

# Agilent

## SLIMS Administration Manual - 6.7.5

Generated on 07 February 2022

---

## **SLIMS Administration Manual**

by Agilent Technologies, Inc.

<support@genohm.com>

Generated on 07 February 2022

## Table of Contents

Preface .....	ix
1. Introduction .....	1
2. SLIMS GUI .....	3
2.1. Login and Session Management .....	3
2.1.1. User Authentication .....	3
2.1.1.1. Local Authentication .....	4
2.1.1.2. LDAP System Authentication .....	5
2.1.1.3. OpenID Connect Authentication .....	6
2.1.1.4. OpenLab Shared Services Authentication .....	7
2.1.1.5. SAML Authentication .....	9
2.1.1.6. API Clients Authentication .....	11
2.1.2. Using Electronic Signatures with Authentication Providers .....	12
2.1.3. Session Time-out .....	12
2.1.4. Introduction to SLIMS Licenses .....	13
2.2. Interface Overview .....	13
2.2.1. General Global Search Features .....	16
2.2.2. Maximize / Minimize Sub-windows in Content Management .....	16
2.2.3. GUI Navigation and Help .....	17
2.3. Grids .....	17
2.3.1. Composition .....	17
2.3.2. Grid Actions .....	19
2.3.2.1. Filter .....	19
2.3.2.2. Sort .....	20
2.3.2.3. Edit Columns .....	20
2.3.2.4. Multi-Selection .....	21
2.3.2.5. Copy From Grid .....	22
2.4. Views .....	22
2.4.1. Create Customized View .....	22
2.4.2. Edit Customized View .....	25
2.4.3. Copy Customized View .....	26
2.4.4. Edit Grid Template .....	27
2.4.5. Queries .....	27
2.4.6. Query Examples .....	29
2.5. Picklists .....	33
2.5.1. Create a New Picklist .....	34
2.5.2. View Picklists .....	34
2.5.3. Copy Picklist .....	34
2.5.4. Add Contents to Picklist .....	35
2.5.5. Unlink Contents from Picklists .....	35
2.5.6. Remove a Picklist .....	36
2.6. Tools .....	36
2.6.1. Data Management using Excel .....	36
2.6.1.1. Export List to Excel .....	37
2.6.1.2. Export Selection to Excel .....	38
2.6.1.3. Create Excel Template .....	39
2.6.1.4. Editing the File .....	39
2.6.1.5. Import Excel File .....	40
2.6.1.6. Advanced Importing vs Standard Importing .....	40
2.7. Rich Text Editor .....	41

2.7.1. Saving .....	42
2.7.2. History .....	42
2.7.3. Drag and Drop .....	43
2.7.4. Hyperlinks .....	43
2.7.5. Editor Size .....	43
2.7.6. Printing .....	43
2.7.7. Spell Checker .....	44
2.8. Actions .....	44
2.8.1. Remove permanently .....	44
2.8.2. History Action .....	44
2.9. Icons Overview .....	44
2.9.1. General Use Icons .....	44
2.9.2. View Icons .....	45
2.9.3. Picklist Icons .....	45
2.9.4. Tool Icons .....	45
2.9.5. Action Icons .....	46
2.9.6. Custom Icons .....	47
3. Content Management .....	49
3.1. Content types .....	49
3.1.1. Content Type Categories .....	50
3.1.2. Moving Categories .....	52
3.2. Kit Configuration .....	53
3.3. Container types .....	54
3.3.1. Applying Custom Fields to Container Types .....	55
3.4. Reference Data .....	57
3.4.1. Reference Data Usage as Dynamic Fields .....	60
3.5. Providers .....	63
3.6. Customers .....	63
3.7. Sources .....	64
3.8. Diseases .....	64
3.9. Locations and Location Types .....	65
3.9.1. Location Types .....	65
3.9.2. Locations .....	68
3.9.3. Actions .....	69
3.9.3.1. Copy Location .....	70
3.10. Contents .....	70
3.10.1. Content Module .....	72
3.10.1.1. Filter: Also Show "X" Of .....	79
3.10.2. Create Contents .....	80
3.10.3. Create Kit .....	80
3.10.4. Create container .....	81
3.10.5. Fill Locations .....	83
3.10.6. Edit Contents .....	83
3.10.7. Copy Contents .....	84
3.10.8. Remove Contents .....	84
3.10.9. Move Contents .....	85
3.10.10. Derive Contents .....	86
3.10.10.1. Multiple Contents Derived at Once .....	88
3.10.11. Aliquot Contents .....	89
3.10.12. Dilute Contents .....	90
3.10.13. Mix Contents .....	92
3.10.14. Change Parent .....	92

3.10.15. Clear Parent .....	93
3.10.16. Merge Two Contents .....	94
3.10.17. Enroll Content .....	95
3.10.18. Create Results .....	96
3.11. Studies .....	97
3.11.1. Studies .....	97
3.11.2. Content event types .....	98
3.11.2.1. Standard Content Event Types .....	98
3.11.2.2. Flag Content Event Types .....	99
3.11.2.3. Adding Flags Automatically in Workflows .....	100
3.11.2.4. Explanation of Consequences .....	101
3.11.3. Content events .....	101
3.11.4. Case report form templates .....	102
3.11.5. Case Report Forms .....	103
3.12. Stability Study Design .....	103
3.12.1. Stability study templates .....	104
3.12.1.1. Stability Study Template Details Tab .....	107
3.12.1.2. Root Specifications Tab .....	107
3.12.2. Storage condition templates .....	109
3.12.2.1. Storage Condition Details .....	109
3.12.2.2. Timepoint templates .....	110
3.12.3. Stability Study Timepoint Template Tab .....	110
3.12.3.1. Timepoint Specifications .....	110
3.12.4. Stability Study Order Types .....	111
3.12.4.1. Pulling and Measurement Orders .....	111
3.12.5. Copying a Stability Study Design .....	112
3.12.6. Versioning a Stability Study Template .....	112
3.12.7. Customize Grids and Reports for Stability Studies .....	113
3.13. Stability Studies .....	113
3.13.1. Stability Study Module Navigation .....	114
3.13.1.1. Selected Tree Details Tab .....	115
3.13.1.2. Enrolled Batches Tab .....	115
3.13.1.3. Pulled Contents Tab .....	116
3.13.1.4. Results Tab .....	116
3.13.1.5. Reports Tab .....	117
3.13.2. Stability Study Order Usage .....	118
3.13.2.1. Behavior of Measuring Orders by Themselves .....	119
3.14. Inventory .....	119
3.14.1. Inventory types Module .....	119
3.14.2. Inventory Module .....	120
3.14.3. Content Module Expansion .....	120
3.14.4. Consuming Inventory .....	121
4. Electronic Lab Notebook .....	122
4.1. Tests and Test groups .....	122
4.1.1. Tests .....	122
4.1.1.1. Dynamic Tests .....	124
4.1.1.2. Test Results .....	124
4.1.1.3. Import and Export Tests .....	125
4.1.2. Test groups .....	125
4.2. Instruments and Instrument types .....	125
4.2.1. Instruments for ELN .....	126
4.2.2. Instruments for Workflows .....	126

4.2.3. Instrument types .....	127
4.2.4. Instrument Calibration .....	127
4.3. Protocols .....	128
4.3.1. Protocol Tab .....	129
4.3.2. Protocol steps .....	130
4.3.3. Sub protocols .....	136
4.3.4. Ad hoc steps .....	137
4.3.5. Final electronic signature step .....	137
4.4. Simplified Protocol Management .....	138
4.5. OpenLab Software Suite Integrations .....	140
4.5.1. Sample Scheduler for OpenLab Integration .....	140
4.5.1.1. Prerequisites for Installation .....	140
4.5.1.2. OpenLab Authentication and Automations .....	141
4.5.1.3. Configuring a Sample Scheduler Workflow .....	141
4.5.1.4. How to Execute an OpenLab Sample Scheduler Protocol .....	145
4.5.2. OpenLab CDS Integration .....	147
4.6. SOPs .....	148
4.7. Electronic Lab Notebook (Routine) .....	149
4.7.1. Projects .....	151
4.7.2. Experiments .....	151
4.7.3. Simplified ELN .....	151
4.7.3.1. Blocks .....	154
4.7.3.2. Adding Blocks .....	157
4.7.3.3. Print Simplified Experiments .....	157
4.7.4. Protocol Runs .....	158
4.7.4.1. Ad Hoc Steps .....	158
4.7.5. Protocol run steps .....	159
4.7.5.1. Sub protocol Step .....	161
4.7.6. Results .....	161
4.7.6.1. Using Value Expressions .....	163
4.7.6.2. Manual Creation of Results .....	164
4.7.6.3. Status of Results .....	164
4.7.6.4. Graphical Representation .....	165
4.7.7. Track Usage .....	166
4.7.8. Notes .....	167
5. Order Management .....	168
5.1. Order Types .....	168
5.2. Orders .....	170
5.2.1. Single Content Per Order .....	173
5.2.2. Order Life Cycle .....	174
5.3. Requestables and Requestable groups .....	179
5.3.1. Requestables .....	180
5.3.2. Requestable groups .....	181
5.4. Products and specifications .....	181
5.4.1. Product Tab .....	183
5.4.2. Specifications Tab .....	183
5.4.3. Importing and Exporting Specification Rules .....	185
6. Workflows .....	187
6.1. Workflow Management .....	187
6.1.1. Workflow Graph Tab .....	188
6.1.2. Requestables Tab .....	191
6.2. Workflow Module .....	192

6.2.1. Searching Records in a Workflow .....	193
6.2.2. Linking and Unlinking Content to/from Workflows .....	194
6.2.3. Using Flags in a Workflow .....	197
6.2.3.1. Flag Warnings and Errors .....	198
6.2.4. Simplified Protocols for Workflows .....	198
6.2.5. Workflow chain of custody .....	201
6.3. Analytical Workflows .....	202
6.3.1. Configuration of Analytical Workflows .....	203
6.3.1.1. Configure Tests .....	203
6.3.1.2. Configure Requestables .....	203
6.3.2. Workflow Management for Analytical Workflows .....	204
6.3.2.1. Analytical Workflow Graph Tab .....	206
6.3.2.2. Analytical methods Tab .....	207
6.3.2.3. Orders Used for Analytical Workflows .....	207
6.3.3. Processing Analytical Results .....	208
6.3.4. QC Orders for Analytical Workflows .....	209
6.3.4.1. Using QC Orders for Analytical Workflows .....	210
6.3.5. Analytical Use Case .....	210
6.3.6. Using Analytical Workflows .....	217
6.3.6.1. Using QC Orders in Analytical Workflows .....	219
6.4. SLIMS-NGS .....	220
6.4.1. NGS Package Installation .....	220
6.4.2. Creation of NGS Instruments .....	222
6.4.3. DNA index sets .....	222
6.4.4. Plate index designs .....	223
6.4.5. Sequencing settings .....	224
6.4.6. MiSeq, HiSeq, NextSeq, and NovaSeq Implementations .....	227
6.4.6.1. General Implementation Configuration .....	227
6.4.6.2. NGS: Library preparation Protocol .....	229
6.4.6.3. NGS: MiSeq execution / NGS: NextSeq execution Protocol .....	229
6.4.6.4. NGS: NovaSeq execution / NGS: HiSeq execution Protocol .....	230
6.4.6.5. Analysis Protocol .....	230
6.4.6.6. NGS Implementation Use .....	231
6.5. Runs .....	235
7. Quality Control .....	237
7.1. QC Templates .....	237
7.1.1. QC Templates of Type Blanco .....	238
7.1.2. QC Templates of Type AQC .....	238
7.1.3. QC Templates of Type Raw Data .....	240
7.1.4. QC During the Night .....	241
7.2. QC .....	241
7.2.1. QC by Card .....	242
7.2.2. QC by Template .....	242
7.2.3. All QC .....	242
7.2.4. Add QC Value .....	242
7.2.5. QC Value Actions .....	242
7.2.6. QC Rules .....	243
7.2.7. QC Statistics Check .....	243
8. Worklist .....	244
8.1. Worklist templates .....	244
8.1.1. Worklist Template - Name .....	244
8.1.2. Layout .....	245

8.1.2.1. Worklist Selection .....	245
8.1.2.2. Worklist .....	246
8.1.2.3. Print .....	246
8.1.3. Sort .....	246
8.1.4. Criteria .....	246
8.1.5. QC .....	246
8.2. Worklists .....	247
9. Reporting and Grids .....	251
9.1. Report Templates .....	251
9.1.1. Core Report Templates .....	251
9.1.1.1. The Report Template Form .....	251
9.1.1.2. Meta Tree .....	252
9.1.1.3. Attachments .....	252
9.1.1.4. Versioning .....	253
9.1.1.5. Generating XML Datasources .....	253
9.1.2. SLIMS GATE Report Templates .....	253
9.1.2.1. The SLIMS GATE Report Template Form .....	253
9.1.2.2. Examples of Report Templates .....	254
9.1.2.3. Where to place Jasper Templates for SLIMS GATE Reports .....	254
9.1.2.4. Examples of XML Datasources .....	255
9.1.3. Export (ZIP) .....	257
9.2. Reports .....	257
9.2.1. The Report Module .....	257
9.2.2. Creation of the Template File .....	257
9.2.3. Core Reports .....	257
9.2.4. SLIMS GATE Reports .....	258
9.2.4.1. Order Reports .....	258
9.2.4.2. ELN Reports .....	258
9.2.5. Export (ZIP) .....	259
9.3. Grid templates .....	259
9.3.1. Stability Study Pulled Contents Grid Templates .....	260
9.3.2. Stability Study Result Grid Templates .....	261
9.3.3. Queue Grid Templates .....	261
9.3.4. Multiple Facets in a Grid .....	261
9.4. Grid .....	262
9.5. Meta Tree .....	263
9.5.1. Root .....	263
9.5.2. Relations .....	263
9.5.3. Criteria .....	263
9.5.4. Having (Aggregators) .....	263
9.5.5. Joins .....	264
9.5.6. One Entry Per Node .....	264
9.6. Dashboard .....	264
9.6.1. Widgets Module .....	264
9.6.2. Grid Widgets .....	264
9.6.2.1. Columns Table .....	265
9.6.3. Graph Widgets .....	265
9.6.4. Rich Text Widgets .....	265
9.6.5. SLIMS GATE Widgets .....	266
9.6.6. Vaadin Flow Widgets .....	266
9.6.7. Macro Widgets .....	266
9.6.8. Create a Dashboard .....	266

10. Miscellaneous .....	269
10.1. Access Rights .....	269
10.1.1. Users .....	269
10.1.1.1. Group Rights Overview .....	271
10.1.1.2. User Lockout .....	271
10.1.1.3. Password Complexity .....	271
10.1.1.4. Password Expiration Period .....	271
10.1.1.5. Group Access Rights .....	272
10.1.2. Groups .....	272
10.1.2.1. Adding Users to a Group .....	272
10.1.2.2. Adding Groups to a Group .....	273
10.1.2.3. User Rights Overview .....	273
10.1.3. Roles .....	273
10.1.4. Defining Access Rights .....	274
10.1.5. Update access rights on records .....	275
10.1.6. ELN Sharing .....	276
10.1.6.1. Sharing Runs .....	276
10.1.6.2. Shared with Me .....	276
10.1.6.3. Shared with Others .....	277
10.2. Analysis Modules .....	277
10.2.1. Analysis Template .....	277
10.2.2. Analysis .....	278
10.3. Attachments .....	278
10.3.1. Create Attachments .....	279
10.3.2. Open and Download Attachments .....	280
10.3.3. Preview Attachments .....	280
10.3.4. Attachment Links .....	281
10.3.5. Attachment types .....	281
10.3.6. Creating Agilent OpenLab ECM Attachment Types .....	282
10.3.7. Drag and Drop .....	283
10.3.8. Binary Search .....	284
10.4. Notification Setup .....	285
10.4.1. Email Templates Module .....	285
10.4.1.1. Template Expression .....	285
10.4.1.2. Scheduled Jobs Module .....	287
10.5. Fields .....	289
10.5.1. Actions Available for Fields .....	289
10.5.1.1. Copying Fields .....	289
10.5.1.2. Import and Export Fields .....	290
10.5.1.3. Field Removal Options .....	290
10.5.2. Default Fields .....	291
10.5.3. Custom Fields .....	291
10.5.4. Rich Text Fields .....	299
10.5.5. Datatypes .....	299
10.5.5.1. Short text .....	299
10.5.5.2. Status .....	300
10.5.5.3. Text box .....	300
10.5.5.4. Rich Text .....	300
10.5.5.5. Whole number .....	300
10.5.5.6. Decimal number .....	300
10.5.5.7. Date .....	300
10.5.5.8. Time .....	300

10.5.5.9. Date and time .....	301
10.5.5.10. Date Difference .....	301
10.5.5.11. Checkbox .....	301
10.5.5.12. Fixed choice .....	301
10.5.5.13. Dynamic choice .....	302
10.5.5.14. URL .....	306
10.5.5.15. Quantity .....	306
10.5.5.16. Comparator Quantity .....	309
10.5.5.17. Barcode .....	309
10.5.5.18. Attachment .....	309
10.5.5.19. Ontology Term .....	309
10.6. History .....	310
10.6.1. Types of History .....	310
10.6.2. Creation of History Records .....	311
10.6.3. History Save .....	311
10.6.4. History Event .....	311
10.6.5. History User Interface .....	311
10.6.5.1. View Filters for History .....	314
10.7. Holidays .....	314
10.8. Lab Settings .....	314
10.8.1. Attachments .....	315
10.8.2. SLIMS .....	315
10.8.3. Orders .....	317
10.8.4. Contents .....	317
10.8.5. Electronic Lab Notes .....	317
10.8.6. Global search .....	317
10.8.7. Users .....	318
10.8.8. API .....	318
10.8.9. Macros .....	318
10.8.10. SLIMS Share .....	318
10.8.11. Locations .....	318
10.8.12. Electronic Signature .....	318
10.8.13. Tests .....	319
10.8.14. History Formatting .....	319
10.9. License .....	319
10.9.1. Chart Tab .....	319
10.9.2. License Tab .....	320
10.9.3. History Tab .....	321
10.10. Notifications .....	321
10.11. Ontology Graphs .....	321
10.12. SLIMS Configuration Import Options .....	322
10.12.1. Package Browser .....	323
10.12.1.1. Imported Packages .....	323
10.12.2. SLIMS Share .....	323
10.12.2.1. Create Package Tab .....	324
10.12.2.2. Import Package Tab .....	328
10.12.2.3. Installed Packages Tab .....	329
10.12.2.4. Importing and Exporting Workflow Protocol Packages .....	331
10.12.3. SLIMS Store .....	333
10.13. Layout .....	333
10.13.1. Add Master Layout .....	334
10.13.2. Alter an existing Master Layout .....	334

10.13.3. Enforcing a Master Layout .....	334
10.13.4. Promote Changes .....	335
10.13.5. Transform into New Layout .....	336
10.13.6. Apply an Alternate Layout .....	336
10.13.7. Configure Visible Fields in a Layout .....	336
10.14. Plugins Module .....	336
10.14.1. Plugin Access rights .....	337
10.14.2. Plugin Module Layout .....	337
10.14.3. Types of Plugins .....	338
10.15. Printing .....	338
10.15.1. Printers .....	339
10.15.2. Label Templates .....	340
10.15.2.1. Standard Label Templates .....	340
10.15.2.2. Scripted Label Templates .....	343
10.15.3. Calibration .....	345
10.15.4. Print Labels .....	346
10.16. Rules .....	346
10.16.1. Conditional value expression .....	347
10.16.2. Result .....	348
10.16.2.1. Match Specification .....	349
10.16.2.2. Script Specification .....	350
10.16.3. Status .....	351
10.16.4. QC .....	352
10.16.5. Compare Results .....	352
10.16.6. Mix Rules .....	352
10.17. Scan .....	353
10.18. Macro .....	354
10.19. Sequences .....	360
10.20. Status .....	361
10.20.1. Status Workflows .....	361
10.20.2. Publication Status Workflows .....	363
10.20.2.1. Publication Statuses .....	364
10.20.2.2. Publication Status Transitions .....	364
10.20.3. Statuses .....	364
10.20.4. Status transitions .....	367
10.21. Study Design Module .....	369
10.21.1. Visit Types .....	370
10.21.2. Specifications .....	371
10.22. Unique Identifiers .....	372
10.22.1. Explanation of SLIMS Default Unique Identifiers Generation .....	373
10.23. Units .....	375
10.23.1. Importance of Unit Definition .....	377
10.24. Variants .....	378
10.24.1. Enable Variants .....	378
10.24.2. Add Variants .....	378
10.24.3. Add Variant observation .....	378
10.25. Versioning .....	379
10.25.1. Create/Edit Versions .....	379
10.25.2. Versions Structure in SLIMS .....	380
11. Groovy Scripting .....	382
11.1. Introduction to Groovy .....	382
11.2. Groovy Editor .....	383

11.2.1. Fields Where Groovy is Available .....	383
11.2.2. Editor .....	383
11.3. Build-in SLIMS Methods and Variables .....	384
11.3.1. oldValue .....	385
11.3.2. empty .....	385
11.3.3. NA .....	385
11.3.4. log .....	386
11.3.4.1. Methods Summary .....	386
11.3.4.2. log.error .....	386
11.3.4.3. log.warn .....	386
11.3.4.4. log.info .....	386
11.3.4.5. log.debug .....	387
11.3.5. tableName .....	387
11.3.5.1. Methods Summary .....	387
11.3.5.2. tableName.get .....	387
11.3.6. authenticatedUser .....	388
11.3.6.1. Methods Summary .....	388
11.3.6.2. authenticatedUser.groupNames .....	388
11.3.6.3. authenticatedUser.role_name .....	388
11.3.6.4. authenticatedUser.role_updateLevel .....	389
11.3.6.5. authenticatedUser.user_email .....	389
11.3.6.6. authenticatedUser.user_pk .....	389
11.3.6.7. authenticatedUser.user_userName .....	389
11.3.7. simultaneousChanges .....	389
11.3.8. Quantity and Unit .....	389
11.3.8.1. quantity(amount, unit) .....	389
11.3.8.2. unit(String abbreviation) .....	390
11.3.9. Comparator Quantity and Position .....	390
11.3.9.1. comparatorQuantity(position, amount, unit) .....	390
11.3.9.2. ComparatorQuantityPosition .....	391
11.3.10. triggerRecalculation .....	391
11.3.11. findByUniqueIdentifier .....	392
11.3.12. macroExecutionDetails .....	392
11.3.13. daoHelper .....	393
11.3.13.1. Methods Summary .....	394
11.3.13.2. daoHelper.fetch .....	394
11.3.13.3. daoHelper.fetch .....	394
11.3.13.4. daoHelper.find .....	395
11.3.13.5. daoHelper.getFetchRequest .....	395
11.3.14. FetchRequest .....	396
11.3.14.1. Methods Summary .....	396
11.3.14.2. FetchRequest.setCriterion .....	396
11.3.14.3. FetchRequest.setAddSecurityCriterion .....	396
11.3.15. SlimsRestrictions .....	397
11.3.15.1. Methods Summary .....	397
11.3.15.2. SlimsRestrictions.equals .....	398
11.3.15.3. SlimsRestrictions gt/lt/ge/le .....	398
11.3.15.4. SlimsRestrictions.in .....	399
11.3.15.5. SlimsRestrictions.like .....	399
11.3.15.6. SlimsRestrictions.not .....	399
11.3.15.7. SlimsRestrictions.and .....	400
11.3.15.8. SlimsRestrictions.or .....	400

11.3.16. Fetch Contents Affecting/Affected By a Content .....	400
11.3.16.1. Fetch Content(s) Affected By a Content .....	401
11.3.16.2. Fetch Content(s) Affecting a Content .....	401
11.3.16.3. Fetch Both Content(s) Affecting and Affected By a Content .....	401
11.3.17. sequences .....	401
11.3.17.1. Methods Summary .....	402
11.3.17.2. sequences.next .....	402
11.3.18. ok .....	402
11.3.19. warn .....	402
11.3.20. error .....	403
11.3.21. notify .....	403
11.3.22. email .....	404
11.3.23. holidayHelper .....	404
11.4. Examples .....	405
11.4.1. Fields, datatype Quantity, table Content: calculate mass from Volume and Concentration field values .....	405
11.4.2. Fields, datatype Decimal number, table Content: calculate the differ- ence between two quantity fields and converts it to double .....	405
11.4.3. Field <code>cntn_fk_group</code> , table Content: set default group if unique .....	406
11.4.4. Fields, datatype Date, table Content: calculate expiration date from pro- duction date and duration .....	406
11.4.5. Rules, Order Status: perform checks on requestables when scheduling an order .....	406
11.4.6. Rules, Conditional value expression: check that all results are verified when stopping a step .....	407
11.4.7. Rules, Conditional value expression: requires a value for a Protocol run step field when stopping the step .....	407
11.4.8. Rules, Conditional value expression: checks that all contents are re- moved when stopping a Protocol run step .....	408
11.4.9. Rules, Result Status: requires a comment for some kind of results .....	408
11.4.10. Fields, datatype Dynamic choice: example of Dynamic filter expres- sion .....	408
11.4.11. Test: calculate from values of other tests .....	409
11.4.12. Test result: calculate result for all results of quantity test .....	409
11.5. Test Groovy Scripts .....	409
12. SLIMS GATE Flows Monitoring .....	411
12.1. SLIMS GATE Flows Monitoring .....	411
13. Specific Use Cases .....	413
13.1. Pedigrees .....	413
13.1.1. Required Setup .....	413
13.1.2. Add/Set Relationship .....	413
13.1.3. View Relationships .....	413
13.1.4. View Pedigrees .....	416
13.1.5. Custom Configurations .....	417
13.1.5.1. Pedigree Camel-Context Configuration .....	419
13.2. SLIMS GATE Flows .....	421
13.3. SLIMS REST API .....	421
13.4. Python API Integration .....	422
13.5. Vaadin Integration .....	422
13.6. Prometheus Integration for Health Monitoring .....	422
13.7. Alamut Integration with SLIMS .....	422
13.7.1. Usage of Alamut in SLIMS .....	422

13.7.2. Configuring Alamut Integration .....	423
13.8. Advanced Advice for Queue Aggregators in Protocols .....	424
13.9. Grid Template Special Context Filters .....	425
13.10. Reagents Default Usage Expressions .....	426
14. FAQ .....	428
14.1. Miscellaneous .....	428
14.1.1. Transport Error when Updating Multiple Records .....	428
14.1.2. Scanner is not Reading Certain Characters .....	428
About this Book .....	cdxxx

---

# List of Figures

2.1. Redirect URI meanings .....	11
2.2. Interface overview .....	14
2.3. Collapsed Filters .....	17
2.4. Grid Components .....	18
2.5. Grid Structure .....	19
2.6. Actions Menu .....	21
2.7. Editing a filter with different clauses .....	24
2.8. Query Language Structure .....	29
2.9. Example of relations .....	30
2.10. Querying all subjects having both parents .....	31
2.11. Querying all records having a DNA sample .....	31
2.12. Querying all records having a result .....	32
2.13. Querying all subjects having DNA contained in the location 'B0001' .....	33
2.14. View Picklists .....	34
2.15. Link Content Options .....	35
2.16. Export to Import flow using Excel Files .....	37
2.17. Rich text fields and history records creation .....	43
2.18. Access Icons for an Element .....	47
2.19. Upload Custom Icons .....	48
3.1. Diagram for example; content types that can be in a container .....	57
3.2. Add a Reference Data Type and Records .....	59
3.3. Use Reference Data in a Field .....	61
3.4. Import Location Types from Excel .....	68
3.5. Content Module .....	72
3.6. Content Description .....	74
3.7. Location Detail .....	74
3.8. Illustration of NGS Mix Flow .....	76
3.9. Derivation Tree with a DNA Content Selected in Content Grid .....	77
3.10. Print Derivation Tab .....	77
3.11. Move Content pop-up window .....	85
3.12. Dilute Content Pop-up Window .....	91
3.13. Mix window .....	92
3.14. Stability Study Template Concept .....	104
3.15. Stability Study Flow from Batch Enrollment to Measuring .....	118
4.1. SOP and Protocol in SLIMS .....	128
4.2. Breadcrumbs Example: .....	149
4.3. Add Experiment with Simplified Protocol: .....	152
4.4. Parts of an Experiment: .....	153
4.5. Elements of a Protocol run step .....	159
4.6. A dynamic test element .....	162
5.1. Orders Module .....	171
5.2. Single Content Per Order Layout .....	174
5.3. Order Workflow .....	176
6.1. Workflow Management Tabs .....	188
6.2. Simple Protocol Steps Example .....	199
6.3. Steps with Single Derivation .....	200
6.4. Steps with Two Derivations .....	201
6.5. Building Blocks of an Analytical Workflow .....	203
6.6. Analytical Workflow Areas .....	205

6.7. Analytical Results Overview .....	208
6.8. Building Blocks of a QC Analytical Workflow .....	209
6.9. Water Preparation Workflow with Protocol Run Grid .....	214
6.10. Queue Grid Template Configuration .....	215
6.11. Results in Water Preparation step .....	218
6.12. Skipped Index Fill Method Order .....	224
6.13. Pickup Results Diagram .....	235
7.1. QC Workflow .....	237
8.1. Workflow of a Worklist .....	248
8.2. Protocol step status in Worklists .....	250
9.1. Generate Report .....	258
9.2. Sort Columns in a Grid Template .....	260
9.3. Create Multiple Facets .....	262
9.4. New Dashboard Options .....	266
9.5. Add Dashboard Widgets .....	267
9.6. Completed Dashboard Display .....	267
10.1. Uploading Attachments in SLIMS .....	282
10.2. Fields Module .....	289
10.3. Example of 'Show in dropdown lists' .....	296
10.4. Example of 'Show in extended tooltips' .....	296
10.5. Selected Categories VS Selected Types .....	298
10.6. URI Example with Localhost as the Port .....	316
10.7. URI Example with Only Port Defined .....	316
10.8. Complex URI Example .....	317
10.9. SLIMS Share Metagraph .....	324
10.10. Create Packages .....	325
10.11. Import Packages .....	328
10.12. Installed Packages .....	330
10.13. Import Using Store .....	333
10.14. Alteration and Enforcement of a Master Layout .....	335
10.15. Label template parameters .....	341
10.16. Scan Module .....	354
10.17. Macro .....	355
10.18. Macro Steps .....	358
10.19. Study Design Module .....	370
10.20. Unique Identifier Example .....	375
10.21. Version Options .....	379
10.22. Structure of Versions in SLIMS .....	380
10.23. Example on how an SOP is stored in SLIMS: .....	381
11.1. Test value expression dependence .....	409
13.1. Example of a pedigree obtained with SLIMS: .....	417

---

## List of Tables

2.1. Authentication Permissions .....	4
2.2. Edit Grid Template Permissions .....	27
2.3. Picklist Permissions .....	33
2.4. Excel Import Permissions .....	36
3.1. Required Permissions for Content Types Module .....	49
3.2. Configure Categories Permissions .....	50
3.3. Kit Category Permissions .....	53
3.4. Container Type Permissions .....	55
3.5. Required Permissions for Reference Data Module .....	58
3.6. Required Permissions for Providers Module .....	63
3.7. Required Permissions for Customers Module .....	64
3.8. Required Permissions for Sources Module .....	64
3.9. Required Permissions for Diseases Module .....	64
3.10. Required Permissions for Location Modules .....	65
3.11. Required Permissions for Content Module .....	70
3.12. Optional Permissions for Content Module .....	70
3.13. Containers Permissions .....	82
3.14. Optional Derivation Permissions .....	87
3.15. Change Parent Permissions .....	93
3.16. Merge Contents Permissions .....	94
3.17. Create Results Permissions .....	96
3.18. Required Permissions for Studies Module .....	97
3.19. Optional Permissions for Studies Module .....	97
3.20. Required Permissions for Content Event Types Module .....	98
3.21. Content Events Permissions .....	101
3.22. Required Permissions for CRF Templates Module .....	102
3.23. Required Permissions for Stability Study Design Module .....	103
3.24. Required Permissions for Stability studies Module .....	114
3.25. Inventory Type Permissions .....	120
3.26. Inventory Permissions .....	120
4.1. Tests and Test Groups Permissions .....	122
4.2. Required Permissions for Instrument Modules .....	126
4.3. Required Permissions for Protocols Module .....	128
4.4. Required Permissions for Simplified Protocols Module .....	138
4.5. Fixed Sample Scheduler Attributes .....	142
4.6. Sample Scheduler Attributes for Configuration .....	143
4.7. Grid Template Example - Sample Scheduler Field Mapping Simple .....	144
4.8. Grid Template Example - Sample Scheduler Field Mapping Complex .....	145
4.9. Required Permissions for SOPs Module .....	148
4.10. Electronic Lab Notebook Permissions .....	149
4.11. Optional Permissions for Protocol Runs .....	158
5.1. Required Permissions for Order Types Module .....	168
5.2. Orders Module Permissions .....	170
5.3. Order Life Cycle Permissions .....	175
5.4. Requestables Permissions .....	179
5.5. Required Permissions for Products and Specifications Module .....	182
6.1. Required Permissions for Workflow Management Module .....	187
6.2. Workflow Module Permissions .....	192
6.3. Workflow Module Optional Permissions .....	192

6.4. Main Product: Aspirin .....	213
6.5. Sub-Product: Aspirin for Australia .....	213
6.6. Sub-Product: Aspirin for CH .....	213
6.7. Grid Template - Columns Table .....	215
6.8. Required Permissions for Setup with NGS Modules .....	220
7.1. Required Permissions for QC Templates Module .....	237
7.2. QC Result Display Permissions .....	239
7.3. QC Required Permissions .....	241
7.4. QC Optional Permissions .....	241
8.1. Worklist Template Module Permissions .....	244
8.2. Worklist Module Permissions .....	247
9.1. Permissions for Report Templates Module .....	251
9.2. SLIMS GATE Report Template .....	254
9.3. Grid Module Permissions .....	262
9.4. Dashboard Module Permissions .....	266
10.1. Required Permissions for Ending User Sessions .....	272
10.2. Create, Access, Update and Remove levels .....	275
10.3. Example of access rights .....	275
10.4. Analysis Module Permissions .....	277
10.5. Attachment Permissions .....	279
10.6. Required Permissions for Scheduled Jobs Module .....	287
10.7. Abbreviations of SLIMS Tables .....	291
10.8. Correlation between dynamic filter expression operators and filter keywords .....	303
10.9. Possible Value Tables for Dynamic Choice Fields .....	304
10.10. Required Permissions for Lab Settings Module .....	315
10.11. Required Permissions for Ontology Graphs Module .....	322
10.12. Permissions for Package Importing Modules .....	322
10.13. Plugin Access rights .....	337
10.14. Required Permissions for Printers and Label Templates .....	338
10.15. Required Permissions for Rules Module .....	346
10.16. Rule Evaluations Module Permissions .....	348
10.17. Operators available for rules .....	349
10.18. Scan Module Permissions .....	353
10.19. Required Permissions for Sequences Module .....	360
10.20. Status Workflow Access rights .....	361
10.21. Permissions for Study Design Module .....	370
10.22. Variants Permissions .....	378
11.1. Permissions for Value Expressions .....	383
11.2. Build-in SLIMS Variables .....	384
12.1. SLIMS GATE Permissions .....	411
13.1. SLIMS REST Access rights .....	421

---

# Preface

This document covers fundamental and advanced concepts and strategies for using the product, as well as details about the features of SLIMS.

---

# 1. Introduction

In May 2018, Agilent Technologies, Inc. acquired Genohm (est. 2002) to enhance their software portfolio. The combination of Genohm's LIMS, workflow management, and ELN allows Agilent to bring greater context to analytical data and generate results for scientists more efficiently. SLIMS and Agilent's OpenLab products work together to provide services that enhance the operations of modern laboratories.

Building upon an extensive bioinformatics consulting expertise within the life sciences R&D world, the unique SLIMS environment was put together to support the ever-increasing complexity of handling, analyzing, and understanding biological data. SLIMS (Simple Laboratory Information Management System) accommodates this by providing research with one integrated SLIMS/VIEWS environment to tackle today's research in domains like cell and systems biology, translational medicine, genomics, and proteomics. After the success encountered at the first release, SLIMS continues to upgrade to adapt its features for diverse research laboratory and hospital settings.

SLIMS was designed with flexibility as its main design principle, ready to be adapted to any type of laboratory. The SLIMS configuration can be permanently modified and fine-tuned by SLIMS users, and is therefore optimally suited for dynamic lab environments. SLIMS has cloud-based installation options as well, allowing a fast out-of-the-box download via packages, and much customization provided within the UI. The system is competitively and transparently priced, without any hidden costs. SLIMS has been designed with an efficient user experience in mind. Moreover, SLIMS was built with the latest web 2.0 technologies and runs in any browser, but with the feel of a desktop application.

This documentation offers explanation and technical reference for the SLIMS software. Using this program's features, you'll learn how to manage your entire laboratory while keeping track of your information.

Note that the available icons and features described in this manual depend on your SLIMS instance's configuration.

## Notices

### Manual Part Number

R4526A-AM6.7.5.0

07 February 2022

### Copyright

2022 © Agilent Technologies, Inc.

No part of this manual may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Agilent Technologies, Inc. as governed by the United States and international copyright laws.

Agilent Technologies, Inc.

5301 Stevens Creek Blvd.

Santa Clara, CA 95051

USA

### **Software Revision**

This guide is valid for the 6.7.5 revision of Agilent SLIMS software.

### **Warranty**

The material contained in this document is provided 'as is,' and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

### **Technology Licenses**

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

### **Restricted Rights Legend**

U.S. Government Restricted Rights. Software and technical data rights granted to the federal government include only those rights customarily provided to end user customers. Agilent provides this customary commercial license in Software and technical data pursuant to FAR 12.211 (Technical Data) and 12.212 (Computer Software) and, for the Department of Defense, DFARS 252.227-7105 (Technical Data - Commercial Items) and DFARS 227.7202-3 (Rights in Commercial Computer Software or Computer Software Documentation).

### **Safety Notices**

⚠ A **CAUTION** notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a **CAUTION** notice until the indicated conditions are fully understood and met.

⚠ A **WARNING** notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in personal injury or death. Do not proceed beyond a **WARNING** notice until the indicated conditions are fully understood and met.

---

## 2. SLIMS GUI

SLIMS GUI chapter contains explanations about all basic features in SLIMS. The most common actions and the associated icons are described.

### 2.1. Login and Session Management

The SLIMS login screen contains:

- *End User License Agreement*: The user must read and accept the End User License Agreement once before they login for the first time. After it is accepted the first time, the EULA does not reappear unless a new version is created, and then the user simply reads the new version and clicks Accept to continue to login.
- *Login Fields*: A unique combination of User Name and Password is required for each user to login to SLIMS. This is dependent on the authentication methods configured in the SLIMS instance. The same user can login on multiple computers. If a user logs out on one computer, he is not logged out on all computers.  
The number of incorrect login attempts allowed before the user is locked out can be specified in the *Authentication module*.
- *SLIMS Version and Update Date*: This presents basic information about the implemented SLIMS version and the date it was compiled.
- *Manual*: The SLIMS reference manual is available in PDF and HTML formats. A link to the documentation is available at the bottom of the login page. The documentation version is updated with every new SLIMS version.

#### 2.1.1. User Authentication

Authentication governs the safe process of validating new users and creating their accounts. There are several methods that can be used with SLIMS to create new accounts and preset the default values for all new users. SLIMS offers different methods because using a third party provider like OpenID Connect, LDAP, OpenLab Shared Services, and SAML reduces the number of credentials users have to remember.

The authentication methods available in SLIMS are: Local authentication, LDAP system, OpenID Connect (Google, Okta and identity providers supporting OpenID Connect Discovery), OpenLab (for Agilent OpenLab Shared Services), SAML 2.0, and API Clients. All of the authentication methods are configured first within the Authentication module to set up the connection between SLIMS and the desired service, and set the default values for a new user's role, license, and whether they are automatically or manually activated. The user will see the SLIMS User Name and Password login fields on the SLIMS login page for all authentication methods (except API Clients), and will see the additional buttons to login with their Google, Okta or another OpenID Connect Provider account only if OpenID Connect is configured.

This module requires the following Role Access rights to work:

**Table 2.1. Authentication Permissions**

Access rights Category	Access rights Name	Purpose
Miscellaneous	Authentication	When enabled, the user can access the Authentication module to add and configure user account authentication methods for the SLIMS instance.

To set up authentication, the five tabs in the Authentication module can be used to turn on and configure the default settings for each desired method. SLIMS Local Authentication cannot be disabled and will always be available as an authentication method. This allows the authentication method to be reverted to local authentication in case there is ever a failure or outage of a third-party authentication system (ex: LDAP server down). One or more authentication methods can be used for a SLIMS instance, but an individual user can only be set up to use one method at a time, and can be changed if needed in the Users module. The below sections detail how to set up each authentication method in the module, how it impacts the login window as the end user sees it, and the login process.

When using any of the LDAP, OpenID Connect, OpenLab and SAML authentication methods, once the external authentication provider is configured, the first successful authentication will create the user account. The account will be created with the user's name pulled from the external provider into the SLIMS "Full Name" field in the Users module. After that, the user is activated if the chosen method is configured to activate users automatically, and the login is successful. If the chosen method is not configured to activate new users automatically, the account will be created but will need to be manually activated by an administrator in the Users module. With any of these external identity providers, the user name and password are not managed by SLIMS, but instead by the provider. Any password resets need to occur in the external provider. However, Local Authentication is always available as a backup method if there are any issues that cannot be worked around.

### 2.1.1.1. Local Authentication

Local Authentication is enabled and available by default. There are only three settings for it on the Internal tab.

- **Password Complexity:** Dropdown choice of values to force the password strength:
  - Simple (0 - 40 Characters)
  - Basic (1 - 40 Characters)
  - Standard (8 - 40 Characters)
  - CLIA Compliant (8 - 40 Characters. At least two each of digits, lowercase letters, uppercase letters, and special characters)
- **Password lifetime in days (Leave at 0 for infinite lifetime):** How long a password should be good for before it expires.
- **Failed login attempts before user is locked:** How many chances a user should have to input correct credentials before they are locked out.

If the user's account is set to "Local Authentication" in the Users module, SLIMS will validate their account using this method when they log in. The user will see the User Name and Password fields in SLIMS on the login page. The user inputs their private User Name and Password that were defined when their account was created with the SLIMS Administrator. For security reasons, the user's passwords are encrypted and can't be retrieved. If a user forgets their password a new one has to be created. A temporary password can be created for new users. The user will define a new password on their next login if "Require password change on next login" is used in the Users module.

### 2.1.1.2. LDAP System Authentication

SLIMS authentication can be done by an external LDAP system if it is already available. In this case, SLIMS authenticates using the user account protections that are setup in the external LDAP system. The user accesses SLIMS with their LDAP user account credentials. The settings are entered on the Authentication module LDAP tab.

- **Enable LDAP Authentication:** (Optional) Enabling this option reveals all the other settings to connect to the LDAP system.
- **Hostname:** (Required) Enter the hostname of the server running LDAP.
- **Port:** (Required) Enter the LDAP server port number.
- **Use SSL:** (Optional) Enable LDAP over SSL (LDAPS).
- **Connection Timeout:** (Optional) Enter the desired timeout of calls in milliseconds. Default value is 5000.
- **Enable two-step authentication:** (Optional) Enabling this reveals the rest of the options to use two-step authentication.
  - **Search Base:** (Required) Distinguished name of the node from which to search for users.
  - **Search Filter:** (Optional) The filter to use when searching users.
  - **Search Username:** (Optional) User to log in to LDAP server and execute queries.
  - **Search Password:** (Optional)
- **Default role for new users:** (Optional) Dropdown of available roles in the SLIMS instance. When selected, new users that are authenticated using LDAP will be created with this role by default.
- **Default license for new users:** (Optional) Dropdown of available license types in the SLIMS instance. New users that are authenticated using LDAP will count against this type of license for the instance by default.
- **Activate new users automatically:** (Optional) When disabled, new user accounts will be authenticated and created using LDAP, but will not be activated in SLIMS. The Admin user will need to activate the user manually as an extra verification step. When enabled, new users accounts will be automatically activated after creation.

If the user's account is set to "LDAP" in the Users module, SLIMS will validate their account using this method when they login. Like other external providers, the user's name information is brought from LDAP, but it uses the "cn" attribute (short for the common name attribute in LDAP directories) of the LDAP user account.

The LDAP connection can be tested using the "Test LDAP Configuration" button. The button opens a window that has a username and password field, and a test button. SLIMS will then test the connection with the LDAP server and provide feedback in the UI to alert you to any errors in the setup. These three stages are verified and return either a success or failure message:

- Basic connection to the LDAP server verifies whether SLIMS can reach the LDAP server.
- User retrieval tests whether SLIMS can find the user in LDAP.
- User authentication tests whether the user credentials match.

### 2.1.1.3. OpenID Connect Authentication

OpenID Connect is an identity layer on top of the OAuth 2.0 protocol. It enables client applications like SLIMS to rely on authentication that is performed by an OpenID Connect Provider to verify the identity of a user. OpenID Connect only handles authentication. Application access and any required authorizations, such as ensuring that users belong to the appropriate groups/roles and have the necessary permissions, should be configured in SLIMS directly. Single sign-on (SSO) is a time-saving and highly secure user authentication process. SSO lets users access multiple applications with a single account and sign out instantly with one click. SLIMS supports OpenID Connect which provides a modern and intuitive Single Sign-on (SSO) experience to SLIMS end users. SLIMS currently supports the following OpenID Connect identity providers:

- Okta
- Google
- Identity providers supporting OpenID Connect Discovery

This authentication method requires setup on the OpenID Connect tab so that SLIMS can redirect to Okta, Google or an OpenID Connect identity provider for authentication, and then redirect back to SLIMS after the user logs in.

When OpenID Connect Discovery is selected, the authentication provider needs to support OpenID Connect Discovery (.well-known/openid-configuration)

The first thing to do in this case is not in SLIMS, but is with the desired OpenID Connect identity provider. The SLIMS application has to be registered with Okta, Google or another OpenID Connect service which will provide a Client ID and Secret. For Okta, you will also get Okta's base URL (not needed for Google). When using OpenID Connect Discovery you will also need the base URL of the authentication provider. Those are values that will be needed for the OpenID Connect tab in SLIMS.

① When the SLIMS application is registered in the authentication provider, the URL SLIMS comes from should be entered in the service. This prevents a third party from pretending to be you.

- **Enable OpenID Connect authentication:** (Optional) Enabling this option reveals all the other settings to connect to an OpenID Connect service.
- **OpenID Connect provider:** (Optional) Selection of Okta, Google or OpenID Connect Discovery
- **Client ID:** (Required) When registering SLIMS as an application with any authentication provider this is the Client ID that is provided back to you.

- **Client Secret:** (Required) When registering SLIMS as an application with any authentication provider, this is the Client Secret that is provided back to you.
- **Service Base URL:** (Not for Google) When registering SLIMS as an application with Okta or OpenID Connect Discovery, this is the URL that is provided back to you. When OpenID Connect Discovery is selected, the following URL needs to work : <Service Base URL>/.well-known/openid-configuration
- **Default role for new users:** (Optional) Dropdown of available roles in the SLIMS instance. When selected, new users that are authenticated using OpenID Connect will be created with this role by default.
- **Default license for new users:** (Optional) Dropdown of available license types in the SLIMS instance. New users that are authenticated using OpenID Connect will count against this type of license for the instance by default.
- **Activate new users automatically:** (Optional) When disabled, new user accounts will be authenticated and created using OpenID Connect and created, but will not be activated in SLIMS. The Admin user will need to activate the user manually as an extra verification step. When enabled, new users accounts will be automatically activated after creation.

If the user's account is set to "OpenID Connect" in the Users module, the user will be able to login to SLIMS using the Okta, Google or OpenID Connect login button below the SLIMS Username and Password fields on the SLIMS login page. There are no extra account login buttons until the SLIMS instance is correctly configured and registered with the OpenID Connect identity provider in the Authentication module. When the user first logs in and authenticates with the OpenID Connect provider successfully, the user account is created in SLIMS.

#### 2.1.1.4. OpenLab Shared Services Authentication

Agilent OpenLab Shared Services is a software that provides the means to access, configure, and manage the Agilent OpenLab software services that SLIMS can integrate with. OpenLab Shared Services can be configured to act as the authentication provider (enable Agilent OpenLab Shared Services authentication) in the Authentication module in SLIMS. Configuring OpenLab Shared Services as an authentication provider for SLIMS is required to use SLIMS integration with other softwares of the Agilent OpenLab software suite, like OpenLab CDS, OpenLab ECM, and OpenLab Sample Scheduler. The versions of software that are compatible with SLIMS integration are listed in the technical requirements which can be provided by your SLIMS contact.

- **Enable Agilent OpenLab Shared Services authentication:** (Optional) Enabling this option reveals all the other settings to connect to OpenLab Shared Services.
- **Server URL:** (Required) Hostname of the server running OpenLab Shared Services. Must point to the OpenLab API base domain URL, including the port. The default port for OpenLab Shared Services is 6625.

Example: <https://server<yourcompany>.com:6625/>, where 6625 is the port used in the example.

- **Domain:** Domain name
- **Admin Name:** (Optional) Username that is used to connect to the OpenLab Shared Services server for specific queries, such as requesting all instruments.

ⓘ Required when enabling Sample Scheduler integration.

- **Admin Password:** (Optional) Password to go with the username used to connect to the OpenLab Shared Services server for specific queries.
- ⓘ Required when enabling Sample Scheduler integration.
- **Default role for new users:** (Optional) Dropdown of available roles in the SLIMS instance. When selected, new users that are authenticated using OpenLab Shared Services will be created with this role by default.
- **Default license for new users:** (Optional) Dropdown of available license types in the SLIMS instance. New users that are authenticated using OpenLab Shared Services will count against this type of license for the instance by default.
- **Activate new users automatically:** (Optional) When disabled, new user accounts will be authenticated and created using OpenLab Shared Services, but will not be activated in SLIMS. The Admin user will need to activate the user manually as an extra verification step. When enabled, new users accounts will be automatically activated after creation.
- **Enable OpenLab ECM Integration:** (Optional) Leave it disabled if OpenLab ECM will not be used. ECM XT is not supported by SLIMS. When enabled, the ECM Server URL field is provided. OpenLab ECM does not depend on enabling OpenLab Shared Services and can be used independently. Instead, any SLIMS user may be configured with credentials specific to ECM. Fields for the user's ECM credentials will be present in the Users module whenever there is an ECM Server URL present in the Authentication module.
  - *ECM Server URL:* The URL must point to the full ECM API path including the trailing sub-path `api/1.6/Data/`. Entering an ECM URL will make the option "Automatically upload an attachment of this attachment type to ECM" available for Attachment Types. This option allows a user to create an attachment type that automatically uploads attachments of that type to ECM. A groovy editor box is available to create a script that identifies the Agilent OpenLab ECM upload location.

Example: `https://ecm.<yourcompany>/ecmwebservices/api/1.6/Data`

- **Enable OpenLab Sample Scheduler integration:** (Optional) Leave disabled if the OpenLab Sample Scheduler will not be used. When enabled, the Sample Scheduler Server URL field is provided. SLIMS' integration with the OpenLab Sample Scheduler software requires use of the OpenLab Shared Services API, so full details must be configured for "Enable Agilent OpenLab Shared Services authentication," including the server and an authenticated user.
  - *Sample Scheduler Server URL:* Entering a Sample Scheduler Server URL will make the Sample Scheduler service available for SLIMS. The Sample Scheduler Server URL must point to the OpenLab Sample Scheduler API base domain URL, including the port.

› *Sample Scheduler version 2.5:* The default port is 7243.

Example: `https://server.<yourcompany>.com:7243`, where the port in the example is 7243.

› *Sample Scheduler version 2.6:* Must include `openlabserver`, and the default port is 50288.

Example: <https://openlabserver.<yourcompany>.com:50288/open-lab/samplescheduler/>, where the port in the example is 50288.

### 2.1.1.5. SAML Authentication

SAML 2.0 (or SAML2) is an open standard that allows identity providers (IdP) to pass user authentication and authorization information between the identity provider and the service provider (SLIMS). Like OpenID Connect, this authentication method allows identity providers that use SAML to connect to SLIMS. SAML is commonly used by enterprises to keep their Single Sign-On logins secure. The SLIMS application has to be registered with the SAML identity provider which will provide a SAML assertion.

⚠ When the SLIMS application is registered in the identity provider, the URL SLIMS comes from should be entered in the service to prevent third parties from pretending to be you.

A SAML assertion (an XML document) is sent to the service provider from the identity provider that contains the user authorization, and involves three pieces of information:

- *Authentication assertions* provide identification of the user, the time the user logged in, and what method of authentication they used (I.e., Kerberos, 2 factor, etc.).
- *Attribution assertions* pass the SAML attributes to the service provider with specific pieces of data that provide information about the user.
- *Authorization decision assertions* confirms that the user is authorized to use the service or if the identity provider denied their request (as in a password failure or lack of access rights).

If a user logs in for the first time through SAML to create their account, their default user name is the "name" information provided by the identity provider. Their email, first name, last name, full name, and locale are blank and need to be filled manually unless those values are provided as attributes in the SAML Attribution assertions. The attribute name needs to be the SLIMS database field (ex: user\_email='test@company.com'). Values can be filled by mapping SAML attributes from the IdP (such as Okta) to the service provider SLIMS. (For example using Okta: Name: userId, Name Format: Unspecified, Value: user.id)

This authentication method requires setup on the SAML tab so that SLIMS can redirect to the identity provider for authentication, and then redirect back to SLIMS after the user logs in. The options on the SAML tab allow the connection to be configured.

- **Enable SAML Authentication:** (Optional) Enabling this option reveals all the other settings to connect to a SAML identity provider.
- **Registration ID:** (Required) A value provided by the identity provider (IdP) as part of setting up SAML. This can be a simple value but needs to match between SLIMS and the IdP. This identifier may be used in URL paths, so take care that no URI encoding is required. It is sometimes used to configure the single sign on URL on the IdP's side, for example.

Example: Okta

- **Single Sign-On Issuer:** (Required) A value provided by the identity provider (IdP) as part of setting up SAML. It is sometimes referred to as the "Entity ID." The issuer is the IdP your application will be accepting authentication requests from.

Example: <http://www.okta.com/abcd1234>

- **Identity Provider Single Sign-on URL:** (Required) A value provided by the IdP as part of setting up SAML. It defines the URL the users will be redirected to when logging in to enter their credentials.

Example: <https://oktapreview.com/app/sso/saml>

- **X.509 Certificate:** (Required) A value provided by the IdP as part of setting up SAML. It is sometimes referred to as the "Signing Certificate." The key usually starts with "----BEGIN CERTIFICATE----". It contains the public key SLIMS uses to verify that all SAML authentication requests received were issued by the IdP.

Example:

```
-----BEGIN CERTIFICATE-----  
QtoQaDqxoRFD6YnHv2hVzM2e5SSxeNIPQnt8sMnwBxWuECnU4gRRTgbAiY  
Z9pzstCtMEXGReKOzjdb3KjMBPz1iu0TP8LfF11BUjebqOpyBoP2VusOJf  
Sgl19yBs0mhd6KP1VMkv7P1Vkj62jqh4pwZz625vKELPvdb81tTUfg1bDk  
9AVt7zX...  
-----END CERTIFICATE-----
```

- **SAML response attribute that identifies the user:** (Optional) This is the attribute name used in the SAML response to identify the user. Some IdP's only provide the username by default, but that alone isn't distinguishable enough to find the existing SLIMS user. SLIMS stores an external ID "user\_externalId" with the ID given by the IdP for a given user. (Using Okta as the IdP for an example, the external ID needs to be configured in Okta. See: *Okta Support Topic* [[https://support.okta.com/help/s/question/0D50Z00008G7V5w/how-do-i-get-the-users-id-via-saml-integration?language=en\\_US](https://support.okta.com/help/s/question/0D50Z00008G7V5w/how-do-i-get-the-users-id-via-saml-integration?language=en_US)].

① Leaving this field empty works but will lead to the username being created as user\_externalId. If the username changes on the IdP, SLIMS will consider it to be a new user.

Example: slims\_userId

- **Default role for new users:** (Optional) Dropdown of available roles in the SLIMS instance. When selected, new users that are authenticated using SAML will be created with this role by default.
- **Default license for new users:** (Optional) Dropdown of all available license types in the SLIMS instance. New users that are authenticated using SAML will count against this type of license for the instance by default.
- **Activate new users automatically:** (Optional) New user accounts will be authenticated and created using SAML, but when disabled, they will not be activated in SLIMS. The Administrator needs to activate the user manually as an extra verification step. New user accounts will be automatically activated after creation when this option is enabled.

These fields are all saved and displayed in the history except for the certificate history, which is hidden.

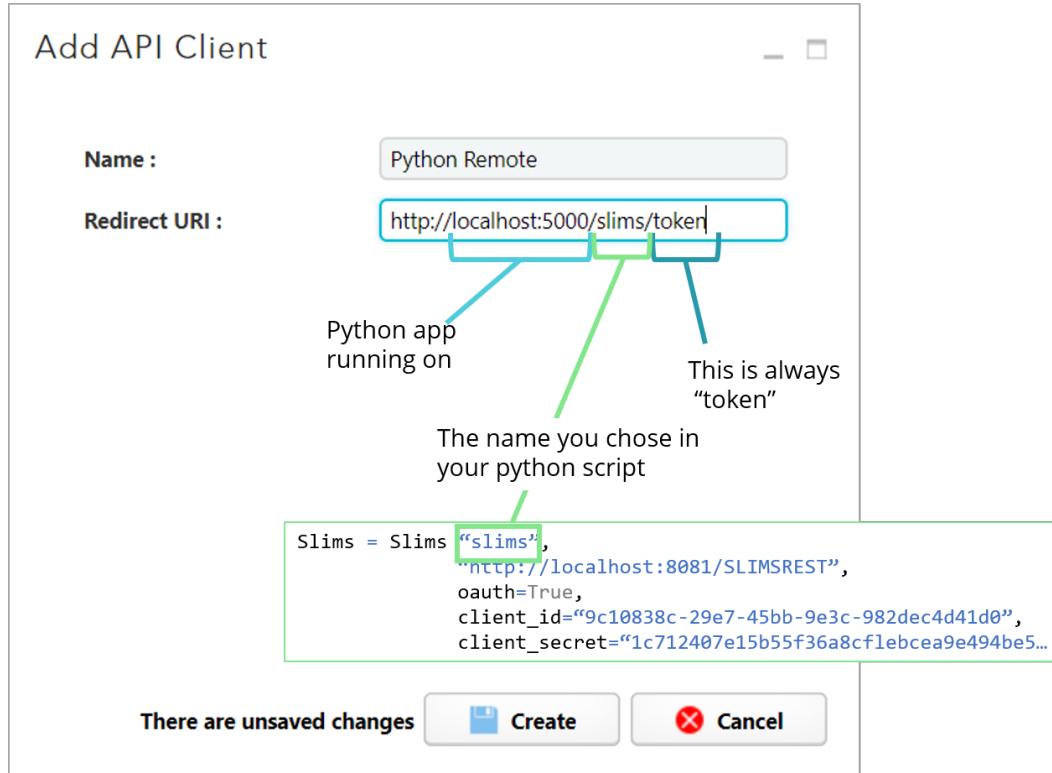
The SAML login button on the SLIMS login page will only show when the single sign-on URL is a valid URL and the certificate is a valid X.509 certificate.

### 2.1.1.6. API Clients Authentication

In this tab, the API Clients that authenticate with SLIMS through OAuth2 can be registered. To register a new API Client, click "New."

- **Name:** The name that represents the API Client so it is recognizable.
- **Redirect URI:** The URI that redirects to the API Client.

**Figure 2.1. Redirect URI meanings**



When the API Client entry is created, more details are supplied in the right panel. Pay special attention to the Client ID and Client Secret, as these fields are used by the remote applications to authenticate SLIMS users.

#### User Account Warnings:

- If the user enters incorrect credentials for any of these methods they receive an 'incorrect credentials' type warning.
- If the user leaves a password field empty, they receive an 'empty credentials' type warning.
- If the user account was initially created with one type of authentication, and the administrator changes the authentication method in the Users module to an incompatible authentication method, the user will receive a 'something went wrong' type warning. This is because SLIMS will try the defined authentication method which in this case does not match the account.

Local Authentication will always be available and reverting any account to local will always work, as it is the back-up method in case any external authentication provider has a failure or outage. However, other changes such as between local authentication, LDAP, and OpenLab Shared Services may not work as intended. Additionally, for OpenID Connect accounts to work, they must have been created with the OpenID Connect system and not with local authentication, LDAP, or OpenLab Shared Services.

- If an external system is used (LDAP, OpenID Connect, OpenLab Shared Services) and the account is created successfully but has not yet been activated in SLIMS by an admin, the user will receive an 'Authentication on external system successful, but the account is disabled in SLIMS' error message.

Once the user name and password (or other Google/Okta/OpenID Connect account actions) are entered and the login button is clicked, a loading progress screen indicates the various tasks and areas that SLIMS is loading.

## 2.1.2. Using Electronic Signatures with Authentication Providers

Some actions in SLIMS require reauthentication to ensure compliance with security standards. The behavior of electronic signatures depends on the lab's needs and the configuration of settings. They are constructed with status workflows, publication status workflows for versionable entities (such as SOPs, protocols, templates, etc.), and electronic signature lab settings. See the options for electronic signature behavior in *Section 10.8.12* and status workflows in *Section 10.20.4*.

The electronic signature action presents the user with a pop-up that asks for the user to reauthenticate. Depending on the lab settings and status workflows, there could be a comment, signature meaning, username, and password. However, when using OpenID Connect or SAML to authenticate, only the username, comment, and meaning may be asked, and the buttons "Sign in with the authentication provider" and "Cancel" are provided. A new tab opens with the fields to sign in with the authentication provider when the user clicks the button to sign. The password is required and the username may also be required depending on the external provider. The signature is applied in SLIMS and the authentication tab can be closed to return to SLIMS.

The reauthentication fails if the user attempts to enter credentials with a different account than the current one or if incorrect credentials are entered.

**⚠ CAUTION:** Not all OpenID Connect and SAML providers support reauthentication. There are varying levels of compliance depending on the capabilities of the providers. Keep this in mind when choosing an authentication provider.

- Okta provides reauthentication with SAML and OpenID Connect.
- Google asks the user to confirm reauthentication but does not ask the user to enter the password. This is not recommended for labs with higher compliance needs as it is not sufficient for electronic signatures.

## 2.1.3. Session Time-out

There is a time-out message defined in SLIMS that users may encounter:

- **Client Session Time-out:** X (in minutes): This occurs after X minutes without mouse-clicks or keyboard presses, when the client terminates the server session (logout) and the user is redirected to the login page where he has to provide his credentials in order to login again.

It is important to save often, as modifications that were not saved will be lost after a session time-out. The exception to this is changes that were made in the rich-text-editor, which is enhanced with an automatic save feature. SLIMS session time-outs can be changed in the Lab Settings module. The default value is 60 minutes.

## 2.1.4. Introduction to SLIMS Licenses

Each SLIMS instance needs a license file, as described in Section 10.9.2. Depending on the license parameters, warnings may be shown to the logged-in user:

- When the user exceeds the number of allowed concurrent users and the 'parameter strict' is set to false, the user will receive a pop-up message indicating he/she is exceeding the license boundary and needs to purchase more licenses. The user will still be allowed to use SLIMS.
- When the user exceeds the number of allowed concurrent users and the 'parameter strict' is set to true, the user will receive a pop-up message indicating he/she is exceeding the license boundary and needs to purchase more licenses. The user will be logged out and can no longer use SLIMS until another user logs out.
- When a user tries to login but there are problems with the license file (removed, changed, etc.), a RunTimeException will be displayed.

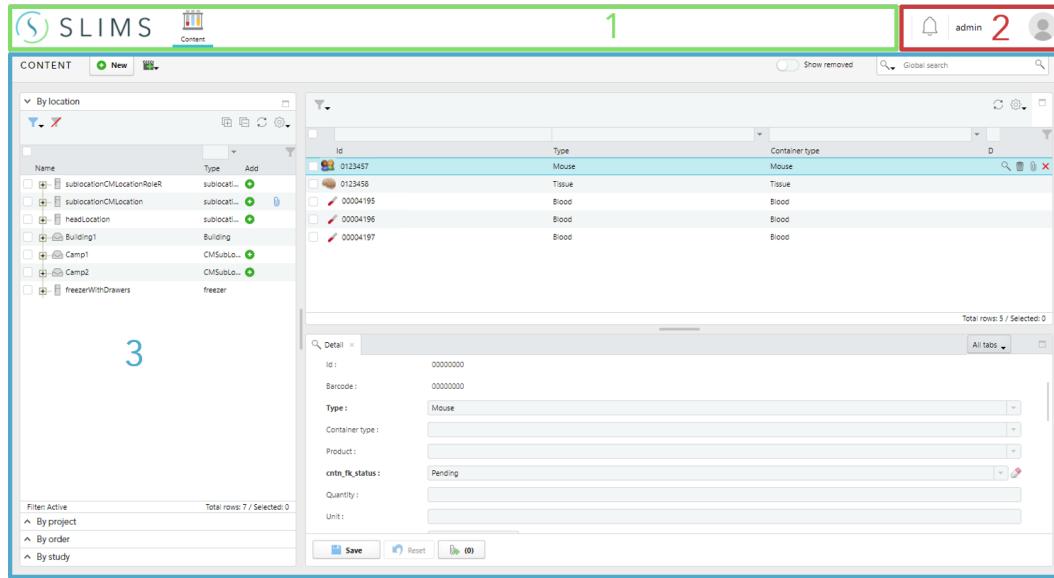
A user session may be terminated manually by the Administrator. In this case:

- When a user's session is terminated by the Administrator, the user will be redirected to the login page.
- When a user logs out in a tab of a browser, all other tabs in that browser containing SLIMS will also be redirected to the login page.

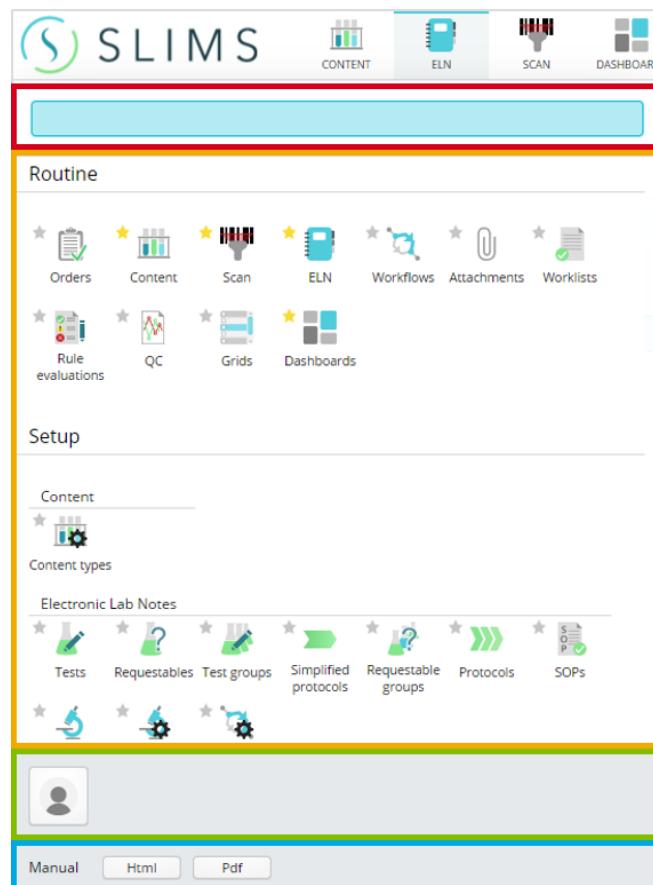
## 2.2. Interface Overview

The SLIMS interface is composed of three main parts as shown in the figure below.

Figure 2.2. Interface overview



1. The first part contains the **SLIMS button** and the **Favorites Bar**.
  - a. **SLIMS button:** Click the SLIMS logo in the top left to access the main menu. The menu is split into 4 sections.



1. Search Bar

2. Functionalities

3. Quick Settings

4. Reference Manual

1. Search Bar: The search bar is auto-focused when the panel opens so the user can immediately start typing to find the module they are looking for. Results are filtered separately in Routine and Setup. If only one result matches the user input then it is highlighted. Once a module is searched, the user can select a result with their mouse or the keyboard arrow keys. Clicking on the icon or pressing the Enter key will open the highlighted module.
2. Functionalities: All the modules available to the user are in the functionalities section grouped by Routine or Setup. The Routine section contains modules regularly used by the user. The Setup section contains modules used to configure SLIMS according to lab specifications.
3. Quick Settings: This section contains modules for user settings. This is where a user can change their password, name, email, and other personal settings.
4. SLIMS Info: This section contains the SLIMS version, instance name, and user manuals. A user can access either the HTML or PDF manual by clicking on the respective button.
  - b. **Favorites Bar:** User favorites are saved along the top bar to the right of the SLIMS logo. This bar provides quick access to the user's 'favorited' modules. Favorites can be added either by dragging the module's icon from the main menu to the Favorites bar or by clicking on the star on top of the module's icon. Modules can be removed by dragging the

icon away from the Favorites bar or by unstarring the module's icon in the main menu.



2. The second part contains a bell icon that contains the status of any notifications, the User Menu, and the status of executed SLIMSGATE flows. Please refer to the [Section 12.1](#) concerning details about flows status. The functions available in the User Menu depend on the user's role permissions, but the following functions are possible.

If available, Profile allows access to reset the user's password, add or choose an avatar to represent their account to other users, and adjust other general user information. Some information, like the user's role, is displayed read only.

The Layout indicates the interface as it appears now. The user can select a different Layout or create a new one by clicking on "Switch Layout." More details and explanations about Layouts are described in the [Layout](#) section.

Settings opens the SLIMS 'Lab Settings' if the user has permission to access.

In addition, a refresh button is available that restarts the user's session without logging them out.

Finally, the Logout button is under the User Menu as the last option, and will securely log the user out of SLIMS.

3. The third part is the actual working environment. The layout and the content are different depending on the chosen module (ELN, Content Management, etc.).

① When a layout is used for the first time, for example, the first time the user logs into SLIMS, or when starting to use a new layout, the third part showed in Figure 2.2 will be blank until a module is selected. After the first use of the layout, SLIMS will always show the latest opened module after logging in.

## 2.2.1. General Global Search Features

Global Search is located under the user menu and notifications icons. This allows you to search any number or text string with behavior that is dependent on your Global Search Lab Settings to easily access content or data in SLIMS (depending on what your role can access). It is present primarily in the Content and ELN modules, and the context of the search updates depending on the module.

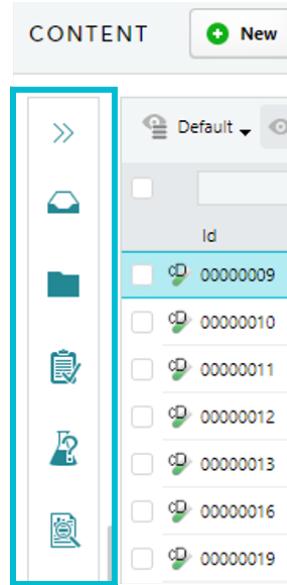
There are some restrictions on Global Search, however. Default and custom fields can be searched, but Flat fields cannot. (To learn about Flat fields, check the [Fields Section](#).)

## 2.2.2. Maximize / Minimize Sub-windows in Content Management

A maximize / minimize button is present in each of the 3 sub-windows present in Content Management. The action can be triggered using the button on the top right corner of each sub-window. When a sub-window is maximized, it takes up more space and the other sub-windows are not visible. Clicking on the minimize button will set the sub-window back to its original size.

The left filter sub-window in the Content module has a different collapse icon that can be used to minimize the filter sub-window, but still leave each filter available with one click.

**Figure 2.3. Collapsed Filters**



### 2.2.3. GUI Navigation and Help

- **URL Navigation:** The SLIMS URL contains the module name so that users can navigate using the browser's next and back buttons. This has the added benefit that users can copy and paste the URL to a colleague to share their view. Users can change the module name after the '#' symbol as an additional way to switch quickly between modules.
- **Hover Help:** Hovering over elements in the system provides helpful information in context. For example, users can hover over a location, such as a laboratory name, and see a help text pop-up listing the sublocations and open positions.
- **Print Grids:** Print options on areas such as grids and derivations have been improved so that separation between rows is more apparent. The print actions are always located at the top right so it is easy to find, and contains the SLIMS logo in the header. The browser will open the file directly in its native print pop-up when the user clicks the print action.
- **System Highlighting:** Cascading dropdown menus, such as the dropdown selection of a new location when moving content, have been updated to improve contrast visually.

## 2.3. Grids

As a user-friendly application, SLIMS provides Spreadsheet style data input throughout its interface that allows efficient data submission as the user would do in a data submission application like Excel.

### 2.3.1. Composition

Grids are composed of three main components as shown in Figure 2.4.

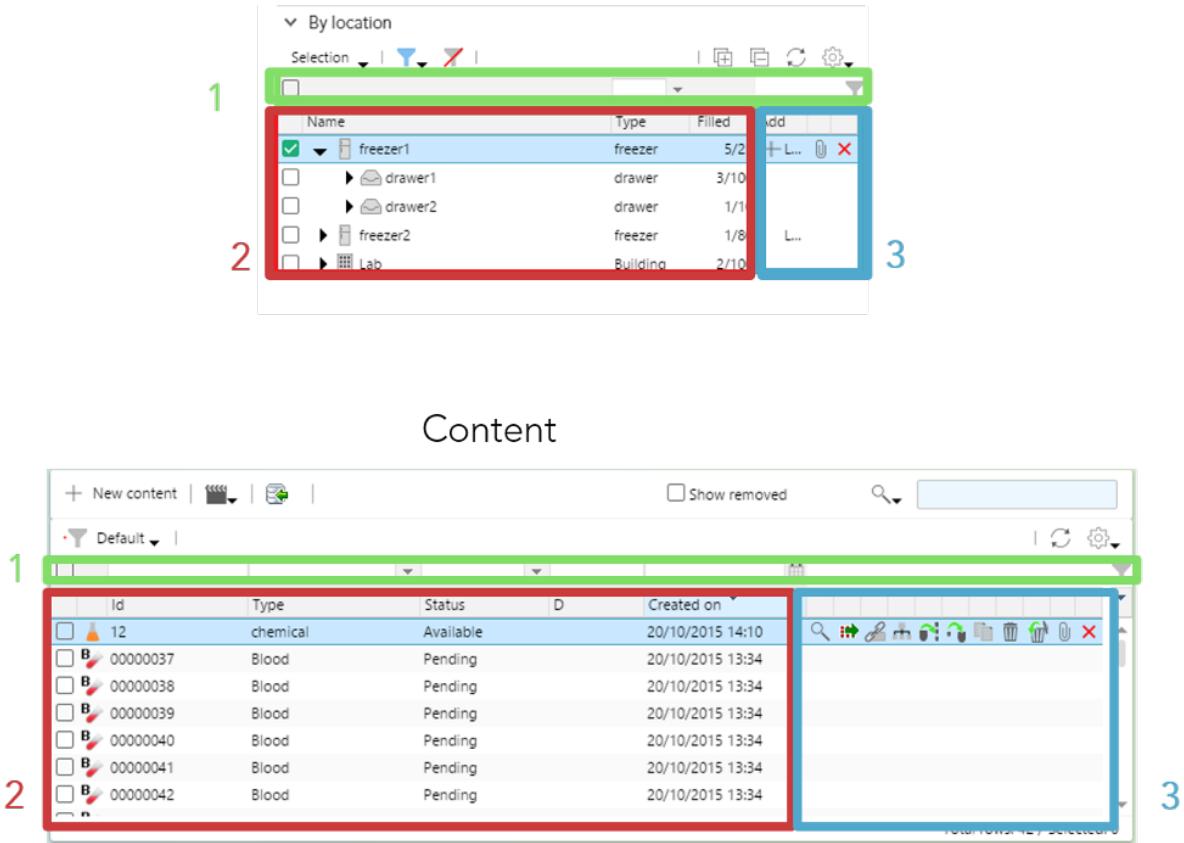
**Figure 2.4. Grid Components**

Grids contain a Tool header ( **a** ) and a Footer ( **c** ) in addition to a central component ( **b** ) representing the grid itself. The header and footer are similar and have the same structure in all grids.

The **Tool header** contains access to actions that can be applied on data records, and Tools and Views that can be used on the grid. More detail is provided on View management and available Actions and Tools in later sections.

The **Footer** contains information about the number of rows available in the grid and the number of selected items.

SLIMS provides different Grid types according to the **central component**. Two types of Grid can be distinguished: **List Grids** and **Tree Grids**. However, they all share the same basic structure and composition. The two types of Grid are shown in Figure 2.5 where the Location Grid is displayed as a '**Tree Grid**' and the Content Grid is displayed as a '**List Grid**'. Both types share the same filter, column, action, and selection structure, highlighted by the green, red and blue colors.

**Figure 2.5. Grid Structure**

Three parts are observed in the Grid's central component as shown in Figure 2.5:

1. The first row is used as a filter as described in the *Filter grid action*.
2. The second part shows the List of records (Contents, Locations, etc.) with their completed field data. By using the view tools, specific fields (the table columns) can be added or removed from the List View. The difference between the types of grid is that **Tree Grids** like the Location Grid are grouped into lists that are expanded into smaller and smaller elements, and **List Grids** like the Content Grid are tables with rows and columns which already show one record per row (Figure 2.5).
3. The third part shows icons for actions that can be executed on the List elements, like "move," "remove," and "add attachment." Actions used in the highlighted row only impact that individual record.

## 2.3.2. Grid Actions

### 2.3.2.1. Filter

The first row of each View is dedicated to filters on each column which provide simple and rapid access to the desired data set. Filtering is possible by using one or more combined criteria in the View columns. The criterion are entered in the field above the columns containing the

content to be filtered. For example, entering "Building" in the field above the "Type" column in the Location View will temporarily update the filtering. Only the locations that are of Building location type will be displayed.

A generic filter option is also available on the top right of the GUI.

Multiple terms can be searched by separating them by space characters.

A tree grid with multiple levels is only filtered on the root nodes.

### 2.3.2.2. Sort

The list of records in a grid can be temporarily sorted according to a column's content by clicking (mouse left click) on the column header. Clicking a second time on a column header changes the sorting order between ascending or descending.

An additional sophisticated sorting option is accessible by right-clicking on a column header:

- **Configure Sort:** Directly opens the View configuration pop-up to the Sort tab. Sort can be configured over multiple columns, and the View can be updated temporarily or permanently.

### 2.3.2.3. Edit Columns

The following options are accessible by right-clicking on a column header:

- **Auto Fit All Columns:** Resizes column width in order to see all the contents' values.
- **Auto fit:** Resizes the selected column in order to see its value.
- **Columns:** Directly opens the View configuration pop-up to the Columns tab. Columns can be added, removed, and reordered. The View can be updated temporarily or permanently.
- **Configure Groupings:** Directly opens the View configuration pop-up to the Groupings tab. Groups can be configured into a tree structure and saved temporarily or permanently.
  - ① Grouping by columns is not possible if the grid contains more than 980 records.
- **Freeze/Unfreeze (column's name):** Freeze keeps the column visible on the left side when there is a horizontal scroll bar in the grid, and Unfreeze disables the freeze.
- **Configure Highlights:** Directly opens the View configuration pop-up to the Highlights tab. One or more highlight rules can be added and saved temporarily or permanently.
- ① The next two options are only available if the Administrator enables the setting in Lab Settings. It is recommended that a similar formula field with groovy be used instead of the column, because these columns do not handle custom data types in an optimal way.
- **Add formula column:** Using the available columns, another one can be created with its value defined by a formula.
- **Add summary column:** Adds a column that can contain a concatenation of multiple fields. For example, to create a column containing "#ID's parents are #mother and #father", where the columns ID, mom, and dad can be selected and completed with the corresponding text.

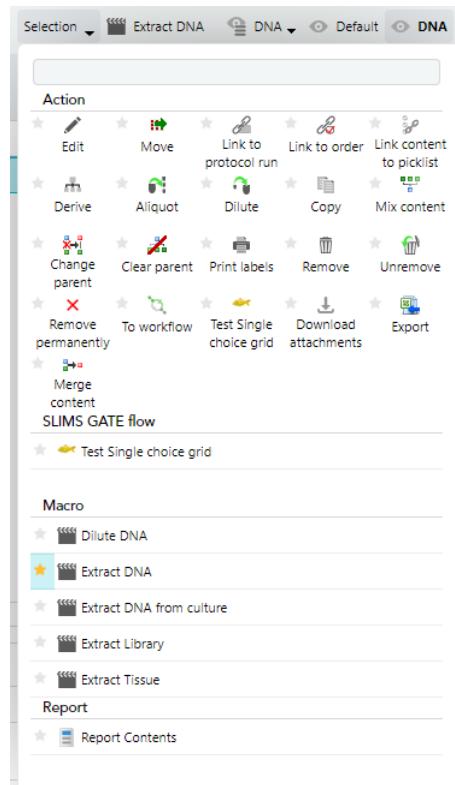
### 2.3.2.4. Multi-Selection

There are two types of record selection in various areas of SLIMS. For single selection, click on a record and it becomes highlighted. Any actions taken will pertain only to that record.

Multiple records in the Grid can be selected in order to apply an action on all selected records at the same time. To do so, check the selection boxes to the left of the records, or use the top-most checkbox to select all.

When multiple records are selected, the 'Selection' menu is available that contains actions that can be taken on the group of selected records, similar to the one represented in Figure 2.6.

**Figure 2.6. Actions Menu**



The Selection menu in most modules is organized in four sub-menus:

- **Action:** Reports all standard actions that can be executed on multiple records simultaneously, e.g. Edit, Move, etc.
- **Macro:** Contains the list of available Macros (if any).
- **SLIMS GATE:** Contains the list of available SLIMSGATE Flows (if any) that can be executed on selected records.
- **Report:** Contains the list of available Reports with selection type Multiselect (if any).

A search field is present on top of the menu which allows the user to find an action, macro, SLIMSGATE Flow, or report quickly by typing its name. The list updates as the user types in-

to the search, so this is useful especially when there are many actions, macros, SLIMS GATE Flows, or reports to sort through.

When records are selected or multi-selected to execute an action, the grid is refreshed after the action is completed. The user will have no records selected directly after taking an action to prevent confusion or needing to clear view filters and sorts to find the original or output records.

Selection actions can only be executed on less than 1000 manually selected records.

When clicking on the select all at the top of the Grid, all the actions will be available. However, the list will be shortened to only the possible actions if the number of records exceeds 10,000.

① A range of records in a grid can be selected by clicking the first item, holding down the shift key, and clicking the last item in the list.

### 2.3.2.5. Copy From Grid

It is possible to select and copy values from a row for all default lists, like the content list. However, lists that allow interaction between rows by drag and drop cannot be copied, like the location tree and ELN tree.

## 2.4. Views

Every SLIMS instance comes with a "Default" view that shows all content that the user has access to. This view is a catch-all, as other views can be added to show a subset of content. Customized Views are important in SLIMS because of their ability to improve action economy and work efficiency. The View feature in SLIMS allows users to create favorite views to switch between with specific filters, columns, sorting, grouping, and highlighting. Users can perform minor adjustments without creating Views, such as the rapid filter method explained in the Grid actions section, and configuring columns, but the changes are temporary unless they are added in a View.

The Views panel  Default ▾ contains existing Views and tools for editing and creating Views, Filters, and Picklists. When the user accesses the View panel, a list of icons are available in the menu. These represent various views and actions that are available to effect the display of the grid content. Any available icons in the menu can be favorited by clicking the star on the icon to add them to the top of the user's grid for quick access. The user can then access any of the favorited icons with a single click and quickly switch between them and perform actions.

### 2.4.1. Create Customized View

By clicking the "Add View" action, the configuration pop-up window is displayed. The window is categorized into tabs where the user can set the criteria for what the View should show. A customized view can be created that has particular columns displayed, has a saved ascending or descending sort, has highlights on column fields, and has filters already assigned, like a DNA content filter that sorts volume by descending values with highlights on a range of low

volumes. To create a new view, the user clicks on the Add view icon  that is available under the Views panel. The configuration pop-up window contains:

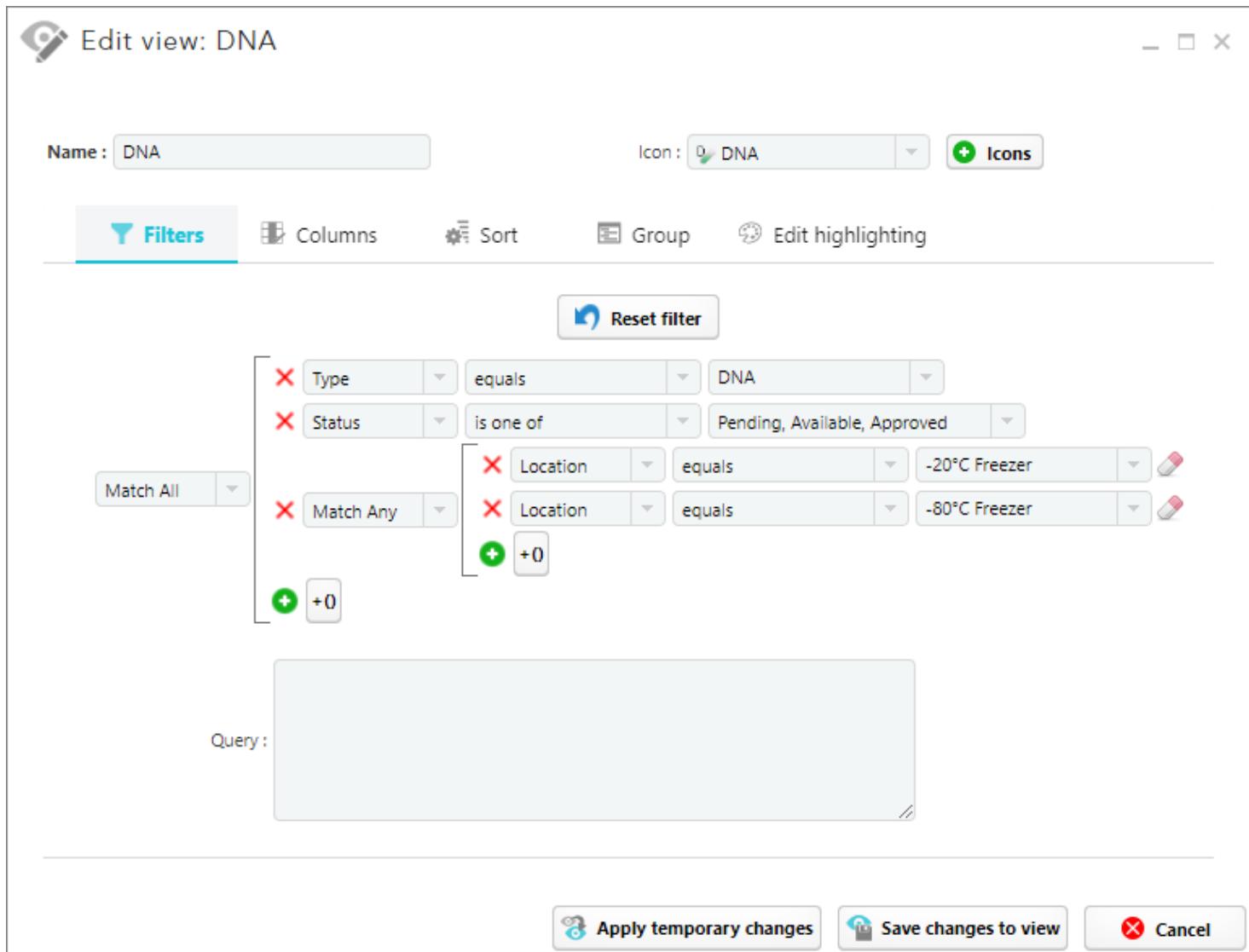
1. The first field is for naming the view. The view can be named one word that indicates the usage, like "DNA" or "VolumeAscending."
2. The second field is for setting a secondary icon for the view (the primary icon is always the Eye icon, to represent that it is a View). Secondary icons are good for quickly recognizing the different views when many are favorited above the grid, and even for saving space, allowing for shorter View names if the icon represents the meaning well enough. The Icon is not required and may be left blank. The icon can be added, updated, and removed after view creation by editing the view.

Custom icons can also be added and used for views. To do so, click on the Icons button beside the dropdown and follow the instructions in the *Custom Icons* section.

① To remove a view icon, change the icon to the first blank row in the dropdown. The transparent icon shouldn't be used for this as it takes up more space, and the meaning of the first blank row is 'icon = unset,' while transparent just sets a transparent icon.

3. The Filters Tab contains advanced options for a series of filters. This allows users to layer more than one filter for more granular results. The advanced filter has two tools. The first tool defines the clause, or the type of relationship of a set of filters to the content data (Match All, Match Any, or Match None). The second tool defines the condition or the set of conditions that filters the data field (Equals, Not Equal, Is One Of, Is Not One Of, Is Null, Is Not Null). Each clause could contain either conditions or other clauses with their own conditions. The icon  is used to add a condition and the icon  to add a Sub-clause.

For example, the following filters in a view will filter data in the content table to DNA contents with the status 'Available' that is located in -20°C Freezer or -80°C Freezer.

**Figure 2.7. Editing a filter with different clauses**

Filters using "Is One Of" and "Is Not One Of" allow you to select all of the desired field entities at once using checkboxes. That way, you do not have to build a filter for each individual entity using "Equals" or "Not Equal."

⚠ Conditions with the null criterion, 'is null' or 'is not null', may not always return the expected results. Therefore, the usage of these two conditions is discouraged.

There are two particularly useful filters:

- The *Context: Logged in User* option is used to filter items for the "User" that created them. When used, it shows only the items created by the currently logged in user.

- The *Context: Logged in User's Groups* option is used to filter items for the ones that are in the user's groups. When used with "Group" it shows only the items created in the currently logged in user's groups.

ⓘ The group function is for data in the user's groups and doesn't extend to the additional group rights from other groups that a user could have.

Two filters are described in the *History* section that can be used for views in history grids.

4. The Columns Tab contains two lists. The left list shows the available fields that can be used in the view (Contents, Orders, Derivations, etc.) and the right list shows fields that are already in the View. Use the arrows (→ or ←) between the two lists in order to customize the columns that are displayed in the View. The order of the fields is denoted by the "Position in the Grid" column, the order can be changed by dragging and dropping the fields in the list into the desired order. The "Frozen" checkbox can be used to freeze fields so they remain static while the rest of the grid scrolls.
5. The Sort Tab contains buttons allowing different sort levels to be created. The levels consist of a Column (the default or custom fields to be sorted) and the Direction (ascending or descending). The levels create a hierarchy in which the top-most level is sorted first, and then by each following criteria. The levels can be adjusted with the Move Up and Move Down buttons, and all unsaved changes can be Reset. This allows the user to set simple to complex sorting as the default for the view.
6. The Group Tab contains similar buttons to the sort tab, but allows content to be grouped within a similar hierarchical structure. Grouping uses the Default Date Grouping Mode.  
 ⓘ Grouping by columns is not possible if the grid contains more than 980 records.
7. The Edit Highlighting Tab contains options to create a simple highlight or an advanced highlight. Highlights allow SLIMS to visually call attention to important data in a grid and can be used as part of designing a distinct and useful View.

When using highlights, the user defines which set of data SLIMS should highlight according to the simple or complex conditions that are configured for fields. Foreground or Background color and specific icons can be chosen for each field row that meets the criteria. Multiple simple and complex highlights can be created and used at the same time. Simple criteria applies a highlight to the fields matching the condition, and advanced criteria applies the highlight to fields other than the one matching the condition. Individual Highlights can be deleted as well with the red X on the right. The Reset button reverts any changes since the View was last saved.

## 2.4.2. Edit Customized View

An existing view can be edited by using the "Edit View" action in the Views menu. The tabs and options are all the same as in "Create Customized View," and the name of the view can be edited with some restrictions. The new name of the view must be unique and therefore cannot match any existing view name for the SLIMS instance.

The Default View is the only one that cannot be edited so that it will always be the catch-all view that shows all content the user can access. Only the "Apply Temporary Changes" button is

available if changes are made. However, the Default View can be copied, renamed, and edited into a new view.

The icon of the selected view turns blue  if any changes are made to the view that could be saved. The color turns back to gray when modifications are saved to the view or canceled. To save any such changes, they must be updated in one of the configuration tabs using "Edit View." The modifications can be cleared by using the cancel icon at the end of the row or using "Reset View."

The possible changes that can be made and saved on a view that cause the icon to turn blue are:

- A column is added or removed
- The column order is changed
- The column sizes are increased or decreased
- The filter is changed
- The grouping is changed
- The highlights are changed
- The sorting is changed
- A new view icon is selected
- The view is renamed

### 2.4.3. Copy Customized View

A customized view can be copied to edit and save into a new view with slightly different properties.

To copy the current view a user should follow these steps:

1. Click the View icon to open the View panel.
2. Click on the 'Copy View' action.
3. The Copy View pop-up appears with the columns and filtering of the current view already completed.
4. The user then alters any of the configuration as desired to create their own view from the current view's properties.
5. The user gives the view a different name and saves it by clicking the 'Create' button, or cancels by clicking the 'Cancel' button.

A new view with the chosen columns and filters will be created, and the view that it was copied from will be unchanged. The configuration of the copied view will be present in the new view.

## 2.4.4. Edit Grid Template

The action Edit Grid Template available in SLIMS allows the user to add columns from other tables, such as result, location, user, and so on to the content grid. This tool can be accessed under the Views menu.

This tool requires the following Role Access rights to work:

**Table 2.2. Edit Grid Template Permissions**

Access rights Category	Access rights Name	Purpose
Routine: Content	Allow Editing of the Grid Template for Content Grids	This permission allows the role user to access the "Edit Grid Template" action available in the views menu in the Content Module. This action allows columns from other tables to be added to the content grid.

This tool allows users to define the meta tree of the content grid in order to add columns related to the contents directly in the content grid. Currently, it is available in the content module and the order module. This allows fields from other tables such as results to be added to the content table in the content grid. In this case, the root of the meta-tree is already pre-configured as Content. For any further details about the meta-tree configuration, see the Meta tree section.

## 2.4.5. Queries

Query scripts in SLIMS are a way for users to filter on existing relationships between different SLIMS records. Relationships are specific links between different SLIMS records, such as 'location of,' which is a relationship between a record and its location. Family relationships such as a content and its mother, father, or son are only available when pedigrees are being used in SLIMS (see *Pedigrees* section for further details). The query tool is available in the query script text box within the *Advanced Filter* panel when editing or creating a view.

The query tool uses a customized language created by Genohm that allows users to define and edit their own advanced queries. The specific language structure (Figure 2.8) has to be respected in order to create queries.

Using the query language, the user defines key names for records with a certain relationship. The key names are 'Nodes,' and the 'End Node' is the key name of what the user is looking for. Each element of the query structure is explained in details below.

- **Nodes** represent labels given by the user to define a key name for records that have a certain relationship. Node labels are chosen by the user. There are no restrictions on node labels besides the 'End Node' (explained below). However, it is recommended to use node names with an easily understood meaning, like 'start,' 'parent,' or 'father' to simplify the script for other users who need to understand it when editing later on.
- The '**End Node**' is the only node that has a fixed name. This name is translated by SLIMS and represents the end records the user is searching for. SLIMS will return a query result containing all records that match conditions specified on the End Node.

- **Point '.'** is used to define a condition on a specific field value of a record. For example, the expression 'end.cntn\_id = "admin"' returns a queried record 'end' that has ID (cntn\_id) equal to "admin". These conditions can also be used for the query: "=", "<", ">", ">=", "<=", "!=" , "like".
- **Square brackets '[- [ ] -'** contain the name of the relationship.
- **Relationship** is the name of an existing relationship between two nodes. Each relationship can be used in the query with two different approaches: either using the notation **is...Of** or **hasAs....** Below are two examples on how the two notations are used:

```
end -[isFatherOf]-> child,  
child -[hasAsFather]-> end,
```

The following supported relationship names are supported by SLIMS:

Location, ContentType, OriginalContent, Result, Source, Disease, Provider, User, Group, WorkList, Test

In addition, family relationships (Father, Mother, etc.) can be used. All family relationship names understood by SLIMS can be found in the *Pedigrees* section. Spaces should not be used in the query relationships, because only capital letters will be used as word separators. For example, the family relationship 'Great GrandChild' would be named 'GreatGrandChild' in the query.

In addition, relations between mixes and ingredients can be used in the query. This type of relationship is not used with the notations 'is..Of' or 'hasAs...', but instead with the relationship name `mix`.

- **Arrows ' > , < :** Represent the edges between two nodes and define in which direction the relationship goes. For example, the next two queries give the same result: `end -[isFatherOf]-> child`, and `end <- [hasAsFather]- child`.
- **Comma ',':** The conjunction between different query expressions. This is used when multiple query conditions need to be assembled together with the 'and' logical connectivity. This is the same as the Boolean AND value.
- **OR :** The disjunction between different query expressions. Used when multiple query conditions need to be assembled with the 'or' logical connectivity. For instance, `cntn.field1="value" or cntn.field2="value2"`, which would allow records in the result that match 'value' for field1, or match 'value2' for field2, or if both match. This is the same as an inclusive OR value.

Used with relations, the following syntax has to be applied: `x -[relation1 or relation2]-> y`. For example: `end -[hasAsMother or hasAsFather]-> parent`.

⚠ Expressions related with a disjunction have to be put in the same line.

- **Notation ':x..y':** This notation is used after the relationship name and it defines the relationship levels where x and y represent levels boundary. For instance if the search is for all records having either a first, second or third original content (parent), the expression to use is: `end -[hasAsOriginalContent:1..3]-> parent`,

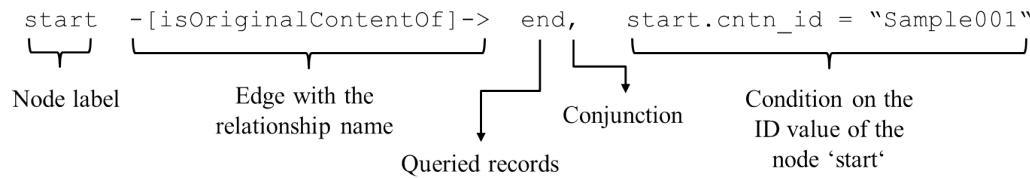
If the search is for a specific relation level the expression ':x..x' is used. For example, if the query is for all records having second original content (parent), the following expression is used:

```
end -[hasAsOriginalContent:2..2]-> parent
```

① Queries with the notation x..y can be very slow if using a big gap between levels boundaries (i.e. y-x shouldn't be too big).

- **IN :** is used for searching records that have a specific field value within a list of values, with the following structure `fieldName IN "value1 || value2 || value3 || value4"`.

### Figure 2.8. Query Language Structure



The shown query will search for all records that have a parent (relation name: `isOriginalContentOf`) and the parent has the id "Sample001".

① No warning or error message is returned when the specified field value is not available in SLIMS or when no query result is found.

△ When there is an error in the query script, SLIMS returns a warning with the message "Server returned FAILURE with no error message performing operation".

## 2.4.6. Query Examples

In order to cover most common query cases with examples, an architecture is used (shown in Figure 2.9) where some family pedigree records from which some samples have been derived (DNA and Blood) are shown. A location and four results obtained from different records are available as well (Figure 2.9). Family relations are created using the "Set parent" and "Set child" actions that are described in the *Pedigrees* section.

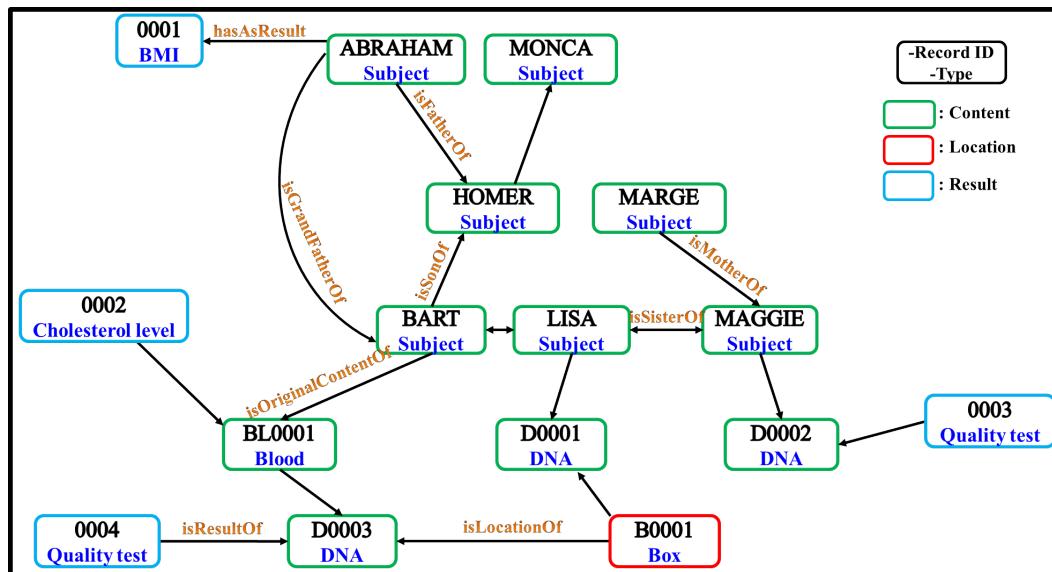
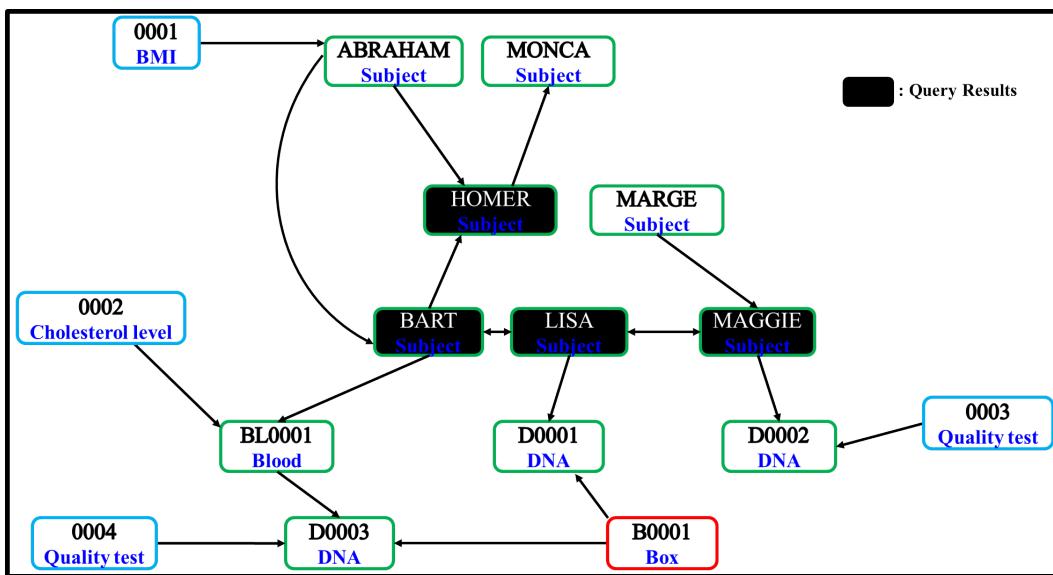
**Figure 2.9. Example of relations**

Figure 2.9 shows an example of relations defined in SLIMS. Note that arrows illustrate relations between records but not all relations are explicitly shown with arrows.

- **Example 1:** All subjects having both parents: (Figure 2.10)

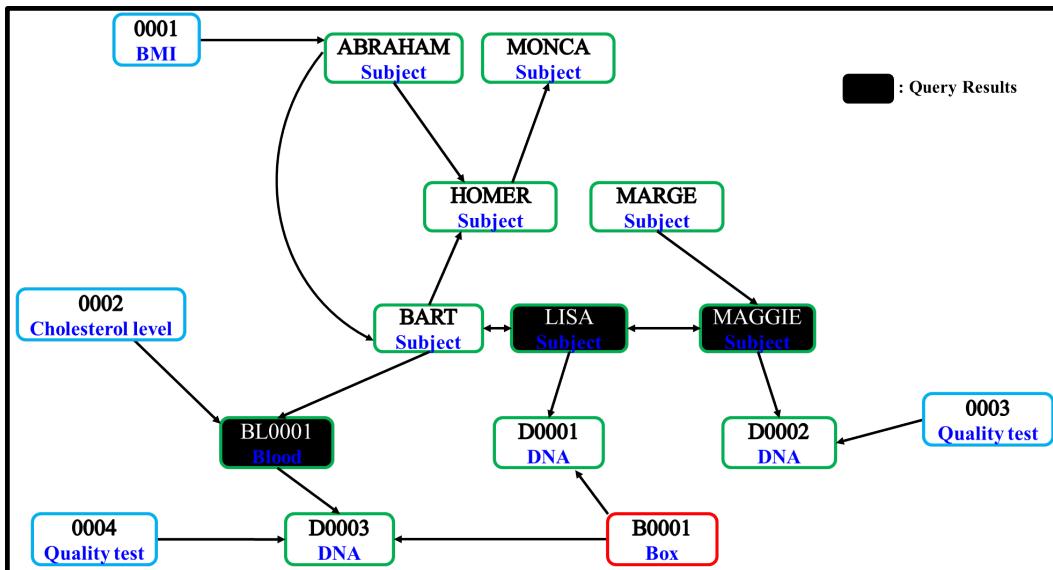
```
end -[hasAsContentType]-> contentType,
contentType.cntp_name="Subject",
end -[hasAsFather]-> father,
end -[hasAsMother]-> mother
```

**Figure 2.10. Querying all subjects having both parents**

Query results show all records having a father and mother. Not all relations are explicitly shown in Figure 2.10 to avoid crossing arrows which leads to confusion.

- **Example 2:** All records having a DNA sample: (Figure 2.11)

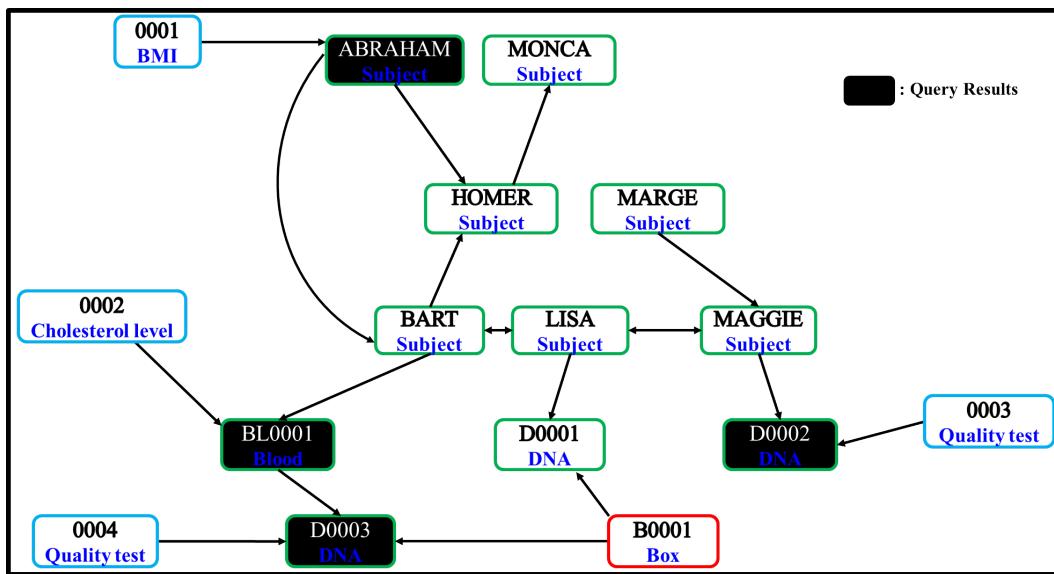
```
end -[isOriginalContentOf]-> start,
start -[hasAsContentType]-> contentType,
contentType.cntp_name="DNA"
```

**Figure 2.11. Querying all records having a DNA sample**

- **Example 3:** All records having a result record: (Figure 2.12)

```
end -[hasAsResult]-> result
```

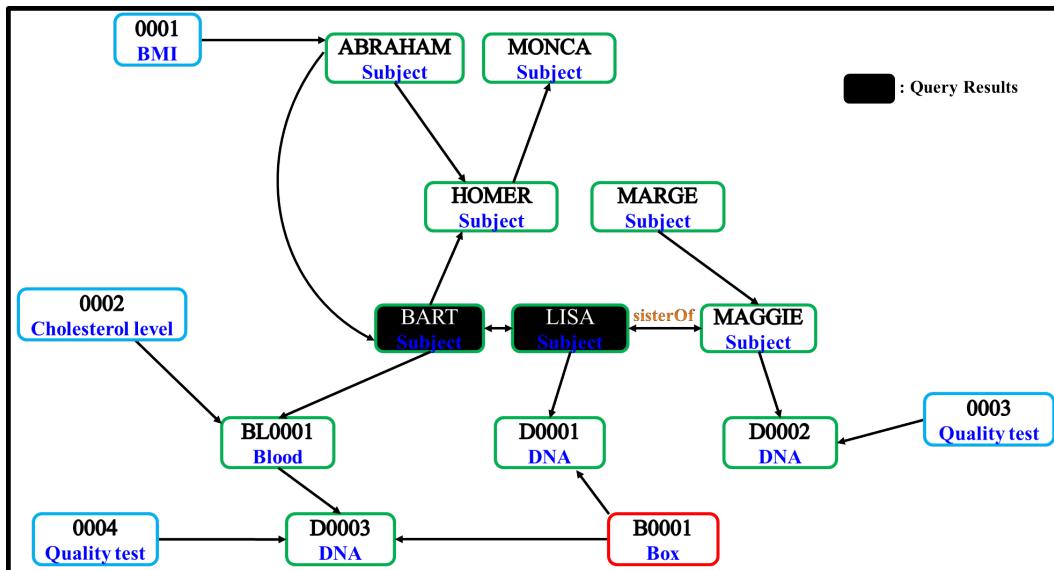
**Figure 2.12. Querying all records having a result**



- **Example 4:** All subjects having a DNA (isOriginalContentOf), could be first or second level, that is contained in a specific location: (Figure 2.13)

```
end -[hasAsContentType]-> contentType,
contentType.cntp_name="Subject",
end -[isOriginalContentOf:1..2]-> dna,
dna -[hasAsLocation]-> location,
location.lctn_name="B0001"
```

**Figure 2.13. Querying all subjects having DNA contained in the location 'B0001'**



- **Example 5:** Querying all ingredients from a mix:

```
end -[mix]-> mix,
mix.cntn_id="MixId",
```

Quering all mixes done with one ingredient:

```
end <-[mix]- ingredient,
ingredient.cntn_id="IngredientId",
```

## 2.5. Picklists

A Picklist is a saved list of contents that can also be added to the favorites above the user's list view.

The following Role Access rights can be configured for this feature:

**Table 2.3. Picklist Permissions**

Access rights Category	Access rights Name	Purpose
Routine: Content	Can Link Content to Picklist	When enabled, the role user has the ability to link (save) content in a grid to a Picklist.
Routine: Content	Can Unlink Content from Picklist	When enabled, the user has the ability to unlink (remove from list) content in a grid from a Picklist.

## 2.5.1. Create a New Picklist

A Picklist can be added using the "Add Picklist" action in the View panel. Adding a new Picklist creates an empty list by default, and then content can be linked to it. It is not possible to create a picklist with the same name as an existing one.

The Picklist configuration pop-up contains the same tabs and options as the Add View configuration pop-up, minus the Filters tab. This is because a Picklist already functions as a filter of particularly selected Contents. Using the other four tabs, the Picklist can be set up with the desired column appearance in the grid.

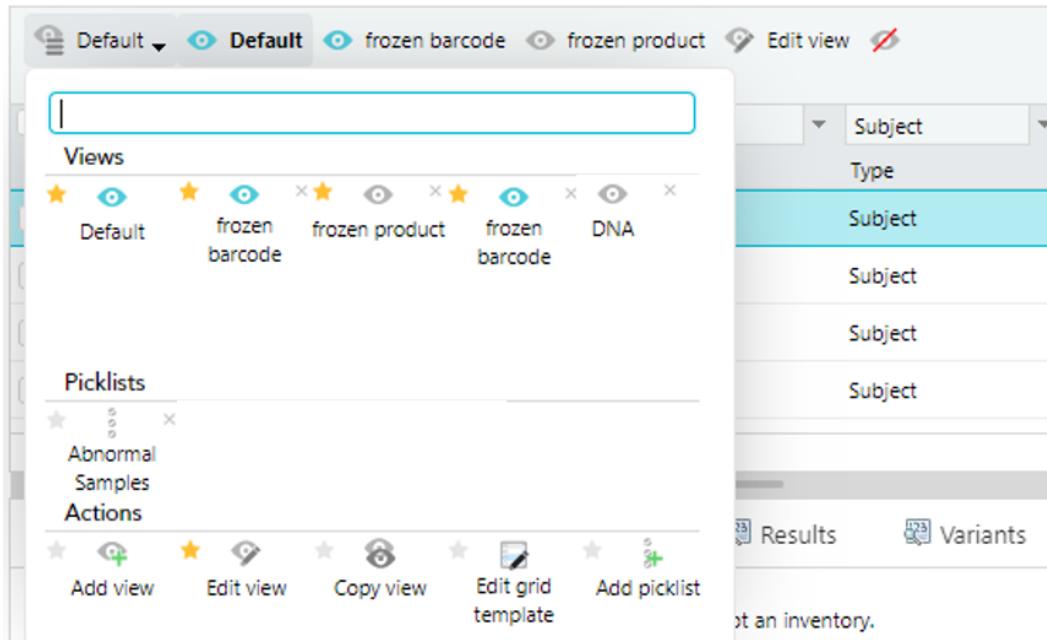
## 2.5.2. View Picklists

The previously created Picklists are available in the View Panel Picklist section. Picklists can be favorited by clicking on the star so they can be quickly accessed in the top row of the grid beside the user's favorited Views and Actions.

While a new or existing Picklist is selected, three new options are present: Link Content to Picklist, Edit Picklist, and Copy Picklist.

Edit Picklist provides the same configuration screen as before, with the option to Apply temporary changes, Save the View, or Cancel. The Picklist name can be edited with some restrictions. The Picklist name must be unique (cannot match any other Picklist names in the SLIMS instance).

**Figure 2.14. View Picklists**



## 2.5.3. Copy Picklist

The "Copy Picklist" action opens the configuration pop-up for a new Picklist, but with all the setup from the one being copied. The user enters a new name for the Picklist and can then ad-

just any of the tabs for the desired visualization. Clicking "Create" will cause the copied Picklist to be created without changing the existing Picklist it was copied from. Copying the Picklist retains the same content that was linked to the Picklist that was copied, so the content is effectively linked to both Picklists.

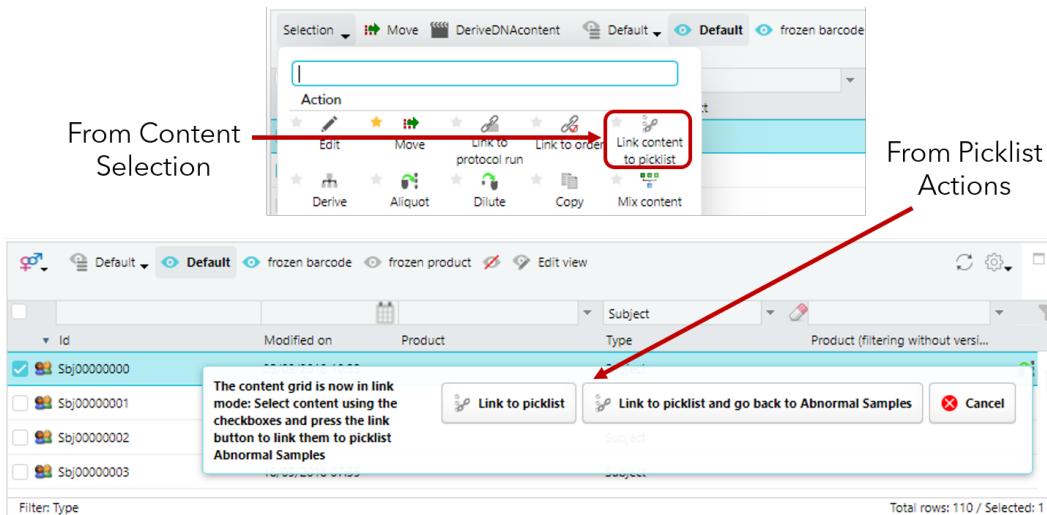
## 2.5.4. Add Contents to Picklist

Content can be added to a Picklist in two ways. Either the user can select content first and link to the Picklist of their choice, or the user can access the desired Picklist and link Content to it via Link Mode.

To use the first method, the user selects a number of content records using the checkboxes, or the context menu on one record, and then clicks 'Link to Picklist.' A pop-up appears where the user selects one of the existing Picklists to add the contents to, then either Confirms or Cancels the operation.

To use the second method, the user selects a Picklist from the View menu first. While the user is in the Picklist, they select "Link Content to Picklist" from the Picklist menu at the top of the grid, the View switches to Default View to show all content, and the grid enters Link Mode. The user selects the desired content with the checkboxes and then has the option to Link to Picklist, Link and return to Picklist, or Cancel. The content is linked to the Picklist and the user either stays on the same screen or is returned to the Picklist.

**Figure 2.15. Link Content Options**



## 2.5.5. Unlink Contents from Picklists

Removing a content item from a Picklist only works from within the selection of contents that are already in the Picklist.

- Select the Picklist from the favorites or from the Views panel.
- Click the selection box for the desired content/s and choose "Unlink from Picklist" in the Selection menu above, or for a single record...

- ...Right click on one content and click on "Unlink from Picklist."
- Confirm the action.

## 2.5.6. Remove a Picklist

Click the grey x next to the picklist to remove it from the favorites at the top of the grid.

# 2.6. Tools

SLIMS provides powerful tools to easily handle the different records available in a grid.

## 2.6.1. Data Management using Excel

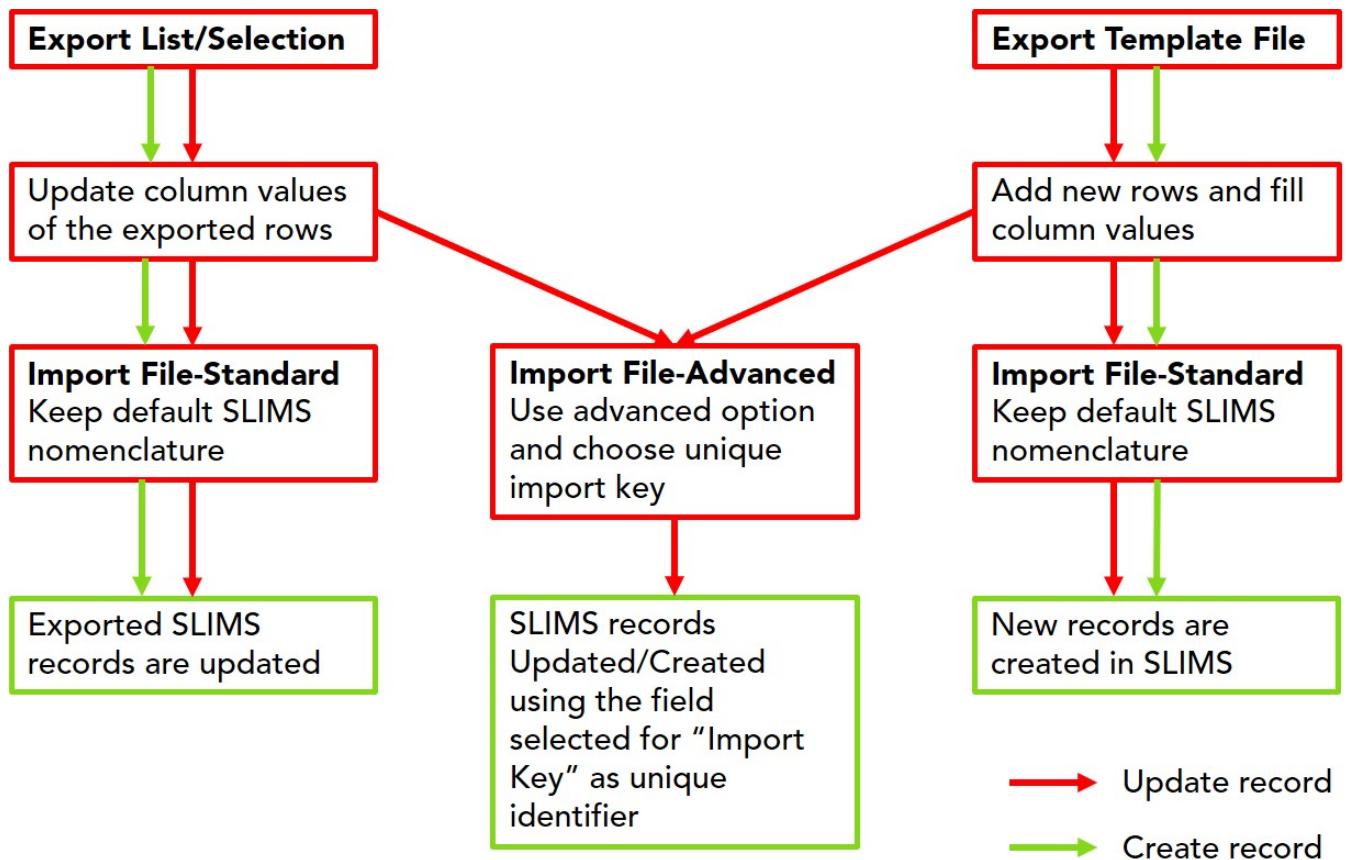
Export and Import for Excel is managed using four actions that are available in most grids with the goal of providing a file that helps users import new records or import changes to existing records. Export (Export list to Excel), Create Template File, and Import can be used from the Execution menu in a grid, and Export (Export selected items to Excel) becomes available from the Selection Menu when one or more items are selected in a grid. The use of each of these actions is detailed in their dedicated sections below. A flowchart follows that describes how to use export and import tools to create or update records in SLIMS.

The following Role Access rights can be configured to restrict the ability to import records to the roles that best understand the import process:

**Table 2.4. Excel Import Permissions**

Access rights Category	Access rights Name	Purpose
Setup: Miscellaneous	Can Import Records in SLIMS via Excel	When enabled, the role user can access the Import from Excel, Export to Excel, and the Create Template Import File actions in grids throughout SLIMS. When disabled, the role user cannot access the Import from Excel or Create Template Import File actions anywhere in SLIMS, but the role user is still able to Export to Excel. This permission is enabled by default for new roles.

Excel files that will be imported must be formatted for SLIMS to read them and turn them into new records or update existing records accordingly. There are a couple of ways to obtain a formatted Excel file for the user to modify.

**Figure 2.16. Export to Import flow using Excel Files**

### 2.6.1.1. Export List to Excel

Exporting an Excel file is best used when trying to update existing records. It is not ideal for creating new records because it introduces the risk that existing rows can be modified along with the new rows that the user adds. It is recommended to keep Update actions separate from Create actions. When a user clicks the "Export list to Excel" action from the Execute Actions menu, a pop-up is displayed with the following fields:

- **Total Rows:** Lists the number of records that will be exported.
- **Advanced Export:** If this box is unchecked, the resulting format is .xlsx, the file name is "tableName\_date\_time.xlsx" and the columns match the display of the grid when starting the action. Advanced Export options:
  - **Format:** Available format options are: .xlsx and .csv.
  - **File name:** Name of the generated file. By default it will be tableName\_date\_time.xlsx
  - **Columns:** This defines the columns that are present in the Excel file. By default, the columns present in the view in the grid when starting the action.

The population of records that will be exported are the ones that were visible in the grid when starting the action.

These changes are performed on the fields while SLIMS creates the Excel file:

- All the fields that refer to other tables are displayed with the display field of the corresponding record.
- All the fields of type quantity are exported as two columns: the first column is the value, and the second field contains the unit.
- All the fields of type attachment are exported as the name of the file. It is not possible to import them back.
- All the fields of type date difference are exported. They cannot be imported as they are automatically computed by SLIMS.
- All the fields that are not relevant for a record are mentioned as 'Irrelevant' and are not imported back.
- If column 'locationPath' is present, then three columns will be present in the Excel file. These are the location path excluding the position indication, the position row, and the position column.
- If the field location was selected, then the 'locationPath' excluding the position indication will be exported instead of the location.
- All the NA values are exported as '=NA()' . (This differs from an import, as cells containing NA values can be exported, but cannot be imported.)

The created Excel file has following properties:

- The GUID of the exported record is present in the hidden first column. This GUID is unique among databases. The GUID is used when re-importing the file to find the right record to update (see import Excel section for more details).
- The first row of the file contains the SLIMS logo and a sentence to explain when the file was created and that it is a file to update records.
- The second row is hidden. It contains the names of the displayed fields.
- The third row contains the label of the fields.
- The following rows (after the third one) each correspond to one record.
- For all the data cells that are numeric (SLIMS datatype quantity, whole number, decimal number, etc.), there is data validation in Excel.
- It is possible to remove entire rows/columns in an exported Excel sheet.

⚠ The order of the records in this file is crucial. Each row contains one record that is uniquely identified by the GUID in the first hidden column, and it causes issues if that GUID is displaced from the other columns relating to a record.

### 2.6.1.2. Export Selection to Excel

This action is available in the selection menu once one or more records are selected. The behavior is exactly the same as the Export List to Excel action, except that it only exports the

records that were selected instead of the entire list. For an advanced export, only the fields relevant to the content types of the selected content will be available (if less than 1000 samples are selected).

### 2.6.1.3. Create Excel Template

Creating an Excel Template is best suited for when the user wants to create new records but can be used to modify existing ones as well. It is recommended to perform Update and Create actions in separate files or import transactions in SLIMS to prevent issues. The action "Create Excel Template" triggers a pop-up that has the following fields:

- **Type:** (Only for contents and locations) the type of records to be created. Only records of the chosen type can be added using this template.
- **Fields:** The columns that will be available in the Excel file. By default, they correspond to all the relevant fields for that type of record. Only required fields for this content type can be selected by clicking on the icon at the end of the field. Read-only fields are not available because they are not editable and therefore there is no need to add them in the Excel template. The fields 'Barcode' and 'ID' are always available in the list when creating the template Excel file for Contents.

The created template file has the following properties:

- If the field 'locationPath' was selected for the export, then three columns will be available in the Excel file: the location path excluding the position indication, the position row, and the position column.
- In the hidden first column, the first cell contains the Primary Key (the key for the content type in the database) of the record type, as well as many other keys relating to the records.
- The first row of the file also contains the SLIMS logo and a sentence to explain when the file was created and that it is a file to create records of a specific type.
- The second row is hidden. It contains the names that map back to the displayed fields in SLIMS.
- The hidden second row (B) contains the SLIMS GUID which must not be changed or moved.
- The third row contains the label of the fields.
- For the rows following the third one the user can enter one new record per line.

### 2.6.1.4. Editing the File

The user opens the newly downloaded file on their local computer and starts the editing process. As stated earlier, the order of the records in the file cannot be changed. Each row contains one record that is uniquely identified by the GUID in the first hidden column which should never be displaced.

The top two rows in the file should not be modified (they contain the internal identifiers of record types, SLIMS logo, the names that map back to displayed fields, and the label of the fields). After the third row, the user can modify data for existing records or add new rows for new records. Entries for fields like dropdown, radiolist, multi-select, etc. should match the op-

tions that are possible for the field in the system. Once the file has been modified it can be imported back into SLIMS. There are a few restrictions and options the user should be aware of when importing.

### 2.6.1.5. Import Excel File

Import Excel File can be selected from the Execution Actions menu in a grid. The action imports the content of an Excel file that was previously generated by SLIMS either with the Export list action or with the Create Excel Template action.

When the Import Excel file action is used, the user is asked to select a file from their local computer. Once the file is selected the user clicks the "Start" button.

SLIMS executes a series of checks before the final import to ensure that the import won't fail. The list of checks is displayed in the pop-up window containing the Start button during their execution. In the event that the validation fails, the user is asked whether he wants to download a file containing highlights on the cells that caused the validation failure. A new Excel file is generated if the user accepts the download. The file has a light red background on the failing content rows and a red background and note for the cause of failure on the failing cells. Error messages are added at the end of each row.

The user has to restart the Import Excel File action for the corrected file to take effect. The whole validation process will start over.

The records are imported in SLIMS if the validation succeeds. The rule that governs whether a record gets updated or created during an import is that if there is a GUID in the Excel file for a record and the GUID can be found in the SLIMS instance, then the record 'exists' already and gets updated. Otherwise, the record does not already exist, and gets created.

#### Notes

- When importing new content with Excel *in the content list of the Orders module*, the imported content will be linked to the selected order and will become visible in that content list.
- When importing new content with Excel *in the content list of the content tab in a protocol run step of the ELN module*, the imported content will be linked to the protocol run step and become visible in that content list for that run step.
- Values entered in the Excel file as NA will not be imported.

#### Hints

- *Import a new Derived Content:* It is possible to specify the parent of the imported content record by using the column 'cntn\_fk\_originalContent'. First, the field 'cntn\_fk\_originalContent' has to be marked as not hidden in the 'Fields' module. Then, the column 'cntn\_fk\_originalContent' needs to be added to the exported Excel file where the Barcode of the parent content record can be specified.

### 2.6.1.6. Advanced Importing vs Standard Importing

An advanced import option is available where the user can select fields to create an import key. An import key is unique to a particular record, existing or new, so any fields that are cho-

sen must be those that have a unique value per record or the import will fail. This method is used when the user wants to update records that are identified with an external import key, and can only be used for updating records, not for new record creation. The following cases describe when an import using a key is subject to fail:

- If no record or multiple records are associated with the selected import key.
- If the Excel file contains two or more records with the same key. Highlights will be shown in the error Excel file on duplicated records with the same import key value.
- If the Excel file does not contain any values for one of the import key's fields.
- For any other error that would occur during an import without an external import key.

## 2.7. Rich Text Editor

A rich text field is in read only mode by default until a user clicks on it. Click inside the text field to enter edit mode. The cursor will replace the "Write your text here" placeholder. Click outside the editing window to quit edit mode and save.

A rich text field provides the following actions in edit mode:

- **Source:** Shows the HTML source text. It is only possible to edit the HTML source when it is visualized.
- Paste
- Paste from Word
- Print
- Undo
- Redo
- Find
- Replace
- Bold
- Italic
- Underline
- Strike through
- Subscript
- Superscript
- Insert/remove numbered list

- Insert/remove bulleted list
- Decrease indent
- Increase indent
- Block quote
- Align left
- Center
- Align right
- Justify
- *Link*: add hyperlink
- *Add a File*: Browse the file system to add a file as a link into the editor. Files can also be added using *Drag and Drop*.
- *Add an Image*: Browse the file system to add an image with preview into the editor. Images can also be added using *Drag and Drop*.
- Table
- Insert special character
- Font name
- Font size
- Text color
- Background color
- Maximize

When the rich text field is in read only mode, it cannot be changed except for file insertion by drag and drop. When it's in edit mode, text edition and drag and drop are enabled.

## 2.7.1. Saving

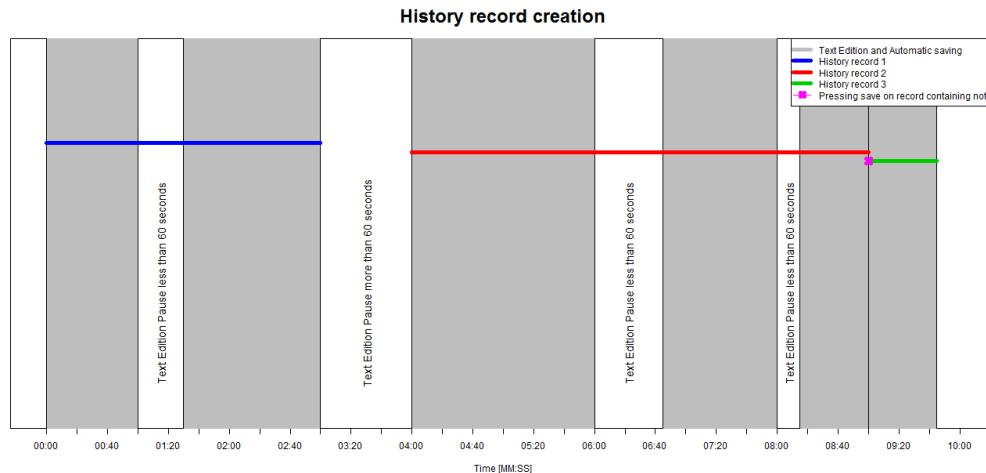
Every rich text field is saved automatically after each change (both key stroke and image insertion). It is also saved when the record containing the rich text field is saved manually.

## 2.7.2. History

A new history record is created for each new text edition when saved manually on the record containing the rich text field. When it is automatically saved, this rule is followed: A new history record is created for the first automatic save. The history record gets updated while editing

the rich text field. A sixty second break while editing creates a new history record. The following graph illustrates the creation of history records while editing text.

**Figure 2.17. Rich text fields and history records creation**



### 2.7.3. Drag and Drop

Drag and Drop is possible in both edit and read only mode. The main difference between the two is when doing drag and drop in read only mode, the file is automatically put at the end of the text, whereas the file is placed where the cursor is in edit mode.

Dropped files are shown as a file link or as an image preview in the RTF and automatically attached to the record containing the RTF. A question pop-up is shown to choose between adding previews or not when dropping files with a supported preview format: JPG, JPEG, PNG, GIF, BMP, PDF, DOCX, PPTX, TIF and TIFF. Files/images previews are surrounded by a 1px border line and have a hyper-link to open the file/image in a new tab of the browser, but the file/image is downloaded if the browser has no support to read the opened file/image.

⚠ Drag and Drop does not work with Internet Explorer, except from Version 11 or when attaching a file into a read only rich text field.

### 2.7.4. Hyperlinks

Whenever the rich text field contains a hyperlink in the text, it can be followed if the rich text field is in read only mode.

### 2.7.5. Editor Size

In read only mode, SLIMS fits editor within the page. SLIMS chooses the best display possible, whether there is space to display all visible lines, or to instead use a scroll bar to show all the lines.

### 2.7.6. Printing

The layout of the text is preserved when printing.

## 2.7.7. Spell Checker

The rich text editor uses the user's native browser spell checker on text in the field. Users have to make sure that their browser spell check is active and configured for the desired language.

The spell check is active only when a rich text field is in Edit mode.

## 2.8. Actions

An "Action" is a functionality that is applied to a record - for example: Edit or Copy. They can be executed on a record either by right-clicking on the record with the mouse or by checking the boxes next to multiple records and then accessing the Selection menu  containing the actions list.

ⓘ After executing an action on multiple records, the request is sent to the server where it is processed. Once the request is sent, the action cannot be stopped and will be executed on all records.

### 2.8.1. Remove permanently

The action "Remove permanently" is available and can be applied for all records in SLIMS.

By clicking the button Remove permanently  , record data is completely deleted from the database and cannot be retrieved.

The permanent remove button applies an irreversible process and therefore its access is restricted by the administrator according to Roles (See the *Access rights* section for more details).

### 2.8.2. History Action

History of records contains the log of events that the records participated in. The *History* section contains full details about what events are logged and how. History can be accessed from grid menus and also from right-click context menus under the "Show" sub-menu. By accessing the context menu for a record, the user can view just the log entries relevant to that record.

## 2.9. Icons Overview

In this section, the use of most of the icons that are present in SLIMS is explained.

### 2.9.1. General Use Icons

Icons that are common throughout SLIMS but do not relate to a specific functionality are explained below:

 **Switch Icons:** Switches replace checkboxes in many areas in SLIMS. When the switch is gray it is disabled / off / false, and when blue, it is enabled / on / true. Click the switch to flip between gray and blue.



**Multi-selection Add All:** When clicked, this icon adds all of the values in a multi-select dropdown list to the field's values with one click. When values are

added to the multi-selection dropdown list, a red X icon next to this Add All icon will instead Remove All of the selected values in the field with one click.

## 2.9.2. View Icons

Actions related to Views are shown below with a short description:



**Views Menu:** Provides access to tools to edit Views and a list of already customized Views.



**Add view:** Creates a customized Grid appearance with choice of filters, columns, sort, groups, and highlights.



**Edit View:** Edits the present view or creates a new one.



**Copy View:** Copies the selected view and creates an identical view, though the name must be different.



**Reset filter:** Clears all additional changes that have been applied to the view and resets the view to its last saved state.



**Save View:** Saves any temporary edits that were made to the present View.

## 2.9.3. Picklist Icons

Actions related to Picklists are shown below with a short description:



**Add Picklist:** Creates a list into which any desired records from the Content Grid can be linked, with a choice of columns, sort, groups, and highlights.



**Edit Picklist:** Edits the present Picklist or creates a new one.



**Copy Picklist:** Copies the selected Picklist and creates an identical Picklist, though the name must be different.



**Save Picklist:** Saves any temporary edits that were made to the present Picklist.

## 2.9.4. Tool Icons

Common tools in SLIMS are shown below with their representative icons and a short description:



**Tools:** Shows a menu of all available tools.



**Refresh:** Refreshes records of a Grid.

-  **Export list to Excel:** Extracts the List Grid data and exports it to the desired Excel format (csv, xls, xlsx).
-  **Create Excel Template:** Creates an Excel Template specific to the records belonging to the current grid so they can be filled out and imported back into SLIMS.
-  **Import Excel:** Import records from a filled Excel template that was generated using the tool Create Excel Template.
-  **Print:** Prints the current List Grid.
-  **History:** Access the history for the records displayed in the Content Grid. Options are available in the history pop-up to see detailed history of changes.
-  **Location Filter (On):** This filter appears in the Location filter in the left side of the content module. When the filter is blue, it shows an applied location filter. Click to remove.
-  **Location Filter (Off):** This filter appears in the Location filter in the left side of the content module next to each location. When the filter is gray, the location filter is not applied. Click to apply a filter (it will turn blue).
-  **Grid Location Filter:** When a location filter is applied on the grid, it appears above the content grid. Click the filter to remove.

## 2.9.5. Action Icons

Below are common actions that can be accessed by right-clicking records, or selecting records and clicking the button  :

-  **Remove permanently** : Deletes records permanently from the database. This action is not reversible.
-  **Unlink:** Unlinks Attachment. An attachment could be a file linked to a Content or a Content linked to an Order.
-  **Edit mode:** Opens a pop-up where the user can modify the data fields of the selected record(s).
-  **Add New:** Adds a new element (content, sublocation, filter clause...)
-  **Remove:** Changes a Content status to 'Removed' and its Location status to 'Available.' A removed content can be recovered using the 'Unremove' button.
-  **Unremove:** Changes a Content status from 'Removed' to 'Available.' A new Location has to be specified.
-  **Copy:** Creates a copy of the selected record, whether it is a content, order, macro, etc.. A new pop-up window will appear for users to specify values of the newly created record. The use and outputs of this action is described in detail separately

for each specific module where it is used. However, usually the Copy form will be a pop-up that provides a name field for the user to name the new record, and a unique identifier field if applicable that automatically suggests a new unique identifier based on the name.



**Derive:** Creates new content that will be derived from the selected content. During derivation, content of the same or a different type is added, and SLIMS maintains the relationship between the derived content and the original content. content.



**Aliquot:** Creates new content that will be aliquoted from the selected content. During aliquot, content of the same type is added and volumes of the aliquotted contents are tracked. SLIMS maintains the relationship between the aliquoted content and the original content.



**Dilute:** Creates new content by diluting the selected content.



**Mix:** Creates new content that will be a mix from the selected contents.



**Move:** Changes the location of the selected content to a new location that the user will have to specify.



**Attachments :** Opens a pop-up window that provides access to the attached files.

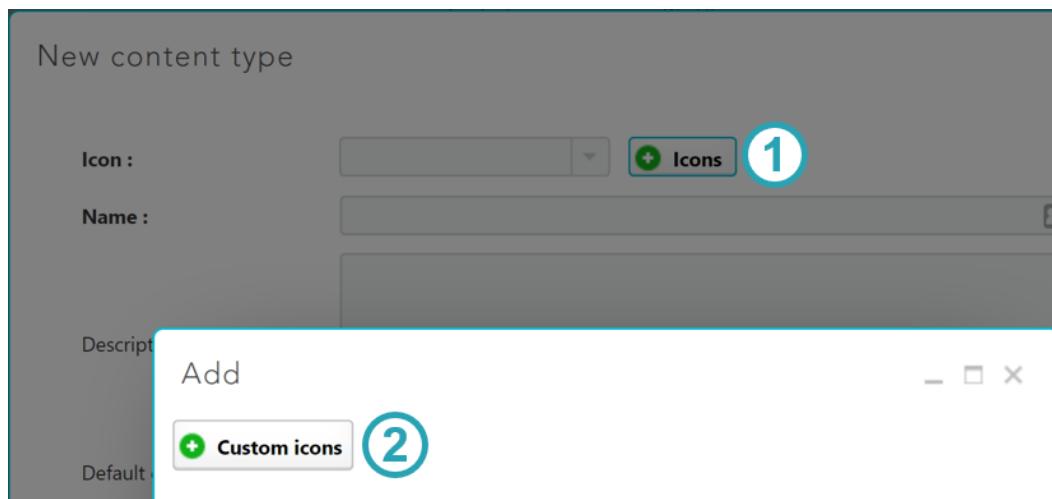


**Execute Flow:** Provides access to the available workflows in SLIMSGATE.

More usage-specific actions can be found in the Selection menu. Further details about them are available in the specific sections/chapter.

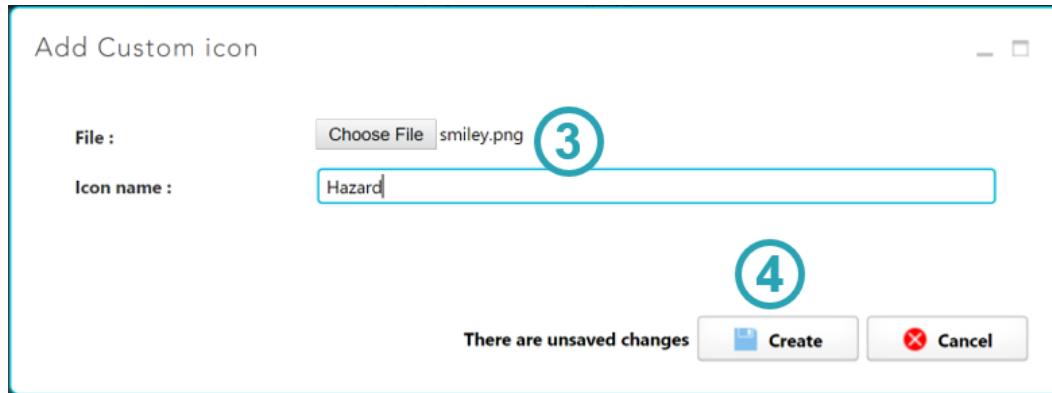
## 2.9.6. Custom Icons

Figure 2.18. Access Icons for an Element



Administrator users can create custom icons for elements like Content Types, Location Types, Inventory Types, and Macros. When adding an element such as a Content Type, a standard icon can be chosen from the dropdown. Alternatively, clicking the "Icons" button beside the dropdown shows the Add Icon pop-up. In this pop-up, previously added custom icons are available, and more can be added after clicking the "Custom Icons" button.

**Figure 2.19. Upload Custom Icons**



Click on the "Custom Icons" button. In the pop-up window, click browse to locate the image file to upload from the computer. Name the icon and upload any 16x16 pixel image file that is a JPG, PNG, or GIF file type. Click "Create" to save the icon, and then select it to use for the element being created.

# 3. Content Management

Within this chapter, all details about contents are described. A content in SLIMS usually represents a sample, but can also represent anything that needs to be recorded, maintained, or tracked. Contents are defined by content types. They are stored in locations which digitally mimic the structure of any lab and track where each content is physically located. Furthermore, contents are associated to barcodes so that a scanner can be used to access the details of a sample or apply an action to it.

Only the two last sections will be of everybody's interest. The previous sections explain the set-up.

## 3.1. Content types

Content components are defined as content types that group contents with common properties, such as Blood, Rat, Chemical, and so on. When setting up content types you can think about the type of samples your lab needs to track and what information is important to each kind of content. For example, the main samples, reagents, ingredients, and kits you use. In addition, the Reference Data related to samples can be created and edited using links available in the Setup menu.

The following Role Access rights can be configured for use in this module:

**Table 3.1. Required Permissions for Content Types Module**

Access rights Category	Access rights Name	Purpose
Setup: Content	Content Types	This permission provides access to the Content Types module in which role users create a standard format for each content type they will collect in their SLIMS instance. This defines the common characteristics of a content type, such as volume of blood in a Blood content, or what content types can be derived from a Subject.

The left side of the Content Types module can be collapsed, hiding the content categories

to save space. The  button is available when the Content Types tab is selected and will switch the page's layout to a vertical alignment. It can be switched back to a horizontal alignment with the  button. The selection is saved in the user's layout.

The module is divided into the left panel and right panel. The left panel lists all of the Categories that have been configured. The right panel shows the Details tab and the Content Types tab relating to the Category selected in the left panel.

- **Details Tab:** Contains the list of features that the content types within a category can have. The purpose for this tab is to set whether the value should be defined on the whole category or on each individual content type.
  - *Apply a value for all content types of this category:* When this is selected, the field value is shown and can be configured for the category. The value set here will be used for every content type within this category and will not appear in the content type forms because it does not need to be individually selected.
  - *Specify a value directly on each content type of this category:* When this is selected, the field will appear on content type forms of this category when the user adds them, and can/must be set on each content type individually.
- **Content Types Tab:** The grid part shows a list of all content types in the selected category. The form part shows the details of the content type selected in the grid. Only fields that must be set on the content types individually appear in the details form.

### 3.1.1. Content Type Categories

Categories are a functionality that make the creation of Content Types easier. There are many fields to define each content type, and categorizing them allows a superuser administrator with the "Can Configure Categories" permission to set up all of the features that are common to the content types within the category. Then a secondary admin or end user that does not have "Can Configure Categories" can add content types to the category, and those preset features will already be configured for the new content types that they add.

The relationship of Categories to Content Types is that categories can define all of the fields of the content types in the category. For example, with a reagent category, you can define all of the fields that all content types in the reagents category have, like adding an amount of material field and an expiration date (both of which work in a Link Reagents protocol step.)

The following Role Access rights can be configured for use in this module:

**Table 3.2. Configure Categories Permissions**

Access rights Category	Access rights Name	Purpose
Setup: Content	Can Configure Categories	This permission allows the role user to create categories and define the features that will be common to all Content Types within that Category. (This permission is useful for when roles are split between a superuser administrator who configures categories and other impactful parts of SLIMS, and other users who configure the content types and don't need to understand SLIMS on the same level as the superuser administrator.)

Content types are described by the following fields:

- **Icon, Name, and Description:** An icon can be set for the category or for each individual content type. When set for the category, every content type in the category will have that icon.  
*Custom Icons* can be added to customize Content Types, Location Types, and Instrument Types.
- **Additional Custom Fields on Content:** Additional custom fields can be created and applied to a whole category or particular content types.
  - **Name:** The name of the custom field from the Fields module.
  - **Description:** The description of the custom field from the Fields module.
  - **Label:** The human-readable label of the custom field from the Fields module.
  - **Show in Simplified ELN:** Check to show the custom field in Simplified ELN.
  - **Apply to All Content Types in this Category:** Check to apply the custom field to all of the content types in the category. It will then show up on the content form of the category's content types by default.
- **Amount of material field:** Select the field that is used to track the amount of material. Only fields with datatype Quantity can be used. The field can be selected among existing fields.
- **Default quantity:** A new Content of this type will get this quantity value by default. Default quantity is assigned to a category and/or content type in the Fields module. For example, the default quantity of an animal may be 1. Default values can be changed.
- **Default unit:** A new Content of this type will get this unit value by default. Default unit is assigned to a category and/or content type in the Fields module. For example, the default unit of a liquid may be ml. Default values can be changed.
- **Expiration Date Field:** This field can be applied to a category or content types to track when the contents expire. Only custom fields with a Date or Date and Time datatype can be applied as an expiration date field.
- **Default tests:** Shows Tests applicable to this Content Type.
- **Additional tests:** Shows all available tests, even those restricting this Content Type.
- **Default location types:** Shows Location Types where this Content Type can be stored. It represents Location Types that did not restrict this Content Type.
- **Additional location types:** Shows all available Location Types, even those restricting this Content Type.
- **Use barcode as id:** This option will use automatically the barcode as id: you will not be able to choose the id for Contents of this Type. Typically we don't activate this option for animals since we prefer to have names as their id and the barcode will represent their identifier.
- **SLIMS generates a barcode for content of this type:** Please refer to the Barcode Definition for more details about barcodes.

Concerning the Barcode mask, the user can either type in a new mask directly (use # for numbers) or select an already existing mask from the list of Sequences of type 'Barcode.' Please refer to the Sequences module for more details.

- **Can be in container:** This is available when Container types are enabled. This option allows the Content type to be added into any created Container types, and provides a field to select a container type in the "Create Content" window.
- **Requires a container:** This is also available when Container types are enabled. Appears if "Can be in container" is enabled. When this option is enabled, Content of this Content type cannot be created in the Content module with the "New Content" action unless a container type is selected.
- **Can only exist in selected Container types:** This is also available when Container types are enabled. Appears if "Can be in container" is enabled. When this option is enabled, a drop-down selection of all available Container types is provided. Container types selected in this list will narrow container types available in the dropdown field in the New Content form for end users.
- **Content type is derivable:** Activating this option allows you to derive Contents of this Type. For instance, blood content type should be derivable in order to derive DNA from it.
- **Content types to derive to are restricted:** Only selected Types can be derived from Contents of this Type. Thus, blood could be restricted to derive DNA but not bones.
- **Content of this content type can only be made by derivation and cannot be added.:** Derived content cannot be added the same way as other content, and is only created by deriving this content type from other content. (As an example, a liver cannot be added, but can be derived from a mouse).
- **Content type is aliquotable:** Aliquoted content is created from content of the same type, dividing the volume of the original into the aliquots' volume and the remaining volume. (For example, blood could be aliquotted into smaller blood samples, and could use up the original sample or leave some remaining.) Using this option allows content to be aliquotted.
- **Content type is dilutable:** Activating this option allows Contents of this type to be diluted.
- **Can be an ingredient:** Activating this option allows Contents of this type to be used as *ingredients*. Ingredients are used to create mixes.
- **Can be a mix:** Activating this option allows this type of Content to be created by mixing other content designated as ingredient types together.
- **Enroll content in study:** Activating this option allows Contents of this Type to be enrolled into studies.
- **Only for selected roles:** Activating this option allows access to Contents of this Type to be restricted to certain roles.

Other fields can be specified at Category, Content types, or Content levels. For further information on it, see the custom fields section.

### 3.1.2. Moving Categories

Content types can be moved from one category to another when needed. The "Change Category" action is available in the right-click context menu and the selection menu. When "Change Category" is clicked, a pop-up form appears so the user can select the category the content type should be moved to. SLIMS looks at the selections of both content types and compares them to see what fields will be gained or lost by the move into the new category. The outcome

is provided in the pop-up once the Target Category is selected. The user should review the message about the affected configuration and the new and lost content fields before confirming the "Move" action.

⚠ "Kits" and "Containers" are special categories that require separate role access rights and a different set of field values. It is not possible to move a content type from a standard category to the Container category or from Container or Kits to a standard category, but a content type can be moved from a standard category to a Kit category.

## 3.2. Kit Configuration

Kits are a useful category in the Content Types module that allow all of the linked Content Types inside the "Kit" to be added and derived from the parent kit all at once.

For example, you could have a kit containing chips, reagents, samples, and tubes (content types and containers), so that when you add the kit in the content module, it is done in one action. The kit is added as a content, and the chips, reagents, samples, and tubes are all automatically derived from the kit and added as content because they were set-up to be the Kit components.

This functionality requires the following Role Access rights to work:

**Table 3.3. Kit Category Permissions**

Access rights Category	Access rights Name	Purpose
Routine: Content	Can create kits	This permission adds a new default category in Content Types called "Kits," and allows the role user to create Kits in SLIMS once all of the components of the kit have been defined. The definition of content types within this category has less to configure and is suited for connecting all the reagents, containers, etc. of a kit in SLIMS. It allows all components of a kit to be derived at once from the main Kit content when it is initially created.

When the Can create kits Access rights is enabled, the new category type "Kits" is added in the Content Types module. This is used as a parent category, or Kit Template, for all the Content Types that are set up to be part of the kit.

The options for Kits are trimmed down from the options available for other categories. Most of the features are defined on the individual Content Types instead of in the Kit Template. The Kit can be designed on the same two tabs used for categories:

- **Category details:** This tab allows the Administrator to specify the kit content itself. You can apply a value for all content types of this category, or choose to specify a value individually for each content type of this category. The kit's list of options are:

- Name
  - Icon
  - Description
  - Default Custom Fields on Content are displayed, if any were set up
  - *Additional Custom Fields on Content:* Custom fields such as expiration dates and kit amount of material fields can be added.
  - *Amount of Material Field:* A custom field can be created to use for the amount of material. This is typically a scalar type quantity, and is used to say how many materials are in the kit. If the kit is linked in a reagent step, this field tracks the amount of materials that are used in the protocol. For example, a kit might contain 300 units, enough reagents and materials to analyze 300 samples.
  - *Expiration Date Field:* Once an expiration date custom field is created in the Fields module, it can be selected as the expiration date field for the kit's content types.
  - Additional Location Types
  - Additional Tests
  - Only for Selected Roles
  - Use Barcode as ID
  - *SLIMS Generates a Barcode for Content of this Type:* It is important to recognize whether a SLIMS-generated barcode is a viable ID for your lab or not. If the barcode needs to match a supplier-particular ID, such as a Lot number, in case faulty kits need to be replaced, manual generation of the ID may be more beneficial.
- **Kit types:** This tab allows a user to create a new kit type. Kit components have an additional dropdown for the user to select the Content types that are part of a kit along with their associated amount. The selected Kit components will be displayed in the table with their Icon, Name, Description, Amount, and Unit. The Amount column is not the same as the "Amount of Material Field" set on the Kit, but comes from the Amount field set individually on each of the Kit components. Thus, the amount type can be different for Kit components like Chip (unit), Dye Concentrate (ml), and Reactants (g). Where the Amount of Materials field on the Kit tracks the total number of kit content, the Kit components Amount fields track the individual content amounts.

### 3.3. Container types

The Container Types module is useful in cases where a container that holds a content needs to be tracked, not just the content inside. For example, a study may have tubes that contain barcoded samples, but the samples might not be added at the start. In this case, the tubes are tracked as empty vessels before the samples are added to the tubes. Therefore, the concept of a container type that behaves similarly to a content type is needed. Container types can be set up for content that use optional containers and content that require containers.

This module requires the following Role Access rights to work:

**Table 3.4. Container Type Permissions**

Access rights Category	Access rights Name	Purpose
Setup: Content	Container types	Allows the user to access the Container types module in order to configure Container types. When enabled, this provides some related Content type Restrictions to choose from in the Content types module.

This module allows the creation of Container types in the Content module. Container types can be set up with a barcode and/or identifier, and require an icon and a name. Container types define Containers, which are content that contain other later-defined content.

Container types are created by defining these fields in the module:

- **Icon:** Like other Content types, an icon should be chosen to represent the icon visually to the user.
- **Name:** The name that identifies the Container Type that will be available for selection to the user when they add Containers in the Content module.
- **Description:** As usual, it is best practice to add descriptions to remind yourself and other users what the Container Type is to be used for.
- **Additional Tests:** The tests a Container Type can be used in can be selected. Any default tests will be automatically listed when the Container Type is created.
- **SLIMS generates a barcode for content of this type:** SLIMS can either generate the barcode automatically for containers the user adds, or the barcode can be entered manually. Please refer to the Barcode Definition section for more details about barcodes.
- **Barcode Mask:** Concerning the Barcode mask, the user can either type in a new mask directly (use # for numbers) or select an already existing mask from the list of Sequences of type 'Barcode.' Please refer to the Sequences module for more details.
- **Only for selected roles:** Like in other areas of SLIMS, the roles in the instance are available to restrict access to who can use this Container Type.
- **Active:** When enabled, this Container Type can be created. When disabled, it cannot.

When Container types are enabled via the access rights permission for a role, several other Content type restrictions become available pertaining to Container types. See the Content types section for the additional Content type restrictions.

See the *Create container* section to see how Container types are used in the Content module.

### 3.3.1. Applying Custom Fields to Container Types

Custom fields can be created that apply to container types, such as the volume of the container. Custom fields can be restricted to particular container types using the parameters:

- **Only for selected categories:** By choosing the DefaultContainers category
- **Only for selected Content types:** By choosing container types from the list

Fields that are unrestricted or are restricted with either of the above options appear in the container creation form.

Any field applied to the container contextually also applies to the content within the container. In SLIMS, this means the field should apply to the container type and also to any of the possible content types the container could be assigned when it becomes filled. This prevents the custom field from being lost when the container type changes its content type. For example, a container like a tube will have the same dimension, temperature, volume, etc. as the sample that fills it. If any content that could fill the tube does not have the custom fields that apply to the container applied to them, SLIMS produces a warning message about it.

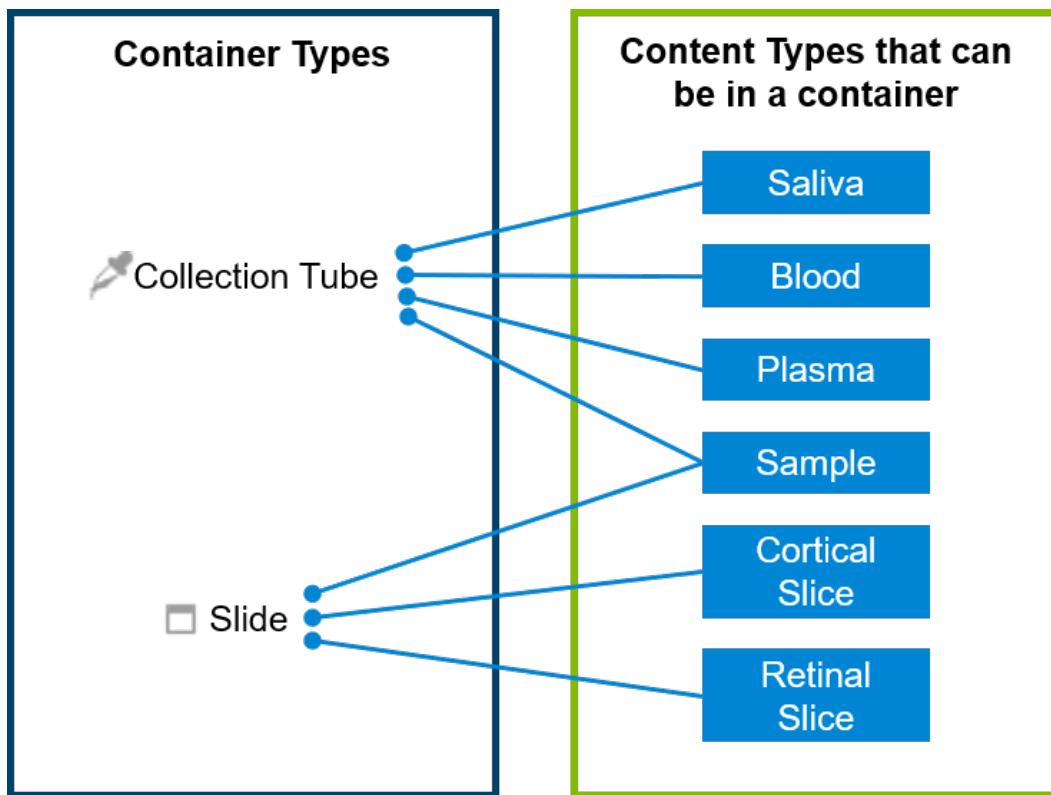
To set up a custom field for a container type and avoid such a warning, the field uses "Only for selected content types" and can use "Only for selected categories" as well.

- When creating the field, select all the particular container types the field should apply to and all the content types that could be in the selected container in the "Only for selected content types" multi-selection list.
- "Only for selected content categories" could be used as well. Select "DefaultContainers" and the category of the content types that the container could contain. When using the category restrictions, either use "Apply to all content types in this category" to ensure the content types that could be in the selected containers are included, or if not all of the category's content types apply to the container, use the "Only for selected content types" restriction to specify them.

If you try to save the custom field and there are content types that are not selected that can fill the selected container types, a warning will appear that blocks the field from saving. It lists the content types that have been left out of the selection.

To illustrate with an example:

- The category "Collectibles" exists with the content types: Saliva, Blood, Plasma, Retinal Slice, Cortical Slice, and Sample.
- Saliva, Blood, and Plasma require a container in order to be created, and the container type they require is Collection Tube.
- Retinal Slice and Cortical Slice require a container, and the container type is Slide.
- Sample requires a container, but it is not restricted to a particular container type.

**Figure 3.1. Diagram for example; content types that can be in a container**

The custom field `cntn_cf_containerVolume` can be created with DefaultContainers and Collectibles selected for the restriction "Only for selected categories" and Apply to all content types is unchecked for both. If only Collection Tube, Blood, and Saliva are selected in "Only for selected content types," a warning will display to inform that Plasma and Sample are content types that can be in the container, but are not selected based on the fact that:

- Sample can be in any container type and could thus be in a Collection Tube, which is selected.
- Plasma requires a Collection Tube but the field is set up to apply to Collection Tubes but not to Plasma.

There would be no warning if Plasma and Sample were added to the selected content types in the restriction.

## 3.4. Reference Data

The Reference Data Module was created to support data that needs to be collected for dynamic custom fields. While the Disease, Products, and Sources modules (discussed in the next sections) are available to categorize some dynamic custom fields, diseases, products, and sources can instead be set up as Reference Data. This allows those types and any additional custom context for data that the lab needs to collect to be configured in one module.

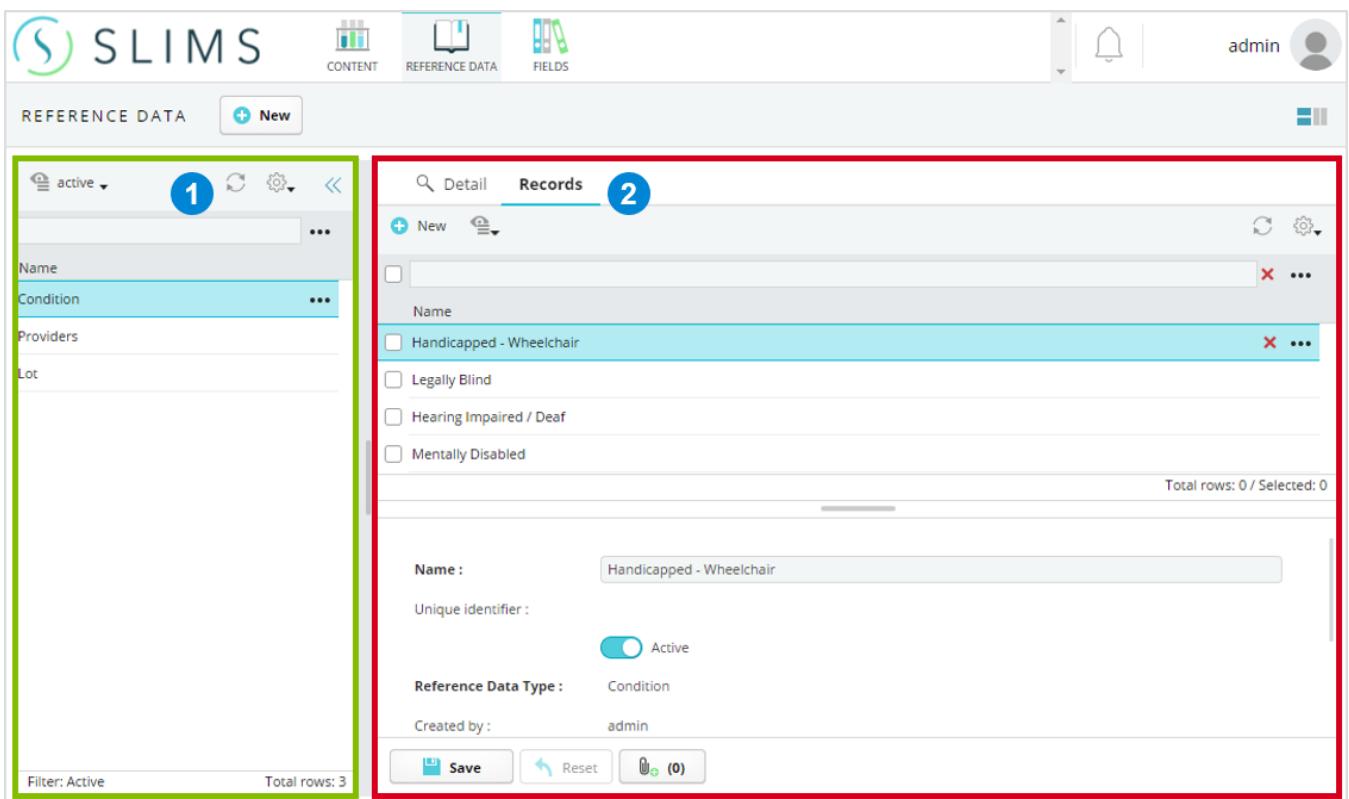
The following Role Access rights can be configured for use in this module:

**Table 3.5. Required Permissions for Reference Data Module**

Access rights Category	Access rights Name	Purpose
Setup: Miscellaneous	Reference Data	This permission provides access to the Reference Data module so the role user can categorize types of data that will be collected.
Setup: Miscellaneous	Can create reference data records from within a dynamic choice custom field	This permission allows a role user to view the "Create New Record" action in dynamic choice fields. The action allows the role user to add reference data records directly in the dynamic choice field in modules outside of the Reference Data module. Whether the "Create New Record" action is enabled or disabled is controlled by a field restriction, not this permission. When this permission is disabled, the role user cannot see the "Create New Record" action.

The  button is available when the Records tab is selected and will switch the page's lay-

out to a vertical alignment. It can be switched back to a horizontal alignment with the  button. The selection is saved in the user's layout.

**Figure 3.2. Add a Reference Data Type and Records**

The module has a similar layout to most others in SLIMS. The left panel (1) contains a list of Reference Data Types. This panel can be collapsed to save space. Each Reference Data Type has the following attributes:

- *Name*: The name of the Reference Data Type, which should be comprehensible to any other users of the same SLIMS.
- *Unique Identifier*: Automatically generated as the user types in the Name field.
- *Description*: Any desired text that helps explain the usage of the data type.
- *Active*: When enabled, this data type is available for selection when configuring Dynamic Choice fields. When disabled, it is no longer available in field configuration. Deactivating the data type does not get rid of the field it was configured with, but it hides the data type from being selected in the form of new subjects.

The right panel (2) contains two tabs. Detail displays the name, description, and status of the Reference Data Type that is selected in the left panel. The Records tab contains the related data options that will be available for selection in the field configured with the Reference Data Type.

- *Name*: The name of the reference data record. The Name field is required and unrestricted by default. The name field can be edited in the Fields module to not be required, and to be restricted to particular reference data types. This means that reference data records do not have to have names.

- **Unique Identifier:** Automatically generated as the user types in the name field. If the name is left empty or is restricted to not be present on a reference data type, the unique identifier will not be automatically generated. The user will need to fill the field manually, or the administrator can configure a value expression to generate a reasonable identifier: see the Unique Identifiers Generation section for an example of an expression.
- **Active:** If disabled, the records will no longer be displayed for selection in a form for new content, though existing selections are retained.

Records can be selected individually or in batch using the multi-select checkboxes, as in other grids. These actions are available upon selection:

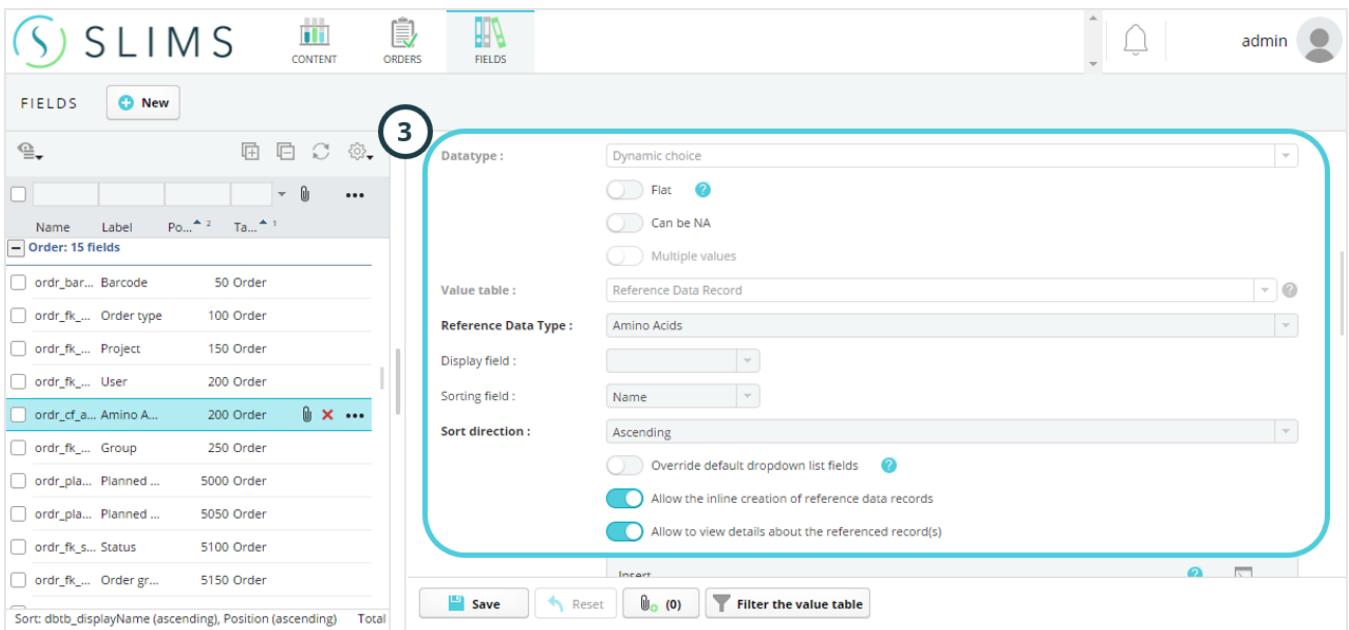
- **Remove Permanently:** This allows you to permanently remove records in batch.
- **Export:** This allows you to export an Excel file of the records in batch.
- **Edit:** The Edit action is available from the Selection Menu and also from the context menu by selecting multiple samples and right-clicking. The edit action opens a pop-up window where you can edit one or more records in batch. You can choose 'Specify New Value' or 'Keep at Current Value' to define what values the selected records will have. The edits apply to all of the records you selected.
- **Copy:** The Copy action is available from the Selection Menu and also from the context menu by selecting multiple records and right-clicking. The copy action opens a pop-up window where you can copy one or more records in batch. You can choose the number of copies that you want to make of each record. While performing the copy action, you can also define any edits to have on the duplicates by Specifying a New Value. SLIMS automatically suggests unique identifiers for the copied records based on the name supplied for the reference data records being copied.

The relationship between Reference Data Type and Records can be explained as a category (like 'pre-existing conditions') with a list of possible dynamic choices (legally blind, hearing impaired, handicapped, etc.).

### 3.4.1. Reference Data Usage as Dynamic Fields

Once created, fields can be added in the Fields module that use the Reference Data Type and the Records. The Field should have "Dynamic Choice" as the Datatype, and in the Value Table, either Reference Data Type or Reference Data Record can be selected.

Figure 3.3. Use Reference Data in a Field



- Using **Reference Data Type** for the value table provides the Display Field and Sorting Field lists. When a Reference Data Type is selected in the value table, that data type will be selectable from the field on the related form.
- Using **Reference Data Record** for the value table prompts the selection of the related Reference Data Type as well, and the Display Field and Sorting Field lists are available. In this case, each of the names configured in the Records tab are available as options in the dynamic field on the related form.
  - ① It is important to select a display field that makes sense for the record. If names are hidden due to reference data type restrictions, another field should be selected as the display field to prevent the list from being empty in the related form.
- Using **Allow the inline creation of reference data records** enables users with the "Can create reference data records from within a dynamic choice custom field" Access rights to add a reference data record directly to the field without having to navigate to the Reference Data module. When disabled, the "Create New Record" option is not available.
- Using **Allow to view details about the referenced record(s)** provides the Details icon  beside the dynamic choice dropdown in the case of a single selection field, or beside each option in the dropdown in the case of a multiple selection field. The details icon opens a read-only pop-up window that shows the record information from the Reference Data module, as well as the attachments on the record. The value tables this action is available on are:
  - Attachment
  - Case Report Form
  - Content
  - Content Event

- Content Event Type
- Content Type
- Experiment
- Group
- Grid Template
- Instrument
- Location
- Location Type
- Order
- Order Type
- Printer
- Product
- Project
- Protocol
- Protocol Run
- Protocol Run Step
- Protocol Step
- Reference Data Record
- Reference Data Type
- SOP
- Sequencing Settings
- Status Workflows
- Study
- Test
- Test Group
- User
- Workflows

The forms for complex items like products, grid templates, or protocols show the Detail form only, not the meta-tree, specifications, or other parts that are too complex to render. For

Tests, the form shows both the test form and the result datatype form combined. Certain fields for Users are hidden for security reasons: Password, Language, Browser notification fields, all OpenLab ECM fields, Locked status, and Logged in status.

① Dynamic choice values can be filtered on the field level using Dynamic Filter Expression and Filter Value Table. The filtering works on the Reference Data Type, not the Reference Data Records themselves.

## 3.5. Providers

Providers is a module to store a list of providers that can be linked to contents or other records to add context.

The following Role Access rights can be configured for use in this module:

**Table 3.6. Required Permissions for Providers Module**

Access rights Category	Access rights Name	Purpose
Setup: Content	Providers	This permission provides access to the Providers module in which you can store a list of the Providers for your lab. Providers can be referred to on any kind of record so you know which provider is associated (to a content, order, test, and more.).

Providers are described by the following fields:

- **Icon, Name, and Description**
- **Address:** Address of provider
- **City:** City of provider
- **Zip code:** Zip code corresponding to the city of the provider
- **Country:** Country of the provider
- **E-mail:** E-mail of the provider for contact purposes
- **Active:** "Provider is active" means the provider can be selected from the list of providers.

## 3.6. Customers

Customers is a module to store a list of customers that can be linked to contents or other records to add context.

The following Role Access rights can be configured for use in this module:

**Table 3.7. Required Permissions for Customers Module**

Access rights Category	Access rights Name	Purpose
Setup: Content	Customers	This permission provides access to the Customers module in which a list of your lab's customers can be stored. The customer can be referred to in other modules so you know who requested it (for example, on content or on an order).

Customers are described by three fields: Name, Description, and Active. Other parameters can be defined directly in the 'Fields' module.

## 3.7. Sources

Sources is a module to store a list of sources that can be linked to contents or other records to add context about where the records came from.

The following Role Access rights can be configured for use in this module:

**Table 3.8. Required Permissions for Sources Module**

Access rights Category	Access rights Name	Purpose
Setup: Content	Sources	This permission provides access to the Sources module in which a list of sources is created that can be used to identify the source organism or tissue of a content.

Sources are described by three fields: Name, Description, and Active. Other parameters can be defined directly in the 'Fields' module.

## 3.8. Diseases

Diseases is a module to store a list of diseases that can be linked to contents or other records.

The following Role Access rights can be configured for use in this module:

**Table 3.9. Required Permissions for Diseases Module**

Access rights Category	Access rights Name	Purpose
Setup: Content	Diseases	This permission provides access to the Diseases module in which a list of diseases is created that can be used to identify

Access rights Category	Access rights Name	Purpose
		fy what diseases content has, if any.

Diseases are described by three fields: Name, Description, and Active. Other parameters can be defined directly in the 'Fields' module.

## 3.9. Locations and Location Types

SLIMS provides Location components to organize and keep track of samples and contents in the lab. If using Contents, at least one location should be defined to show where the content resides.

Like Contents, Locations need to have their properties defined by Types. Location Types group locations based on similar properties and allow them to be managed using the Location module (For example, labs, sample boxes, refrigerators, shelves, and data banks can all be defined with different properties using location types).

The following Role Access rights can be configured for use in these modules:

**Table 3.10. Required Permissions for Location Modules**

Access rights Category	Access rights Name	Purpose
Setup: Location	Location Types	This permission provides access to the Location Types module in which role users define the standard properties of a location type. (For example, the kind of content a box in a fridge can contain, or how many slots are in a plate well.)
Setup: Location	Locations	This permission provides access to the Locations module in which role users can create locations to show where content is at any time.

### 3.9.1. Location Types

Every Location requires a Location Type, which defines a group of locations that are identical. An example of this relationship are the locations Building A, Building B, and Building C that all have the same setup, such as the available space and equipment that they hold, and are all defined as the location type 'Building.' The Location Types module is accessible under the Setup tab.

In the right panel of the Location Types module, two tabs are present: *Detail* and *Excel Template*. Within the first tab, a detailed description of the Location Type is given whereas in the second tab an Excel file can be generated and other Excel files can be added as attachments.

A Location Type is described with the following fields:

- **Icon, Name, and Description**

- **Don't require positions:** Content and sublocations stored in this type of location will not require a position. This means that an infinite number of content and sublocations can be added. A location type can be switched from requiring positions to being positionless because any defined positions can be included in an infinite position structure, but not the other way around (because an infinite number of positions cannot fit into a finite set of positions). When a location type is changed from requiring positions to being positionless, all locations of that location type will be updated.
    - The update will remove column and row values for all content and sublocations associated with the locations of that location type if the number of filled positions in the location is lower than 1000.
    - The update will not clear the column and row values associated with the locations of that location type if the number of filled positions are 1000 or more. In this case, when a content record in one of these locations is updated, the values will be cleared.
  - **Default number of rows/columns:** Locations in SLIMS have two dimensions called rows and columns. This is a two-dimensional representation of the space that sublocations (such as racks or boxes) or samples can occupy in a location. When creating a new Location, the number of rows and columns has to be specified unless "Don't require positions is checked." A newly created Location of this Type gets by default these values. Typically, a 96 well plate has 8 rows and 12 columns by default. Changing the location's number of rows and columns remains always possible during or after the location type creation. As a location can contain sublocations, one can represent any kind of Location.
  - **Name mask (use # for numbers)**
  - **Short name mask (use # for numbers):** The mask of the generated short names by using the 'Add multiple' feature.
  - **Adding content is allowed:** Activating this option allows Content to be stored in Locations of this Type.
    - **Adding content is restricted to certain content types:** Only the selected Content Types can be stored into Locations of this Type, e.g., cells can be stored into a plate Location but humans cannot.
  - **Adding sublocations is allowed:** Activating this option allows you to add sublocations to Locations of this Type. It is typically allowed for freezers or buildings, but not for plates.
    - **Default type of sublocation:** The default Type assigned to sublocation of Locations of this Type. Changing the sublocation's Type remains always possible during or after its creation.
    - **Default type of sublocation:** Sublocations Types are restricted to the specified Types. For example, freezers could be restricted to contain boxes or plates but not a building.
    - **Adding sublocations is allowed in the Content section:** Activating this option allows you to add sublocations in the Content section. This is useful for Roles that don't have access to the Location setup but need to add sublocations, for instance, lab technicians that add plates to a freezer.
- Adding sublocations in the Content section is restricted to certain roles:** In the Content section, only the selected Roles can add sublocations to Locations of this new Type, for example lab technicians or power user.

**Adding sublocations in the Content section is restricted to certain location types:** In the Content section, only the selected Location Types can be added as sublocations to Locations with this new Type.

- **Use barcode as name:** Activating this option allows SLIMS to automatically use the barcode for the location as the location's name. In this case, the user will not have to / be able to enter a name for locations of this type, and will not be able to edit the name after it is added. This is true of sublocations as well. When users add sublocations that have "Use barcode as name" enabled on their location type, the user will not have to / be able to add a name when creating the sublocation. It will automatically be named as its barcode upon creation.
- **SLIMS generates a barcode for locations of this type:** Please refer to the Barcode Definition section for more details about barcodes.

Concerning the Barcode Mask, the user can either type a new mask directly (use # for numbers) or select an already existing mask from the list. Please refer to the Sequences Module for more details.

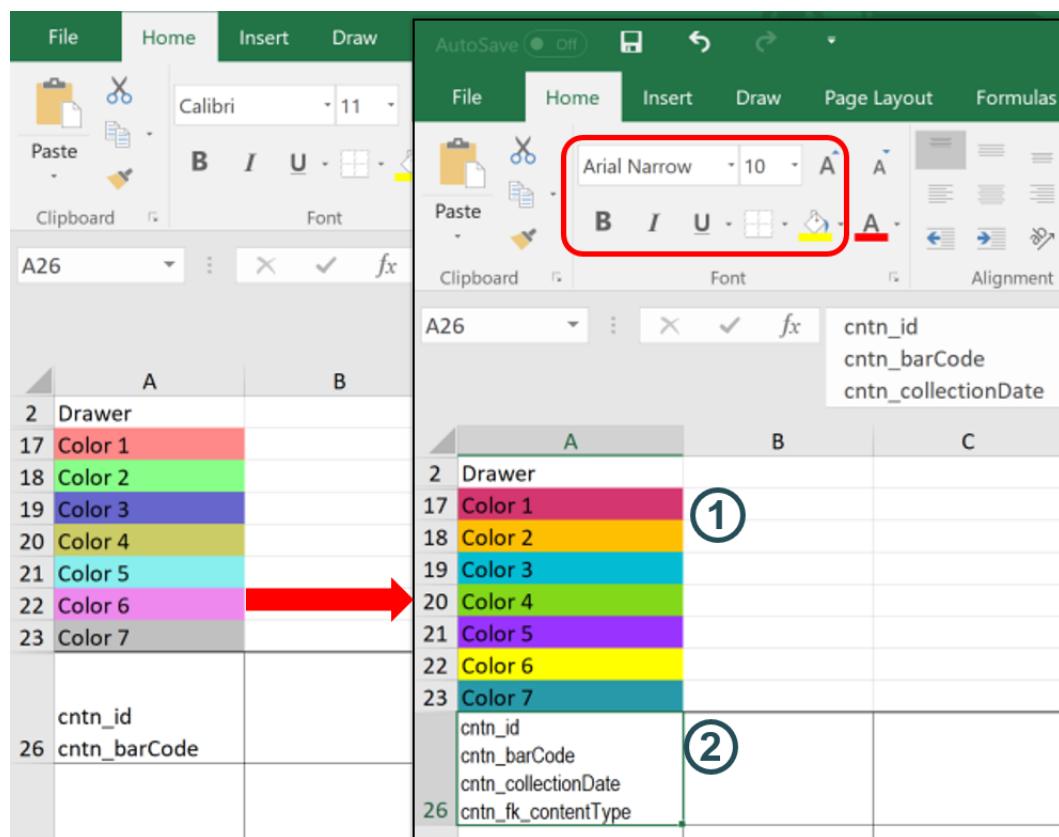
- **Rows alphabetic, Columns alphabetic:** Activating this option allows rows and/or columns to be named alphabetically instead of numerically. This means that for example the first column will be named A, then B, ..., Z, AA, AB, etc. When this option is enabled, rows and/or columns can be specified via their alphabetical position but also via their numerical position.

In the Excel Templates tab, the "New" button enables users to attach new Excel template files. The pop-up window provides several parameters to set while importing the Excel template:

- › **Excel file:** Browse and select the Excel template file to upload.
- › **Default discriminator field:** Choose the field that should be used for highlighting.
- › **Number of columns and Number of rows:** Number of rows and columns of the location.

The second icon is "Download Default" which can be used to download a default Excel template. The user can download it, edit what information should be shown (by adding field names into the Excel cells), changing the colors, and changing the styling. The content information, style, and size need to be changed within the first location position cell (2 in the image below). The colors of the position rows (1 in the image below) can be updated by changing the background color of the colored cells above the first location position cell. .

Figure 3.4. Import Location Types from Excel



### 3.9.2. Locations

Locations can be added, modified, or deleted in the Locations module  that can be accessed from the Setup menu. Adding or editing a sublocation is possible from the *Contents module* by using the Location grid if the user has the *appropriate access rights*. A green plus (+) icon indicates that the user has permissions necessary to add a new sublocation, and light gray icon  indicates that a sublocation cannot be added. Hovering over the icon displays a tool tip to clarify whether sublocations can be added or not.

Locations are defined by the following fields:

- **Barcode and Name:** The character '/' is not allowed in the name because of the conflict with the location path. SLIMS automatically removes the '/' character when used in the name.

If the barcode is being used as the location name, then the name will not be available and will be filled automatically with the barcode when the location or sublocation is created. If the barcode is generated by SLIMS, the user will not need to fill it manually.

- **Path:** The complete Location path. For example `freezer1/drawer2[1,9]` indicates the position [row=1,column=9] of drawer2 in freezer1.

⚠ The location path cannot be used for Excel export/import. Sublocations can be imported by using the field 'lcnt\_fk\_parent' that should contain the parent Barcode.

- **Type:** The Location Type, e.g., building, box, plate.
- **Short name:** A short name to be used instead of the name. It is used in the path (for example, frz1/drw2[1,9]). Using short names is helpful for printing location paths in small labels. The character '/' is not allowed in the Short name either, and will be removed automatically by SLIMS if used.
- **Description:** A description of the location, what can be stored there, or any other helpful text can be added.
- **Rows, Columns:** The number of rows and columns in the location. For instance, a 96 well plate has 8 rows, 12 columns and thus 96 sites in total.
- **located at Row / Column:** Indicates the position in the Location. For example, a drawer located in the third level of a freezer would have 3 and 1 as rows' and columns' numbers respectively. By default, the value of the row and the column correspond to the first free position. These should be left empty if the location is not contained, e.g., a building.
- **User:** The user will have access to the location and the possibility to modify or delete it (if he has access to the Location module). The location will be accessible only by that user if the group is not filled in (personal item).
- **Group:** User belonging to the group will have access to the Location and can add Contents or sublocations. This is possible only if a group member have at least group access.
- **Active:** An active location or sublocation can have content added to it if allowed by the configuration of the Location Type. Making a location inactive prevents content from being added to it. This status also relates to the filter in the left panel of the Content module, allowing users to view all, active, or inactive locations depending on the filter they choose.

ⓘ All fields are available in the location module and as columns in Locations grid.

### 3.9.3. Actions

This section contains a detailed description of the actions that can be used on Locations.

Once added, locations can be accessed in the Location Tree in the left panel of the Content module. The following actions can be performed when right-clicking on a location in that panel:

- **Show in location module:** Redirects the user to the Location module and selects the given location.

The following actions are available when right clicking on the location:

- **View in context:** Shows the location in the complete location tree. This action is only available when a filter is applied to the location tree.
- **Location Link:** This provides a pop-up window with a URL that leads back to the specific location that was right-clicked. This link can be copied and pasted into an email or message to send to another team member and will link them to the same location, provided that they have access.

ⓘ Location links are disabled if the lab setting 'Use pagination in Location Trees' is checked.

### 3.9.3.1. Copy Location

The 'Copy' action is available in the Locations module by right-clicking on a location. This action provides a pop-up to select the **parent location** in which the copied location will reside.

The location (child) created by the Copy action will be added to the selected location (parent) as a sublocation. If no parent is selected, the copied location will not have a parent. In addition, a new unique barcode will be assigned to the new location. The user can change the name of the new location by typing a new name into the "New Name" field. All the other data (Type, Short Name, etc.) will be the same as the original location.

ⓘ All Sublocations are copied as well when a location is copied.

## 3.10. Contents

Once the desired Content types have been created and configured, you can start managing your samples using the Content module found under the Routine menu.

The following Role Access rights can be configured for use in this module:

**Table 3.11. Required Permissions for Content Module**

Access rights Category	Access rights Name	Purpose
Routine: Content	Content	When enabled, the role user has access to the Content module in which they can view, filter, and update content depending on their other permissions.

**Table 3.12. Optional Permissions for Content Module**

Access rights Category	Access rights Name	Purpose
Routine: Content	Filter By Location	When enabled, allows the role user to use the quick filter in the left panel of the Content Module by the location they reside in (if there is a location). Content that has no location (is positionless) can still be seen by clearing the filter. If disabled, this filter is not present, but locations can still be filtered using Views or the column filters on top of the grid.
Routine: Content	Filter By Project	When enabled, allows the role user to use the quick filter in the left panel of the Content Module by their project association. If disabled, this filter is not present, but projects can still be filtered using Views or

Access rights Category	Access rights Name	Purpose
Routine: Content	Filter By Requestable	the column filters on top of the grid. When enabled, allows the role user to use the quick filter content in SLIMS by their requestable association. The filter appears in the left panel of the Content Module. If disabled, this filter is not present.
Routine: Content	Filter By Order	When enabled, allows the role user to use the quick filter in the left panel of the Content Module by a content's order association. If disabled, this filter is not present, but content can still be filtered by Order using Views or the column filters on top of the grid.
Routine: Content	Filter By Study	When enabled, allows the role user to quick filter content in the left panel of the Content Module by their study association. If disabled, this filter is not present, but content can still be filtered by Study using Views or the column filters on top of the grid.
Routine: Content	Can Create and Update Records in Content Module	Is enabled by default. When enabled, this permission makes the actions that create or update content, locations, and other entities in the content module visible and accessible to provide full module functionality. If this option is enabled, but other functionalities disable certain actions, those actions will still be disabled. If this permission is disabled, all actions that create or update content, locations, and other entities in the content module are disabled or not visible for the role user. All of the information the role user has access to is readable but not editable. Other functionalities that influence actions in the

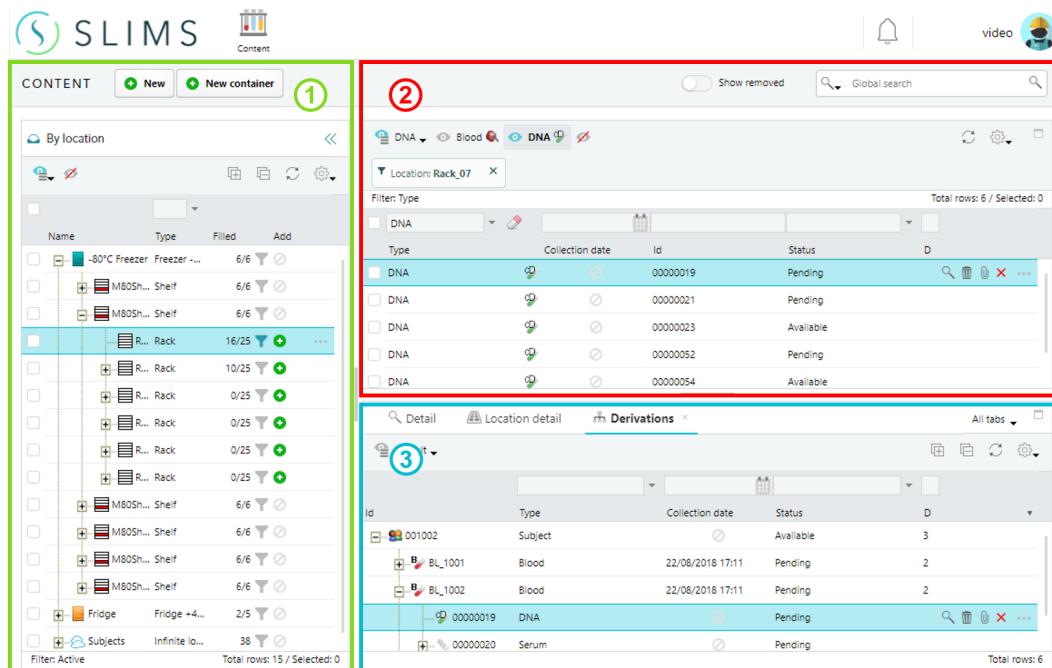
Access rights Category	Access rights Name	Purpose
		content management module will be read-only.

### 3.10.1. Content Module

The Content module can be separated into three main windows as shown in Figure 3.5.

The  button is always available and will switch the page's layout to a vertical alignment. It can be switched back to a horizontal alignment with the  button. The selection is saved in the user's layout.

**Figure 3.5. Content Module**



1. The first window contains the tree list of locations, and provides an overall filter of content by its location. The column with the location filter icon is used to apply a location filter on the content in the top right grid. To apply a location filter, click on the gray icon  and it will turn blue . This will also add a filter indicator over the grid header in the right panel that can be clicked to clear the filter . From the locations list, the filter can be cleared by clicking the blue icon (which turns it back to gray). When a filter is applied, the top right grid will filter down to the Content stored in the applied Location. If no filter is applied (all location filter icons are gray), the grid will show all Contents the user has access to.

Using the Locations tab in the left panel, the "Available positions" column shows at a glance how many open positions are available at a location. This will be displayed in a numerical

0/0 format, where the first number is the number of open positions, and the second is the total positions, filled or unfilled. The "Edit Highlighting" tool can be used to make it even more obvious if a location is running out of positions. For example, the Available Positions column could be highlighted different colors based on whether the positions are full or empty.

This window can also be used to filter contents overall by their associated Projects, Orders, Studies, or Requestables (if those functionalities are enabled). Those filters can be used instead of, or in parallel with locations.

The Active filter in this panel can be used in three states: Filled, Checkmarked, Empty. If the filter is Filled (default state), then all locations are shown, whether they are active or inactive. If the filter is Checkmarked (click once to place check in filter), then only active locations are shown. If the filter is empty, (click again to empty the filter), then only inactive locations are shown. The filter does not proceed to the sublocation level, so an active sublocation of an inactive location will be displayed when the filter is Checkmarked, and vice versa when the filter is Empty.

Orders can be selected in the order module by clicking on the icon *"select in order section."*

△ When a user has no rights to filter content by location, order, project, study, or by requestable, the left panel is removed entirely (the content grid becomes full-screen) and the content grid automatically loads all of the content the user has access to.

2. The second window lists all Contents if no filters are applied in the left panel. On top of this list quick links can be found to add new content  **New content**, to execute workflows on contents , to filter or to add views to the content list  (see Section 2.4 ), and other actions are available from the menu  (see Section 2.6 ).

Some actions are available when selecting multiple content(s), and others are directly available by right-clicking on a content. This action is available by right-clicking a content:

- **Content link:** This action will show a pop-up window with a URL generated for the selected content item.
3. The third window provides information about the selected content organized by the type of task. Content information is categorized in the different tabs ( Figure 3.6 ) that are described below:

**Figure 3.6. Content Description**

- **Detail:** This tab contains all of the content details including the standard and custom fields added to collect content data.
- **Location Detail:** This is a two-dimensional graphical representation of the Location where the Content is stored (Figure 3.7). The Location Detail tab shows details of the location where the content selected in the Contents grid is located. If no content is selected, it shows the details of the selected location from the location tree on the left window. When hovering over the location name in the grid, the location path is shown. The "Show:" drop-down menu allows the user to select up to two fields to display for each content in the location. If more than two fields are selected, a warning message appears, telling the user to only select one or two fields. This helps the tab retain the most useful and efficient display.

Locations that have no position or are too big to be displayed will still show the location name and path when the user hovers over the location icon in the tab.

**Figure 3.7. Location Detail**

In the top-right of the visualization, an export to Excel icon is present. It enables to export the location detail to Excel using any Excel template that has been defined in the location types module. The discriminator field is the field that is used to decide which color the cell should have.

By holding the CTRL (Command on a MAC) key and clicking more than one filled position, multiple samples in the Location Detail can be selected, which allows the Selection menu

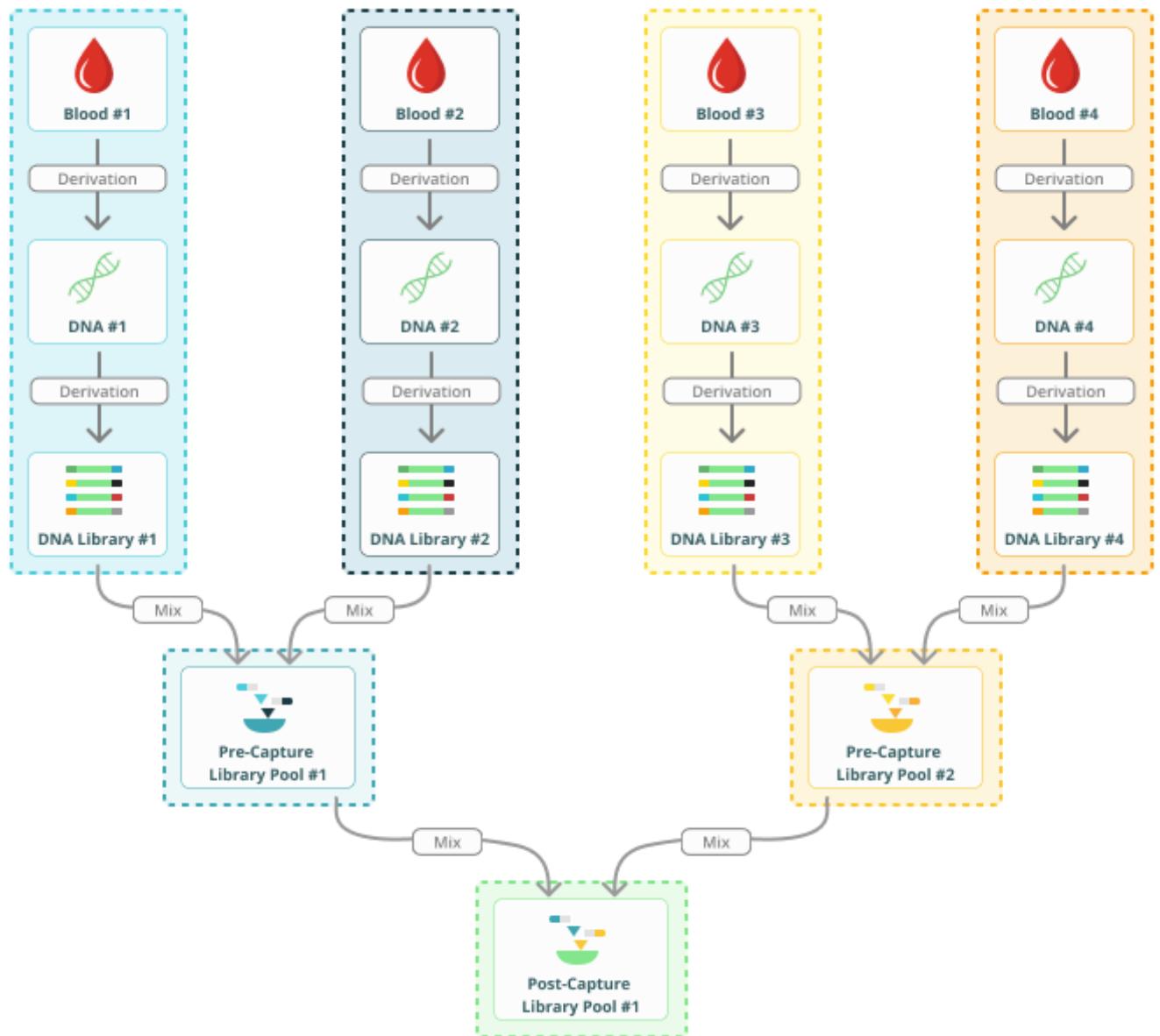
to be accessed. Users can also right-click on a filled position to access a similar menu with a list of actions that can be taken on the sample.

- The "*Select in Content List*" action selects the chosen sample from the Location Detail in the content grid above.
- The "*Show Derivations*" action in the "Show" menu produces a pop-up that contains any derivations that resulted from the selected sample.
- The "*Select samples in same location in content list*" action filters the samples in the content grid for those that are stored in the same location as the one selected in the location detail grid.
- Any available *Macro* actions that can be performed on the selected samples are also available in the right-click menu.

Users can right-click on an unfilled position to access a menu with a list of actions that can be taken on the sample.

- The "*Block*" action can be used from the right-click menu for one selected empty position or selection menu for multiple empty positions to prevent content from filling that position. Block can be used to hold a position for later, or from being used indefinitely.
- The "*Unblock*" action is used from the right-click or selection menu as well, and removes the block from the "*Block*" action.
- The "*Create Content at this Position*" action will open a Create Content form so the user can create content directly on that one position. This action is only available from the right-click menu on an empty position, so is used to create content one at a time. In the form, the location and position are pre-filled but can be adjusted if needed.
- **Derivation:** Shows the derivation tree of the selected content that is tracked when the content is derived, aliquoted, diluted, or mixed. You can see the parents, derivations, and mixes in which the selected content or its derivations/parents are included. If the selected content was an ingredient in a mix, it will be labeled "Ingredient." The mix in which the content was used will be seen derived from the ingredient, but the other ingredients used in the mix will not be seen here. If the selected content was the result of a mix, or one of its derivations is a mix, it will be labeled "Mix." The ingredients of the mix can be viewed in a pop-up if you click on the Mix label.

The figures below illustrate how this visualization is useful in an NGS situation. This shows four blood contents, each derived into a DNA content, with each DNA content derived into a DNA library. The DNA libraries are first pooled two-by-two in a pre-capture pool, then both pre-capture pools are pooled together in a post-capture pool. The second figure shows what the user sees in the Derivation tree when they select one of the DNA content.

**Figure 3.8. Illustration of NGS Mix Flow**

**Figure 3.9. Derivation Tree with a DNA Content Selected in Content Grid**

Id	Type	D	I	M
Blood #1	Blood	1		
DNA #1	DNA	1		
DNA Library #1	DNA Library	1		
Pre-Capture Library Pool #1	Library Pool	2	1	
Post-Capture Library Pool # 1	Library Pool	2		

- **Print Tab Option:** The Derivation tab contents can be printed by clicking the gear icon to access the action menu. The derivation sample list should be expanded or collapsed to accommodate the desired display, and then the print icon can be clicked. The list is printed as it is displayed on the screen.

The view in the derivation tab inherits the view from the Content grid above. If the view needs to be adjusted, users can make the desired view in the content grid, refresh SLIMS, and then the same view will be applied to the derivation tab.

- **Figure 3.10. Print Derivation Tab**

Print																																																																																																																																								
	<table border="1"> <thead> <tr> <th data-bbox="262 1178 290 1199">Id</th> <th data-bbox="649 1178 763 1199">Collection date</th> <th data-bbox="784 1178 829 1199">Type</th> <th data-bbox="878 1178 931 1199">Status</th> <th data-bbox="953 1178 1006 1199">Quantity</th> <th data-bbox="1024 1178 1054 1199">Unit</th> <th data-bbox="1073 1178 1155 1199">rounded bmi</th> <th data-bbox="1176 1178 1196 1199">D</th> <th data-bbox="1274 1178 1535 1199">Product (filtering without version)</th> </tr> </thead> <tbody> <tr> <td data-bbox="262 1205 404 1227">00000001</td><td data-bbox="654 1205 768 1227">26/01/2018 09:59</td><td data-bbox="789 1205 833 1227">Sample</td><td data-bbox="855 1205 909 1227">Pending</td><td data-bbox="930 1205 959 1227"></td><td data-bbox="979 1205 1008 1227"></td><td data-bbox="1028 1205 1057 1227"></td><td data-bbox="1077 1205 1096 1227">2</td><td data-bbox="1116 1205 1135 1227"></td></tr> <tr> <td data-bbox="262 1233 453 1254">BL00000020</td><td data-bbox="654 1233 768 1254">26/01/2018 09:59</td><td data-bbox="789 1233 833 1254">Blood</td><td data-bbox="855 1233 909 1254">Pending</td><td data-bbox="930 1233 959 1254"></td><td data-bbox="979 1233 1008 1254"></td><td data-bbox="1028 1233 1057 1254"></td><td data-bbox="1077 1233 1096 1254">6</td><td data-bbox="1116 1233 1135 1254"></td></tr> <tr> <td data-bbox="262 1269 474 1290">00000002</td><td data-bbox="654 1269 768 1290">26/01/2018 10:04</td><td data-bbox="789 1269 866 1290">DNA Library</td><td data-bbox="855 1269 909 1290">Pending</td><td data-bbox="930 1269 959 1290"></td><td data-bbox="979 1269 1008 1290"></td><td data-bbox="1028 1269 1057 1290"></td><td data-bbox="1077 1269 1096 1290">5</td><td data-bbox="1116 1269 1135 1290"></td></tr> <tr> <td data-bbox="262 1305 540 1326">DNA00000004</td><td data-bbox="654 1305 768 1326">26/01/2018 10:07</td><td data-bbox="789 1305 833 1326">DNA</td><td data-bbox="855 1305 909 1326">Pending</td><td data-bbox="930 1305 959 1326"></td><td data-bbox="979 1305 1008 1326"></td><td data-bbox="1028 1305 1057 1326"></td><td data-bbox="1077 1305 1096 1326"></td><td data-bbox="1116 1305 1135 1326"></td></tr> <tr> <td data-bbox="262 1341 540 1362">DNA00000005</td><td data-bbox="654 1341 768 1362">26/01/2018 10:07</td><td data-bbox="789 1341 833 1362">DNA</td><td data-bbox="855 1341 909 1362">Pending</td><td data-bbox="930 1341 959 1362"></td><td data-bbox="979 1341 1008 1362"></td><td data-bbox="1028 1341 1057 1362"></td><td data-bbox="1077 1341 1096 1362"></td><td data-bbox="1116 1341 1135 1362"></td></tr> <tr> <td data-bbox="262 1377 540 1398">DNA00000006</td><td data-bbox="654 1377 768 1398">26/01/2018 10:07</td><td data-bbox="789 1377 833 1398">DNA</td><td data-bbox="855 1377 909 1398">Pending</td><td data-bbox="930 1377 959 1398"></td><td data-bbox="979 1377 1008 1398"></td><td data-bbox="1028 1377 1057 1398"></td><td data-bbox="1077 1377 1096 1398"></td><td data-bbox="1116 1377 1135 1398"></td></tr> <tr> <td data-bbox="262 1412 540 1434">RNA00000000</td><td data-bbox="654 1412 768 1434">26/01/2018 10:58</td><td data-bbox="789 1412 833 1434">RNA</td><td data-bbox="855 1412 909 1434">Pending</td><td data-bbox="930 1412 959 1434"></td><td data-bbox="979 1412 1008 1434"></td><td data-bbox="1028 1412 1057 1434"></td><td data-bbox="1077 1412 1096 1434"></td><td data-bbox="1116 1412 1135 1434"></td></tr> <tr> <td data-bbox="262 1448 540 1469">RNA00000001</td><td data-bbox="654 1448 768 1469">26/01/2018 10:58</td><td data-bbox="789 1448 833 1469">RNA</td><td data-bbox="855 1448 909 1469">Pending</td><td data-bbox="930 1448 959 1469"></td><td data-bbox="979 1448 1008 1469"></td><td data-bbox="1028 1448 1057 1469"></td><td data-bbox="1077 1448 1096 1469"></td><td data-bbox="1116 1448 1135 1469"></td></tr> <tr> <td data-bbox="262 1484 540 1505">00000003</td><td data-bbox="654 1484 768 1505">26/01/2018 10:04</td><td data-bbox="789 1484 866 1505">DNA Library</td><td data-bbox="855 1484 909 1505">Pending</td><td data-bbox="930 1484 959 1505"></td><td data-bbox="979 1484 1008 1505"></td><td data-bbox="1028 1484 1057 1505"></td><td data-bbox="1077 1484 1096 1505"></td><td data-bbox="1116 1484 1135 1505"></td></tr> <tr> <td data-bbox="262 1520 540 1541">00000004</td><td data-bbox="654 1520 768 1541">26/01/2018 10:05</td><td data-bbox="789 1520 866 1541">DNA Library</td><td data-bbox="855 1520 909 1541">Pending</td><td data-bbox="930 1520 959 1541"></td><td data-bbox="979 1520 1008 1541"></td><td data-bbox="1028 1520 1057 1541"></td><td data-bbox="1077 1520 1096 1541"></td><td data-bbox="1116 1520 1135 1541"></td></tr> <tr> <td data-bbox="262 1556 540 1577">00000005</td><td data-bbox="654 1556 768 1577">26/01/2018 10:05</td><td data-bbox="789 1556 866 1577">DNA Library</td><td data-bbox="855 1556 909 1577">Pending</td><td data-bbox="930 1556 959 1577"></td><td data-bbox="979 1556 1008 1577"></td><td data-bbox="1028 1556 1057 1577"></td><td data-bbox="1077 1556 1096 1577"></td><td data-bbox="1116 1556 1135 1577"></td></tr> <tr> <td data-bbox="262 1592 540 1613">00000006</td><td data-bbox="654 1592 768 1613">26/01/2018 10:58</td><td data-bbox="789 1592 833 1613">Plasma</td><td data-bbox="855 1592 909 1613">Pending</td><td data-bbox="930 1592 959 1613"></td><td data-bbox="979 1592 1008 1613"></td><td data-bbox="1028 1592 1057 1613"></td><td data-bbox="1077 1592 1096 1613"></td><td data-bbox="1116 1592 1135 1613"></td></tr> <tr> <td data-bbox="262 1628 540 1649">00000007</td><td data-bbox="654 1628 768 1649">26/01/2018 10:59</td><td data-bbox="789 1628 833 1649">Plasma</td><td data-bbox="855 1628 909 1649">Pending</td><td data-bbox="930 1628 959 1649"></td><td data-bbox="979 1628 1008 1649"></td><td data-bbox="1028 1628 1057 1649"></td><td data-bbox="1077 1628 1096 1649"></td><td data-bbox="1116 1628 1135 1649"></td></tr> <tr> <td data-bbox="262 1664 540 1685">BL00000021</td><td data-bbox="654 1664 768 1685">26/01/2018 10:00</td><td data-bbox="789 1664 833 1685">Blood</td><td data-bbox="855 1664 909 1685">Pending</td><td data-bbox="930 1664 959 1685"></td><td data-bbox="979 1664 1008 1685"></td><td data-bbox="1028 1664 1057 1685"></td><td data-bbox="1077 1664 1096 1685"></td><td data-bbox="1116 1664 1135 1685"></td></tr> </tbody> </table>	Id	Collection date	Type	Status	Quantity	Unit	rounded bmi	D	Product (filtering without version)	00000001	26/01/2018 09:59	Sample	Pending				2		BL00000020	26/01/2018 09:59	Blood	Pending				6		00000002	26/01/2018 10:04	DNA Library	Pending				5		DNA00000004	26/01/2018 10:07	DNA	Pending						DNA00000005	26/01/2018 10:07	DNA	Pending						DNA00000006	26/01/2018 10:07	DNA	Pending						RNA00000000	26/01/2018 10:58	RNA	Pending						RNA00000001	26/01/2018 10:58	RNA	Pending						00000003	26/01/2018 10:04	DNA Library	Pending						00000004	26/01/2018 10:05	DNA Library	Pending						00000005	26/01/2018 10:05	DNA Library	Pending						00000006	26/01/2018 10:58	Plasma	Pending						00000007	26/01/2018 10:59	Plasma	Pending						BL00000021	26/01/2018 10:00	Blood	Pending					
Id	Collection date	Type	Status	Quantity	Unit	rounded bmi	D	Product (filtering without version)																																																																																																																																
00000001	26/01/2018 09:59	Sample	Pending				2																																																																																																																																	
BL00000020	26/01/2018 09:59	Blood	Pending				6																																																																																																																																	
00000002	26/01/2018 10:04	DNA Library	Pending				5																																																																																																																																	
DNA00000004	26/01/2018 10:07	DNA	Pending																																																																																																																																					
DNA00000005	26/01/2018 10:07	DNA	Pending																																																																																																																																					
DNA00000006	26/01/2018 10:07	DNA	Pending																																																																																																																																					
RNA00000000	26/01/2018 10:58	RNA	Pending																																																																																																																																					
RNA00000001	26/01/2018 10:58	RNA	Pending																																																																																																																																					
00000003	26/01/2018 10:04	DNA Library	Pending																																																																																																																																					
00000004	26/01/2018 10:05	DNA Library	Pending																																																																																																																																					
00000005	26/01/2018 10:05	DNA Library	Pending																																																																																																																																					
00000006	26/01/2018 10:58	Plasma	Pending																																																																																																																																					
00000007	26/01/2018 10:59	Plasma	Pending																																																																																																																																					
BL00000021	26/01/2018 10:00	Blood	Pending																																																																																																																																					

- **Results:** Shows the results of the selected Content that have been obtained through tests in the ELN or in orders. Right-clicking on a result provides access to "view history," details on who added or modified the results, and details about when the modifications were

made. In the right-click menu, the "Select in Content List" action selects the chosen sample from the Results tab in the content grid above. If the role functionality "Routine->Content->Create results" is enabled, it is possible to add a new result for the selected content by clicking on the "New" button.

- **Reagents:** Provides a display with details about reagents: content linked in a Link Reagent step in a Protocol that typically belong to a Reagent content type. The Reagent tab has two modes to choose from which display the details inversely from each other. With "Reagents Used with this Content" mode, you see the reagents that were used with the content (the contents linked through a Link Reagent step in a protocol where the selected content was present).

The "Contents Used with this Reagent" mode shows the contents that were used with the reagent that is selected in the main grid (the contents that were present in the step when the selected content was linked as a reagent). There are no further options with this mode selected.

In the right-click menu, the "Select in Content List" action selects the chosen sample from the Reagents tab in the content grid above.

- **ELN:** Shows in which Project/Experiment/Run this content was or is involved.
- **History:** Provides a display with details about all of the modifications made on this content, including the user who performed these changes and the date.
- **Location History:** Provides details about the location changes made on the selected content, including the user who performed these changes and the date.
- **Printer Jobs:** Indicates which label and printer were used for the impressions.
- **Orders:** Shows the orders that the selected content is linked to. Custom fields created for the order content table can be displayed in the "Order Content Data" section if they are not hidden and are active. They will be displayed in the view by default if they are configured with "Show in list grid" enabled, but can be added manually if "Show in list grid" is disabled. Order content custom fields are read-only in the Content module.
- **Mix:** Displays all of the mixing relations of the content such as what it was used to create, what ingredients were used with it, or what content was mixed together to create it. In the right-click menu, the "Select in Content List" action selects the chosen sample from the Mix tab in the content grid above.
- **Workflows:** Shows all Protocol runs created in the *Workflows Module* using the selected content. The right window inside this tab shows all contents used in the selected protocol run. The *Open Run* button behavior is as follows:
  - *Run Finished:* Opens a new pop-up window with the run details in read-only mode.
  - *Run in Progress:* Opens the run in the *Workflows Module*.
- **Studies:** This tab shows all of the Studies in which the selected content was enrolled. The checkbox "show studies from derivatives" additionally shows the studies that the sample's derivatives were enrolled in. In the right-click menu, the "Select in Content List" action selects the chosen sample from the Studies tab in the content grid above.

- **Events:** This tab shows all events that occurred that pertain to the sample, including study enrollment. Case Report Forms can be added with the green plus button grouped by template. Then, by right-clicking on the header, any desired columns from the CRF can be added to format the display of the table.

Users can select one or more events at a time to perform actions from the selection menu, or perform actions on each row individually. Users that have permission to permanently remove records will see the option to "Permanently Remove" content events.

In the right-click menu, the "Select in Content List" action selects the chosen sample from the Events tab in the content grid above.

- The "All Tabs" icon  provides access to previous tabs in case some were removed from the window. Tabs can be favorited by clicking the yellow star so they appear in the bottom panel, and unfavored to hide them again.

### 3.10.1.1. Filter: Also Show "X" Of

A filter can be enabled on several Content module tabs which will display the related records of a selected content and its family's records, depending on the selection. The context of the records depend on the tab the user is viewing. The filter is saved per tab so the user's selection can be different across multiple tabs. These tabs have the filter available:

- Results
- Variants
- Reagents
- ELN
- Workflow
- Orders
- Events
- Studies

The filter allows you to see useful relationships like the metadata of the mixes, ingredients, parents, and derivations of a selected content. Or the QC Results of the DNA Libraries that are ingredients of the pool that is currently selected. Or even the orders associated to the parents or derivations of the selected content. Put more generally, the filter shows the metadata associated to the contents influenced by the selected content, the metadata associated to the contents that influence the selected content, and both options together.

When the filter is enabled, the specific options are "Also show X of:"

- **Contents affected by selected content:** The contextual records of contents that are influenced by the selected content are listed. Examples of contents affected by the selected content are derivations of the content and mixes in which the content is an ingredient.
- **Contents affecting selected content:** The contextual records of contents that influence the selected content are listed. Examples of contents affecting the selected content are parents

of the content, ingredients of the content if the content is a mix, and ingredients of the parents of the content if one of the parents is a mix.

- **Both contents affecting and affected by selected content:** The contextual records of contents that influence and are influenced by the selected content are listed.

### 3.10.2. Create Contents

To create new content, the button  'New content' has to be used that will generate a pop-up window where the user enters the content data. Multiple contents can be added rapidly using the same button.

A content is defined with default and custom fields. The following Default fields are included for all content types but can be hidden and customized according to the team's needs:

- **Icon, Id, and Barcode**

ⓘ If adding multiple contents at once that have a content type with "barcode not generated by SLIMS," there will be a barcode field to fill out for each content.

- **Derivation count:** Display the number of derivation proper to the content.
- **Category:** The user can select a category to filter the content types in the next dropdown. If the user skips this and chooses the Type first, the category field is updated to match the category that the selected content type is part of.
- **Type:** The user must select the content type of the content. This list can be filtered by the category selection above. Content types are defined as explained in the *last section*.
- **Collection date:** Date type custom field used to register the real collection date of the content. By default, this value is set as the creation date.
- **Status:** Register the status of the content. These six default statuses come with the system: Pending, Available, Labeled, Approved, Removed and Cancelled. Statuses can be added, removed, and modified, so could be different than the six listed here.
- **Amount:** (for aliquotable/dilutable samples only): The value for the Amount of material field that is used to track the remaining Amount of the content.
- **User:** The user will have access to the content and the possibility to modify or delete it. The content will be accessible only by that user if the group is not filled in (personal item).
- **Group:** Users belonging to the group will have access to the content and can edit it. This is possible only if a group member have at least group access.  
 ⓘ Inactive users and groups are not available in the selection list.

### 3.10.3. Create Kit

Kits are a content category that define certain things about the components that are in the kit. They are a content themselves, and a way to organize all the parts of a kit that would be received in the lab. When a kit content is added in SLIMS, the components (contents defined as

belonging to the kit) are derived from the kit content automatically. Depending on the configuration of the kit details and the individual content types, information about the kit's components can be added automatically or manually.

To create kits, use the  button which is only available to role users that have the "Can create kits" permission enabled.

The New kit pop-up window provides the fields that were configured in the kit details. For example:

- **Type:** A dropdown with a selection of the types of kits that were configured. Choosing one determines the options that are shown on the rest of the form.
- **Additional Custom Fields:** Other custom fields can be added to this form from the kit configuration.
- **Kit components:** A read-only list of the components that are in the kit (the Content types defined as belonging to the kit) is shown with their individual amounts. If a kit component does not generate its barcode automatically, then a required text field is shown so the user can manually fill in the barcode. Otherwise, there is no field and SLIMS will create the barcode for the content.

When a kit content is created using the "New Kit" button, it is immediately derived to create a content for each kit component defined in the kit type. If configured, each component will have an individual amount pre-filled based on the information entered when configuring the kit type.

Additionally, you can create a kit using the "New content" button to create a content of the Kit type, but without any Kit components. This creates an empty kit content from which you can manually derive reagents, which is similar to creating an empty kit bag.

Both kits and reagents can be used in Link Reagents protocol steps. There, too, the kit Amount of Material and the kit component Amounts can be tracked.

### 3.10.4. Create container

Containers are a type of content that can contain other content. Various types of container can be defined using the *Container types* module. Then, containers can be added in the Content module. The main difference between containers and content is that containers start out as an empty Container Type and then take on the Content type of the content that is added to them. Containers can then be returned to the empty Container Type when the content has been used. This allows the container to be retained in the system and reused for other content if required.

Containers can be not used, optionally used, or required for various content types depending on the configuration of the content types. Containers can be added first using the "New Container" action in the Content module, or content can be added with the "New Content" action that uses a container. If the content type requires a container type, then the content cannot be created without choosing the container type as the field is required.

Containers can have a location, a barcode, an identifier, an expiration date, and be associated with lots. When a content is added to them, the other fields related to the content can be filled in.

This module requires the following Role Access rights to work:

**Table 3.13. Containers Permissions**

Access rights Category	Access rights Name	Purpose
Routine: Content	Can Create Containers	Allows the role user to create containers in the Content module once one or more Container types exist in SLIMS.

When at least one Container Type exists in SLIMS and a user has the above access permission, containers can be added in the Content module. Instead of using the "New" button the user would traditionally use to add content, the user should click the "New Container" button. Multiple containers can be added at one time.

Clicking "New Container" opens a window to define the container with default and custom fields. The following Default Fields are included for all Container types but can be hidden and customized according to the team's needs:

- **Icon, Id, and Barcode**
- **Amount:** One or more containers can be added at a time by entering the desired amount of containers here.
  - ① If adding multiple containers at once that have a Container Type with "barcode not generated by SLIMS," there will be a barcode field to fill out for each container.
- **Location:** If the location has already been selected in the location tree when the user clicked "New Container," the desired location should already be filled. If not, this provides a dropdown list of all available locations and a number to indicate how many empty positions remain.
- **Fill Method:** See the explanation for content, as the fill method for containers is identical.
- **Container Type:** This dropdown contains all of the available created Container types to define the desired container. After selecting a Container Type, fields that are not relevant to the selected type will hide, and ones that are relevant to that container type or category will show.
- **Status:** This field registers the status of the container. These six default statuses come with the system: Pending, Available, Labeled, Approved, Removed, and Cancelled. Statuses can be added, removed, and modified, so could be different than the six listed here.
- **User:** The user will have access to the container and the ability to modify or delete it. The content will be accessible only by that user (will be a personal item) if the group is not filled in.
  - ① Inactive users and groups are not available in the selection list.
- **Access Rights:** Determines whether the container can be accessed by anyone within the set group, only the person that created the container, or anyone who can access the SLIMS instance. Users belonging to the group will have access to the container and can edit it. This is possible only if a group member have at least group access.

Once a container is created, it appears like other content in the Content Grid in the right panel. A container can either be filled or empty. This is designated by selecting the new "Type" for the container in the Details tab below the grid. To add content to the container:

1. Select the desired container in the grid.
2. Navigate to the Details tab below the grid for the selected container.
3. The container has a "Type" and a "Container Type." Change just the "Type" field to the desired Content type. The relevant content fields will appear so that they can be collected, and the Category will update depending on the content type selected (but will display as read-only). Adjust those to suit the content and then save the updates.
4. The container takes on the content's type (thus the content is inside the container), has the relevant details, and has the same barcode.

The content within a container can be used, and if needed, the container can be made 'empty' again.

1. Select the container in the Content Grid.
2. In the Details tab below the grid, change the "Type" field (not the "Container Type" field) back to the Container Type instead of the content type. The fields relevant to the content will hide.
3. Save the updates. The container is now considered 'empty' again.

### 3.10.5. Fill Locations

One or multiple empty locations can be filled with contents in one action.

The "Fill" action is available in the location tree in content management by:

- Right-clicking on a location and using the "Fill" action;
- Selecting multiple locations and using the "Fill" action from the selection menu.

A filling method can be chosen (Column wise or Row wise).

⚠ Note that only empty locations can be filled with this method.

### 3.10.6. Edit Contents

Once content has been added, SLIMS provides three different ways to edit content records as follows:

- **Edit Mode:** Double-click on a sample directly in the content grid to enable edit mode. This causes all of the fields in the content row to become editable.
- **Details:** Access the details tab in the bottom panel, or right-click on a content record and select 'Open Details.'
- **Multiple Edit:** Check the boxes for multiple content records, open the 'Selection' menu, and then click 'Edit.' Fields that show up in the edit pop-up are all fields available in the detail tab of the content record and can be edited.

SLIMS allows a maximum of 1000 records to be recorded at the same time. Multiple edit operations have to be done if more than 1000 records need to be edited.

ⓘ After using the multiple edit and confirm, the request is sent to the server where it is processed. Once the request is sent, the action cannot be stopped and will be executed on all selected records.

Any module that uses history tables, like the content module, implements optimistic locking. This means that SLIMS performs a check to detect concurrent changes so that users are aware of whether their updates will overwrite each others'. SLIMS checks the modifiedOn field to see if it matches the actual modifiedOn value in the database, and if it does not, then the update fails with a message to the user. If the field doesn't match, it indicates that the user is updating a record based on values that do not contain the most recent changes. The message the end user gets indicates that 'a concurrent change occurred' and that he should refresh in order to try updating again.

It is recommended to use optimistic locking to ensure users are always on the most current version of their data, but this feature can be turned off by enabling the lab setting "Suppress concurrent modification errors" in the *Lab Settings Module*.

### 3.10.7. Copy Contents

Contents can be created as a copy of one or multiple existing contents. The copy action is available:

- By right-clicking on original content
- By selecting multiple contents and using the selection menu
- Using a macro step

When the copy step is initiated, users choose between the original value or a new value for:

- Location
- Content Fields
- The number of copies

If contents are copied in the Content tab of a Name, the created contents are also linked to the same Name.

### 3.10.8. Remove Contents

SLIMS provides two remove possibilities for contents. The first option adds the "Removed" status to the Content but does not delete the content. To achieve this, the user clicks the "Remove" button  in the Content's row. When a Content is removed, its status changes from 'Available' to 'Removed' and its location becomes Available. (The Removed status name can be changed in the Statuses module, though the name only changes in the related grid column. Hovering over the remove icon still shows the tooltip "Mark Content as 'Removed.'") This is not an irreversible action because it does not permanently delete the content. The removed content can be accessed by enabling the 'Show Removed' switch  Show removed, and recovered to an Available status with the Unremove button  if desired. This option is available for Content data only.

The second option is to remove a Content record permanently. By using this option, the content data will be completely lost and can no longer be retrieved. See the section "Remove Permanently" for more details.

Whenever a content has derivations and the user has the functionality 'Can permanently remove derivations' or 'Can remove derivations' enabled, it is possible to remove the derivations while removing the parent content. A pop-up dialog box asks the user whether the derivation of the selected content should be removed as well or not.

When executing an action or a macro, a progress bar is shown that displays how many contents were handled by the action.

### 3.10.9. Move Contents

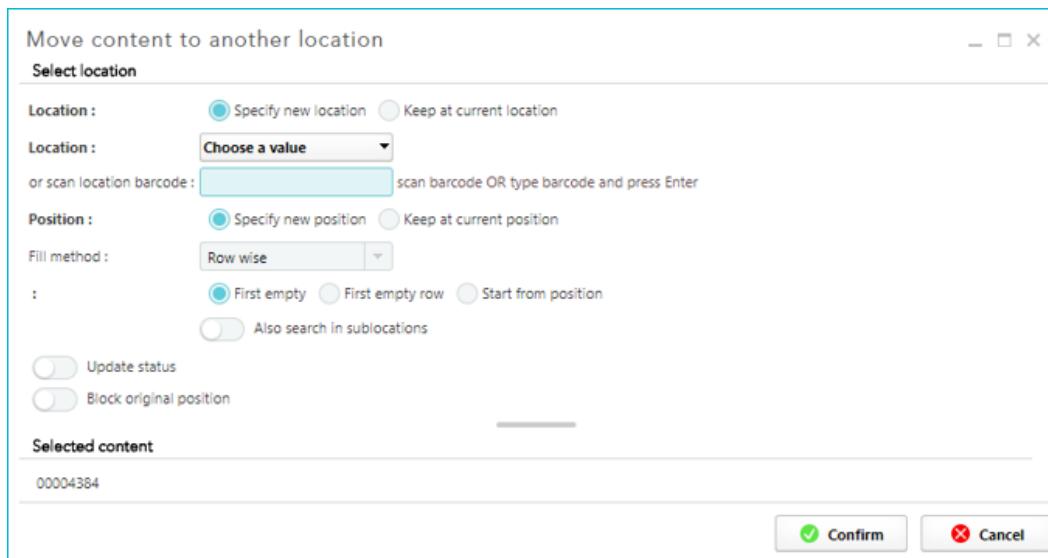
Contents can be moved:

- By dragging and dropping samples to a different position in the *Location Detail* bottom tab;
- By right-clicking on a *single Content* and selecting the Move button;
- By selecting one or more Contents and clicking on the "Move" button in the Selection menu.

⚠ Note that the right-click refers only to a single Content (the selected one), even if more Contents are selected using the checkboxes.

The following *pop-up window* allows to select the destination.

**Figure 3.11. Move Content pop-up window**



- **Location:** Selected Content(s) can be moved to a new location, or moved to a different position in the same location. When specifying a new location, the cascading location list shows a numerical value next to each of the locations. This is the amount of available positions for locations that accept content and the amount of available positions in all sublocations for locations that cannot accept content.

The location can be selected using the cascading location list or by typing (or scanning) the location barcode in the dedicated field and pressing Enter.

If a location exists corresponding to the scanned barcode, it will be shown in the location list in the left panel.

If the specified location doesn't have enough available positions, an error will be generated and the Contents won't be moved.

- **Position:** The exact position in the intended location can be:

- **Specify New Location:**

- *Specify New Position:* By selecting the radio button "Start from position", the row and column that the content is moved into can be specified.
- *Keep at Current Position:* Means that the same row and column index will be kept when the content moves to the destination location.

- **Unspecified:** The first empty position in the location will be filled including the gaps. The following criteria can be specified for defining what the first empty position is.

- *Fill Method:* Choose whether content is filled at the first empty column position, or the first empty row position.
- *First Empty:* The first empty position that is found, column or row wise, depending on the choice made in the Fill method.
- *First Empty Row (column):* The first empty position that is found in the first empty row (column).
  - *Preserve column (row) positions:* A checkbox that becomes available when selecting 'First empty row' to be used in order to preserve the current column (row) position of the content.
- *Start From Position:* Two boxes appear for specifying the row number and/or column number to move the content.

⚠ If a destination position is not empty, a warning is generated and the Content won't be moved.

- A content's status can be changed to a new selected status during the move action if desired.
- The original location can be freed or blocked depending on the **Block original position** option.

The move action appears in the History of the Content.

### 3.10.10. Derive Contents

A derivation is the action of creating a new content by taking it from an existing content. This could result in content that is the same type as the original, or a different type from the original. The relationship between derivations and content they were derived from are tracked in SLIMS.

As an example, a blood content can be derived from a mouse and further analyzed. The mouse is the parent content, and the blood is its derivation.

The following Role Access rights can be configured for use in this module:

**Table 3.14. Optional Derivation Permissions**

Access rights Category	Access rights Name	Purpose
Routine: Content	Can Remove Derivations	When enabled, allows the user to use the "remove" action on derived content. This marks the derived content as removed and does not permanently remove the derived content.
Routine: Content	Can Permanently Remove Derivations	When enabled, allows the role user to use the "permanently remove" action on derived content. It is recommended to only provide administrators with this permission as permanently removed content cannot be retrieved.

The new obtained content (derived content) does not have to be of same content type or category, nor have exactly the same properties. See further down for more details. The parent content is not affected by this derivation, except that its derivation count gets incremented, and its related derivatives can be viewed.

Depending on the parent content's Type definition (see [Content types section](#) ), the derivation can be restricted to particular content types, or the content type might only be able to be created by derivation.

Contents can be derived using the derive action  that is available in these areas:

- Right-click on the original content to see it in the row's action menu.
- Select multiple contents (of same content type) to see it in the Selection menu.
- Use the Derive Icon on the content record.
- Use a derive action in a macro step.

A derivation has the same parameters as performing a content creation with the corresponding content type. However, the parent content's values are already present on the content creation pop-up so that they can be quickly reused (useful for derivations of the same content type). First the number of derivations should be specified (**Number of derivations per content**). If more than one identical derivation has to be performed, the two following fields should be filled in order to dissociate the derivations.

- **Suffix mask (use # for numbers):** Suffix that is added to the content IDs which can be used to make the IDs unique.

- **Start index:** Identifies the index from which the incrementation begins in the suffix mask.

Then the location should be defined.

- **Use parent location:** Use the same location as the parent content.
- **Specify new location:** Choose a new location (in the list or by barcode scanning).
- **Use parent position:** Use the same position as the parent content. (It is not recommended to use this if using the same location as the parent content, since the parent content remains and does not empty out that position. Be careful using this option as the position may not be empty.).
- **Specify new position:** Choose a new position by specifying row and column, or leave empty so the first available position in the selected location is supplied.

Finally, the details about the derived contents should be set up, primarily by specifying whether to use the same value as the parent's value, (**Use parent value**) or by selecting a new value (**Specify new value** and adding the new value in the additional field). For some specific fields, some more options are available:

- **Suffix parent value:** Add a suffix to the parent value (for Content id).
- **Use barcode as id:** Use the barcode of derived content for its Id (If the derived content type is set to 'Use barcode as id,' then this option is set to the default value).
- **Use value expression:** This field will be evaluated whenever the field contains a Value expression.
- **Myself:** for User: To attribute the derived content to the logged-in user.

When SLIMS does not automatically generate a barcode for the derived content, meaning the derived content type has the option "*SLIMS generates a barcode for content of this type*" set to false (disabled), then a new barcode should be entered for the derived content.

### 3.10.10.1. Multiple Contents Derived at Once

Multiple contents can be derived at once by selecting multiple contents and clicking on the derive icon in the selection menu. The creation of derivations depends on both the content type of the parent content and the derived content's type. It is better not to derive contents of different content types together because these properties may differ between content types. Additionally, the creation of new IDs can be tricky, especially with the *Specify New Value* option. In case this is unavoidable, examples of the possible choices are provided below.

- **Id: Use Parent Value**

**Number of derivations per content: 2, Suffix mask (use # for numbers): child#, Start index: 1**

Initial Contents	Corresponding Derivations
Content1	<ul style="list-style-type: none"> <li>○ Content1_child1</li> <li>○ Content1_child2</li> </ul>
Content2	<ul style="list-style-type: none"> <li>○ Content2_child1</li> </ul>

Initial Contents	Corresponding Derivations
○ Content2	Content2_child2

- **ID: Specify New Value, ID prefix: derivation, ID start index: 10**

**Number of derivations per content: 2, Suffix mask (use # for numbers): child#, Start index: 1**

Initial Contents	Corresponding Derivations
○ Content1	derivation_0010_child1
○ Content2	derivation_0011_child1

- **ID: Suffix parent value, ID suffix: child**

**Number of derivations per content: 2, Suffix mask (use # for numbers): child#, Start index: 1**

Initial Contents	Corresponding Derivations
○ Content1	Content1_child_child1
○ Content2	Content2_child_child1

### 3.10.11. Aliquot Contents

An aliquot is similar to a derivation that has the same content type as its parent content. The major difference between an aliquot and a derivation is that aliquots keep track of the amount of the parent and children samples. The amount of the aliquotted contents are deducted from the amount of the parent content while aliquotting.

As an example, a blood sample of 100 microliters is transferred into new tubes that can only contain 30 microliters. To accomplish this, three aliquotes that each contain 30 microliters of the parent blood content are moved to their own tubes, and the original content has only 10 microliters remaining.

Only content that has a content type definition that allows aliquoting can be aliquoted (see *Content types section*).

Contents can be aliquoted using the aliquot action  that is available using these methods:

- Right-click on original content to access the single row's action menu
- Select multiple contents (of same content type) and use the selection menu
- Use the aliquot icon on the content record

- Use a Macro step

While aliquoting, the location has to be defined except for contents with status 'Pending' (or the status that means Pending if the status name was modified) which can be aliquoted without specifying a location or a position. The parent content must have a value for the field defined as the amount field.

- **Use parent location:** Use the same location as the parent content.
- **Specify new location:** Choose a new location in the list or by scanning the location's barcode.
- **Use parent position:** Use the same position as the parent content (It is not recommended to use this in the same location, as the parent content doesn't get removed automatically, so the position is likely not empty.).
- **Specify new position:** Choose a new position by specifying a row and column, or leave it empty to assign the first available position in the selected location.

These fields are specific to aliquots:

- **Number of aliquots:** How many aliquots should be done.
- **Suffix mask (use # for numbers):** A suffix that is added to the content IDs to help with keeping them uniquely named.
- **Start index:** Identifies the index from which the incrementation begins in the suffix mask.
- **Amount label:** The amount that is distributed into each aliquot.
- **Unit:** This is predefined by the amount specification of the parent content (the abbreviation of the unit is shown).
- **Remove original sample:** Check the box if the parent content can be removed after aliquoting (either it is used up or the rest is thrown away).

⚠ The aliquot action inherits all field values from its parent by default, except for the following fields:

1. The aliquoted volume must be entered by the user because it is not automatically defined.
2. If a Value expression is defined in the field definition by the administrator and set to be always evaluated, the field value will be the calculated Groovy expression.
3. If the aliquot content type is set to 'use barcode as ID', the resulting aliquots will have their barcode as ID. In this case, the mask and start index from the configuration will be ignored.

ⓘ The unit in an aliquot is determined by the content type that is aliquoted. When no content is available (e.g. during macro config), the unit field is not visible.

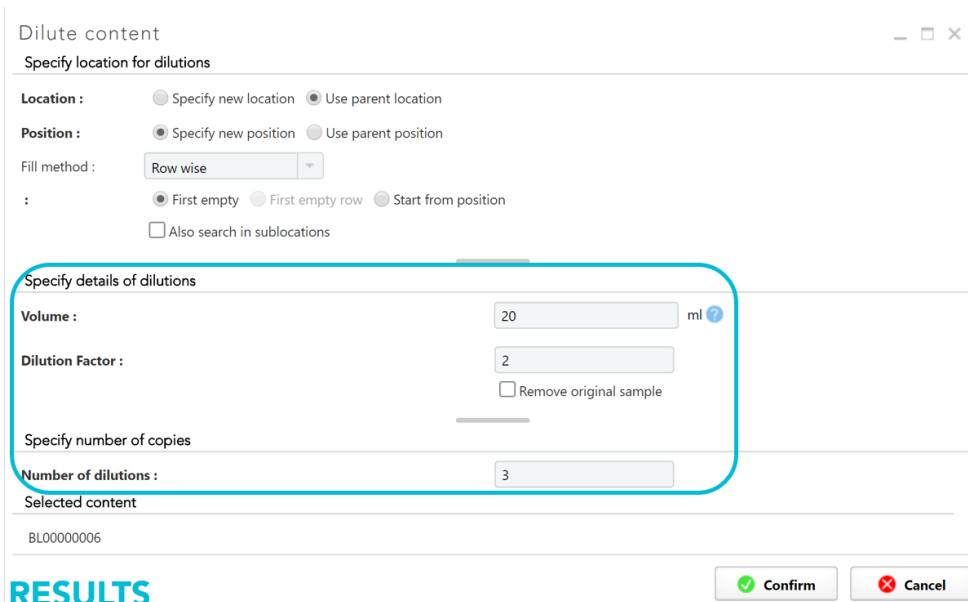
## 3.10.12. Dilute Contents

The action of dilution tracks the creation of a new sample from the dilution of an original sample (the sample that is being diluted) and a buffer (which is not recorded in SLIMS). Like the Aliquot and Track Usage actions, dilution tracks the amounts of contents.

The dilute action is triggered by clicking on the dilute icon .

For example, we dilute 20 ml (Quantity that is extracted from the parent) of a sample with initial quantity 100 ml, by a Dilution Factor of 2, 3 times (Number of dilutions=3). We get 3 (Number of dilutions) new samples with quantity 40 ml each (Dilution Factor X Quantity that is extracted from the parent). The initial sample quantity is 40 ml (initial quantity - Number of dilutions X Quantity that is extracted from the parent).

**Figure 3.12. Dilute Content Pop-up Window**



## RESULTS

Derivations						
Id	Collection date	Type	Stat...	Volume	D	
B 001007	03/05/2018 17:03	Blood	Pending	40 ml	3	
B 001007	03/05/2018 17:03	Blood	Pending	40 ml		
B 001007	03/05/2018 17:03	Blood	Pending	40 ml		
B 001007	03/05/2018 17:03	Blood	Pending	40 ml		

(Initial Quantity - # dilutions X Quantity extracted)  
(100 ml - 3 X 20 ml) = 40 ml

(Dilution factor X Quantity extracted)  
(2 X 20 ml) = 40 ml (each)

The Dilution pop-up window contains these parameters:

- **Number of dilutions:** The number of new samples that will be created.
- **Suffix mask (use # for numbers):** The suffix that is added to the content IDs to retain their uniqueness.
- **Start index:** Identifies the index from which the incrementation begins in the suffix mask.
- **Location:** of the diluted contents (similar to Derivation).
- **Quantity that is extracted from the parent:** The amount that is extracted individually for each diluted sample (not the total).

- **Dilution Factor:** The diluted contents will have an amount equal to the Quantity that is extracted from the parent multiplied by the Dilution Factor. This relates to how much of the buffer goes into the sample to dilute it.
- **Remove original sample:** This can be checked to remove the parent content at the end of the action in case the original content has either been used up or the rest was discarded.

### 3.10.13. Mix Contents

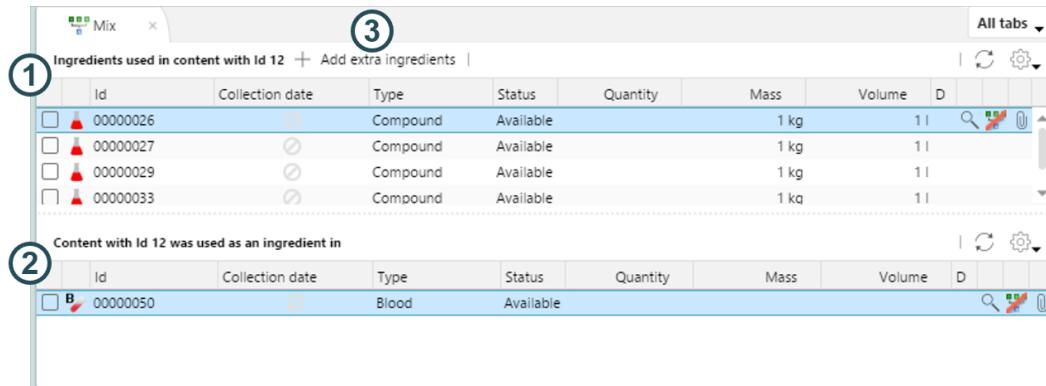
The mix action combines multiple contents that are defined as ingredients together into a new content that is defined as a mix.

The mix action consists of the following steps:

- First, select the samples (contents with content type defined as ingredients) to be mixed.  
Then use the "Mix" action.
- **Id:** The resulting mix's Id.
- Enter the parameters for the new content (the mix).

There is a tab available in the bottom of the content management module that contains information on how this sample was used in mixes. The top list (1) shows the ingredients that were used to create this mix. It also allows more ingredients to be added to the mix. The bottom list (2) shows the mixes that were created with the selected sample as an ingredient.

**Figure 3.13. Mix window**



The Mix Module is illustrated in the above image. The details refer to content *Mix1*. Part 1 represents all the ingredients that are used in *Mix1*, whereas part 2 represents all the mixes for which *Mix1* is an ingredient. If more ingredients should be added to *Mix1*, click on Add extra ingredients (3), select the ingredient contents, and then click Add as ingredient.

① Mix and Ingredient relationships can also be viewed in the Derivation tree in the Derivations tab of the content module.

### 3.10.14. Change Parent

This function allows a user to change the parent of a content, but only if the user's role has permission to do so.

This action requires the following Role Access rights to work:

**Table 3.15. Change Parent Permissions**

Access rights Category	Access rights Name	Purpose
Routine: Content	Can Change Parent	When enabled, the role user has the Can Change Parent action in the selection menu. This action enters link mode to choose the new parent content for the selected content. The original content will now show up as derived from the parent content in the Derivations tab.

To change the parent of a content the user follows these steps:

- Click the content or check the row's selection box for the content that needs to have its parent changed.
- Right-click the selected content to show the Content context menu, or open the Selection menu.
- Click the option "Change Parent."
- An overlay will appear. Choose the content that should be the new parent and check the box, then click on the "Choose Parent" button.
- A message will be displayed to show that the parent changed successfully.

① This action has some limitations:

- The *Change Parent* option in the Content Context Menu will not be available for roles who are not allowed to change the parent of records. This should be limited to an administrator action.
- An error message will be shown if the new parent is part of the derived content tree of the first record or if the selected parent is the same as the first record. This will prevent infinite loops.
- An error message will be shown if the user does not select a new parent.
- Only multiple contents with the same parent can be changed at a time.

## 3.10.15. Clear Parent

This function allows a user to clear the parent of a content if the user's role has permission to do so. A role is able to clear the parent of a content if the "Routine > Content > Can Change Parent" box is checked in the *functionality access rights* tab in the Roles module.

To clear the parent of a content the user can follow these steps:

- Click the content or check the row's selection box for the content that needs to have its parent changed.

- Right-click the selected content to show the Content Context Menu.
- Click the option "Clear Parent."
- A message will be displayed to show that the parent has been cleared successfully.

As a result, the selected content will no longer have a parent.

① This action has some limitations:

- The *Clear Parent* option in the Content Context Menu will not be available for roles that do not have permission to change the parent. This action should be limited to administrators.
- An error message will be shown if the content needs to be derived because of its content type and cannot be dissociated from its derivation tree.
- Only multiple contents with the same parent can be changed at the same time.

## 3.10.16. Merge Two Contents

This action allows the user to merge two contents together and if their role has permission to do so. ① This functionality should not be used often and only by people who know about the internal workings of SLIMS. The merge action will result in keeping one of the two content (the "remaining content") and permanently removing the other content (the "merging content").

This action requires the following Role Access rights to work:

**Table 3.16. Merge Contents Permissions**

Access rights Category	Access rights Name	Purpose
Routine: Content	Can Merge Content	When enabled, provides the "Merge Content" action in the selection menu (with multiple contents selected). The action is still restricted by the state of the selected content (for example, if a content is being used in the ELN or is linked to an order).

To merge content the user should follow these steps:

- Check the boxes for the two contents that should be merged in the content module.
- Click on the *Merge Content* button from the *Selection* dropdown menu.
- A window will appear in which the user can select which content will be the remaining content. It will be shown like this:  
Id: 0 Option 1: content1 0 Option 2: content2
- A second window will appear in which all the differences between the contents are displayed. It will be shown like this:

Field name:  Main: option1  Merging: option2  
The user can select the desired values for the different fields.

- Finally the user clicks the "Confirm" button.

The remaining content has the following properties:

- The selected values for the fields that had differences.
- The following properties from the merging content:
  - Derived contents
  - Attachments
  - GUID if the main content did not have a GUID itself

The merging content will be permanently removed and an event will be generated in the history of the main content to track that the merging content was merged into this resulting content.

ⓘ This action has some limitations:

- The "Merge Content" button will only appear when two contents are selected, and will be hidden if one, or more than two are selected.
- An error message will be shown in any of the following cases:
  - When one of the contents is linked to an order or the ELN. More specifically, if the content is referred to in any of the following tables: *ExperimentRunStepContent, Request, Result, RuleRun, WorkListRow, ContentRequest, OrderContent*.
  - When the contents have a different Content Type from each other. This would cause problems when evaluating the differences, because the fields may be different between different Content Types.
- Some fields that differ between two contents are not shown:
  - Modified by and modified on
  - Created by and created on
  - Location, Position column, and position row: these are derived from the location path
  - Barcode: the barcode of the remaining content is used by default
  - Hidden fields

### 3.10.17. Enroll Content

In Content management, the "Enroll Content" action is only available within the Selection menu and not within the right-click menu of a content. By checking one or multiple content(s), the Selection button becomes available and the action "Enroll Content"  is shown within the Selection menu. After selecting the "Enroll content" action, a pop-up window will appear where one can select a study to enroll the sample into.

By enrolling a content in a study, the list of Content Event Types and CRF Templates that can be added to that content get filtered by the selected Study.

Custom Fields can be added to the enrollment form by using the 'Content Study' table with the metadata tag "cnst\_name." This allows the field to be entered when the content is enrolled by the user, and it can be viewed by right-clicking on the enrolled content in the Studies Tab and selecting 'Detail.' The custom fields cannot yet be pre-filled using an "Enroll Content" step in a macro.

Content study can be used as a value table in a dynamic choice field to provide selection of the content study on forms. The details about the content study in the field can be displayed if "Allow to view detail of the referenced record(s)" is enabled on the field.

⚠ Contents can only be enrolled into a study if their Content type has the option *Can Enroll in a Study* enabled. A warning will be provided if a user tries to enroll a content in the same study more than once to prevent issues with custom enrollment fields.

The left side of the Content Module contains the "By Study" filter. This is enabled for specific roles by the role functionality "Filter by Study." Filtering by a study filters the content list to samples that have been enrolled in the selected study. The content event tab will also be filtered to content events for the selected study. Resetting the content list filter will clear the study selection and show all content and content events again.

### 3.10.18. Create Results

Role users can add a new result by clicking on the "New" button available in the "Results" tab if they have the below permission.

This action requires the following Role Access rights to work:

**Table 3.17. Create Results Permissions**

Access rights Category	Access rights Name	Purpose
Routine: Content	Create Results	When enabled, the role user can add a new result in the Results tab in the Content module by clicking the "New" button. A pop-up window opens so the user can select any available test for the content and enter a value. The tab still displays information when this permission is disabled, but the "New" button to add results is not available.

After clicking on "New," a pop-up window opens so the user can select any available test and enter a value. The following actions are available in the pop-up window:

- *Create and add another record:* The result is created, but the pop-up window stays open so the user can immediately add another result.

- *Create*: The result is created and the pop-up window is closed.
- *Cancel*: Closes the pop-up window without creating a result.

It is also possible to update or create results using the Export, Create Template, and Import from Excel actions.

## 3.11. Studies

### 3.11.1. Studies

In the Studies module, users can create, update, and delete studies. The "Add" button on the top-left is used to create a new study. Studies are defined by the following fields:

The following Role Access rights can be configured for use in this module:

**Table 3.18. Required Permissions for Studies Module**

Access rights Category	Access rights Name	Purpose
Setup: Studies	Studies	This permission provides access to the Studies module in which users can create, manage, and delete studies to use in the SLIMS instance.

**Table 3.19. Optional Permissions for Studies Module**

Access rights Category	Access rights Name	Purpose
Routine: Content	Study Tab in Contents Module	When enabled, the Study Tab is available for the role user to add to the visible tabs at the bottom of the Content Module.

- *Name*: Display name of the study
- *Description*: Description of the study
- *Group*: The group used to restrict access.

① The Group permission works differently for Studies, where the permission only needs to be defined at the Study level. Once that is done, the related components of a study inherit the group permission, so the permission does not need to be defined on the Content Events, Case Report Forms, etc. that are related to the study.

- *Only for selected content event types*: Restrict content event types that can be added to this study
- *Only for selected CRF templates*: Restrict CRF templates that can be added to an event of this study
- *Active*: Content can only be enrolled in active studies

Additional custom fields can be added by using the study table in the *Fields* module. The related meta data name "stud\_(name)" will be automatically entered for the unique identifier as the user types. Any added custom fields will show up on the details in the Studies Module.

ⓘ Contents can only be enrolled into a study if their Content type has the option *Can enroll in a study* enabled. If users try to enroll a content in the same study twice, a warning will be displayed. This prevents any issues from updates to custom fields relating to enrollment.

## 3.11.2. Content event types

The functionality "Content event types" needs to be enabled for a specific role in order for it to access the module Content event types under the Setup menu.

In this module, the user can create new event types using the "Add" button on the top-left. There are two kinds of Content Event Types: Standard, and Flag. Standard events are study events that occur to a content during the course of a study and need to be tracked. Flags indicate to users that something has happened to a content during an experiment run that they should pay attention to.

The following Role Access rights can be configured for use in this module:

**Table 3.20. Required Permissions for Content Event Types Module**

Access rights Category	Access rights Name	Purpose
Setup: Studies	Content Event Types	This permission allows the role user to access the Content Event Types module in which study event types and flags can be created.

### 3.11.2.1. Standard Content Event Types

Content type events of 'Standard' type are defined by the following fields:

- *Name*: Name of the event type
- *Description*: Description of the event type
- *Type*: Standard or Flag
- *Barcode mask*: Barcode mask for the content event type
- *Date type*: The following options are available
  - *Date*: The related event of the event type with this date type will have an extra Date field in order to fill in the date of the event.
  - *Date time*: The related event of the event type with this date type will have an extra Date time field in order to fill in the date time of the event.
  - *Date range*: The related event of the event type with this date type will have an extra 'Date from' and 'Date to' field in order to fill in the date range of the event.
  - *Date time range*: The related event of the event type with this date type will have an extra 'Date time from' and 'Date time to' field in order to fill in the date time range of the event.

- *Additional custom fields*: Restricted Content event fields can be added to the Content event type. Custom fields can be added in the *Fields* module for the Content event table.
  - ① Default Content event fields (available for all Content events) are visible and can only be restricted to specific Content event types from the Fields module.
- *Use Now as the Default Date(Time)*: If enabled, the default date (time) on creation of an event with this event type will be filled with the current date.
- *Only for Selected CRF Templates*: Defines which case report templates can be added to an event of this event type (see Case report form templates for more information).
- *Active*: If disabled, inactive event types cannot be chosen for new events, but existing events of an inactive event type remain available.

A copy action is available in order to copy a CRF template and all its restricted CRF fields.

### 3.11.2.2. Flag Content Event Types

Flags are different from other content events because they have consequences for content in a workflow. Some consequences require the affected protocols to be particularly selected. Content events of "Flag" type are defined by the following fields:

- *Name*: Name of the flag
- *Short Name*: Short Name of the flag for the system to recognize it.
- *Color*: A color to associate with the flag so that it stands out and is recognizable to the user.
- *Description*: Description of the flag to outline its usage or application.
- *Type*: Standard or Flag
- *Consequence*: Determines what happens to a content that is flagged, such as whether it can proceed in the protocol or not, whether it must be cancelled, requeued, or can proceed in a different protocol. (See explanation of the consequence options below.)

If a Forbid consequence is selected, the Forbidden Protocols table appears allowing particular protocols to be selected to close off parts of a workflow to a flagged content.

- *Only allow adding this flag in selected protocols*: Restricts the flag to particular protocols. When adding a flag in a certain step, the user will be able to choose from unrestricted flags, or the flags restricted to that step. The protocol will need to be versioned manually when these restrictions are specified. Protocols and flags can also be restricted from the workflow configuration.

When turned on, a table with the Protocol Steps appears so that the restricted ones can be selected.

- *Active*: Inactive flags can no longer be applied to new experiments, but existing flags of an inactive event type remain present.

When a flag is added or removed from the list of allowed flags for a protocol step, the protocol with that step will need to be manually versioned. The version button will be displayed as yel-

low to indicate that there are changes. This doesn't apply if the flag is not restricted to any particular protocols.

⚠ Disabling the restriction on a flag impacts **all versions** of the protocol including the draft and old versions, even if protocol runs are still in progress for these versions.

### 3.11.2.3. Adding Flags Automatically in Workflows

Adding flags automatically is possible in protocol steps of a workflow based on a Groovy expression (not in the Protocols or Simplified Protocols modules). In the "Expression to add flags" field, Groovy can be used to add flags based on conditions on the content, results, or on other more complex criteria. From within the value expression field, the content, its results, the protocol run step, and the protocol step are directly accessible, as well as the usual daohelper and utilities.

Flags that have already been created in the Content Event Types module (using the tutorial in the above sections) can be used in the expression and added automatically. Groovy expressions for flags that do not yet exist can also be included in a protocol step, but they will not work unless there is a corresponding flag in Content Event Types. The Groovy expression will be reevaluated whenever contents or their results are updated and saved in the current protocol step, and when the user proceeds to the next protocol step by clicking Next, or Finish if it is the last step. If the condition to trigger the flag on a result is no longer true after the update, then the flag will be removed on that protocol step. If a condition on a content is no longer valid after the update on a different protocol step, the flag triggered in the current protocol step while that condition was met will not be removed.

The new Groovy syntax `flag(flagName, condition)` can be used in the Expression to add flags field. `flagName` is the full name of the existing Flag, and the `condition` is a statement that should resolve as either true or false. If the flag exists and the condition is met for the current protocol step, the flag will be attached to the content. If the condition is not met, or is no longer met after an update on that protocol step, the corresponding flag will be removed.

The "Expression to add flags" field is checked when the content is updated and saved, when the content's results are updated and saved, and when the user clicks Next or Finish in this workflow protocol run step.

Example of some simple Groovy flag expressions:

ⓘ In this example, `cntn_cf_concentration` and `cntn_cf_mass` are decimal number fields and not quantity fields, so the return type of the groovy matches the datatype of decimal number.

- Condition on content:

```
flag("Concentration Too Low", content.cntn_cf_concentration < 5)
```

- Condition on content results for test "pH":

```
flag("pH Too High", content.results.pH.rslt_value.amount < 7)
```

A more complex flag expression with a condition written separately:

```
groovy
concentration = content.cntn_cf_mass / content.cntn_cf_volume
condition = concentration < 3 || concentration > 8
flag("Concentration Out of Range", condition)
```

### 3.11.2.4. Explanation of Consequences

Flags are added manually in a protocol run step inside a workflow, and can be added to one or more content records. The Add Flag option is in the context and selection menus in the content tab.

The flag consequences result in these behaviors:

- *No Consequence*: This is presented as a purely informational flag, and there is no restriction to content movement through the workflow.
- *Forbid a set of protocols for samples with this flag until requeue*: The content can no longer proceed through the restricted protocols and will need to take another path through the workflow or be requeued. Requeueing will allow the content to continue in the protocol if it is requeued to a queue that is earlier in the flow to the one that activated the flag on the content.
- *Forbid a set of protocols for samples with this flag indefinitely*: The content can no longer proceed through the restricted protocols and will need to take another path through the workflow. Requeueing will not be possible if there is no path through the workflow from the start queue around the restricted protocol/s, so the content in that case would have to be cancelled.
- *Require Requeue*: The content cannot continue through the workflow and will have to be requeued.
- *Require Cancel*: The content cannot continue through the workflow and will have to be cancelled.

① Flags can be added to content from the Content module, but flags added this way do not show up in workflows.

### 3.11.3. Content events

The 'Events' tab allows the user to access, edit, and add events to the selected content in the content list.

This feature requires the following Role Access rights to work:

**Table 3.21. Content Events Permissions**

Access rights Category	Access rights Name	Purpose
Routine: Content	Content Events	When enabled, the role user can access the "Events" tab in the Contents module. Role users can view and update the Content Events for the selected content in the tab.

In the top part of the Events tab, events can be added with the "Add" button at the top-left of the Content event list. A pop-up window will be shown with a form of fields for the new event. First, a study for which the content has been enrolled can be chosen for the event. Secondly, a

Content event type can be selected. The selection is filtered by the selected study. Additional custom fields could be available depending on the configuration of the Content type event (see [Content event types section](#) for more information).

In the bottom part of the Events tab, any studies the content is currently enrolled in are shown in the table, and the content can be enrolled or removed from a study directly from the tab options. The context menu is available by right-clicking on an event and shows the Open Detail option.

Finally, CRFs are grouped by Case report form templates and can be added with the green plus icon that appears in the highlighted Event's row. A grid with the CRF's datapoints can be set up by right-clicking the header of the CRF in the event and using the "columns" option to add the desired columns. The Open Detail menu is also available in the right-click context menu in the CRF grid.

ⓘ It is not possible to change the Content event type after creation.

### 3.11.4. Case report form templates

The following Role Access rights can be configured for use in this module:

**Table 3.22. Required Permissions for CRF Templates Module**

Access rights Category	Access rights Name	Purpose
Setup: Studies	CRF Templates	This permission allows the role user to access the CRF Templates module where Case Report Form Templates can be added and managed.

In this module, the user can create new Case Report Form Templates using the "Add" button on the top-left. Case report form templates are defined by the following fields:

- *Name*: Name of the Case report form template.
- *Description*: Description of the Case report form template.
- *CRF Unique Identifier Mask*: Generates a root name and incremented number based on the mask CRFR##### for each CRF that is generated from the template, or a mask customized by the administrator. See the Name mask (use # for numbers) definition for more details.
- *Additional custom fields*: Restricted CRF fields can be added to the CRF template. Custom fields can be added in the *Fields* module for Case Report Form Template and Case Report Form tables.

- ⓘ Default CRF fields (available for all CRFs) are visible and they can only be restricted to specific CRF templates from the Fields module.
- *Active*: Inactive Case report form templates can no longer be chosen for new case report forms, but existing case report forms of an inactive Case report form template remain available.

A copy action is available in order to copy a CRF template and all its restricted CRF fields.

### 3.11.5. Case Report Forms

Case Report Forms can be added to an existing Content event by using the "Add" button next to the detail button of the Content event. A pop-up window will be shown with a form of fields for the new Case Report Form. The selection of the CRF templates are filtered by the Content event type and the study to which the content is enrolled to.

CRFs can also be imported from Excel. Case report forms for multiple Content events can be imported at once. A "CRF template file" can be downloaded from the tools menu in the Events tab. The link between the CRF and the content event is done by filling the barcode of the Content event in the Excel file. This file can then be imported into SLIMS using the "Import CRF" action.

⚠ Case report forms can only be added to Events that have an Event type that allow adding their Case report form templates (See *Content event types section*).

Once a Content event has some Case Report Forms, a gray triangle will be visible on the left of the Content event. Toggling the triangle will open/close a sub-component that displays the existing Case Report Forms of the Content event. The open detail button used for case report forms allows the user to review the created case report form in a pop-up window.

ⓘ It is not possible to change the Case report form template after creation.

## 3.12. Stability Study Design

The concept of Stability Studies is supported in SLIMS using the Stability Study Design and Stability studies modules. Stability studies help ensure compliance with the relevant standards when determining how well contents retain their properties over time.

Stability studies are supported in SLIMS using the Stability Study Design module where the various stability study templates are first created. The Stability Study Design module is accessed



by clicking the icon from the main menu. Any number of Stability study templates can be created in this module to process batches of content and determine how well they retain their properties over time to ensure compliance with the relevant standards. A template can be created for each study with their own timepoints and storage conditions to test different environments, such as how well the product retains certain properties when stored in a hot humid environment over a period of time versus a temperate environment.

Stability studies can be used with Products and specifications to determine location-dependent tolerances, such as the temperate climates found in some countries versus the tropical humid or cold dry climates of others. Stability studies also use Workflows to aide in processing tests that measure contents' properties, such as weight retention, dissolution, disintegration, spectrophotometry, etc.

This module requires the following Role Access rights to provide access to a role user:

**Table 3.23. Required Permissions for Stability Study Design Module**

Access rights Category	Access rights Name	Purpose
Setup: Specification	Stability Study Design	Disabled by default. When enabled, this permission provides

Access rights Category	Access rights Name	Purpose
		access to the Stability Study Design module which allows the creation of Stability study templates to establish conformity to a standard for content over time in various storage conditions.

### 3.12.1. Stability study templates

The Stability Study Design is made up of templates that define the stability study you want to create. The template can be reused for additional studies as needed, and thus is versioned so a study can be associated to a particular version of the template. Stability study templates require a product (from the Products and Specifications module) in order to be used for a stability study. However, templates can be created without products (in which case, they serve as copyable templates of templates).

Stability study templates consist of Storage condition templates which are composed of Timepoint templates. They are built in a tree structure in the left panel.

**Figure 3.14. Stability Study Template Concept**

Name	Active	Product	
Paracetamol Stability	<input checked="" type="checkbox"/>	Pill Tests	  
Cold room	<input checked="" type="checkbox"/>		 
Hot room	<input checked="" type="checkbox"/>		 
Timepoint 1	<input checked="" type="checkbox"/>		
Timepoint 3	<input checked="" type="checkbox"/>		
Timepoint 6	<input checked="" type="checkbox"/>		
Temperate room	<input checked="" type="checkbox"/>		 
Timepoint 1	<input checked="" type="checkbox"/>		
Timepoint 3	<input checked="" type="checkbox"/>		
Timepoint 6	<input checked="" type="checkbox"/>		

The history can be accessed to show each level of the tree. The gear menu in the tree panel provides the "Show Stability Study Template History," "Show Storage Conditions History," and "Show Timepoints History" actions. The "Show History" action is available by right-clicking any of the tree levels to see their respective history. The root template level also shows the histo-

ry of storage conditions, and the storage condition levels also shows the timepoints that were added, edited, or deleted from the selected storage condition.

Clicking the  button opens a form to create the new template. Once the template is created with details, most of the selections can be edited in the "Stability study details tab" on the right. Existing Stability study templates are selected from the dropdown at the top of the window to filter the details in the module so only that template is shown. Creating a new template selects that one by default.

The form provides these options to configure the stability study behavior:

- **Name:** (Required) The name of the stability study template.
- **Unique identifier:** (Required) Auto-filled as the name is typed with the format "stst\_name\_draft." The unique identifier cannot be edited later.
- **Description:** Can be filled after creation.
- **Use Pulling Orders:** Enables usage of pulling orders (optional). This is functionality which automatically generates orders of a type specifically created for retrieving the desired number of pulls at certain times based on the timepoints when batches are enrolled in the stability study based on this template. A pulling order will be created automatically and registered for each timepoint to be executed in the Workflows module. When the pulling order is scheduled, the content is queued into the workflow. In this way, the pulling order links what needs to be tested and on which content, and then edits a measurement order automatically with its output. It is not possible for users to edit pulling orders to add or remove content or requests from them; only the stability study can create pulling orders with requests and linked content.

This option hides the pulling-related fields in other tree levels if it is disabled. The "Pulled Contents" tab in the finished stability study will also be hidden. If settings are entered in the pulling fields and saved, hiding them by disabling pulling does not delete the settings, it just makes them inactive so they can be retrieved.

If pulling is enabled, the related fields are shown and are "Validated" when a version of the stability study template is created. Validated fields can initially be left blank and saved, but must be filled in order to create a version.

These options are available only when "Use Pulling Orders" is enabled:

- **Pulling Range:** (Validated when creating a version) The number of leeway (flexibility) days to allow in case the pulling date is on a weekend or it is too busy to complete on time. Allows a setup where samples outside of this range are considered unviable.
- **Pulling Order Type:** (Validated when creating a version) The list provides a selection of all the order types of order type Workflow available in your instance. Only Workflow Order type orders can be used as pulling orders. An order of the selected type will be created in the Orders module when batches are enrolled in the study.
- **Pulled Content grid template:** (Required) The pre-selected default is "Default Stability Study Pulled Content" which is always available. Only grid templates of type "Stability Study Pulled Content" can be selected. They can be created and edited in the Grid Templates module. See how to create them here: [Section 9.3.1](#)

- **Measuring Range:** (Validated when creating a version) The number of leeway (flexibility) days to allow in case the measuring date is on a weekend or it is too busy to complete on time.
- **Measurement Order Type:** (Validated when creating a version) The list provides a selection of all the order types of order type QC available in your instance. Only QC order type orders can be used as measurement orders. Measurement orders are specifically for the operation of stability studies to facilitate performing the selected test specifications at certain times. They are created automatically when batches are enrolled in a stability study based on this template, and when using pulling orders, are updated directly from the pulling order output linked to the same batch. The specifications are assigned as requests depending on the timepoint specifications of the stability study template.
- **Results grid template:** (Required) The pre-selected default is "Default Stability Study Result" which is always available. Only grid templates of type "Stability Study Result" can be selected. They can be edited and created in the Grid Templates module. See how to create them here: [Section 9.3.2](#)
- **Product:** A selection list to the products in the Products and Specifications module, but only "draft" products are available. The product field cannot be edited later. This selection ties the stability study to the batches that will be enrolled and impacts the specifications for the timepoints.
  - When a product is not selected, the stability study template can be used as a template for other templates. A stability study cannot be created based on a template without a product, but the template can be copied and reused for other similar stability study templates. The field becomes read-only when the template is versioned.
  - When a product is selected, the stability study template can be used as the basis to create a stability study. Batches (contents) that have the same product can be enrolled in the stability study. Additionally, the following field becomes available.
- **Product specification version to reuse:** (Available when a product is selected.) A selection list of the versions of the product selected in the above field. When this field is left empty, only the specifications created at the stability study template root level and timepoint level will be used. When a product specification is selected, the specifications that are propagated in the selected product from the Products and Specifications module will be propagated to the stability study template root specifications. The specifications are propagated without an "amount to pull" field because pulling is not relevant to products alone. If using pulling orders, the "amount to pull" can be filled on each of the inherited specifications in the stability study template's root specifications. Specifications that are not propagated from Products and Specifications will not show in the selection field. (See the Products and Specifications section for where to enable propagation: [Section 5.4](#))

The specifications from the product will show as inherited in the stability study template root specifications. The specifications cannot be edited in the root specification except for the "Active" attribute, but can be edited from the product in Products and Specifications. The selected product can be changed later and the inherited specifications will be removed and replaced with the new ones.

Versioning the product in Products and Specifications will not impact stability studies that are run on the stability study template. A warning is provided on the stability study template to let you know if "This stability study is using an old product specification version." Even if the product specification version to reuse is updated to the new version, and the template is

saved and versioned, it will not impact the stability study that was built and running on the prior template version. This protects the running stability study's specifications.

### 3.12.1.1. Stability Study Template Details Tab

All of the configuration added to the Stability Study Template is available on the Details tab. Several fields are added to the tab to provide visibility on the configuration of the stability study template's child levels.

- **Applicable Specification Rules:** A read-only grid that shows the overview of root specifications at a glance that exist for the stability study template, as well as tests, operators, expected values, and more. Columns can be added or removed in the grid.
- **Storage Conditions (grid on template details tab)** Any storage conditions that are added to the stability study template appear in the "Storage Conditions" grid. The grid is read-only and displays the name of each storage condition and all of the custom fields that are added to storage conditions that have the "Show in dropdown list" option enabled.
- **Total Amount for Study** (Hidden when pulling is disabled) This read-only field reports the totaled number of pulls from the specifications of all child timepoints in the stability study template, and the number of additional pulls of all child storage conditions. This also includes the number of pulls that were propagated into timepoints via the root specification rules (inherited rules that are active on the child timepoints). This value gets updated when any specification or storage condition is deactivated (which lowers the amount to pull) or when the number of pulls is updated on any of the specifications or child storage conditions.

### 3.12.1.2. Root Specifications Tab

Rules can be associated with a Stability Study Template that will be checked on new instances of the stability study. There are no specification rules on a template at the start (unless they have been purposefully propagated from Products and Specifications), but they can be added by clicking the "New Specification Rule" button. Once a rule is created, it is added to the Root Specifications grid which is read-only. However, the specifications can be edited by clicking the "View Details" (magnifying glass) icon to open a pop-up to view and edit the details.

Root specifications can be imported, exported, and template files can be created according to [Section 5.4.3](#)

The purpose of root specifications is to create the specification rules that are common across timepoints in the stability study, so root specifications are propagated by default to all their child timepoints. The propagations can be controlled by deactivating or leaving the specifications active on the individual child timepoints.

Root specifications can be created with the following configuration to be shown in the grid:

- **Name:** Name of the specification rule.
- **Amount to Pull:** (Optional, hidden if pulling orders are disabled) The number of content that is needed to complete this test. (For propagated root specs, this amount to pull will be propagated to each timepoint with the specification rule.)

① This field is "0" by default for specifications inherited from the Products and Specifications module, but it can be edited to the desired value.

- **Mode of Evaluation:** Match specification, script specification, or no evaluation. "No evaluation" rules are available for specifications but not for global rules (created in the Rules module). A "No evaluation" mode of evaluation allows a rule to be created that saves the executed test value for later statistical analysis without imposing any limits on the value. The evaluation of the result goes directly to "Verified." See the *Rules* section for an explanation of how to create rules with the other modes of evaluation.
- **Tests:** The selected test(s) that the specification checks.
- **Minimum # of automatically created results:** (Optional) Set the number of results that should be automatically created in the results step. This should be set with care because SLIMS creates whichever are higher between the number of results configured here or configured in the results protocol step. More details: *Number of Automatically Created Results*
- **Aggregation Method:** (Optional, only available on tests with numeric result values) Choose standard deviation or mean to create an automated aggregated result over all the results in the results step. Keep in mind that all the results requested via the specification and the results protocol step will be aggregated. More details: *Aggregation Methods*

Independent specification rules can be created to check the user-entered results and aggregated results. Specifications created to ensure that the aggregated result values are within the desired ranges only apply to the aggregated result rows. Likewise, those created to ensure that the user-entered results are within range do not apply to the aggregated results. For example, given there are 5 results with a specification that checks for a standard deviation between "0g" and "1g," and another specification for a mean between "5g" and "10g," and the user-entered results are: "1g," "2g," "3g," "4g," and "5g."

- The user-entered results go to status "Verified."
  - The mean and standard deviation rule evaluations trigger and the results go to status "Available."
- If the user-entered results are updated to: "6g," "6g," "6g," "6g," and "6g."
- The user-entered results are re-evaluated and are still at status "Verified."
  - The mean and standard deviation recalculate, are re-evaluated, and go to status "Verified."
- **Operator:** When available (for match specifications), the operator provides options for setting up the acceptable ranges for the rule. The datatype of the test filters the operator list so only those that make sense are available.
  - **Expected Value(s):** The value(s) that are considered *acceptable* for the test result. An entered value outside of the expected range will trigger the warning/error for the specification rule.
  - **Min:** The minimum required value when using a test that has such a value, such as between (inclusive).
  - **Max:** The maximum allowable value when using a test that has such a value, such as between (inclusive).
  - **Default Analytical Method:** The default analytical method through the analytical workflow that checks the specification rule for the timepoint. This is just the path through the workflow that is suggested automatically and can be changed when the user schedules the measurement order. The default path can also be restricted by setting the

"schd\_fk\_workflowRequestable" field to read only, in which case the user will not be able to choose a different analytical method when scheduling (useful in cases where only default analytical methods will ever be used to test specifications in SLIMS).

- **Active:** Denotes whether the specification rule is active or not. Keep in mind that a propagated root specification that is inactivated will also deactivate its propagations.
- **Inherited:** Denotes whether the specification is inherited or not (whether it is propagated from a parent). Root specifications could be inherited from products if they were intentionally propagated from the Products and Specifications module.

## 3.12.2. Storage condition templates

This tab is available when a Stability Study Template is selected. Storage conditions can be used to store batches of contents in the stability study. Storage conditions can be refined by creating custom fields on them such as humidity, temperature, etc. Use the "StabilityStudyStorageTemplate" table to add custom fields. Storage conditions can be added from the "Storage

Condition Templates" tab or by clicking the button in the root template's row with the  icon. Most of the details for the selected storage condition can be edited afterward by selecting a storage condition and viewing its "Storage Conditions Details tab" on the right.

The "New Storage Condition" form provides these options:

- **Name:** (Required) Name of the storage condition.
- **Unique identifier:** (Required) autofills as the name is typed with the format "sst\_nameOfTemplate\_templateVersion\_nameOfStorageCondition."
- **3-letters Abbreviation:** A maximum of any three characters can be entered.
- **Color:** Use the html color code or the color picker next to the field to specify a color for the storage condition icon. The appearance of the icon will change to match in the stability study template tree and in the finished stability study.
- **Description:** A description can be entered after creation.
- **Pulling Requestable:** (Hidden if pulling orders is disabled, validated when a version is created) This specifies the requestable of a standard workflow that will be used to retrieve pulls from the enrolled batches. Workflow requestables that have "Link the output of this requestable as a fulfillment in order management" enabled can be selected in this list.
- **Additional Storage Amount:** (Optional, hidden if pulling orders is disabled) Aside from the pulls needed for the root specification rules at the template level and the individual timepoint specification rules, this is the number of spare content that will be needed to pull for testing. For example, if it is desired to have some extras in the freezer or hot room in case of breakages.
- **Active:** A switch to indicate if the storage condition is active.

### 3.12.2.1. Storage Condition Details

A storage condition can be selected in the stability study template tree once it has been created. This provides further overview information in the "Storage Condition Details" tab beyond what was entered when creating the storage condition.

- **Timepoints** A read-only grid that displays the child timepoints within the storage condition. It shows whether each timepoint is active or inactive. Additional columns can be added by enabling the "Show in dropdown lists" attribute on fields in the Fields module, which will then show in this grid. For example, the timepoint schedule information "amount" and "unit" fields could be added to show an overview of the amount of time between the stability start and each of the timepoints.
- **Amount for Storage** (Hidden if pulling orders are disabled) Displays a read-only report of the total amount of pulls from the storage condition's child timepoints and its additional storage amount. This value gets updated if a specification is removed from a timepoint or the additional storage amount gets updated.

### 3.12.2.2. Timepoint templates

Once a Stability Study Template Storage Conditions is created, the timepoints can be added from the tree in the storage condition's row with the  icon, or with the "Timepoints" tab in the right panel while the storage condition is selected.

Timepoints can be added with the following attributes:

- **Name:** Enter a recognizable name for the timepoint.
- **Unique Identifier:** Automatically filled as the Name is entered.
- **3-letters Abbreviation:** Enter 3 digits or letters that will show up next to the timepoint icon in the stability study template and the created stability study.
- **Amount:** Enter a number to indicate how much time should pass between the start of the stability study and this timepoint (example: "10" days).
- **Unit:** A selection of units of time that works with the Amount to determine how much time should pass between the start of the stability study and this timepoint (example: 10 "Days").

Timepoint specifications can be imported, exported, and template files can be created according to Section 5.4.3

### 3.12.3. Stability Study Timepoint Template Tab

The details of created timepoints appear on the "Stability Study Timepoint Template" tab. Additional details show up on this tab as well:

- **Total Amount to Pull:** (Hidden if pulling orders are disabled) Read-only field that shows the totaled number of pulls from this timepoint's specification rules and all of its inherited root specification rules. This field gets updated when a specification is deactivated, reducing the number of pulls needed.

#### 3.12.3.1. Timepoint Specifications

The Specifications tab in the context of timepoints is for the creation of specification rules that apply to the individual selected timepoint. They are created the same way as the specification rules on the root stability study template specifications. However, the number of pulls in this case defines the pulls needed for the test on this individual timepoint.

The root specification rules that have been inherited from the stability study template level can be deactivated on the individual timepoints to control which timepoints the specifications will be tested in. Inherited specifications cannot be edited otherwise on the timepoint level. It is easier to create specification rules common to many timepoints on the root specification level, and create specification rules used for individual timepoints on the individual timepoint specification level.

## 3.12.4. Stability Study Order Types

Stability studies need to use automatic creation of orders in the Orders module. Orders of two different types are made: Workflow Orders and QC Orders. Their purpose is explained below.

### 3.12.4.1. Pulling and Measurement Orders

Pulling orders are a special usage of fulfillment orders (see more about fulfillment orders here: [Section 5.1](#)) with functionality that supports stability studies. Pulling orders retrieve batches from storage and link to the workflow in SLIMS to produce the contents needed to do stability study testing. They use the pulling information from the stability study design to track batches fetched from storage and any sample preparation time designed in the workflow. Pulling orders allow a workflow to produce content through its protocols and can be set up to prevent the workflow from finishing until the number of requested contents are available.

Measurement orders facilitate the testing of pulled content(s) by automatically creating orders that queue into one or more analytical methods or workflows. The specifications from the timepoints of the stability study template are requested for the batch or pulled contents. Measurement orders are edited by the outcome of pulling orders if pulling orders are used, but can alternatively be used by themselves.

Some configuration is required to set up stability study orders. Pulling orders need a standard workflow with specific options enabled on its requestable and an order type of Workflow Order. Measurement orders need one or more analytical workflows with analytical methods configured with the tests related to the specifications set up in the stability study design. They require the order type QC Order to route to analytical workflows.

These steps are used to create automated stability study orders from pulling to measurement:

1. Create a standard workflow with these options enabled on its requestable. (See the options in context here: [Section 6.1.2](#).)
  - **Link the output of this requestable as fulfillment in order management:** (Required to make it a pulling order) The output of the workflow will be visible in the Orders module in the Fulfillments tab. The pulled content(s) can be seen in the Fulfillments tab when the batch content in the order is selected.
  - **The output of this requestable is more than one content:** (Not required) This option allows the order to restrict the workflow completion based on the number of pulls that are needed to fulfill the request. For example, when an order based on a timepoint is scheduled and the queued batch gets through the workflow, there need to be enough contents pulled during the workflow to satisfy the requested number of pulls or the workflow cannot be finished.
2. Create an analytical workflow with results step(s) for the desired tests. There can be different analytical workflows for different tests, several different analytical methods to measure

the same test with different paths through a workflow, or even several paths in the same workflow to measure different tests.

3. On the stability study template, enable pulling orders and set pulling range, pulling order type, and pulled content grid template. Also set measuring range, measurement order type, and results grid template. If a product has not been set, and the stability study will be created from this stability study template, add the product.
4. On the storage conditions, set the pulling requestable so pull requests will go through the desired workflow. Add any additional storage amounts needed on the storage conditions.
5. On the stability study template root level and timepoints levels, ensure any needed amounts to pull are set in the specifications. The default analytical method may also be set so the user does not have to choose the analytical method when scheduling the requested measurements. However, the user will have the selection choice of applicable analytical methods when scheduling the order.
6. The versioned stability study template can be selected to create the Stability Study. See how the pulling and measurement orders are used in this section: *Section 3.13.2*.

### 3.12.5. Copying a Stability Study Design

There might be cases where much of the same stability study template is useful for more than one product. For example, a product that has a swallowable and a dissolvable format that uses some of the same specifications, but not all of them. Stability study templates can be copied using the row action menu at the root template level.

Selecting the copy action provides a pop-up to enter the values of the copy. At least the Name and Unique Identifier are required to copy the template. Copying results in a new draft stability study template with all of the entities (storage conditions, timepoints, and specifications) of the original.

A stability study template can be created to use as a template for the others by leaving the product field empty. This means the stability study template will never be selectable to base a stability study on. It can be used to retain a common build and copied into new drafts of the template, which can then be associated with products.

A stability study template version can also be copied into a new draft template. To do so, select the version in the "Versions" dropdown. Then, use the row action menu to select the "Copy" action and change the desired information for the new draft.

### 3.12.6. Versioning a Stability Study Template

Stability study templates can be versioned just like other versionable entities such as protocols or report templates. The status workflow "Default Stability Study Template Publication Status Workflow" exists by default to govern the versioning statuses from the Status Workflows module (see more about Publication Status Workflows in *Section 10.20.2*). Clicking the version button on any level of the template tree creates a version of the entire Stability Study Template. Only the latest active version of a stability study template can be used to run a stability study.

ⓘ Electronic signatures can be required to create new versions if the publication status workflow is configured as such. Keep in mind that products from the Products and Specifications module are governed by the Default Product Specifications Publication Status Workflow which have no bearing on the specifications created in the stability study design module.

Like other versionable entities, a newly created Stability Study Template is a Draft. When the new template is versioned, the Draft is copied and becomes a Version. Records (root specifications, storage conditions, timepoints, and timepoint specifications) can be added to the draft, removed, and edited until the next version is created. Removing any level of the tree also removes its children levels. For example, removing a storage condition also removes its timepoints and the specifications for those timepoints. Changing a draft and creating a new version does not influence the stability studies running on an earlier version of that stability study template. (See more about versioning behavior in Section 10.25.)

Old versions of the Stability Study Template are retained and can be accessed with the versions dropdown. Old versions are read-only, but version columns can be added to the tree grid and specification grids at the root and for the timepoints. The "Version" number column can be added to all three grids, whereas the "Version Comment" column can only be added to the root level.

### 3.12.7. Customize Grids and Reports for Stability Studies

Aside from the pulled contents and results grid templates described earlier, metadata relations are available to pull stability study information into custom report templates and grid templates. Creating a report template on a stability study table makes the report available for generation at that stability study level.

- **Stability Study:** A report on this table can be generated from the stability study level in the stability study tree.
- **Stability Study Storage:** A report on this table can be generated from the storage condition level in the stability study tree.
- **Stability Study Timepoint:** A report on this table can be generated from the timepoint level in the stability study tree.
- **TimepointEnrollment:** A report on this table can be generated from the batch level in the stability study tree. It indicates the enrollment of an individual content at a timepoint. The "StabilityStudyEnrollment" metatree relation can be added on this table, which indicates the enrollment of a content into the stability study.

#### Customize Other Grids and Reports with Stability Study Data

Grid templates and Report Templates can be designed on the "Order" and "Result" root tables with the metatree relations: Stability Study, Storage Condition, Stability Study Timepoint, Batch, and Timepoint Enrollment (which is the record of the batch enrollment into the timepoint).

The columns for stability study, storage condition, timepoint, and batch are available to be added to the Orders grid in the Orders module and the Result Overview grid in an analytical workflow by editing the views.

## 3.13. Stability Studies

The Stability Study module is where a stability study is created and run based on a stability study template. Batches of content are enrolled in the study from this module to start the pulling or measuring processes for each timepoint in each storage condition.

This module requires the following Role Access rights to provide access to a role user:

**Table 3.24. Required Permissions for Stability studies Module**

Access rights Category	Access rights Name	Purpose
Routine: Stability studies	Stability studies	Disabled by default. When enabled, this permission provides access to role users to the Stability studies module which allows the usage of created Stability study templates to ensure compliance with a standard for testing contents over time in various environments.

A new stability study is created by clicking the "New Stability Study" button. The creation form provides these fields:

- **Name:** (Required) Enter the name of the stability study which should be recognizable by other users.
- **Unique Identifier:** (Required) Filled automatically as the user types the name of the stability study.
- **Description:** (Optional) A description can be provided with any desired study criteria or supplemental information.
- **Stability Study Template:** (Required) A stability study template must be selected to base the build of the stability study on. The dropdown provides a selection of all stability study templates that have a product associated with them and that have been versioned.
- **Active:** Allows the study to be created as active or inactive.

### 3.13.1. Stability Study Module Navigation

The stability study is created with the same tree structure as the stability study template it's based on. The tree structure has the study at the root level which can be expanded to show the storage conditions. The storage conditions expand to show their timepoints, and once batches are enrolled into the study, the timepoints expand to show their enrolled batches.

The tree has the "Name" column and actions menu enabled by default, but header columns can be added, filtered, and sorted. Additional columns can be enabled to show created by, created on, modified by, modified on, attachments, and active.

Each level of the stability study tree can be selected to show its relevant tabs on the right. Depending on whether the stability study template has been configured to use pulling orders or not, these tabs may be available:

- *Selecting the root stability study:* The available tabs are Stability Study Details, Enrolled Batches, Pulled Contents, Results, and Reports.
- *Selecting a storage condition:* The available tabs are Storage Condition Details, Pulled Contents, Results, and Reports.
- *Selecting a timepoint:* The available tabs are Timepoint Details, Pulled Contents, Results, and Reports.

- *Selecting an enrolled batch:* The available tabs are Batch Details, Pulled Contents, Results, and Reports.

### 3.13.1.1. Selected Tree Details Tab

When the root stability study is selected, the details tab shows the information that was entered on its creation. Only the stability study details can be changed. When any other level in the tree is selected, the details tab shows read-only information about the selected tree level (though additional custom fields that were added are editable unless configured otherwise).

When an enrolled batch for a timepoint is selected, several important date values are displayed in the batch details tab.

- **Start Pull Date:** The earliest allowable date on which pulling can start. Comes from the stability start date value + the stability study timepoint time amount. This date is copied to the "Planned on Date" field in the Orders module, which is a column that can be added to the Orders grid. Empty if pulling orders are not used.
- **End Pull Date:** The latest allowable date on which pulling must end, otherwise the pulls are not a pull for this timepoint anymore. Comes from the pull date start + the pulling range (the leeway days). Empty if pulling orders are not used.
- **Start Measurement Date:** The earliest allowable date on which measurement can start. Comes from the stability start date value + the stability study timepoint time amount.
- **End Measurement Date:** The latest allowable date on which measurement must be performed, otherwise the results are no longer valid. Comes from the measurement date start + the measuring range (the leeway days).

### 3.13.1.2. Enrolled Batches Tab

This tab is only provided on the root stability study level. The "Enroll batch in stability study" action allows an existing content to be selected and enrolled. A content (which will count as a stability study enrolled batch) shows up in the selection list when the content is associated with the same product as the stability study. The same content cannot be enrolled in the same stability study more than once.

The batch (content) is enrolled at every timepoint of every storage condition in the stability study. It shows up under the timepoints in the tree.

The enrollment form also asks for the stability start date which defaults to today's date. The stability start date can be calculated by a value expression if desired.

The Enrolled Batches tab provides the following columns (by default) in a grid, though views can be created like in other modules in SLIMS:

- Collection date/time
- Category
- Content Type
- Content ID

- Status
- Location (including sublocations)
- Derivation Count
- Stability Start Date

### 3.13.1.3. Pulled Contents Tab

The Pulled Contents tab is available at every level of the stability study tree if pulling orders are being used. The columns shown in the grid are from the grid template selected in the stability study template. The grid shows the contents that were pulled using pulling orders and depends on the tree level that is selected.

- *Stability study is selected:* Shows the pulled contents for all storage conditions, timepoints, and batches of that study.
- *Storage condition is selected:* Shows the pulled contents for all batches and timepoints of that storage condition.
- *Timepoint is selected:* Shows the pulled contents for all batches of that timepoint.
- *Enrolled batch is selected:* Shows the pulled contents for the selected batch of that timepoint.

These columns are available for the Pulled Contents grid by default (if the grid template was not modified):

- Barcode of the Content
- Status
- Content Type
- Pulling Order ID
- Barcode of the Batch
- Stability Study Name
- Storage Condition
- Timepoint

### 3.13.1.4. Results Tab

The Results tab is available at every level of a stability study tree. The columns shown in the grid are from the grid template selected in the stability study template. The grid shows the results / rule evaluations that were completed for the enrolled batches (and any of their pulled contents). The results shown depend on the tree level that is selected:

- *Stability study is selected:* Shows the results for all storage conditions, timepoints, and batches of that study.

- *Storage condition is selected:* Shows the results for all batches and timepoints of that storage condition.
- *Timepoint is selected:* Shows the results for all batches of that timepoint.
- *Enrolled batch is selected:* Shows the results for the selected batch of the expanded timepoint.

These columns are available for the Rule Evaluations grid by default (if the grid template was not modified):

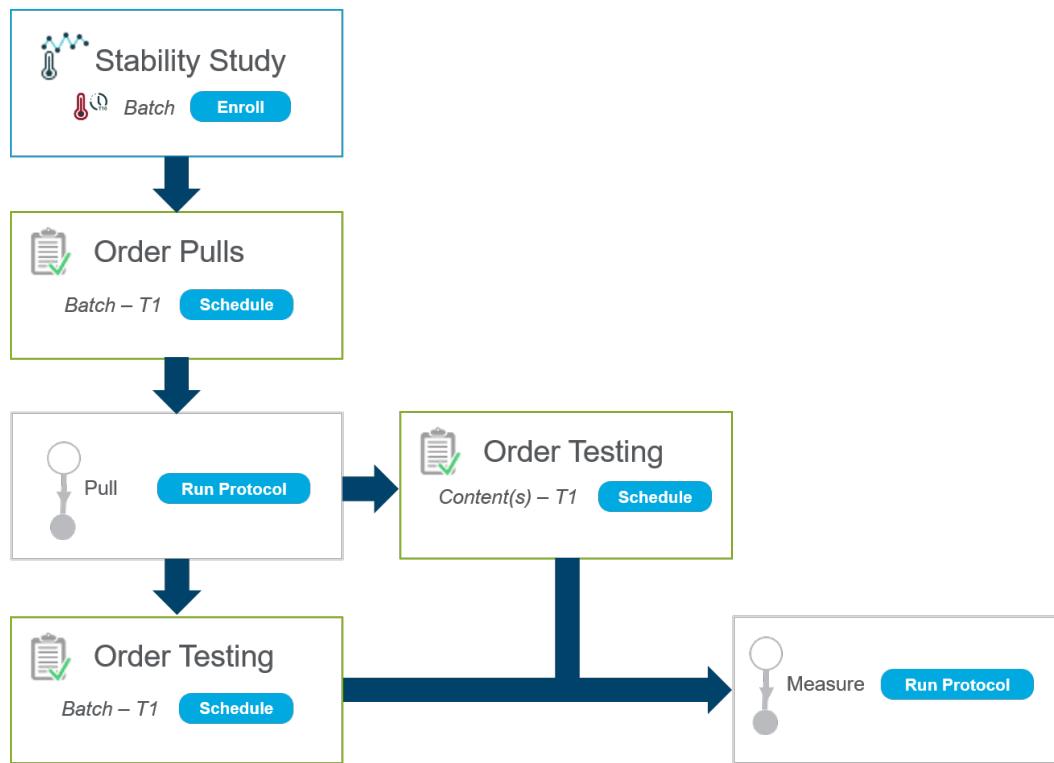
- Status
- Fields for the specification: Operator, Expected Value, Minimum, Maximum
- Value: The entered result value for the specification rule.
- Created On Date/time
- Storage Condition
- Study Name
- Study Version
- Sample Barcode: Could be different from batch barcode depending on the study setup.
- Batch Barcode
- Test
- Stability Start Date
- Timepoint
- Pulling Order ID
- Measurement Order ID
- Workflow Icon: Provides a view of the run in which the selected result was created. By clicking on the workflow icon, if the run is still in progress, the user is redirected to the Workflows module with the protocol run opened. If the run is finished, a pop-up opens with the protocol run displayed as read-only.

### 3.13.1.5. Reports Tab

The Reports tab is available at every level of the stability study tree. A report can be added on a tree level if a report template has been defined for that level. The reports are generated by selecting a stability study, storage condition, timepoint, or enrolled batch and clicking "Generate Reports" on the Reports tab. You can also right-click those levels and access the "Generate Reports" action. Clicking on the Reports tab accesses the previously generated report for the selected level and below (example: You can see reports generated for the selected storage condition and those generated for its timepoints and enrolled batches.). The prior reports provide the information: version, user, creation date, report template, and attached report.

### 3.13.2. Stability Study Order Usage

**Figure 3.15. Stability Study Flow from Batch Enrollment to Measuring**



A batch (content) that has a product assigned that matches the stability study's product can be enrolled in that stability study. When pulling and measurement orders are being used for the study and a batch is enrolled, it causes a pulling order to be created and set to "Registered" for each timepoint in the Orders module. A measurement order is also created for each timepoint and set to "Pending." The output of the pulling order will be used as input for the measurement order upon its completion.

The Orders grid can be edited to add the columns "Stability Study," "Storage Condition," "Time-point," and "Batch" to see which the orders are linked to.

Pulling and measurement orders cannot be edited like other order types. Because of their automatic operation, they are generated based on the setup of stability study design. Therefore, users cannot link or unlink content from either order type, or add or remove requests for either order type. The order status can be changed like normal so that content can be received, orders can be scheduled, canceled, etc.

The user can schedule the pulling order to queue the batches in the workflow. The amount of pulls needed comes from the setup of specifications in the stability study design, so the user does not need to specify the amount of pulls needed. The right workflow is chosen by the pulling order, so the user can select the content and start the protocol run. The workflow can be completed as long as enough content has been created to fulfill the number of pulls. Otherwise, an error will be provided that there are not enough pulls and the workflow cannot be finished yet. Once there are enough contents, the workflow can be completed and the output

shows in the Orders module. Selecting the original content in the order shows the pulled content in the Fulfillments tab. The pulling order is changed to "Completed" when all queue elements are done or canceled.

When the pulling order is completed, content is linked to the measurement order. The linked content will either be the output pulls from the pulling order or the batch itself. Also, requests are automatically added on the content(s). The requests are the "Specifications to be checked" and are set by the product for each timepoint. The measurement order is updated from "Pending" to "Registered." There are three scenarios for content(s) and request(s) linked automatically to the measurement order:

- **Single content:** When a pulling order is completed with only one content, then the single output content of the pulling order is linked, and all the requests are added on the content.
- **Multiple content, same number of specifications:** When a pulling order is completed with multiple contents and there are the same number of contents and specifications for the timepoint, then all the contents are linked to the order and get a single request each (1:1 content to specification). In this case, the content with the lowest barcode gets the specification with the unique identifier that comes first when sorted alphabetically, and so on.
- **Multiple content, different number of specifications:** When a pulling order is completed with multiple contents and there are a different number of contents and specifications for the timepoint, the contents that check the specifications cannot be automatically decided. So in this case, the batch is linked to the order instead and all the requests are added to the batch so the user can manually decide which get tested.

The measurement order is scheduled to send the content(s) / batch to the next workflow. Either the user chooses the analytical method for the desired workflow and path when scheduling, or accepts the default analytical method that was suggested by the study design. Then, the user can navigate to the workflow, select the contents, and start the protocol run for testing.

### 3.13.2.1. Behavior of Measuring Orders by Themselves

It is possible to use measuring orders by themselves without pulling orders. When the batch is enrolled in the stability study, a measuring order is created for each batch at each timepoint. The measurement orders are set to the "Registered" status. The batch is always linked and all the requests are added to the batch so the user can decide which content to use for each of the tests in a workflow.

The user can schedule the order and choose an analytical method and then begin the protocol run for the analytical workflow(s). On completion, the results are available in the order and the stability study.

## 3.14. Inventory

The Inventory module and the Inventory types module are used together to allow users to manage a lab's inventory, track the amounts that are on hand, and see the usage history of each inventory record.

### 3.14.1. Inventory types Module

Adding an Inventory Type is similar to adding a Content Type, but with a more limited set of parameters. Inventory types also require the "Amount of Material Field" dropdown menu to be

filled with a content field with a quantity datatype. This field tracks the amount of inventory in the system. This field is also used with the "Show warning when amount of material drops below a threshold" checkbox. When this box is selected, an additional field called "Warning threshold" appears where the user can specify an amount. When the amount of material goes below this threshold, the inventory content will be highlighted red in the Inventory Module.

This module requires the following Role Access rights to work:

**Table 3.25. Inventory Type Permissions**

Access rights Category	Access rights Name	Purpose
Setup: Content	Inventory Types	When enabled, the role user can access the Inventory Types module to create the types of inventory with their amounts and warnings when the amount runs below a certain threshold.

## 3.14.2. Inventory Module

This module requires the following Role Access rights to work:

**Table 3.26. Inventory Permissions**

Access rights Category	Access rights Name	Purpose
Routine: Content	Inventory	Allows the role user to access the Inventory module to see the current inventory and the history of inventory items. Also allows the creation of inventory content in the content module, and adds the inventory history tab to the Content module.

The inventory module has three main elements:

- **Left Panel:** Shows all the types of inventory in the system. Selecting one of the types shows all the inventory items of that type in the upper right panel.
- **Top Right Panel:** This is where the individual inventory items are displayed when an inventory type is selected in the left panel.
- **Bottom Right Panel:** Displays the Inventory history tab that shows the usage history of an individual inventory item selected in the top right panel.

## 3.14.3. Content Module Expansion

Inventory records are created in the Content module. Click the "New Content" button, select one of the defined Inventory types in the "Type" field, and then enter a value into the applicable quantity field. If the quantity field used to track the amount remaining (as defined in the Inventory type) is not filled, SLIMS will display a warning showing which field must be completed.

There is also a new tab available in the lower right panel of the Content module called Inventory history. This is identical to the Inventory history information found in the Inventory module. It is only visible if an inventory record is selected.

### 3.14.4. Consuming Inventory

Inventory can be consumed in the following ways:

- **Usage tab in experiment run steps:** If a protocol run step has the Track Usage option selected and a type is specified in the Content Types for Usage field, the Usage tab can be used to consume inventory items.
- **Usage taken from content action:** Found in the content module via the right-click context menu or the selection menu.
- **Usage taken from content macro step type:** The macro step setup menu allows the user to configure the amount, dimension, and the number of uses.

# 4. Electronic Lab Notebook

The Electronic Lab Notebook is a SLIMS tool created to supplement and even replace the paper version of laboratory notebooks. This chapter describes all the setup needed for the electronic lab notebook to be configured, used, and maintained efficiently.

## 4.1. Tests and Test groups

Tests organized with Test groups allow data to be structured in a meaningful way. Both modules are described in this section.

These modules require the following Role Access rights to work:

**Table 4.1. Tests and Test Groups Permissions**

Access rights Category	Access rights Name	Purpose
Setup: Electronic Lab Notes	Tests	This permission provides access to the Tests module in which the role user can structure the various tests that can be measured in a workflow (as in a Results protocol step), entered in the standard ELN, and in the Results tab of the Content module.
Setup: Electronic Lab Notes	Test Groups	This permission provides access to the Test Groups module in which role users can group tests that regularly occur together, and causes the results to be grouped together in a collapsible sub-list in the Orders module.

### 4.1.1. Tests

The Test Module is accessible under the setup menu.

A Test is a way of storing results corresponding to any physical tests that may be run on content in a structured way. It is mainly suggested for results of real tests, like blood group or pH measurements, but can also contain more abstract results. The Tests Module is used to define new tests and is divided into two parts. The list of existing tests is listed in the left panel, and the right panel contains the details of the selected test.

The details needed to define a new test are:

- **Name:** (Required value) The name of the test (mainly used by the system to relate values to the test), which should be kept to a single word.

- **Label:** (Required value) The name of the test as it will appear inside the protocol templates. The label should indicate the usage of the test so other users can easily recognize it.
- **Description:** Description of the test, and an area to provide any desired details.
- **Configure Result Datatype:** (Required value) This is used to define the datatype of results created from this test. For instance, a datatype could be Short text, Quantity, Date, or Fixed choice. Further details on the different datatypes and corresponding parameters are given in the Datatype section.
  - **Value Expression:** After clicking "Configure Result Datatype," the Value Expression is available. Groovy script can be added that will be used to return a result value for this test. The groovy evaluation works as it would for a custom field for the Result table, and then calculates the result value (rslt\_value). This allows a calculation based on other custom fields of the current result, or a calculation based on values of other tests not linked to the same run step.
    - ① The result value datatype returned by the groovy must match the test datatype.
- The groovy can also be a calculation depending on other tests. The tests on which the value expression depends need to be specified in the "Tests" multiple choice list. Examples of such scripts can be found in Section 11.4.11 and Section 11.4.12.
- **Result Unique Identifier mask:** Generates a root name and incremented number based on the mask RSLT##### for each result that is generated from the test, or a mask customized by the administrator. See the Name mask (use # for numbers) definition for more details.
- **Default Protocol:** A protocol can be selected that should be associated to the test by default.
- **Test groups:** Select a test group to add it to other relevant tests. This allows the results of the group to be viewed in a collapsible list. A test group can contain more than one test, but the Test Groups field can only have one group defined in it.

(ex: The Blood PH test and the Cell Count test can each have the "Blood Tests" group defined so the results will be viewed in the same list.)
- **Only for Selected Content Types:** Can be used to restrict the tests to particular content types.
- **Content Types:** A list of content types that are allowed to be tested with this test.
- **Dynamic:** This allows a test to contain multiple results over time for a single content.
- **Format Observations:** This can be used only with dynamic tests to format the kind of observation point (date, date and time, time, or your own text).
- **Behavior Observations:** These can only be used with dynamic tests and includes three options. Dynamic adds observations at the time the test is run, Series indicates the sequence to be given, and Interval is the difference between observation to be set up.
- **Adding Results Possible in Orders:** If checked, then the customer is allowed to enter a value for the test before sending the order.
- **Internal:** If checked, then the test is not visible in orders.

### 4.1.1.1. Dynamic Tests

Dynamic tests can be created by checking the Dynamic box in the test form. These type of tests consist in a series of measures over time. (Also called *Time-Based Series*.) During the execution of the experiment run containing this test, the Date and Time have to be defined to 'start observations.' (If no such field is present, the step's start date is used as the 'start observations' date.)

If Dynamic is checked, 'Format Observation' and 'Behavior Observation' parameters have to be set. 'Format Observation' represents the format of the label you associate to each measure, and can be a date, date and time, time, or text. If "date" is chosen, when a new measure is added to the test, the current day will be associated to it. If the test consists of more than one measure per day and spans several days, the format "date and time" is preferable. If the test lasts less than one day, the format "time" is the most accurate one. Finally, if the labels need to be of a non-date format, "text" is the right option.

- If a numeric Format Observation (date, date and time, or time) is chosen, a "Dynamic", an "Interval" or a "Series" Behavior Observations can be chosen. With the choice "Dynamic," no constraint is added on the date/time when taking measures. This is the simplest and most general choice. If the choice is "Interval" or "Series," observations will correspond to specific dates/times.
  - With the Behavior Observation set to "Interval," the interval between observations (in minutes for Time, and in days for Date) and the Amount of observation are asked. For example, if the value for "Start Observation" date is March 14th, the Interval is 5 and the Amount is 3, the test contains a result for today, the 19th, and the 24th of March.
  - With the Behavior Observation set to "Series," a series of intervals between observations is requested. For example, if the value for "Start Observation" date is August 1st, and the Series is 2,4,5, the test will contain three results corresponding to observations for August 3rd, 5th, and 6th.
- If the Format Observation is text, only a "Dynamic" or a "Series" Behavior Observation can be chosen. The "Dynamic" case corresponds to no constraints, as in the date case. The Observation label has to be set each time a new measurement is added. If the Behavior Observation is set to "Series," a series of comma separated labels that each correspond to one observation have to be set during the creation of the test.

### 4.1.1.2. Test Results

Results represent the actual values obtained from a test for a sample. For instance, a test defining the blood group would have a fixed choice as values: "A," "B," "AB," or "O," and each sample that this test was performed on will get a result record storing the sample's actual blood type.

Result records can be entered into the ELN or a workflow by executing a protocol containing a run step of type "Result." During the creation of steps of "Result" type in a Protocol, a dropdown list of the defined "Tests" is available where one or multiple tests can be selected for the same step. Results can be configured so they are added automatically or manually and associated to the test in a result step.

In the ELN or workflow, Result Protocol run steps have two more tabs, "Sample Results" and "Results Overview," in order to enter result records for the samples attached to the run step. More details about these tabs are available in the *Protocol Runs Section*. If the Result step

contains tests of the "Dynamic" type, a field to enter the date when to Start Observations will be required (in the General Info tab), and an additional tab with the name of the test will be available where multiple Result records for the dynamic test can be entered.

### 4.1.1.3. Import and Export Tests

Tests can be imported and exported with Excel in the Tests module. The actions are available in the gear menu.

Tests can be exported by selecting the "Export" action in the gear menu, and exports all of the tests in the module. A more complex export can be done by checking the "Advanced Export" box. This allows you to select the Format, File Name, and the desired Columns that should be exported. The datatype configuration values are not exported with tests. If the user wants to export/import (as though they were performing a 'copy' function), he/she would have to combine the GUID of the records from the export with the datatype configuration fields that are present in the template.

Tests can either be imported from a previously exported Excel file, or from an edited Excel Template File. The "Create Template File" action can be accessed from the gear menu. The import adds all fields for the datatype configuration regardless of datatype so that the user can create tests with different datatypes in the same template file. The advanced export allows you to select other tables from which to display fields.

ⓘ Importing many-to-many links does not work for Tests.

The datatype configuration is similar to those for Fields, so the section on importing fields and the section about datatypes for fields are helpful references.

Additionally, if the user has difficulty finding an accepted value for a field, he/she can find the expected value for a field by exporting a new or existing field with that value in the Fields module.

### 4.1.2. Test groups

Test groups provide the ability to regroup tests that are used inside a global analysis in the Order module. When configured, this groups similar results into a collapsible sub-list. For example, if a blood analysis is to be performed, the blood group and the blood pH could be part of the testing. In the system, these two tests can be regrouped into a test group named Blood Tests, and the results will be displayed grouped in a collapsible list.

The left panel of the test group module contains a list of all the available test groups, and the right panel contains details about the selected test group. To create a new test group, click on  (in the top left panel) and fill out the pop-up by choosing a name, a short description, and selecting whether to group results or not. Individual tests are not added to a group within the Test groups module, but in the Tests module. Once the created tests are assigned to a group, the Main Test field will have the tests that are part of the group available so that a main test can be selected if desired.

## 4.2. Instruments and Instrument types

Instruments (and their definition within Instrument types module) allow users to digitally reproduce what was done in the lab by specifying the instrument was used in performing an experiment.

The following Role Access rights can be configured for use in these modules:

**Table 4.2. Required Permissions for Instrument Modules**

Access rights Category	Access rights Name	Purpose
Setup: Electronic Lab Notes	Instrument Types	This permission provides access to the Instrument Types module in which role users can standardize the parameters for instruments that have the same settings.
Setup: Electronic Lab Notes	Instruments	This permission provides access to the Instruments module in which role users can add the specific instruments in their lab that need to be tracked.

## 4.2.1. Instruments for ELN

Instruments are part of the Electronic Lab Notebook in the setup menu. Experiments can be run directly on an instrument.

An instrument is defined by a name, a description, an instrument type, a user, a group, and any custom fields added to it. A location can be added to the instrument once it has been saved.

There is a particular way of executing Protocols on an instrument. First, if the protocol requires an instrument, the protocol won't be available in the list of Protocols inside Projects, but will instead be available inside instrument, and vice-versa. To be able to run an experiment, or part of an experiment, inside a Project, a sub-protocol step that requires an instrument is needed.

## 4.2.2. Instruments for Workflows

Instruments have expanded options when used with Workflow protocols. Administrators are able to require the use of Instruments and control the consequences of the calibration status the instrument can have in a protocol run. Users are able to link an instrument they used to a protocol run step.

This is done by configuring a few other features. The instrument status can be created in the Statuses module (the default statuses "Available" and "Unavailable" are provided for instrument calibration, but more can be added if these are not sufficient). Additionally, the protocol run can employ any flags created for that purpose in the Content Event Types module for uncalibrated instruments. When building the protocol steps, the options to prohibit or allow the use of uncalibrated instruments, to warn users, to flag content that has been treated with uncalibrated instruments, and track the status in the History are all available.

See the Protocol steps section for the details on how to include an instrument in a protocol step. See the Instrument Calibration section below to see how to calibrate instruments.

### 4.2.3. Instrument types

Instrument types are available to group all instruments of the same type together that have identical parameters. To create a new instrument type, click the "New" button in the Instrument Types module, choose an icon, a name, a description (optional), and define the type of location that can hold the instrument. A newly added instrument type has "Can be calibrated" disabled by default, and its status set to "Available." Define whether the instrument can be calibrated or not with the "Can be calibrated" option. This will allow the calibration status of the instrument to be tracked in the instrument history and for the administrator to be able to assign consequences if an instrument is not calibrated.

*Note: It is recommended to create a location type that is specific to the instrument type, and that has a similar name as the instrument type. The instruments will not appear inside the location tree, but are attached to it, so doing this makes the location tree clearer for showing where instruments are kept.*

### 4.2.4. Instrument Calibration

When the "Can be calibrated" field is enabled for an Instrument Type, the calibration status can be reset and will be tracked as a calibration history event.

Calibrating an instrument is done in the Instruments module. It consists of these fields:

- **Status:** These are the Instrument Statuses created in the Statuses module. "Available" and "Unavailable" are the two statuses created by default, but any further desired statuses can be created and visible in this dropdown.
  - ① This field is not updated automatically by SLIMS. Once the instrument is calibrated, the user will need to update the status manually and save the page.
- **Calibrated:** Displays as Yes or No. This field is read only and is updated automatically when the instrument is calibrated.
- **Calibration expiry date:** This field is read-only but is updated by clicking the Calibrate Instrument button. SLIMS checks once a day to see what instruments have expired, and if the expiry date is passed, the calibration status is updated to Uncalibrated (Calibrated: No).
- **Calibrate Instrument:** When this button is clicked, a pop-up appears to provide a date selection dropdown and calendar icon to allow for easy date selection. The date must be in the future. Click "Calibrate" to set the new expiration date or "Cancel" to cancel calibration. Once Calibrated, the pop-up closes, "Calibrated" is set to Yes, and the expiry date is updated to the selected one.

The job that checks for expiration runs at midnight, so the expiry date is the 'first day the instrument cannot be used' and not 'the last day the instrument can be used.'

The History tab records each calibration event and collects: Date, User Full Name, Username, Status, New Expiry Date, and if they're relevant, Can have a run (added via API when calibration is done as part of a protocol and not using the calibrate button) and API (SLIMS GATE) to add calibration history events including the link to the runstep.

## 4.3. Protocols

New protocols can be created in the Protocols Module. Protocols are standard procedures, or the series of steps that it takes to complete an experiment. The purpose of protocols are to predefine the most common experiments used by lab technicians and researchers to simplify their work. Protocols can be modular by using Ad hoc protocol steps, and can be used to automate some steps in an experiment.

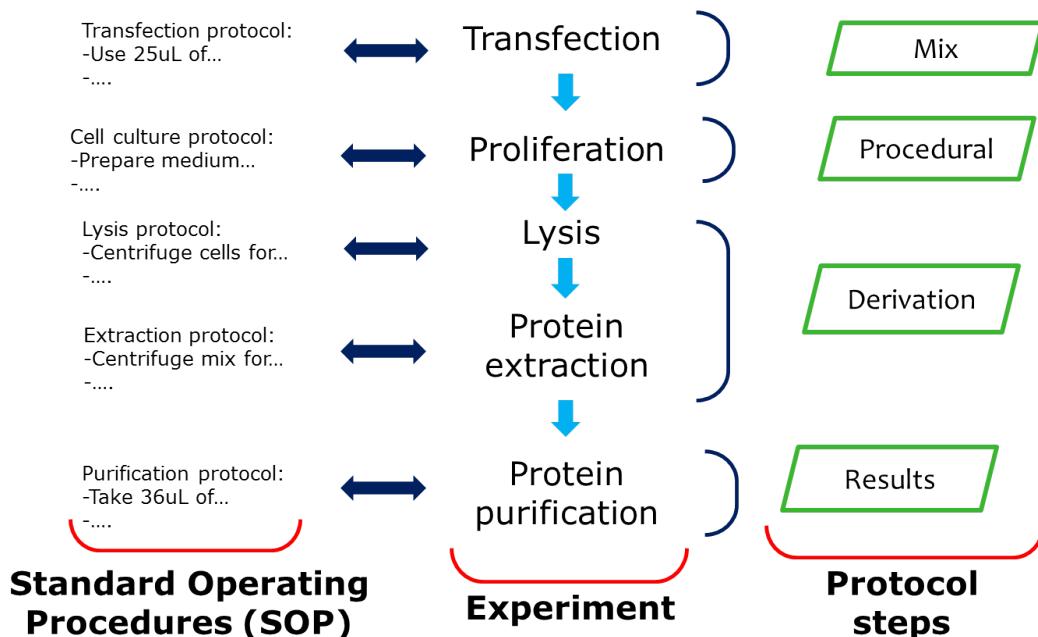
The following Role Access rights can be configured for use in this module:

**Table 4.3. Required Permissions for Protocols Module**

Access rights Category	Access rights Name	Purpose
Setup: Electronic Lab Notes	Protocols	This permission provides access to the Protocols module in which the role user can create standard protocols that can be used in the experiments of an ELN experiment run.

Figure 4.1 shows an example of how an experiment used in a lab can be structured in SLIMS. The protocol steps available in SLIMS can be used to match the individual parts of an experiment to digitally track what was done in the laboratory.

**Figure 4.1. SOP and Protocol in SLIMS**



Comparison between a real-life experiment and the type of Protocol Steps that represent it in SLIMS.

The Protocols Module allows existing protocols to be copied in full by right-clicking on a protocol and using the dropdown menu. The module is divided into three parts:

- **Left Panel** : Contains the list of all existing protocols.
- **Middle Panel (Hidden by default)**: This panel is only present if the 'Show All' button is clicked next to the Versions dropdown. Otherwise, the panel is not present, and the versions are shown instead in the Versions dropdown. The versions are retained, but hidden by the 'Hide All' function of the button to save space. To see the configuration of a different version, the desired version can be selected in the dropdown. The panel shows all versions of the protocol selected in the left panel. The versions are kept to ensure that the history of changes are retained, especially when a protocol is run on a version before the current one was updated. (See the *Versioning Section* for more details.)

In Protocols, the Version button is always available but the color of the button changes to yellow when any change is made to indicate that there are unsaved changes present. Versioning the protocol changes the color back to white. This includes when Content Event Type flag restrictions are placed on a protocol - the button changes to yellow in Protocols to indicate that unsaved changes have been made, and the protocol should be versioned.

- **Right Panel**: This area is where the protocol setup is done, and contains multiple tabs on the top. The tabs (Protocol, Protocol Steps, Ad Hoc Protocol Steps, and Sign Off Step) are detailed in the sections below.

### 4.3.1. Protocol Tab

- **Protocol**: General details of the protocol containing the following fields:
  - **Name**: The name of the protocol should clearly indicate its usage so that other users could easily understand it, especially if there are more than one protocols involved in the same kind of experiment.
  - **Protocol Type**: Default or OpenLab Sample Scheduler Execution. Most protocols use the Default type. To learn to set up an OpenLab Sample Scheduler protocol for use with Agilent OpenLab Shared Services, see the *Sample Scheduler for OpenLab* section.
  - **Description**: Provide any desired description and details about the protocol.
  - **Experiment Run Unique Identifier Mask**: Generates a root name and incremented number based on the mask XPRN##### for each protocol run that is generated, or a mask customized by the administrator. See the Name mask (use # for numbers) definition for more details.
  - **Instrument Required**: Check this option when an instrument is required for the protocol. This means that the protocol will not be available in the dropdown list of Protocols when starting a new experiment inside a Project, but only when starting a new experiment on an instrument. In order to use the instrument inside a Project, create two Protocols: the first one should contain all the steps that are specific to the instrument, and the second contains the whole procedure and a step of type 'sub-protocol' using the previously defined protocol.
  - **Instrument Type**: Specifies which type of instrument is used for the protocol.
  - **Execute steps in correct order**: Select this if it is required to execute all the steps of this protocol in the correct order. It will not be possible to start a step before the previous steps are finished.

- *Validator roles*: Choose which roles can validate results defined in the protocol. Validation of results can be done outside the context of an order, so if a role has the potential to validate results, it can be selected from the dropdown. This does not require the order status transition from 'Completed' to 'Validated' to be enabled.
- *Automatically create a protocol run when creating an experiment*: If this option is checked, the user does not need to manually create a protocol run when creating an experiment for the first time.
- *Automatically open the first protocol run step when creating a protocol run*: If this option is checked, the user is automatically redirected to the first protocol run step when a run is created.
- *Active*: Check the box for the protocol template to be available for usage. Archived protocols are marked as 'Inactive' so they are no longer available in the dropdown list of protocols.
- *Default custom fields for Protocol*: All the custom fields that apply for all the Protocols (no specific protocol chosen).
- *Additional custom fields for Protocol*: Any custom fields that can be added for the protocol.

### 4.3.2. Protocol steps

The Protocol steps tab contains the setup of the protocol steps and their details. The Steps are defined by their Name, Seq no, and Type.

- › **Name**: The name is associated with a specific part of the experiment being performed, like 'Measure PH.'
- › **Seq no**: Defines the order of the step relative to the rest of the steps in the same Protocol. Steps with low numbers will be executed earlier than steps with higher numbers.
- › **Type**: The fields and functionality of a step are different depending on their type. The available step types are as follows:
  - *Link content*: Associates the protocol step with the contents that the experiment is being performed on. This is generally required in all protocols to have traceability between what was done, and what it was done to.
  - *Link inventory*: Used to link inventory for tracking purposes. The linked inventory does not continue to the next protocol step.
  - *Link location*: This is used to associate locations that are involved in the ELN experiment or workflow protocol. This is generally used to get a graphical representation of the location, not to link the contained contents. This step is typically used after a step which links contents, such as a Link content, Derivation step, etc., so that the locations for the contents can be found. The option "Automatically link locations of linked content on start of step" is available. When enabled, all locations of the linked contents are displayed in the Locations tab on the step, and a tab named for each individual location is created that shows the location details. If multiple linked content share the same location, then the location is only linked once.

A macro with a "Create Location" step can be used in a workflow protocol as an alternative to do something similar. This macro can be selected in a link location protocol step and will automatically link the created location to the step. If the macro starts with a Create Location step *and* any of the following macro steps require the selection of content (such as move content or derive content), this macro will be shown as a selection macro on Content, and not at the top with the other Create location macros. It will still automatically link the location to the step.

- *Link order*: This associates the orders that are used in the experiment with the protocol step. It is useful to group multiple orders together and execute them inside one single protocol run.
- *Link reagents*: This associates reagents that are used in the experiment with the protocol step. It is useful to link the reagents to a step for traceability and to provide the reagent context to the linked samples. Reagents are no different from other content types outside of context, and thus any type of content can be linked to a Reagent step. Users can link any content type, including types that do not have an "Amount of material" field default definition, and the content types can be restricted by configuring additional options on the Link reagents step. There are usage options available specifically for reagents with "Amount of material" fields, and content without that field cannot have their usage tracked in this step. The reagents are not transferred to the next step because reagents assist in the experiment, rather than being the subject of the experiment.
- *Procedural*: Any actions that can be performed manually in the lab without the help of SLIMS, but that should still be tracked in the workflow of an experiment. For example, the execution of an SOP that should be tracked, for which there is not a test with appreciable results.
- *Derivation*: A step that allows the linked contents to be derived (See the *Derive Content Section*).
- *Mix*: A step that allows the mixing of linked contents (See the *Mix Content Section*).
- *Result*: This step captures the results of a test. Tests can have multiple purposes; a simple test, where the values just need to be stored in SLIMS, or a more complicated one, where the result value allows data collection about the sample at the same time. (See the *Results Section*)
- *Link content and Add Results*: This step allows the execution of a procedure (available tests) that is predefined according to the needs of the laboratory using SLIMS. This can be a simple computation that creates a result given the specification of the contents.
- *Conclusion*: Conclusion of the experiment. It is usually a note into which the conclusion is written.
- *Sub protocol*: This step allows a predefined protocol to be selected and executed as a sub-protocol in this step (See the *Sub protocols Section*).
- *Variant observation*: Variant Tests are the only tests that can be included in this step. When running a protocol in the ELN with such tests, no results are automatically created for Variant tests (See the *Variants Section*).

- › **Description** : An instruction or description of what the user should do in SLIMS during this step (derive contents, enter results, etc.).
- › **SOP (Standard Operating Procedure)** : Selection of an SOP to follow during this step from the list of SOPs created in the *SOPs Module*.
- › **Executor of this step needs to be different than previous step**: If checked, it indicates that the executor of this step should differ from the executor of the previous step (the user that finishes this step shouldn't be the same as the one finishing the previous step).
- › **Executor of this step needs to be different than all previous steps**: If checked, this indicates that the executor of this step should differ from all of the previous steps (the user that finishes this step shouldn't be the same as the one finishing any of the previous steps).
- › **Electronic signature is needed to finish or to reopen the step**: If checked, this indicates that the user will have to enter their credentials again in order to finish the step. Depending on the lab settings of the SLIMS instance, the user may have to enter their username, password, signature meaning, and a comment. When using an external authentication provider, the password is not asked for in the pop-up in SLIMS, but the username, signature meaning, and comment may be asked. Instead, a button is provided to "Sign in with the authentication provider" which opens another tab to perform reauthentication. The password and username, depending on the authentication provider, are required to sign. More details are provided in *Section 2.1.2*.
- › **Only for selected content types**: Only content of the selected content types can be linked to the protocol. Contents inherited from the previous step are not validated against the restriction (they will be inherited even if they are of a *restricted* content type).
- › **Default Rules**: A read only field that displays the mix rules that were automatically applied to the mix protocol step. Any mix rules that are unrestricted in the rule's configuration are listed as default mix rules ("Only applicable in selected mix protocol steps" parameter is disabled).
- › **Additional Rules**: A multi-select list which allows additional mix rules to be applied to the mix protocol step. The mix rules that are available in the dropdown were created in the Rules module and have the "Only applicable in selected mix protocol steps" parameter enabled. If a rule has "Only applicable in selected mix protocol steps" enabled and also is selected in the rule's multi-select list, it appears in Additional Rules automatically.
- › **Only selected roles can execute**: It is possible to define which roles can start, finish, and reopen a step.
- › **Automatically start this step when previous step is finished**: If the box is checked, then this step will start as soon as the previous one is finished. If it is unchecked, the user will have to click the start button to initiate the step.
- › **Restrict and specify extra options about macros**: Only the selected Macro will be available in this step. In addition, the Macro can be started automatically as soon as the step is reached by using the "Run on Start of Step" box. The order in which Macros are executed can be set using the sequence number.
  - ① By default (if this option is not selected), all Macros will be available during the execution of the Protocol step.
- › **Specify extra options on linked reagents and usages**: This option is only available for Link Reagents steps. It provides access to several other restrictions and configurations for linking reagents and input content during the protocol step. Reagents have usage options that

are different from the "Track Usage" and "Content Types for Usage" options that exist on other types of steps. When this option is enabled, usage options for both reagents and input content can be specified.

- *Reagent content types*: This list allows you to choose the reagent content types for the step, and define their usage and features. For each of the reagent types selected, there is a row of options:
  - *Description*: The description for each reagent content type.
  - *Required*: The link reagent step cannot be finished unless at least one reagent of this content type has been linked. During execution, the user will be prevented from finishing the step by an error message that explains which reagent/s are missing.
  - *Usage Tracking Required*: When checked, prevents the step from being finished if a usage value for the reagent or input content is not entered by the user or by SLIMS.
  - *Default Usage*: The default usage is calculated automatically if you fill in a groovy value expression. Default Usage value expression will return the default amount of material to be subtracted from the linked reagent or input content. This is a text field that can be edited to add a small groovy.
  - *Groovy Console Button*: If a longer groovy calculation needs to be entered, the button next to the Default Usage column pops out the groovy value expression editor box. There are additional completion functions available in the groovy editor for Default Usage expressions:

$N$  = The number of linked contents that are linked to the step as Input (not the reagents).

Content = the linked reagent.

experimentRunStep = The run step for which the usage is added.

ⓘ See some use cases for Default Usage expressions.

- *Default Unit*: The unit associated to the default usage amount. It comes from the default unit defined for the content type.
- *Only allow linking selected content types as reagents*: When enabled, this option restricts the content types that can be linked as reagents to those selected in the "Reagent content types" list. In execution, the error message will show up if the user attempts to link or create a reagent type in the step that is not allowed.

Kits can be linked to a Link Reagents protocol step as well, including their Kit components. This depends on the allowed Content types, however. If a user links only the kit, the components will be linked with it, unless some of the component Content types are restricted. The components that are not allowed will not be linked. If a kit contains no components that can be linked but the kit type itself is allowed on the step, the kit will be linked without its components. If the kit type is not allowed but all of its components are, the kit will be linked along with its components anyway. And finally, if the kit type and none of its components are allowed, none of them will be linked.

- *Create a default usage instance for each content type at start of the step:* If enabled, the default usage that you have defined for each reagent or input content is automatically filled in at the start of the step.
- › **Allow linking of expired reagents:** This option allows you to decide what should happen when an expired content is linked to a Link Reagents step. Expiration Date checking is done when the content is linked to the step. If Allow Linking of Expired Reagents is enabled, users will be able to link expired reagents, and the below configuration options appear. If this option is disabled, the user will get an error every time they try to link an expired reagent. The expiration date referred to for this option comes from the Content Types module. First, a field with either the "Date" or "Date and Time" datatype can be created on the table "Content." Then it can be applied to the reagent's category or content type as the expiration date.
  - *Warn the user when an expired reagent is linked:* When this is enabled, a warning is displayed to the user when they link an expired reagent. The user must confirm the link, but the warning does not prevent them from completing the link action.
  - *Add the following flag(s) if an expired reagent is used:* When enabled, the Contents linked to this step as Input will be flagged when the step finishes. The flag listed in the next field "Flags" will be applied to every content. The following option is shown when using flags:
    - **Flags:** This lists any of the Flag content event types that can be applied to contents used with expired reagents. Select the desired flags from the list to add them to the step so they can be applied to content used with expired reagents.
- › **Minimum content status:** Defines the minimum status a content must have in terms of the status sequence number in order to be linked to this step. (This option is only present for steps where link content is possible: Link content, Link order, Derivation, and Link content and Add Results). It does not check contents inherited from a previous step.
- › **Content transferred to next step:** (This option is only present for steps of type Derivation, and Sub protocol.) This allows contents to be selected that will be used for the next steps. By default, it is 'Output,' but it can be changed to 'Input,' 'Input and output,' 'Input if no output available,' 'Output if no input available,' and 'Output and unused input.' The last option allows for any unused input to proceed to the next step with the output from any used input.
- › **Test:** (This option is only present for steps of type Result and Link content and Add Results.) Identifies which tests should be linked to the step. For each test selected, the following options can be configured:
  - **Allow Creation of Manual Results:** This option can be enabled by itself or in conjunction with Automatically Create Results. If this option is enabled, results can be manually added for this test for each linked content. This option cannot be edited for dynamic tests and will be enabled by default. If this option is used in the context of an order requesting results for this test (with the Order Management Order Type type), Result Rules or Products and Specifications defined for this test will not apply to any result created manually.
  - **# of automatically created results:** A number can be set by itself or used in conjunction with "Allow Creation of Manual Results." The value is set to "1" by default to create one result automatically for this test at the start of the step for each linked content and for each selected test in the protocol definition. The option should be set to "0" if the test should not create any results automatically at the start of the step. Otherwise, the value can be

set to a number higher than 1 to create the desired number of results automatically (for instance, to measure the weight of a content 5 times and then take the average result).

ⓘ The # of automatically created results can also be set on specifications (for Products and Stability Study Design), so SLIMS will create the higher number of requested results for the test on the content in the protocol run. Take this into account when designing the number of results.

- **Aggregation Methods:** Only applicable to tests with numeric result values. Automatically calculate the "Standard Deviation" and/or the "Mean" over all the requested results, including the automatically and manually created results. The aggregate is calculated over the results for the same content, such as a batch (a collective content, ex: 5 samples from the same box). Enabling one or both aggregation methods creates a result row for each method that is read-only to the user so SLIMS can aggregate the values. The aggregations update as the user inputs the other results.

An aggregation method can be set additionally on specifications (for Products and Stability Study Design), so SLIMS will create an aggregated result row for each requested method. As an example, if a timepoint specification rule for a test requests 5 results and an aggregated mean, and the results step requests 3 results and an aggregated standard deviation, the protocol run will have 7 results created: 5 for the user-entered values, a row for the mean, and a row for the standard deviation.

ⓘ Keep in mind that the aggregates are calculated over all the results for the protocol run step when designing the protocol step. In the example above, the standard deviation applies to all 5 results, not just the 3 requested by the result step.

If manual results are enabled and the automatic result value is set to 1 for a test, SLIMS will create one result automatically for this test for each linked content, and the technician executing the protocol step will also be able to add new results for this test manually to any of the linked contents. If manual results are disabled and the automatic result value is set to 0 for a test, SLIMS will not create results automatically and the button to add new tests manually will not be present. It is still possible to create results using other means (like with a SLIMS GATE).

- › **Flows:** (This is only available if the option "Can execute SLIMS GATE flows" is enabled.) This option makes it possible to select which flows should be available when executing the step.
- › **Track Usage:** This option identifies whether some content's usage should be recorded or not (mainly for solvents or consumables). If this option is enabled, the 'Add usage instance (with defaults) at start of step' option is added as well as a table containing the content types to be tracked. In this table the user can define:
  - A default usage, useful in particular when 'Add usage instance (with defaults) at start of step' is true. This value will be used as default in all usages.
  - Whether tracking on the content is required.
- › **Add usage instance (with defaults) at start of step:** This option identifies whether a 'first usage instance' will be added when the run step is started with these defaults already filled in. Even contents that a user adds later on will already have the default usages filled in.
- › **Restrict and specify extra options about macros:** This indicates whether a Macro can be used in the step. Once a Macro is selected, two more options become available:

- *Run on start of step*: Used to start the Macro with the start of the step.
- *Skip inapplicable content types*: Ensures that the Macro is only applied on contents of content types that were not of a selected restricted type.
- › **Attachment 1, Attachment 2, Attachment 3**: Three attachment type fields. If a file is defined as a value of one of these fields, that file will be available in the Electronic Lab Notes as part of the step definition.
- › ⓘ The following information relating to Instruments is only relevant for Workflows.

**Requires an instrument**: Is used to specify that the protocol step in the Workflow requires an instrument to be completed. Only one instrument can be linked per run step. A protocol run step that requires an instrument cannot be finished unless an instrument is selected. When enabled, these configuration options appear:

- *Only for selected instrument types*: Allows selection of the available instrument types to restrict the list for the user.
- *Instrument types*: A dropdown list of all the configured instrument types in SLIMS.
- *Allow selection of uncalibrated instruments*: When enabled, this allows users to select instruments that have and that have not been calibrated. When disabled, this issues an error if a user tries to finish the step after selecting an instrument that has the "Uncalibrated" calibration status. The instrument selection dropdown the user sees during this step will display a list of all instruments with their name and calibration status displayed to make it easier for the user to select one in the appropriate status.
- *Warn the user (s)he is using an uncalibrated instrument*: When enabled, this displays a pop-up warning message that the user has selected an uncalibrated instrument when they try to proceed after selecting an instrument with the "Uncalibrated" status. When disabled, the user does not get a warning and can proceed to the next step regardless of the selected instrument's calibration status.
- *Add the following flag(s) if an uncalibrated instrument is used*: When enabled, this provides a list of flags that can be added to the run step contents when finishing the step if the instrument used was "Uncalibrated."
- *Flags*: A selection of the flags that were created in Content Event Types. If any flags are desired specifically for this case, then they can be created first in that module.

### 4.3.3. Sub protocols

Each protocol can contain an execution of a second protocol as one of its steps. This is referred to as a Sub protocol. To use this configuration, a Protocol run step of type 'Sub protocol' can be defined and the protocol to execute in this step has to be selected.

ⓘ If multiple Protocols are selected in one Sub protocol step, the user will be able to select which protocol to use during the execution in the ELN.

In addition to the common fields for a step, the following fields are available for Sub protocol steps:

- *Sub protocol*: Contains a dropdown list of available Protocols to choose the one that should be executed in this Protocol step.
- *Multiple sub runs possible*: This checkbox allows the protocol to be run multiple times during this step.
- *Sub run sharing possible*: This option allows an existing Protocol run to be used as a sub run, and therefore multiple runs can be performed together. This is useful to run an experiment in batch mode. △ Modifications made to this Sub protocol run will be reflected in the original Protocol run, and at the end, all contents are pushed back to their original run.
- *Automatically create a new subrun when scheduling this step*: If this protocol step has only one Sub protocol as a choice, SLIMS will automatically create a new sub run when this step is started in the ELN. Disabling this option only becomes available if one of the options 'Sub run sharing possible' or 'Multiple sub runs possible' is enabled.
- *When repeating a run of this protocol, also repeat the used sub protocol*: By enabling this option, SLIMS will automatically repeat the run of this sub protocol if the protocol run is repeated.
- *When repeating a run of this sub protocol, also repeat this protocol*: By enabling this option, SLIMS will automatically repeat the protocol run if a run of this sub protocol is repeated.
- *Select all samples by default*: If checked, all of the contents linked to the previous step are selected.

#### 4.3.4. Ad hoc steps

New Ad hoc step can be created using the Ad hoc protocol steps window. Ad hoc steps are used to complete a protocol during its execution.

For example, if a test is not automatically performed, a result step corresponding to this test can be created, and then it can be added during the Protocol run. Ad hoc steps are created the same way as normal steps.

#### 4.3.5. Final electronic signature step

The Electronic Signature is required on a Final electronic signature step and requires the user to enter their password to sign the record electronically. Re-entering the username, selecting a meaning for the signature, or entering a comment may also be required depending on the choice of Lab Settings. The Electronic Signature is saved in the History along with the event that occurred, and that is linked to the change of the record.

A protocol can end with a Final electronic signature step which defines its behavior once the protocol has been finished. There are two behaviors available for a Final electronic signature step:

- Block reopening of other steps when final electronic signature is done.
- Reopen final electronic signature step when other steps are reopened

Other options are available with the same behavior as for standard steps (refer to Section 4.3.2):

- Executor of this step needs to be different than previous step

- Executor of this step needs to be different than all previous steps
- Electronic signature is needed to finish or to reopen the step
- Automatically start this step when previous step is finished
- Only selected roles can execute

If results were present in the protocol, the sign off step can behave in the following ways:

- *Automatic validation on sign off*: All verified results will be validated automatically when signing off the protocol run.
- *Manual validation required before sign off*: All results are required to be validated or cancelled before it is possible to sign off the protocol run.
- *No automatic validation and no manual validation required*: Validation of results is not handled by sign off.

## 4.4. Simplified Protocol Management

Simplified Protocol Management allows a user to define Simplified protocols. This module is accessible within the setup menu. Users create new Simplified protocols using the new button.

The following Role Access rights can be configured for use in this module:

**Table 4.4. Required Permissions for Simplified Protocols Module**

Access rights Category	Access rights Name	Purpose
Setup: Electronic Lab Notes	Simplified Protocols	This permission provides access to the Simplified Protocols module in which the role user can create simplified protocols that can be used in the ELN. Simplified protocols can be built from a selection of block types with simplified UI.

A simplified protocol has the following fields:

- **Name**
- **Description**
- **Restrict Block Types**: When disabled, all block types are available to choose from in the simplified protocol. When enabled, the following option is provided:
  - *Allowed Block Types*: By default, Note, Attachment, Link Content, and SOP are listed, but any simplified protocol block can be added or removed from the list of allowed block types.
- **Active**
- **Open Blocks by Default**: Defines whether blocks are open (when button is enabled) or closed (when button is disabled) by default.

- **Audit Fields**

A simplified protocol also contains a number of blocks. The blocks have the following fields:

- **Seqno:** To define the order in which the blocks are presented.
- **Name**
- **Type**

The type has seven possible values:

- **Note**
- **Link Content**

It is possible to restrict the Macros that can be executed on the contents that are linked to this kind of block.

If the **Restrict and specify extra options about macros** option is checked, then it is possible to select the Macros that will be available in this block.

For each selected Macro, it is possible to select a Destination block. This Destination block is the block where the contents produced by the Macro (e.g a Derivation Macro) will be linked.

① By default (if the option is not checked), all Macros are available.

- **Attachment**
- **SOP**
- **Link Reagents**
- **Use a Timer**
- **No Predefined Type**

Users can add, modify and delete the blocks. To see how each type of block is used, refer to Section 4.7.3

Simplified protocols have the same version control as normal protocols. When a change is made, saving the version updates the protocol to the latest set of changes, but the previous versions are retained for protocols that were started prior to the update. The version button is always displayed, but is more prominent (the color changes from white to yellow) when modifications have been made.

A sign off step can be set for simplified protocols just as it can for standard protocols.

A simplified protocol can end with a Final electronic signature step which defines its behavior once the protocol has been finished. It is possible to define two possible behaviors for a Final electronic signature step:

- Block reopening of experiment when signoff is done.
- Unsign experiment when it is reopened

The option "Only selected roles can execute" is available (refer to Section 4.3.2).

## 4.5. OpenLab Software Suite Integrations

Agilent's OpenLab Software portfolio offers a suite of products that work together to cover an analytical workflow from the moment the request is generated until the data is archived. SLIMS has integrated with several OpenLab products and add-ons to support the submission of samples for analysis through the core functionalities of SLIMS. Retrieving results after they have been analyzed is not a primary aim through core functionalities, but is possible through plugin customization.

These areas have been targeted for analytical integration so far:

- OpenLab Shared Services allows users to authenticate in SLIMS to connect to OpenLab software. This also enables fetching of instruments and their status from OpenLab.
- OpenLab CDS can be integrated with SLIMS to select projects, instruments, and acquisition/processing methods based on the user's rights in OpenLab Shared Services.
- Sample Scheduler for OpenLab is an add-on to OpenLab CDS that enables SLIMS to submit sequences from the workflow to OpenLab CDS without the manual file manipulation that OpenLab CDS requires if integrated by itself.

ⓘ OpenLab CDS and Sample Scheduler do not run in the cloud, so only SLIMS with on-premises installations can integrate with either software.

ⓘ Sample Scheduler for OpenLab includes the requirements for OpenLab CDS, so use the section for Sample Scheduler if integrating Sample Scheduler for OpenLab.

### 4.5.1. Sample Scheduler for OpenLab Integration

Sample Scheduler enables SLIMS to automate the scheduling and assignment of samples to run them on the right instrument using the right methods. The interface provides a view of instruments and samples in process, allowing the user to assign the sample to the required or next free instrument for higher instrument utilization, and includes the instruments from the SLIMS integration. It allows chromatography labs using routine automated analytical methods to be managed from a simple and intuitive interface without having to open OpenLab CDS itself. The integration maps Sample Scheduler concepts into SLIMS concepts. Sample Scheduler for OpenLab requires OpenLab CDS and OpenLab Shared Services but only certain versions. Check the technical requirements to see the versions that are supported with integration to Sample Scheduler and SLIMS.

The following steps guide administrators on how to set up a SLIMS OpenLab Sample Scheduler execution protocol to allow contents from a SLIMS protocol run step to serve as inputs to Sample Scheduler. This allows SLIMS to create analyses sequences in a supported version of an OpenLab environment. This section explains how to configure and use the protocol in a clean installation of SLIMS.

#### 4.5.1.1. Prerequisites for Installation

This configuration is required in order to access all the information needed for the configuration of the protocol:

1. The customer must have a full installation of OpenLab CDS software in order to run: including an OpenLab Server, Agilent Instrument Controller which manages the communication

between OpenLab CDS and instruments, and OpenLab CDS Client which controls the analyses that are run on the instruments.

2. The customer must have a full installation of Sample Scheduler for OpenLab including the Sample Scheduler Web Interface.

#### 4.5.1.2. OpenLab Authentication and Automations

These authentications must be configured in order to establish API connections between SLIMS and OpenLab.

1. Two parameters need to be enabled and configured in the Authentication module. On the OpenLab tab, enable OpenLab Shared Services authentication.
  - The Server URL field must contain the base OpenLab API URL.
  - The Admin Name and Admin Password must be valid credentials for a user with administrator privileges on the OpenLab software.

On the OpenLab tab, also enable OpenLab Sample Scheduler integration. The Server URL field must contain the base Sample Scheduler API URL.

The credentials provided here are used to request necessary information from the OpenLab software like available instruments, projects, and templates.

2. At least one user must be configured with Agilent OpenLab Shared Services Authentication in the Users module. SLIMS will use this user's credentials to interact with Sample Scheduler administrative features, so user accounts created for this purpose need to have the necessary privileges for that level of access.
3. SLIMS GATE must be running to make use of the necessary plugin.
4. The "CDS" package is available in the Package Browser module in SLIMS. This package includes the protocol, instrument types, and more that are required for integration.
5. The "olss-samplescheduler" SLIMS GATE plugin is required to get results back from OpenLab CDS and can be obtained by contacting a SLIMS engineer. If SLIMS GATE is running, a log output will be available in the Plugins module that confirms when the plugin is successfully loaded.
6. Instrument types are imported by the CDS package, and the instruments need to be synced with OpenLab to import the instrument information into SLIMS. Click on "Sync with OpenLab Shared Services" in the Instruments module to do so before executing the protocol for the first time. The synced instruments will be the ones from which lab technicians will choose when creating analysis sequences.

#### 4.5.1.3. Configuring a Sample Scheduler Workflow

The majority of the configuration for an analytical workflow including a Sample Scheduler protocol is the same as any other analytical protocol until the configuration of the protocol itself. See the ??? section for help on general analytical workflow setup.

1. **Create a test.** A test and requestable are both necessary to send SLIMS content to the workflow.

Example: Create a test in the Tests module (e.g. "Urea Concentration"). Rememeber to con-  
figure the result datatype when creating the test.

2. **Create a requestable.** Create a requestable in the Requestables module. The type must be "Default" and the execution mode must be "Analytical." Select the test created in the previous step.
3. **Link the test and requestable.** If the lab setting "Create default requestable for new test" is not enabled, return to the test and select the requestable created in the previous step for the "Requestables" option.
4. **Create Analytical Workflow.** Create an "Analytical" type workflow in Workflow Management.

Example: Create a workflow named "Sample Scheduler" with type "Analytical."

5. **Add "OpenLab Sample Scheduler execution" protocol.** Protocols can come before and after this protocol, but this section only discusses how to set up the Sample Scheduler protocol. Create a new protocol in the workflow and choose type "OpenLab Sample Scheduler execu-  
tion."

Example: Create a protocol named "Waiting" with the first queue named "Start" and the sec-  
ond queue named "End."

6. **Open the "Submit Sample Scheduler sequence" protocol step.** Click on the protocol to open the details on the right. Select the Protocol Steps tab on the top. The "Submit Sample Sched-  
uler sequence" step is present by default.
7. **Set the "Submit sequence in status" parameter in the protocol step.** The "Submit sequence in status" parameter available under Details is required, and is set to "Scheduled" by default. A sequence with the "Waiting" status will require use of Sample Scheduler software to be scheduled for execution, whereas a sequence created with the "Scheduled" status should be scheduled for execution automatically, provided there are no errors.

Example: Set the "Submit sequence in status" parameter to "Waiting."

8. **Configure the Grid Template in the protocol step.** The Grid Template must be used to map values between SLIMS fields and Sample Scheduler parameters, so SLIMS is unable to use anything outside of the below "Supported Sample Scheduler Attributes." Click on the Grid Template tab on the bottom-right. The Grid Template handles important configuration that maps SLIMS record fields to Sample Scheduler attributes. The grid template contains a col-  
umn named "Sample Scheduler Attribute" with fixed choice, one-to-one parameters from the list below.

The limsId1 and limsId2 fields are present too, but are used by SLIMS for the content bar-  
code and unique identifier. These fields cannot be mapped in the grid to be filled in by the user.

**Table 4.5. Fixed Sample Scheduler Attributes**

Fixed Fields	Datatype in Sample Scheduler
Project	Text
Instrument	Text
Acquisition Method	Text

Fixed Fields	Datatype in Sample Scheduler
Processing Method	Text
Sequence Template	Text

**Table 4.6. Sample Scheduler Attributes for Configuration**

Fields for Grid Template Configuration	Datatype in Sample Scheduler
sampleAmount	Float
level	Integer
vial	Text
sampleDescription	Text
injectionVolume	Float
calibrationRuntype	Enum (text)
injectionSource	Enum (text)
sampleType	Enum (text)
injectionsPerSample	Integer
samplePrepMethodName	Text
sampleName	Text
dataFile	Text
internalStandardAmount1	Float
internalStandardAmount2	Float
internalStandardAmount3	Float
internalStandardAmount4	Float
internalStandardAmount5	Float
target1	Float
target2	Float
target3	Float
target4	Float
target5	Float
multiplier1	Float
multiplier2	Float
multiplier3	Float
multiplier4	Float
multiplier5	Float
dilutor1	Float
dilutor2	Float
dilutor3	Float
dilutor4	Float
dilutor5	Float
limsId3	Text

**9. Optional: Create custom fields to map additional SLIMS fields to Sample Scheduler attributes.** The fields `cntn_barCode`, `xrsc_cf_acquisitionMethod`, and `xrsc_cf_processingMethod` are created by default when importing the CDS package. However, Sample Scheduler has many more attributes that can be mapped to SLIMS. To make use of them in the protocol grid template, create custom fields with the desired datatype in the Fields module on the "Protocol Run Step Content" table. The table above lists the datatype of the source fields so the administrator can match the datatype in SLIMS. See how to create custom fields in the ??? section. Keep in mind that mapping a lot of custom fields / parameters may cause the grid to become cluttered.

Example: Create a grid template that uses short text custom fields in SLIMS created for: `xrsc_cf_injectionSource`, `xrsc_cf_injectionVolume`, `xrsc_cf_vial`, `xrsc_cf_injectionsPerSample`, and `xrsc_cf_sampleType`. Map each field to the Sample Scheduler attribute of the same name.

This table describes a grid template for a simple workflow that only tracks samples for a Sample Scheduler implementation that uses a sequence template.

**Table 4.7. Grid Template Example - Sample Scheduler Field Mapping Simple**

Meta Tree	Field	Title	Sample Scheduler Attribute	Seq No
Content	Barcode ( <code>cntn_barCode</code> )	Barcode	Sample Name	10
Protocol Run Step Content	Sample Amount ( <code>xrsc_cf_sampleA</code> )	Amount	Sample Amount	20
Protocol Run Step Content	Injection Source ( <code>xrsc_cf_injectionSource</code> )	Injection Source	Injection Source	30
Protocol Run Step Content	Vial ( <code>xrsc_cf_vial</code> )	Vial	Vial	40
Protocol Run Step Content	Sample Type ( <code>xrsc_cf_sampleType</code> )	Sample Type	Sample Type	50
Protocol Run Step Content	Internal Standard Amount 1 ( <code>xrsc_cf_internalS</code> )	Internal Standard Amount 1	Internal Standard Amount 1	60
Protocol Run Step Content	Multiplier 1 ( <code>xrsc_cf_multiplier1</code> )	Multiplier 1	Multiplier 1	70
Protocol Run Step Content	Dilutor 1 ( <code>xrsc_cf_dilutor1</code> )	Dilutor 1	Dilutor 1	80

This table describes a more complex grid template with calibration standards that does not require use of a sequence template.

**Table 4.8. Grid Template Example - Sample Scheduler Field Mapping Complex**

Meta Tree	Field	Title	Sample Scheduler Attribute	Seq No
Protocol Run Step Content	Barcode (xrsc_cf_barCode)	Barcode	Sample Name	10
Protocol Run Step Content	Vial (xrsc_cf_vial)	Vial	Vial	20
Protocol Run Step Content	Sample Type (xrsc_cf_sampleType)	Sample Type	Sample Type	30
Protocol Run Step Content	Injections per Sample (xrsc_cf_injection)	Injections per Sample	Injections per Sample	40
Protocol Run Step Content	Calibration Run Type (xrsc_cf_calibrationRunType)	Calibration Run Type	Calibration Run Type	50
Protocol Run Step Content	Level (xrsc_cf_level)	Level	Level	60
Protocol Run Step Content	Multiplier 2 (xrsc_cf_multiplier2)	Multiplier 2	Multiplier 2	70
Protocol Run Step Content	Dilutor 2 (xrsc_cf_dilutor2)	Dilutor 2	Dilutor 2	80

**10 Add "Result" protocol step.** Click the "New" button to add a new protocol step to the protocol. Select "Results" for the type and select the test created earlier.

**11 Configure "Analytical Methods."** On the Workflow Graph screen, click on the Analytical Methods tab. Add an analytical method to the workflow. Select the Start and End Queues and the tests created for the workflow.

Example: Create Analytical Method called "scheduler-waiting," assign it to the "Start" and "End" queues so it will apply just to the protocol created between Start and End. Select the test created earlier (e.g. Urea Concentration) for the Tests option.

**12 Version the protocol and / or the workflow.**

#### 4.5.1.4. How to Execute an OpenLab Sample Scheduler Protocol

An analytical workflow with the Sample Scheduler protocol works like other analytical workflows, but there are differences when users reach the point at which the sequences are sent to Sample Scheduler. Like other analytical workflows, a request can be created by scheduling content via an order in the Orders module. When the request is created, users choose the requestable that routes to the available analytical methods for the Sample Scheduler's workflow. Then, when scheduling the order, users set the analytical method that routes to the OpenLab Sample Scheduler Execution protocol. For details on how to perform those actions, see the ??? section.

The content should now be waiting in the analytical workflow when the user navigates to the Workflows module. The rest of the instructions guide users on how to use the Sample Scheduler protocol.

1. **Start the protocol run.** Go to the Workflows module and select the Sample Scheduler workflow. Click on the Sample Scheduler protocol. Select the content in the grid that was scheduled via the order. Create a new protocol run for the content.
2. **Submit the sequence.** The "Submit Sample Scheduler sequence" step contains the fields to select an Instrument and select an OpenLab project. It contains a grid with the status of each analysis and the status of the sequence (the group of all analyses). Each row is an analysis and the sequence status can be seen at the bottom left of the footer near the "Create Sequence" button. The Analysis Status shows the status of the sample analysis. An instrument and an OpenLab project must be selected in order to continue the workflow. The values available for project and instrument depend on the user's OpenLab account permissions and what was configured in the OpenLab software by the lab administrator. The available acquisition and processing methods depend on the selected project.

The other values in the grid are optional by default and depend on configuration in the OpenLab software.

Click on the "Create Sequence" button once the values are set appropriately. The user can set the desired values for each of the analysis rows. The individual analyses' states will be displayed in the "Analysis Status" column.

The states can be viewed in SLIMS and are displayed with a similar shape and color to Sample Scheduler to maintain consistency:

- *Waiting:* An initial status of analysis. The analysis has been associated with an instrument and method and is ready to be started. Will remain in Waiting status until the user clicks the "Scheduled" button.
- *Scheduled:* An initial status of analysis. The analysis is queued and will start as soon as the instrument is idle and all analyses with higher priority have been launched. Does not require a user interaction to start.
- *Holding:* An initial status of analysis. The analysis is not associated with an instrument/method and cannot be edited or started. Interaction is required in Sample Scheduler.
- *In Edition*
- *Submitted:* A status of analysis after launch. The analysis is starting on the instrument. Interaction is required in Sample Scheduler.
- *In Progress:* A status of analysis after launch. The analysis is acquiring an instrument.
- *Error:* A status of analysis after launch. The analysis encountered an issue which could be a block or error on the instrument or in OpenLab CDS.
- *Waiting on error:* A status of analysis after launch. Multiple analyses can be launched on the same instrument. If there is an Error on the current acquiring analysis, the following scheduled ones have the Waiting on error status. The user will need to restart them when the issue is resolved.

- *Stopping*: A status of analysis after launch. The analysis has been stopped by a user in Sample Scheduler and it is currently stopping in OpenLab CDS.
- *Stopped*: A status of analysis after launch. The analysis has been stopped by a user via Sample Scheduler.
- *Ended*: A status of analysis after launch. The analysis has been acquired.
- *Incomplete*: A status of analysis after launch. The analysis does not contain a value for all attributes that are required by OpenLab CDS to perform the acquisition.
- *Rollout Pending*: A status that relates to analyses with a sequence template applied in the Sample Scheduler web interface. When sequence templates are applied to the analyses, this status informs users that the sequence is about to be rolled out and cannot be edited. Any actions would be performed in Sample Scheduler until Rollout reaches the next status: Waiting or Incomplete.
- *Sequence not Created*: A status that is specific to SLIMS to indicate that the sequence was not created and thus could not progress to Sample Scheduler.

### 3. Optional: Other useful actions.

- A "Sequence template" dropdown may be available along with the instrument and OpenLab project fields if there are templates associated with the chosen OpenLab project in Sample Scheduler. Sequence Templates can be used with SLIMS, but the rolled-out sequence where the sequence template was applied to the samples submitted by SLIMS can only be seen in the Sample Scheduler UI. The sequence lines added will not be shown in the grid within SLIMS.
- The list of instruments updates live the first time the user selects an instrument. After that, they do not automatically refresh on the step, but clicking the Refresh icon next to the instrument will refresh the status of the selected instrument at any time.
- The Sequence Status displayed below the grid shows the latest requested status of the created sequence. The Refresh icon can be clicked to get the latest status at any time.

SLIMS uses the acquisition and processing methods to create the sequence in Sample Scheduler, which submits the sequence to OpenLab CDS. The instrument the user selected is used, and the methods must be suitable for the instrument. The analysis is executed by OpenLab CDS and a report is generated. Using plugin customization, the reports can be sent back from OpenLab CDS in the results protocol step. Contact your SLIMS engineer to acquire and setup the plugin for obtaining results in SLIMS.

## 4.5.2. OpenLab CDS Integration

The goal of integration with OpenLab CDS is to sync with the chromatography instruments used for analysis in Agilent OpenLab Shared Services, and use the instruments in chromatograph runs in SLIMS with customizable OpenLab CDS sample sheets that the chromatograph run results are imported to, and import of the chromatograph run results on samples in SLIMS. The versions of OpenLab CDS that SLIMS supports integration with are listed in the technical requirements that can be provided by your SLIMS contact.

① Sample Scheduler for OpenLab is preferred over a standalone integration with OpenLab CDS because it works with later versions of OpenLab CDS, and has the benefit of transferring data in an automated way. Files have to be transferred manually using an OpenLab CDS integration, making Sample Scheduler a beneficial choice with its automation.

## 4.6. SOPs

Standard Operating Procedures (SOP) are managed in SLIMS with the **SOPs** module available under the Setup menu. An SOP refers to the lab protocol used in one step of a SLIMS protocol (Figure 4.1 ).

The following Role Access rights can be configured for use in this module:

**Table 4.9. Required Permissions for SOPs Module**

Access rights Category	Access rights Name	Purpose
Setup: Electronic Lab Notes	SOPs	This permission provides access to the SOPs module in which role users can create and manage the versions of SOPs used in the SLIMS instance.

① The same SOP can be used in multiple protocol steps.

The SOPs module contains three parts:

- **List of SOPs** (left panel): Contains the list of all SOPs in a grid. Therefore, all grid actions, SLIMS GUI tools, and view actions (create a view, add/remove columns, sort, filter, highlight, etc.) can be used on the SOP list.
- **Versions** (middle panel): See the *Versioning section*. When creating a new Version of an SOP whose Versions are already in use by protocols, the user is asked whether to use the new Version in these protocols or keep using the old Versions.
  - △ If a protocol is used in the ELN, a new protocol version is created when choosing to use the new Version in this protocol.
- **SOP Details** (right panel): This window contains two tabs:
  - **SOP**: Contains these SOP fields:
    - **Name**: The SOP name, which is independent of the Version.
    - **Type**: SLIMS offers three types of SOP:
      - **Text**: Contains a rich text field where the lab procedure has to be entered.
      - **Attachment**: The lab procedure is generally described in an attachment file that can be directly attached to the SOP.
      - **Link**: Contains a link to an URL where the lab procedure is described. The user can choose the displayed name of the URL link.

- **User and Group:** Defines the access rights to the SOP. Please refer to the Access Rights module for more details.
- **Version:** Takes a snapshot of the actual draft of the SOP and creates a new Version. See the *Versioning Section*.
- **Protocol steps:** Shows all protocols using the selected SOP, the Version of the protocol, the name, and the type of the protocol step. This tab is only available for specific versions of the SOP and shows the protocols using the selected version.

## 4.7. Electronic Lab Notebook (Routine)

The Electronic Lab Notebook (ELN) is a feature used to easily store and retrieve information about ongoing and past Experiments/Protocols, orders, samples, results, etc. In the ELN, Projects can be defined, experiments can be added, and experiment runs can be recorded.

This module requires the following Role Access rights to work:

**Table 4.10. Electronic Lab Notebook Permissions**

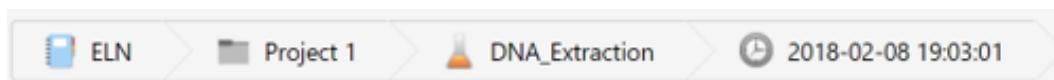
Access rights Category	Access rights Name	Purpose
Routine: Electronic Lab Notes	Electronic Lab Notes	This permission provides access to the ELN module for a role user of the SLIMS instance.

The following elements are available to streamline navigation in the ELN module:

- **Breadcrumbs (Navigation aid):** Within the bar situated on top of the ELN module, the Breadcrumbs (navigation aid) of the actual position within the ELN are shown. Breadcrumbs display the path to a Project, Experiment, or a Protocol run, but do not include the path to a step or a Sub Protocol run.

For instance, *Figure 4.2* shows that the left part of the ELN contains all the protocol run steps and notes contained in the Protocol run 2018-02-08 19:03:01. This protocol run is part of the experiment 'DNA\_Extraction' that is part of the project 'Project 1.' Clicking on an item in the breadcrumb brings the user directly to the item containing it. For example, a click on Project 1 brings the user to the content of Project 1.

**Figure 4.2. Breadcrumbs Example:**



- ① Projects can be concatenated in order to add clarity and structure to the ELN. Different projects can be reordered by right-clicking on a project to access the Move action in the menu . A project can be moved into another project by selecting either the desired project or the ELN main folder (root). Multiple projects can be moved at the same time by selecting multiple projects and using the Selection menu.

- **Favorites Bar:** The bar under the Module title and breadcrumbs line is the favorites bar. It contains the *Back* options on the left and all favorites on the right. The *Back* option allows users to return to the previously selected item, or to one of the previous selected items by choosing it from the dropdown menu, accessible via the down arrow just next to the *Back* button. Favorites can be added either by drag and drop of favorite items into the bar, or by right-clicking on the item and choosing *Add to Favorite* in the context menu. A favorite item can be accessed by clicking on its icon from the dropdown menu, and can be added in the top bar for rapid access by clicking on the star. A favorite item can be removed from the Favorite Bar by dragging and dropping it somewhere else on the screen. Otherwise, it can be removed by clicking the cross over the icon in the Favorite dropdown menu. A pop-up is displayed for the user to confirm the action.
- **Recently Worked On:** A dropdown list called Recently Worked On is available next to Favorites in the left panel. This list contains the last 10 ELN entries that the logged in user worked on, including standard experiment runs, simplified experiment runs, and notes. When the user edited text, linked content, or added attachments to these items, they're considered by SLIMS to be 'worked on.' The list is refreshed each time the menu is opened.
- **Hyperlinks:** Each element in the ELN has a hyperlink that can be used to access that element by entering (or pasting) the URL in a web browser. The link will require the user to login and then opens SLIMS into the right ELN element.

To create a URL for an element, right-click the ELN element and then select 'Generate link' in the context menu.

ⓘ This is needed to show results to different people by simply sending an email with the hyperlink pasted into it.

The rest of the ELN module space is divided into two submodules. The first one on the left side shows all Projects / Orders / Instruments and their underlying experiments, runs, and notes. Additionally, the left panel allows Projects, experiments, or runs to be added to the Electronic Lab Notebook by using the quick links on its top bar. The left panel can be collapsed to hide the Projects / Orders / Instruments to save space.

- **Left Window:** The first time the ELN is opened, the left module shows a list of projects. These projects are the main ones, but can contain other projects as well. The top bar of this submodule is a Global searching tool which contains two search options:

1. In records: It searches within all the element names and note texts that are present within the currently opened item.
2. In files: It searches within the text of attachments that are linked to all the elements within the currently opened item. Therefore it is also possible to search on filenames with a query similar to 'Name:file.txt' (See the **Attachments binary search section** for more details).

In this section, an item refers to a project, an experiment or a protocol run. The item is the last element of the breadcrumb in the top bar.

⚠ Protocol run steps are covered by searches, but only the protocol runs containing them are shown.

In order to navigate between the different items, single-click on an item name (when it is underlined) to open it, and double-click on an item name to edit it. Click the back button above to return to the previous item.

In the left window, right-click on an item to access the context menu. The menu contains options to generate *Reports* or export a Zip document (Export (ZIP)). This creates a Zip file with all attachments and information that are present in the selected item. A file containing the script to generate all symbolic links of the attachment can be generated as well using the context menu option "Generate symbolic links of the ELN tree" (See the **Attachments Symbolic Links Section** for more details) .

- **Right Window:** The window on the right shows the details and description of the selected element from the left window.

### 4.7.1. Projects

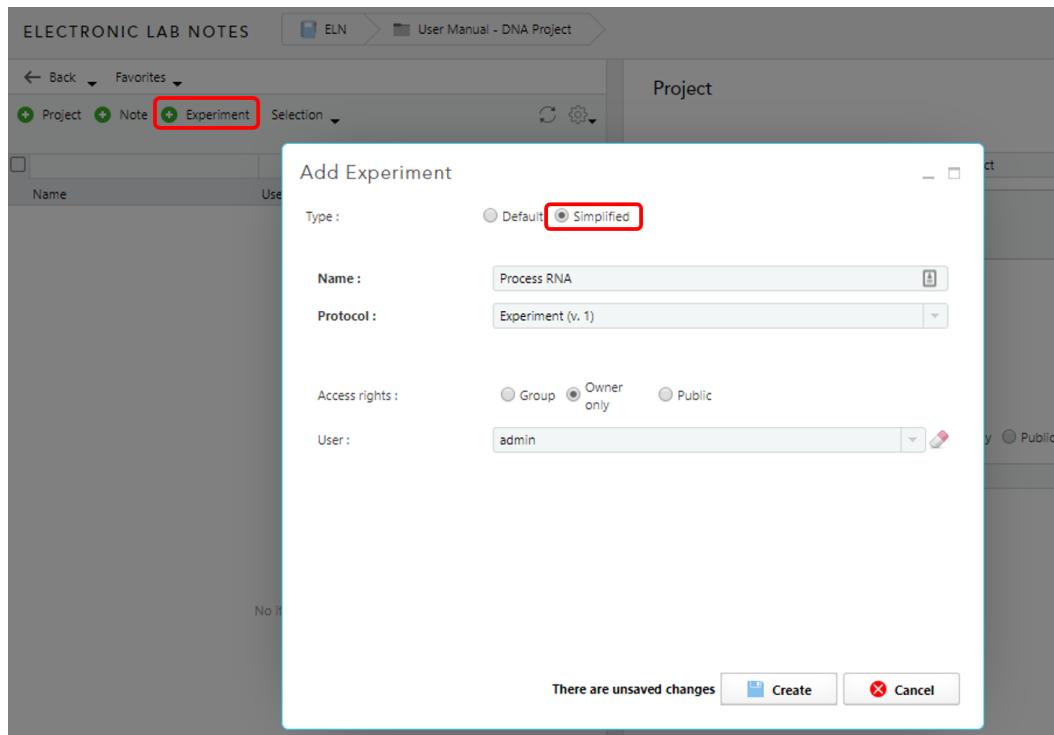
Projects are defined in SLIMS within the ELN workspace, and need to be defined first in order to register experiment data. Projects in the context of the ELN are a means to categorize sub-projects, experiments, and experiment runs into a neatly organized tree. A Project is defined with a name and description. In addition, access to a project can be restricted to a specific user or group. For collaboration Projects, a new group containing all the involved users needs to be created and the Project should be assigned to this new group. Projects generally contain several experiments that could all be in process at the same time.

### 4.7.2. Experiments

Experiments (default ones) are elements contained in the Projects. Each experiment has a name and a description. In addition, **an experiment follows a specific protocol template**. Once the experiment is started, it is not possible to change its protocol template. Hypothesis and Success criteria can be defined in the Description text field. A new experiment is automatically assigned the same user (owner) and group as the project it is contained in, but the user and group values can be edited during the experiment's creation or in the experiment details on the right after it is added.

### 4.7.3. Simplified ELN

Users can run a simplified protocol by adding a simplified protocol run in ELN. This is done using the  Experiment button that is visible after the user navigates to a project. In the pop-up, the user can choose to use a Default or Simplified Protocol. The user can then choose which existing simplified protocol to use and is required to enter a name for the simplified protocol run. Simplified protocols must be Active and Versioned in order to be available for selection here.

**Figure 4.3. Add Experiment with Simplified Protocol:**

Users can view and edit simplified protocol runs by clicking on them in the ELN. This opens the Simplified ELN screen on the right. Initially the simplified ELN screen shows a header and a number of blocks corresponding to the blocks defined in the simplified protocol.

**Header**

Type to add tags ...  Completed  Show removed  ...

**Link Content Block** 04/05/2018 11:11

**Note Block** 04/05/2018 11:04

**SOP Block** 04/05/2018 11:03

**Link Reagents Block** 04/05/2018 11:05

**Timer Block** 04/05/2018 11:03

**Undefined Block** 04/05/2018 11:03

**Add New Block - Hovering**

**Attachment Block** 04/05/2018 11:08

**Add New Block**

### Header:

The header shows the title of the simplified protocol run. A user can click on the title to edit it inline, and then needs to press the enter key to save the change (a help text reminder will appear as well).

Next to the name users can add tags by typing some text inline and pressing Enter. The last tag in the list can be deleted by pressing Delete. Users can also delete tags by clicking the Remove button on a tag. When entering a tag value, previously entered values are suggested automatically. Clicking a tag will show other simplified protocol runs that are associated with it. This changes the global search in the ELN Tree to #\*[TAG]. Tags are useful for entering directly in the Global filter, especially for searching multiple tags at the same time.

In the header there is also a 'Completed' checkbox that allows users to mark the protocol run as completed. Checking Completed makes the entire header and all the blocks read only, and prevents users from adding additional blocks.

Next to the Completed checkbox, a 'Show Removed' checkbox is present. It causes the blocks that were removed (not permanently removed) to be displayed.

The header contains two buttons to expand and collapse every block.

The last entry in the header is a button (...) that brings up a context menu with extra actions:

- *Detail:* Opens a form that shows the simplified protocol run details. Additionally, the fields "User" and "Group" can be edited in order to restrict access rights.
- *Generate Link:* Creates a hyperlink that refers to that simplified protocol run.
- *Print:* Expands all blocks and shows a print preview. If the protocol run is still editable, edit buttons and help texts will be removed before showing the preview, buttons, labels, and margins will be adjusted for a printed format. Tables with too many columns may not print in the most ideal width.

ⓘ Blocks can be printed individually or together in full depending on where the print action is initiated: in the individual block header, or the experiment header.

### **4.7.3.1. Blocks**

After the header there are a number of blocks. All blocks will initially be shown expanded. A scroll bar on the right allows the user to scroll the entire run. The title is shown on the top-left of the block and can be edited inline. On the top-right the following information and icons are available:

- Last Event Date: This shows the date of the last event.

Events are actions that change the data of the block, like editing the text or linking content. Moving a block is not an event (but it would change the last modification date).

- Each block has a context menu available (three vertical dots icon) that contains actions to show the history of the block, to copy the block, to soft-remove the block, and to remove the block permanently. If it is a text block, the create SOP button is also available.
- Collapse/Uncollapse Button
- Remove button which removes the block from the display (soft-remove), but allows it to be added back.

Collapsing and expanding blocks is saved until the next time that the user clicks Refresh.

The ordering of the blocks can be changed via drag and drop, which is constrained to the canvas area. When a block is dragged over another, they are displayed switching places live so the user can see the change before letting go.

Each block has a type that triggers a different visualization when it is expanded. The seven types are: Note, SOP, Link content, Attachment, Link Reagents, Use Timer, and No Predefined Type.

- Note Block:

Note blocks use the *Rich Text Editor* that allows users to add and edit text entries. Initially, the note block appears as read only. Click inside the text field where it says "Click here to start editing" to begin editing the note block, and it turns into an editable text workspace. Click outside of the text field and it closes edit mode and goes back to a read only state.

The Rich Text Editor takes the entire height of the available text area in read only mode as well as when starting the edit mode. The tools header is fixed on the top of the Rich Text Field or top of the page when scrolling in Edit mode.

⚠ The text editing height area does not adjust dynamically when editing the note block.

ⓘ While there is a specific block type for SOPs, a text block that has already been written to contain SOP material can be turned into an SOP. This action is available in the block actions (by clicking the three vertical dots icon) and choosing 'Create SOP.' This will save the text block as an SOP and bring the user directly to that newly created SOP in the SOP's module. This action is available only if the user has access to the SOP's module.

- SOP Block:

**Based on a Rich text/URL SOP:** The SOP block is the same as a note block, but the initial text is the contents or the URL of an SOP.

ⓘ In Note blocks or SOP blocks it is possible to Create an SOP from the list of actions. **This action is accessible only if the user has access to the SOP module.**

**Based on an attachment:** The SOP block is an attachment tile as viewed in the attachment tile grid.

SOP blocks in a Simplified Experiment run have a label that shows the name and version of the SOP that was used in the block. This allows the full history of SOP changes to be tracked when they are versioned between experiment runs.

- Link Content Block:

The link content block adds a content grid into which users can link content. The grid is initially blank until content is linked. Content linking can be done by scanning a barcode or using the *select content in content management* mechanism (the Create/Link Content button). After content is linked, the content grid in the block contains content columns and data similar to what is seen in the Content module. The columns in the content grid are the columns that are applicable for at least one of the linked records as long as the fields were defined as *Show in grid* or *Show in simplified ELN*. Adding extra columns in a Link Content Block grid is not possible; however, the open detail option (from the right-click or Selection menu) can be

used to view all content details. Additionally, the "select in content list" link is available in the context menu by right-clicking on a content.

The toolbar of a link content block provides the same actions available in the standard content module. Users can scan barcodes to link content, create new content, use the right-click and selection menus, and execute link/create macros on selected content. It also contains the SLIMS GATE icon to perform any "simplified" flows that are available to link or create contents. The right-click menu also contains the same actions available in content module (edit, derive, etc.)

Link content blocks adjust their height according to the number of linked contents in order to keep all records displayed.

- Attachment Block:

The attachment block allows the user to add files. Attached files are displayed in a list grid that has the same functionalities as the standard grid in the Attachments module. For files that are of standard (supported) image formats, users can hover over the file column to display a preview of the file. New attachments can be added using the 'Choose File' button on top of the grid. It is possible to select a file from the user's local computer or from the server (when enabled in Lab Settings). The 'Attachments' action in the actions menu (three vertical dots icon) of the attachment block opens the full attachment details window, and provides access to attachment details, links, and preview tabs.

The attachment block automatically resizes depending on the amount of attachments in the grid.

An action menu is available by right-clicking on a single attachment or by selecting multiple records, and the same actions as in a standard attachment grid are available: Copy to clipboard, Download, View, Show direct editor link, and Detach. 'Remove Permanently' is available for users with a role that enables deletion of entries.

- Timer Block:

The timer block provides a digital timer within the system that will count down the hours, minutes, and seconds that the user specifies. The user can choose from up to 168 hours in the first dropdown, 59 minutes in the second dropdown, and 59 seconds in the third. Then the user can click the Start button to begin the count down, and Stop to halt it at any time. If the clock reaches 0000, a notification will appear in the user's Notification menu. If notifications for the website have been enabled, the user will also receive a pop-up indicating that the timer expired. This can be used to count down a set time for any procedure the user is running in case the user prefers to not use a hardware timer outside the system.

- No Predefined Type:

This block has no type to start with and is only designated when the user is running the experiment. This allows the user to choose the type later, in case it is not known what type it needs to be before the experiment, or if the block needs to change type once the experiment has begun.

### 4.7.3.2. Adding Blocks

New blocks can be added before and after each existing block in the simplified experiment. The line "+ new block" is always visible after the last block or when no blocks are available. However, it is only shown when hovering over the blank space in between blocks.

Adding a new block prompts users to choose a button to create a block of one of the six types. The block types that are available to choose from are affected by the "Restrict Block Types" option that can be set during configuration of the simplified protocol. The default name "New Block" will be entered automatically, but can be updated as the block is added, or after. There is also a cancel button. When the user clicks on the button to create an SOP block, they are prompted to select an SOP. When the block is created it will be highlighted immediately by scrolling to it. If it is a note block it will start in edit mode.

ⓘ Note that the Sign off step is always the last step in an experiment and blocks cannot be added/moved after it.

### 4.7.3.3. Print Simplified Experiments

Simplified experiments can be printed one at a time, or multiple simplified experiments can be selected up to ten at a time and printed. The print action is available from the right-click context menu, or from the selection menu after selecting multiple simplified experiments. The print action is not available in those menus if any notes, standard experiments, or projects are selected. If more than 10 simplified experiments are selected, an error will prevent the user from printing.

All of the selected simplified experiments will be rendered into one html page, and then in the browser's print function, each new experiment starts on a new page.

The printed simplified ELN experiment contains these traits:

- **Header:** The SLIMS logo (left), the simplified experiment name (centered), and the print date and name of user (right).
- **Experiment Details:** The metadata about the experiment.
  - Tags on the experiment
  - Protocol and protocol version
  - Project name
  - User and Group names
  - Created and Modified user name and date
  - Completion date (if the experiment was completed)
  - The custom fields that were displayed on the experiment run.
- **Body:** The details of the simplified experiment.
- **Footer:** The SLIMS instance name (centered).

## 4.7.4. Protocol Runs

Protocol runs represent executions of an experiment. Therefore, the protocol template of the experiment will be initiated when a run is started and all the protocol steps will be available and have to be finished in order to complete the run. A new Run is initiated from the experiment by using the button *Add Protocol Run* in the top bar (Left window). If no 'Name' is specified for a new run in the form, the current date and time will be used by default. A new protocol run is automatically assigned the same user (owner) and group as the person that created the run, but the user and group values can be edited when the run is created or in the protocol run details form. When a Protocol Step is finished, the user is updated to be the person that finished the step. Between creation and finish of a step, the user can be updated (for example, to assign it) to another user.

ⓘ Value expressions are ignored in relation to the user field.

⚠ The user field (xprs\_fk\_user) does not determine the access rights to this step. The access rights to protocol run steps are managed (inherited) by the protocol run user and group.

The following Role Access rights can be configured for use in this module:

**Table 4.11. Optional Permissions for Protocol Runs**

Access rights Category	Access rights Name	Purpose
Setup: Electronic Lab Notes	Can run old versions of protocols	When enabled, the "Use specific version" field is available to the role user during the creation of a new run, and can be selected in order to choose a specific version of a protocol from the list. When disabled, the default behavior applies, which is to use the latest version of the protocol in the experiment.

⚠ A run that has already started will always keep the protocol version it started with.

The version of a protocol being run is indicated in the field 'Protocol' along with the name of the protocol.

### 4.7.4.1. Ad Hoc Steps

Protocol run steps that are not included in the protocol can be added to a run using the Ad hoc feature *Ad Hoc Step*.

After clicking the button, the first parameter to set is where to put the step. It can be added in the 'First' position, or after one of the existing steps.

The next parameter is to specify whether to use a *Preconfigured ad hoc step* that has been previously set up, or a *Custom ad hoc step* with all new definition.

A *Preconfigured ad hoc step* was defined during the creation of the protocol. Users define such steps when they know it will be needed in the protocol, but the number of repetitions that will be needed is not known until the protocol is run, or the user does not know at what time the

step will be needed. This is also useful when a step needs to be performed one or more times based on the results of a previous step, meaning it is dependent on something that is not predictable.

For the other option, a Custom ad hoc step should be defined in the same way that it would be configured in 'Protocols.'

The use of an Ad hoc step is the same as a normal step.

## 4.7.5. Protocol run steps

When selecting a Protocol run step from the left window of the ELN working space, the right window will display the step description and provides access to the following elements:

**Figure 4.5. Elements of a Protocol run step**



1. **Note:** Observations can be edited in the rich text editor.
2. **Detail:** Contains general information on the step like the user and the status.
3. **Content:** Contains a Content List View where Contents involved in the step are shown. Content can be right-clicked to access the "Select in content list" action which navigates directly to the Content module with the content selected. The Rules regarding the content linked to a step follow:
  - a. *Starting a step for the first time:* Contents of the previous step (even if it is not finished) are recorded as contents in the present step. Note that if a step is skipped (not started), contents will not be linked to later steps. A step that was never started does not have any Content, so no content will be forwarded from the step.

Concerning steps of type derivation or subprotocol, their definition determines which samples are transmitted to the next step. 'Content transferred to next step' in the top part displays the definition.

- b. *Redoing a step:* The recorded contents would be contents that were already linked in this step and contents inherited from the previous step. If a step is redone and content is un-linked (removed from the run step), the content is not unlinked from the step that inherited from it.

*Link contents:* Contents can be linked in two ways:

- Scan content Barcode or by selecting contents from the Contents module using the 'Link content' button. Only the following types of steps allow linking contents: Link content, Link content and Add results, Derivation, and Mix. When opening the Content tab in a 'Link content' type step, the cursor is automatically placed on the 'Scan Barcode' field.
  - ① When using the *Link Content* button, the user is redirected to the content module with a bottom box that takes up most of the grid width and contains three actions: Back to ELN, Cancel, and Link and Back to ELN (only when a content is selected).
- 'Link orders' which links the order contents to the step. All order contents can be linked by using 'Link complete orders,' or only some of the selected contents can be linked by using 'Link partial order.' An Order can also be linked by scanning its Barcode.

While executing a derivation step, the derivation should be done inside the content tab. This is a bit different from a normal content tab as it has two grids within the tab. The top grid is for the inputs of the derivation and the bottom grid is for the outputs of the derivation.

**Scan and execute a macro:** Allows content to be scanned and a macro to be executed in one step.

① A derivation step displays any available macros that have a "create content" action in the top right window.

△ If a step is In Progress and content is added to the previous step, it will need to be restarted in order to load the newly added contents from its previous step.

4. **Usage:** This element is only accessible for steps of types Link content, derivation, mix, procedural, result, and flow. It allows the usage of a consumable, solvent, or similar contents to be tracked. The content that needs to be tracked must be linked to the Protocol run step. Only content types configured as explained in Section 4.7.7 can be tracked.

Each time some of this content is used, the amount is recorded on the Usage Tab in the Usage column. Multiple Usage columns can be added (Usage 1, Usage 2, etc.) The usage columns are associated to a quantity custom field whose amount will decrease by the amount that is put inside each usage column. To add a new usage (which indicates that the tracked content is used another time), the *add usage* button needs to be clicked and a value needs to be entered in the new column. A usage can be removed as well, which will increase the initial quantity field by the amounts of the deleted usage column.

5. **Locations:** This element is accessible only for steps of type Link location. In this tab, the location can be added either by scanning the location's barcode or using the *Link location* button. A new tab will appear next to the Locations tab for each linked location to show the location's graphical details. These tabs will remain for the rest of the run, except for steps of type Sub protocol.

6. **Sample results, Dynamic test results and Results overview:** These elements are accessible only for steps of type 'Result' or 'Link content and Add results' and they are described in the Results section. The tab Results Overview is automatically selected when opening a step of type Result.

7. **Attachments:** Allows files or images to be added as attachments from the user's local computer or the server to the Protocol run step. This can also be done by drag and drop.

8. **Attachments Preview:** Allows files or images to be added as attachments from the user's local computer or the server to the Protocol run step with a preview of the images. Drag and drop is also enabled here.

9. **History:** This displays a history of the run step without results changes.

10. **Flow History:** This displays the history of the flows execution. This tab is only available when a flow is configured.

Each Protocol run step has three statuses: Not started, In progress, and Done. The step status is changed using the icons to start the step  , mark the step as done  , and redo the step  . These icons are available to the right of the step name (in the left window of ELN).

① The Experiment run status is calculated automatically by SLIMS and therefore when all steps of a specific run are set to "Done," the run status is automatically set to "Done".

① Experiment run steps details, content, and results tabs can be refreshed within the execution of a SLIMS GATE flow.

#### 4.7.5.1. Sub protocol Step

A step of type Sub protocol is composed of run(s) of its defined Protocols (*Sub protocol*). This type of step will be present if a Sub protocol was built into the Protocol, and is not added like Ad Hoc steps in the ELN. The right window of a Sub protocol step contains a "General info" tab that shows an overview of the input/output contents. In addition, a new tab will be created for every new Sub protocol run that is added and each one shows the overview of input/output contents in the run.

① The Subprotocol runs and their steps are shown in the ELN grid in the left panel as a tree and not in a list view.

New runs of the Sub protocol can be added using the  Add button available in the right window next to the "General info" tab. If *multiple sub runs* are NOT possible, the button will not be available anymore once the sub protocol is added. A Sub protocol run is defined by its name, the protocol selected from the list (See *Sub protocol* ), and the selected contents to be used. The checkbox "Hide used content" will show only contents that are not used in other runs of the subprotocol.

For steps that allow sub run sharing, an additional tab will be available next to the Add and General Info tabs to insert an existing Protocol run  **Insert existing protocol run** This tab allows users to select a Protocol run by choosing the Project and experiment it is part of. An Insert button  **Insert** is available down the right side of the page to finish the Protocol run selection.

① A Sub protocol step Run is started automatically if the step contains only one protocol and doesn't allow multiple subruns or protocol sharing.

#### 4.7.6. Results

Results are obtained in SLIMS using "Tests" in a Protocol step of type "Result" or "Link content and Add results." Runs of Protocols containing Result steps can be created in ELN or a workflow with these approaches:

- *Order:* Scheduling an order in the Orders module creates a Protocol run. Some order types are used to schedule tests in the ELN, and others are used to schedule tests in a workflow.
- *ELN Experiment:* If not using an order, a "Link content" step followed by a "Results" step, or "Link content and add results" step can be used to acquire the content to test an experiment run.
- *Workflow:* If not using an order, content can be sent to some types of workflows from the Content module. (Some types of workflow require orders so they can be scheduled, like analytical QC using products.)

When a Protocol step of type "Result" or "Link content and Add results" containing tests is started, the following tabs will be available in the right window:

- **Sample Results:** Each line shows a Content and the tests to be performed in this experiment or protocol run step. The tests are only shown if there was one automatically created result. Tests for manual results and >1 automatically created results are shown on the Results Overview tab instead. Results can be entered directly by double-clicking on a record, and the entered values will be saved and can be accessed from the Content grid using the Results tab in Content (Section 3.10).

SLIMS keeps track of all changes done on a result when it is edited using the *History* option that can be accessed by right-clicking on a result record in the "Result Overview" tab.

- **Dynamic Test Results:** For each dynamic test a new tab is opened. Depending on the test definition, either all the observation points are already defined and shown, or they can be added at run time. When adding new observations at run time, the user clicks the add button and sets an observation label. In the example below, the test is called *Test dynamic integer for 3.6* and the observation points have a numeric label. Three observations were added.

**Figure 4.6. A dynamic test element**

Test dynamic integer for 3.6						
Observation		3.6.1	3.6.2	3.6.3	3.6.4	Status
<input type="checkbox"/>	Observation	1	2	3	4	
<input type="checkbox"/>	00000006	34	45	2	34 Verified	
<input checked="" type="checkbox"/>	00000007	45	65	74	34 Verified	

- **Variant observation:** Results classified as Variant observations can be created by using the Variants tab. Users add a new Variant observation by clicking the 'Add Variant Observation' button. Then users can select a content linked to the run step, one of the tests that were linked to the protocol step, and a Variant matching the selected test (See *Variants* section).
- **Results Overview:** This tab is available for Results steps and provides more detailed information about every result record. Each line is dedicated to one test entry (or one measure, if the test is dynamic) with a column to enter the result for each test.

An optional configuration may be set up to create aggregated results automatically. If they are, result rows are created for each of the defined aggregation methods. "Standard Deviation" and "Mean" are possible aggregated results. These rows are automatically updated as the user enters values and are not editable. Any rules on the aggregated results operate the same as for other values. To help identify which results are the aggregated ones, the "Aggregation Method" column can be added to the grid to show which are the mean and standard deviation results.

These actions are available in the menu when right-clicking on a result record and at the top of the grid when one or more result records are selected. When results are multi-selected, only the actions that apply to all of the statuses of the results are available. (For example, if a non-rejected result with a broken rule and a result with no broken rule are selected, the validate and restore options will not be present.) Some of the options are available in each result row as action buttons:

- *Open Details:* Users can use this option to view the details of the result.
- *Repeat:* This action is only available for results created in the Electronic Lab Notebook, and results in analytical workflows that were not added manually and that have a request associated with them.

The repeat action launches another whole protocol run with the same linked contents as the current protocol run. Only the results of the last protocol run are used for the order. If a subprotocol run is available to repeat, SLIMS checks if the subprotocol has not already been repeated and reports it to the user.

In order to undo the repeat action, the *Undo Repeat* action can be used on a new result. This will completely remove the current run and show the results of the previous run in the order.

- **Validate:** (The option appears in the right-click menu after a value is entered for the result.) Validates the obtained result from a test and changes its status to "Validated."
- **Revert:** This option allows the status of the result to be changed. It is particularly useful when rules are defined and the value of one result needs to be changed. First revert the result to 'status pending' and then enter the new value. Performing the action this way causes the rules to be evaluated again.
- **Reject:** This option allows the status of the result to be changed to rejected. The "Reject All" button is available above the grid if one or more results are selected so that results can be rejected in batch.
- **Cancel/Restore:** This option changes the status of the result to Canceled / restores the status to the prior status the result had before it was canceled or rejected.
- **View Previous/Following:** The repeats of a result can be viewed by right-clicking on it and selecting the *View Previous/Following* action. This contains all the results in a series of repeated results. It includes the current result itself, the results it was a repeat of, the results that are a repeat of the current result, and the results that are a repeat of the current result's original result.

In order to know if this result is the original, a repeat, or has been repeated, the columns containing the corresponding names can be displayed.

- **View Rule Evaluations:** This window displays all rules on the results with details.
  - **View Events:** This displays details on the events, all the repeats, and the reason for the repeat.
  - **View History:** This displays the history of all the actions and modifications done on the result excluding the events.
- ① An electronic signature will only be requested once for a batch selection of results if signature is required for the status transition.
- ① Id and Barcode are the only content fields available as columns in the Results tabs 'Sample results,' 'Dynamic test results,' 'Variant observation,' and 'Results overview.' A more comprehensive display of content and results data can be achieved by using the Grids module where the user can create a more detailed grid with appropriate data hierarchy and column display.

#### 4.7.6.1. Using Value Expressions

The value expression is used to calculate the result value when defined for a test.

The result is read only, but the checkbox Manual Value can be checked to allow the user to edit the result value.

#### 4.7.6.2. Manual Creation of Results

If the option "Allow creation of manual results" is activated, a new result can be added by clicking on the "New" button available in the "Results overview" tab. Clicking on "New" opens a pop-up window where user can select:

- A content from the list of linked contents
- A test from the list of tests that were assigned to the protocol step
- A value for the test

The following actions are available to continue or complete the action:

- *Create and add another record*: Creates the result and keeps the pop-up window open so more results can be created.
- *Create*: Creates the result and closes the pop-up window.
- *Cancel*: Closes the pop-up window without creating a result.

Results can be updated or created using the Export, Create template, and Import from Excel actions. If the result is created with Import from Excel, it will be linked to the Protocol Run Step.

#### 4.7.6.3. Status of Results

The result life cycle should normally go through four statuses:

1. **Pending**: The starting status of a newly created result. In this status the result still does not have a value because the test has not been performed yet.
2. **Available**: The status of a result when all the tests associated to it have received a value after they have been performed. If there are no *Rules* to evaluate or all the rules are fulfilled, this status will be updated automatically to 'Verified.' If all values are available, but some rules could not be evaluated yet or were not fulfilled, the status of the result stays at 'Available.'
3. **Verified**: The status of a result when all the tests associated to it have received a value and all rules were fulfilled and/or acknowledged.

Note that only normal result rules can block the result status from transitioning from available to verified. *Specification Rules* do not block the status from being promoted to verified. The warning specification rules are for informative purposes.

4. **Validated**: This status can only be given to a verified result *manually*. This is the finishing status of a result.

The life cycle can also be "interrupted" by putting a result into the status **Cancelled**. If a result is also available in a sub run and is being cancelled, rejected, or restored, the user is asked to confirm the action on the results of the sub run.

During the execution of an Order, a result can be **Repeated**. This action generates a new run where the test can be done again. The Repeat action is recorded in the *Events* of the result.

#### 4.7.6.4. Graphical Representation

Whenever the datatype of results for a dynamic test is a numeric (Whole number, Decimal number, or Quantity), the user has the ability to see a graphical representation of their results. To activate this, the result records of contents that are interesting to the user have to be selected. Then the 'Show Dynamic Results in a Chart' icon  can be used.

The graph representing the values is generated and displayed in a pop-up window. Right-clicking in the window provides a few customization options. The customizations that can be made are: Chart Type (choose the visualization type), Fill (fills the area under line), Stack (sums up the values), and Swap Facets (inverts the axis that represents observations and contents). The following are descriptions of the available chart types:

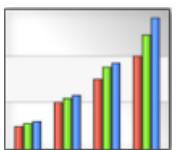
- **Area** (Default: filled and stacked):



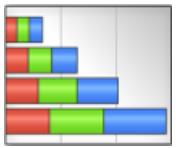
- **Column** (Default: filled and stacked):



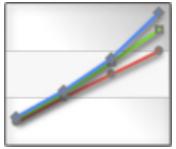
- **Column** (filled and unstacked):



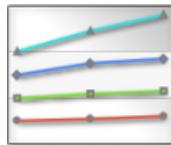
- **Bar** (Default: filled and stacked):



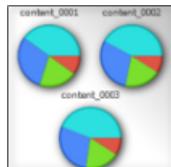
- **Line** (Default: unfilled and unstacked):



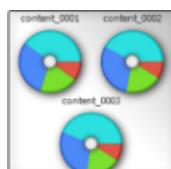
- **Line** (swapped facets):



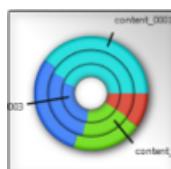
- **Pie** (Default: filled and unstacked):



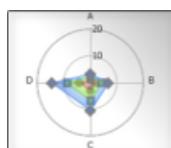
- **Doughnut** (Default: filled and unstacked):



- **Doughnut** (filled and stacked): each concentric doughnut ring corresponds to a different content



- **Radar** (Default: filled and stacked):



ⓘ Note: There is no distinction in the displayed labels of content IDs in a chart, so it is better to try not to have the same ID for all of the contents so they can be distinguished from each other.

#### 4.7.7. Track Usage

It is possible to track the usage of specific content types and keep the amount that is available to be used updated in the Electronic Lab Notebook.

Track Usage of a Content Type can be enabled by selecting one Amount of material field in the Content Type definition. Only fields of type 'Quantity' and with table 'Content' can be chosen. If usage tracking was enabled on a content type when it was set up, it will be possible to identify

it as a content type that should be tracked in a Protocol step. To do so, Track Usage should be activated in the step, and the content type should be selected so it will be tracked.

When a Protocol run step has Track Usage enabled, a Usage tab will be added in the right window of the step. This tab shows a line for every linked content that has a content type with Track Usage enabled. One or more usage columns can be added in the tab to track anything that uses part of the content. Double-clicking in the added usage column(s) allows the user to enter the amount of content that was used. Every amount entered is subtracted from the total remaining amount of the content. For the first usage column, the default usages (if defined) are used and immediately subtracted. When a usage is altered, the amount of content is re-calculated with the new values.

The Track Usage information for a Content can be retrieved in the Content Management module through:

- The Amount of material field which contains the available quantity.
- The ELN tab, which reports all Protocol runs where the content was linked.

#### 4.7.8. Notes

A note is an unstructured way of storing information in SLIMS. In this way, it can capture information that cannot be collected in standard fields, notes used to communicate with other users, and procedural documentation that doesn't align with other modules.

Adding a note is done in two steps. First, click *Add a Note* in the top left bar and give it a name. Then, enter information in the rich text editor in the right window with comments, observations, figures, or any unstructured data that needs to be captured.

Notes can be added at the Project, Experiment run, and Run Step levels. Multiple notes can be added at the same level. The note name and group or user access restrictions can be defined in the note creation form, or on the Detail tab after the note has been created. In addition, notes are enhanced to track data information using History. This is accessed by right-clicking on the note and accessing the History action (See the section *History* for further details).

The rich text editor used with notes enables direct drag and drop from the user's local computer into the text area (See the section *Attachments* for further details). Figures can be dropped in notes where they are displayed with the text. Figure dimensions and other properties can be modified by using one of the rich text editor tools available under the icon "image".

Notes are automatically saved every second so work is not lost.

Modification on notes are tracked in the history, which is history is updated after one minute of inactivity.

To see full details and instructions on the behavior of the rich text editor, see the *Rich Text Editor* section.

# 5. Order Management

The Order management Module allows the interaction of the laboratory with external people or labs, which are referred to generally as customers.

Any Order basically consists of a request a customer makes to a laboratory or a request a lab receives for the analysis of samples. The laboratory will fulfill the customer requests, or Orders, by running specific experiments recorded in the Workflows module as any of the defined protocol runs. The customer can access the results of the requested experiments from the Order management module. Usually the lab will interact with Workflows, and the customer will not access the details of experiments in Workflows, because it is only interesting to the laboratory, but rights to access the workflow can be given to the customer if the customer is a user of SLIMS as well.

## 5.1. Order Types

Like with other SLIMS components, Order Types need to be defined before Orders can be created and used. There are advantages to defining multiple order types.

The following Role Access rights can be configured for use in this module:

**Table 5.1. Required Permissions for Order Types Module**

Access rights Category	Access rights Name	Purpose
Setup: Order	Order Types	When enabled, the role user can access the Order Types module to define various order types. At least one must be present before the Orders module can be used, but the "Default Order Type" already exists as a built-in order type in SLIMS.

Each Order is required to have an Order Type which is associated with the configuration below.

- *Order Type*: The following types of order types are available:
  - *Order Management (default)*: One default Order Management order type already exists for the module. This order type allows you to make:
    - Orders to be executed in the ELN using Requestables from the Requestables module with execution mode "ELN" to request the execution of tests.
    - Orders to be executed in Analytical Workflows using Requestables from the Requestables module with execution mode "Analytical Workflows" to request the execution of tests.
  - *QC Orders*: This order type allows you to make:
    - Orders to be executed in the ELN using Requestables coming from the Products and Specifications module to request both test execution and rule (specification) evaluation.

- Orders to be executed in Analytical Workflows using Requestables coming from the Products and Specifications module to request both test execution and rule (specification) evaluation.
- *Workflow Orders*: This order type allows you to make orders to be executed in standard workflows (in which "Analytical" is disabled). There are three possible use cases:
  - This order type uses Requestables that are configured in the Requestables tab of a workflow in the Workflow Management module.
  - This order type is used for fulfillment orders that are executed in standard workflows which have been designed to handle content fulfillment requests, such as to order content that comes from another lab or different part of the same lab.
    - The option "Link the output of this requestable as a fulfillment in order management" on the workflow requestable must be enabled. This is used for a fulfillment where the input content is the same as the output content, or the output content may be created or derived, aliquoted, etc. from the input content. The output will be linked in the "Fulfillments" tab of the order.
    - The option "The output of this requestable is more than one content" on the workflow requestable can be enabled or disabled. If disabled, the workflow can be completed as long as there is an output content. If enabled, the user will be asked how many content need to be present before the workflow can be completed. This question appears when they add the request to the order. The workflow is prevented from being completed until that number of output content are present. Also, the request will not be "Finished" and the order will not be "Completed" until the workflow is completed with all the required content.
    - The "Number of samples" can be added to the protocol run grid to make the required number of output contents more visible to users. It doesn't show in the protocol run step, but in the list of protocol runs. Use the "Protocol Run Grid Template" on the workflow to add these leaves to the meta tree: "QueueElement: From QueueElement" > "QueueElementRequest: From QueueElementRequest" > "Request: To Request." Then, add the column field "Number of samples" under the "Request" meta tree.
    - Aside from within the order, the fulfillments can be traced back in the Content module. The Derivations tab shows the derived fulfillments, the Workflow tab provides access to the completed run(s), and the Orders tab shows the fulfillment orders and provides navigation back to them.
  - This order type is used for stability studies to use pulling orders linked to standard workflows which have been designed for batch pulling. Pulling orders are a special usage of fulfillment orders but are used only for stability studies. Their configuration is fully detailed in [Section 3.12.1 Stability Study Templates](#).
- Requestables can be restricted to certain Order Types, ('requestables' being anything that is defined as a request in an order, such as content actions, tests, etc.).
- Fields for an order can be restricted to certain Order Types.
- *Barcode mask for order*: The barcode mask of newly created orders can be set up for each order type. Please refer to the [Barcode Definition](#) for more details about barcodes.

To create a Barcode mask, the user can either type a new mask directly (use # for numbers), or select an already existing mask from the list. Please refer to the Sequences Module for more details.

- *Add all available requestables by default when registering order:* When this option is used, even if the user did not manually add requests to the contents linked to that order, all available requestables for that order type and role will automatically be added to the contents linked to the order when the order status transitions to "Registered" or "Scheduled" (if registration is skipped).

## 5.2. Orders

The Order Module is accessed through the *Routines Menu*. This module layout varies slightly according to the option 'Allow only one content record per order' that can be selected in *Lab Settings*. Changes to the module layout and accessible features when this option is selected will be described later in the *Single Content Per Order* section.

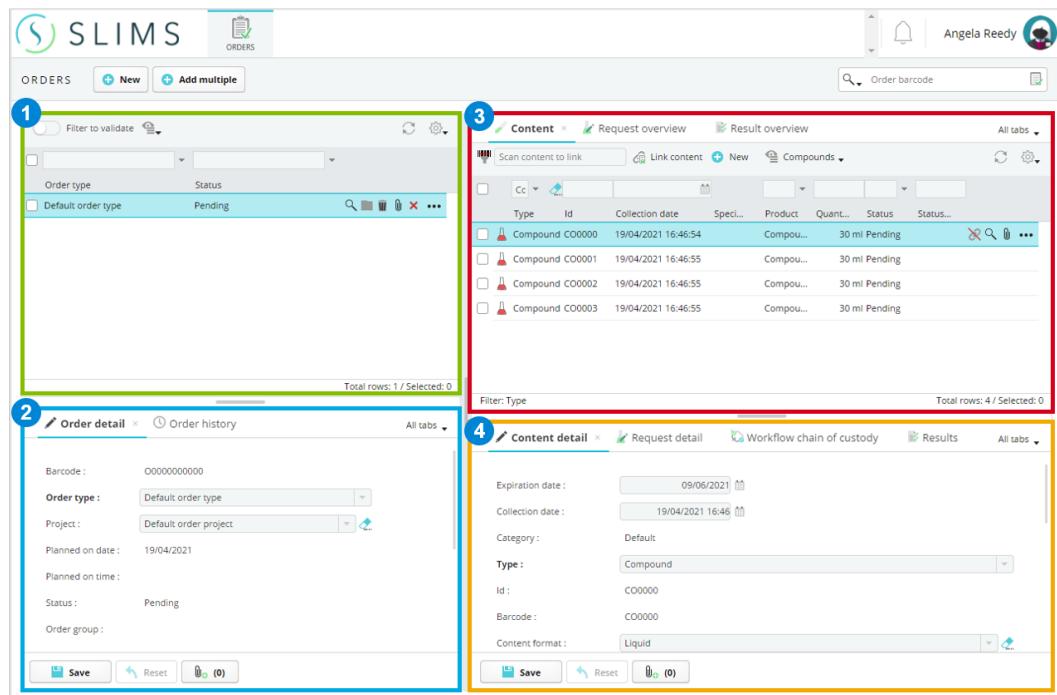
This module requires the following Role Access rights to work:

**Table 5.2. Orders Module Permissions**

Access rights Category	Access rights Name	Purpose
Routine: Order Management	Orders	When enabled, the role user has access to the Orders module in which orders can be created, linked to requestables, and sent to workflows.

The Orders Module is separated into two main parts as shown in Figure 5.1. The windows on the left (1 and 2) mainly contain details of the order. The windows on the right (3 and 4) describe the linked contents, Requestables, and results.

Figure 5.1. Orders Module



Visualization of the orders module and its four main sections. Section (2) shows the details of the order selected in section (1). Section (4) shows the details of the content selected in section (3) *if the tab 'Content' is visualized*. Instead, if the tabs "Request Overview" or "Result Overview" are opened in section (3), section (4) is hidden.

1. The first window (top-left) contains the list of orders. Order actions can be performed in this window. These are: all status transitions, execution of Macros or flows (if defined), and generation of reports.
2. The second window (bottom-left) includes two tabs that are related to the order selected in the first window:
  - **Order Detail**, where the details of the order selected in the first window can be viewed and edited;
  - **Order History**, where the history of the order is reported (primarily changes in the details or in the status)
3. The third window (top-right) contains a list of contents and an overview of requests of the order selected in the first window:
  - **Content**: A list of linked contents is shown in this tab with a display similar to the Content Management module. In this tab it is possible to create new content (automatically linked to the order) or to link an existing content.

Create Content macros and Link Content to Order macro steps can be used in the Content Grid to create or process content before they are scheduled with the order. The macro step can be configured to link content created in a previous macro step or link an existing content to the selected order. (See more about the Section 10.18 [359].)

Right-clicking a content linked to an order provides a list of actions:

- View/Add Attachments
  - Select in content list, which navigates directly to the Content module with the content selected.
  - Open Detail
  - Show Derivations, showing the derivation tree (see Section 3.10.10 for derivation)
  - View Rule Evaluations, showing the outcome of the defined rules
  - Show Content History, which shows history of changes to the selected content
  - Show Order Content History, which shows history of changes to the selected content data that relates to the order.
- ① These actions are also available in the Content module.
- **Request Overview:** This tab offers an ensemble overview of contents and requests. The Requestable, Test, and Protocol associated to each content are available. One row per content and per request is present.
  - **Result Overview:** This tab offers an ensemble overview of contents and their test results, with test and the value entered. One row per content and per test is present once the result has been entered in the ELN or Workflow.

① SLIMS selects a tab by default according to the order's status. If the status is before 'Scheduled,' the Content tab will be automatically selected; otherwise, the Result overview tab will be selected.

4. The fourth window (bottom-right) contains detailed information about the content selected in the third window.

△ This window is not present when the tabs 'Request Overview' or 'Result Overview' are shown in the third window.

- In workflow orders, the Requestable can be assigned to a particular user with the "Assigned To" dropdown. This affects the "Assigned to" filter in the Workflows module which will show only the content with requests assigned to the selected user in the workflow.
- **Content Detail:** The details of the selected content are shown and editable. There are three sections in this tab that separate the types of information about content in the order. The "Content Data" section is for data relating to the content regardless of its relationship to the order. The "Order Content Data" section is for data relating to the content such as specifications to be checked and any custom fields created on the Order Content table. The "Content Access Rights" section has the selection of group, owner, or public access rights as seen in other areas of SLIMS.

Custom fields created for order content can be displayed in the grid by default if the field is created with "Show in list grid" enabled, or displayed manually if the field is created with "Show in list grid" disabled. Additionally, the "Show in list grid" option adds the field to the Orders tab view in the Content module by default, or allows it to be manually added. Custom fields for order content are editable from the Orders module like other content

fields, but they become read-only if the order becomes read-only, such as when the order is scheduled.

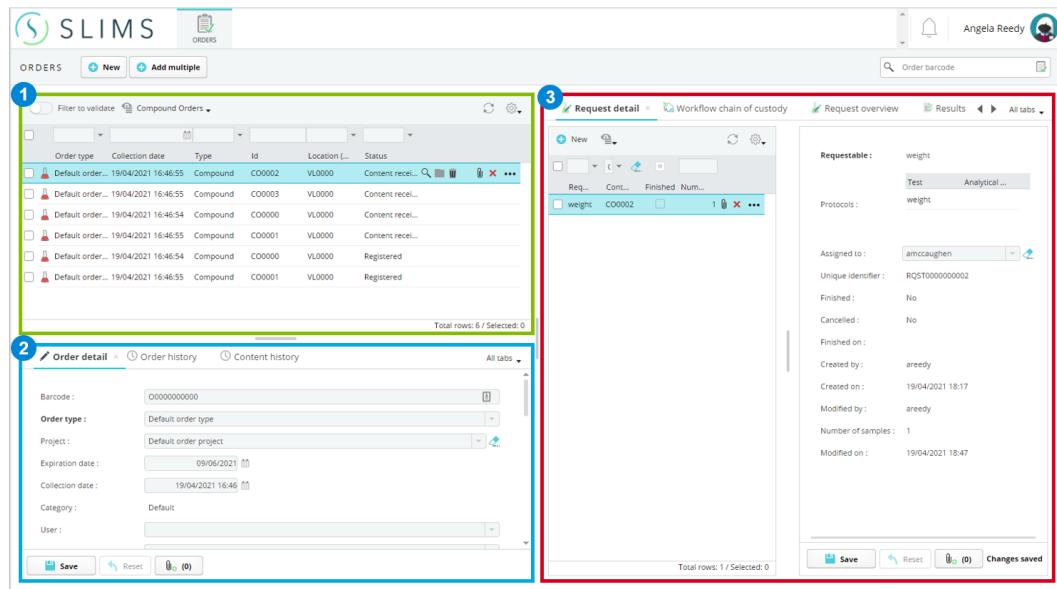
Unique identifiers are used to keep track of the content that is linked with an order. SLIMS uses the format rdcn\_orderBarcode\_contentBarcode. (Example: If content with barcode SA0002 is linked to order with barcode DO0003, then the order content barcode is recorded as rdcn\_DO0003\_SA0002.)

- **Request Detail:** Shows a list of Requestables associated to the selected content. Each Requestable is detailed on the right side of the window, with the list of tests included in the Requestable, and the protocol used to perform the test.
- **Fulfillments:** This tab is present if the order type is of type Workflow Order and a requestable has been requested that has "Link the output of this requestable as a fulfillment in order management" enabled. The tab contains a list of samples for which the request is fulfilled. In case of a derivation, the derived contents will be shown in the list.
- **Workflow chain of custody:** All the information about the requests, the protocols, and the protocol runs are stored here. This allows the sample to be followed through the workflow. A protocol run can be opened from that tab to see the details of the run by selecting the request and protocol run and clicking on the "Open Run" button.
- **Results:** During the execution of the Order, as soon as a result is defined in the ELN, it will also be visible in this tab.
  - ① Results obtained from derivatives of the linked contents are shown by default as well.
- **Local Results:** Local results are results that can be entered in the Order Management independently from the execution of a run. All tests with the option Adding results possible during order entry enabled appear in this tab. In this tab it is possible to enter a value for those tests or leaving them empty. When a value is entered, a result will also appear in the Content Management 'Results' tab.

### 5.2.1. Single Content Per Order

When selecting the option 'Allow only one content record per order' in *Lab Settings*, the form to create a new order is merged with the fields of a new content. Therefore, a new content record is created along with the creation of the new order. The feature to link an existing content to the order is disabled because it is not needed in this method. It is not possible for this type of order to go through workflows.

In addition, the overview screen of the Order module is simplified as shown in Figure 5.2 for this special case.

**Figure 5.2. Single Content Per Order Layout**

Orders module when 'Allow only one content record per order' is selected. Section (2) shows the details of the order selected in section (1). Section (3) shows requests linked to the order selected in section (1).

1. The first window (1) is a grid that contains columns for the fields of both the order and the content related to it.
2. The second window (2) has three tabs:
  - *Order Detail*: Contains the content fields merged together with the order fields.
  - *Order History*: Contains all details on the changes made to the order.
  - *Content History*: This tab tracks history of the content related to the order.

① SLIMS selects a tab by default according to the order's status. If the status is before 'Scheduled,' the Request tab will be automatically selected; otherwise, the Result tab will be selected.
3. The third window (3) shows details on the linked Requestables and the obtained results using the tabs 'Request detail,' 'Request overview,' 'Results,' and 'Result overview.' More details about each tab are available in the previous section describing the order module with general usage.

Only if one content per order is used, a mini-order grid is present in all the tabs of the right window of the module.

## 5.2.2. Order Life Cycle

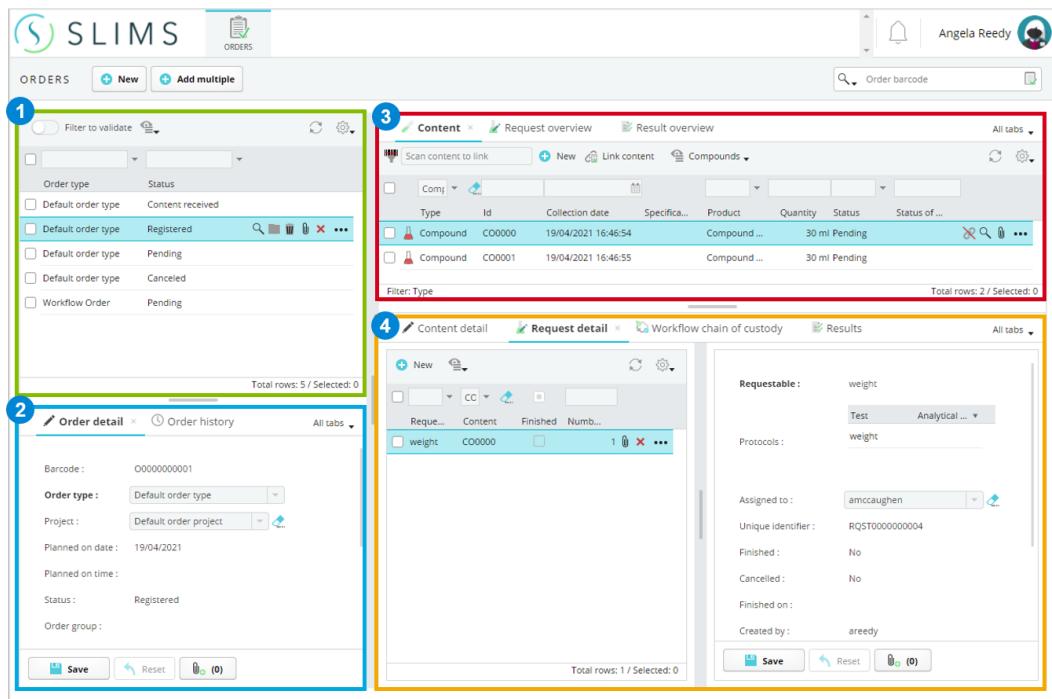
The following walkthrough illustrates the main steps to be achieved in order to process an order.

The following Role Access rights can be configured for use in this module:

**Table 5.3. Order Life Cycle Permissions**

Access rights Category	Access rights Name	Purpose
Routine: Order Management	Add Order	This permission allows the role user to add orders of the Order Types that are configured in the Orders module.
Routine: Order Management	Add Multiple	This permission allows the role user to add multiple orders at once using a start date/time schedule with an interval in Days, Hours, or Minutes in the Orders module.
Routine: Order Management	Copy Order	This permission allows the role user to copy an order from the selection or context menu in the Orders module.
Routine: Order Management	Receive Content	This permission allows role users to receive content on registered orders in the Orders module. The role user can assign a location to the contents linked to the order and print labels for the content.
Routine: Order Management	Can Add Content in Orders	This permission allows role users to create content in SLIMS from the Orders module in the Content grid when an order is selected.
Routine: Order Management	Can Link Content in Orders	This permission allows role users to link existing content to the selected order using the button in the Content grid in the Orders module.
Routine: Order Management	Can Assign Requests	When enabled, the role user is able to assign requests in the Orders module to a user of their SLIMS instance. This allows users to then filter for requests that have been assigned to them when working in the Workflows module.

Figure 5.3. Order Workflow



⚠ Rules can be added for each status of an order. An order will not get to a certain status unless all rule evaluation for this specific status are validated. A warning message is shown when a rule is violated.

Each step of an order workflow shown in Figure 5.3 is explained in the following list:

1.

**Create Order:** A new order is created by accessing the orders module  and using the button 'add new'  on the top left of the orders module.

When creating a new Order:

- The Order Type is required.
- The barcode can be entered and is required if it is not generated by SLIMS (defined in the order type).
- The Project, User, and Group can be specified. By default, 'User' will be the user creating the order.

Multiple orders can be created using a time schedule where the start date/time, the interval in Days, Hours, or Minutes, and the overall amount of orders to be created, are indicated. Weekends and holidays can be excluded from the order plan. Holidays are managed using the Holidays Module. When multiple orders are created, a progress bar opens in a pop-up window. The creation of the orders can be Cancelled from that window until the creation is complete.

Once the order is created:

- Existing contents can be linked to the order by using the icon  Using the *Orders Tab* in the Content management module, all orders that a content is linked to can be viewed.
- New contents can be created from the Orders module and will be automatically linked to the selected order.
  - A Product can be added to contents when they are created for a QC Order. To add a Product while in link mode in the Content module, use the "Open Detail" action and add one to the content. Then complete the link action.
- One or more Requestables can be added for each content. If the same requestable is added more than once, a warning is provided to alert the user: "One or more request(s) from this requestable have already been created for this content." The functionality 'Only Requestables from Requestable Groups allowed' can be used to hide the combobox to select a Requestable that's not in a group.
  - If a Requestable has a custom field, a pop-up window will open when selecting the Requestable in order to select a value for the custom field value.
  - Requestable custom fields are always visible in the request form of the Request Detail tab.
  - For each content record in a QC Order, only Requestables corresponding to the test Specifications defined on the selected Product version of a content can be selected as Requestables (see *Products and Specifications* module).

- 2. Register Order:** Once all the contents are linked, and the Requestables and local results are defined, the order can be registered using the button  Register which can be found in the right-click menu or the Selection menu.

*The statuses can be edited to control whether a user can edit a registered order or not (this configuration can be changed in the Statuses module). The default behavior is to allow users to edit a registered order.*

- 3. Receive Contents:** This is the first step done in the laboratory on a registered order and it consists of:

- Assigning a location to the contents linked to the order (often the user creating the order is not able to assign a location to the linked content because it has just been ordered and not shipped or stored at a location yet).
- Labels may potentially be printed for the received content.

- 4. Schedule Order:** The laboratory receiving the order will have to schedule the requests using the button  Schedule Order, which can be found in the right-click menu or the Selection menu.

- An order is scheduled for execution in the ELN by selecting the protocols to be run in the ELN for each request.
- An order can be scheduled for execution in a standard workflows after adding requests to the contents linked to the order.

- An order for execution in an analytical workflow can be scheduled after adding requests to the contents linked to the order. At that time, the tests can be multi-selected to assign users to work on them, or to choose or update the selected analytical methods.

Protocols contain tests which are linked to Requestables. Using this link, a user can select only protocols that have at least one test related to the Requestable. On top of the schedule window, the list of protocols that contain all the tests related to the selected Requestables is available, and a protocol can be chosen at once for all requests, if any exists.

△ Orders executed in the ELN are directly promoted to status 'Validated' when scheduling without specifying a protocol because SLIMS considers all results to be validated and all runs to be finished.

An experiment run will be created for each selected protocol. For instance, if the same protocol is selected for multiple requests, only one experiment run will be created in the ELN.

△ For Orders executed in the ELN, once protocols are chosen, all tests in these protocols will have a result, not only tests related to the ordered Requestables.

## 5. Execute Order (ELN):

The execution of an order consists in the execution of protocol runs in the Electronic Lab Notebook.

① The order in the ELN can be easily found from the Order module by right-clicking on it and selecting the "Select in ELN" action.

The order status is 'In Progress' during the execution of the Order in ELN until all steps are done and the Protocol Run status becomes 'Done.'

The order status will automatically change to 'Completed' when the execution of all the protocol runs is 'Done.'

① Details on how to enter results in ELN can be found in the *Results* section under the Electronic Lab Notebook chapter.

**Execute Order (Workflow):** The execution of an order consists of the execution of protocol runs in a Workflow. As the protocol run proceeds through the configured steps, the Order status also changes to 'In Progress.' When the execution of all of the protocol run steps in a workflow are completed, the order status changes to 'Completed.'

**Execute Order (Worklists):** The execution of an order can be simplified by using Worklists. The Worklists module furnishes a simple interface for entering results without needing to access the Electronic Lab Notebook.

## 6. Validate Order:

Validation of an order needs to be done manually and will automatically validate all of the results in the order that have the status "verified." For Orders executed in the ELN only, when all results have the status "validated," the order status automatically transitions to validated.

## 7. Receive Results:

In a typical configuration, a customer has access to only the Order module. Through the order module, the customer can check the status of the order and of the related results at any time.

- *Upper/Lower Specifications and Tests:* When the test result is entered in the ELN or the result step of an Analytical Workflow for the selected Requestable, a rule evaluates the defined Min and Max values selected in the specifications, and in rules from the Rules mod-

ule. A warning is shown for a result in the result tab if it does not satisfy the product specifications, or when it breaks a rule on a result.

- **Result Validation:** Result values can be validated in the Result Overview tab on the right. The actions available in the Orders module are the same as those in a Result step of the ELN. See the *Results* section for details on those actions.

In addition, the status of an Order can be

- **Canceled**
- **Reverted to a previous status**

The Status Workflows module provides a way to configure which transitions are possible. For example, the transition from Scheduled to Canceled may be forbidden. If that transition is configured, an order of status Scheduled cannot not be canceled. This helps to prevent undesirable status transitions.

## 5.3. Requestables and Requestable groups

Requestables along with Requestable groups allow customers to choose what is needed within the Requestables. It also provides freedom of choice for the lab person to decide which protocol and test need to be performed in order to fulfill the order.

The following Role Access rights can be configured for use in this module:

**Table 5.4. Requestables Permissions**

Access rights Category	Access rights Name	Purpose
Routine: Order Management	Show all Requestables (ignore restrictions on content type and order type)	This permission allows the role user to view all requestables in the Requestables module regardless of the content type and order type (useful for read only review type roles).
Routine: Order Management	Only Requestables from Requestable Groups Allowed	When enabled, this permission hides the searchable dropdown to select a Requestable that's not in a group so that the role user cannot select a requestable that is not part of a group. When disabled, all choosable requestables are available regardless of group assignment.
Setup: Electronic Lab Notes	Requestables	This permission provides access to the Requestables module in which the role user can create Requestables that link to tests created in the Tests

Access rights Category	Access rights Name	Purpose
Setup: Electronic Lab Notes	Requestable Groups	module in order to request the tests in the ELN or in an Analytical Workflow.

### 5.3.1. Requestables

A Requestable is a pre-configured request for whatever a client needs to request via an Order. For example, a client could request a specific protocol/workflow, like DNA extraction/NGS, to be executed on blood samples linked to an Order. Or the user could request that specific tests, like blood group and red blood cell count, be tested on blood samples linked to an order.

The Requestables for Electronic Lab Notes and for Analytical Workflows will be created in this module, however, Requestables for standard workflows are managed differently and are created in the Workflow Management Module.

The Requestables window is divided into two main panels. The first panel on the left contains a list of all Requestables and the second panel on the right contains details about the selected Requestable.

A Requestable is defined by its name, description, type, execution mode, and the tests that are required to fulfill the request. Only Requestables for ELN and Analytical Workflows are configured in this module. A Requestable can also be restricted to specific order types. For instance, a request for DNA quality control would require results for tests for the DNA concentration and DNA purity.

Requestables can be created to use with orders of type Order Management. The following options are available:

- *Request Unique Identifier Mask:* Generates a root name and incremented number based on the mask RQST##### for each request that is generated from this Requestable, or a mask customized by the administrator. See the Name mask (use # for numbers) definition for more details.
- *Execution Mode:* Determines whether the results of the tests for this Requestable will be entered in an ELN protocol or a workflow protocol. If "ELN" is selected, the Requestable will be available only for ELN. If "Analytical Workflow" is selected, the Requestable will only be available for *Analytical Workflows* in the Workflow Management module. New Requestables will default to Analytical Workflow.
- *Tests:* Select the tests that can be requested when this Requestable is requested in an order. (The default analytical method can also be set for analytical workflow requestables so that

when multiple analytical methods can measure a test, the default is automatically suggested when the lab user schedules the order.)

- *Allow only one test to be scheduled:* If multiple tests are linked to a Requestable, by default, all tests will be scheduled. But if this option is enabled, one test to be scheduled from this Requestable should be selected by the user during order scheduling. The default selected test depends on content type restrictions.
- *Only for selected order types:* A list of the existing Order Types that have been created. If enabled, the Requestable cannot be used in any of the order types that are not selected.
- *Only for selected roles:* A list of the existing roles that have been created. If enabled, the Requestable cannot be used by any of the roles that are not selected.

⚠ Restrictions do not apply on Roles where the functionality 'Show all requestables (ignore restrictions on content type and order type)' is enabled.

To create a new Requestable, first click on  in the left panel. Then add the name, description, and the list of related tests in the right panel.

ⓘ A request custom field is visible in the Request form of the Request Detail tab available in the Orders module. Also, a Requestable custom field can be added to Requestables that will be available in ELN and workflow Requestables, and analytical methods for Analytical Workflows.

### 5.3.2. Requestable groups

Multiple Requestables can be grouped inside a *Requestable group* which displays the Requestables in a more organized way. When adding a request to a content, the user will by default always see a dropdown of all available Requestables in the Orders module without specific organization. However, if the Requestables are grouped, the user will see categorized requestables as well as the dropdown, and be able to multi-select one or more requestables, or even select a whole group by clicking on the "Select All" checkbox next to the group name. To create a new one, click on , then provide a name, and add a description if desired.

The sequence number corresponds to the position of the Requestable group in the order form. Select the Requestables that are part of the group before clicking on *Create*.

## 5.4. Products and specifications

Products and their Specifications are managed in SLIMS with the **Products and specifications** module available under the Setup menu. Products are a set of Specifications for a content which define the tests that should be measured for this content and corresponding result acceptance limits for the content. An example might be adding a drug as a Product with Specification that it must have a minimum weight of 2g and maximum of 8g, and a minimum pH of 6, and max of 8. That Product could also have sub-Products with more stringent requirements on pH and weight called "Drug for US" and "Drug for JP" to capture the required result ranges that the drug must meet specifically for the US and for Japan. In SLIMS, a Product refers to tests that can be requested in QC Orders. Each Product has specifications that define its related tests (see below for more details). The distinct specification versions are associated to a content in the QC Order and pre-filtered by the product.

Created products can be associated to content to determine the specifications that need to be checked for the content in a stability study or analytical workflow. Contents can be created without products so that the specifications to be checked can be filled out later, if it is not known which are needed now. Contents can also be created with versioned products which are from the Study Design module, or draft products which are from Products and Specifications. Contents with blank or draft products associated to them can be linked in QC Orders and sent to an analytical workflow.

The following Role Access rights can be configured for use in this module:

**Table 5.5. Required Permissions for Products and Specifications Module**

Access rights Category	Access rights Name	Purpose
Setup: Specifications	Products and Specifications	When enabled, the role user can access the Products and Specifications module to add and define the tests that can be requested in the Orders module and impose boundaries on valid values of these tests (ex: provide evaluation for products to check if they have too high or too low mg of an active ingredient). (This permission does not impact the specifications in the Study Design module.)

The Products and specifications module contains three panels.

- **List of Products** (left panel): Contains the list of all Products in a tree grid. All Grid actions, SLIMS GUI tools, and View actions (create a view, add/remove columns, sort, filter, highlight...) can be used on the Product list.

A Product can have sub-Products which inherit all of the defined Specifications from the parent Product. Additional Specifications can be added to the sub-Product without updating its parent Product. See versions and specifications below for more details on how specifications are passed from a Product to its sub-Products

- **Versions** (Hidden, middle panel): The Versions dropdown provides the list of draft and all versions. If the 'Show All' button is clicked, versions are expanded into a list in a middle panel.

① Making a new version of a Product will create new versions of its sub-Products by copying their last version and including the new specifications of the parent Product if the parent is propagated. Editing a sub-Product causes a read-only version button to be displayed to remind the user to create versions on the parent Product. Like most versionable entities, products and specifications can be revised and removed until they are versioned. Then, new sub-Products and specifications added to the draft can be deleted until the parent is versioned again. Removing a parent also removes the children sub-Products and specifications.

The rest of versioning behavior adheres to the details in the *Versioning Section*.

- **Product Details** (right panel): This panel contains two tabs.

### 5.4.1. Product Tab

The Product tab contains the following product fields:

- *Name*: Enter the Product name, keeping in mind that it is version-dependent.
- *Version*: Takes a snapshot of the actual draft of a Product and creates a new Version. The button to create a new version shows up in the right panel near the versions dropdown in the Product Draft when edits have been made and saved. Changes made to the child products cause the version button to turn orange on the parent product. Versioning can be triggered on root products only. This button is available only for Roles that have the functionality 'Can create versions'. See the *Versioning* section for more details.
- *Description*: A text box to enter the description of a Product.
- *Publication status*: Displays the status of the selected product version.
- *Specifications*: Displays a brief read-only overview of specifications for the selected product. By default, it only has the specification name and active column, but more can be added to display further information. Enable "Show in dropdown list" on a default or custom field for the "Rules" table in the Fields module to show a column in this grid.

### 5.4.2. Specifications Tab

Three kinds of specifications can be created by choosing the Mode of Evaluation on the Specifications tab.

Click the "New specification rule" button to add a new specification and use these steps:

1. Enter a name for the specification rule.
2. Choose one of the three modes of evaluation:

- *Match Specifications*: Create standard Specifications, which are tests that have rules for acceptable values. First select an operator, then enter the acceptable values. The operators available in the selection list are pre-filtered to those that make sense for the datatype of the selected test(s). Details about the operators can be found here: [Section 10.16.2.1](#)

Single-test Specifications, or Specifications with only one test selected, will always evaluate if they are used in an analytical workflow regardless of the method used to generate the results. These could be generated with manual results, requested results (generated by a technician or by plugin automation), and repeated results.

The default analytical method can also be set for analytical workflow specification requestables so that when multiple analytical methods can measure a test, the default is automatically suggested when the lab user schedules the order. This is also an option available on script specifications through scripting with Value expression.

① The "schd\_fk\_workflowRequestable" field in the Fields module can be set to read only by the administrator in conjunction with this option if they want the default analytical

method to be selected for each requested test and be uneditable by the lab user scheduling the order.

- **Script Specifications:** Create Specifications that contain advanced criteria that are defined using a groovy script (see the Groovy Scripting chapter).

The structure of the groovy script for these purposes requires a "check statement" (evaluates a particular condition) and a "warning generation" (shows a warning to the user if the condition is met).

As an example, if the test name is "test\_color" and the purpose is to have a warning when the color is not red, the script specification would need the following values:

Name: Not red

Test: test\_color

Value expression: `if (test_color.rslt_value != "red")  
{ warn("test_color","The observed color is not red"); }`

① Note that this example can be created more easily using a Match Result rule.

- **No Evaluation:** Creates a Specification to execute a test and save the value for later statistical analysis, but does not impose any limits on the value.

3. Choose one or more tests that will be tested for the Product and that will use the specification rule.
4. Enter the configuration of the specification rule that appears depending on the mode of evaluation that was selected.
5. Choose whether to specify the "Minimum # of automatically created results" and an "Aggregation method" (only applicable to tests with numeric result values). These settings can also be specified on the results step of a protocol and SLIMS will use the higher number of results and combined aggregation methods that are configured, so the configuration needs to make sense. See more about these settings here: *Number of Automatically Created Results* and *Aggregation Methods*.

Independent specification rules can be created to check the user-entered results and aggregated results. Specifications created to ensure that the aggregated result values are within the desired ranges only apply to the aggregated result rows. Likewise, those created to ensure that the user-entered results are within range do not apply to the aggregated results. For example, given there are 5 results with a specification that checks for a standard deviation between "0g" and "1g", another specification for a mean between "5g" and "10g," and the user-entered results are: "1g," "2g," "3g," "4g," and "5g."

- The user-entered results go to status "Verified."
- The mean and standard deviation rule evaluations trigger and the results go to status "Available."

If the user-entered results are updated to: "6g," "6g," "6g," "6g," and "6g."

- The user-entered results are re-evaluated and are still at status "Verified."
- The mean and standard deviation recalculate, are re-evaluated, and go to status "Verified."

6. Choose whether the rule is propagated or not. Propagated rules will be added to all of the product's sub-Products. Rules can be propagated on any product or sub-product. Rules can only fully be edited on the product they were created on, not the children. Only the propagate and active attributes can be edited on the children.

① Propagating specifications makes them available for stability study templates that use the same product, but only with an explicit user interaction to turn on the link in the stability study template with the "Product version to reuse" field (see [Section 3.12.1](#) for more details). If they are reused, the specifications show up as inherited in the Stability Study Design module in the root specifications level.

7. Choose whether the rule is active for the product or sub-product. This is useful for rules propagated from the parent product because not all sub-products use every one of the parent product's rules.

Any specifications that are checked on a content that goes through a QC Order are added to History. A history event is captured that can be viewed in the Order History and the History tab on the content. An example of the history event might be: "Content [contentID] was scheduled in order [orderID] to check product specifications of [specificationID]."

### 5.4.3. Importing and Exporting Specification Rules

Specifications for Products and Specifications, Stability Study Templates [Section 3.12.1](#), and Stability Study Template Timepoints [Section 3.12.2.2](#) can all be imported and exported with Excel, and template files can be created.

When exporting specification rules for Products, Stability Study Templates, or Timepoints:

⚠ Script specifications (rule evaluation mode = script specification) cannot be imported or exported.

- The displayed specification records without the script specification are exported.
- The visible columns in the grid are exported by default with an additional column for the linked record. The "Unique identifier" is the display field of the linked record.
- The "Label" is the display field for tests.
- The advanced export allows users to configure the columns, file name, and format.

The column selection pop-up allows selection of the record (Product, Stability Study Template, or Timepoint) for further context when creating a template file. The exported file or template file can be updated and imported to make changes to the existing or new draft, respectively.

When importing specification rules for Products, Stability Study Templates, or Timepoints:

- Importing from an exported file updates the draft specification of the same strain, identifying it by the same internal ID.
- Importing from a template file creates a new specification under the draft of the record with the same display field. However, if the record column is removed so it can't be identified, importing the template file creates specifications on the record that is currently selected.
- Advanced functionality import can be done from a template file. One or more important keys can be selected so only the draft record will be searched, and when located, will be updated

with the imported values. An error is provided instead if no matching specification draft is found.

# 6. Workflows

The Workflow Module allows work to be scheduled and run per instrument or experiment by tracking actions performed on contents from multiple orders.

The basic building blocks of a Workflow are a sequence of queues and protocol runs. In order to use the workflow module, an order with contents and corresponding requests should be created and scheduled. There are two ways to send content to a workflow: the "To Workflow" option can be used from the Orders module and the Content module from the selection menu. Then the content that is associated to the request goes through different queues and protocol runs. How the content proceeds through the workflow depends on the way the requestable (the defined content or action related to the request) is set up, and on the choice of the user during the launch of each protocol run.

## 6.1. Workflow Management

The Workflow Management module is used to create and manage workflows in SLIMS. This module allows users to define protocols and queues, edit the grid for each queue, and create requestables for each workflow. The Workflow Management module must be enabled in the Roles module in order to be used.

The following Role Access rights can be configured for use in this module:

**Table 6.1. Required Permissions for Workflow Management Module**

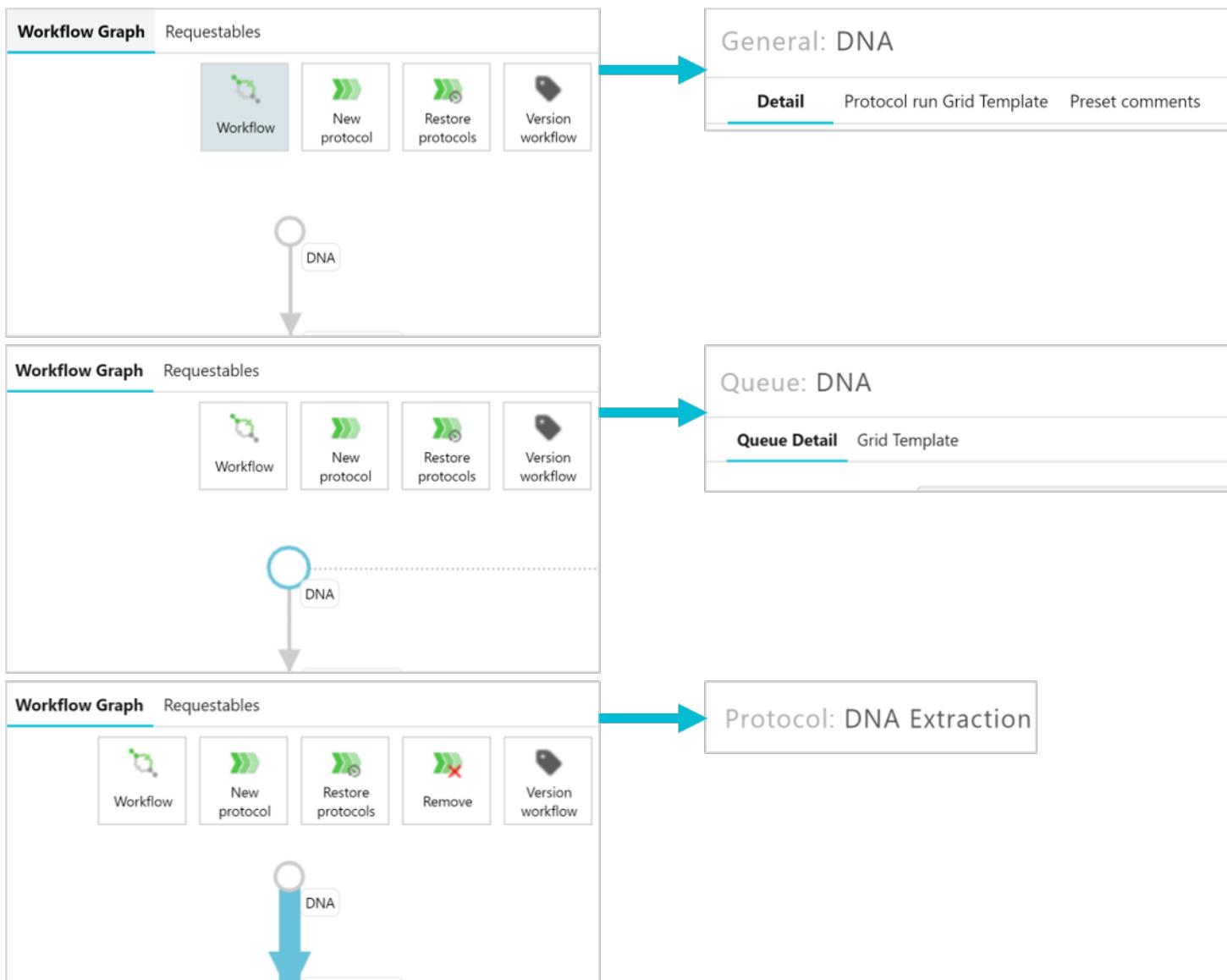
Access rights Category	Access rights Name	Purpose
Setup: Electronic Lab Notes	Workflow Management	This permission provides access to the Workflow Management module in which role users can build and manage their lab's workflow structure in SLIMS to prepare for its usage in the Workflows module.

The Workflow Management module is divided into two side-by-side windows with a set of tabs on each side (see Figure 6.1). The left window has two tabs.

- **Workflow Graph:** Displays a visual representation of the selected workflow.
- **Requestables:** Displays the list of the requestables associated to the workflow.

The right window contains different options depending on which tab is selected in the left window:

- **When on Workflow Graph Tab:** The "General" window is displayed when nothing has been selected yet. "Queue" is displayed in the right window when a Queue is selected in the workflow graph, and "Protocol" is displayed in the right window when Protocol is selected in the graph.
- **When on Requestables Tab:** The list of Requestables, and details and options for a selected Requestable are displayed.

**Figure 6.1. Workflow Management Tabs**

To create a new workflow, click the "New" button at the top of the left window of the module. This will open a new pop-up window where the name and the description of the workflow is defined.

### 6.1.1. Workflow Graph Tab

The creation and configuration of the protocols and queues that make up the structure of a workflow happens in the Workflow Graph tab.

In the top-right of the workflow graph are three (sometimes four) buttons:

- **Workflow:** Clicking on the Workflow button displays the overview of the workflow with the General view in the right window. This has four tabs:

- *Detail*: Contains the name and description of the workflow.
- *Protocol Run Grid Template*: Shows the meta tree (with the selected columns in the bottom panel) used during a protocol run in the Workflow module. This grid can be customized in the same manner described in the **Grids** section.
- *General Content Grid Template*: Shows the meta tree (with the columns in the bottom panel) used in a queue grid (content associated to a request and an order) using the workflow module.
- *Preset Comments*: Allows a user to create some preset comments to use when a run or a request is cancelled.
- **Remove Workflow**: This button is only available if none of the workflow's protocols are versioned.
- **New Protocol**: Opens a pop-up window with details to complete to create a new protocol.
- **Restore Protocols**: Allows a user to restore a deleted protocol.
- **Remove**: When a visible protocol is selected, allows the user to remove the selected protocol from the workflow. This is the same as making a protocol inactive in the ELN. Protocols that have been removed can be restored using the "Restore Protocols" button.
  - ① When a protocol is required for a request that is already scheduled, it appears as a dotted line in the graph until the related requests are no longer pending. The protocol will be removed when the requests are no longer pending.
- **Version Workflow**: This button is available to create a new version of the whole workflow at once. It is available whether there were changes in any of the protocols of the workflow or not, and allows the user to conveniently version all parts of the workflow at one time instead of being required to version each protocol individually. Protocols have an individual version button as well.

When Version Workflow is used, all protocols within the workflow will share the same versioning comment. If reauthentication is required for versioning any of the protocols in the workflow, versioning the whole workflow with this button will still require reauthentication.

A new protocol can be created using the New Protocol button. The following parameters must be defined for the new protocol.

- **New Protocol**:
  - *Create a New Protocol*: A new protocol will be created that requires the following fields:
    - Type: Default or Simplified
    - Name: Enter a name for the protocol.
    - Protocol type: This is only available if preceding Type is Default.
  - *Copy Existing Protocol*: Allows the user to copy a protocol that already exists in the workflow module. The Protocol Type and the Protocol that will be copied must be selected.
- **Protocol Queues (from and to)**

- *Create a New Queue*: A new queue will be created with the name provided.
- *Use an Existing Queue*: Select a queue that already exists in the workflow.
- *Copy Existing Queue*: Duplicate a queue that already exists in the protocol. When a protocol or a queue is copied, "Copy" will be added to the name of the queue or protocol.

Once a protocol has been created, the user can select the Queue (circle) or the Protocol (arrow connecting the circles). Clicking the Queue in the Workflow Graph tab on the left shows the Queue editing tabs in the right window, and selecting the Protocol shows the protocol options. After selecting the protocol, the details and the steps of the protocol can then be modified in the right window.

- Define the steps of a protocol of type Default as described in the **Protocols** module.
- Define the blocks of a protocol of type Simplified as described in the **Simplified Protocols** module.

The following details of a Queue may be edited in the right window by selecting the Queue in the workflow graph:

- **Name**: The display name of the queue.
- **Aggregator**:
  - *Keep*: The elements posted on a Keep Queue go through the Queue normally with no special changes made to them (no split or join changes).
  - *Split*: Elements posted on a Split Queue will be separated in order to have a different Queue element for each Request. For example, if a DNA content with two requests is posted on a Split Queue, two Queue elements will be created in the Queue so the DNA content can be processed twice, once for each request.
  - *Join*: Each element in the queue will combine all requests related to the same Content so there will be one Queue element per content. An exception is made for requests with different workflow sub-graph paths. These will not be joined in order to avoid processing a request in the wrong workflow sub-graph path.
- **Restrict Content Type**: Restricts the content that is allowed to be added to this queue. This means when a user adds a request on a content that does not match an allowed content type for the queue, only the analytical methods or requestables that do not start with that queue are available for selection. The requestables and analytical methods are filtered in the dropdown of requests and the Schedule Order pop-up window.
- **Grid Template**: The Grid Template is used to design the columns in the queue grid where content waits to go through a protocol run. Information related to the content in the queue can be displayed in the grid by defining the metadata (to show the content barcode as well as identifier, show the content's order identifier and barcode, show the content's batch number/lot number, show or hide all of the flags that are active on content that are going through the workflow, etc.).

The Grid Template is not created or modified in the Queue Detail tab. Instead, it is created in the Grid Templates module as a "Queue" grid template type. Creating Queue grid templates will allow complex grid templates to be created once and used many times across any of

your new protocols. Two Queue grid templates exist in the Grid Templates module by default, but you can create more custom ones. Select the grid template on the Queue Detail tab for the protocol to apply it to the selected queue.

Please refer to the Queue Grid Templates section for instructions on how to create a Queue Grid Template.

Exploring Queue Aggregators further requires an advanced dive into the topic, which can be found in the Use Cases: Queue Aggregators section.

## 6.1.2. Requestables Tab

Requestables for the selected workflow are defined and edited in this tab. Multiple requests can be created with different Start/End queues or different workflow paths. Click the "Add" button to begin.

The following fields are required:

- **Name:** A name to indicates the type of Requestable.
- **Start Queue:** The first queue where the content of the scheduled order will be placed.
- **End Queue:** The queue following the last protocol in the Requestable. After the last protocol run is completed, the content will be removed from the workflow (meaning it will not actually reside in this queue).
- **Target:** The Requestable can be characterized as None, Sequencing Settings, or Gene.

Other optional attributes are available:

- **Description:** A text box to enter any desired description about the Requestable.
- **Link the output of this requestable as a fulfillment in order management:** This option is used for fulfillment orders and also is required for pulling orders (a type of fulfillment order specifically for stability studies). The output of your workflow will be visible in the Orders module under the Fulfillment tab when this option is checked. This option is required for a requestable to be used as a "Pulling requestable" in the Stability Study Design module.
- **The output of this requestable is more than one content:** The user will be asked to specify an amount of samples that will need to be present when the workflow ends when this option is checked. However, for pulling orders, the user will not be asked to specify the amount of samples. Instead, the number of pulls required to finish the workflow will be determined from the stability study setup (# of pulls needed for specifications and additional # of pulls for storage conditions). Details about stability study design can be found here: [Section 3.12.1](#)
- **Only for Selected Order Types:** The Requestable can be restricted to certain Order Types.
- **Choose Workflow Path when Scheduling:** When this is checked, the path a requestable takes through a workflow is not predefined during setup so that it will be selected by the user when the order is scheduled. To predefine a workflow path for a requestable, see the Workflow Graph button description below.
- **Only for Selected Roles:** Access to the requestable can be restricted to particular roles.

Workflow Graph Button:

Once the requestable has been created, the workflow graph can be selected using the Workflow Graph button (located at the bottom of the right window). The path that content takes as it moves through the workflow corresponds to the protocols that will be run. The paths may be added or removed by clicking on the queues and protocols in the diagram pop-up. This can be used in two helpful ways. If used with "Choose Workflow Path when Scheduling," the whole workflow can be selected to provide all path options for the user to choose from when they schedule an order, or the paths can be restricted to the ones that should be available to the user by adding or removing them in the diagram accordingly.

If used without "Choose Workflow Path when Scheduling," then the setup should define which path the requestable will take automatically when the order is scheduled. In this case, the user does not choose the path when scheduling and the order moves through the path that was defined in the workflow diagram pop-up.

## 6.2. Workflow Module

The workflow module is available in the SLIMS Routine menu. The module is split into two windows. The left window contains a graphical representation of the workflow which can be collapsed to save space. Depending on the area selected in the left window, the right window contains either the general elements of the workflow, specific elements concerning the selected queue or protocol, or details about the currently executed protocol run. When a user opens the module, the default Workflow shown on the left will be the last Workflow they had selected.

This module requires the following Role Access rights to work:

**Table 6.2. Workflow Module Permissions**

Access rights Category	Access rights Name	Purpose
Routine: Workflows	Workflows	When enabled, the role user is able to access the workflows module and use any active workflows that they have permissions to access that were created using the Workflow Management module.

**Table 6.3. Workflow Module Optional Permissions**

Access rights Category	Access rights Name	Purpose
Routine: Workflows	Submit Content to Workflows	When enabled, the option "To Workflow" is available in the Content module so that content can be submitted directly to a workflow without being scheduled in an order.
Routine: Workflows	Can Requeue Queue Elements	When enabled, the role user is able to requeue queue elements (content that is waiting in a workflow queue) to send them back to a previous workflow queue.

Access rights Category	Access rights Name	Purpose
Routine: Workflows	Can Cancel Queue Elements	When enabled, the role user is able to cancel queue elements (content that is waiting in a workflow queue) to reject them from the workflow.
Routine: Workflows	Runs	When enabled, the role user is able to access the Runs module, which provides a list of previously run protocols and experiments from workflows and ELN. Specific runs can be located using filters and opened directly from the module.

The workflow module can be used when working on orders that have been scheduled on "workflow" requests, or if the functionality "Submit Content to Workflows" is enabled, when samples are submitted "To workflow" with a request from Content Management. Once an order has been scheduled or a sample has been submitted to a workflow, it appears in one of the 'From' queues. (This is indicated by the number in parentheses next to a Queue name in the workflow diagram in the left window.) A new protocol run is started by selecting a protocol in the workflow graph, selecting the contents/requests to use, and clicking on the "Create New Protocol Run" button.

### 6.2.1. Searching Records in a Workflow

The Search function in the Workflows module can be used to find records that are in progress:

- Order Barcodes
- Content Barcodes
- Any fields that have "Allow searching for this field in the workflow module" checked. This option is only available for custom fields of type "Short text" and "Barcode."

If an order barcode is found, the workflow will be filtered to show only:

- Content on a queue linked in the found order
- Runs with content linked in the found order

When content is found by searching a barcode:

- *If one content record is found in one place:* SLIMS navigates to the place where the content is in the workflow, as in, it opens the queue or the run the content is in.
- *If one content record is found in more than one place:* SLIMS opens a pop-up to show where the content is found and provides the option to navigate to it.
- *Otherwise, SLIMS filters the workflow on content with matching "Allow searching for this field in the workflow module" field values:* Based on content on a queue where one of the fields match, and runs with content where one of the fields match.

## 6.2.2. Linking and Unlinking Content to/from Workflows

There are three distinct concepts to understand when linking content to a workflow which affect how content proceeds through the workflow.

### Link and Select Content Concepts

1. The most commonly used concept is the "Select Content from Queue" step. Content can either be selected in the queue when starting a new protocol run, or if not selected, the "Select Content from Queue" step will be offered as the first step in a protocol. The back button in the following step (not the user's browser), will return the user to the "Select Content from Queue" Step in case any content was forgotten. This way, the selected contents/requests will disappear from the 'From' queue and appear in the process of the protocol run. The general work with the protocol run (what is done in one of the experiments as part of the workflow) displays the protocol run steps in a graphic from left to right at the top of the protocol run tab. If the protocol run is associated to an SOP, this is displayed under the run steps; otherwise, a reminder to select contents is displayed. To finish a step and progress to the next one, users click the 'Next' button at the bottom of the window. Once a protocol run is finished the content/request appears in the 'To' queue.

Content selected in this manner can continue through the entire workflow depending on the requests.

2. Protocol steps that are of "Link Content" type can also link content to a workflow. However, content linked in a Link Content step will only continue to the next steps in the protocol, and will not continue to the next protocol in the workflow.
3. Protocol steps that are of "Link Reagents" type can link content as well (like reagents), but in this case the content is confined to the protocol step. Content does not proceed to the next protocol steps in the workflow, or any further protocols in the workflow.

There are ways outside of the main concepts to link content. It can also be selected or linked using a SLIMSGATE flow if enabled. All "simplified" flows will be available.

Additionally, you can scan a location barcode to link all of the queued contents stored in the location in the Select Content from Queue step. The "Scan Content/Location to Link" option in the Select Content from Queue step of a workflow links the *intersection* of all contents stored in the location **and** the contents already present in the protocol queue to the step. If a content stored in the location is linked multiple times in the queue from different requests, an error indicates that it is not possible to select the same content multiple times in one protocol run, along with the barcode of the first duplicate content. Linking in this method follows the same rules as selecting content with concept 1 above.

### Cancel, Requeue, or Unlink Content

It is possible to cancel a run or a request. If the user clicks on the 'Cancel' icon then a pop-up window will be displayed. The pop-up window displays a grid containing the element records used by the run. For each element record, it's possible to select the queue that the element should be moved into after the run has been cancelled. By default, the intake queue of the protocol run is selected. A comment can be added by selecting a preset comment (defined earlier in the Workflow Management module) or by adding a custom comment. If the user clicks on the confirm button of the pop-up, then the run is cancelled. This means that the run will no

longer be displayed in the protocol run grid of Workflow Management. A cancelled run stays available in the tab workflow chain of custody of the Orders module.

Unlinking content can also be used to affect content in a workflow so it does not reach the end. Content that was added with the "Select Content from Queue" step cannot be unlinked. However, it can be requeued to a previous queue of the workflow, or cancelled to remove the request from the workflow. Only content that was linked from a link content step or a link reagents step can be unlinked. Also, content cannot be unlinked from a workflow when it is related to a request and there are no other content in that step related to the same request.

It is not possible to unlink content in previous steps using the back button. If the back button is used to unlink content in a previous step, it will not unlink the content from all of the following steps. However, Requeuing or Canceling content added through a "Select Content from Queue" step from previous steps after using the back button will remove the content/request in the subsequent steps as well.

### How to Requeue or Cancel a Request

- Clicking the Cancel button  provides a pop-up window where the user can enter a custom comment before confirming the cancellation.
- Clicking the Requeue button  provides a pop-up window where the user can enter a custom comment and select which queue to requeue the element to before confirming the change. Queue elements that are in the initial queue do not have the Requeue action available because there is no queue before the initial one to go to. However, cancellation is still an option in the initial queue. If at least one selected queue elements is not in its initial queue, the Requeue action is available, but only the non-initial queue elements will show up for requeue.
- If the content/request is a Mix, the pop-up will show the list of ingredients to cancel or requeue, and selections can be made for each individual ingredient.

### Examples of Link and Unlink Behavior

The following two examples illustrate this behavior when the back button in the protocol (not the user's browser) is clicked in a link content step of a workflow. Both examples use a workflow set up with Step 1: Link Content, and Step 2: Procedural. The content used in the examples are called Blood A, Blood B, and Water C.

- Two content are linked only using "Select Content from Queue."
  1. Blood A and Blood B are sent to the workflow queue by assigning a request and sending them to the workflow.
  2. A new protocol run is started without selecting content.
  3. Blood A and Blood B are selected from the queue in the "Select Content from Queue" step.
  4. Click Next to go to Step 1 of the protocol without doing anything else. This means that Step 1 has been created with Blood A and B in it.
  5. Click Next to go to Step 2 of the protocol without doing anything else. This means that Step 2 has been created with Blood A and B in it.

6. Click Back to return to Step 1.
7. Cancel Blood A to remove it from the workflow.
8. Click Next to go to Step 2 without doing anything else. Only Blood B exists in Step 2 even though it was created with both Blood content. This is expected because the content was added to the workflow with Select Content from Queue, and then Cancelled to remove the associated request from the workflow.
- Three content are linked using "Select Content from Queue" or using the Link Content step.
  1. Starting over, Blood A and Blood B are requested in the queue the same way as before.
  2. A new protocol run is started without selecting content.
  3. Blood A and Blood B are selected from the queue in the "Select Content from Queue" step.
  4. Click Next to go to Step 1 (Link Content step) of the protocol.
  5. Link Water C in Step 1.
  6. Click Next to go to Step 2 of the protocol and click on the Content tab to see Blood A, Blood B, and Water C.
  7. Click Back to return to Step 1.
  8. Water C is still present in Step 1 as expected.
  9. Unlink Water C and Cancel Blood A in Step 1 so that the only remaining content is Blood B.
  10. Click Next to go to Step 2.
  11. Click on the Content tab. Step 2 still has Blood B and Water C because the step has been created before with both of those content. It is not possible to unlink content in previous steps using the back button. If the back button is used to unlink content in a previous step, it will not unlink the content from all of the following steps. However, Requeuing or Cancelling will remove content/requests from subsequent steps as seen in the first example.

It is possible to search for content in the Workflow module using the 'Search' field in the top bar. Contents can be searched for using their barcode (search will find barcodes that match exactly what is entered). Any field of type barcode can also be used to search for a content. The search will be done on all the workflows and not only on the opened workflow. If the search gives one match, the workflow queue or protocol the content is in will automatically be opened. If a content is present in more than one queue, protocol, or workflow, a pop-up window will open to display all the results of the search. The queue, Content, Protocol Run, Workflow and Requestable will be shown to facilitate the selection of the content. Selecting the searched content will open the workflow queue or protocol that the content is in.

The hyper-link of each protocol run can be obtained from the "Generate Link" action in the right-click context menu. The hyper-link can be used to access that protocol run (after being logged in) by pasting or entering the URL into a web browser.

### 6.2.3. Using Flags in a Workflow

Flags are added manually in a protocol run step inside a workflow, and can be added to one or more content records. "Add Flag" is an action in the Content tab within a protocol run, available from the Selection menu and the Context menu.

Flags in a workflow are shown in the Content tab of a run. There are two columns.

- *Flags*: Flags displayed here are the ones that were added in the current protocol step and on that specific content.
- *Previous Flags*: Flags displayed here are the ones that affect the current journey through the workflow.
- Examples of flags:
  - Flags that were added in previous steps
  - Flags that were added in this step, but on different content. For example: flags on ingredients of a mix will be visible on the mix in the output grid.
  - Flags that were added in previous runs, unless the samples were requeued further back than the step that these flags were added. For example: A Require Requeue flag is added to a sample and then that sample is requeued to the start queue. The Require Requeue flag will not show up in the runs that follow.
  - Forbid Indefinitely or Require Cancel flags that were added in previous runs, even if those samples were requeued.

Flag details can be viewed in two places in the workflow. By clicking on the flag in the content tab, the detail pop-up opens with information about the flag. Otherwise, right-clicking on the content and choosing "Show Flags" displays a pop-up window that lists the current and previous flags.

Flags can be deleted by clicking on the flag and selecting "Delete Flag" in the details.

When adding a View Window in the content grid of a workflow step, a new filter is available for flags. The only flags available for selection in the filter are those not restricted to any protocol step, unless the flags are restricted to a protocol step for a content grid that corresponds to a run of that protocol step. The supported operators are:

- *Equals*: The flags must exactly match the selected flags. The criteria "equals Flag 1, Flag 2" will show content with flags "Flag 1, Flag 2" or "Flag 1, Flag 1, Flag 2," but not "Flag 1" or "Flag 1, Flag 2, Flag 3."
- *Not Equals*: Behaves the opposite way of Equals.
- *Is One Of*: The flags must be a non-empty subset of the selected flags. The criteria "Is One Of Flag 1, Flag 2" will show the contents with flags "Flag 1, Flag 2," or "Flag 1, Flag 3," but not "Flag 3, Flag 4," or contents with no flag.
- *Is Not One Of*: Behaves the opposite way of Is One Of.

Flags can also be filtered in the top column of the grid using the dropdown in the header. Multiple flags can be selected to filter. The result displayed in the grid contains contents that use at

least one of the selected flags (only if the intersection of the content's flags and the filter flags are not empty).

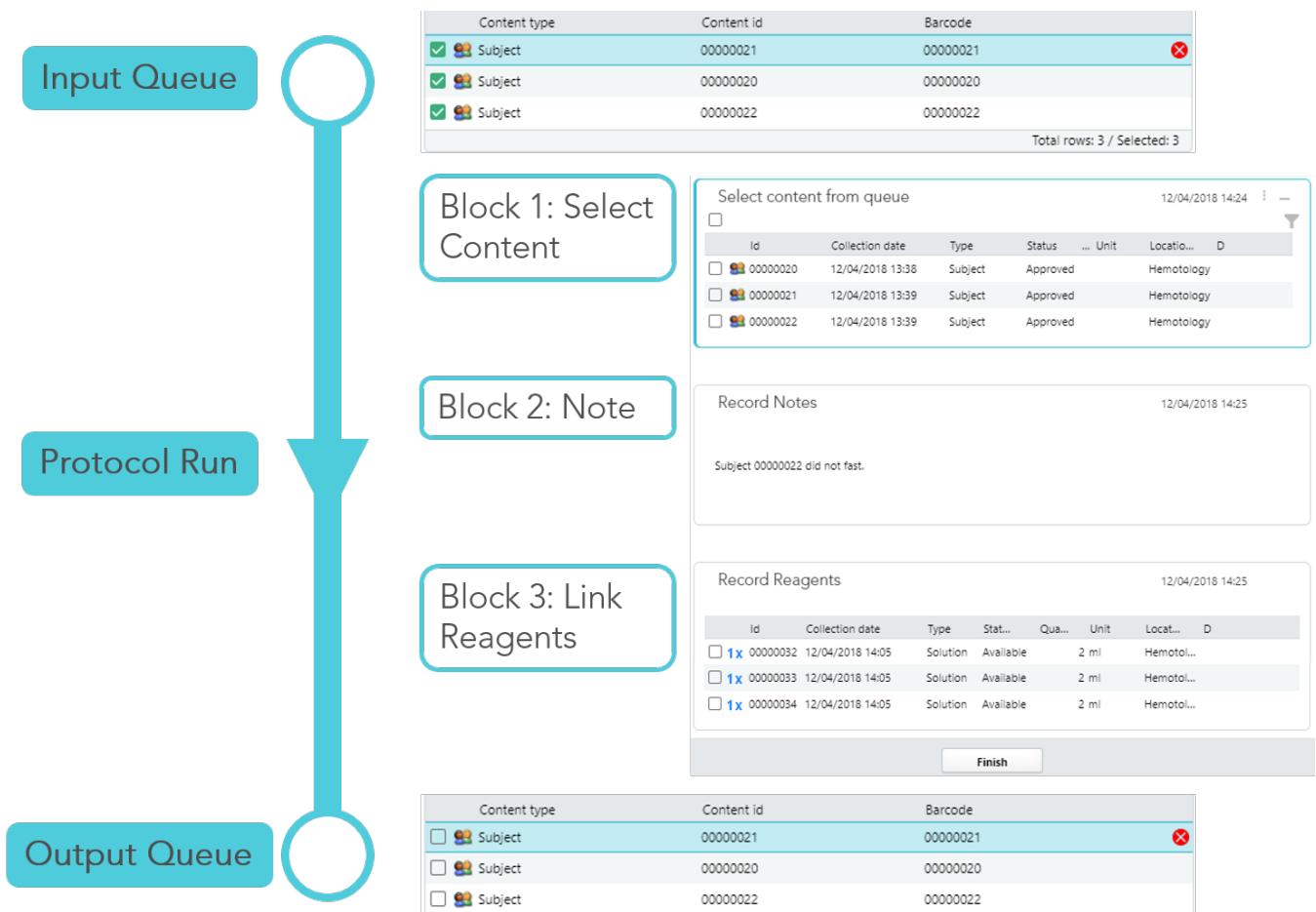
### 6.2.3.1. Flag Warnings and Errors

An error message is displayed when a user clicks "Next" or "Finish" to proceed to the next protocol step with content that cannot continue due to one or more flags. This is situational, so that content is naturally filtered out of content selection if it would get stuck along the path of the workflow due to its flag. It occurs when content cannot continue from its target queue or when it cannot continue from the start queue after being requeued due to flags. In these cases, content may need to be sent down a different workflow path, requeued to a queue such as the start queue that is prior to the one which activated the flag, or cancelled depending on the situation.

- *Step cannot be finished until the content(s) marked with flags FLAG NAME are requeued or cancelled.*
- *Requeue to queue QUEUE NAME is not possible for contents marked with flag(s) FLAG NAME. Sample CONTENT NAME must be requeued to another queue, or cancelled, or flag should be removed.*
- *No requeue is possible for contents marked with flag FLAG NAME. All paths are forbidden, sample CONTENT NAME must be cancelled.*

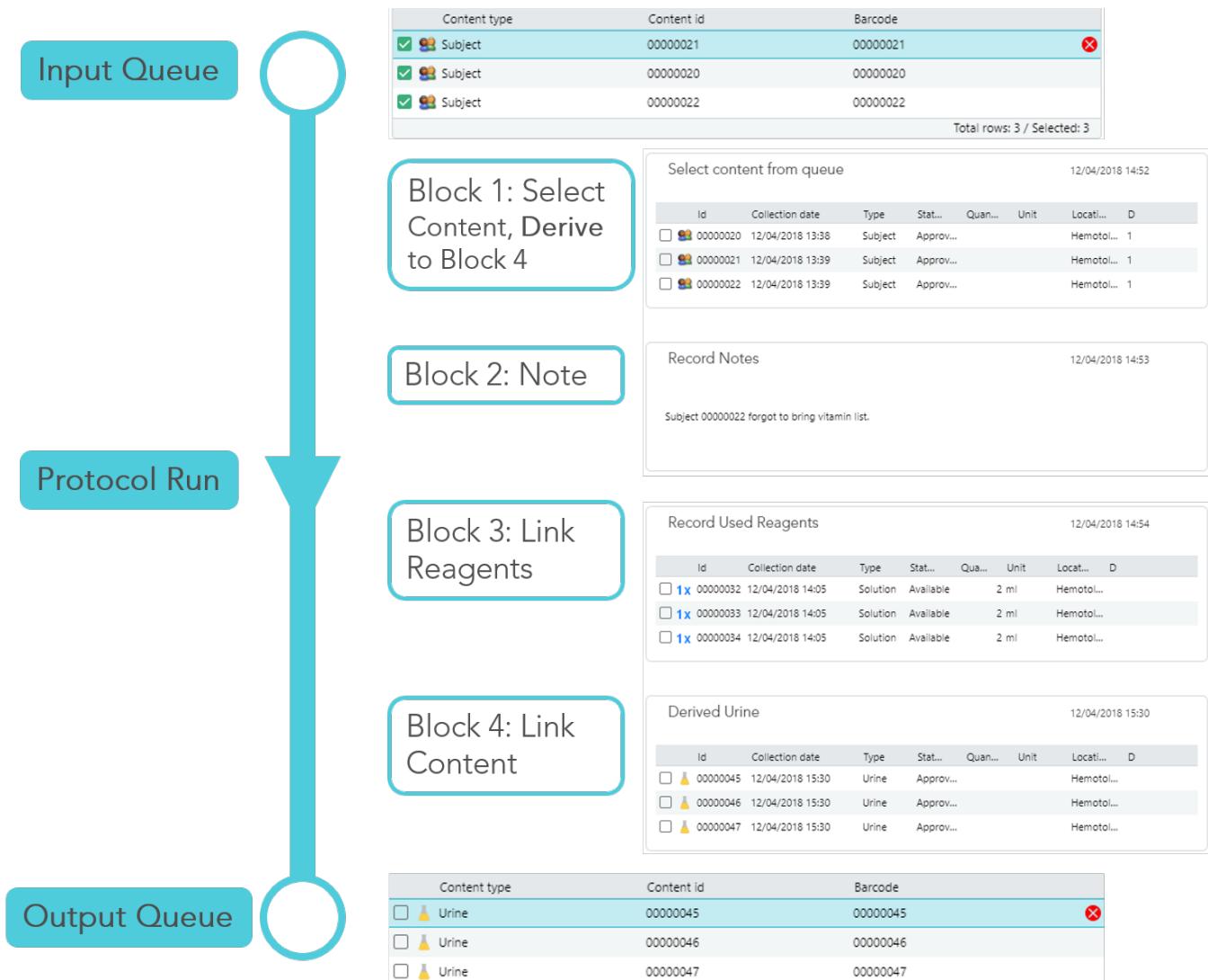
### 6.2.4. Simplified Protocols for Workflows

Several examples of simplified protocols follow, describing how content enters a protocol step, proceeds through the experiment (protocol run), and is loaded into the output queue for that protocol step. Each step consists of a series of blocks in a simplified protocol. Remember that the content from the last Link Content block in a protocol step is the content that will proceed to the next queue. Additionally, content that has been treated disappears from the workflow after it completes the last protocol run, because it is considered treated. Content can be seen in an Output Queue when there are further steps after that queue.

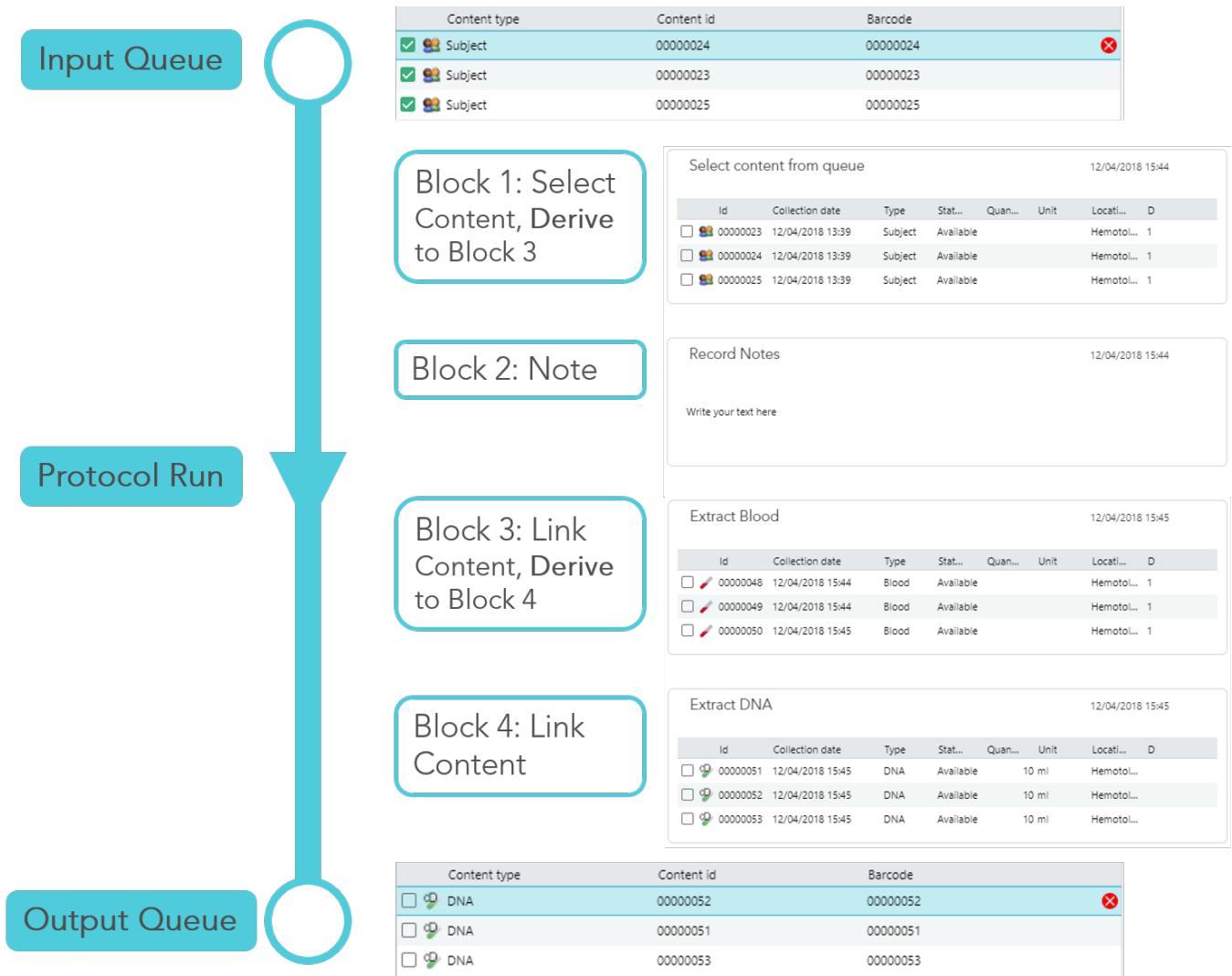
**Figure 6.2. Simple Protocol Steps Example**

In this example, content is selected in the first block but no further actions are taken on it. The content is tracked in the protocol run and is attached to the Output Queue.

Figure 6.3. Steps with Single Derivation



This example shows content selected in the first block and derived using a macro. Derived content is stored in a Link Content block, so this content was derived from block 1 into block 4. The content that comes to the Output Queue is the last content linked in the protocol run; in this case, the two derived urine from the last block. Even if block 3 and 4 were switched so that the Link Reagents block was last, Reagents are untracked content so the derived urine would be the content that comes to the Output Queue.

**Figure 6.4. Steps with Two Derivations**

The image above shows Content selected in the first block and derived into the Link Content block 3. Content can be derived as many times as needed in a simplified protocol, as seen with the block 3 samples derived into the Link Content block 4. The content linked last is what goes to the output queue. In this case, though Subject samples enter the workflow, the derived DNA samples from block 4 are attached to the Output Queue.

Blocks can be removed from a simplified protocol run in the same way as in the simplified ELN. Predefined blocks of type "Link Content" are the only exception.

## 6.2.5. Workflow chain of custody

Chain of Custody is a feature that stores all the information about the use of a content in a workflow. It is available by right-clicking on a Content in the Content module, and in the Order module in the bottom tab.

In the Content module the view consists of:

- **Tabs on the top:** Orders in which the records were involved (by link or creation) and order-less requests
- **Left Panel:** List containing all the requests linked to the selected order.
- **Middle Panel:** List of all the runs associated with the selected request.
- **Right Panel:** Contains compact information regarding the run:
  - **Input:** Input for the run
  - **Output:** Output of the run
  - **Results:** Any results created during the run
  - **Started and Finished By/On:** Details about the run creation and completion date and user
  - **Open Run:** A button that provides a view similar to the one in the Workflow module, except that everything is read only. From the Open Run window, the back arrow button at the top returns the user to the main chain of custody view.

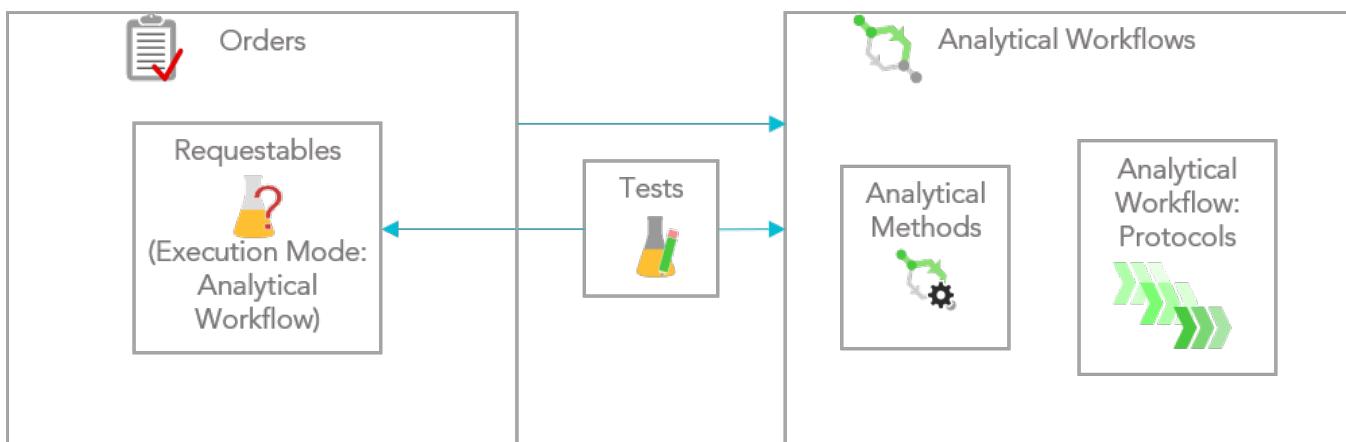
In the Orders module, the Workflow chain of custody is displayed as one of the available bottom tabs. It contains the same elements as in the Content module view described above, but relating to the order that is selected in the top left panel.

## 6.3. Analytical Workflows

Analytical Workflows allow results to specific Tests to be required on a sample navigating a workflow. They differ from standard workflows in the sense that on a standard workflow, one can request a sample to follow a given path in a workflow, and on an analytical workflow, one can request test results to be entered on a sample while the sample follows a given path in the workflow. For example, tests like quantifying the copper amount in water, or measuring the concentration of a medicine could be requested and results could be captured during a workflow. Analytical Workflows use the same building blocks as regular workflows, but they require slightly different configuration on most of those pieces so that analytical Tests can be requested in a workflow.

This module requires the same Access rights as regular workflows.

This section will cover the configuration and usage of Analytical Workflows and then provide a use case example to better illustrate how the building blocks fit together in this context.

**Figure 6.5. Building Blocks of an Analytical Workflow**

## 6.3.1. Configuration of Analytical Workflows

The Analytical Workflow is closely linked with the Tests that are run in its protocols.

### 6.3.1.1. Configure Tests

There are no specific requirements on test configuration to use Tests in analytical workflows. The execution of Requestables is what will cause the Tests to be related to analytical workflows, which will be covered below. See the *Tests* and *Test groups* sections to set up the desired Tests for use in an analytical workflow.

Analytical Tests that are scheduled to run through an analytical workflow via an order can have Rules that act on their results, and are set up the same way as for standard ELN. See the *Rules* section to set up any desired test Rules. All results will be checked against rules. Rule evaluations provide indicators when a result breaks a rule which can then be reviewed. In analytical workflows, the broken rules can be acknowledged and contents can be sent back in the workflow for results to a test to be repeated, which is covered in the *Result Protocol Steps* section.

### 6.3.1.2. Configure Requestables

Analytical workflows start to differ from traditional workflows with how to configure requests that can be added to content linked to an order. This is done with Requestables which are used to create pre-configured requests for orders. Analytical workflows require Requestables configured in the Requestables module to be able to request results and schedule specific Tests for content in an order.

① An easy way to automatically create a Requestable for each test is by enabling the Lab Setting "Create default Requestable for new test." This automatically creates Requestables for the product tests and specifications in the Requestables module. A Requestable can also be configured to request several tests simultaneously. For example, a Requestable called "Bacteriological water analysis" would require both tests "Escherichia coli" and "Pseudomonas aeruginosa."

Otherwise, the Requestables will need to be created manually in the Requestables module for each test by clicking "New" and entering the details from the Requestables section.

An option is available for Requestables to make it more convenient for lab users to select the right analytical methods when scheduling orders. If there is only one analytical method that could possibly measure the test, it will be suggested by default when the lab user schedules an order. However, if there are multiple analytical methods that could measure the test, selecting the default method in the Requestable test table causes it to be automatically suggested when a lab user schedules the order. If an analytical method is updated to no longer measure the test, the Requestable test table will also be updated to remove the default analytical method.

Requestables with "Analytical Workflow" execution mode are not visible in non-analytical workflows, non-analytical orders, or in the 'To Workflow' selection action on content. They are only used in the context of analytical workflows and their orders.

Requestable groups puts Requestables together like Test groups is used to group Tests, except that Requestables can be in multiple groups while tests can only be in one group. It is used to add single or multiple Requestables at one time, and also creates a better display for the user to choose Requestables from. If there are no Requestable groups configured, the user will see a dropdown of all of the available Requestables. If there are Requestable groups configured, the user will see group headers and the list of grouped Requestables, which makes it easier to see and choose individual categorized Requestables or the whole group. There is no difference in how Requestable groups are created as far as analytical workflows are concerned. However, do not group Requestables that have different execution modes together. See the *Requestable groups* section for more on how to create groups.

### 6.3.2. Workflow Management for Analytical Workflows

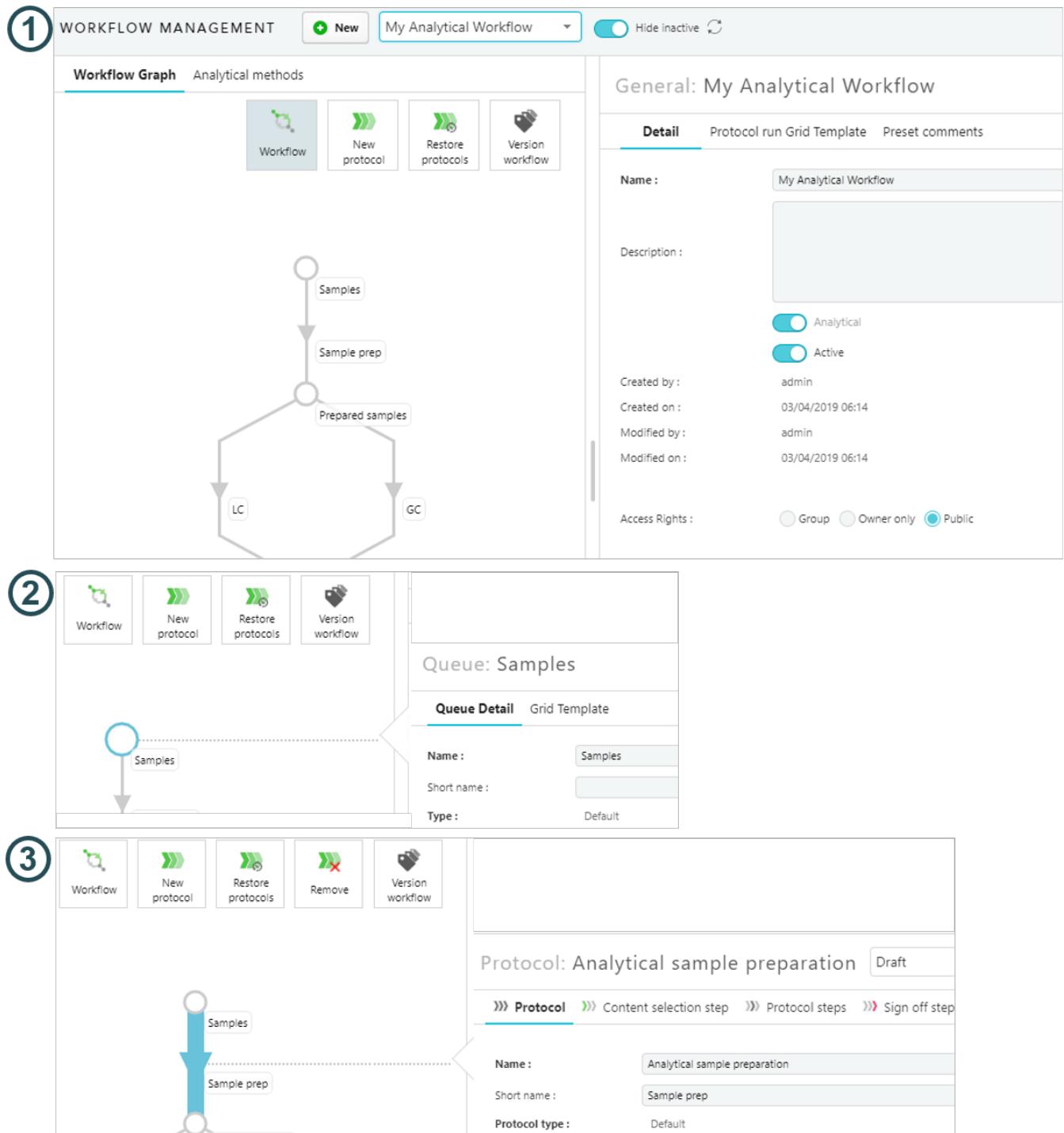
Analytical workflows are created and managed in the Workflow Management module. This module allows users to define protocols that contents will go through between queues, select the Tests that will be used in each protocol, and create Analytical methods for each workflow.

The Workflow Management module is divided into two side-by-side windows with a set of tabs on each side (see Figure 6.6). The left window has two tabs:

- **Workflow Graph:** Displays a visual representation of the selected workflow.
- **Analytical methods:** Displays the list of the Analytical methods associated to the workflow.

The right window contains different options depending on which tab is selected in the left window:

- **When on Workflow Graph tab:** The "General" window (1) is displayed when nothing has been selected yet. "Queue" (2) is displayed in the right window when a Queue is selected in the workflow graph, and "Protocol" (3) is displayed in the right window when Protocol is selected in the graph.
- **When on Analytical methods tab:** The list of Analytical methods, and details and options for a selected Analytical method are displayed.

**Figure 6.6. Analytical Workflow Areas**

To create a new analytical workflow, click the "New" button above the left window. This will open a new pop-up window where the name and the description of the workflow is defined.

- **Enable the "Analytical" option to create an analytical workflow.** Once this is enabled and a workflow Requestable or Analytical method is defined for the workflow, the created analytical workflow cannot be changed back to a standard workflow ("Analytical" cannot be disabled afterward).

### 6.3.2.1. Analytical Workflow Graph Tab

Analytical workflows are setup the same way as traditional workflows as far as the Workflow Graph tab is concerned. See the Workflow Graph section for more details.

Protocols are created the same way as those for other workflows, but there are some pieces of configuration that will help with the display and usage from the user's perspective.

- *Join Queue Type:* Due to the high amount of tests that can be run in an analytical workflow, using the Join Queue Type will cause all of the requests that are on the same content to be joined into a single element in the queue. Additionally, if the same content is sent to the workflow multiple times through different orders, all these requests will be joined into a single element in the queue as well. That way the content only needs to run once through the protocol. If Keep Queue Type is used instead, there will be one element per request in the queue, independent from the fact that several requests can be on the same content. (For example, subject 1001 is listed once for a Join queue, with weight, pH, and concentration as the tests that will be run. The same subject in a Keep queue is listed three times: 1001 - weight request, 1001 - pH request, 1001 - concentration request. This means the user would have to run the sample through the protocol once for each request, as different requests on the same content cannot be processed as part of the same protocol run.)
- *Queue Grid Template:* It is useful to configure the queue grid template so that users can view not only the content, the content type, and identifiers, but also information such as tests that were requested, products, order types, order barcodes, and more. An example of a grid template using these columns is covered in Step 7 of the *Use Case* section.

#### Result Protocol Steps

Each path on the workflow must be able to execute the Tests scheduled by an order, and an analytical workflow needs to have some result protocol steps configured for those Tests. Result protocol steps specify the Tests a given protocol step can measure.

Result type steps for analytical workflows are much the same as those for traditional workflows. Any of the desired Tests can be added if they can be performed in the particular protocol step. However, they have an additional option to "Only create results for the requested Tests" which is very useful in this context. Much of the time analytical labs have a huge amount of different tests that can be run on instruments and equipment. If "Only create results for the requested Tests" is not enabled, all of the tests are present in the Results step and can be filled out. Enabling this option narrows the Tests that are created to only the ones that were requested in the order. This is useful if a lab has 200 tests normally, but sometimes only a few of them need to be done.

Results can be created manually if the option is enabled, but manually created results cannot be repeated. Additionally, only results that have a request associated to them can be repeated (coming from tests that are requested, and not tests that just happen to be executed as well while going through the workflow).

### 6.3.2.2. Analytical methods Tab

Analytical methods define workflow paths that are configured to measure specific Tests. Analytical methods for the selected workflow are defined and edited in this tab. Multiple requests can be created with different start/end queues or different workflow paths. Click the "Add" button to begin.

The following fields are required:

- **Name:** A name to indicate the type of Analytical method.
- **Start Queue:** The first queue where the content of the scheduled order will be placed.
- **End Queue:** The queue following the last protocol in the Analytical method. After the last protocol run is completed, the content will be removed from the workflow (meaning it will not actually reside in this queue).

Other optional attributes are available:

- **Description:** A text box to enter any desired information about the Analytical method.
- **Calculate Tests Automatically:** The system will automatically determine which tests can be measured from the path of the requestable. When this is enabled, this keeps the Tests table on the Analytical methods tab and the one on the results step in sync - when Tests are added in the step, they are automatically added to the Tests table in Analytical methods. When disabled, the Tests an Analytical method can perform must be manually selected.
- **Tests:** This table displays the Tests that will be run during the selected Analytical method.
- **Only for Selected Roles:** Access to the Analytical method can be restricted to particular roles.

**Workflow Graph button:** Once the Analytical method has been created, the workflow graph can be selected using the Workflow Graph button (located at the bottom of the right window). If there is only one possible path from start to end queue, the path will automatically be selected and the graph doesn't need to be edited manually. The path that content takes as it moves through the workflow corresponds to the protocols that will be run. Every path for an Analytical method must have the test execution at some point. The paths may be added or removed by clicking on the queues and protocols in the diagram pop-up.

The setup should define which path the Analytical method will take automatically when the order is scheduled. In this case, the user does not choose the path when scheduling and the order moves through the path that was defined in the workflow diagram pop-up.

### 6.3.2.3. Orders Used for Analytical Workflows

Orders are handled in a similar manner to standard workflows but have a few differences. The default order type "Order Management" must be used to add a request for Tests that are located on an Analytical method path. Once the order is created, the user needs to follow these steps:

1. Link or create content for the order.
2. Select the linked content and add the request with the "Add Request" button. Choose from the Requestables from the Requestables module, which will select the test to be measured for the content.

3. Select the request in the Request Detail or Request Overview tab and edit the "Analytical method" that needs to be used to measure the requested test. If only one Analytical method is available to measure the test, or a default Analytical method is set on the Requestable, it will be pre-filled.
4. Schedule the Order to make the content(s) linked to the Order appear in the workflow start queue. While scheduling, the user can select multiple tests and assign them to users or update the Analytical method in batch, if needed.

For more details on Orders, see the *Orders* section.

### 6.3.3. Processing Analytical Results

Results for analytical workflows are entered on the Results tab of the result protocol step. After results have all been captured, there are three places the lab manager can review the results.

- From the workflow on the Results tab, the rule evaluation can be viewed by clicking the icon (1), or from the context menu (right-click or access with the three-dots menu). This opens a pop-up window where accepted values, errors for the broken rules, and rejected values are indicated.
- From the Order module on the Results Overview tab, the rejected and accepted values are also present.
- The Rule Evaluation module lists the results for tests that were mentioned in rules.

**Figure 6.7. Analytical Results Overview**

Content	Test	Value	Comments	Value entered on	Status	Modified on
WT-00000051	pH	7.6		26/04/2019 11:43	Verified	26/04/2019 11:43
WT-00000051	Weight	5 g		26/04/2019 11:13	Verified	26/04/2019 11:13

#### Rule Evaluation Outcomes

If a rule is broken, an error message will be present. The lab manager has a couple of options to deal with broken rules in analytical workflows.

The below options can be accessed in the related result's row, by right-clicking on a result row, or by selecting one or more results with the multi-select boxes. This allows results to be acknowledged, repeated, rejected, etc. in batch. Note that if a result with a broken rule and a result without a broken rule are both selected, only the options that are relevant to the status of all of them will be available as batch options. (Ex: Validate option will not be available because a result that breaks a rule is selected, and Restore is not available if a result that is not rejected or canceled is selected, etc.)

ⓘ An electronic signature will only be requested once for a batch selection of results if signature is required for the status transition.

- Click the repeat icon (2). Select where the content should start its journey over in the workflow. Provide your electronic signature. The message "Results reschedule successful. Con-

tent will not continue in workflow." is given and the content goes to the selected step to be retested. The new result will appear as a repeated test and there will be an associated repeat test event in the history. The result that is repeated will have an "is repeated" checkbox checked on it, while repetitions of a result (the new value) have "is a repeat" checked.

The repeat icon is not available for manually created results or results that were not requested (coming from a test that just happened to be executed along the workflow path).

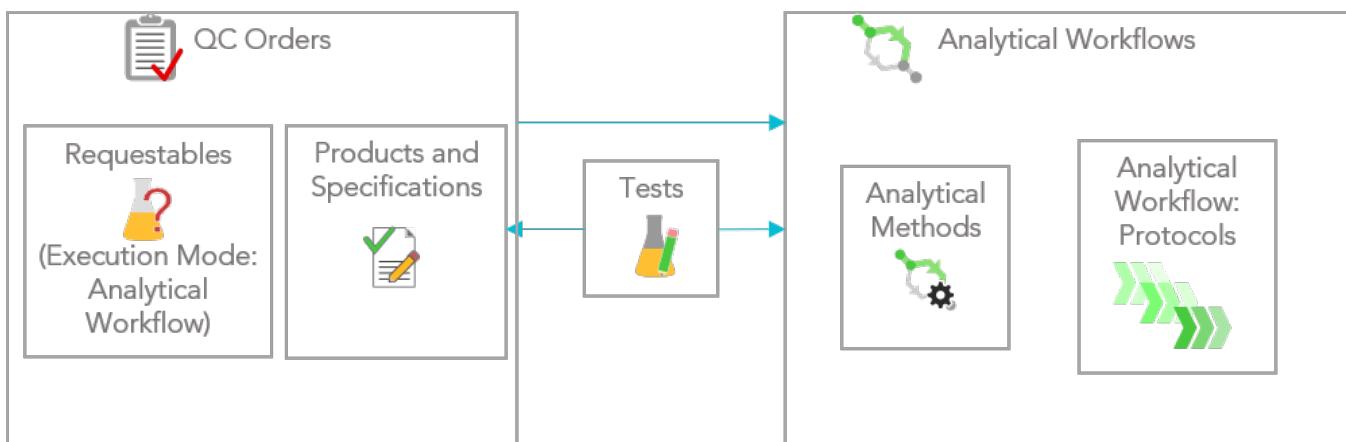
- The error can be acknowledged (3) to accept the rejection. A comment can be added with the electronic signature to note the outcome. There are any number of reasons to acknowledge an error, but some examples might be because a rule does not apply, a rule is incorrect and needs to be fixed, etc.
- The result can be rejected by right-clicking the result and selecting the "Reject" action. This changes the status of the result to "Rejected."

### 6.3.4. QC Orders for Analytical Workflows

QC Orders are useful for scheduling a list of tests to be executed and a set of Specifications on these test results to be checked on a content.

Standard rules on results can also be used with non-QC orders in analytical workflows. QC Orders are intended to only work with products, though standard rules will be checked in addition to the Specifications of a Product. QC Orders combine with Products and Specifications to provide a method to define layered rules with multiple different ranges that are applied to the same content. Failing one Specification does not fail all of the Specifications for the test. This adds a useful layer of complexity to testing that suits Analytical Workflows.

**Figure 6.8. Building Blocks of a QC Analytical Workflow**



To set up QC Orders for Analytical Workflows, these features need to be in place:

- Create the *Product Tree* that will be tested.
- Configure the *Specification* for each Product and sub-Product to define each required value range.

The Default Analytical method can be set on Product Specifications to make it more convenient for lab users to select the right Analytical methods when scheduling QC orders. If there is only one Analytical method that could possibly measure the test, it will be suggested by

default when the lab user schedules an order. However, if there are multiple Analytical methods that could measure the test, selecting the default method in the Specification's test table causes it to be automatically suggested when a lab user schedules the order. If an Analytical method is updated to no longer measure the test, the draft Specification test table will also be updated to remove the default Analytical method, but a versioned Specification test table will not change.

It is also possible for the administrator to combine this option with locking the Analytical method field so that it cannot be edited. To do so, a default Analytical method would need to be set for each required test, and the field "schd\_fk\_workflowRequestable" in the Fields module can be set to read only. In this way, when the lab user schedules an order, the default Analytical method will be selected for each requested test and cannot be edited.

- "QC Order" Order Types need to be created in the Order Types module to work in Order Management.
  - "Add all available requestables by default when registering order" will add the possible requests by default when the order is scheduled so the user doesn't have to choose each test and Specification from the Product manually.

Results for QC Orders will be checked against both Rules and Product Specifications.

#### 6.3.4.1. Using QC Orders for Analytical Workflows

In the Orders module, create a new Order using a QC order type. Link or create the content(s) as usual. If the content has a product draft associated to it, then the "Specifications to be checked" can be assigned. The latest specification version is proposed but the latest version of sub-products are available if something else needs to be selected. If the content has no product associated to it, then the latest version of all products from the Products and Specifications module are available.

ⓘ Contents associated to a versioned product cannot be linked to or created in a QC Order because versioned products are from the Study Design module and are not relevant to QC Orders. Only draft products are from the Products and Specifications module allowing for selection of a product version or one of its sub-products.

If "Add all available requestables by default when registering order" was selected on the Order type, the Order can be scheduled directly and the requests required by the Product will be scheduled automatically. An Analytical method will need to be selected for each request, and will be automatically selected if it is the only Analytical method available that can measure the required test. Otherwise, the user needs to choose the method manually. The QC Order content then goes to the workflow.

#### 6.3.5. Analytical Use Case

This use case covers a simple demonstration for an analytical workflow that measures the pH and weight of water that will be used to make Aspirin. Each part of the setup for the workflow is described below, as well as the rationale.

1. Two tests were created in the Tests module for this example.

##### Weight

- *DataType*: Quantity

- *Test Group*: Water Results
- *Dimension*: Mass
- *Unit*: g

### pH

- *DataType*: Decimal Number
  - *Test Group*: Water Results
  - *Show in result overview*: Enabled
2. The test results can be grouped into collapsible lists so the display is better for the lab manager later on.

### Test Groups

- *Name*: Water Results
  - *Use to group results*: Enabled
  - *Main test*: pH
  - *Seq no*: 10
3. These Requestables were added automatically when the tests were added (thanks to the lab setting) and should have the correct execution mode already.

### Weight Requestable

- *Name*: Weight
- *Type*: Default
- *Execution Mode*: Analytical method
- *Tests*: Weight
- *Default Analytical method*: water analysis (Can be entered once the workflow and Analytical method is created.)

### pH Requestable

- *Name*: pH
- *Type*: Default
- *Execution Mode*: Analytical method
- *Tests*: pH
- *Default Analytical method*: water analysis (Can be entered once the workflow and Analytical method is created.)

4. The Requestables can be grouped in the Requestable Groups module so the user has a categorized list to choose Requestables from when scheduling the order.

### **Requestable Groups**

- *Name:* Water Requests
- *Requestables:* pH, Weight
- *Show in order screen:* Enabled

5. General rules can be added that apply to all results of the selected test (and specific Content Types). Two will be created as checks that the user enters a reasonable value for the weight and pH.

### **Rule for pH Tests**

- *Name:* pH must be plausible
- *Type:* Result
- *Mode of evaluation:* Upper/lower limit
- *Test:* pH
- *Minimum:* 0
- *Maximum:* 14

### **Rule for Weight Tests**

- *Name:* Weight must be plausible
- *Type:* Result
- *Mode of evaluation:* Upper/lower limit
- *Test:* Weight
- *Minimum:* 0
- *Maximum:* (leave empty)

6. Now analytical tests have been created that can be requested in the workflow. But the lab also wants to test Product usage for Aspirin to be distributed in Australia and Switzerland. This uses configuration in the Products and Specifications module.

Aspirin: The main Product

- **Table 6.4. Main Product: Aspirin**

Name	Test	Minimum	Maximum	Default Analytical method
Weight Upper/Lower (Aspirin)	Weight	2	6	water analysis (Can be entered once the workflow and analytical method is created.)

Aspirin: The Sub-Product for Australia, which requires a stringent pH in addition to the weight specification inherited from the root aspirin product.

- **Table 6.5. Sub-Product: Aspirin for Australia**

Name	Test	Minimum	Maximum	Default Analytical method
pH Upper/Lower (Aspirin for Australia)	pH	6.5	8	water analysis (Can be entered once the workflow and analytical method is created.)

Aspirin: The Sub-Product for Switzerland, which requires an even more stringent pH.

- **Table 6.6. Sub-Product: Aspirin for CH**

Name	Test	Minimum	Maximum	Default Analytical method
Weight Upper/Lower (Aspirin)	pH	5	7	water analysis (Can be entered once the workflow and analytical method is created.)

7. The analytical workflow was built next to assign the Analytical method workflow path that Requestables can take through the workflow. A result step was included in which the requested tests will be fulfilled.

#### Workflow Management: Workflow creation

- *Name:* Water Preparation
- *Analytical:* Enabled

#### Workflow Management: Protocol creation

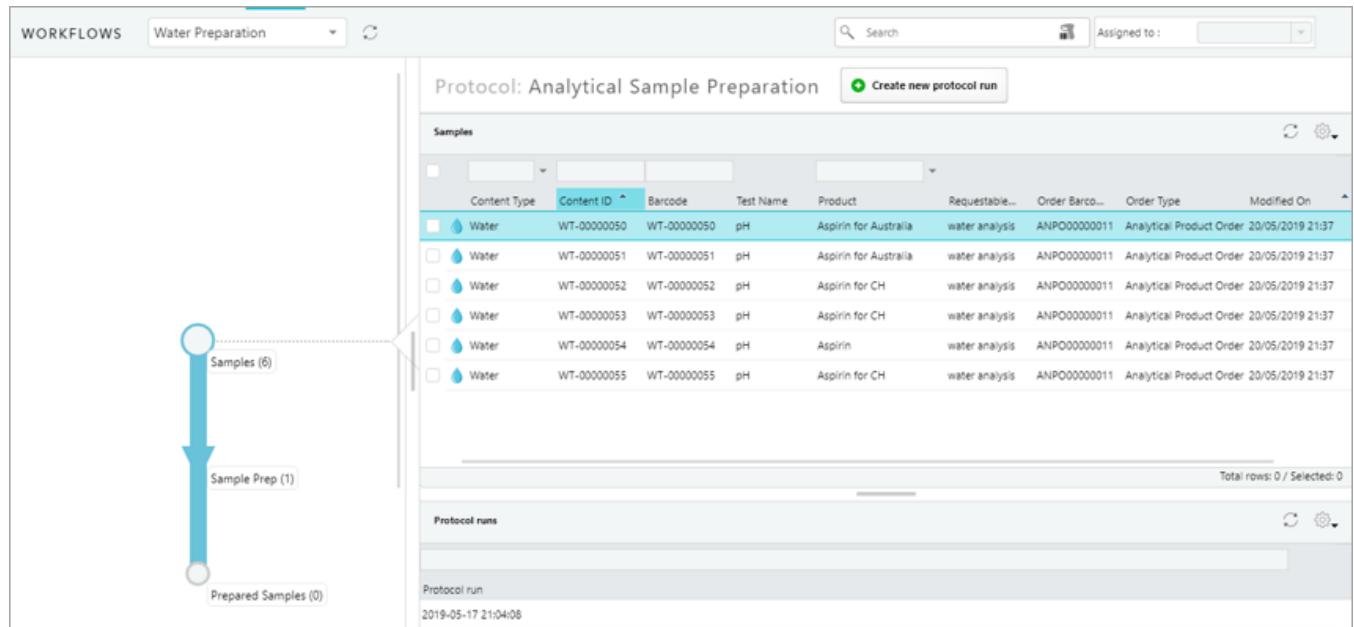
- *Name:* Analytical Sample Preparation

- *Short Name:* Sample Prep
- *Protocol Type:* Default
- *Queue From:* Samples
- *Queue To:* Prepared Samples

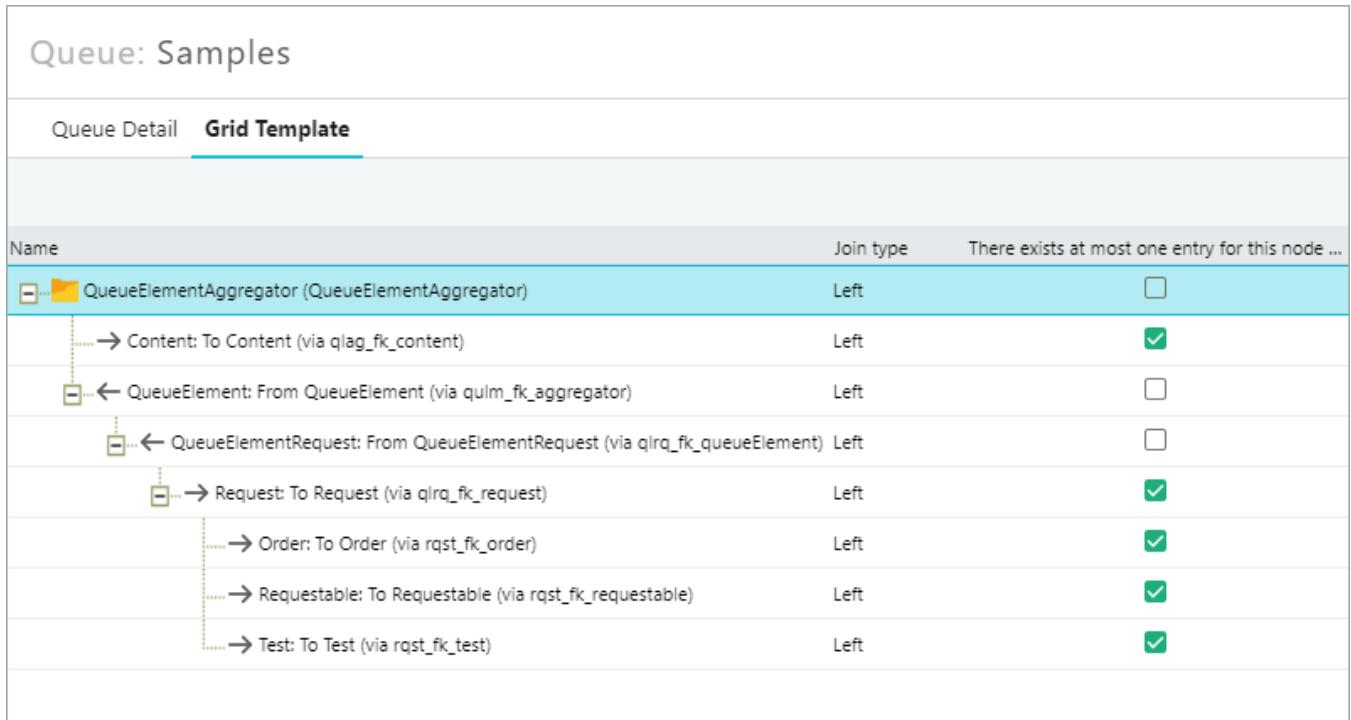
### Workflow Management: Queue Type

1. Click on the "Samples" Queue.
2. Change the Aggregator from Keep to "Join."
3. Click Save.
4. This will group requests on the same content in one single queue element so that the content only has to go through the protocol once - instead of a content going through the protocol once for each separate request.

**Figure 6.9. Water Preparation Workflow with Protocol Run Grid**



**Queue: Samples** - A helpful Grid Template to show Requestables and Tests in the Queue grid.

**Figure 6.10. Queue Grid Template Configuration****Table 6.7. Grid Template - Columns Table**

Meta Tree	Function	Field	Title	Seq No
Content	(leave blank)	Icon	Icon	10
Content	(leave blank)	Content Type (cntn_fk_contentT)	Content Type	20
Content	(leave blank)	Id (cntn_id)	Content ID	30
Content	(leave blank)	Barcode (cntn_barCode)	Barcode	40
Test	Comma-separated values	Label (test_label)	Requested Test(s)	50
Content	(leave blank)	Product (cntn_fk_product)	Product	60
Request	(leave blank)	Requestable (rqst_fk_requestableName)	Requestable	70
Order	(leave blank)	Barcode (ordr_barCode)	Order Barcode	80
Order	(leave blank)	Type (ordr_gk_orderType)	Order Type	90
QueueElement	(leave blank)	Modified On (qllm_modifiedOr	Modified on	100

**Workflow Management: Protocol Step creation**

The protocol result step is created to perform the requests for the analytical workflow.

- *Seq No:* 10
- *Name:* Measure sample physical characteristics
- *Type:* Result
- *Tests:* pH, Weight
- *Only create results for the requested tests:* Enabled
- *Can requeue in this step:* Enabled

**Workflow Management: Analytical method creation**

- *Name:* water analysis
- *Start queue:* Samples
- *End queue:* Prepared Samples
- *Calculate tests automatically:* Enabled
- *Workflow Graph:* Path from Samples to Prepared Samples
- *Tests:* When Workflow Graph is selected and saved, the tests table calculates automatically and adds pH and Weight tests.

8. Two Order Types were created in the Order Types module for the analytical workflow and for QC.

**Analytical Workflow Order Type**

- *Name:* Analytical Order
- *Type:* Order management
- *SLIMS generates a barcode for orders of this type:* Enabled
- *Barcode Mask:* ANO-#####

**Analytical QC Order Type**

- *Name:* Analytical Product Order
- *Type:* QC orders
- *SLIMS generates a barcode for orders of this type:* Enabled
- *Barcode Mask:* ANPO-#####
- *Add all available requestables by default when registering order:* Enabled

### 6.3.6. Using Analytical Workflows

This section will step through every part of an analytical workflow to show how it can be used from a user's perspective, assuming the setup follows the earlier sections' advice.

1. Navigate to the Orders module and click the "New" button. Select the analytical order (which has type: Order Management) and click Create.
2. Create or link Content in the top right panel. Choose the amount, location, type, and any other required relevant information. Click "Create" or "Link and back to Orders."
3. Click on the Request Overview tab in the top right panel to assign the Analytical method for each request. This can otherwise be done when scheduling the order. If there is only one Analytical method possible for each request, it will already be selected. If there are more than one, check the boxes to select all the tests (or use SHIFT + click to quickly check them all) and use the Assign Analytical method menu to assign the appropriate workflow path.
4. Click on the three-dot option menu next to the order in the top left panel, then click on Schedule Order. The pop-up to Schedule Order allows you to assign or adjust the Analytical methods of each request. Click Confirm.
5. Navigate to the Workflows module and select the Water Preparation workflow. There should be content waiting in the queue for each request in the order. If the queue is a Join type, there will be one content for each set of requests assigned to that content. If the queue is a Keep or Split type, there will be a content for each individual request, so the content ID may be present more than once.
6. Make the selection of content that should go through the protocol (if IDs listed are not unique), or use the top checkbox to select all, and click Create New Protocol Run.
7. Perform and complete each step until the Result step is reached.
8. On the result step, enter values in the Value column for each of the tests. As soon as this is done, it is possible to see whether a rule was broken. If the warning icon appears, click on it to view the issue. Click Next to proceed/Finish to complete the workflow.

**Figure 6.11. Results in Water Preparation step**

Content	Test	Value	Manual...	Status	Is a repeat	Is repeated	Created by	Created on	Modified by	Modified on
WT-000...	pH	3	<input type="checkbox"/>	Available	<input type="checkbox"/>	<input type="checkbox"/>	admin	17/05/2019 15:04	admin	17/05/2019 15:06
WT-000...	Weight	2.5 g	<input type="checkbox"/>	Available	<input type="checkbox"/>	<input type="checkbox"/>	admin	17/05/2019 15:04	admin	17/05/2019 15:09
WT-000...	Weight	3 g	<input type="checkbox"/>	Verified	<input type="checkbox"/>	<input type="checkbox"/>	admin	17/05/2019 15:04	admin	17/05/2019 15:09
WT-000...	Weight	4.7 g	<input type="checkbox"/>	Verified	<input type="checkbox"/>	<input type="checkbox"/>	admin	17/05/2019 15:04	admin	17/05/2019 15:09

9. The Lab Manager generally follows the user entry and views the results. Results can be viewed in the Orders module even after the content finishes the workflow and is no longer visible in the protocols.

10. In the Orders module, the Lab Manager clicks on the analytical order.

11. Click on the Result Overview tab in the top right panel.

12. Review the results: there are four actions to take. It is possible to select multiple records and use the options that appear at the top of the grid to repeat, acknowledge, or reject them all at once.

- You can view the rule evaluations by clicking the error message icon, or access them via the right-click context menu or the three-dot menu. This displays all of the rules that the results have broken, if any.
- You can validate the result by clicking the green check mark icon in the row.
- You can also send the test back to a workflow step to re-test by clicking the icon with the two blue arrows ("Repeat" icon). This will reschedule the test and add another result entry for that test so the user has the opportunity to recheck the values. Clicking "Repeat" brings up a pop-up for you to select the queue the content should go back to.
- You can reject the result by right-clicking on the row and using the "Reject" icon.

13. This will create an event for the History, and will keep any review comments, such as which countries the tested content is suitable for.

### 6.3.6.1. Using QC Orders in Analytical Workflows

This section will step through every part of an analytical workflow with QC Orders to show how it can be used from a user's perspective, assuming the setup follows the earlier sections' advice.

1. Navigate to the Orders module and click the "New" button. Select the analytical product order (which has type: QC Order) and click Create.
  2. Create or link Content in the top right panel. Choose the amount, location, type, product (if pre-selecting a product with the specifications to use), and any other required relevant information. Click "Create" or "Link and back to Orders."
  3. Edit the content grid to select the desired specification version in "Specifications to be checked." If a product was associated to the content, the specifications will be pre-filtered to those available for the product or sub-product.
  4. Click on the three-dot option menu next to the order in the top left panel, then click on Schedule Order. The pop-up to Schedule Order allows you to assign the Analytical methods of each request, though the Water Analysis method should already be selected for each. Click Confirm.
  5. Navigate to the Workflows module and select the Water Preparation workflow. There should be content waiting in the queue for each request in the order. If the queue is a Join type, there will be one content for each set of requests assigned to that content. If the queue is a Keep or Split type, there will be a content for each individual request, so the content ID may be present more than once.
  6. Make the selection of content that should go through the protocol (if IDs listed are not unique), or use the top checkbox to select all, and click Create New Protocol Run.
  7. Perform and complete each step until the Result step is reached.
  8. On the result step, enter values in the Value column for each of the tests. As soon as this is done, it is possible to see whether a rule was broken. If the warning icon appears, click on it to view the issue. Click Next to proceed/Finish to complete the workflow.
  9. The Lab Manager generally follows the user entry and views the results. Results can be viewed in the Orders module even after the content finishes the workflow and is no longer visible in the protocols.
10. In the Orders module, the Lab Manager clicks on the analytical product order or the analytical order.
11. Click on the Result Overview tab in the top right panel.
12. Review the results: there are four actions to take. It is possible to select multiple records and use the options that appear at the top of the grid to repeat, acknowledge, or reject them all at once.
- You can view the rule evaluations by clicking the error message icon, or access them via the right-click context menu or the three-dot menu. This displays all of the rules that the results have broken, if any.

- You can validate the result by clicking the green check mark icon in the row.
- You can also send the test back to a workflow step to re-test by clicking the icon with the two blue arrows ("Repeat" icon). This will reschedule the test and add another result entry for that test so the user has the opportunity to recheck the values. Clicking "Repeat" brings up a pop-up for you to select the queue the content should go back to.
- You can reject the result by right-clicking on the row and using the "Reject" icon.

13. This will create an event for the History, and will keep any review comments, such as which countries the tested content is suitable for.

## 6.4. SLIMS-NGS

SLIMS has specific configurations to support integration between the Workflow module and NGS, with particular focus on adaptability for Illumina systems. MiSeq, HiSeq, NextSeq, and NovaSeq implementations need plenty of configuration in order to be integrated efficiently, but regardless of which sequencer you use, the MiSeq package contains the configuration required to use SLIMS integration. The configuration of MiSeq, HiSeq, NextSeq, and NovaSeq executions are similar and differ only in the execution step, so the below explanation works for all of them. If a lab is using a combination of sequencers a great deal of the workflow configuration can be reused. This section steps through configuration in the order that each unit would naturally be encountered and built.

The following Role Access rights can be configured for use in this module:

**Table 6.8. Required Permissions for Setup with NGS Modules**

Access rights Category	Access rights Name	Purpose
Setup: Index Sequences	DNA Index Sets	This permission provides access to the DNA Index Sets module where DNA Indexes are associated to their sequences.
Setup: Index Sequences	Plate Index Design	This permission provides access to the Plate Index Design module where the DNA index set is mapped to the wells of a plate to provide digital location details.
Setup: Index Sequences	Sequencing Settings	This permission provides access to the Sequencing Settings module where the configuration to Illumina is set up.

### 6.4.1. NGS Package Installation

A package is available in the Package Browser module that hastens your configuration. The below steps can be used to install the MiSeq package.

1. Navigate to the Package Browser module.
2. Click the "Browse Locally" button.
3. Select the "MiSeq" package.
4. Click the "Import" button in the lower-right corner.
5. A pop-up window opens listing all of the entities that are included in the package. You can choose to import the custom field in the package as a new field, or if there's another record that could match in your SLIMS already, you can elect to use the existing one instead for any entities that are being imported.

① An existing record has to have the same name as the one in the package for it to work, because the configuration that's being installed with the package expects those particular names. For example, the HiSeq instrument type that is imported can only be matched with an existing instrument type that is also called "HiSeq."
6. Make your selections for each of the records to be imported.
7. Refresh SLIMS for the installation to take effect.

The package brings these entities:

- **Content Types:** DNA Library, Library Pool
- **Instrument Types:** MiSeq, HiSeq, NovaSeq, NextSeq
- **Fields:** target, gene, panel
- **Custom Fields for Barcodes:** i7, i5
- **Other Custom Fields:** sample sheet, sample sheet name, cartridge barcode, flow cell barcode, sent to instrument, index number, read type, read 1, read 2, custom primer 1, custom primer 2, custom primer index, reverse complement, adapter trimming, queue grid template, finish files, important files, sample sheet folder, results pickup folder, pickup max wait, plate display fields, nucleic acid type, assay, adapter, application, chemistry, workflow, illumina index adapters, lane, HiSeq lane count, SLIMS GATE flow generating the sample sheet, SLIMS GATE flow processing instrument runs
- **Test Forward FastQ and Test Reverse FastQ**
- **Custom Field Type Restrictions:** barcode i7 to DNA Library content type, barcode i5 to DNA Library content type, sample sheet folder to MiSeq, HiSeq, NovaSeq, NextSeq instrument types, results pickup folder to MiSeq, HiSeq, NovaSeq, NextSeq instrument types, pickup max wait to MiSeq, HiSeq, NovaSeq, NextSeq instrument types, finish files to MiSeq, HiSeq, NovaSeq, NextSeq instrument types, important files to MiSeq, HiSeq, NovaSeq, NextSeq instrument types, lane to Library Pool content type, SLIMS GATE flow generating the sample sheet to protocol step, SLIMS GATE flow processing instrument runs to protocol step
- **Custom Field Result Value Fields:** Forward FastQ, Reverse FastQ
- **Custom Field results path for HiSeq pools**
- **Custom Field run statistics summary for HiSeq**

- **Protocol Types:** Library Preparation, MiSeq Execution, HiSeq Execution, NovaSeq Execution, NextSeq Execution, Analysis, NextSeq Basespace Execution

## 6.4.2. Creation of NGS Instruments

The sequencer instrument types are now installed from the package, but the instruments need to be created. Follow the steps to add an Instrument of the Illumina type you're targetting using the instructions that would apply for any other instrument, like in the Instruments section. The package adds some NGS-specific fields that will also need to be filled in:

1. **Finish File:** (Optional) There is a default hardcoded behavior for each sequencer type that is applied to indicate when the sequencer run is completed. The finish file gets attached to the protocol run step when the Illumina instrument runs. If the default behavior does not suit the customer's needs, or they use a different file as the finish file, a single file name can be entered for "Finish File." SLIMS will look for that file and change the sequencer run status to "Done" when the file is found in the watched folder.

① Only finish files that will be in the root instrument run folder will be found by SLIMS, so files that will be present in a subfolder or parent folder cannot be used as a custom finish file.

2. **Important Files:** (Optional) There may be default important files for each sequencer type that get attached along with the finish file to the protocol run step when the Illumina instrument runs. SLIMS can look for files in addition to the default important file as well. In order for SLIMS to know what files are important, a list of file "matchers" separated by commas can be entered for "Important Files." Matchers include the parts of the file name to look for and can include wildcard characters. For example: *Run\*.xml,samplesheet\_id\_????.csv*.

① Only important files that will be in the root instrument run folder will be found by SLIMS, so files that will be present in a subfolder or parent folder cannot be used as custom important files.

3. **Results Pickup Folder:** (Required) Enter the path into the root folder for the instrument run results pickup. The pickup routine will look into subfolders two levels deep for the relevant finished run folders.

4. **Sample Sheet Folder:** (Required) Enter the path of the folder where the sample sheet file will be created when users click the "Send to Instrument" button in the sample sheet step of the workflow.

If any content types outside of DNA Library and Library Pool are needed, such as an "FFPE" type that can be derived into DNA, and DNA that can be derived into DNA Library, create those in the Content Types module.

## 6.4.3. DNA index sets

The next step to set up NGS configuration in SLIMS is to create the DNA Index Sets. This module gives you tools to create the barcode adapter, or unique identifiers that represent the digital sequence of DNA, so that SLIMS and the sequencer can understand the index.

The DNA index sets module window is made up of two sections. The left section contains the list of DNA index sets with their instrument type and the right section contains one tab that

shows details about the selected set and another tab where DNA indices and their associated sequences can be added, edited, and viewed. A DNA index consists of a name, a sequence, and a type. They can be added using the "Add" button at the top of the DNA index tab.

① A validation check will run to ensure that the combination of Sequence and Type is unique in a Set.

Each DNA Index expects:

- **Name:** Name of an index (ex: N701)
- **Sequence:** DNA sequence fragment corresponding to the index (ex: TAAGGCGA)
- **Type:** i5, i7, or both if applicable
- **Active:** Indicates whether this index can be used or not

A fast way to create new DNA Indexes is to install them from the SLIMS Store. There are several packages available for the most common indices on the market. The NGS category in the Store has the Illumina Nextera Library Prep Kit for MiSeq and Nextera for MiSeq, for example.

The packages include:

- Plate Index Design
- Instrument Type
- DNA Index Set
- Sequencing Settings

Otherwise, the "Create Template File" and "Import" buttons are available to facilitate importing multiple DNA index sets. The generated Excel file is specific to the DNA index set that was used to generate the template file.

#### 6.4.4. Plate index designs

The next module that needs to be configured allows you to map the positions of a DNA Index Set to the wells of a plate, such as a 96 well plate, for labs that use plates. The Plate index design module is divided into two panels. The left panel contains the list of Plate index designs while the right contains details about the selected Plate index design. The DNA index location combines the plate design with the DNA index from the selected DNA index set. This assigns the i5 and i7 indices of each position/location in the plate.

A Plate Index Design consists of:

- **Name:** It is suggested to use the index brand in the name for clarity
- **DNA Index Set:** The DNA Index Set restricts which indices can be assigned to positions in the plate design. Only indices belonging to the selected set can be used.
- **Allow Skipping Indices:** It is not frequently needed, but this option allows skipping indices to leave wells empty if needed, or disallows skipping if the option is disabled.

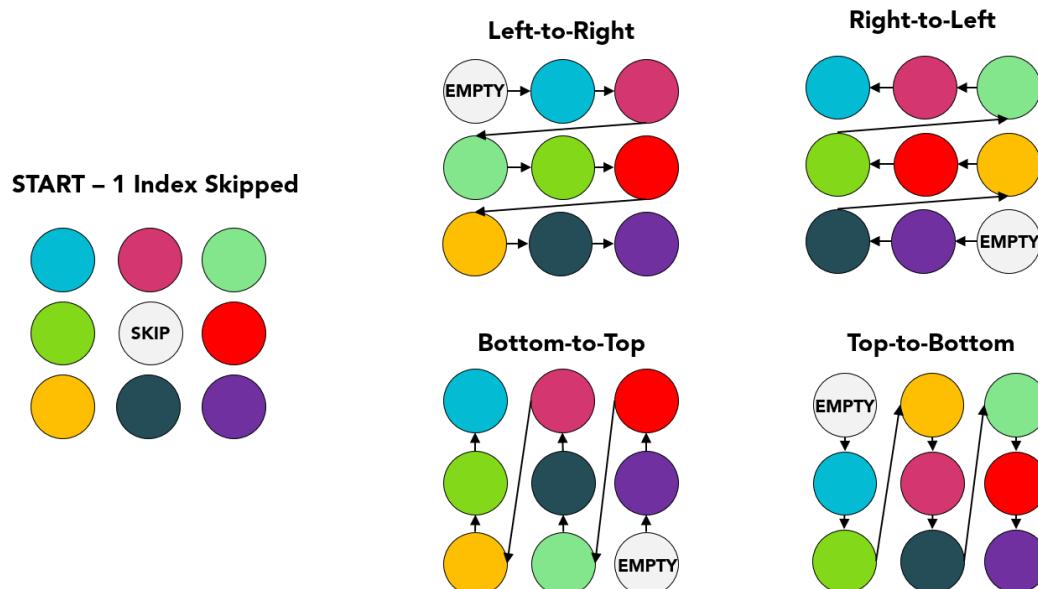
- *Skipped Index Fill Method*: If Allow Skipping Indices is enabled, the direction that the indices are applied to a plate can be adjusted. This is irrelevant unless at least one index is skipped. If skipping indices, the direction in which they are skipped can be chosen with "Skipped index fill method." Whatever selection is made is applied during the plate assignment execution.
- *Do not check for unicity*: By default, the combination of the position and Index 1 and Index 2 needs to be unique in a set. While an index may need to be repeated for multiplexing, the exact same combination of index 1 and index 2 cannot be repeated. This can be turned off by enabling "Do not check for unicity."

The options for the skipped index fill method are:

- *Right-to-Left*: Remaining indices are filled by row, packed toward the top-left.
- *Left-to-Right*: Remaining indices are filled by row, packed toward the bottom-right.
- *Bottom-to-Top*: Remaining indices are filled by column, packed toward the top-left.
- *Top-to-Bottom*: Remaining indices are filled by column, packed toward the bottom-right.

The figure below shows an example of each of the fill methods to show how the indices will fill the gap left by the skipped index.

**Figure 6.12. Skipped Index Fill Method Order**



## 6.4.5. Sequencing settings

The last step before moving on to workflow configuration is the Sequencing Settings module, which allows you to define the advanced parameters for your run. This includes settings that allow SLIMS to work with your machine, such as how many cycles there are for sequencing and how many reads there are per second. The Sequencing settings module consists of two sections: the left one contains the list of Sequencing settings and the right one displays details about the selected Sequencing settings. Sequencing settings are defined by the following pa-

rameters (some of them come from information about the instrument/software itself, especially those referring to a file in the SamplePrepKits or the Application folder):

- **Name:** Name of the Sequencing settings
- **DNA index sets:** Reference to existing DNA index sets. This is required in order to execute a protocol of type NGS: Library preparation
- **Illumina Assay:** The assay (reagent kit) that will be used. This information is in the assay file in the SamplePrepKits folder under the section *[Name]*
- **Illumina Adapter:** Information obtained in the assay file in the SamplePrepKits folder under the section *[Settings]*. The DNA sequence comes from the line that starts with "Adapter." If a second line starting with "AdapterRead2" is present, the Illumina Adapter value is a concatenation of both sequences separated by a ";" (semicolon).

Example:

```
[Settings]
Adapter AGATCGGAAGAGCACACGTCTGAACCTCCAGTCA
AdapterRead2 AGATCGGAAGAGCGTCGTAGGGAAAGAGTGT
```

then the value is

AGATCGGAAGAGCACACGTCTGAACCTCCAGTCA;AGATCGGAAGAGCGTCGTAGGGAAAGAGTGT

- **Illumina Chemistry:** The type of chemistry that will be used for sequencing. Select "Ampli-con" for dual indexing. Otherwise, choose "Default."
- **Illumina Application:** Depends on the application and what it does, such as sequencing for research and many other things. The machine will run differently depending on the application entered. This information is in the assay file in the Application folder under the section "*[Display Name]*."
- **Illumina WorkFlow:** The workflow for results pickup. This information is in the assay file in the Application folder under the "*[Workflow Name]*" section.

The link to MiSeq information refers to the folder structure of the Illumina Experiment Manager (*link to download*) [[http://support.illumina.com/sequencing/sequencing\\_software/experiment\\_manager/downloads.html](http://support.illumina.com/sequencing/sequencing_software/experiment_manager/downloads.html)] . They are also valid for NextSeq and HighSeq sequencers.

① The following values are used in the example that is presented in the MiSeq configuration:

- *Name:* TruSight HLA v1
- *Well Count:* 1
- *DNA index sets*
- *Illumina Assay:* TruSeq LT
- *Illumina Adapter:* AGATCGGAAGAGCACACGTCTGAACCTCCAGTCA+AGATCGGAA-GAGCGTCGTAGGGAAAGAGTGT
- *Illumina Chemistry:* Default

- *Illumina Application*: FASTQ Only
- *Illumina WorkFlow*: GenerateFASTQ



The graph to the left shows a simple workflow covering the entire NGS data management process. Dots represent queues and arrows represent protocols. The numbers in the parentheses correspond to the number of samples in a given queue and the number of open runs of a given protocol.

**"DNAextr"**: This is the DNA extraction protocol. In SLIMS, this is equivalent to a derivation of the incoming content (a sample or FFPE) into DNA content.

**"libPrep"**: This is the DNA library preparation (NGS Library Preparation) protocol that it is specific to NGS. It combines the use of DNA



Index Set with the Plate Index Design  
It is the primary method for assigning NGS indices to each prepared library.

**"MiSeq"**: This protocol generates the sample sheet, places it on the MiSeq instrument PC, and then fetches the results. This step can be set up for a HiSeq execution as well.

**"Galaxy"**: The fourth protocol is dedicated to secondary analysis. Currently, Galaxy and GenePattern analysis are integrated with SLIMS.

**"Variants"**: The final protocol is dedicated to tertiary analysis and variant calling.

There is an advantage in using a Workflow instead of the ELN: Content from different protocol runs are grouped depending on the type of protocol or instrument used.

The configuration and implementation of each element is explained in the following sections. Two other features provide important information for workflows: the *Workflow Chain of Cus-*

today groups information about the different Workflow runs performed on Content, and the *Runs* module has a simple filter that enables users to query runs that were executed in both the ELN and the Workflow module.

## 6.4.6. MiSeq, HiSeq, NextSeq, and NovaSeq Implementations

MiSeq, HiSeq, NextSeq, and NovaSeq implementations need specific configuration in SLIMS in order to be implemented efficiently. This section explains how to configure a workflow with a general NGS execution and how to use it. The configuration of each type of NGS execution differs only in the execution step. A lot of the workflow configuration can be reused if a lab is using both a MiSeq and a HiSeq, NextSeq, or NovaSeq execution.

### 6.4.6.1. General Implementation Configuration

The following procedure configures a general NGS implementation in SLIMS. The MiSeq execution serves as an example.

1. The local package called "*MiSeq*" needs to be downloaded using the Package Browser module in SLIMS.
2. An *Instrument* needs to be added in the Instruments module and given the necessary information for results pick up and sample sheet folder structure. If needed, the finish file and important files can also be entered to define what files get attached to the protocol run step, and what finish file designates the run as "Done." For MiSeq we select the "*MiSeq*" instrument type from the package (the one with the helix icon).
3. Content Types DNA Library and Library Pool come with the package, but any additional content types that are needed should be added. In this case, the samples "FFPE" (that can be derived into DNA), and "DNA" (that can be derived into DNA Library) are added to the available content types.
4. Create an Order Type called "NGS Order" with the "Type" field set to "Workflow orders."
5. Define a DNA index set in the DNA Index Set module (see DNA index sets).
6. Define a Plate index design (see Plate index designs)
7. Define a Sequencing settings (see Sequencing settings)
8. A workflow called 'SLIMS-NGS' has to be defined and the following protocols have to be added:

Protocol	Queue (from)	Queue (to)
DNA extraction	Queue 0: Samples	Queue 1: DNA
DNA Library Preparation	Queue 1: DNA	Queue 2: DNA Library
MiSeq Execution	Queue 2: DNA Library	Queue 3: FastQ
Galaxy Analysis	Queue 3: FastQ	Queue 4: Variants
Variant Analysis	Queue 4: Variants	Queue 5: End Queue

9. These details describe the protocols in the graph above:

- A DNA extraction protocol containing one step of type derivation. A macro can be configured to simplify the derivation process. This macro will appear as the default macro option when content is selected in the derivation step if the "Restrict and specify extra options about macros" checkbox is selected and the macro is added to the list.
- A DNA Library Preparation protocol of type NGS: Library Preparation (see NGS: Library preparation).
- An Execution protocol of the particular NGS type:
  - For the MiSeq or NextSeq setup see NGS: MiSeq execution / NGS: NextSeq execution.
  - for the HiSeq or NovaSeq setup see NGS: NovaSeq execution / NGS: HiSeq execution.
- An Analysis protocol of type Analysis with the required information entered in the protocol step (see Analysis).
- A Variant analysis protocol of type Default that contains a step of type Variant observation.

10. The queues mentioned in the graph above have to be created. These are named and created by the user during the creation of a new protocol:

- Queue 0 (Type : Default, Aggregator: Join) Samples
- Queue 1 (Type : Default, Aggregator: Split) DNA
- Queue 2 (Type : Default, Aggregator: Keep) DNA Library
- Queue 3 (Type : Default, Aggregator: Keep) FastQ
- Queue 4 (Type : Default, Aggregator: Keep) Variants
- Queue 5 (Type : Sink, Aggregator: Keep) End Queue

11. Create a requestable in the Requestables tab of the workflow (in the Workflow Management module) called *NGS request* with the following attributes:

- **Type:** Workflow
- **Workflow:** SLIMS-NGS
- **Start Queue:** Queue 0: Samples
- **End Queue:** Queue 5: End Queue
- **Target:** Sequencing Settings. The "Fixed Sequencing Settings" option can be used also. This means that the target sequencing settings cannot be selected when adding a request. If enabled, the requestable will display as a checkbox in the Add Request window. If disabled, the requestable will display as a dropdown in which multiple sequencing settings can be selected.

### 6.4.6.2. NGS: Library preparation Protocol

The *NGS: Library preparation* protocol contains a unique content selection step where samples are placed on a plate. On the left is a grid showing the contents of the input queue. This grid can be customized using the *Grid Template* tab of the *Content selection* step tab in the Workflow Management module. Each element should correspond to one requested panel. The "Display Field" option controls the value displayed on the wells.

The "Index Allocation" protocol step is added by default and cannot be removed. The "Display Field" option controls the value displayed on the wells here as well.

On the right side, a 96-well plate is displayed with each well containing a Queue Element. Queue Elements (content in the queue) can be dragged over and assigned positions in the open wells in this step.

⚠ If protocols are added between the Library Preparation and the Sequencer Execution protocol steps, the option to see the libraries as a plate in the sequencer execution content selection step is unable to represent them visually. This functionality can only fetch plates from one protocol previous, so keep in mind that if protocols are inserted between Library Preparation and Sequencer Execution steps, a visualization option other than Plate will need to be used.

### 6.4.6.3. NGS: MiSeq execution / NGS: NextSeq execution Protocol

This protocol has three steps (by default): The first a MiSeq Run preparation step / NextSeq Run Preparation step for content selection, then a Sample Sheet preparation step, and finally a MiSeq run / NextSeq Run step. Additional steps may be added.

The steps in this protocol all differ from the default protocol step types in the following ways:

- The content selection step has both a list and a plate design selection.
- All step options are default values for fields inserted in the generated sample sheet. The Sample Sheet generation step asks for instrument parameters such as the assay configuration (number of indices, read type, cycle read 1 and 2, adapter trimming), custom primers (1, 2, and custom index primer) or application configuration (reverse complement).
- If you have a custom SLIMS GATE Flow created by a SLIMS Engineer to generate the Illumina sample sheet, the SLIMS GATE Flow can be selected from the "SLIMS GATE flow generating the sample sheet" dropdown. A flow can be selected if SLIMS GATE is running and has a SLIMS GATE Flow present in your SLIMS that is for usage in a workflow. If the flow is selected in this step, clicking the "Confirm" button in the sample sheet at run-time will use the flow to generate the sample sheet with customization. If it is not selected, the protocol will create the normal sample sheet.
- The MiSeq run / NextSeq Run step has no parameters to set up, and runs the sequencer-specific processing. However, if you have a custom SLIMS GATE Flow created by a SLIMS engineer to perform custom actions on the results of the run, the SLIMS GATE Flow can be selected from the "SLIMS GATE flow processing instrument runs" dropdown. A flow can be selected if SLIMS GATE is running and has a SLIMS GATE Flow present in your SLIMS that is for usage in a workflow. If the flow is selected in this step, SLIMS attempts to execute the SLIMS GATE flow once for each instrument run when the run is updated to the "Done" status. If two instrument runs of the same instrument are finished ("Done") in the same round, the SLIMS GATE flow is executed once for each, but with different arguments. If a flow is not selected, the protocol will run without the flow's additional processing.

The SLIMS GATE flow may make use of the result pickup location field that is stored on the instrument run. When the instrument run is completed, the result pickup location is stored in the "nsrn\_cf\_resultFolder" custom field for instrument runs.

#### 6.4.6.4. NGS: NovaSeq execution / NGS: HiSeq execution Protocol

This protocol has three steps (by default): The first is a NovaSeq run preparation step / Hiseq run preparation step where content is selected, then a Sample Sheet preparation step, and finally a NovaSeq run step / Hiseq run step. Additional steps may be added. This covers the 2 or 4 lanes for a NovaSeq flow cell on which contents can be inserted and involves the lane attribution option to set the number of lanes to 2 or 4, or the 2 or 8 lanes for a HiSeq flow cell to set the number of lanes to 2 or 8 respectively.

- The content selection step has both a list and a plate design selection.
- All step options are default values for fields inserted in the generated sample sheet. The Sample Sheet generation step asks for instrument parameters such as the assay configuration (number of indices, read type, cycle read 1 and 2, adapter trimming), custom primers (1, 2, and custom index primer) or application configuration (reverse complement).
- If you have a custom SLIMS GATE Flow created by a SLIMS Engineer to generate the Illumina sample sheet, the SLIMS GATE Flow can be selected from the "SLIMS GATE flow generating the sample sheet" dropdown. A flow can be selected if SLIMS GATE is running and has a SLIMS GATE Flow present in your SLIMS that is for usage in a workflow. If the flow is selected in this step, clicking the "Confirm" button in the sample sheet at run-time will use the flow to generate the sample sheet with customization. If it is not selected, the protocol will create the normal sample sheet.
- The Hiseq run / NovaSeq run step has no parameters to set up. However, if you have a custom SLIMS GATE Flow created by a SLIMS engineer to perform custom actions on the results of the run, the SLIMS GATE Flow can be selected from the "SLIMS GATE flow processing instrument runs" dropdown. A flow can be selected if SLIMS GATE is running and has a SLIMS GATE Flow present in your SLIMS that is for usage in a workflow. If the flow is selected in this step, SLIMS attempts to execute the SLIMS GATE flow once for each instrument run when the run is updated to the "Done" status. If two instrument runs of the same instrument are finished ("Done") in the same round, the SLIMS GATE flow is executed once for each, but with different arguments. If a flow is not selected, the protocol will run with the default behavior.

The SLIMS GATE flow may make use of the result pickup location field that is stored on the instrument run. When the instrument run is completed, the result pickup location is stored in the "nsrn\_cf\_resultFolder" custom field for instrument runs.

#### 6.4.6.5. Analysis Protocol

This protocol has a single step that, when configured, connects to an analysis server. Current analysis options include Galaxy and GenePattern. The following parameters must be defined for the given analysis tool:

- Galaxy
  - Server Configuration: Host and API key from Galaxy server.

- Workflow Configuration: Selection of workflow, which comes from the Galaxy workflow (eg RNA-Seq STAR Mapping)
- Inputs: Link between SLIMS and Galaxy. Link the correct test or attachment in SLIMS or file path to the corresponding input for Galaxy.
- Outputs: Link between SLIMS and Galaxy. Link the correct test in SLIMS to the corresponding output from Galaxy.
- GenePattern

#### 6.4.6.6. NGS Implementation Use

These steps should be followed to execute the NGS Workflow without problems:

- Create an Order in the Order Module of type "*NGS Order*", then link or create samples in the order. Select the samples and click "Add Request" (located above the content grid), then select "*NGS Request*". Choose a panel and click "Create." Now right-click the order in the left window and schedule the order. The samples in the scheduled order should now show up in Queue 0 of the *SLIMS-NGS Workflow*.
- Click the first protocol (the line with the arrow) and create a new run of the protocol where new DNA will be derived from the input samples.
- The second protocol (DNA library preparation) retrieves the DNA samples and prompts the user to place them in a well plate exactly as they are in the real plate in the lab.
  - ① The content type being used as input for library preparation needs to be derivable so it can be derived into DNA Libraries.
    - In the Library Preparation step, Queue Elements can be dragged from the grid to the plate and from the plate to the grid. Multiple elements can be dragged all at once and will be placed from the cursor's position in the closest open wells. They fill row-wise or column-wise set by the checkbox labeled "Fill by Column" above the plate. Multiple elements can be selected by clicking while holding down the CTRL or Shift key, and then can be dragged and dropped into the plate or grid. Otherwise, the option "Use checkboxes to drag samples" drags only the samples that have their checkboxes selected.

A visual representation of the plate can be printed by clicking the 'Printer' button.

- In the next step, select a "DNA Index Plate Template" that will apply a plate design; in other words, it will apply DNA indices to each sample and derive all contents into DNA Libraries automatically. All input contents need an output library. This means every position with a content needs to have at least one index assigned to it on the plate design being used. Applied indices will be stored as the fields "Barcode - i5" and "Barcode - i7" on the DNA Library record.

Once a design has been applied, the user can alter the specific indices on a well by selecting it and clicking the "Change indices of selected wells" button, or by double-clicking the well. The user can only use indices belonging to the index set linked to the plate design being used. Indices can only be changed on a position contained in the plate design (ex: with at least one not-null index in the design).

A warning appears if the same combination of index is used in two wells.

Libraries are created automatically in this step, but will fail if there are required fields without default values, or if the initial content being used cannot be derived.

- In the third protocol, a MiSeq, HiSeq, NextSeq, or NovaSeq run is prepared depending on the chosen workflow path. Content can be selected from the plate and put in the flow cell. The queue of contents can be visualized in three ways: a list, a plate, and by location. Using the dropdown, the user can view either of the three visualizations:
  - **List View:** The usual queue grid view.
  - **Plate View:** The dropdown lists all of the plates that contain at least one queue element.
  - **Location View:** The dropdown lists all locations that contain at least one queue element, that have positions, and that have less than 24 rows and columns. The view of Locations is similar to the one displaying samples inside a plate. The difference is that they are shown in the context of their location instead. The location shows samples that are not in the workflow as well, but those samples are gray and cannot be dragged and dropped. The samples in the workflow are blue and interactive.

The DNA Libraries can be pulled to the run selection grid (right side). Those are the samples that will be sequenced in the run. Content for single lane instruments can be dragged and dropped into the single section on the right side. However, there are as many sections on the right side as there are lanes for multiple lane instruments. Contents can be dragged to any of the lanes, and can be dragged to multiple lanes unless they are of DNA Library Pool type. SLIMS will aggregate and track the contents even when they are dragged to multiple lanes. Content can also be dragged from one lane to another.

- In the next step of the protocol, the sample sheet that is required by the sequencing instrument is generated when the "Send File to Instrument" button is clicked. It can then be either downloaded to the computer or found in the Sample Sheet folder. The actions for the sample sheet step are:
  - **Flow cell barcode:** An additional text field on the sample sheet.
  - **Sample Sheet Name:** Name of the generated file.
  - **Select Instrument:** The instrument tied to the configuration that SLIMS needs so it knows where to upload the sample sheet to and which folder to monitor for result files.
  - **Generate Sample Sheet:** Opens a pop-up where the user can alter step parameters at run-time. Chosen values are stored on the Instrument Run for subsequent generations. The generated sample sheet is also stored as a string on the Instrument Run and displayed in the lower right text area.
  - **Download:** Downloads the latest generated sample sheet to the user's computer.
  - **Send File to Instrument:** Sends the sample sheet file to the configured sample sheet folder on the instrument.

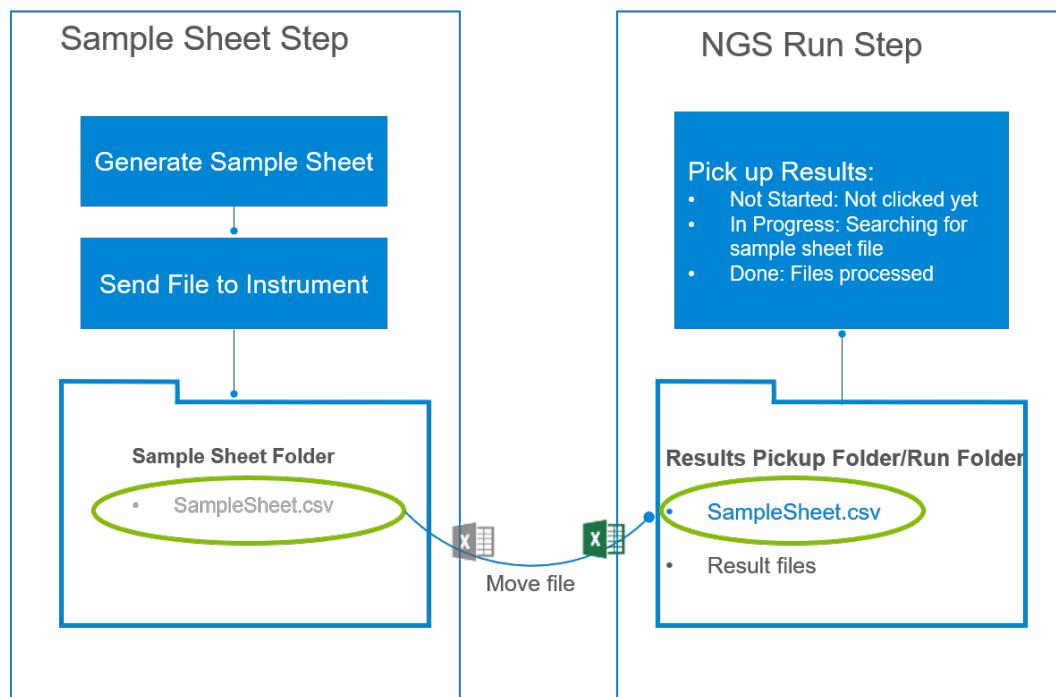
The behavior of sending the file to the instrument is described below:

1. The sample sheet file is named "sample sheet name".csv by SLIMS. It contains a GUID that links the file to the specific protocol run step in SLIMS.

2. When the user clicks "Pick up Results," the sample sheet file is put into the Sample Sheet Folder of the computer mounted on the sequencer. The generated file has to be moved into the Run folder by the user. The Run folder is the subfolder associated with the Run inside the Results Pickup folder.
3. SLIMS watches for the Instrument Runs associated to Instrument(s) in the "Waiting" status and filters out instrument runs that have timed out (which get set to the "Timeout" status, or for which protocol runs have been cancelled (which get set to the "Failed" status). This leaves a list of active instrument runs with the "Waiting" status which correspond to instruments. The inactive instruments are filtered out, and SLIMS finds the folders that must be watched based on the Waiting instrument runs of those active instruments. The monitoring service looks for new files in the Instrument's Pick up Results folder that it is watching. It reacts to directory creation and file creation to look for particular completion files. This can be the default finish behavior respective to the sequencer run type, or the file matching the "Finish File" name that was configured for each instrument in SLIMS:
  - **HiSeq:**
    - SLIMS looks for the default finish file CopyComplete.txt, or the specified finish file. The default important files are the First\_Base\_Report.htm and any specified important files.
    - The results of the sequencer are raw data that SLIMS does not use, but are available for the customer to demultiplex/analyze. SLIMS picks up the run information in a combination of xml and binary files.
    - SLIMS recognizes the default completion files and parses the binary files in order to produce run information and provides result files including cluster density phasing and %>-Q30.
    - Sequencer specific processing uses the instrument run folder, interop summary, and links from the First\_Base\_Report.htm to the instrument run.
  - **NovaSeq:**
    - SLIMS looks for the default finish file CopyComplete.txt, or the specified finish file. The default important files are the RTAComplete.txt file and any specified important files.
    - Sequencer specific processing uses the instrument run folder and interop summary.
  - **NextSeq:**
    - Has no default finish or important files.
    - Sequencer specific processing uses the interop summary.
  - **MiSeq:**
    - Like the other sequencers, MiSeq uses the run folder as the sample sheet location, but also Data/Intensities/BaseCall.

- MiSeq does not have a default finish file, so the sample sheet file finishes the run unless a different file is configured as the finish file. The default important files are the CompletedJobInfo.xml, or any specified important files. The folder watch service for MiSeq reacts when the CompletedJobInfo.xml file is created. This file can be deleted and recreated in order to trigger SLIMS results processing.
- Sequencer specific processing: MiSeq is the only sequencer that uses status "Failed" (statuses listed below). At the end of a run as soon as the sample sheet is dropped into the Sample Sheet folder, SLIMS searches for two FastQ files for each library. If any FastQ file is missing, the run status is changed to Failed. If it finds all the relevant files, the status changes to Done. Customization of the Done/Failed acceptance can be done by a SLIMS Engineer on request.

4. The status indicates a brief report of the MiSeq, HiSeq, NovaSeq, or NextSeq run.
  - *Not Started*: The Pick Up Results button is not clicked yet, so the monitoring system has not found the file yet. (Some sequencer types will be able to discern when new files are added without needing to click Pick Up Results, but then the button will be grayed out to indicate it is already processing.)
  - *In Progress*: The button has been clicked and the monitoring system is searching for the sample sheet file with the GUID. It will continue looking until it finds the file or times out.
  - *Done*: The sample sheet / finish file was found and the results files have been processed. The finish file and important files are attached to the protocol run step.
  - *Failed*: Relevant for MiSeq only. The files were not able to be processed because one or both of the FastQ files for each library were missing from the MiSeq run.
  - *Timeout*: The monitoring system searched for the sample sheet file with the GUID in it but could not find it and timed out.

**Figure 6.13. Pickup Results Diagram**

- The fourth protocol performs secondary analysis of the result files. It has no run display. The user selects the library to analyze and starts a run. This will transmit the necessary information to the analytical tool. There is no feedback displayed in this step, though it is possible to view the execution of the Galaxy analysis directly in the Galaxy software.
- The fifth protocol is used for tertiary analysis. The user can add the applicable variant as described in the Variant section of this manual.

## 6.5. Runs

The Runs Module provides a display of all protocol runs. It contains filters specific to protocol runs within a compact information screen. A complete run can be viewed by clicking the "Show Run" button which opens a view similar to the view of a run in the Workflow Module.

ⓘ Functionality *Runs* needs to be enabled in order to access the module.

The module is divided in three panels. The left panel contains all the available filters.

- Include runs in subprojects: Extends filtering for projects additionally to subproject level.
  - Workflow: Filter on which workflow these runs were executed.
  - Protocol: Filter on which protocol was used for the runs.
  - Protocol Type: Filter the protocol runs by Default (all runs except simplified runs) or simplified (all simplified runs).
  - Requestable: Filter on which requestable was used for the runs.

- Content: Filter the protocol runs by the content that was involved. Only the runs in which the content was linked as a content, linked as a reagent, created as a derivation of another content, or created as a mix of several other contents will be displayed.
- Reagent: Filter on which reagent was involved. Only the runs in which the content was linked as a reagent will be displayed.
- Instrument: Filter on which instrument was used for the runs.
- Project: Filter the protocol runs by which project the runs were executed in.
- Status: Filter on the status of the run (Not Started, In Progress, etc.).
- User: Filter on user.
- Created by: Filter on the user that started the run.
- Start and End Date: Filter on date and hours.

The middle panel contains the protocol list, showing the protocols that result from the filter.

The right panel contains a screen with compact information regarding the run.

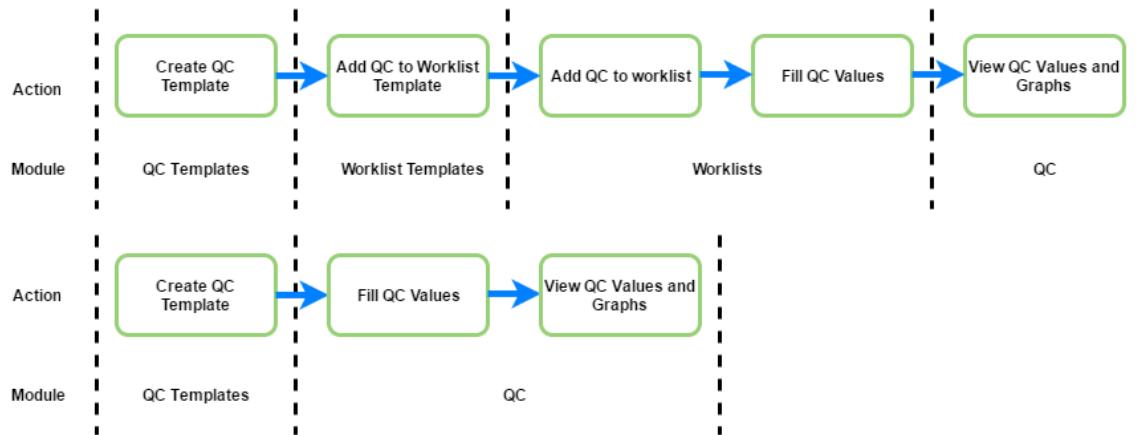
- Input: The input for the run.
- Output: What came out of the run.
- Results: Any results created during the run.
- Started and finished by/on: Who started and ended the run, and when it started and ended.
- A button to open the run. The run will be opened in a view much like the protocol run view from the Workflow Module, except everything is read-only. The protocol run view will also have a back button which takes the user back to the main view of the Run Archive.

# 7. Quality Control

Quality Control (QC) allows samples that are not stored in SLIMS to be treated in order to assess the quality of the results. This chapter covers the configuration and use of QC within SLIMS.

The following diagram illustrates all steps to go through for using QC in SLIMS:

**Figure 7.1. QC Workflow**



Usage of QC with worklists is described with the sections [Worklists](#) and [Worklist templates](#). QC templates and QC modules are detailed in the following sections.

## 7.1. QC Templates

The QC Template Module manages QCs that can be used with Orders and Protocols and is available under the setup menu.

QC Templates are of three types: Blanco, AQC, and Raw Data. The parameters and measures are dependent on the type. Each QC Template refers to a Protocol step test. The corresponding test should have datatype quantity.

The following Role Access rights can be configured for use in this module:

**Table 7.1. Required Permissions for QC Templates Module**

Access rights Category	Access rights Name	Purpose
Setup: Quality Control	QC Templates	This permission allows the role user to access the QC Templates module where various types of QC templates can be created for use with Orders and Protocols to aid in the detection of abnormal result values.

### 7.1.1. QC Templates of Type Blanco

A Blanco template type only has a lower and upper boundary and does not provide a chart visualization. A QC using a QC Template of type Blanco will only check that the value is between the boundaries, and provides a warning if either limit is exceeded. It is mostly used to check whether an instrument is running properly on an empty sample.

These parameters can be set for a Blanco QC Template:

- **Name:** Name of QC Template
- **Type:** Blanco
- **Description:** Description of what the QC Template is for.
- **QC Card Unique Identifier Mask:** Generates a root name and incremented number based on the mask QCCR##### for each QC Card that is generated from the template, or a mask customized by the administrator. See the Name mask (use # for numbers) definition for more details.
- **Variable Decimals on QC Card:** The amount of decimals that should be shown on the QC card.
- **Protocol step test:** Protocol step that has the test linked to it, combined with the test itself.
- **Lower limit:** Lower limit of the acceptable values (the minimum acceptable value).
- **Upper limit:** Upper limit of the acceptable values (the value that should not be exceeded).
- **Apply Precision to QC result:** If disabled, the precision of the Test datatype will be ignored.

### 7.1.2. QC Templates of Type AQC

A QC using a QC Template of type AQC will put the value in a graph defined in areas. These areas are set up using the mean and standard deviation of the QC. This is used most often to measure a QC value of a known concentration and plot it on a Levey-Jennings chart with Westguard rules evaluation.

The below parameters can be set for a QC Template of type AQC. QC measures of AQC Template type are treated by cards. Each card treats a certain amount of days and gives a graphical feedback of the values. The graph has an additional filter to instead sort by the order of results, and to display a chosen amount of results at a time.

- **Name:** Name of QC Template.
- **Description:** Description of what the QC Template is for.
- **QC Card Unique Identifier Mask:** Generates a root name and incremented number based on the mask QCCR##### for each QC Card that is generated from the template, or a mask customized by the administrator. See the Name mask (use # for numbers) definition for more details.
- **Protocol step test:** Protocol step that has the test linked to it, combined with the test itself.

- **Type:** AQC
- **Mean:** Expected mean of the measures. This value is first set up equal to the theoretical value. Once a card is finished, its values are used to compute the mean of a new card. This value is fixed for each card, except if it is changed manually.
- **Standard deviation:** Expected standard deviation. This value is first set up and then it will be computed for each new card.
- **Theoretical value:** Theoretical mean of the measures.
- **Maximal Bias Mean:** For QC Bias check. The check will fail if the absolute difference between the new calculated mean and the theoretical value is more than the value for this property.
- **Maximal Standard Deviation:** For QC Bias check. The check will fail if the new calculated standard deviation is more than the value for this property.
- **Variable Decimals on QC Card:** The amount of decimals that should be shown on the QC card.
- **Card span:** Number of cards that will be taken in consideration when computing the new mean and standard deviation.
- **Number of results:** Number of days by card. ① By default, if multiple results are entered on the same day, they will all be on the same date (same x coordinate). All x positions (days) have at least one observation, i.e. there might be multiple "real" days between two days on the card. For instance, if work is done on Monday, Tuesday, and Friday every week and the first Monday is May 1st, then the day of the card will be day1: Monday 1st, day2: Tuesday 2nd, day3: Friday 5th, day4: Monday 8th, day5: Tuesday 9th, and so on.

The 'Display Graph by Results' option allows the user to enter a range, and instead of displaying by date, the x axis displays the results in the set range in order. The user can then use the left and right arrow buttons to show the next range of results. This allows results that fall on the same day to be displayed on separately. The result dots are connected by a line when the filter is turned on.

The 'Display Graph by Results' is only available if the user has the following Role Access rights:

**Table 7.2. QC Result Display Permissions**

Access rights Category	Access rights Name	Purpose
Routine: Quality Control	Can Display QC Card by Results	When enabled, the role user will have the "Display Graph by Results" option in the QC Templates Module.

As an example, if three results are entered on Monday, and two are entered on Wednesday, and the user sets a range of 1 to 5, then Monday's results will display on x axis points 1, 2, and 3, and Wednesday's results will be on 4 and 5 all connected by lines to make the change in mean more apparent.

- **Start date**

- **S bound (%)**
- **Fixed standard deviation and mean:** If ticked, the values that are entered here will be used every time. If unticked, SLIMS should compute new mean and standard deviation for each new card.
- **Apply precision to QC result:** If disabled, the precision of the Test datatype will be ignored.

### 7.1.3. QC Templates of Type Raw Data

This type can be used to observe the trend of raw data results of a machine and will give warnings when QC result values exceed a given limit.

Before creating a raw data QC card, make sure tests and protocols (with a result step that uses the test) have been defined. Go to the QC template module and click on the "New" button. A pop-up will appear for the user to fill in the details:

- **Name:** Name of QC Template
- **Type:** Raw data
- **Description:** Description of what the QC Template is for.
- **QC Card Unique Identifier Mask:** Generates a root name and incremented number based on the mask QCCR##### for each QC Card that is generated from the template, or a mask customized by the administrator. See the Name mask (use # for numbers) definition for more details.
- **Protocol step test:** The related protocol step test for which raw data is being monitored. This setting also allows Raw Data QC values to be entered into a worklist configured with this QC template and protocol step test (This is also possible for AQC and Blanco QC templates).
- **Raw Data Test:** This test contains the datatype that will be used to save raw data results.
- **Variable Decimals on QC Card:** The amount of decimals that should be shown on the QC card.
- **Lower limit:** The lower limit of the raw data QC value. If the value is less, a warning message will be shown.
- **Upper limit:** The upper limit of the raw data QC value. If the value is more, a warning message will be shown.
- **Apply precision to QC result:** If disabled, the precision of the Test datatype will be ignored.
- **Active:** If ticked, the values that are entered here will be used every time. If unticked, SLIMS should compute new mean and standard deviation for each new card.

When creating a Raw data QC template, a QC card will be created automatically. The QC card has an infinite amount of measurement days, but for convenience, only the last 30 days are shown. With some navigator buttons it is possible to navigate to previous measurement days. The QC card can be closed manually (for instance after cleaning/calibration of the machine) and then a new card will be created automatically.

## 7.1.4. QC During the Night

In order to make QC work, some scripts have to run every night. This has to be set up by a SLIMS engineer. This describes what SLIMS is capable of overnight.

- SLIMS picks up the list of changed cards
- Picks up the list of results of the current day
- Evaluates QC again and sends an email about anything that may have gone wrong
- Checks if some changed cards are full in order to close them.
- Checks all the assumptions on closed cards and emails about any problems. A card that is defined as bad won't be used to compute the mean and standard deviation of the next card.
- Opens a new card for each closed card and computes mean and standard deviation for each of them (with exclusion of outliers).

## 7.2. QC

After defining QC templates, QC values can be created in two ways as shown in Figure 7.1 using either *Worklist templates* and *Worklists* modules or the QC module.

This module requires the following Role Access rights to work:

**Table 7.3. QC Required Permissions**

Access rights Category	Access rights Name	Purpose
Routine: Quality Control	Quality Control	This permission enables the QC module which allows role users to assess the quality of the results from sample testing. This permission also controls whether the role user will have the "View QC Card" action available in the context (right-click) menu in Worklists.

**Table 7.4. QC Optional Permissions**

Access rights Category	Access rights Name	Purpose
Routine: Quality Control	Close QC Card	When enabled, the role user will be able to close QC cards in the QC module.

The role user can display the QC card by results as well if their role has the "Can Display QC Card by Results" access right enabled: *QC Card by Results definition*.



Go to the  button available under Routines. The QC module contains three tabs that are explained in detail in the next sections.

① QC status are the same as Results status which are described in the *ELN chapter*.

## 7.2.1. QC by Card

Cards are only available for the QC types "AQC" and "Raw data."

The list of QC cards is available in the left window. QC records in the right window are filtered according to the card that was selected on the left. In addition, a graph is available for each QC card.

## 7.2.2. QC by Template

The list of QC templates is available in the left window. QC records in the right window are filtered according to the selected template on the left.

## 7.2.3. All QC

Shows all QC records in a list grid. Views can be created and tools available with the  button can be used on the grid.

## 7.2.4. Add QC Value

Click on the "New" button and a pop-up will appear where the value can be entered. The unit of the value will depend on the chosen test.

## 7.2.5. QC Value Actions

Right-clicking on a QC Value provides the following actions:

- **Open Detail:** Opens QC details in a pop-up window. The QC value and the comments field can be edited in the opened detail window.
- **Validate:** Validates a QC value and changes its status to "Validated."
- **Reject:** Rejects a QC value and changes its status to "Rejected."
- **Revert:** Reverts the QC record status back to "Pending." This is particularly useful when rules are defined and the QC value needs to be changed. The revert action works in two steps: first, the QC value reverts to status "Pending," and then a new value is inserted. This way all rules will be re-evaluated.
- **Cancel:** This status is available once QC has gone to status "Validated". This action will create an event and change the QC status to "Cancelled" after asking the user for a reason. Details of this event can be viewed afterward by selecting the "View Events" action.
- **Restore:** Changes the status of "Cancelled" QC back to "Validated."
- **View Events:** Events are created whenever a QC status is changed to "Cancelled." This action will show all events and their details.
- **View History:** Shows all changes made on a QC value or its status. See the *History* section for more details.

- **Edit Mode:** Allows editing (alternatively, double-click on the record for the same action).
- **Remove Permanently:** Delete the QC record permanently. The QC record will be lost forever when using this action. It is recommended to "Cancel" a result instead of permanently removing its record.

## 7.2.6. QC Rules

To set QC Rules, go to the *Rules* module in the setup module and create a rule of "QC" type.

A "QC" rule has two possible values in the 'QC Type' field.

- Warning: If the rule is violated, the result will be marked with an orange warning.
- Error: if the rule is violated, the result will be marked with a red warning.

QC rules are backwards compatible with existing rules that use 'Error' in older SLIMS versions.

## 7.2.7. QC Statistics Check

When a QC card of type AQC is closed, some QC card statistics and bias checks are performed in order to determine if the calculated mean and standard deviation of the successor QC card is compliant.

To perform bias checks, the user will need a SLIMSGATE Flow to mark the results and a previous QC card with at least 4 validated samples.

After closing the QC card, an attachment with calculations will be added.

# 8. Worklist

Worklists are used to simplify a common execution such as filling in results and completing protocol steps for a selection of content. In order to use the Worklist module, Worklist templates need to be defined first.

## 8.1. Worklist templates

Worklist templates handle the different templates of Worklists, and must be defined first in order to use the Worklists module. Names manage which steps will be included in a Worklist, Worklist layout, entry sorting, content filtering, and QC settings.

This module requires the following Role Access rights to work:

**Table 8.1. Worklist Template Module Permissions**

Access rights Category	Access rights Name	Purpose
Setup: Miscellaneous	Worklist Templates	This permission provides access to the Worklist Templates module which allows the role user to design worklist templates to fill results and complete protocol steps for a selection of content.

The Worklist templates module is separated into two panels. The left panel contains the list of templates and the right panel contains the details about the template that is selected in the left panel.

The details about a template (right panel of the module) is divided into five tabs: Worklist Template, Layout, Sort, Criteria, and QC. All these tabs will be described in the following subsections.

The definition of the Worklist Template Name is important as it will help lab users to execute work efficiently. For each Worklist Template a clear purpose should be established before its creation, either to report in SLIMS procedural work that can be done in batch mode, or to enter results in batch mode. In both cases, the steps to be executed should be part of a known and common routine that the lab user is familiar with.

### 8.1.1. Worklist Template - Name

The Worklist Template tab contains the main details about the Worklist Template. This is the only tab that needs to be defined in order to create a Worklist.

Here is the description of the different parameters of the Worklist Template:

- **Name:** Name of the Worklist Template
- **Name mask:** Default name of a newly created Worklist. It can either be a fixed name ('worklist name') or a date-related one (ddMMyyyy-'worklist') such that SLIMS will replace the date

format that is given (ddMMyyyy) with the date that the Worklist was created. Fixed string parts should be between quotes. Example: the 1st of April 2014 will be shown as: 01042014.

- **Description:** Description of the Worklist Template.
- **Worklist Unique Identifier Mask:** Generates a root name and incremented number based on the mask WRKL##### for each worklist that is generated from the template, or a mask customized by the administrator. See the Name mask (use # for numbers) definition for more details.
- **Only add results and protocol steps that did not appear on another worklist:** Whenever this option is checked, it prevents users from entering a result in multiple Worklists.
  - ① Repeating a result will still be possible.
- **Maximum number of rows:** Defines the maximum number of rows in the Worklist, i.e. maximum number of contents and QC rows that can be added to a Worklist.
- **add all QC configured for this template, also for non requested tests:** Defines whether a QC result should be added to the worklist if no normal result of the same test is available on the worklist. For instance, both the tests pH and Temperature and their respective QC templates are configured in a worklist template. If a content is added to such a worklist with only the pH test requested, and QC is added, this defines whether Temperature QC should also be added to the worklist.
- **Protocol step tests:** Defines whether results need to be filled up with this kind of Worklist. Step tests that are relevant should be added. Step Tests are steps that have the test linked to it combined with the test itself.
- **Protocol steps:** If procedural work needs to be reported as done, allows selection of the steps that are relevant. Steps of type Link content, Procedural, Link content and Add Results, and Result can be selected.
- **Informative protocol step tests:** Similar to *Protocol step tests* but read-only. For instance, result values of another 'related' test that could already be filled in, in another worklist.
- **Flows:** Flows (mostly export/import of data) that should be used to reduce direct interaction with SLIMS.

## 8.1.2. Layout

The layout tab allows the selection of which parameters should be displayed while executing the Worklist.

⚠ The Layout of all Worklists with this Worklist Template is defined here. It is not possible to change the layout directly within the *Worklists* module.

### 8.1.2.1. Worklist Selection

The first panel manages the layout of the Worklist module to allow selecting the contents / results / steps that should be part of the Worklist.

- **Content columns:** Columns that will be displayed relating to the content record.

- **Fields for results:** This part relates to Results.
  - **Result columns:** Select the columns that will be displayed concerning the result record.
  - **Show previous results:** Choose whether the previous evaluation of the results should be shown (only when a result is repeated).
- **Fields for protocol steps:** This part relates to protocol steps (of type procedural).
  - **Protocol run step columns:** Select the columns that will be displayed concerning the protocol run step record.
  - **Protocol run step content columns:** Select the columns that will be displayed concerning the protocol run step content record.

### 8.1.2.2. Worklist

The second part configures the layout of the actual Worklist when it is confirmed and also the bottom part during the work selection step.

The same parameters as in the Worklist Selection part can be defined.

### 8.1.2.3. Print

The third part allows the user to define which columns should be printed.

If the box is unchecked, the same columns and selections that are already defined will be printed. If the box is checked, then the same parameters are available to set up as in the Worklist Selection part. Furthermore, the following parameters can be set up.

- **Print extra columns:** Creates new columns that will be printed. The name corresponds to the header of the column. (It doesn't need to be related to a field that is present in SLIMS.)

### 8.1.3. Sort

The Sort tab allows the entries to be sorted according to a selected criteria that depends on the type of Worklist.

### 8.1.4. Criteria

The Criteria tab allows the contents to be filtered that will be used within Worklists using the selected Worklist Template. More details on the filtering options can be found in the *Advanced Filter Section*.

### 8.1.5. QC

This is available only for a QC template referring to exactly the same protocol step test as the one selected for the Worklist Template.

- **Name:** Name of the QC for this Worklist.

- **QC templates:** Which QC template it is referring to.
- **Alternating group:** Row templates can be added to a group so their rows will not be added at the same time to a worklist. Only the least recently used row will be kept. This can be used to create an alternating pattern.
- **# of QC to start with:** Number of QC samples to add before the list of contents.
- **Block size:** Number of contents between two QC samples.
- **Block offset:** Number of places that a block should be shifted to the bottom. The default value is 0.
- **# of QC to end with:** Number of QC samples to add at the end of the list of contents.
- **Seq no:** At which sequence position this QC should be.

#### Examples of values for QC details

- 10 samples, # of QC to start with : 2, Block size : 3, Block offset : 0, # of QC to end with : 4  
QC1, QC2, sample1, sample2, sample3, QC3, sample4, sample5, sample6, QC4, sample7, sample8, sample9, QC5, sample10, QC6, QC7, QC8, QC9
- 10 samples, # of QC to start with : 3, Block size : 4, Block offset : 2, # of QC to end with : 2  
QC1, QC2, QC3, sample1, sample2, QC4, sample3, sample4, sample5, sample6, QC5, sample7, sample8, sample9, sample10, QC6, QC7

## 8.2. Worklists

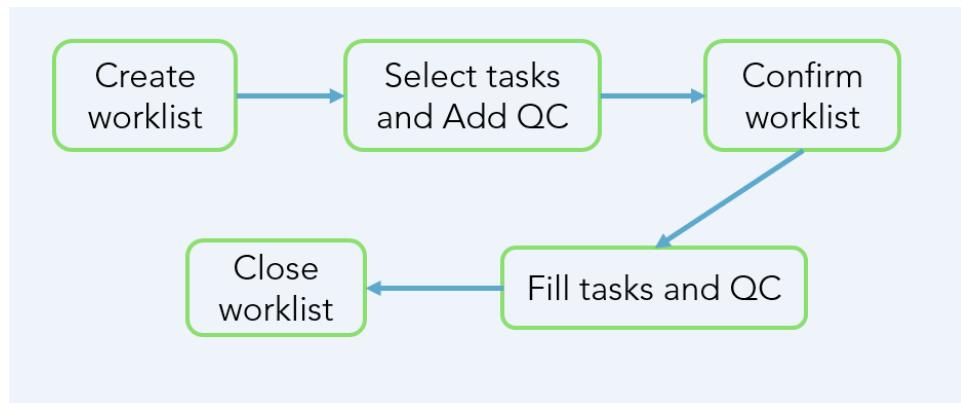
The Worklist module contains a principal tab that has a list of all existing Worklists in a grid. This principal tab is called 'All Worklists' and custom views can be added to the grid. In addition, all *grid tools* are available through the  button.

This module requires the following Role Access rights to work:

**Table 8.2. Worklist Module Permissions**

Access rights Category	Access rights Name	Purpose
Routine: Worklists	Worklists	This permission provides access to the Worklists module which allows the role user to simplify commonly used executions, like filling results and protocol steps for a selection of content.

The following diagram illustrates all the steps to pass through in order to fulfill tasks using Worklists.

**Figure 8.1. Workflow of a Worklist**

A Worklist task could be: 1- Entering a result value 2- Completing a protocol step.

1. **Create Worklist:** New Worklists can be added in the 'All Worklists' tab using the  New button. After adding a new Worklist, a new tab is automatically created.

The Worklist will contain content records linked to an ELN step with status 'In progress' and fill one of the next two conditions:

- Have unfilled results of tests specified in 'Protocol step tests' field in the *Worklist Template*.
- Be linked to one of the protocol steps specified in the *Worklist Template*.

2. **Select tasks and add QC:** Two parts will be available in the Worklist tab. The top part named 'Worklist Selection' containing the list of content records, and the lower part where the actual Worklist is shown. Tasks and QC rows can be added to the Worklist as explained below:

- **Add Tasks:** Results or protocol steps can be selected from the top part 'Worklist Selection' and transferred to the actual Worklist using the button  'Transfer to Worklist'. Note that not only entire rows but also specific content results or protocol steps can be selected and transferred to the Worklist. Filtering and sorting on content and result fields is possible in the Worklist Selection (upper) part.
- **Add QC:** The button  provides a menu with the available QC actions described below:
  - *Add predefined QC:* Adds all QC defined in the Worklist Template. Predefined QC can be added at the beginning, end, or between a defined number of content rows. See *QC in Worklist Templates* for more details on the configuration of predefined QC.
  - *Add new QC rows:* A button will be available for each QC added to the Worklist Template. (See *QC in Worklist Templates*). It allows adding QC at a certain position.
  - *Randomize AQC:* QC of type AQC will have their position changed in the Worklist to be random.
  - *Remove all QC rows:* Remove QC rows from the actual Worklist.

In the Worklist (lower part), the order of rows can be edited by using the available arrows on the right or by selecting multiple rows and moving them simultaneously using the  button available in the grid header.

Tasks can be removed from the Worklist using the  button available in the grid header. Removed tasks will be put back in the Worklist Selection window.



**Add empty row** button available in the grid header is used to add empty rows in the Worklist form. This is used to make the Worklist form easier to read or for printing purposes, but it has no impact in SLIMS.

① The context menu available by right-clicking is cell-based. Thus, a specific result or a protocol step can be selected and right-clicked to access the list of actions. Result actions are described in the *Results Section* under ELN.

△ Available columns in the Worklist are defined in the *Worklist Template*, so they cannot be edited in this module.

3. **Confirm Worklist:** The Worklist can be confirmed using the  button available at the bottom of the page.

4. **Fill tasks and QC:** Once the Worklist is confirmed, tasks are filled either by entering a result value or by completing a protocol step as described below:

- *Entering results:* By clicking once on a record row, the row can be edited so that result values can be entered. The result status is changed automatically to "Verified" when the result value passes all predefined rules. See the *Rules Module* for more details on how to define rules.
- *Completing protocol steps:* A protocol step can be set to done for each content record independently by clicking on the  button. The protocol step will only be set to status 'Finished' in ELN when all content records linked to it have their protocol step finished in Worklist.

For example:

- SampleB is set as finished, but SampleA and SampleC are still in progress: the protocol run step is in progress.
- Both SampleD and SampleE are set as finished: the protocol run step is finished.

**Figure 8.2. Protocol step status in Worklists**


The screenshot shows two windows side-by-side. The left window is titled 'Worklists' and contains a table with three rows. The columns are 'Content', 'Id', 'Name', 'Status', 'Created on', and 'protocol step'. The rows show tasks: 'sampleB' (protocol step, In progress, 02/02/2016 12:01), 'sampleC' (protocol step, In progress, 02/02/2016 12:01), and 'sampleA' (protocol step, In progress, 02/02/2016 12:01). The right window is titled 'ELN' and shows a table with two rows. The columns are 'Content', 'Name', 'Status', 'Created on', and 'protocol step'. The rows show protocol steps: 'protocol step (Workflow)' (In progress, 02/02/2016 12:01) and 'protocol step (Workflow)' (Done, 02/02/2016 12:07). The status in the ELN table matches the 'Status' column in the Worklist table.

Content	Id	Name	Status	Created on	protocol step
sampleB	1	protocol step	In progress	02/02/2016 12:01	✓
sampleC	2	protocol step	In progress	02/02/2016 12:01	✗
sampleA	3	protocol step	In progress	02/02/2016 12:01	✗

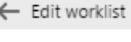
Content	Name	Status	Created on	protocol step
sampleE	1	protocol step	Done	02/02/2016 12:07
sampleD	2	protocol step	Done	02/02/2016 12:07

Content status in Worklist and corresponding protocol step status in ELN.

Viceversa, if a protocol run step is set to status 'Finished' in ELN, all Worklist tasks relative to that protocol run step will be set to 'Finished'.

- *Entering QC values:* Similar to result entry. QC values are also evaluated and QC status is changed automatically to "Verified" when the QC value passes all QC rules.

Actions available through a context menu for Results and QC values are the same actions available for results in ELN. Description of all actions is available under the *ELN Module*. See the *Rules Module* for more details on how to define rules on QC.

①  button available on the bottom can be used in order to re-edit the confirmed Worklist.

5. **Close Worklist:** After filling all tasks, a Worklist can be closed using the  button available at the bottom of the page or by clicking on the  next to the Worklist tab name.

A closed Worklist can be made available by selecting 'Open Worklist' in the context menu (right-click) of the Worklist record in the 'All Worklists' window.

- ① A Worklist can be closed and reopened at any step.

# 9. Reporting and Grids

A report in SLIMS is an automated procedure initiated by the user that creates a report file according to a predefined template. Users can define report templates in the UI, upload their own Jasper templates, and execute created report template to generate PDF, HTML, RTF, DOCX, and CSV documents.

Users can define custom grids in the UI to export information from different tables to one file.

## 9.1. Report Templates

Report Templates are split into two types: Core and SLIMS GATE. Core Report Templates allow users to configure a report completely in the UI, especially the structure of the XML files which will be given as an input to the Jasper Template. The Jasper Template is part of the definition of a Core Report Template and can be uploaded as an attachment. The Core Report Templates use the traditional versioning mechanism for SLIMS modules.

SLIMS GATE Report Templates refer to an alternative type of templates and are executed in SLIMS GATE. This kind of template is used when a special structure of the input XML file is required, and when the SLIMS GATE bean needs to be developed to generate it.

The following Role Access rights can be configured for use in this module:

**Table 9.1. Permissions for Report Templates Module**

Access rights Category	Access rights Name	Purpose
Setup: Report	Report Templates	This permission allows the role user to access the Report Templates module where a report can be configured completely in the UI to produce Core and SLIMS GATE reports.
Setup: Report	Can Execute Drafts of Report Templates (if enabled on the template)	This permission allows the role user to execute the draft version of a report template when that template has the option "Draft is executable" enabled. This allows the user to execute report templates easily without having to add a new version.

### 9.1.1. Core Report Templates

#### 9.1.1.1. The Report Template Form

The Selection Type determines how users can generate a report of this template.

- *Right-click*: Users can right-click a record of the target table to generate a report for that record.

- When a right-click report is generated, the report is linked as an attachment to the original record that the user right-clicked on.
- An attachment is generated from right-click reports, so the "Attachment type of the created attachment" becomes available for you to choose the type of attachment the report will be.
- *Multi-select*: Users can select multiple records in a grid that the target is in and generate a report for all these records at once.
- *Criteria*: Reports are generated from within the Report Module. The exported records are solely determined by the meta tree. These reports are also not bound to one table.

*Report Unique Identifier Mask*: Generates a root name and incremented number based on the mask RPRT##### for each report that is generated from the template, or a mask customized by the administrator. See the Name mask (use # for numbers) definition for more details.

The export format determines the format that the report will be generated in. These are the possible format options:

- PDF
- HTML
- RTF
- DOCX
- CSV

Users can choose the export format when generating a report: The export format can be changed when the user generates a report if the flag is checked.

*Draft is executable*: Users can also execute the draft version of a report template. This option exists so report templates can easily be executed without having to add a new version all the time. In order to do this, the user's role also needs the extra functionality "Can execute drafts of report templates" in the Roles: Functionality Access Rights tab.

### 9.1.1.2. Meta Tree

The meta tree tab allows users to define the meta tree for the XML Template that will be used to generate the report with Jasper. See Meta Tree Section

### 9.1.1.3. Attachments

In the attachment tab users can upload their Jasper templates (.jrxml files). This can be a single file or multiple files in case of sub-reports. Users can also upload images they use in their reports here. One of these files needs to be marked as the "Main Report." This marks the Jasper Template that will be generated as the main report. The others will be sub-reports that are included in the main report.

Sub-reports can be included via the `REPORT_DIR` parameter in Jasper.

```
<subreportExpression><![CDATA[JasperCompileManager.compileReport(new  
FileInputStream(${P{REPORT_DIR}+"SubReport.jrxml"})]]></subreportExpression>
```

Likewise images can also be included like this:

```
<imageExpression><![CDATA[$P{REPORT_DIR} + "Image.png"]]></imageExpression>
```

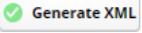
### 9.1.1.4. Versioning

Versioning behaves like in other modules except that the attachments are also copied in this module. It is possible to create a custom publication status workflow to govern versioning by creating a status custom field on the Report Template table. More instructions are available in the *Publication Status Workflows* section.

### 9.1.1.5. Generating XML Datasources

Data extracted from SLIMS database are converted to XML format and used as an input data source for the report template. Therefore, it is useful to have examples of XML datasources when designing the report template.

The XML datasources generated for a report can be obtained with the following steps:

- The "Draft is executable" option needs to be set to true in the report template.
- Generate a report by selecting the "Report (Draft)" option  Draft >. (Role functionality "Can execute drafts of report templates (if enabled on the template)" needs to be true).
- Select "Generate XML" .

① XML datasources can be generated before Jasper Templates are defined in the Report Template.

## 9.1.2. SLIMS GATE Report Templates

A SLIMS GATE Report Template definition has less options than a Core Template.

In particular the XML datasource will be generated in a customized way. To do that, it is necessary to:

- Write the `Filler` bean generating the XML datasource
- Declare the name of the filler bean in the Report Definition

### 9.1.2.1. The SLIMS GATE Report Template Form

A Report Template has the following parameters:

- Name: The name of the template. The name also defines the format to which the template is exported. By default a template is generated in PDF format. To obtain reports in Word or RTF format, place the string `docx` or `rtf` respectively in the report template name.

For example:

- A Report Template named "*Title of the report template*" generates a report in the PDF format.
- A Report Template named "*Title of the report template (docx)*" generates a report in Word format.
- A Report Template named "*Title of the report template (rtf)*" generates a report in RTF format.
- Table: Specifies the target table of the template, which can be one of the following: Order, Project, Experiment, Protocol Run, Protocol Run Step.
- File Name: The name of the Jasper Template: (for example `report.jrxml`).
- Filler Bean: The name of the filler bean used to generate the XML datasource.

### 9.1.2.2. Examples of Report Templates

Some examples of report templates for the ELN elements are available. To access them a Report Template needs to be defined with the following parameters:

- Name: Any name for the report, but as explained previously, the format can be specified inserting the string (docx) or (rtf) at the end.
- Table: Project, Experiment, Protocol Run, or Protocol Run Step.
- File Name: The file name is `examples/project.jrxml`, `examples/experiment.jrxml`, `examples/protocolrun.jrxml`, or `examples/experimentrunstep.jrxml`.

### 9.1.2.3. Where to place Jasper Templates for SLIMS GATE Reports

Each template (e.g. `report.jrxml`) needs to be put in a folder with the same name as the template without the extension (e.g. `report/`). The folder needs to be placed in the `slimsgate/conf/META-INF/jasper/` folder on the server. So, the complete path of a report (from the `slimsgate` folder) is `slimsgate/conf/META-INF/jasper/report/report.jrxml`. The `jasper/` can also contain subfolders, for example: `jasper/my/subfolder/report/report.jrxml`. The path must be reported in the parameter `File name` of the Report Template.

**Table 9.2. SLIMS GATE Report Template**

Path from META-INF	Value of parameter File name
<code>META-INF/jasper/report/report.jrxml</code>	<code>report.jrxml</code>
<code>META-INF/jasper/my/reports/report1/report1.jrxml</code>	<code>my/reports/report1.jrxml</code>

In other words: there is always one extra folder on the server (with the same name as the main report) which is NOT specified in the File name.

① NOTE: The folder containing the report (in the example, the folder `slimsgate/conf/META-INF/jasper/report/`) is contained in the parameter `REPORT_DIR`. This is NOT a default parameter of JasperReport (Jasper does not have such a parameter). However, it is a useful one, primarily for report expressions such as the path of sub-reports or images.

### 9.1.2.4. Examples of XML Datasources

⚠ The following examples refer to XML datasources generated by SLIMS GATE. When using Core Report Templates, the XML datasource is defined in the report template itself and can be generated as explained in Section 9.1.1.5.

Report Templates are filled with an XML datasource. The structure of the XML datasource and the XPath (used by JasperReport to fetch the datasource) depend on the Target, which can be an ELN element (Section 9.1.2.4.1) or an Order (Section 9.1.2.4.2).

Two general rules:

- One field is represented by one node holding a text node, for example:

```
<name_of_field>Value of Field</name_of_field>
```

We will call this kind of node a *Field Node*.

- One record is represented by one node holding other nodes, typically Field Nodes, but also nodes representing related records. The name of the record node depends on the kind of record and the target (see specific sections).

#### 9.1.2.4.1. XML Datasource of ELN Records

The XPath is /export/entityName where `entityName` can be one of the following:

- Project
- Experiment
- ExperimentRun
- ExperimentRunStep

One record is represented by one node with a name equal to the kind of record. For example, a "Project" is represented by a `<Project>` node. Each record node holds nodes corresponding to the fields of the record, the attachments, the notes, and the related records.

Example:

```
<export>
<Project>
  <prjc_field>text of field</prjc_field>
  <Note>
    <note_contents>text of note</note_contents>
    <note_field>text of field</note_field>
  </Note>
  <Attachment>
    <attm_field>text</attm_field>
  </Attachment>
<Experiment>
  <xprm_field>...</xprm_field>
  <Note>...</Note>
  <Attachment>...</Attachment>
<ExperimentRun>
  fields, Notes and Attachments
```

```
<ExperimentRunStep>
  fields, Notes and Attachments
  <Content>
    fields and attachment of a content record linked to the experiment run step
    <Result>
      fields and attachments of a result of that content
    </Result>
  </Content>
</ExperimentRunStep>
</ExperimentRun>
</Experiment>
<Order>
  fields and attachment of this order
</Order>
</Project>
</export>
```

Attachments and notes are <Attachment> and <Note> nodes holding the related field nodes. The records related to another record are those records which in the ELN appear as contained in the record.

#### 9.1.2.4.2. XML Datasource of Orders

The XPath is /export/order

One order is represented by an <order> node holding

- Nodes describing the fields of the order
- A <contents> node holding <content> nodes, one per each content linked to the order. Each <content> node holds
  - nodes describing the fields of the content
  - A <requests> node holding a <request> node per requestable
  - A <localResults> node holding a <result> node per result obtained in the context of the order
  - A <results> node holding a <result> node per result linked to the content (but obtained somewhere else)

Example:

```
<export>
<order>
  <order_field>value of the field</order_field>
  <contents>
    <content>
      <content_field>value of the field</content_field>
      <requests>
        <request>
          <requestable_field>value of requestable field</requestable_field>
        </request>
      <localResults>
        <result>
          <result_field>value of result field</result_field>
        </result>
      </localResults>
    <results>
```

```
<result>
  <result_field>value of result field</result_field>
</result>
</results>
</requests>
</content>
</contents>
</order>
</export>
```

### 9.1.3. Export (ZIP)

Report templates of type Export (ZIP) are used to generate a ZIP document containing data relative to an entry in the ELN.

A default Export (ZIP) report template is available in SLIMS and it can be used to export all data relative to an entry in ELN. To create a new report template of type Export (ZIP), the user has to provide the Template Name and a jrxml file that will be used to generate a PDF report in the ZIP document.

Attachments (files) can be added in the Attachments tab as described in detail with Core Report Templates.

## 9.2. Reports

### 9.2.1. The Report Module

In the Report Module users can see the report templates and the reports that have been generated for each report template. Note that only reports generated from non-draft (versioned) report templates will be saved. Users can see the report template version that was used to create a generated report in the Report Template Version column. The report template version column can be filtered, sorted, highlighted, and grouped using Views. This is also the place where users can generate criteria reports.

### 9.2.2. Creation of the Template File

The template file is written in the Jasper Report format (.jrxml). The JasperSoft-Studio Software [<http://community.jaspersoft.com/project/jaspersoft-studio>] provides reliable assistance to define a Jasper Report Template.

Rich text fields can be rendered using the `html` component in a Jasper Template file. This component is included in SLIMS, but it is not part of the standard JasperReports package and may not be well supported.

### 9.2.3. Core Reports

When generating a report, either by right-clicking, multi-selecting, or starting it from the report module, the generated file will immediately start downloading. The only choice provided is the export format (if enabled in the report template). When a draft of a report template is executable, the XML that is used to fill the report can be generated. This can come in handy to develop the report in Jasper.

Reports generated by right-clicking are linked as an attachment to the original record the user right-clicked on. These attachments will be of the attachment type that is set in the Report Template of the report.

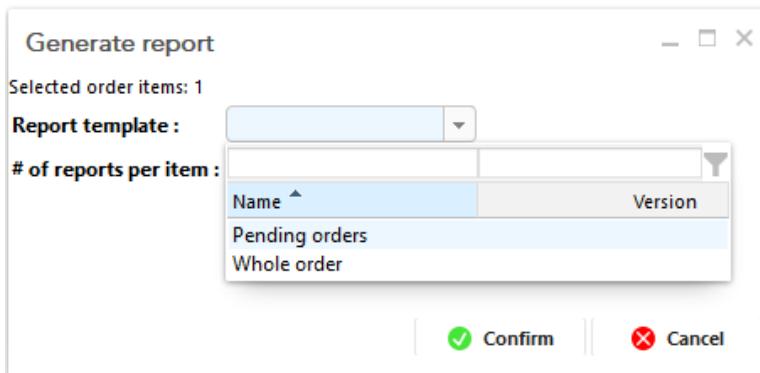
## 9.2.4. SLIMS GATE Reports

Reports of orders or ELN records (projects, experiments etc.) can be generated by selecting the action "Generate report." A pop-up window (Figure 9.1) asks the user which report template to use and the number of reports to generate.

The report template is the canvas of the report. During the generation of the report, the chosen template is filled with the record data, and an output file (in PDF format by default) is generated. The output file name contains the name of the report template.

A report template is specific to the kind of record (order, experiment run step and so on) or "target." For example, the report of an experiment run can be generated only using a report template with target "Experiment Run." Note that only templates of the right target are shown in the pop-up window (Figure 9.1).

**Figure 9.1. Generate Report**



① NA is exported in report XML as <na>

### 9.2.4.1. Order Reports

The report of an Order can be generated by right-clicking the order and selecting Generate Report (Figure 9.1).

Multiple Orders can be reported in a single file by selecting them and generating the report through the selection menu. A unique file will be generated containing the report of all the selected orders.

### 9.2.4.2. ELN Reports

The report of a Project, Experiment, Protocol Run, or Protocol Run Step can be generated by right-clicking the element and selecting Generate Report (Figure 9.1).

One report template per type of record (Project, Experiment, Protocol Run, or Protocol Run Step) is present in SLIMS. These templates serve as examples of possible configurations that a lab can base their build on to satisfy their individual needs.

To print the sequence of SOPs of a protocol, generate the Protocol Run Report called "Print SOPs."

⚠ Report are generated (compiled) on the server side. Therefore, fonts used in rich text fields (as notes) must be installed on the server to be used in the generated PDF report. Otherwise, missing fonts are simply ignored by the report compiler.

## 9.2.5. Export (ZIP)

A project, experiment, or any other record in the Electronic Lab Notebook can be exported in a ZIP folder. Right-click on the record to be exported and choose the action Export (ZIP). A pop-up window opens and the user chooses the Report template to use.

The ZIP folder is both:

- Attached to the record
- Sent to the browser so it can be directly download

If the ZIP file is too big, the action sending the file to the browser could fail. However, the file can be accessed from the record attachments after refreshing SLIMS regardless of if the browser was able to handle the file size or not.

The default Export (ZIP) template provided by Genohm creates a folder that contains:

- A PDF report with all data relative to the reported ELN element.
- An XML version of the report.
- A log file with info and possible (non fatal) errors.
- An attachment folder with all attachments organized in a tree structure which resembles the ELN tree structure.
- An HTML folder containing all rich text with no particular structure.

## 9.3. Grid templates

A new grid can be defined in the Grid template Module by clicking the "New" button and entering the Name, Type of grid template, and Main Table that should be related to the new grid.

A Grid Template is defined using two tabs.

- Grid template: Reports general information about the Grid Template, especially the Name, whether the template is for standard grids or for workflow queue grids, and whether the grid template is active or not.
- Meta: Defines the meta tree and the columns that will be present in the grid.

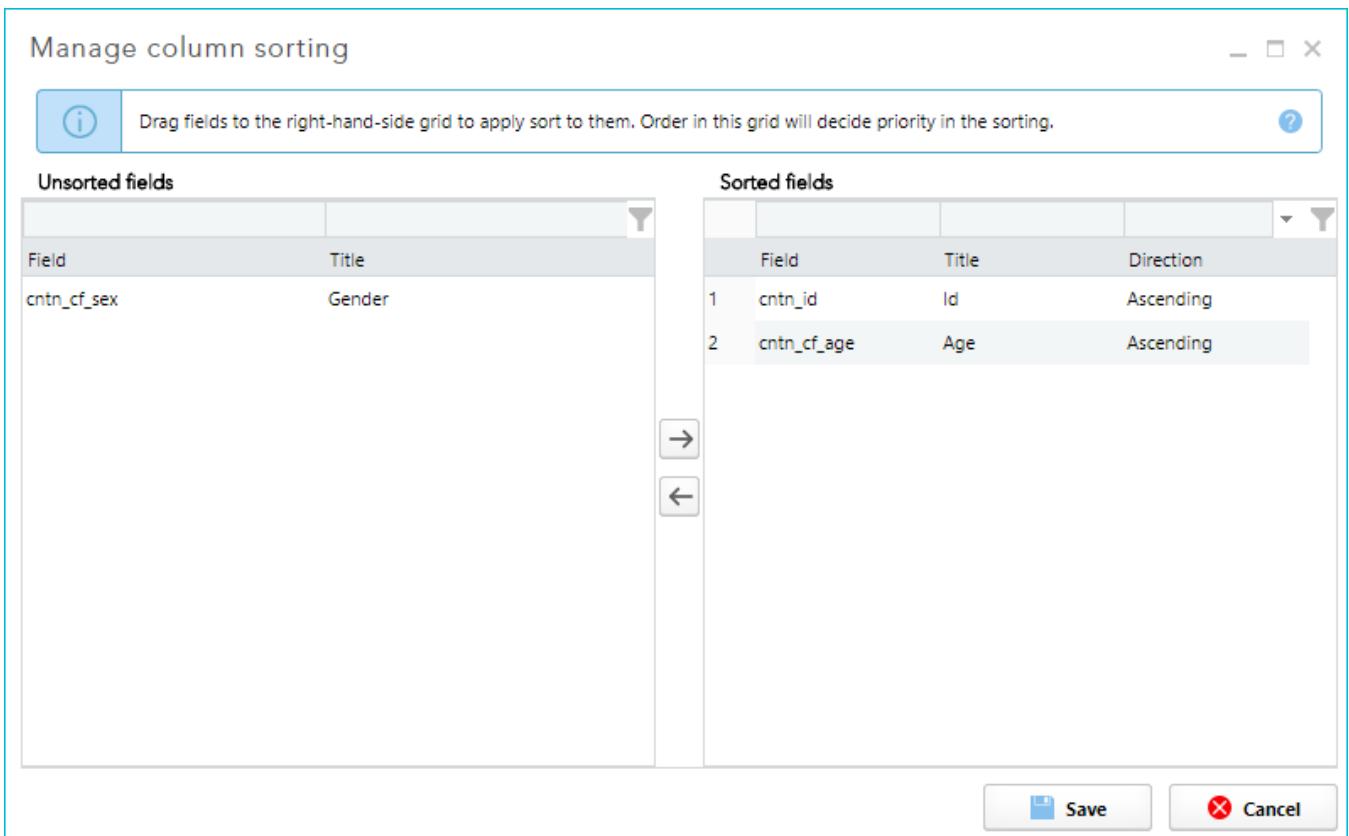
In the Meta tab, the meta tree can be defined in the same way as it can be for *Report Templates*.

In the same tab, the columns shown in the grid also need to be defined. A user can add new columns for the custom grid by clicking the "New Column" button. A window will be displayed in which the user must first select a table from the created meta tree. Then the user can

choose a column from the table to show in the custom grid and give the column a different title and a position in the grid.

Pre-sorting of the columns in the grid can be configured in the Columns tab. The user can click on the gear icon to access the Execution Menu, and then on the 'Sort On' icon  to access the Manage Column Sorting pop-up. The left grid contains the columns that were added to the grid template on which no sorts have been configured yet. They can be dragged to the right grid or selected and moved with the arrow icons. The number appearing beside the columns indicates the priority of the sort. In the image below, content in the grid will be sorted primarily by ID ascending, and secondarily by Age ascending. The sort 'Direction' can be changed by double-clicking on the direction field of the column and selecting either 'Ascending' or 'Descending,' but the Direction column can only be set to one or the other.

**Figure 9.2. Sort Columns in a Grid Template**



① The right-click menu for a Grid Template contains the Copy action. All of the Grid Template definition is copied into a new template, including the meta-tree. The Grid Template Copy will have the same name with the suffix "- (Copy)".

### 9.3.1. Stability Study Pulled Contents Grid Templates

This grid template type is only used for stability study functionality. A "Default Stability Study Pulled Contents" grid template is present with a pre-built structure. The default grid template can be modified to add new metatree nodes and adjust or add columns as you can any oth-

er custom grid template. New Stability Study Pulled Content grid templates can be created as well, which come with the predefined original structure seen in the default.

These columns are prebuilt by default: Barcode (Content), Status, Type (Content), Pulling Order, Barcode (Batch), Stability Study, Storage Condition, and Timepoint.

### 9.3.2. Stability Study Result Grid Templates

This grid template type is only used for stability study functionality. A "Default Stability Study Result" grid template is present with a pre-built structure. The default grid template can be modified to add new metatree nodes and adjust or add columns as you can any other custom grid template. New Stability Study Results grid templates can be created as well, which come with the predefined original structure seen in the default.

These columns are prebuilt by default: Status, Operator, Expected Value, Minimum, Maximum, Value, Created On, Storage Condition, Study Name, Study Version, Sample Barcode, Test, Stability Start Date, Timepoint, Pulling Order, and Measurement Order.

### 9.3.3. Queue Grid Templates

If your SLIMS instance uses Workflows, Grid Templates can be designed here and used for the queue grid of a workflow. This allows many columns to be added to a template once, and then re-used across many protocol queue grids.

When creating a new grid template, the type must be "Queue" to designate it as a grid template that can be used in Workflows.

There are two Queue grid templates present by default which cannot be deleted, but which can be modified:

- Workflow Queue that has a basic column presentation for standard workflows.  
It contains: Content Type Icon, Content Type, Content ID, and Content Barcode.
- Analytical Workflow Queue which has a useful column presentation for analytical workflows.  
It contains: Content Type Icon, Content Type, Content ID, Content Barcode, Test Label, Product, Analytical Method, Order Barcode, Order Type, and Modified On.

Only the QueueElementAggregator table is available for Queue grid templates. To get a start on building your own queue grid template, it is possible to copy one of the default templates, re-name it, and modify the columns as desired.

There are a few special case filters available for Grid Templates in a protocol run step that are useful. There are examples in the *Specific Use Cases* section.

### 9.3.4. Multiple Facets in a Grid

Multiple facets can be placed into a grid if there is a grouping and a dependency to a different grid that can be used to add other facets. It isn't possible to have multiple facets if there is only one grid (root), however; a dependency is required.

For example, a grid can be created that shows the number of orders created, grouped by month, and has a second facet to show the order type.

**Figure 9.3. Create Multiple Facets**

Number of orders by month - 2 Facets - Riad

Graph

Active

Column

Name Join type There exists at most one entry ... Criteria Aggregator Aggregator file...

OrderType (OrderType) Left   Count

Orders: From Order (via ordr\_fk\_orderType) Left   Count

Total rows: 2

New column

Meta tree Field Title Seq no Graph datatype

Orders value Value 0 Value

Orders ordr\_createdOn Created on 1 Facet

OrderType rdtp\_name Name 3 Facet

Total rows: 3

## 9.4. Grid

The grid module is found in the routine menu of SLIMS. This module contains all custom grids and can display them for the user. Columns of grids can be sorted and filtered, but only on the columns of the root of the meta tree. The contents of a grid can be exported to Excel or CSV format.

This module requires the following Role Access rights to work:

**Table 9.3. Grid Module Permissions**

Access rights Category	Access rights Name	Purpose
Routine: Grids	Grids	This permission enables the Grids module so the role user can view the created custom

Access rights Category	Access rights Name	Purpose
		grids and the data values that will be displayed in the grids.

## 9.5. Meta Tree

### 9.5.1. Root

The meta tree of a report template defines a template for the XML that will be sent to Jasper to generate the report. For right-click and multi-select reports, a root meta tree record is automatically created on the target table. For right-click and multi-select reports, the root cannot be deleted and no further root records can be added. For criteria report templates, no root meta tree record is created automatically and users can add multiple root records.

The meta tree of a grid template defines the data sent to the grid in order to fill one line. The root meta tree record is created automatically when the grid template is created.

### 9.5.2. Relations

Users can add extra table relations recursively to narrow down the data that gets displayed in the resulting grid. The relations can go in two ways: incoming (From) or outgoing (To).

For example assume a meta tree record on the Content Table exists. Users can add an outgoing relation to Content Type via the cntn\_fk\_contentType field. This means that in the generated XML in the <Content> element there will be a sub-element <ContentType>. In the other direction, adding an incoming relation from a Content Type meta tree record causes all the content of that content type to also be exported.

A special case of incoming dependencies is the attachment dependency, which will also export all the attachments.

Note that Grids are built so that one line per root record will be shown. So in the case of incoming relations when multiple results exist, it is not possible to predict which one will be shown in the corresponding column. It is recommended to use filters in order to limit the possible results so the finished grid is comprehensible.

### 9.5.3. Criteria

Users can also add filters on every level in the meta tree. This means that only the records that match the defined criteria will be exported.

This has special behavior when adding criteria on the root records for single and multi-select report templates. When some criteria are specified on the root of the report template and a record doesn't match the criteria, users won't be able to generate the report on that record. This means the option to generate the report doesn't show up in the UI for records that don't match. For example, a report template could be restricted to only work on content of type DNA. It will not be available when selecting contents of type Patient.

### 9.5.4. Having (Aggregators)

Aggregators can be added to leaf levels (relations) of the tree in the Having column. The available Having aggregators are Min and Max, *for which a Having Field needs to be specified*. This

implies that only the record with the minimum/maximum value for that field will be exported to the XML. Min and max nodes can have children.

① Dependencies to be used as an additional filter can be created on tables that have a Having Aggregator present, but the dependencies of the aggregated row will have an "Inner join."

## 9.5.5. Joins

On leaf levels of the tree, users can specify a join type (Left or Inner). The join type determines a different behavior when records corresponding to the parent of the leaf have a depending record or not:

- *Left*: Records corresponding to the parent of the leaf are always included in the report.
- *Inner*: Records corresponding to the parent of the leaf are included in the report only if records corresponding to the leaf exist.

For example, assume a meta tree record on the Content table (parent) with a dependency on the Result table via the rslt\_fk\_content field (leaf) exist. Depending on the join type defined for the leaf, the report will behave as follows:

- *Left*: All contents are reported even if they do not have results.
- *Inner*: Only contents that have at least one result are reported.

## 9.5.6. One Entry Per Node

The option 'There exists at most one entry for this node for each corresponding root node' can be enabled on each relation. For outgoing relations, this option is enabled by default. For incoming relations, this option is disabled by default and can be enabled if the end user can acknowledge that there is at most one entry for each corresponding root node.

This option is relevant for Grids only. If it is enabled, it will be possible to filter on columns of the corresponding sub-meta tree. For incoming relations, enabling the option and not being sure about the uniqueness of content to the root nodes could cause unexpected results in the grid. For instance if you have a root content meta tree with an incoming relation to a result meta tree and there is a content A and two results R1 and R2, the grid will show one row containing information of content A and R1. Filtering on result columns could cause the grid to show one row containing information of content A and result R2.

# 9.6. Dashboard

## 9.6.1. Widgets Module

The Widgets Module is a place to create, configure, and store the visualizations used in dashboards. These elements consist of grids, graphs, and rich text blocks.

Widgets can only be added to a dashboard after they are created in this module.

## 9.6.2. Grid Widgets

Grid Widgets are not created in the Widgets module, but are instead created in the Grid Templates module and use the Meta Tree in the top table. Any existing grids in Grid Templates will

be available as Grid Widgets in a Dashboard. Having aggregators and filters can be used as well in the top table.

ⓘ A note about migration: When a Grid type widget is created in SLIMS version 6.1 and SLIMS is upgraded to version 6.5, the grids defined in the Widgets module are migrated to the Grid Templates module and keep the same name appended with the prefix `Widgets_`.

### 9.6.2.1. Columns Table

The columns that are displayed in the grid are set up in the bottom table. Once saved, the Grid will be available in the list of Dashboard widgets.

The Columns Table has the columns Meta Tree, Function, Field, Title, Sequence Number, and Graph Datatype. Function has the possible values: Count, Mean, or Comma-Separated. Additionally, if grouping is enabled, only one of Having Aggregator, Grouping and any function field can be a custom field.

- *Count*: Displays the amount of entities in this metatree node. Field is hidden and defaulted to Value.
- *Mean*: Displays the mean value of the chosen field.
- *Comma-separated Values*: Displays the value of the chosen field for every entity in the node as a comma-separated string. This is only visible if "There is only one value for this relation" is not checked in the table above (there is a non-unique path from its related metatree to the root metatree). Filtering is re-enabled for columns that have this option enabled.

Users can export grid widgets to Excel directly from the dashboard.

### 9.6.3. Graph Widgets

A graph needs to be created in the Widgets module before it can be used in the Dashboard. Building a graph is similar to building a grid, and begins with choosing the Meta Tree elements in the top table, and any Having aggregators and filters. When 'graph' is selected for the type, the chart type must also be defined so SLIMS understands what the graph should look like. The options are: Area, Bar, Column, Doughnut, Line, Pie, and Radar. When the graph is placed in a dashboard, the chart type can be changed with the right-click menu as well.

Graphs use x and y axis to define the trend of data that is represented. The x axis (horizontal) is the independent value. In general, this is data that is not measurable, like a disease type. The y axis (vertical) is dependent. This is data that can be meaningfully measured, like the number of patients with the disease.

When adding columns in the bottom table, there should be one Y coordinate defined as a Value column, and one or more X coordinate(s) defined as Facet(s).

The user can switch between a grid and a graph display with the "Show grid" icon when viewing the dashboard. The graph can be downloaded as a PNG image file, and the grid can be downloaded for Excel.

### 9.6.4. Rich Text Widgets

A rich text widget needs to be created in the Widgets module to be added to a Dashboard. This creates a CKeditor widget similar to a Note Block in Simplified ELN. When a user enters text or edits the note, it will be edited for all users with Dashboard access. This does not affect the

original Rich Text widget in the Widget module, just the display of the widget on the dashboard. Rich text widgets can be defined as read only so they cannot be edited after import into the dashboard.

## 9.6.5. SLIMS GATE Widgets

SLIMS GATE Widgets are not edited within the Widgets module, but are displayed if they can be used in a dashboard. Only SLIMS GATE flows with ContentManagement usage are displayed.

## 9.6.6. Vaadin Flow Widgets

Vaadin Widgets are not edited within the Widgets module, but are displayed if they can be used in a dashboard.

## 9.6.7. Macro Widgets

Likewise, Macro Widgets are not edited within the Widgets module, but are displayed if they can be used in a dashboard. Only Create macros are available for selection.

## 9.6.8. Create a Dashboard

The Dashboard module allows users to specify visualizations such as grids and graphs, and add quick links for their users. The Dashboard uses Macros, SLIMS GATE flows, Vaadin flows, Grids, and Widgets that are created separately in their respective modules. (Users must have the appropriate permissions to create widgets and dashboards.)

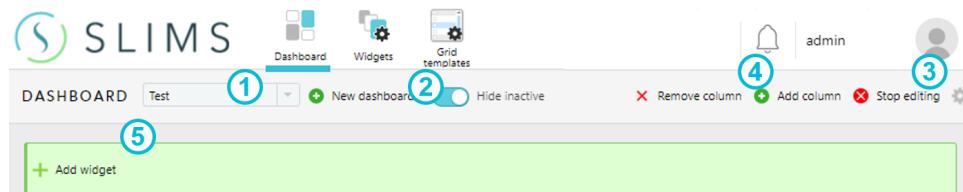
This module requires the following Role Access rights to work:

**Table 9.4. Dashboard Module Permissions**

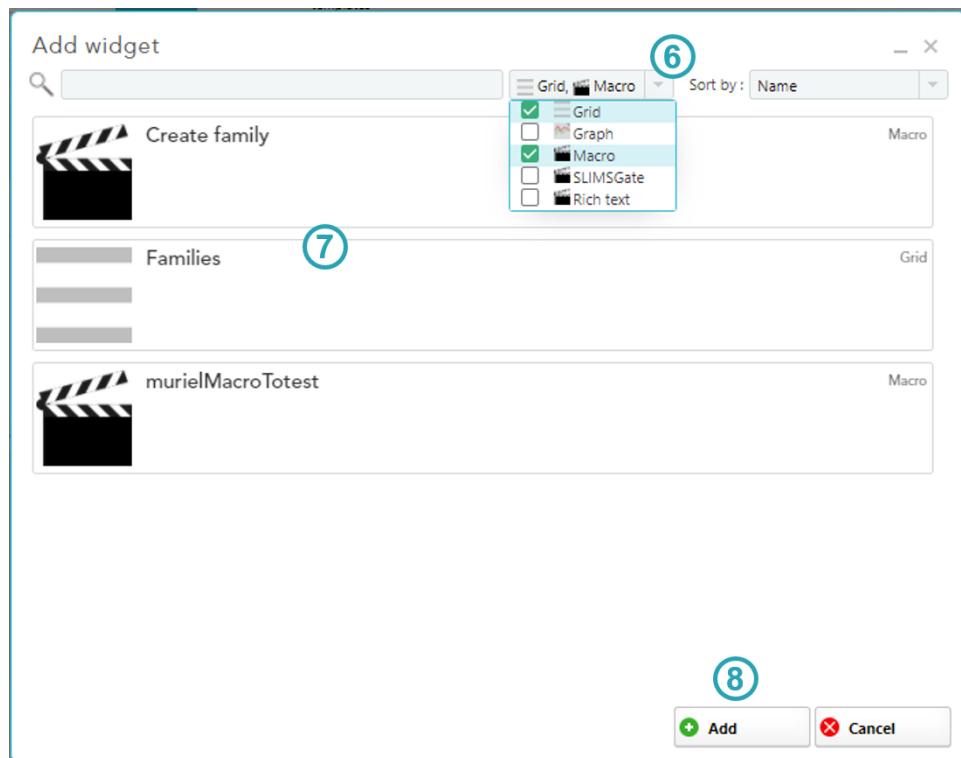
Access rights Category	Access rights Name	Purpose
Routine: Dashboard	Dashboards	This permission provides access to the Dashboards module in which a layout of rich text, charts, grids, widgets, and macros can be designed for users of the SLIMS instance.

The figure below shows a new Dashboard that has not been edited yet.

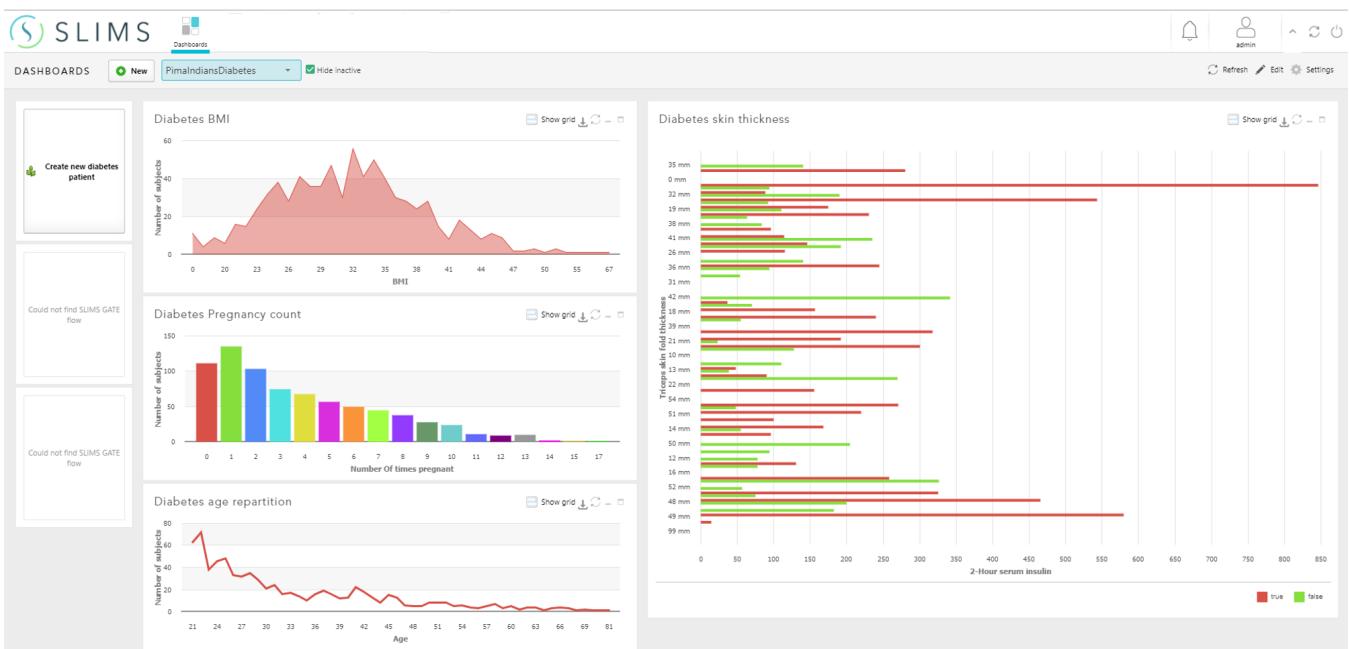
**Figure 9.4. New Dashboard Options**



For users with access, the Dashboard can be accessed to (1) switch between existing dashboards, (2) create new dashboards, (3) enable editing or stop editing, (4) add and remove columns to the dashboard design, and (5) add widgets to the columns in the design.

**Figure 9.5. Add Dashboard Widgets**

The Add Widget pop-up provides search, filter, and sort (6) options to find what should be added to the Dashboard column. To add the filtered widget, click on it in the display (7) and click the Add button (8).

**Figure 9.6. Completed Dashboard Display**

Once the Dashboard is configured, click Save or Stop Editing. The display loads data according to the grids and graphs that have been set up, and the SLIMS GATE and Macro links become clickable. This display is similar to what users without edit permissions will see when they access the Dashboard.

---

# 10. Miscellaneous

The Miscellaneous chapter contains the description of user access and permission modules as well as any other modules that are more narrow in scope, or that are not directly related to core SLIMS setup principles.

## 10.1. Access Rights

Access Rights are defined using the three modules "Users", "Groups," and "Roles." Each user has a Role and can be part of multiple groups.

### 10.1.1. Users

The Users module available under the Setup menu allows SLIMS Administrators to add new SLIMS users or change a user's password. The following parameters can be edited in this module as well:

- › *User Name*: The user's identification name checks for uniqueness to prevent different users from having the same user name. The user name is displayed in SLIMS histories.
- › *Password*: The login password of the user. Passwords are encrypted in SLIMS after being entered and thus cannot be retrieved even by SLIMS staff. The user must contact their SLIMS administrator to define a new password if they forget their current one.
- › *First Name and Last Name*: The user's first and last name.
- › *Full Name*: This field is automatically filled with a concatenation of the user's first name and last name fields when an account is made using Local Authentication. The full name field will remain empty if first name and last name are not filled, but it can be manually entered. When using a third party provider, the full name field is automatically populated with the user's name data from the provider. The full name field is displayed in SLIMS histories. If left empty, only the user name will be displayed in SLIMS histories.
- › *Email*: Contact email address of the user.
- › *Role*: Assigns the Role of the user in the system, which defines the level of access the user has (All, Group, Personal, None) and which modules the user can access. See the *Roles* section for details about defining access.
- › *License Type*: Service User, Named, Concurrent, and Read-Only. Named users are identified specifically, and only a certain number of them can be active in the system. Concurrent users count against the total number of users that a customer can have active at one time in the system, according to their license. Read-Only and Service accounts do not count against either of those totals. Service Users can be for SLIMS staff, SLIMSGATE accounts, Vaadin accounts, or other support-specific engineer accounts are considered as 'Licenseless' users.
- › *Authentication Method*: Authentication methods are configured in the Authentication module. This dropdown shows the options available and shows the authentication method that was used to create the particular user account. Local Authentication means that the user's

name and password are handled by SLIMS. LDAP means that the user credentials are handled by an LDAP system instead of SLIMS. OpenID Connect means that SLIMS relies on authentication that is performed by an OpenID Connect Provider to verify the identity of a user (Okta, Google, and identity providers that support OpenID Connect Discovery are supported). Finally, OpenLab Shared Services means that the user's credentials are handled by Agilent OpenLab Shared Services.

After an account is initially created using one method of authentication, it can be changed with this dropdown if needed. For example, updating an account of any kind to use local authentication will always work as it provides a back-up method for users to access SLIMS in case any external provider has an outage or failure. However, other changes such as between local authentication, LDAP, and OpenLab Shared Services may not work as intended. Additionally, for OpenID Connect accounts to work, they must have been created with the OpenID Connect system and not with local authentication, LDAP, or OpenLab Shared Services.

- › *Language*: SLIMS supports English, French, Dutch, German, and Chinese (Mandarin). However, it is recommended to choose one language and use it for all users in the same organization.
  - › *Display notifications with this level or higher*: Info, Success, Warning, or Error. These notification types are considered in order as various degrees of severity, so if a user's account was set to receive any Warning notifications, they would also receive Success and Info notifications. Notifications correlate with the Bell Icon in the top bar of SLIMS, and may also correspond with an email being sent to the user.
    - ⓘ *Enable Browser Notifications (Switch)*: This allows the notification to appear in the user's browser, not just on the Bell Icon in the top bar of SLIMS.
  - › *Avatar*: This section shows an edit icon when hovering with the mouse. When clicked, administrators can select one of the existing SLIMS avatars to represent the user to other users, or even upload an avatar of their own making. Avatars must be 128 x 128 pixel image files and either JPG, PNG, or GIF file types. Users can only delete the avatars that they added - not default system avatars or those uploaded by other users.
  - › *Active*: Inactive users cannot log into SLIMS and can no longer be selected as users when creating new records.
  - › *Locked*: Read-only field that shows if a user is locked out of the system from too many wrong password attempts.
  - › **Fields for ECM3**: Four fields are available when using ECM3 for attachments that allow you to connect to a user account for ECM3:
    - *ECM Account*: The account name for the ECM3 server
    - *ECM Domain*: The ECM3 domain name
    - *ECM Username*: The user's ECM3 Username
    - *ECM Password*: The user's ECM3 Password
- ⓘ A user can only be removed if they have never used their account, like when an account is created by mistake. This helps retain the history of all users who logged into SLIMS.

### 10.1.1.1. Group Rights Overview

The Group Rights Overview tab contains a display of all groups that were created in SLIMS and the possible operations the selected user is allowed to take on records belonging to each group. When selecting a group, further detail in the right panel explain why certain operations are enabled or disabled.

### 10.1.1.2. User Lockout

A user will become locked out of the system if local authentication is enabled on their account and the amount of failed login attempts is exceeded. The number of login attempts before locking the user out can be specified in the *Lab Settings module*.

When a user is locked out or unlocked a new event will be created. These events are visible in the *License Module* in the History tab. When a user is locked out and attempts to login, a failed login attempt event will be created. The user will get an error message on the login page to inform that they are locked out.

A user can be unlocked by right-clicking on the User Name and choosing the 'Unlock' action. Access to the *License Module* is required for a role in order to access the 'Unlock' feature.

### 10.1.1.3. Password Complexity

The complexity required for passwords can be specified in the *Authentication module*. This feature is only effective if Local Authentication is enabled on the user's account. If a method besides Local Authentication is used for user accounts, the password is managed by the configuration of the third party authentication provider. The complexity can be set to four levels:

- *Simple*: 0 - 40 characters
- *Basic*: 1 - 40 characters
- *Standard*: 8 - 40 characters
- *CLIA Compliant*: 8 - 40 characters, and requires at least two each of digits, lowercase letters, uppercase letters, and special characters.

When a new complexity is set, all the users have to redefine their password at the next login.

### 10.1.1.4. Password Expiration Period

The expiration of passwords can be specified in the *Authentication module*. This feature is only effective if Local Authentication is enabled on the user's account. If a method besides Local Authentication is used for user accounts, the password is managed by the configuration of the third party authentication provider. For Local Authentication, when the set number of days passes, the user's password expires and they are asked to change it after logging in.

ⓘ The upgrade to SLIMS version 5.0 reinitializes the last change date to be the upgrade date.

The functionality "Can Terminate User Sessions" can be activated for administrative users. This allows a user to see the sessions of users that are logged in and terminate their sessions by right-clicking them and using the 'Terminate User Session' action in the context menu.

The following Role Access rights can be configured for user sessions:

**Table 10.1. Required Permissions for Ending User Sessions**

Access rights Category	Access rights Name	Purpose
Setup: User	Can Terminate User Sessions	When enabled, the role user can see the sessions of other users that are logged in and terminate their sessions by right-clicking them and using the "Terminate User Session" action in the context menu. This is most suited for administrators.

### 10.1.1.5. Group Access Rights

This option controls the Access Rights options that are displayed at the bottom of a record form, location form, etc. The grouping of access rights can be enabled or disabled in the *Lab Settings module*. When enabled, the user and group field will be shown together at the bottom of the form preceded by a radio button with the options:

- *Group*: If Group is filled, users that are part of the selected Group will be able to access the record. The User and Group fields will be visible underneath.
- *Owner only*: If Owner Only is filled, only the user creating the record (who counts as the 'owner') will be able to access it. Anything filled in the Group field will be cleared, and only the User field will be available underneath.
- *Public*: If Public is filled, any users of the SLIMS instance can access the record. User and Group will be hidden, and will be cleared if they contained any values.

### 10.1.2. Groups

When a Role's access level is defined as "Group," a user of that role can only access records that give rights to a group they are a member of. Groups are defined in the "Groups" module, and have a Name, Description, and a list of users or possibly other sub-groups that the Group consists of.

The Detail tab shows the group Name, Description, Active option, and other custom fields.

Users and Groups can be added to a group in the Rights tab.

#### 10.1.2.1. Adding Users to a Group

When adding a user to a group, it is possible to choose to enable/disable an operation (Add, Update, Remove) for that user in the chosen group. It is only possible to change the operation if the related role level is 'Group' for that user. If the operation-related role level is 'All,' the operation will be shown as enabled and checked in the User Rights list by default, because the user can execute the operation of records belonging to the group. If the related operation level is 'Personal' or 'None,' the operation will be shown as disabled and unchecked because the user cannot execute the operation on records belonging to the group.

### 10.1.2.2. Adding Groups to a Group

When adding another group to a group, all users of the sub-group will be included in the selected group. The operation of all the users of the included sub-group in the selected group can be managed with the following settings:

- Yes: The user can execute the operation
- No: The user cannot execute the operation
- Inherit: The user can execute the operation if they can execute this operation in the other group.

If the user is present in both a group and in included sub-groups, the operation values of the user in the group take priority over the operation values of the user in the included sub-groups. If the user is not present in a group, but in multiple included sub-groups, the maximum of all operation values of the user in the included sub-group is used. For example, if one of the included sub-groups has an operation value of Yes or one of the included sub-groups has operation value Inherit and the user can execute the operation in the included group, then the user can also execute the operation in the selected group.

### 10.1.2.3. User Rights Overview

In the User Rights Overview tab, all users and the possible operations they are allowed to do on records belonging to the selected group are listed. When selecting a user, a detailed form appears to explain why certain operations are enabled/disabled.

- ① Inactive groups and users are not available in the selection list when creating new records.

### 10.1.3. Roles

A role defines which functionality a user can access and their different access levels.

The main parameters of a role are as follows:

- › *Name*: Name of the role.
- › *Description*: Internal help to describe what a user with this role can do.
- › *Create Level*: See Section 10.1.4.
- › *Access Level*: See Section 10.1.4.
- › *Update Level*: See Section 10.1.4.
- › *Remove Level*: See Section 10.1.4.
- › *Content Types*: List of content types that are restricted to this role.
- › *Active*: A role can be inactivated only if this role is assigned to no active users. Roles cannot be permanently removed in SLIMS, so unused roles can be inactivated to hide them from the list of roles.

A full list of functionalities is available in the second tab of the right panel of the Roles module. Whenever a functionality is enabled, it means that this functionality is available for the selected role.

## 10.1.4. Defining Access Rights

The accessibility of a record can be:

- › **Group:** *Group filled in (user filled in or not)*, the record is accessible by the members of the group (if the group members have at least group access).
- › **Owner Only:** *User is filled in, and group is not filled in, then the record is only accessible by that user.*
- › **Public:** *Neither the user or group are filled in:* the record is accessible by everyone.

When the labsetting 'Group access rights fields together' is enabled, the user and group field will be shown together at the bottom of the record form preceded by a radiobutton with the options:

- › **Group:** this option will be marked if a group is filled in. The user and group field will be visible underneath.
- › **Owner only:** this option will be marked if only a user is filled in. The user field will be visible underneath. When changing to this option, any filled in group will be cleared.
- › **Public:** this option will be marked if no group an user are filled in. The user and group field will not be shown. When changing to this option, any filled in user and group will be cleared.

Regardless of the labsetting, this functionality is not used in content specific actions like edit, derive and also not in macro's.

There are four possible levels for Create, Access, Update or Remove data:

- Nothing
- Personal
- Group
- All

For 'Create level' they mean:

- users with create level **Nothing** are not allowed to create records;
- users with create level **Personal** can only create records belonging to themselves;
- users with create level **Group** can create records owned by themselves or by a group they belong to;
- users with create level **All** can create public, group or owner only records.

For Access, Update and Remove level, they mean:

- users with access level **Nothing** have access to no record;
- users with access level **Personal** have access to public records and to records owned by themselves;

- users with access level **Group** have access to public records and to records owned by themselves or by a group they belong to
- users with access level **All** have access to everything, independent of the filled in user and group.

and similarly for update and remove level.

Table Table 10.2 shows the access of the four possible levels. Each line reports a level and the records that level has access to. For example, line "Personal" shows that an access level Personal implies that level can access records that are public or owned by the user.

**Table 10.2. Create, Access, Update and Remove levels**

	Group	Owner only	Public
<b>Nothing</b>			
<b>Personal</b>		✓	✓
<b>Group</b>	✓	✓	✓
<b>All</b>	everything	everything	everything

Additionally, remove level has a fifth option, 'Unrestricted'. This level allows the removal of records that cannot be restricted by a user or group (for instance Content types, Macros, Label templates, ...) This level does not allow the removal of records that can be restricted by a user or group (for instance Content, Location, Project, ...) Suggested usage: power users that need removal rights for certain configuration they have access to, but disable removal of routine data.

Following example shows the different access rights a user "User\_A" will have according to the access level he is given (Nothing, Personal, Group or All):

**Table 10.3. Example of access rights**

Access Level User: User_A	Owner on- ly items: User: user_B Group: ---	Group items: User: user_B Group: Lab_A	Public items: User: --- Group: ---
<b>Nothing</b>	---		
<b>Personal</b>	---		✓
<b>Group</b>	User_A $\notin$ Lab_A		✓
<b>All</b>	User_A $\in$ Lab_A	✓	✓
	---	✓	✓

## 10.1.5. Update access rights on records

Access rights on a record can be modified by changing the value of field User and/or Group.

If you have create level 'Group' you will only be able to select groups in which you have create rights. It is also not possible to change an existing group (of for instance a content record) to a group in which you do not have create rights

As a general rule, updating access rights takes effect only on the updated record(s). Exceptions to this rule are:

- Simplified runs: if the access rights of a simplified run is updated, the modification is propagated to the attachments of all its blocks.

## 10.1.6. ELN Sharing

### 10.1.6.1. Sharing Runs

In the ELN tree of the ELN module, it is possible to share a simplified or default protocol run with other users who do not have access to the run.

The share action is available in the context menu and opens the Share Window, consisting of these options:

- A share link that will redirect an enduser directly to the run, if he has access to it.
- The property 'Users can share records of this group by adding additional users or groups':
  - This can only be changed by users who can edit the original run.
  - Enabling this allows users that have shared edit rights to add additional users/groups to the shared run.
  - Disabling this gives users that have shared edit rights to the shared run a readonly view of the Shared window.
- Shared Users: Similar functionality to adding a user to a Group, but now only for the shared run (Only the update and remove operation are available).
- Shared Users of Groups: Similar functionality to include users of a group into a Group, but now only for the shared run (Only the update and remove operation are available).
- Unshare Button: Undo the sharing by changing the shared group of the run back to the original group.

Sharing is only possible for runs that already have a group assigned. The only users who can share the run are those who can edit the original run, and belong to the same group as the run.

① Even a user that has admin rights to edit everything cannot share a run unless they belong to the same group as the run.

Sharing a run results in changing the group of the run to a new shared group which takes the name of the original group and adds a star: group A -> group A\* Unsharing a run results in changing the group of the run back to the original group, removing the star. For example: group A\* -> group A

### 10.1.6.2. Shared with Me

The ELN tree of the ELN module has a default folder on top called 'Shared with Me.' The folder contains all runs that are shared with the current logged in user (i.e. the user is added to the shared run or the user is part of a group that is added to the shared run). Content that is linked

in the shared run will be visible to the shared user regardless of the access rights by the group of the content itself, and update and remove rights remain. Attachments that are linked to the shared run will be propagated to the shared group by the sharing action, so users that can access the shared run can also access its attachments.

### 10.1.6.3. Shared with Others

The ELN tree of the ELN module has a default folder on top called 'Shared with Others.' The folder contains all runs that are shared by the current logged in user (i.e. the user is part of the original group of the shared run or the user is part of a group included in the original group of the shared run).

## 10.2. Analysis Modules

The analysis modules allow R scripts to be run on data available in SLIMS using Grid Templates.

These related modules require the following Role Access rights to work:

**Table 10.4. Analysis Module Permissions**

Access rights Category	Access rights Name	Purpose
Routine: Analysis	Analysis	This permission provides access to the Analysis module in which the analysis runs are started, viewed, and the run outputs are logged.
Setup: Miscellaneous	Analysis Templates	This permission allows role users to access the Analysis Templates module to create analysis templates that are used in the analysis module. Analysis templates require a grid template and an R script.

### 10.2.1. Analysis Template

In this configuration module, the grid template is selected and the R script to be run is uploaded.

The module displays the list of existing templates in the left panel. The details on the template are available in the main panel, split into two tabs:

- *Details:* Standard details including Name, Creation Date, and User
- *Processing:* This tab contains the main script as well as all the other files necessary to its execution.

A new template can be created by pressing the 'New' button at the top of the list. Three parameters are available:

- **Name**
- **Grid Template:** The grid template used to extract data from SLIMS. This selection cannot be altered after the creation of the analysis template.
- **Analysis Script:** The main script run by R. The script can also be added later on.

A script example can be downloaded in the processing tab. It contains a function called `compute` which takes two arguments:

- `gridDataFile`: The absolute path to the CSV file containing the data in the Grid used. To read the file, one can use: `read.csv(gridDataFile)`
- `outputDirectory`: Absolute path to the directory from which the output will be downloaded back into SLIMS. To write a `data.frame` to a CSV file: `write.csv(data, file=paste(outputDirectory, 'data.csv', sep='/'))`

The data as it is available in the grid can also be downloaded. It is compiled in a CSV file in the exact same format as will be provided to the script when run on the server.

The attachment grid displays all the files that will be uploaded to the R working directory except for the grid data itself. Exactly one of the attachments must be flagged as the Main Script for the analysis to run successfully.

## 10.2.2. Analysis

The analysis module is divided into three panels:

- The left list contains all the created analysis templates.
- The second list contains all the runs started for a specific template.
- The last panel contains all the details and data of a specific run, including logs output by R.

Pressing the 'Run Analysis' button creates a new run. All the files attached to the template are uploaded to the server running Rserve and the main script's `compute` function is called.

Once the run is finished, all the files in the output folder are downloaded back into SLIMS and attached in the Output tab.

All standard and error outputs from R are displayed in the Logs tab. Debug information can therefore be provided using the R `print` function. In case of failure, the error message is displayed in the Details tab as the status message.

## 10.3. Attachments

SLIMS' records such as experiments, contents, locations, or orders can possess links to files (images, documents, etc.) by using the Attachments tool. This tool can be accessed with its icon  with which files can be added, viewed, and removed.

The following Role Access rights can be configured for use with this feature:

**Table 10.5. Attachment Permissions**

Access rights Category	Access rights Name	Purpose
Routine: Attachments	Attachments	This permission provides access to the Attachments module which allows the role user to view and manage all attachments in the SLIMS instance that they have access to (user/group permissions to view).
Setup: Miscellaneous	Attachment Types	This permission provides access to the Attachment Types module which allows the role user to arrange attachments into groups based on common properties.

Records containing attachments can be easily recognized by the attachment icon. The icon is gray for records without attachments and light blue for records with attachments. The number of attached files is displayed by hovering the mouse cursor over the attachment icon .

### 10.3.1. Create Attachments

By clicking on the Attachments icon, a pop-up window appears and all the attached files are shown. New attachments can be created in three different ways:

1. *Click the "New" button and choose Local File:*
  - Works for large files.
  - Can upload up to 10 files at once.
  - Progress is shown in the status bar of the browser.
2. *Click the "New" button and choose Server File:* This option is available only if the 'Server attachment root directory' was specified in the Lab Settings.
  - Works for large files.
  - This is a fast method, but the files have to be accessible from the SLIMS Server.
  - Allows the selection of an entire directory to be added as an attachment in SLIMS. Downloading such a directory yields a ZIP file.
  - Allows the selection of multiple files at once with no limitation in number using the Ctrl (Command for MAC) or Shift key.
  - Progress is shown in a progress bar in SLIMS. If the user clicks "Cancel," all files that were already uploaded will be removed.
3. *Drag and Drop on the Attachment Pop-up:* More details of this method are described in the *Drag and Drop* section.

- Works for large files.
- Progress can be consulted via a button in the top right corner .
- Can be used to upload up to 10 files at once by Dragging and Dropping multiple files at once.
- Works in Chrome, Safari, and Firefox, but not in Internet Explorer.

An attachment can be unlinked from a record using the break link icon . This action does not delete the attachment, it just removes the connection between the attachment and the record. If all of the links to an attachment are removed, then the attachment cannot be viewed from the records anymore. However, the History will show that the link was removed and the unlinked attachments can be viewed in the Attachments Module. In this way, breaking the link acts as a 'soft remove' instead of a permanent removal.

① Users can add an unlimited amount of attachments to a record. The amount and size of attachments are limited only by the storage server size.

### 10.3.2. Open and Download Attachments

Attachments can be opened and/or downloaded from the Attachments pop-up.

- A single attachment can be opened in the browser by clicking the "Open" icon (magnifying glass). However, the ability to open a file directly in the browser depends on the browser settings. If it cannot be opened, the file will be downloaded instead.
- A single attachment can be downloaded clicking on the "Download" icon (page with red arrow).
- Multiple attachments can be selected and downloaded in a ZIP file by clicking on the action "Zip and download" (ZIP icon in Selection menu). The name of the ZIP file follows the format [Table]\_Attachments\_yyyyMMdd\_hhmmss.zip.

### 10.3.3. Preview Attachments

Supported image previews are: JPG, JPEG, PNG, GIF, BMP, PDF, DOCX, PPTX, TIF and TIFF (images, PDF, Word, and PowerPoint). Previews show only the first document page and are available in:

- *Attachment Preview Tab*: The preview is automatically shown for supported file and image formats.
- *Rich Text Editor*: The user can choose to add a Preview for supported file/image formats when Dragging and Dropping an image into the editor. Please refer to the Rich Text Editor section for more details.

Previews for DOCX files have the following limitations:

- Images available in the Word document are not shown in the preview.
- Text highlights are not supported in the preview.

When the supported image types listed above are attached to content, and an Attachment Field is added to show up in the Content Grid, the text link to the attachment can be hovered to show a small image preview.

### 10.3.4. Attachment Links

One file can be attached to multiple records using the "Copy to Clipboard" tool. This makes it possible to download a file once and create links to it from different records. "Copy to Clipboard" is available when right-clicking on the attachment within the Attachments Module or in the Attachment pop-up window. The following steps explain how to use the tool.

- First, the file has to be attached to a SLIMS record using the steps previously explained.
- Then a link to the attachment has to be created. This can be done either by right-clicking on the attached file and selecting "Copy to Clipboard," or by selecting multiple attachments and clicking "Copy to Clipboard" in the Selection menu.
- Finally, select a different record and open the pop-up window for attachments. Then click the "Paste from Clipboard" button to paste the link to the attachment(s) that were copied before.

⚠ If an attachment is deleted permanently, all attachment links will be deleted from all records containing it.

ⓘ As stated earlier, when the attachment link is broken using the 'Break Link' action, the attachments disappear from the record but are not deleted. The action can be seen in History, and the attachment is still accessible in the Attachments Module.

Symbolic links of the attachments on the server can be created using a workflow in SLIMS. See the *Attachments Symlinks Section* for more details.

### 10.3.5. Attachment types

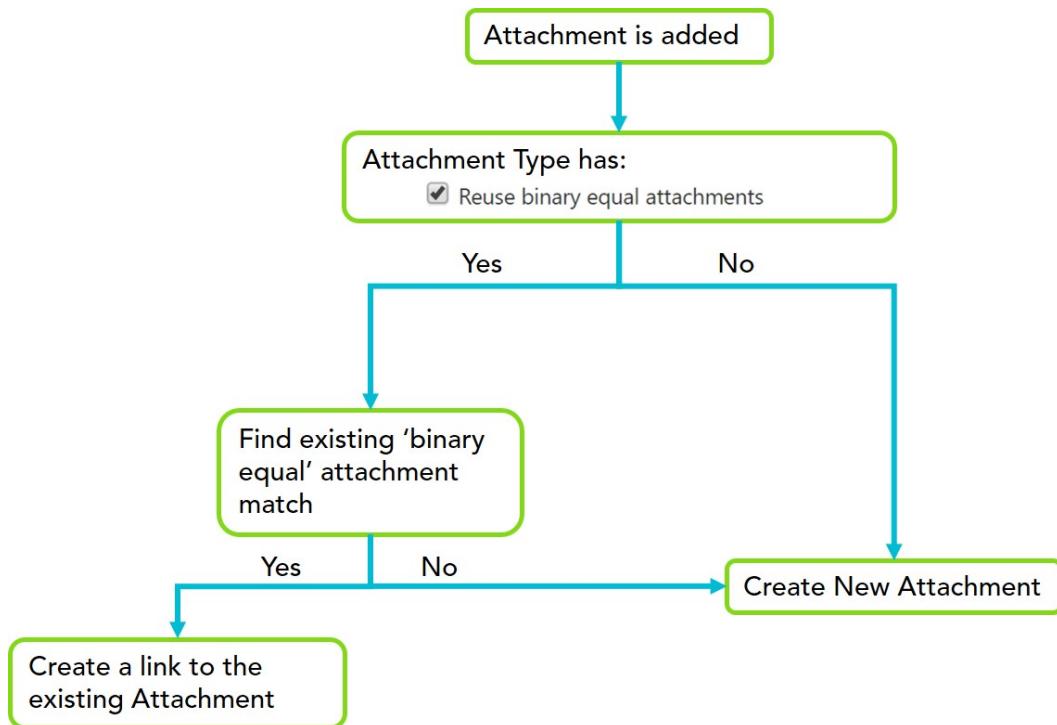
Attachments can be arranged into types to group attachments based on common properties such as the file type or the source of the file.

These can be added using the "Attachment Types" module available under the Setup Menu. An attachment type is defined with a name, a description, and the parameter "Reuse Binary Equal Attachments."

ⓘ Specific custom fields can be added for attachments with the same type. See the Fields section for more details on how to add custom fields.

"Reuse Binary Equal Attachments" is used to save disk space by preventing file duplication on the server. Every time an attachment is uploaded, SLIMS checks if its attachment type has "Reuse Binary Equal Attachments" enabled. If it does, SLIMS queries all attachments to look for those with the same inherent binary code and searches for matches. If it finds no matches, it creates the new attachment (Figure 10.1), but if it finds a match, it creates a link to the existing attachment (the matching attachment that was created first if multiple matches are found) instead of creating a new one.

This is useful to save space when the same file needs to be reused in multiple records and does not change. If attachment types have been defined, the user can choose a type for a new attachment when adding it.

**Figure 10.1. Uploading Attachments in SLIMS**

⚠ The default behavior will be used if no attachment type is specified, meaning that a new file will be created for every attachment added in SLIMS.

### 10.3.6. Creating Agilent OpenLab ECM Attachment Types

You can configure Attachment types so that attachments of that type will automatically upload to ECM. The option "Automatically upload an attachment of this attachment type to ECM" can be enabled on an attachment type, but it will not be available unless there is an ECM URL added in the OpenLab tab of the Authentication module.

Your SLIMS contact can provide the technical requirements that list the ECM versions supported by SLIMS.

Once an attachment type with this option has been created, a groovy editor box will be available. Attachment types that upload to OpenLab ECM require a groovy script that determines the path to save attachments in OpenLab ECM. To do so, you can configure the groovy script to return an object of type `StorageLocation`. The `StorageLocation` represents a location (cabinet, drawer, folder) in ECM. They can be constructed like this:

```

return StoragePosition.builder()
    .location("Customer")
    .cabinet("SLIMS")
    .drawer(attachment.user.userName)
    .folder(attachment.attm_name.getAt(0).toUpperCase())
    .build()
  
```

The example script results in a location in ECM like this:

Contact \ SLIMS \ SLIMS \ Admin \ S

Attachments that are changed to an attachment type that uploads to ECM will be automatically uploaded to OpenLab ECM. Attachments that are uploaded to ECM have fields in the attachment details that reflect the status of the upload to ECM. For an attachment to be uploaded to ECM the attachment also requires an ECM upload user to be set. These users need to have all the OpenLab ECM fields filled, which you can read details about in the User Section.

The ECM3 upload status has the following options:

- **Upload Pending:** The upload has not begun processing, and could be held up by missing information. If the ECM URL (in Lab Settings) is not set or no ECM Upload User has been set (in the attachment details), the attachments will remain at "Pending" status until those necessary fields are filled in.
- **Uploading:** The necessary information has been completed and the upload is processing.
- **Upload Successful:** The attachment has been successfully uploaded to ECM in the location governed by the groovy script of the attachment type. The Upload User field in the details can't be changed anymore and a URL pointing to the attachment in ECM is shown.
- **Upload Failed:** The upload should not fail due to a missing user or ECM URL, which only cause the upload to be stuck at "Pending" (like above). If other issues are encountered and a failure occurred, a log message will be available in the Upload Failed Event with details about what went wrong. After fixing the issue, the user can retry the upload manually. The attachment can be right-clicked to access the context menu, and the action "Repeat attempt at ECM upload" can be selected to trigger the upload again.

### 10.3.7. Drag and Drop

Files can be attached to SLIMS records easily using 'Drag and Drop.' Files can be dragged from a user's local machine and dropped into the SLIMS web page. This can be done by selecting one or more files on the user's computer (from a folder or desktop interface), holding the mouse button down, and hovering over the SLIMS page. When files are dragged over the SLIMS interface, blue panels are displayed to highlight any zones where files can be dropped.

These are zones where files can be dropped in SLIMS:

- › **Normal Grids:** The files get attached to the record the user is hovering over.
- › **Attachment Grids:** The files get attached to the record corresponding to the attachment grid.
- › **Attachment Tile Grid:** The files get attached to the record corresponding to the attachment tile grid.
- › **Rich Text Fields:** The files get attached to the record that is currently being edited. A link or image will also be inserted into the rich text. Please refer to the *Rich Text Editor* section for more details.

SLIMS attachments can be linked to other records using Drag and Drop as well. Note that this action will only create the link to the same file from a different record (see "Attachment Links" for more details).

When using Drag and Drop, an additional tool shows the download progress of attachments accessible from the icon . The progress of the uploading files is displayed with the following upload statuses: Not Started, In Progress, or Finished. As long as the upload has not completed yet, the upload of files can be Cancelled during progress.  When the upload of multiple

files is cancelled, the unfinished files as well as the files that have finished uploading as part of the action are cancelled.

ⓘ Drag and Drop is possible in most browsers, but replaces SLIMS with a preview of the file in Internet Explorer because it is not supported. This is default browser behavior and cannot be changed by SLIMS.

⚠ When using Drag and Drop without the "Drop here" overlay, the browser replaces SLIMS with a preview of the file.

## 10.3.8. Binary Search

Attachment grids and the ELN global search have a Query field available which allow searching for text in attachments.

The following queries are possible:

- A single term, or a phrase delimited by double quotes. Examples:

```
Mouse
"Lorem ipsum dolor ut nacem"
```

- Multiple terms or phrases. These can be combined with the words AND and OR. If neither is used, OR is implied. Examples:

```
Color Colour
Color OR Colour
Color AND Colour
```

- Text limited to a field. Current fields are "Contents", "Name," and "Pk." Examples:

```
Pk:11 OR (Name:Report AND Contents:"Foo Bar")
```

- Ranges. Square brackets for inclusive ranges, curly for exclusive. The following are equivalent:

```
Name:[2001 TO 2014]
Name:{2000 TO 2015}
Name:{2000 TO 2014}
```

- Wildcards: ? denotes a single character, \* denotes 0 or more characters. Wildcards do not work within double quotes. Wildcards cannot be used as the first character in a term. Examples:

```
Blood_*_Level
201?
```

- Fuzzy searches, with an optional parameter denoting the allowed difference between search term and result. Default is 0.5. Fuzzy search does not work with double quotes. Examples:

```
Mendelejev~
Mendelejev~0.8
```

- Words that are at most a certain distance (a number of words) apart. Examples:

```
"Chicken Rice"~20
```

- NOT and "-" for terms that are forbidden. "+" for required terms. Examples:

```
+Chicken -Curry
Error NOT "Low Priority"
```

The following characters need to be escaped by putting a \ before the character:

```
+ - ! ( ) { } [ ] ^ " ~ * ? : \
```

For example, the equation

```
"d=b^2-4ac"
```

is searchable by the query:

```
d=b\^2\^-4ac
```

In addition, "&&" and "||" are parsed as "AND" and "OR". This can be avoided by escaping one or both of the characters.

Both the query and the grid restrictions will be applied to the attachments. If there are less than 1000 matches for the query, or there are less than 10,000 records in the grid, at most 1000 matching attachments will be shown in the grid. If neither of these restrictions are true, the query will fail and it should be refined to decrease the number of matches.

Files with the following extensions can be searched:

```
.txt .pdf .rtf .xml .html .htm .doc .docx .xls .xlsx .ppt .pptx .odt .fodt .ods .fods .odp .fodp
```

## 10.4. Notification Setup

### 10.4.1. Email Templates Module

Email templates are created in the Email Templates module. An email template is used to send email(s) periodically or upon a specific event expressed in a Rule of type *Status* or *Conditional Value Expression*. Refer to 'Rules' for detailed explanation on Rules triggering emails.

When creating an email template, the following fields should be filled:

- **Name:** The name of the template that may be referred to in a 'Rule' using  
`email("nameOfTheTemplate")`
- **Description:** Any desired explanation for how the email template is used.
- **Template Expression:** The subject, body, and list of receivers can be built with Groovy script to define the email template.

Once the expression is defined, the email can be set up to trigger via CRON, which is done in the Scheduled Jobs Module. Refer to the *Scheduled Jobs* section for details.

#### 10.4.1.1. Template Expression

Clicking on *Insert* at top left corner of the Groovy editor will produce a default empty email template:

```
String subject = null;
String body = null;
Email.Builder builder = new Email.Builder(subject,body);
//complete builder here
return builder.build();
```

In the Groovy script, built-in SLIMS method and variables may be used to build the subject, the body, or the list of receivers. The 'Groovy Scripting' chapter shows examples on how to write Groovy scripts.

Note that if the email is triggered by a Rule, then the `updatedRecord` object can be used in the Groovy script. (see Attachment example below)

Here are the methods dedicated to email objects.

To add the sender, you will use:

```
builder.withSender("sender.address@xxx.yyy")
```

To add a receiver, you will use:

```
builder.addReceiver("receiver.address@xxx.yyy")
```

or

```
builder.withReceivers(receiverList)
```

where `receiverList` is a list of addresses.

To add a CC, you will use:

```
builder.addCC("receiver.address@xxx.yyy")
```

or

```
builder.withCCs(CCList)
```

where `CCList` is a list of addresses.

To add a BCC, you will use:

```
builder.addBCC("receiver.address@xxx.yyy")
```

or

```
builder.withBCCs(CCList)
```

where `CCList` is a list of addresses.

To attach files, you will use:

```
builder.addAttachment(attachmentName, attachmentInputStream)
```

or

```
builder.withAttachments(attachments)
```

where `attachments` is a map as in following example where all files attached to a content are attached to the email :

```
String subject = "Content with Attachment " + updatedRecord.get("cntn_id");

Map<String,InputStream> attachments = new HashMap<String,InputStream>();

// Get the attachments_links for the content that triggered the email
List<Map<String, Object>> attachmentLinks = daoHelper.fetch("AttachmentLink",
    SlimsRestrictions.and(SlimsRestrictions.equals("atln_broken", 0),
        SlimsRestrictions.equals("atln_recordTable", "Content"),
        SlimsRestrictions.equals("atln_recordPk", updatedRecord.get("cntn_pk"))));
// Get the attachments and add the content to a list
for(Map<String, Object> link : attachmentLinks){
    Map<String, Object> attachment = daoHelper.find("Attachment",
        link.get("atln_fk_attachment"));
    //Add an attachment to this email
    InputStream inputStream =
        attachmentHelper.getAttachmentStream(attachment.get("attn_pk"));
```

```

    // Note that this example expects all attachment names to be unique, otherwise they will
    be replaced in the Map
    attachments.put(attachment.get("attm_name"), inputStream );
}
// Build the email with attached files
Email emailContent = new Email.Builder(subject, "no body").withSender("sender")
    .addReceiver("receiver.address@xxx.yyy").withAttachments(attachments).build();

//Emails returned
List<Email> emails = new ArrayList<Email>(0);
emails.add(emailContent);
return emails;

```

⚠ In order for emails to be delivered, it is necessary to set the SMTP server URI in the Lab Settings as *Email templates SMTP URI*.

## 10.4.2. Scheduled Jobs Module

The following Role Access rights can be configured for use in this module:

**Table 10.6. Required Permissions for Scheduled Jobs Module**

Access rights Category	Access rights Name	Purpose
Setup: Miscellaneous	Scheduled Jobs	This permission provides access to the Scheduled Jobs module which allows role users to write a condition (Cron job) to define when a given groovy script should be executed in time (to schedule it). The target script content can be used to communicate different ways with a user: from sending an email based on an email template, to send a browser notification, to show up as a warning or an error in SLIMS.

Scheduled Jobs can be created to send notifications or retrigger evaluations on a schedule. A CRON expression needs to be assigned to each scheduled job so it can perform its task on schedule, whether that is to send an email based on the email template that was created, or re-trigger the value expression set up in the scheduled job.

When updating, a liquibase script will take care of creating an equivalent scheduled job for every email template having the same CRON expression and as a value expression:

```
email('EMAIL_TEMPLATE_NAME')
```

where EMAIL\_TEMPLATE\_NAME is the name of the email template that was constructed in the Email Templates module.

The syntax of the CRON expression is the same as Apache Quartz's which can be found [here](http://www.quartz-scheduler.org/documentation/quartz-2.x/tutorials/crontrigger.html#format). The available functions in the value expression are:

- email(String emailTemplateName)

- `notify(Map config)`
- `warn(String message)`
- And the standard ones, such as log, empty, etc.

See [http://docs.oracle.com/cd/E12058\\_01/doc/doc.1014/e12030/cron\\_expressions.htm](http://docs.oracle.com/cd/E12058_01/doc/doc.1014/e12030/cron_expressions.htm) for more on CRON expressions

The Scheduled Jobs Module contains the list of jobs in the left panel. These consist of the following parameters:

- *Name*: The name of the scheduled job. As best practice, this should correlate with the email template name in some way to aid with other users' comprehension.
- *Description*: Any helpful text to describe the usage of the email or job.
- *CRON Expression*: Enter the CRON expression described earlier to schedule the job.
- *Maximum execution time (minutes)*: Default = 5, minimum = 1, maximum = 60. Identifies the number of minutes the scheduled job is allowed to execute. This setting prevents unending jobs from running which could cause problems. A job is canceled if it takes too long to finish running. ① No two scheduled jobs are allowed to run at the same time.
- *Value Expression*: Specify an expression that will be evaluated as the value of this field. Expressions should be written in the 'Groovy' language.

A value expression can be written to trigger evaluation of a value expression on a custom field: `triggerRecalculation(String table, Long recordPk, boolean separateTransaction)`

#### **Parameters:**

`String table` - The table that the record belongs to.

`Long recordPk` - The primary key of the record to find.

`boolean separateTransaction` - True or false. Determines whether the job will be executed in an existing transaction (false) or within a transaction of its own (true).

#### **Purpose:**

This is a way to trigger Groovy that should be recalculated from a scheduled job, like evaluations that should be redone overnight.

#### **Example:**

Given that I have a custom field that has a value expression to update (like `return new Date()`) and "value expression is evaluated on every save" enabled, and I have a scheduled job with a CRON expression "0 \* \* \* \*?", then the below value expression on the scheduled job will update content BL0001's custom field every minute.

```
content = daoHelper.fetch("Content",
    SlimsRestrictions.equals("cntn_barCode", "BL0001"))
triggerRecalculation("Content", content[0].cntn_pk, false);
```

- *Active*: During configuration, if the scheduled job is active and is saved, SLIMS checks to ensure the CRON expression is valid, and that the value expression is syntactically correct. If

either or both are invalid, the scheduled job cannot be saved. However, inactive jobs can be saved even if they are incorrect (including empty values), so that they can still be worked on. As best practice, scheduled jobs are best left inactive until the configuration is finished and checked for errors.

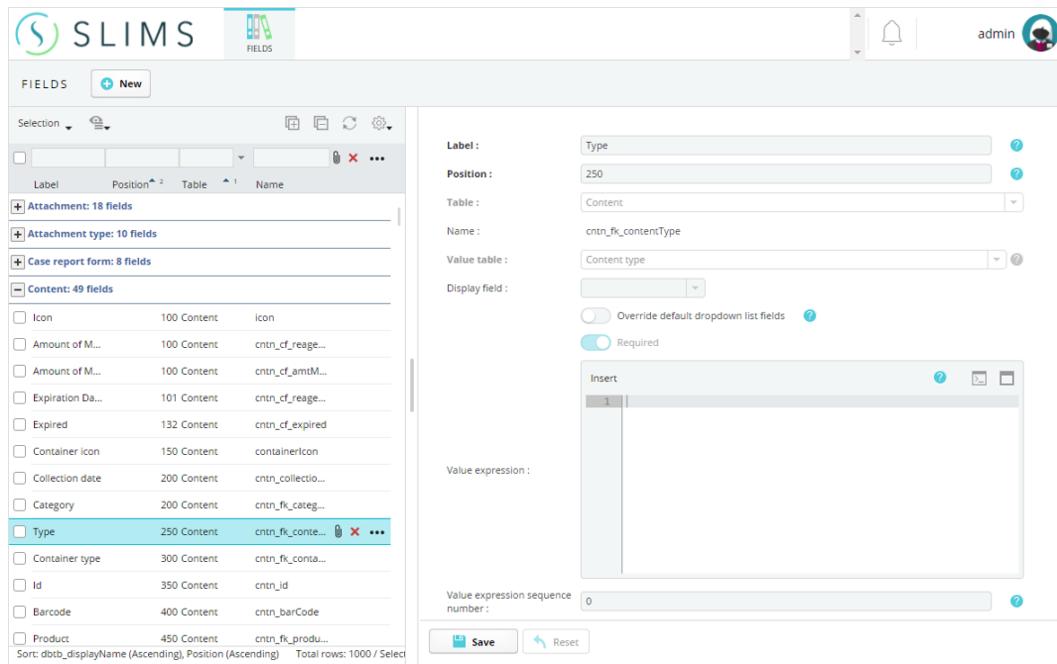
## 10.5. Fields

Each table (Location, Content, Test, etc.) is defined by several fields: Name, Description, and so on. Fields depend on the specific table. For example, the definition of a Content includes a Quantity Field, and the definition of a Location includes a Path Field. Default fields are allocated automatically by SLIMS. Aside from the standard fields, the Fields feature in SLIMS allows default fields to be modified, and custom fields to be added.

The Fields module can be accessed from the Setup menu and is used to customize existing

fields or add new ones .

**Figure 10.2. Fields Module**



The left window shows a collapsible list of fields from which one can be selected at a time. The right window shows the details and descriptions of the selected field.

The left window shows a collapsible list of fields from which one can be selected at a time. The right window shows the details and descriptions of the selected field.

### 10.5.1. Actions Available for Fields

#### 10.5.1.1. Copying Fields

It is possible to copy a custom field by accessing the "Copy" action from the right-click menu. A pop-up window opens where the user can specify the name and label for the custom field copy.

### 10.5.1.2. Import and Export Fields

Fields provide microscopic control over every field on a form, but if a lot of them need to be changed at once, such as when migrating from a different LIMS to SLIMS, importing can help speed the process. As in other areas of SLIMS, the gear icon provides the Import and Create Template File actions. The Export action is not available in the Selection dropdown menu until one or more fields are selected using the multi-selection boxes.

Only custom fields are eligible for exporting/importing. The default SLIMS fields cannot be exported or imported, but you can import previously exported custom fields or custom fields that were created with the template. Flat fields will be exported as regular fields, and they cannot be imported. They can be imported as standard fields and made flat after the fact. Additionally, it is not possible to export the entire Fields grid.

There are two other restrictions to understand for field imports and exports:

- Field restrictions (only for selected Content Categories/Content Types and only selected roles can access/update) are recognized by their name, so they will need to be imported using cntp\_name/role\_name. To restrict to multiple records at once, separate the record names with commas. "Dropdown list fields" and "Expression depends on" use a special field key as an identifier composed of the table name, a dot, and the field name (for example: Content.icon and Content.cntn\_id).
- The records referred to in the dropdowns need to exist in SLIMS already. Also, newly imported custom fields cannot depend on other fields that are imported in the same file.

Additionally, if the user has difficulty finding an accepted value for a field, he/she can find the expected value for a field by exporting a new or existing field with that value in the Fields module.

### 10.5.1.3. Field Removal Options

Fields can be inactivated to hide them from the Fields module and from the SLIMS user interface, or Permanently Removed to delete them from SLIMS entirely. However, this is only possible in some circumstances.

The "Active" property on a field can be switched off to deactivate the field. This is useful to hide fields that are no longer being used while still retaining the field history for audits and reporting. Fields that are defined as an amount of material field or an expiration date field cannot be deactivated. Deactivated fields can still be sorted on and filtered on. Deactivated fields:

- Are no longer visible in the SLIMS user interface,
- Are no longer visible in the SLIMS REST API,
- Are no longer validated (for example, the field is no longer required, SLIMS does not perform status restrictions from status workflows, the field restrictions on types are not in effect, etc.),
- No longer has its Groovy executed if Groovy expressions were configured,
- And values for the field are no longer updated via the API if this was configured.

The field can only be permanently removed in cases when the field was never used, meaning it does not have data values entered and never has had a value entered. Fields can be per-

manently removed using the right-click context menu or with the  icon in the list of fields. Permanent removal is useful when a field is created by accident or with mistakes. Many of a field's parameters cannot be edited once it is created, so the field can be deleted and re-added with the corrections.

## 10.5.2. Default Fields

The default Fields below are generated automatically by SLIMS:

- **Icon:** A symbol used as a quick and intuitive graphical representation of the component.
- **Name:** The full name of the Component. In some special cases, like creating new Custom Fields, the name field can be used for programming purposes and is subject to restrictions, such as not containing spaces. In these cases, a label field is used to name the Component, so the label will be displayed on the user's screen instead of the name.
- **Label:** The name describing the Component as shown on the screen.
- **Name mask (use # for numbers):** The root name generated when using the feature 'Add multiple'. The different new Components will contain a root name and an incremented number. By using # characters you specify how many digits the incremented number should contain. Example: The Content Type "Mouse" is created and the user wants to add 15 mice. If the Name mask (use # for numbers) for Mouse has been defined "Mouse###," 15 mice will be added and automatically named Mouse001 through Mouse015 when they are added. Although the names are set by default according to these settings, the name can be changed afterwards.
- **Barcode:** The barcode can be generated either by SLIMS or by the User and it is the unique identifier of a Component. When SLIMS generates the barcode it searches in the existing barcodes and checks those with the same mask name (the root), finds their highest barcode number and establish a higher number for the new component (does not fill gaps).
- **Address:** The record's location, street name, etc.
- **Description:** Any desired text that is needed to specify further details or describe the component.

## 10.5.3. Custom Fields

Custom fields provide options beyond default fields. They can be configured with the following parameters:

- **Name:** The name of a Custom Field has a particular format in SLIMS because it can be invoked with SLIMS' scripts. To create a name, the user should follow the set of restrictions below. The name starts with the following abbreviations separated by underscores ( \_ ).
1. This list shows the abbreviations of the tables where a Field is contained.

**Table 10.7. Abbreviations of SLIMS Tables**

Table	Abbreviation
Attachment	atm
Case Report Form	crfr

Table	Abbreviation
Case Report Form Template	crft
Content	cntn
Content Event	cnnv
Content Event Type	cnvt
Content Study	cnst
Content Type	cntp
Customer	cstm
Disease	diss
Experiment	xprm
Group	grps
Grid Template	grtm
Instrument	nstr
Instrument Run	nsrn
Location	lctn
Location Type	lctp
Macro	scnr
Order	ordr
Order Content	rdcn
Order Type	rdtp
Plugins	slpg
Printer Job	prjo
Product	prdc
Project	prjc
Protocol	xptm
Protocol Run	xprn
Protocol Run Step	xprs
Protocol Run Step Content	xrsc
Protocol Step Test	xsts
Protocol Step	xpst
Provider	prvd
QC Card	qccr
QC Template	qctm
Queue	queu
Reference Data Record	rdrc
Reference Data Type	rdty
Report Template	rptm
Request	rqst
Requestable	rqbl

Table	Abbreviation
Result	rslt
Rule	rule
SOP	stop
Source	sorc
Stability Study	stsd
Stability Study Storage	sssr
Stability Study Template	stst
Stability Study Template Storage Conditions	ssts
Stability Study Template Timepoints	ssst
Status Workflow	stwf
Study	stud
Test	test
User	user
Test Group	tsgr
Variant	vrnt
Worklist	wrkl
Workflow	wrfl

2. cf initials have to be added in the created Custom Field name in order to distinguish it from a default Field.
3. If the Field will contain a dynamic choice as *data type* (explained later), the fk initials must be used (foreign key).
4. Finally, a name describing the Field should be added (e.g. quantity, unit, Father).

Examples:

- › A Custom Field with Father as dynamic choice (explained later) in the Content table (cntn) would be named: cntn\_cf\_fk\_Father
- › A Custom Field with gender as a choice in the Content table would be named: cntn\_cf\_sex
- **Label and Description** Labels take the place of names for Fields. The Label is the text that will be shown inside SLIMS to indicate what the field is about, since the field name is rather programmatic.
- **Tooltip:** The written text will be displayed in a small box as an explanation when you roll over the field with the mouse cursor.
- **Help:** Specify a help text for this Custom Field. If the text is small (1 line), it will appear as a hint inside or next to this field in corresponding forms. Bigger texts will appear in a popup window when one clicks the help icon next to this field.
- **Position:** Specify the position of the Field inside the table. Fields are shown in ascending order down the page, so fields with lower numbers are toward the top, and fields with higher

numbers are toward the bottom of a form. With this method related fields can be grouped nearer to each other.

- **Table:** Specify the table where the Field would be available.
- **Flat:** When enabled, this option marks the field to be transformed into a flat field on the next SLIMS service restart. A warning will be displayed on saving a field that has just been made flat that the option will not take effect until the next SLIMS restart. Once the field has become flattened, the option becomes read-only and cannot be undone.

Flat fields are beneficial in cases where there are so many records for SLIMS to fetch and update that they begin to impact performance. Standard custom fields that are created on a given table are stored in a separate table of the SLIMS database, as compared to default fields for the same table, which are actually columns in the database. This gradually impacts performance as the SLIMS database grows bigger, as the query to look up a custom field in the database grows more complex to include the separate table where custom fields values are stored. Using the Flat option for a custom field turns the field into a new table column instead of becoming a new entry of the separate table. This makes querying the content of this field much faster, because it functions similarly to default fields.

Flat Fields have some restrictions. They are incompatible with "Can be NA," "External," "Multiple" on dynamic choice fields, and they cannot be used with bulk propagation. If any of those options are enabled, the Flat option on the field will be grayed out with an explanation. If Flat is enabled, NA, External, and Multiple will be hidden. The datatype Ontology Term is not supported and will not be shown for flat fields. Searching for flat fields in the Global Search field in the Content module is not supported. Finally, fields will be imported/exported as non-flat. Excel and SLIMS Share are not able to recognize flat fields, and thus will just import/export them as regular fields.

- **Datatype:** Choose the data type of this Field from the choices in the list. Further details on the different data types and corresponding parameters are given in the Datatypes Section.
- **Multiple values:** Multiple fixed choice or multiple dynamic choice values can be entered by checking this box.

In Excel import/export, the list of values is exported as a comma-separated list. For multiple fixed choices, the list of choices will be sorted alphabetically regardless of how they are specified in the field definition.

Multiple fixed choices can be entered in grids and in forms. Filtering in grids can be done on one value. If users want to query for combinations of options they can use the filter builder.

Multiple dynamic choices can only be entered in forms. Filtering and editing is not supported in grids.

- **Can be NA:** Allows custom fields to be set to an NA (Not Available, Not Applicable) value.

If this option is checked, the field will be shown with an NA circle next to it. To save the value as NA, the user clicks on the NA circle instead of entering a value for the field. To remove the NA value, the user clicks on the NA circle to clear it and then a value can be entered. An NA value is a filled response, so it is considered a valid response to a required field. △ The NA option is not available for fields of type Rich Text.

- **Value Expression:** Specify an expression that will be evaluated as the value of this field. Expressions should be written in the 'Groovy' language. Value Expressions are used in fields to do set tasks like calculating the mass value from the volume and concentration values.

① Note that the Value Expression contains a method where the returned value (using `return returnValue;`) will become the new value of the field where the value expression is evaluated.

Please refer to the '*Groovy Scripting*' chapter for details with examples on how to write Groovy scripts and explanations of the built-in SLIMS methods and variables.

The user can choose when a value expression is evaluated during setup of the expression:

- **Before creation:** Using this option, the value expression is evaluated only once; during the creation of the record.
- **On every save:** The value expression will be evaluated whenever the record is edited and changes get saved.

△ The evaluation of the value expression is triggered only for the record that is saved. Therefore, if a value expression depends on a second record and this one is updated, then the value expression will not be re-evaluated unless the initial record that has the value expression specified is updated.

Users can also specify fields on which the expression depends, so that SLIMS will Evaluate the expressions while editing one of these fields.

① Value expressions with evaluation "**Before creation**" are calculated when the creation form is closed. This means the calculated expression is not shown in the creation form. This behavior can be modified by specifying a field on which the expression depends. In this case, the calculated value will be shown (and calculated run time) if the dependent field is edited.

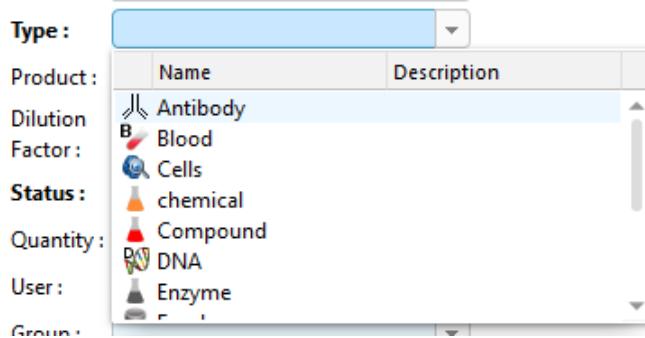
- **Value Expression Sequence Number:** The number entered here determines which order value expressions are evaluated in. The lowest number in the list of fields gets evaluated first, and the highest is evaluated last. If two fields have the same sequence number, the alphabetically lower name gets executed first. When a value expression is present for a field, the value expression sequence number field becomes available and is required.

For example, if you have the fields blood volume, gender, and study, and the blood volume collected depends on the gender of the patient and the study they are in, then the study and gender would need to be evaluated before blood volume. The value expression sequence numbers could be set to something like blood volume = 300, gender = 200, and study = 100. If you accidentally had something like blood volume = 200, gender = 200, the blood volume would be evaluated first (alphabetically) and the expression would not work as intended.

- **Read only:** The value appears in the form but cannot be edited.
- **Required:** The component creation will not be permitted without entering a value in this Field.
- **Hidden:** Not shown to the user, neither in the editing or detailed view.
- **Show in list view:** This Field would be by default present in a column of the default view of the table list. For instance, we can include the field quantity to be present by default in the Content List View.

- **Show in Dropdowns:** If the chosen table of this Field is used in another table then a column for this Field would be shown when selecting a Field pointing to its table. (For example: Content Type table is used as a Field in the Content tab.) When selecting the Content Type in a Dropdown List, there are columns for the icon, name, and description because these three fields have 'Show in selected items' checked (Image below).

**Figure 10.3. Example of 'Show in dropdown lists'**



① It is possible to override the default Dropdown List fields of a dynamic choice field with a different set of fields when defining the field.

- **Show in extended tooltips:** The use is the same as with 'Show in dropdown lists' but the Field will be shown in a row of the tooltip instead of the selected item tab. For example, the tooltip in the image below shows the field's Name, Description, Default Quantity, and Default Unit as the user hovers over a blood sample in the Content Type column of the Content grid.

**Figure 10.4. Example of 'Show in extended tooltips'**

Collection date	Type	Product	Status
<input type="checkbox"/> 08/02/2018 10:43	Blood		Available
<input type="checkbox"/> 08/02/2018 10:43	Blood		Available
<input type="checkbox"/> 08/02/2018 10:43	Blood		Available
<input type="checkbox"/> 08/02/2018 10:43	Blood		Available
<input type="checkbox"/> 08/02/2018 10:43	Blood		Available
<input type="checkbox"/> 08/02/2018 10:43	Blood		Available

- **Show extended tooltip to referenced record:** This parameter is available when choosing the data type 'Dynamic choice.' Enabling this parameter will show the extended tooltip for this field as explained with *Show in extended tooltips*.
- **Show in simplified ELN** Show the field in simplified ELN.
- **Propagate changes to derivations** All the derivations of contents that have this field will also contain this field. The value shown is that of the original content for all the derivations, but the value is read only unless the Allow update in derivations is checked.
- **Use bulk propagation** This parameter is available when **Propagate changes to derivations** is enabled. If checked, then the fast propagation algorithm is used, which means that a change on this field will be propagated to derivations of the current entity but will not trigger recalculation of the groovy expressions for the derivations, so **no** groovy expression should depend on a field with "Use bulk propagation." An error will be displayed in the logging if a groovy ex-

pression contains a reference to a custom field with this option enabled (even if it is only a `log.info()`).

⚠ In the case that multiple fields are updated, the default propagation algorithm is used for all of the updated fields even if some fields have bulk propagation enabled.

- **Allow update in derivations** If this is enabled, the user can modify the custom field value of a derived content item, and the modified value will be propagated to other content items in the derivation tree. This option is only available if the option *Use bulk propagation* is enabled.
- **Only show if other field is checked:** If this option is enabled, the field will only be shown if another field on the same record is checked. The other field data type should be a checkbox. E.g. If the field "smoker" is checked, then the current field "number of cigarettes per day" will be present. But if "smoker" is unchecked, the "number of cigarettes per day" field is not present and its value will be removed if it was previously set.
- **Only selected roles can access:** The field can be accessed and viewed by selected roles, and is hidden to roles that are not selected. There is an important technical difference in the implementation of this feature between default fields and custom fields. Making a default field hidden for certain roles means that these roles cannot see it in the SLIMS UI. To ensure that SLIMS behaves correctly, the values of those fields will still be sent from the server to the client, even if the user does not have access to them. This behavior is different for custom fields. If a user does not have access to a certain custom field, SLIMS will not send the values from the server to the client.
- **Only selected roles can update:** The field can be updated and edited by only the selected roles.
- **Only for Selected Content Categories:** Available only with Contents. It works with or without "Only for Selected Content Types" to provide different results. Only for Selected Content Categories filters the Content Types option if a category is selected, and the additional Apply to All Content Types in this Category will apply the field to all of the Content Types in the category. This figure shows the possible overlaps and usages of these options.

**Figure 10.5. Selected Categories VS Selected Types**

Selected Options	Which content types are included
<input type="radio"/> Only for selected Categories <input type="radio"/> Only for selected Content Types	The field applies to all content types.
<input checked="" type="radio"/> Only for selected Categories <input checked="" type="checkbox"/> Apply to all content types in this category <input type="radio"/> Only for selected Content Types	The field applies to all content types within the category and "Only for Selected Content Types" is not used.
<input checked="" type="radio"/> Only for selected Categories <input type="checkbox"/> Apply to all content types in this category <input type="radio"/> Only for selected Content Types	The field does not apply to any content types because a category is expected but has not been selected.
<input checked="" type="radio"/> Only for selected Categories <input type="checkbox"/> Apply to all content types -- Category 1 <input checked="" type="checkbox"/> Apply to all content types -- Category 2 <input checked="" type="radio"/> Only for selected Content Types + Content Type from Category 1	The field applies to all content types within category 2, and the added content type from Category 1. The other content types from Category 1 are available for selection under "Only for selected Content Types."
<input type="radio"/> Only for selected Categories <input checked="" type="radio"/> Only for selected Content Types + Content Type from Category 1 + Content Type from Category 2	The field applies to the two selected content types, and all other content types are available for selection under "Only for selected Content Types."
<input type="radio"/> Only for selected Categories <input checked="" type="checkbox"/> Apply to all content types -- Category 1 <input checked="" type="radio"/> Only for selected Content Types	The field applies to all content types within Category 1 only, and no additional content types are available for selection.
<input type="radio"/> Only for selected Categories <input type="checkbox"/> Apply to all content types -- Category 1 <input type="radio"/> Only for selected types	The field applies to no content types at all, and only the content types from Category 1 are available for selection in the Content Types dropdown.

- **Only for Selected Type:** Available only with Contents, Locations, Reference Data Types, and Orders tables. It is recommended to use this option often to simplify the form that describes a record because it narrows the fields to only show up for the pertinent Content, Location, Reference Data Types, and Order types that are selected. To use this option, check the "Only for Selected Type" box and then select the types that need the custom field.

When the data type is set to Quantity, the Unit, Lower Unit, and Upper Unit to be used for each type can be specified. Only units that have the same dimension (Lower and Upper Unit) as the custom field dimension are available. An example concerning custom fields and units definition is given in the unit section Section 10.23.1.

- ① When a unit of a particular type is changed, only the new entered quantity values will get this modified unit. Quantity records entered before the change will keep their original unit and will not be affected with this modification.
- **External:** Enables live fetching. This feature allows a custom field value to be fetched from a table that is not contained in SLIMS. This is only available for custom fields on the table Content with simple data types such as short text, text boxes, date and time, checkboxes, and numbers. Once checked, the user must provide the table and key column used to identify the record content in the remote database. The key should refer to a column with a UNIQUE constraint. The field can only be filled with SLIMSGATE or directly in the database by filling the field cntn\_externalId from the content table with the key column value from the remote database. This is used to identify the record to fetch in the remote database.

For Example, given a MySQL database "livefetchdb" with a table "remote" and two columns ("rmte\_pk", "rmte\_value") the following fields need to be entered in SLIMS:

- *External table:* livefetchdb.remote
- *External key column:* rmte\_pk
- *External value column:* rmte\_value

And the following statement needs to be executed:

```
UPDATE Content SET cntn_externalId=1 WHERE cntn_pk=1;
```

- **Show in Portal:** This option is only used in combination with the SLIMS Requests Add-On and determines whether a field will show up in the SLIMS Requests Add-On.  
△ The DataType, the unit dimension (for quantities), and the Table choice cannot be modified after the Custom Field is created. All other parameters can be edited after the custom field creation.

## 10.5.4. Rich Text Fields

Fields of type "Rich Text" contain a rich text editor allowing the use of advanced tools similar to a word processor. Refer to the section "*Rich Text Editor*" for detailed instruction and descriptions.

Usage of rich text fields is intuitive and allows the user to write and edit text with fonts, tables, and colors similar to using a favorite text editor (Notepad, Microsoft Word, etc.). A tool bar is available on top of the field to access all of the features available in the rich text editor.

Images can be dragged and dropped into the field to insert them in the text area (See *drag and drop in the Attachments* section for further details) and they will be displayed with the text. Figure dimensions and other properties can be modified using one of the rich text editor tools available under the icon "image." ⓘ Inserted images are turned into attachments so they can always be accessed through the attachment window of the record that contains the field.

ⓘ Rich text fields with the inserted tables and images can be printed using the "print" button available in the tool bar on top.

## 10.5.5. Datatypes

SLIMS contains data types that can be assigned to field values or results. The following is the list and definition of existing data types in SLIMS.

### 10.5.5.1. Short text

A text field shorter than one line which may contain numbers, letters, or both. For example, a name or an address.

### 10.5.5.2. Status

The Status datatype is used to force a record (content type, instrument type, order type, test, etc.) to follow a particular status workflow and provide a dropdown of the status workflow's available statuses in the record's form. Multiple Status datatype fields can be created on the same table that follow different status workflows in order to support record statuses in different contexts.

When a status datatype custom field is created, a status workflow must be selected for the field. Once the field is created and saved, the selected status workflow cannot be changed. (Because of that, status datatype fields make good flat fields.)

When a record with a status field is created, the user can choose any of the active statuses that the record can currently transition to in the status field. Records can have more than one status datatype field specified, meaning they can also utilize more than one status workflow.

### 10.5.5.3. Text box

A text field longer than one line which may contain numbers, letters, or both. For example, a comments field that may take up several lines.

### 10.5.5.4. Rich Text

Allows the user to input text using a tool similar to a word processor. The height of this field can be specified. Further data type description is available in the "Rich Text Field" section.

### 10.5.5.5. Whole number

An integer number. A lower and upper limit can be specified which will limit the entered value to one that fulfills the lower and/or upper constraint.

### 10.5.5.6. Decimal number

A decimal number. A lower and upper limit can be specified to limit the entered value to one that fulfills the lower and/or upper constraint.

### 10.5.5.7. Date

A date displayed according to the format defined in *Lab setting*. The default format is dd/mm/yyyy. A calendar icon is available next to the field so the user can select a date. The calendar allows choosing between 15 years more or less than the current year, but any year can be entered by typing the value manually into the date field. Date fields are not sensitive to the local timezone that a user has set and will display the exact data that was entered in the field. If a date field needs to display in a given user's local timezone, a Date and time field can be used instead. (When the data is exported, it displays in the user's timezone in the exported file.)

### 10.5.5.8. Time

A time entered in format HH:MM. The time that is entered does not display differently depending on the user's timezone. (REST API users will notice such times are internally stored in the UTC timezone.) If a time field needs to display in a given user's local timezone, a Date and time field can be used instead. (When the data is exported, it displays in the user's local timezone in the exported file.)

### 10.5.5.9. Date and time

A date and time displayed according to the format defined in *Lab setting*. The default format is dd/mm/yyyy hh/mm. A calendar icon is available next to the field so the user can select a date. The time is 00:00 by default and has to be entered manually. Date and time fields are sensitive to a user's timezone and will display in a given user's local timezone when they view data entered in the field.

### 10.5.5.10. Date Difference

A *read only* field that calculates the difference between two Date fields. This data type only works if the following are defined:

- **Date Difference Field 1**, fixed choice between fields of type Date or Date and time of the same table and "Current time"
- **Date Difference Field 2**, fixed choice between fields of type Date or Date and time of the same table and "Current time"
- **Date Difference Unit**, fixed choice between Second, Minute, Hour, Day, Week, Month, Year.

The difference is calculated as

#### Date Difference Field 2 - Date Difference Field 1

In the unit defined in **Date Difference Unit**.

⚠ The unit is not automatically displayed, but can be added in the Label. For example: "Age (years)," or as a Tooltip.

A lower and upper limit can be specified for this field so that only values fulfilling the limits will be displayed.

### 10.5.5.11. Checkbox

Data type that can have one of two values: true or false (checked = true, unchecked = false). A box will be available for this data type that gets checked or unchecked by clicking on the box or the field label.

Checkbox fields cannot be required (the "required" parameter is not available for them) because checkboxes have two states: checked or unchecked. This is to prevent confusing configuration because either checked or unchecked responses are valid responses for checkbox fields, and to require the field would mean only the checked response is valid.

### 10.5.5.12. Fixed choice

Provides a list of choices for the user to select a value from. If this data type is selected, an additional text field will appear where the possible field choices can be entered, each value on a separate line.

Example with the defined Fixed Choices for a Custom Field Gender:

Male

Female

### Unknown

The option Strict is available: Select this if you want the value to be limited to the above list of 'Possible values'. If not selected, the possible values will be suggested but any other value will also be accepted.

If Strict is selected, then 'Display mode' will be available with the following options:

- **Selection list:** The possible values can be chosen from a dropdown list.
- **Group of horizontal radio buttons:** The possible values are visualized as horizontal radio buttons.
- **Group of vertical radio buttons:** The possible values are visualized as vertical radio buttons.

Radio choices are considered strict and do not allow Multiple Values. This means that Multiple Values will only be available for Selection Lists. Also, the radiolist options are not present unless the fixed choice custom field is changed to Strict.

ⓘ If zoom is applied in the browser, the dropdown list may not show the last entry. This is fixed by resetting the zoom to its default level.

### 10.5.5.13. Dynamic choice

The Dynamic choice field points to a list of values from another table and inserts them into the desired component's field. For example, a Dynamic choice field "Source" can be created to a reference data type containing the records "arm," "leg," and "head" which can be displayed in the content creation form. The value of the Dynamic choice field is chosen from an existing table.

These options exist for Dynamic choice fields:

- **Value Table:** Specify the table that the list values come from. (Ex: Reference Data Record) A list of possible tables can be found below, after the filters explanation.
- **Reference Data Type:** Only for Dynamic choice fields to the value table "Reference Data Records." A required field that restricts the reference data records that will appear in the dropdown to a specific reference data type. (Ex: Reference Data Type "Sample Collection")
- **Display Field:** The column that the list of options in the resulting dropdown will use as a display field. (Ex: Name)

ⓘ It is important to select a display field that makes sense for the items that will be listed in the dropdown. If a field like the name is hidden due to type restrictions, another field should be selected as the display field to prevent the list from being empty in the related form.

Additional columns that make sense for the type of field may be included by design. For example, stability study entities used as a value table have secondary display columns by design to help show which stability study they relate to.

- **Sorting Field:** The column that the list of options in the resulting dropdown will be sorted on. (Ex: Creation Date)

ⓘ It is important to select a sorting field that makes sense for the items that will be listed in the dropdown. If a field like the name is hidden due to type restrictions, another field should be selected as the sorting field to make sure it's applicable to the items in the list.

- **Sort Direction:** Appears when there is a Sorting Field selected. The sort direction can be ascending or descending, which is useful if the sorting field is a date or a number.
- **Filter the Value Table:** Allows for more complex filtering on the output list; detailed below.

A filter can be applied to the selected table using the action  **Filter the value table**. As an example, a dynamic choice can be added for Patients with a value 'father' where the father from the existing content table can be chosen, with the content type restricted to Patient and the sex being male. This filters the options the user can choose from so only male patients are available to select from, reducing human error and narrowing a potentially long list into one that suits the field.

A dynamic filter can be written in the "Dynamic Filter Expression." For example, to restrict the available contents that can be selected in a dynamic choice field, a dynamic filter expression can be used to filter on the barcode of the contents. The below expression, for example, restricts the choices in the field to content with barcodes BLD00000132 and BLD00000130.

```
criteria = [
  operator: "or",
  criteria:[
    [
      fieldName: "cntn_barCode",
      operator: "equals",
      value: "BLD00000132"
    ],
    [
      fieldName: "cntn_barCode",
      operator: "equals",
      value: "BLD00000130"
    ]
  ]
]
return criteria;
```

Operators that can be used in the dynamic filter expression work in the same way as the keywords found in the choice menu of "Filter custom fields". The table below shows how the operators correlate with these keywords.

**Table 10.8. Correlation between dynamic filter expression operators and filter keywords**

Filter keyword	Expression operator	Comments
Match All	and	
Match Any	or	
Match None	not	
equals (match case)	equals	
not equal (match case)	notEqual	
greater than	greaterThan	
less than	lessThan	
greater than or equal to	greaterOrEqual	
less than or equal to	lessOrEqual	

Filter keyword	Expression operator	Comments
between (match case)	between	instead of <b>value:</b> , <b>start:</b> and <b>end:</b> are used
between (inclusive, match case)	betweenInclusive	instead of <b>value:</b> , <b>start:</b> and <b>end:</b> are used
is null	isNull	<b>value:</b> is not used
is not null	notNull	<b>value:</b> is not used
contains (match case)	contains	
starts with (match case)	startsWith	
ends with (match case)	endsWith	
equals	iEquals	
not equal	iNotEqual	
contains	iContains	
starts with	iStartsWith	
ends with	iEndsWith	
does not contain (match case)	notContains	
does not start with (match case)	notStartsWith	
does not end with (match case)	notEndsWith	
does not contain	iNotContains	
does not start with	iNotStartsWith	
does not end with	iNotEndsWith	
is one of	inSet	<b>value:</b> expects a list of values
is not one of	notInSet	<b>value:</b> expects a list of values

Dynamic Filter Expressions are being added to the available setup for default fields, starting with cntn\_fk\_product. While this is the only default field to currently allow filter expressions, more can be enabled on request by using the Allow Dynamic Filters attribute in the right ds.xml file. However, this can only be done on certain fields without foreign keys or layers of logic (E.g. will not work on content type because of the layers of reference).

① If zoom is applied in the browser, the dropdown list may not show the last entry. This is fixed by resetting the zoom to its default level.

**Table 10.9. Possible Value Tables for Dynamic Choice Fields**

Available Value Tables
Analysis
Analysis Template
Case Report Form
Case Report Form Template
Content

**Available Value Tables**

Content Event
Content Event Type
Content Study
Content Type
DNA Index
DNA Index Location
DNA Index Set
Email Template
Experiment
Grid Template
Group
Instrument
Label Template
Location
Location Type
Meta Tree Column
Order
Order Type
Plate Index Design
Plugins
Printer
Product
Project
Protocol
Protocol Run
Protocol Run Step
Protocol Step
Queue
Reference Data Record
Reference Data Type
Requestable
Result
Sequencing Settings
Stability Study
Stability Study Storage
Stability Study Template
Stability Study Template Storage Conditions
Stability Study Template Timepoints

### Available Value Tables

Stability Study Timepoint

SOP

Status Workflow

Study

Test

Test Group

User

Workflow

## 10.5.5.14. URL

The custom field of type URL allows a hyperlink to be defined for the following:

- *http*: A link to the specified internet URL will be created.
- *mailto*: Contains a link to send an email to the specified address.
- *file*: A link to a file with the specified URL will be created.

⚠ The file link is blocked by all browsers for security reasons. As a solution, the address can be copied using right-click and then pasted in the browser address bar. In addition, some add-ons can be installed in the browser that allow file links to be opened.

## 10.5.5.15. Quantity

Quantities are numeric values associated to a unit that are variable but within the same dimension. The unit dimension should be predefined and cannot be changed after the creation of the field.

When the Quantity data type is selected, the Dimension and the Unit (the default unit, if one exists) can be selected. A quantity field with a predefined unit can be set by leaving "Allow Choice of Unit" unchecked, or the user can choose a unit when filling the form if "Allow Choice of Unit" is checked. A user can also choose the highest and lowest unit that is allowed for that field.

Groovy value expressions can be written to return a quantity and a unit. A relevant value expression is described here: *Section 11.3.8 Quantity and Unit*. They can even be written when using "Allow choice of unit." SLIMS performs conversions to respect the selected unit choice on quantity fields with value expressions. The unit entered by the user will be kept if it is in the right dimension and subdimension and is between the highest and lowest units allowed for the field. If the unit is:

- *In the wrong dimension*: An error is logged in the groovy console and the value is kept null.
- *In the wrong subdimension*: The value is converted to the default unit of the field. For example, a length dimension has the metric subdimension with the unit centimeter, but length also has the unit inch. A quantity field allowing choice of unit specifies that the subdimension should be metric, and the default unit is centimeters. When the value expression on the field returns 2 inches, SLIMS converts the value to 5.08 centimeters when the groovy executes on a record.

- *Outside the allowed upper/lower units:* The value is converted to the closest boundary unit. For example, given the length dimension has a default unit of centimeters and the following cases:
  - *Is higher than highest allowed unit:* The field's upper unit is centimeters and the groovy returns 5 meters, the value is converted to 500 centimeters when the value expression executes on a record.
  - *Is lower than lowest allowed unit:* The lower unit is millimeters and the groovy returns 5 micrometers, the value is converted to 0.005 millimeters when the value expression executes on a record.
- *Not defined in SLIMS:* If it's a prefix unit, the value is converted to the default unit of the field. Otherwise, an error is logged in the groovy console and the value is kept null. For example, the length dimension has a default unit of centimeters, but a field that allows choice of unit has a value expression that returns a value with an undefined unit. If it's a prefix unit, SLIMS converts the value to centimeters when the groovy executes on a record.

Using a highest and lowest possible unit implies that 'highest' and 'lowest' are meaningful so that units have a meaningful order. Unit ordering is sometimes arbitrary and not meaningful in relation to each other. For example, clearly  $g < kg$  and  $l > ml$ , but it makes less sense to say that  $^{\circ}C > ^{\circ}K$  or  $^{\circ}C < ^{\circ}K$ . Thus, sometimes it makes sense to set an upper and lower unit, and sometimes it does not.

The default unit and the limits can also be configured per restricted type. For example, a user can change the default quantity, unit, and unit limits for a certain content type.

A unit within a dimension can also be assigned a Sub Dimension. The Sub Dimension is merely symbolic, but is used as a way to distinguish units within the same dimension. For example, the units  $dm^3$  and  $l$  are in the same dimension, but it is not common to have a situation where users need to choose between the two, as they would do everything in liter or in  $m^3$ . A Field can consequently also be assigned a Sub Dimension to add this context.

The following options are available with a Quantity data type:

- *Sub Dimension:* The sub dimension from within the choice of units is restricted.
- *Allow Choice of Unit:* The user will be able to select the unit when filling the quantity value at runtime if this option is enabled.
- *Unit:* The default unit for this field and a selection between the units contained in the same field dimension.
- *Lower Unit:* Lowest unit that can be selected.
- *Upper Unit:* Highest unit that can be selected.
- *Value Expression:* There are some functions particularly useful for creating value expressions for quantity and unit functions. The functions `quantity(amount, unit)` and `unit(abbreviation)` are available in the Insert dropdown menu in the value expression editor, and can be used to create, return, or calculate with quantities or fetch a unit. See some examples in the Groovy Scripting sections: `quantity(amount, unit)` and `unit(abbreviation)`.
- *Minimal Decimals:* Restricts the decimal value to prevent users from entering fewer decimal places than the minimum decimal place. For example, if Minimal Decimals = 2, then 2 places are expected after the decimal.

- 2.30 will be accepted
  - 2.3 will be refused
  - 2.34e1 will be refused
- *Maximum Decimals*: Decimal places exceeding the set amount will be rounded after the maximum decimal place. For example, if Maximum Decimals = 4, then decimal places after the 4th will be rounded and cut off to restrict the number length.
  - 2.34345 → 2.3435
  - 2.34345e1 → 23.4345 (not rounded)
- *Minimum Significant Digits*: Prevents users from entering values with fewer significant digits than this. For example, if Minimum Significant Digits = 2
  - 2 won't be accepted (write 2.0)
  - 0.0002 won't be accepted (write 0.00020)
  - 2e1 won't be accepted (write 2.0e1)
  - 20 will be accepted
- *Maximum Significant Digits*: Digits exceeding the set significant digit amount will be rounded at the specified digit. For example, if Maximum significant digits = 4
  - 2.0005 → 2.001
  - 20000 → 20.00e3
  - 2000000 → 2.000e6 (note the convention of writing only powers multiple of 3)
- *Apply Logarithm to Maximum Significant Digits*: Enables smaller values to have fewer significant digits (this only applies on values greater than one). If you enable this option, the actual maximum number of significant digits will be  $\min(\text{ceil}(\log_{10}(\text{value})) + 1, \text{maximum significant digits})$ . For example, if maximum significant digits = 3
  - 0.09 → 0.09 (rule not applicable)
  - 0.20 → 0.20 (rule not applicable)
  - 3.71 → 3.7 (because  $\text{ceil}(\log(3.7)) + 1 = 2$  so max significant digits = 2)
  - 10.3 → 10.3 (because  $\text{ceil}(\log(10.3)) + 1 = 3$  so maximum significant digits = 3)
  - 1122 → 1.12e3 ( $\text{ceil}(\log(1122)) + 1 = 4$  but maximum significant digits = 3)
- *Enable Scientific Notation*: SLIMS automatically writes quantities in a scientific notation, so this option is enabled by default. This option has to be disabled if users do not want to have the quantity shown in scientific notations in their forms.

### 10.5.5.16. Comparator Quantity

This is another quantity data type that provides a different context, and allows values like the following to be specified:

< 5.67

> 4.7

The same options as Quantity are available for this data type.

### 10.5.5.17. Barcode

Allows a barcode to be captured (essentially a string of characters), but also detects uniqueness so that duplicate values cannot be entered.

### 10.5.5.18. Attachment

This provides a fixed choice between any existing attachments so they can be linked to the record's details. When editing this field it is also possible to add a new attachment.

### 10.5.5.19. Ontology Term

This type is only available if the Ontology Graph module is enabled in the role functionality settings.

Allows users to add an Ontology Term from a single target graph. A single target graph must be selected so SLIMS knows what graph the term relates to. The following options are available with a field of data type Ontology Term:

- *Display Field*: 'Label' is the default.
- *Multiple Values*: Allows multiple Ontology terms to be entered as values.
- *Fixed Ontology Terms*: Predefine a fixed list of Ontology Terms for the field from which values can be picked.
- *Root Ontology Term*: Limits the values of the field to children of the root term. This option is only possible if the checkbox "Fixed Ontology Terms" is disabled.

Fields of data type Ontology Term are displayed as described below:

- *In Grids*: A comma-separated list of the display fields is shown - not editable
- *In Forms*:
  - *Fixed Terms*: A combobox item is shown from which one can pick an Ontology Term
  - *Non-fixed Terms*: An add button is shown that opens the term browser to pick a term in the graph that is attributed to the field
  - *Multiple*: A list grid is shown with a remove button per row
  - *Single*: The selected term is shown in the combobox item for fixed terms or next to the add button for non-fixed terms

Default Ontology fields are present for Content Type, Location Type, and others. Contrary to the custom fields, it is possible to select the Ontology graph while adding a term. The Ontology browser will open in a pop-up window in order to search and add an Ontology Term. When clicking on the "Remove" button of an already added Ontology Term, it will be removed.

The linked Ontology Terms are visible in a separate field "*Record Ontology Terms*" in the form of the specific record. The field is only accessible after creation of a record (Content type, location type, etc.), not in the Creation Form when adding a new record.

Ontology term fields support Excel export/import:

- *Export*: The display field is exported as a comma-separated list.
- *Import*: By entering a comma-separated list of display fields in the excel file. SLIMS will need to query the graph attributed to the field to find the other Ontology Term information (label or URI).

## 10.6. History

### 10.6.1. Types of History

The History (since SLIMS 4.3) provides a detailed granular record of the changes that occur across the various concepts in SLIMS. Most importantly, this means that changes (adds, updates, and removes) are tracked on the following tables (\*Subject to change):

Attachment, AttachmentType, Content, ContentRelationType, ContentRequest, ContentType, Customer, Datatype, DbTable, Dimension, Disease, EventType, Experiment, ExperimentRun, ExperimentRunStepContent, ExperimentRunStep, ExperimentStep, ExperimentTemplate, Field, Functionality, FunctionalityGrant, Groups, Holiday, Instrument, InstrumentType, LabelTemplate, LabSetting, Location, LocationDetailExcelTemplate, LocationType, Module, Note, OrderContent, Order, OrderType, Perspective, Printer, Product, Project, Provider, QcCard, QcTemplate, QcTemplateRule, ReportTemplate, Requestable, Request, Result, Role, Rule, Scenario, SOP, Source, Status, StatusTransition, TableField, Test, Unit, User, Variant, WorkList, WorkListTemplate

Every change is kept as one record in the changelog table. The changes to the fields are stored in the changelogentry table that refers to this changelog table. Changelogentries refer to the fields by their name and track their old and new values (in both a technical format and a displayable format).

In case of an addition the old value will always be null. In case of a remove the new value will always be null. When updating both, the old value and the new value may be filled. The only way to see a remove history entry is to look into the database.

The values of the changes should be displayed clearly and be readable: quantities should show as quantities, foreign keys should show the correct display fields, etc.

Legacy history (previous to SLIMS 4.3) will be kept in the new changelog table but will not be parsed and will thus be shown in the old fashion (enumeration of the changes in one line).

Upon upgrading, the state of every table that has history will be stored in the changelog table as an "upgrade" operation. This operation will be similar to an "Add" operation. This means

it only has new values. These change logs may be hidden later on (\*Subject to change). This change is applied only for customers that used a SLIMS version lower than 4.3.

The lab setting "*History Language*" can be used to set the language in which all users see history events for the whole SLIMS instance.

## 10.6.2. Creation of History Records

All adds, updates, and removes through SLIMS DAO API on tables that require history will trigger creation of new rows in the changelog table. Creation of these changelog rows will happen asynchronously after a database transaction is successfully submitted.

If multiple operations happen on the same record within the same transaction they may be combined in the following cases:

- Update into Update => Condensed into one combined update
- Add into Update => Condensed into one combined add
- Add into Remove => No history is created

ⓘ Saving a record without changing any field value always creates a history record. Audit values are updated and a changelog record is created, even though there is no field value change.

## 10.6.3. History Save

History does not get created when automatic saves happen in SLIMS UI (for example, the automatic save that occurs after editing rich text fields). Creation of History is deferred until one minute passes without changes. At that point, one history record is made that reflects the complete change. If the user saves manually, such as when they click the save button or exit the rich text editor, history is created immediately.

## 10.6.4. History Event

The last operation type in the changelog is "Event." These events represent important occurrences in the life cycle of one or multiple records. A good example of this would be "*Content is linked to an experiment run step*." In addition, history events are captured in many cases where records related to the selected record are updated, created, removed, linked, or unlinked. Additional events can also be added in various points via the EventService API. Adding an event means adding a record in the Event table. The records associated with that event will get a new entry in their changelog that will point to this event.

An event is explained by event descriptions generated by the system. Event descriptions are translated automatically into the language set by the History Language lab setting, and therefore cannot be customized.

## 10.6.5. History User Interface

There are a few places in the UI where users can view history on a per record basis. History can be seen in selection menus, in grid gear menus, and in right-click context menus wherever history is enabled. They can be seen per content, order, result, note, and protocol run step. The history is shown with the most recent change on top. The new changelogs can be seen on a

per field basis. The legacy history (previous to SLIMS 4.3) is still shown as one big text. Events are visualized with their type and the description.

The history is displayed in a table with these columns:

- *User*: The name from the "Full Name" field in the user's account and the user name below it. Only the user name will be displayed if the user's full name is missing.
- *Date (Local time)*: This is the date/time in the user's local time zone that a change occurred. It is set by the "*History Date Format*" lab setting and includes the time up to the second.
- *Date (UTC)*: This is the date/time in the UTC time zone that a change occurred, for comparison to the user's local time, though it also follows the format set by the "*History Date Format*" lab setting. Time includes the time value up to the second.
- *Type*: New record, update, signature, etc. as seen in Types of History and Creation of History records.
- *Changes*: The specific details of the history event. This could include the fields that were added with a new record, or the old and new values for a field that was updated. When fields of type time, date, or Date and time are updated, the fields will show their value in the date format set by the "*History Date Format*" lab setting up to the second. "Date and time" fields are sensitive to the user's timezone and display their values relative to the user's local timezone. "Date" and "Time" fields, however, are not sensitive to the user's timezone and display as the value was entered by a user initially.

Time and Date and time fields display the hours, minutes, and seconds in the UI. The database saves the milliseconds as well, though milliseconds are not displayed in the UI.

Users can only view changes to fields they have access to. For the legacy history (previous to SLIMS 4.3) this means if a user does not have access to all fields, they cannot see it. For history from SLIMS 4.3 onward, only the changes to fields they do not have access to are hidden.

N/A values are displayed in history as a circled NA.

There is also history on a per table basis. It can be accessed via the general context menu in SLIMS grids. This will show all the changes that happened to a certain table. Note that this is unfiltered and may show changes to records that are not in the view the user is currently using. The user can only see the history of the records that they have access to. If record ownership changes, the ownership of the history records changes with it.

Changes related to many-to-many fields (those with multiple selections from a list, like an "Only for selected Tests" attribute) have their own grids in the "Changes" column. The "Updated field" shows the field that the update related to. the "Old value" and "New value" grids show the changed values of the many-to-many. The updated attributes are color coded to show what changed.

- Highlights in **blue** indicate updates.
- Highlights in **green** indicate additions.
- Highlights in **red** indicate deletions.

Comments are captured in a "Comments" row in history grids. This comments field only shows in history for events that users can enter comments for. Users can enter comments for the event types below:

- Order cancelled
- Order reverted
- Result rejected
- Result cancelled
- Result repeated
- Result remarked
- Result reverted
- Rule acknowledged
- Result removed from worklist
- Step removed from worklist
- Electronic signature
- Reason for change

Authentication and License events are created systematically when certain actions are taken. This type of event is captured in the "Authentication Description" row in history grids. Authentication descriptions are captured for these events:

- On a failed login attempt
- On update to a role that causes the license limits to be exceeded
- On update to a user that causes the license limits to be exceeded
- On attempt to login when all licenses were in use
- On user login
- On user logout

The ELN History is accessed by clicking the history button on the upper right side of the ELN tree. If the history button is not present, click the gear icon in the upper right side of the ELN tree to show all available actions. Clicking the history button will display a pop-up window in which a table can be selected to display its history. Some technical details should be kept in mind when choosing a table:

- Protocol Run: This will show the history of all run information, including both standard protocol runs and simplified experiments used in either the ELN or Workflow modules.
- Protocol Run Step: This will show the history of all run step information, including both standard protocol run steps and simplified experiment blocks used in either the ELN or Workflow modules.

The History List can be filtered by ID. This is done by entering the ID in the Record column. The filter only works with the *equals* operator, not *contains*. If the option *Show Changes* is disabled, the details of the changes can be seen in a tooltip when hovering over the Record column.

When there are no changes for an update, the change value will display as *No changes were made*. By default, updates without a change are not shown. Those records can be shown in the history by checking 'Show updates without changes.'

#### 10.6.5.1. View Filters for History

Wherever history is collected in a grid, the view can be edited to make use of filters that facilitate the review of lengthy record histories. (Check out the *Create Customized View* section to see how to broadly use filters in a view.) Filters can be created for views with fields outside of the current context. For example, users with permissions to Requests, Content, Orders, and Requestables will have content, orders, and requestables available to use in history view filters when in the request table context. Two particularly useful filters exist for use with history grids that can filter to show only the selected event types, or filter to show only entries where the selected fields changed.

The "Event Type" option in the filter builder is available with the operators "equals (match case)," "not equal (match case)," "is one of," and "is not one of." Only the event types that are applicable for the module in which the user is accessing history are populated in the value dropdown (on the right) because history is context-specific. For example, in the Order module history, only values such as Request removed from content to order, Content linked to order, etc. are available. If the user filtered for "Event Type" "equals" "Content linked to order" the grid would show only history entries of content linked to order.

The "Field Updated" option in the filter builder is available with the operators "equals (match case)" and "is one of." Only one field can be selected with "equals," but multiple can be selected with "is one of." The view shows all changes to the record where the selected field(s) were modified with this filter because the filter applies to the transaction, not the individual field to keep the relevant details in the view. The filter also works with records that have many-to-many relationships defined to any other records, such as container types and content types. The view shows the multiple values in a table and highlights the changes.

The "Changes" column can be removed from a history grid as well to hide the old and new values entered for an updated record. This is done by removing the Changes column from the view.

### 10.7. Holidays

Holidays are useful when creating multiple orders at once. When adding multiple orders at once, you can specify the interval between two orders and whether to exclude holidays and weekends. If the order should be planned on a holiday or weekend, then SLIMS will plan it on the next working day. Correctly defining the planned holidays implies that no customer will be able to plan an order when nobody is working.

The Holidays module is separated in two parts: the left contains the list of holidays, and the right contains the details about the selected holiday. A Holiday is defined by a date and a description.

### 10.8. Lab Settings

The Lab Settings module is accessible through the *Setup Menu*. This module allows a user with access to set general SLIMS options that may affect multiple modules or edit a common layout in SLIMS.

The following Role Access rights can be configured for use in this module:

**Table 10.10. Required Permissions for Lab Settings Module**

Access rights Category	Access rights Name	Purpose
Setup: Miscellaneous	Lab Settings	This permission provides access to the Lab Settings module so the role user can manage the core setup of the SLIMS instance.

## 10.8.1. Attachments

- *Server attachment root directory:* This defines the root directory from which users can select a file when adding an attachment in SLIMS and choosing Server instead of Local. Users can only choose a file from the server when a value is given to the option "Server attachment root directory" and Tomcat has access to that server folder.
- *Maximum number of files allowed to upload at the same time:* If 0 or a negative integer are set, then there is no limitation on the number of attachments users can upload. The number of attachments a user can upload can be restricted to a particular amount by setting this value to a positive integer.
- *Always reuse binary equal attachments instead of checking attachment type:* This allows SLIMS to search for matching attachments that already exist every time an attachment is uploaded, independent of the attachment type option "Reuse Binary Equal Attachments," and in this way, save space by preventing file duplication. See the *Attachment Types* section for context and details.

## 10.8.2. SLIMS

These parameters are unique for each instance and will be shown for all SLIMS users of the same instance.

- *SLIMS instance name:* Name of the SLIMS instance that will be displayed below the SLIMS icon on the top left of the screen.
- *SLIMS instance name color:* Enter a color name or HTML color code to change the color of the instance name. The default color is grey.
- *SLIMS instance menu color:* Enter a color name or HTML color code to change the color of the menu bar on the top of SLIMS. The default color is white.

Instance name and colors are useful for when a customer is using multiple SLIMS instances within the same organization. Otherwise default colors should be kept.

- *SLIMS date format:* This feature displays all the dates in SLIMS according to predefined formats like European 24/05/2008 or Japanese 2008/05/24 or US 05/24/2008.
- *SLIMS languages:* Select the languages that will be available in SLIMS (language can be selected for each user in the users module).

- *Grid page size*: Select the number of items loaded per page in Grids (a number between 50 and 500 can be selected).
- *Suppress concurrent modification errors*: By enabling this option, concurrent change errors will be suppressed. See the *Edit Contents* section for more information.
- *Enable status 'Labeled'*: Enable or disable the "Labeled" status option. It can only be disabled if there is no content with the status "Labeled." Disabling this option also hides the "Update status" checkbox in the "Print labels" dialog box.
- *Allow addition of formula and summary columns*: By default this is disabled to prevent users from adding a Formula or Summary column to grids, as usage with custom data types is not optimal. However, this option can be enabled by administrators to allow users to add the column to grids.
- *Use deprecated content status field*: When activated, Content statuses cannot be edited and status items in forms and grids stay the same as in previous versions. When inactivated, Content statuses can be configured in the Statuses module, aligned on orders.
- *Text match method for grid filters*: This sets whether grid filters search for records that "Contain" or "Start With" the typed characters.
- *Email templates SMTP URI*: Set the address of the SMTP server to identify where SLIMS sends emails. See the *Email Templates* section to see how to define what the emails report.

Some examples of URI settings are displayed in the screenshots below. Generally only the Port must be listed so SLIMS knows which connection to send to the server through.

An example of a complex format that can be used is:

```
smtp://[emailaddress]server:[port#]?password=[password]&username=[sender]
```

In this example, the user is not defined. If the user's server is on the SLIMS server, using localhost could work for their SLIMS as shown in Figure 10.6.

**Figure 10.6. URI Example with Localhost as the Port**

Keyword	Value
Email templates SMTP URI	smtp:localhost

In this example, the user is also not defined. The Port is defined so SLIMS knows which connection to use to send emails to the server as shown in Figure 10.7.

**Figure 10.7. URI Example with Only Port Defined**

Keyword	Value
Email templates SMTP URI	smtp://gate.mycompany.local:25

This example defines the email address that receives email, the port to send to, a password, and the sender's name. Figure 10.8.

### Figure 10.8. Complex URI Example

Keyword	Value
Email templates SMTP URI	smtp://mycompany.mailserver:30?password=passxyz&username=

For more assistance on writing SMTP email address definition, visit the *Camel Apache* [<http://camel.apache.org/mail.html>] support page.

## 10.8.3. Orders

The following settings can be edited for orders:

- *Allow only one content record per order*: By enabling this option, there will be one content record per order. Thus, a new content record will be created for each new order.
- *Create default requestable for new test*: This option triggers the creation of a new Requestable for each newly created test.

The newly created Requestable will have the following field values identical to the created test: "Name, Description, Active." Remaining Requestable parameters will be left empty and can be edited in the *Requestables* module.

## 10.8.4. Contents

*Pedigree enabled*: This option has to be enabled if family pedigrees will be used in SLIMS. See the *Pedigree* module.

## 10.8.5. Electronic Lab Notes

- *Automatically finish protocol steps when results are filled*: If this checkbox is checked, protocol steps containing tests will be automatically changed to the "Finished" status in ELN when all their results are filled in the step.
- *Variants enabled*: Everything about Variants is hidden when this is not checked. This includes the functionality or the availability of the special tabs in Content management, Order management, and ELN.
- *Default simple protocol*: If this checkbox is checked, the protocol type "Simplified" will be preselected when creating a new experiment.

## 10.8.6. Global search

This impacts the way the Global search acts. The following options are available:

- *Include text fields in global search*
- *Text match method for global search*: Equals, Starts with, or Contains
- *Case sensitive global search (Only for Oracle databases)*

## 10.8.7. Users

- *Session timeout (minutes):* This sets the number of minutes before the client's session is terminated after taking no action (saving the page, moving or clicking with the mouse, typing) for an extended period of time. The default is 30 minutes. SLIMS validates the value that is entered for the session timeout to ensure that it is reasonable: between 0 minutes and 1500 minutes (a little more than a day).
- *Group access rights fields together:* By enabling this option, the user and group field will be grouped together. See the *Group access rights* section for more details.

## 10.8.8. API

- *SLIMS Python-api allowed urls and flow-ids (url -> flowid,url -> flowid2,url2 -> flowid)*
- *BaseSpace access token:* The Illumina BaseSpace access token.

## 10.8.9. Macros

- *Show value specification during macro execution:* When this functionality is disabled, only the macro fields which are set to Specify new value can be "Shown during execution."

## 10.8.10. SLIMS Share

- *SLIMSShare server:* The URL where exported packages will be stored. The field is empty by default. If a URL is present, SLIMS verifies the URL is in a valid format and provides access to various SLIMS Share module areas. An example of a proper URL format for SLIMS Share Server is:

`https://slims-customer.net/slimsshare`

- *SLIMSShare user password:* The password to download packages from the SLIMS Share server. The field is empty by default and when a value is entered, the password is hidden by generic password obfuscation characters (ex: \*\*\*\*\*).

## 10.8.11. Locations

- *Use pagination in Location trees:* Checking this option uses paged fetches to handle large amounts of locations, but using it disables all link actions for locations.

## 10.8.12. Electronic Signature

- *Electronic signature requires a username to be provided.:* When enabled, this option requires both the username and password of the logged in user to be entered any time an action is taken that requires the Electronic Signature. When disabled, only the password of the currently logged in user is needed for that user to sign a record.
- *Electronic signature requires comments to be filled.:* When enabled, this option requires at least one character to be entered into the comment text box when an Electronic Signature is provided for an action. The comments box is available whether this is enabled or not, but is not required when this setting is disabled.

- *Electronic signature requires a meaning to be provided.*: When enabled, this option provides a dropdown with various Meanings for the Electronic Signature, and requires the user to select a provided option. The Meanings provided are: Author, Review, Approve, and Reject.
- *Require electronic signature on results repeat and rule evaluation acknowledgement*: When enabled, this option requires the electronic signature when the user clicks "Repeat Result" or "Acknowledge Rule Evaluation" for a result. An electronic signature event is added in history and in "View Events" on the results and rules. When disabled, the electronic signature is not required when the user clicks "Repeat Result" or "Acknowledge Rule Evaluation," but a comment is required. A "Result Repeated" or "Rule Acknowledged" event is added instead and contains the comment that was entered.

## 10.8.13. Tests

- *Show groovy configuration for tests*: Enabling this option makes the value expression field on a test visible. Showing the value expression field on tests is obsolete and groovys entered in the value expression field when the setting is enabled might not be supported in later versions of SLIMS. (The value expression field to configure the test result value can be configured on result values, and is available in the *Configure Result Datatypes* section.)

## 10.8.14. History Formatting

- *History Date Format*: The default selection for new instances is "SLIMS Default w/timezone." This option provides a dropdown with multiple date formats with and without time zone to cover various labs' compliance needs. For example, a human readable time zone and an ISO 8601 format are options in the dropdown. The user's local time and the UTC time columns are displayed in the selected format, and any date, time, and datetime fields in the Changes column are displayed in the selected format including hours, minutes, and seconds.
- *History Language*: Set the language of history (audit events) for the whole SLIMS instance. This is set for the server, so users will see history in this language even if their user setting specifies a different language. Changing the language will not alter the language of prior/existing history entries, so it should be the first setting Administrators set on a new SLIMS instance. These areas will be displayed in the selected language: filter options, setting options, column labels, event types, field labels, event descriptions, and button names in the instance histories.

## 10.9. License

The License module contains all the information about the license and login data of users. It contains three tabs:

- **Chart**: A chart of 'User logged in' and 'User logged out' events is displayed
- **License**: A plain visual representation of the contents of the license file
- **History**: A list of all login events

### 10.9.1. Chart Tab

In this tab, the user can see a chart of the 'User logged in' and 'User logged out' events for a certain resolution. Events are represented in a two dimensional chart where the X axis repre-

sents the date and time of the event, and the Y axis represents the number of logged in users with a slightly different meaning depending on the chosen resolution.

The 'Resolution of chart' and 'Date range' can be selected in the corresponding menus. Possible values for Resolution of chart are:

- Event: Each 'log in' and 'log out' event is shown and the Y coordinate represents the number of users logged in at that time point.
- Daily: One point per day is shown and the Y coordinate represents the *maximum number* of users logged in *at the same time* during one day.
- Monthly: One point per month is shown and the Y coordinate represents the *maximum number* of users logged in *at the same time* during one month.
- Yearly: One point per year is shown and the Y coordinate represents the *maximum number* of users logged in *at the same time* during one year.

The points on the chart can be hovered over to see the time and the amount of users during that event.

## 10.9.2. License Tab

In this tab, the user can see a plain visual representation of the contents of the license file and a grid explaining the License Type with breakdown of user metrics.

Each SLIMS instance needs a license file that looks like this:

```
instance=(insert instance name)
serviceUsers=(comma delimited list of user names of SLIMS service engineers.)
bypassUsers=(comma delimited list of user names used by the customer; not required.)
licenseType=(premium or standard)
maxNamed=(any integer)
maxConcurrent=(any integer)
strict=(true or false)
```

The parameters are:

- Instance: The name of the instance that the license file is applicable to.
- Service Users: A list of user names used by SLIMS service engineers, which do not consume a concurrent user license. These users will also be read-only in the SLIMS Users Module.
- Bypass Users: Optional list of customer user names that are protected from being denied access to SLIMS due to insufficient user licenses. These should be users that can resolve license problems, like modifying other users' accounts to terminate access and free up concurrent space, or make a named user account inactive.
- License Type: Standard and Premium license types determine the modules that the customer has access to in SLIMS.
- Max Named: This is how many users with a named license can be created. A named user license is associated with a single person. It does not count inactive users.
- Max Concurrent: A concurrent user license is shared between multiple users. Every logged-in user with a concurrent user license consumes a concurrent user license at that point in

time. (Read-only users that have access rights to the Order Module only do not count against available concurrent licenses.)

- Strict: This boolean determines if excess users are allowed if the amount of users exceeds the maximum number.
  - ① It is difficult to strictly enforce maxNamed and maxConcurrent in every case.

The license file is accompanied by an electronic signature and a public key. The license file is verified when starting the SLIMS instance to make sure that it was provided by SLIMS.

The grid shows the license type, the maximum number of users that was set for the type, the Actual which reports how many users logged in for those types, and Comments explains more about the restriction that is used.

### 10.9.3. History Tab

In this tab, the user can see all login events:

- User logged in: created on login and associated with the logged in user.
- User logged out: created when logging out, when user session is terminated and when the user's session on the server is destroyed. The event is associated with the logged out user.
- User session terminated: created when an admin terminates a user's session. This will be associated with the terminated user and the terminating user will be mentioned as well.
- User exceeds limit: created when somebody tries to login when the maximum amount of concurrent users are already logged in. The event will contain all the logged in users and will be associated with the user who tries to log in.
- Failed login attempt: Created on wrong login credentials and contains the failure reason: user does not exist, wrong password, or user not active. If there is a user with the username, the event is associated with that user. Otherwise it is attributed to the SLIMSGATE user.

## 10.10. Notifications

Notifications can be created using SLIMSGATE flows or within *Status Rules* or *Conditional Value Expression Rules*.

These notifications will then appear in the top-right menu bar, under the bell menu (which is also the SLIMSGATE flow monitoring menu). When the bell turns red, the user has an unread notification. Once all the notifications have been read the bell turns grey.

The user setting "Enable Browser Notifications" makes it possible to show notifications as browser notifications. The minimum level of the notifications to display this way can be selected among (in increasing order of importance): 'INFO,' 'SUCCESS,' 'WARNING,' 'ERROR'.

## 10.11. Ontology Graphs

The Ontology Graphs module is available under the Setup menu and its access can be role restricted. See the *Roles* module for more details about access rights.

The following Role Access rights can be configured for use in this module:

**Table 10.11. Required Permissions for Ontology Graphs Module**

Access rights Category	Access rights Name	Purpose
Setup: Miscellaneous	Ontology Graphs	This permission provides access to the Ontology Graphs module which allows role users to define graphs for ontology terms.

An Ontology graph is defined with:

- *Name*: The Ontology graph name
- *End Point*: The URL of the Ontology repository. It represents the SPARQL endpoint where SLIMS can query for terms.
- *Graph*: Path of the File context of the Ontology repository
- *Version*: Ontology graph version

New Ontology graphs can be added in the Ontology graphs module by using the *Add* icon on the top left.

Ontology terms are defined in SLIMS by an URI value, a Label, and a source Ontology Graph. An Ontology term can be added in SLIMS either in:

- *Content types/Location types*: By using right click. Ontology terms that have already been added cannot be visualized yet in SLIMS.
- *Fields*: By creating a custom field of type "Ontology term." See the *Fields* section for more details.

Terms of an Ontology graph can be shown in a SLIMS pop-up window that displays a SPARQL Browser window by right-clicking on an Ontology graph. Users can browse terms in the SLIMS pop-up window, which executes queries on the SPARQL endpoint-based source Ontology graph and returns Ontology terms from the SPARQL endpoint.

## 10.12. SLIMS Configuration Import Options

Three options exist in the SLIMS user interface that alleviate the setup of a fresh SLIMS instance by providing packages - small bundles of configuration that are ready to install - in various ways. The Package Browser module is a way for our SLIMS engineers to share complicated configurations out-of-the-box to your SLIMS instance. SLIMS Share is for exporting parts of a SLIMS instance (\*the parts that can be turned into packages) and importing them into another similar SLIMS instance for further configuration, sans data. Finally, SLIMS Store has many browsable prebuilt packages that can be installed in full or in part into a SLIMS instance.

The following Role Access rights can be configured for use in these modules:

**Table 10.12. Permissions for Package Importing Modules**

Access rights Category	Access rights Name	Purpose
Setup: Miscellaneous	Package Browser	When enabled, the role user can access the Package

Access rights Category	Access rights Name	Purpose
Setup: Miscellaneous	SLIMS Share	Browser module to access out-of-the-box packages (groups of related SLIMS configuration) to use in the setup of a SLIMS instance. (For example: A package that includes MiSeq instrument types and protocol types to allow SLIMS to be configured correctly for NGS.)
Setup: Miscellaneous	Store	This permission allows role users to access the SLIMS Share module in which configuration from one SLIMS instance can be transformed into a package and imported into another SLIMS instance. This does not transfer data, only the SLIMS setup.  This permission allows role users to access the SLIMS Store module in which out-of-the-box SLIMS configurations are stored. Users can search for useful packages and install all or part of a package into their SLIMS instance.

## 10.12.1. Package Browser

The package browser module is used to browse through the packages available for this instance from the xio website (Browse public) or SLIMS local packages (Browse locally).

A package can be imported by searching for the name of the package within the folder structure of packages and then clicking on the *Import* button at the bottom-right corner. This opens a pop-up window that displays the package and its dependencies that will be imported simultaneously. Clicking the "Import" button finishes the import.

### 10.12.1.1. Imported Packages

The imported package tab provides the list of packages that have already been imported.

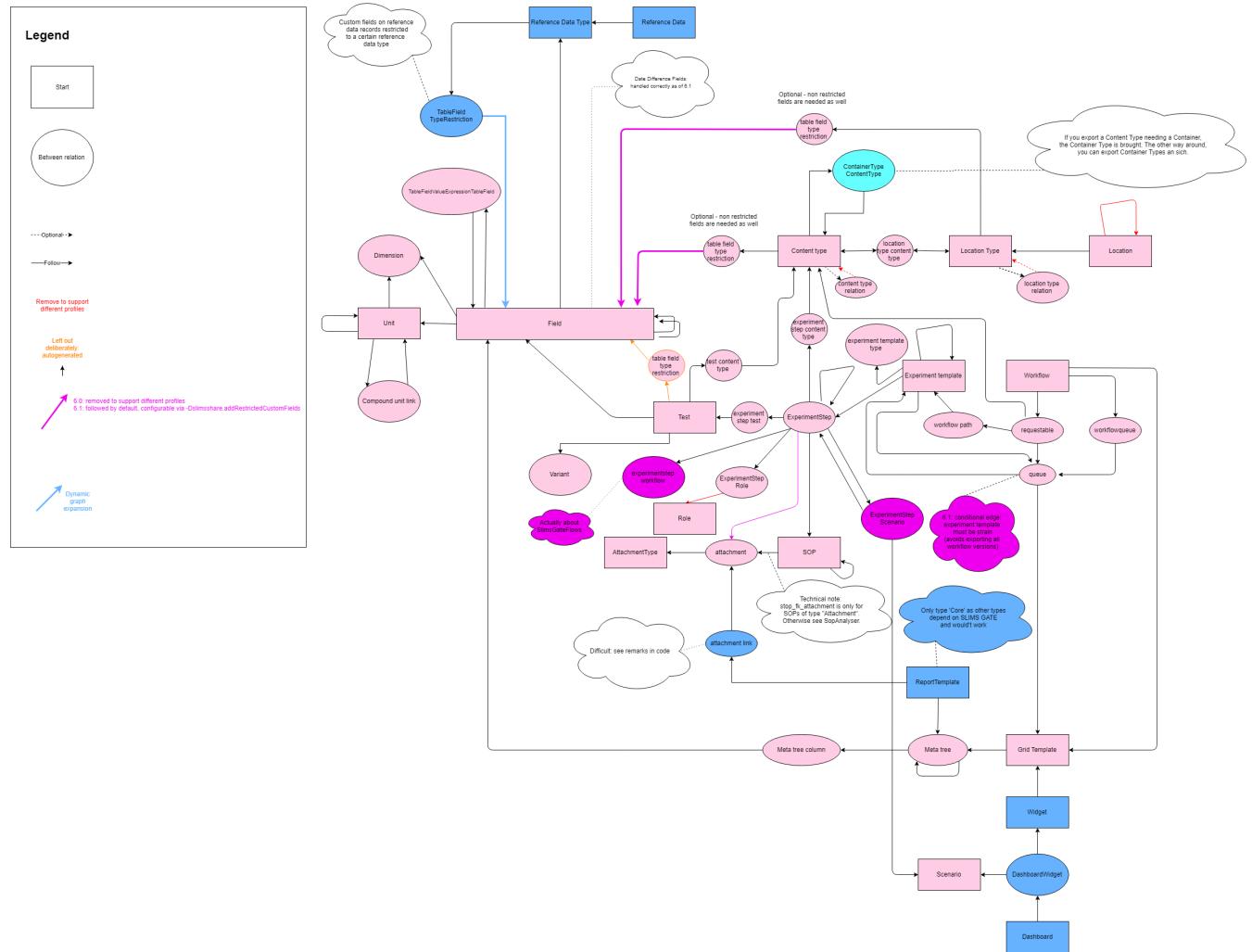
## 10.12.2. SLIMS Share

SLIMS Share is a platform for sharing configuration between customers. Agilent will give access to one SLIMS Share website that contains all of the preconfigured packages. It is possible to setup which SLIMS Share the user's instance will use in Lab Settings, in case multiple SLIMS Share would be present (ex: in the event that a customer has their own repository).

In the metagraph below, the dependencies between the different tables are shown. Note here that the direction of the arrow should be the dependency (not the direction of the foreign key),

and also that squares are to indicate the things that make sense to export on themselves while circles are for dependency only.

## Figure 10.9. SLIMS Share Metagraph



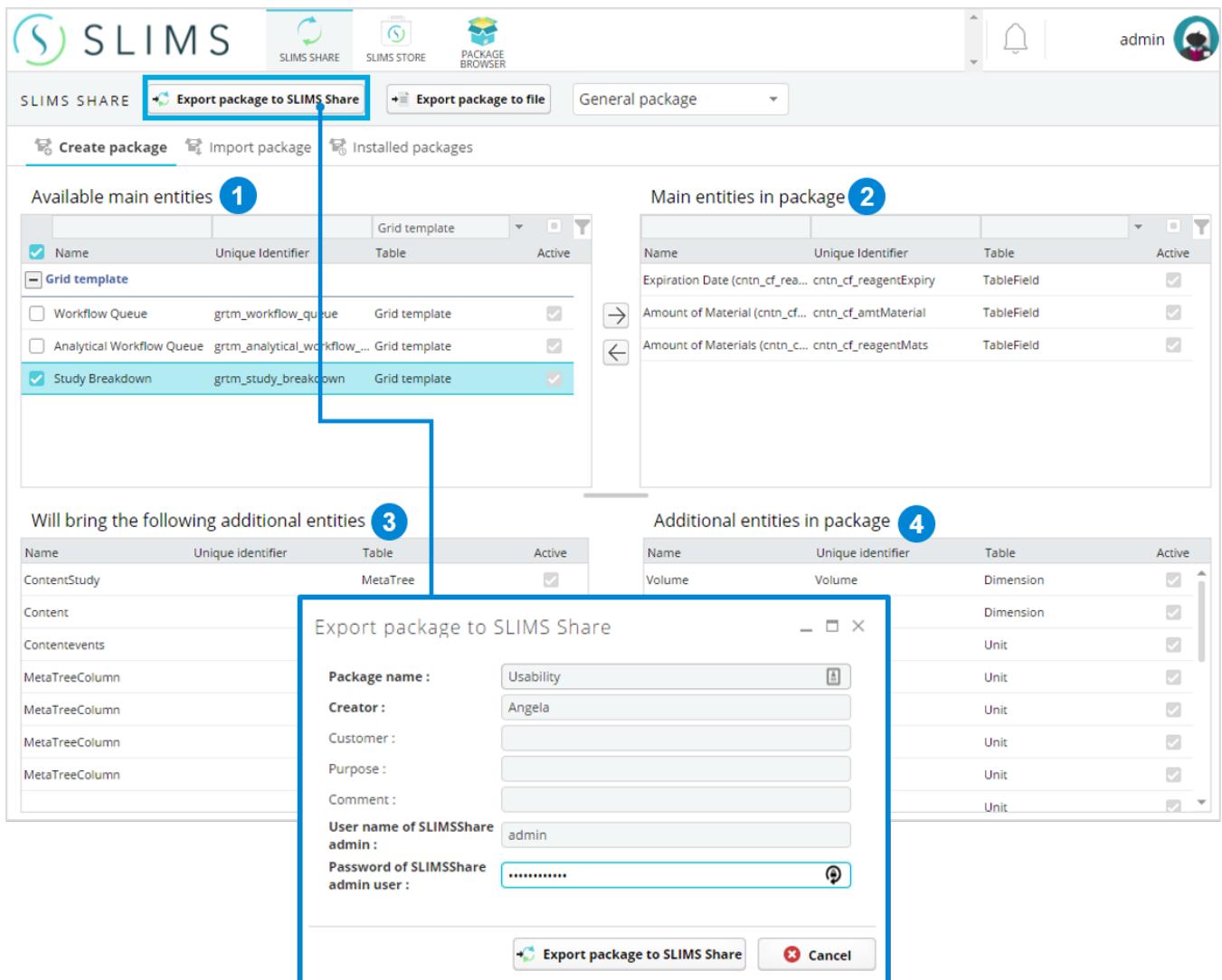
The SLIMS Share module is split into three tabs. The first tab, "Create Package," is for creating and exporting packages to be shared, the second tab, "Import Package," is for importing existing packages, and the third tab, "Installed Packages," displays the packages that have been installed.

### 10.12.2.1. Create Package Tab

A package is made up of a collection of one or more main entities that exist in the SLIMS instance. The available main entities are listed in the top-left grid (1) grouped by type. Some main entities are made up of other entities that are required (like a protocol that contains various protocol steps and tests). These are listed in the bottom-left grid (3) when a main entity is selected for a package and moves into the "Main entities in package" grid on the top-right (2). This is done by checking the box for one or multiple main entities and then clicking the "right arrow" icon.

Whenever there are more than 1,000 records in the Available Main Entities or Main Entities in Package grids, it can be difficult to filter by the name of the table. The "Table" column of both of the grids offers a dropdown of all the available tables to make finding a table easier. The table names are displayed as user-friendly names instead of the technical names so they are easy for customers to identify.

**Figure 10.10. Create Packages**



Begin by sorting the available main entities with the filter above the first grid (1). The "General" package kind can be used to export any combination of configuration entities as packages, and the "Workflow Packages" package kind can be used to export a single protocol from a workflow. Then you can open the various entity categories and check the boxes to include packages. Use the right arrow icon to move the entities into the main entities selected for the package (2).

The "Export Package to File" button will open a pop-up to enter a file name and create a file that contains the selected entities that can be downloaded and then imported within the Import Package tab. After adding a file name, click the "Export Package to File" button in the pop-up.

The "Export Package to SLIMS Share" button opens a pop-up window where a Package Name, Creator, SLIMS Share Admin user name, and SLIMS Share Admin password must be added, and then the "Export Package to SLIMS Share" button can be clicked. This action will export the package and make it available for the user's company (on the Import Package tab to be shared with their other instances). This button is only available when a valid SLIMS Share URL is present in the Lab Settings module.

① Notes about exporting packages:

- **About Roles:** When roles are imported, the functionality rights attached to the role come with it. The system checks for uniqueness so that it will not duplicate functionalities that are already present. To prevent confusion, SLIMS Share also checks for uniqueness with imported roles to stop the import of a role name that already exists.
- **About Status Workflows:** Status fields that identify a status workflow to be used on content (in other words, a tbfl\_fk\_statusWorkflow field) export their related status workflow when included in a package for export. SLIMS comes with several default status workflows which exist by default and have their own static identifiers. Therefore, while default status workflows can be exported, importing them does not create new status workflows. Importing custom status workflows into a SLIMS instance will create new entities for the status workflow, statuses, and status transitions. Content Order status workflows are deprecated and cannot be exported.
- **About Pedigrees:** It is possible to import and export Add Child and Add Parent macro steps that are commonly needed for Pedigrees and Biobanks, though these steps do not show up unless Pedigrees is enabled.
- **About Macros:** Macros that include Add Event steps, Create Location steps, Enroll Content steps, and Execute SLIMS GATE Flow steps are supported. Macros with Print Label steps are partially supported and can be exported with SLIMS Share, but the preconfigured printer and label template will not be preserved and will need to be reconfigured manually after the import. This allows the macro step to be used in a larger macro that can be exported.
- **About Printers:** Printers are also supported, but are currently exported without role restrictions and possible label templates.
- **About Layouts:** It is possible to easily share created Dashboards and Widgets using SLIMS Share, as these features can be imported and exported. You can even export entities from the "Perspective" and "GridTemplate" tables to export Views and Layouts from one instance to another.
- **About Rules:** When a rule is selected that is linked to a content type (by way of "Only applicable for selected Content Types"), the link is retained for export from SLIMS Share.
- **About QC Templates:** QC Templates can also be imported and exported, and QC type Rules are exported as well.
- **About Plugins:** When plugins are exported, the dependencies that are declared in the jar file (by a SLIMS plugin developer) will be exported when the package is created. That way the plugin will run right after import since its dependencies will import with it. Missing dependencies will prevent a plugin from running.
- The following tables are supported in SLIMS Share:

Attachment type	Case report form template	Content event type
-----------------	---------------------------	--------------------

Content type	Dashboard	DataSourceField
DNA index set	Email template	Grid template
Holiday	Icon	Instrument type
Label template	Lab Setting	Location
Location type	Macro	Order type
Plate index design	Plugin	Printer
Product	Protocol	QC template
Reference data record	Reference data type	Report template
Requestable	Requestable group	Role
Rule	Scheduled job	Sequencing settings
SOP	Source	Stability study template
Status Workflow	Study	TableField
Test	Unit	Widget
Workflow	Worklist template	

- Entities from the following tables may be brought as dependencies:

Attachment	Attachment link	Compound unit link
Content type relation	Crf applicability	Dashboard widget
Dimension	DNA index	DNA index location
DNA index set	Experiment template type	Functionality
Functionality access right	Location type content type	Location type relation content management
Location type relation location type management	Macro content type	Macro step
Metatree	Metatree column	Order type
Product	Protocol step	Protocol step content type
Protocol step macro	Protocol step role	Protocol step sub protocol
Protocol step test	Protocol step workflow	Qc template rule
Qc worklist cell template	Qc worklist row template	Queue
Requestable order type	Requestable requestable group	Requestable test
Rule content type	Rule test	Specification
Status	Status Transition	Storage condition
Study Crf template	Study event type	TableField type restriction
TableField value expression TableField	Test	Test content type
Test value expression test	Timepoint	Timepoint specification
Variant	Workflow path	Workflow queue
Worklist template protocol step	Worklist template Protocol step test	Worklist template workflow

- The following macro steps are supported:

CONTENT_CREATE	CONTENT_DERIVE	CONTENT_ALIQUOT
CONTENT_DILUTE	CONTENT_MOVE	CONTENT_EDIT
CONTENT_MIX	CONTENT_COPY	CONTENT_REMOVE
CONTENT_UNREMOVE	CONTENT_ADD_EVENT	CONTENT_ENROLL
RELATION_ADD_CHILD	RELATION_ADD_PARENT	CREATE_LOCATION
CONTENT_PRINT_LABELS	EXECUTE_SLIMSGATE_FLOW	

### 10.12.2. Import Package Tab

This tab is split into three parts. The list of available packages are in the upper-left grid (1), and the description of entities in the selected package are in the right grid (2). The general details of the package are in the bottom-left grid (3). Finally, the import actions are at the bottom of the Main Entities and the Available Packages grids (4).

**Figure 10.11. Import Packages**

**Available packages (1)**

ID	Name	Creator	Custom...	Purpose	Comm...	SLIMS v...	Packag...
346	Unofficial ...	valerie	SLIMS devs	Develop...	Built from ...	6.6	1
347	MathewMi...	Mathew				6.6	1
348	MathewTest	Mathew				6.6	1
349	reagentsP...	Angela	Agilent SLI...	Training	contains 4 ...	6.6	1
350	LocationTe...	Mathew				6.6	1

**Main entities in package (2)**

Name	Unique identifier	Table	Active
Reagents		ContentType	✓
Buffer	Buffer	ContentType	✓
Blue Dye	Bdye	ContentType	✓
Green Dye	Gdye	ContentType	✓
Gel Matrix	matrix	ContentType	✓
Amount of Materials	cntn_cf_reagentMats	TableField	✓
Expiration Date	cntn_cf_reagentExpiry	TableField	✓
Buffer		Icon	✓
Blue Dye		Icon	✓
Green Dye		Icon	✓
Flask		Icon	✓
cl	cl	Unit	✓
I	I	Unit	✓
dm^3	dm^3	Unit	✓
dm	dm	Unit	✓

**reagentsPackage (3)**

- Creator: Angela
- Customer: Agilent SLIMS
- Purpose: Training
- Comment: contains 4 DNA Reagents

**Import actions (4)**

Import from file (1)

Import selected main entities (4)

To begin importing packages, determine whether to import packages from the SLIMS Share Server or from a file. If from the server, the Available Packages grid will be populated and one or more desired packages can be selected to import. The Available Packages grid loads packages from the SLIMS Share Server only when a valid SLIMS Share server URL is present in Lab Settings. To import packages from a file (JSON type files), use the "Import from File" button in-

stead. After the file is selected, it is only cached in SLIMS for a while. The package shows up in the Available Packages grid but is not yet imported.

ⓘ The automatically generated name of the file is displayed based on your browser and does not correspond to the path of the selected file in the file system.

Next, select all of the Main Entities in Package if you want to import the entire package, or select only the desired package's entities by checking the boxes for the ones you want. Then click the "Import Selected Main Entities" button in the bottom-right to initiate the import.

A pop-up provides a log of the import. The text within the pop-up can be copied for convenience if investigation into log issues is required. SLIMS will auto-scroll to the last line of the log so it is faster to see the result of the import. The log lists the process that is occurring (Start, Found, Skipped, Updating), the entity name, the unique identifier (if the entity has one), and the GUID in case troubleshooting is required.

### 10.12.2.3. Installed Packages Tab

This tab displays information about the packages that have been installed in the SLIMS instance. They could have been installed from the SLIMS Store, from an exported package file, or from another SLIMS Share instance, whether it is your own separate SLIMS or a custom SLIMS Share. This tab exposes the details about the packages that were installed so that a SLIMS engineer or super user can tell at a glance when packages were installed, who by, and if they have become outdated.

Figure 10.12. Installed Packages

The screenshot displays the SLIMS Share interface, specifically the 'Installed packages' tab. The main grid lists five packages installed on SLIMS Version 6.6, all by 'admin' on different dates. A context menu is open over the last package in the list. A modal window titled 'Package entities' shows a JSON object representing a package entity. Another window shows the details of a package, including 'Main entities in package' (administrator, technician) and 'Additional entities in package' (FunctionalityGrant entries).

Old packages can be up-versioned with some actions performed from this tab.

The grid on this tab can be independently refreshed, exported, and printed from the gear menu. The grid contains these details:

- **Source column:** Explains the origin of a package with "File" if the package came from a file, "SLIMS Store" if it came from the SLIMS Store, or the URL of a SLIMS Share instance for custom SLIMS Share installs.
- **Package Name column:** Displays the file name of a source file, package name of installed SLIMS Store packages, or "Customer created package" for custom packages from SLIMS Share.
- **Package Version column:** Displays the version of the package so engineers/SLIMS super users can tell whether a version is old.
- **SLIMS Version column:** Displays the version of SLIMS the package was installed on. This is important because SLIMS cannot have an older version of SLIMS' packages installed on it. (For example, an error will prevent a package from SLIMS v6.5 from being installed on SLIMS v6.6.)
- **Installed By and Installed On columns:** Provide the username, date, and time on which the package was installed.

- **Partially Installed column:** Shows a checked box for packages for which only some of the entities were installed, and an empty checkbox if all of the entities were fully installed. This way, engineers or SLIMS super users can tell whether a package should be retried.
- **Show Entities button:** An action button that opens a pop-up to display the entities that were installed in the package for easy read-only visibility.
- **Version button:** An action button that allows engineers or SLIMS super users to create a new version of the installed package.

An alternate view of the Create Package tab opens when the SLIMS engineer or super user clicks on the Version button. The grids update to show all of the entities in the installed package in the "Main entities in package" grid, and the dependent entities in the "Additional entities in package" grid. If any entities were removed since the installation of the original package version, a warning message can detect it and warn the user that some entities are missing (and provides their identifiers). The user can download the package to bring the entities back if they encounter this warning.

This view of the Create Package tab makes it possible for the engineer/super user to install entities to update a package. First, the user can add any desired entities to the package, if applicable. The "Export new package version to SLIMS Share" and "Export new package version to file" buttons provide two ways to do so. When either is clicked, a new package version is created that is linked to the parent package.

ⓘ The first button is only present if a valid SLIMS Share URL is entered in the Lab Settings module.

When the user clicks "Export new package version to SLIMS Share," they can enter the package details in the pop-up.

**Package Name:** The name is already filled with the original package name and is read-only.

ⓘ For packages imported before SLIMS 6.6, the name will be empty and users will get a warning when trying to push a new version to SLIMS Share. The warning is the same as when the user tries to create a new package with a name that already exists. The warning provided is "A package already exists with this name: [name]. Would you like to create a new version?"

When the user clicks "Export new package version to file," they can enter the file name in the pop-up and then click the Export button. The file can then be installed like it was initially in the Import Package tab.

#### 10.12.2.4. Importing and Exporting Workflow Protocol Packages

Workflow Protocol packages are different from other packages. The Package Kind filter was created initially for making it easier to find the appropriate entities to be part of a protocol package. Workflow Protocol packages are created by exporting them from a SLIMS instance into the SLIMS Share database. They cannot be exported to file. After exporting a protocol package, administrators will be able to import them into their existing workflows. This will be possible in a different SLIMS instance and in the same SLIMS instance that the protocol package was created on. This makes importing protocol packages useful for transferring complicated protocols, or re-using the same protocol.

1. To create a Workflow Protocol package, select the package kind "Workflow Protocol." This will filter out all of the workflow entities and leave the protocol related entities. Protocol packages cannot include a workflow, and must include exactly one protocol.

2. Look in the ExperimentTemplate category for the protocol defined in workflows that you want to include in the package. Protocol entities are recognized by the name of the workflow and the start and end queue names in parentheses (for example: "Add Catalyst (DNA extraction: q1 > q2)").
3. Select only one protocol entity with the checkbox, then click the right arrow icon to move it into the Main Entities in Package box.
4. Use the export instructions from earlier in this section. Remember that protocol packages can only be exported to SLIMS Share, and cannot be exported to a file. The package will be exported to your SLIMS Share database and can then be imported in other workflows in the same instance or in another of your SLIMS instances.

Error checking:

- Package of kind Workflow Protocol cannot contain workflows: You probably have selected a workflow entity and not a protocol entity.
- Package of kind Workflow Protocol cannot contain more than one workflow protocol, you have selected... : The error lists the protocol entities you have selected.
- Package of kind Workflow Protocol cannot contain more than one workflow protocol, you have selected 0: There is not enough information in a workflow protocol package so it cannot be created.

Importing the workflow protocol package is not performed in SLIMS Share, but instead in Workflow Management. This is because the protocol needs to be imported into a workflow directly so that SLIMS knows how they are connected.

To do so, perform these steps:

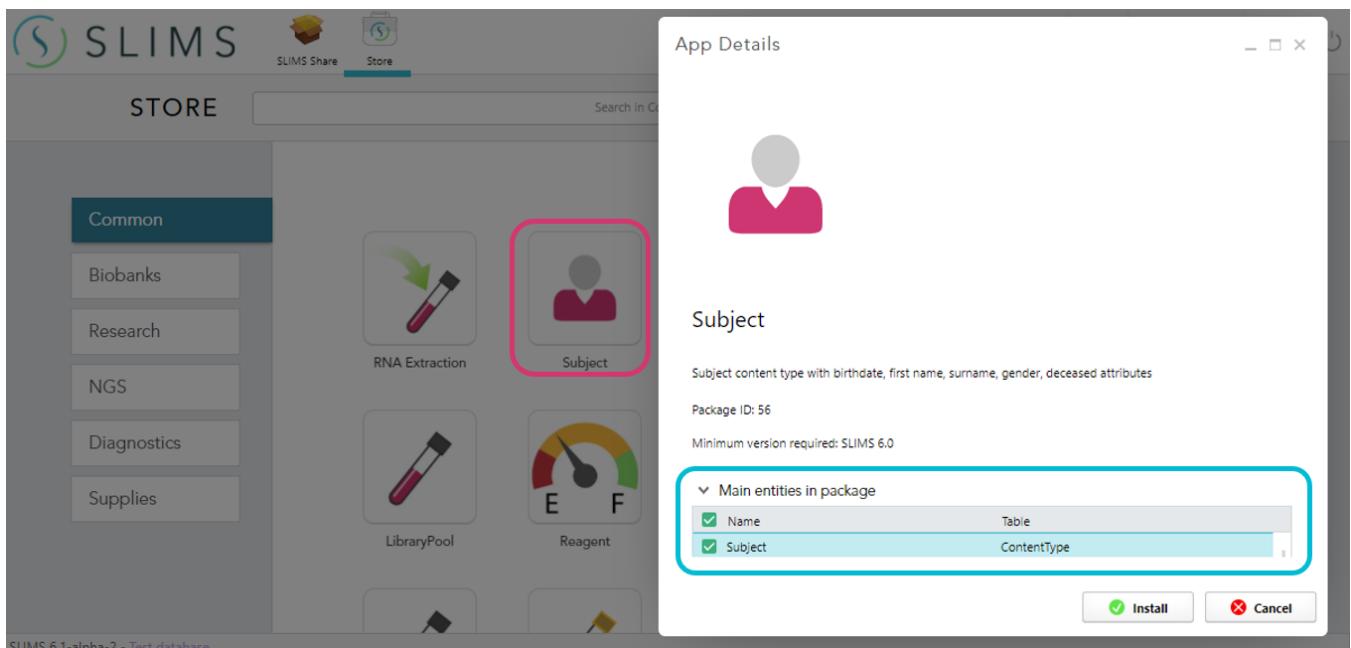
1. Select a workflow, or create a workflow in the Workflow Management module.
2. Click on the "Import Protocol" button on the Workflow Graph tab.
3. The Import Protocol pop-up window allows you to define the start and end queues just as if you were adding a protocol from scratch.
4. Below that are the available packages: all of the workflow protocol packages that have been exported to your SLIMS Share database, and that are available for import into SLIMS. Select the desired package and click the "Confirm" button.
5. The protocol will be added and connected to the new/existing start and end queues you defined.

The same protocol package can be imported on more than one workflow. Also, the same protocol package can be imported more than once in the same workflow. SLIMS produces unique identifiers for protocols and protocol steps, so the entities have a unique identifier going into the package. When SLIMS imports the protocol package into a workflow, it regenerates the unique identifiers to ensure that they will be unique even if the same package is imported more than once. This means the unique identifiers for the protocol and protocol steps could be different in the target workflow than they were in the original package.

### 10.12.3. SLIMS Store

The SLIMS Store module is an alternate way to browse and import packages which has a more user-friendly interface. The user will see icon representations of packages that are available to import. Packages are searchable and are categorized by type. Users will not be able to add a package to what is available on the Store, but requests for packages can be submitted to SLIMS Support. Users can see everything that is available by going to [store.genohm.com](http://store.genohm.com), but it is not possible to download the packages directly from this website. The store contains an icon representation of only the packages that are available to import.

**Figure 10.13. Import Using Store**



### 10.13. Layout

Every user in SLIMS has their own personal Layout where modifications they make are saved automatically, such as when the user adds a view in the Content Grid, adds favorites to the bar, etc. Every time the user logs in, SLIMS uses the same view that was active when the user last logged off.

In addition to the personal Layout, a 'Master Layout' can be defined for a certain Role so all users assigned to that Role share the same Layout. Layouts can be defined for specific Roles, so that a simplified interface with only the necessary buttons and lists can be tailored to suit that Role.

There are two related role permissions required for updating a layout. Users cannot access personal layout configuration (allows the actions: create, delete, or restore to default) without the "Layout Configuration" Functionality Access Right under the Account Settings section. In order to access the Layouts Module to create and manage Master Layouts, a user needs the "Layouts" Access Right under the Miscellaneous section.

### 10.13.1. Add Master Layout



A new Master Layout can be created from the Layouts Module  by using the 'Add New' button located at the top left of the window.

When creating a master layout you can choose a role to see SLIMS in. Your actual role will not change while impersonating and changing data is discouraged and not allowed. An error will appear if you attempt to create, edit, or delete records while impersonating. Additionally, impersonation will end automatically after one hour of inactivity. The User button at the top-right of SLIMS will change to blue while impersonating and show the role you are impersonating as a subtitle in parentheses.

### 10.13.2. Alter an existing Master Layout

An existing Master Layout can be altered by right-clicking on a Layout and selecting 'Alter master layout.'

When updating a master layout you can choose a role to see SLIMS in. Your actual role will not change while impersonating and changing data is discouraged and not allowed. Additionally, impersonation will end automatically after one hour of inactivity. The User button at the top-right of SLIMS will change to blue while impersonating and show the role you are impersonating as a subtitle in parentheses.

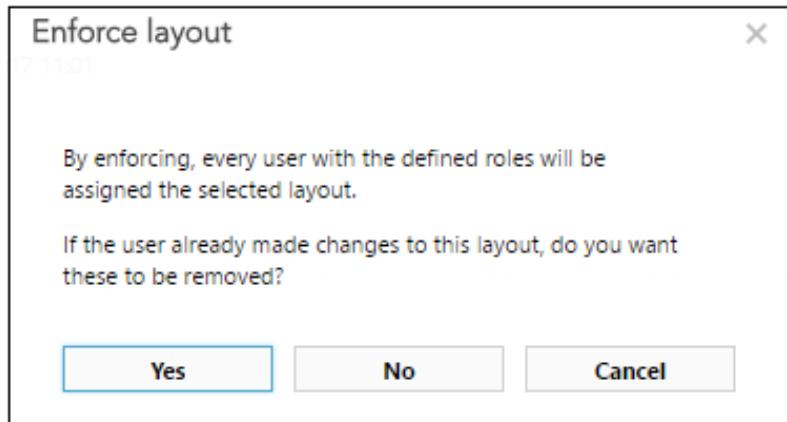
The Master Layout stores the configuration of multiple components: Content Grid, Views, Orders list, etc. When a user changes one of these components, like adding a new column in the Content Grid, this modification will apply only to that user. In other words, the current user's layout diverges from the master layout, but only in the Content Grid where the changes were made. This is represented by the User 'x' in Figure 10.14 (Alteration 'b' in Content list). Therefore, if the Content list of Master Layout is changed (Alteration 'a' in Content list), this change won't overwrite the alteration that User 'x' made to their own layout.

Rules concerning layout alterations are summarized in the following points:

1. Everything that is added in the master layout (new layout components) is added to all users using this layout. For example, Views that are added in the Master Layout are added to all users' layouts. However, existing components that were already changed by the user will not be overridden (like the view they adjusted personally).
2. When altering an existing view in the master layout, select  Enforce layout in the right-click menu to propagate the changes to all users using this layout.

### 10.13.3. Enforcing a Master Layout

Enforcing is used to apply changes that have been made on a Master Layout to all user layouts (see Figure 10.14). When enforcing a layout, the user is presented with the following pop-up window:

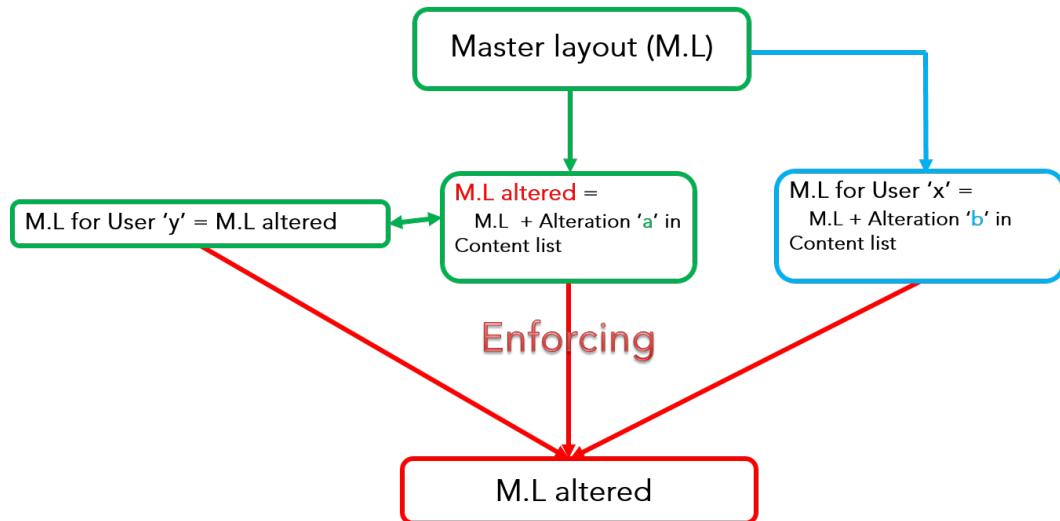


**Yes:** This will be the layout for users of the selected roles the next time they log into SLIMS and all personal changes to their existing 'personal' layout will be removed so that the layout is consistent across all of the role users.

**No:** This will be the layout for users of the selected roles the next time they log into SLIMS, but their personal changes to the layout will not be overwritten.

**Cancel:** Cancels the enforce action.

**Figure 10.14. Alteration and Enforcement of a Master Layout**



#### 10.13.4. Promote Changes

After a user with appropriate permissions makes changes to their layout (for example adding a new view in the content module), the user can copy these changes so they are implemented in the master layout. Promoting changes to the master layout will override layout components of the master layout with layout components of the selected layout.

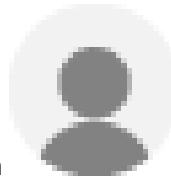
Users can promote changes by right-clicking on their username in the Layouts Module and selecting "Promote Changes."

### 10.13.5. Transform into New Layout

Transforming into a new master layout will copy all layout components into a new master layout.

Layouts can be transformed by right-clicking on a user's username in the Layouts Module and selecting "Transform into New Layout." The Name field is provided and prefilled with the name of the copied layout with "(Copy)" appended to it and an automatic suggestion for the unique identifier field. These can be updated before saving the new layout.

### 10.13.6. Apply an Alternate Layout



To change the active Layout, click on the user button in the top-right of the SLIMS window to open the User Menu. Then, click on the  Switch Layout icon. This opens the Layout configuration window which lists all the layouts available based on the User and their Role. Select the desired Layout and click the "Switch current layout to this layout" button . This will reload SLIMS and apply the desired layout.

### 10.13.7. Configure Visible Fields in a Layout

Fields in most forms (New Content, Macro execution, etc.) can be hidden using the Layouts Module. This allows users to configure the information on screen so that only the fields relevant to a certain task show up. To do this, begin altering the the desired Master Layout as described in Section 10.13.2. Then navigate to the desired form and click  Field Configuration. Here, the user can select what fields should be shown in the form for the Layout currently being edited.

⚠ Adjusting fields with this setting will give a global change to these forms, which can lead to hiding fields in places where they are needed. E.g. the details tab of protocol run steps, where a field is needed to advance in one step, but not in the step where it was set to hidden.

## 10.14. Plugins Module

A Plugin is a small piece of code that is responsible for one functionality. For example, a plugin can handle logic to connect to an instrument, implement a piece of specific business logic, or connect to a specific portal. Plugins do not directly affect each other and work independently to perform their own specific tasks. Plugins are imported from SLIMS Share by a SLIMS engineer so they can be viewed, modified, reset, activated, and deactivated in one place. Plugins either run directly in SLIMS or run in SLIMS GATE.

You will need to take the SLIMS API Training before you can use this module and create and use Plugins in your SLIMS instance. Please contact your SLIMS Engineer to request a training session.

## 10.14.1. Plugin Access rights

This module requires the following Role Access rights to work:

**Table 10.13. Plugin Access rights**

Access rights Category	Access rights Name	Purpose
Setup > Plugins	Can Edit Configuration of Plugins	Provides the role user with import and editing access to Plugins in the module.
Setup > Plugins	Plugins	Provides the role user with access to the Plugins module so they can view information about configured Plugins.

## 10.14.2. Plugin Module Layout

The Plugins module consists of a list of created Plugins in the left panel and additional tabs in the right panel. The left panel provides view and filter options, and the "Active" status column. The tabs on the right provide:

- **Plugin Tab:** This tab shows the details of the Plugin selected on the left.
  - *Name:* The name of the selected Plugin that was assigned on its creation. Can be modified, but needs to be unique.
  - *Type:* This displays the type of Plugin so you can see what it runs on and what dependencies it has (see below list of types).
  - *Configuration:* This displays the Plugin configuration in YAML format. It can be modified directly in this text box within the SLIMS UI without needing to upload another file.
  - *File:* When a Plugin requires an update but has already been created, upload the compiled Plugin (new version) .jar file to replace the old one.
  - *Active:* When activated, this Plugin will be available wherever Plugins can be used. When deactivated, this Plugin will no longer run or be accessible.
  - *Reset Configuration to Default:* Default values can be specified in the Plugins configuration file. Since you can make changes in the text editor box for the configuration, the Reset button is available if you ever need to restore back to the default values.
  - *Reset Values of Plugin Dependencies:* Plugins can be configured so they require one or more fields of their dependencies to have a particular value. When starting a plugin, all the required values of the dependencies are checked before running. An error message will be provided and the plugin will not be executed if some values do not match. When the Reset Values of Plugin Dependencies button is clicked, a pop-up window with a grid displaying the old value and the new value of the requirements is shown. The old and new values can be clicked to expand them if they happen to be large values.
- *Update Values:* The Update Values button is provided in the Reset Values of Plugin Dependencies pop-up. After the values have been reviewed, clicking the Update Values but-

ton resets the values of the fields to what the plugin specifies so the plugin can be executed.

- **Log:** A text box that shows any information, warnings, messages, and outputs when the Plugin is run or tested. The "Restart" button is available to restart the Plugin on demand.
- **History:** This contains the audit History of events related to the Plugin, such as creation, update, runs, and deletion. The list of changes captures the before and after configuration changes.
  - *Show Updates Without Changes:* This can be enabled to filter the list and make it shorter if there were events created that did not have any changes, or disabled to show the full list.

### 10.14.3. Types of Plugins

Three types of Plugins exist that can interact with SLIMS:

- **SLIMS:** This type of Plugin is a Vaadin Plugin used for custom UI components. They run inside SLIMS, which means that SLIMS has to be running for the Plugin to work.
- **SLIMS GATE:** This type of Plugin is a SLIMS GATE Plugin used for custom automation or instrument interaction (such as importing weight values from a balance into a protocol step). SLIMS GATE Plugins require that SLIMS GATE is running.
- **Remote:** This type of Plugin is a remote Plugin that interacts with a local daemon, and is typically used for SLIMS cloud installations. This daemon is normally set up by the SLIMS engineer during the cloud installation. The daemon makes it possible for the customer's local server to interact with their SLIMS cloud instance.

## 10.15. Printing

The use of barcode scanners and printers is recommended for the SLIMS software. This section explains how to configure printers and design label templates to use with SLIMS.

The following Role Access rights can be configured for use in this module:

**Table 10.14. Required Permissions for Printers and Label Templates**

Access rights Category	Access rights Name	Purpose
Setup: Printer	Printers	This permission allows the role user to access the Printers module where label printers can be configured to use with SLIMS.
Setup: Printer	Label Templates	This permission allows the role user to access the Label Templates module so that label templates can be created that can be sent to a label printer. The Label Templates and Printers modules work in conjunc-

Access rights Category	Access rights Name	Purpose
		tion, as labels can't be printed from SLIMS without a connection to label printers.

## 10.15.1. Printers

In order to use the printers, the SLIMS server needs to be installed within the same network range as the lab printers and computers (either through physical network connects or via VPN bridges). SLIMS requires the printer IP address for the setting. This information should be provided by the IT department of your organization.

① Label printers should only be used by SLIMS and not shared with other software because SLIMS controls the settings of these printers. If a user wants to use the label printers with other softwares anyway, then they should use "scripted" label templates rather than label templates designed by SLIMS.

Printers are set in SLIMS using the Printers module in the Setup menu. A new printer is added using the add  button available on the top left of the window and then following the two steps described below:

- First, the printer type has to be selected. Supported barcode printers by SLIMS include:
  1. Brady BBP11/BBP12
  2. Brady BBP33
  3. Brady IP
  4. PRINTMATE
  5. Zebra
- In the second step, the user is asked to enter the following information:
  - *Name*: The name that SLIMS refers to this printer as. This should be recognizable to personnel.
  - *IP Address*: The printer IP address on the network.
  - *Printer Resolution*: The resolution defaults to 300dpi but can be changed to match the lab's higher or lower resolution label printers. This field is only available for Brady BBP11/Brady BBP12 and Zebra printers. Labels will be printed in the resolution set by this field.
  - *Density*: Refers to the ink density used for the label. Should contain a whole number from 0 to 15, with 0 being the lightest and 15 the darkest level.
  - *Possible Label Templates*: Specify which labels this printer can print. Only specified labels will be available on the list of labels related to this printer.
  - *Self Peeling*: Enable this option if the printer is equipped with a self peeling mechanism.
  - *Only for Selected Roles*: Enable this option if the printer is accessible only for users having a certain role.

- *Cutter unit available*: Enable this option if the printer is equipped with a cutter unit for labels.
- *Scripted*: See following paragraph for further details

Jobs sent to the printer can be viewed in the Printer job tab in the right window. In addition, unfinished jobs can be cancelled by right-clicking on a specific job.

## 10.15.2. Label Templates

Label Templates can be added or edited in the Label templates module available in the setup menu. To add a new label template, use the add  button available in the top left window.

### 10.15.2.1. Standard Label Templates

The printer type has to be selected before choosing any option concerning the label template. The label template can be used as a value table in a dynamic choice field to provide selection of the label template on forms. The details about the label templates in the field can be displayed if "Allow to view detail of the referenced record(s)" is enabled. The following label details have to be provided in the label template for every type of printer:

- *Name*: The name that SLIMS refers to this label template as.
- *Target*: The table of records that will be used to print labels with this label template.
- *Possible printers*: Specify which printers can be used with this label template. ⓘ If no printer is specified, the user will not be able to select this label template while printing.
- *Only for selected content types*: Allows the content types that can use this label template to be selected.
- *Active*
- *Scripted*: See following paragraph for further details
- *Only for certain roles*: Restricts usage to the selected roles.

The following parameters concern all the printer types except PRINTMATE:

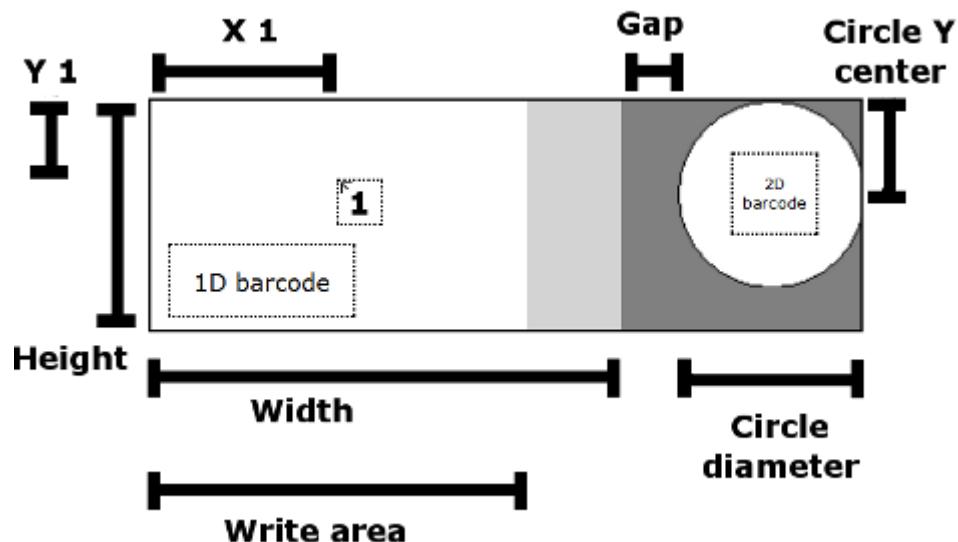
- *Width and Height (mm)*: The label dimensions. This information is available on the label package or the provider web page. ⓘ If this data is not as accurate as possible, shifts will be observed over time in the printed text on the label.
- *Labels/Row*: Number of labels present in a row of the label roll.
- *Landscape Orientation*: Depending on this option, the printer will print the label in landscape or portrait orientation. ⓘ Label template design will always be in landscape mode. For instance, labels with multiple rows are usually in a portrait orientation but the design will be in landscape mode.
- *Restricted Write Area*: Restrict the area where text can be printed on the label.
- *Vial Top*: Used for labels containing a small round label that is applied to the vial top in addition to the rectangular label that will be used on the side of the tube. When enabling this op-

tion, additional information concerning the vial top position and diameter have to be provided.

A last parameter can be set for printer type BRADY IP, the vertical offset: vertical shift of values.

The following image shows a visual representation of the label template:

**Figure 10.15. Label template parameters**



⚠ The graphical representation provides a quick and easy way of modifying the elements position on the label and estimates their distribution, but these are a best guess and not guaranteed to be accurate. It is recommended to print test labels and adjust the parameters according to the output before committing to printing finalized labels.

Once the configuration of the label template parameters are finished, label elements have to be included in the label template.

A label element represents the text contained in one of the record custom fields. Each label element can be printed in a text, 1D barcode, 2D barcode-PDF 417, or 2D Barcode-Datamatrix format. After selecting a format when adding a label element, additional information will be requested according to the format that was selected:

Format	Description
X and Y position	All Defines the element position on the label. Using the graphical representation of the label template, elements can be dragged and dropped to adjust their position in the label.
Format	All Available formats depends on the data type of the selected element (field): <ul style="list-style-type: none"> <li>• <i>Time and Date</i>: Different formats are available for selection - HH:mm or HH:mm:ss</li> </ul>

Format	Description
	<ul style="list-style-type: none"> <li><b>Quantities:</b> Specify the number format (ex: scientific) and whether to print the unit or not.</li> <li><b>Others:</b> Two formats are available: either to print only the value: "&lt;Value&gt;" or to include the text 'Label' before it: "Label: &lt;Value&gt;"</li> </ul>
Font Size	Text The size of the text on the label with 10 being the largest that can be set and 1 the smallest. Depending on the exact printer type, font 6 may be smaller than 5, and 9 smaller than 8.
Height	1D Barcode (BBP11, Brady IP, Zebra) and Barcode (BBP33)  1D barcode (BBP11, Brady IP, Zebra) and 2D Bar- code-PDF 417 (Brady IP)
Narrow Element	The space in between the different bars of the 1D barcode. The smaller the space, the smaller the barcode.
Barcode Type	1D Barcode (BBP11, Brady IP, Zebra) and Barcode (BBP33)  2D bar- code-PDF
Bar Height	417 (BBP11, Brady IP, Zebra)  2D bar- code-Datam- atrix
Dot Size	Any value can be entered.
Module Size	2D bar- code-Datam- atrix (BBP11, Zebra)
Module Width	2D bar- code-PDF 417 (BBP11, BBP11 and is selected from a fixed dropdown list. Zebra)

The list of Label Elements with their parameter values is available on the right window below the visual label presentation. Parameters of a label element can be edited by double-clicking or by opening the element details (right-click > open details).

The *Value Expression Label Element* allows users to define their own label elements, that are not tied to a single record value.

Apart from selecting "text" or one of the barcode options as with other elements, the user also gets a text field such as in the static text element. The contents of this field will be parsed as groovy before being sent to the printer. Apart from standard groovy syntax, the user can obtain the value of any field of the record being printed.

An example of a script:

```
if (content.cntp_name == "chemical") {
    return content.cntp_id;
} else {
    return content.cntp_id + " (" + content.cntp_name + ")";
}
```

will print the content id for chemicals and the content id and type for other types. It is possible to create much more complex scripts as well.

## 10.15.2.2. Scripted Label Templates

⚠ All scripts in this section are made for the Brady IP 600. Script syntax will differ from printer to printer.

Scripted labels are a new way to define labels in addition to the current label editor included in SLIMS. A scripted label template works by sending the output of a value expression in groovy straight to the printer, without any alteration. In its simplest form, this allows the user to paste the output script of other label printing software straight into SLIMS and replace certain values with groovy expressions.

First of all, a user would need to create a new printer in SLIMS. The "scripted" checkbox for the newly-made printer should then be checked.

⚠ If a printer will be used for both scripted and regular labels, the printer will have to be added to SLIMS twice – once as a scripted variant, and once unscripted. This ensures that both label types can be used, but acts as two different printers.

Second, the label template creation form also has a "scripted" checkbox, which allows the user to create a label for scripted printing. The label design screen looks very different for scripted templates, as there is no label visualizer, and no label element grid. Instead, there is a large "script" textfield which should contain the complete printer script.

An example of a script:

```
"""m m
J
H 100
S 0,0,25,27,50.0
O R
T 5,23,90,3,pt 10;"Hello"
```

```
A 1""""
```

will print a single label with the word "Hello."

Because this is using Groovy, it is possible to replace "Hello" with attributes such as the content id:

```
"""m m
J
H 100
S 0,0,25,27,50.0
O R
T 5,23,90,3,pt 10;${content.cntn_id}
A 1""""
```

These are the basic scripts for one label.

It is possible to print multiple labels on one row. To do that, the labels/script value should be set to the number of labels the user wants to print on each row. If labels/script is not set or is set to one, the script will not work with the multi-label syntax shown below. If labels/script is set to a value greater than one, the syntax for the label will change. Instead of using, for instance, `content.cntn_id`, the user must now use `contents.get(0).cntn_id`, `contents.get(1).cntn_id`, and so on. In addition, there is an additional constant `max`, which contains the number of labels being printed at the same time minus one. In other words, it contains the current highest label number. The number of labels being printed at once is never greater than the labels/script value, but it can be smaller if only a few labels are queued for printing with that script.

An example of a script that prints multiple labels:

```
s="""m m
J
H 100
S 0,0,25,27,50.0
O R
"""

for(i in 0..max)
{
    s += """T """ + ((10*i) +5) + """,23,90,3,pt 10;"""+ contents.get(i).cntn_id + """
"""
}

s += """A 1"""
return s
```

The string `s` is initialized with the first part of the earlier scripts, which tells the printer how big the label is and which printer options to use. Then, the for-loop adds the same line once for each label: the line starting with `T`, as in the earlier examples, but using `contents.get` instead of `content`, and with the first number on that line replaced by an expression `((10*i) +5)` that increases for every label. This will print the first label at an x-coordinate of 5, the second one at 15, then 25, and so on. In the end, the label is finished with a last printer command, and the string is sent to the printer.

① Note that the above script will generate the same label as the second script in this section if only a single record is printed.

At the moment, two additional properties are declared: `logger`, which returns a logger that can be used to debug scripts, and `authorizedUser`. `authorizedUser` is label-specific,

meaning you will need to use `locations.get(2).authorizedUser` to get the user that started printing label 2, for instance.

Example script:

```
s="""m m
J
H 100
S 0,0,25,27,50.0
O R
"""

logger.info("started loop")

for(i in 0..max)
{
    s += """T """ + ((10*i) + 5) + """,23,90,3,pt 10;"""+ contents.get(i).authorizedUser + """
"""

}

logger.info("loop completed")

s += """A 1"""
return s
```

will print labels with just the usernames of the users that requested the different prints, and log some extra info.

Of course, sequences can be used in these scripts, but as with value expression label elements in regular labels, the sequences will increment every time the label is sent to the printer, not every time it is printed, so they should be used with care.

### 10.15.3. Calibration

Calibration should be done for each new printer or upon changing the label template in a printer. The following is a short guide on how to calibrate a printer of type **Brady BBP11** to use with SLIMS. For more details outside of calibration for SLIMS, please refer to the printer/labels provider.

According to the type of calibration and preferences, the printer can be calibrated using one of these two methods:

- *Using Printer Software:* Printer software can be used to read printer/label configuration and calibrate them. It is accessible by entering the IP address of the printer on a browser which gives the user access to the printer setting page.
  - Calibrating label offsets is required to get labels printed correctly and it helps having the graphical representation similar to the real printed labels. The calibration can be achieved by editing the X and Y offsets in the software.
- *Using Printer Feed Button:* Printer Sensor Type can be set manually on the printer after changing a label type. This is done by holding Feed button when turning on the printer and release on a certain blink color:
  - *Black Mark Sensor:* During Green/Amber blink (4th blinks series)
  - *Gap Sensor:* During Red/Amber blink (5th blinks series)

## 10.15.4. Print Labels

Labels can be printed using the action "Print labels" available from the right-click menu on a record or from the Selection menu for multiple selected records. The user is asked to select the printer, label template, and number of labels. If the printer includes a cutter unit, the check-box for cutting after the last label will be available and 'true' by default.

## 10.16. Rules

Rules allow various ways of checking data entered by users in SLIMS. The evaluation of a rule can take place in several occasions, depending on the type of rule. A generic example is a rule which is defined to check that the value entered in a field fulfills some condition (see Section 10.16.1). Another example is a rule which is defined to check if the results of an experiment are within the theoretical limits (see Section 10.16.2). In both cases, the rule is evaluated when the value is entered. Depending on the rule type, the outcome of the evaluation is either stored in a rule evaluation or shown in a pop-up window.

The following Role Access rights can be configured for use in this module:

**Table 10.15. Required Permissions for Rules Module**

Access rights Category	Access rights Name	Purpose
Setup: Miscellaneous	Rules	This permission provides access to the Rules module where the role user can define conditions and configure events such as warnings, errors, notifications, and emails that will be triggered whenever the conditions are met. Rules can be used to evaluate data entered in SLIMS, raise the user's awareness, and prevent errors.

SLIMS contains 6 types of rules:

- Conditional value expression, which is the most general and widely used case;
- Result, specific to results (test) of experiments linked to an order whose order type is part of the module Order management;
- Status, evaluated when the status of a record (Order or Result) is updated;
- QC, used to check QC results;
- Compare results, used to compare the same result obtained in different orders whose order type is part of the module Order management.
- Mix, used to restrict things about ingredients, such as checking for the correct content type or expiration date.

A rule basically has two parts:

- **Trigger Condition:** The condition that will trigger the evaluation of the rule. This depends on the rule type:
  - *Status:* when a result has a status.
  - *Result and Compare Result:* when a result is filled in.
  - *Conditional Value Expression:* custom defined by a groovy expression.
- **Rule Evaluation:** The actual evaluation of the rule defined by a groovy script.  
① *Feedback:* there are several ways to give feedback about rule evaluation:
  - *Warning or Error* messages can be displayed to the user. *Error* messages block the transaction, *Warnings* do not.
  - *Notification* can be sent.
  - *Email* can be sent.

Please refer to the 'Groovy Scripting' chapter for details with examples on how to write Groovy scripts and the built-in SLIMS methods and variables.

The *Rule Evaluation Unique Identifier Mask* generates a root name and incremented number based on the mask RLVL##### for each rule evaluation that is generated from the rule, or a mask customized by the administrator. See the Name mask (use # for numbers) definition for more details.

## 10.16.1. Conditional value expression

The "Conditional Value Expression Rule" will be evaluated on each add or update of a record of the specified table if the "Conditional Value Expression" returns true. Conditional Value Expressions can be evaluated on the tables: Case report form, Content, Content event, Content event type, Content study, Content type, Customer, Disease, Experiment, Location, Instrument, Product, Provider, Request, Requestable, SOP, Source, Study, Order, Result, Project, Printer job, Note, Protocol run and Protocol run step.

A rule of type Conditional value expression can be defined by the following steps:

1. Create a new rule and give it a name.
2. Select type Conditional value expression.
3. Select the table to which the rule applies.
4. Specify the Conditional value expression. The 'Conditional Value Expression Rule' will be applied on each add or update of a record of the specified table if the "Conditional value expression" returns true.
5. Specify the Value expression.

In the Value expression field, one can generate error messages with the method `error('field', 'error message')`. The first argument of the `error` function is the

field it refers to. The second argument is the message displayed. When a rule fails, the add or update will be aborted and the generated error messages will be shown in the according fields of the form.

EXAMPLE 1: the table is Content, the Conditional value expression is

```
return content.cntn_cf_notCiao == 'ciao'
```

and the Value expression

```
error('cntn_cf_notCiao', 'cntn_cf_notCiao can not be ciao')
```

An error with message 'cntn\_cf\_notCiao can not be ciao' will be generated whenever the word 'ciao' will be assigned to the custom field `cntn_cf_notCiao` and the actual assignment will be prevented.

EXAMPLE 2: the table is Experiment Run Step, the Conditional value expression is

```
return "DONE".equals(experimentRunStep.get("xprs_status"));
```

and the Value expression

```
if (experimentRunStep.get(xprs_cf_certainField) == null) {
    error("xprs_cf_certainField", "Field is required on status Finished");
}
```

It will not be possible to finish an experiment run step if the field `xprs_cf_certainField` is not filled. An error with message "Field is required on status Finished" is generated.

## 10.16.2. Result

A Rule of type Result is defined to check the value of a result or to compare the value of two results (or more) referring to the same content. An execution of a rule will lead to rule evaluations for every result on which it is executed.

The following Role Access rights can be configured to use this related module:

**Table 10.16. Rule Evaluations Module Permissions**

Access rights Category	Access rights Name	Purpose
Routine: Rule Evaluations	Rule Evaluations	This permission enables the Rule Evaluations module which contains the list of rules that were triggered on content in a workflow and the ELN. They can be repeated, acknowledged, and opened from this module.

A rule of type Result can be defined by the following steps:

1. Create a new rule and give it a name. (Filling a name also automatically fills the unique identifier.)

2. Enter the # of pulls.
3. Select type Result.
4. Choose a mode of evaluation from the choices: Match Specification, or Script Specification.

No Evaluation rules are available in the Products and Specifications module and Stability Study Design module, but are not relevant for the Rules module. Match and Script Specifications are explained below.

5. Select the test(s) the rule is about. The rule will be evaluated only if results of all selected tests are available for a content record. If the rule is only about a single test (only one test is selected) and is for an analytical workflow, then the rule will always evaluate regardless of whether the results are created manually or created from requested results (by a technician or by plugin automation), or are results that are repeated.
6. Rules can be restricted to certain content types.
7. Fill the configuration of the rule depending on the selected mode of evaluation.
8. If "Expected Value" is shown, then enter the acceptable result value for the rule. If the user enters a value outside of the range or expected value, a failed rule evaluation will be triggered.

### 10.16.2.1. Match Specification

Match Specification mode of evaluation is a way to write a simple rule that is performed on a test. When a test is selected with this rule configuration, an Operator dropdown becomes available. The operator options in the list are filtered based on the datatype of the test (ex: quantity, decimal number, and whole number test datatypes have "Between (inclusive)" as a selectable operator, but a fixed choice test does not).

The operator options depend on the type of test:

**Table 10.17. Operators available for rules**

Operator	Applicable Test Datatype	Behavior
Is	Checkbox	The rule can have expected value of true or false. Enable expected value to indicate that a checked box is the acceptable result and an unchecked box should trigger a failed rule evaluation. Disable expected value to indicate that an unchecked box is the acceptable result and a checked box should trigger a failed rule evaluation.
Is one of	Fixed Choice, Dynamic Choice	The rule can have an expected value of one or more fixed choice answers. When one of the acceptable values is en-

Operator	Applicable Test Datatype	Behavior
Between (inclusive)	Quantity, Decimal number, Whole number	tered as a result, the rule passes, but if any of the unacceptable values are entered, the rule evaluation fails. Dynamic choice works the same way, but allows a rule to be created with reference data records as the choice options. The rule can have an expected value of any of the reference data records on the selected reference data. If the user enters an unacceptable value, a failed rule evaluation is triggered.  The rule can have a minimum value and a maximum value. If a result is entered that falls outside of that range, a failed rule evaluation is triggered.

## 10.16.2.2. Script Specification

Script Specification mode of evaluation is a way to script a simple or complex rule that is performed on one or more tests using Value expression. Documentation is available in the 'Groovy scripting' chapter.

For example if you want to check that the value of test A is smaller than test B you have to select tests A and B in the definition of the Rule. To write the value expression for the rule you have access to the content under the variable `content` and to the full result values. The result variables are named after the test. So the value of the result of the test A is `A.rslt_value`. Additionally, the test is also available in the value expression `A.test.test_name`. The first argument of the `warn/error` function is the test it refers to. The second argument is the message displayed in the rule evaluation list. For example: `warn('A', 'A warning about A')`. Only one notification per test can be done. When multiple warnings or multiple errors are reported in the value expression only the most recent one will be kept in SLIMS. If both errors and warnings are given, the errors have precedence. When the value expression has an exception while executing (for example due to a syntax error), the rule evaluations of all the results will be `ERROR`. The message will be the exception.

An example rule is

```
if (A.rslt_value < B.rslt_value)
  warn('A', 'A can not be smaller than B')
```

This will result in two rule evaluations: one warning on test A with the message 'A can not be smaller than B' and one 'OK' on test B.

You can view the rule evaluations in the rule evaluation section, by right-clicking on the result and selecting 'view rule evaluations.' There you can see the rule evaluation with the content, test, rule, and result associated to it.

The following methods are used as a possible outcome of the rule evaluation:

- **OK.** When nothing is specified in the rule or the `ok` method is used, the evaluation will be OK.
- **WARNING.** This expression in the value expression specifies a warning: `warn('A', 'A warning about A')`.
- **ERROR.** This expression in the value expression specifies an error: `error('B', 'An error about B')`.

The first argument of the `ok/warn/error` function is the test or field it refers to. The second argument is the message displayed in the rule evaluation list.

If the evaluation of a rule does not give an OK outcome, the result will be marked with a warning or error symbol, but the value will nevertheless be saved (differently than the Conditional value expression rules which don't allow a `wrong value` to be entered).

① Rules are evaluated only if the results belong to a protocol run linked to an Order (generated from a 'Schedule' action in Order management). Protocol runs started in ELN and not linked to an Order are not subjected to rule evaluation. Results that are scheduled via workflow orders are also not subjected to rule evaluation.

If all rules applying to a result are fulfilled, the status of result is automatically *Verified*. When not all rules are fulfilled, the status result is *Available* and the result is marked with a Warning/Error symbol. In this case, several possibilities present to the User:

- *Edit* the result. The modification will be written in the history and will trigger a new rule evaluation. If rules are fulfilled after the editing, the result will become *Verified*.
  - ① A result with status *Available* may be not editable (see section *Statuses*).
- *Acknowledge* the result. This means you are okay with the rule evaluation giving a warning or error. You can also enter comments to say why you acknowledge the evaluation instead of changing it.
- *Repeat* the result. A new run will be scheduled with the ability to enter new results. The repetition of the result is reported in the *Events* of the repeated result (the first one).

A result will be promoted to *verified* when all the rule evaluations associated to all its rules are *okay* and/or *acknowledged*. In our example the result on test 'B' will not be promoted to *verified* because there is still a warning on A that needs to be fixed either by making A bigger than B or acknowledging the rule evaluation.

### 10.16.3. Status

A 'Status rule' will be applied on each add or update of a record of the specified table if it has the specified status.

A 'status rule' can be defined in the Rules module. From the Rules module:

1. Create a new rule and give it a name.
2. Select 'Status' type.

3. Select the table to which the status refers (options: Result, Order).
4. Select the status to which the rule applies. The statuses are populated from the default status workflow related to the record type selected in the table dropdown.
5. Write the Value expression. Documentation is available in the 'Groovy scripting' chapter.

## 10.16.4. QC

QC Rules are rules that are run on QC results of one QC card. To create an error here you can call `createError(errorList)` where error list is the list of results that violate the rule. However, most of the time you can use the following functions: `westgard(limit, amount)` and `westgardCont(limit, amount)`. The first one gives an error when the difference between the mean of the card and the result value exceeds the limit times the standard deviation for more than the amount times. For example, the expression `westgard(3, 2)` will give an error when two results are found whose difference with the mean exceed 3 times the standard deviation. `westgardCont` checks the same, but the results have to be on the same side of the mean and have to follow up.

## 10.16.5. Compare Results

Rules of type Compare Results are similar to Result Rules. However, in this case, no test needs to be selected because two result values of the same test of two different orders will be compared.

The rule of type Compare Results applies on two related orders obtained by copying an order and *specifying that the content is derived*. This way the two orders are linked and checks will be run on the pairs of tests that are in both orders. These variables can be used in the Value Expression: `originalContent`, `derivedContent`, `originalResult`, and `derivedResult`.

It is recommended to use the unique identifier of the test instead of the name in the expression in order to trigger the expected warning.

## 10.16.6. Mix Rules

Rules of Mix type can be created to put restrictions on ingredients used in a mix. They can provide a warning to the end user but allow the mix to continue, or provide an error to the user and prevent the mix from continuing. The mix rule is executed when a mix record is created, and then evaluated again every time a mix is modified, such as adding additional ingredients to a mix. From the Rules module:

1. Create a new rule and give it a name. It is recommended to indicate its use with the name.
2. Select 'Mix' type.
3. Choose whether the mix rule applies to protocol steps of mix type.
  - If "Only applicable in selected mix protocol steps" is disabled, then the mix rule is unrestricted and will apply on all mix protocol steps and in the Content module by default.
  - If "Only applicable in selected mix protocol steps" is enabled, a multi-select list is provided with your choice of the mix type protocol steps that are available in the SLIMS instance.

The rule will ignore any protocols that are not selected, and will ignore mixes made in the Content module.

- Also, if a mix protocol step is selected in the multi-select list, then the rule will apply to the selected protocol step as an "Additional rule" automatically.
- Write the Value expression. Documentation is available in the 'Groovy scripting' chapter.
  - The mix rules that apply in a protocol step can alternatively be restricted from the protocol step definition in Workflow Management. If a mix rule is applied to a protocol step definition, then the protocol step will be listed in the applicable mix protocol steps in the rule also. You can refer to the 'Protocol Steps' section for details.

This mix rule example detects when ingredients are expired or are near expiration. A warning is provided if they are near expiration, though the user can proceed with the mix, or an error is shown if they are already expired that prevents the user from proceeding with the mix.

```
def expiredIngredients = ingredients.findAll({it.get("cntn_cf_expired")})

if (expiredIngredients.size() > 0) {
    expiredNames = expiredIngredients.collect { it.get("cntn_id") }.join(",")
    error("Ingredients are approaching expiration: " + expiredNames)
}
warn("These ingredients should be used soon!")
```

This mix rule example checks the number of ingredients being used. An error is provided if less than three or more than five ingredients are used.

```
numberIngredients = ingredients.collect{ it.get("cntn_fk_contentType") }.unique().size()
log.info(numberIngredients)
if (numberIngredients < 3) {
    error("A pool must contain at least 3 different ingredients!")
} else if (numberIngredients > 5) {
    error("A pool can contain no more than 5 different ingredients!")
}
```

## 10.17. Scan

The scanning module allows researchers to execute a Macro in a lab environment with a simple scan of their records and without requiring a direct interaction with the computer.

This module requires the following Role Access rights to work:

**Table 10.18. Scan Module Permissions**

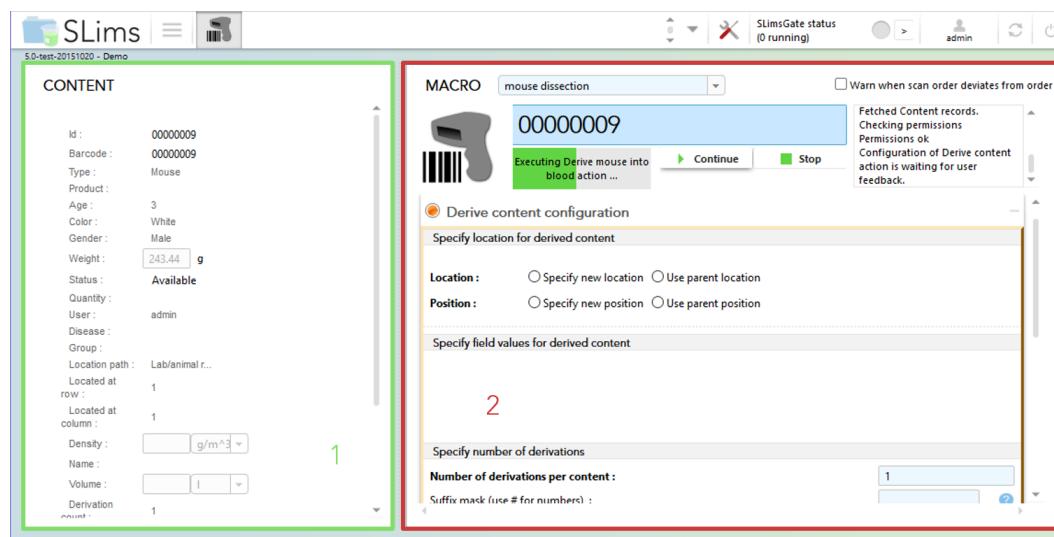
Access rights Category	Access rights Name	Purpose
Routine: Content	Scan	When enabled, the role user can access the scan module to facilitate specific tasks using macros, and requiring less interface with the computer.

Defining the Macro that will execute is the first and main operation to do before beginning to scan. Once the desired Macro is defined, the scan of contents and locations can be started, de-

pending on the Macro that was chosen. The link to the scan module is available under the Routine menu. After each scan, the Macro will be executed automatically until the end unless pauses were specified in the Macro steps.

The scan window is split into two main parts. Figure 10.16 illustrates the different parts in the scan window.

**Figure 10.16. Scan Module**



The first part on the left window shows details about the scanned record. The second window on the right shows the different steps to be executed with the selected Macro. In addition, there is a window describing the full history of actions that were executed during the Macro run that displays the details of any errors if they occurred.

When a Macro needs to be executed on a series of contents, the Warn when scan order deviates from order in location option can be enabled. In this case, the Macro will check that the contents are scanned in the same order as they are in the location (row wise). For example, if three contents are at position (1,1), (1,2) and (1,3), after having executed the macro on content in (1,1), the Macro will warn the user if called to be executed on content in position (1,3) instead of (1,2). This is the same as checking the option Check contiguity in the Macro definition.

When a Macro needs a manual input, the Manual input required can be enabled. This will trigger a focus on the first field in the Macro form when pressing enter (the scanner can be configured to do it automatically).

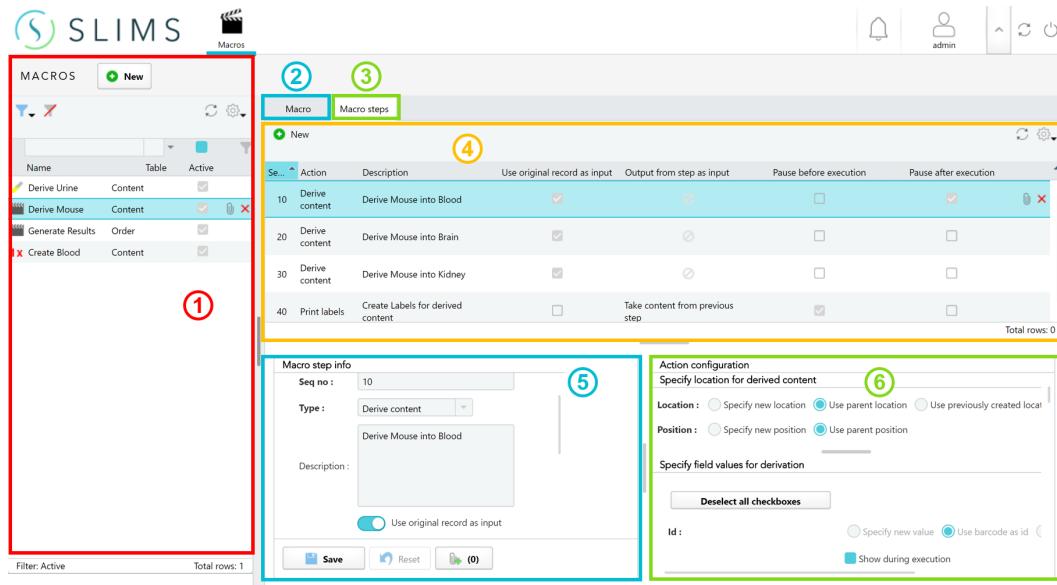
ⓘ Pressing shift-enter simulates clicking the *Continue* button.

## 10.18. Macro

In order to automate procedures that are often replicated (a repetitive series of steps that take up the user's time), SLIMS provides the Macro module where an entire process can be defined in a Macro with different steps. Once this is set up, an entire procedure could be executed with one mouse click.

Even if the definition of a Macro is similar to the definition of a protocol, the module containing the definition of the steps is a little more complicated.

**Figure 10.17. Macro**



The left panel (1) contains the list of defined Macros and the right panel contains the details about the selected Macro. Let's have a closer look into the right panel.

The first tab (2) contains the same options as the pop-up window to create a new Macro:

- **Icon**
- **Name**
- **Description**
- **Table** Two possible values Content or Order. This indicates on which table the Macro is available.
- **Check contiguity** (Available only when selected Table is Content). If the Macro is used in the Scan module on a series of contents, it will check that the contents are scanned in the same order as they are in the location (row wise). For example, if three contents are at position (1,1), (1,2), and (1,3), after having executed the macro on content in (1,1), the Macro will warn the user if called to be executed on content in position (1,3) instead of (1,2). This behavior can be activated directly in the Scan module, checking the option Warn when scan order deviates from order in location.
- **Execute each macro step on all records at once in batch mode** Execute each step on all the selected content at the same time, otherwise execute whole Macro on one content at a time.
- **Hide detail form when executing this Macro** This option is only available when "Execute each macro step on all records at once in batch mode" is disabled and the selected table is "Content." When this option is enabled, the informational Content Details panel that is displayed on the left of the Macro execution window is hidden, and when it is disabled, the pan-

el is shown. Hiding the panel may improve the performance speed of Macros that are responsible for processing many fields at a time. This option can be enabled on new or existing Macros.

- **Auto close the popup window when the macro is finished successfully.** Once all the Macros are executed the window will close automatically.
- **Auto start execution of the macro** The Macro starts immediately upon clicking the related button. If this option is not checked, clicking on the Macro will open the window, but not start the Macro. The Macro can be started clicking on the start button.
- **Only for selected roles** Specify the roles that can execute this macro.
- **Only for selected content types** This option is available if the content table is selected. Allows the Macro to be restricted to specific content types.
- **Only for selected statuses** This option is available if the order table is selected. Allows the Macro to be restricted to orders which have a specific status.
- **Only for use in selected modules** When disabled, the user can select the macro from the list of macros in any module where macros are available. When enabled, the only modules the macro will be available in are the ones that are selected. The modules that the macro can be restricted to are: Electronic Lab Notes, Scan, Content, Flows, Orders, and Dashboard. This selection can help ensure that macros only appear in the modules in which they are relevant.
- **Active** Defines whether the Macro is available for use or not.
- **Number of Executions** This is a count of how many times the Macro has run.
- **Last execution date** This indicates the date that the last execution was run.

① Macros can be restricted to specific protocol steps or executed automatically when a protocol step starts. Additional details concerning the use of Macros in the Protocols are available in the *Protocol Step* section.

The second tab (Macro Steps) contains the definitions of the Macro's steps. It is separated into three parts: the first (4) contains the list of the steps, the second (5) contains general information about the selected step, and the third one (6) contains the configuration of the actions that should be performed in the step.

The step info contains the following fields:

- **Seq no** This is set to execute the steps in the desired sequence, where lesser numbers are performed earlier, and greater numbers are performed later.
- **Type** This defines which action type should be performed while executing this step.
- **Description**
- **Use original record as input** This option will automatically use the original record (as opposed to a record derived from the original in previous steps) entered in the Macro as an input for this step. For example, this option can be used in a step of type 'Remove' to remove the original content after a derivation step.

- **Output from step as input** This option appears whenever the option Use original record as input is unchecked. By default, its value is Take content from previous step, meaning the contents created from the previous step are taken as input for this step. It is also possible to choose content from another step that is prior to the current step and not skippable.
 

⚠ If a step is selected and afterwards that step is made skippable, the link will be lost.
- **Pause before execution** Check this option if the user will need to enter a field's values before starting the step; otherwise, the step will be executed immediately and the field's values will be taken by default.
- **Pause after execution** Check this option if the user will need to check or take notes on the output of the step run. A pause will be added at the end of the step to give the user time to do so.
- **Can be skipped** This checkbox will add a skip button next to the cancel button allowing the user to omit the step during execution.

Each action has its own options, though they are generally the same as while executing the action directly on the content/order. The following table illustrates the different actions.

Action	Content Table	Order Table
Move	x	
Link content	x	
Link content to order	x	
Edit content	x	
Create content	x	
Copy content	x	
Derive content	x	
Aliquot content	x	
Dilute content	x	
Enroll content	x	
Add content event	x	
Add case report form	x	
Print labels	x	
Unremove content	x	
Remove content	x	
Mix content	x	
Execute SLIMS GATE flow	x	
Fulfill request	x	
Add parent	x	
Add child	x	
Create new location	x	
Usage taken from content	x	
Generate report		x

Action	Content Table	Order Table
Receive content		x
Copy order		x
Edit order		x

⚠ A step of type Add case report form can only be added if a previous step of type Add content event is added because the CRF is added to the latest created Content Event associated to the selected content.

After the Macro creation, steps have to be added by using the tab *Macro Steps* where as many steps as needed can be added. A step can be copied by right-clicking on the step and selecting the "Copy" action. Macro steps do not have names, so the pop-up form provides a field to enter the sequence number and unique identifier. A new unique identifier is automatically suggested but can be updated if desired.

**Figure 10.18. Macro Steps**

Macro		Macro steps				
S...		Action	Description	Use original record as input	Pause before execution	Pause after execution
10	Derive content	Derive mouse into blood		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
20	Derive content	Derive mouse into heart		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
30	Derive content	Derive mouse into kidney		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
40	Print labels	create labels for derived contents		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

More customization of a Macro step can be done by using the *Action Configuration* window available under the list of the Macro steps. This part is action-specific and defines which values will be given to the record by executing the Macro. For example, by selecting a move Macro step the location the content records will be automatically moved to after executing this Macro can be chosen.

The tab 'Action Configuration' contains the same action parameters as when the action is executed on the record directly from the Content/Orders module. For example, the action 'Print labels' when executed from the Content module asks the user to choose the printer, the label template, and the number of labels, and the same parameters are available in the macro step of type 'Print labels'. However, the following parameter is only available with macros:

- **Block original position:** Available in a Macro step of type **Remove** (soft-remove). This parameter is also available using a direct action in Content Management, but only with the 'Move' action.

The following choices are available for record fields that can be edited by the Macro action:

- **Keep at current value:** Keeps the field at its current value and doesn't change it (doesn't even calculate the value expression).
- **Specify new value:** Adds a new field where the user can specify the new field value.
- **Use parent value:** Copy the field value from the parent if we are executing a derivation action.

- › **Use value expression:** If this option is selected, the field's new value will be calculated from the value expression that is specified on the field in the Fields module instead of allowing the user to enter a value.
- › **Specify value expression:** This option is available on content fields in these macro steps: Edit Content, Derive Content, Copy Content, Create Content, Create Container, and Mix Content. When enabled on a field in a macro step, the value expression editor opens below to allow the user to enter a groovy expression to calculate a result for the field. When Specify value expression used on a field is executed, the field value will be computed by the value expression written on the macro step even if a default value expression is specified in the Fields module on that field.

The execution order of the expressions depend on the groovy sequence number specified in the Fields module.

- › **Show during execution:** Shows the current field during the execution of the Macro.

This option is mainly used when the user needs to enter a value while executing the Macro and the step option "**Pause before execution**" has to be enabled. Otherwise the value has to be predefined while setting up the Macro step.

- ① Newly created custom fields will not be shown during the Macro execution by default and will have their Value Expression (with Groovy script) calculated in a Macro step like edit, derive, or copy. If no "Groovy script" was specified, the fields will keep their current/parent value.

### Link Content to Order Action Configuration

The Link Content to Order macro has specific action configurations. There are two options: Current Order and Existing Order. During configuration, if Current Order is selected, the order selection grid will not be visible for the user to choose which order to link content to, because it is assumed the user has an order currently selected. If the user tries to use the macro from another module, the macro validation will fail since no order is currently selected. Restricting the macro usage to the Orders module will prevent that from happening. Otherwise, Existing Order can be used to let the user select the order to link the content to.

Link Content to Order steps can also link content that already exists, or content that has been created in the previous step (using "Take Content from Previous Step").

**Execute SLIMS GATE Flow** A Step of action type 'Execute SLIMS GATE Flow' is used to execute a SLIMS GATE flow bean during a Macro execution. However, there are some limitations on the SLIMS GATE flow to consider:

- Only flows with the new usage 'Macro' are allowed.
- Only flows with one step are supported.
- Only flows that execute synchronically are supported.
- Flows with file parameters are not supported.

During the execution of the Macro step:

- The first step of the SLIMS GATE flow is visible in the Macro step.

- If the Macro step is paused, it is possible to fill in SLIMS GATE flow parameters (except files).
- When execution of the Macro step is continued, the SLIMS GATE flow is started with the filled-in SLIMS GATE flow parameters and the input content records of the Macro step as the 'SLIMS\_SELECT\_SAMPLES' SLIMS GATE flow parameter.
- If the next Macro step has 'Use original record as input' disabled, then the output of the SLIMS GATE flow is used as input.

## 10.19. Sequences

The Sequences module is available in the Setup menu under Miscellaneous.

The following Role Access rights can be configured for use in this module:

**Table 10.19. Required Permissions for Sequences Module**

Access rights Category	Access rights Name	Purpose
Setup: Miscellaneous	Sequences	This permission provides access to the Sequences module where the role user is able to view all of the barcode templates that are available in their SLIMS instance and add more options.

This module gives an overview with a list of all sequences available in SLIMS. The following fields are available for a sequence:

- **Sequence Mask:** The mask of the generated sequence. # is used for numbers.
- **Last Sequence Number:** Indicates which number was last used for a sequence. For instance, there is a sequence record with sequence mask: SAMPLE### and last sequence number: 15. The content type Sample has barcode mask SAMPLE##. This means the next sample you create will have barcode SAMPLE016.
- **Type:** Can be 'Barcode' or 'Value expression':
  - **Barcode:** Sequences of the barcode type are not editable and are used for Contents, Orders, and Locations.
  - **Value expression:** The last sequence number field of Sequences that are a Value Expression type are editable.

The Sequences module enables the user to:

- Create a new 'value expression' type sequence.
- Export the list of sequences to Excel.
- Sort the grid with the list of sequences.

① Whenever a content, order, or a location is created with a Barcode Mask that hasn't been used yet, a new sequence record with the barcode type is made in sequences.

Whenever sequences.next() is called in a value expression with a sequence name that hasn't been used yet, a new sequence record with value expression type is made in sequences.

## 10.20. Status

A Status defines any state that a record could be in. Status Transitions are the ability for records to go from one status to another. Status Workflows allow administrators to define the statuses and status transitions for various types of records, allowing for customization and multiple workflows in the same SLIMS. The management of statuses is explained in this section.

This module requires the following Role Access rights to work:

**Table 10.20. Status Workflow Access rights**

Access rights Category	Access rights Name	Purpose
Setup > Miscellaneous	Status Workflow	This permission allows a role user to access the Status Workflows module to add and manage status workflows, statuses, and status transitions. Administrators can enable or disable the access right to allow or prevent access to the module.

### 10.20.1. Status Workflows

In this module, status workflows can be edited and added to set up the management of statuses and status transitions. The Status Workflow module has the "New Status Workflow" button to allow the creation of new custom status workflows and a dropdown of existing status workflows. The existing status workflows dropdown can be filtered to hide inactive status workflows and can be refreshed. The module contains three tabs:

- **Status Workflow tab:** Displays the definition of the status workflow that is selected in the dropdown above.
- **Status tab:** Contains the definition of all statuses that are linked to the selected status workflow.
- **Status Transitions tab:** Contains the definition of all possible status transitions linked to the selected status workflow.

Status Workflows have simple details because the majority of the configuration is done with the statuses and status transitions they contain. It is recommended to create a custom field for each status workflow the administrator creates, as detailed in the Status Datatype section. A field with the datatype Status allows you to assign a particular status workflow to the field.

This gives the administrator a way to restrict record types (content, instruments, orders, results, etc.) to that particular status workflow. Status Workflows can be defined with:

- **Name**
- **Unique Identifier**
- **Description**
- **This status workflow controls the active property:** This option is available on custom status workflows, not on default status workflows. It is set on the workflow, not the statuses within the workflow. When this option is enabled, the Active switch on records that have a status field pointing to this status workflow is read-only and controlled by the status field on the records. The "Records in this status are considered active" option becomes available on the statuses within the status workflow.
- **Active/Inactive switch**
- **Read-only audit fields.**

Some default status workflows exist in SLIMS at the start to support the default functionality of statuses and status transitions that records can go through. This means administrators do not need to set anything up for statuses to function right out of the gate, but they can be fine-tuned for the lab's needs in this section.

① *A note about migrating from an older SLIMS version to version 6.6:* Status Workflows did not exist in prior versions of SLIMS. However, the statuses and status transitions in prior versions have been preserved and migrated into the default status workflows outlined here. Additionally, the Status Workflow permission is enabled for administrators by default, and for a role which existed in the prior SLIMS version and had permissions for both Statuses and Status Transitions. Otherwise the access right will not be enabled for the role.

The default status workflows available in SLIMS have some editing restrictions and contain some specific default fields:

- **Default Content Status Workflow:**
  - Captures all statuses and transitions for content.
  - The status workflow's existing statuses and transitions can be edited and new ones can be added.
  - The default field `cntn_fk_status` is the default *Status* datatype field that is linked to this status workflow. It cannot be unlinked from the status workflow.
- **Default Instrument Status Workflow:**
  - Captures all statuses and transitions for instruments.
  - The status workflow's existing statuses and transitions can be edited and new ones can be added.
  - The default field `nstr_fk_status` is the default *Status* datatype field that is linked to this status workflow. It cannot be unlinked, but the instrument types that use this status workflow can be restricted by editing the field.

- **Default Order Status Workflow:**
  - Captures all statuses and transitions for orders.
  - The status workflow's existing statuses and transitions can be edited, but new ones cannot be added.
  - The default field `ordr_fk_status` is the default `Status` datatype field that is linked to this status workflow. It cannot be unlinked, but the order types that use this status workflow can be restricted by editing the field.
- **Default Result Status Workflow:**
  - Captures all statuses and transitions for results.
  - The status workflow's existing statuses and transitions can be edited, but new ones cannot be added.
  - The default field `rslt_fk_status` is the default `Status` datatype field that is linked to this status workflow. It cannot be unlinked, but the tests that use this status workflow can be restricted by editing the field.
  - Normal result status transitions have a `Normal Results` label, and QC result status transitions have a `QC Results` label.
  - The Lab Setting `"Show groovy configuration for tests"` is required if you want to be able to add value expressions to statuses in this status workflow.

## 10.20.2. Publication Status Workflows

Various publication status workflows exist for each record type that can be versioned. Publication status workflows allow the flexibility of requiring electronic signatures or reasons for change in order to version records of different types: SOPs, Protocols, Product Specifications, or Report Templates. For instance, the SOP publication status workflow could be updated to require an electronic signature when versioning SOPs, and the products and specifications publication status workflow could be updated to require reasons for change instead.

Default Status Workflows exist for each record type that can be versioned:

- *Default SOP Publication Status Workflow*
- *Default Protocol Publication Status Workflow*
- *Default Product Specifications Publication Status Workflow*
- *Default Report Template Publication Status Workflow*
- *Default Stability Study Template Publication Status Workflow*

This section describes the setup of publication status workflows. Each captures all the statuses and transitions for versioning their respective record types: SOPs, Protocols, Product Specifications, Stability study templates, or Report Templates. A Publication Status field exists for each record type. It references the publication status of its own workflow to provide the status

Draft, Published, or Obsolete for the record. (Publication status workflows are tied to their respective record types with a default (recordAbbreviation)\_fk\_publicationStatus field.)

### 10.20.2.1. Publication Statuses

Publication status workflows come with three statuses: Draft, Published, and Obsolete. New statuses and transitions cannot be added to publication status workflows.

The "Draft" status cannot be inactivated, and some options are not editable, but you can define whether records in Draft status can be removed or not. The "Published" and "Obsolete" statuses cannot be inactivated and are not editable.

### 10.20.2.2. Publication Status Transitions

The transitions that are possible are Creation to Draft, Creation to Published, and Published to Obsolete. The existing active transitions can be edited to restrict the users that can perform a transition, or require an electronic signature or reasons for change for all users of the status workflow. All of the other transitions (such as Obsolete to Published) are disabled and read-only so they cannot be edited to protect the fidelity of the versioning process.

The behavior of versioning is detailed in: Create/Edit Versions. In terms of a publication status workflow:

1. Draft is used to add new records and revise. The status transition that creates a draft is Creation to Draft.
2. When all status fields present on the draft are pointing to a status that has "Can version records in this status" enabled, the Version button is available. The current state of the record's draft is versioned into the Published status, using a required signature, comment, or reason for change depending on the status transition configuration and lab settings. The status transition that publishes a draft is Creation to Published.
3. All previously saved versions are Obsolete so they cannot be edited or used. The Draft remains as draft so it can be edited, and if other status fields exist on the record, they are reset to the status selected for "On publication of a new version, reset draft status to:".

① A note about migrating from an older version of SLIMS: Any versionable record will enter its respective status workflow with an intuitive publication status when coming from an older version of SLIMS, such that:

- Outgoing record's version type is Draft > Incoming record has publication status "Draft"
- Outgoing record's version type is Version and is Active > Incoming record has publication status "Published"
- Outgoing record's version type is Version and is not Active > Incoming record has publication status "Obsolete"

## 10.20.3. Statuses

In this tab, statuses can be inactivated, modified, and the behavior associated to a status can be configured. New Statuses for custom status workflows and some default status workflows

can be added as well. For example, the user might want to make Orders that are in the "Registered" status uneditable.

The statuses tab has a grid on the left and form on the right. The grid lists all of the statuses that currently exist in the selected status workflow. The tool bar has a "New Status" button that offers options for creation of new statuses. Selecting an existing status displays its details form on the right. Statuses can be edited or removed as long as there are no associated records (and if they are in a default status workflow, the workflow allows it).

When this configuration is modified on a status, SLIMS will have to be refreshed for the changes to take effect.

The following options are available for *all* statuses:

- **Name**
- **Unique Identifier**
- **Seq No:** Automatically added so the system can track and refer to the status.
- **Description**
- **Can permanently remove records with this status:** Available on all statuses of all status workflows.
  - When enabled, a multi-select dropdown allows the roles, users, and groups that can perform the action to be selected. By default, all statuses allow removing records that have that status. This only applies to customized records since most records do not have any status fields by default.
  - In any list where records that can normally be permanently removed, the remove button is only visible for records that are in a status that can be removed, and only to the user with this permission. If a user, plugin, or REST API tries to remove a record with an unremovable status, an error is provided that prevents the removal of the record.
  - If there are multiple statuses on a record, it can only be removed if all statuses on the record allow the user to remove the record.
- **Only selected users can permanently remove record with this status:** Provides a multi-select dropdown list that allows the administrator to select a combination from the list of users, roles, and groups to whom they want to provide permission.
- **Can edit records with this status:** Controls whether or not updates are allowed on a record that has this status, whether it's a default status or a custom field.
  - Enabled by default. When enabled, a dropdown allows the roles, users, and groups to be selected that can perform the action. If enabled, records with this status cannot be updated by users outside of the selected user groups.
  - When disabled, the edit/details form for the record becomes read-only. The record can still be transitioned if there are valid status transitions available for it, and the status fields of the record are always editable.
  - If the record has multiple status fields and at least one is a status that disallows editing, the record cannot be updated.

- **Only selected users can edit records with this status:** Provides a multi-select dropdown list that allows the administrator to select a combination from the list of users, roles, and groups to whom they want to provide permission.
- **Records with this status can be versioned:** This option is only relevant for record types that can be versioned (SOPs, Protocols, Products and Specifications, and Report Templates). However, it exists as an option on all statuses in case it is needed on some other record types. When this option is enabled on a status, it indicates that records in this status can be versioned - meaning that the version button becomes available when they are in this status. If the record has multiple custom status workflow fields that influence whether the record can be versioned depending on its status in the custom workflow, all of the statuses for the record must allow versioning before the version button will be available. For more detail on how versioning works, see Section 10.25.
  - *On publication of a new version, reset draft status to:*  
When a record is versioned, the status field on that record is reset to the status configured by this option. The dropdown list is populated with the statuses of the current status workflow. Only one status can be selected, and by default, the current status is selected. This option points to "Draft" and is read only on publication status workflows. This option is not available on other statuses of publication status workflows.

- **Active**
- **Status Workflow:** A read-only field that displays the Status Workflow that the status related to.
- **Created by, Created on, Modified by, Modified on**

The following options are available for *only some* statuses:

- **Records in this status are considered active:** Only available for custom status workflows when the option "This status workflow controls the active property" is enabled. Records that gain a status that has this option enabled are changed to active automatically. They are likewise changed to inactive when they gain a status that has this option disabled. If a record has more than one status fields that are governed by status workflows that impact its active/inactive property, the record becomes / stays inactive when any of its statuses indicate that it is inactive.
- **Can select instrument with this status in an experiment run:** Instrument Statuses
- **Requires Location:** Content Statuses
- **Content is Removed:** Content Statuses
- **Only selected roles can generate reports for records with this status:** Order Statuses
- **Execute macro Only for Registered, Scheduled, Canceled Statuses:** Order Statuses, able to select a macro that will be executed when this status is reached.
- **Configure status rules:** see Section 10.16.3
- **Execute value expression:** this option states whether the Value expression (for tests having one) should be executed if the result of the tests *called within* the Value expression are in this status.

EXAMPLE: There are 2 tests, ASR and ORG, and ORG is a read only test with a Value expression calculated from ASR

100 - ASR

If "Execute value expression" is true for 'Verified' and false for 'Pending' and 'Available' result, then the ORG result value will only be calculated if the ASR status is 'Verified.'

So, if ASR = 75 and the ASR status is 'Available' (or 'Pending'), then the value of ORG is not evaluated. When the status of ASR becomes 'Verified,' then the value of ORG will be calculated: ORG = 25.

## 10.20.4. Status transitions

This tab configures the messages that show in the history when an event is created and defines whether the transition between various statuses are possible or impossible. For example, this tab would allow a user to designate that an Order in the "Pending" status can transition to the "Cancelled" status, but an Order in the "Registered" status cannot transition to "Cancelled."

The status transitions tab has a grid on the left that shows all transitions that exist for the selected status workflow. The transitions in the grid can be multi-selected. Selecting a transition shows the details in the form on the right.

Initial status transitions are created automatically when a new status is added to a status workflow, but they can be inactivated or activated. When a status is first added to a status workflow it gets a transition:

- From "Creation" to the new status. This transition can be disallowed to prevent creation of a record with this status. Only statuses with active initial transitions will show up in the record's status dropdown fields when the user creates a record.
- All possible status transitions are created when a new status is added to a workflow. Transitions are created from the new status to all other existing statuses in the status workflow, and from the existing ones to the new one. By default, the automatically created status transitions are active for statuses that are active, and are inactive for statuses that are inactive.
- It is not possible to remove status transitions, though they can be inactivated/reactivated.
- When a status is deactivated, the status transitions that go "To" the inactive status are also deactivated. The transitions that move "From" the inactive status are left active because records could still be present in that status.

These options are available for each status transition:

- **Status Transition:** Read-only, shows the status transition "From" and "To" status.
- **Label**
- **Description**
- **Active:** Activate or deactivate the status transition. This is the first level of control over whether users can perform this transition. By inactivating some status transitions, adminis-

trators can require records to follow a certain status update path through the status workflow. For example: If Creation > Pending is active, Pending > Available is active, Available > Labeled is active, but Creation > Available and Creation > Labeled are inactivate, it would prevent a record from being approved or labeled as soon as it was created.

- **Only selected users can perform this status transition:** Available on all status transitions of a status workflow. When enabled, a multi-select dropdown is available to choose the users, roles, and groups that can perform the status transition on records. Users that are not defined in the dropdown cannot perform the transition.
- **Status transition is restricted to certain role:** The administrator can specify a combination of users, groups, and/or roles that can execute this status transition.
- **An electronic signature is required to execute this status transition:** When enabled, the electronic signature window appears and requires the user to re-authenticate when he/she saves a record performing that status transition. Its behavior depends on the Lab Settings that are enabled. When enabled:
  - The password is always required.
  - The user can only sign as the currently logged in user.
  - If the lab setting "Electronic signature requires a username to be provided" is enabled, then a username is required.
  - If the lab setting "Electronic signature requires a meaning to be provided" is enabled, the meaning must be entered.
  - If the "Electronic signature requires comments to be filled" lab setting is enabled, comments must be entered when signing electronically for all status transitions in SLIMS. If the lab setting is disabled and an electronic signature is required for the status transition, the comment field will be offered but can be required or not required on a case-by-case basis.
  - When using an external authentication provider, the electronic signature pop-up can ask for the username, meaning and/or comments when configured with the above options and the "Sign with the authentication provider" and "Cancel" buttons are provided. The password is not asked for until the user clicks the button to sign with the provider. This opens another tab to ask for the username and password, depending on the provider. After signing, the tab can be closed and the user can return to SLIMS where the signature is added.
  - Plugins created with Vaadin show an electronic signature when needed. Its functionality is the same to the user as SLIMS but its appearance is styled by Vaadin.
  - SLIMS GATE Flows do not have a user interface for signature boxes so one will not be shown. In this case, SLIMS Engineers can configure the flow with temporary permissions to perform its task.
  - When multiple content is selected (multi-select checkboxes in a grid) and the electronic signature is required, one signature box will appear for both of the selected content.
- **Offer the user to enter a comment / reason for change when executing this status transition:** Disabled by default. When enabled, a comment field is provided for the user to enter comments that are saved in history while executing the status transition, but the transi-

tion can be executed even if the comment is left empty. "Comment / reason for change is required" is provided when this parameter is enabled. When disabled, the comment field is not provided to the user.

If the status transition requires an electronic signature, this parameter is enabled and read only so that a comment field is always offered to the user when they electronically sign to execute a status transition.

If the "Electronic signature requires comments to be filled" lab setting is enabled, this parameter and "Comment / reason for change is required" are enabled and read only.

① The "Creation" to "Published" status transition on every default publication status workflow allows comments but does not require them.

- **Comment / reason for change is required:** Disabled by default. Available only when comments / reason for change are being offered. When enabled, requires the user to enter a comment / reason for change in the supplied comment field. The status transition is prevented if no comment is entered.

When the status transition requires an electronic signature, this parameter can still be enabled or disabled unless the lab setting "Comment is required for electronic signature" is enabled. In that case, this parameter is also enabled and read only for all status transitions.

① *A note about migrating from older versions:* If "Require comment on signature" lab setting was enabled in the source SLIMS version, then both comment parameters in the upgraded SLIMS will be enabled on transitions that require signatures. Both comment parameters will be disabled if the lab setting was disabled.

- **Add event:** This option is only available for backward transitions, and when enabled will generate a pop-up text box for the user to provide a justification that will be saved with the created event.

- **Unique Identifier**

① The Revert action is available in the GUI and allows *backward* transitions only if allowed in this module.

For example, if all transitions are checked, an Order in the "Validated" status can be reverted to the "Content received" or "Completed" status. Unchecking the transition to "Completed" will make it unavailable to the Revert action.

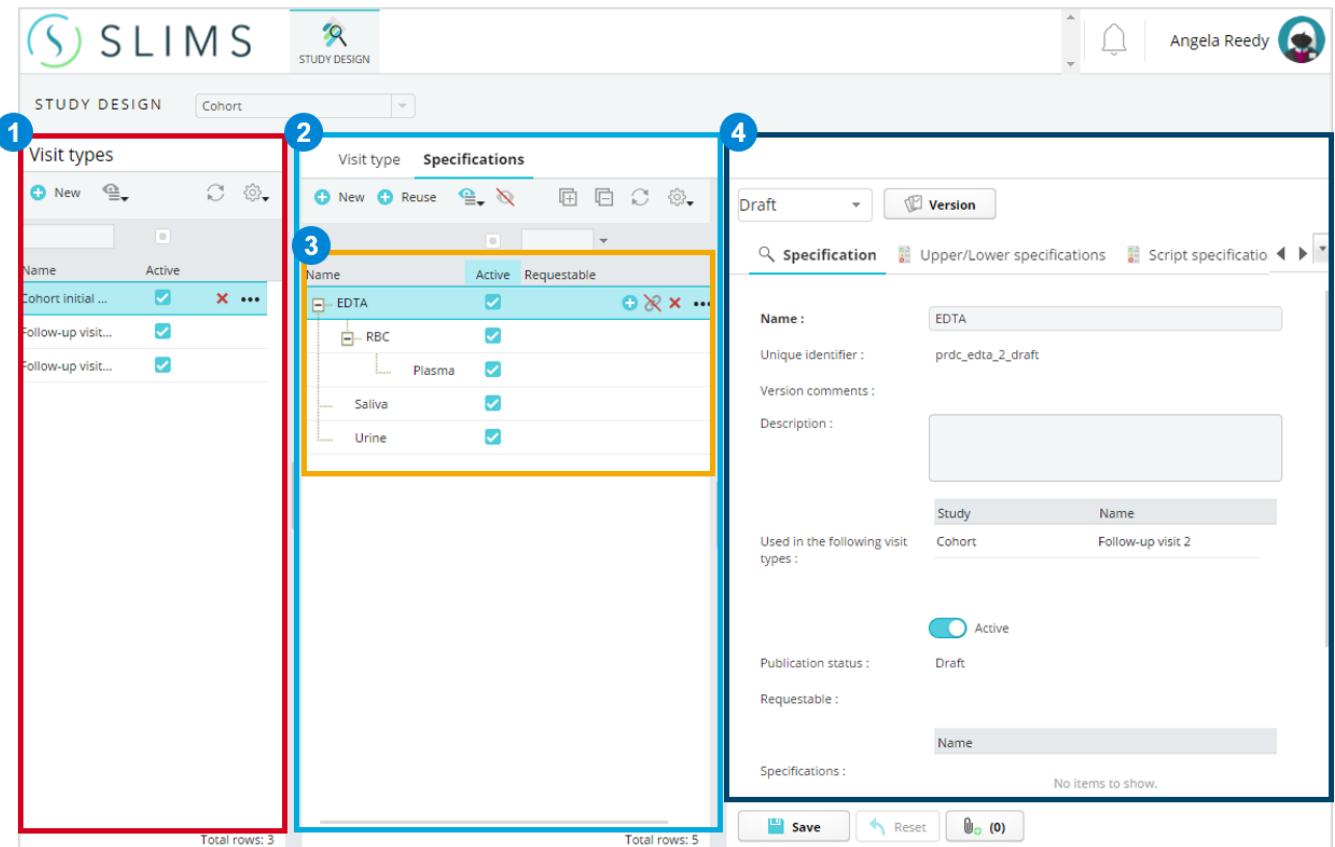
## 10.21. Study Design Module

Study Design is a configuration-only module that was designed to make the data structures for biobanks into core concepts in SLIMS. Configuration in this module allows mapping between Study, Workflow, Requestable, and Products and allows the products for designed studies to be separated from the products used in QC Orders. It has no influence on actual behavior otherwise, but will offer services to make customization easier. The module consists of Visit Types and Specifications as part of a study's design. Module access is controlled with the role functionality "Study Design Management."

The following Role Access rights is required to access this module:

**Table 10.21. Permissions for Study Design Module**

Access rights Category	Access rights Name	Purpose
Setup: Studies	Study Design	When enabled, the role user can access the Study Design module to create a data structure for their study(s) in SLIMS.

**Figure 10.19. Study Design Module**

The study selection menu provides a list of studies that were created in the Studies module. The menu shows a table in the dropdown list that includes any standard study fields and custom fields that were added to the "study" table that have the "Show in dropdown lists" option enabled in their field configuration. The dropdown can be sorted by clicking on the column headers. The user can also type text into the box and the study name and any of the added study fields or custom fields added to the display will be filtered based on what is typed.

## 10.21.1. Visit Types

Visit Types are technically a kind of Content Event Type, but they are entirely governed by setup in the Study Design module. They cannot be seen or interacted with in the Content Event Types module or the Study Module. This means that they do not show up in the list of Content Event Type restrictions for studies and cannot be added or deleted from that list. A Visit Type cannot

be changed into another kind of Content Event Type. Visit Types are special also in that they are always related to exactly one Study and cannot exist without that study.

After the name of a Study is selected in the title bar of the module, the Visit Types associated with the study can be viewed, updated, created, or removed from the left panel (1) or the Visit Type tab in the middle panel (2).

## 10.21.2. Specifications

A Specification is a product associated with one or more visit types within the same study or from multiple studies. Specifications in this context consist of a Root Specification which is associated with at least one visit type (ProductContentEventType table), and if desired, one or more Child Specifications which are descendants of a root (child, child of a child, etc.). Children are not individually associated to visit types and thus are only associated back through the associations of its root specification. (No entries for children appear in ProductContentEventType table.)

Specifications are versioned, which works the same way as in the Product and Specifications module, meaning that making a new version of a child product uses the specification of the last version of the parent product. Making a new version of a product will create new versions of its child products by copying their last version and including the new specifications of the parent product.

To add or edit specifications, first select a Study and a Visit Type. Then in the middle panel (3), Specifications can be added or edited on the Specifications tab.

Actions available in the header:

- *New*: Creates a new root specification.

For root specifications, the form is similar to the Products and Specifications form, but it provides the "Used in the following Visit Types" table to associate Visit Types and the studies that the Visit Types exist in.

- *Reuse*: Allows a root specification that is already associated with any number of Visit Types to be associated to the selected Visit Type also.

Actions available in context:

- *New*: Using "New" when a root specification is selected creates a new child specification related to the root.

For child specifications, the form is the same as the Products and Specifications form.

- *Unlink*: Allows dissociation of a root specification from the currently selected Visit Type as long as the root specification will still be associated with at least one visit type (as long as it will not be orphaned). Child specifications cannot be unlinked.
- *Remove Permanently*: Deletes and removes the root specification's association from all Visit Types and should be used with care. Remove cannot be used if the root or child specification has descendants because it would orphan them.

The actions available in the third panel of Specifications (4):

- *Specification*: The details of the selected specification.
- *Workflow*: Allows a Requestable to be linked or unlinked from a Specification. Allows the details of the Requestable to be viewed for any versions, but updated only for draft specifications.

Each version of a Specification can be linked at most to one Workflow Requestable (across versions of a Specification, the linked requestable may differ).

A requestable cannot be linked to multiple specifications but is linked to subsequent versions of the same specification. Requestables already in the Workflow module cannot be used, so linking a requestable here will create a new one. Vice versa, requestables created in Specifications will not be available in the Requestable tab of Workflow Management.

Upon creating a requestable, the user must supply the workflow, start queue, and end queue. After the creation of the requestable, the workflow, start queue, and end queue cannot be edited, but the workflow graph becomes editable for draft specifications.

A requestable linked to a draft specification may be unlinked. Unlinking is not possible for non-draft versions and has the following effects:

- If there are no requests for the requestable and no non-draft versions of the specification referring to it, the requestable will be removed.
  - If there are requests for the requestable or a non-draft version of the specification refers to it, the requestable will be inactivated.
- △ Requestables are not versioned. If the draft version of the specification is linked to the same requestable as previous versions of the specification, updating that requestable will affect all versions.

## 10.22. Unique Identifiers

Records in SLIMS have a unique identifier field with the primary purpose of providing a consistent, readable, fixed naming convention so that groovy, SLIMS GATE Flows, plugins, and other programmatic customizations can be built on as reliable a foundation as possible. This unique identifier is shown in the event history in the Changes column to provide further traceability. Auditors also benefit from reliable unique identifiers because it is easier to associate changes in data to the configuration records. The methodology of creating unique identifiers makes it easy for administrators too: the identifiers do not have to be typed by administrators during configuration, but are created automatically to provide ease of use and consistency.

It can be useful to query tables and records based on unique identifiers. To do so, check out the groovy section *findByUniqueId*.

When an administrator creates a new configuration record, such as creating a new attachment type, the unique identifier is automatically filled to match the label of the attachment type, but with a consistent suffix. For example, if the user types "images" as the label, SLIMS prefills the unique identifier field with "attp\_images" as he/she types. The administrator can still update the identifier to suit the lab's needs, but it must still be unique within the table (ex: the name "images" must still be unique within the attachment types table). Administrators can also instruct the naming conventions they want by specifying a value expression for the unique identifier field (in the example, for attachment types) so the expression generates the identifier instead of the SLIMS default method of generation.

## 10.22.1. Explanation of SLIMS Default Unique Identifiers Generation

This section is slightly technical but is intended to explain how the unique identifiers for various types of records in SLIMS are generated using the default generation method. This list is not exhaustive and is provided to show a few examples in SLIMS:

- **Status Workflows:** Default status workflows suggest the prefix "stwf" appended with "\_label" (for example: stwf\_cntn for the Default Content Status Workflow). Custom status workflows suggest the prefix "stwf" appended with "\_status workflow name" (for example: stwf\_review for the custom status workflow "Review.")
- **Built-in Statuses:** Built-in statuses are those that come with the default Status Workflows of SLIMS. They suggest the prefix "stts\_" appended with the status workflow identifier, and then the status label. Underscores separate the sections. (For example: stts\_cntn\_pending, stts\_nstr\_available, stts\_ordr\_in\_progress.)
- **Custom Statuses:** Custom statuses are those that come from custom status workflows. Custom status identifiers check for uniqueness to ensure they do not conflict with built-in status identifiers and append with a numeral if they are not unique. Built-in statuses do not get appended with a number so the non-unique custom statuses get appended. If there are more than one non-unique custom status identifiers, they are appended in order of sequence number. (For example: content table "Identified" status with sequence number 10 has the suggested ID "stts\_cntn\_identified1" and identified status with sequence number 20 has "stts\_cntn\_identified2" suggested. As long as it is the next sequentially, "Identifier" status with sequence 100 would have the ID "stts\_cntn\_identified3" suggested.)
- **Status Transitions:** The unique identifier cannot be created in the UI, so an identifier is generated that is the unique identifier of the prefix "sttr\_"\_related status workflow identifier"\_"From status label"\_to\_"To status label" (Example: sttr\_cntn\_pending\_to\_available). Initial status transition identifiers use the text "initial" instead of the "From" status label.

Status transitions of the result table used for QC additionally get appended with "\_qc."

- **Workflows:** Workflows have an identifier suggested based on the name of the workflow. (Example: A workflow named "Sample Collection" has this ID suggested: wrfl\_sample\_collection.)
  - **Queues:** The workflow name is included in the suggested queue identifier. (Example: The queues "Samples" and "Extracted" in the Sample Collection workflow have these suggested IDs respectively: queu\_sample\_collection\_samples, queu\_sample\_collection\_extracted.)
  - **Protocols:** Workflow protocols and ELN protocols have an indication of the draft or version number in their suggested identifiers. (Examples: The workflow protocol "Extraction" in the Sample Collection workflow has the suggested ID xptm\_sample\_collection\_extraction\_draft for the draft and xptm\_sample\_collection\_extraction\_v1 for the first version. A protocol of the same name but for ELN has the suggested ID xptm\_extraction\_draft for the draft and xptm\_extraction\_v1 for the first version.)

- **Protocol Steps:** The identifier for protocol steps includes the unique identifier of the protocol, version information, and name of the step. (Example: The workflow protocol step "Derive Sample" for the Extraction protocol has the suggested ID xpst\_extraction\_draft\_derive\_sample, and after the protocol is versioned, the ID for the step version is xpst\_extraction\_v1\_derive\_sample.)
- **Requestables:** There are different requestable types, each with their own prefix:
  - Analytical Methods are prefixed by rqlb\_anal\_meth\_"workflow unique identifier"\_"requestable name"
  - Test requestables are prefixed by rqlb\_test\_"requestable name"
  - Content requestables are prefixed by rqlb\_cntn\_"requestable name"
  - Standard workflow requestables are prefixed by rqlb\_wf\_"workflow unique identifier"\_"requestable name" (Example: The "Preparation" requestable has the suggested ID rqlb\_wf\_sample\_collection\_preparation).
- **Content Events:** The barcode is unique, so SLIMS uses the barcode as the unique identifier.
  - *Flags:* The barcode mask FLG##### is used as the unique identifier.
- **Units:** Their abbreviations are used as the unique identifier.
- **"Type" configuration entities:** "Type" entities used to configure the fields that are collected for records have a unique identifier barcode field. Administrators can update the barcode template that records generated from the "type" entity will have, and the records will have an incremented count using the barcode for their unique identifier. Some type entities are: Test, Case Report Form Template, Report Template, Rule, Worklist Template, QC Template, etc.
- **Imported Entities:** Entities or records that are imported from the SLIMS Store or Package Browser module have a special prefix added to the beginning of their unique identifiers to ensure they can be imported by preserving uniqueness in the target SLIMS instance. The "store\_" prefix is added for imported entities and persists to their children. For example, if an "Extraction" protocol is added to the "Sequencing" workflow imported from SLIMS Store, the workflow has unique identifier "store\_wrfl\_sequencing" and the protocol created by the user will have the identifier "xptm\_store\_wrfl\_sequencing\_extraction\_draft" suggested.

Unique identifiers do not possess some of the configuration options that other fields have, because they do not make sense for this type of field. Unique identifiers are immutable, read-only, and required to ensure they are protected. Therefore, these options are not available for unique identifiers in the Fields module:

- Execution Value: Hidden because unique identifiers cannot be changed.
- Read-only: Hidden because they are already read-only by nature.
- Required: Hidden because they are required by nature.
- Only for selected [type]: Hidden because they cannot be restricted.
- Only selected roles can access: Hidden because access is based on context.

- Only selected roles can update: Hidden because they cannot be updated.

If the default method of unique identifier generation does not yield results that suit the customer's needs, or there is no value for the field the unique identifier generation is based on, a value expression can be written for the unique identifier in the Fields module for that record type. The expression needs to use execution mode "Value expression is evaluated before creation." If the administrator also supplies the "Expression depends on" as well, the unique identifier that is generated will actually update live as the user types in the value of the other field (usually label). For further reference see the *Custom Fields* section.

This is an example of a value expression that could be used on the study unique identifier field that would generate an ID that has: prefix 'stud\_-' with the Lead Scientist field output with the study name separated by underscores.

- Create custom field `stud_cf_leadScientist` on the Study table and make it required.
- Refresh SLIMS.
- Search for the `stud_uniqueIdentifier` field. Enter the groovy below in the Value Expression box.

```
return "stud_" + study.get("stud_cf_leadScientist") + "_" + study.get("stud_name");
```

- Select `stud_cf_leadScientist` and `stud_name` in the Expressions Depend On dropdown.
- Refresh SLIMS and try it out.

**Figure 10.20. Unique Identifier Example**

Name :	Diabetes
Lead Scientist :	Leslie, Paul
Unique identifier :	stud_Leslie, Paul_Diabetes

## 10.23. Units

Units are defined for fields of type Quantity. For each content type, the most appropriate unit within the same dimension can be selected. A dimension is a set of units that represent the same measurement. For example: cubic meters and liters both belong to the volume dimension.

A Unit can be defined in many ways.

- Units can be defined from scratch. From the table "Unit," add a new unit and select *Standard*. In the new window, three fields need to be filled: the Abbreviation (what will appear next to the values in the fields), the Name (the full name - optional) and the Dimension it refers to.

*Example: abbreviation = l, name = liter, dimension = Volume*

Note: It is better not to choose this option if a unit already exists for the same dimension, but instead to choose *Value Expression*.

- A unit can be defined that refers to an existing one (like celsius degree can refer to kelvin degree). Add a new unit and select *Value Expression*. In the new window, five fields need to be set: Abbreviation, Name, Original Unit (which unit it is referring to), Value Expression to convert to original unit (Groovy script that converts the new unit to the original unit), Value Expression to convert from original unit (Groovy script that converts the original unit to the new unit).

*Example:*

- *abbreviation = °C,*  
*name = celsius degree,*  
*original unit = (K, kelvin),*  
*value expression to convert to original unit =*  
*return input.add(new BigDecimal("273"));*  
*value expression to convert from original unit =*  
*return input.subtract(new BigDecimal("273"));*
- *abbreviation = °F,*  
*name = fahrenheit degree,*  
*original unit = (°C, degree celsius),*  
*value expression to convert to original unit =*  
*value1 = input.subtract(new BigDecimal("32"));*  
*value2 = value1.multiply(new BigDecimal("5"));*  
*value3 = value2.divide(new*  
*BigDecimal("9"),20, BigDecimal.ROUND\_HALF\_UP);*  
*return value3;*  
*value expression to convert from original unit =*  
*value1 = input.multiply(new BigDecimal("9"));*  
*value2 = value1.divide(new BigDecimal("5"));*  
*value3 = value2.add(new BigDecimal("32"));*  
*return value3;*

Groovy scripts need an input (input value) and then all four basic operations can be performed (.add, .subtract, .multiply, .divide). The class *BigDecimal* is used to get precise conversion. More details about this class can be found at <http://docs.oracle.com/javase/1.5.0/docs/api/java/math/BigDecimal.html> [<http://docs.oracle.com/javase/1.5.0/docs/api/java/math/BigDecimal.html>]. Special attention has to be paid to the divide function. In the example above, the solution is round to 20 decimals up.

- You can define an alias for an existing unit which does not impact the value, just the display. For example, if it is preferred to have *liter* instead of *l* shown after the value, a new alias can be defined. Start by creating a new unit, and choosing *Alias*. Three fields need to be defined in the next window: Abbreviation, Name, and Original Unit.

*Example: abbreviation = liter, name = "", original unit = (l, liter).*

- A new prefix can be defined for an existing unit, like milliliter for liter. This can be done by adding a new unit and selecting *Prefix*. Only two fields from the dropdown lists need to be

defined: Original Unit and Prefix. For example, original unit = (l, liter), prefix = m. The powers associated to a prefix are implemented in SLIMS, so for example, SLIMS knows that a milliliter is a thousandth of a liter since it has been tied to the original unit.

- A unit that is a composite of two or more units can be defined. Add a new unit and choose *Compound*. In the left part of the new window, select one unit at a time and click on the green right arrow to add it to the compound unit. Once all units have been selected, define the power of all the chosen units. Then give the composite unit a name, the abbreviation will automatically be filled out, and dimension will be filled out as well if this dimension already exists; otherwise, a new dimension can be entered.

*Example:*

- *abbreviation = m, power = 2 => square meters*
- *abbreviation = m, power = 1; abbreviation = s, power = -1 => meter per second, speed*
- *abbreviation = g, power = 1; abbreviation = l, power = -1 => gram per liter, concentration*
- *abbreviation = g, power = 1; abbreviation = cm, power = -3 => gram per cubic centimeter, concentration*

Note that SLIMS recognizes the dimension of a compound unit and in some known cases suggests a name. For example, the dimension of the square of a *length* will be called *surface*

### 10.23.1. Importance of Unit Definition

It is important to define the units with care because a powerful mechanism is set up in SLIMS that converts units within the same dimension and returns the value corresponding to the unit for a particular field.

As an example, the concentration of a content given its mass and volume need to be calculated. To accomplish this, three custom fields need to be defined, all of data type quantity and referring to the content table:

- `cntn_cf_mass` with unit gram [g]
- `cntn_cf_volume` with unit liter [l]
- `cntnt_cf_concentration` with unit gram per milliliter [g/ml] and expression =  
`return content.cntn_cf_mass / content.cntn_cf_volume ;`

There is no need to define which conversion to do in the expression field, because the concentration unit is already set up as a compound unit using a mass component divided by a volume component. The example above does not use exactly the same units, but one could instruct the expression to simply multiply by one thousand. But in reality, the units could be expected to change for all content types, so it is challenging to define in advance which conversion will be needed in the Groovy script. To prepare for that circumstance, the expression is pared down to instruct that the mass should be divided by the volume.

Another example concerns the different metric systems. Imagine a customer measuring the temperature of its rats in fahrenheit, but the lab measures in celcius. Then only two custom fields would need to be defined: customer's temperature (`cntn_cf_customerTemperature`) and

temperature ( `cntn_cf_temperature` ). That way, the temperature can be simply defined with an expression that return the customer's temperature. `return cntn_cf_customerTemperature;` and the conversion will be done automatically.

## 10.24. Variants

The Variants module supports recording Variants data and assign them to contents as Variant observation by using the ELN and workflows.

This module requires the following Role Access rights to work:

**Table 10.22. Variants Permissions**

Access rights Category	Access rights Name	Purpose
Routine: Content	Create Variants	Allows the user to create Variants if the module has been turned on and set up.
Setup: Electronic Lab Notes	Variants	Allows access to the Variants module if the Lab Setting is also enabled.

### 10.24.1. Enable Variants

In order to use Variants, the Lab Setting "Variants enabled" has to be activated. Otherwise, everything about Variants will be hidden. This includes the functionality or the availability of the special tabs in Content management, Order management, ELN, and workflows. The Variants module can be accessed under the Setup menu once the "Variants" functionality access right is enabled.

### 10.24.2. Add Variants

The Variants module allows users to define special tests with the data type "Variant observation." Such a variant test has a limited amount of possible fields: a name, label, description, barcode unique identifier, and the ability to associate it to requestables. Each variant test has a number of variants associated with it. Variant is a new table that by default only has a name. Extra custom fields can be defined on the variant table.

### 10.24.3. Add Variant observation

Before being able to add Variant observation, a protocol with a step of type "Variant observation" has to be created. A "Variant observation" step can only be associated to Variant tests as tests. When running a protocol with such Variant observation tests in ELN or in workflows, no results are automatically created for Variant tests.

Results (Variant observations) can be created in the protocol run in ELN or the workflow protocol step by using the Variants tab. When clicking the "New" button, users select a content linked to the run step, one of the variant tests that were linked to the protocol step, and a Variant observation that was defined in the selected test.

An additional "Variants" tab is available in the Contents and Orders modules to show the Variant observation of the selected content records. The tab in the Contents module has an option to "Also show Variants of derivations." If this is disabled, only the Variants for the selected content will be shown in the tab. Enabling the option allows any Variants for content that are derived from the selected sample to also be shown. The identifier of the content is displayed to differentiate between the Variants belonging to the parent and derived content.

## 10.25. Versioning

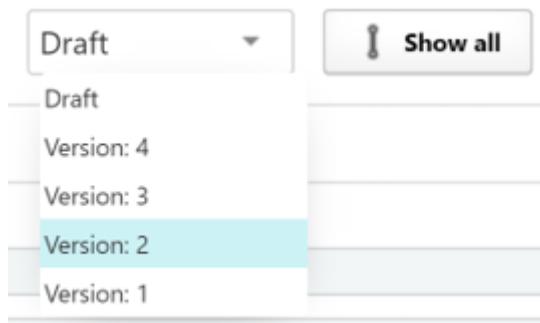
SLIMS supports version control for revisions done in SLIMS regarding Protocols, SOPs (Standard Operating Procedures), Report Templates, Stability Study Design, and Products and specifications. Versioning is managed the same way in most of the modules where it is used, and how versioning behaves in SLIMS is explained in detail in this chapter.

The Version button is disabled or not available when the records are in a status that does not allow versioning, or no changes have been made since the last version. The version button changes color to yellow to indicate that changes are present that have not yet been saved to a version, and then when it is clicked to create a new version, returns to its prior appearance.

### 10.25.1. Create/Edit Versions

Modules with a version control have a dropdown menu available that displays the versions that have been created. The dropdown contains the draft and all versions that have been saved of the record in question. When a version is selected, the information displayed on the module screen changes and shows the configuration that was saved to that revision. The 'Show All' / 'Hide All' button is next to the dropdown to open a window that shows the list of all created versions and the draft entry (explained below).

**Figure 10.21. Version Options**



The Draft is used to add and revise changes. By default, the Draft is considered to have the publication status "Draft." Once the revision is finished, a user can create a new Version by using the 'Version' button available either in the top-right or down near the save button in the Draft. When a record is versioned, it is considered to have the publication status "Published." Depending on the Lab Setting in SLIMS, if the status of the publication status workflow requires a signature, creating a new Version may require the user to enter their username, a meaning for the signature, and a comment in addition to their password to add their Electronic Signature to the history. When using an external authentication provider, the password is not asked for in the pop-up in SLIMS, but the username, signature meaning, and comment may be

asked. Instead, a button is provided to "Sign in with the authentication provider" which opens a pop-up or another tab to perform reauthentication. A new tab appears to sign with the password and username, depending on the authentication provider. A version saved in SLIMS is a read only copy of the current state of the draft. It can be viewed and used, but editing is no longer possible. Additionally, all previously saved versions (all versions besides the last one) have publication status "Obsolete" so they cannot be edited or used.

① Creating new versions and running old versions of a protocol are functionalities that can be made available for specific roles only (See *Roles module*). These functionalities' names in SLIMS are: 'Can create versions' and 'Can run old versions of protocols' respectively.

Records in SLIMS may have custom publication status workflows created, meaning that records may have more than one status that impacts the ability to version. If there are more than one statuses for a record, it must be in the statuses that allow versioning before the Version button shows in yellow and can be clicked. See more about Status Workflows in this section: Section 10.20.1.

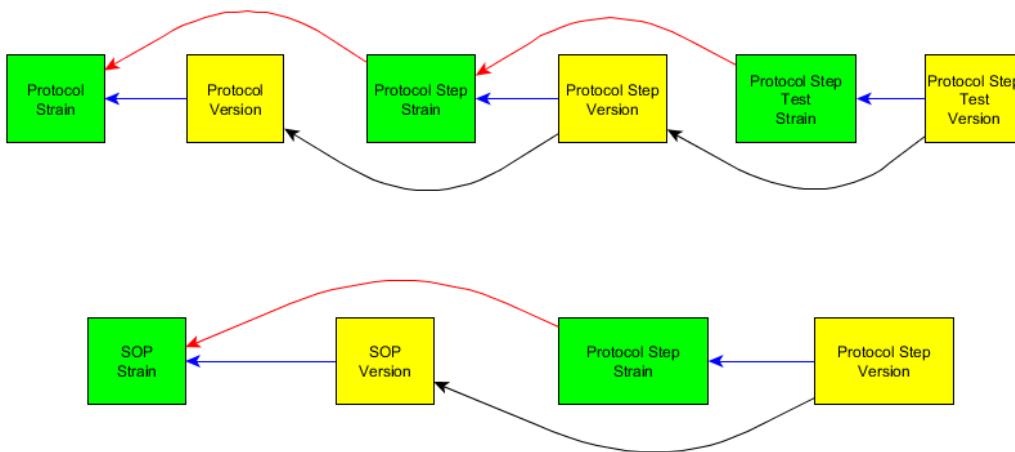
## 10.25.2. Versions Structure in SLIMS

This section describes how versions are structured and stored technically in SLIMS.

SLIMS uses strains to relate all versions of a record. Strains help manage the different versions of a record on the development side. For example: using strains we can find all protocol runs used with a particular protocol regardless of the protocol version.

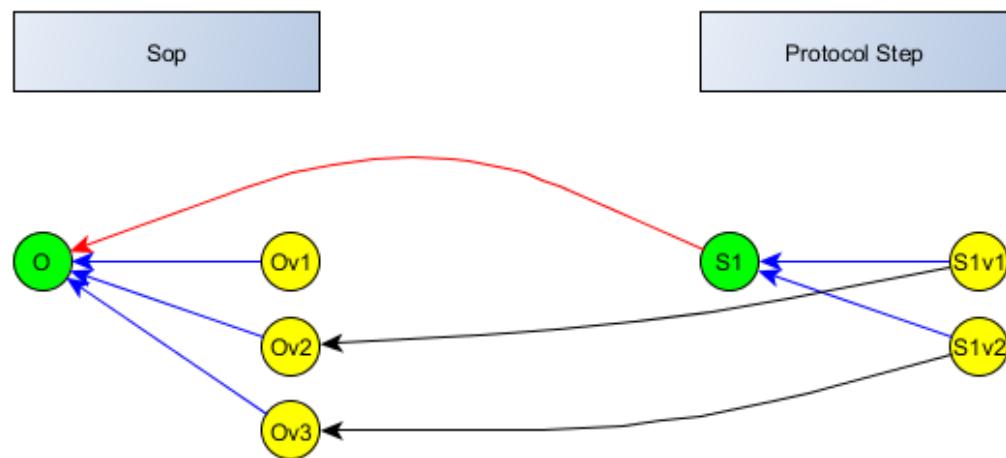
Figure 10.22 Shows the structure used in versions with Protocols and SOPs. The main principle is that Versions refer to Versions and Strains refer to Strains.

**Figure 10.22. Structure of Versions in SLIMS**



The same principle is applied to all records using versioning: Versions refer to Versions and Strains refer to Strains.

① Note that Tests also have versions and strains since their usage with Protocols is version specific.

**Figure 10.23. Example on how an SOP is stored in SLIMS:**

---

# 11. Groovy Scripting

This section contains a short introduction to the Groovy language, descriptions of methods and variables that are available to help access SLIMS data, and detailed examples of common usages.

## 11.1. Introduction to Groovy

Groovy is a scripting language based on Java. Most valid Java code is a valid Groovy code. Additionally, Groovy code can be made more compact and readable than Java.

The main things to know about Groovy that are relevant in our context are:

- **Variables:** Defining the type of the variable is not required and Groovy will assign the type automatically depending on the value. For example:

```
number = 5 // similar to: int number = 5;  
text = "text" // similar to: String text = "text";
```

- **Logical statements and looping:** Groovy supports the use of logical statement as 'if,' 'else,' and looping: 'for.' Their syntax can be used in the same way as when writing Java code. Please refer to the official Groovy documentation for *Control Structures* [[http://groovy-lang.org/semantics.html#\\_control\\_structures](http://groovy-lang.org/semantics.html#_control_structures)] for more details.
- **Methods:** Methods are defined by simply using the expression 'def.' For example, a method that prints in the log "Hello *user\_name*" (*user\_name* is the input parameter) is defined as following:

```
def printHello(userName){  
    log.info("Hello " + userName)  
}
```

⚠ The message is printed in the SLIMS log file (slims.log) available on the server and it is not shown to the user in the GUI. More details on the 'log' variable will be explained in *the next section*.

A message using

```
log.info("My message")
```

can also be printed in the console directly in SLIMS. The console can be opened by clicking the console icon:  available on the top right of the groovy script box.

Groovy scripting in SLIMS also offers some predefined methods which are available from the 'Insert' menu on the top left of the groovy scripting box. These methods will be described in section *Build-in SLIMS methods and variables*.

For more details, please refer to the *Groovy official documentation* [<http://groovy-lang.org/documentation.html>].

## 11.2. Groovy Editor

### 11.2.1. Fields Where Groovy is Available

The following Role Access rights can be configured for Groovy:

**Table 11.1. Permissions for Value Expressions**

Access rights Category	Access rights Name	Purpose
Setup: Miscellaneous	Can Edit Value Expressions	When enabled, the role user is able to add and update Groovy expressions in value expression fields. When disabled, the role user is able to see expressions written by users with this permission, but is not able to add or update Groovy expressions in value expression fields.

Fields that require a value expression in the Groovy language are editable in an advanced source code editor. Those fields are:

- Value expression (tbfl\_groovy) in Fields module.
- Dynamic filter expression (tbfl\_pickListGroovy) in Fields module.
- Script (lbtm\_script) in Label templates module.
- Value expression (lbtl\_script) in Label template elements module.
- Conditional value expression (rule\_condition) in Rules module.
- Value expression (rule\_expression) in Rules module.
- Value expression (test\_groovy) in Tests module.
- Value expression to convert from original unit (unit\_fromOriginalUnitGroovy) in Units module.
- Value expression to convert to original unit (unit\_toOriginalUnitGroovy) in Units module.

### 11.2.2. Editor

The header of the editor contains the following options:

- *Insert*: Opens a menu with all available/supported variables and functions.
- *Value expression console*: Opens a non-modal pop-up window that displays live Groovy logging and problems for the executions of the current value expression.

- **Maximize:** Maximizes the editor in order to provide a better overview of your value expression.

The editor furthermore supports:

- Syntax highlighting for the Groovy language.
- Search and Replace (Ctrl/cmd + F).
- Auto-completion (Ctrl + Space) of the following type of words:
  - *SLIMS:* cfr. variables and functions that can be added with the above 'Insert' menu.
  - *local:* already typed words in the current editor.
  - *keyword:* basic Groovy language keywords.

## 11.3. Build-in SLIMS Methods and Variables

SLIMS provides a certain environment with variables and methods (SLIMS API) to be used with Groovy. First, this section will show a table presenting the API provided by SLIMS and where it is available. Secondly, this section will give a detailed description with short examples on the provided API.

**Table 11.2. Build-in SLIMS Variables**

Variables	Available In
<code>authenticatedUser</code>	Field expression, Status rules, Conditional value expression rules,
<code>FetchRequest</code>	Field expression, Status rules, Conditional value expression rules, Result rules and Compare result rules.
<code>SlimsRestrictions</code>	Field expression, Status rules, Conditional value expression rules, Result rules and Compare result rules.
<code>empty</code>	Field expression
<code>daoHelper</code>	Field expression, Status rules, Conditional value expression rules, Result rules and Compare result rules.
<code>error</code>	Status rules, Conditional value expression rules, Result rules and Compare result rules
<code>log</code>	Always (Whenever Groovy can be used in SLIMS)
<code>NA</code>	Field expression
<code>ok</code>	Result rules and Compare result rules
<code>oldValue</code>	Field expression
<code>sequences</code>	Field expression
<code>simultaneousChanges</code>	Field expression
<code>tableName</code>	Field expression, Status rules and Conditional value expression rules
<code>warn</code>	Status rules, Conditional value expression rules, Result rules, Compare result rules and QC rules

Variables	Available In
<code>notify</code>	<i>Status rules and Conditional value expression rules</i>
<code>email</code>	<i>Status rules and Conditional value expression rules</i>
<code>holidayHelper</code>	<i>Field expression</i>

① Tests only have the `'log'` as integrated API. However, tests result values can be fetched in the value expression by using the test name followed by `'.rslt_value'` like the following: `testName.rslt_value`. The returned test result value will be the one obtained in the same experiment run step. If none is found then it will take the most recent one.

### 11.3.1. oldValue

A variable containing the value before evaluation of the field on which the field expression is calculated.

### 11.3.2. empty

A variable used to return an empty value in the evaluated field.

△ In fields with datatype Quantity an empty map should be returned instead: "return []"

△ Returning a null value in the field expressions generates an error and thus the field expression will not be evaluated. Therefore, "return empty" always has to be used instead of "return null."

### 11.3.3. NA

NA means that the value of the field is not available.

To set a field to NA

```
return NA
```

should be used. Note that NA is coerced to false therefore this 'if' will never be executed:

```
if (NA) { println("will never happen") }
```

To check if a field value is NA, the value of the `name_of_the_field.appended with _isNa` can be checked. For example:

```
if (content.cntn_cf_field_isNa) {
    println("cntn_cf_field is NA")
}
```

will print if the value of the field is NA (and do nothing otherwise).

The expression

```
if (content.cntn_cf_field == NA)
```

can also be used *if it refers to the record the value expression is evaluated on*. Note that the expression will not work, for example, on a record fetched by a dao method (so not the same record on which the value is evaluated).

## 11.3.4. log

This variable is used to write messages in the SLIMS log file that is available on the SLIMS server. Please contact your system manager to get access to the SLIMS log file.

### 11.3.4.1. Methods Summary

Modifier and Type	Method and Description
void	error(String message) Log a message with error log level. The default log level of SLIMS is 'info' and therefore logs with this level will be written in the SLIMS log file.
void	warn(String message) Log a message with warn log level. The default log level of SLIMS is 'info' and therefore logs with this level will be written in the SLIMS log file.
void	info(String message) Log a message with info log level.
void	debug(String message) Log a message with debug log level. The default log level of SLIMS is 'info' and therefore this message will not be written in the SLIMS log file unless the log level of SLIMS is modified to debug.

## 11.3.4.2. log.error

```
void log.error(String message)
```

Log a message with error log level.

**Parameters:**

message - log this message.

## 11.3.4.3. log.warn

```
void log.warn(String message)
```

Log a message with warn log level.

**Parameters:**

message - log this message.

## 11.3.4.4. log.info

```
void log.info(String message)
```

Log a message with info log level.

**Parameters:**

`message` - log this message.

### 11.3.4.5. `log.debug`

```
void log.debug(String message)
```

Log a message with debug log level.

**Parameters:**

`message` - log this message.

## 11.3.5. `tableName`

Map that contains all the values of the record being evaluated and it is named after the table that the Groovy script is written for. This variable is a Map of `<String, Object>` key-value pairs where the `String` is the field name and the `Object` is the field value. Depending on the datatype of the field, the `Object` could be a `String`, `Integer`, `Quantity`, `Date`...

### 11.3.5.1. Methods Summary

Modifier and Type	Method and Description
Object	<code>get(String key)</code> Returns the value to which the specified key is mapped, or <code>null</code> if this map contains no mapping for the key.

### 11.3.5.2. `tableName.get`

```
Object tableName.get(String key)
```

Returns the value to which the specified key is mapped, or `null` if this map contains no mapping for the key.

ⓘ The call to this method can be done using the expression `map.key` instead of `map.get("key")`. For instance, 'content.cntn\_barCode' can be used instead of 'content.get("cntn\_barCode").'

The list of field names used in the Groovy script is accessible through the Fields module in SLIMS.

**Parameters:**

`key` - the key whose associated value is to be returned

**Returns:**

the value to which the specified key is mapped, or `null` if this map contains no mapping for the key

**Examples:**

`content.cntn_id` to obtain the content id when executed on records in the *Content* table. The variable '`content`' will refer to the content for which the Groovy script is evaluated.

`result.rslt_value` to obtain the result value when executed on records in the *Result* table. The variable '`result`' will refer to the result for which the Groovy script is evaluated.

## 11.3.6. authenticatedUser

This variable holds a User object containing details about the authenticated user that triggered the evaluation of the field expression.

### 11.3.6.1. Methods Summary

Modifier and Type	Method and Description
String[]	<code>groupNames</code> Returns the groups' names that the user is assigned to.
String	<code>role_name</code> Returns the name of the role assigned to this user.
String	<code>role_updateLevel</code> Returns the update level of the authenticated user.
String	<code>user_email</code> Returns the email address of the authenticated user.
Long	<code>user_pk</code> Returns the primary key of the authenticated user.
String	<code>user_userName</code> Returns the user name of the authenticated user.

### 11.3.6.2. authenticatedUser.groupNames

`String[] authenticatedUser.groupNames`

Returns the groups' names that the user is assigned to.

**Returns:**

Names of the groups that the user is assigned to.

### 11.3.6.3. authenticatedUser.role\_name

`String authenticatedUser.role_name`

Returns the name of the role assigned to this user.

**Returns:**

The name of the role assigned to this user.

#### **11.3.6.4. authenticatedUser.role\_updateLevel**

```
String authenticatedUser.role_updateLevel
```

Returns the update level of the authenticated user. Possible values are: "All," "Group," "Personal," or "Nothing".

**Returns:**

The update level of the authenticated user.

#### **11.3.6.5. authenticatedUser.user\_email**

```
String authenticatedUser.user_email
```

Returns the email address of the authenticated user.

**Returns:**

The email address of the authenticated user.

#### **11.3.6.6. authenticatedUser.user\_pk**

```
Long authenticatedUser.user_pk
```

Returns the primary key of the authenticated user. It is the unique identifier of a user.

**Returns:**

The primary key of the authenticated user.

#### **11.3.6.7. authenticatedUser.user\_userName**

```
String authenticatedUser.user_userName
```

Returns the user name of the authenticated user.

**Returns:**

The user name of the authenticated user.

#### **11.3.7. simultaneousChanges**

A variable of type `Map` that allows updating other fields of the updated record. The update is done by adding new fields' values to the `simultaneousChanges` map.

#### **11.3.8. Quantity and Unit**

##### **11.3.8.1. quantity(amount, unit)**

```
quantity(amount, unit)
```

Identifies the quantity, quantity field, and unit so it can be returned, used in a calculation, or created.

Groovy can be written for fields that allow choice of unit. Read about the conversions in [Section 10.5.5.15 Datatype Quantity](#).

**Parameters:**

amount - The amount value to create/return.

unit - The unit/unit abbreviation to use for the quantity value.

**Returns:**

The quantity amount and unit are returned and can be used in a calculation.

**Example:**

A value expression on a field "Volume" (datatype Quantity, Dimension Volume) that returns "5 ml".

```
quantity("5", unit("ml"))
```

### 11.3.8.2. unit(String abbreviation)

```
unit(String abbreviation)
```

Used in other calculations, this simply identifies the unit abbreviation.

**Parameters:**

String abbreviation - The unit abbreviation written in parentheses, as ("x").

**Returns:**

Returns the unit abbreviation in the final calculation.

**Examples:**

A value expression on a field "Volume" (datatype Quantity, Dimension Volume) that returns complex quantity with conversion "0.005 l \* 5.5 ml" aka "27.5 ml".

```
return 2 * quantity("0.005", "l") * quantity("5.5", unit("ml"))
```

A value expression on a field "Volume" (datatype Quantity, Dimension Volume) that returns calculated quantity with conversion written a different way.

```
return quantity("5", "l").convertTo(unit("ml"))
```

### 11.3.9. Comparator Quantity and Position

#### 11.3.9.1. comparatorQuantity(position, amount, unit)

```
comparatorQuantity(position, amount, unit)
```

Returns a value for the comparator quantity field with a position, amount, and unit.

**Parameters:**

position - The position (<, >, =) value to create/return. Can only be a ComparatorQuantityPosition.

amount - The amount value to create/return. Can be a string, number, or BigDecimal.

unit - The unit abbreviation to use for the quantity value.

**Returns:**

A quantity comparator with the specified quantity position, amount, and unit.

**Example:**

A value expression on a field "Mass" (datatype Comparator Quantity, Dimension Mass) that returns "< 1" in the field.

```
return comparatorQuantity(ComparatorQuantityPosition.LESS, "1", "mg")
```

### 11.3.9.2. ComparatorQuantityPosition

ComparatorQuantityPosition.POSITION

This is a helper group with the options EQUAL, LESS, or GREATER to be used in the comparatorQuantity expression.

**Parameters:**

EQUAL - One of three choices of position, adds equals sign into field: =.

LESS - One of three choices of position, adds less than sign into field: <.

GREATER - One of three choices of position, adds greater than sign into field: >.

**Returns:**

Returns the position in the final calculation.

**Examples:**

See the Section 11.3.9.1 example which includes ComparatorQuantityPosition.

### 11.3.10. triggerRecalculation

```
triggerRecalculation(entityName: String, recordPk: Long)
```

The "Reevaluate value expression after save to the database (not recommended)" option should be enabled to use the triggerRecalculation API.

**Parameters:**

entityName - Name of the entity type that should be recalculated.

recordPk - The primary key of the record to find.

**Returns:**

This is a way to identify a different record than the one being added/updated and trigger the Groovy for that record. triggerRecalculation can be used in-line with other groovy to compare a record's value so the identified record's Groovy will be re-triggered or skipped depending on the criteria. It only triggers an entity once if it has already been triggered to prevent cycles in calculation, and a maximum of 5000 entities can be triggered.

**Example:**

Trigger the Lot value to re-evaluate if updated content value is not the same as the old value. If it is the same, keep the old value.

```
if (!oldValue.equals(newValues.cntn_cf_fk_lot)) {  
    if (oldValue) {  
        triggerRecalculation("Content", oldValue);  
    }  
    triggerRecalculation("Content", content.cntn_cf_fk_lot);  
}
```

### 11.3.11. findByUniquedIdentifier

All Table Queries and Table RecordQueries have a method findByUniquedIdentifier. DaoHelper has a findByUniquedIdentifier method.

**Parameters:**

entityName - Name of the entity type.

uniqueIdentifier - uniqueIdentifier to query for.

**Returns:**

Returns the record associated with the queried unique identifier. In case the table does not have a unique identifier column, a TableDoesNotHaveUniquedIdentifiersException is thrown.

**Example:** You can use this expression to fetch contentTypePKs. (This example uses a category field called cntn\_fluid\_category.)

```
fluidCategoryPk =  
    daoHelper.findByUniqueIdentifier("ContentType", "cntn_fluid_category").cntp_pk;
```

### 11.3.12. macroExecutionDetails

```
macroExecutionDetails.GROUPCHOICE()
```

This is a helper group with context information on the macro step this groovy is executed in. Null if executed outside macro context.

**Parameters:**

`getOriginalRecord()` Only for derive and copy macro steps. Get the record being derived or copied.

`getBatchSize()` The total amount of input records when executing a macro in batch mode.

`getBatchSeqNo()` The sequence number of the execution when executing a macro in batch mode.

`getMultiplicitySize()` The expected total amount of output records for the current input record in macro steps with output records (e.g. Derive steps).

`getMultiplicitySeqNo()` The sequence number of this output record for the current input record in macro steps with output records (e.g. Derive steps).

`getMacroPk()` Primary key of the macro.

`getMacroStepPk()` Primary key of the macro step.

`getExperimentRunStepPk()` Primary key of the experiment run step (protocol run step).

**Returns:**

Returns the scripted information from the context of the macro step the groovy was executed on.

**Example:** You can use this expression to fetch the protocol run step pk from a protocol derivation step and output the pk value into a custom field on the content that was derived by the macro. (The example does not show this setup: A custom field is created first on the Content table with a String type datatype, and "Value expression is executed before creation" is enabled on the field. Then a macro with a Derive step is created and set up to be used in a Derivation protocol step.)

```
return macroExecutionDetails?.getExperimentRunStepPk();
```

### 11.3.13. daoHelper

This variable holds a daoHelper object. This object has methods to query records using a SLIMS criterion or the primary key of a record.

The table name is needed in the query methods to indicate in which table the query should be done. Table names are found in the Fields module in SLIMS. Additionally, there are options for the Execution Mode that choose when the expression is evaluated.

- Value expression is evaluated before creation (default value)
- Value expression is evaluated on every save

The "Reevaluate value expression after save to the database (not recommended)" is offered primarily to use the triggerRecalculation API later in this section.

ⓘ When using the **table name** in the script, spaces have to be removed and only capital letters will be kept as word separators. For instance the table 'Content type' will be named 'Content-Type' in the script.

### 11.3.13.1. Methods Summary

Modifier and Type	Method and Description
List<Map<String, Object>>	fetch(String tableName, SlimsCriterion slimsCriterion) Fetches all records from a <code>tableName</code> that satisfy a <code>slimsCriterion</code> .
List<Map<String, Object>>	fetch(String tableName, FetchRequest fetchRequest) Fetches all records from a <code>tableName</code> that satisfy a <code>fetchRequest</code> .
Map<String, Object>	find(String tableName, Long pk) Queries for a record in a table <code>tableName</code> with a primary key <code>pk</code> .
FetchRequest	getFetchRequest() Returns the fetch request of the <code>daoHelper</code> object.
Map<String, Object>	findByUniqueId() Searches the record in the <code>entityName</code> table by its <code>uniqueIdentifier</code> .

① Quantity values are returned differently between `fetch` and `find` methods. For `find` methods, a map (`String, Object`) with amount and unit entries is returned, whereas for `fetch` methods, the `Quantity` object is returned.

### 11.3.13.2. `daoHelper.fetch`

```
List<Map<String, Object>> daoHelper.fetch(String tableName, SlimsCriterion slimsCriterion)
```

Fetches all records from a `tableName` that satisfy a `slimsCriterion`. `slimsCriterion` are created by using `SLimsRestrictions`

#### Parameters:

`tableName` - name of the table to query on. For instance: `Content`, `Order`...

`slimsCriterion`.

#### Returns:

List of the fetched records or `null` if nothing is found.

#### Example:

Fetch all contents of type DNA:

```
contentType = "DNA";
contents = daoHelper.fetch( "Content", SlimsRestrictions.equals("cntp_name",
contentType));
```

### 11.3.13.3. `daoHelper.fetch`

```
List<Map<String, Object>> daoHelper.fetch(String tableName, fetchRequest FetchRequest)
```

Fetches all records from a `tableName` that satisfy a `fetchRequest`. Please refer to the `FetchRequest` section for more details on how to use `FetchRequest`.

**Parameters:**

`tableName` - name of the table to query on. For instance: `Content`, `Order`...

`fetchRequest` - contains criterion for the fetch.

**Returns:**

List of the fetched records or `null` if nothing is found.

**Example:**

Fetch all contents of type DNA:

```
contentType = "DNA";
FetchRequest request =
daoHelper.getFetchRequest().setCriterion(SlimsRestrictions.equals("cntp_name",
contentType));
contents = daoHelper.fetch("Content", request);
```

#### 11.3.13.4. `daoHelper.find`

```
Map<String, Object> daoHelper.find(String tableName, Long pk)
```

Queries for a record in a table `tableName` with a primary key `pk`.

**Parameters:**

`tableName` - name of the table to query on. For instance: `Content`, `Order`...

`pk` - The primary key of the record to find.

**Returns:**

Map of the found record or `null` if nothing is found.

**Example:**

Find the content type record of a content:

```
//content.cntn_fk_contentType returns the primary key of the content type.
contentType = daoHelper.find("ContentType", content.cntn_fk_contentType)
```

#### 11.3.13.5. `daoHelper.getFetchRequest`

```
FetchRequest daoHelper.getFetchRequest()
```

**Returns:**

`FetchRequest` The fetch request of the `daoHelper` object.

**Example:**

Create a fetch request with the criterion `cntp = "DNA"`:

```
contentType = "DNA";
FetchRequest request = daoHelper.getFetchRequest();
request.setCriterion(SlimsRestrictions.equals("cntp_name", contentType));
```

## 11.3.14. FetchRequest

FetchRequest is an object containing defined fetch criterion in order to fetch SLIMS records.

### 11.3.14.1. Methods Summary

Modifier and Type	Method and Description
FetchRequest	<pre>setCriterion(SlimsCriterion criterion)</pre> <p>Set a SLIMS criterion to the fetch request.</p>
FetchRequest	<pre>setAddSecurityCriterion(Boolean addSecurityCriterion)</pre> <p>Defines if the security criterion concerning access rights on the authenticatedUser will be applied during the fetch.</p>

### 11.3.14.2. FetchRequest.setCriterion

```
FetchRequest FetchRequest.setCriterion(SlimsCriterion criterion)
```

Set a SLIMS criterion to the fetch request.

**Parameters:**

criterion - SLIMS criterion that can be added using SLIMS restrictions

**Returns:**

FetchRequest containing the set criterion.

**Example:**

Create a fetch request with the criterion cntp = "DNA":

```
contentType = "DNA";
FetchRequest request = daoHelper.getFetchRequest();
request.setCriterion(SlimsRestrictions.equals("cntp_name", contentType));
```

### 11.3.14.3. FetchRequest.setAddSecurityCriterion

```
FetchRequest FetchRequest.setAddSecurityCriterion(Boolean addSecurityCriterion)
```

Defines if the security criterion concerning access rights on the authenticatedUser will be applied during the fetch. The default setting is `True`. It can be set to `false` if the groovy script needs to access data of all SLIMS independently of the access rights of the authenticated user that triggers the groovy script.

**Parameters:**

addSecurityCriterion - Boolean parameter that defines whether to add access right security to the fetch or not.

**Returns:**

FetchRequest

**Example:**

Fetch all contents of type "DNA" without applying access right security checks on the authenticated user triggering the groovy:

```
contentType = "DNA";
FetchRequest request = daoHelper.getFetchRequest();
request.setAddSecurityCriterion(false);
request.setCriterion(SlimsRestrictions.equals("cntp_name", contentType));
contents = daoHelper.fetch("Content", request);
```

## 11.3.15. SlimsRestrictions

SlimsRestrictions is used to create SlimsCriterion in order to fetch for records.

### 11.3.15.1. Methods Summary

Modifier and Type	Method and Description
	equals(String fieldName, Object value)
SlimsCriterion	Creates a criterion where the fieldName value is equal to the specified value.
	gt/lt/ge/le(String fieldName, Object value)
SlimsCriterion	Creates a criterion where the fieldName value is greater than/lower than/greater or equal/lower or equal to the specified value.
	in(String fieldName, Object[] values)
SlimsCriterion	Creates a criterion where the fieldName value is one of the specified values.
	like(String fieldName, String pattern)
SlimsCriterion	Creates a criterion where the fieldName value matches the pattern.
SlimsCriterion	not(SlimsCriterion criterion)
	Creates a negation to the input criterion.
SlimsCriterion	isNaSlimsRestrictions.isNa(fieldName)
	Checks if a field is N/A.
	and()
SlimsJunction	Creates a SlimsJunction with conjunctions.
SlimsJunction	or()

Modifier and Type	Method and Description
	Creates a SlimsJunction with disjunctions.

### 11.3.15.2. SlimsRestrictions.equals

```
SlimsCriterion SlimsRestrictions.equals(String fieldName, Object value)
```

Creates a criterion where the fieldName value is equal to the specified value.

**Parameters:**

fieldName - name of field in SLIMS.

value - value that should be matched in the field with the name fieldName.

**Returns:**

SlimsCriterion.

**Example:** Creates a criterion on the field "cntn\_id" to be equal to "12345678":

```
SlimsRestrictions.equals("cntn_id", "12345678")
// Corresponding sql criterion: WHERE cntn_id = '12345678';
```

### 11.3.15.3. SlimsRestrictions gt/lt/ge/le

```
SlimsCriterion SlimsRestrictions.gt(String fieldName, Object value)
```

```
SlimsCriterion SlimsRestrictions.lt(String fieldName, Object value)
```

```
SlimsCriterion SlimsRestrictions.ge(String fieldName, Object value)
```

```
SlimsCriterion SlimsRestrictions.le(String fieldName, Object value)
```

Creates a criterion where the fieldName value is greater than/lower than/greater or equal to/lower or equal to the specified value.

**Note:** Unit is not taken into account when comparing Quantity objects.

**Parameters:**

fieldName - name of field in SLIMS.

value - value that should be matched in the field with the name fieldName.

**Returns:**

SlimsCriterion.

**Example:** Creates a criterion on the field "cntn\_amount" to be greater than "42":

```
SlimsRestrictions.gt("cntn_amount", "42")
```

```
// Corresponding sql criterion:      WHERE cntn_amount > '42';
```

#### 11.3.15.4. SlimsRestrictions.in

```
SlimsCriterion SlimsRestrictions.in(String fieldName, Object[] value)
```

Creates a criterion where the fieldName value is one of the specified values.

**Parameters:**

fieldName - name of field in SLIMS.

value - value that should be matched in the field with the name fieldName.

**Returns:**

SlimsCriterion.

**Example:**

```
cntn_cf_name Contains one of the values "ABC", "DEF", "GHI"
```

```
SlimsRestrictions.in("cntn_cf_name", ["ABC", "DEF", "GHI"])
//Corresponding sql criterion:      WHERE cntn_cf_name IN ('ABC', 'DEF', 'GHI');
```

#### 11.3.15.5. SlimsRestrictions.like

```
SlimsCriterion SlimsRestrictions.like(String fieldName, String pattern)
```

Creates a criterion where the fieldName value matches the pattern.

① The percentage (%) is a wild-card character used to indicate that the pattern could contain at this position: nothing, single or multiple characters.

**Parameters:**

fieldName - name of field in SLIMS.

value - value that should be matched in the field with the name fieldName.

**Returns:**

SlimsCriterion.

**Example:**

```
cntn_cf_note Contains the pattern "%C"
```

```
SlimsRestrictions.like("cntn_cf_note", "%C")
//Corresponding sql criterion:      WHERE cntn_cf_note LIKE '%C';
```

#### 11.3.15.6. SlimsRestrictions.not

```
SlimsCriterion SlimsRestrictions.not(SlimsCriterion slimsCriterion)
```

Creates a negation to the input criterion.

**Parameters:**

fieldName - name of field in SLIMS.

value - value that should be matched in the field with the name fieldName.

**Returns:**

SlimsCriterion.

**Example:**

```
not(SlimsRestrictions.equals("cntn_createdBy", "12345678"))
//Corresponding sql command:      WHERE NOT (cntn_createdBy = '12345678');
```

### 11.3.15.7. SlimsRestrictions.and

```
SlimsJunction SlimsRestrictions.and()
```

Creates a SlimsJunction with a conjunctions between the added criterion.

**Returns:**

SlimsJunction. This object is extended from SlimsCriterion.

**Example:**

Creates a SlimsJunction to add multiple SlimsRestrictions:

```
SlimsRestrictions.and().add(SlimsRestrictions.equals("cntn_createdBy", "12345678")
.add(SlimsRestrictions.like("cntn_id", "%C"))
//Corresponding sql criterion:
//WHERE cntn_createdBy = '12345678' AND cntn_id LIKE '%C';
```

### 11.3.15.8. SlimsRestrictions.or

```
SlimsJunction SlimsRestrictions.or()
```

Creates a SlimsJunction with disjunctions between the added criterion.

**Returns:**

SlimsJunction. This object is extended from SlimsCriterion.

**Example:**

Creates a disjunction between multiple SlimsRestrictions:

```
SlimsRestrictions.and().or(SlimsRestrictions.equals("cntn_createdBy", "12345678")
.add(SlimsRestrictions.like("cntn_id", "%C"))
//Corresponding sql criterion:
//WHERE cntn_createdBy = '12345678' OR cntn_id LIKE '%C';
```

## 11.3.16. Fetch Contents Affecting/Affected By a Content

This change relates to the Affecting/Affected By filter in the various Content module tabs. See the filter described in the ??? section.

### 11.3.16.1. Fetch Content(s) Affected By a Content

```
myAffectedContents = daoHelper.fetch("Content",
    SlimsRestrictions.equals("affectedBy", contentPk))
```

This groovy will return all contents affected by the content with primary key contentPk within myAffectedContents.

```
myAffectedContents = daoHelper.fetch("Content",
    SlimsRestrictions.in("affectedBy", contentPk))
```

This groovy will return all contents affected by the content with primary keys contentPk within myAffectedContents.

### 11.3.16.2. Fetch Content(s) Affecting a Content

```
myAffectingContents = daoHelper.fetch("Content",
    SlimsRestrictions.equals("affecting", contentPk))
```

This groovy will return all contents affecting the content with primary key contentPk within myAffectingContents.

```
myAffectingContents = daoHelper.fetch("Content",
    SlimsRestrictions.in("affecting", contentPk))
```

This groovy will return all contents affecting the contents with primary keys contentPk within myAffectingContents.

### 11.3.16.3. Fetch Both Content(s) Affecting and Affected By a Content

```
myContentFamily = daoHelper.fetch("Content",
    SlimsRestrictions.equals("affecting_or_affectedBy", contentPk))
```

This groovy will return all contents affecting and affected by the content with primary key contentPk within myContentFamily.

```
myContentsFamilies = daoHelper.fetch("Content",
    SlimsRestrictions.in("affecting_or_affectedBy", contentPk))
```

This groovy will return all contents affecting and affected by the contents with primary keys contentPk within myContentFamily.

## 11.3.17. sequences

Sequences are groovy constructs that keep track of the number of times they have been executed. Each sequence requires a unique id. Sequences can be used in any groovy field in SLIMS. An example of a sequence statement:

```
return sequences.next("LocationNumber");
```

which could be used as a default value in a custom LocationNumber field, for instance.

### 11.3.17.1. Methods Summary

Modifier and Type	Method and Description
Long	<p>next(String id)</p> <p>Checks to see if there is already a sequence with the specified id. If so, it will increment the number stored in that sequence and return it. If not, it will create a new (persistent) sequence, set it to 0, and return 0.</p>

### 11.3.17.2. sequences.next

```
Long sequences.next(String id)
```

The next-method requires a String id. Every time it is called, it checks to see if there is already a sequence with that id. If so, it will increment the number stored in that sequence and return it. If not, it will create a new (persistent) sequence, set it to 0, and return 0.

Sequences can go up to Java's `long.MAX_VALUE`, after which they throw an exception.

Depending on the relevant settings for the encapsulating groovy field, sequences will increase every time a new record with this sequence is made using the relevant groovy script, or every time one is updated. Sequences will not update when the script is called upon opening the "create new" window, but the value the sequence will have on the next update is inserted into the script, in order to preview the result.

Inserting sequences into label templates will result in the sequence incrementing every time the label is sent to the printer, which may be several times if the printer is not connected, not switched on, or otherwise not responding.

Sequences are updated directly from the groovy script, which means that, for instance, a sequence in an if-block that is not executed, will not be increased. Similarly, a sequence in a for-loop will be incremented as many times as the code loops.

### 11.3.18. ok

Sets the rule evaluation of the evaluated result to 'Ok.' This method is available for rules of type *Result rules* and *Compare result rules*.

⚠ This method is evaluated only if the result belongs to a protocol run linked to an Order.

```
void ok(String test, String message)
```

**Parameters:**

`test` - name of the test on which the ok rule will be shown.

`message` - text message.

### 11.3.19. warn

This method name refers to two different methods that are used in different rule types:

- *Status rules, Conditional value expression rules, Result rules, Compare result rules* and *QC rules*: Shows a Warning message to the User. Transaction is not blocked.

```
void warn(String message)
```

**Parameter:**

message - text message.

- *Result rules* and *Compare result rules*: Sets the rule evaluation of the evaluated result to 'Warning'.

⚠ This method is evaluated only if the result belongs to a protocol run linked to an Order.

```
void warn(String test, String message)
```

**Parameters:**

test - name of the test on which the ok rule will be shown.

message - text message.

## 11.3.20. error

This method name refers to two different methods that are used in different rule types:

- *Status rules* and *Conditional value expression rules*: Stops/fails the record update. If possible, the message will be shown in the form into the according field.

```
void error(String field, String message)
```

**Parameters:**

field - the field on which the error message will be shown.

message - text message.

- *Result rules* and *Compare result rules*: Sets the rule evaluation of the evaluated result to 'Error'.

⚠ This method is evaluated only if the result belongs to a protocol run linked to an Order.

```
void error(String test, String message)
```

**Parameters:**

test - name of the test on which the ok rule will be shown.

message - text message.

## 11.3.21. notify

Sends a notification. This method is available for rules of type *Status rules* and *Conditional value expression rules*.

The notification does not block the current operation.

```
void notify(String title, String text, Notification.NotificationLevel
level, Long[] userPk)
```

**Parameters:**

`title` - the title of the notification.

`text` - the text of the notification.

`level` (optional) - the level of the notification: INFO, SUCCESS, WARNING or ERROR. If level is not set, INFO will be used by default.

`userPk` (optional) - the list of users the notification should be sent to. If userPk is not set, the current user will be used by default.

Simple example:

```
notify(title: "Experiment signed", text: "Experiment has been signed off");
```

Advanced example:

```
notify(title: "Order validated", text: "Order with barcode $barcode has been validated",
level: 'SUCCESS', userPk: [adminUser.user_pk, technicianUser.user_pk]);
```

## 11.3.22. email

Sends an email. This method is available for rules of type *Status rules* and *Conditional value expression rules*

⚠ In order to deliver emails, it is necessary to set the *Email templates SMTP URI* in the Lab Settings and to define an email template. See *Email Templates* for details.

```
void email(String templateName)
```

**Parameter:**

`templateName` - name of the template that builds the email(s).

## 11.3.23. holidayHelper

This variable can be used to calculate new dates based on working days or to calculate the date difference in working days. To correctly skip holidays, the Holiday module in SLIMS needs to be kept up-to-date.

### Calculate a new date

For example, to calculate the date that is 6 working days from the current date, while excluding both weekend and holidays, the following value expression can be used:

```
excludeWeekends = true;
excludeHolidays = true;
dateCalculator = holidayHelper.createDateCalculator(excludeWeekends, excludeHolidays);
```

```
dateCalculator.setStartDate(org.joda.time.LocalDate.now());
dateCalculator.moveByBusinessDays(6);
newDateOnlyWorkingDays =
dateCalculator.getCurrentBusinessDate().toDateTimeAtStartOfDay().toDate();
```

To use a different date field as the start date of the calculation, replace `org.joda.time.LocalDate.now()` with `org.joda.time.LocalDate.fromDateFields(dateField)`.

#### Calculate working days date difference

To calculate the number of working days between two given dates, the *value expression* needs to return the following:

```
excludeWeekends = true;
excludeHolidays true;
dateDifferenceInWorkingDays = holidayHelper.createPeriodCalculator(excludeWeekends,
excludeHolidays).dayDiff(dateField1, dateField2)
```

## 11.4. Examples

In the following section, use case examples of Groovy scripts applied with Field expressions and Rules will be shown.

### 11.4.1. Fields, datatype Quantity, table Content: calculate mass from Volume and Concentration field values

**Context:** Fields, datatype Quantity, table Content, calculated at every save.

```
//Verify that volume and concentration values are not null otherwise an error will be thrown
if (content.cntn_cf_volume && content.cntn_cf_concentration) {
    //return the calculated the mass.
    return content.cntn_cf_volume * content.cntn_cf_concentration
}
//Return an empty value if volume or concentration values are null
return empty
```

① Unit conversions are managed automatically by SLIMS as long as the fields of data type 'Quantity' have been correctly configured.

△ The value expression is evaluated only when the record is modified.

### 11.4.2. Fields, datatype Decimal number, table Content: calculate the difference between two quantity fields and converts it to double

**Context:** Fields, datatype Decimal number, table Content, calculated at every save.

```
if (content.cntn_cf_mass.getAmount() != null && content.cntn_cf_massThre.getAmount() != null) {
```

```
        difference = content.cntn_cf_mass.getAmount() - content.cntn_cf_massThre.getAmount();
        return difference.toDouble();
    }
```

The conversion to double is useful when the difference value is used in highlight rules; for example, when records with mass below a certain threshold need to be highlighted.

### 11.4.3. Field `cntn_fk_group`, table Content: set default group if unique

**Context:** Field `cntn_fk_group`, table Content, calculated before creation.

```
if (authenticatedUser != null && authenticatedUser.getGroupPk() != null &&
authenticatedUser.getGroupPk().length == 1) {
    return authenticatedUser.getGroupPk()[0];
} else {
    return null;
}
```

If the authenticatedUser belongs to only one group, that group will be assigned by default to the created content, otherwise no group will be assigned by default.

### 11.4.4. Fields, datatype Date, table Content: calculate expiration date from production date and duration

**Context:** Fields, datatype Date, table Content, calculated before creation.

```
import groovy.time.TimeCategory
if (content.cntn_cf_duration) {
    duration = content.cntn_cf_duration;
}
if (content.cntn_cf_productionDate && duration) {
    Date production = content.cntn_cf_productionDate;
    use(TimeCategory) {
        int months = duration;
        return production + months.month;
    }
}
```

Note the use of `TimeCategory`. This package allows to add months to a date.

### 11.4.5. Rules, Order Status: perform checks on requestables when scheduling an order

**Context:** Rules, type Status, table Order, status Schedule.

```
orderPk = order.ordr_pk;
if (orderPk) {
    contentPk = daoHelper.fetch("OrderContent", SlimsRestrictions.equals("rdcn_fk_order",
orderPk).get(0).get("rdcn_fk_content"));
    if (contentPk) {
        content = daoHelper.find("Content", contentPk);
        if ("Blood".equals(content.cntp_name)) {
            requests = daoHelper.fetch("Request",
SlimsRestrictions.and()
.add(SlimsRestrictions.equals("rqst_fk_order", orderPk))
```

```
        .add(SlimsRestrictions.equals("rgst_fk_content", contentPk)));  
  
    requestables = requests.collect{it -> it.rqbl_name};  
    if (requestables.contains("hematocrit") && !requestables.contains("whiteCells"))  
{  
    error("Requestables", "When hematocrit test is requested on blood, whiteCells  
test should also be requested");  
}  
}  
}  
}  
}
```

If contents of type "Blood" are linked to an Order, it won't be possible to schedule it if "hematocrit" is requested and "whiteCells" is not requested. The following message will be shown: "When hematocrit test is requested on blood, whiteCells test should also be requested".

## 11.4.6. Rules, Conditional value expression: check that all results are verified when stopping a step

**Context:** Rules, type Conditional value expression, table Protocol run step

### Conditional value expression:

```
return "DONE".equals(experimentRunStep.get("xprs_status"));
```

## Value expression:

```

table = daoHelper.fetch("DbTable", SlimsRestrictions.equals("dbtb_name", "Result") );

status = daoHelper.fetch( "Status",
    SlimsRestrictions.and()
        .add( SlimsRestrictions.equals("stts_fk_table", table.dbtb_pk) )
        .add( SlimsRestrictions.equals("stts_name", "Verified") )
);

if ( daoHelper.count( "Result",
    SlimsRestrictions.equals("rslt_fk_experimentRunStep", experimentRunStep.xprs_pk)
) != daoHelper.count("Result",
    SlimsRestrictions.and()
        .add( SlimsRestrictions.equals("rslt_fk_experimentRunStep",
experimentRunStep.xprs_pk) )
        .add( SlimsRestrictions.equals("rslt_fk_status", status.stts_pk) )
)
{
    error("", "Not all the results are in status Verified. You cannot finish the step");
}

```

If the number of results linked to the present experiment run step is not equal to the number of *verified results* an error is thrown and the step can not be finished.

## 11.4.7. Rules, Conditional value expression: requires a value for a Protocol run step field when stopping the step

**Context:** Rules, type Conditional value expression, table Protocol run step

### Conditional value expression:

```
return "DONE".equals(experimentRunStep.get("xprs status"));
```

Value expression:

```
if ( ! experimentRunStep.get("xprs_cf_requiredField") ) {  
    error("xprs_cf_requiredField", "Please enter a value for required field");  
}
```

## 11.4.8. Rules, Conditional value expression: checks that all contents are removed when stopping a Protocol run step

**Context:** Rules, type Conditional value expression, table Protocol run step

Conditional value expression:

```
return "DONE".equals(experimentRunStep.get("xprs_status"));
```

Value expression:

```
exprunstepcontents = daoHelper.fetch("experimentRunStepContent",  
    SlimsRestrictions.equals("xrsc_fk_experimentRunStep", experimentRunStep.xprs_pk));  
contentMaps = exprunstepcontents.collect { daoHelper.find("Content", it.xrsc_fk_content) };  
// Different kind of checks on the content fields can be defined here  
notRemovedContents = contentMaps.findAll { it.cntn_status != 50 };  
  
if (notRemovedContents.size() > 0) {  
    error("Contents", "Please remove all contents before to continue");  
}
```

## 11.4.9. Rules, Result Status: requires a comment for some kind of results

**Context:** Rules, Status, table Result, status Verified

```
if(result.rslt_value == "Not OK" && result.rslt_comments == null) {  
    error("rslt_comments", "A comment is required when result is Not OK");  
}
```

## 11.4.10. Fields, datatype Dynamic choice: example of Dynamic filter expression

**Context:** Fields, table Protocol run, Dynamic choice with Dynamic filter expression

```
targetContentType = content.cntn_cf_fk_targetContentType;  
criteria = new HashMap();  
criteria.put("cntn_fk_contentType", targetContentType);  
return criteria;
```

The dynamic choice list for this field will be all contents of the content type defined in the field cntn\_cf\_fk\_targetContentType.

A dynamic filter expression can also be used to filter with multiple values:

```
criteria = [  
    operator: "or",
```

```

criteria:[
  [
    fieldName: "prvd_name",
    operator: "equals",
    value: "Genohm"
  ],
  [
    fieldName: "prvd_name",
    operator: "equals",
    value: "A Lab Inc"
  ]
]
]
return criteria;

```

### 11.4.11. Test: calculate from values of other tests

**Context:** Test

The Value expression needs to depend on the Tests from which the value will be calculated (Figure 11.1).

```
// The expression depends on tests PH_1 and PH_2
return return (PH_1 + PH_2) / 2;
```

**Figure 11.1. Test value expression dependence**

Label	Description
PH 1st value	PH_1
PH 2nd value	PH_2

### 11.4.12. Test result: calculate result for all results of quantity test

**Context:** Test result

This groovy is on a result for a Test called "Concentration," which has datatype Quantity and Dimension Concentration, will return 12.3 µg/ml in the rslt\_value field of all results to this test.

A more complex computation of the concentration to return (based on custom fields of the same result, on results to other tests on the same content or any other business logic) is possible using the usual Groovy tools (daoHelper, see earlier in this section).

```
// Example Groovy on Result for a Test of datatype Quantity
unit = daoHelper.fetch("Unit", SlimsRestrictions.equals("unit_abbreviation", "&mu;g/ml"))

Map<String, Object> concentrationQuantity = new hashMap<String, Object>();
concentrationQuantity.put("amount", 12.3)
concentrationQuantity.put("unit_pk", unit.get(0).unit_pk)

return concentrationQuantity;
```

## 11.5. Test Groovy Scripts

There is a groovy web console available to test groovy scripts: <http://groovyconsole.appspot.com/>

All that is required is to fill in the objects/bindings the way that SLIMS would normally fill them in order to test the scripts. This way of writing scripts will easily solve typos or other compilation problems.

### Example:

A custom field 'rslt\_cf\_expired' should be filled depending on some other result fields. The groovy script to use in the groovy web console to test the script is the following:

```
1
result = new HashMap();
result.put("rslt_workListDate", new Date(year: 2014, month: Calendar.APRIL, date: 1,
hours: 12, minutes: 12, seconds: 0));
result.put("rslt_cf_expiryDate", new Date(year: 2014, month: Calendar.APRIL, date: 1,
hours: 14, minutes: 12, seconds: 0));
5
if (result?.rslt_workListDate && result?.rslt_cf_expiryDate
&&
result.rslt_workListDate.clearTime().after(result.rslt_cf_expiryDate.clearTime())) {
    println(true);
} else {
10
    println(false);
}
```

The final groovy script for the field in SLIMS would be:

```
1
if (result?.rslt_workListDate && result?.rslt_cf_expiryDate
&&
result.rslt_workListDate.clearTime().after(result.rslt_cf_expiryDate.clearTime())) {
    return true;
5
} else {
    return false;
}
```

### Remarks:

- **result?.rslt\_workListDate** uses the Safe Navigation Operator and will return true if result is not null and if rslt\_workListDate is not null
- **clearTime()** is a method in Groovy that's available for the Date class that will clear the hours, minutes and seconds

# 12. SLIMS GATE Flows Monitoring

This chapter explains how to view SLIMS GATE flow statuses and usage in your SLIMS instance.

## 12.1. SLIMS GATE Flows Monitoring

The following Role Access rights can be configured for use in this module:

**Table 12.1. SLIMS GATE Permissions**

Access rights Category	Access rights Name	Purpose
Routine: SLIMS GATE	Able to View SLIMS GATE Runs from Other Users	When enabled, the role user can view SLIMS GATE runs that were started by themselves and other users in the Notifications > SLIMS GATE menu. This permission also adds a button in the SLIMS GATE menu to filter "Only my flows."
Routine: SLIMS GATE	Can Shut Down SLIMS GATE	When enabled, the role user has the "Shutdown SLIMS GATE" button available in the Notifications > SLIMS GATE menu.

Information about executed SLIMS GATE flows and their statuses are available in the Bell menu (bell icon) on the right side of the top bar.

The button 'Flows Status' contains two indications:

- Number of currently running process in SLIMS GATE.
- Notification button that turns:
  - **Grey:** No new notification and no running flow.
  - **Green:** One or multiple flows are currently running.
  - **Red:** One of the executed flows failed. Red color will turn to grey again after clicking on the Flows Status button.

Clicking on the Flows Status button shows all the flow records in a new grid. Each flow record has a:

- **Name:** The name of the flow that was executed.
- **Status:** Contains five possibilities:
  - *In progress:* The flow process is still running.

- *Done*: The flow finished successfully.
- *Failed*: An error occurred during the flow execution. The error log can be accessed by using the info button as explained below.
- *Interrupted*: The user can interrupt long running flows by right-clicking and using the 'Interrupt' button on a flow in status 'In progress.' After clicking on 'Interrupt,' a flow execution is stopped only when reaching specific checkpoints that are defined in the flow itself in order to make sure that the flow interruption goes smoothly.
- *Partially done*: The flow has started, and some of the steps have finished successfully, but the next step to be executed has not started yet.
- **Started/Finished date**: Date and time when the flow was started/finished.
- **Info button**: Clicking on this button will open a new window that shows all relative information to the flow:
  - *Parameters*: Input parameters used to start the flow.
  - *Log*: Shows the log messages generated by the flow execution. Log messages are dynamically updated when the flow is still running.

# 13. Specific Use Cases

This chapter shows some use cases with details on how SLIMS is configured to deal with specific lab requirements.

## 13.1. Pedigrees

SLIMS supports the structured tracking of relationships between family members to trace inheritable traits and conditions between related subjects. Each family member is represented by a content record. As much information as needed can be collected by using custom fields.

Pedigrees can be used to trace the inheritance of a specific trait, abnormality, or disease. SLIMS uses the Proband (the person under study) as the main record in a pedigree to define family relationships, and as many other family members as needed can be added as related content.

The sections below cover how to set up SLIMS with the required configuration to manage Pedigrees, how to use relationships with a pedigree, and finally, describes further optional configuration that your company's engineers can use to optimize Pedigrees in your SLIMS instance.

### 13.1.1. Required Setup

In order to start using relationships between family members, the following setup is required:

- › Go to the "SLIMS Store" module and import the "Pedigree" package. This will create the necessary fields, content types, and relations for a basic starting configuration.
- › Navigate to the "Labsettings" module and ensure that the "Pedigree enabled" option is checked.

### 13.1.2. Add/Set Relationship

All subjects in a relationship are derived from a Pedigree which needs to be added first. Then the Proband, or subject who is being studied, should be derived from the Pedigree and identified as the Proband with the field in the Derivation Form. Multiple other subjects can be added using the Add Parent and Add Child actions.

It is important to understand the "Parent" and "Child" tools which are essential for explaining all relationships in a Pedigree. All relationships are added using these two relations. A relationship can be added by using the "Add Parent" or "Add Child" actions which creates a new record with that relation to the Proband, or by setting this relation between two existing records using the actions "Set Parent" and "Set Child." This works with existing subjects that were derived from the Proband's pedigree. The actions "Add parent/child" and "Set parent/child" are available as well with Macros.

### 13.1.3. View Relationships

The subject's relationship to the Proband is shown in the column 'Relation to proband' in the Details Tab in the Content module. In addition, the columns "Father" and "Mother" show the parents' ID.

⚠ The column 'Relation to proband' is available only for contents of type "Subject," and derivations of the Subject (for example: Blood, DNA, urine) do not get the value propagated to them.

The following list shows relations that are recognized in SLIMS with their priority rules. Each subject will get the highest priority that it can be categorized as:

Relationship	Priority
Child	1
Son	20
Daughter	20
Father	20
Mother	20
Partner	17
Grandchild	2
Grandson	12
Granddaughter	12
Grandparent	2
Great-grandchild	2
Great-grandparent	0
Great-grandson	12
Great-granddaughter	12
Paternal Grandparent	7
Maternal Grandparent	7
Grandfather	12
Paternal Grandfather	17
Maternal Grandfather	17
Grandmother	12
Paternal Grandmother	17
Maternal Grandmother	17
Great-grandparent	0
Paternal Great-grandparent	5
Maternal Great-grandparent	5
Great-grandfather	10
Paternal Great-grandfather	15
Maternal Great-grandfather	15
Great-grandmother	10
Paternal Great-grandmother	15
Maternal Great-grandmother	15
Grandaunt	0
Paternal Grandaunt	5
Maternal Grandaunt	5

Relationship	Priority
Granduncle	0
Paternal Granduncle	5
Maternal Granduncle	5
Naunt	-2
Paternal Naunt	3
Maternal Naunt	3
Grandnaunt	-5
Paternal Grandnaunt	0
Maternal Grandnaunt	0
Weak Sibling	2
Paternal Weak Sibling	10
Maternal Weak Sibling	10
Sibling	16
Paternal Half-sibling	16
Maternal Half-sibling	16
Half-sibling	14
Weak Brother	3
Paternal Weak Brother	11
Maternal Weak Brother	11
Brother	17
Paternal Half-brother	17
Maternal Half-brother	17
Half-brother	15
Weak Sister	3
Paternal Weak Sister	11
Maternal Weak Sister	11
Sister	17
Paternal Half-sister	17
Maternal Half-sister	17
Half-sister	15
Uncle	8
Paternal Uncle	13
Maternal Uncle	13
Aunt	8
Paternal Aunt	13
Maternal Aunt	13
Nibling	-3
Cousin	-5

Relationship	Priority
Paternal Cousin	10
Maternal Cousin	10
Nephew	10
Niece	10

When a relation is unknown to SLIMS (it could be either too far in the pedigree tree or too complicated), the field "Relationship to Proband" is set to "Unknown" but the relation can still be found using the custom fields "Father" and "Mother."

⚠ Father and/or mother fields can only be calculated by SLIMS if the parent gender (cntn\_cf\_sex) has a value. Otherwise these fields are left empty even though the parents are known.

### 13.1.4. View Pedigrees

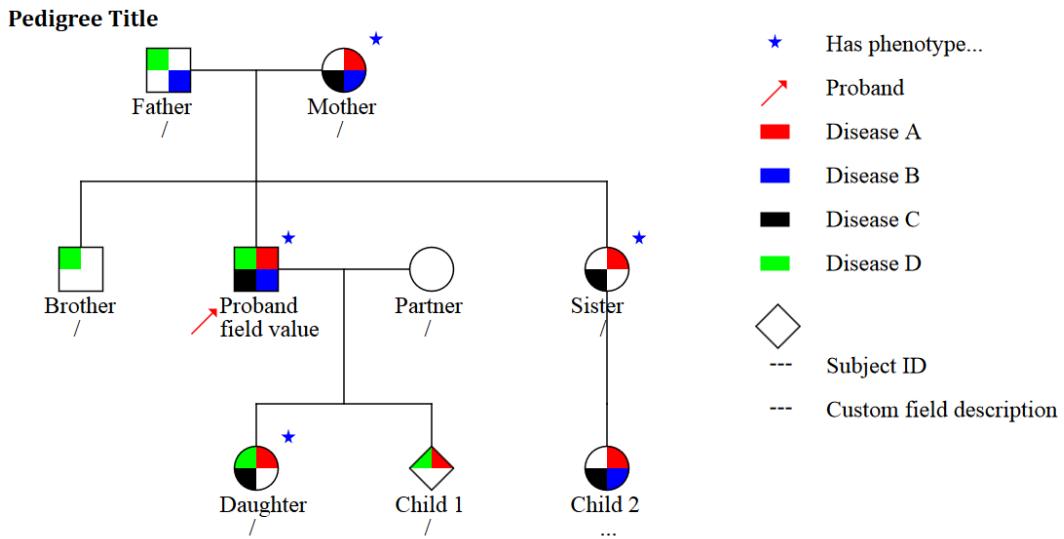
The "Pedigree" Tab is available to view the family pedigree when selecting either a Subject or a Pedigree content record. The pedigree drawing is shown in the browser as a PNG image. However, it can be downloaded in an editable vector format (SVG) using the link "**Download as SVG**" available at the bottom of the page.

⚠ The pedigree tab is available only for contents of type "Subject." Therefore, the pedigree cannot be visualized when a content type other than Subject is selected (for example: Blood, DNA, Mouse).

SLIMS uses symbols to represent males, females, mating (marriage), and offspring. A horizontal line between two subjects shows that they are parents and a vertical line describes the parent-child relationship. A square means male, a circle means female, and a diamond means a subject of unknown gender.

The drawing layout can be specified on the pedigree record for each family. Thus, each family can have a differently colored highlight, legend, and extra information to show in the pedigree drawing.

Custom fields and highlights can be added to the pedigree drawing but your SLIMS administrator has to submit a request to Genohm to add them.

**Figure 13.1. Example of a pedigree obtained with SLIMS:**

## 13.1.5. Custom Configurations

The below sections can be used by developers to customize SLIMS Pedigrees even further.

① SLIMS GATE is available with every SLIMS instance since version 6.1. To work appropriately with Pedigrees, ensure that `pedigree.graph.factory=defaultPedigreeGraphFactory` in the `slimsgate.properties` file.

The below parameters are created with the packages installed from the Store to create only relations between the members of the family.

- Content Types:
  - **Subject:** Contents have to be of type "Subject" to allow adding relationships between them.
  - **Pedigree:** All of the "Subjects" from the same family have to be derived from the same "Pedigree" to make it possible to add relationships between them.
- Custom Fields:
  - **cntn\_cf\_proband:** A switch (slider) custom field to select which "Subject" is the proband.
  - **cntn\_cf\_affected:** A switch (slider) custom field to select if a subject is affected.
  - **cntn\_cf\_sex:** Required fixed choice custom field with possible values "female," "male," and "unknown."
  - **cntn\_cf\_deceased:** A switch (slider) custom field to select if the subject is deceased.
- Content Relation Types:
  - **Parent:** A switch (slider) that sets the relation between parents and children.

- **Proband:** A switch (slider) that sets the relation between the proband and the other members of the family.

These default settings can be modified by editing the database table 'configuration.' The default Pedigree settings available in the column 'cnfg\_configuration' are as follows. Using a different content type than "Subject" or "Pedigree," or using different field names could be configured by Genohm if requested by a SLIMS administrator. This can be done by editing default configurations in the database described below:

```
<?xml version="1.0" encoding="UTF-8"?>
<java version="1.6.0_38" class="java.beans.XMLDecoder">
<object class="com.genohm.slims.common.xml.PedigreeConfiguration">
<void property="affectedField">
<string>cntn_cf_affected</string>
</void>
<void property="femaleValue">
<string>female</string>
</void>
<void property="maleValue">
<string>male</string>
</void>
<void property="parentRelation">
<string>Parent</string>
</void>
<void property="patientContentType">
<string>Subject</string>
</void>
<void property="pedigreeContentType">
<string>Pedigree</string>
</void>
<void property="probandField">
<string>cntn_cf_proband</string>
</void>
<void property="probandRelation">
<string>Proband</string>
</void>
<void property="sexField">
<string>cntn_cf_sex</string>
</void>
</object>
</java>
```

Additional configuration is done with the packages to set up quadrants which allow various characteristics to be followed through the generations in the family tree. (see example from Figure 13.1)

- Custom Fields:
  - **cntn\_cf\_"position"Quadrant:** Where "position" is upLeft, downLeft, upRight and downRight. These fields allow the selection of which of the subject's characteristics (default or custom field) the quadrant coloration will be based on. This field is a fixed choice datatype with multiple values possible (label of any field from the subject) and will only be for the "Pedigree" content type.
  - **cntn\_cf\_"position"Quadrant"color"if:** Where "position" is upLeft, downLeft, upRight, and downRight and where "color" can be any color predefined in the Java Color class. Multiple colors can be used for the same position but only one color will be shown on the Pedigree graph. These fields allow the condition that determines the field color to be selected. This field is a fixed choice which contains the value the previous parameter can take. For ex-

ample, (as configured in the package) if "Eyes color" is selected in the field "Up left quadrant," the field "Up left quadrant orange if" is a dropdown list of different eye color possibilities, and brown could be selected. This means that if the Subject's eye color is brown, the up left quadrant will be colored in orange (or as defined in the camel context). These quadrants are set up for the "Pedigree" for each family and can be different from one family to another.

- **cntn\_cf\_eyesColor**: Fixed choice custom field with possible values "Brown," "Blue," and "Green."
- **cntn\_cf\_hairColor**: Fixed choice custom field with possible values "Brown," "Black," "Blonde," and "Ginger."
- **cntn\_cf\_birthdate**: Additional custom fields for the Subject.
- **cntn\_cf\_age**: Calculated based on the birth date.
- **cntn\_cf\_firstname**: Additional custom fields for the Subject.
- **cntn\_cf\_surname**: Additional custom fields for the Subject.
- Diseases: Two diseases are imported along with the Pedigrees package.
  - **Diabetes**
  - **Myopia**

Any additional information can be shown on the pedigree graph for each subject using the "extraInfo" property.

### 13.1.5.1. Pedigree Camel-Context Configuration

The following routes have to be defined in your camel-context file:

```
<route id="slimsgate-visualize-proband-svg">
  <from uri="direct:visualize-proband-svg"/>
  <to uri="svgPedigreeGraphVisualizer"/>
</route>

<route id="slimsgate-visualize-proband-png">
  <from uri="direct:visualize-proband-png"/>
  <to uri="pngPedigreeGraphVisualizer"/>
</route>
```

The following beans have to be defined in your camel-context file:

```
<bean id="svgPedigreeGraphVisualizer"
  class="com.genohm.slimsgate.camel.beans.common.relation.PedigreeGraphVisualizer">
  <property name="pedigreeGraphFactory" ref="defaultPedigreeGraphFactory"/>
  <property name="laneDrawer">
    <bean class="com.genohm.slimsgate.camel.beans.common.relation.SvgLaneDrawer"/>
  </property>
</bean>

<bean id="pngPedigreeGraphVisualizer"
  class="com.genohm.slimsgate.camel.beans.common.relation.PedigreeGraphVisualizer">
  <property name="pedigreeGraphFactory" ref="defaultPedigreeGraphFactory"/>
  <property name="laneDrawer">
```

```

        <bean class="com.genohm.slimsgate.camel.beans.common.relation.PngLaneDrawer"/>
    </property>
</bean>

<bean id="defaultPedigreeGraphFactory"
    class="com.genohm.slimsgate.camel.beans.common.relation.DefaultPedigreeGraphFactory"
    p:recordColor = "WHITE"
    p:pedigreeTitleField = "cntn_id"
    p:pedigreeRecordName = "cntn_id"
    p:pedigreeRecordNameDescription = "Subject ID"
    p:deadCustomField = "cntn_cf_deceased"

    p:pointProbandWithArrow = "true"
    p:arrowColor = "BLACK"

    p:starMark = "true"
    p:starColor = "PINK"
    p:starMarkFullCustomField = "cntn_cf_affected"
    p:starMarkDescription = "Affected">

    <!-- Other possible marker (only one star marker can be shown at a time)
    p:starMarkEmptyCustomField = "cntn_cf_hasGlasses"
    p:emptyStarMarkDescription = "Has glasses"
    p:crossedMarkCustomField = "cntn_cf_longHair"
    p:crossedStarMarkDescription = "Has long hair">
    -->

    <property name="extraInfo" > <!--The fields defined in the extraInfo will be shown below
    the "pedigreeRecordName" for each subject -->
        <map>
            <entry key="cntn_cf_firstname" value = "First name"/>
            <entry key="cntn_cf_surname" value = "Surname"/>
        </map>
    </property>
    <property name="upLeftQuadrantFieldName" value="cntn_cf_upLeftQuadrant" />
    <property name="upLeftQuadrantConditions">
        <list>
            <map>
                <entry key="color" value="ORANGE"/>
                <entry key="fieldName" value="cntn_cf_upLeftQuadrantOrangeIf"/>
            </map>
            <map>
                <entry key="color" value="BLUE"/>
                <entry key="fieldName" value="cntn_cf_upLeftQuadrantBlueIf"/>
            </map>
            <map>
                <entry key="color" value="GREEN"/>
                <entry key="fieldName" value="cntn_cf_upLeftQuadrantGreenIf"/>
            </map>
        </list>
    </property>

    <property name="upRightQuadrantFieldName" value="cntn_cf_upRightQuadrant" />
    <property name="upRightQuadrantConditions">
        <list>
            <map>
                <entry key="color" value="BLACK"/>
                <entry key="fieldName" value="cntn_cf_upRightQuadrantBlackIf"/>
            </map>
            <map>
                <entry key="color" value="YELLOW"/>
                <entry key="fieldName" value="cntn_cf_upRightQuadrantYellowIf"/>
            </map>
            <map>
                <entry key="color" value="ORANGE"/>
                <entry key="fieldName" value="cntn_cf_upRightQuadrantOrangeIf"/>
            </map>
        </list>
    </property>

```

```

<map>
    <entry key="color" value="RED"/>
    <entry key="fieldName" value="cntn_cf_upRightQuadrantRedIf"/>
</map>
</list>
</property>
<property name="downLeftQuadrantFieldName" value="cntn_cf_downLeftQuadrant" />
<property name="downLeftQuadrantConditions">
    <list>
        <map>
            <entry key="color" value="BLACK"/>
            <entry key="fieldName" value="cntn_cf_downLeftQuadrantBlackIf"/>
        </map>
    </list>
</property>

<property name="downRightQuadrantFieldName" value="cntn_cf_downRightQuadrant" />
<property name="downRightQuadrantConditions">
    <list>
        <map>
            <entry key="color" value="BLACK"/>
            <entry key="fieldName" value="cntn_cf_downRightQuadrantBlackIf"/>
        </map>
    </list>
</property>
</bean>

```

## 13.2. SLIMS GATE Flows

SLIMS GATE Flows are automated procedures that are coded in Java and can be started by clicking a button and entering the needed input parameters. SLIMS GATE flows can be used to:

- *Interact with instruments:* Either send information to the instrument or process instrument outputs.
- *Export/Import data from SLIMS:* Export/import data in special formats.
- *Process data:* Analyze data using Java code or trigger custom scripts (R, Matlab, etc.) to be executed on the server.

## 13.3. SLIMS REST API

The SLIMS REST API allows developers to securely interact with SLIMS without using the user interface to perform operations such as fetch and update of data. SLIMS REST API can fetch information for reporting or perform data updates using external tools through the API without any manual input from end users.

A generic SLIMS REST API manual can be provided on demand.

Access to SLIMS REST API requires the following Role Access rights to work:

**Table 13.1. SLIMS REST Access rights**

Access rights Category	Access rights Name	Purpose
Setup > REST API	Can use the REST API	This permission provides access to the SLIMS REST server. If this access right is not enabled for a role user, REST

Access rights Category	Access rights Name	Purpose
		returns a HTTP status 403. SLIMS REST must be restarted to make the changes effective immediately.

## 13.4. Python API Integration

The SLIMS database can be accessed using slims-python-api. This allows any customer to access and analyze their data using Python scripts. Please contact a SLIMS Engineer to get the correct server configuration. The documentation about the slims-python-api can be found on <http://slims-python-api.readthedocs.io/en/latest/> and on the git repo <https://github.com/genohm/slims-python-api>.

## 13.5. Vaadin Integration

Custom Vaadin development can be done by a SLIMS Engineer. Please contact them for further information.

## 13.6. Prometheus Integration for Health Monitoring

Prometheus can be integrated with SLIMS REST to monitor the health of connectivity and various metrics of your on-premise SLIMS installation, such as the total time SLIMS spent fetching contents in the last minute. Please contact a SLIMS Engineer for further information.

## 13.7. Alamut Integration with SLIMS

SLIMS is capable of integrating with Alamut to enable gene sequencing in the same environment as sample control, order management, ELN, and the other modules. In SLIMS, sequencing starts with an NGS Workflow in the Workflows Module to perform sequencing on HiSeq and MiSeq. Then the files are prepared. The Alamut batch reads the files that were created from the previous pipeline and analyzes the genes.

### 13.7.1. Usage of Alamut in SLIMS

The integration with SLIMS allows the user to click on a SLIMSGATE that opens the Alamut software directly based on the file. The analysis is set up to run in the Alamut software, and once completed, the result is shown directly in Alamut Visual. The link to open Alamut Visual is present in the Variants tab. The steps to follow are listed below:

1. In an NGS Workflow when the Alamut step has been reached, click the SLIMSGATE button to open the Alamut software.
2. Start the Alamut analysis.
3. SLIMS will connect according to the set parameters and will ssh into the analysis server. Then it will launch the analysis and return the job ID of the Alamut batch. This should be done automatically and not require user input.

4. SLIMSGATE will show that it is in progress until the operation is completed, as Alamut determines what analyses should take place.
5. If there are no issues and the analyses are completed, a link to Alamut Focus is created in SLIMS where the results can be viewed.
6. Clicking on the link opens Alamut Focus locally.
7. In Alamut, the user can select one or more Variants and choose "Export back to SLIMS."
8. This starts communication between Alamut and SLIMSGATE to import the selected variant observations.
9. A link to the individual variants in Alamut Visual is created in SLIMS. This is accessed from the Variants tab in Content Management.

### 13.7.2. Configuring Alamut Integration

A client's engineers can follow these steps to configure the integration of Alamut and SLIMS.

⚠ Contact your IT department to get a personal license if one does not exist for the steps below that require license information.

#### Set Up Alamut Focus

1. Download Alamut Focus from: <http://downloads.interactive-biosoftware.com/AlamutFocus-1.2.0.dmg>
2. If using Windows, download from: <http://downloads.interactive-biosoftware.com/AlamutFocus-1.2.0-64-bit-Slims.zip>
3. Perform the installation process on the local computer: the one that SLIMS is used on.
4. Genohm activates the license (2 concurrent allowed) with the required credentials (what's given below are examples):
  5. • ID: The ID may be a text string, ex: CHARLIE
  - License: The license number, a numeric string, ex: 12345678
  - User: The initials of the license owner, ex: ABC
  - Location: Where the software was obtained, ex: [alamut-focus.interactive-biosoftware.com](http://alamut-focus.interactive-biosoftware.com)

#### Set Up Alamut Batch

- a. `/srv/alamut-batch/alamut-batch.pl`
- b. `/srv/alamut-batch/alamut-batch-daemon.pl`
- c. `/srv/alamut-batch/file.conf`

1. Install daemon on an analysis server, ex: the SLIMS server itself.
2. The following are the required filepaths:

#### Set Up Alamut Visual

1. Download Alamut Visual from: <http://downloads.interactive-biosoftware.com/>
2. Perform the installation process on the local computer: the one that SLIMS is used on.
3. Genohm activates the license (2 concurrent allowed) with the required credentials (what's given below are examples):
  4. • ID: The ID may be a text string, ex: CHARLIE
    - License: The license number, a numeric string, ex: 12345678
    - User: The initials of the license owner, ex: ABC
    - Location: Where the software was obtained, ex: alamut-focus.interactive-biosoftware.com
5. Enable HTTP-based access in the Preferences menu under Options > API.

## 13.8. Advanced Advice for Queue Aggregators in Protocols

This use case is an advanced topic to explore what a Queue Aggregator for a workflow protocol is and how to use it.

First it is important to understand these concepts:

- The records that you see waiting in a queue are called queue elements.
- A queue element is always composed of 2 components: a content, and a set of requests.
- To be posted in a queue, a queue element needs to have at least one associated request. You associate one or more requests to a content and a queue element(s) appear(s) in the queue for them.
- Initially, the default behavior of requests in the queue was that when several requests were associated to the same sample in an order, it would automatically post one queue element per request to the queue.
- Keep, Split, and Join queue aggregators were introduced to expand the options for that behavior.
  - **JOIN:** When a queue element gets posted on a queue with a set of requests, there will first be a check if that sample is already on the queue. If it is, then no new queue element is made but the requests are appended to the previous queue element for that sample.
  - **SPLIT:** When a queue element gets posted on a queue with N requests linked to it, instead N queue elements will be posted on the queue linked to each request separately.
  - **KEEP:** The queue elements remain posted. When coming from an order, when several requests are added to a sample, one queue element will be added per request (default behavior).

With those core concepts in mind, let's clarify the difference between a KEEP and a SPLIT queue. Imagine 3 requests are added to the same sample. There is a workflow with 3 protocols

in an inverted "Y" shape such that the first protocol splits into two other protocols. This workflow is for genetic analysis.

The 3 requests are:

1. Measuring the DNA Concentration can be fulfilled in the first protocol: "DNA Measurement"
2. Checking gene BRCA1 presence can be fulfilled in the second protocol: "Check BRCA1"
3. Checking gene BRCA2 presence can be fulfilled in the third protocol: "Check BRCA2"

The initial queue should be of the JOIN type because there is only one initial sample on which the DNA concentration is measured. It doesn't make sense to run the sample through that protocol 3 times.

However, the workflow forks in the second queue. This queue should use aggregator SPLIT type because the two remaining requests will go on to two different paths, and you want to see two elements in the queue, one for each request.

If KEEP was chosen for the second queue, there would only be one queue element with two requests in that queue after the first protocol, because the single queue element that entered the workflow with three requests will continue as it is with the two remaining requests.

If JOIN was chosen for the second queue, the same result as KEEP would be seen with the previous scenario.

Let's add one more thing to the scenario. There is a single element on the sample with two remaining requests waiting in the second queue. Now, an additional queue element arrives for the same sample but with different requests.

In a JOIN queue, the two elements will be merged together and all requests will be added to the element. Whereas for a KEEP queue, even if it's for the same sample, if a later queue element with additional requests is posted to the queue, the elements are not merged.

## 13.9. Grid Template Special Context Filters

Grid Templates are useful for customizing exports and supporting an Agilent OpenLab CDS workflow. There are three context filters to be aware of that allow administrators to define how grid templates should be filtered by the context of the user looking at it.

These can be used when configuring a grid template from the Grid Templates module, when editing the grid template of the Content module, when configuring a Graph Widget, when configuring a Queue type grid template for the protocol run grid, or from a specific step like an Agilent OpenLab CDS or NGS step.

- The *Content* root has the available filter "Context: Linked in"
- The *Result* root has the available filter "Context: Recorded in"
- The *ProtocolRunStepContent* root has the available filter "Context: Created in"

These possible values can be selected using a context filter:

- Current protocol run step

- Current protocol run
- Previous protocol run step

Combining these options provides some useful cases. For example:

- *"Context: Recorded in" equals "Current Protocol Run"*

A grid template like this is created by default when an Agilent OpenLab CDS step is created:

- The root "Content" with the filter "Context: Linked in" equals "Current protocol run step"
- A child "ProtocolRunStepContent" with the filter "Context: Created in" equals "Current protocol run step"

An example where the administrator chooses to expand content to a child of results so they can filter the result on Weight tests:

- *"Context: Recorded in" equals "Current protocol run"*  
Shows the weight measured in this run.
- *"Context: Recorded in" equals "Previous protocol run step"*  
Shows the weight measured in the previous step.

If the grid relies on the current protocol run step, only the results that were recorded in the current protocol run will show up whenever the grid is exported from within a protocol run step. If a SLIMS GATE flow exports the grid to CSV, the flow will get a grid filtered on the current protocol run step if executed from a run step.

If a grid has the filter "Context: Linked in" equals "Current protocol run step" and the grid is viewed in the Grids module, which means the user is not viewing it from within a protocol run step perspective, the content will not be filtered and they will see all content.

## 13.10. Reagents Default Usage Expressions

When the user links reagents during a Link Reagents step in ELN or Workflows and adds a usage or has one created automatically, the groovy will be evaluated to create a new usage for the reagent and subtract it from the amount field total. These are examples of valid value expressions to compute default usage for a reagent type "Buffer" that has an amount of material field "Volume" with the default unit "ml."

- Subtract 5 m from the linked chemical:

5

- Subtract 5 ml multiplied by the number of samples in this protocol step from the linked chemical:

5 \* N

- Subtract 5 ml multiplied by the number of samples in this protocol step and 25% volume overage from the linked chemical:

```
5 * N * 1.25
```

- Adapt the default usage of the linked buffer to how concentrated the buffer stock is:

```
concentrationFactor = content.cntn_cf_concentrationFactor
overageFactor = 25

return (5 / concentrationFactor)*N*(1+overageFactor)
```

# 14. FAQ

Answers for Frequently Asked Questions are grouped in this chapter.

## 14.1. Miscellaneous

### 14.1.1. Transport Error when Updating Multiple Records

Updating 839 content records ends with the following error:

```
Transport error - HTTP code: 503 for URL: https://customer-website.ch/slms/slms/sc/IDACall [^]
```

Does this mean that the samples were or were not updated?

Am I supposed to close the "edit selected items" window or "close" after this error?

The samples are updated and you can close the form after getting the error. Here is the detailed explanation of the error cause:

After using the multiple edit and confirm, the request is sent to the server where it is processed. Once the request is sent, the action will be executed on all records. The warning you get is normal because SLIMS doesn't get an answer from the server for a long time so it complains but the action is still being executed on the background (server side).

If you refresh the grid after a few minutes, you will see that all records got updated.

### 14.1.2. Scanner is not Reading Certain Characters

The scanner is not reading the '-' character in 2D barcodes.

How do I change the scanner or computer settings to fix this?

This occurs when the scanner is not set to the correct keyboard setup that the computer uses, like if the computer is set to Swiss French but the scanner is set to English. There are various manuals that support how to change your settings, but the one to use depends on the type and brand of scanner used. These directions can be used as a general guide to fixing the settings. In the case of Gryphon and Xenon keyboards, the correct country mode can be activated with steps like the following.

Gryphon Scanner: Change Country Mode

1. Link to Gryphon Scanner Website with Guides [[http://www.datalogic.com/eng/products/hand-held-scanners/gryphon-i-gbt4100-pd-169.html#file\\_1348](http://www.datalogic.com/eng/products/hand-held-scanners/gryphon-i-gbt4100-pd-169.html#file_1348)]
2. Scan the "Enter/Exit Programming Mode" barcode to enter programming mode
3. Scan the relevant "Country Mode = COUNTRY" barcode
4. Scan the "Enter/Exit Programming Mode" barcode to exit programming mode

Xenon Scanner: Activate Correct Country Mode

1. Honeywell Xenon 1902 Quick Start Guide [<https://country.honeywellaidc.com/CatalogDocuments/CrdIss-NG2D-QS%20Rev%20D.pdf>]
2. Scan the "Program Keyboard Country" barcode to activate programming mode
3. Scan the relevant Country Code barcode
4. Scan the "Program Keyboard Country" barcode again to end programming mode and save the change

---

# About this Book

*www.agilent.com*

© Agilent Technologies, Inc. 2022

February 2022

R4526A-AM6.7.5.0