). According to P21, this is an under-utilized feature that has the potential to provide tremendous time savings. First, the user will need to obtain a copy of the aCRF and save it locally. While the initial name of the file is not important at this point, for submission it will need to be named acrf.pdf.

To import the aCRF the user needs to click the **+Import** button in the top right and then select **Annotated CRF** from the drop down. On the next window click **Browse** and navigate to the location of the appropriate document. Then hit **Submit**.

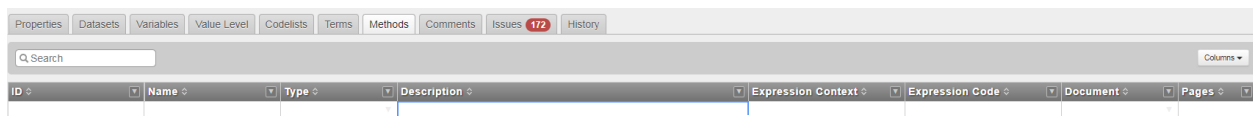


This step may take a moment, but while importing P21 Enterprise will read the annotations and provide accurate page numbers for all variables mentioned and place them into the define module. The user will want to go back and verify the page numbers, especially if there are variables that have the same name across multiple domains or if you have annotations that may not specifically identify all variables. For example, due to space limitations, I will often annotate all –GRPID variables with one annotation that states that –GRPID variables across domains come from page x. P21 Enterprise cannot distinguish this, so those variables will need to be manually set to CRF.

TIP: If changes were made to the aCRF it can be reimported. Any pages in the new aCRF will be updated in the define. However, if there was a variable that was in the old version of the aCRF but is not in the new version of the aCRF, the old page references for that variable will still be in place. This means that P21 Enterprise will not alert the user to any issues with this variable’s origin, and it may be overlooked.

METHODS

Setting a variables Origin to **Derived** requires the creation of a method to be linked to the variable. The method must be created on the **Methods** tab before it can be typed into the Method column on the **Variable** tab. To create a method, select the **Methods** tab and then populate the required fields as follows:



The **ID** column needs to be populated with a unique identifier for the derivation. This ID will not be visible in the define and will act as the hyperlink linking the variable to the method. More than one variable can be linked to the same method by entering the same method ID into the methods column on the **Variables** tab.

The **Name** is the label for the method as populated in the Computational Algorithms section at the bottom of the final define.xml.

TIP: As many reviewers are not aware that this is redundant information already presented with the variables in the define.xml, it will be important to have a clear naming convention here. For example domain.variable can be used for variables that may exist in multiple domains but with different derivations. For methods that are the same within each domain, a more general method ID may be used.

Type values include Computation or Imputation.

Description is where the text of the derivation should be typed.

TIP: As the background of the reviewer is varied and unknown the use of coding lingo, pseudocode, functions, symbols, and abbreviations should be avoided. It is preferred that all derivations except for purely mathematical operations to be related in prose.

Expression Context is an optional field that can be used to express the coding environment, i.e. SAS 9.4.

Expression Code is an optional field that can be used to present the actual programming code used in the derivation.

Document is where a link to a document further describing a derivation may be added. This document must have already been added to the **Properties** tab prior to being linked in the **Methods** tab.

TIP: The goal of the define.xml is to be machine readable, however due to the stylesheet limitations, long derivations are often not “human readable” as spacing, bullets, and other clarifying markups can not be made. To rectify this, the text of the derivation can be placed in the Description column satisfying machine readability. Additionally, a Computational_Methods.pdf document can be created where the text of long complex derivations can also be presented in a more “human readable” format.

The **Pages** column is to be filled with relevant page numbers from a specified document in the Document column. Multiple page numbers are separated by a space so they may be properly hyperlinked upon creation of the final define.xml document.

When all required fields are completed the user should click Save and then return to the **Variables** tab. The value that was used in the ID column can then be typed into the Method column for the variable in question linking the two together.

COMMENTS

Creating comments works much the same way as methods, except there are fewer fields. To create a comment the user must navigate to the **Comments** tab and then populate the fields.

The **ID** is similar to the **Methods** tab with the exception that the ID values will be visible in the Comments section at the bottom of the final define.xml. The ID value will be the link between the Comment and the Variable.

Description is where the text of the comment should be typed, - noting the same restrictions to use prose and not pseudocode for any derivations.

Document and **Pages** columns work the same as for the **Methods** tab.

When all required fields are completed the user should click Save and then return to the **Variables** tab. The value that was used in the ID column can then be typed into the comment column for the variable in question linking the two together.

TIP: Comments are expected with most Origins of “Assigned” but can also be placed along side any other origin for additional clarification.

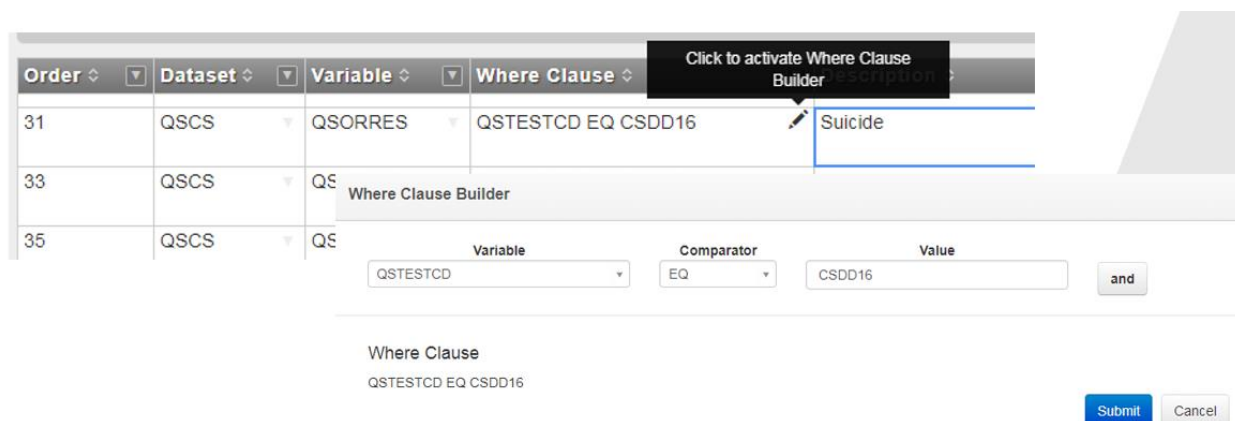
THE VALUE LEVEL TAB

For the most part the **Value Level** tab works the exact same way as the **Variable** tab. Origins, Comments, and Methods are set up the same way as for the **Variables** tab. The exception for the **Value Level** tab is the Where Clause column, which sets the condition in which the value level derivation

applies. For example, in SDTM EG each parameter will likely have a different derivation for EGORRES making it difficult to store the complete derivation of EGORRES in the **Methods** tab. The value for the overall interpretation of the ECG results, EGORRES where EGTESTCD equals INTPT, would have the Where Clause column populated with "EGTESTCD EQ INTPT". This allows for the derivation of this result to be specified separately from other ECG results like QTCf QTCr etc., making documentation easier to parse.

Note: The Where Clause column has the option to have other operators beyond EQ and can even have multiple complex conditions. The operators available to use are EQ, LT, LE, GT, GE, NE, IN, and NOTIN. As mentioned above, if there are multiple Origins expressed for a variable on the **Value Level** tab, the Origin for that variable on the **Variables** tab will be left blank. The Origin of a Variable and its Value Level cannot be in conflict.

TIP: To assist in the creation of complex Where Clauses there is a where clause builder inside the tool. The user should hover the mouse over the Where Clause cell to be edited and then click the pencil icon that appears in the top right of the cell. See Image below:



P21 Enterprise auto-populates multiple codelists based on the metadata, but it is not able to create them all. It is the responsibility of the user to review each codelist to ensure they are complete, necessary, and presented logically.

Codelists in P21 Enterprise consist of three parts, the information on the **Codelists** tab, the information on the **Terms** tab, and the link in the Codelist column on the **Variable/Value Level** tab.

To create a codelist, the user must first navigate to the **Codelists** tab and populate the form. The **ID** column here works like the ID column in the **Methods** tab or **Comments** tab, as the link between the codelist and the variable. The **Name** column will be the display value of the codelist and likely should reflect the value of the associated variable label. The **NCI Code** will need to be added for columns that are required to use controlled terminology if they do not auto-populate. The **Type** must match the type of the variable.

| ID | Name | NCI Code | Type |
|----------|------------------------|----------|------|
| EGTESTCD | ECG Test Code | C71153 | text |
| ELEMENT | Description of Element | | text |
| EPOCH | Epoch | C99079 | text |
| ETCD | Element Code | | text |
| ETHNIC | Ethnic Group | C66790 | text |
| EXDOSFRM | Dose Form in EX | C66726 | text |
| EXDOSU | Dose Units in EX | C71620 | text |
| EXTRT | Name of Treatment | | text |

Once the codelist has been populated on the **Codelists** tab the user should then navigate to the **Terms** tab and fill out the form for each term in the codelist. The **Order** column sets the order in which terms are displayed (more details will be given in the **Codelist Issues** section of this paper). The **Codelist** column is the same as the **ID** from the **Codelists** tab and links the term to the associated codelist. The **Term** column is where each value for the codelist is populated. The **NCI Code** column is auto-populated and cannot be edited (more on NCI Code in the **Codelist Issues** section of this paper). Finally, the **Decoded Value** column is where any definition or further explanation for the codelist is populated.

| Order | Codelist | Term | NCI Code | Decoded Value |
|-------|----------|--|----------|--|
| 5 | EGTEST | QTcF - Fridericia's Correction Formula | | |
| 6 | EGTEST | Summary (Mean) QT Duration | | |
| 7 | EGTEST | Summary (Mean) Ventricular Rate | | |
| 1 | EGTESTCD | INTP | C41255 | Interpretation |
| 2 | EGTESTCD | PRMEAN | | Summary (Mean) PR Duration |
| 3 | EGTESTCD | QRS DUR | | Summary (Mean) QRS Duration |
| 4 | EGTESTCD | QTcB | | QTcB - Bazett's Correction Formula |
| 5 | EGTESTCD | QTcF | | QTcF - Fridericia's Correction Formula |
| 6 | EGTESTCD | QTMEAN | | Summary (Mean) QT Duration |
| 7 | EGTESTCD | VRMEAN | | Summary (Mean) Ventricular Rate |

Once the **Codelists** tab and the **Terms** tab have been populated the user should return to the **Variables/Value Level** tab. Here the **ID** from the **Codelists** tab can be entered into the Codelist column to link the variable to the appropriate codelist.

TIP: Sometimes it is necessary to remove a codelist from the define. To remove a codelist navigate to the **Codelists** tab and right click on the codelist that you wish to remove. Then select "Remove row" to delete the codelists. Hitting save will also remove the codelist from the **Terms** tab and remove the link on the **Variables/Value Level** tab.

CODELIST ISSUES

Codelists just might be the most frustrating and time-consuming part of P21 Enterprise -they take a lot of effort, queries on the data may arise, manual entry must be performed, and then the program itself has some quirks that new and experienced users may find difficult. This section addresses these issues and provides solutions for users to eliminate (or at least *lessen*) their frustrations.

UNNECESSARY CODELISTS

Up first, unnecessary codelists. P21 Enterprise auto-populates codelists based on the metadata. Depending on the Value Level selections when initializing the define module, P21 Enterprise likely created a separate codelist for each one of the xxTESTCD's or xxORRES's in the define. This will likely mean that there are multiple codelists with numerical values that do not add any useful information to the define file. For example the numeric values of the mean QT interval in the EG domain are not indicative of a useful codelist.

| | | | | |
|-------------------------------|---------------------------------|---|-------------------------------|-----|
| EGORRES.EG.EGTESTCD.EQ.PRMEAN | Summary (Mean) PR Duration | 3 | EGORRES.EG.EGTESTCD.EQ.QRSDUR | 84 |
| EGORRES.EG.EGTESTCD.EQ.QRSDUR | Summary (Mean) QRS Duration | 4 | EGORRES.EG.EGTESTCD.EQ.QRSDUR | 80 |
| EGORRES.EG.EGTESTCD.EQ.QTMEAN | Summary (Mean) QT Duration | 5 | EGORRES.EG.EGTESTCD.EQ.QRSDUR | 100 |
| EGORRES.EG.EGTESTCD.EQ.VRMEAN | Summary (Mean) Ventricular Rate | 1 | EGORRES.EG.EGTESTCD.EQ.QTMEAN | 356 |
| | | 2 | EGORRES.EG.EGTESTCD.EQ.QTMEAN | 372 |
| | | 3 | EGORRES.EG.EGTESTCD.EQ.QTMEAN | 396 |
| | | 4 | EGORRES.EG.EGTESTCD.EQ.QTMEAN | 382 |
| | | 5 | EGORRES.EG.EGTESTCD.EQ.QTMEAN | 400 |
| | | 6 | EGORRES.EG.EGTESTCD.EQ.QTMEAN | 360 |
| | | 1 | EGORRES.EG.EGTESTCD.EQ.VRMEAN | 86 |
| | | 2 | EGORRES.EG.EGTESTCD.EQ.VRMEAN | 77 |
| | | 3 | EGORRES.EG.EGTESTCD.EQ.VRMEAN | 69 |
| | | | | -- |

When working with auto-populated codelists, the first step is to locate these unnecessary lists in the **Codelists** tab and delete them by highlighting them all, right clicking, and then selecting “Remove row”. P21 Enterprise will also create a codelist for every variable with fewer than approximately 10 unique values, even if they are free text fields. Based on requirements, these may also need to be deleted.

MISSING BLANK ENTRY ROW

The second issue, a P21 Enterprise nuance, is not having a row to add an entry. While this can happen on any of the other tables in the module, it is more common when working with the **Terms** tab. Many times, a user will scroll to the bottom of a table to add a new entry, only to find that there is not a blank row in which to add the new value to. To fix this select any cell in the last row and hit enter and then save. That should enable the user to use the down arrow key to get a new blank row.

CRAZY TERMS ORDER

For auto-populated terms, P21 Enterprise puts them in what can only be described as a random order. Most reviewers prefer codelists to be ordered logically. If all terms deal with time, relation, or something that can be ranked I like to use that to order the codelists, otherwise I put them in alphabetical order. If the codelist only has 5 terms, this can easily be alphabetized, however, what about the LBTESTCD with 150 terms?

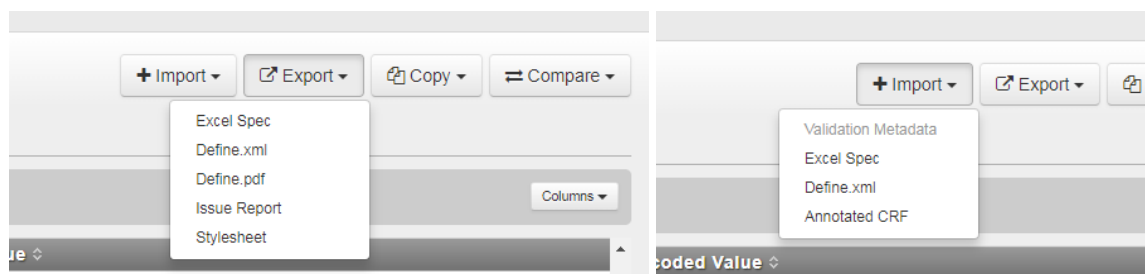
A simple solution to this is to use Microsoft Excel to reorder the values. Select the Order, Codelist, Term columns to copy and paste into a Microsoft Excel sheet. In Excel sort the data by the ‘Term’ column and create a new order column by adding the numbers 1 to n (number of terms in the codelist) in the next empty column. Sort all columns by the original order column and then copy the numbers from the “new” order column. Go back to the **Terms** tab in the P21 Enterprise and paste those numbers over the original Order number column for the code list. When you hit save it will put the codelist terms into alphabetical order.

Note: Sometimes this will cause P21 Enterprise to freeze, especially if a filter was used to limit the terms. The changes will have been saved but the application must be exited and re-opened.

NCI CODE WILL NOT POPULATE

Sometimes P21 Enterprise will not auto-populate the NCI Code on the **Terms** tab. If the **Codelists** tab is populated with an NCI code but the **Terms** tab will not auto-populate the simplest solution is to export the define by selecting **Export** from the top right and choosing **Excel Spec** from the drop down. The program will save the spec locally (probably in the computers “Downloads” folder). Once the spec is exported, immediately import it with no changes by selecting **+Import** from the top right and choosing

Excel Spec from the drop down. After you browse to the location of the spec and submit, it will cause the define to reinitialize and fixes the NCI code.



WHAT TO DO WHEN DATA CHANGE

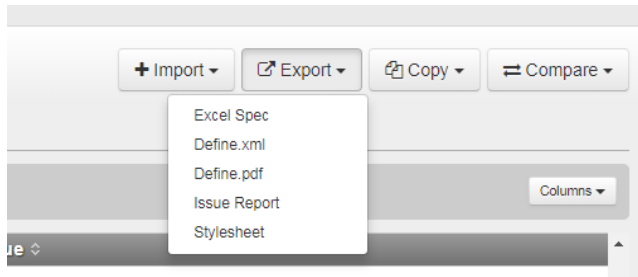
Often when working on the define.xml the data will change. P21 Enterprise has the ability to import the changed metadata after it has been validated again, however I strongly urge against this if the changes are minor. It is better to request a change list from programmers and make manual changes to the define. Here is why: When newly validated metadata are merged into P21 Enterprise, everything that it initially auto-populated will be recreated. Additionally, anything that was altered after the original auto-population will be over-written. This sounds terrible, and it is. If the changes are so large that manual updates are not practical, however, there is some good news. Anything that was written by the user in the **Methods**, **Comments**, **Codelists**, or **Terms** tabs will still exist, however, the links to the **Variables\Value Level** tab have been removed or over written. If you must bring in newly validated metadata be prepared to spend some time fixing all those links.

There is a way to make it easier (loud cheers), and that is with the **History** tab. After every major change in the define a versioning record is created on the **History** tab. It is even possible to revert to an old version. You can also compare two versions by selecting their check box and then hitting the blue **Compare** button. This will produce a report of all the differences between the two versions by tab. This can be exported to Microsoft Excel so that you can read it while making changes in the module.

| Version | Revert | Reason | Created By |
|---------------------------------------|--|--|--------------|
| <input checked="" type="checkbox"/> 5 | Current Version | Import Excel | Frank Menius |
| <input checked="" type="checkbox"/> 4 | Revert to this version | Upload CRF | Frank Menius |
| <input type="checkbox"/> 3 | Revert to this version | Reverted to previous version (1) | Frank Menius |
| <input type="checkbox"/> 2 | Revert to this version | Upload CRF | Frank Menius |
| <input type="checkbox"/> 1 | Revert to this version | Initial Creation via Import Validation | Frank Menius |

WRAPPING UP THE DEFINE

Once all the tabs in the define module have been addressed it is time to export the define.xml file. P21 Enterprise makes this easy, simply click on the Export button from the top right and select Define.xml. The application will save this file locally (probably the computer's default downloads file). The export will be named for the study. The program can also export a define.pdf and a stylesheet. After the define is exported it is important to place it, the stylesheet, and any linked documents into the location of the study XPT files and rerun the validation. This will cross reference the define with the documents and data. In the resulting P21 validation report there are likely to be define issues that did not show up in the module, such as label mismatches and values not being included in the user-defined codelists. It will be important to go through the P21 validation report and address the issues. Repeat the export and validation steps until all issues have been addressed as best as possible, and document any that are not able to be addressed in the corresponding reviewer's guide.



TIP: In order to properly open and view the define.xml file, the stylesheet and all linked documents must be in the same folder. Then right click on the define.xml file and open with an internet browser. I find that Google Chrome or Mozilla Firefox work the best.

CONCLUSION

P21 Enterprise is a useful tool that creates a consistent method of communication of the information contained in a study database with a regulatory authority. As with all tools, the more a user practices the smoother the process will become. My hope is that my experience will improve your understanding and ability to jump into this tool feet first and be successful with your documentation procedures.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Franklin E Menius Jr.
Covance
4000 Center Green Way # 300
Cary / 27513
Frank.Menius@Chiltern.com
<https://www.linkedin.com/in/frank-menius/>

Brand and product names are trademarks of their respective companies.