##### 

|  |
| --- |
| *Kellogg Soil Survey Laboratory (KSSL)* |
| **Laboratory Information Management System (LIMS)** |
| National Soil Survey Center  Natural Resources Conservation Service  Lincoln, NE  05/14/2020  Version 1.0 |

##### 

# Revision History

| Version # | Implemented By | Revision Date | Approved By | Approval Date | Description of Change |
| --- | --- | --- | --- | --- | --- |
| *1.0* | Rick Nesser | 05/14/2020 |  |  | Initial Version |

# Table of Contents

Revision History 2

Table of Contents 3

1 Laboratory Information Management System (LIMS) 5

1.1 Overview 5

1.2 System Requirements 5

2 Database Detail 7

3 Workstation Application 7

4 Calculation Engine 8

5 Reporting System 8

6 Relationship to NCSS Lab Data Mart (NCSS LDM) 15

Appendix A – Acronyms 17

Appendix B – Setting Up the Server 17

Appendix C – Setting Up SQL Server 19

Appendix D – LIMS Calculation Engine 17

Appendix E – GnuPlot 17

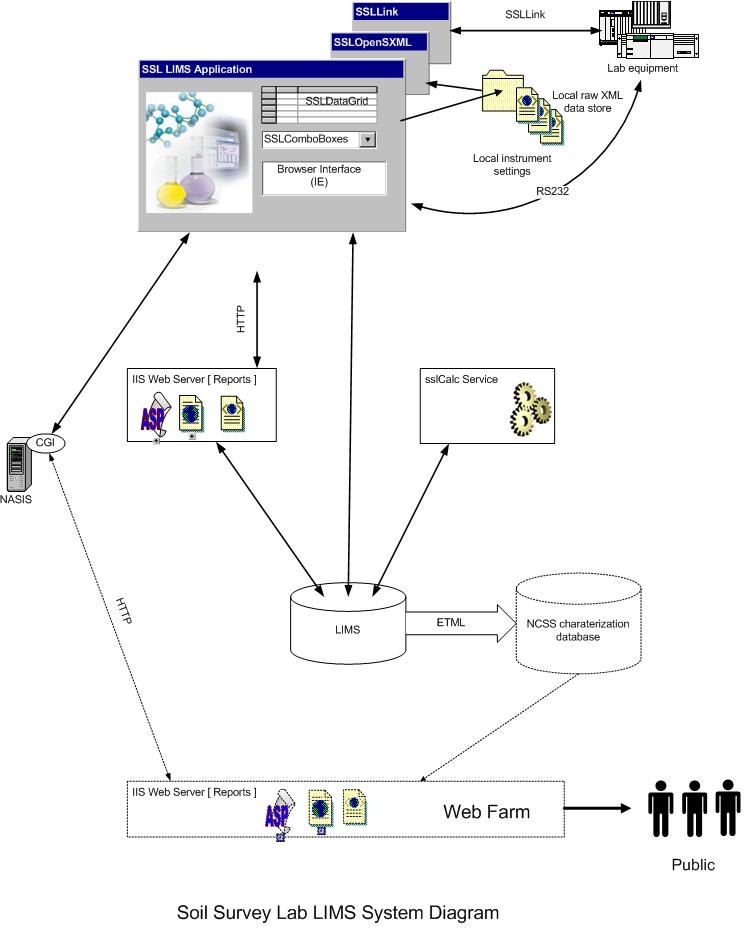
Appendix F – Day-to-Day Monitoring and Maintenance 17

# Laboratory Information Management System (LIMS)

The primary purpose of LIMS is to provide quality soil information to NRCS field offices and conservation partners in support of the National Cooperative Soil Survey or other end-products. The primary purpose of the LIMS application suite is to manage the data regarding reception, analysis, and reporting of soil samples. Information about the data is used to inform KSSL employees of work accomplished and work to be scheduled. The LIMS system is made up of components which will be described in subsequent chapters.

## Overview

The KSSL LIMS is pictured below. The gray boxes represent KSSL LIMS software running on KSSL workstations, both employee workstations and instrument-connected workstations. The boxes representing Lab equipment with lines drawn between them and the gray boxes indicate that data flows between the KSSL workstations connected to instruments either directly, via a RS-232 connected cable, or KSSL LIMS integrated software from files prepared by the instrument software. Below this are two white boxes that represent the KSSL LIMS reporting service, provided by a web site on the KSSL LIMS server, and the KSSL LIMS Calculation Engine, provided by a computer program on the KSSL LIMS server. Below these is the KSSL LIMS database, also on the KSSL LIMS server. To the left of this is a gray box that represents the NASIS (National Soil Information Service) web site. This indicates that KSSL LIMS workstation software downloads data from this web site. To the right of the KSSL LIMS database is the Characterization database, also called the Repository or NCSS Lab Data mart database. Completed projects of the KSS LIMS are copied in this database which is a collection of lab data gathered by various USDA and university laboratories (members of the National Cooperative Soil Survey). This data base is publicly accessible via a web site in a web farm, as indicated in the box below the database symbol. The documentation for the NCSS Lab Data Mart is found in its own document.



## System Requirements

KSSL LIMS requires a computer server with the latest Windows Operating System offered and supported by USDA FPAC CEC (Client Experience Center). This server should be provisioned with a minimum of 4 CPUs at 2.6 GHertz speed, 2 Tbytes disk space, and 16 Gbytes RAM.

In order to assist in the maintenance and configuration of the system and the software running on it, at least one employee of the NSSC must have a special administration Active Directory account. Currently, that account is identified using the letters ‘rc.’ Prefixing the employees usual Active Directory account. That special administration account must have the ability to alter operating system roles and features, administrate Internet Information System services, update Windows Firewall rules, modify permissions on disk drive folders, identify folders as network shares, install and configure third-party and internally written software, etc.

Certain Roles and Features must be configured in the system to support functionality used by KSSL LIMS reporting. Those Roles and Features are detailed in Appendix B. Internet Information Service Management services must be turned on and configured and is detailed in Appendix B. Remote Desktop services must be turned on and configured to allow the NSSC employee with administrative rights the ability to login remotely from their workstation. Windows Firewall has certain rules to protect the server from being attacked by hackers and other rules to permit certain activity. These rules, listed in Appendix B, must be present (and may need to be created) for KSSL LIMS to operate. In addition to the rules, CEC must turn on network connections between the server and the range of IP network addresses to which these rules apply, described in Appendix B.

The principal software installed should be Microsoft SQL Server, latest version. At the time this software is installed, the NSSC employee with administrative rights must have a SQL Server login created that has system administrative rights so that the KSSL databases may be created and restored, jobs created and activated, etc. The details of these jobs are indicated in Appendix C.

There is in-house written software, the LIMS Calculation Engine, that must be copied into a folder on the server. How this software is set up and run is detailed in Appendix D. There is third-party freeware software, GnuPlot, that must be copied into a folder on the server. How this software is set up and run is detailed in Appendix E.

KSSL LIMS also requires one or more workstations to run the software that interacts with the server. These workstations may be desktop or laptop computers. The minimum system requirements do not go beyond the minimum system provisioned by the CEC, this includes any requirements by instrument software. The KSSL LIMS workstation software is in-house written and is compiled based on the .NET framework. No extra software is required.

# Database Detail

The KSSL LIMS has a database as its foundation. Though it was designed around the needs of a working laboratory, data is the foundation of recording laboratory data, conveying analysis results, preparing reports for quality control, indicating when certain actions have been completed, indications of the status of work, etc. KSSL LIMS, then, becomes a communication system.

The KSSL LIMS database is prepared using a DBMS (data base management system). The current one is Microsoft SQL Server. Language describing the database will be using terms that can be associated with DBMSs in general unless specific SQL Server terms must be used.

The KSSL LIMS database is made up of tables, columns, key columns, indexes, triggers, views, stored procedures, and functions. These database items work together and are defined for a purpose. The KSSL LIMS workstation application relies on these database items to be there and working. Database administrators use SQL Server Management Studio (SSMS) to monitor and alter the database definitions. This software must be requested of CEC employees for installation. Altering the database definitions via SSMS may require altering the content of certain tables so that the workstation application will continue to function properly.

## Security

The KSSL LIMS database contains PII (Personably Identifiable Information) which must be protected. Also, the nature of laboratory analysis results is that, until the data is verified accurate and complete, it should not be distributed and therefore must be protected. Yet data must be accessible if KSSL employees are to do their work. Authentication is a must to manage protection and accessibility.

KSSL LIMS uses a simple authentication scheme to reduce management. Rather than have each employee maintain their own login account and password, KSSL LIMS has a non-domain account (lims\_application) defined in the database used in connecting to the database and then uses the employee’s Windows Active Directory account as the authentication. This way the database administrator maintains only one database login and the database tables that identify the employees permitted to run the workstation application.

There are three tables involved: person, ssl\_user, and ssl\_user\_role. The person table has the PII. Its columns are typical contact information with three being an exception: project\_submitter\_flag, project\_coordinator\_flag, and ssl\_employee\_flag. The three flags are used by the workstation application to drive choices when associating a person with projects. In the case of security, the ssl\_employee\_flag must be set ‘on’. The ssl\_user table links to the person table. Most of its columns are not used with one as the exception: windows\_name. For the employee to be authenticated during login, the windows\_name column must have the Active Directory domain name of the employee. The ssl\_user\_role table links with the ssl\_user table. Every user must have at least one role. The role\_type column is most important as it establishes whether the user may alter a project’s status or not. The ssl\_user\_lab\_section is populated for ‘Tech’ role type to narrow which sections of the lab they need to work in. The workstation application uses the values to display analysis batches for analyses performed in the sections of the KSSL.

## Projects

Government entities, private companies, and others submit soil samples to the KSSL for analysis for a purpose. Most such submissions are from USDA NRCS Soil Survey Offices in for characterization in support of the National Cooperative Soil Survey. A submission of a group of related soil samples is called a ‘project’. The KSSL LIMS database has a table named ‘project’ to hold the meta data pertinent to a project. Some columns in this table document the meta data while others drive the workstation application.

The columns that drive the workstation application are: proj\_status – determines when certain actions take place when the status changes, i.e. when a project status is changed to ‘complete’ it is then locked against changes and the contents are copied to the SSL\_Repo database; proj\_priority – used in sorting the list of projects and work in the KSSL; private\_flag – determines if a completed project is copied to the SSL\_Repo database; project\_folder\_path – used in storing reports generated by the workstation application.

The project\_contact table is used to store the persons that should be contacted in regards to a project. The workstation application makes a choice list from the person table

## Sites

Sites are locations where the soil samples submitted for a project are from. This information is stored in the lims\_site table and are identified by the column user\_site\_id. The user site id can be whatever the sample submitter needs to relate to the physical location. Typically, it is in a standard form that includes the year the samples were taken, the state FIPS code, the county FIPS number, and the sequence number of site within the year and county. An example would be S01KS-077-088 representing the year 2001, Kansas, Harper County, sequence 88.

One form of location is latitude with longitude. There are actually two sets: projection and decimal degrees. The columns representing projections include the type or name of the projection, column horizontal\_datum\_name, and the latitude and longitude in degrees, minutes, and seconds relative to the projection.

The other form of location is geopolitical boundaries (country, state, county, Major Land Resource Area, Soil Survey Area, National Park, and National Forest). These are stored in the table site\_area\_overlap as links between the lims\_site and the area tables. The area table is the list of all the different types of boundaries with codes and names. These lists must be managed as boundaries and names change. The NASIS database is regularly updated and should be used as the source of change to the KSSL LIMS area table. Refer to Appendix F to learn how to do this. The area table also includes special columns related to Animal Plant and Health Inspection Service (APHIS). These columns are crucial to the handling of soil that may contain pests in defense against their spread. APHIS updates its list of counties that are known to have pests and must be updated in the area table. Appendix F has information about this.

However, some soil samples are not identified by location and do not require a site. For this reason, the lims\_site table is linked to the layer table and not directly with the project table. Unfortunately, this may create problems for a project destined for the SSL\_Repo database as it requires a site to be linked to the project.

## Pedons

Pedons are the intrusions (pits, cores, etc.) into the soil used to collect soil samples. They are called soil profiles and is a way to identify it as a whole. Information about the pedon is stored in the lims\_pedon table. Like the site, a pedon is identified by the column user\_pedon\_id, which is often the same as the user site id. One or more pedon records are linked to a site record so the user pedon id may require suffixing to provide unique ids. The natural\_key column is automatically assigned by KSSL LIMS as a calculated field: current fiscal year plus the letter ‘N’ plus a sequence number. This is done by a trigger when the project status is changed from ‘pre-submission’. The values for the particle size control section columns are determined by the sample submitter. They are used by the Calculation Engine when calculating pedon level properties.

Associated with the pedon are taxonomic classifications which are determined by soil scientists when collecting soil samples, when analyzed at the KSSL, and when correlating the pedon with others. The lims\_ped\_tax\_hist table is used to record taxonomic classifications with child table taxon\_family\_detail to record one or more family term details.

## Layers

Layers (or horizons) represent the soil at distinct depths within a pedon. There must be one or more horizons in a pedon for there to be a pedon record. This rule is enforced by a stored procedure when entering soil samples into the KSSL LIMS database via the workstation application. There can be layers associated with a project without sites and pedons, however. The information about a layer is recorded in the layer table. Each layer is at least linked to a project record and may be linked to a site and a pedon. The natural\_key column is automatically assigned by KSSL LIMS as a calculated field: current fiscal year plus the letter ‘N’ plus a sequence number. This is done by a trigger when the project status is changed from ‘pre-submission’. The lay\_rpt\_seq\_num column is automatically assigned starting with 1 for the first layer entered for the project. Changing the sequence number using the workstation application will renumber the sequence. The lay\_field\_label columns are to be assigned values meaningful to the sample submitter. Traditionally, the first one is assigned the user-pedon id followed by a sequence number within the pedon. The horizon\_designation column, if entered via the workstation application, will be split into the horz\_desgn columns. The aphis\_permit column is populated at sample login using the workstation application. It is important in the handling of soil samples where there is identifiable plant material in the layer.

Several tables are associated with the layer. The layer\_horz\_desgn\_ls table holds the horizon designation suffixes found in the horizon\_designation column. The layer\_texture table holds the list of texture classes identified in the texture\_desc\_abbrev column and the layer\_texture\_mod table identifies any modifiers associated with the texture classes. The layer\_fraction table holds the field-observed volume or percent of particle sizes at various size ranges. This data is used in calculating certain soil properties on a “whole soil” basis – typically, soil properties are measured and reported on a “< 2mm” basis, sometimes called “fine earth”. Other tables associated with layers are involved with the analysis work of the KSSL and will be described later.

## Samples

Samples represent the physical quantities of material submitted as part of a project to be analyzed by the KSSL. Information about the material is recorded in the sample table. The smp\_id column is automatically generated as a sequence. Bar code labels are printed with the smp\_id number and affixed to the container of material, at the time of the login process, so that the material may be identified later during the preparation and analysis process. The smp\_type column is key to identify what the material is, i.e. loose bulk soil, dried ground and sieved soil, and even water, and how it is to be prepared for analysis – even what analyses are appropriate. The aphis\_reg\_code is populated depending on where the material came from at the lowest level geopolitical boundary, country, state, or county, and is assigned at login.

## Preparations

Though samples are what is received in the KSSL, they are not typically what is analyzed. Each analysis performed at the KSSL require the material to be in a specific condition. In KSSL terms, the sample material may need to be processed and made into a preparation. The best example of this is bulk soil samples. They are submitted to the KSSL in a moist field state, loose, void of very large (> 75 mm) particles, and in a bag. Some analyses must be done on this loose, moist, soil while most analyses are done on a dried, ground, sieved, and homogenized soil. In both cases, the material prepared to be analyzed is called a preparation. There are preparations that are permanently stored in the KSSL Soil Archive facility while others are discarded after an analysis is completed or after all the analyses are completed.

These preparations are requested to be done based on the analyses requested. The kinds of preparations are defined in the preparation table. The values in columns master\_prep\_id, prep\_sort\_order, and prep\_list\_order drive the reporting system. Preparation information is found in the analytical\_sample\_disp table. The smp\_id column links to the sample table. The prep\_id column links to the preparation table. The prep\_seq column is a sequence number, allowing for several preparations of the same type to be created for a sample. Together, these three columns must be unique. The columns prefixed with ‘store’ refer to who, what, where, and when a preparation was stored. The columns prefixed with ‘smp’ refer to who, what, and when set the disposition of the preparation, i.e. discarded, consumed, available. The columns aphis\_reg\_treated and aphis\_reg\_contaminated indicate whether a regulated preparation has been treated in such a way that it can be handled as a non-regulated preparation or a non-regulated preparation has been contaminated so that it must be handled as a regulated preparation.

The box table identifies the various cardboard boxes and plastic trays use to transport preparations around the KSSL. Some cardboard boxes represent permanent storage (these are always placed in the Offsite or other locations) while others are used to rotate temporary preparations. Trays are always rotated with temporary preparations.

The preparation\_wt\_data table is used by the workstation application to record the weights when processing a sample to make the preparations. The workstation application actually considers the project’s bulk samples in an analysis batch with weighing the sample being the analysis with columns of weights for different processing stages. This way a data grid may be prepared and displayed to the technician for entry. Also, the processing\_detail\_history table has records inserted automatically when a processing weight is taken, identifying who and when for what project.

## Procedures

Laboratory procedures define how an analysis is performed. It is well defined and follows a pattern. They are devised, written, and posted in the Kellogg Soil Survey Laboratory Methods Manual (<https://www.nrcs.usda.gov/wps/portal/nrcs/detail/soils/survey/tools/?cid=nrcs142p2_054247>). Procedures are coded for easy reference. Procedure data is stored in table procedure. Of importance is the column proced\_abbrev, which appears in many reports, the proced\_code, which must match the writ-up found in the KSSL Methods Manual, the proced\_raw\_tab\_name column, which must point to the database table where the analysis data will be stored, the lab\_section column, which must have the laboratory section where the analysis is performed because it is used by the workstation application to show the technicians the analysis batches they may work with, and the anal\_id column, which links to the analysis table.

It is important to note that analyses to be performed on a sample preparation is assigned by analysis (described in the next section) but the analysis batch is assigned to a procedure. There are many tables that are linked to the procedure table because of this. The procedure\_analyte table links the analytes that are expected to be calculated from a procedure being performed with a procedure. The procedure write-up should show those analyses and how to calculate them from the measured data. The proced\_method\_code column is the way to identify a method code for that analyte which is used in certain reports. The following tables work together to automatically and manually record when tasks are completed for an analysis batch assigned for a procedure: procedure\_detail, procedure\_detail\_procedure, procedure\_detail\_history, and processing\_detail\_history. The tables prefixed with the word instrument work together to describe the sets of instruments used in a procedure – these will be described in a later section as well as the LIMS Link system, which is used to prepare files for instrument software and ingest instrument data.

## Analyses

While procedures describe how an analysis is performed, an analysis allows variations in how a procedure is done. The analysis table contains information used by the workstation application for the request of analyses on sample preparations. The columns anal\_status, default\_airdry\_prep\_id, and default\_moist\_prep\_id identify whether an analysis may be selected, what preparation is the default for air-dried prepartions, and what preparation is the default for moist preparations. The columns anal\_qc\_quantity, anal\_qc\_date, and anal\_qc\_count support the KSSL Quality Control program monitored by the Quality Control Officer. These columns ensure that sample preparations are run in duplicate during a time period so that quality control statistics may be gathered. These columns are used by the workstation application as prompts for action depending on the number of quality control samples (0 means never), the date the last one was assigned, and the number of quality control samples yet to be assigned for the cycle.

The analysis\_preparation table lists the possible choices of preparation types that can be assigned to an analysis. This list drives the workstation application analysis request for custom preparations for an analysis. The analysis\_detail table lists the choices and default values of instructions for analyses where options are permitted. The analysis\_suite table lists the names of groups (suites) of analyses that can be requested at one time via the workstation application. The analysis\_suite\_analysis table links these names to the analysis table. The check\_sample\_analysis table lists the special check (sometimes called quality control) samples that may be assigned to an analysis batch as regular samples are added. These check samples are assigned internally in the lab, known to have soil properties that can be repeated, have numbers that are less than 1000, and have a single preparation assigned.

The layer\_analysis\_request table represents the data created when an analysis is requested for a sample preparation. The columns link to the layer, the analysis, and the preparation; store when it was made and by which KSSL employee; whether the request is an original or a rerun and whether it was completed (by assignment to a batch) or not; how many times to repeat this analysis on the preparation; and whether the request is part of the Quality Control program. The layer\_analysis\_request\_detail table allows the analysis request to show the detail of the instructions chosen for this request. The layer\_preparation\_req table contains the preparation requests spawned by the layer analysis requests and those preparation requests manually entered that may not be associated with an analysis, i.e. a container destined for another laboratory. It should be noted that a number of analyses may share the same preparation but only one request will be generated.

## Batches

A batch is a collection of preparations that are to be analyzed as a unit. Sample preparations that are requested for analysis from projects that are in progress or review are queued in project priority order, with the exception that rerun analyses requests are highest priority. A technician creating a batch must regard this priority and assign preparations in that order. The batch may be interspersed with check samples and/or blanks (water) which is used for quality control, thinking that if the check and/or water samples do not produce expected results then the whole batch cannot be trusted and must be rerun. There is no limit on the number of preparations in a batch but is usually related to the limitations of the instruments used in the procedure. The analysis\_batch table contains a record of the batches created. The columns record when the batch was created and by which KSSL employee; the current status of the batch; whether data was linked (ingested from an instrument file); and any batch notes taken while performing the analysis. The batch is linked to the analysis while the procedure used is chosen by the technician as well as the instrument set.

The analysis\_batch\_sample table shows the detail of the preparations in a batch. The batch\_ord\_pos column is populated when the batch is created and in the order the technician requests via the workstation application. The lay\_anal\_req\_id links this preparation back to the layer analysis request. Note that check and blank samples are never requested in the same way as a sample, so this column is empty for them. The anal\_smp\_disp\_id column links back to the preparation for the sample being analyzed. This column is always filled in since a preparation must exist and be available. The column anal\_batch\_smp\_status shows the status of the analysis on this preparation and is used in reporting. The value is automatically set on certain events but may be manually set by the technician during running the analysis. The analysis\_batch\_instrument table is populated when the technician chooses a particular instrument within a set of instruments assigned for the analysis. This selection is used by the LIMS Link function so that the proper script is used when preparing files for instrument software and ingesting the data from the same.

## Instrumentation

The KSSL performs analyses using various pieces of equipment. These are identified in the Methods Manual as procedure write-ups. Not all equipment is accounted for but those that are can be found in the instrument table. These are recorded so that they can be tied back to a batch. This way, if a trend of bad data is discovered, the cause may be traced back to the instrument and corrected. Many of the columns in this table is for documentation. The instr\_model\_id column, however, is useful when specific instruments are grouped together by a model. The KSSL purchases more than one instrument of the same model when its workload warrants. A batch may be analyzed on any of these instruments knowing that the results are equivalent. The technician chooses which specific instrument was used at the time the analysis is performed. The instrument\_model table facilitates grouping like instruments, providing a name for association.

Procedures determine the sets of instruments that are used for collecting data. The instrument\_set table provides a name so that it can be associated with the analysis analytes. The instrument\_model\_set table links the models of instruments to a specific instrument set. The instrument\_set\_proced table then links the instrument sets to a procedure.

The LIMS Link system is programming in the workstation application that can be used to create files that instrumentation software will use to tie an instrument run back to the analysis batch samples. The sample\_info\_file facilitates this by joining a procedure (selected for a batch) to a specific instrument (also selected for a batch) and identifying a Visual Basic script file that is programmed to write the preparation information into a file that the instrumentation software can read and follow. The instr\_data\_acq facilitates the reading of the instrument software data files via Visual Basic scripts or via a Microsoft Access query.

When a new instrument is purchased by the KSSL, it may be to replace an existing instrument or to support a new procedure. Either way, decisions must be made to whether it requires a new model or added to an existing one, does a new instrument set need creating, what models need to be added to the set, and what procedures are associated with the set. Visual Basic scripts must be created to support the LIMS Link system which will work with the new instrument.

## Raw Data

KSSL analyzes sample preparations according to a procedure yielding measurements which culminate into a reportable property. The measurements must be recorded into a database table (called a raw data table) so that the reportable properties (called analytes) may be calculated, reviewed, and validated. The procedures involve a set of instruments -- the raw data tables must be created to accommodate the measurements from these instruments. There are other aspects of running an analysis that require a reagent (called a reagent batch) to be prepared or instrument calibrations (called a calibration set) to be recorded. The raw data table must be able to accommodate the association of a reagent batch or calibration set. The procedure write-up gives clues on what measurements are necessary for recording and which are involved in calculating the analytes.

Raw data tables can be segregated into two groups: single and master/detail. Single raw data tables allow for the recording of measurements associated with a single sample, like sample weight, concentration of an element when there is only one element, etc. These raw data tables are named with ‘data’ at the end but are void of ‘mas’ and ‘det’ (for master and detail). Master/detail tables are pairs of tables for recording measurements for a procedure. The master table is like the single raw data table, used for recording single measurements for a sample while the detail table is for recording many measurements for a sample, like cations. The pair of tables are named with the same prefix and end with the word ‘data’ but are distinguished with ‘mas’ (for master) and ‘det’ (for detail) within the name. The workstation application uses this information for preparing a data window for the technician to do the actual recording. In the case of a single table, the window will appear as a single spreadsheet. In the case of a master/detail pair of tables, the window will appear as a split screen with the master table in a spreadsheet in the top portion and the detail table in a spreadsheet in the bottom portion.

## Reagents, Calibrations, and Tares

Certain procedures call for the preparation of reagents with measured values that have a part in calculating analytes or is note-worthy. The table reagent\_batch is flexible enough to record information about different kinds of reagents. The reagent\_batch\_id column is recorded in the raw data table for the analysis batch for which a reagent batch took part in the analysis. The workstation application allows for the management of reagent batches and presenting a choice list to the technician for selecting the reagent batch in the data window of the analysis batch.

Certain procedures call for the calibration of an instrument that is used in an analysis. The calibration is performed on a set of samples with different concentrations of a chemical. The measurements for these samples are taken and recorded as pairs. These pairs of concentration and measurements form a calibration curve. The curve is then used in calculating the concentration in an unknown sample using the measurement attained by the analysis. The table calibration\_master stores the singular identification of a calibration set, such as what procedure, when the set was prepared, and whether the set is actively used or not. The table calibration\_detail stores the individual sample data of known concentrations with measurements. The workstation application allows for the management of calibration sets and presenting a choice list to the technician for selecting the calibration set in the data window of the analysis batch.

Certain analyses use permanent sets of containers to weigh soil preparations as part of the procedure. The preparations are placed in the container which has a unique identifier stamped or marked on it. This identifier is entered in the analysis batch (usually from a choice list) for the sample so that the container weight may be subtracted from the total weight when calculating the analytes. These container sets are named and dated. This information is stored in the tare\_master table, along with who and when the tare containers were weighed. The status column indicates if the set is actively being used – under this scheme, only one set should be active. As new sets are created because the container sets are reweighed, older sets are marked inactive. The tare\_detail table has the list of containers for a set with its weight. The workstation application allows for the management of tare sets and the preparation of the choice lists from the tare sets.

## Analytes

When the workstation application is used to record measurements and other data via the data window, calculation request records are automatically recorded in a table (calc\_queue) which will be processed by the KSSL Calculation Engine computer program, described later. The calculation requests identify the analysis batch and position. The analytes to be calculated are discerned based on the procedure associated with the analysis batch and the analytes associated with the procedure (procedure\_analyte table). The analyte table lists each analyte that can be calculated from the raw data tables, which are defined in the procedure table. The columns in that table take an active part in the calculations and reporting systems.

There are two levels of analytes: sample analytes and layer analytes. Sample analytes are calculated for each sample in an analysis batch. The columns in the analyte table used are the calc\_script\_path, which identifies a file that is a Visual Basic script directing the calculation logic, and adod\_dependency, which is a flag indicating if the ADOD analysis result is needed as part of the calculation logic. The calculation results are stored in the sample\_analyte table. The columns contain links to tables showing the analysis batch, procedure, preparation, analyte, instrument set, when it as calculated, and what its status is. Sample analytes must be reviewed and validated before an analysis batch is marked as completed. Layer analytes are prepared by the KSSL Calculation Engine from validated sample analytes. When a sample analyte is marked as valid, a database trigger is fired which records a calculation request. The KSSL Calculation Engine then uses the agg\_method column of the analyte table to decide whether to average (mean) or to concatenate (concat) the validated sample analytes into a single value for the layer analyte. The calculation results are stored in the layer\_analyte table. Many of the columns come from the sample\_analyte table columns.

The reporting system uses the following columns from the analyte table: analyte\_abbrev, a unique identifier that is used in queries and column headings, analyte\_type, indicates whether the value is to be treated as integer, floating point, or character, and analyte\_format, used in the formatting of numerical values for rounding.

## Results

KSSL provides property values from data measured in an analysis of samples. KSSL also provides calculated property values based on those measured values. These calculations are described in the KSSL Methods Manual. They are defined in the calc table. Like the analyte table, the columns take an active part in the calculations and reporting systems. One of the main differences is that analytes are calculated from the raw data table values while the results are not – they are calculated from analytes and/or other calculations. For that reason, the column calc\_depth is set as an indication of the dependency of calculations upon other calculations. A calc\_depth of 1 means that only layer analytes are involved. A calc\_depth of 2 means that at least one calculation with a depth of 1 is involved. Higher depths mean that at least one calculation is involved at one lower. This information is used by the KSSL Calculating Engine to order the calculation requests so that lower depth calculations are performed before the higher ones, making the data available at the proper time.

The calc\_analytes table associates the layer analytes with the calc table entries to support the calculation script. The calc\_analytes table records are used by the Calculation Engine to retrieve the data and replace the script variables with the values so the script can be run. Logic in the script determine what to do if no value was found in the database, such as substitute a default value or do no calculation and return an empty value. The calc\_analytes narrow an analyte by five columns: analyte\_id, proced\_id, master\_prep\_id, instr\_set\_id, and size\_frac. Variations in the value of any of these columns will require a different calc entry along with a different set of calc\_analytes records.

The calc\_sub\_calc table associates the layer results with calc table entries to support the calculation script. The calc\_sub\_calc table records are used by the Calculation Engine to retrieve the data and replace the script variables with the values so the script can be run. Logic in the script determine what to do if no value was found in the database, such as substitute a default value or do no calculation and return an empty value. Since different calc table entries represent the same calculations with variations in the layer analyte identification columns, there will be different calc table entries representing variations in the lower level calculations. Combinations of layer results involved in calculations will create a permutation requiring many calc table entries.

## Miscellaneous

Many tables are prepared to support the workstation application and other aspects of managing the KSSL LIMS system. The following paragraphs describe those tables in alphabetic order.

The application\_error table records when there is a programmatic fault with the workstation application. The columns identify when, who, which workstation, what the fault was, and the specific line of programming code where the fault took place. Sometimes the fault is due to human error and the user will see a window pop up with the fault message. The user response should be to discern if it is indeed caused by them and take corrective action or to notify the person managing the KSSL LIMS workstation application. Sometimes the fault is due to some general fault of the Windows system. These faults can be ignored since they cannot be traced to a specific cause. Sometimes the fault is due to an error in programming. These are indicated by a specific line of code in the stack\_trace column. A computer programmer who has access to the code repository should investigate the severity of the code error and determine if code updating is required. The contents of this table should be monitored daily.

The calc\_error table records when there is a fault during the processing of a calculation request by the Calculation Engine. The columns identify when and what caused the fault. Sometimes the fault occurs because necessary data is needed to run the calculation script and was not found. This is common for changes to the pedon or layer records which would require a re-calculation but the original data was not found, maybe because the original data was never calculated due to no analyte data. In this case, the fault is expected and should be ignored. Other faults may be because there was a database time-out in retrieving the supporting data. In this case, the calculation status should be changed for requeuing the request. Sometimes the fault occurs because there is a programmatic error in the script. The script and/or calc\_analytes and calc\_sub\_calcs should be examined and corrected. The contents of this table should be monitored daily.

The calc\_queue table holds calculation requests to be processed by the Calculation Engine. The requests may be inserted by the workstation application as a response to the entry of data or the validation/invalidation of calculated analytes or they may be inserted manually when a calculation is needed but was not automatically created. The columns identify when the request was recorded and then processed. Other columns support the needs of the Calculation Engine while others support the needs of the workstation application. These will be described in more detail in the section on the Calculation Engine. The contents of this table should be monitored daily.

The change\_log table holds a record of changes made to KSSL LIMS project data after a project has already been marked as completed. When a KSSL LIMS project is marked as completed, its data is destined to be copied into a repository database for publishing to the public. A database job is scheduled to run every night with SQL to check when projects are newly completed. Candidate projects are then copied entirely. However, completed projects may be reopened and data entered and updated. Database triggers are fired when this happens which record these data changes into the change\_log table. The same database job then reads the change\_log records and processes them so the repository database is updated. This system of updating is described in a later section, Relationship to NCSS Lab Data Mart

The tables data\_tier, data\_tier\_column, dt\_mcode, report, header, and report\_body work together to support the KSSL LIMS Reporting System. This system will be described in detail later.

The display\_table is used by the workstation application to display grids of data in windows and allow certain functionality. This way the grids may be altered in the table with immediate response without reprogramming the workstation application. The form\_name column is the programmatic name of the window (form). The data\_grid\_name column is the programmatic name of the grid in the window. The other columns are flags indicating the functionality settings. The display\_columns table shows the column specifications of each grid in the display\_table. The column\_name refers to the column name from the query used to generate the data being displayed. Most of the other columns in this table are intuitive.

There are a set of tables prefixed with inventory. This set of tables were designed to work together for creating an inventory of whatever group of items can be managed, such as equipment, chemicals, etc. However, the time spent setting up and updating the inventory proved to be cost ineffective.

The laboratory table contains the identifiers for the laboratories involved in the layer\_analyte and result tables. The KSSL LIMS system automatically assigns the lab\_id for analytes and results produced by it. If data from other labs are inserted manually, a record should be recorded in the laboratory table. However, the laboratory identifiers are not used in the reporting system or anywhere else.

The limsdd\_domain\_master table records represent kinds of data that have fixed data. The workstation application uses this table to prepare a choice list on forms. These choice lists are valuable in limiting users of the workstation application to certain choices and provides the convenience of not having to type the exact spelling, eliminating such errors. Managing the choice lists (domains) is accomplished without reprogramming the application. The column domain\_key is the unique domain identifier that is used in other tables when referring to the domain, such as the limsdd\_table\_column tale. The column domain\_name is a unique name used in other tables, such as the display\_columns table. The domain\_choices column contains the count of choices in the limsdd\_domain\_detail table for the domain. The column domain\_ordering\_method may have the value Explicit (order by the sequence number) or Choice (order domain choices alphabetically). The limsdd\_domain\_detail table contains the choice list values. The choice\_inactive\_indicator is a flag indicating if the choice is active or obsolete. This value is used by the workstation application to display obsolete choices in gray and should not be chosen. Note that values should never be removed from a list but rather be controlled via the flag.

The limsdd\_table table documents all of the KSSL LIMS tables. The table\_type column drives the workstation application, particularly the data grid of the analysis batch form. The value ‘single’ indicates a table is singular in nature while ‘master’ indicates a table is coupled with a ‘detail’ table in a one-to-many relationship. The workstation application uses these values to prepare a single data grid for ‘single’ tables and a split grid (master table at the top and detail on the bottom) for ‘master and ‘detail’ tables. The limsdd\_table\_column column documents the columns in each of the tables. The values in the columns should match the column definitions in the database schema, such as column\_name, column\_data\_type, column\_field\_size, column\_precision (for numeric columns), and column\_not\_null flag. Other columns (for data tables) drive the workstation application in building the data entry grids. Column column\_label is displayed as column labels for the grid. Column display\_width is used to size the grid column in pixels. Column column\_type is used by the workstation application to decide how to act with data entry. Type ‘auto\_list’ is used with the domain\_key column to automatically create a detail row of data for each value in the domain for each master row of an analysis batch in anticipation that these detail rows will be filled in with data. Type ‘choice\_list’ is used with the domain\_key column to prepare a choice list for the data entry column. Type ‘data’ is used with columns column\_minimum and column\_maximum to test entered values against these and will display an error message to the user when data outside the boundaries are entered and the value will not be accepted. Type ‘foreign\_key’ is used with the columns foreign\_table\_name (table that the column links to), foreign\_column\_name (column in the foreign table), foreign\_display\_columns (a SQL based string to form what the user will see in the grid column), and foreign\_table\_where\_clause (a SQL based string to filter the choices) to produce a choice list the user may select from. Type ‘primary\_key’ identifies the primary key column for the table and will not be displayed in a data grid. Column column\_default is used to provide a value in the case the user does not. The cursor\_com column contains a string of cursor movement instructions that the workstation application will take after a value is entered in a cell of that column. The values may be L, R, U, and D (for left, right, up, down) and comma separated. For example, D,L,L,L is interpreted as ‘move the cursor down one row, then move the cursor to the left three columns’. If this column is void, the natural cursor movement is down one row. The proced\_det\_id column identifies the table column with a procedure task (see Procedures section). The workstation application uses this column to automatically record when a procedure task for the analysis batch is completed by data being entered in the column cell.

The note table is used to record KSSL employee notes for these types of database tables: layer, lims\_pedon, lims\_site, project, and sample, which is identified in the source\_type column. The source\_id is the primary key value for the specific table. This allows some flexibility to use one table for notes regarding various items. The report\_flag column indicates whether the note may be displayed in a report that may be distributed. The private\_flag column indicates whether the note may be seen by other than the author. Notes are made available to be entered/edited/viewed by the workstation application whenever a Note button is present. There is a checkbox associated with the button to indicate if a note already exists. Check samples must have a note record which describes the check sample, including what analyses it may be used with. These notes cannot be entered via the workstation application and must be added via Microsoft Access or SSMS.

The person table is used to record the people associated with the KSSL and its work. These people may be sample submitters, project coordinators, technicians, supervisors, and even external contacts. Most of the columns are for information only. Column project\_submitter\_flag is used by the workstation application to make a choice list for the project table column proj\_submit\_id. Column project\_coordinator\_flag is used by the workstation application to make a choice list for the project table column proj\_coordinator\_id. Column ssl\_employee\_flag is used by the workstation application to make a choice list for every table where an employee id is needed.

The period table is a static look-up list of dates with various formats. The date\_id value would be stored in various tables where a date id is appropriate. It should be noted that the highest date is 2030-12-31. After this date, the period table must be revised.

The report\_list table is a list of web-based reports served by the KSSL LIMS web server. These reports are accessible via the workstation application in a few ways, driven by the rpt\_section column value. The value ‘Application’ means the report is accessed via a link on a form or a choice list. The value ‘Bench’ means the report is a bench sheet report accessible from the analysis batch window in the report list (lower left corner). The value ‘Predict’ means the report is accessible from the Preview button of the Analysis Batch Creation Wizard. All other values refer to sections of the laboratory and appear in a choice list on the Report Selection form. The rpt\_name column is the name of the report that would appear in a choice list, if it is exposed that way. The rpt\_pointer is the base URL for the report and would be used with a web browser window. The rpt\_param column is used with the bench sheet and predict reports to identify the procedure id appropriate for the report. The requires\_proj flag indicates if the report is a project level report and requires the project id parameter and value added to the base URL

The status\_history table is a way to capture the status changes of certain KSSL LIMS items over time. Many tables have a status column and a status date but only if a singular event is required. This tables is for recording what, when, and who for a status that needs recording different status values over time. The type of items is stored as a value in the source\_type column. Currently, the values are ‘project’ and ‘analysis\_batch’. The source\_id is the value for key field of the table matching the source\_type. The column status\_code is the value of the status. A project’s natural progression of status over time is ‘pre\_submission’ (samples are logged in)->’ validated’ (the project analysis scheme is entered and the project may be worked)->’ in progress’ (the project samples are processed and may be analyzed)->’ review’(all analyses completed and being reviewed)->’completed’(all lab data reviewed and attested to be distributable). Other values may be ‘abandoned’(the project is not allowed to continue) and ‘distributed’(lab data was distributed). Most of these status values are entered manually in the workstation application and affect how the application behaves. For example, a project that is ‘complete’ is locked to further changes. For that reason, a completed project may have the status changed back to ‘review’ to allow further analyses or updates. After the new analyses or updates are made, the project status should be changed back to ‘completed’.

The study and study\_detail tables were meant to link projects together as part of a study. This concept was never implemented in KSSL LIMS, but the tables remain in case an interest gains momentum.

The thermal\_standard table was designed to record information about standard samples used in the DTA analysis. It was never implemented, but the table remains in case it becomes important.

# Workstation Application

The KSSL LIMS workstation application controls the insertion/update/deletion of data for certain tables in the database. It will give a user the rights and privileges of a database administrator without giving them autonomy. It does this in the programming by presenting table columns to the user to allow entry in a controlled way. Though restrictive, the user is allowed to interact with the database to fulfill their duty to the KSSL.

The workstation application serves as a communication device. Employees of the KSSL use the application windows and reports to find work that needs to be accomplished. As work is accomplished, the employees update database tables which then appear to other employees for them to work on. For example, when a shipment of soil samples arrives at the KSSL, employees are instructed to log them in the KSSL database via the workstation application (Sample Submission). Employees in charge of preparing a project for work in the lab will see the submission and begin their work. When the project is prepared, the employee will change the project status which will alert other employees that work is ready to be done. This process continues until all samples are processed, all analyses completed and validated, all results reviewed and approved, and the project is completed.

At a given moment the KSSL will have many active projects in various stages of work. The KSSL workstation application displays data for projects, showing the work that is accomplished and what remains to be done. The following paragraphs describe the parts of the application in terms of this work in the context of its command bar:



The Application command controls the appearance of the main window display. The choices for the command are toggles to show the toolbar (icons for commands), status bar (identification of who is the user, etc.), calculations (shows the results of scheduled calculations started at the workstation by the user), and unit conversion ( provides a way to convert units of measure to aid in data entry). Of these, the calculations panel is the most helpful to employees who are working with batches of analyses. Calculations are requested when data is entered/modified for a batch. The panel is updated to show the batch calculation request status so the employee will know and act.

The Sample command is geared to the set of employees who work with logging in and processing samples. The choices for the command are Label Print, Submission, Preparation, and Disposition. Label Print command opens a window used to print bar code labels for sample numbers to be affixed to sample bags (or other containers) for reference in sample submission. Note that a label printer must be attached to the workstation and set up in the Serial Ports Setup Tool to be effective.

The Submission command is the first command used for a project (a submission of samples to be treated as a unit). The window shows a choice list made up of existing projects by fiscal year, which is itself a choice list. There is a button to create a new project. The user clicking on that button is led to build a project name based on project type and location (country and state) – fiscal year and sequence are given. If the user accepts the name, the application will insert a record in the project table, filling in columns with only the known values, and then updates the project choice list.

The normal sequence of sample login follows in two paths –the pedons are in NASIS or not. If the pedons are in NASIS the user clicks on the Import Pedon button which directs the application to download the site-pedon-layer records (with some of the child tables) using a web connection to the NASIS web server and runs a NASIS KSSL report. The application reads the report results and constructs a tree view of the site, matching the view as if it was part of the project. The user may accept the view to be inserted into the project or not. If the pedons are not in NASIS the user must add sites, then pedons, then layers in a hierarchical way. For each there is a minimum of information required that is pre-filled as much as possible but could be overridden. This is presented as a tree view. A member of the tree view has the focus of activity and becomes the context for updating or additional entry. If the member having the focus is a layer, a sample grid is presented so that samples may be entered.

The Preparation command is used to set up and record the processing of logged-in samples for analyses. The grid displayed is built from a query of projects that need to have preparations done. Before this time, projects are given an analysis scheme which determines the preparations required to fulfill the analysis requests. At this time additional preparations may be requested that are not tied to an analysis. In any case, when the Preparation command is invoked, a stored procedure is run that matches the preparation requests with the preparation records already created (analytical\_sample\_disp table) and creates the records needed. Then the grid is built. If it is discovered that there are samples logged without a preparation request or preparations requested that cannot be tied back to a sample, the row in the grid is highlighted in yellow as an alert. The buttons at the bottom of the window are used to perform various aspects of the processing: collecting weights from bulk samples, printing the worklist and other reports, printing preparation labels, and recording the preparation completion and location (disposition).

The Sample Disposition command is used mainly for altering the disposition of preparations already made. There are three tabs in the window: Sample Search, used to search for preparations based on project/site/pedon/layer/sample/analysis batch; Box/Tray, used to search for preparations based on which box or tray they were placed in/on; and Selected Samples, used to “collect” information about the preparations found in the other tabs. The first two tabs allow for updating the different columns for each row or for a number of selected rows.

The Project command is meant for all KSSL employees to search for a specific project in order to view/alter information about it at any moment. There are one or two choices for the command depending on the employee’s role type in the ssl\_user\_role table. All employees have access to the Laboratory Listing command. Clicking on this command will build a grid of projects from the entire system for all time. The user can narrow focus based on fiscal year and/or project status or search for a project based on characters found in the lab project name or based on a known identifier for sites/pedons/layers/samples (Find Sample button). Only administrators of the KSSL, having the role LabSuper, will have access to the Priority command. Clicking on this command will build a similar grid as the other list but will allow changing the priority of a project or the note column of the project table. In either case, selecting a project leads to the opening of a window showing a tree view of the project with buttons that allow opening other windows exposing different parts of the project for viewing, printing, and editing.

The Analysis command is meant for technicians of KSSL to create batches of sample preparations requested for analyses and to record the results of those analyses. The analysis scheme that was entered for a project generated preparation requests so that samples would be processed. These preparations would be given a status of ‘available’. Projects that are either ‘in progress’ or ‘review’ status, have analysis requests that have status ‘requested’, and have matching preparations ‘available’ will appear in the list of preparations that can be added to an analysis batch. Clicking on the Batch Creation command initiates a query to build a set of candidate preparations by KSSL section and analysis. The sections a technician is assigned, via the ssl\_user\_lab\_section column of the ssl\_user\_role table, is used to minimize the choices of analyses. The user selects an analysis and initiates a wizard that guides the user to “build” a batch that will be given to a technician to analyze. Clicking on the Data Acquisition command initiates a query to build a set of analysis batches that exist for all time. The presented list is narrowed by the sections a technician is assigned. At first the list is narrowed by the active batches (status not ‘completed’) but can be opened to include completed batches. A selected batch is highlighted and then opened with clicking the Open batch button. The window presented is used to select a procedure and instrumentation used in the analysis, enter data, print reports, and validate the analytes produced by calculations of the data entered. Once the analytes have been validated/invalidated, the Completed flag may be checked to indicate the batch may be removed from the list of active batches.

The Reports command is meant for all employees of the KSSL to view information about the projects. The list of reports come from a query of the report\_list table. The window displayed shows a choice list that narrows the report list by sections of the laboratory or by purpose (General or Management). The General reports are typically used to generate a product that can be delivered to a customer or displays information about a project to be used in a general way, i.e. Project Analysis Status. The Management reports are typically not about a specific project but the system as a whole and used by the KSSL management to report statistics. The reports by laboratory section can either be to report project data for a specific analysis or show general information about the section that assists the technicians to do their work. The user selects one or more reports by clicking on the check boxes next to the report name. The window displays a list of projects that can be narrowed by fiscal year and/or completed status and/or identifying parts of the laboratory project name. The grid may be sorted by clicking on the column headings. The user selects one or more projects by either double-clicking on the row in the grid or clicking once and clicking on the Add button. The reports are generated by clicking on the Open Reports button. The result is a cross product of the projects and reports selected with each one displayed in a separate window. Each window is a web browser that uses a URL prepared from the report\_list entry and the necessary parameters identified in the report\_list entry, i.e. project identifier.

The Tools command is a menu list of system-wide tools in support of the other commands. The Tare Weight command opens a window with a set of controls used to create/edit sets of objects where the object is reused, and the weight is maintained over time. The object’s identifier (usually stamped or labelled on the physical object) can then be entered in a data grid for a batch and the weight is determined by lookup. How the lookup is handled can be found in the limsdd\_table\_column table in the ‘foreign’ columns. See section 2.13 about how the sets are stored. The Reagent Batch command opens a window with a grid allowing creation/editing of information on reagents prepared and used in various analyses. The batch id column is then entered in a data grid for a batch and the data (weight, pH, or normality) is determined by lookup. The Calibration Set command opens a window with a set of controls used to maintain sets of instrument calibration sets based on procedure. An instrument is calibrated by measuring concentrations of known standards and recorded as a set. The set is then used to calculate a concentration for an unknown sample. The set’s identifier is selected in a data grid for the batch.

The Sample Information File command opens a window with controls that facilitate creating what is called a Sample Information File (SIF). A SIF is a file that can either be read by an instrument’s software or whose contents are copy-and-pasted into an instrument’s software control panel to guide the instrument’s analytical run. The end game is to direct the run so that analytical results can be read back in by the companion command Instrument Data Acquisition to load the results into the LIMS database for the batch. This is done by providing the batch identifiers in the SIF which are output by the instrument software and in turn read by the IDA process (referred in the KSSL as LIMS Link). Since different procedures require different data and different instrument’s software require different parameters the controls on the SIF window help narrow the information needed to do the work. Each KSSL LIMS batch is assigned a procedure and set of instruments, including identifying a specific instrument, if available. Therefore, it is required that a batch or list of batches be entered. Usually a batch has 24 samples, but an instrument run may allow 96 to over a hundred samples. Therefore, the technician may choose more than one batch by entering a comma-separated list. All the batches must have the same procedure and instrument sets selected for this to work. Clicking the Get Batch initiates the gathering of the procedure id and the instrument list. The samples’ data from the procedure raw data table is displayed as a grid. The instrument list is presented as a choice list. The technician then chooses the instrument and then the samples in the grid. The SIF is named based on requirements by the instrument software and can be altered by the technician, if desired. The Make SIF button is clicked to start a process that reads and runs a Visual Basic script that is programmed for the procedure-instrument combination. KSSL LIMS reads the script into memory and alters it by putting in the list of samples as an array. Then the script is run using a system shell command with the results returned to KSSL LIMS as a character string. That string is then saved to the file. The technician will then set up the instrument’s autosampler with the physical tubes of sample extracts to analyze and use the instrument’s software to “read” the SIF followed by running the software.

The Instrument Data Acquisition command opens a window with controls that facilitate reading the file(s) resulting from an instrument run. Like the SIF process, a list of batches that are involved in the run is entered in the batch list. Clicking the Get Batch initiates the gathering of the procedure id and the instrument list. The instrument list is presented as a choice list. The location of the data file(s) is presented. Depending on the instrument, it may be necessary to navigate to the directory where the files are found and selected. In any case, the IDA process treats each file separately one-by-one. Clicking the Get Data button starts a process that reads and runs a Visual Basic script that is programmed for the procedure-instrument combination. KSSL LIMS reads the script into memory and alters it by putting in the list of batch numbers as an array and the full path of the file. Then the script is run using a system shell command with the results returned to KSSL LIMS as a character string. That string should be a comma-separated list of values read from the file which is then converted to a grid and displayed to the technician. This process is repeated for each file selected. The technician may then choose which lines of data to have transferred to the LIMS data base raw data table for the procedure and batch.

The Serial Port Setup command opens a window of controls that allow the identification and set up of serial devices and their communication protocol for those connected to the workstation via cables and the serial ports, including those that use a USB port and serial converter. There are basically three instrument types – balance, bar code reader, and bar code printer – that fit into this scheme. Though choices can be made to set the protocol, most of these instruments come out of the box using factory settings that match the KSSL LIMS default settings. Therefore, the typical operation is click on the row in the grid for the port the instrument will be attached, set the instrument type, and set the instrument name (balances only). The Save Input and Sound Bell checkboxes should be set for balances. The Save button is clicked to save the information to a special configuration file which is read every time the KSSL LIMS application is opened.

The Instrument Setup command opens a window with a grid listing the active instruments. If there is an instrument from this list attached to the workstation the Acquisition checkbox must be checked for the SIF and IDA tools to work.

# Calculation Engine

The KSSL LIMS Calculation Engine is a stand-alone computer program. Its purpose is to fulfill calculation requests to supply data that will satisfy customer’s requests, including the KSSL internal customer. The program is intended to run 24 hours a day, 365 days a year. It was originally programmed to run on a server computer as a service that automatically starts whenever the server is booted and can be stopped, paused, or restarted. Something changed when Windows Server 2012R2 was installed and the program stopped functioning that way. Currently, the program must be started manually either from the server or from a workstation. The caveat is that the user starting the program must remain logged in to keep the process running.

The Calculation Engine performs by constantly reading the LIMS database table calc\_queue via a stored procedure that collects the requests and orders them based on the priority column and the status column in the table. The priority is set by the KSSL LIMS workstation application when requests are inserted into the calc\_queue table. In general, sample analyte calculations are priority one with layer analyte calculations next, followed by result calculations with increasing value depending on the level of calculation found in the calc table. These requests can be manually inserted but are mainly inserted automatically in response to certain events. Those events are: entering/updating data into the Data window of a batch which triggers the sample analyte calculations; the validation of a sample analyte which triggers the layer analyte calculations; the creation/update of a layer analyte which triggers a result calculation dependent on the layer analyte; the creation/update of a result calculation which triggers a result calculation dependent on the result; the creation/update of a layer analyte which triggers a pedon result calculation dependent on the layer analyte; the update of a layer depth which triggers a pedon result calculation dependent on the layer; the update of a pedon control depth which triggers a pedon result calculation. Each of these events are expressed in a calculation class which is recorded as a number in the calc\_class column. This value is used by the Calculation Engine to direct the processing of the request. The scheme is defined in Appendix \_\_. Understanding the scheme is important to develop requests that are inserted manually in order to produce results that did not take place under normal circumstances.

The calc\_status column displays the status of the request and is represented with the values 0, 1, 2, and 4. All requests begin with status 0 – the Calculation Engine chooses the requests to process by filtering on this status value. Once a request has been selected for processing the status is updated to 1. After a request is processed, the status is updated to 2 – success – or 4 – failure. Failures are logged in the calc\_error table. The columns in that table link back to the calc\_queue record and indicate what went wrong. Failures can happen when the time of processing goes too long – the response would be to reschedule the request by changing the 4 back to a 0; pedon calculations were requested due to changes in the control section but the data is not there to actually calculate – the response would be to delete the request; or the calculation script has an error – the response would be to fix the script an reschedule the request. When errors are mitigated, the calc\_error records should be deleted.

Other columns in the calc\_queue table are calc\_ident and calc\_sub\_ident, whose values are determined in the context of the calc\_class column (see Appendix \_\_). The columns calc\_req\_date and calc\_comp\_date are the time stamps of when the request was inserted in the queue and when the processing of the request terminated. If column client\_display\_flag is set to on, the columns machine\_name and ssl\_user\_id will be set to the workstation name and the KSSL employee id who used the KSSL LIMS workstation to make the request. These columns also work together with the KSSL LIMS to return notification when the requests terminate successfully or in error. There is a stored procedure -- sp\_resetCalcQueue – which removes successful calculation requests that are more than three days old to keep the queue small. A SQL Server Job -- Maintain calc\_queue table entries – is scheduled to execute the stored procedure once every night.

# Reporting System

The KSSL LIMS reporting system handles different needs of personnel at the KSSL, displaying data from the database as a report. Every report is web based and is served using a Windows server running the Internet Information Services service. See Appendix B on how this should be set up on the server. The web pages are organized in folders which coincide with categories in the KSSL LIMS workstation application. Most reports center around one or more queries that are formatted with HTML for viewing. The queries typically use a LIMS Object for the database connection which returns a dataset object. The report then loops through the dataset, formatting the data for each row. These reports can be found in the report\_list table.

An exception are the reports that require special formatting which allow different types of data to be arranged in a group, called a tier. Several tables work together – report, report\_body, header, data\_tier, data\_tier\_columns, and dt\_mcode – to define how these tiers are aligned with a report and formatted. There is a special program that is compiled named LimsRptCV3 which accepts a pedon identifier and a tier identifier and queries for the data, preparing a grid. The web page then passes that grid through an XSL file that formats it with HTML for viewing. The report table simply lists the different reports – these are present in the report\_list table so that they can displayed to the user as choices in the Report command window of the KSSL LIMS workstation. The URLs point to the web pages of the web site that do the work of collecting the pedons and tiers and calling upon the LimsRptCV3 program to format each pedon and tier combination, ultimately displaying the formatted tier. The report\_body table links to the report table records. Each record represents a tier which is either a header or data (section\_type column) with a sequence number (section\_seq\_num column) to order the tiers.

The header type tiers are used as headers to the pages of the report with a flag (column section\_header) to indicate if the header is on the first page of the report or on the second and subsequent pages. Reports are intended to be printed on 8.5 by 11-inch paper in landscape mode with a .25 inch margin all around. To aid in the paging, column section\_row\_count indicates the number of lines taken up by the tier as a fixed number, if the value of column section\_count\_type is ‘static’, or a least value, if the value of column section\_count\_type is ‘dynamic’. The counting can be as simple as counting the output lines in the XSL file – this is usually the number of <TR> tags. The section\_xsl\_path is the file name of the XSL file used in the page formatting of the data. The full path is relative to the web site’s path on the server with a subfolder of ‘rptXSL’.

The data\_tier type tiers are expected to be grids -- columns of laboratory data with rows tying that data to pedon layers. The data\_tier table contains rows simply listing the data tiers, linked back to the report\_body table. The data\_tier\_columns table contains a row for each column of data for each tier. The column column\_index is a sequence number starting at one. The column column\_name is important in that it’s values must be referenced in the XSL file spelled exactly. The column column\_type must have the value ‘required’ to indicate that this column must have data for the tier to appear regardless if other columns have data or not. The dt\_mcode table supplies the laboratory data identifiers used in preparing the grids based on the tier and column. The analyte\_id column has either an analyte\_id from the analyte table or a calc\_id from the calc table. This is differentiated by the value in the analyte\_type column which is ‘aggregate analyte’ for analyte\_id or ‘derived\_analyte’ for calc\_id. The other columns are intended to help the LimsRptCV3 program match up the calculated results with the analyte results based on the analyte’s procedure, preparation code, instrument set, and size fraction. These columns do not require population for a tier of homogenous data. Any number of analytes and calculations may appear in a column. This allows for more than one piece of laboratory data to fit into one column. An example of this is Total Clay, which can be found on tier 6 and column 2. Total clay can be determined using a number of procedures, each with a different analyte\_id. There is a different dt\_mcode row for each of these. If a layer has Total Clay measured with more than one procedure, then the grid will have more than one row of data for that tier. The grid will carry the procedure with it to help the LimsRptCV3 arrange the data. If necessary, the program will create multiple grids to account for differences due to procedure and the procedure code will be displayed in the grid as a column heading. It is important, then, to maintain the dt\_mcode by adding new rows when new analytes or calculations are added due to new procedures, preparations, or instrument sets.

# Relationship to NCSS Lab Data Mart (NCSS LDM)

The KSSL LIMS is a complete, whole system, with the goal to produce data that can be delivered to a customer. The USDA is mandated to make this data public. The KSSL LIMS fulfills this mandate by feeding laboratory data to a repository database (named SSL\_Repo) that is accessible to the public via a web site with web pages that provides choices and text the user enters, queries the database, and returns a result. The National Cooperative Soil Survey (NCSS) is a corroboration between USDA and its partners to carry out soil survey in the United States. In the case of laboratory data, USDA has the KSSL and the partners are university laboratories. The NCSS Laboratory Data Mart (NCSS LDM), like the KSSL LIMS, is a system of servers, database, web site, and users. Because the NCSS LDM serves the same laboratory data as the KSSL LIMS, the database has much the same structure as the KSSL LIMS database. The differences are: analytes and calculations are combined in a single analyte table – since the NCSS LDM does no calculations, this works well; no sample data and no sample related data is migrated since laboratory data is reported on a layer basis; the layer analytes and all of the calculated results are migrated into a single results table; a set of bridge tables exist to link information in the repository database back to the original system from which the data came – university lab databases, previous USDA laboratories, and LIMS.

The migration of data from LIMS to SSL\_Repo is automatic via SQL Server Job named REPO Synchronization Routine that is scheduled to run once every night to synchronize the lookup list tables and to migrate laboratory data, including updates. It performs two steps: runs the stored procedure sp\_sync\_ProcessChanges and runs a SQL to copy whole projects from LIMS to SSL\_Repo. The sp\_sync\_ProcessChanges stored procedure calls upon other stored procedures to effect changes to the tables in SSL\_Repo. First, it calls these sp\_sync\_Period, sp\_sync\_Organization, sp\_sync\_Area, sp\_sync\_Person to migrate changes from LIMS by looking for differences between the tables. Second, it looks for changes made to the tables analysis, procedure, analyte, calc, instrument\_set, master\_preparation according to entries in the LIMS change\_log table (these entries are recorded by triggers on the tables whenever changes are made) and migrates these changes. Third, it looks for changes made to the tables layer, lims\_pedon, lims\_site, project, layer\_analyte, result, analytical\_sample and migrates these changes. The SQL to copy whole projects from LIMS to SSL\_Repo collects a list of projects that have not been migrated before (not found in the LIMS table transfer\_project\_log), has a proj\_status of 'completed', and the private\_flag is ‘off’. Then the stored procedure sp\_sync\_LoadProject is the run against each of the projects. The sp\_sync\_LoadProject calls upon other stored procedures to insert the project/site/pedon/layer/preparations.

# 

# Appendix A – Acronyms

|  |  |
| --- | --- |
| **Acronyms** | **Definitions** |
| NASIS | National Soil Information Systems |
| KSSL | Kellogg Soil Survey Laboratory |
| LDM | Lab Data Mart |
| LIMS | Laboratory Information Management System |
| MLRA | Major Land Resource Areas |
| NCSS | National Cooperative Soil Survey |
| NHQ | National Headquarters |
| NSSC | National Soil Survey Center |

# Appendix B – Setting Up the Server

A server provisioned by CEC will either be an application server, a database server, a batch server, or a file share server. The server for KSSL LIMS is all four rolled into one. Yet, not all of the Windows Operating System services are made available nor even configured for a specific application. When working with CEC, it is important that the server is tailored to work with KSSL LIMS. Either the CEC representative may do this or an employee of the KSSL may do this provided they have an administrative account that has been given privileges to access and work on the server. Indeed, an employee of the KSSL must have an administrative account and be given the privileges to access and work on the server in order to maintain its operation. Currently, to gain an account, the employee must work with the NSSC’s POC to fill out a form naming the server. The employee must work with OCIO to gain the right to login on the server using Remote Desktop Connection. The CEC employee provisioning the server will set up the hard disks. The number of drives is unimportant. What is important is that at least four folders need to be created – one to hold Internet Information Services files (web sites and supporting folders), one for SQL Server files (database files, backup files, etc.), one for read-only files (files that may be shared, or not, but cannot be altered by anyone other than the KSSL employee with privileges), and one for read-write (files that may be shared, or not, and can be altered by anyone). The CEC employee must add the KSSL employee as a user with full control to each of these four folders. Once the KSSL employee has full control, they should set up the folders that the KSSL LIMS is expecting to be available. Currently, they are:

L:\IIS\Domains –web sites

L:\IIS\LIMS\_download – temporary files created by LIMS web pages

L:\IIS\NCSS\_labdatamart\_download –temporary files created by NCSS LDM web pages

L:\Read\_Only\CalcScripts –calculation script files used by the Calculation Engine

L:\Read\_Only\CalcService – Calculation Engine executable files with log file

L:\Read\_Only\Developer – documentation and developer notes of KSSL LIMS

L:\Read\_Only\IDAScripts – script files used by the LIMS Link process

L:\Read\_Only\MIR\_Spectral\_Library – spectral files collected from MIR analysis

L:\Read\_Only\MIR\_Truncated\_Library – files from the MIR Spectral Library but truncated to a wavelength range and formatted to OPUS and CSV

L:\Read\_Only\RaCA\_Spectral\_Library – files from VNIR spectroscopy collected during the Rapid Carbon Assessment project in the field

L:\Read\_Only\SIFScripts – script files used by the KSS LIMS to prepare sample information files

L:\Read\_Only\VNIR\_Spectral\_Library – spectral files collected from VNIR analysis

L:\Read\_Write\ACCESS\_database\_text\_files – files used in the creation of an Access database from the SSL\_Repo database for distribution via the NCSS LDM web site

L:\Read\_Write\Combine\_NASIS\_and\_NCSS – files used in the download of NASIS pedon data and update to the SSL\_Repo database

L:\Read\_Write\LIMS\_PROJECTS – project folders to hold attachments, i.e. reports, spreadsheets

L:\Read\_Write\MIR\_Spectral\_Library – spectral files from MIR analysis waiting to be migrated into the library

L:\Read\_Write\NCSS\_calculations – VB scripts that calculate common results for cooperating laboratories that were added to a staging database for insertion into SSL\_Repo

L:\Read\_Write\VNIR\_Spectral\_Library -- spectral files from VNIR analysis waiting to be migrated into the library

L:\SQL\_Backups – folders and files from the database backup jobs run by SQL Server

The following folders must be network-shared so that the KSSL LIMS and KSSL employees may access the files they contain: CalcScripts, IDAScripts, MIR\_Spectral\_Library, MIR\_Truncated\_Library, RaCA\_Spectral\_Library, SIFScripts, VNIR\_Spectral\_Library, and the entire Read\_Write folder.

One of the Windows services required by KSSL LIMS is the Internet Information System service (IIS). It is the service that hosts web sites which provide web pages and web services to users. The service is a feature of Windows Server operating systems but must be configured when the server is provisioned. The following pictures show how to access Roles and Features of Windows Server 2016 R2 indicating what roles and features must be set in order to support KSSL LIMS. These pictures may be invalid with future Windows Server OS versions, but the roles and features must be set in any case:

# Appendix C – Internet Information Service (IIS) service

# Appendix D – Windows Firewall rules

# Appendix E – LIMS Calculation Engine

# Appendix F – GnuPlot

# Appendix G – SQL Server Details