Title:	CLUWE Web Tool Design Specifications

Document Number:	ITLIB-CSV-005862-CLUWE Web Tool Design Specifications

Version: 9	Reviewers – Date (YYYY-MM-DD) and Time Approved. ITL Reviewer - Sheppard Christopher - 2017-07-14T10:27:02 ITL Reviewer - Stokes Mike - 2017-07-14T10:39:38
Status (at print): Effective	
Effective Date: 2017-08-13T00:04:20	Approvers – Date (YYYY-MM-DD) and Time Approved. ITL Approver - Rees John Jason - 2017-07-14T11:00:32

If you are using a printed copy of this document, check that the version number is consistent with the current version number located in the Regulus system. This page does not display more than 25 signatures or signatures obtained outside the system. Refer to local standard operating procedures for additional information.

TITLE: CLUWE WEB TOOL DESIGN SPECIFICATIONS

Training Requirement

Place an "X" in the appropriate box.

New Procedure

Training Required Yes □ No □

Revised Procedure

Retraining Required Yes ☒ No □

REVIEWERS

Your electronic signature indicates that you have reviewed this document and that for your area of expertise, agree it is accurate and complete.

Information Technology (IT) Subject Matter Expert (SME)

Quality Analyst

APPROVER

Subject Matter Expert

Your electronic signature attests that

- The design specifications are technically accurate, address how the computer system requirements are met
- Meet the requirements of the Lilly Quality Practice *Requirements and Design (LQP302-16)* and are traceable to one or more requirements

TITLE: CLUWE WEB TOOL DESIGN SPECIFICATIONS

PURPOSE

This Design Specifications document provides information necessary for support teams to build and maintain the CLUWE Web Tool. It will be kept updated in order to always present the current configuration information throughout the system lifecycle.

SCOPE

The following are in scope:

- Job Scheduling
- Versioning
- Electronic Signature (eSignature)

The following are out of scope:

- M2C Application Design
- Integration Broker Application Design
- Statistical Analysis System (SAS) Grid Implementation

ACRONYMS AND DEFINITIONS

The terms and acronyms in this document are defined at their first occurrence.

DOCUMENT REVISION HISTORY

Version	Revision Date	Reason for Revision (Include CR#, if Applicable)	Revised By, Title
9	14-Jul-2017	Updated for Release 4.0, CHG1092912 The following changes were made: • Added the following tables • CLUSTER_ARTIFACT • CLUSTER_ARTIFACT_DTL Updated the content in 1.4.5 Execute a scheduled SAS job • Added the content for 1.4.8 Automated File Versioning • CLUWE_OWNER.COMPUTE_JOB, updated the column type and description for RUN_THROUGH_AVA • CLUWE_OWNER.AUTH_LOG added UPDATED_UTC • added CLUWE_OWNER.AVA_EVENT • CLUWE_OWNER.ESIG_LOG added CREATED_UTC, UPDATED_UTC • CLUWE_OWNER.FILE_LOCK_LOG added FILE_CREATED_UTC, FILE_UPDATED_UTC • CLUWE_OWNER.ESIG_MEANING added	Sara Conner, Developer, Naveen Singireddy, Developer

Version	Revision Date	Reason for Revision (Include CR#, if Applicable)	Revised By, Title
		CREATED_UTC, UPDATED_UTC • CLUWE_OWNER.JOB_CONTEXT added CREATED_UTC, UPDATED_UTC • CLUWE_OWNER.JOB_EXECUTION_LOG added UPDATED_UTC • CLUWE_OWNER.JOB_INFO_LOG added CREATED_UTC, UPDATED_UTC • CLUWE_OWNER.SDDAPPROVAL added CREATED_UTC, UPDATED_UTC • CLUWE_OWNER.SDDFILE added UPDATED_UTC • CLUWE_OWNER.SDDMETADATA added CREATED_UTC, UPDATED_UTC • CLUWE_OWNER.USER_FAV added CREATED_UTC, UPDATED_UTC • CLUWE_OWNER.VERSION_LOG added CREATED_UTC, UPDATED_UTC • CLUWE_OWNER.RELOCK_LOG added CREATED_UTC, UPDATED_UTC • CLUWE_OWNER.JREVIEW_LOG added CREATED_UTC, UPDATED_UTC • CLUWE_OWNER.CAMEL_MESSAGETRACED added CREATED_UTC, UPDATED_UTC • CLUWE_OWNER.COMPUTE_JOB added CREATED_UTC, UPDATED_UTC, RUN_THROUGH_AVA • CLUWE_OWNER.COMPUTE_JOB_STATUS added CREATED_UTC, UPDATED_UTC • CLUWE_OWNER.COMPUTE_JOB_EVENT added CREATED_UTC, UPDATED_UTC • CLUWE_OWNER.COMPUTE_JOB_STATUS added CREATED_UTC, UPDATED_UTC • CLUWE_OWNER.SAS_JOB_CONTEXT added CREATED_UTC, UPDATED_UTC • CLUWE_OWNER.SAS_JOB_DTLS added CREATED_UTC, UPDATED_UTC • CLUWE_OWNER.SESSION_LOG added UPDATED_UTC • CLUWE_OWNER.USER_PRED added UPDATED_UTC	
8	14-Jun-2017	Updated for Release 3.0, CHG1074979 The following changes were made • Updated the entity and repository classes in database design section. • CLUWE_OWNER.AUTH_LOG : Added column SID • CLUWE_OWNER.ESIG_LOG : Added columns	Nikhil Das Nomula, Developer

Version	Revision Date	Reason for Revision (Include CR#, if Applicable)	Revised By, Title
		COMPLETE_TIMESTAMP_UTC, STATUS_MESSAGE, CLUSTER_ID. • CLUWE_OWNER.FILELOCK _LOG : Added columns STATUS_MESSAGE, CLUSTER_ID • CLUWE_OWNER.ESIG_MEANING : Added columns IS_ACTIVE, SORT_ORDER • CLUWE_OWNER.SDDMETADATA: Update column descriptions KEY and VALUE • CLUWE_OWNER.RELOCK_LOG: Changed type of columns ARTIFACT_PATH, STATUS_MESSAGE, Changed description of the table • CLUWE_OWNER.JREVIEW_LOG: Changed description of table • Added the following tables a) AUDIT_ESIG_MEANING b) CAMEL_MESSAGEPROCESSED c) CAMEL_MESSAGETRACED d) COMPUTE_JOB e) COMPUTE_JOB_ERR_STATUS f) COMPUTE_JOB_EVENT g) COMPUTE_JOB_STATUS h) SAS_JOB_CONTEXT i) SAS_JOB_DTLS j) SESSION_LOG k) USER_PREF • Updated GUI Design for Schedule Tab	
7	23-May-2017	Added Training grid on the first page. Removed OSS section as it is available in the System Overview document.	Priyanka Mehra, Validation Lead
6	17-May-2017	Updated the below sections pertaining to CHG1078626: CLUWE_OWNER.SDDFILE: Column length for SDD_PATH field updated to 1000 characters	Nidhi Gupta, IT SME
5	10-Mar-2017	Updated for Release 2.5, CHG1057702. The following changes were made: • Added sections 1.1.15 and 1.6.6 for JReview integration	David Fleig, IT SME
4	8-Nov-2016	Updated for release 2.3, CHG1006633 Added table 1.1.13 LOG_RELOCK	John Jason Rees, IT SME
3	27 Sep-2016	Updated for CHG0988306, R2.1. 1.1.6: Added new line to table; STARTED_UTC	Emily Harshman, IT SME
2	22-Sep-2016	Updated the below sections pertaining to CHG0981297: • CLUWE_OWNER.ESIG_LOG : Column Length Updated To 1000 Characters • CLUWE_OWNER.FILELOCK_LOG : Column Length Updated To 1000 Characters	Sharmistha Bhattacharya, IT SME

Version	Revision Date	Reason for Revision (Include CR#, if Applicable)	Revised By, Title
		• CLUWE_OWNER.JOB_EXECUTION_LOG : Column Length Updated To 1000 Characters • CLUWE_OWNER.SDDFILE : Column Length Updated To 1000 Characters • CLUWE_OWNER.SDDMETADATA : Column Length Updated To 1000 Characters; Added the purpose of column SDDFILE_FK • CLUWE_OWNER.USER_FAV : Column Length Updated To 1000 Characters • CLUWE_OWNER.VERSION_LOG : Column Length Updated To 1000 Characters • External Resources : Updated Unix To Linux • References : Added New Folder Location • Configuration Parameters: Updated description of fileLockerMaximumAgeInHours and fileLockerMinimumAgeInMinutes properties • File Browser: Rephrased the content • Reports: Updated referred document name • Data Conversion/Migration: Updated referred document name • Traceability: Updated referred document name *_Numerous updates have been made to correct spelling and grammar; abbreviations, etc. These updates did not change their intent or meaning and have been added to increase clarity and readability._	
1	17-Mar-2016	Release 1.2 - CHG0090403	David C. Fleig Execution Lead
0	7-Oct-2015	New Document.	David C. Fleig, Execution Lead

Table of Contents

1. DETAILED DESIGN SPECIFICATIONS

1.1 Database Design

The Oracle 12c database stores both transactional data as well as long term audit data generated by the system. Spring Data JPA (Java Persistence Application Programming Interface (API)) is used in conjunction with Hibernate to perform Object Relational Mapping (ORM) functionality for the application. The tables, sequences, indexes etc., are created under the CLUWE_OWNER schema in the development environment using Hibernate automatic schema generation and a **Structured Query Language (SQL)** script provided by the Quartz java library. Once the schema is complete it is migrated to qa and prd by the Global Data Management team. The complete Data Definition Language (DDL) for the tables can be obtained from the database.

All database columns with "_UTC" appended to the name are date and time values stored in Universal Coordinated Time (UTC) format.

1.1.1 CLUWE_OWNER.AUTH_LOG

This table stores a record for every login into the system.

JPA Entity	com.lilly.cluwe.integration.ldap.model.log.AuthLogEntry.java	
Spring Repository	com.lilly.cluwe.integration.ldap.model.log.IAuthLogRepository.java	
Column Name	**Type**	**Purpose**
AUTH_LOG_ID	NUMBER(19,0)	Primary Key
ACTION	VARCHAR2(255 CHAR)	Describes the action performed by the user.
ADGROUPCOUNT	NUMBER(10,0)	If authentication was successful, this is the count of the number of Active Directory (AD) groups a user is a member of. This data is stored for diagnostic purposes in case Kerberos token size issues occur.
CREATED_UTC	TIMESTAMP(6)	The data and time in Universal Coordinated Time the event occurred.
ROLES	VARCHAR2(255 CHAR)	If authentication was successful, contains the list of Spring Security Roles assigned to the user.
SERVER	VARCHAR2(255 CHAR)	The hostname of the server through which the user logged in.
TRANSACTION_ID	VARCHAR2(255 CHAR)	The system generated identifier used to map user transactions to database log entries for diagnostic purposes.
USERDN	VARCHAR2(255 CHAR)	If authenticated, the AD Distinguished Name (DN) of the user.
USERID	VARCHAR2(255 CHAR)	The System ID the user entered in the login page.
USERNAME	VARCHAR2(255 CHAR)	If authenticated, the AD Displays Name of the user.
SID	VARCHAR2(255 CHAR)	The value that specifies the security identifier (SID) of the user.

UPDATED_UTC	TIMESTAMP(6)	The date and time in Universal Coordinated Time the event was updated.

1.1.2 CLUWE_OWNER.AVA_EVENT

This table stores a record for every AVA event.

JPA Entity	com.lilly.cluwe.service.ava.audit.AVAEventRecord.java	
Spring Repository	com.lilly.cluwe.service.ava.audit.IAVAEventRepository.java	
Column Name	**Type**	**Purpose**
EVENT_ID	NUMBER(19,0)	Primary Key
USER_ID	VARCHAR2(255 CHAR)	The id of the user who performed the versioning
EVENT_TIME	TIMESTAMP(6)	The date and time that the event occurred
TRANSACTION_ID	VARCHAR2(255 CHAR)	The system generated identifier used to map user transactions to database log entries for diagnostic purposes.
JOB_ID	NUMBER(19,0)	Foreign Key to the Compute Job Table
EVENT_DESC	VARCHAR2(255 CHAR)	Description of the AVA Event
DATA	CLOB	Any relevant data about the AVA event such as file location or Stringified HTTPRequest
CREATED_UTC	TIMESTAMP(6)	The data and time in Universal Coordinated Time the event occurred.
UPDATED_UTC	TIMESTAMP(6)	The date and time in Universal Coordinated Time the event was updated.

1.1.3 CLUWE_OWNER.ESIG_LOG

This table stores a record for every Electronic Signature performed in the system. Even a single signature can be applied to multiple files so each file has its own record. The transaction id can be used to group by signature event.

JPA Entity	com.lilly.cluwe.service.signature.SignatureLogEntry.java	
Spring Repository	com.lilly.cluwe.service.signature.ISignatureLogRepository.java	
Column Name	**Type**	**Purpose**
ESIG_LOG_ID	NUMBER(19,0)	Primary Key
APPROVAL_TIMESTAMP_UTC	TIMESTAMP(6)	The data and time in Universal Coordinated Time the signatory applied the signature.

ARTIFACT_ID	VARCHAR2(255 CHAR)	The Isilon file identifier of the signed file.
ARTIFACT_MOD_DATE	TIMESTAMP(6)	The date and time the file was last modified at the time of signing.
ARTIFACT_PATH	VARCHAR2(1000 CHAR)	The path of the signed file.
DELEGATEFOR_NAME	VARCHAR2(255 CHAR)	The name of the user the signatory is a delegate for.
DELEGATEFOR_ID	VARCHAR2(255 CHAR)	The system id of the user the signatory is a delegate for.
IS_DELEGATED	NUMBER(1,0)	Boolean value indicating if the signature was delegated.
SIGNATORY_NAME	VARCHAR2(255 CHAR)	The name of the signatory.
SIGNATORY_ID	VARCHAR2(255 CHAR)	The system id of the signatory.
TRANSACTION_ID	VARCHAR2(255 CHAR)	The system generated identifier used to map user transactions to database log entries for diagnostic purposes.
SIG_MEANING_FK	NUMBER(19,0)	The id of the signature meaning record.
STATUS	VARCHAR2(255 CHAR)	The current status of the operation
COMPLETE_TIMESTAMP_UTC	TIMESTAMP(6)	The date time stamp in UTC when the signature operation is completed.
STATUS_MESSAGE	CLOB	Descriptive message detailing the status.
CLUSTER_ID	VARCHAR2(255 CHAR)	Cluster ID of Isilon.
CREATED_UTC	TIMESTAMP(6)	The data and time in Universal Coordinated Time the record was created.
UPDATED_UTC	TIMESTAMP(6)	The date and time in Universal Coordinated Time the record was updated.

1.1.4 CLUWE_OWNER.FILELOCK_LOG

This table stores a record for every lock operation performed on the isilon file system.

JPA Entity	com.lilly.cluwe.service.repository.FileLockLogEntry.java	
Spring Repository	com.lilly.cluwe.service.repository.IFileLockLogRepository.java	
Column Name	**Type**	**Purpose**
FILELOCK_LOG_ID	NUMBER(19,0)	Primary Key
ARTIFACT_ID	VARCHAR2(255 CHAR)	The Isilon file identifier of the locked file
ARTIFACT_PATH	VARCHAR2(1000 CHAR)	The Isilon path of the locked file.
LOCKED_UTC	TIMESTAMP(6)	The time and date the file was locked.
STATUS	VARCHAR2(255 CHAR)	The final status of the lock operation.
TRANSACTION_ID	VARCHAR2(255 CHAR)	The system generated identifier used to map user transactions to database log entries for diagnostic purposes.
UPDATED_UTC	TIMESTAMP(6)	The file system timestamp of when the file was last updated.
CREATED_UTC	TIMESTAMP(6)	The file system timestamp of when the file was created.
CLUSTER_ID	VARCHAR2(255 CHAR)	Cluster ID of Isilon.
STATUS_MESSAGE	CLOB	Descriptive message detailing the status.
CREATED_UTC	TIMESTAMP(6)	The data and time in Universal Coordinated Time the event occurred.
FILE_CREATED_UTC	TIMESTAMP(6)	The date and time in Universal Coordinated Time the file was created on the Isilon
FILE_UPDATED_UTC	TIMESTAMP(6)	The date and time in Universal Coordinated Time the file was updated on the Isilon

1.1.5 CLUWE_OWNER.ESIG_MEANING

This table stores information of the signature meaning that the users choose before they perform a sign operation.

JPA Entity	com.lilly.cluwe.service.signature.SignatureMeaning.java	
Spring Repository	com.lilly.cluwe.service.signature.ISignatureMeaningRepository.java	
Column Name	**Type**	**Purpose**
SIG_MEANING_ID	NUMBER(19,0)	Primary Key

SIG_MEANING	VARCHAR2(255 CHAR)	The Signature Meaning text.
TRANSACTION_ID	VARCHAR2(255 CHAR)	The system generated identifier used to map user transactions to database log entries for diagnostic purposes.
IS_ACTIVE	NUMBER(1,0)	Boolean value that determines whether the signature meaning should be shown in the UI or not.
SORT_ORDER	NUMBER(38,0)	This determines the order in which signature meaning is shown in the UI
CREATED_UTC	TIMESTAMP(6)	The data and time in Universal Coordinated Time the record was created.
UPDATED_UTC	TIMESTAMP(6)	The date and time in Universal Coordinated Time the record was updated.

1.1.6 CLUWE_OWNER. JOB_CONTEXT

This table stores the Statistical Analysis System (SAS) Grid Job Context information displayed to the user and passed to the SAS Grid.

JPA Entity	com.lilly.cluwe.integration.sasgrid.model.JobContext.java	
Spring Repository	com.lilly.cluwe.integration.sasgrid.repository.IJobContextRepository.java	
Column Name	**Type**	**Purpose**
JOB_CONTEXT_ID	NUMBER(19,0)	Primary Key
CONTEXT_DISPLAY_NAME	VARCHAR2(255 CHAR)	The value displayed to the user in the dropdown control on the Job Scheduling page.
CONTEXT_DESCRIPTION	VARCHAR2(255 CHAR)	A short description of the context.
CONTEXT_NAME	VARCHAR2(255 CHAR)	The context name passed to the SAS Grid through sasgsub. This value must match a value on the grid.
CREATED_UTC	TIMESTAMP(6)	The data and time in Universal Coordinated Time the record was created.
UPDATED_UTC	TIMESTAMP(6)	The date and time in Universal Coordinated Time the record was updated.

1.1.7 CLUWE_OWNER.JOB_EXECUTION_LOG

This table stores a record for every SAS program that's scheduled for execution on SAS Grid.

JPA Entity	com.lilly.cluwe.service.scheduler.JobExecutionLogEntry
Spring Repository	com.lilly.cluwe.service.scheduler.IJobExecutionLogRepository

Column Name	Type	Purpose
JOB_EXECUTION_LOG_ID	NUMBER(19,0)	Primary Key
BATCHTRANSACTIONID	VARCHAR2(255 CHAR)	The system generated identifier used to map user transactions to database log entries for diagnostic purposes. This id applies to the entire group of files selected for execution.
CLUWE_JOB_ID	VARCHAR2(255 CHAR)	Unique job identifier
COMMAND_LINE	VARCHAR2(1000 CHAR)	The command line string passed to the Linux shell for execution.
ERROR_OUTPUT	CLOB	The results of STANDARD ERROR (STDERR)
EXECUTION_OUTPUT	CLOB	The results of STANDARD OUTPUT (STDOUT)
EXECTRANSACTIONID	VARCHAR2(255 CHAR)	The system generated identifier used to map user transactions to database log entries for diagnostic purposes. This id applies to the individual executions.
HIDEFROMUI	NUMBER(1,0)	Boolean to indicate if the record should be hidden from the UI.
RESCHEDULEDFLG	NUMBER(1,0)	Boolean to indicate if the job was rescheduled at any time.
COMPLETED_UTC	TIMESTAMP(6)	The data and time the quartz job completed.
CREATED_UTC	TIMESTAMP(6)	The date and time this record was created.
QUEUE	NUMBER(10,0)	The SAS Grid queue the job was placed in.
SUBMITTED_UTC	TIMESTAMP(6)	Unused in this release
LOG_PATH	VARCHAR2(255 CHAR)	The path of the log file produced by SAS Grid
NATIVE_JOB_ID	VARCHAR2(255 CHAR)	The SAS Grid Job Id.
PROGRAM_NAME	VARCHAR2(255 CHAR)	The name of the program file.
PROGRAM_PATH	VARCHAR2(255 CHAR)	The full path of the program file.
EMAIL	NUMBER(1,0)	Unused in this release
STATUS	VARCHAR2(255 CHAR)	The final status of the submission.
USER_ID	VARCHAR2(255 CHAR)	The id of the user that scheduled the job
EXIT_CODE	VARCHAR2(255 CHAR)	The final status of the submitsasjob.sh script.

JPA Entity	com.lilly.cluwe.service.scheduler.JobExecutionLogEntry	
Spring Repository	com.lilly.cluwe.service.scheduler.IJobExecutionLogRepository	
Column Name	**Type**	**Purpose**
JOB_QUEUE	VARCHAR2(255 CHAR)	The SAS Grid queue in which the job was placed.
SCHEDULED_UTC	TIMESTAMP(6)	The timestamp of when to submit the job.
DISPLAY_PATH	VARCHAR2(255 CHAR)	The file path displayed to the user.
STARTED_UTC	TIMESTAMP(6)	The timestamp when the job starts.
UPDATED_UTC	TIMESTAMP(6)	The date and time in Universal Coordinated Time the record was updated.

1.1.8 CLUWE_OWNER.JOB_INFO_LOG

This table stores the details of the last time the Lock-In-Place process ran.

JPA Entity	com.lilly.cluwe.integration.datacore.model.CluweJobInfo.java	
Spring Repository	com.lilly.cluwe.integration.datacore.repository.ICluweJobInfoRepository.java	
Column Name	**Type**	**Purpose**
JOB_INFO_LOG_ID	NUMBER(19,0)	Primary Key
BEGIN_WINDOW	TIMESTAMP(6)	The date and time of the current BeginWindow value.
END_WINDOW	TIMESTAMP(6)	The date and time of the current EndWindow value.
FIRE_TIME	TIMESTAMP(6)	The date and time the job started.
JOB_NAME	VARCHAR2(255 CHAR)	The Quartz name of the job.
RUN_TIME	NUMBER(19,0)	The amount of time in milliseconds that the job ran.
CREATED_UTC	TIMESTAMP(6)	The data and time in Universal Coordinated Time the record was created.
UPDATED_UTC	TIMESTAMP(6)	The date and time in Universal Coordinated Time the record was updated.

1.1.9 CLUWE_OWNER.SDDAPPROVAL

This table stores a record for every migrated SAS Drug Development (SDD) electronic approval. This table has a many-to-one relationship with the SDDFILE table.

JPA Entity	com.lilly.cluwe.service.sdd.SddApprovalEntry.java

Spring Repository	N/A	
Column Name	**Type**	**Purpose**
SDDAPPROVAL_ID	NUMBER(19,0)	Primary Key
APPROVED_UTC	TIMESTAMP(6)	The date and time the record was approved.
APPROVER_NAME	VARCHAR2(255 CHAR)	The name of the approver.
APPROVER_ID	VARCHAR2(255 CHAR)	The SDD Id of the approver.
REASON	VARCHAR2(255 CHAR)	The approval meaning selected by the approver.
SDDFILE_FK	NUMBER(19,0)	The primary key of the SDD File entry this record is associated to.
CREATED_UTC	TIMESTAMP(6)	The data and time in Universal Coordinated Time the record was created.
UPDATED_UTC	TIMESTAMP(6)	The date and time in Universal Coordinated Time the record was updated.

1.1.10 CLUWE_OWNER.SDDFILE

This table stores a record for every SDD file migrated by the M2C application. This record has a one-to-many relationship with the SDDMETADATA and SDDAPPROVAL tables.

JPA Entity	com.lilly.cluwe.service.sdd.SddFileEntry.java	
Spring Repository	com.lilly.cluwe.service.sdd.ISddRepository.java	
Column Name	**Type**	**Purpose**
SDDFILE_ID	NUMBER(19,0)	Primary Key
CLUWE_ID	VARCHAR2(255 CHAR)	The Isilon file id.
CLUWE_PATH	VARCHAR2(1000 CHAR)	The path the file was stored in. This value is before the file is versioned so the final location of the file may be in the prior versions folder.
CREATED_UTC	TIMESTAMP(6)	The date and time in UTC that the file was migrated.
SDDOBJECT_ID	VARCHAR2(255 CHAR)	The SDD identifier
SDD_PATH	VARCHAR2(1000 CHAR)	The SDD path of the file
SDD_VERSION	NUMBER(10,0)	The SDD version number of the file
TRANSACTION_ID	VARCHAR2(255 CHAR)	The primary key of the SDD File entry this record is associated

JPA Entity	com.lilly.cluwe.service.sdd.SddFileEntry.java	
Spring Repository	com.lilly.cluwe.service.sdd.ISddRepository.java	
Column Name	**Type**	**Purpose**
		to.
UPDATED_UTC	TIMESTAMP(6)	The date and time in Universal Coordinated Time the record was updated.

1.1.11 CLUWE_OWNER.SDDMETADATA

This table stores a record for every metadata entry migrated by the M2C application. This table has a many-to-one relationship with the SDDFILE table.

JPA Entity	com.lilly.cluwe.service.sdd.SddMetadataEntry	
Spring Repository	N/A	
Column Name	**Type**	**Purpose**
SDDMETADATA_ID	NUMBER(19,0)	Primary Key
KEY	VARCHAR2(255 CHAR)	The metadata key name
VALUE	VARCHAR2(1000 CHAR)	The metadata value
SDDFILE_FK	NUMBER(19,0)	The primary key of CLUWE_OWNER_SDDFILE table
CREATED_UTC	TIMESTAMP(6)	The data and time in Universal Coordinated Time the record was created.
UPDATED_UTC	TIMESTAMP(6)	The date and time in Universal Coordinated Time the record was updated.

1.1.12 CLUWE_OWNER.USER_FAV

This table stores a record for every user favorite created through the UI.

JPA Entity	com.lilly.cluwe.service.user.model.UserFavorite.java	
Spring Repository	com.lilly.cluwe.service.user.data.IUserFavoriteRepsitory.java	
Column Name	**Type**	**Purpose**
USER_FAV_ID	NUMBER(19,0)	Primary Key
CREATED	TIMESTAMP(6)	The date and time the favorite was created

JPA Entity	com.lilly.cluwe.service.user.model.UserFavorite.java	
Spring Repository	com.lilly.cluwe.service.user.data.IUserFavoriteRepsitory.java	
Column Name	**Type**	**Purpose**
NAME	VARCHAR2(255 CHAR)	The name of the favorite. This is what is displayed in the UI.
PATH	VARCHAR2(1000 CHAR)	The file path selected from the UI.
POSITION	NUMBER(10,0)	Unused in this release.
SCOPE	VARCHAR2(255 CHAR)	Unused in this release.
TRANSACTION_ID	VARCHAR2(255 CHAR)	The system generated identifier used to map user transactions to database log entries for diagnostic purposes.
USER_ID	VARCHAR2(255 CHAR)	The id of the user who created the favorite.
CREATED_UTC	TIMESTAMP(6)	The data and time in Universal Coordinated Time the record was created.
UPDATED_UTC	TIMESTAMP(6)	The date and time in Universal Coordinated Time the record was updated.

1.1.13 CLUWE_OWNER.VERSION_LOG

This table stores a record for file versioned by the system.

JPA Entity	com.lilly.cluwe.service.version.model.log.VersionLogEntry.java	
Spring Repository	com.lilly.cluwe.service.version.data.log.IVersionLogRepository.java	
Column Name	**Type**	**Purpose**
FILE_LOCK_ID	NUMBER(19,0)	Primary Key.
ARTIFACT_ID	VARCHAR2(255 CHAR)	The Isilon file identifier.
ARTIFACT_PATH	VARCHAR2(1000 CHAR)	The path of the file before versioning.
RULE_OVERRIDE	NUMBER(1,0)	Boolean to indicate if the business rules were overridden.
STATUS	VARCHAR2(255 CHAR)	The final status of the version action.
TRANSACTION_ID	VARCHAR2(255 CHAR)	The system generated identifier used to map user transactions to database log entries for diagnostic purposes.

USER_ID	VARCHAR2(1000 CHAR)	The id of the user who initiated the action.
VERSIONED_UTC	TIMESTAMP(6)	The time and date the file was versioned.
VERSIONED_PATH	VARCHAR2(255 CHAR)	The path of the file after versioning.
COMPLETE_TIMESTAMP_UTC	TIMESTAMP(6)	The date time stamp in UTC when the version operation is completed.
STATUS_MESSAGE	CLOB	Descriptive message detailing the status.
CLUSTER_ID	VARCHAR2(255 CHAR)	Cluster ID of Isilon.
CREATED_UTC	TIMESTAMP(6)	The data and time in Universal Coordinated Time the record was created.
UPDATED_UTC	TIMESTAMP(6)	The date and time in Universal Coordinated Time the record was updated.

1.1.14 CLUWE_OWNER.RELOCK_LOG

This table stores a record for every folder object relocked by system

JPA Entity	com.lilly.cluwe.service.provision.log.ProvisionLogEntry	
Spring Repository	com.lilly.cluwe.service.provision.log.IProvisionLogRepository	
Column Name	**Type**	**Purpose**
RELOCK_LOG_ID	Identity	Primary Key
USER_ID	VARCHAR2 (255 CHAR)	User’s system ID
STATUS	VARCHAR2 (255 CHAR)	Status
RELOCK_UTC	TIMESTAMP (6)	Time at which the relock was requested.
TRANSACTION_ID	VARCHAR2 (255 CHAR)	Id for tracking the request.
ARTIFACT_PATH	VARCHAR2 (1000 CHAR)	Full path of the directory being relocked.
STATUS_MESSAGE	CLOB	Descriptive message detailing the status.
COMPLETE_TIMESTAMP_UTC	TIMESTAMP (6)	Timestamp of the completion of the relock request.
CREATED_UTC	TIMESTAMP(6)	The data and time in Universal Coordinated Time the record was created.
UPDATED_UTC	TIMESTAMP(6)	The date and time in Universal Coordinated Time the

		record was updated.

1.1.15 CLUWE_OWNER.QRTZ_* Tables

These tables are defined and managed by the Quartz library. The system interacts with the tables through the Quartz java libraries.

1.1.16 CLUWE_OWNER.JREVIEW_LOG

This table stores the details that are associated with every JReview run.

JPA Entity	com.lilly.cluwe.service.jreview.model.log.JReviewLogEntry.java	
Spring Repository	com.lilly.cluwe.service.jreview.data.log.IJReviewLogRepository.java	
Column Name	**Type**	**Purpose**
JREVIEW_LOG_ID	NUMBER(19,0)	Primary Key
ADDTNL_DTLS	VARCHAR2 (255 Char)	Miscellaneous execution information.
NO_OF_ENTRIES	NUMBER (10)	The number of permission entries written to the .csv file.
END_TIME_UTC	TIMESTAMP(6)	UTC Timestamp of when the process ended.
START_TIME_UTC	TIMESTAMP(6)	UTC Timestamp of when the process started.
RUN_SERVER_NAME	VARCHAR2(255 CHAR)	The server on which the process executed.
STATUS	VARCHAR2(255 CHAR)	The final status of the process.
TIME_TO_RUN_MILLIS	NUMBER (19, 0)	The amount of time required to execute the entire process in milliseconds.
TRANSACTION_ID	VARCHAR2(255 CHAR)	Id for tracking the request.
CREATED_UTC	TIMESTAMP(6)	The data and time in Universal Coordinated Time the record was created.
UPDATED_UTC	TIMESTAMP(6)	The date and time in Universal Coordinated Time the record was updated.

1.1.17 CLUWE_OWNER.AUDIT_ESIG_MEANING

This table logs all changes that are made to ESIG_LOG table.

JPA Entity	N/A
Spring Repository	N/A

Column Name	Type	Purpose
AUDIT_ESIG_ID	NUMBER(38,0)	Primary Key
AUDIT_OPERATION	VARCHAR2(50 CHAR)	Indicates the type of operation that was performed on ESIG_LOG table
AUDIT_LAST_UPDT_DT	DATE	Timestamp of when the audit record was last updated.
AUDIT_USER	VARCHAR2(50 CHAR)	User associated with changing the ESIG_LOG table.
SIG_MEANING_ID	NUMBER(19,0)	Signature meaning id
SIG_MEANING	VARCHAR2(255 CHAR)	Signature meaning text
TRANSACTION_ID	VARCHAR2(255 CHAR)	Stores the user id
IS_ACTIVE	NUMBER(1,0)	Boolean value that determines whether the signature meaning should be shown in the UI or not.
SORT_ORDER	NUMBER(38,0)	This determines the order in which signature meaning is shown in the UI
LAST_UPDT_DT	DATE	Timestamp of when the ESIG_LOG record was last updated.

1.1.18 CLUWE_OWNER.CAMEL_MESSAGEPROCESSED

This table is defined and managed by camel library. The system interacts with the tables through the spring repository java libraries.

1.1.19 CLUWE_OWNER.CAMEL_MESSAGETRACED

This table stores records of camel messages details when exceptions occur.

JPA Entity	com.lilly.cluwe.service.computejob.model.CamelMessageTraced.java	
Spring Repository	com.lilly.cluwe.service.computejob.repository.ICamelMessageTracedRepository.java	
Column Name	**Type**	**Purpose**
TRACED_ID	NUMBER(19,0)	Primary Key
BODY	CLOB	Body of the Camel message
CAUSED_BY_EXC	CLOB	Exception that caused the error
EXCHANGE_ID	VARCHAR2(255 CHAR)	Camel message exchange id
EXC_PATTERN	VARCHAR2(255 CHAR)	The ID of the related job
FROM_URI	VARCHAR2(255 CHAR)	Camel message from endpoint

HEADERS	CLOB	Camel message headers
PROPERTIES	CLOB	Camel message exchange properties
ROUTE_ID	VARCHAR2(255 CHAR)	The id of the Camel route
TIMESTAMP	TIMESTAMP(6)	The time the exception occurred
CREATED_UTC	TIMESTAMP(6)	The data and time in Universal Coordinated Time the record was created.
UPDATED_UTC	TIMESTAMP(6)	The date and time in Universal Coordinated Time the record was updated.

1.1.20 CLUWE_OWNER.COMPUTE_JOB

This table stores the details that are associated with every job that is scheduled through CLUWE.

JPA Entity	com.lilly.cluwe.service.computejob.model.ComputeJob.java	
Spring Repository	com.lilly.cluwe.service.computejob.repository.IComputeJobRepository.java	
Column Name	**Type**	**Purpose**
COMPUTE_JOB_ID	NUMBER(19,0)	Primary Key
ARTIFACT_PATH	VARCHAR2(1000 CHAR)	The entire path of the file including the path of the cluster
ENGINE_JOB_ID	VARCHAR2(255 CHAR)	The identifying job id given by the engine
ENGINE_NAME	VARCHAR2(255 CHAR)	The underlying engine used to execute the job.
FILE_PATH	VARCHAR2(1000 CHAR)	The display path that the user sees in the file tree
HIDE_FROM_UI	NUMBER(1,0)	Boolean value indicating if the job is to be reflected on the UI
JOB_CREATION_DATE	TIMESTAMP(6)	The point in time the job was created
SEND_COMPLETION_EMAIL	NUMBER(1,0)	Boolean value indicating if the completion email should be sent
SEND_COMPLETION_TEXT	NUMBER(1,0)	Boolean value indicating if the completion text should be sent
SUBMITTER_ID	VARCHAR2(255 CHAR)	The id of the user who submitted the job
TRANSACTION_ID	VARCHAR2(255 CHAR)	The system generated identifier used to map user transactions to database log entries for diagnostic

		purposes.
ARTIFACT_ID	NUMBER(19,0)	The Isilon file identifier of the file for which a job is scheduled.
LOG_PATH	VARCHAR2(1000 CHAR)	The path of the log file generated during the execution of the program on SAS grid.
CLUSTER_ID	VARCHAR2(255 CHAR)	Cluster id associated with Isilon
PRFD_PHONE_NUMBER	VARCHAR2(255 CHAR)	Preferred phone number of User with respect to Job
CREATED_UTC	TIMESTAMP(6)	The data and time in Universal Coordinated Time the record was created.
UPDATED_UTC	TIMESTAMP(6)	The date and time in Universal Coordinated Time the record was updated.
RUN_THROUGH_AVA	NUMBER(1,0)	Boolean which indicates if the job was run through AVA

1.1.21 CLUWE_OWNER.COMPUTE_JOB_ERR_STATUS

This table stores the details of various error status codes for various compute job scenarios

JPA Entity	com.lilly.cluwe.service.computejob.model.ComputeJobErrorStatus.java	
Spring Repository	com.lilly.cluwe.service.computejob.repository.IComputeJobErrorStatusRepository.java	
Column Name	**Type**	**Purpose**
ERROR_STATUS_ID	NUMBER(19,0)	Primary Key
ERR_DESCRIPTION	VARCHAR2(255 CHAR)	Details of the particular error code
EVENT_SOURCE	VARCHAR2(255 CHAR)	The source associated with the event
RESULT_CODE	NUMBER(10,0)	Details of the particular error code
CREATED_UTC	TIMESTAMP(6)	The data and time in Universal Coordinated Time the record was created.
UPDATED_UTC	TIMESTAMP(6)	The date and time in Universal Coordinated Time the record was updated.

1.1.22 CLUWE_OWNER.COMPUTE_JOB_EVENT

This table stores the details of events that are associated with all compute job entities.

JPA Entity	com.lilly.cluwe.service.computejob.model.ComputeJobEvent.java
Spring Repository	com.lilly.cluwe.service.computejob.repository.IComputeJobEventRepository.java

Column Name	**Type**	**Purpose**
CJ_EVENT_ID	NUMBER(19,0)	Primary Key
DESCRIPTION	CLOB	Details associated with the event.
EVENT_CREATED_DATE_UTC	TIMESTAMP(6)	The point in time the event was created in UTC.
EVENT_ENGINE_DATE_UTC	TIMESTAMP(6)	The point in time the event occurred based on the associated engine in UTC.
EVENT_SOURCE	VARCHAR2(255 CHAR)	The source associated with the event.
EVENT_NAME	VARCHAR2(255 CHAR)	Name of the event
RESULT_CODE	NUMBER(10,0)	The result information associated with the event source
TRANSACTION_ID	VARCHAR2(255 CHAR)	The system generated identifier used to map user transactions to database log entries for diagnostic purposes.
COMPUTE_JOB_ID	NUMBER(19,0)	Foreign Key to the Compute Job Table
CREATED_UTC	TIMESTAMP(6)	The data and time in Universal Coordinated Time the record was created.
UPDATED_UTC	TIMESTAMP(6)	The date and time in Universal Coordinated Time the record was updated.

1.1.23 CLUWE_OWNER.COMPUTE_JOB_STATUS

This table is used to get the status of a job.

JPA Entity	com.lilly.cluwe.service.computejob.model.ComputeJobStatus.java	
Spring Repository	com.lilly.cluwe.service.computejob.repository.IComputeJobStatusRepository.java	
Column Name	**Type**	**Purpose**
STATUS_ID	NUMBER(19,0)	Primary Key
EVENT_TYPE	VARCHAR2(255 CHAR)	Name of the event
CURRENT_STATUS	VARCHAR2(255 CHAR)	Current Status of a job
EVENT_SOURCE	VARCHAR2(255 CHAR)	The source associated with the event.
RESULT_CODE	NUMBER(10,0)	The result information associated with the event source

CREATED_UTC	TIMESTAMP(6)	The data and time in Universal Coordinated Time the record was created.
UPDATED_UTC	TIMESTAMP(6)	The date and time in Universal Coordinated Time the record was updated.

1.1.24 CLUWE_OWNER.SAS_JOB_CONTEXT

This table stores the available context and display name for sas jobs.

JPA Entity	com.lilly.cluwe.integration.sasgrid.model.SasJobContext.java	
Spring Repository	com.lilly.cluwe.integration.sasgrid.repository.ISasJobContextRepository.java	
Column Name	**Type**	**Purpose**
CONTEXT_ID	NUMBER(19,0)	Primary Key
CONTEXT_NAME	VARCHAR2(255 CHAR)	Name of the context
CONTEXT_DESCRIPTION	VARCHAR2(255 CHAR)	Description of the context
CONTEXT_DISPLAY_NAME	VARCHAR2(255 CHAR)	Name to display the context as
CREATED_UTC	TIMESTAMP(6)	The data and time in Universal Coordinated Time the record was created.
UPDATED_UTC	TIMESTAMP(6)	The date and time in Universal Coordinated Time the record was updated.

1.1.25 CLUWE_OWNER.SAS_JOB_DTLS

This table stores the details of all the jobs that we fetch by running “bjobs”.

JPA Entity	com.lilly.cluwe.integration.sasgrid.bjobs.SasJobDetails.java	
Spring Repository	com.lilly.cluwe.integration.sasgrid.bjobs.ISasJobDetailsRepository.java	
Column Name	**Type**	**Purpose**
SAS_JOB_DTLS_ID	NUMBER(19,0)	Primary Key
CPU_USED	VARCHAR2(255 CHAR)	CPU used by the job
EXEC_HOST	VARCHAR2(255 CHAR)	Host executed against
FINISH_TIME	VARCHAR2(255 CHAR)	Time the job finished
FROM_HOST	VARCHAR2(255 CHAR)	Host that initiated the job
JOB_NAME	VARCHAR2(255 CHAR)	Name of the job

MEM	VARCHAR2(255 CHAR)	Memory used by the job
PIDS	VARCHAR2(255 CHAR)	Process ids associated with the job
PROJ_NAME	VARCHAR2(255 CHAR)	Project name associated with the job
QUEUE	VARCHAR2(255 CHAR)	Queue associated with the job
SAS_JOB_ID	VARCHAR2(255 CHAR)	Job id as identified within SAS
START_TIME	VARCHAR2(255 CHAR)	Time the job was started within SAS
STATUS	VARCHAR2(255 CHAR)	Status of the job within SAS
SUBMIT_TIME	VARCHAR2(255 CHAR)	Time the job was submitted to SAS
SWAP	VARCHAR2(255 CHAR)	Total virtual memory (swap) usage of all processes in a job
USER_ID	VARCHAR2(255 CHAR)	ID of the user who is associated with the SAS job
CREATED_UTC	TIMESTAMP(6)	The data and time in Universal Coordinated Time the record was created.
UPDATED_UTC	TIMESTAMP(6)	The date and time in Universal Coordinated Time the record was updated.

1.1.26 CLUWE_OWNER.SESSION_LOG

This table stores the details that are associated with the web socket connection details.

JPA Entity	com.lilly.cluwe.webapp.websocket.SessionLogEntry.java	
Spring Repository	com.lilly.cluwe.webapp.websocket.ISessionLogRepository.java	
Column Name	**Type**	**Purpose**
SESSION_LOG_ID	NUMBER(19,0)	Primary Key
ACTION	VARCHAR2(255 CHAR)	Action of the web socket event.
CREATED_UTC	TIMESTAMP(6)	Time when the web socket event is created in UTC.
SESSIONID	VARCHAR2(255 CHAR)	Session id associated with the user activity.
USERDN	VARCHAR2(255 CHAR)	User Dn associated with the user.
USERID	VARCHAR2(255 CHAR)	User id associated with web socket event.
USERNAME	VARCHAR2(255 CHAR)	User name associated with the web socket event.

UPDATED_UTC	TIMESTAMP(6)	The date and time in Universal Coordinated Time the record was updated.

1.1.27 CLUWE_OWNER.USER_PREF

This table stores the details that are associated with user preferences.

JPA Entity	com.lilly.cluwe.service.user.model.UserPreferences.java	
Spring Repository	com.lilly.cluwe.service.user.data.IUserPreferencesRepository.java	
Column Name	**Type**	**Purpose**
USER_PREF_ID	NUMBER(19,0)	Primary Key
CREATED_UTC	TIMESTAMP(6)	Point in time the user preference was created in UTC.
PREF_TYPE	VARCHAR2(255 CHAR)	Type of preference.
PREF_VALUE	VARCHAR2(255 CHAR)	Value of the preference.
TRANSACTION_ID	VARCHAR2(255 CHAR)	The system generated identifier used to map user transactions to database log entries for diagnostic purposes.
USER_ID	VARCHAR2(255 CHAR)	User id associated with the preference.
UPDATED_UTC	TIMESTAMP(6)	The date and time in Universal Coordinated Time the record was updated.

1.1.28 CLUWE_OWNER. CLUSTER_ARTIFACT

This table stores a record for every cluster/artifact interacted for the first time.

JPA Entity	com.lilly.cluwe.service.repository.cluster.ClusterArtifact	
Spring Repository	com.lilly.cluwe.service.repository.cluster.IClusterArtifactRepository	
Column Name	**Type**	**Purpose**
CLUSTER_ART_ID	NUMBER(19,0)	Primary Key
ARTIFACT_ID	NUMBER(19,0)	Isilon Id
CLUSTER_ID	VARCHAR2(255 CHAR)	Cluster id associated with Isilon
TRANSACTION_ID	VARCHAR2(255 CHAR)	The system generated identifier used to map user transactions to database log entries for diagnostic purposes.
CREATED_UTC	TIMESTAMP	The point in time the entry was created

1.1.29 CLUWE_OWNER. CLUSTER_ARTIFACT_DTL

This table stores a record for every cluster/artifact on a per cluster basis.

JPA Entity	com.lilly.cluwe.service.repository.cluster.ClusterArtifactDetail	
Spring Repository	com.lilly.cluwe.service.repository.cluster.IClusterArtifactDetailRepository	
Column Name	**Type**	**Purpose**
CLUSTER_ART_ID	NUMBER(19,0)	Foreign key to Cluster Artifact
ARTIFACT_ID	NUMBER(19,0)	Isilon Id
CLUSTER_ID	VARCHAR2(255 CHAR)	Cluster id associated with Isilon

1.2 System Inputs and Outputs

A high level view of the system inputs and outputs can be found in the *CLUWE Web Tool System Overview* document.

1.2.1 Configuration Parameters

The following properties are stored in the application.properties file and affect how the system operates.

Property Name	**Description**
repo.papi.uri	The Isilon Platform API (PAPI) URL. This value should be different for the dev, qa and prd environments.
repo.admin.username	The PAPI username. This value should be different for the dev, qa and prd environments.
repo.admin.password	The PAPI password This value should be different for the dev, qa and prd environments.
repo.server.root	The Isilon file system root.
repo.windows.root	The CLUWE Common Internet File System (CIFS) root. This value should be different for the dev, qa and prd environments.
repo.display.parent.path	The parent path of the repo.ifs.root path. This value should be different for the dev, qa and prd environments.
repo.ifs.root	The path to the CLUWE files. This value should be different for the dev, qa and prd environments.
repo.ifs.display.root	The folder that contains the qa and prd directories. This value should be different for the dev, qa and prd environments.

Property Name	Description
version.prefix	The string value to prepend to the version string.
version.digits	The number of digits to display in the version string.
version.foldername	The name of the folder where versioned files are stored.
scheduler.maxDaysToDisplay	The number of days of scheduled jobs to show the user.
fileLockerTurnedOn	Enables or disables the Lock In Place (LIP) functionality.
fileLockerFolder	The folder at which LIP should start locking files.
fileLockerJobRunIntervalInSeconds	How often the LIP job should run in seconds.
fileLockerBatchSize	The maximum number of files LIP should process in each run.
fileLockerMaximumAgeInHours	Files as old or newer as the fileLockerMaximumAgeinHours value are considered for LIP. Considered with the fileLockerMinimumAgeInMinutes parameter to build a sliding window for LIP.
fileLockerMinimumAgeInMinutes	Files this old or older are considered for LIP.
security.role.user	The AD groups used to designate CLUWE Users. The list of AD groups is comma separated. This value should be different for the dev, qa and prd environments.
security.role.admin	The AD groups used to designate CLUWE Admin users. The list of AD groups is comma separated. This value should be different for the dev, qa and prd environments.
security.role.webservice	The AD groups used to designate CLUWE Users. The list of AD groups is comma separated. This value should be different for the dev, qa and prd environments.
security.role.deny	The AD groups used to designate deny users' access to CLUWE. The list of AD groups is comma separated. This value should be different for the dev, qa and prd environments.
security.role.itsupport	The AD groups used to designate CLUWE IT Support users. The list of AD groups is comma separated.
security.role.sasgrid	The AD groups used to designate CLUWE SAS Grid users. The list of AD groups is comma separated.
ldap.url	The AD Global Catalog LDAP URL. This value should be different for the dev, qa and prd environments.

Property Name	Description
ldap.userdn	The service account used to search for users. This value should be different for the dev, qa and prd environments.
ldap.password	The password for the ldap service account. This value should be different for the dev, qa and prd environments.
permissions.template.readonly	The directory to use as the permission mask.
JReview Parameters	
jReviewTurnedOn	Enables or disables the JReview functionality.
jReviewJobRunIntervalInSeconds	The number of seconds to wait between each job execution.
jReview.csv.file.path	The location of the exported .csv file.
jReview.create.permission.dataset.path	The location of the exported SAS dataset.
jReview.study.reg.sas.path	The location of the SAS program that converts the JReview registered trials to .CSV.
jReview.study.reg.csv.path	The location of the .csv file produced by the Study_reg_jreview.sas SAS job.
jReviewTimeToWait	The amount of time to wait for the SAS Job to export the list of trials registered in JReview.
jReview.job.user	The user the SAS Grids jobs should execute under.
jReview.job.context	The SAS Context the SAS Grids jobs should execute under.
jReview.groups.to.exclude	The groups to exclude from including in the permission export.

1.3 Design Constraints

1.3.1 Large Number of Files and Folders in the Isilon NAS

It is expected that the Isilon Network Attached Storage (NAS) will contain hundreds of thousands of files and folders over time so File Browser design relied on “lazy loading” of folder contents which means the web service call to retrieve the folder contents was not performed until the folder was opened by the user. This results in a delay when opening a folder proportional to the number of objects in the folder. Later releases of the system could add caching and paging to mitigate the delay further.

This same constraint also impacts the Lock-In-Place feature whose task is to monitor the file system for newly created files then change the permissions to read-only. Isilon Smartlock functionality was investigated but the business did not like the inability to unlock files at a later point. The only mechanisms to monitor for new files are: a full drive scan, monitor audit events or use the Isilon query Application Program Interface (API). Scanning all the files on the NAS was immediately ruled out due to the potential number of files and amount of time required to perform the scan. Monitoring the audit events was discounted because the events are only available through command-line tools, syslog files or a syslog network stream, none of which are both reliable and timely. The chosen solution was to use the Isilon query functionality provided through the Isilon Platform API (PAPI). This approach is faster than a filesystem scan, more reliable than monitoring the event logs but is still impacted by the number of files and folders on the NAS. Later releases will consider other approaches of achieving the goal of read-only files in prd.

1.3.2 Sasgsub utility does not support passing credentials

The SAS sasgsub command-line utility does not support passing credentials through parameters so the Linux sudo functionality was implemented to emulate the web user running the command. The sudo command is limited to users in the “sas” Lightweight Directory Acces Protocol (LDAP) group to ensure only SAS Grid users can execute sasgsub. The sasgsub command-line string is intentionally specified as constant in the java code instead of an application parameter to ensure that any changes pass through multiple levels of review.

1.4 Algorithms

1.4.1 File Versioning

The following steps are an overview of the actions performed by the system to version a file. The logic is implemented in the com.lilly.cluwe.webapp.service.impl.RepositoryServiceImpl class with the details regarding each step and associated error handling in the createNewVersion method.

If more than one file was selected, perform the following steps for each file.

a) Create a version log entry with the request information.
b) Check that the file actually exists. If not, update the version log entry and return an error.
c) Check if the file is in the /prd directory structure. If not, update the version log entry and return an error.
d) Check if the file is located in a prior versions folder. If so, update the version log entry and return an error.
e) Check if the user has write access to the file’s parent folder. If not, update the version log entry and return an error.
f) Check if the prior versions folder already exists. If not, create the folder.
g) Set the permission of the prior versions folder to read only using the File Locking algorithm described below.
h) Call the Isilon move web service to move the file to the previous versions folder and rename it using the Determine Next File Version algorithm below.
i) Set the permissions of the file to read-only using the File Locking algorithm described below.
j) Update the version log entry with the file path and successful status of the operation.

1.4.2 File Locking

The following steps are an overview of the actions performed by the system when removing user read access to a file or folder. The logic is implemented in the com.lilly.cluwe.webapp.service.impl.RepositoryServiceImpl class with the details regarding each step and associated error handling in the setObjectToUserReadOnly method.

a) Create an Access Control List (ACL) mask by calling the Isilon ACL web service for the /lillyce directory. The ACLs returned in this call will be not be impacted by any security changes in further steps with the exception of the SCE_Admin AD group mentioned in later steps.
b) Call the Isilon ACL web service to retrieve the ACLs applied to the object.
c) Iterate through the list of ACLs returned in the previous step performing the following actions for each ACL.
    a) Does the ACL refer to the SCE_Admin group and was the method told to make changes to the group. If so, set the permissions on the ACL to read-only. If not go to the next step.
    b) Is the ACL in the ACL mask? If so, then move to the next ACL.
    c) Is the ACL a DENY ACL? If so, then move to the next ACL.
    d) Update the ACL with read-only permissions.
d) Call the Isilon ACL web service to apply the updated ACL to the object.
e) Add an entry to the FILELOCK_LOG table.

1.4.3 Determine Next File Version

The following steps are an overview of the actions performed by the system to determine what the version string to be appended to the filename of the versioned file should be. The logic is implemented in the com.lilly.cluwe.webapp.service.impl.RepositoryServiceImpl class with the details regarding each step and associated error handling in the createNewVersion, buildVersionString and getLastVersionNumber methods.

a) Remove the extension from the file.
b) Retrieve the contents of the prior versions folder filtering the results to only filenames that start with the base filename from the previous step.
c) Is the results list empty? If so then return the string "_v001".
d) Iterate through the results list parsing the version strings to find the highest version number.
e) Increase the highest version by one, append it to the string "_v" and return the value.

1.4.4 Lock In Place (LIP)

The following steps are an overview of the actions performed by the system to determine which files should be set to read-only after creation. The logic is implemented in the multiple classes so they won't be listed here. These steps are performed every five minutes by the Quartz scheduler component. Quartz will coordinate the execution of the steps across the servers and ensure the previous LIP operation completes before another begins.

a) Update the JOB_INFO_LOG table with the current time, the current time minus four hours (WindowBeginTime) and the current time minus four days (WindowEndTime).
b) Call the Isilon Query API with to search for files with the following criteria. The query results are limited to the number of items specified in the fileLockerBatchSize property of the application.properties file.
    a) Creation time older than the WindowEndTime value and newer than the WindowBeginTime value.
    b) Modified time older than the WindowEndTime value and newer than the WindowBeginTime value.
    c) Isilon metadata value user.lockStatus is not equal to "True".
    d) The file is not in a prior_versions folder.
c) The returned results are sorted by created and modified date whichever is older.
d) Iterate through the files to be locked the permissions of the file using the algorithm described in the File Locking section of this document.

1.4.5 Execute a Scheduled SAS Job

The following steps are an overview of the actions performed by the system when a scheduled job is ready for submission to the SAS Grid. The logic implemented in the com.lilly.cluwe.service.computejob.quartz. ComputeNativeJob class provides the details regarding each step and associated error handling. Quartz is responsible for maintaining and executing the scheduled jobs so the logic it uses internally will not be listed here.

a) Retrieve the compute job data transfer object created when the job was scheduled initially or rescheduled. This compute job data transfer object contains various information such as userId, file path to the SAS program, which needs to be executed, type of notifications requested by the user when the job is completed on SAS grid, etc.
b) Quartz calls the executeJob method of class SASExecutionEngine
c) The previous step in turn calls ExecuteSASJob class, this class constructs the sudo command and opens a Linux terminal and execute the command
d) The sudo command calls the shell script /opt/cluwe/sasgsub/submitsasjob.sh that sets several SAS environment variables then executes the sasgsub command.
e) While the command is executing any errors from any of the Linux commands or scripts are captured and published as an error event related to this job and stored in COMPUTE_JOB_EVENT table.
f) Once the SAS program is sumitted successfully to the SAS grid, then a submitted event is published for this job and stored in COMPUTE_JOB_EVENT table.

1.4.6 Electronic Signature

The following steps are an overview of the actions performed by the system to electronically sign a file. The logic is implemented in the com.lilly.cluwe.webapp.service.impl.SignatureServiceImpl class with the details regarding each step and associated error handling in the signFile method. The DynaSignController is responsible for verifying the credentials of the signature before the following steps are performed.

a) Record the current time and date as the approval date of the document(s).
b) If a more than one file was selected, perform the following steps for each file.
c) Check if the file is in the /prd directory structure. If not, update the signature log entry and return an error.
d) Check if the user has read access to the file. If not, update the version log entry and return an error.
e) Create a signature log entry with the Electronic Signature information.
f) Set the value of the “user.eSignatureId” metadata entry on the file to the transaction id.
g) Set the permissions of the file to read-only using the File Locking algorithm described below.

1.4.7 Displaying Files and Folders

The following steps are an overview of the actions performed by the system to display the contents of a folder on the Isilon NAS. The logic is implemented in the com.lilly.cluwe.webapp.data.impl.IsilonFileRepository class with the details regarding each step and associated error handling in the getChildren methods. The DynaFileTreeController and FancyTree javascript tree control are responsible for converting user actions on the File Browser such as open folder to parameters required by the getChildren method. The getChildren results are then interpreted by the DynaFileTreeController and the tree control for display.

a) Call the Isilon Query web service to return the contents of the folder passed in from the DynaFileTreeController. The authorization header for this web service call is populated with the credentials the user entered in the system’s login page so the results from the web service reflect the permissions applied within the Isilon.
b) Iterate through the results of the query performing the following actions for each object.
    a) If the object is a Folder, create a Folder object populated with the metadata values returned from the query. Move to the next object.
    b) Create a File object populated with the metadata values returned from the query.
    c) Set the isSignable flag on the File object to true if it’s in the /prd branch and not in a prior versions folder.
    d) Set the isVersionable flag on the File object to true if it’s in the /prd branch and not in a prior versions folder.
    e) Set the isExecutable flag on the File object to true if the filename extension is “sas”.
c) Return the collection of File and Folder objects to the DynaFileTreeController for processing.

1.4.8 Automated File Versioning(AVA)

The following steps are an overview of actions performed by system to automated versioning. The logic is implemented in com.lilly.cluwe.service.ava.impl package. Whenever AVA service receives any request:

a) The request must contain the directory path, userId, JobId and transactionId.
b) AVA service compares each folder and file that we have in staging directory (/lillyce/.staging/{userId}) depending upon the request received. This comparison is done with the destination directory (where the output is expected) which depends on the SAS program.
c) As a part of the request AVA determines the root directory and traverse through all of its children.
d) If the child is a folder then check whether the folder exists in the destination directory. If yes, then iterate through each child and repeat the same process.
e) if the folder does not exist in the destination directory, check whether the user who scheduled the job has write permission in the destination directory. If yes, then simply add the folder to move list so that it can moved from staging directory to destination.

f) If the child is a file, check whether the user who scheduled the job has write permission in the destination directory. If the user has write permission, then check whether the file exists or not in destination directory.

g) If the file exists in the destination directory, add it to the version process List and move list. If not, then only add it to move list.

h) Version the files in the version list then after move the files in move list.

i) After moving each file, set the permission identical to its parent.

j) After setting the permission, lock each file.

1.5 External Resources

1.5.1 Linux sudo Command

The Linux sudo command must be enabled by the Lilly Global Service Delivery - Linux team for use by the Job Scheduling functionality. The sudo entry required is:

```
cluwe-webui cluwe-p1.am.lilly.com,cluwe-p2.am.lilly.com=(%sas)
NOPASSWD:/od/cluwe/sasgsub/submitsasjob.sh
```

1.5.2 Isilon NAS

The Isilon NAS is used by the Global Statistical Sciences team to store SAS programs, SAS datasets as well as other Clinical Trial results related artifacts. It is a clustered environment with a second cluster configured to act as a backup and managed by the Lilly Global Service Delivery – Storage team as they manage the other Isilon environments. The CLUWE program has requested that the following features be enabled:

- Merging of Windows and Linux permissions to ensure the same authorization rules are enforced regardless of whether a user accesses the NAS from Windows or Linux. The feature requires that users have both an Active Directory account and a Linux LDAP account or they cannot access the NAS. The ability to change file permissions from UNIX was also disabled to prevent possible user confusion when viewing Linux based changes from Windows.
- Auditing of all read, write, permission change and delete events to provide traceability for all file system activities.
- Isilon Platform API access described in a later section.
- Prevent users from change permissions of files they own. This default Windows and Linux behavior was disabled to prevent users from change the status of read-only files generated by the SAS Grid on the user's behalf.

1.5.3 RedHat jBoss EAP

JBoss Enterprise Application Platform (JBoss) provides the Java 8 Java Virtual Machine (JVM) and Java Server Pages (JSP) compliant container required to host the CLUWE Web Tool. This platform is supported by the Lilly Web Hosting Organization on servers dedicated to the system. The following changes to the base JBoss service are required:

- JBoss must run under the cluwe-webui LDAP account since that account must access the Isilon file system.
- The java option "jsse.enableSNIExtension" set to false to allow web service calls to the Isilon PAPI which uses a self-signed Secure Socket Layer (SSL) certificate.
- A Java Naming and Directory Interface (JNDI) datasource named: "java:jboss/datasources/CluweDataSource" that connects to the Oracle database.

1.5.4 SAS sasgsub command-line utility

The /sasfs file system exported by the SAS Grid must be mounted on the system web servers so Quartz can execute the sasgsub command-line utility. This utility submits sas program files to the SAS Grid for execution. Upon successful submission it returns the following:

a. SAS Grid Job ID
b. Location of the SAS program's log file
c. Location of the work directory assigned by SAS

1.6 External Interfaces

1.6.1 Versioning Web Service

These two RESTful web services are provided by the CLUWE Web Tool to expose the file versioning logic to other applications. Two flavors of the service exist but both of them use the same internal methods as the CLUWE Web Tool User Interface (UI). The first is exposed for other applications. The second service is exposed specifically for the Migrate to CLUWE (M2C) application as it must version files regardless of location or file permissions. Both services use base 64 encoded credentials in the Hyper Text Transfer protocol (HTTP) authorization header to authenticate the client.

RestController	com.lilly.cluwe.webapp.rest.RepositoryServiceController
URL	https://cluwe.am.lilly.com/rest/repository/version/v1/<path to file to version>
HTTP Verb	POST
Return Value	HTTP status codes are returned with text explaining error conditions if they occur.
Parameter Name	**Description**
userid	Optional. The id of the user for which a version is to be created. If not specified, the id from the authorization header is used.
transactionid	Optional. A client generated identifier that can be used to track transactions between the client and the web service logs.

1.6.2 Repository Web Services

This RESTful web service is provided by the CLUWE Web Tool to expose a basic set of file operations. This service uses base 64 encoded credentials in the HTTP Authorization header to authenticate the client.

RestController	com.lilly.cluwe.webapp.rest.RepositoryServiceController
URL	https://cluwe.am.lilly.com/rest/repository /v1/<path to file or folder>
HTTP Verb	GET
Return Value	HTTP status codes are returned with text explaining error conditions if they occur. If command parameter is not included, the metadata values for the object are returned in JSON format if the object exists. If canread or canwrite command is specified an HTTP 200 OK is returned if the result is true. Otherwise, an HTTP 403 Forbidden is returned.
Parameter Name	**Description**
command	Optional. When value is “canread” the service checks whether the specified user can read the contents of the file or folder. When value is “canwrite” the service checks whether the specified user can modify the contents of the file or folder.
userid	Optional. The id of the user for which the “canread” or “canwrite” command is executed. If not specified, the id from the authorization header is used.
transactionid	Optional. A client generated identifier that can be used to track transactions between the client and the web service logs.

The following service allows the client to change the owner of a file.

RestController	com.lilly.cluwe.webapp.rest.RepositoryServiceController
URL	https://cluwe.am.lilly.com/rest/repository /v1/<path to file or folder>
HTTP Verb	POST
Return Value	HTTP status codes are returned with text explaining error conditions if they occur.
Parameter Name	**Description**
command	Required. When value is “setowner” the service changes the owner of the file to the value of the userid parameter.
userid	Optional. The id of the user for which the “canread” or “canwrite” command is executed. If not specified, the id from the authorization header is used.
transactionid	Optional. A client generated identified that can be used to track transactions between the client and the web service logs.

1.6.3 SDD Migration Data Web Service

This service is exposed specifically for the M2C application to store data about files migrated from SDD. Unlike the other web services provided by the CLUWE Web Tool, this service uses no custom code as the functionality is provided by Spring Data Rest annotations on the ISddRepository class. The service uses exposes the URL:

https://cluwe.am.lilly.com/api/m2c/<sdd file id>

This service uses base 64 encoded credentials in the HTTP Authorization header to authenticate the client. HTTP verbs define the operations to perform.

HTTP Verb	Operation
GET	Retrieve the specified SDD file record with associated metadata and eApprovals.
POST	Create a new SDD file record with included metadata and eApproval values. The body of this message must be in JSON format.
DELETE	Delete the specified SDD file record and associated metadata and eApprovals.

1.6.4 Oracle

Oracle 12c serves as the persistence layer for the system. Database access and structure is detailed in the Database Design section of this document.

1.6.5 Isilon Platform API

The Isilon Platform API (PAPI) consists of a set of web services hosted from the Isilon NAS. The CLUWE Web Tool uses the PAPI to move files, retrieve the contents of folders, manage file metadata and manage permissions. All actions are authorized by the same permission model applied to the file system such that files that aren't available to user from the file system cannot be accessed by the same user through the PAPI. When access to all files and folders is required, a system account with expanded permissions over the file system is used when authenticating against the PAPI.

1.6.6 JReview

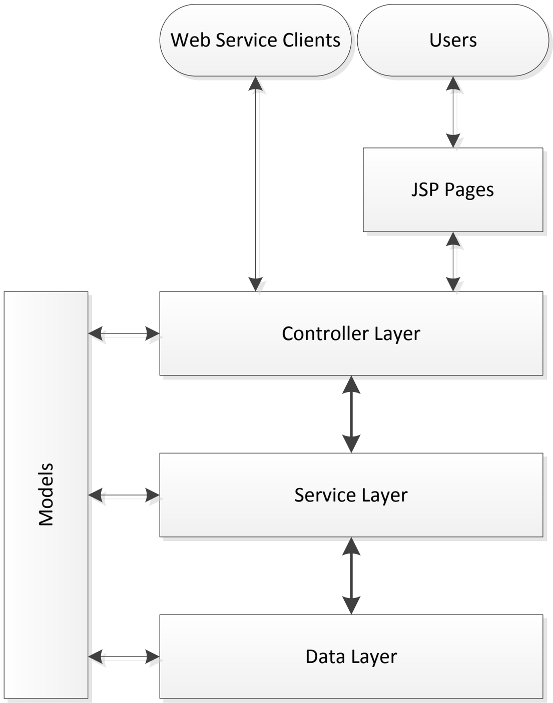
The JReview system is unable to interpret the Windows or NFS permissions applied to the Isilon file system so a process was setup to export the permission information to JReview in a form that it could ingest. The process is executed from Quartz every two hours after which the following steps are performed:

1. Submit the SAS job specified in the 'jReview.study.reg.sas.path' property to the SAS Grid. This job reads the studies registered in JReview then exports the list to a .csv file, the location of which is specified in the 'jReview.study.reg.csv.path' property.
2. Open the .csv file from step 1 and loop through the folders in the file performing the following steps for each:
    a. Query the Isilon for users and groups with at least Read permission on the folder.
    b. Query Active Directory for the members of any groups used to apply permissions to the folder excluding groups listed in the 'jReview.groups.to.exclude' application property.
    c. Export the list of users to the file specified in the 'jReview.study.reg.csv.path' property.
3. Submit the SAS job specified in the 'jReview.create.permission.dataset.path' property to the SAS Grid to convert the .csv file created in step 2 to a SAS dataset for JReview to ingest.

A SAS encrypted password file is used by the process in step 1 to access the JReview system. This password is stored in the /lillyce/.cluwe/jreview/file/pwd.txt directory which is limited to Read/Write access for only the cluwe_apps AD group (service accounts) and the cluwe_it_support group.

1.7 Internal Interfaces and Architecture

The application was designed with modularity, scalability and supportability as the primary design factors. The Model View Controller (MVC) pattern is implemented using Spring MVC. The service layer implements the business logic and the data layer provides the interfaces to the database and Isilon file repository.



As depicted in the diagram, each layer only communicates with the layer above and below it and the model objects are passed throughout. Spring Security provides the authentication and authorization functionality for all the layers.

1.8 Customization of Vendor-Supplied Software

Not applicable.

1.9 Process Automation System Design

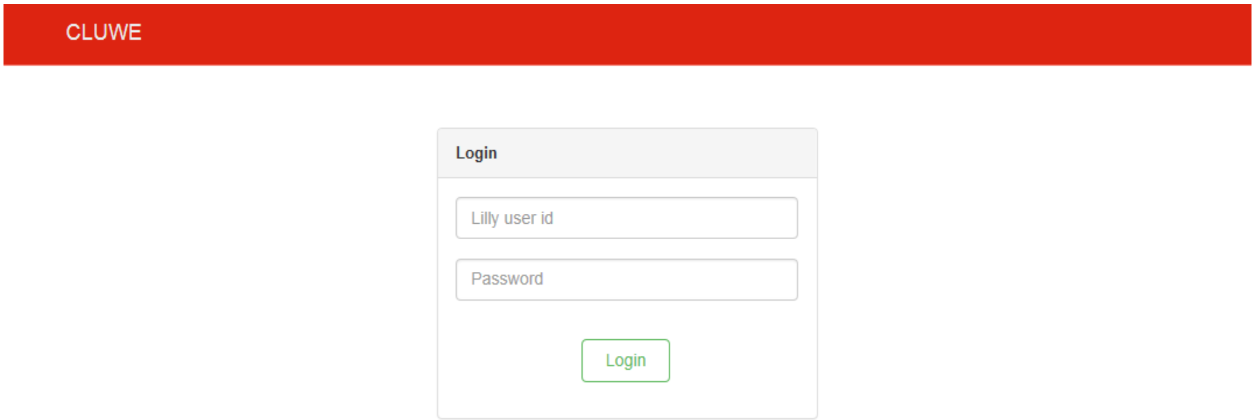
Not applicable.

1.10 Graphic User Interface (GUI) Design

Users interact with the system through a website hosted on two RedHat JBoss web servers. The user interface is built using Java Service Pages (JSP), HTML5, Bootstrap, jQuery and a web tree control named FancyTree. The pages and AJAX service calls are managed through Spring MVC. Internet Explorer 11 (IE) and Google Chrome 45 (Chrome) were used for development and supported for the users.

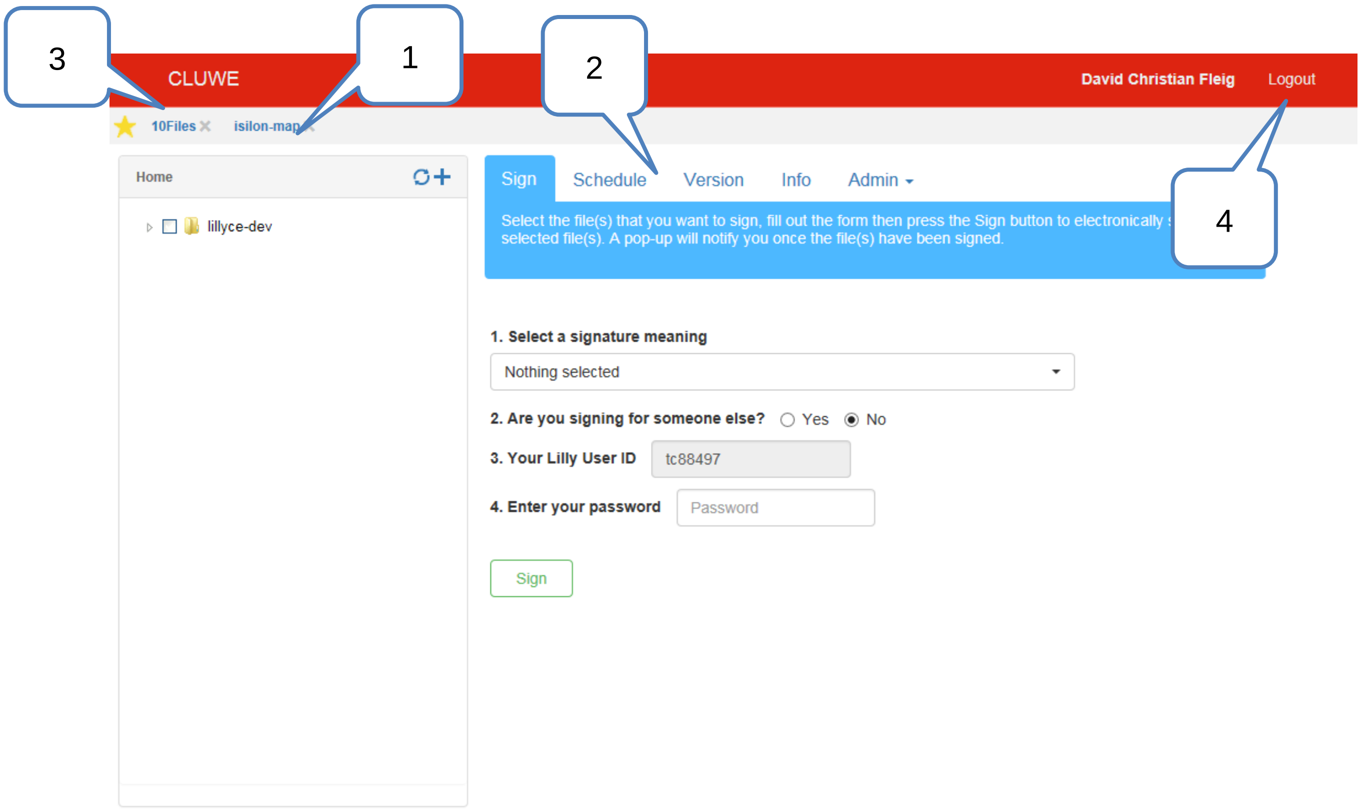
1.10.1 Login Page

The login.jsp page is where users enter their Lilly AD credentials to login to the system. The AD domain should not be included with the user id. Once the credentials have been entered, pressing the Login button will submit them to AD for verification and the page will either display an authentication failure message or redirect the user to the home page. This page is available to anyone.


CLUWE
Login
Lilly user id
Password
Login

1.10.2 Home Page

The Home.jsp page is where all the user actions take place. The left side of the screen contains a file browser that displays the files and folders available to the users from the Isilon Network Attached Storage (NAS) device. The right side is dedicated to a set of tabs which represent actions the user can take on the files selected in the file browser. Each time the user changes tabs on the right, the File Browser display on the left updates the files displayed based on whether it is applicable to the selected tab. Changing tabs also clears any selected files to prevent users from accidently selecting unwanted file in the new tab.



Select the file(s) that you want to sign, fill out the form then press the Sign button to electronically s
selected file(s). A pop-up will notify you once the file(s) have been signed.

1. Select a signature meaning

Nothing selected ▾

2. Are you signing for someone else? ○ Yes ● No

3. Your Lilly User ID tc88497

4. Enter your password Password

Sign

1. File Browser: See section File Browser section below.
2. Action Tabs: See Tab sections below.
3. Favorites Bar: See Favorites Bar section below.
4. Logout: This link logs the user out of the system. Automatic logout occurs after 15 minutes of inactivity.

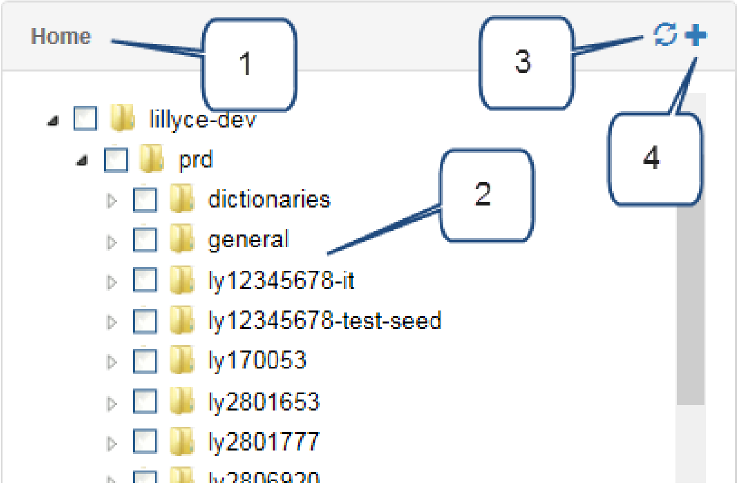
1.10.3 Favorites Bar

The Favorites Bar displays the favorites created by the user. Favorites can be removed by clicking the "X" icon to the right of the favorite. These entries are stored in the CLUWE_OWNER.USER_FAV table. This section of the home page is serviced by the DynaFavoritesController class.

1.10.4 File Browser

The File Browser represents the files available to the user through the Isilon NAS. The same security applied to the files and folders on the NAS are applied to File Browser view. This means that files and folders which the user is not allowed to see on the NAS are not displayed in File Browser. The contents of each folder are “lazy loaded” meaning the AJAX web service call to retrieve the folder contents only occurs when the folder is opened and on each subsequent opening. This results in a short delay in opening each folder while the web service call completes but prevents the web browser from crashing due to the large number of files and folders in the NAS. Additionally, Windows Thumbs.db files are not displayed.

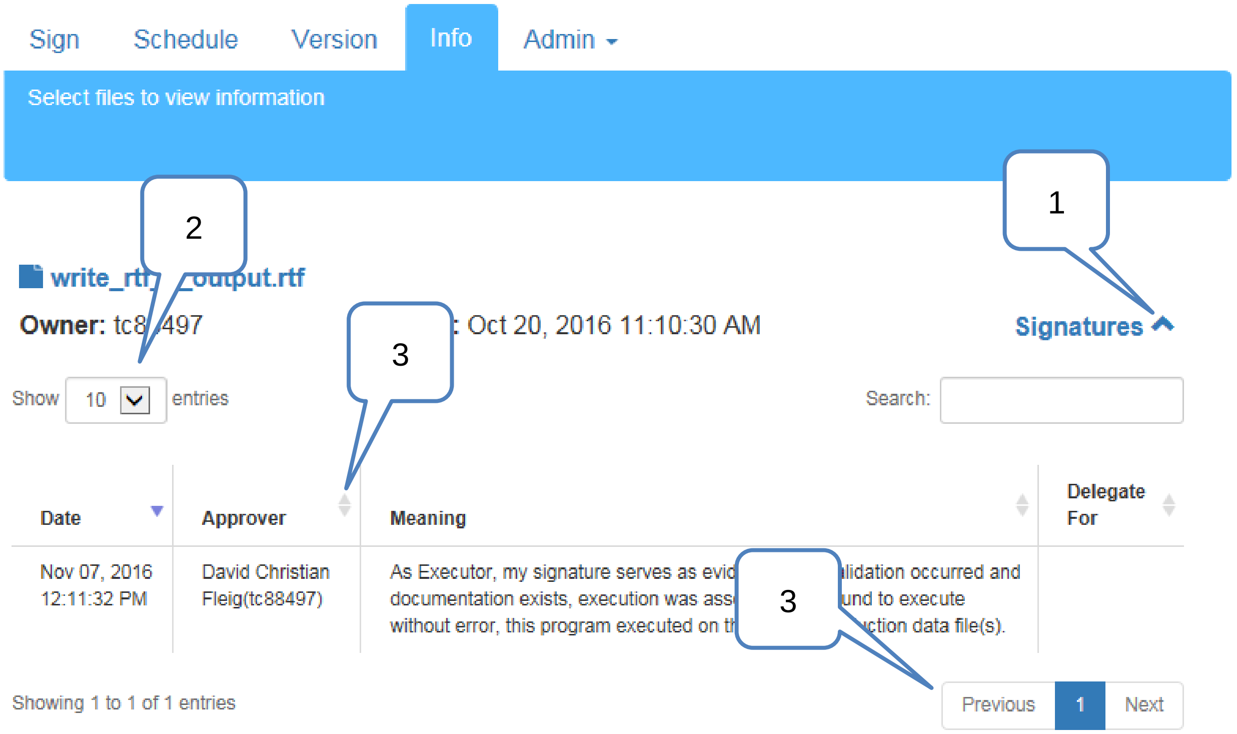
The process to retrieve and display the files is found in the 1.4.7 Displaying Files and Folders section of this document. This section of the page is serviced by the DynaFileTreeController class.



1. File Browser label. When a favorite is selected the name of favorite is displayed here.
2. Files and Folders available for selection and sorted by name.
3. Press this refresh icon to reset the File Browser back to Home view and reload the contents.
4. Press the plus icon to add a new favorite.

1.10.5 Info Tab

The Info tab is where users see information about the files selected in the File Browser. This tab is serviced by the DynaInfoController class.



Select files to view information

write_rtf_output.rtf

Owner: tc8 497 : Oct 20, 2016 11:10:30 AM

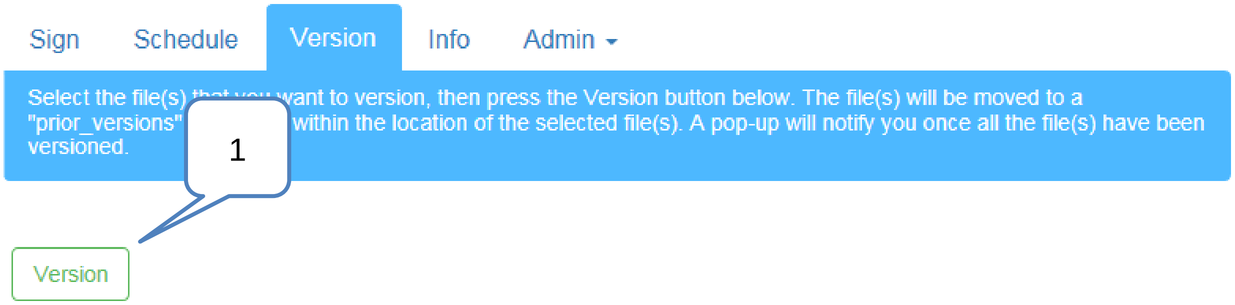
Date	Approver	Meaning	Delegate For
Nov 07, 2016 12:11:32 PM	David Christian Fleig(tc88497)	As Executor, my signature serves as evid lidation occurred and documentation exists, execution was ass und to execute without error, this program executed on th ction data file(s).	

Showing 1 to 1 of 1 entries

1. Select the arrow to open and close the Electronic Signature information for the file.
2. This dropdown is displayed for all grid controls and it controls the number of items displayed in the grid at one time.
3. The sort controls are displayed for all grid controls and it allows the user to sort the columns by value.
4. The paging control is displayed for all grid controls and it allows the user to browse through multiple pages of grid data.

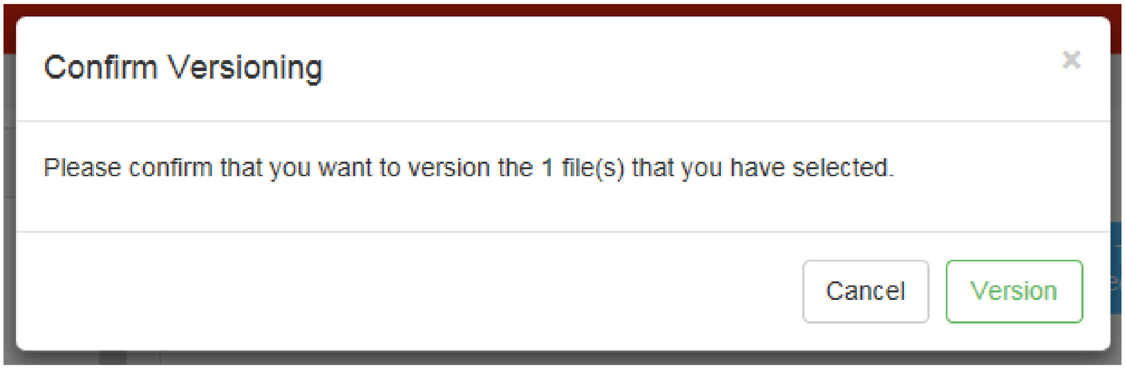
1.10.6 Version Tab

The Version tab allows users to version one of files selected in the File Browser. When pressed, the Version button starts the versioning process for the selected files. This tab is serviced by the DynaVersionsController class.



Select the file(s) that you want to version, then press the Version button below. The file(s) will be moved to a "prior_versions" within the location of the selected file(s). A pop-up will notify you once all the file(s) have been versioned.

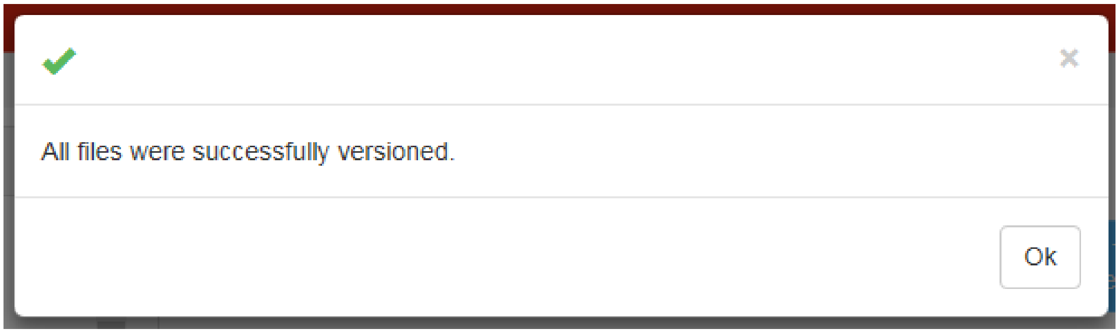
The following modal dialog box is displayed before the versioning operation starts.



Confirm Versioning

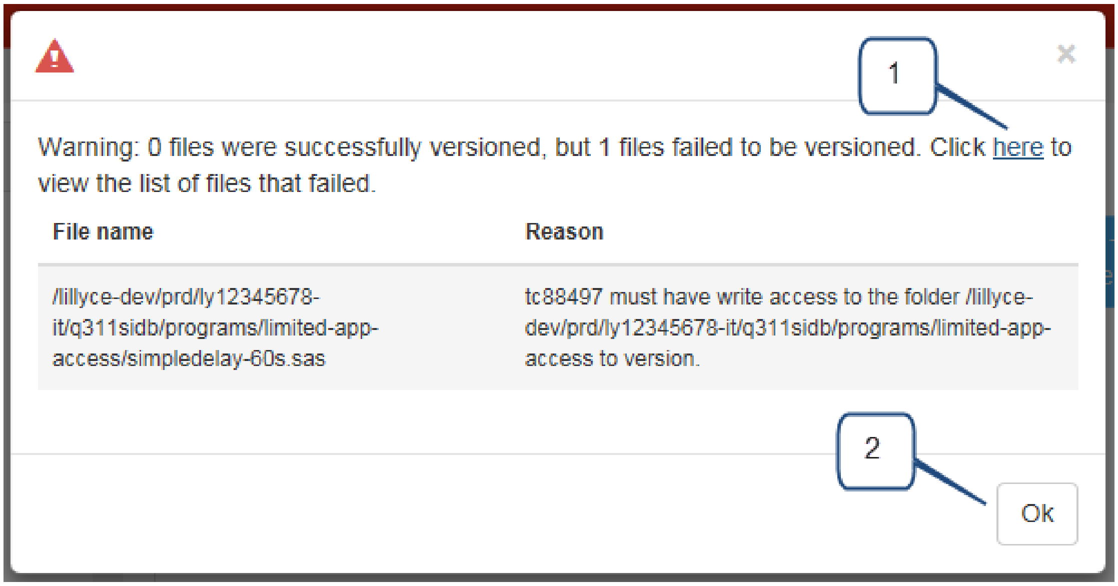
Please confirm that you want to version the 1 file(s) that you have selected.

If the operation completes successfully the following dialog is displayed:



All files were successfully versioned.

If the version operation completes with any errors, the following modal dialog is displayed to inform the user of the files on which the errors occurred.



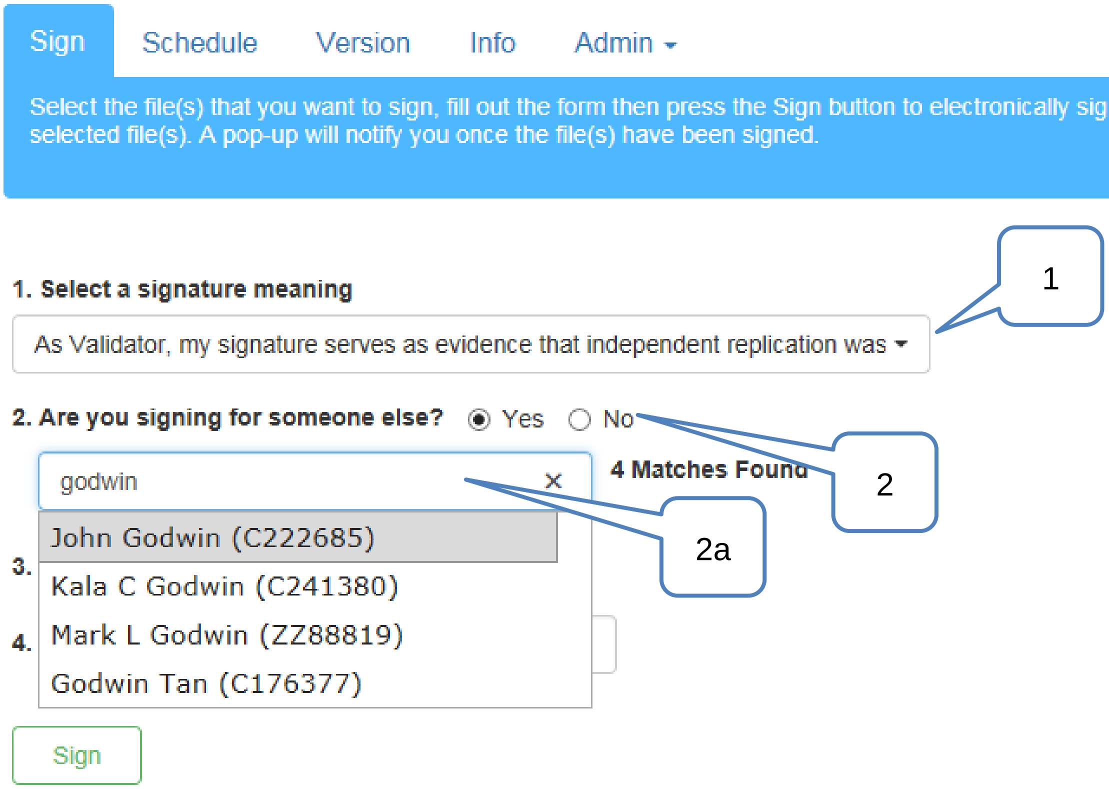
Warning: 0 files were successfully versioned, but 1 files failed to be versioned. Click here to view the list of files that failed.

File name	Reason
/lillyce-dev/prd/ly12345678-it/q311sidb/programs/limited-app-access/simpledelay-60s.sas	tc88497 must have write access to the folder /lillyce-dev/prd/ly12345678-it/q311sidb/programs/limited-app-access to version.

1. Press the “here” link to see the list of files with errors and the reason for the error.
2. Press the “OK” button to close the dialog.

1.10.7 Sign Tab

The Sign tab provides Electronic Signature capabilities to the users. This tab is serviced by the DynaSignController class.



Select the file(s) that you want to sign, fill out the form then press the Sign button to electronically sign the selected file(s). A pop-up will notify you once the file(s) have been signed.

1. Select a signature meaning

As Validator, my signature serves as evidence that independent replication was

2. Are you signing for someone else? ● Yes ○ No

godwin ✕

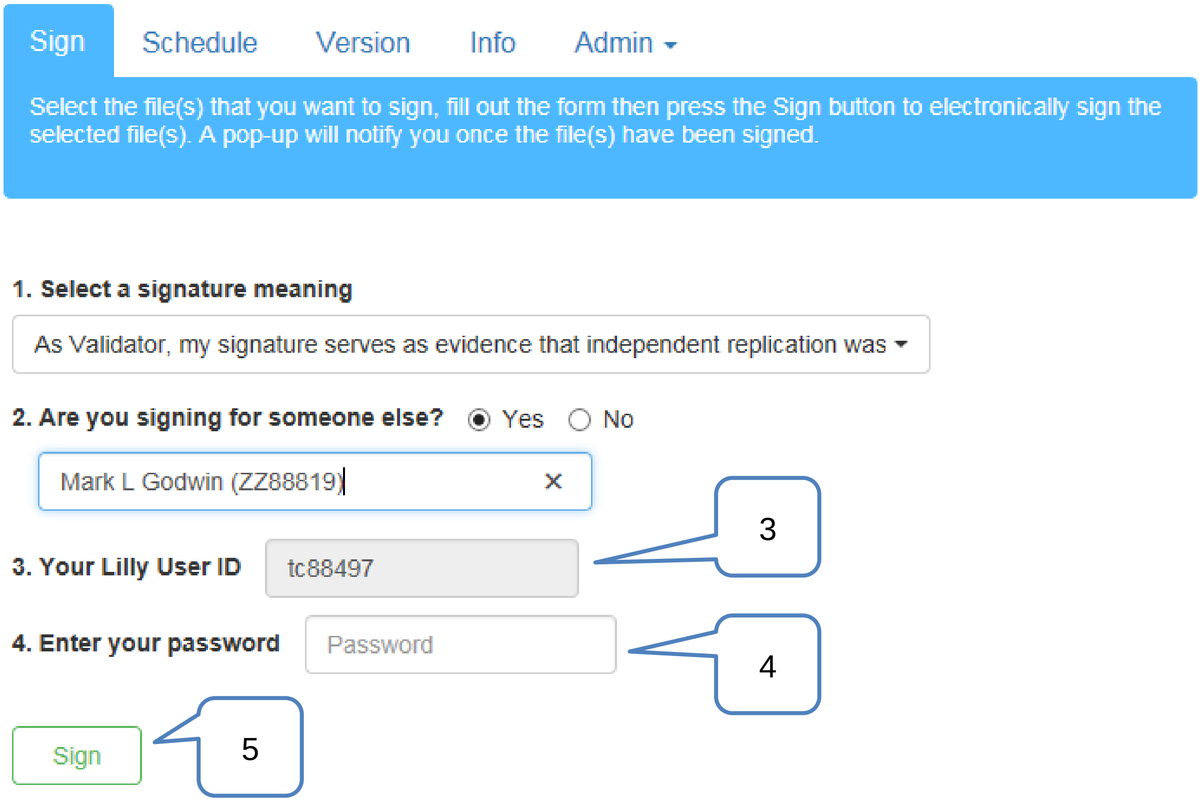
4 Matches Found

John Godwin (C222685)
Kala C Godwin (C241380)
Mark L Godwin (ZZ88819)
Godwin Tan (C176377)

3.

4.

1. Select the appropriate signature meaning. These values are loaded from the ESIG_MEANING table.
2. Select “Yes” or “No” to indicate whether this signatory is a delegate for another user. Selected “Yes” exposes the 2a control.
    a. The string entered into this text control is used to search AD for any users whose last name, first name or user id starts with the string. While typing, a pause of one or more seconds will result in matching names appearing just below the text box. Any of the names from the list can be selected.



Select the file(s) that you want to sign, fill out the form then press the Sign button to electronically sign the selected file(s). A pop-up will notify you once the file(s) have been signed.

1. Select a signature meaning

As Validator, my signature serves as evidence that independent replication was ▾

2. Are you signing for someone else? ● Yes ○ No

Mark L Godwin (ZZ88819) ✕

3. Your Lilly User ID tc88497

4. Enter your password Password

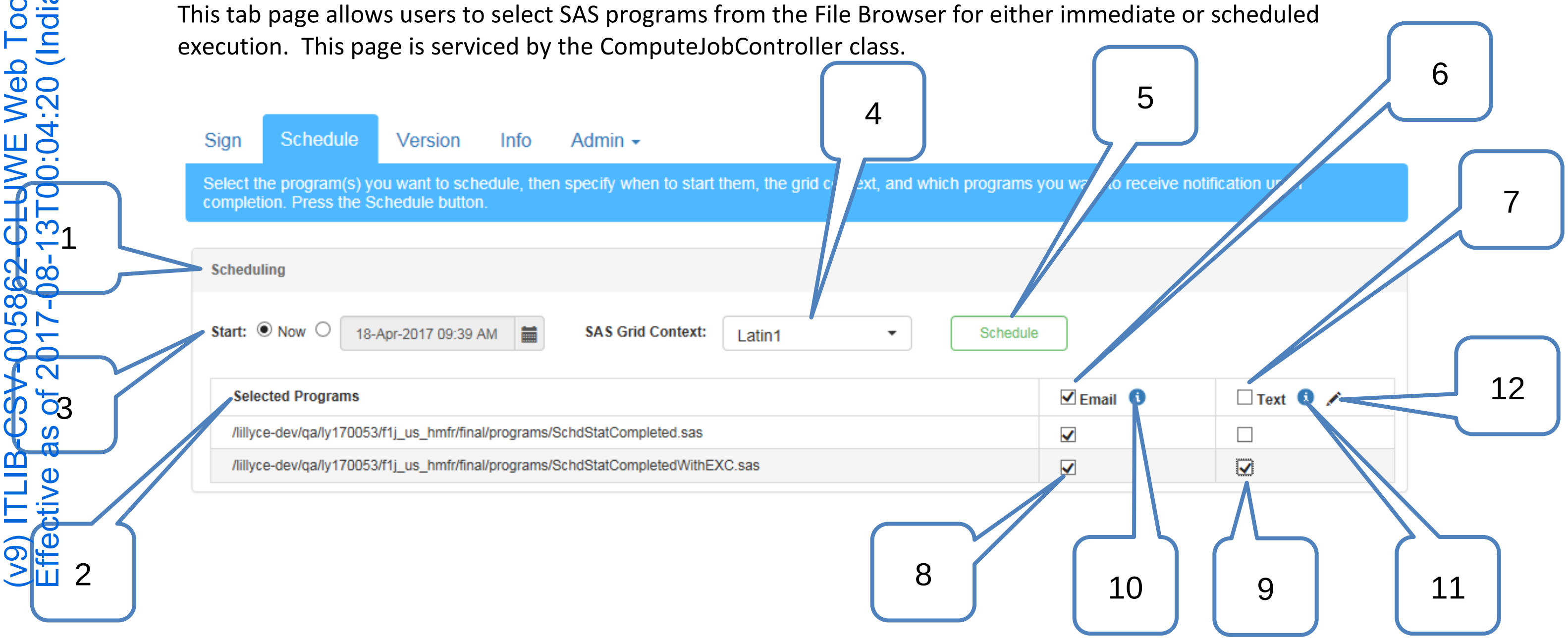
Sign

3. The ID of the logged in user is assigned here.
4. The user enters their password in this password control. The id and password are verified against AD when the Sign button is pressed.
5. Press the “Sign” button to start the Electronic Signature process.

The same verification and status dialog boxes used for the Version tab are also used for this tab with the only difference being version related text is replaced with signature related text.

1.10.8 Schedule Tab

This tab page allows users to select SAS programs from the File Browser for either immediate or scheduled execution. This page is serviced by the ComputeJobController class.



Select the program(s) you want to schedule, then specify when to start them, the grid c ext, and which programs you wa o receive notification u completion. Press the Schedule button.

Scheduling

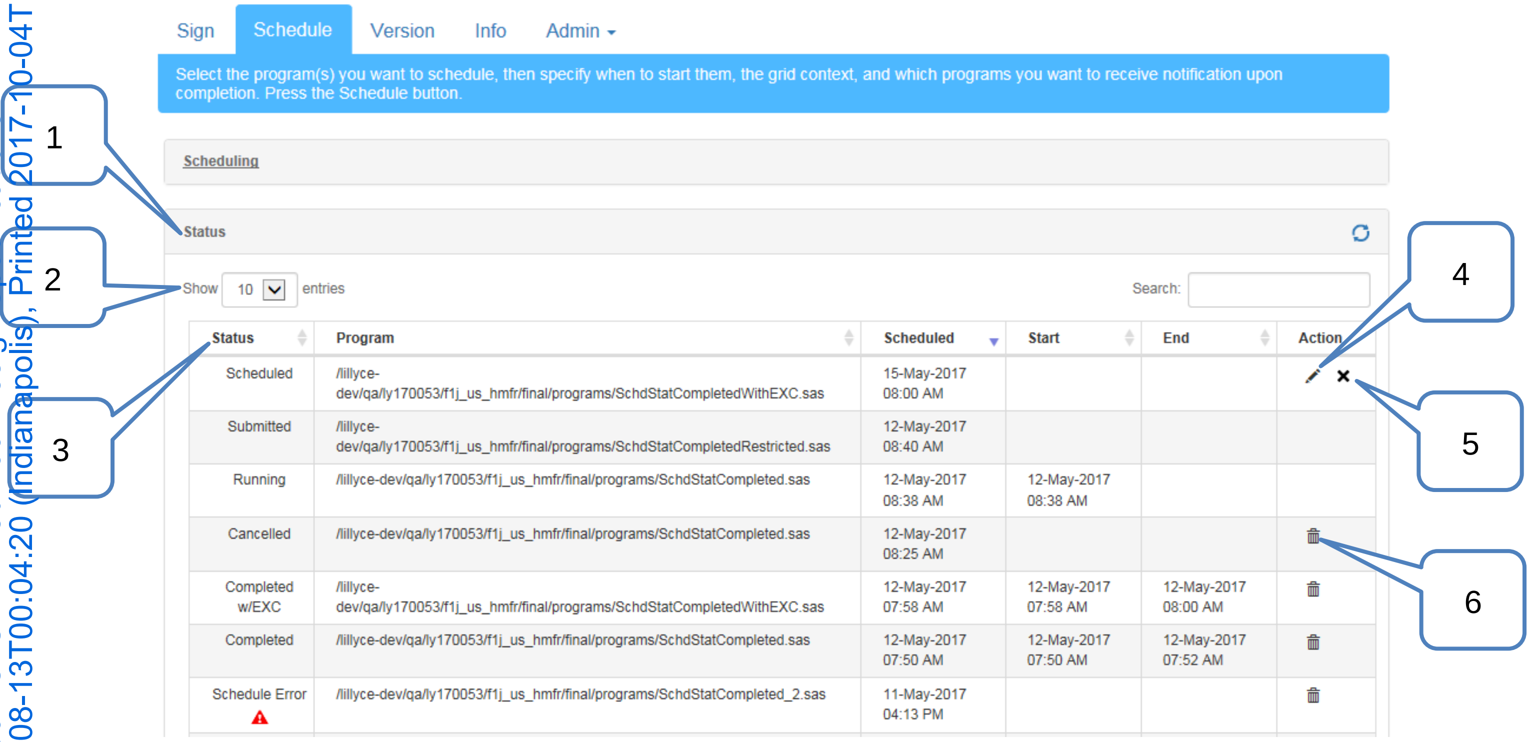
Start: ● Now ○ 18-Apr-2017 09:39 AM

SAS Grid Context: Latin1 ▾

Schedule

Selected Programs	☑ Email	□ Text
/lillyce-dev/qa/ly170053/f1j_us_hmfr/final/programs/SchdStatCompleted.sas	☑	□
/lillyce-dev/qa/ly170053/f1j_us_hmfr/final/programs/SchdStatCompletedWithEXC.sas	☑	☑

1. Click on the this to open and close the scheduling section.
2. This shows the list of selected programs to schedule.
3. Specify when to start the programs. Selecting the “Now” option will instruct the system to submit the job to the SAS Grid immediately upon clicking the “Schedule” button. When the date/time picker control is selected the user can choose a date and time for the job to be submitted to the grid. Only values in the future can be selected.
4. The list of SAS Grid Contexts available for job submission.
5. When pressed, the Schedule button starts the scheduling process.
6. Checking/unchecking this will enable/disable email notifications for all the programs selected.
7. Checking/unchecking this will enable/disable text notifications for all the programs selected.
8. Checking/unchecking this will enable/disabe email notifications for that particular program.
9. Checking/unchecking this will enable/disabe text notifications for that particular program.
10. Hover mouse over to see email address to which the notificaitons will be sent.
11. Hover mouse over to see the mobile number to which the notificaitons will be sent.
12. Clicking on this icon will open a modal that will enable you to enter an alternate mobile number to send text to.



Select the program(s) you want to schedule, then specify when to start them, the grid context, and which programs you want to receive notification upon completion. Press the Schedule button.

Status	Program	Scheduled	Start	End	Action
Scheduled	/lillyce-dev/qa/ly170053/f1j_us_hmfr/final/programs/SchdStatCompletedWithEXC.sas	15-May-2017 08:00 AM			✎ ✕
Submitted	/lillyce-dev/qa/ly170053/f1j_us_hmfr/final/programs/SchdStatCompletedRestricted.sas	12-May-2017 08:40 AM			
Running	/lillyce-dev/qa/ly170053/f1j_us_hmfr/final/programs/SchdStatCompleted.sas	12-May-2017 08:38 AM	12-May-2017 08:38 AM		
Cancelled	/lillyce-dev/qa/ly170053/f1j_us_hmfr/final/programs/SchdStatCompleted.sas	12-May-2017 08:25 AM			🗑
Completed w/EXC	/lillyce-dev/qa/ly170053/f1j_us_hmfr/final/programs/SchdStatCompletedWithEXC.sas	12-May-2017 07:58 AM	12-May-2017 07:58 AM	12-May-2017 08:00 AM	🗑
Completed	/lillyce-dev/qa/ly170053/f1j_us_hmfr/final/programs/SchdStatCompleted.sas	12-May-2017 07:50 AM	12-May-2017 07:50 AM	12-May-2017 07:52 AM	🗑
Schedule Error ⚠	/lillyce-dev/qa/ly170053/f1j_us_hmfr/final/programs/SchdStatCompleted_2.sas	11-May-2017 04:13 PM			🗑

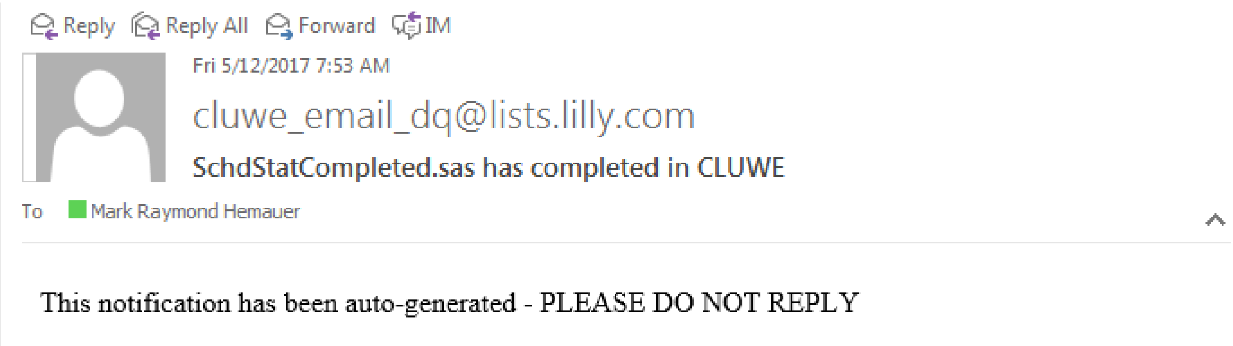
1. Click this link to open and close the Status grid which displays the status of all programs awaiting submission and those that have been submitted.
2. Shows the number of entries per page.
3. Shows the the status of the jobs. The job can have any of the following statuses

- Schedule Error

- Scheduled
- Cancelled
- Submit Error
- Submitted
- Running
- Completed
- Completed w/ EXC

4. Click the icon to edit schedule date/time of the job.
5. Clicking on this icon to cancel scheduled program.
6. Clicking on this icon will delete the row from the status list.

The email and text notifications are as shown below:



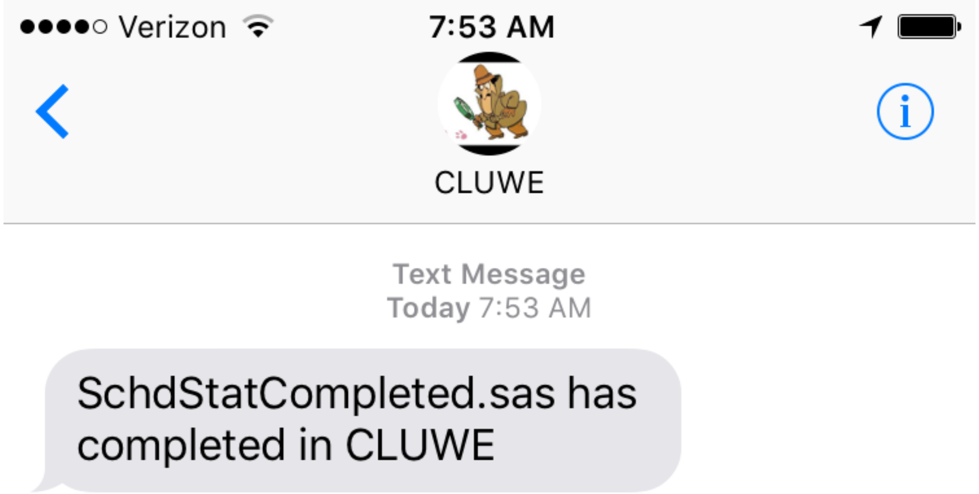
Fri 5/12/2017 7:53 AM

cluwe_email_dq@lists.lilly.com

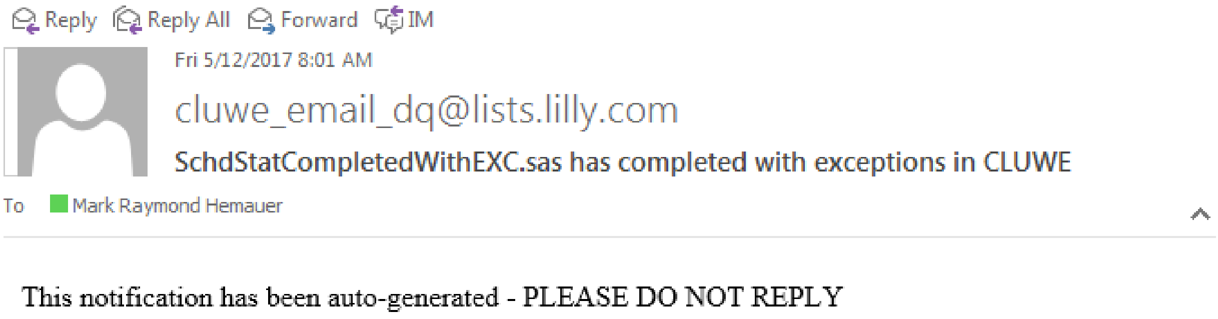
SchdStatCompleted.sas has completed in CLUWE

To 🟩 Mark Raymond Hemauer

This notification has been auto-generated - PLEASE DO NOT REPLY



SchdStatCompleted.sas has completed in CLUWE



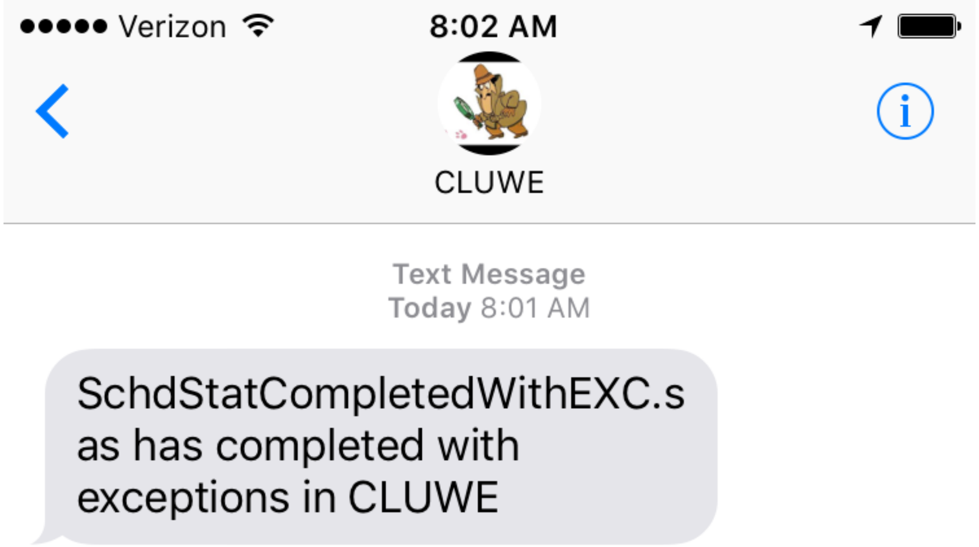
Fri 5/12/2017 8:01 AM

cluwe_email_dq@lists.lilly.com

SchdStatCompletedWithEXC.sas has completed with exceptions in CLUWE

To Mark Raymond Hemauer

This notification has been auto-generated - PLEASE DO NOT REPLY

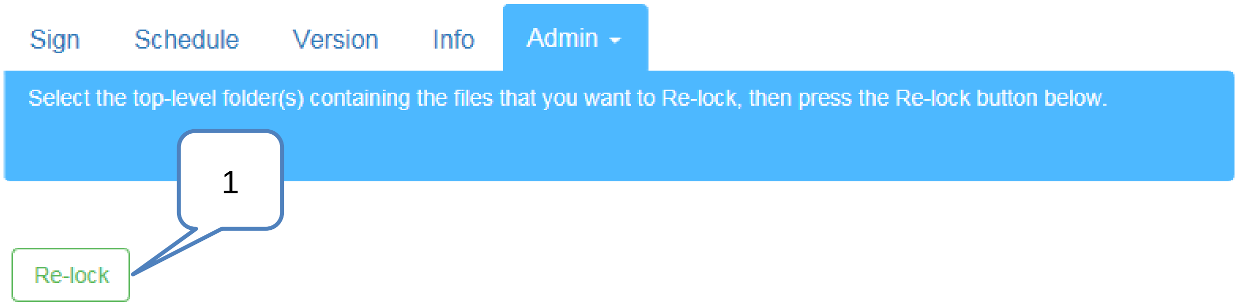


Text Message
Today 8:01 AM

SchdStatCompletedWithEXC.sas has completed with exceptions in CLUWE

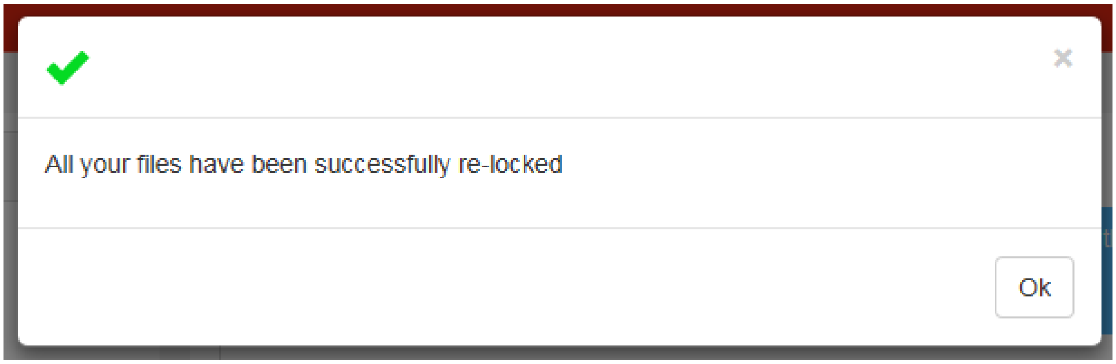
1.10.9 Admin Tab

The Admin tab allows Business Support Team members to re-lock files starting at the folder selected in the File Browser. When pressed, the Re-Lock button starts the re-lock process for all files and prior_versions folders under the selected tab. This tab is serviced by the ProvisionController class.



Select the top-level folder(s) containing the files that you want to Re-lock, then press the Re-lock button below.

If the operation completes successfully the following dialog is displayed:



All your files have been successfully re-locked

1.11 Reports

File and Folder open, read and delete events are stored in the Isilon NAS log files for review. The process to request this data is documented in the *CLUWE Web Tool System Admin and Support SOP*.

No reports were implemented. See the Graphical User Interface section for web page design and functionality.

1.12 Data Conversion/Migration

The Version and SDD Migration web services exposed by the CLUWE Web Tool are utilized by the M2C application which does perform data migration. Refer the *M2C Validation Plan* for detail of data migration.

1.13 Data Maintenance Activities

Data maintenance activities are documented in the *CLUWE Web Tool System Administration and Support SOP*.

1.14 Batch Processing

No batch processing is performed by the system.

1.15 Security

Security design and implementation is documented in the *CLUWE Web Tool Security Plan* and *CLUWE Web Tool Security Administration SOP*.

1.16 Error Handling

Error handling is performed using standard Java error handling patterns.

1.17 Alarm Handling

No alarm handling is performed by the system.

1.18 Electronic Record Audit Trails

The Electronic Signature functionality maintains audit logs within the ESIG_LOG table. Refer to the Database Design section of the document for table details. Security for Electronic Records is detailed in the *CLUWE Web Tool Security Plan*.

1.19 Electronic Signature

The Electronic Signature functionality provided within the Sign tab can be applied to any file within the /prd branch of the Isilon file system as long as it has not been previously versioned. The algorithm behind this functionality is described in the Electronic Signature portion of the Algorithms section of this document.

2. TRACEABILITY

The *CLUWE Web Tool Requirements Traceability Matrix,* present in Quality Center (QC), provides proof that requirements are traceable to design and testing.

3. REFERENCES

A current list of system-specific documents stored in Regulus can be generated using Regulus. The documents are stored in the IT Library Regulus repository at:

IT_Library > LRL IT > Systems C to D > CLUWE

IT_Library > LRL IT Secure > Systems C to D > CLUWE