Agilent SLIMS

SLIMS 7.2 Version

Technical Release Notes

Notices

Document Identification

DE003272 Rev. 7.20

January 13, 2025

Copyright

Agilent Technologies, Inc.
5301 Stevens Creek Blvd.
Santa Clara, CA 95051
USA

Software Revision

This guide is valid for the SLIMS 7.2 revision of Agilent SLIMS software.

Warranty

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

Technology Licenses

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

Restricted Rights Legend

U.S. Government Restricted Rights. Software and technical data rights granted to the federal government include only those rights customarily provided to end user customers. Agilent provides this customary commercial license in Software and technical data pursuant to FAR 12.211 (Technical Data) and 12.212 (Computer Software) and, for the Department of Defense, DFARS 252.227-7015 (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

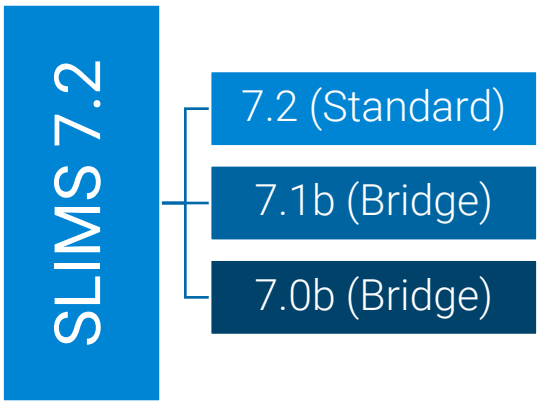
DANGER indicates a hazardous situation which, if not avoided, will result in death or serious injury.

Table of Contents

1 Introduction

This document provides a cumulative listing of the major feature modifications made in version 7.2 of the SLIMS Software. It includes the installation and use of enhancements and any requirements needed to enable the new features such as role permissions and SLIMS 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 SLIMS is developed and tested in more frequent and feature-focused versions.



7.0b is the first bridge release for Agilent SLIMS 7, building toward a cumulative 7.2 version that is supported for 3 years with 1-year limited support. The updates in this document are cumulative for SLIMS 7.2 so customers can see the features that were added in each version making up SLIMS 7.2.

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 SLIMS or enhance the customers' ability to programmatically customize SLIMS.

2.1 Installation Changes

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

- Supported Operating Systems and Databases:

  The Technical Requirements can be checked to see the full list of supported operating systems and databases.

  - Windows Server 2022 is now supported for SLIMS installations. See the Requirements documentation for more details.

- JDK Version:

  - SLIMS upgraded to Java 21 for the SLIMS 7 releases.

- SLIMS converted to Spring Boot so Tomcat and JDK now come bundled with SLIMS. With this change, an actuator was introduced which provides some stats about the system health.

- Technology upgrades:

  - Upgraded to Camel 4. No migrations should be needed.
  - Upgraded to Jasper 6.20. No migrations should be needed.

- New Startup Flag: By default, Excel with macros is now disallowed for imports.

  - An error message was added to guide users when using Excel imports. With the necessary role functionality access rights, users can update content by importing an edited Excel file from SLIMS. However, if an Excel macro is added to manipulate the file and the datatype changes to .xlsm, import of the file can cause a crash. To prevent the issue, this filetype is no longer allowed and an error message warns the user when they attempt to import .xlsm files.
  - For labs that rely on macro manipulation, contact your SLIMS administrator to see how to use .xlsm files with SLIMS. Refer to the SLIMS Installation Manual for the necessary startup flag.

2.1.1 Simplified Installation for SLIMS

SLIMS is installed with a Windows wizard or a Linux package installation starting in version 7.0b. This makes it easier and faster to install SLIMS and standardizes the process to cut down on human error.

- On Windows, an executable file is downloaded and runs the wizard which asks for the SLIMS installation path and SLIMS properties file path. An option is provided to install Agilent Licensing services and Apache HTTP Server at the same time. If selected, the installation process will set up Agilent Licensing services and/or the http service.
- Linux installation comes as an RPM package, not a graphical installation wizard. One command is entered to install SLIMS, SLIMS-R, and SLIMS GATE without customization options. The Apache HTTP Server comes as a dependency, and Agilent CLL must be installed separately.

The Windows installer deploys the SLIMS, SLIMS GATE, SLIMS-R files, and a Java 21 JDK installation in the selected file path. If an attachments folder to store SLIMS attachments on the local hard disk doesn't exist, the installer creates one. The SLIMS, SLIMS GATE, and SLIMS-R services are started and can be monitored using Windows Services.

SLIMS Rest is included in the SLIMS installation to simplify its installation as well. The URL scheme to access SLIMS Rest documentation changed. Customers with "Can use the REST API" enabled now go to …/slims/slimsrest/v1/doc to access the SLIMS Rest documentation.

2.2 Upgrade Report

The upgrade report is provided after migrating from SLIMS version 6.9 to a SLIMS 7 version. Technical customers or SLIMS delivery helping with the migration can read the report to see everything that was touched during migration and use it to guide retesting. The upgrade report combs through records from all modules, as well as filters, views, grids, macros, Groovies, location types, locations, instrument types, instruments, etc. to find what changed. It includes Groovies that could be suspected of not working after the upgrade and Groovies that would make use of backwards compatibility so they can be checked and fixed.

The upgrade report is provided as HTML via SLIMS Rest (for example: …/slims/slimsrest/v1/upgrade/report).

2.3 Customization Changes

These upgrades are relevant for customers who do their own customization in SLIMS. The changes for SLIMS 7.0b – 7.2 are as follows:

- Upgrade to Vaadin 24: Customers using Vaadin customizations were on Vaadin 8 which is reaching its end of life, so we are enabling Vaadin 24 to make migration possible to a supported version.

- Tests are migrated to JUnit 5. Customization developed by customers can still use JUnit 4 or can convert to use 5.
- Import compatibility:
    - SLIMS Share, SLIMS Store, and plugin package development is changed to support the new cumulative release process. This ensures preconfigured packages are compatible from 7.0b (the first version) to the last SLIMS 7 version. Note that a 6.9 package still won't be applicable for SLIMS 7 versions, nor a SLIMS 7 version for a 6.9 SLIMS.

      In addition, later packages cannot be applied on earlier versions. For example, a plugin from 7.1b will be importable on 7.3, but a plugin from 7.3 will not be importable on 7.1b.
- Groovy compilation validates when saving fields. It was already possible to do so by self-monitoring the execution log of Groovies when saving records in the relevant areas of SLIMS. However, Groovy expressions now attempt to compile as soon as the Save button is clicked. It functions in the most common use cases: the Fields module and conditional value expression rules. This is not intended to catch every problem, but it catches basic syntax errors (like missing brackets).

  The line numbers in the log are not relevant to the actual expression; however, it enables administrators and engineers to quickly tell when the compiler cannot compile the Groovy.
- SLIMS is no longer shipped with default dependencies for Groovy because several were not being used. The core libraries were trimmed down to make the system leaner and there is no expected impact to existing customer scripts. These are the Groovy libraries that are still packaged to support SLIMS scripting:
    - `groovy`
    - `groovy-datetime`
    - `groovy-json`
    - `groovy-nio`
    - `groovy-xml`
- The way of storing hard-coded constants in the database was replaced with a more flexible mechanism using a queries/service class. They describe the default status for results, orders, and are used to configure the custom fields added to pedigree drawings, affected field, male value field, etc. It no longer made sense to change the default values for orders and results, so they are not editable anymore.

  Any nonstandard configuration that customers had is captured in the upgrade report when they migrate to SLIMS 7.x. However, the configured custom fields can still be modified by finding their keys in the properties file and updating them. The key names can be provided if needed. (Reference: SLIMS Administration Manual → 6. Studies → 6.3. Pedigrees → 6.3.5. Custom Configurations)
- 9.6.9. Remote Daemon Authentication for Agilent Hosted SLIMS. The library was changed for SLIMS 6.9 which introduced a requirement that the remote daemon needed to reauthenticate periodically (every 12 hours). This was changed so that remote daemons are valid for a year so they will only need reauthentication each year.
- These were the changes to SLIMS API services between SLIMS 6.9 and 7.2. The specific SLIMS versions the APIs can be found on are:

- 6.8.8. RuleEvaluationService: Several methods were added with this service that support the acknowledgement of rule evaluations via plugins. Rule evaluations from plugins work similarly to result rules or product and specification evaluations.
- 6.8.8. EnrollContentInStudyService: The service already existed and allowed content to be enrolled in a study but did not allow adding enrollment details. The service is being extended with a method to allow enrollment details to be added when enrolling content and provides checks to prevent the wrong content type from being enrolled, and prevents redundant enrollment.
- 6.8.9. DisplayableEntityService: Provides a way to transform a record into a DisplayableEntity, which contains relevant useful information about a record (like the field names, titles, values, display values, etc.). It can be used to provide a human readable version of a record.
- 6.9.1. recordQueries classes for all tables now have findByUniqueIdentifier available. recordQueries didn't contain a method to find by unique identifier so it has been added to allow any recordQueries class to fetch by unique identifier. The return value is a `Map<String, Object>`.
- 7.0.0. authenticatedUser can be used in a dynamic filter expression on built-in dynamic choice fields such as products and analytes, and on custom dynamic choice fields. authenticatedUser was added as an option in the Groovy Insert menu and allows a Groovy expression on a field to filter the values of the field based on the current user's rights.
- 7.0.0. differencesMap can be used on conditional value expression rules and status rules. This change allows rules to be defined based on the update of a value on specified fields for a given record. For example, a conditional value expression rule could be made that checks the "Lot number" field for the status "Available" and provides a warning "This content field cannot be updated in status Available" if the user tries to update it.
- 7.0.0. Rest calls account for paged protocols to ensure that basic fetches and advanced operations on entities in the children protocols of ELN can be performed. (Paged protocols are described in a later note.)
- 7.0.0. Users can return the Unique Identifier from the target table to allow for simpler expressions to find records. SLIMS will automatically find and link the records whose UID was returned.

  Instead of an expression like this:

  ```
  return daoHelper.findByUniqueIdentifier("ReferenceDataRecord",
    "rdrc_countries_belgium").rdrc_pk
  ```

  This simple expression can be used:

  ```
  return "rdrc_countries_belgium"
  ```

  There are some behaviors to be aware of:

  - Target tables where the UID is a number, like content barcodes, are special cases. SLIMS can't tell if an all-numeric return value is a barcode or primary key, so it is always interpreted as a primary key. In the past, a numeric return value could have only been a primary key, so this behavior preserves Groovies that existed before this change.
  - In fields with single choice, the field will be left empty and reported in the log if the Groovy doesn't find the record. In fields with multiple

choices enabled, the found records are returned, and any missing ones will likewise be skipped and reported in the log.

- 7.0.0. Criteria allow use of UIDs in place of Primary Key values when defining a criterion that searches in a Foreign Key field. When writing Groovies or plugin code, it was already possible to use UIDs, but it came in two expressions: one to query the UID to find out the primary key of the entity, and another to use the query result in what you wanted to fetch. This has been simplified so users can pass in UIDs to daoHelper and query-classes. If the UID isn't a number, SLIMS can tell it's the UID of an entity like a specific content type and not a primary key.

  UIDs can be used this way in Groovy, plugins, and SLIMS Rest calls. The results of the queries are cached to help ensure it will be as fast as possible. Existing queries written with the two-part structure will still be supported.

  This change enables SLIMS GATE filters to use UIDs to filter on value tables. This was possible before using hard-coded values, but using UIDs provides a more consistent and transferable result (for example, when importing to another SLIMS via SLIMS Share). Existing filters with hard-coded values are still supported as well.

  Example of new use of UIDs:

  ```
  allBlood = daoHelper.fetch("Content",
  SlimsRestrictions.equals("cntn_fk_contentType", "cntp_blood"))
  ```

- 7.1.0. Fluent Groovy API has several improvements. This list is not exhaustive, but the most important updates are listed:

  - You can start a query from many more tables. For example: `global.attachments()`
  - You can now pass unique identifiers instead of a PK to `.with("fieldname", "uniqueIdentifier")`
  - You can now pass in a collection of values to:

    `.with("fieldname", [1, 2, 3])`

    which can also be a list of unique identifiers. This is interpreted as a "one of" filter.
  - Methods have been added for working with versions. For example:

    Queries: `global.experimentTemplates().withStatusPublished()`
  - Links: `experimentTemplate.retrieve().draft()`

- 7.2.0. Groovy provides methods to supply the environment in which an action/macro is running (in a simplified protocol run). These completions will be available in the insert box of Groovy grouped under `.getActionBlock()` to use as context in more complex expressions.

  - Return the action block pk

    `macroExecutionDetails.getActionBlock().getActionBlockPk()`
  - Return the input block pk

    `macroExecutionDetails.getActionBlock().getInputBlockPk()`
  - Return the output block pk

`macroExecutionDetails.getActionBlock().getOutputBlockPk()`

- Return the page pk

  `macroExecutionDetails.getActionBlock().getPagePk()`

- Return the run pk

  `macroExecutionDetails.getActionBlock().getRunPk()`

- 7.2.0. Methods were added to the Fluent Groovy API to get to records via access rights and workflow protocol links.
  - Navigate from a group to its user and vice versa

    `group.retrieve().users()`

    `user.retrieve().groups()`
  - Navigate from a role to its users and vice versa

    `user.retrieve().role()`

    `role.retrieve().users()`
  - Navigate from any record to its attachments

    `record.retrieve().attachments()`
  - Navigate from Protocol step to Protocol

    `experimentStep.retrieve().protocol()`
  - Navigate from Protocol Run to Protocol

    `experimentRun.retrieve().protocol()`
- 7.2.0. A method was added that sends a delayed email. It was already possible to have Groovy that sends an email immediately triggered by an action, but the email is sent even if an issue occurs after that. For example, an order is Completed, an email is sent to notify about the status change, and then an issue in the order makes the status revert. This method delays the email until the whole transaction is completed to ensure there aren't errors, and if anything happens to make the email fail, it doesn't send. In the example, the order is Completed but the email waits for the whole transaction to process, so when the issue occurs and the order reverts, the email doesn't send.

  The method is only available for Validation timing (new feature for conditional value expression rules). These options were added:
  - `triggerDelayedEmail(String emailTemplateName)`
  - `triggerDelayedEmail(String emailTemplateName, Map params)`
- 7.2.0. As paged protocols were added to SLIMS with the 7.x versions, one of the parameters used in a SLIMS GATE flow in an action block changed what it returns. In 6.9, the `slimsFlowInitParam.getExperimentRunStepID()` parameter returned the primary key of the action block it was executed in. In 7.x, the parameter returns the primary key of the page the block is on, and not the block itself. To get the pk of the block the SLIMS GATE flow was started in, or the pks of that block's input or output blocks, you can use new parameters.

  The `slimsFlowInitParam` object can return the pk of:
  - `SLIMS_ACTION_BLOCK`

- SLIMS_INPUT_BLOCK
- SLIMS_OUTPUT_BLOCK

using calls like:
`slimsFlowinitParam.getInputParameterValues().get(SlimsgateParameter.SLIMS_ACTION_BLOCK)`

- 7.2.0. Filter operators are now supported in slimsgate.xml so that valueMap elements can have more complex filters. For example, filtering the input parameter of multiple-choice fields for a particular value. The filters are described in the SLIMS Development Manual, section 3.5.7. Parameters of Type SingleChoice and MultipleChoice.

3 Enhancements in Version 7.0 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 SLIMS Version 7.0b.

3.1 SLIMS User Interface Refactored *

Agilent SLIMS version 7.0b comes with a refreshed look and feel that matches the rest of the Agilent OpenLab software suite. These changes affect the color and style of these areas:

- Login Page: Contains the same fields, but the color, shape, and size are adjusted.
    - Third-party authentication provider login buttons
    - Progress page when loading into a SLIMS session
- Top Bar: Contains the same options and pinned modules, but the colors, icons, hover colors, outlines, and highlights are adjusted.
- SLIMS Main Menu: The list of modules is shown based on role access, but their icons, hover color, text, search bar color, and favorite action icon are adjusted.
    - The modules have been reorganized into different groups to help find them more easily by purpose.
- Buttons: Changed to a matching color scheme and squared off shape throughout the software with refreshed icons.
- Grids: Lines are added between rows and columns to better distinguish between data in the grids

INFO

Reference:

- ✓ SLIMS Administration Manual (throughout)

3.2 Locations are now Content *

Locations were defined by Location Types in previous versions of SLIMS but are now being folded into Content Types so labs can more easily set up locations that are used as contents to link them to orders, run them through a workflow, and track them alongside their contents. This also consolidates the setup modules and makes them easier to use.

Locations are now defined as content types with the new “Can be used as a Location” option enabled. Contents designated as locations appear in the location tree of the Contents module and get the benefit of location behaviors from prior versions of SLIMS. For example, they can still be linked in a “Link Location” protocol step and used to filter the contents they contain, but they gain the benefits of useful content type definitions.

Location content types are collected in a “Locations” category with “Can be used as a Location” turned on for the whole category. Customers can add more location categories if desired, but the default one is provided to collect existing locations from prior versions of SLIMS when upgrading. Location content types get the relevant content type fields as options but also additional fields that are just for locations.

Location content statuses are managed with the default content status workflow like other contents. Active location contents are built into a location tree in the Locations module or Content module. Location contents come with barcodes, barcode mask options, short ids, and short id prefixes to give them identification. Then, the location contents can be processed in a workflow alongside their sample contents. The location (such as a plate) can be linked in its entirety or scanned to link all of its sample contents into the protocol.

An exhaustive explanation to set up and use content locations is found in the SLIMS Administration Manual. However, customers migrating from SLIMS 6.9 should proceed to the next note for details.

INFO

Reference:

- ✓ SLIMS Administration Manual → 5. Content Management → 5.1. Content Types and subsections
- ✓ SLIMS Administration Manual → 5. Content Management → 5.2. Contents and subsections

3.2.1 Migration of Locations to Content from 6.9 to 7.0b

With the change from locations into contents, the process to migrate from another version of SLIMS to 7.0 has been thoroughly considered. Existing location types are migrated into content types of the "Location" category and the fields that define them are migrated to new content fields related to the locations category. The upgrade should keep the following: existing location types, macros, grid template definitions, custom fields definitions and values, views, report templates/reports, location links in runs, location detail Excel templates, and Groovy value expressions without losing data.

- Location barcodes are merged into content barcodes on migration. In cases where a barcode is already taken, the location barcode will be appended with the prefix "lctn70_". The prefix may cause issues because the changed barcodes may not match printed labels. However, any modified barcodes will be included in the migration report so customers know which were changed and can adjust them afterwards.
- Default fields could be set by using Groovy expressions, for example, to give location barcodes a specific naming scheme. If such changes exist on a prior SLIMS, they are ignored when migrating to SLIMS 7 to prevent them being applied across unintended content types. However, any ignored changes to default fields are listed in the upgrade report.
- Descriptions are migrated into a "descriptions migration field" for location (and instrument) types.
- Custom fields: The fields for location types were mapped to content types. Custom fields on locations could be restricted to certain location types. Unrestricted fields become content fields that are restricted to the Locations category, and restricted fields become content fields that are restricted to the correct content types. Sublocation restrictions that dictated which kind of locations could be stored inside another location have migrated into an "Adding content is restricted to certain types" option. The restricted content types can further be selected, as well.
- Dynamic fields: Custom field dynamic choice values could have been created that point to locations or location types. They are mapped so that after migration they point to the migrated content or content type. Both the field definitions and values were considered so no data is lost.
- Status: Locations could only be active or inactive before, but contents have a status workflow with multiple statuses. During migration, locations that are active get the "pending" content status, and inactive ones get the "removed" content status. To circumvent issues with making root locations "available," statuses that would usually require root locations do not enforce in such a way for root locations.
- Flat fields: Fields for locations, instruments, location types, and instrument types are deflattened during migration if they were flat in 6.9 so that they can be migrated to 7.0 without problems. However, once they are in 7.0 as contents or content types, they can once again be flattened.
- Content macros like those with derive, copy, and creation steps, and content event macros with content event creation steps could have dynamic choice fields pointing to locations and location types on them. These values are retained during migration so the macros are upgraded and should still work. The "Create location" macro step is replaced with "Create content." They are noted in the upgrade report so they can be reviewed.
- Report templates that were created on location and location type are migrated to be reports on content and content type, respectively. There is a new "tag" attribute for Jasper reports that is filled automatically during migration if there were metatrees on locations, instruments, or their types. Only power users will see the tag when executing the report draft. The tag ensures the report uses the correct tree structure to get the necessary references.
- History: The legacy prefix "lctn" for custom fields is allowed in the history and for making fields flat, as well as the new "cntn" designation for locations, because the history must retain the original values. This allows the printer job history and location history to be retained properly. (The same is true of instrument history, but with the legacy prefix "nstr.")

- Metatrees were adapted to filter correctly after migration. Locations and location types used as the content root have “can be used as a location” applied as a filter so they show the location contents. The filter is added to the children and grandchildren of the root so child locations are returned as well, and not contents for all the children.
- Runs: Location links in runs are retained by transforming the ExperimentRunStepLocation table into ExperimentRunStepContent. That way, the locations linked to experiment steps of type "Link Location" are still linked to the experiment after migration.
- Location Excel templates can be generated to export a 2D representation of a location if it has positions. This used to be accessed from the Location Detail tab in the Content module, but will now be in the Content Types module, available for content types in the Location category (and for any content types that can be used as a location). After migration, any Excel templates that were attached to location types are attached to the content types instead.
- Groovy: SLIMS administrators and engineers may have to rewrite Groovy expressions after migration, but to make the transition more straightforward, some backwards compatibility with SLIMS 6.9 has been introduced. Anytime backwards compatibility is used, a warning appears in the log to alert the admin. SLIMS scans Groovies and reports when they are suspect and need to be rewritten. For example, writing a daoHelper.fetch on the Location table that fetches all locations should work after migration and returns all contents with "Can be used as a location." A warning using backwards compatibility might look like: “Use of the backwards compatibility layer detected, please convert any daohelper calls on <location> to <content>.” daoHelper.count on a location type (or instrument type – see Migration of Instruments) should give the same counts in 6.9 and 7.0. More complex criteria do queries on the contents to check for content that can be used as a location (or instrument). Those should have the same count and results in 6.9 as 7.0.
- REST API: Backwards compatibility extends to SLIMS Rest as well, so other systems querying the Rest API don’t have to be adapted. Endpoints will still work and can be filtered, and the returned entities look like content records with location attributes, though they still have location-relevant columns like “lctn_name.” However, only reading data (GET) is supported with backwards compatibility, not POST / PUT / DELETE.

  The migration process cannot handle a limited number of specialized criteria, like detachedCriteria, which will need to be translated manually. An upgrade report is provided on migration that identifies Groovy issues to make them easier to find and fix. Additionally, a full list of incompatibilities is available in the Upgrade Report including guidance of how to rewrite those criteria.

INFO

Reference:

✓ Migration changes are noted in this document, not the SLIMS Administration Manual.

3.3 Instruments are now Content *

Instruments were defined by Instrument Types in previous versions of SLIMS, but they are now defined by Content Types to benefit from content features, like the ability to add

results to them and send them to workflows. The Instrument Types module has been removed to further consolidate setup features into Content Types.

Instruments are content types with the “Can be used as an Instrument” option enabled, defined by fields on the content type. Contents designated as instruments can be created, imported, and exported in the Instruments and Content modules. The “Can be calibrated” switch has been removed to simplify calibration configuration, leaving a “Calibration Expiration Date Field” setting on content types. A date custom field can be created and restricted to any desired instrument content type categories and then selected to be the calibration expiration date. (This is set up by default if migrating: details in next note.) When selected as the calibration expiration date, the field is used in conjunction with the default and simplified protocol settings involving whether uncalibrated or expired instruments can be linked, whether the user is warned when linking them, and for default protocols only, whether flags are applied.

Protocol runs can be restricted by instrument content types. Protocols for ELN are run under the required instrument in the ELN instead of within a project when they require an instrument content type. Workflow protocols can restrict which instrument content types to use for a protocol step, including the content selection step. Linking contents can be permitted or disallowed depending on their status as well. This is controlled using a status workflow. Contents use the Default Content Status Workflow by default, but a custom one could be created specifically for instrument maintenance. The key status configuration is the setting “Records with this status can be linked in protocol runs” which is enabled by default but can be disabled to filter out any contents (including instrument contents) that are in that status during a protocol run. Once a run in ELN or Workflows is started, it can be found in the Runs module and searched by instrument.

Instrument content types are collected in an “Instrument” category with “Can be used as an Instrument” turned on for the whole category. Customers can add more instrument categories if desired, but the default one is provided to collect existing instruments from prior versions of SLIMS when upgrading.

An exhaustive explanation to set up and use instrument contents is found in the SLIMS Administration Manual. However, customers migrating from SLIMS 6.9 should proceed to the next note for details.

INFO

Reference:

- ✓ SLIMS Administration Manual → 5. Content Management → 5.1. Content Types → 5.1.2 Category/Content Type Field Explanations
- ✓ SLIMS Administration Manual → 5. Content Management → 5.1. Content Types → 5.1.4. Instrument Content Types
- ✓ SLIMS Administration Manual → 5. Content Management → 5.2. Contents → 5.2.4. Manage Instruments
- ✓ SLIMS Administration Manual → 8. Experiments and Workflows → 8.4. Simplified Protocol Design and subsections
- ✓ SLIMS Administration Manual → 8. Experiments and Workflows → 8.5. Default Protocol Design
- ✓ SLIMS Administration Manual → 9. Audit Tools and Quality Control → 9.6. Status and subsections

3.3.1 Migration of Instruments to Content 6.9 to 7.0b

The process to migrate from instrument types into content types from 6.9 to 7.0b has also been thoroughly considered. Existing instrument types are migrated into content types of the "Instrument" category and the fields that define them are migrated to the new content type fields related to that category. The upgrade should keep the following: existing instrument types, macros, custom fields, definitions and values, views, report templates/reports, runs related to instruments, ability to import/export instruments, and ability to use the OpenLab CDS and NGS packages.

- UID: Instruments did not have barcodes in 6.9, so the unique identifier of an instrument becomes the barcode of the instrument content after migration. Instruments did not have IDs either, so the name becomes the ID after migration. In the rare case when a content with a barcode exists that matches the unique identifier, then the instrument gets "nstr70_" prepended to the barcode to ensure it is unique.
- Location path: The location selected for each instrument is migrated, but only if it was positionless, because the available spots for a content in positionless locations are infinite. The instrument location field becomes empty during migration if the location has positions, because there could be contents already taking up the available spots in the location.
- Calibration: The default instrument status workflow is retired and no longer used, so it is not migrated. The "Can be calibrated" field wasn't used and has been removed in SLIMS 7. Any instruments that had this field enabled and had a calibration expiration date will retain their expiration dates. A setting for "Calibration expiration date field" will be present in the Content Types module, and it will be set to a custom field labeled "Calibration expiration date" for instrument content types after migration. The custom field is restricted only to the instrument content types that had calibration enabled so that it won't be displayed on irrelevant content types. Any instruments that had a calibration expiration date will have the date value migrated.

  Simplified and default protocols that allowed or disallowed using uncalibrated instruments are migrated accordingly. The setting to warn the user is migrated as well. They are triggered according to the calibration expiration date value. However, "Add the following flags" is currently unsupported for simplified protocols.

- Status: Instrument types had an "Active" status and Content types do as well, so the active or inactive status migrates neatly. However, Instruments used both:
    - A Default Instrument Status Workflow in 6.9 that had "Unavailable" and "Available" statuses, whereas Contents use the Default Content Status Workflow with a range of statuses. The "Records with this status can be linked in a protocol run" option was added to status workflows including the Default Content Status Workflow and custom ones. However, it is not possible to migrate the status workflows neatly, so it will need to be recreated after migration. A custom status workflow can be recreated for instruments with a status that allows linking to protocol runs and a status that disallows it, or this option can be disabled on a selection of the Default Content Status Workflow statuses.
    - An "Active" status. Inactive instruments are migrated to become contents with the "Removed" status and active instruments are migrated to become contents with the "Pending" status.
- Custom fields: Custom fields on instruments that were not restricted to an instrument type are migrated to be custom fields on content, restricted to the instrument category. Custom fields that were restricted to some instrument types are migrated to be custom fields on contents restricted to the equivalent instrument content types.
- Dynamic fields: Any custom field definitions that point to instruments are kept. For example, dynamic choice fields pointing to the instrument table now point to content, filtered to those whose type "Can be used as an instrument." The same field values are retained on upgrade, whether they are flat or nonflat fields, and point to the correct instruments.
- Filter fields: Custom fields in views can point to many tables, including instruments. They now point to contents instead. Filters and highlights on custom fields that refer to instruments are the same after migration, so customers should still see the same filtered contents and highlights.
- History that existed on instruments and instrument types is migrated and retained to show all events. The history of all instruments can be found using the action menu where the history shows even the deletion of instruments.
- Macros referring to instruments are migrated such that custom fields pointing to instruments now point to contents and still refer to the right instruments. They should still run correctly and provide the same configuration and outcome.
- Jasper reports on instrument and instrument type are migrated to content and content type, instead.
- Metatrees were adapted to filter correctly after migration. Instruments and instrument types used as the root have "Can be used as an instrument" applied as a filter so they show only the instrument contents. The filter is added to the children and grandchildren of the root so child instruments are returned as well, and not contents for all the children.
- Rules on instruments were migrated to become rules on contents. To prevent cases where instrument rules apply to all contents, cases will be included in the upgrade report. The Groovy expressions for the rules will need to be modified after migration.
- The NGS package was adapted so it still imports and works as expected with instruments as contents. Attachments and reports linked to instruments or instrument types also become linked to the migrated contents and content types respectively.

INFO

Reference:

✓ Migration changes are noted in this document, not the SLIMS Administration Manual.

3.3.2 CDS Package Updated to Work with Migrations *

The OpenLab CDS package provided via the Package Browser module was adjusted to work with the changes to instrument types. Some default fields were dropped with this change because they were never used:

- olssIsLocked
- olssApplicationId
- olssDefaultProjectId
- olssDefaultProjectName
- olssAlwaysUseDefaultProject
- olssIsConfigured

The package now comes with four content types that are imported into the Instruments category: Gas Chromatograph, Liquid Chromatograph, Mass Spectrometer, and Generic Instrument. The custom field "nstr_cf_olssId" is imported as well, but it is restricted to the four imported instrument content types. After the package is imported, the instruments can be synchronized from OpenLab CDS to import them as contents.

INFO

Reference:

✓ SLIMS Administration Manual → 8. Experiments and Workflows → 8.6. OpenLab Software Suite Integrations

3.4 Simplified Protocol Pages *

TIP

The "Default Simple Protocol" lab setting was added that allows simplified protocols to be selected by default when users create new experiments in ELN.

Both default and simplified protocols have been consolidated into the "Protocols" module to use in ELN, and the "Simplified Protocols" module has been deprecated. These changes improve the user experience as SLIMS moves toward primarily using simplified protocols and phasing out default protocols. Simplified protocols now resemble a physical lab notebook with pages to separate lists of blocks into several screens. This makes the operation of a long protocol easier in an ELN or workflow. Default protocols do not get the upgrade to pages; however, they can be converted into simplified protocols (more information is provided in the next tech note).

Simplified protocols are created with a default page so blocks can be added quickly, and if the protocol is for a workflow, a "Pop Block" (replacing the "Select Content from Queue" step) is automatically added as the first block on the page. Other blocks can be dragged and dropped into a different order on the page and can even be moved between pages with a selection list.

In workflows, the breadcrumb chevrons in a protocol run now show navigation between pages in a simplified protocol. They still show navigation between steps in a default protocol. Editing is also still restricted when navigating with the chevrons. The “Back” and “Next” buttons should be used to edit prior pages or steps. In ELN, the breadcrumbs show the Project > Experiment > Experiment Run > Experiment Pages. Also, in ELN, pages and steps can be finished out of order (unless configured otherwise) so users can freely navigate between them. Pages and steps are progressed using a protocol status workflow that can be selected when the protocol is designed (more information about protocol status workflows in a separate tech note).

Tagging has changed because of the adjustment to pages. ELN tags are at the top of the experiment canvas and now apply to entire experiment, not to individual pages. Printing has been improved for ELN and workflows, providing the option to print the whole protocol run or just the page. ELN allows users to select multiple runs or pages and print them in batch.

Content progresses through a workflow with simplified protocol pages in consideration with input, extra, and output blocks. Link content blocks have the “Can requeue or cancel in this block” option unless they are “extra” blocks so administrators can permit requeuing or cancelling in certain blocks. Content progresses from the queue through a protocol from one page to another. Input blocks allow content to be linked and processed in other blocks in the page or the following pages. Content created (or derived) from the input is captured in an output block and progresses to the next pages and then the next protocol. Content linked into an output block just goes to the next page, not the next protocol. Finally, content in extra blocks doesn’t progress to the next page or protocol. The “Content transferred to next page” selections allow these behaviors to be adjusted.

There are some situations to consider when building pages:

- Pop blocks are always on the first page. Validation messages warn if there is more than one input block on a page, so an input block can’t be put on the first page in addition to the pop block. Input blocks can be on a different page from result blocks but must be before the result block. Action blocks are similar but must be on the same page as the block they act on.
- “Use as experiment level” allows there to be multiple runs in a simplified protocol experiment. This makes it easier to migrate default protocols to simplified protocols. The converted protocol will keep using the experiment level as a simplified protocol.
- The “Instrument Required” option was moved to the page level on simplified protocols and to the block level on Form blocks.
- A grid template filter was added so records can be filtered to “Context: Linked in = Current page.”

NOTICE Migration Note:

- **Simplified protocols from 6.9 had no pages, so after migration, they are created as a single page simplified protocol with the migrated blocks on it. Single page protocols have a different layout in workflows with no breadcrumb chevron at the top, and in ELN, don’t have levels, but just go straight to the page for the run. In both cases, the page status converts Completed runs to Finished runs.**
- **Users with access to the Simplified Protocols module will have access to the Protocols module after migration to prevent losing access to simplified protocol**

setup. Existing simplified protocols will be migrated onto one page. If desired, more pages can be created, and the blocks can be moved around after migration.

- Plugins such as SLIMS GATE flows are translated so they retain the most sensible behavior possible. For example, a migrated flow should get the results of all the blocks on the page when fetching the results of a page, instead of returning nothing or fetching the results of a single result block on the page.

INFO

Reference:

✓ The primary changes are throughout SLIMS Administration Manual → 8. Experiments and Workflows

3.5 Default to Simplified Protocol Conversion

Simplified protocols are being improved to take the place of default protocols in the future. 7.0b introduces the ability to convert default protocols into simplified protocols while retaining all possible setup. Each protocol step converts into a page, and each step tab converts into a block on the page. Most protocols can be safely converted, but if you find the changes to be undesirable, protocols can be reverted to a known version to undo the conversion.

For example, an admin converts their protocol V2 to a simplified protocol and makes changes until they have a V3. They decide they liked the old version better, so they revert, which makes the draft the same as V2's build. When the admin versions again, they create a new V4 based on V2's default protocol with any changes they applied after they reverted.

The protocol step type determines what blocks will be created on the converted simplified protocol page. The conversions are described in a table in the Administration Manual. However, the conversion considers:

- Content type restrictions
- Content usages
- Derivation, aliquoting, mixing
- "Content to be transferred" selections
- Descriptions and notes
- Linking of expired contents, warning when linking expired contents
- Automatic and manual result creation, aggregation selections, and analyte group selections
- Steps with macros or SLIMS GATE Flows are converted to action blocks
- Executor needs to be different than previous/all steps
- Protocol status workflow selection
- SOPs, Observations, Attachments
- Conclusion steps

NOTICE

SLIMS checks for unsupported situations and warns the user when it detects that the conversion is not possible.

- ✓ Result restrictions: Sample Results is an unsupported tab and will not be in the converted simplified protocol. Dynamic Results are not supported and prevent the conversion.
- ✓ Unsupported protocol types: Only Default protocols can be converted.
- ✓ Unsupported steps: Link Location and Variant Observation are not currently supported.
- ✓ Unsupported functionalities: Steps that require an instrument, steps with flags, steps with expressions to add flags, and steps with mix rules are not yet supported.

INFO

Reference:

- ✓ SLIMS Administration Manual → 8. Experiments and Workflows → 8.5. Default Protocol Design → 8.5.6. Convert into a Simplified Protocol

3.6 Protocol Status Workflows *

Custom status workflows already existed that could control the versioning of protocols as templates for experiments. This change is for protocol run statuses, not for protocol versioning, and introduces additional statuses for protocol runs in ELN and workflows.

With this release, the concept of Protocol Status Workflows was added to replace the original approach to protocol run statuses. This change provides more statuses than before, makes status transitions more apparent, and supports review and amendment flows. Reviews and amending reviews can be tracked beyond simply starting and finishing a workflow or ELN protocol.

The Default Protocol Status Workflow exists by default in 7.x releases and comes with the statuses: Not Started, In Progress, In Review (inactive by default), Done, Amending (inactive by default), and Amending Review (inactive by default). In Review, Amending, and Amending Review can be activated with status transitions if needed. It is possible to create a new custom status workflow of type “Protocol” which automatically comes with the same blueprint. Default and custom protocol status workflows are locked so that new statuses and status transitions cannot be added, but the existing ones can be (mostly) modified. All protocol runs rely on protocol status workflows, so Not Started and Done cannot be deactivated, and run steps/pages in these statuses cannot be edited to protect the operation of protocols.

The intent is to allow administrators the flexibility of one or more review flows to dictate which statuses each kind of experiment goes through, which status transitions require sign off, and which roles can access or amend reviews for their lab. Each simplified protocol page and default protocol step now has a field to select the protocol status workflow that it follows. During a run, buttons at the bottom of ELN experiments and workflow protocol runs correspond to the selected status workflow. Instead of only having the Back, Next, or Finish buttons, there could be more depending on which user is viewing the run: a Back button, text that says In Progress, a Review button, and a Next or Finish button. On Finished pages, there may be an Amend button.

NOTICE Migration Note:

- Default protocols still have a sign off step available, but simplified protocols lose their sign off tab. Note blocks are taking the place of signature blocks for simplified protocols in ELN and workflows for SLIMS migrating into 7.0b.
- Existing signature blocks turn into note blocks to store the information they had before migration. If there were signatures, the paged protocol is forced into a "Review protocol status workflow" and migrates into an appropriate status. No e-signature options are enabled on the status transitions for that status workflow, but they can be enabled after migration. The transition from "In Progress" to "Done" is disabled so the pages must pass through "In Review" instead.
    - If the protocol run wasn't closed and the blocks were in progress, the run migrates into a page that can be put "In Review" and then can be finished.
    - If the protocol run was closed and e-signature wasn't done yet, the page has the status "In Review."
    - If the protocol run was closed and the block was signed, it migrates into a page in the "Done" status.
- The original sign-off behavior and any original sign-off SLIMS GATE flows will be listed in the upgrade report. The original field xprs_status field is supported for backwards compatibility, but new SLIMS GATE flows should instead use the new field xprs_fk_status to update the status of a run.
- The "E-signature is needed to finish or reopen the step" and "Only selected roles can execute" options were considered during migration. If both were disabled on signature steps in the prior SLIMS version, then the default protocol status workflow is sufficient in 7.0. However, if signature and/or role restrictions are required, another protocol status workflow will be created to migrate the values into. For example, they could migrate to a protocol status workflow that has e-signatures required for the logical status transitions and restricted to the defined set of user roles. SLIMS attempts to create as few new status workflows as possible during migration.

INFO Reference:

- ✓ SLIMS Administration Manual → 8. Experiments and Workflows → 8.4. Simplified Protocol Design → 8.4.1. Simplified Protocol Pages
- ✓ SLIMS Administration Manual → 8. Experiments and Workflows → 8.5. Default Protocol Design → 8.5.5. Final Electronic Signature Step
- ✓ SLIMS Administration Manual → 8. Experiments and Workflows → 8.7. Electronic Lab Notebook (Routine) → 8.7.5. Protocol Run Steps
- ✓ SLIMS Administration Manual → 9. Audit Tools and Quality Control → 9.6. Status → 9.6.1. Status Workflows

3.7 Restrict to One Session Per User with Lab Settings *

TIP A new lab setting was added: "Multiple sessions per user allowed"

A user session begins as soon as a user logs in and ends when they log out. Previously, users could login across multiple browsers or devices at a time. Now, to allow customers to choose their level of workstation security, a lab setting has been added so the user sessions can be restricted.

If “Multiple sessions per user allowed” is enabled, users can still be logged in on multiple browsers or devices at the same time. But if it’s disabled, users are logged out of a previous device or browser whenever they log into a new one. Within the active user session, users can still have multiple tabs open to SLIMS in the same browser.

The new lab setting is enabled by default, so customers won’t see a difference unless they choose to disable it.

INFO

Reference:

- ✓ SLIMS Administration Manual → 3. SLIMS GUI and Navigation → 3.9. Lab Settings → 3.9.7. Users
- ✓ SLIMS Administration Manual → 4. Access Control and Session Management → 4.1. Login and Session Management

3.8 Selection Actions Work during “Link Content to Order” Mode *

One of the ways to link content to an order is by entering link mode from the Orders module and then selecting contents to link in the Content module. Most actions were prevented during this selection in previous versions. You can now use actions while selecting content in link mode (for example, to derive contents or get an Excel output of the grid).

INFO

Reference:

- ✓ Too fine of detail for manual.

3.9 Edit Reference Data Records Inline

Given a user has the necessary permissions, it was already possible to view the details of reference data records and even add new records inline. Users can now also edit the existing reference data records without leaving the context of the field. This is useful for fixing typos or incorrect records that would previously have required access to the Reference Data module to correct.

Users need the right access in order to edit records inline.

- User rights: The “access level” on the reference data record is set to a user or group the user can update.
- Role rights: The role functionality access right “Can create Reference Data Records from within a dynamic choice custom field” is enabled.
- Field rights: The dynamic choice field allows inline creation of new records.

INFO

Reference:

✓ SLIMS Administration Manual → 5. Content Management → 5.5. Reference Data

3.10 Sorting is Possible on Function Columns *

Administrators can add columns to grid templates and can even add columns that count values or generate the mean instead of just displaying the record values. The count and mean function columns can now be sorted, whether accessing the sort action from the gear menu or view menu. The change works on standard, workflow, block, and stability grid templates.

INFO

Reference:

✓ Too fine of detail for the manual.

3.11 Restrict Macros by Category

Administrators were already able to create macros that were restricted to certain content types but can now restrict to categories also. Macros have the option “Only for selected content types” which now includes a list of categories and content types to choose from. Categories are filtered on top and then filtered alphabetically. A “level” column was added to the list to show whether an entity is a content type or a category.

The macro can run on content of the selected types or categories. When any other content is selected, the macro will be unavailable. In a default protocol, the macro can be automatically started on the start of a step. The “Skip inapplicable content types” is useful when content from both the allowed and disallowed content types and/or categories will be linked to the step, preventing the macro from being started. The setting allows the macro to run while skipping the disallowed contents.

INFO

Reference:

✓ SLIMS Administration Manual → 10. Automations → 10.9. Macros

3.12 Remove Unused Custom Statuses *

If there are no records using a status on a custom status workflow, then the status can be permanently removed. Built-in statuses for default status workflows can’t be removed even if they’re unused.

INFO

Reference:

- ✓ SLIMS Administration Manual → 9. Audit Tools and Quality Control → 9.5. Status → 9.5.3. Statuses

3.13 Entities in Store Module can be Filtered *

In the Store module, a pop-up displays all the entities that are imported with a package. Packages can be installed in full, or a selection of entities can be made to install a partial package. The headers for the grid were made filterable so it is easier to find the desired entities by name and/or table.

Typing in the Name column provides a selection that contains the entered text. Selecting one or more tables in the Table column provides a selection containing entities from the selected tables.

The main entities grid on the Import Package tab in the SLIMS Share module also is filterable now. The sensible column headers are filterable, such as name, table, and UID. This grid is populated when importing a SLIMS Share package or Json file that you want to install.

INFO

Reference:

- ✓ SLIMS Administration Manual → 10. Automations → 10.6. SLIMS Share Configuration Import Tool → 10.6.2. Import Package Tab
- ✓ SLIMS Administration Manual → 10. Automations → 10.7. SLIMS Store Configuration Import Tool

4 Enhancements in Version 7.1 Bridge

The following updates were included in SLIMS Version 7.1b. **Default updates are designated with an asterisk (*) next to the title.**

4.1 Sample Scheduler in Simplified Protocols

INFO

- ✓ New block type “Custom” comes with a Renderer option. To start with, renderer only provides “Submit Sample Scheduler” and “Complete Sample Scheduler Sequence” blocks.
- ✓ Existing default Sample Scheduler protocols can be converted into simplified Sample Scheduler protocols to gain their benefits.
- ✓ This note does not describe the full Sample Scheduler for OpenLab and OpenLab CDS integrations with SLIMS, but they are covered in other guides, such as the SLIMS Administration Manual and Installation Manual.

NOTICE

No automatic conversion will be provided for the CDS Execution Protocol in SLIMS. Therefore, it is recommended to use these new Sample Scheduler simplified protocols instead if your lab is just starting to use Sample Scheduler for OpenLab and OpenLab CDS with SLIMS.

Starting with this release, SLIMS supports Sample Scheduler for OpenLab with simplified protocols in ELN and workflows. Customers will need to configure their authentication with OpenLab Shared Services and import the CDS package from the Package Browser module. Then they will have access to Sample Scheduler Renderers under the “Custom” block type. There are two renderers to choose from:

- “Submit Sample Scheduler” creates a block that sends the samples to Sample Scheduler and runs them through an optional sequence template, so you can get back a whole sequence to any Complete blocks that follow the Submit block. The Submit block comes with a Grid Template tab of its own. The Submit block needs to be associated with a Content block to take input samples from. The grid template is used to build the mapping that translates SLIMS fields to Sample Scheduler fields. Anything linked to the Content block is also linked to the Submit block. Once the user chooses an instrument, project, and sequence template, they can create the sequence. The sequence status updates to Waiting (by default but Scheduled can be used instead) and the samples can be seen in Sample Scheduler for OpenLab.
- “Complete Sample Scheduler Sequence” creates a block that maps the sequence to the correct samples in SLIMS. An Auto Completion plugin is available that can be adapted to help automate the connection between contents and the blanks, system suitability samples, and calibration standards.

When the protocol is versioned, SLIMS validates that the order of the blocks makes sense. The order of the blocks on a given page is not important, but if they’re on different pages,

any Complete blocks should be after the Submit block. The Link Content block associated to the Submit step also cannot be on a page after the Submit block's page. This allows you to have one or more Complete blocks that separate the calibration standards, blanks, etc. on different pages if you prefer. There must be a Submit block if there is a Complete block. The Sample Scheduler blocks can be deleted because the protocol is not in itself special, but then it will no longer be used for Sample Scheduler.

INFO

Reference:

- ✓ SLIMS Administration Manual → 8. Experiments and Workflows → 8.4. Simplified Protocol Design → 8.4.3. Defining Block Types → 8.4.3.13. Custom
- ✓ SLIMS Administration Manual → 8. Experiments and Workflows → 8.6. OpenLab Software Suite Integrations (and subtopics)

4.2 Link Location Blocks Added to Simplified Protocols

Link location blocks have been added to simplified protocols which allows default protocols with link location steps to be converted to simplified protocols. This change comes with improved location UI design to visualize the wells of the plate/location and the contents inside.

Link location blocks may use a content block as input. If a content block is selected, the locations associated with contents linked to the content block are auto populated in the location block. Otherwise, the user can manually scan locations or select them from a pop-up menu to link them to the block. The locations are displayed in an expandable accordion list that shows the location details. If mistakes are made, locations can be unlinked by selecting them and using the unlink action from the selection menu. Locations can be selected to use unlink, move, and export to Excel actions from the selection menu.

It's possible to restrict the location types that can be linked to the block to narrow the scope of choices for users and prevent unwanted locations from being linked. In that case, only the allowed locations are automatically linked or scanned, and the disallowed location types are filtered out of the manual selection pop-up. If expiration/calibration is available on locations, it's also possible to prevent linking expired /uncalibrated locations and enable a warning to pop-up when the user links them. Currently, the warning only appears when the user links expired/uncalibrated locations manually.

Action blocks allow macros or SLIMS GATE flows to run on one or more records during the simplified protocol run. This depends on the macro settings "requires manual input selection" and "run on start of step." If manual input selection is required, one or more contents can be selected in multiple linked locations at a time by using the Ctrl or Shift key (Command or Shift on Mac, respectively) and clicking them. An arrow shows the direction in which the contents were clicked. The action button runs the macro or flow only on the selection of contents. Actions that run on the start of step or that do not require manual input selection are run on all contents in the linked locations, not just those linked to the run.

Default protocols with link location steps can be converted to simplified protocols with link location blocks. A link location step that has a macro defined will be converted to a page with the former name of the step. The converted simplified protocol page has a link content block, an action block with the macro and the link content block selected as the input block, and a link location block.

If the link location step had “Automatically link locations of linked content on start of step” enabled, then the link location block is associated with the link content block to preserve that behavior. The protocol can be reverted to a previous version to convert it back into a default protocol.

NOTICE

There is a limit on the size of the location details. Positionless locations and locations bigger than 100 rows x 100 columns are too big to display effectively so the location details are not displayed.

INFO

Reference:

- ✓ SLIMS Administration Manual → 8. Experiments and Workflows → 8.4. Simplified Protocol Design → 8.4.3. Defining Block Types → 8.4.3.4. Link Location
- ✓ SLIMS Administration Manual → 8. Experiments and Workflows → 8.5. Default Protocol Design → 8.5.6. Convert into a Simplified Protocol
- ✓ SLIMS Administration Manual → 8. Experiments and Workflows → 8.7. Electronic Lab Notebook (Routine) → 8.7.3. Simplified ELN → 8.7.3.1. Blocks

4.3 Location Detail and Plate Visualization *

The digital representation of plates has been improved with new functionality and Agilent UI design. It is easier for users to see what’s in a plate on the location details tab.

The grid display of the location detail can handle plate wells of common sizes such as 8x12 and 16x24. Existing functionality is retained so it is still possible to block/unblock wells, view the position history, create content from an empty position, and more. The actions available from the right-click menu and selection menu are retained and are detailed in the Administration Manual. "Selection Mode" was added so users can choose how positions are selected when they drag over the location detail grid: in a rectangle, by column, or by row.

Content positions can be changed in the UI by clicking “Rearrange location” to enter rearrangement mode. Users make all the desired changes to positioning, then click confirm to commit (or cancel to revert) the changes. In rearrangement mode, users drag and drop the contents into an empty well or onto another content to switch their positions in the location/plate.

NOTICE

- There is a limit on the size of the location details. Positionless locations and locations bigger than 100 rows x 100 columns are too big to display effectively so the location details are not displayed.
- This feature is still in development in 7.1b, so the content icon and two lines of custom information is not yet visible on a well.

INFO

Reference:

- ✓ SLIMS Administration Manual → 5. Content Management → 5.2. Contents → 5.2.1. Content Module → 5.2.1.1. Content Module Tabs

4.4 SLIMS Share Improvements *

The SLIMS Share module has been redesigned to give super users more insight into what goes into the configuration package after it has been validated for their production SLIMS. The Create Package screen was decluttered to only show a grid of all main entities in a SLIMS instance, and a grid of everything that was selected to be part of the final package.

Instead of creating many packages over time, there will only be one validated "General" package. As main entities are tested, they are moved over to be included, and they stay persistent in the "Main Entities in Package" grid. This is the package intended to synchronize the validation SLIMS instance with the production SLIMS instance.

With this change, super users don't have to predict all the dependencies because SLIMS checks everything in the package when the user clicks an export action. If there are any issues with the package, a pop-up points out the issues. Without leaving the context of the pop-up window, the super user can choose whether to include dependencies for the main entities in question or to exclude the main entities.

"Ad Hoc" packages are a new package type that can be created to fulfill the role of picking smaller changes for one-time packages. They allow the super user to cherry-pick across all main entities (except for individual protocols from a workflow). SLIMS can't check for dependencies in ad hoc package main entities, so it's important to keep in mind what dependencies exist and whether your target instance has what it needs for the entities in the package. If a package cannot be imported on the target SLIMS, an error message is provided; however, the error cannot provide as much help as it can for the General package.

The "Workflow Protocol" package still exists to create packages that include exactly one workflow protocol, but they are not intended for customer use. Whole workflows can instead be included in the General package.

NOTICE

- ✓ **In SLIMS 7.1b, a partial import cannot be performed. If the super user imports from a file and checks only some of the main entities, all the main entities in the file will be imported. Partial import is reinstated in 7.2.**
- ✓ **Access rights on configuration parts should be verified after SLIMS Share transfer. Users may exist in the source and target SLIMS instances, but they do not exist in SLIMS Share; therefore, the user access right field will always be cleared and set to public. Group access rights may transfer if the group exists in both source and target instances, or if the group is included in the imported package. If not, it will also be cleared and set to public.**

INFO

Reference:

- ✓ SLIMS Administration Manual → 10. Automations → 10.6. SLIMS Share Configuration Import Tool (and subtopics)

4.5 Aggregated Results Behavior for Rejected Results *

Aggregation methods such as maximum, mean, and standard deviation can be used to calculate a value for multiple results. However, SLIMS took rejected results into consideration for the calculation. Starting in 7.1b, that will no longer be the case, as it is preferable to only have a calculated value for validated results and exclude rejected results from the calculation.

This change also impacts product specification rules which can be built to require a minimum number of results. Product specifications now also exclude rejected results from the result count, so they won't conflict with aggregated results.

INFO

Reference:

- ✓ SLIMS Administration Manual → 8. Experiments and Workflows → 8.5. Default Protocol Design → 8.5.2. Protocol Steps
- ✓ SLIMS Administration Manual → 8. Experiments and Workflows → 8.7. Electronic Lab Notebook (Routine) → 8.7.6. Results
- ✓ SLIMS Administration Manual → 8. Experiments and Workflows → 8.14. Products and Specifications → 8.14.2. Specifications Tab

4.6 Macro UI Refactoring *

The pop-up macro execution screen users see when executing macros was redesigned to focus on the most useful information. The Scan module was updated along with this change to gain the same benefits.

Most of the changes were aesthetic:

- Color of side bar, background, and progress bar
- Icon per step type was added
- Information is styled in boxes
- The position of buttons was optimized
- The log was removed and info about the step was added in feedback
- "Show Feedback" panel is opened when the user corrects an error.

INFO

Reference:

- ✓ SLIMS Administration Manual → 10. Automations → 10.8. Scan

4.7 Hide Irrelevant Action Buttons *

Actions like Derive, Aliquot, Mix, etc. are available on records in right-click and selection menus in many modules across SLIMS. They are now filtered out depending on the record that is selected so users cannot click on an action that cannot be used on the selected record. This change applies to the Content module and protocol runs in Workflows and ELN.

For example, a content whose content type does not have derive enabled will no longer have the derive action available. If multiple records are selected, only the intersecting actions will be available. If a blood is derivable and mixable, and a solvent is mixable but not derivable, then when both are selected, only the Mix action will be available.

INFO

Reference:

- ✓ SLIMS Administration Manual → 3. SLIMS GUI and Navigation → 3.2. Grids → 3.2.2. Grid Actions → 3.2.2.4. Multi-Selection

4.8 Restrict Category in Simplified Protocols

It was already possible to restrict the content types that could be linked in a simplified protocol, but this has been updated so content linking can be restricted by category as well. When a category (or categories) is restricted in the protocol block setup, the content types in the categories can be selected in addition to further refine the configuration options (such as default usage amount). This change is available in simplified protocols for workflows and ELN.

When the 'Required' option is set to true for a given category on a block, then the protocol block requires at least one content within that category to be linked at runtime. If usage tracking is configured on the block and the category has an amount field applied for all its content types, then the usage options are applied on any linked contents of that category. If the category has no amount field set, then the usage options are disabled on the block configuration.

INFO

Reference:

- ✓ SLIMS Administration Manual → 8. Experiments and Workflows → 8.4. Simplified Protocol Design → 8.4.3. Defining Block Types → 8.4.3.3. Link Content

4.9 Use UID for URL Links to Records *

It was already possible to generate URL links to navigate to records in SLIMS using the GUID, but you can now access records using their UIDs without having to generate a link first. Records acquire a GUID when the "Generate link" right-click action is used, and that method still produces a GUID link. But additionally, users can paste the URL link using UIDs

of a record directly into the browser to go to the content, order, protocol run, or protocol run step.

The new format only works for the tables Content and Order, and for the Experiment Run and Experiment Run Step tables for both ELN and Workflows. Links can be copied using actions in the UI or automatically generated using emails and notifications from rules or Groovy expressions. If the user is not logged in when they click the link, they are brought to the record after logging in.

INFO

Reference:

✓ SLIMS Development Manual

4.10 UID of Users is Recorded in Expression Logs *

The unique identifier of a user who initiated an action in the user interface is now captured in the log. When a user starts a Groovy execution, SLIMS GATE flow, or SLIMS REST flow, their UID is reported in the log.

Note that customers can use custom logging configurations in which case the UIDs might not appear by default, and the configuration would require changes to allow this.

INFO

Reference:

✓ SLIMS Administration Manual → 12. Scripting and Customization → 12.2. Groovy Editor → 12.2.2. Editor

4.11 Download SLIMS GATE Remote Daemon in Customer-Hosted SLIMS *

Customers using a SLIMS on their own cloud environment need the ability to deploy SLIMS GATE remote daemons to interact with their equipment. The “Download Remote Daemon” button was added to SLIMS so users with access to the Plugins module can click it to download the files they need.

INFO

Reference:

✓ Agilent Hosted Installation Manual

4.12 Restrict Content Events

Content event types and flags can now be restricted by content types and category. This enables Administrators to filter the content events that can be added to relevant content, or to filter the flags available to add for a content. For example, the administrator can prevent the list of checks from showing up for fridge contents except for temperature.

INFO

Reference:

- ✓ SLIMS Administration Manual → 6. Studies → 6.1. Studies → 6.1.2. Content Event Types → 6.1.2.1. Standard Content Event Types
- ✓ SLIMS Administration Manual → 6. Studies → 6.1. Studies → 6.1.2. Content Event Types → 6.1.2.2. Flag Content Event Types

5 Enhancements in Version 7.2 Release

The following updates were included in the SLIMS Release Version 7.2. **Default updates are designated with an asterisk (*) next to the title.**

5.1 Bench UI for Fixed-Bench Mobile Devices

INFO

- ✓ SLIMS must be set up with protocols ready to use and contents in the queue if using workflows. All setup is done in SLIMS and Bench UI is not standalone.
- ✓ Administrator or manager needs the role functionality access right “Can assign requests” to make it easy for users to find their work.
- ✓ Users need the role functionality access rights “Electronic Lab Notes” and/or “Workflows” to create and access protocol runs in Bench UI for the respective modules.

The Bench UI is a web-based interface that can be run on tablets like the Agilent tablet to execute protocols. The user experience is simplified to work on touch-screen mobile devices to guide users through a protocol execution while they are at a bench. Bench UI makes it easier to have a fixed-bench device that processes a subset of specific protocols. Bench UI works with ELN and Workflow simplified protocol runs.

NOTICE **The Agilent tablet is recommended to use with Bench UI, but other tablet devices are supported like those running Google Chrome or Safari. It is recommended to use tablet devices 10 inches or larger for the best viewing experience.**

About the setup:

There is no setup in Bench UI, so any configuration needs to be done beforehand in SLIMS. Protocol runs and requests need to be assigned to users so they can locate their work to-do easier. ELN experiments are created with access rights to determine what roles and groups can view the protocol runs. For workflows, managers can select samples in a workflow queue and assign them to a user.

When SLIMS is set up, a URL is provided that can be used to connect to Bench UI.

Views are configured with this in mind to show users the most relevant records in each block on a protocol page. By tapping “Configure view in Bench UI,” administers can pick the columns that should be shown in pop blocks and link content blocks. Hints/tooltips can be added to custom fields on the protocol run step table to provide a pop-up hint to users when they tap on a hint icon on a run. Changes made in SLIMS are reflected instantly in Bench UI. However, SLIMS needs to be refreshed to see changes made in Bench UI. This

makes it handy for work done in a third-party software or instrument (such as measuring weight on a balance) to be automatically refreshed in Bench UI.

About the user's workflow:

Users navigate to the URL provided by their admin and log in with their regular SLIMS credentials on a mobile device. Users start on the Home page which has a filter to show all runs and queued samples or only the runs and samples assigned to them. There are two grids: a list of available protocol runs with their popped samples and a list of queued samples with their associated requests. Users can start a protocol run directly from the "Runs" grid or begin by selecting some samples. Samples can be added to the "Run Basket" from the "Not Started" grid, and when the user is ready, they choose from the applicable protocol runs to start the samples.

Users can return to Home or log out at any time. Users can search runs and samples by barcode to find them again with the search bar. Once started, the protocol run shows the run name and the first page in the protocol with all the instructions and blocks on the page. A stepper is shown at the top to show what page the user is on, and they navigate between pages with buttons at the bottom as work is completed. Tapping on protocol blocks expands and focuses them on the screen.

When the user taps the "Information" icon in a sample row in a block, a side panel opens on the right where the user can edit information, view details about the selected sample, and view usages with the taken quantities. Usages are only shown when they are applicable.

Users can only edit pages of a protocol run that are in a status that permits it—by default, the "In Progress" status—to guide them to where work needs to be done. Pages that are "Done" are read-only. This behavior works with the statuses and role permissions on the status workflow associated with your protocol design. Users can navigate with "Next" and "Back" buttons to complete or redo pages and put pages into review if your status workflow requires it. Statuses can be restricted so only lab managers can edit "In Review" pages.

Supported block types are used in the Bench UI with mobile design in mind:

- Link Content: Tapping the Pop/Link Content block automatically focuses on the scan box so the user can scan a content or location in the queue right away. The scanned record links the relevant content to the block. Users can also link contents manually via a pop-up that filters the content types to show only the relevant ones. Users cannot link contents that don't exist, and for workflow protocols, that aren't in the queue. You can also set up content restrictions to provide a warning when linking expired content, to restrict content types, or to prevent linking contents in certain statuses. Content can be unlinked by tapping the three-dot icon in its row of the Pop/Link Content block grid.
- Result: Results can be seen in a Result block with content barcode, test, analyte, and the value. The user can select a result by tapping the "Information" icon to update or enter result values in the panel that opens on the right. Several test datatypes are currently supported: text, dynamic choice, fixed choice, checkbox, quantity ("Allow choice of unit" is supported), whole number, and decimal number.
- Attachment: Attachments are shown in two view modes: as a grid of tiles with previews for supported file types, or as a table with details about the files. Users can tap the tiles to view the attachments or tap the download icons to download the files. Users tap the "Upload" button to add attachments using the device's file browsing system or drag and drop over the file upload area in the pop-up.
- Form: Users can see data in a Form block and can tap on the fields to enter data. The form block autosaves as the user fills it in. Several field datatypes are currently supported: Date*, date and time*, textbox (multiline supported), short text, rich text

(only displays as read-only), dynamic choice (single-choice and multiple values), fixed choice (multiple values supported), checkbox, quantity (“Allow choice of unit” supported), whole number, decimal number, barcode, status, and attachment.

*Date field format in Bench UI does not conform to the one in SLIMS. Based on the technology, the only date format currently possible in Bench UI is YYYY-MM-DD; however, the SLIMS date format is applied whenever the data is viewed in SLIMS.

- Note: Users can view Note blocks including any images. Note blocks cannot be edited in Bench UI.
- Grid Template: The grid is displayed in the block and any context filtering is applied to show just the desired data in the protocol run or on the protocol run page. If rows overflow, the grid is scrollable.
- Action: SLIMS GATE flows and macros can be run as Action blocks. They are displayed as buttons the user can tap on. If action blocks are configured to require selection, the buttons are disabled until the user selects some contents to run them on. If the blocks do not require selection, the user can tap the buttons without picking contents first. Currently, Bench UI only supports macros on the content table with content step types, and users cannot provide input (like on an edit step) during the macro execution. Reports are not currently supported.
- Custom – Renderer: Sample Scheduler: Bench UI supports integration with OpenLab CDS and Sample Scheduler for OpenLab. However, only the Submit Sample Scheduler block is usable for now. The user needs an OpenLab Shared Services account (just as they would in SLIMS) and can then select the project, sequence template, and instrument in the block. The projects and instruments are those available through OpenLab Shared Services, and the sequence template selection are for the templates available in OpenLab CDS for the selected project. Once the required fields are filled, the user taps a button to create the sequence. Once it’s been submitted, the user can return to the block to see a link to the sequence in Sample Scheduler for OpenLab.

  Since the operation in Bench UI is so simple for the user, it requires the configuration to be considerate of any needed automations.

INFO

Reference:

- ✓ SLIMS Administration Manual → 8. Experiments and Workflows → 8.13. Bench UI: SLIMS Interface for Tablets

5.2 Requester Portal Add-on for Lab Orders

INFO

- ✓ Requires Requester Portal add-on license and role entitlement Standard, Premium, or Basic (Requester).
- ✓ Requires role functionality access right "Can use Requester Portal" (and "Can use SLIMS" can be disabled).
- ✓ Requires the Requester Portal package from the SLIMS Store which comes with required custom fields for order types, demo report templates to use, and plugins to customize the branding of Requester Portal.

You can set up the Requester Portal, an add-on to SLIMS, to provide customers with a simple interface to create orders and request analyses from your lab. The portal is separate from SLIMS but connects to it, allowing you to control your customers' access with user accounts and SLIMS's built-in permissions. It is installed by the SLIMS engineer and a URL is provided so your customer users can access the portal without accessing SLIMS. The customer role requires "Can use Requester Portal" to be enabled for customers to login to the portal. The access right "Can use SLIMS" was added so it can be disabled for customer roles to prevent login to SLIMS.

Plugins from the SLIMS Store package can be used to customize the branding of the UI, so your customers see your logo on the login and landing pages. You can also create short FAQs using reference data and the customization plugin to provide information in the portal.

The customer logs into the portal and is authenticated to the portal's dashboard where they see the orders to which they have access. The customer can edit those orders, create new orders, add samples with sample data to the orders, and add requests to the orders to indicate what testing they would like your lab to perform. The dashboard provides tiles showing the order statuses and QR codes to scan the barcodes of orders they've created. The dashboard can show orders in a list view instead of the tile view with more detailed information. In list view, the grid columns can be rearranged or hidden.

Orders can be created, edited, or canceled from the portal. The customer can create orders of any order type. The "Create Order" form pop-up shows any fields on the order table for which you have enabled "Show in Portal" during setup. Not all field datatypes are supported*. Editing orders that have been created, such as updating their status, updates the order in SLIMS (and vice versa).

Customers can add samples to an order during order creation or afterwards and can update sample information afterwards. The "Add Samples" form pop-up shows any fields on the content table for which you have enabled "Show in Portal" during setup. Not all field datatypes are supported*. For example, your sample form might have sample type and numerical fields for starting volume/mass. Alternatively, samples can be added by enabling imports on the order type being used for the portal. The grid is paginated so the screen does not become cramped, displaying five samples on each page.

The "Add Requests" section of the "Add Samples" form provides the products and / or requests dropdown with a selection of products or requestables filtered to go with the selected order type. After selecting the product/requests and submitting the selected samples and their data, the grid within the order details updates to show the samples and their specifications.

The order type for the portal can be configured to provide a shipping manifest structured by a report template so the lab's user can download it and include it for shipping contents to the lab. After the samples are all processed, the Certificate of Analysis is linked within SLIMS. A Certificate of Analysis field was added that shows the linked certificate in the order details of the portal and the Orders module in SLIMS. The order grid in the portal can be updated to show the certificate of analysis column as well. If an order has a Certificate of Analysis attached, a preview of the file replaces the QR code in the order's tile in the portal dashboard.

**Supported field datatypes: Date, date and time, textbox (multiline supported), short text, rich text (only displays as read-only), dynamic choice (single-choice and multiple values), fixed choice (multiple values supported), checkbox, quantity ("Allow choice of unit" is supported), whole number, decimal number, barcode, status, and attachment.*

INFO

Reference:

- ✓ SLIMS Administration Manual → 4. Access Control and Session Management → 4.3. Access Rights → 4.3.3. Roles
- ✓ SLIMS Administration Manual → 8. Experiments and Workflows → 8.12. Requester Portal Configuration and Preparation
- ✓ SLIMS Administration Manual → 8. Experiments and Workflows → 8.13. Requester Portal Customer Use
- ✓ SLIMS Administration Manual → 10. Automations → 10.2. Fields → 10.2.3. Custom Fields
- ✓ SLIMS Administration Manual → 10. Automations → 10.4. Plugins → 10.4.3. Types of Plugins

5.3 Scan to Run Macros in Simplified Protocols

INFO

- ✓ The new option appears only for macros in Action blocks.

Macros that need an input to perform an action, like derive and aliquot macros, now have three ways to choose the input. The macros are built as normal in the Macros module. When the macro is selected in the simplified protocol design (Protocols module for ELN, Workflow Management for Workflows), there are three choices for "Input Selection."

- All: The macro takes everything from the input block. If there is no input block, the user is prompted to choose one.
- Selected Samples: The macro button is inactive until the user selects records from the block they want to use as input. Then, the button activates and can be clicked.
- Scan: The macro provides a scan box and the user can scan or type in barcodes. The macro runs on the entered barcodes.

SLIMS considers where the samples are coming from and whether they are already in the protocol or not. If the barcodes match samples already in the input block, they are processed, and the output is linked to the output block. If the samples aren't in the input block, they are linked to the input block, processed, and the output is linked to the output block. If the protocol run is in a Workflow and the samples aren't in the queue, the macro fails and provides an error.

INFO

Reference:

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

5.4 Macro can Position Content in Link Location Block

An option was added for macro steps that can place records in a location starting from the position selected by the user. In the macro step configuration, the "Use output location block" option was added to the ones that existed: "Specify new location," "Keep at current location," and "Use previously created location." This allows the administrator to specify that a location block serving as output will be filled by the records processed by the macro. When combined with "Use output location block," the fill method (by row or column) determines the order the macro starts filling from the position selected by the user.

A simplified protocol is set up with an action block. The action block requires an input or some kind and an output block. The input is the source of the content to be processed, such as a Link Content block, a Link Location block already filled with content, or an Action block with a macro step that creates content. The Action block itself is configured with the macro that 'uses an output location block.' The output for the Action block needs to be a Link Location block.

When the user selects a position (well) in the location block and runs the macro, it processes the input and fills the link location block starting from the position they selected. The location is filled column or row wise as the macro was designed.

INFO

Reference:

- ✓ SLIMS Administration Manual → 8. Experiments and Workflows → 8.4. Simplified Protocol Design → 8.4.3. Defining Block Types → 8.4.3.6. Action
- ✓ SLIMS Administration Manual → 8. Experiments and Workflows → 8.8. Electronic Lab Notebook (Routine) → 8.8.3. Simplified ELN → 8.8.3.1. Blocks
- ✓ SLIMS Administration Manual → 10. Automations → 10.9. Macros → 10.9.4. Action Configuration Options

5.5 Support Integrations with Sample Sheet Rows

Grid Templates have been expanded to support sequence template functions for equipment beyond the OpenLab ecosystem. These changes introduce a broader sample

sheet concept to allow integration with various Agilent MassHunter or other systems. MassHunter comes with different needs which include samples that associate with content, lines in a sequence that should not be associated with content and may be left blank, and meta-attributes that come from the content and other attributes that don't come from content. The goal is to make a sequence and implement sequence templates generic enough to be used for any MassHunter.

This implementation requires grid templates to create a sequence template with lines, custom fields to map the data and pull it into the sequence, and simplified protocols with blocks that use the grid template as a sequence. Plugins may be used to automatically fill data that doesn't come from contents and integrate with MassHunter to export and import data.

- Custom fields that will be used to collect data for the sequence should be on the "sample sheet row" table. They can then be used in the grid template.
- A grid template of type "block" that serves as the sequence template should be created on the "sample sheet row" table. This allows it to make use of the custom fields on sample sheet row.
    - Grid Templates created on the sample sheet row table come with a partial meta tree to make it faster to build:

      Samplesheetrow: Sample Sheet Row (Root)

      ∟Protocolrunstepcontent: To Protocol Run Step Content

      ∟Content: To Content

      ∟Protocolrunstep: To Protocol Run Step
    - Values can be stored on the content and stored on the sample sheet row as well (for example, used for empty rows not associated to any content.)
    - When sample sheet row columns do not use a function, they get an additional option to identify the "Source metatree" and "Source field" to copy values from. For example, you might use "Source metatree: Content" and "Source field: cntn_cf_amount" on a sample sheet row "Amount" column. At runtime, the amount value on the content will be copied into the sample sheet row amount value. For this to work, the source and target fields must have the same datatype (and other attributes like dimension).
    - It is suggested to have at least the columns "Seqno" from sample sheet row and "Content barcode" from content, though more can be added.
- A simplified protocol should be created in Protocols (to use for ELN) or in Workflow Management (to use in Workflows). A Grid Template block needs to be added to the protocol page with the new grid template selected. A Link Content block needs to exist earlier in the protocol (before it on the page, or on a preceding page) and associated to the Grid Template block.
- Optionally, rules can be created on the sample sheet row to provide evaluations on values in grids that use custom fields on that table.

At runtime, the same number of sample sheet rows are created in the Grid Template block as there are contents linked to the Link Content block before it. Fields from content are brought over via the grid template's source mapping, and fields on the sample sheet row are useful for sequence lines that aren't backed by content in SLIMS. The "Recalculate" action lets the block pick up any changed values. Empty rows can be added with a button click. The sequence lines can be edited, removed, and reordered as needed with arrow buttons or by using the Seqno of each row. The protocol page status applies, so if the page

is not in an editable status (ex: Done), the rows cannot be recalculated, edited, removed, or reordered.

INFO

Reference:

✓ SLIMS Administration Manual → 8. Experiments and Workflows → 8.7. Agilent MassHunter Integrations (new topic)

5.6 Aggregated Results Evaluate After Contributing Results *

Aggregated results were evaluated, and their status was updated for each result value as they were filled. Now, the status updates from Pending to Available when the first contributing result value is filled but stays in that status until all the expected results contributing to the aggregated result are filled. When the last is filled, the aggregated result is recalculated and the status updates to Verified if no rules are triggered or stays at Available and provides the rule errors. This change ensures the user only needs to electronically sign one time per aggregated result, and not each time a result that contributes to the aggregated result is entered.

INFO

Reference:

✓ SLIMS Administration Manual → 3. SLIMS GUI and Navigation → 3.9. Lab Settings → 3.9.12. Electronic Signature

5.7 Default Choice of Unit in Simplified Protocols *

INFO

✓ Applies to quantity fields with “Choice of unit” enabled.

Link Content blocks in simplified protocols can track the usage of reagents and any other content with a quantity field. These blocks now support "Choice of unit" which may be enabled on a content's quantity field. A quantity custom field requires a unit but may optionally have "Choice of unit" enabled and may be further restricted to a range of units using "Upper unit" and "Lower unit." The Link Content block will need "Track Usages" or "Track usages and restrict content types" to take advantage of the quantity fields, and then the default quantity and unit can be set. The unit choice is filtered by the dimension of the quantity field (ex: mass, volume, etc.), and if Upper/Lower units have been set (ex: liter, ml, µl, etc.), only the units within the range are available as a choice.

The unit chosen in the protocol design is the unit used at runtime. Since the unit is predefined for the user, they can't change it during the protocol run.

Migration Note: Simplified Protocols stick to the quantity and unit that are already selected. Administrators can choose units for fields with "Choice of unit" in the setup, but it requires versioning a protocol to go into effect on new runs. Users can't change the unit during the protocol run and won't see a difference unless a new unit is set in the protocol design.

INFO

Reference:

- ✓ SLIMS Administration Manual → 8. Experiments and Workflows → 8.4. Simplified Protocol Design → 8.4.3. Defining Block Types → 8.4.3.3. Link Content

5.8 SLIMS Share Import Visibility *

INFO

- ✓ Requires that the Lab Setting SLIMS Share Server and Password match the SLIMS Store server URL and password to install Store packages.

SLIMS Share's visibility was improved to show what goes into a validation package and how it will change the SLIMS instance it is imported on. SLIMS Store was also linked to Share so users gain the same benefits when installing a Store package.

When exporting the main package, the pop-up added in version 7.1b shows the missing dependencies. The pop-up was improved in 7.2 with two tabs to show the required dependencies that would prevent the package from functioning, and the optional dependencies that wouldn't stop the package from functioning but would prevent the configuration from being identical. Super users can choose from the pop-up whether to exclude main entities or include their dependencies.

SLIMS Share's import tab was improved to make it easier to validate a package that you wish to import on a SLIMS instance. The tab shows the available packages from the SLIMS Store if the right information was provided in the Lab Settings "SLIMS Share Server" and "SLIMS Share Password." Store packages for versions 7.0b through the current version are available, but there are installation restrictions for compatibility with the SLIMS version. A 6.9 package cannot be installed on a SLIMS 7.x version. Additionally, packages for earlier versions can be installed on a SLIMS, but packages for later versions cannot. As an example, packages for 7.0b and 7.1b can be installed on a SLIMS 7.1b, but a 7.2 package cannot.

The import tab shows all the main entities in a selected Store package or file that you import. They are displayed in expandable trees (one branch deep) to group dependencies by their main entities and make the contents of a package easier to read. The grid also shows whether the entities will be new to the SLIMS on which you perform the import, or whether they will be an update to an existing entity. If they will update an entity, the entity can be selected to show details about the change so you can review the old value and new value. When a super user installs a package from the SLIMS Store, they will be taken automatically to the SLIMS Share import tab to take advantage of the details for review, and to select the entities they want to install in the package.

To ensure super users get the most beneficial information for review, it is no longer possible to install packages from the SLIMS Store without entering the Store server URL and password in Lab Settings.

Options that were only usable to the SLIMS Store team are now hidden from administrators in SLIMS Share to make it easier to use.

INFO

Reference:

- ✓ SLIMS Administration Manual → 3. SLIMS GUI and Navigation → 3.9. Lab Settings → 3.9.10. SLIMS Share
- ✓ SLIMS Administration Manual → 10. Automations → 10.6. SLIMS Share Configuration Import Tool
- ✓ SLIMS Administration Manual → 10. Automations → 10.7. SLIMS Store Configuration Import Tool

5.9 Rule Expansion to Enable Data Edition *

To centralize where calculation and validation of values is managed, rules have been expanded so they can take on the duties of field Groovies. This enhancement makes it possible for conditional value expression rules to modify data for records on which they are triggered so it will be less necessary to create value expressions across several SLIMS modules. Conditional value expression rules are created to set a behavior that applies when the rule's conditions are met, such as scripting a warning that notifies a user when something happens to a content. Now, they will also be able to change data values* when a record is added, updated, or removed.

*Custom field values and a subset of default field values can be changed. A list of possible default fields that can be used is included in the Administration Manual.

This enhancement provides several benefits. The "Description" field was added to conditional value expression rules so engineers can describe their purpose. A dropdown is also provided so administrators can limit the record types that the rule applies to. The record types available in the dropdown are based on the table the rule is for. As an example, rules on the Content table allow restriction to Content Types and/or Categories, rules on the Order table allow restriction to Order Types, the Content Events table allows restriction to Content Event Types, and so on.

There are additional options to set the order in which rules fire, what the trigger is, and the timing (or purpose for which the rule runs).

- Rules have a sequence number to dictate the order in which they run so the administrator can choose. For example, an administrator might want a rule to be triggered that copies content and only after that have a second rule trigger that sets a value on the copied content. If rules have the same sequence number, the one with the lowest pk runs first.
- Administrators can set the action that precedes the rule execution: when records are created, updated, and/or removed. For example, if "Execute rule on creation of records" is enabled but "on update" is disabled, the rule is executed when a content is added but not when it is edited.
- Lastly, the Timing can be chosen to align the rule with a phase. Generally, one phase is updating data and doing calculations, and the next phase is validating that the values you got were sensible. Set the timing to make the most sensible methods available according to the functions you want to use and what the rule should do. The “Set Value” timing means the rule will be written to change values or do work such as performing a calculation based on another field. The “Validation” timing means the rule provides warnings, errors, email notifications, delayed trigger, etc. based on what it

checks. If “Set values” is selected for timing, then the `setValue` method can be used in the script and the notifying methods (warn, error, etc.) cannot be used. If “Validation” is selected, then `setValue` cannot be used. Rules that `setValue` overwrite values supplied by users.

In cases when a macro with an edit content step is run, a rule could be triggered. The rule can take different actions if it’s running from a macro, so the rule can be made aware of the macro execution using `macroExecutionDetails`. It is only suggested on conditional value expression rules on the Content table.

NOTICE

Migration Note: When migrating to version 7.2, all rules are given the sequence number of “0,” so since the sequence number could be the same, the one with the lowest pk will run first (they will run in order of lowest to highest pk).

INFO

Reference:

- ✓ SLIMS Administration Manual → 10. Automations → 10.1. Rules → 10.1.1. Conditional Value Expression

5.10 Easier Health Checks for Super Users and Support

The Spring Boot technology enhances the SLIMS Rest API to supply information. Customer super users and Agilent support users can view the Spring Boot Actuator pages to look at an instance’s memory usage, access logs, and alter log levels at runtime. Login is required to access the pages to protect the instance’s health monitoring information.

INFO

Reference:

- ✓ Too technical for the Administration Manual

5.11 Utilities for Dashboards *

Some quality-of-life changes were made for Dashboards. Previously, create macros were the only kind that could be used in a dashboard, but other macros can now be used. Create macros still appear as a simple button to click, but macros of other kinds require user input, so they show up as a tile with an input field. The user scans a barcode, and the macro executes on the scanned record. Errors are provided if the scanned information doesn't match any barcodes or matches multiple barcodes.

Additionally, grid templates created on the Order table have a new column called "Link to Order" that can be added to the grid. This provides a link icon in the grid column when it's used in a dashboard so the user can click it to go to the relevant order.

Finally, graph and rich text widgets can be copied to make designing them faster. If a similar graph or rich text widget is needed, copy the widget, and supply a new name. The meta tree and columns or the text in the rich text box (if there is any) will be copied into a new widget.

INFO

Reference:

- ✓ SLIMS Administration Manual → 3. SLIMS GUI and Navigation → 3.12. Dashboard → 3.12.2. Grid Widgets
- ✓ SLIMS Administration Manual → 3. SLIMS GUI and Navigation → 3.12. Dashboard → 3.12.3. Graph Widgets
- ✓ SLIMS Administration Manual → 3. SLIMS GUI and Navigation → 3.12. Dashboard → 3.12.4. Rich Text Widgets
- ✓ SLIMS Administration Manual → 3. SLIMS GUI and Navigation → 3.12. Dashboard → 3.12.7. Macro Widgets

5.12 Utility for Grids Module *

INFO

- ✓ Requires role functionality access right “Submit Content to Workflows.”

Users now get the “To Workflow” action available in the Grids module. For the action to be available, the root table of the grid template meta tree must be “Content” and the user’s role needs to have “Submit Content to Workflows” enabled. They will then be able to select one or more content and use To Workflow from the Grids module.

INFO

Reference:

- ✓ SLIMS Administration Manual → 9. Audit Tools and Quality Control → 9.12. Grids

5.13 Views Are Available for Attachment Blocks *

The ability to use views has been added to Attachment blocks in simplified protocols. Attachments are commonly uploaded to protocol runs in these blocks by users or lab equipment. It’s now possible for users to style the blocks with views the same way as they do Link Content and Result blocks. Creating views for an Attachment block in the first run of a protocol will assign the views to the blocks and on subsequent executions of the protocol, SLIMS ‘learns’ which views to assign to future iterations of the Attachment blocks.

INFO

Reference:

- ✓ SLIMS Administration Manual → 8. Experiments and Workflows → 8.7. Electronic Lab Notebook (Routine) → 8.7.3. Simplified ELN → 8.7.3.1. Blocks
- ✓ SLIMS Administration Manual → 8. Experiments and Workflows → 8.7. Electronic Lab Notebook (Routine) → 8.7.3. Simplified ELN → 8.7.3.3. Using Views in Simplified ELN Blocks

5.14 Dynamic Choice Field Enhancements

INFO

- ✓ Requires role functionality access right “Requestable Groups.”

Dynamic choice custom fields create a field with a selection list of records from the target value table. The Requestable Groups table can now be used as the target value table. Additionally, when using “barcode” as the display field for a value table like orders, content, or instruments, scanning now works for the field. This lets users scan the barcode of the chosen record, like scanning order barcodes to associate them to content or scanning an instrument into a protocol run. When using “unique identifier” as the display field for a table like reference data records, the unique identifier of records can be entered into the dynamic choice field to select them.

INFO

Reference:

- ✓ SLIMS Administration Manual → 10. Automations → 10.2. Fields → 10.2.4. Datatypes → 10.2.4.13. Dynamic Choice

5.15 Sort by Name Instead of Version *

Records that are versioned show up in selection lists in various modules. In these lists, records were sorted by version, but they are now sorted alphabetically by name and then by version number. This makes them easier to find in a long list of records. The change impacts these modules:

- Orders: Products in Specifications to be Checked dropdown
- Stability Studies: Stability Study Templates
- Anywhere reports are generated: Report Templates in Report dropdown

INFO

Reference:

✓ Too fine of detail for the Administration Manual

5.16 Improved Awareness of Field Edition and Page Status*

Fields and page status could conflict if a field in a simplified protocol run was edited before the page changed status to one that permitted edition. For example, a rule was not able to update a custom field via calculation when the user clicked the “Next” button on the page while the page was still Not Started, even though the user expected the status to become In Progress and permit the update. This behavior was improved to look at the To and From page statuses and better support the intention of the status workflow. The change also applies to REST calls that update content if the content has a custom field pointing to a status workflow (for example, when the call concurrently changes the data and status at the same time).

Now, the fields can be edited if any of these are true:

- The To status permits edition
- The From status permits edition
- The To and From statuses both permit edition

The field cannot be edited if neither the To nor From statuses permit edition.

INFO

Reference:

✓ SLIMS Administration Manual → 8. Experiments and Workflows → 8.8. Electronic Lab Notebook (Routine) → 8.8.3. Simplified ELN → 8.8.3.1. Blocks

5.17 Better Context for Starting Experiments and Runs

The “Default simple protocol” lab setting could be enabled to have “simplified protocol” preselected when the user opened the protocol creation form in the ELN. The lab setting was removed because this is now the default behavior.

The UI was changed to make it more clear to users what they are starting in the ELN. Before, there was one button to create experiments and runs, and it depended on where the user was in the context of the ELN. Now, there are two respective buttons for experiments and runs. Only the relevant button shows up depending on whether runs or experiments can be started in the user’s context.

INFO

Reference:

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

5.18 Highlight a Whole Row in Views

It is possible to highlight important information in a grid using Views. When creating an Advanced Highlight, it was possible to change the appearance (text or background color, etc.) of selected fields, and the field selection was a required setting. The field selection is no longer required so the dropdown can be left empty. If empty, the advanced highlight will apply to the whole row in the grid instead of just the selected field(s).

INFO

Reference:

- ✓ SLIMS Administration Manual → 3. SLIMS GUI and Navigation → 3.3. Views → 3.3.1. Create Customized View

5.19 Faster Date Difference Field Calculation *

The calculations behind date difference fields are considerably faster now. However, as a side effect, the filtering behavior does not match the displayed data in some cases. If exact filtering is required, consider filtering on the date fields that are the basis for the date difference field instead.

INFO

Reference:

- ✓ Too fine of detail.

5.20 Location Detail Display Field *

The Display Field was temporarily unavailable in the Location Detail tab while the feature was being developed in 7.1b, but two lines of custom information can be selected once again. The chosen lines are shown in the location wells of the Content module and in the Link Location blocks of ELN and Workflow simplified protocol runs.

INFO

Reference:

✓ SLIMS Administration Manual → 5. Content Management → 5.2. Contents → 5.2.1. Content Module → 5.2.1.1. Content Module Tabs

6 SLIMS Store Content Related to this Version

This section explores a nonexhaustive list of SLIMS Store packages related to release changes for SLIMS 7.0b – 7.2. These packages help facilitate or expand on the new features. Unless otherwise stated, these packages are available starting in version 7.2.

6.1 Standard Protocol Status Workflows Package

This package is visible for SLIMS version 7.0b and up. The Default Protocol Status Workflow contains the statuses Not Started, In Progress, Done, In Review, Amending, and Amending Review, but only has transitions activated between Not Started, In Progress, and Done. This package includes three additional protocol status workflows in the Status Workflows module, so administrators don't have to create review flows from scratch.

- Amending without review: A protocol status workflow that enables update/correction to protocol data after it has been processed. Comes with transitions between Not Started, In Progress, Done, and Amending.
- Standard Review: A protocol status workflow that enables protocol data to be reviewed after it has been processed. Comes with transitions between Not Started, In Progress, In Review, and Done.
- Amending Review: A protocol status workflow that enables review and minor corrections/update of protocol data. Comes with transitions between all the statuses.

INFO

✓ Full details are documented in the package description on the SLIMS Store.

6.2 Requester Portal Basics Package

INFO

✓ Requires Requester Portal.

The package adds options needed to make the Requester Portal usable. With installation, a "Show in portal" field is added to Products, Requestables, Order Types, and Content Types. You can enable the field on the desired entities to filter between the products, requests, order types, and content types that should be used in the Portal and the ones that should not. When "Show in portal" is enabled on an order type, other settings become available so

you can configure how Portal users add samples and select a report template to be used as a shipping manifest.

The package also contains a role preset to use only the Requester Portal so you don't have to create it from scratch, as well as reference data that may be useful, such as customer addresses, countries, and a starter FAQ. The FAQ can be used in the included "Requester Portal" plugin that's used to customize the Portal's logo and add any additional FAQ tabs.

INFO

- ✓ List of fields and other entities are documented in the package description on the SLIMS Store.
- ✓ Instructions to set up the Portal with the package included in: SLIMS Administration Manual → 8. Experiments and Workflows → 8.12. Requester Portal Configuration and Preparation

6.3 Requester Portal with Shipping Manifest Example

INFO

- ✓ Requires Requester Portal.

The package contains a basic Jasper report template that comes with some custom fields that may be useful to collect for customers shipping samples to your lab. The fields include a storage type, hazardous classifications, customer request id, and customer. They come with reference data types for customers, storage type, and hazardous classifications.

On the order type(s) you choose to use for the Requester Portal (those with "Show in portal" enabled), select the report template you want to use as the shipping manifest. The package expects the "Can add samples in portal" field to be enabled. If it is, the report will include the samples and their details when the customer generates it. If "Can add samples in portal" is disabled, there will be empty lines so the customer can manually fill the sample information in the report. The customer will be able to generate a shipping manifest in the Portal and download it to send with their shipment to the lab.

INFO

- ✓ List of fields is documented in the package description on the SLIMS Store.
- ✓ Instructions to set up the Portal included in: SLIMS Administration Manual → 8. Experiments and Workflows → 8.12. Requester Portal Configuration and Preparation

6.4 Sample Sheet Interaction Plugins Package

INFO

- ✓ Requires SLIMS GATE to run plugins
- ✓ Requires a sample sheet row grid template block in a workflow or ELN simplified protocol

The default behavior for sample sheets allows users to associate contents to samples and to create empty rows one row at a time. This package contains plugins, custom fields, and a sample sheet grid template that let the user associate content in multiple rows instead of one by one.

The first plugin makes it possible to add multiple empty rows with one button click. The rows are added in predetermined locations dependent on the content in the grid template block's input block and may even fill values in chosen fields of the sample sheet grid block. The user can also select empty rows and link content from a block other than the grid block's input block so they can link non-sample records that might need results linked after the instrument has been run.

The second plugin can create a sample sheet in .csv format to import into an instrument's software.

INFO

- ✓ Full details are documented in the package description on the SLIMS Store.

6.5 General Updates

All packages have been updated after 7.0 to use location types and instrument types as content types.

6.6 iLab Package Functionality Has Changed

The iLab integration is supported through customization instead of through a SLIMS Store package. This change allows us to support your integration with iLab better than the previous package permitted. Reach out to your Agilent representative for help with iLab integration in SLIMS 7.2 and up.

7 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 SLIMS 7.0b – 7.2.

7.1 Remove Location Type and Instrument Type Modules

As described in the related enhancement notes, these two modules are obsoleted with SLIMS 7 and the features needed to manage locations and instruments are now supported with content types. The role functionality access rights that provided access to the modules are removed as well.

7.2 Default Field Removals

Some changes came with the protocol status workflow enhancement. The cntn_status field was replaced with the cntn_fk_status field and the “Use deprecated content status field” lab setting that existed in SLIMS 6.9 was removed. The “Enable status ‘Labeled’” lab setting was removed as well because the status has been governed by status workflows for several releases.

SLIMS’ that are migrated from 6.9 to 7.0b or higher still have views, macros, SLIMS GATE flows, and plugins correctly applied as the replaced status field is adapted during migration. For Groovies, SLIMS scans the expressions to find the ones that need to be adapted and reports them in the upgrade report with the message: “This groovy uses cntn_status. This will need to be converted to cntn_fk_status.” Hard-coded statuses in Python scripts will need to be adapted as well and are not included in the report.

Field name:

- cntn_status is replaced with cntn_fk_status.

The legacy default fields for unit and quantity were removed from contents and content types because they are no longer used. They were used to input a static quantity and unit on contents, but the fields could not be used for calculations. Instead, customers have been creating datatype quantity fields that can be calculated and have better usability.

When upgrading to a SLIMS 7 version, the default unit and quantity fields are removed if they were never used. Otherwise, they are converted and kept as flat custom fields. The custom fields replacing the default fields will have the same names.

Field names:

- cntn_unit, cntp_defaultUnit
- cntn_quantity, cntp_defaultQuantity

INFO

Reference:

✓ SLIMS Administration Manual → 3. SLIMS GUI and Navigation → 3.9. Lab Settings → 3.9.2. SLIMS

7.3 "Link Order" and "Link Partial Order" Protocol Steps

The protocol step type "Link Order" and the action "Link Partial Order" have been removed because they were not used.

7.4 Move Location Store Package is Deprecated

The Move Locations package in the SLIMS Store is deprecated from SLIMS v7.x onward. The core abilities of SLIMS can accomplish what the package delivered, so it is no longer needed.

7.5 Galaxy and GenePattern Protocols are Removed

The Galaxy and GenePattern protocols for NGS workflows were deprecated in 6.9 and have been removed in 7.0. With this change:

- Dao, query, and model classes for Galaxy and GenePattern integration are removed in 7.2.
- The related database tables are kept for existing customers to retain historic data, but they are not created for new instances.

8 Appendix 1: SLIMS License Information

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

The following table lists the modules by the groups that are found in the SLIMS 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 SLIMS 7.x 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: SLIMS 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: SLIMS License Information

Table 1 Product Modules and License Type

	Units			○	○	
	Sequences			○	○	
	SLIMS Store			○	○	
	SLIMS 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 UI	A module included with SLIMS		○		○	

Version History

Version	Changes	Date
7.0b	Enhancements in 7.0 Bridge were added. Installation and upgrade changes, and deprecated features were added.	03/OCT/2023
7.1b	Enhancements in 7.1 Bridge were added. Explanation of bridge versus regular releases was updated. Installation and upgrade changes were added. License table was updated.	08/MAY/2024
7.2	Enhancements in 7.2 Release were added. Customization changes were added. Added additional license table.	07/JAN/2025

In This Book

This document provides a cumulative listing of the major feature modifications made in the SLIMS 7.2 release of the Agilent SLIMS Software.

www.agilent.com

DE003272

5994-8082EN

Edition 01/2025

Agilent