Agilent

Agilent OpenLab Sync

OpenLab Sync 7.4 Version

Technical Release Notes

Notices

Document Identification

DE-017715 Rev. A.00

September 16, 2026

Copyright

© Agilent Technologies, Inc. 2026

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 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 OpenLab Sync 7.4 revision of Agilent OpenLab Sync 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 252.227-7015 (Technical Data - Commercial Items) and DFARS 227.7202-3 (Rights in Commercial Computer Software or Computer Software Documentation).

Safety Labels

NOTICE

NOTICE indicates a situation which, if not avoided, results in damage to or destruction of the device or data.

CAUTION

CAUTION indicates a hazardous situation which, if not avoided, could result in minor or moderate injury.

WARNING

WARNING indicates a hazardous situation which, if not avoided, could result in death or serious injury.

DANGER

DANGER indicates a hazardous situation which, if not avoided, will result in death or serious injury.

Table of Contents

1 Introduction 7
2 Installation and Upgrade Changes 8
2.1 Installation Changes 8
2.2 Customization Changes 9
3 Enhancements in Version 7.3 Bridge 12
3.1 Create Recipes as SOPs 12
3.1.1 Restrict SOPs by Requestable 14
3.1.2 Refined Links Between Ingredients and Mixes 14
3.1.3 View a Mix's Ingredients with their Amounts 15
3.2 Groups for Training Regulation 15
3.2.1 Group and Subgroup Selection 16
3.3 Actions Support in Bench View 17
3.3.1 Grid Template Blocks in Bench View 18
3.4 Improved Analytical Result Experience 18
3.4.1 Analytical: Result Order of Operations * 19
3.4.2 Analytical: Coding System for Rules 19
3.4.3 Analytical: 2D Results Grid * 20
3.4.4 Analytical: Repeat, Requeue, Cancel Actions on Results * 20
3.4.5 Analytical: Check Results with an External Database 21
3.4.6 Analytical: Schedule with Analyte Groups 22
3.4.7 Analytical: Create Aggregated Result Rules on an Analyte Group 22
3.4.8 Analytical: Result Calculations Based on Other Results 23
3.4.9 Analytical: New "Result Block" Rule Type 25
3.4.10 Analytical: Result Block Rules can Access Specification Outcomes 26
3.4.11 Analytical: Import Specification Rules with a Specification Type * 26
3.5 Upgraded Rich Text Editor 27
3.6 Track Reagents with Simplified Protocols 28
3.7 Edit Results is Controlled by a Permission 28

3.8 Ingredient Removal is Controlled by a Permission 29
3.9 Edit Content Type after Content is Created * 29
3.10 Import Multiple Content Types * 30
3.11 Import a Product via Excel * 30
3.12 Requestables can be Restricted by Content Type/Category * 31
3.13 Dashboard can Display Report Previews 31
3.14 Download Graph in Dashboards * 32
3.15 Relink Content with a Copied Order * 32
3.16 Cancel Loading of Slow Grids * 32
3.17 Remove Content from a Location * 33
3.18 Nice Multiple Choice Pop-up for Extensions Flows 33
3.19 Client IP Address Lab Setting 34
3.20 Scheduled Jobs are Run by OpenLab Sync Extensions User * 34
4 Enhancements in Version 7.4 Release 35
4.1 SLIMS is Renamed to OpenLab Sync 35
4.2 Reusable Simplified Protocols as Templates 36
4.2.1 Static Reusable Protocols 37
4.2.2 Planned Ad Hoc User Reuse 38
4.2.3 Unplanned Ad Hoc User Reuse 38
4.3 Expansions to Bench View for OpenLab Sync 39
4.3.1 Recipe Blocks are Supported 39
4.3.2 SOP Blocks are Supported 40
4.3.3 Link Location Blocks are Supported 40
4.3.4 Create and Edit Usages in Bench View 41
4.3.5 Create Manual Results in Bench View 42
4.3.6 Add Ingredient to Mix (Without Recipe) 42
4.3.7 Support Software Date Formats and N/A Values * 43
4.3.8 Customizable Grid Columns for Bench View 43
4.3.9 Edit Content Details and Scan Linked Samples * 44
4.3.10 Select Instruments in Form Blocks 45
4.3.11 Expanded OpenLab Sync Extensions Flow and Macro Support 45

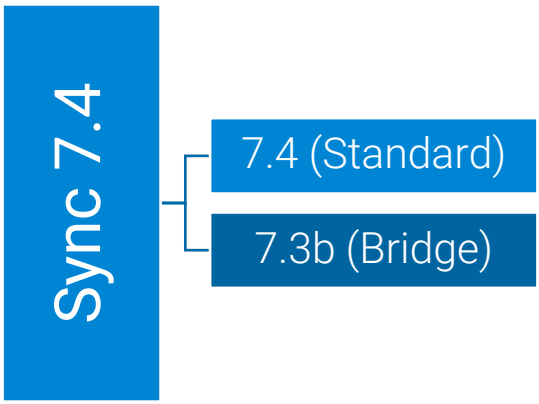
4.4 Updates to Requester Portal for OpenLab Sync 46
4.4.1 Import Multiple Content Types in Portal with Excel * 46
4.4.2 Customizable Grid Columns for Requester Portal 46
4.4.3 Add Requests Table * 47
4.5 Flags May be Used in Simplified Protocols 48
4.6 Recipe-Related Improvements 49
4.7 Aggregated History for Protocol Runs * 50
4.8 Expanded History on Test Datatype * 51
4.9 Refreshed Content Type Icons* 51
4.10 Drag and Drop to Reorder Sample Sheet Rows * 51
4.11 Import Protocol Run Step Content Values via Excel 52
4.12 Requests Respect Content Type Restrictions * 53
4.13 Control the Download of Attachments * 53
4.14 New Queue Aggregator Type: Automatic * 54
4.2 Automations Quality of Life Changes 55
4.2.1 Scan to Execute Extensions Flows * 55
4.2.2 Empty Number of Content is Allowed in Macros During Design * 55
4.2.3 Mix Rules can Set Values 56
4.3 Can Create Multiple Content with Same ID * 56
4.4 Able to Restrict Roles to Edit Dashboard 57
4.5 Version SOPs in Batch * 57
4.6 Support for Kit Collection Store Package 58
4.7 NGS Content Queue Progression 58
4.8 Product Specifications Form Display Improvements* 59
4.9 Orders Hidden in ELN by Default 59
4.10 Expanded Tables for Conditional Value Expression Rules 59
4.11 Agilent Store Package Audit Collection * 61
4.12 OpenLab Sync Installations Come with SI Units * 61
5 Store Content Related to this Version 62

5.1 Pharma QC 1.0 (Small Molecules) Collection 62
5.2 Training Plan Package 63
6 Deprecated Features 64
7 Appendix 1: OpenLab Sync License Information 65

1 Introduction

This document provides a cumulative listing of the major feature modifications made in version 7.4 of the OpenLab Sync Software, which was known as SLIMS until version 7.4. It includes the installation and use of enhancements and any requirements needed to enable the new features such as role permissions and OpenLab Sync lab settings.

A **bridge** release (marked with the letter **b** in its commercial name) includes major or minor enhancements in a new version that is supported for several months. Multiple bridge versions are used to build new functionality toward a **standard** release and provide early adopters a chance to test the latest updates before production. The standard release (accumulated with its bridge releases) is actively supported for 3 years with 1-year limited support. The advantages to this versioning approach are that regulated and non-regulated customers need to upgrade less frequently and OpenLab Sync is developed and tested in more frequent and feature-focused versions.



7.3b is the first bridge release building toward a cumulative 7.4 version that is supported for 3 years with 1-year limited support. The updates in this document are cumulative for OpenLab Sync 7.4 so customers can see the features that were added in each version making up OpenLab Sync 7.4.

For our Regulated Customers

When any change is made to Agilent software, the validation status of the software needs to be re-established by the user/customer.

Whenever software is changed, a validation analysis should be conducted not just for the validation of an individual change, but also to determine the extent and impact of that change on the entire software system.

2 Installation and Upgrade Changes

The following upgrades impact the installation and running requirements of OpenLab Sync or enhance the customers' ability to programmatically customize OpenLab Sync.

2.1 Installation Changes

These upgrades impact the installation and running requirements of OpenLab Sync. The changes for OpenLab Sync are as follows:

- When Requester Portal was developed with OpenLab Sync 7.2, it could only be installed on the same server as OpenLab Sync. Development of the installation methods progressed so the Portal can be installed on a different server from OpenLab Sync for Windows and Linux environments. The benefit is OpenLab Sync can be used internally in the lab with limited connection from the outside world and Requester Portal can be installed on a separate server to provide access outside the lab to your customers.

  Standalone installation of the Portal still requires connections to OpenLab Sync with these requirements:

  - Database
  - Hazelcast port
  - OpenLab Sync must be running
  - The Portal and OpenLab Sync must be on the same patch version.

  These details and the full installation procedure are described in the OpenLab Sync Installation Manuals for Windows and Linux.

- Endpoints were added to REST API so that you can POST a request to a designated URL to return and request licenses programmatically instead of doing it manually in the License module. This is useful when you copy a production instance to a test instance and want to return your licenses so the licenses for production are not interrupted.

2.2 Customization Changes

These upgrades are relevant for customers who do their own customization in OpenLab Sync. The changes for version 7.3b –7.4 are as follows:

- Part of the V2 REST API documentation is available with version 7.4. OpenLab Sync engineers and customer engineers can access the documentation by changing the URL path after `openlab-sync/…` to `slimsrest/doc`. On the interactive documentation page, a dropdown in the upper-right corner allows users to toggle between the V1 and V2 documentation. Currently, the V2 documentation includes endpoints for fetching macros and Extensions flows, an improved method for fetching records, and methods for creating or updating an order. A technical description of the API is also available as a YAML file, which can be downloaded from the link below the page header.

  The PDF of the documentation includes V1 and V2 methods.

- SLIMS GATE flows were renamed to OpenLab Sync Extensions flows in version 7.4. Parameters, file paths, and other code-dependent names were not changed so that existing flows and configuration will not break. The changed names can be seen in the UI.

- Long-running OpenLab Sync Extensions flows provide a message when they time out on the client side to confirm that they are still running on the server side. When the info message is closed, the Extensions window will now also close so the user doesn't think that they must keep it open for the flow to finish. The status of the flow is available in the Notifications menu.

- Toast messages can be configured to provide feedback in Bench View (formerly called Bench UI) after Extensions flow steps and at the end of the flow. However, there is only a default message available for the end of the flow: "<extensions name> has been executed successfully". To define a custom message, this line can be added within the Extensions plugin:

  ```
  slimsProxy.setToast(String toast)
  ```

These were the changes to OpenLab Sync API services until 7.4. The specific OpenLab Sync versions the services can be found on are:

- 6.9.12, 7.0.3. ArchiveGeneratorService: A service was added so plugins can be written that add extra options for getting a generated zip archive of ELN records like projects, experiments, and runs.
- 6.9.15. TransactionService: Wrap code in a new database transaction. Useful when you need to create a sub-transaction or when you are writing code that interacts with the database and may end up being called by code which is not transactional.
- 6.9.23. InteractiveImportApi: Expanded to import records into OpenLab Sync from a Sync-formatted data file with the option to create a separate transaction for the import or not.
- 6.9.24. AddRequestService: Expanded so there is an option for all requestables to allow or disallow the creation of duplicate requests.
- 6.9.27. ContentTypeMoveCategoryService: Move a content type to a different category.

- 6.9.33. PluginConfigurationService: Methods to get a plugin's configuration YAML as a String, serialize the plugin's configuration YAML to a matching object, set a plugin's configuration to a given YAML String, or set a plugin's configuration to an object that the service will transform into a YAML String.
- 7.0.4. FilledPositionService: Calculates the number of filled positions in a chosen location.
- 7.1.3. SampleSchedulerInstrumentService: CRUD operations for Sample Scheduler for OpenLab instruments that are for OpenLab analyses.
- 7.2.10. ExperimentRunStepStatusService: Change the status of an experiment run step (e.g. put a Page in Amending, Amending review, or In review status).
- 7.2.11. LicenseDetailsService: Service for retrieving license details and user counts
- 7.3.0. findByPks: A method was added to the API so plugin developers can use `findByPks` from all RecordQueries instances that they can inject in their plugins.
- 7.3.0. Dynamic choice custom fields have a dynamic filter expression Groovy box where more options have been added to the insert menu for common criteria you might use in this context.
    - "Criteria" inserts this expression template:

      ```
      [fieldname: "name", operator: "equals", value: Object]
      ```
    - "Criteria operators" provides a list of operators.
    - "Combining criteria" provides a list of operators to combine criteria into compound criteria.
- 7.3.0. Three services were added for use in plugins to support recipes.
    - RecipeExecutionService: Add recipe executions to a recipe block of a protocol run
    - LinkIngredientToRecipeService: Link/unlink ingredients to the instructions of a recipe execution and record how much material was used
    - RecipeRunService: Fetch recipe executions with contextual information like the recipe that was used, the mix that was created and so on.
- 7.3.1. AdHocAnalyticalWorkflowService: Schedule some requests into an analytical workflow and check them for the specifications of some list of products. This allows an Extensions flow to be built that can use one product's specifications for some sample types and a different product's specifications for another type (like blanks) in the same analytical workflow.

  With the option "Only create requested results" enabled, the created results are limited to just the tests requested by the plugin.
- 7.3.1. Trigger API in Mix Rules. There are several Trigger API entries in the Insert menu for mix rules in the Rules module. This change ensures that any triggers defined in a mix rule are evaluated when a mix is created or when content is added/linked into the mix. The rule evaluates whether mixes are created with the dedicated Mix action or are created in Recipe blocks.
- 7.3.1. Batch API is available only for results. It is supported by Excel import and File importer plugins and requires usedatabasequences and batchsave.enabled flags to be enabled, and then large numbers of analytes and samples are much faster to import

- 7.4.0. OpenLab Sync Extensions flows that upload files can be run in macros. This usage was unsupported in previous versions but can now be used. An Extensions flow with a step that has the `<type>File</type>` parameter can be used in a macro step of type "Execute OpenLab Sync Extensions Flow." Then, the user can run the macro and upload a file during its execution.
- 7.4.0. OpenLab Sync MCP Server is a standalone Spring Boot application, like OpenLab Sync Extensions, that is available for engineers who wish to expose OpenLab Sync data and operations as tools, resources, and prompts to AI clients that support the Model Context Protocol. Only users with the role "Can use MCP" are authorized to call MCP tools, resources, and prompts. A full description is available in the OpenLab Sync Development Manual.
- 7.4.0. mix.setValue is available in the Groovy expression box for mix rules. It allows mix rules to set values from linked ingredients to the mix. This note has more details.
- 7.4.0. recipeMeasurement usage is now available in Extensions flows. A measurement Extensions flow can be made to weigh ingredient amounts using a balance and import the values to OpenLab Sync or Bench View recipe instructions. Flows with usage `<usage>recipeMeasurement</usage>` can be used in your recipe instructions. This note has full instructions and context.
- 7.4.0. `actionContext` is available from the Groovy insert menus for field value expressions, rules, test datatypes, recipe instructions, specification types, and macro value expressions. Like `macroExecutionDetails`, `actionContext` provides contextual information. Currently, it only provides `orderPk()`; for example, if the macro is run in the context of an order. A field with a `return actionContext.toString()` value expression in a step of type "create content," "edit content," "derive content," or "copy content" shows the order in which it ran. If the macro doesn't run in the context of order, then order will be null. Only the order context is supported at first, but this change enables development of more contexts in the future.
- 7.4.0. `StabilityStudyExpansionService` is available to programmatically create a new stability study timepoint in an existing stability study based on a stability study timepoint template. The operation and maintenance of stability studies are rigid because they are fully defined before they start, so this service provides a way to lengthen a study when needed by adding further timepoints. This works by adding timepoints from a future version of the stability study template, so if the study was started on version 1, version 2 of the template must have the added timepoints. The content must already be enrolled in the study. Additionally, the storage condition being used must exist in version 1 of the template.

  `StabilityStudyEnrollmentService`: The newly created timepoints will not have content enrolled in them by default, but this service allows them to be handled programmatically as well.

3 Enhancements in Version 7.3 Bridge

Enhancements are improvements that have been made to the system as a direct result of your feedback.

Manual updates are features that need to be enabled, require back-end configuration, or require set-up in your instance to start using, such as new modules and major functionality changes. Default updates come ready to use and do not require setup, such as design-only changes or minor updates that improve the default functionality. Default updates are designated with an asterisk (*) next to the title.

The following updates were included in OpenLab Sync Version 7.3b. This was the last bridge version before the product name changed from SLIMS to OpenLab Sync.

3.1 Create Recipes as SOPs

TIP

Uses protocols in ELN and Workflows. Requires:

- Role functionality access rights Workflow Management and Workflows with a Premium license

Recipes are a new SOP type added with this release that are used in simplified protocols. Recipes allow you to define the detailed steps it takes to get a mix while providing guided instructions to the user. The recipe is used in a Recipe block in simplified protocols for ELN or Workflows. The main setup is in recipes so you can make generic protocols to use them and choose which recipes are run based on requestables. This allows a lab to run hundreds of methods with less setup.

Recipes are created in the SOPs module. They are defined by a target content type — the mix that will be produced and the base amount you want to obtain of the mix. In the recipe you can define each ingredient (the content types that are part of the mix), the target amount you will take of each ingredient, whether that amount is proportional with or without any Groovy expressions to make the proportions scale differently, whether there is tolerance in the deducted amount or not, and associate any macro that will be used to create the mix. Recipe steps can be included that don't use ingredients by leaving out the content type on the step (ex: Heat the water in a 500 ml beaker). Ingredient filters can be used to filter fresh and non-expired ingredients.

Once a recipe is versioned, the newest version can be used in a recipe block of a simplified protocol. You can restrict recipe blocks so only certain recipes can be executed in them to guide the user's work. Requestables/analytical methods in the Workflow Management module can also restrict the recipes (and SOPs) available in their protocol runs. Then, the recipes available at execution will be the intersection of what's allowed by the recipe block

and by the requestable. Recipe blocks can also create executions of each recipe automatically, but only if there is a restriction on the protocol block and/or the requestable. This prevents a recipe block from running every recipe that exists.

When the protocol is run, the user gets each recipe execution as a collapsible section with the identifiers of the mix that will be generated and the instructions for each step. That way, the container for the mix can be labeled before the mix is complete while ingredients are added or removed from the mix's composition. Any automatic executions are performed when the protocol is started, or the user runs manual executions. If there are multiple recipes possible for a block, the user chooses the desired recipe(s) from a dropdown. Then they associate the ingredients (fresh ones depending on the filter you set) that were used and enter the amounts that were taken of each. They can also adjust the size of the mix and see the proportional scaling of ingredient amounts they will need.

Note: The target mix content is prepared from the recipe block, so the user doesn't have a chance to fill any required fields (such as the content status). To use recipe blocks, you will need a rule or macro to fill any required fields on creation.

- You can create a macro with a single create content step that sets the values for the status and any additional required custom fields, then select the macro in the Recipe SOP configuration.
- Otherwise, you can create a conditional value expression rule with an expression like:

  ```
  content.setValue("cntn_fk_status", "stts_cntn_pending");
  ```

  It can be restricted to the recipe's content types and should execute only “on creation” and with the timing "set values."

Action blocks otherwise use the mix from recipe blocks as input for their operations. Macros only use recipe blocks as input and take “All” mixes from the executed recipes or the user’s “Selected samples” or “Scanned” mixes. Extensions flows in action blocks can use recipe blocks as input or output depending on their design. If the recipe is the input block, the flow gets the pk of the mix, and with the recipe as the output block, the flow gets the pk of the block and decides what to do with it according to its scripting.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.4. Simplified Protocol Design → 8.4.3. Defining Block Types → 8.4.3.10. Recipe
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.8. Electronic Lab Notebook (Routine) → 8.8.3. Simplified ELN → 8.8.3.1. Blocks
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.14. Workflow Management → 8.14.2. Requestables Tab
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.19. Analytical Workflows → 8.19.2. Workflow Management for Analytical Workflows → 8.19.2.2. Analytical Methods Tab
- ✓ OpenLab Sync Administration Manual → 9. Audit Tools and Quality Control → 9.2. SOPs
- ✓ OpenLab Sync Administration Manual → 10. Automations → 10.2. Fields → 10.2.2. Default Fields

3.1.1 Restrict SOPs by Requestable

As a benefit of these updates, SOPs can be restricted in a similar way. In a protocol SOP block, you can select an SOP to provide just one SOP at runtime, restrict to a selection of SOPs to give the user a choice, or leave it unrestricted to provide all SOPs as a choice. The requestable/analytical method of a workflow can be restricted to a combination of SOPs and recipes in the same dropdown. In this way, you can restrict the SOPs that are relevant for your lab's methods. OpenLab Sync will offer the intersection of SOPs from the protocol and requestable.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.4. Simplified Protocol Design → 8.4.3. Defining Block Types → 8.4.3.9. SOP
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.14. Workflow Management → 8.14.2. Requestables Tab
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.19. Analytical Workflows → 8.19.2. Workflow Management for Analytical Workflows → 8.19.2.2. Analytical Methods Tab

3.1.2 Refined Links Between Ingredients and Mixes

TIP

Some beneficial changes came with the development of recipes for ingredients and mixes used without recipe blocks.

Usages from ingredients in a Link Content block on a protocol page can be taken and associated with a mix in a Link Content block on the same page. In this case, the mix can be created during the protocol run or premade in the Content module. If there is a unique usage per ingredient on the same page as the linked mix, then its usage is associated to the mix. If there is more than one usage of an ingredient, OpenLab Sync doesn't know which ingredient to associate, so they are not associated with the mix.

Extensions flows can be made to associate usages as well. If a flow takes ingredients from an input Link Content block and has an output Link Content block for the Mix, then users can select one of each desired ingredient and run the flow. The ingredient usages are associated to the mix. They can add a second usage to some of the ingredients and run the flow again and get a second mix associated to the second usage of the ingredients.

Macros can be made to work with extra ingredients for mixes in simplified protocol runs. The macro needs an Add Usage step and an Add Ingredient to Mix step (followed by any other desired steps like Print Labels, Move Content, etc.). It is used in an Action block with the ingredients of an input Link Content block and the mixes of an output Link Content block. The user selects ingredients and the mix to associate and executes the action. The macro takes the usages and adds the ingredients to the mix.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 5. Content Management → 5.2. Contents → 5.2.15. Mix Contents
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.4. Simplified Protocol Design → 8.4.3. Defining Block Types → 8.4.3.3. Link Content
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.8. Electronic Lab Notebook (Routine) → 8.8.3. Simplified ELN → 8.8.3.1. Blocks

3.1.3 View a Mix's Ingredients with their Amounts

As another benefit of these updates, the capture of usages is improved in other modules. A mix can be created via a recipe block. It can also be made via a mix macro in an action block sandwiched between a Link Content block for the ingredients that captures usages and a Link Content block for the mix. An "Amount" column was added to the Mix tab in the Content module to show how much was used. You can also see what mix the amounts were taken for via the Mix button in the Derivations tab, and the Show Ingredients right-click action in a protocol content block with a mix in it. The Usage grids in a Link Content block of an ingredient also shows the "Mix" column to identify where it was used in that protocol run.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 5. Content Management → 5.2. Contents → 5.2.1. Content Module → 5.2.1.1. Content module Tabs
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.8. Electronic Lab Notebook (Routine) → 8.8.3. Simplified ELN → 8.8.3.1. Blocks
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.8. Electronic Lab Notebook (Routine) → 8.8.7. Track Usage

3.2 Groups for Training Regulation

TIP

Impacts access to execute protocols in ELN, Workflows and Bench View. Requires:

- Role functionality access rights Workflow Management and Workflows
- License Premium and/or Basic (Read-Only) (see Appendix 1)

The groups defined in the Groups module are now called "Standard Groups." The module was expanded so you can set up a new group type, "Training Groups," to which users can be added. The "Start Membership Date" and "End Membership Date" are used to indicate when a user's training begins and ends, or in other words, when the training is valid.

About the dates:

- The start and end date are not required but the end date cannot be before the start date.
- The start date can be in the future to indicate when a user will be trained.

- The end date can be blank to indicate training that is tracked but does not restrict access to any records or procedures.
- Users who try to perform operations restricted to the training will be unable to if their end date is expired (if the current date is past the chosen end date).

Once training groups are set up, various entities in other modules can use them in their restriction settings. The multiselect option "Only selected users can run this protocol/perform this status transition" allows you to restrict permissions to users, role groups, standard groups, and training groups. This option appears in the following areas:

- Protocols in Workflows: Restrict the user on the protocol configuration in Workflow Management. If the user is not one of the selected users, roles, is not in the standard group, or if their training date has expired in any of the required training groups, they cannot run the protocol, and existing runs are read only for them.
- Protocols in ELN: Restrict the user on the protocol configuration in the Protocols module. If the user doesn't match the selection or any of their trainings have expired, the restricted protocol doesn't show up in the list of protocols they can run and existing runs are read only for them. In an experiment, they get an error if they try to start a restricted protocol run.
- Status Transitions: Restrict the user on a status transition. If the user doesn't match the selection or any of their trainings have expired, they can't perform that transition.
- Bench View: The user can't perform runs or status changes that are restricted if they don't match the selection or any of the selected trainings have expired. Restricted protocol runs are accessible but read only.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 4. Access Control and Session Management → 4.3. Access Rights → 4.3.2. Groups
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.5. Default Protocol Design → 8.5.1. Protocol Tab
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.8. Electronic Lab Notebook (Routine) → 8.8.3. Simplified ELN
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.14. Workflow Management → 8.14.1. Workflow Graph Tab
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.16. Bench View: OpenLab Sync Interface for Tablets → 8.16.1. Configuring OpenLab Sync for Bench View
- ✓ OpenLab Sync Administration Manual → 9. Audit Tools and Quality Control → 9.6. Status → 9.6.3. Statuses

3.2.1 Group and Subgroup Selection

There were a couple of additional benefits from the development of training groups.

In a standard group there is a grid for "User rights for users of another group." You could already add other standard groups as subgroups and identify what access they had to the selected group. You can now also add roles as a subgroup (ex: Administrators role group, Technicians role group, etc.)

Additionally, training groups may be selected in dynamic choice fields with the "Group" value table.

INFO

Reference:

✓ OpenLab Sync Administration Manual → 4. Access Control and Session Management → 4.3. Access Rights → 4.3.2. Groups → 4.3.2.4. Adding Sub-groups and Roles to a Standard Group

3.3 Actions Support in Bench View

The first implementation of Bench View supported only basic macro content steps that did not require user input. This release expands the capabilities of Bench View to handle macros that require user input, simple OpenLab Sync Extensions flows that do or do not require user input, support more macro scenarios, and execute them correctly.

Contents can now be scanned into action blocks. Scanned contents will be linked to the selected link content input block if they were not already linked (and popped from the queue if the protocol is from a workflow). If the macro in the action block requires input – meaning it has "Pause before execution" and "Specify new value" for some of the fields – then a pop-up is provided to enter data in Bench View.

Result blocks now have a customizable view for Bench View just as Link Content blocks do.

Macros and Extensions flows with multiple steps run in sequence and have a stepper (breadcrumbs) to show the order of execution and what step the user is currently on. Steps that require input provide fields and pause for entry. Steps that do not require entry (and Extensions flow steps that are hidden) run automatically with an indication that something is being processed, then are completed and arrive at the next step. If no step in the macro requires user input (or flows in which all the steps are hidden), a pop-up is not shown, and the macro runs automatically in the background. A pop-up banner confirms when the action has successfully finished.

Choosing a new location and/or position provides the user with a location tree and position grid to choose from, depending on what is made visible in the macro setup. The location tree starts on whichever level was selected as the location in the macro and can be restricted with "Use selection as root at execution." The user's selection overrides the macro's position settings because their choice represents the sample position in the real-life lab. The location and position can be set automatically and hidden, so the user doesn't need to select them. Bench View also supports "Use previously created location" in multi-step macros.

These macro step types are now supported:

- Content actions: Create, Edit, Copy, Move, Derive, Mix, Aliquot, Dilute, Enroll, Add Content Event, and Add Usage
- Print Labels

Extensions flow steps currently support these field types:

- String
- Text
- Integer

- Float
- Boolean
- Date
- Datetime
- Dropdown

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.16. Bench View: OpenLab Sync Interface for Tablets → 8.16.1. Configuring OpenLab Sync for Bench View
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.16. Bench View: OpenLab Sync Interface for Tablets → 8.16.2. How to Use Bench View → 8.16.2.3. Using Blocks in Bench View
- ✓ OpenLab Sync Administration Manual → 10. Automations → 10.9. Macros → 10.9.4. Action Configuration Options

3.3.1 Grid Template Blocks in Bench View

When a simplified protocol is run in Bench View, its grid template blocks now have a checkbox for each sample's row. If the grid template block is used as input for an Extensions flow in an action block, then the user can tap the checkboxes to choose samples and run the action on them. The Extensions flow needs the setting "Requires manual input selection" activated so the action button will become enabled after the user selects the desired samples.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.16. Bench View: OpenLab Sync Interface for Tablets → 8.16.1. Configuring OpenLab Sync for Bench View
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.16. Bench View: OpenLab Sync Interface for Tablets → 8.16.2. How to Use Bench View → 8.16.2.3. Using Blocks in Bench View

3.4 Improved Analytical Result Experience

Changes were made in patches after 7.3 that improved the performance and visualization of analytical results, especially for labs with a lot of results, specifications, or rules. The visual changes include a coding system for rules to see at a glance which of them passed and which failed and a 2D grid view that focuses on just the results and analytes/tests they relate to.

New features allow you to schedule requests with analyte groups, get aggregated results on analyte groups to get the sum of all the group's analytes, check multiple results measured at the same time in result blocks, and compare the outcome of other regular result rules and specifications. Technical changes enable an OpenLab Sync Extensions

flow to compare specification results with limits from an external database and several related API improvements.

3.4.1 Analytical: Result Order of Operations *

This change affects the technical operation of rules and results to make them more performant. Result values used to be set to "Available" status and then all checked to see which could go to "Verified." Results now go to "Verified" status first. After they are all saved, test specifications and result rules are run, then result blocks are run, rule evaluations become available, results and extra custom fields are updated when necessary, and finally, if something caused the results to fail, they are set to "Available."

As a side effect, specifications and rules are no longer scheduled and it is not possible to see what rules will be executed on the result. Some technical tables storing that information have been retired. This change doesn't affect legacy customers who require that information.

"Show rule evaluations" was an existing option on individual result rows, but it also appears in the result block menu, Result Overview tab in the Orders module, and the Results tab in the Content module. The grid has been consolidated to make the min/max/expected value columns less busy.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 5. Content Management → 5.2. Contents → 5.2.20. Create Results
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.8. Electronic Lab Notebook (Routine) → 8.8.6. Results → 8.8.6.3. Status of Results
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.11. Orders → 8.11.1. Order Life Cycle

3.4.2 Analytical: Coding System for Rules

TIP

Set up in Products and Specifications module and Rules module. Used primarily in analytical workflows.

Result rules and product specification rules can now be given a brief code (for example, Nickel specification "Ni," Concentration too high "HIGH"). The "Codes" column is displayed in result blocks in simplified protocols as a comma-separated list. Labs with many specifications per analyte look at many rule evaluations, so users who know the codes will be able to tell immediately which rules failed on a result value without needing to open the details of each rule. The codes are also available to add to grid templates that accompany Jasper reports so the data can be included in reports.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.8. Electronic Lab Notebook (Routine) → 8.8.3. Simplified ELN → 8.8.3.1. Blocks
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.18. Products and Specifications → 8.18.2. Specifications Tab
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.19. Products and Specifications → 8.19.6. Analytical Use Case (Updated to show Code in examples)
- ✓ OpenLab Sync Administration Manual → 10. Automations → 10.1. Rules → 10.1.2. Result

3.4.3 Analytical: 2D Results Grid *

The option “Show in 2D View” was added to result blocks in simplified protocols in ELN and workflows and result grids in the Orders and Content modules. This option opens a window that focuses on only the result values in the grid.

In Workflows, the results are colored yellow for values with warnings and red for values with errors (or if there are both, the cell will be red). The user can hover over the colored cells to see a tooltip with a comma-separated list of rule evaluation messages.

- If the setup has different tests but no analytes, the grid has one line per content and one column per test.
- If the setup has different analytes with the same test, the grid has one line per content and one column per analyte.
- If the setup has different analytes and tests, the grid has one line per content and one column per analyte and test. The column header displays the test and analyte name (if there is one).

This visualization is only for cases with one result for one analyte and test. An error is provided if there are multiple results for an analyte or test.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 5. Content Management → 5.2. Contents → 5.2.20. Results
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.8. Electronic Lab Notebook (Routine) → 8.8.3. Simplified ELN → 8.8.3.1. Blocks
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.11. Orders → 8.11.1. Order Life Cycle

3.4.4 Analytical: Repeat, Requeue, Cancel Actions on Results *

Results that were created by mistake or were intentionally not filled could cause an empty result row. You can now cancel or reject result rows that have empty result values.

You could already repeat results or requeue samples if the result values needed to be rechecked, but more control has been added to permit or restrict results from being repeated. The result block configuration has an option for “Can repeat results” now so the administrator can choose whether the repeat result action is shown or not in each results

block. The option is disabled by default for new blocks, though if it was enabled in a prior version of OpenLab Sync, it is enabled when the instance is migrated.

Protocol runs disappear from the Workflows module once they are finished which prevented samples from being requeued from finished protocol runs. There is now a "Show completed contents" switch when you click on the end queue which shows the options "Export," "Cancel request," and "Requeue." Contents can be requeued or cancelled until the order has been validated. Requeuing contents changes the order status from "Completed" to "In Progress."

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.4. Simplified Protocol Design → 8.4.3. Defining Block Types → 8.4.3.11. Result
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.8. Electronic Lab Notebook (Routine) → 8.8.6. Results
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.8. Electronic Lab Notebook (Routine) → 8.8.6. Results → 8.8.6.3. Status of Results
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.15. Workflow Module → 8.15.2. Linking and Unlinking Content to/from Workflows

3.4.5 Analytical: Check Results with an External Database

The Products and Specifications module allows for the creation and management of as many specification rules as needed to validate result data. The software has been updated to manage large amounts of specifications while maintaining fast behavior, but in an edge case where there are too many to maintain (as an example, 2 million specifications with constantly changing limits) a lab may prefer to use an external source (for example, the European Union's pesticides database) to check against their rules.

For that rare case, a new "Evaluated by OpenLab Sync Extensions" evaluation mode is available in product specification configuration. Specification rules with this evaluation can be programmed with an Extensions script that calls out to an external system so the Extensions flow can evaluate the rule. The limits come from the external database to check if the results in OpenLab Sync are within range or not. The feedback is provided in OpenLab Sync, and non-passing values are flagged. The provided message can be scripted by the programmer, and although expected values can't be shown, extra values may be stored in a custom field on the result instead (like min/max, version of the EU pesticides database, etc.).

To work, OpenLab Sync Extensions and your plugins need to be installed and properly configured. There must be an analytical workflow with a result block to fill in the results and an action block with the Extensions flow in it pointing to the result block.

INFO

Reference:

✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.18. Products and Specifications → 8.18.2. Specifications Tab

3.4.6 Analytical: Schedule with Analyte Groups

TIP

- This change only applies to requestables from the Requestables module, so it works with analytical default orders, not QC orders.
- This change benefits labs that use a lot of analytes by requesting them at the group level instead of one by one.

An option was added in the Requestables module to "Schedule this requestable as a complete analyte group." When enabled, the requestable schedules the samples to a workflow based on the analyte group that was requested instead of by individual analyte. The possible analytical methods are all the ones capable of executing all the analytes in the requested group.

On the technical side, this change reduces the number of internal requests to improve how fast requests are scheduled. This is also visible when users schedule analytical orders. There will be a smaller number of rows in the pop-up: one for each analyte group instead of one for each separate analyte.

INFO

Reference:

✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.9. Requestables and Requestable Groups → 8.9.1. Requestables

✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.19. Products and Specifications → 8.19.1. Configuration of Analytical Workflows → 8.19.1.2. Configure Requestables

3.4.7 Analytical: Create Aggregated Result Rules on an Analyte Group

TIP

- Requires a test that measures the analyte group in a workflow, and an analytical method that has the test in its path.
- Available for scripted specification rules and for match specification rules using "Sum" aggregation.

It was already possible to have rules on individual analytes or on the sum of a couple of analyte results. You can now create result rules and product specification rules that evaluate on an analyte group instead of the individual analytes. To set up a specification rule:

- Choose "Script Specification" as the mode of evaluation and select a test. In the test configuration row, choose an analyte group and check the "Evaluate this on a result on the analyte group" box.

  Or

- Choose "Match Specification" as the mode of evaluation and select a test. Choose an analyte group, operator "Between (inclusive)," and aggregation method "Sum," then enable the "Evaluate this on a result on the analyte group" option.

This feature is useful in cases where you want to measure the sum of every analyte in the group and provide feedback about the total. For example, instead of writing a rule to sum each chemical Glyphosate, Diuron, Atrazine, etc., you can just write a rule to measure the sum of the group Pesticides. At runtime, this provides a result row for Pesticides in addition to the other created results and checks its value with the rule.

The analyte group result value takes the sum of any results in the result block belonging to analytes in its group. If the protocol result block has "Only create results for the requested tests" enabled, then the specification creates the analyte group aggregated result for all the individual analyte results that are automatically created. If extra results are added, they are included in the sum, and the rule shows if the specification is exceeded (such as if the total sum of all pesticides is out of bounds). Also, if there are multiple results for the same analyte, the analyte group aggregated result takes the sum of the analytes' means.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.2. Analytes and Analyte Groups → 8.2.4. Using Analytes in a Specification or Rule
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.2. Analytes and Analyte Groups → 8.2.5. Using Analytes in a Workflow
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.8. Electronic Lab Notebook (Routine) → 8.8.3. Simplified ELN → 8.8.3.1. Blocks
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.18. Products and Specifications → 8.18.2. Specifications Tab
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.19. Analytical Workflows → 8.19.1. Configuration of Analytical Workflows → 8.19.1.3. Configure Analytes
- ✓ OpenLab Sync Administration Manual → 10. Automations → 10.1. Rules → 10.1.2. Result

3.4.8 Analytical: Result Calculations Based on Other Results

Result calculations can be defined on a test to automatically calculate a target result field from other results on the same content. This is useful when a lab needs calculated values such as averages, sums, ratios, relative differences, or other derived results based on other results.

The Tests module UI now has tabs to organize the configuration and separate each calculation on a test. The "Detail" tab is the same, but the "Datatype" tab now contains the datatype setup. "+ Add Result Calculation" is used to add a new tab for each calculation that the administrator creates. On the Datatype tab, the result field must be set to Read Only so that the result value can be used in the calculation; otherwise, it will not be available. Each calculation tab has two parts: the expression area and the calculation parts area.

A result calculation is configured with a name, a target table field, and a Groovy expression. The calculation also contains one or more input parts. Each part has a name that is used as a binding variable in the expression, optional criteria to filter the candidate input results (for example, analyte equals Mg and test equals a specific test), and a selection method that determines which results are used. You can choose an aggregator and selection field to instruct the calculation; for example, to take the newest result based on the Created On date/time. Selection methods are:

- Use the most recent matching result on the same sample,
- Results with the same analyte, test, and protocol run block on different content,
- All results linked to the same order content (useful for when you're deriving and want a calculation based on the original content), or
- Custom.

Choosing Custom lets you select an existing custom selection method from a dropdown. Custom selection methods can be defined once and reused by calculation parts, but only by service users (OpenLab Sync Delivery Engineers). These should rarely be used.

If a calculation part has "Bind complete result" enabled, the full result record is exposed to the calculation instead of just the result. This allows the expression to use a custom field from the selected input result as part of the calculation.

When the result block is run, rows are added for the results and an additional row for the test calculation (ex: Hardness). When a result is added, OpenLab Sync determines whether existing results on the same sample depend on it by recognizing their calculation parts. The relevant formulas are evaluated, and the target field is updated on the affected aggregated result. The same recalculation occurs whenever an input result is updated. If new results are added, the calculation remains based on the original input results.

Users can manually review and refresh calculation inputs. From a result, the right-click option "View input results" opens a popup with tabs showing the input results used for each calculation so they can always tell which results were the basis of the calculation. The popup includes "Scan for new input results for all calculations," which rescans the available input results and uses the newest versions of the result calculations.

The Groovy expression is built with named bindings for each calculation part, the current aggregated result as "result," and the calculation math. A "mathHelper" object with statistical functions was added to the Insert menu to help the builder such as average, sum, minimum, maximum, median, count, standard deviation, relative standard deviation or coefficient of variation, relative percent difference, and square root. These functions support number fields and quantity values.

Several safeguards are included to keep the calculated values controlled. Only one active result calculation is allowed for the same target field on the same test. Rejected and canceled input results are excluded from formula evaluation. If any calculation part has no matching inputs, the expression is not executed, and the calculated result is blank. Formulas are also not re-evaluated for aggregated results that are validated or belong to a protocol run page with status "Done," so completed work is not changed. If result calculations form a circular dependency, recalculation stops after the few loops it takes OpenLab Sync to recognize it, and an error is provided.

Result calculations are versioned to provide stability. When a calculation is published, a new version is created, the input parts are copied to the new version, and only the latest version is executed. When a test is exported or shared via the Share module, its result calculations, calculation parts, and any referenced custom selection methods are included in the package.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.1. Tests and Test Groups → 8.1.1. Tests
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.1. Tests and Test Groups → 8.1.2. Result Calculations
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.8. Electronic Lab Notebook (Routine) → 8.8.6. Results

3.4.9 Analytical: New “Result Block” Rule Type

TIP

- Uses the Rules module.
- It is best practice to use the simplest rule to suit your needs because the more a rule looks at, the more costly it is in terms of performance. Conditional value expression rules are for general use and scripted result rules provide access to fewer other results compared to result block rules, but they still cover many use cases.

Result rules allowed you to check result values on one result or create a more complex Groovy script, but it was not possible to check multiple results measured at the same time in the same result block. The new “Result Block” rule type allows a Groovy script to be written with access to all the results in the same result block. You can also look across all the analytes in the block if the script looks at the analyte group.

The new rule type is useful for cases where you want to compare results against other results. For example, to look at every two results, calculate their difference, and add an error if the difference is too big. Another case may be to compare blanks against other blanks or compare analytical samples against quality samples to check if they were switched on accident. If a blank (which has a value you already know) has a higher value than an analytical sample, it could indicate that the blank was tested as the sample.

As another example, you can show the relative difference using a quantity custom field on the “Result” table that is filled by the rule script with `setValue`. That way, there’s a relative difference column that can be added to the result block’s view that displays the calculated value alongside the results.

A Value expression sequence number can be added to result block rules to determine the order in which the expression executes compared to other result block rules. This is useful in cases where there are multiple result block rules on the same result block. It also allows a result block rule to look at the results of an earlier result block rule (one that has a lower sequence number than itself).

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 10. Automations → 10.1. Rules → 10.1.3. Result Block

3.4.10 Analytical: Result Block Rules can Access Specification Outcomes

TIP

Uses the Rules module and may use rules from Products and Specifications.

Just as result block rules can check result values on all results in one result block, they also see the evaluations of regular result rules and product specification rules that have been run on single results including specification rule codes that have been applied. You can use them to compare results and find outlying values among rule evaluations.

For example, if you measure a blank and find that it's higher than analytical samples (code HIGH was applied), it indicates that the samples may have been switched on accident. The result block rule can see the code that was applied on blanks in the result block and produce errors on analytical sample results in the same block that have values lower than the blank.

INFO

Reference:

✓ OpenLab Sync Administration Manual → 10. Automations → 10.1. Rules → 10.1.3. Result Block

3.4.11 Analytical: Import Specification Rules with a Specification Type *

TIP

- Only for Product Specification Rules that have their scripting in the selected Specification Type.
- Only for users who have Import from Excel role functionality access rights.

Specification rules can be created as match specifications (like using a wizard) or as script specifications using Groovy scripts. It was already possible to import match specification rules because they don't have Groovy scripts on them. For script specification rules, it is best practice to craft the scripts once in the Specification Types module and then use the specification type across multiple product specification rules. This makes it easier to manage and prevent errors across a lot of different scripted rules.

Given the best practice, rules that have their scripting written in the selected specification type (so that the Groovy box on the rule itself is empty) can be imported through Excel. To enforce this, rules that have no specification type selected cannot be imported because it is too difficult to ensure data quality and prevent import errors.

The Specification Types column must be included in the Excel file for the import to succeed. Therefore, if you are exporting and then altering a file to reimport, use the advanced export options to include the column.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.17. Specification Types
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.18. Products and Specifications 8.18.3. Importing and Exporting Specification Rules

3.5 Upgraded Rich Text Editor

OpenLab Sync now uses CKEditor version 5 to take advantage of its modern UI and improved features for WYSIWYG editing. The rich text editor is accessed anywhere that long text can be edited in OpenLab Sync. For example, the description field on most entities, notes, note and form blocks in simplified protocols, rich text widgets for dashboards, and rich text custom fields.

Most of the features from CKEditor version 4 were retained and several were gained. There are some differences between the versions that impact OpenLab Sync:

- Rich text blocks and fields used to save automatically, but saving will only be automatic in some areas now. A warning is provided whenever the user has made unsaved edits. Notes and note blocks in protocols for ELN and Workflows automatically save every few seconds and when the user clicks outside the field. Dashboard text widgets autosave the same way. Other areas, like description fields and text SOPs, require the user to click Save.
- A full screen mode is available that shows the whole text area to make edition easier. Some of the actions are moved into a categorized top toolbar in full screen mode like in any other text editing program.
- The rich text editor retains most of the formatting if users paste formatted text from Google Docs, Microsoft Word, etc. Images and other documents can be dragged and dropped directly into the editor to show up as attachments on the parent record, and if possible, the file preview can be shown in the rich text field.
- Versioned entities can't be edited, but users can view versioned rich text fields (like a versioned SOP) in full screen mode. Search (without Replace) and Print are also available actions.
- Design features are available in the new version like line graphics and other formatting options.
- Checklists are available to create lists with checkboxes. Users can check the boxes as tasks are completed if the rich text block/field is not read only.
- Table creation and design offers greater flexibility and is easier to use.
- Bookmarks can be created so that you can link to them within the same rich text context.
- Clicking on an inline element like a line of text, an image, a table, etc. brings up a contextual toolbar icon. You can edit the inline element with the contextual toolbar or reorder it by dragging and dropping it by the toolbar icon.

Migration Note: Rich text created in CKEditor 4 in a prior version of SLIMS should be readable by CKEditor 5 in 7.3 and after.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 3. OpenLab Sync GUI and Navigation → 3.6. Rich Text Editor → 8.16.1. Configuring OpenLab Sync for Bench View
- ✓ OpenLab Sync Administration Manual → 3. OpenLab Sync GUI and Navigation → 3.12. Dashboards → 3.12.4. Rich Text Widgets
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.8. Electronic Lab Notebook (Routine) → 8.8.3. Simplified ELN → 8.8.3.1. Blocks
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.8. Electronic Lab Notebook (Routine) → 8.8.8. Notes

3.6 Track Reagents with Simplified Protocols

Simplified Protocol Link Content blocks were made more aware of each other so reagents can be related to the contents they are used with. Simplified Protocols do not have a designated Reagent step like Default Protocols and they are more freeform. Therefore, a Link Content block with usage “Extra” is considered a reagent block for a Link Content block with usage “Input” or “Output” if they are in the same protocol run. The blocks can be on different pages as long as they are in the same run. Then, the Reagents tab in the Content module will show the contents and reagents used together.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 5. Content Management → 5.2. Contents → 5.2.1. Content Module → 5.2.1.1. Content Module Tabs
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.4. Simplified Protocol Design → 8.4.3. Defining Block Types → 8.4.3.3. Link Content
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.8. Electronic Lab Notebook (Routine) → 8.8.3. Simplified ELN → 8.8.3.1. Blocks

3.7 Edit Results is Controlled by a Permission

TIP

The new role functionality access right “Can edit results outside of the context of their protocol run” is enabled by default (standard and premium roles).

The existing role functionality access right “Can create results” allows users to create results in the Content module. Users could also edit results in the Content module’s Results tab even if they were created in a protocol run.

The new permission makes the results read-only outside of the protocol run’s context. Result actions like reject, repeat, and validate will be accessible, but the values will not be editable from the Content and Orders modules for roles with the permission disabled. This change lets you choose the level of protection for result values.

Migration Note: “Can edit results outside of the context of their protocol run” is set to true to preserve the original behavior when upgrading to 7.3.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 5. Content Management → 5.2. Contents → 5.2.20. Create Results
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.11. Orders

3.8 Ingredient Removal is Controlled by a Permission

TIP

The new role functionality access right “Can remove ingredients from a mix” is enabled by default (for standard and premium roles).

Users can remove ingredients from a mix with a button in the Mix tab of the Content module and in the Mix step of default protocols. Administrators can choose the roles that can remove ingredients from mixes by disabling the new role functionality access right.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 5. Content Management → 5.2. Contents
- ✓ OpenLab Sync Administration Manual → 5. Content Management → 5.2. Contents → 5.2.15. Mix Contents

3.9 Edit Content Type after Content is Created *

In prior versions, a content’s type could be changed after the content was created, though this was not commonly done. This ability is now explicit in 7.3 as an option that can be enabled or disabled at the category or content type level.

When “Content of this type can be changed to another type after creation” is enabled, the type can be changed after contents are created in case the wrong content type was selected. When the option is disabled, the content type is read only after contents are created. This option is disabled by default because it is recommended not to use it outside of correcting mistakes.

“Content of this type can be changed…” is not available on container types or kit components, and content types cannot be changed using macros (such as in a macro edit step). Use of container types is also discouraged because their functionality is superseded now that all locations and contents are defined by content types. Contents like vials, plates, and other containers can be created as contents that hold other contents without needing container types.

Migration Note: Content types migrated from 7.2 will have “Content of this type can be changed to another type after creation” enabled to retain the previous behavior.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 5. Content Management → 5.1. Content Types → 5.1.2. Category/Content Type Field Explanations
- ✓ OpenLab Sync Administration Manual → 5. Content Management → 5.3. Container Types

3.10 Import Multiple Content Types *

TIP

Available to users with role functionality access right “Can import records in OpenLab Sync via Excel.”

Users can import multiple contents with different content types in one file. To do so, they can generate an Excel Template File in the Content module. For the content type dropdown, they can choose “Undefined” so they can fill the column in the resulting file. Users can specify advanced options and choose a display field for the content type. The list of fields is not restricted by content type since an undefined type has no restrictions. When the user enters content types for each row of content in the template file and imports it, the contents with their respective content types are created in the grid.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 3. OpenLab Sync GUI and Navigation → 3.5. Tools → 3.5.1. Data Management using Excel → 3.5.1.4. Create Excel Template

3.11 Import a Product via Excel *

Products can be imported via Excel in the Products and Specifications Module. They cannot be imported at different levels at the same time because of their tree structure, but multiple products or multiple subproducts of the same product can be imported at once.

A Template File can be generated that has the needed columns including the parent product column. This identifies which product the imported subproducts will go under in the tree.

This update doesn’t apply to products in the Study Design or Stability Study modules.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.18. Products and Specifications

3.12 Requestables can be Restricted by Content Type/Category *

Administrators can restrict requestables to certain content types and categories so that end users will see only the applicable requestables for the samples they selected. The option “Only for selected content types” is provided to multiselect the content types and categories relevant to a requestable. This restriction can be configured on requestables for standard workflows in Workflow Management and on requestables in the Requestables module for analytical workflows.

Users select samples in the Content module or link them to an order in the Orders module or the Requester Portal. Then, when they add requests with the To Workflow action, add requests to the order content, or choose requests in the Requester Portal, the available requests are filtered so they only see the applicable ones for their samples.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.9. Requestables and Requestable Groups → 8.9.1. Requestables
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.12. Requester Portal Configuration and Preparation → 8.12.1. Setup Considerations
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.14. Workflow Management → 8.14.2. Requestables Tab

3.13 Dashboard can Display Report Previews

TIP

Dashboards can only use criteria R report templates that have a single input file.

Administrators can make custom graphs for the Dashboard using criteria reports. They must set up a report template using the R engine and selection type “criteria.” The report template will show up when the Administrator edits a Dashboard to add widgets.

End users of the Dashboard can see a draft of the report template as a preview. The report is rerun when they refresh the widget. Only some file types can be shown as a preview: png, jpg, html, pdf and txt.

INFO

Reference:

✓ OpenLab Sync Administration Manual → 3. OpenLab Sync GUI and Navigation → 3.12. Dashboard → 3.12.5. Report Widgets

3.14 Download Graph in Dashboards *

It was not possible to download graphs from the Dashboard in version 6.9. With this release, you can now click the button at the top of graphs (such as pie and bar charts) from the Dashboard to download them as PNG images.

INFO

Reference:

✓ No change

3.15 Relink Content with a Copied Order *

Previously, when copying an order, a lab manager could choose to copy or derive its contents. It isn't always desirable to create new contents, so the option "Relink" was added to relink the same contents and their requests to the order's copy. The requests are copied from the original and have the status "Unfinished," and the workflow chain of custody, results, and local results for the order are empty. The order is "Pending" and ready to be scheduled with the relinked contents to the requested workflow.

INFO

Reference:

✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.11. Orders → 8.11.1. Order Life Cycle

3.16 Cancel Loading of Slow Grids *

Grids with large amounts of records may load more slowly than others. For example, the run grid of a busy protocol or the content grid with no filters applied. Users navigating between modules may not always need to see all the data loaded in a grid. In this case, they can click the cancel icon (red X) on the loading message to stop loading its records. The grid will be empty and the loading message changes to "Loading Canceled." Refreshing the grid will reload the records.

INFO

Reference:

✓ OpenLab Sync Administration Manual → 3. OpenLab Sync GUI and Navigation → 3.2. Grids → 3.2.1. Composition

3.17 Remove Content from a Location *

Contents and sublocations could be moved to a new location in the Content module, Locations module, and using various macro steps. However, you can now remove content from a location by moving it and leaving the destination blank. The content will still be tracked but will not have a position or be stored in any location.

You can see this change in:

- Default Move action in Content module and content grids
- Default Move action in Locations module and location trees
- Location definition in macro steps

INFO

Reference:

✓ OpenLab Sync Administration Manual → 5. Content Management → 5.2. Contents → 5.2.3. Create Location Contents → 5.2.3.1. Location Actions

✓ OpenLab Sync Administration Manual → 5. Content Management → 5.2. Contents → 5.2.11. Move Contents

✓ OpenLab Sync Administration Manual → 10. Automations → 10.9. Macros → 10.9.4. Action Configuration Options

3.18 Nice Multiple Choice Pop-up for Extensions Flows

OpenLab Sync Extensions flows with table parameters that have MultipleChoice columns now provide a pop-up that makes it easier to select records when there are many choices. The pop-up has the “Records” box on the left and the “Selected Records” on the right and moving records into the right-side box makes the records selected for the flow when you close the pop-up.

INFO

Reference:

✓ OpenLab Sync Development Manual → 3. Extensions Development → 3.5. Definition of slimsgate.xml → 3.5.3. Input Parameters

3.19 Client IP Address Lab Setting

The lab setting “Record IP Address in audit trails” may be enabled to capture the IP address of users performing actions that end up in the audit trail. The lab setting is optional and disabled by default. For every user, their IP address will be saved and visible in the Users module (read only) and in the history records. User IP addresses can change sometimes (f.e. when bringing a laptop home), so if it changes, the IP address is updated in the Users module, history, and records acted upon while the new IP address is used. None of this is visible when the lab setting is disabled.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 3. OpenLab Sync GUI and Navigation → 3.9. Lab Settings → 3.9.14. History Formatting

3.20 Scheduled Jobs are Run by OpenLab Sync Extensions User *

Scheduled jobs used to be run by the user account that created them. However, the jobs were then reliant on the user’s access rights which could prevent jobs from accessing some data. Starting with version 7.3, scheduled jobs will be run by the Extensions user instead, which provides access to all data the jobs need.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 10. Automations → 10.11. Notification Setup → 10.11.2. Scheduled Jobs Module

4 Enhancements in Version 7.4 Release

The following updates were included in the version 7.4 release. **Default updates are designated with an asterisk (*) next to the title.**

4.1 SLIMS is Renamed to OpenLab Sync

Version 7.4 introduces the name OpenLab Sync alongside a group of features that broaden OpenLab Sync's scope to customers looking for an LES. The software outgrew the name SLIMS, offering functionalities beyond those of a LIMS, yet the name did not indicate that. The rename to OpenLab Sync emphasizes features SLIMS already had that work as a lab execution system to integrate the lab's instruments into one workflow and track data end to end through the lab's digital ecosystem.

Existing customers will not lose the UI they are used to, nor the features that serve their ecosystems.

What changes:

- The product name and logo along with dependent technologies, modules, and add-ons.
    - SLIMS GATE is renamed OpenLab Sync Extensions
    - Bench UI is renamed Bench View for OpenLab Sync
    - Requester Portal is renamed Requester Portal for OpenLab Sync
    - SLIMS Store and SLIMS Share are renamed Store and Share
    - SLIMS Python/Rest/etc. APIs are renamed OpenLab Sync Python/Rest/etc. API
    - SLIMS-R is renamed OpenLab Sync R Integration
- URLs that serve the product (in general, `/slims)` are now `/openlab-sync`. The previous URLs will still redirect to new ones for customers with dependencies on old URLs, but customers are encouraged to use the new URLs when possible.
- Manuals, validation documentation, training materials, and resources are labeled for OpenLab Sync from version 7.4 onward
- Features in 7.4 targeted for LES Pharma laboratories
- Features and quality of life improvements that benefit customers more broadly
- Store packages that support LES Pharma

What we keep:

- Existing features so that current customers lose nothing from SLIMS
- Existing generic Store packages and those for other lab industries

INFO

Manuals that are Impacted:

- ✓ OpenLab Sync Administration Manual
- ✓ OpenLab Sync Installation Manual
- ✓ OpenLab Sync Development Manual
- ✓ Documentation on Agilent Community and Agilent.com

4.2 Reusable Simplified Protocols as Templates

TIP

- ✓ Simplified Protocols only (for Workflows and ELN).
- ✓ Reusable protocols are configured in the Protocols module only and can then be reused in other protocols.
- ✓ Flags are only available in Workflows, not ELN, so reusable protocols that use flags cannot be designed.

Reusable Simplified Protocols let administrators define a simplified protocol as a reusable template and insert it into other protocols in ELN or Workflow Management or run it in ELN just like a normal protocol. This reduces configuration work for repeated processes because pages and blocks that have been checked for correctness can be used across multiple similar procedures. The template protocol gets pulled in with its blocks and structure, including placeholders. Reused content remains connected to its source protocol so administrators can identify where the configuration came from and whether a newer template version is available.

Three situations were accounted for that benefit the protocol designer to make configuration easier:

- Static reusable protocols to make standardized protocol templates that the designer knows are correct (non-ad hoc),
- Planned reusable protocols using a “Reuse Protocol” block to make a list of possible reusable protocols the technician can choose from (planned ad hoc),
- And unplanned reusable protocols that can insert a reusable one page protocol when a run is being done (unplanned ad hoc) even though a “Reuse Protocol” block wasn’t preconfigured.

Setup

To set up a reusable protocol, create or open a simplified protocol in the Protocols module and enable "This protocol can be reused as a template for other protocols." The protocol template will always be created from its latest version, and if the template is ever updated, indicators are provided on protocols that reuse the template to say they are out of date. Once enabled, you can optionally define placeholders so the template can adapt to a destination protocol that might already have a page or link content block in it.

- For a placeholder page, enable "When this protocol is reused allow to use an existing page" so protocol designers can map the template page to a page that already exists in the destination protocol.
- For a placeholder Link Content block, enable "When this protocol is reused allow to use an existing link content block" so the protocol designer can map the template link content block to an existing one on the page. Placeholder blocks may only be used on Placeholder pages, not by themselves. This is practical for templates that you expect to be reused by workflow protocols, because each of their pages start with a default Pop block.

4.2.1 Static Reusable Protocols

When editing a simplified protocol in Protocols or Workflow Management, the “Reuse a protocol” button is available on the Pages and Blocks form. When clicked, a selection window opens that lists the reusable templates. The list is based on the latest active reusable protocol versions and can be browsed or searched to find the desired template. After a template is selected and confirmed, OpenLab Sync copies its pages and blocks into the current protocol. If the selected template does not contain placeholders, the reuse is completed as-is.

If the reusable protocol contains placeholder pages or placeholder Link Content blocks, OpenLab Sync prompts the designer to choose how those placeholders should be handled during reuse. For each placeholder page, the designer can either create a new page from the template or select an existing page in the destination protocol. If a selected page contains placeholder Link Content blocks, each of those blocks can also be created from the template or mapped to an existing Link Content block in that page. Configurations like macros are only brought if the block is created from the template.

Reusable content remains traceable after insertion. Pages and blocks copied from a reusable protocol display the source protocol and version in a "From protocol" field and column, and the protocol form shows a "Reused protocol(s)" list so administrators can review all template relationships in one place. When there is a newer version of a reused protocol, the “From protocol” fields turn yellow to show they are out of date, and if relevant, a banner appears on the Page Info or Block Info (details forms). On the Protocol tab, in the Reused Protocols table, designers can click the “Update” button to bring in the changes. The update process uses the same mapping workflow as initial reuse so designers can preserve existing placeholder choices or change them if the newer template version should create new pages or blocks instead.

Protocol templates can be reused inside other templates (nested reuse). The reusing protocol traces back to the first direct template it reused to retain linkage. Also, templates reused in a protocol can be updated in the destination protocol. However, if the template is ever changed and the updates are pulled to the destination protocol, they will overwrite the manual changes that were made. Blocks that are removed from a template are not removed from the destination, so nothing is lost, but the link to the template is removed. Otherwise, the template is the version of authority over the pages and blocks in its design.

Protocols that reuse templates can still be copied, versioned, reverted, and shared. When a protocol that already uses reusable protocols is copied, the new protocol retains the same reuse relationships. When such a protocol is versioned or later reverted to an earlier version, the system preserves the reused protocol references that were valid for that version so designers can see the correct template versions for each historical state. Reusable protocols and protocols that depend on them can also be transferred with Share,

allowing reusable configurations to be moved between environments while maintaining their template relationships.*

*For more detail about how version number recalculation from the source environment to the target environment, works, see the Administration Manual.

4.2.2 Planned Ad Hoc User Reuse

Ad hoc protocol reuse is possible during protocol execution. Designers can configure reusable one-page protocols for insertion at runtime so users can add them into an in-progress run. This allows a whole complex one-page protocol to be crafted that can be inserted on the fly at runtime.

Designers can add the new block type “Reuse Protocol” to a one-page protocol for this purpose. This block type has a “Sub protocol” dropdown with a list of possible templates so the designer can restrict the list that should be available to the user at runtime. The “Reuse Protocol” appears as a button like the Flows or Macros buttons from Action blocks and provides the user with a filtered selection of eligible templates. Once the user chooses, the action inserts the blocks from the template, taking the Link Content blocks on the page that make sense and linking them to the inserted blocks where needed (such as Result blocks that need input samples).

4.2.3 Unplanned Ad Hoc User Reuse

It is possible for users to add ad hoc blocks with the blue Add Block bar that appears when they hover in a Simplified Protocol. The “Reuse a Protocol” block is available as an option, in which case the user can choose from any of the one-page reusable protocol template in a dropdown. The protocol designer can restrict block types in the protocol design if reusing templates should be prohibited or planned instead.

If there is no Link Content block on the page that the reused template can use (and it has blocks that rely on Link Content blocks, such as Result blocks), it creates one.

Protective Design Limitations

Some limitations exist to keep impossible builds from being created:

- A protocol template cannot be reused within itself.
- Cycles are not possible. For example, Protocol A cannot be reused inside Protocol B while B is reused within A, and bigger cycles with similar loops cannot be constructed.
- Protocols may accidentally have two Input Link Content blocks on one page due to reuse, which is not allowed. For planned reuse, the issue is detected and the protocol is not versionable until it is fixed. For reuses made during protocol execution, an error is provided if inserting the reusable protocol would cause you to have two Input Link Content blocks on the same page.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.3. Protocols
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.4. Simplified Protocol Design → 5.2.4. Manage Instruments → 5.2.4.2. Instruments for Workflows
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.4. Simplified Protocol Design → 8.4.1. Simplified Protocol Pages
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.4. Simplified Protocol Design → 8.4.3. Defining Block Types → 8.4.3.14. Reuse Protocol
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.4. Simplified Protocol Design → 8.4.3. Defining Block Types → 8.4.3.3. Link Content
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.4. Simplified Protocol Design → 8.4.4. Reusable Simplified Protocols
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.4. Simplified Protocol Design → 8.4.5. Simplified Protocol Validation
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.8. Electronic Lab Notebook (Routine) → 8.8.3. Simplified ELN → 8.8.3.1. Blocks
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.14. Workflow Management → 8.14.1. Workflow Graph Tab

4.3 Expansions to Bench View for OpenLab Sync

TIP

- ✓ Simplified Protocols only (for Workflows and ELN).
- ✓ Requires Bench View role functionality access rights

Bench UI was renamed to Bench View for OpenLab Sync in this release. Bench View is a tablet interface for users working with instruments and samples at the bench. These changes extend simplified protocols in Bench View by supporting more block types, expanding customization options, and aligning behavior more closely with core OpenLab Sync.

4.3.1 Recipe Blocks are Supported

Recipe blocks are now supported in Bench View, bringing guided recipe execution to the bench. A horizontal carousel shows all recipe executions for the block. Users can scroll with the arrow icons or by tap-holding and sliding. The 3-dot menu can be used to remove executions unless the run or page is read only. If only one recipe is available to execute, it is selected and displayed automatically. If multiple recipes are available, the user can tap one to display it.

The user can select an execution to use the mix as input for an Action block. An ingredient or mix may also be scanned to find a recipe it was used in. If it relates to multiple mixes,

Bench View finds the last recipe that the ingredient or mix was used with. Feedback is provided if the scanned ingredient or mix is not found.

Below the carousel is the list of the selected recipe execution instructions. When the user creates a recipe execution, an ingredient may be scanned to link it, and a confirmation message is provided. An error is provided if something is scanned or selected that doesn't exist or is of an ineligible content type. Ingredients can otherwise be linked from a dropdown list and can be unlinked if needed. When amounts are entered, green checkmarks or exclamation points indicate whether they are in or out of range, and the target range is provided as a column in the instructions.

The target amount can be changed to make a larger mix. If an Extensions flow is used to get the amount, there will be an icon in the amount column next to each amount to launch the flow, which automates actions like talking to a balance, getting the measurement, and using context to link the values to the right recipe instruction.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.16. Bench View for OpenLab Sync: Interface for Tablets → 8.16.1. Configuring OpenLab Sync to use Bench View
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.16. Bench View for OpenLab Sync: Interface for Tablets → 8.16.2. How to Use Bench View for OpenLab Sync
- ✓ OpenLab Sync Administration Manual → 9. Audit Tools and Quality Control → 9.2. SOPs → 9.2.1. Recipe SOPs

4.3.2 SOP Blocks are Supported

SOP blocks are now supported in Bench View as an additional way to provide in-context guidance during execution. They can present an SOP attachment, note, or hyperlink while the user works. SOP blocks may be restricted at the protocol and workflow requestable levels so that only contextually relevant SOPs are presented.

If only one SOP is possible based on those restrictions, Bench View preselects and displays it for the user. If multiple SOPs are possible, then the user gets a choice of all possible SOPs to select from and it will be displayed (shown, opened, or downloaded).

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.16. Bench View for OpenLab Sync: Interface for Tablets → 8.16.1. Configuring OpenLab Sync to use Bench View
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.16. Bench View for OpenLab Sync: Interface for Tablets → 8.16.2. How to Use Bench View for OpenLab Sync

4.3.3 Link Location Blocks are Supported

Link Location blocks are now supported in Bench View so location-based work can be managed alongside samples and actions at the bench. They track lab locations that hold content, such as fridges, drawers, and well plates. Bench View respects block restrictions such as location content type filtering, whether uncalibrated or expired locations are

allowed, whether warnings are shown for expired or uncalibrated locations, location statuses that prevent linking to a protocol run, and interactions with macros and Extensions flows. Location blocks display the configured plate view when custom fields or color highlighting are in use.

Location blocks in Bench View have a scan box and selection button so users can link locations using either method. A vertical carousel on the left displays linked locations and users can tap one to display it as a well plate on the right. A message is shown instead if the location is positionless or too big to display. Users can tap and drag to select multiple positions and can even select them within multiple locations. A chip is shown on each location so users can tell how many positions they selected within each.

Users can unlink or manually move locations using the 3-dot menu on each location. Samples, however, need to be moved via a Link Content block or a macro instead.

Locations and selected positions within them can be used as input for macros and Extensions flows. How to configure these options is detailed in the Administration Manual.

When barcodes are scanned, Bench View processes them when the scan box for the Link Location block is focused:

- An error is provided if the barcode doesn't exist
- If a location is scanned that isn't already linked, it is linked and pulled up in focus
- If a location is scanned that is already linked, it is pulled up in focus
- If a sample is scanned that is already in a linked location, the location is pulled up with the sample selected
- If a sample in an unlinked location is scanned, the whole location is linked and pulled up with the sample selected
- An error is provided if a sample is scanned that has no location or that has a location that cannot be linked.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.4. Simplified Protocol Design → 8.4.3. Defining Block Types → 8.4.3.4. Link Location
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.16. Bench View for OpenLab Sync: Interface for Tablets → 8.16.1. Configuring OpenLab Sync to use Bench View
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.16. Bench View for OpenLab Sync: Interface for Tablets → 8.16.2. How to Use Bench View for OpenLab Sync → 8.16.2.3. Using Blocks in Bench View for OpenLab Sync

4.3.4 Create and Edit Usages in Bench View

Bench View also adds more direct support for usage tracking during protocol execution. Default usages could already be added in Bench View, but users had to open the side panel to view or edit them. A "Usage" column appears in the Link Content block's grid now but only when the block has usage tracking enabled or if a macro creates usages in the block. If usages already exist, or if usages are expected but still empty, the Usage column shows the sum of usages (one sum per row) for samples in the Link Content block. Users can tap

the sum to open the details panel and update the usage. As in OpenLab Sync, usages take the default unit and amount when they are defined by the protocol or by a macro with an add usage step. Usages remain synchronized between OpenLab Sync and Bench View.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.16. Bench View for OpenLab Sync: Interface for Tablets → 8.16.1. Configuring OpenLab Sync to use Bench View
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.16. Bench View for OpenLab Sync: Interface for Tablets → 8.16.2. How to Use Bench View for OpenLab Sync

4.3.5 Create Manual Results in Bench View

Result entry has also been extended in Bench View so users can create manual results in addition to automatic ones. Users can add a manual result by tapping the plus icon. If multiple tests are available for the sample, a pop-up prompts the user to choose the test for which results should be added. If no samples are selected, the pop-up asks which result to add, and the user can select or scan the relevant sample.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.16. Bench View for OpenLab Sync: Interface for Tablets → 8.16.1. Configuring OpenLab Sync to use Bench View
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.16. Bench View for OpenLab Sync: Interface for Tablets → 8.16.2. How to Use Bench View for OpenLab Sync

4.3.6 Add Ingredient to Mix (Without Recipe)

The macro step “Add Ingredient to Mix” also works in Bench View outside of recipe execution. This step may work alone but commonly has an “Add Usage” step before it. There need to be two Link Content blocks: one as input and one as output for an Action block that uses the configured macro. The macro steps should have “output to block” disabled, so the input of the macro doesn’t appear in the output block as well. At runtime, the user must select an ingredient in the input block and a mix in the output block, then tap the macro to associate the ingredient with the mix. A usage will be taken of the ingredient and reflected on the mix.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.4. Simplified Protocol Design → 8.4.3. Defining Block Types → 8.4.3.3. Link Content
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.16. Bench View for OpenLab Sync: Interface for Tablets → 8.16.1. Configuring OpenLab Sync to use Bench View
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.16. Bench View for OpenLab Sync: Interface for Tablets → 8.16.2. How to Use Bench View for OpenLab Sync
- ✓ OpenLab Sync Administration Manual → 10. Automations → 10.9. Macros → 10.9.2. Macro Step Types → 10.9.2.1. Steps Available for Table: Content

4.3.7 Support Software Date Formats and N/A Values *

Bench View now supports the date formats configured in the core software, making date entry more consistent across OpenLab Sync. Date and Date Time fields in Bench View follow the chosen lab setting format, including European, Japanese, and US date styles. This reduces the need for users to adjust to different date-entry behavior when moving between the main software and Bench View.

Bench View also supports fields that allow N/A. A small "N/A" button is available for fields that permit that value, and the user can tap it to enter N/A instead of the field's value. If N/A must be removed, the user can tap "Clear."

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 3. OpenLab Sync GUI and Navigation → 3.9. Lab Settings → 3.9.2. OpenLab Sync
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.16. Bench View for OpenLab Sync: Interface for Tablets → 8.16.1. Configuring OpenLab Sync to use Bench View
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.16. Bench View for OpenLab Sync: Interface for Tablets → 8.16.2. How to Use Bench View for OpenLab Sync

4.3.8 Customizable Grid Columns for Bench View

A configuration plugin can be used to customize the Home Page grids in Bench View. Administrators can define which columns are shown by default in the tables, so the Home Page emphasizes the information most relevant to the lab's work.

- To adjust the Bench View Home Page grids, use the configuration plugin that comes from the Store package for Bench View. It is separate from the Portal plugin, but an OpenLab Sync instance can have both and they will not interfere with each other.
    - To add columns to the Bench View Runs grid, include the parameter `runDefaultColumns:`

- To add columns to the Bench View Not Started (Samples) grid, include the parameter `contentDefaultColumns:`

List the field names to show under each respective parameter for the respective grids. If no custom columns are defined, Bench View uses the system defaults. Any columns added through the plugins are shown to users on the Home Page, so it is best practice to keep the default list focused and avoid overloading the tables with too many fields.

Users can still personalize the grids if they do not need every default column. The Home Page grids in Bench View provide a gear icon with the option "Configure columns." The user's visible-column choices are saved in the browser cache, so the same browser on the same machine will keep those preferences until the cache is cleared, at which point the configured defaults from the plugin are applied again.

Example of the Bench View plugin with custom columns added for both grids:

```
---
contentDefaultColumns:
- "cntn_cf_expiry"
runDefaultColumns:
- "xprn_status"
- "xprn_completedOn"
```

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.16. Bench View for OpenLab Sync: Interface for Tablets → 8.16.1. Configuring OpenLab Sync to use Bench View
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.16. Bench View for OpenLab Sync: Interface for Tablets → 8.16.2. How to Use Bench View for OpenLab Sync

4.3.9 Edit Content Details and Scan Linked Samples *

Bench View also provides more direct ways to work with linked samples within the context of a run. Sample data can be edited directly from the Content Detail panel. When the user opens the side panel for a linked sample, the panel shows the sample details in context so values can be reviewed and updated without navigating through the protocol page. This supports a more controlled workflow at the bench and helps reduce editing errors.

If a Link Content block contains many linked samples, users can scan samples in the block's scan box and Bench View will locate and focus on them instead of requiring the user to search manually. Multiple samples may be scanned in sequence, and Bench View will multiselect them. This is especially useful in high-throughput work where the user needs to find the correct linked records quickly and with less chance of error.

If the Link Content block uses protocol run step custom fields, those fields are also available from the content detail panel in the context of the run. For example, if the block includes fields used to map analytical sample types, the side panel shows the relevant fields for the selected sample and the user can edit their values there. This allows run-specific sample information to be maintained in Bench View alongside the linked sample record.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.16. Bench View for OpenLab Sync: Interface for Tablets → 8.16.2. How to Use Bench View for OpenLab Sync

4.3.10 Select Instruments in Form Blocks

Form block support has also been expanded in Bench View to include instrument selection during a protocol run. Users can choose an instrument from a selection pop-up or scan an instrument to link it.

The instrument-selection pop-up filters out instruments that are not eligible for the step. If warnings for ineligible instruments are enabled in the configuration, those warnings are also shown when appropriate.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.16. Bench View for OpenLab Sync: Interface for Tablets → 8.16.1. Configuring OpenLab Sync to use Bench View
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.16. Bench View for OpenLab Sync: Interface for Tablets → 8.16.2. How to Use Bench View for OpenLab Sync

4.3.11 Expanded OpenLab Sync Extensions Flow and Macro Support

Support for OpenLab Sync Extensions and macros in Bench View has also been broadened so guided workflows can accommodate more interaction patterns at the bench. Bench View now supports additional OpenLab Sync Extensions flow field types and behaviors. Password fields are supported, and their text is obfuscated as the user types. In addition to the already supported single-select dropdown, Bench View now supports `SingleChoiceGrid`, `Multiselect`, and `MultipleChoiceGrid` datatypes. These allow users to select one or more records through either dropdowns or tabular choice views, depending on how the flow is designed.

The `Feedback` datatype may also be used in multiple-step flows. This allows a later step to show hardcoded guidance or provide feedback that references information from an earlier step in the same flow.

Macros can now also execute at the start of a page in Bench View. If multiple macros are configured to run when the page opens, they execute in ascending order of their sequence numbers when sequence values have been defined.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.16. Bench View for OpenLab Sync: Interface for Tablets → 8.16.1. Configuring OpenLab Sync to use Bench View
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.16. Bench View for OpenLab Sync: Interface for Tablets → 8.16.2. How to Use Bench View for OpenLab Sync

4.4 Updates to Requester Portal for OpenLab Sync

TIP

- ✓ Requires Standard or Premium license and Add-on license.
- ✓ Requires Requester Portal role functionality access right.

Requester Portal is an add-on functionality that allows your lab's customers to request orders and analyses through a portal separate from core OpenLab Sync. Some changes were made to expand customization options and align behavior with OpenLab Sync.

4.4.1 Import Multiple Content Types in Portal with Excel *

TIP

- ✓ "Can use default Excel import" must be enabled on your order type.

The lab customer can create orders and add samples in Requester Portal, or they can import them from Excel. In core OpenLab Sync, you can import multiple content types in one sheet by choosing "Undefined" as the sample type, and this now works in Requester Portal as well. The user exports an Excel Template file that contains a column for Sample Type (equivalent to the content type), all the columns that have "Show in portal" enabled on them, and the relevant Request column for products and/or requests depending on your order type configuration.

The corresponding samples are imported with the chosen sample types (content types) and they are linked to the created order.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.12. Requester Portal Configuration and Preparation → 8.12.1. Setup Considerations
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.13. Requester Portal for OpenLab Sync Customer Use → 8.13.2. Create Order

4.4.2 Customizable Grid Columns for Requester Portal

A configuration plugin can be used to customize the Orders table in Requester Portal. Administrators can define which columns are shown by default, so the table emphasizes the information most relevant to the lab requester's work.

- To add columns to the Requester Portal Orders grid, use the configuration plugin that comes from the Store package "Requester Portal Basics." This is the same plugin that lets Administrators add company branding and FAQ tabs to the portal. Include the parameter `defaultOrderColumns:`

List the field names to show under the parameter. Custom fields for Requester Portal must have "Show in Portal" enabled or they will not appear in the portal. If no custom columns are defined, Requester Portal uses the system defaults. Any columns added through the plugin are shown on the Orders table, so it is best practice to keep the default list focused and avoid overloading the table with too many fields.

Users can still personalize the grids if they do not need every default column. The List View in Requester Portal allows the user to hide columns. The user's visible-column choices are saved in the browser cache, so the same browser on the same machine will keep those preferences until the cache is cleared, at which point the configured defaults from the plugin are applied again.

Example of the Requester Portal plugin with custom columns added:

```
---
defaultOrderColumns:
- "ordr_cf_priority"
- "ordr_cf_customerID"
```

INFO

Reference:

✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.12. Requester Portal Configuration and Preparation → 8.12.2. Technical UI Customization

4.4.3 Add Requests Table *

When users add samples in Requester Portal, they may have many requests to choose from, depending on the analyses available from the lab. Requests can now be selected from an organized table. Custom fields can be added as columns in the requests table and sample details which users can edit by clicking the pen icon on the row. Values in the fields only appear for relevant requestables and they become Read Only once the order is submitted. These custom fields must be defined on the request table in OpenLab Sync, use a datatype supported by Requester Portal, and have "Show in Portal" enabled. Any restrictions configured on Requestables are respected.

The "All" and "Selected" chips show the number of available and selected requests. Users can filter requests by name by typing in the filter box. As a reminder, Administrators can define request groupings by creating Requestable Groups that appear as Quick Selection chips above the table, allowing users to select an entire group at once. Users can also select individual requests with the checkboxes on the table.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.12. Requester Portal Configuration and Preparation → 8.12.1. Setup Considerations
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.13. Requester Portal for OpenLab Sync Customer Use → 8.13.2. Create Order

4.5 Flags May be Used in Simplified Protocols

Flags could already be used in Default Protocols, but they can now be used in Simplified Protocols also. Each flag is defined with a long name, short name, color, and consequence, such as forcing the user to cancel or requeue flagged samples. Flags are not relevant for blocks without samples linked in them, so they may only be added in Pop blocks and Link Content blocks. This change also makes it possible to convert default protocols with flags into simplified protocols.

The dropdowns for the flags' consequences and restricting flags to selected protocol steps now include the allowed simplified protocols and block types in your instance.

The order of blocks on pages in Simplified Protocols impacts when the flags are applied and how they are viewed. Flags applied by scripting in a content block can be triggered by result values in a Result block on the same or a different page. The scripting must go on the content block serving as input for the Result block, and then the samples with result values that trigger the scripting will be flagged in the content block. However, if the content block is on a page before the Result block, the flags scripted on the content block will not be applied.

Once set up, users can add and remove flags manually from Pop and Link Content blocks. The Flags and Previous Flags columns can be added to the block's view to show all the flag labels that were applied on the linked samples. Feedback is shown that prevents protocol progression depending on the consequences set by a flag.

Converting a Default Protocol with Flags

A default protocol with flags can now be converted into a Simplified Protocol. The flags are retained during conversion and are put on a Link Content block to serve as input for the block that used to have flags on it. For example, a default protocol with additional flags on the content selection step and a derivation step that has an expression to add a flag can be converted. It results in a simplified protocol with two pages. The first has a Pop block with the same additional flags selected on it. The second has a Link Content input block and a Link Content output block. The input block has the expression to add a flag.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 5. Content Management → 5.2. Contents → 5.2.4. Manage Instruments → 5.2.4.2. Instruments for Workflows
- ✓ OpenLab Sync Administration Manual → 6. Studies → 6.1. Studies → 6.1.2. Content Event Types → 6.1.2.2. Flag Content Event Types
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.4. Simplified Protocol Design → 8.4.3. Defining Block Types → 8.4.3.3. Link Content
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.5. Default Protocol Design → 8.5.2. Protocol Steps
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.5. Default Protocol Design → 8.5.5. Convert into a Simplified Protocol
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.8. Electronic Lab Notebook (Routine) → 8.8.3. Simplified ELN → 8.8.3.1 Blocks
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.15. Workflow Module → 8.15.3. Using Flags in a Workflow

4.6 Recipe-Related Improvements

TIP

Simplified Protocols only (for Workflows and ELN)

Several updates were made to improve Recipes in core OpenLab Sync and its components.

- The Seq No attribute was added to Recipes so you can choose the order in which recipes in the same block operate. The recipe executions will be shown in the defined order, lowest to highest number, and null Seq Nos last. If it doesn't matter what order they execute in, this can be left empty. However, it is useful when the lab has recipes that need to be executed first to produce mixes that are used in later recipes.
- Recipes can automatically use a default amount for recipe ingredients for which the lab doesn't care about an exact measurement. This is meant to save work for the technician who executes the recipe. Recipe ingredients have the option "Automatically take a usage equal to the target amount" which fills the default amount when enabled so the user doesn't need to enter a specific amount.
- A Recipe Measurement Extensions flow can be made to weigh ingredient amounts using a balance and import the value to OpenLab Sync or Bench View recipe instructions. It can be scripted with useful actions like talking to a balance, getting the weights, and using context to link weight amounts to the right recipe instruction. When a flow with usage `recipeMeasurement` is used for a recipe ingredient, it appears as a button next to the ingredient amount to save screen space.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.4. Simplified Protocol Design → 8.4.3. Defining Block Types → 8.4.3.3. Link Content
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.4. Simplified Protocol Design → 8.4.3. Defining Block Types → 8.4.3.10. Recipe
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.8. Electronic Lab Notebook (Routine) → 8.8.3. Simplified ELN → 8.8.3.1. Blocks
- ✓ OpenLab Sync Administration Manual → 9. Audit Tools and Quality Control → 9.2. SOPs → 9.2.1. Recipe SOPs

4.7 Aggregated History for Protocol Runs *

History is recorded for every table in OpenLab Sync, including protocols that consist of several steps, pages and each block in every step or page. However, history for protocol runs was stored in each of those individual tables. The history mechanism has been expanded so users can see a convenient aggregated history for protocol runs in these primary places: Experiment/Protocol Run, Experiment Run Step/Protocol Run Step, and Content.

Expanded History:

- At the **experiment/protocol run history level,** the pop-up contains audit trail for pages and blocks in the run, content and results linked into blocks, quantities recorded in blocks, instrument usage, recipe and recipe instruction executions, sample sheet rows, flags applied during the run, etc.
- At the **step/block level**, the history pop-up contains audit trail for all content linked into the block, any results and quantities that were captured, any instrument usages in the step/block, executions of the step/block, flags on the step/block, etc.
- At the **Content level**, the history pop-up shows any of its created results, any quantities taken, any recorded events, and any steps or blocks it was linked to.

For each of these aggregated histories, a “Table” column shows which table the history context originated from. The table column can be used to filter the history records.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 5. Content Management → 5.2. Contents → 5.2.1. Content Module → 5.2.1.1. Content Module Tabs
- ✓ OpenLab Sync Administration Manual → 5. Content Management → 5.2. Contents → 5.2.21. Create Usages
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.8. Electronic Lab Notebook (Routine) → 8.8.3. Simplified ELN
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.8. Electronic Lab Notebook (Routine) → 8.8.5. Protocol Run Steps
- ✓ OpenLab Sync Administration Manual → 9. Audit Tools and Quality Control → 9.1. History → 9.1.5. History User Interface

4.8 Expanded History on Test Datatype *

When permitted changes are made to a test's datatype, they are now visible in the test's history. Test datatype information is stored in behind-the-scenes fields that are not visible, but the historical links were improved to reveal pertinent information from those fields in the test's history. You can now also filter on the Test and TableField in history. When migrating from a previous version of OpenLab Sync, you can see the history of the datatype in 7.4.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.1. Tests and Test Groups → 8.1.1. Tests

4.9 Refreshed Content Type Icons*

A new suite of content type icons and other system icons have been added with a refreshed look. Migrating customers will get the new icons by default and may view them in the Content Types module.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 3. OpenLab Sync GUI and Navigation → 3.8. Icons Overview

4.10 Drag and Drop to Reorder Sample Sheet Rows *

Grid templates with the root `samplesheetrow` can be used in a grid template protocol block as an editable sample sheet for your equipment. Previously, users could reorder sample sheet rows with arrow buttons to move individual rows up and down, or by opening a pop-up to enter a sequence number to move rows to a specific position.

Now, users can also reorder sample sheet rows by dragging one or more rows to a different position in the grid. The reordered sequence is saved and retained when the protocol is reloaded. Read only grids, such as those in finished runs, cannot be reordered. A maximum of 100 rows can be dragged at one time. Reordering is disabled while the grid is filtered, whether by drag and drop or with the arrow buttons, because OpenLab Sync cannot determine the correct position when only a subset of rows is displayed.

INFO

Reference:

✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.7. Agilent MassHunter Integrations → 8.7.4. Running a Sequence in ELN or Workflows

4.11 Import Protocol Run Step Content Values via Excel

It was already possible to interact with samples linked in a protocol run with Excel, but the actions didn't have access to the protocol run step content table, so users couldn't interact with custom fields on that table. A new set of actions has been added to give users a choice between interacting with linked samples and their custom fields or interacting with custom fields on the protocol run step content table. If users select a Link Content block during a simplified protocol run, the buttons can be found in the page header Actions menu. There are two sets of the Export, Import, and Create Template File actions.

The “Content” set are the actions that previously existed and allow users to import linked samples to update their custom fields and export the list of linked samples. They are:

- Export content
- Create cntn template file
- Import content

The “Run Step Content” set are the new actions which allow users to update custom fields on the protocol run step content table. They have access to the protocol run step content fields, not the usual content fields, so users can't use them to link new samples or update content field values. They are useful for updating field values on content in the context of the protocol run:

- Export run step content
- Create xrsc template file
- Import run step content

There are a few restrictions to prevent errors. Samples that are exported from one block cannot be imported to another block. Also, if a user tries to import samples that aren’t linked to the block, an error is provided.

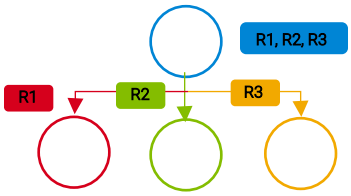
INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.8. Electronic Lab Notebook (Routine) → 8.8.3. Simplified ELN

4.12 Requests Respect Content Type Restrictions *

Content type restrictions in the pop step of workflows are respected, and requests will now split queue elements where it is appropriate. If you have a sample that is derived in the first protocol into three different types, and the workflow splits into three protocols with content type restrictions on them, the request is split to take that into account. The derived children of the original sample get the request for their matching content type and proceed to the correct protocol.



However, all three content types need to make it through the workflow and fulfill their requests. An error will be provided if one or more gets left behind.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.4. Simplified Protocol Design → 8.4.3. Defining Block Types → 8.4.3.3. Link Content
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.5. Default Protocol Design → 8.5.2. Protocol Steps → 8.5.2.2. Protocol Step Settings for Specific Step Types
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.16. Bench View for OpenLab Sync: Interface for Tablets → 8.16.1. Configuring OpenLab Sync to use Bench View

4.13 Control the Download of Attachments *

A field called “Track Download” was added to Attachment Types so you can see how many times an attachment is downloaded and use the field in rules. The field can be enabled or disabled across different attachment types so you can pick and choose which are tracked and restricted by rules.

An event is created in history that shows who downloaded attachments and how many times it was downloaded. A “Downloaded” column in the Attachments module can be added to the view to show the number at a glance.

Conditional value expression rules written on the “attachment” table may be scripted to prevent some user roles from downloading attachments. The rule can be restricted to the tracked attachment types. It needs to be executed “On update” and have timing “Validation.”

Tracking is disabled by default for all attachment types when migrating to 7.4 from an older version of OpenLab Sync.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 9. Audit Tools and Quality Control → 9.3. Attachments → 9.3.2. Open and Download Attachments
- ✓ OpenLab Sync Administration Manual → 9. Audit Tools and Quality Control → 9.3. Attachments → 9.3.5. Attachment Types

4.14 New Queue Aggregator Type: Automatic *

To make workflow design easier, the new “Automatic” queue aggregator type has been added. Instead of needing to choose between “Keep,” “Join,” and “Split” behaviors in a workflow, the administrator can leave a queue set to Automatic so that OpenLab Sync will choose the most appropriate behavior for them.

As an example, an analytical workflow that has a Start queue > Protocol 1 > Middle queue > Protocol 2 > and End queue:



All the queues are set to Automatic by default. When samples run from the Start to End queue, the Start automatic queue behaves as a Join and the Middle automatic queue behaves as a Keep queue to keep the samples flowing through as-is. However, when samples start in the Middle queue and run to the End, the Middle automatic queue behaves as a Join so that the protocol run does not error out on duplicate sample IDs.

Non-analytical workflows do not commonly end up with duplicate ID issues the way that Analytical workflows do, so Automatic queues in non-analytical workflows always behave as Keep queues.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.20. OpenLab Sync-NGS → 8.20.6. MiSeq, HiSeq, NextSeq, and NovaSeq Implementations → 8.20.6.1. General Implementation Configuration
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.14. Workflow Management → 8.14.1. Workflow Graph Tab
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.19. Analytical Workflows → 8.19.2. Workflow Management for Analytical Workflows → 8.19.2.1. Analytical Workflow Graph Tab
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.19. Analytical Workflows → 8.19.6. Analytical Use Case
- ✓ OpenLab Sync Administration Manual → 13. Specific Use Cases → 13.1. Advanced Advice for Queue Aggregators in Protocols → 13.1.1. The Automatic Queue Aggregator

4.2 Automations Quality of Life Changes

Macros and OpenLab Sync Extensions (previously called SLIMS GATE flows) have received some quality-of-life updates.

4.2.1 Scan to Execute Extensions Flows *

The “Input Selection” options for OpenLab Sync Extensions in simplified protocol configuration were different from the options for macros, but this has been made consistent. When you select an Extensions flow in an Action block, you may select whether to run the flow on “All,” “Selected samples,” or “Scan.” The Scan option provides a box at runtime where a content can be scanned to run the flow. An error message is provided if the scanned barcode does not exist in OpenLab Sync.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.4. Simplified Protocol Design → 8.4.3. Defining Block Types → 8.4.3.6. Action
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.8. Electronic Lab Notebook (Routine) → 8.8.3. Simplified ELN → 8.8.3.1. Blocks

4.2.2 Empty Number of Content is Allowed in Macros During Design *

Macros can be used to create one or more content or create children from parent content and preconfigure their values. If you enter a value during configuration for how many contents will be created, then that is the default that will be created at runtime. The macro configuration for such a step used to require the number value for how many contents to make, but this field can also be left empty now to force the user to pick the number of contents at runtime. This is useful when the number varies from run to run.

The user can enter a non-negative number to create the desired number of contents. They can also enter "0" to skip the step if it isn't relevant for that run.

You can see this behavior in macro step types that create content or child content: Aliquot content, Copy content, Create container, Create content, Derive content, and Dilute content.

INFO

Reference:

✓ OpenLab Sync Administration Manual → 10. Automations → 10.9. Macros → 10.9.4. Action Configuration Options

4.2.3 Mix Rules can Set Values

Mix rules can now use `mix.setValue` to update values on the mix when the rule runs. The method is available from the autosuggest menu for mix rules This expands mix rules beyond validation-only behavior so they can also set values from linked ingredients to the mix. For example, a mix rule can set a concentration field on the mix based on a main ingredient, copy an expiry date from an ingredient so the mix is used within the correct time frame, or record ingredient barcodes in a custom text field for traceability.

The rule runs when a mix is created, when ingredients are linked to the mix, or when ingredients are removed from it. As a result, the mix can adapt when its ingredient composition changes.

This is useful because standard conditional value expression rules do not have the same early relationship between a mix and its ingredients while the mix is being created. With mix rules, administrators can validate the mix and set mix values based on the ingredients before the workflow continues.

INFO

Reference:

✓ OpenLab Sync Administration Manual → 10. Automations → 10.1. Rules → 10.1.7. Mix Rules

4.3 Can Create Multiple Content with Same ID *

You are now able to create new content with the same ID in bulk. It was possible to create content in bulk already, but to create them with the same ID, they had to be created one by one by entering the ID manually. If OpenLab Sync generates the barcode and the barcode is not used as the ID, then it is possible in bulk.

When creating new content in bulk, the user specifies the amount of content to create and enters a prefix for the ID, but they may leave the start index empty. As a result, all the contents are created with the same prefix value and ID. If a start index is filled, then the IDs have prefixes with incremented numbers starting from the start index. Macros with Content Creation steps may also generate content with the same ID in bulk.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 5. Content Management → 5.1. Content Types → 5.1.2. Category/Content Type Field Explanations
- ✓ OpenLab Sync Administration Manual → 5. Content Management → 5.2. Contents → 5.2.2. Create Contents
- ✓ OpenLab Sync Administration Manual → 10. Automations → 10.2. Fields → 10.2.2. Default Fields
- ✓ OpenLab Sync Administration Manual → 10. Automations → 10.9. Macros → 10.9.4. Action Configuration Options

4.4 Able to Restrict Roles to Edit Dashboard

The role functionality access right to Access Dashboard already existed, but this change adds a new one to "Can edit dashboard." This allows administrators to choose which roles manage dashboards, so they aren't inadvertently changed, and which roles use them in read-only mode. Users that edit dashboards will need both permissions enabled to access the module and have edit options.

"Dashboards" allows role users to access the Dashboards module, select between existing active dashboards, view them as read-only, and refresh. "Can edit dashboard" gives role users the abilities that existed before: Create dashboards, enter edit mode to change columns and widgets, hide inactive dashboards, update settings, refresh, and save.

The new permission is available on Basic (Read-only) and Premium role entitlements. "Can edit dashboard" is enabled by default for those role entitlements on a new or migrated instance.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 3. OpenLab Sync GUI and Navigation → 3.12. Dashboard → 3.12.9. Create a Dashboard

4.5 Version SOPs in Batch *

SOPs can be created manually one by one, but sometimes many are installed at one time in a Store package. Administrators can now version multiple SOPs at once by selecting them with checkboxes, then using the Version action from the Selection menu. This makes it easier to version the bunch of freshly imported SOPs at once. If e-signature is required to version records, only one signature is required for the selected batch.

INFO

Reference:

✓ OpenLab Sync Administration Manual → 9. Audit Tools and Quality Control → 9.2. SOPs

4.6 Support for Kit Collection Store Package

TIP

✓ Kit Collection can only be used if the Store package is installed and the role functionality access right “Can use kit collection” is enabled.

Kit Collection is especially useful for Biobank labs to facilitate the design of different studies, their sample types, how samples will be processed, stored, and collected. The package already existed, but the changes for 7.4 enable the Kit Collection window to be launched from three places: the Dashboards and Content modules and with Extensions flows in Simplified Protocols. The “Kit collection” widget is available and can be added to a Dashboard, and then the user can click it to open the kit collection window. The buttons available for various actions can be changed through the configuration plugin from the store package. The specific plugin UID must match `store_slpg_kitcollection_configuration`.

INFO

Reference:

✓ OpenLab Sync Administration Manual → 6. Studies → 6.4. Kit Collection

4.7 NGS Content Queue Progression

The MiSeq, NextSeq, NovaSeq, and HiSeq protocol steps are used in NGS workflows to process DNA library pools. A DNA library is a prepared DNA mix with primed sequences attached so it can be run through the desired sequencer. A DNA library pool contains multiple libraries that can later be demultiplexed back into individual libraries. Some labs need libraries produced by the sequencer run step to continue through the workflow, and other labs need the library pools to continue.

With this release, a “Content transferred to next step” dropdown was added to the sequencer sample sheet step so the administrator can choose which content is placed in the run step’s output queue after the step is completed. If “Input” is selected, then the library pools continue to the next queue. If “Output” is selected, the libraries continue to the next queue.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.20. OpenLab Sync-NGS → 8.20.6. MiSeq, HiSeq, NextSeq, and NovaSeq Implementations → 8.20.6.3. NGS: MiSeq execution / NGS: NextSeq execution Protocol
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.20. OpenLab Sync-NGS → 8.20.6. MiSeq, HiSeq, NextSeq, and NovaSeq Implementations → 8.20.6.4. NGS: NovaSeq execution / NGS: HiSeq execution Protocol

4.8 Product Specifications Form Display Improvements*

The columns on specification rules were consolidated to save screen space and display only the most useful information. The test table that appears when a specification rule uses a specification type was too small to read the headers, so the table headers appear as tooltips when the user hovers on them. An option that was recently added that allows specification rules to aggregate on an analyte group was renamed for clarity and is now called “Aggregate on the analyte group.”

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.18. Products and Specifications → 8.18.2. Specifications Tab

4.9 Orders Hidden in ELN by Default

Historically, if the “Project” was not set on an order, the order was shown in ELN (Electronic Lab Notebook) at the root of the ELN tree. For most customers, orders are only used to schedule content into Workflows, so the list of orders cluttered the ELN module. Orders are now hidden in ELN by default if there is no project defined on them. If a project is defined, the orders appear in the selected project folder in ELN.

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.8. Electronic Lab Notebook (Routine)
- ✓ OpenLab Sync Administration Manual → 8. Experiments and Workflows → 8.11. Orders → 8.11.1. Order Life Cycle

4.10 Expanded Tables for Conditional Value Expression Rules

When a conditional value expression rule is created, the tables “Study Event Type” and “Attachment link” can be selected. The “Study Event Type” table is for the Study Design

module. By creating rules on these tables, administrators can do things like provide errors that forbid attachments from being linked or unlinked, or provide errors that forbid creating certain visit types in Study Design.

INFO

Reference:

✓ Too fine of detail for a reference

4.11 Agilent Store Package Audit Collection *

When you import a Share package from a server, OpenLab Sync notifies the Share server that the package was imported. This allows Agilent to keep track of which Store and Share packages are in use. During this, no personal information is collected. The only information that is transferred is the version of OpenLab Sync, the name of the entitlement used to activate OpenLab Sync, and the holder of the license (name of the company).

INFO

Reference:

- ✓ OpenLab Sync Administration Manual → 10. Automations → 10.6. Share Configuration Import Tool → 10.6.2. Import Package Tab

4.12 OpenLab Sync Installations Come with SI Units *

Instances that already have units on them could prevent the SI Units package from being installed from the Store module because of unicity errors. To alleviate this, new OpenLab Sync installations come with the SI Units package preinstalled.

INFO

Reference:

- ✓ Too fine of detail for a reference

5 Store Content Related to this Version

This section explores a nonexhaustive list of Store packages related to release changes for OpenLab Sync 7.3b – 7.4. These packages help facilitate or expand on the new features. Unless otherwise stated, these packages are available starting in version 7.4.

5.1 Pharma QC 1.0 (Small Molecules) Collection

The Pharma QC 1.0 (Small molecules) package collection was created to more quickly and easily get pharmaceutical labs digitized in OpenLab Sync. This collection is released starting with version 7.4 and expansions are coming soon with the next couple of patches.

The pharma starter kit can be used in a fresh instance to prepare the build and contains several common content types and tests for various applications. It can be used in combination with various protocol templates available for pharma or as support for a custom workflow setup.

The next large component that adds to the basic kit is checks with control samples, such as blanks and standards. The collection contains many protocol templates to be imported, adjusted for the way your lab works, and versioned in a workflow. In 7.4, there will already be several protocol templates to import:

- Appearance protocol template: One of the first tests performed in pharma QC on a batch. This protocol tracks the result of a visual inspection compared to the SOP as well as visual proof images.
- Loss on Drying protocol template: Compares the weight of a sample before and after it is dried under predefined conditions to track the loss on drying. This protocol tracks the specified weights of the number of representative samples according to the SOP and calculates the difference so the technician can determine the constant weight determination.
- Hardness protocol template: Tests the representative samples repeatedly to ensure their hardness is within limits. This protocol tracks the value of each hardness test and provides a mean aggregation result from each sample's results.
- Friability protocol template: Tests the representative samples to ensure the mechanical strength of the batch against wear and impact. This protocol measures the percentage of weight lost compared to the starting weight based on the SOP. The instrument used, time, and RPM are stored on the protocol. The percentage of weight lost is calculated from the before and after weights.

- HPLC via OpenLab CDS Complete protocol template: Includes a complete HPLC sample processing flow with sample preparation and analysis via OpenLab CDS. It requires the CDS package installed with the Package Browser module. Comes with various basic SOPS, recipes, a rule, and expiration dates suitable for various laboratories.
- HPLC Preparation protocol template: A similar package that only includes the first part of the HPLC complete flow: HPLC sample preparation. Comes with the various basic SOPS, recipes, a rule, and expiration dates suitable for various laboratories.
- HPLC Run via OpenLab CDS protocol template: A similar package that only includes the second part of the HPLC sample processing flow: the instrument run via OpenLab CDS. Comes with the basic SOPs and recipes.

Many of the protocols rely on samples being weighed in the lab, so 7.4 comes with a new balance integration. The first integration supports Mettler-Toledo balances with MT-SICS support on a network, a common balance type, and currently weighs content data and recipe ingredients. The plugin communicates with the balance to discover the options it supports, such as target setting for a recipe ingredient. It comes with supportive Extensions flows to retrieve the balance capabilities and to store weighed values in OpenLab Sync.

Coming for 7.4.1: Customers may also purchase an additional plugin to USP MethodConnect's digitalized information for a wealth of monographs. OpenLab Sync integrates with USP MethodConnect with the plugin so customers can download the monographs they need and update when new versions become available. These packages include products and specifications to import acceptance criteria from USP MethodConnect and SOPS and Recipes with monograph information.

SOPs, recipes, and products must be versioned after they are imported before they can be used in a protocol. They can then be restricted to certain workflows or protocols and wired into a protocol using workflow paths.

5.2 Training Plan Package

This package contains basic custom fields and reference data types to facilitate training groups, including email alerts about upcoming training expiration dates, a training matrix report, and more. This package was released after the 7.3b release and is compatible with OpenLab Sync 7.3 and after.

6 Deprecated Features

Features may be removed or replaced between versions to provide improvement and to keep up with dependent software. These notes are about features that are being replaced or deprecated in OpenLab Sync 7.3b – 7.4.

- The lab setting “Suppress concurrent modification errors” is heavily discouraged. The concurrent modification error is necessary and useful to prevent loss of data from users saving over each other.
- Container types are a feature without a clear alternative; however, it was found that they do not suit many use cases. Usage is discouraged if there is a better alternative for your use case.

7 Appendix 1: OpenLab Sync License Information

The modules that are available in a customer's instance differ based on the OpenLab Sync License Type that the customer has purchased: Standard or Premium.

The following table lists the modules by the groups that are found in the OpenLab Sync main menu and indicates whether the modules are available under a Standard License. Modules that are not available under a Standard license require a Premium license to be available in the customer's instance. Some modules require an additional Add-on license before they can be used.

The table lists the modules for OpenLab Sync 7.4 as they are added.

Table 1 Product Modules and License Type

Module Groups	Modules	Basic Requester	Basic Read-Only	Standard License	Premium License	Requires Add-on License
Inventory	Content	○	○	○	○	
	Locations		○	○	○	
	Instruments		○	○	○	
	Content Types		○	○	○	
	Container Types		○	○	○	
	Printers		○	○	○	
	Reports		○	○	○	
	Attachments		○	○	○	
	Attachment Types			○	○	
	Analytes		○		○	
	Providers		○	○	○	
	Customers		○	○	○	
	Sources		○	○	○	
	Diseases		○	○	○	
	Reference Data		○	○	○	
Process	ELN		○	○	○	
	Protocols			○	○	
	Workflows		○		○	
	Workflow Management				○	
	Requestables				○	
	Requestable Groups				○	

Appendix 1: OpenLab Sync License Information

Table 1 Product Modules and License Type

	Worklists		○		○	
	Worklist Templates				○	
	Orders	○	○		○	
	Order Types				○	
	Runs		○	○	○	
	SOPs		○	○	○	
Monitoring	Dashboards		○		○	
	Grids		○	○	○	
	QC		○		○	
	Rule Evaluations		○		○	
	Stability Study		○		○	○
Design	Grid Templates			○	○	
	Widgets				○	
	Tests			○	○	
	Test Groups				○	
	DNA Index Sets				○	
	Plate Index Design				○	
	Sequencing Settings				○	
	Content Event Types				○	
	CRF Templates				○	
	Studies		○		○	
	Study Design		○		○	
	Products and Specifications				○	
	Specification Types				○	
	Stability Study Design				○	○
	QC Templates				○	
Automation	Scan			○	○	
	Macros			○	○	
	Plugins			○	○	
	Status Workflows			○	○	
	Rules			○	○	
	Scheduled Jobs			○	○	
	Email Templates			○	○	
	Report Templates			○	○	
	Label Templates			○	○	
General	Fields			○	○	

Appendix 1: OpenLab Sync License Information

Table 1 Product Modules and License Type

	Units			○	○	
	Sequences			○	○	
	Store			○	○	
	Share			○	○	
	Package Browser				○	
	Lab Settings			○	○	
User Management	Users			○	○	
	Roles			○	○	
	Groups			○	○	
	Layouts			○	○	
	Holidays				○	
	Authentication			○	○	
	License	○	○	○	○	

Table 2 Additional Features and License Types

Feature	Type	Basic Requester	Basic Read-Only	Standard License	Premium License	Requires Add-on License
Requester Portal	Separate Add-on functionality	○		○	○	○
Bench View	An interface for tablets included with OpenLab Sync		○		○	

Version History

Revision	Changes	Date
Draft (for 7.3b)	Enhancements in 7.3 Bridge were added. Installation and upgrade changes, and deprecated features were added.	07/AUG/2025
A.00 (for 7.4)	Enhancements in 7.4 release were added. Installation, upgrade, and deprecated features were added. Product name was changed from SLIMS to OpenLab Sync. Enhancements done between versions were added in Version 7.3b section.	16/SEP-2026

In This Book

This document provides a cumulative listing of the major feature modifications made in the OpenLab Sync 7.4 release of the Agilent OpenLab Sync Software.

www.agilent.com
© Agilent Technologies, Inc. 2026
DE-017715

Edition 09/2026

Agilent