

Purpose

The use of electronic laboratory notebooks (ELNs) for regulated bioanalysis requires validation of numerous ELN workflows (templates). Presented here is a strategy for using common ELN functionality to validate workflows such that reviews are more efficient and comprehensive than for traditional software validation. The resulting electronic records of the complete workflow validation process are readily accessible for future reference.

Segregation of Templates During Development and Validation

Templates are built, then validated, and then are made available for use in three separate, segregated parts of the hierarchy, each having different security permissions.



Template Building Folder

- Near universal access and permissions for all users
- For designing, building, and informally testing template prototypes
- Templates in this folder are prevented from being used elsewhere

Template Validation Folder

- Only ELN Administrator may move templates into or out of this folder
- Access and permissions very limited
- QA and Mgmt. review template validation results prior to Mgmt. approval

Approved Template Folder

- Upon approval, ELN Administrator moves locked template to this folder
- Logic embedded into each template ensures that only approved templates may be executed to support bioanalytical studies

Supporting Test Records

Approved Template Validation Records Folder (Archives)

- Numerous experiments and templates executed in support of template validation are moved by ELN Administrator to corresponding archive folder



Execution of Scripts as in Traditional Software Validation... with an Electronic Twist

Template Candidate: Workflow being validated

Validation Plan Experiment: Includes instructional scripts to be executed with a specific Template Candidate

The "Validation Plan" and "Template Candidate" are executed simultaneously in separate ELN experiments. Each time the Tester performs a script designated in the Validation Plan experiment, he or she version-saves the Template Candidate experiment in which it was executed, rendering the results forever retrievable. For each individual script in the Validation Plan Experiment, the Tester records the saved version of the corresponding Template Candidate experiment, along with any comments and unexpected observations.

Unlike traditional software validation, where successful script execution is affirmed by a generic statement that "expected results were observed", the strategy described here yields electronic confirmation of observed results, as the result of each script execution is saved in a unique, retrievable version of the executed template candidate experiment.

Fig. 1 Portion of a Validation Plan Experiment Table

Objective	Test Procedure	Expected Outcome/Acceptance Criteria	Test Date	Actual Outcome	Unexpected Observations	Version number of Template Candidate Experiment	Pass/Fail
Test 10.0 Functionality: User Corrects "BAM Version" Entry in "Template Attributes Table".	Complete entries per "Data for Entry by Tester" table. Version Save. Sign off.	green stating "Yes". This indicates a match between the "BAM Version" stated in the "Instrumental_Analysis_v3" template and the "Study Summary" experiment associated with Study 0015-1049.	4-Feb-2011	Performed as Expected		11	Pass
Test 11.0 New Template Functionality: Demonstrate Field Check For Correct BAM Validation Level.	Complete entries per "Data for Entry by Tester" table. Version Save. Sign off.	In "Status" table, "Instrumental_Analysis" (experiment) ID re-appears, as both "Correct BAM Version?" and "Field Checks Passed?" entries become red, stating "No". This indicates a mismatch between the "BAM Validation Level" stated in the "Instrumental_Analysis_v3" template and the "Study Summary" experiment associated with Study 0015-1049.	4-Feb-2011	Performed as Expected		12	Pass
Test 12.0 Normal Template Functionality: User Corrects "BAM	Complete entries per "Data	In "Status" table, "Instrumental_Analysis" (experiment) ID re-appears, as both "Correct BAM Version?" and "Field Checks Passed?" entries become green stating "Yes". This		Performed			

Fig. 2 Portion of Another Validation Plan Experiment Table

Test Number	Table	Field Name	Data for Input
29	Table	BAM Version	BAM.9998.01
30	Template Attributes Table	BAM Version	BAM.9998.01
31	Template Attributes Table	BAM Validation Level	Non-regulated B
32	Template Attributes Table	BAM Validation Level	Regulated

❖ Fig. 1 lists scripts such as **A** from Validation Plan Experiment

❖ Fig. 2 is from the same Validation Plan Experiment, and lists specific input such as **B** required by Tester

❖ **C** describes expected results of script **A**

❖ Fig. 3 is a specific table from the Template Candidate Experiment

❖ **D** is the actual result in the Template Candidate Experiment which **C** describes in the Validation Plan Experiment

❖ **E** is where the Tester documents the saved version of the Template Candidate Experiment containing the result of executing script **A**

Fig. 3 Table From Template Candidate Experiment

Instrumental_Analysis	
Is BAM Template Appropriate For This Sponsor?	Yes
Is BAM Template Appropriate For This Study?	Yes
Correct BAM Version?	No
All Mandatory Fields Complete?	Yes
Field Checks Passed?	No
System Suitability Acceptable?	Yes
Template Formulas Intact?	Yes
User Entered Issue Comments	
Optional Additional Supporting Information or hyperlink	

Results and Conclusion

Strategic use of common ELN functionality can provide a straight-forward and comprehensive way to execute, review, and even reconstruct ELN workflow validations. Templates being built and validated are securely segregated from those approved for use. A complete electronic validation "package" corresponding to a specific ELN workflow, including supporting experiments and templates, is reviewed by Quality Assurance and approved by Management. This review process is efficient, yet thorough, as a view of every executed script result is a click away, and yields exactly what the tester observed during validation.