OS PROTOCOL AND PROCEDURE: DMP – DATA MANAGEMENT

PREPARED BY	**ORGANIZATION**	**DATE**
Tera Del Priore	SCI	05/01/2023
Cody Flagg	SCI	04/02/2019
Courtney Meier	SCI	01/02/2018
Sarah C. Elmendorf	DPS	01/05/2015
Kate Thibault	SCI	01/05/2015

APPROVALS	**ORGANIZATION**	**APPROVAL DATE**
Kate Thibault	SCI	05/05/2023

RELEASED BY	**ORGANIZATION**	**RELEASE DATE**
Tanisha Waters	CM	05/05/2023

See configuration management system for approval history.

The National Ecological Observatory Network is a project solely funded by the National Science Foundation and managed under cooperative agreement by Battelle. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Change Record

REVISION	DATE	ECO #	DESCRIPTION OF CHANGE
A	05/30/2014	ECO-01835	Initial release
B	01/05/2015	ECO-02633	Added information regarding recent developments of web interfaces and mobile applications for data entry
C	04/04/2016	ECO-03640	• Updated formatting to latest Protocol/SOP template (previous version did not have SOP sections) • Added more text to Background section • Added clarifying steps for data entry QA procedures • Updated acronyms in Table 2 • Section 8.2 – updates to best practices section
D	03/09/2018	ECO-05383	• Changed title from “Manual Data Transcription” to “Data Management” to better reflect the contents • Added text in Background section to address mobile data collection with no paper datasheets • Expanded Method section that provides more rationale and language describing data quality concepts • Updated definitions to better match digital data collection concepts • Previous Table 1 was deleted as it was no longer relevant. • Deleted old Table 2, no longer relevant • Added Table 4, Field Sampling Completeness table for Process Quality review • Added Table 5, Protocol Sampling Completeness table for Process Quality review • Replaced Figures 1 with a more up to date version of the data ingest process. • Added Figure 2, workflow diagram from field data collection to data quality review. • Re-arranged SOPs to emphasize the mobile/digital elements of data collection (SOPs originally started with paper datasheet collection workflow) • Added SOP A: general overview of data collection and data quality review process

REVISION	DATE	ECO #	DESCRIPTION OF CHANGE
			• Added SOP B: data quality methods for detecting duplicate records • Added SOP C: data quality methods for assessing process quality (complete record sets) • Removed old Appendix A “Best Practices for Data Entry and Data Quality Management” • Removed old Appendix B “NEON Specific Guidelines and Tips for Taxonomy related Data Entry” • Old Appendix C, “Data entry training checklist” is now Appendix A • Added Appendix B “QA/QC for Digital Hemispherical Photos” • Added Appendix C “Vegetation Structure QA/QC”
E	04/16/2019	ECO-06020	• Section 3.1.2: Added paragraph about how checklists should be developed and used for QA/QC • Section 3.1.3: Added a new table as “Table 1” – general description of parser validation rules. • Section 3.1.3: Added more descriptive text about the data ingest process. • SOP E: New SOP outlining how to create and use QC checklists, and when they should be implemented. • Removed Appendix B “QA/QC for Digital Hemispherical Photos”, this should go into the new protocol specific QC checklists that are being developed. • Removed Appendix C “Vegetation Structure QA/QC”, this should go into the new protocol specific QC checklists that are being developed. • Added Appendix B “How to Export Fulcrum Data”
F	03/20/2023	ECO-06963	• Migrated content into newest protocol template Rev L • Re-organized SOP’s order • Added SOP A.4: Review Remarks • Added SOP D: Quality Control Shiny Applications • Added SOP E: How to Use the Magpie Data Viewer Application • Added SOP H: Parser Error Resolution • Added Appendix B: How to Filter and View Fulcrum Data in the WebUI • Expanded Appendix C to include how to download full Fulcrum version history • Added Appendix D: Supplemental Parser Error Resolution Workflows
G	05/05/2023	ECO-07012	• Updated Table 2 • Updated the title of SOP H.2

TABLE OF CONTENTS

LIST OF TABLES AND FIGURES

1 OVERVIEW

1.1 Background

This document provides a cross-protocol set of data management instructions for the NEON Terrestrial and Aquatic Observation Systems. It includes a framework for data quality control and assurance measures that should be applied before, during, and after data collection, best practices for data collected on mobile devices, guidelines for transcription and storage when field and lab data are recorded with paper datasheets, and finally, a reference for how to troubleshoot, correct, and report data that fail to load into the NEON database.

Data management is the application of a "…consistent methodology that ensures the deployment of timely and trusted data" (Fisher 2009). There are two primary goals that staff should understand before engaging in the data management process. The first goal is to ensure that high quality data are collected, maintained, and delivered to end users by preventing or identifying data quality issues. *Data quality* issues can be defined as any condition that is an obstacle to the data consumer's use of those data (Sebastian-Coleman 2013). Data management in this context is the process by which we prevent, remove, or reduce obstacles to the effective use of data. The second goal of data management is therefore to establish a responsive process that identifies, measures, tracks, and resolves data quality issues.

The aim of quality control (QC) during data management is to prevent the introduction of errors throughout all stages of data collection, transcription (where relevant), and storage. The purpose of quality assurance (QA) is to detect and correct errors, and prevent future data quality issues. Identifying and resolving data quality issues early in data collection processes is particularly important to NEON's strategic goal of providing standardized, long-term data sets to the ecological community. It is critical that personnel understand the data quality process and their role in it at NEON.

This document is formatted such that the individual standard operating procedures (**Section 7**) can and should be referenced frequently before, during, and after data collection by field staff to ensure data quality control and assurance is performed correctly and any issues are communicated via the proper mechanisms.

1.2 Scope

This document provides a change-controlled version of Observatory protocols and procedures. Documentation of content changes (i.e. changes in particular tasks or safety practices) will occur via this change-controlled document, not through field manuals or training materials.

1.2.1 NEON Science Requirements and Data Products

This protocol fulfills Observatory science requirements that reside in NEON's Dynamic Object-Oriented Requirements System (DOORS). Copies of approved science requirements have been exported from DOORS and are available in NEON's document repository, or upon request.

Execution of this protocol procures samples and/or generates raw data satisfying NEON Observatory scientific requirements. These data and samples are used to create NEON data products, and are documented in the NEON Scientific Data Products Catalog (RD[03]).

1.3 Acknowledgments

N/A

2 RELATED DOCUMENTS AND ACRONYMS

2.1 Applicable Documents

Applicable documents contain higher-level information that is implemented in the current document. Examples include designs, plans, or standards.

AD[01]	NEON.DOC.004300	EHS Safety Policy and Program Manual
AD[02]	NEON.DOC.004316	Operations Field Safety and Security Plan
AD[03]	NEON.DOC.050005	Field Operations Job Instruction Training Plan
AD[04]	NEON.DOC.004104	NEON Science Data Quality Plan
AD[05]	NEON.DOC.014051	Field Audit Plan

2.2 Reference Documents

Reference documents contain information that supports or complements the current document. Examples include related protocols, datasheets, or general-information references.

RD[01]	NEON.DOC.000008	NEON Acronym List
RD[02]	NEON.DOC.000243	NEON Glossary of Terms
RD[03]	NEON.DOC.002652	NEON Data Products Catalog

2.3 Acronyms

Acronym	Definition
AOS	Aquatic Observation System
DEA	Data entry application
FSCI	Field Science
HQ	Headquarters
MDR	Mobile data recorder
PDR	Processed data repository
QA	Quality assurance
QC	Quality control
SSL	Sampling Support Library
TOS	Terrestrial Observation System

2.4 Definitions

Common terms used throughout this document are defined here, in alphabetical order.

Canary: A python script managed by NEON Science staff that produces a spreadsheet of current parser errors caused by records rejected by the parser.

Child record: In data entry applications, the nested form/screen where sub-sample data or multiple observations are recorded; child records inherit metadata from the parent record. The paper datasheet analog would be the rows where sample, trap, or observation specific data are recorded.

Data entry application (DEA): An electronic, protocol-specific user interface managed by NEON Science to provide controlled data entry that can be accessed through mobile devices or desktop/laptop computers. Currently, NEON DEAs are created and deployed via the Fulcrum platform. These applications are used by NEON field staff to submit data to the NEON Processed Data Repository (PDR).

Data quality dimension: Characteristics of data that can be measured through which data quality can be described and quantified.

Data quality context: The description of when and where data quality tasks are conducted within a workflow.

Data quality focus: The combination of data quality dimensions that build towards organizational data quality goals (example: a process quality focus is the assessment of both sampling completeness and timeliness).

Downstream application: An application that receives metadata from another application.

Field datasheet: A pdf document to be printed and filled out by hand in the field or the lab; datasheets are typically used when DEAs are not available.

Fulcrum: Software tool used to create and deploy NEON electronic data entry applications (DEAs).

Ingest workbook: A spreadsheet that outlines all the validation rules for a particular data product and is read by the parser upon an ingest attempt to validate data against the rules outlined.

L0 data: Level zero data i.e. raw data collected in Fulcrum DEA's and loaded via the parser.

L1 data: Published and processed L0 data that is made publicly available on the NEON Data Portal.

Load Delay: The application/protocol specific number of days before records in Fulcrum are ingested and evaluated by the parser. Number of days are counted from date of record creation. Records that successfully "pass" the parser validation rules are set to a read-only state; rejected records remain editable.

Load Group: The set of applications whose data are ingested at the same time. Applications related to the same protocol are generally within the same load group (e.g. Litter: Trap Deployment, Litter: Field Sampling, and Litter: Lab Mass Data).

LOV: The 'List of Values' is a list of constrained values accepted by the parser for a particular field in a data product.

Magpie: A NEON created and managed Shiny app used to view and QA/QC data collected in Fulcrum applications.

Metadata: Information that describes where observations or samples were collected, when those data were collected, who collected them, and other observations about the data.

Mobile data recorder (MDR): Handheld, field-portable equipment that runs protocol-specific data entry applications created by NEON Science.

NEON Data Portal: The public interfacing website to access NEON data as well as other resources and information regarding the observatory, found here**:** https://data.neonscience.org/

Sampling Support Library (SSL): A webpage housed on the NEON Sharepoint site which provides links to access all current documents and resources for OS protocols, IS instruments, plus field and lab equipment found at each domain support facility.

Sharepoint: A customizable Cloud-based content collaboration and management platform that allows organizations to store, organize, share, and access information.

Orphan Record: A record whose unique identifier (e.g. sampleID, individualID etc.) is not logically traceable to an upstream record. Field data are often related to downstream domain lab and/or external lab generated data. An example orphan record would be a lab sample whose sampleID does not match metadata for a field collection event (e.g. a Soil Moisture sample "GRSM_006-O-11.5-34-20180409-sm" *should* match an upstream Soil Core Collection sample "GRSM_006-O-11.5-34-20180409"; if it does not, the Soil Moisture record is an orphan.).

Parent record: In data entry applications, the first form/screen where metadata are recorded; parent records often contain and are linked to child records. The paper datasheet analog would be the "header" section where sampling location, sampling date, and field personnel information is recorded.

Parser: Data validation software that evaluates Fulcrum records against protocol specific rules. The parser either "passes" and locks records that successfully pass validation and moves data on for web publication, or "rejects" records that fail validation from the publication process resulting in parser errors.

Parser Errors: Records that fail the front-end validation (Parser) and are rejected and not ingested into the database and will result in an error with an accompanied error message that is viewable within the Canary spreadsheet. This prevents 'bad data' from getting into the database.

Quality Assurance: Methods for detecting and correcting errors, and for preventing future data quality issues, abbreviated as 'QA'.

Quality Control: Methods for preventing the introduction of errors throughout all stages of data collection, transcription, and storage, abbreviated as 'QC'.

Record: The collection of data values that describe a sampling entity or event. If data are viewed in a spreadsheet or tabular format, each row is a record and each column is an attribute of that entity or event.

Recorder: The person who enters the data.

ServiceNow: Software tool used for problem/incident tracking and resolution.

Shiny App: Shiny is a web application framework that enables developers to build an interactive interface for data viewing and analysis.

Syncing/synchronization: The process of transferring digital records from a DEA to a cloud database for long-term storage. On mobile devices, this specifically refers to the act of pressing the “sync” button. On laptop or desktop computers, syncing occurs as soon as a record is saved.

Upstream application: An application that is the *source* of metadata such as plotID, collectDate, sampleID etc. These are typically *field* applications such as BET: Field Sampling.

UUID: A Universally Unique Identifier value used to identify information in a system. The NEON database creates a UUID for every data record ingested.

3 METHOD

This protocol provides Standard Operating Procedures (SOPs) for all aspects of data management that are relevant to field staff.

- **SOP A**: Mobile Data Recorder – describes how data collected with mobile data recorders should be managed and reviewed
- **SOP B**: Paper Datasheet Quality Checking – addresses management and review of data collected on paper datasheets.
- **SOP C**: Using Checklists for Quality Control – describes checklists available to field staff to use to perform data quality checks
- **SOP D**: Quality Control Shiny Applications – describes tools available to field staff to perform data quality checks
- **SOP E**: How to Use the Magpie Data Viewer Application – describes the Magpie data viewer application and how and when to use it.
- **SOP F**: Referential Integrity Data Quality Review – describes how to perform data integrity quality checks
- **SOP G**: Detecting Process Quality Issues – describe how to perform data process quality checks.
- **SOP H**: Parser Error Resolution – describes the workflow to resolve data errors.

In addition, each protocol may contain instructions, under the "Data Entry and Verification" SOP, which supplement procedures listed here.

3.1 Data Quality Framework

Data quality can be defined as the degree to which data meet the expectations of data consumers per the intended use of those data (Sebastian-Coleman 2013). More specifically, the proposed framework helps to clarify answering what (data quality dimensions), where and when (data quality context), and why (data quality focus) of the entire data quality process. While these expectations are not objective and can vary widely among the broad range of potential NEON data users, there are dozens of data quality attributes that can be measured, reported, and assessed by users. These attributes can be directly measured by NEON staff as well, and are summarized as **data quality dimensions** (from Sebastian-Coleman 2013):

1. **Completeness**: the expected data exist and are complete according to pre-defined characteristics (e.g. all relevant attributes are measured and the amount of data is correct).
2. **Timeliness**: data are collected, processed, and delivered at the expected time according to a set schedule or when an event occurs.
3. **Validity**: data conform to the expected syntax, allowable type (string, integer, date etc.), ranges, or other pre-defined rules.
4. **Consistency**: the absence of variance or change in comparison to some pre-defined expectation.

5. **Integrity**: data contain all relevant relationship linkages (e.g., no orphans or missing child records).

Where and when data quality is controlled and assessed is just as important to consider as *how* to measure quality, and is referred to as the **data quality context**. At NEON, the following contexts should be familiar to experienced staff:

- **Pre-Collection QA/QC**: activities that take place prior to data collection, aimed at maximizing data collection consistency – e.g. protocol training, calibration for field staff collecting data (for observations e.g. phenophase evaluation or estimating percent cover), etc.
- **Collection QA/QC** (or "field QA/QC"): activities that take place during or immediately after data collection, but before leaving a field site, aimed at maximizing data validity and completeness.
- **Post-Collection QA/QC** (or "office QA/QC"): activities that generally take place after field collection, and are designed to maintain data completeness, sampling timeliness, and between bout observation consistency.

Measuring multiple data quality dimensions simultaneously is important in assessing the quality of underlying processes of data collection within an organization. These are referred to as a **data quality focus**, of which the OS subsystem is primarily interested in or focused on:

- **Referential Integrity (validity and integrity)**: the absence of duplicate, orphaned, and/or 'childless' records from datasets.
- **Process Quality (completeness and timeliness):** the presence of all expected data and/or samples within a well-defined time interval.
- **Data Plausibility (validity and consistency):** a qualitative assessment of whether data are "reasonable" or not. Data Plausibility issues and methods are NOT addressed in this document, but should be documented within protocols as a "Data QA/QC Checklist" (as indicated in **SOP C**).

3.2 Data Quality Review Workflow

Mobile applications are the preferred mechanism for data collection and entry. The first stage of the process is to review critical field metadata values soon after collection, preferably before leaving a field site. Critical metadata values, such as plotID or collection date, are important data that cannot be easily inferred and/or corrected after staff leave a field site, and include data that pass DEA validation rules but could be ***inaccurate*** or ***incorrect*** in some way (e.g., the wrong plotID was accidentally selected; see **Box 1** and **SOP A.1** for specifics).

It is recommended that field staff use digital tools to detect, report, and correct issues, such as Shiny applications created and managed by Field Science and Headquarters staff scientists. If a Shiny QAQC application does not exist for a given data product, it is recommended that data are first analyzed with filters and Pivot Tables in Microsoft Excel to summarize patterns in large datasets. Viewing raw data can be useful, but is the least desirable option because it's easy to overlook errors when faced with hundreds of data values.

Referential quality checks are concerned with maintaining the "referential integrity" of datasets by ensuring that ingested records are free of potentially misleading data (e.g. duplicate records with slightly different data values are confusing to end users) and have concrete relations to relevant records (e.g. records have all necessary metadata). Duplicate records are a major concern as even a single duplicate record will cause an entire batch of data to fail upload for many NEON data products. Duplicate records can be identified as the presence of two or more records in a data set with the exact same sample identifier.

Ideally, field staff review data for referential quality (e.g., duplicates) once an entire bout's worth of data have been entered into (or collected with) a protocol-specific DEA. The exception is for bouts that extend for long periods (i.e. bouts longer than 2 weeks), in which case data reviews should occur before the bout is completed. At a minimum, data should be reviewed before the load delay is reached for a data product, as this simplifies the data editing and record deletion process; see **Section 3.3** below *Timing ("load delays")* for more details regarding load delays. Technicians use the Magpie data viewer application to conduct QA/QC tasks that are standard across multiple OS protocols. Magpie can be accessed via the Data Management page of the Sampling Support Library (SSL), or via the NEON Aviary, and contains instructions on how to use the interface. Technicians assess the results of each query conducted with Magpie, then determine whether corrective action needs to be taken, such as submitting record deletion requests or correcting data, see **SOP E** and **SOP F**.

Identifying process quality issues allows NEON to improve data collection procedures over time while also ensuring that all expected data/samples are collected and published. Incomplete data sets can be particularly detrimental to scientific end users as fewer samples or data points decrease the ability to detect significant ecological changes or trends. Analyzing process quality includes assessing whether domains and field staff are correctly implementing NEON protocols, and encompasses 'bout-level' metrics such as sampling completeness and sampling timeliness, as prescribed by protocols. Quantitatively analyzing data for process quality issues means checking that:

- Expected number of field records are present per site and protocol ("Field Sampling Completeness")
- Expected number of records are present for all SOPs in a protocol per sampling location, bout, and protocol ("Protocol Sampling Completeness")

Finally, domain staff should use existing QA/QC checklists for assessing protocol-specific data quality criteria. Most all sampling protocols should contain a QA/QC checklist detailing key data attributes to review before data are ingested. These checklists can be found on the landing page for the respective protocol on the SSL. Many checklist items describe important aspects of a data product's quality that are not, and may never be, detected by the OS Parser. For example the Parser cannot inspect the photographic quality of images collected for the measurement of Leaf Area Index, thus the QA/QC checklist highlights the following checks:

- "Are images in focus?
- Is ISO within acceptable range?
- Is f/stop within acceptable range?"

Checklists should be used by staff at all levels throughout the data review process, and are intended to ensure consistency, to convey best practices, to help avoid poor decision making, and to sustain patterns of success.

3.3 Data Validation and Ingest Process

Validations

Field staff must understand the process and timing of how data are transferred, or "ingested", from data entry applications to the NEON Data Portal, as digital records are only directly editable for a limited time frame. Automated data validation checks are documented in a data product's "Ingest Workbook", which details a two-stage process. The first round of data validation occurs directly within Fulcrum applications. Data that do not meet the expectations of these rules are generally prevented from being saved and synchronized with the Fulcrum cloud database. Fulcrum validation rules are limited to checking data within a single parent and if applicable child record(s) within that parent record, thus rarely checking for referential integrity issues. For example, these validation rules cannot check for duplicate sampleID or barcode values across multiple parent or child records. A piece of software known as the "OS Parser" (or "parser" for short) carries out the second round of validations and makes up for some of the limitations of the Fulcrum rules. The parser executes several referential integrity checks to prevent duplicates and orphaned data from entering the database, in addition to checking the Fulcrum rules again (**Table 1**). This second check of Fulcrum rules is done to ensure that the applications have been designed and implemented correctly by application developers.

Table 1. OS Parser Data Validation rules and associated error messages when incoming data do not meet requirements.

Method	Checks for…	Error Message
Geographic Range Constraint (integrity, validity)	Valid plotID names for a specific protocol are selected. Sampling cannot occur at certain types of plots or transects.	NAMED_LOCATION_TYPE
Sample Identifier Check (integrity)	Whether sample identifiers are unique within the system and/or whether downstream sample identifiers correctly match upstream identifiers e.g. checks for duplicate and orphan records. • 'ReferenceCount' indicates duplicates within the batch of records being ingested	'ReferenceCount' 'Sample already exists' 'Samples does not exist'

Method	Checks for…	Error Message
	• 'Sample already exists' indicates a record attempting to ingest is a duplicate of a record already ingested • 'Sample does not exist' typically indicates a required upstream record is missing	
Barcode Check (integrity)	Whether scanned barcodes are unique, whether a barcode has been associated with more than one sample type (e.g. a barcode scanned in ALG and again in CFC would fail validation)	'Has different tags' 'Has different barcodes' 'Configuration error changing tag'
Taxonomic Value Constraint (validity)	Only animal or plant taxa that have been identified to exist within a domain's geographic boundaries are selected.	ELEMENT_OF
Required Fields (completeness)	Whether required fields have been filled out; conditionally dependent on data entered, whether certain fields have been left blank. Fields are usually required as a way of delivering a minimum usable data product for end users (e.g. many data are unusable if collection dates are missing).	REQUIRE, REQUIRE_NULL
List of Values Constraint (validity)	Whether selected data values match against a pre-specified list. More of a check on the application's design because end users cannot enter custom values.	LOV
Numeric Range Constraint (validity)	Whether entered values fall within a specified range.	GREATER_THAN, LESS_THAN

Timing ("load delays")

Every record entered into a DEA is "stamped" with a creation date (**Figure 1**). Once a record is synchronized with the cloud database, a "load delay" countdown begins that starts from the record creation date. The load delay defines how long a record can be edited by field staff. Load delays vary from 14 to 365 days, and protocol-specific load delays are available to view per DEA via the Magpie Shiny application. The Magpie application link, like links to all OS Data QAQC Shiny applications, is available from The Aviary (https://den-prodshiny-1.ci.neoninternal.org/TheAviary/). Once a record exceeds the load delay, the data are tested by the OS Parser. Records that "pass" Parser validation are then locked and the data eventually appear on the NEON Data Portal. Field staff can view but not directly edit locked records, with an exception for records re-opened on a tablet that were not synced properly after creation, then the Fulcrum application can roll back the record to have a load status of NONE. Records that "fail" validation generate error messages that are collated across protocols and distributed to FSCI and HQ staff by a tool named Canary (see **SOP** Error! Reference source not found.). The OS parser attempts to re-ingest most failed records nightly, except those records in the Shipping application. Failed shipping records are set to skip ingest for a week, this allows for other records to load. The same error messages for data issues are provided until they are addressed. Failed records must be corrected by field staff. Changes to locked and loaded data records may be necessary if field staff determine the need for updates (e.g. a valid taxonID was selected, but is known to be the incorrect sub-species for a particular site), or if field staff detect data errors not caught by the parser (e.g. a valid but incorrect collection date was entered). These updates must be submitted through the Magpie application (i.e. “Update Requests”) and are handled by HQ staff.

Difficulties

One major feature of the OS data ingest system is that records are not ingested individually; instead, records are ingested in ‘batches’ or ‘sets’ referred to as ‘load groups’. Load groups contain all the records from a suite of related applications for a given domain. This means that a single record fail results in all related records within and across applications failing at the same time (all records will also receive a load_status of “PARSE_FAIL”, which can be misleading for troubleshooting). All related records are rejected as there may be duplicate records present that need to be resolved or deleted before the data can be ingested. OS data are sometimes heavily interrelated across field, domain lab, external lab, and shipping applications. Since metadata are “inherited” across these applications, errors in one application may cause the Parser to flag records in other linked applications. This inter-application complexity can make resolving parser errors difficult if one does not understand what the Parser is checking for or how metadata are related across applications. This is why it is important to have fully trained staff members perform data QA/QC activities. Experienced staff understand the data structure and know where to obtain the resources to effectively and efficiently correct data issues. Finally, post-parser “Update Requests” can be very time consuming and difficult to work through because records can no longer be edited through Fulcrum applications.

As a rule of thumb, experienced Field Science staff should spend no more than 1 hour trying to determine the root cause of any given parser error (or set of related parser errors). If the root cause and required steps to resolve the error(s) remain unclear after 1 h of due diligence, submit an Incident via ServiceNow and engage the Science team for assistance.

Standard Operating Procedures (SOPs), in **Section 7** of this document, provide detailed step-by-step directions, resources, and best practices for implementing this procedure.

The value of NEON data hinges on consistent implementation of this protocol across all NEON domains, for the life of the project. It is therefore essential that field personnel carry out this protocol as outlined in this document. In the event that local conditions create uncertainty about carrying out these steps, it is critical that technicians document the problem and enter it in NEON’s problem tracking system.

Quality assurance is performed on data collected via these procedures according to the NEON Science Data Quality Plan (AD[04]).

4 SCHEDULE

4.1 Data Entry and Review

It is preferable to use MDRs and NEON data entry applications to capture data electronically in real-time whenever possible. Data collected on an MDR should be synchronized **as soon as possible.**

Data collected on **paper datasheets should be entered within 14 days of collection** for protocols with long bouts (e.g., Vegetation Structure) *or* within 14 days of the end of a bout for protocols with short bouts (e.g., Mosquitoes).

All digital records, whether originally recorded with an MDR or paper datasheet, must be reviewed and edited ***before*** the load delay occurs. Load delays per protocol are defined on the "About" page of the Magpie application: https://den-prodshiny-1.ci.neoninternal.org/Fulcrum-QA/

The following tasks should be completed in descending order of priority **within 14 days of collection or the end of a bout**:

- Field datasheets and data entry application records should be quickly reviewed for critical errors by technicians on a daily basis (**SOP A.1**, step 2). Critical errors are incorrectly recorded sampling locations and sampling dates that cannot be accurately corrected after leaving the field.
- Sync MDR data when a web/data connection is available (**SOP A.1**)

The following review tasks should be completed in descending order of priority and completed within the **load delay period**:

- Perform referential integrity reviews for a given protocol before data are ingested and locked (**SOP F**)
- Perform a process quality review for a given protocol at the end of a sampling bout and field season (**SOP G**)
- Perform additional quality control procedures outlined in protocol specific "Data QA/QC" checklists (**SOP C**)
- Scan and store field datasheets (where applicable) on a weekly or bout-level interval (**SOP B.2**)
- Manually enter paper data (where applicable), starting with the earliest collection date. *Every paper data entry bout should include paper datasheet data quality checking* (**SOP B.4**)

Scheduling considerations

Data QA/QC tasks can take significant time, therefore it is important to schedule for and allow ample time to complete these tasks. It is important to set aside time a) at the end of the field day to QC data and b) after the bout/end of the week/after lab processing/etc to QA the data. Time is scheduled for data QC/QA activities after each scheduled field or lab sampling activity per protocol and can be found on the master schedule on the NEON Sharepoint.

5 SAFETY

This document identifies procedure-specific safety hazards and associated safety requirements. It does not describe general safety practices or site-specific safety practices.

Personnel working at a NEON site must be compliant with safe field work practices as outlined in the Operations Field Safety and Security Plan (AD[02]) and EHS Safety Policy and Program Manual (AD[01]). Additional safety issues associated with this field procedure are outlined below. The Field Operations Manager and the Lead Field Technician have primary authority to stop work activities based on unsafe field conditions; however, all employees have the responsibility and right to stop their work in unsafe conditions.

6 PERSONNEL

Technicians entering data should familiarize themselves with the protocol and its datasheets for which they are entering data. As a best practice, technicians transcribing paper data sheets should be those who collected the field data, or be personnel trained in the specific protocol. Lead field staff should take a guiding role in directing all data management activities, from paper datasheet data entry and data quality checking (DQC) to data quality review.

6.1 Training Requirements

All technicians must complete protocol-specific training as required in the Field Operations Job Instruction Training Plan (AD[03]). Additional protocol-specific required skills and safety training are described here.

Field staff should have basic knowledge of the software used to host data entry applications and should have familiarized themselves with the relevant data ingest sheets and protocol-specific applications involved in their work.

6.2 Specialized Skills

Prior experience viewing, sorting, filtering, and manipulating data in spreadsheet software such as Microsoft Excel is desirable but not required.

7 STANDARD OPERATING PROCEDURES

SOP Overview

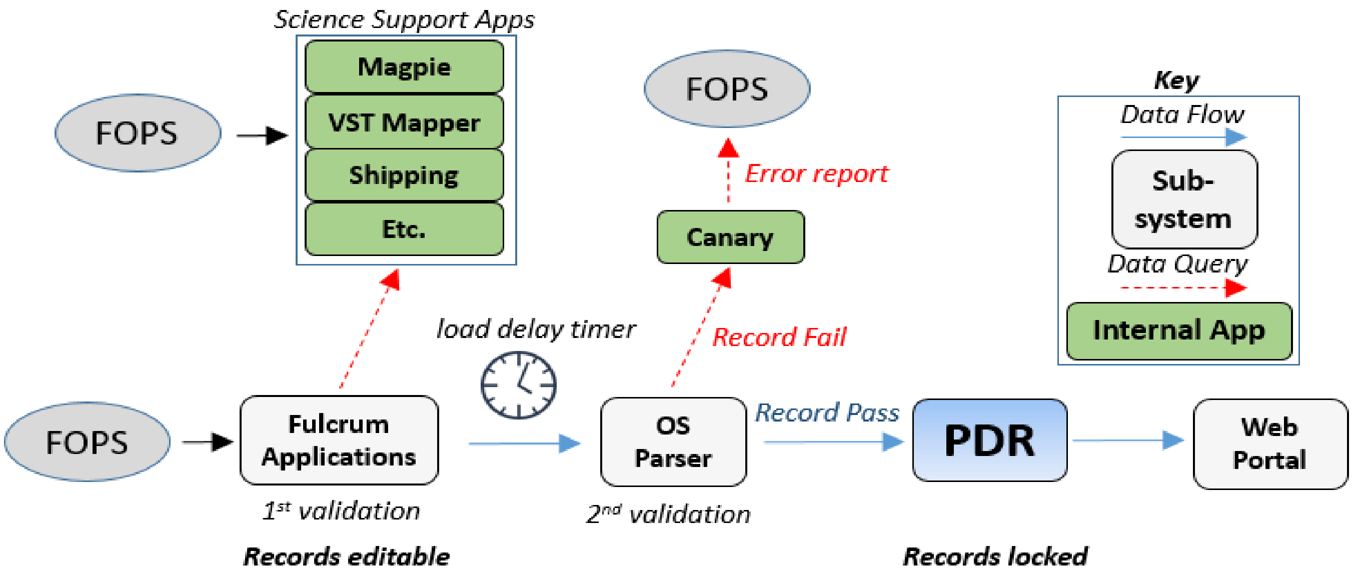


Figure 1. Observatory System data ingest process. Records begin moving through the ingest process once the “load delay”, or number of days since record creation, is exceeded. Data are automatically validated twice in the process, before being stored in the Processed Data Repository (PDR) and accessed via the NEON Data Portal. Data for specific protocols are manually validated with various “Science Support Applications”. The Canary tool analyzes and distributes error messages to Field Science and HQ staff when records fail to pass through the OS Parser.

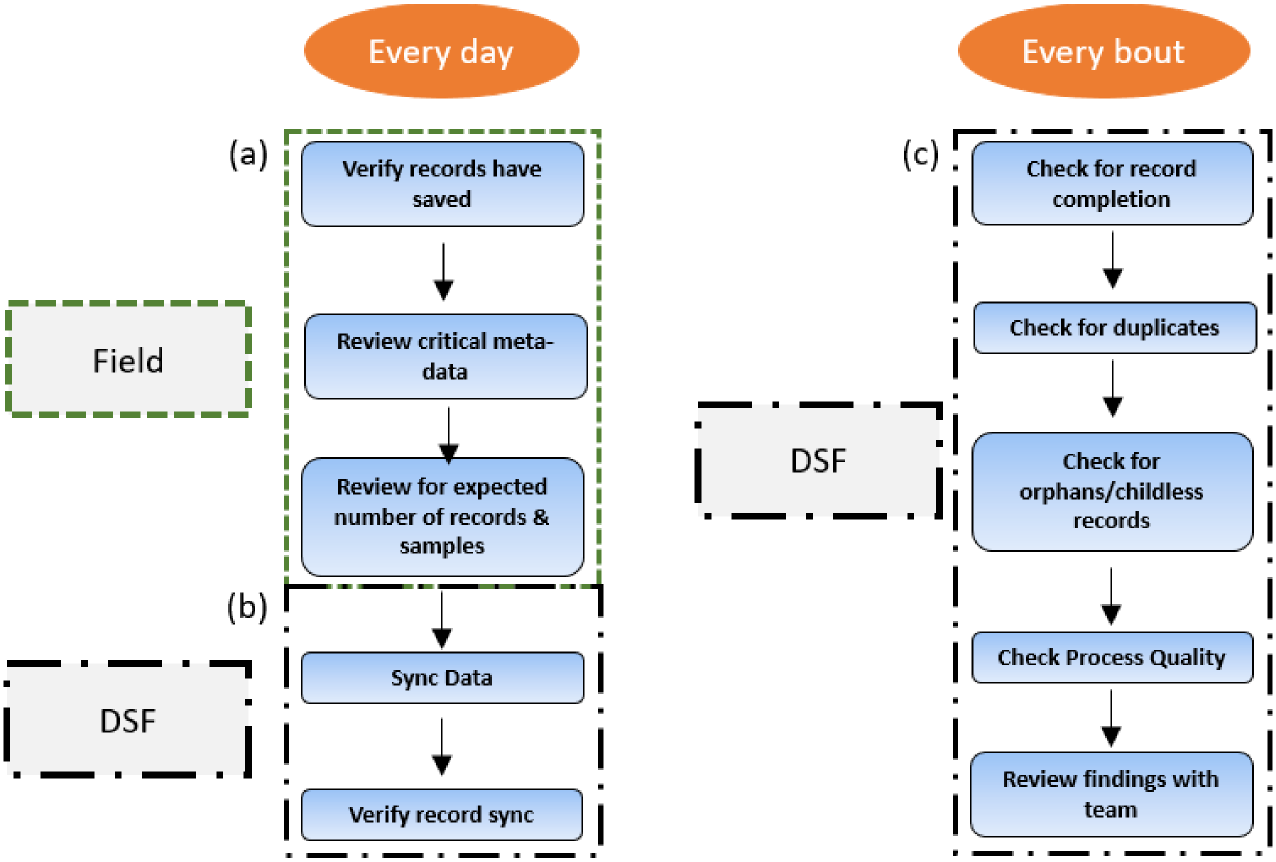


Figure 2. Workflow diagram for reviewing data quality per day and per bout. Dashed boxes represent *where* data should be reviewed, orange circles represent *when* in the workflow review steps should happen.

- **SOP A**: Mobile Data Recorder: General Data Quality Workflow for all Protocols
- **SOP B**: Paper Datasheet Quality Checking
- **SOP C**: Using Checklists for Quality Control
- **SOP D**: Quality Control Shiny Applications
- **SOP E**: How to Use the Magpie Data Viewer Application
- **SOP F**: Referential Integrity Data Quality Review
- **SOP G**: Detecting Process Quality Issues
- **SOP H**: Parser Error Resolution

SOP A Mobile Data Recorder: General Data Quality Workflow for all Protocols

Ensure that the person conducting data entry has read through the training materials associated with the protocol-specific data entry application, and that they have practiced entering data using the *Training* version of that application – i.e., the [CERT] version. A training checklist for mobile data recorder data entry is provided in **Appendix A**. Staff entering data for the first time should work through the checklist, in combination with reading the SOPs below, to ensure an understanding of the NEON OS Data Quality Workflow.

A.1 Every Day: Before Leaving a Field Site

Context: Collection QA/QC

The component of the Quality Workflow that is implemented daily before leaving the field site is shown in **Figure 2a**. Steps include:

1. Verify that entered data have been saved on the MDR
    a. For parent-child records, be sure to save the **parent** record. Saving child records does NOT mean that a parent record has been saved.

    NOTE: It is a common misunderstanding that the data entry application will automatically save any entered data. A person who is not the data recorder should double-check that the recorder has properly saved all data before leaving a field site. All too often, data are entered and then discarded by accident only for the data loss to be recognized after leaving a field site.

    WARNING: Records can be saved in a temporary state called a "draft" when one or more required fields are left blank (appears as a yellow pencil icon on mobile devices). ***Draft records are never synced to the cloud*** and should be completed as soon as possible to prevent data loss – i.e., the draft record will only exist on the device used to collect data.
2. Review critical metadata fields for accuracy and completeness. This step is important to prevent incorrect, but technically valid data from being entered (**Box 1**).
    a. Check for valid and correct *sampling location* values (e.g. siteID, plotID, clipCellNumber, trapID etc.)
    b. Check for valid *sampling date and time* values (e.g. setDate, collectDate, setTime, collectTime etc.)
    c. Check for valid *event identifier* values (e.g. weekBoutBegan, yearBoutBegan, eventID, etc.)
    d. Check that the expected number of records has been created and saved on the MDR.
    e. ***Examples***:

i. For many data entry applications, one parent record is often created per sampling location.

  1. If the field crew is expected to sample 10 mosquito traps in one day, there should be 10 mosquito records on the MDR.

ii. In other data entry applications, one parent record represents a single sampling location, and the child records represent sub-samples within that same location.

  1. If a field crew is visiting three plots to take digital hemispherical photos there should be:

    a. Three parent records (one per plot).

    b. Either 12 or 24 child records per parent record (12 or 24 photos taken per plot).

Box 1. Inaccurate but technically valid data

Definition: Entered data values that pass DEA validation rules but are mis-recorded or factually incorrect. Inaccurate data records are particularly troublesome because while they can often be identified as incorrect, how to correct the data is not always obvious or possible. Delaying the data review process leads to the loss of useful information from field staff that might help rectify errors.

- ***Example***: Data recorded for plot “X” when observations are actually from plot “Y”
  - Actual plotID = “CPER_012”; recorded plotID = “CPER_014”
  - *Results in*…a data set with fewer records than expected AND duplicate records; lower process quality and spatio-temporal data gaps that reduce statistical power.

- ***Example***: A recorded tree diameter that is within range but an order of magnitude off from the actual measurement
  - Actual Tree Diameter = “12 cm”; Recorded Tree Diameter = “120 cm” (min value = 1 cm, max value = 400 cm)
  - *Results in*…a data set with more outliers; reduced confidence in the plausibility of the data; inaccurate plot-level biomass estimates.

3. If field samples were collected:

  a. Check that the number of samples matches the number of recorded digital records.

  b. Check a subset of sample labels and/or barcodes for accuracy.

  c. At a minimum, check 10% of the total samples or 10 samples (whichever is greater).

    i. For samples with barcodes, the barcodes can be scanned with the MDR via the “Search” function or using a handheld barcode scanner

      1. Records with a matching barcode will appear in the record list *if they exist.*

        2. **If no records** are returned after scanning a barcode, this means the barcode wasn't initially scanned correctly or at all, and you should find the appropriate record and ensure that the barcode is scanned with that record.
        3. **For records that appear** after scanning, verify that the record's metadata match the information on the sample.
        4. **If more than one record appears** after scanning, a barcode has likely been scanned more than once and associated with different unique samples. This is considered a duplicate record situation, and should be resolved before data are loaded.
    - ii. For non-barcoded samples, the information on the sample should match metadata in the record.

A.2 Every Day: Returning to the Domain Support Facility or Field House

Context: Post-Collection QA/QC

The component of the Quality Workflow that is implemented daily upon returning to the domain support facility or Field House is shown in **Figure 2b**. Steps include:

1. Sync the MDR's data
    - a. Be sure that the device is connected to a secure Wi-Fi network or has a data connection before attempting to sync.
    - b. The synchronization mechanism can fail for multiple reasons, meaning that data are not fully transferred from the MDR to the cloud database.

    NOTE: Un-synced draft records are isolated to the device they were collected on. They will not appear on another device or in a web browser because the data have not been properly transferred.

    IMPORTANT: In isolated cases, if a LOADED record is re-opened on a tablet that was not synced properly after creation, then the Fulcrum application can roll back the record to have a load status of NONE and thus the parser will attempt to re-load the record resulting in a duplicate record!
2. Verify that all MDR data have been synced: Use a web browser to check that records have transferred from the MDR to the cloud.
    - a. Navigate to the appropriate DEA (application names are identical between the MDR and web browser).
    - b. Sort the data table by the "Created date" column (this is the date and time a record was created) by selecting "Sort Dec -> Jan".

   c. The most recently created records will appear at the top of the table.
   d. Locate and identify the records that should have been synced from the MDR.
   e. If not all expected records appear in the table, double-check the MDR to see if the syncing process failed and try to sync the device again.

3. If physical specimens were collected:
   a. Transfer samples to the appropriate storage containers and locations.
   b. Record this information in a sample inventory.
   c. Verify that the number of samples stored matches the expected number of records.

A.3 Every Bout: Data Quality Assurance

Context: Post-Collection QA/QC

This section gives a brief overview of the rest of the data quality review process, the components of which are shown in **Figure 2c**. Expanded step-by-step details are provided below in **SOP F** and **SOP G**.

1. Check that all entered records have been completed.
   a. All data values entered into each record:
      i. Certain applications allow users to save records without entering key metadata in order to match the field workflow. These data values should always be recorded even if they are not specifically required in the application, else records will be rejected by the ingest system.

      Example: Mosquito Collection ("MOS") application records can be saved without a **collectDate** or **collectTime**, however records ingested without these values will be rejected by the OS Parser.

   b. All related records have been entered (for protocols with multiple SOPs and data entry applications)

      Example: Herbaceous Clip Harvest ("HBP") has two applications: Field and Lab. One Lab record should exist for each Field record where targetTaxaPresent = "Yes".

   c. Follow additional protocol-specific instructions from the "Data Entry and Verification" SOP within each protocol.
   d. Check for duplicate records: Use the Magpie application to find and resolve duplicate records (**SOP E.1**).
   e. Prevent orphan records from being created: The parser will detect and reject orphan records

        i. There are currently no tools to find orphan records, however field staff can implement procedures that prevent creation of orphan records (**SOP E.2**).

2. Perform quantitative process quality review:
    a. Check that all expected digital records for a given protocol and bout exist.
    b. Check that all expected sampling locations for a given protocol and bout have been visited.
3. Perform additional quality control review outlined in protocol specific "Data QA/QC" checklists (**SOP C**).
    a. Check dataset attributes that are not detected by the parser

A.4 Every Bout: Review Remarks

A 'Remarks' field is usually present in the data recording applications to allow Field Science staff to comment for end users about a particular circumstance regarding the collection of field or laboratory data. Because remarks are published to the end user, the remarks field should be written professionally and only used to communicate pertinent information to an end user that isn't captured elsewhere within the data record. Remarks should not be used for internal information such as placeholders, reminders, or notes for field staff, unless a separate "internal remarks" field exists in the app. **Table 2** outlines some examples of useful remarks that should be included in a record and those that are not useful and should be omitted.

Table 2. Example remarks, both relevant and non-relevant for end users from various protocols.

Subsystem	Protocol	Remark	Useful (Y/N)	Explanation
AOS	INV	Partial sample collected with small sieve	N	Unclear – not enough detail
AOS	INV	Sediment in 19.5m tow. Tow taken at 19m instead	Y	Informs end user depth is different than expected
AOS	APL	Crispy brown morphospecies	N	Redundant – recorded in morphospecies app
TOS	MAM	Bad door	N	Internal note – irrelevant to end user
TOS	MAM	Nontarget: amphibian	Y	Provides end user data not available elsewhere
TOS	MOS	Catch cup found on ground	N	Redundant – recorded in CatchCupStatus
TOS	MOS	Parasitic mites present on mosquitoes within catch cup	Y	Provides end user data not available elsewhere

SOP B Paper Datasheet Quality Checking

Paper datasheets can be used as a backup data recording method in cases of adverse field conditions, equipment failure, or missing equipment, however all efforts should be made to enter data into an MDR. Data recorded on paper should be entered within 14 days of collection.

B.1 Every Day: Review Field Datasheets before Leaving the Field

1. Review the field datasheets for completeness:
    a. Check BOTH sides of the datasheets for extra taxa or notes.
    b. Be sure to clarify and annotate any shorthand notes on the datasheet; don't assume the shorthand will be meaningful to staff in the future.
    c. Finally, check that all required fields/columns have been filled out.
2. Protocol leads should review critical metadata values on datasheets at the end of the sampling day, similar to steps 1 and 2 outlined in **SOP A.2**. Critical metadata fields should be reviewed for accuracy and completeness (see **Box 1**):
    a. Check for valid and correct *sampling location* values (e.g. siteID, plotID, clipCellNumber, trapID etc.).
    b. Check for valid *sampling date and time* values (e.g. setDate, collectDate, setTime, collectTime etc.).
    c. Check for valid *event identifier* values (e.g. weekBoutBegan, yearBoutBegan, eventID, etc.).
    d. Check that the expected number of rows has been recorded.
3. If physical specimens were collected:
    a. Check that the number of samples matches the number of rows on the datasheet.
    b. Check all sample labels for accuracy.

B.2 Every Bout: Scan Field Datasheets

1. Scan reviewed and annotated datasheets.
2. Save scanned datasheets in the folder designated on the NEON shared drive, such as N:\DSF\Workspace\Domain XX\Subsystem\etc using the following naming convention: "moduleAbbreviation_fds_SAMPLINGLOCATION_YYYYMMDDx", where:
    a. moduleAbbreviation is three letter code for protocol such as PHE, APL, etc.
    b. 'fds' refers to field datasheet; 'lds' should be used for lab datasheets.

c. SAMPLINGLOCATION is the plotID (i.e. siteID_plotNumber e.g. "HARV_003") which can also be appended to the filename, after the siteID, to avoid confusion if multiple field crews have been working at the same site and module for a given date.

d. 'YYYYMMDD' is the most recent date on the datasheet;

  i. 'x' is an optional character to be used in the event of multiple files (representing a-z). Multiple files can result from multiple datasheets needed to accommodate the quantity of data collected on a given date for a given module and/or a module requiring multiple distinct field (or lab) datasheets. If multiple files are not needed, there is no need to use 'x'.

Example: 'phe_fds_CPER_20130710b' (this example is 22 characters long, which is less than the maximum length allowed by some models of scanners).

B.3 Every Bout: Data Entry Procedures

1. Be sure you are connected to the NEON network (use of VPN may be necessary at remote sites). Use only supported web browsers, Google Chrome, Mozilla Firefox, or Microsoft EDGE, for data entry. Using an unsupported browser may result in lost data. Supported browsers may change in the future.
2. Login to the webpage that hosts NEON data entry applications (e.g. Fulcrum) and navigate to the appropriate, protocol-specific application. Links and application names can be found on the protocol-specific pages of the Sampling Support Library.
3. The person conducting data entry must have practiced entering data using the *Training* version of that application.
   a. **Training Applications (CERT)**
      i. The NEON data entry webpage provides a copy of each data entry application for the sole purpose of training. These are labeled with the DEA name plus CERT for 'certification'. Technicians should use these to enter test data for training purposes, as the data are not stored in the NEON data repository.
      ii. New technicians must practice data entry in CERT applications *for each protocol where they plan to enter data*. Domains should consider generating test data that highlight common local data entry issues to assist with training.
      iii. Entered training data should be exported as a .CSV and reviewed by experienced field staff.
4. Once training requirements are met for that application/protocol, data can be entered into the protocol-specific *Production* (PROD) version of the data entry application.

   a. **Production Application (PROD)**

      i. The data entry webpage provides data entry applications for actual field and lab data entry. These data will be stored in the NEON data repository and published on the NEON data portal.

      ii. It is difficult to locate and fix incorrectly entered data once they have been locked and transitioned to the NEON PDR. *Extra time and effort should thus be devoted to careful data entry and data review*.

5. Data entry applications are customized for each individual protocol. Detailed training for each protocol interface is available in the 'Supporting Documents' section of the protocol page in the SSL, but here are some general rules:

   a. Sync MDR devices often in order to ensure that up-to-date versions of all DEAs are available.

   b. Remember that most of the applications have a 'nested' structure, meaning that some of the higher-level information (plotID) is entered once but applies to a group of samples.

   c. Data entry through a data entry application is a multi-step process:

      i. The 'Create' button (+) will allow you to create a new record

      ii. Once all required information about that record has been entered, use the '**Save**' button (✓) to save the data and store it locally.

   <u>**NOTE**</u>: Once records are created, saved, and synced, they cannot be deleted without submitting a request to NEON HQ.

      iii. The data entry application will allow you to review and edit data you have recorded until the load delay expires and the data are ingested to the NEON data repository. Review can be accomplished by clicking on the 'Quick View' option for a specific record, while editing can be accomplished using 'Edit.'

      iv. Use the **Status** column to identify records that have been flagged as needing further attention – they will be highlighted in red, orange, or other colors.

6. All bouts of data entry must be accompanied by the Paper Datasheet Data Quality Checking procedure, outlined below. <u>This procedure must be conducted before the data entry bout is considered complete.</u>

B.4 Every Bout: Paper Datasheet Quality Checking

1. Field technicians should quality check a minimum of 10% of the entered data records or 10 records, whichever is higher. These targets are either on a per sampling/data entry bout or per SOP level (clarified below). Data quality control is accomplished by comparing entered values to the original paper datasheet values.

   a. **Per sampling or data entry bout**: Data quality checks should be carried out for each field sampling/data entry bout during a field season. This is done so that errors in data entry and problematic data entry habits can be detected and corrected.

   b. **Per SOP**: the number of records reviewed should be considered per protocol SOP rather than all protocol records totaled together.

   Example*:* After a sampling/data entry bout, a domain has 40 records from Litterfall SOP C (Field Sampling) and 390 records from Litterfall SOP D (Lab Processing – Dry Mass) for bout 3 of the season. Staff should therefore review 10 records from SOP C (because 10% of 40 is only 4 records) and 39 records from SOP D (because 10% of 390 is 39).

2. After a dataset is transcribed, it can be checked in one of two ways – either by exporting as a spreadsheet directly from the DEA, or using the NEON custom data viewer "Magpie". Both options will provide relatively fast ways to compare entered data to paper datasheet values.

   a. If choosing to export data as a spreadsheet from the DEA, refer to Appendix B.

3. If using the Magpie data viewer, use the instructions provided in the 'About this application' tab to select and filter the records relevant to the bout. Once selected and filtered, the records can be exported in spreadsheet form using the 'Download Query' button. From there, they can be further filtered as described above to facilitate data quality review.

4. Using the filtered data, compare the entered data values of each field to the original paper datasheet with one of the following procedures:

   a. **Read Aloud Review**: Data checking conducted by two technicians: one technician should read the values on the original paper sheet aloud, while the other visually checks against the electronically entered values.

   b. **Independent Data Review**: If it is not possible for two technicians to conduct data quality review together, then a technician who did not enter the data should perform the review.

   c. **Redundant Data Review**: If a separate technician cannot conduct data quality review, then the technician who entered the data should also visually re-check the data.

      i. It is strongly recommended that technicians employ the Read Aloud Review or Independent Data Review procedures over Redundant Data Review.

5. For any records that have errors, navigate to the relevant data entry application and correct them.

SOP C Using Checklists for Quality Control

Incorporating checklists into routine procedures has been shown to reduce the number of serious errors committed in both the aviation and healthcare industries (Clay-Williams and Colligan 2015). Checklists are best suited to situations where performance requires standardization (i.e. reviewing data quality), time is not critical, tasks may be forgotten or skipped, and the number of tasks may be too great to accurately memorize. Checklists are not just a memory aide but should also be considered a tool for discussing important aspects of a procedure with other staff members.

Research indicates that checklists are more likely to be effective and adopted into workflows when teams are actively engaged in the process of developing and adapting them to their particular contexts or work environments (Gillespie and Marshall 2015). As such, NEON protocol authors have not specified uniformly formatted, "one size fits all" data quality checklists for each protocol. Rather, protocol authors identify: (1) tasks that define the minimum standard for delivering high quality data; and (2) effective guidance that supports consistent decision making when minimum data quality standards are not met. In general, minimum data review standards:

- Clearly define the frequency of data review.
- Stipulate when data review must occur during a bout.
- Generate binary outcomes (e.g. pass/fail, yes/no), and
- Provide guidance when expected outcomes are not met.

Each field office should use the minimum data review standards and decision-making guidelines as a starting point for implementing checklists, while also acknowledging and incorporating data quality conditions unique to a domain (e.g. biological soil crust cover estimates are only carried out in D14 and D15).

C.1 Checklist Preparation

1. Consult the protocol's Data Entry and Verification SOP for key items that have been identified as critical to check.
    a. The protocol specific "Data QA/QC Checklist" is described within the SOP and where it is located on the Sampling Support Library (SSL) is typically linked.

    <u>NOTE</u>: The Measurement of Vegetation Structure protocol does not have a Data QA/QC Checklist, checks are conducted through the QAQC Shiny application instead

    b. The minimum data standards specified by the document should be effective for each bout unless otherwise specified.
    c. Any additional checks NOT specified in the "Data QA/QC Checklist" are entirely optional.
2. Two documents are provided for each protocol's data quality review process:

a. A Word document that lists specific details on how to conduct data review, referred to as the "**checklist**".

    i. The checklist provides step-by-step instructions on how to review data, and lists only the key procedures that need to be carried out.

    ii. Anything not listed on the checklist is not considered to be critical to the data product.

b. An Excel spreadsheet that contains abbreviated checklist items referred to as the "**worksheet**".

    i. The QC worksheet is intended for domain staff to track and communicate where data are in the review process, when review occurred, and who carried out review.

    ii. QC worksheet documents *can be modified* by Field Staff as needed, but ***must still contain the QC items listed in the Word document checklist***

3. Review the various checklist worksheet templates provided and adapt as appropriate for your domain's workflow.
    a. There is not a single right or wrong way to format a checklist.
    b. These templates are examples of the different ways a checklist can be implemented.
4. Create a checklist for your domain by combining the items listed in a protocol's Quick Reference section (or Knowledge Base article) and information from the QC Checklist with a worksheet template.
5. Review the content of your checklist worksheets and implementation strategy with fellow domain staff.
6. Store the digital version of the domain-specific QC Worksheet in a well-known, easy to find location.
    a. For example, save QC Worksheets to the All Collaboration Library and set the Category to 'QC Checklists' to enable easy discovery and retrieval.
    b. Alternatively, save checklists in a folder designated on the NEON shared drive, such as N:\DSF\Workspace\Domain XX\Subsystem\Protocol\etc. using the following naming convention: "moduleAbbreviation_DXX_QCChecklist_YYYYMMDDx"
7. Incorporate use of QC Checklists into training for all protocols.

SOP D Quality Control Shiny Applications

Several internal web applications (Shiny apps) have been developed to aid Field Science staff with quality control processes. These QC applications help to automate specific data quality checks, have enhanced data viewing features built in to identify potential data quality issues, and can save time.

The Aviary houses these applications, direct link here: NEON Aviary. In addition to QC specific applications, the Aviary also hosts various other data viewing applications such as Magpie and the Canary tool, which are discussed in **SOP E** and **SOP H** respectively.

> **REMINDER**: The NEON Aviary will not load if you are working outside of the Battelle network and are not connected to the VPN.

Refer to an individual protocol for more information regarding how and when to use a particular QC application. These applications are subject to change as some may be decommissioned, new ones added over time, or changes or enhancements are made to existing applications.

SOP E How to Use the Magpie Data Viewer Application

The Magpie shiny application is an online interface to view NEON data pulled from the NEON Fulcrum data recording applications. Magpie is used for the following tasks:

1. Viewing load delays for a given application suite and data product (loadGroup).
2. Submitting requests to HQ Science staff for data deletions or data updates when Fulcrum records have passed the load delay and are LOADED and locked.
3. Conducting data QA/QC (see **SOP F**).

Magpie can be accessed via the Data Management page of the Sampling Support Library (SSL) or via the NEON Aviary located here: NEON Magpie Application (or copy-paste: den-prodshiny-1.ci.neoninternal.org/Fulcrum-QA/).

The application itself has detailed descriptions of terms and instructions on how to use the interface in the 'About this application' tab, and training on using Magpie is included in the training plan for this protocol.

E.1 Workflow for How and When to Use Magpie

The Magpie application has four tabs, of which two are most pertinent to field staff, the 'About this application' and 'QA Data' tabs.

About this application tab:

- Orient yourself to this page before using the application for the first time. There is pertinent information about general data checks as well as instructions about how to use the application itself.
- The 'Data Delivery Deadlines' section outlines timelines between when data are entered into Fulcrum and when the parser attempts to load.
    - The load delay for all Fulcrum application DEA's are listed under the **daysBeforeLocked** column and can be queried per application. Reference this section of the app to confirm how many days after a record is *created* you have to QC data before there is a load attempt by the parser.
    - Remember that the parser ingests data per **loadGroup**, meaning it is possible data from more than one application will be pulled by the parser for a load attempt. For example, the Herbaceous Biomass loadGroup includes both the *HBP: Field Sampling* and the *HBP: Lab Masses* applications.
    - The **daysBeforeDeleted** column lists how long records in a given load group are kept in Fulcrum following creation before they are deleted. Applications that produce samples that are shipped to external facilities for analysis or archive have a long period before

deletion to ensure field staff have shipped the samples and the receiving institution has received them without any problems.

NOTE: Fulcrum record deletion is a manual process performed by HQ staff after a series of checks have been performed to ensure data have been successfully ingested into the NEON database, and any samples have been shipped and properly received by external facilities. Fulcrum records are deleted to ensure the applications run smoothly and do not get slowed down by old records. Deleted Fulcrum records are no longer visible in the Fulcrum DEA; however, these records can be found in the NEON L0 database or the NEON Data Portal.

QA Data tab:

This tab is used for two purposes:

1. To QC data for duplicates in a dataset as part of the standard quality control workflow outlined in this document in **SOP F.1**.
2. To Report records for either deletion or modification that have known issues, either discovered during the QC process or from ingest failures:
    a. Enter your **NEON email address**.
    b. Choose a **subsystem**.

    NOTE: The Shipping: Shipment Review app is only available under the TOS subsystem.

    c. Constrain your dates – i.e., **Start and End dates.** These dates should bound the dates the records were created in Fulcrum and do not *necessarily* correspond with the field collection date(s).
    d. Select your **domain**, **site**, and **application**
        i. If the application has child records, a box will pop up prompting selection of a subtask – select either Parent-Only or Child records.
    e. **Select a Query Type** – should be 'Raw Data' for putting in a request for a known issue.
    f. After these parameters have been set, a table will populate with any pertinent records.
    g. Find the record(s) of interest and select by clicking anywhere along the line. Multiple records may be selected at the same time, and a selected record will become highlighted in blue.
        i. Clicking the '**Edit Record**' (loadStatus=NONE) or the '**View Loaded Record**' (loadStatus=LOADED) button will open the record in another Fulcrum browser window.

    TIP: The search bar at the top right is useful for querying records via fulcrumID, loadStatus, or another attribute.

    h. For **Report Records?** select either Deletion Request or Update Request.

- i. **Deletion Request** – Request to completely delete a load_status=NONE record from the Fulcrum application. Note that Field Science staff can delete child records from a parent record with load_status = NONE without further assistance.
    1. Select a **Reason for Deletion.**
    2. The **Deletion Explanation** field can be used to further clarify the reason for deletion.
    3. Click the **Report Fulcrum Records for Deletion** button.
    4. A message pop-up box will say either “Deletion Request Completed” for success, or if you accidentally requested deletion of a LOADED record, it will fail and say “Deletion Request >REJECTED<”
- ii. **Update Request** – Request to modify a load_status=LOADED record in the NEON database and possibly in the Fulcrum application if necessary.
    1. Select the field that is errant in **Which column is in error?**
    2. Type in the **Corrected Value** – best practice is to copy/paste to avoid possible transcription errors.
    3. The **Reason for Update** field lists several choices for record modification, choose the best option.
        - a. DELETE RECORD is an option here, this means you want to delete a LOADED record already ingested to the NEON database, not just in Fulcrum.
    4. Add any **Additional Remarks** in free-form text that the HQ staff member may need to know in order to better understand and process your request. Please include any relevant ServiceNow Incident or Request numbers if applicable.
    5. The **Fulcrum Record Update Required?** field auto-populates to ‘No’ (Most Common), but can be modified if the record does need updating for other reasons.
        - a. In most cases, updating the Fulcrum record is NOT necessary, as edited data can be viewed in the NEON L0 database.
    6. Click the **Report Fulcrum Records for Deletion** button.
    7. A message pop-up box will say either “Update Request Completed” for success, or if you accidentally requested deletion of a NONE loaded record, it will fail and say “Update Request >REJECTED<”.

E.2 How to View and Edit Submitted Magpie Requests Within the Magpie Fulcrum Applications

After a Magpie request has been submitted it will create a Fulcrum record in either the **Magpie Deletion Requests** application (for Deletion Requests of UNLOADED records) or in the **Magpie Update Requests** application (for Update Requests to LOADED records). Field Science staff can use the Fulcrum applications to view their requests. It is best practice to review Magpie requests after submitting to ensure accuracy.

1. To view either of your request types (deletion or update), navigate to the appropriate Fulcrum application.
    a. The **Status** field defaults to ‘In Review’ for newly submitted records.
        i. If a record has been processed – i.e., Resolved=Y, the status will automatically set to ‘Processed by HQ’.
        ii. If a record was submitted for deletion by mistake, or if there is another type of problem with the request, Field Science staff can manually set the status to ‘Ignore Request’.
            1. Additional notes can be written to HQ staff members in the ‘Field Notes’ section to communicate why a record should be ignored.
        iii. When changing the status of a record to ‘Ignore Request’ it will automatically set the Resolved status to NA. The following text is displayed in the application itself:

    NOTE: *Changing this record's status to "Ignore Request" will update the "resolved" value to "NA". HQ will not process your request. If you need to reverse an "Ignore Request" edit, change the record status to "Edited by FS". This will change "resolved" to "N", and HQ will then proceed with processing the record.*

    b. The **Resolved** field defaults to ‘N’ for newly submitted records.
        i. A status of ‘NA’ will be ignored and not processed.
        ii. A status of ‘Y’ means the request has been processed.
    c. The **HQ Remarks** field is a free form text field for HQ staff to include links to data edits or write notes etc.

NOTE: Most Magpie Update Requests will not correlate to an update in the Fulcrum application record unless explicitly requested. This does not mean your request has not been fulfilled! If the Resolved status=Y, assume the request has been processed.

SOP F Referential Integrity Data Quality Review

Context: Post-Collection QA/QC

F.1 Analyze Data for Duplicate Records

Sample identifier fields are commonly named "sampleID", but also include a number of variations that all end with the suffix "ID" as described in the relevant protocols – e.g., individualID, subsampleID, fieldSampleID, massSampleID, moistureSampleID, archiveID etc. Sample identifiers are composed of human-readable unique ***metadata*** from each record, such as:

- plotID
- date
- sub-location, and
- sometimes a suffix or prefix related to the module (e.g. "***hbp***.CPER057163").

To analyze data for duplicate records:

1. Navigate to the NEON Magpie Application in a web browser (the link is in The Aviary, or copy-paste: den-prodshiny-1.ci.neoninternal.org/Fulcrum-QA/). Review more detailed instructions for using the Magpie application in **SOP E**.
2. Enter the appropriate search parameters for the data set you are checking – e.g., **subsystem**, **date range**, **domainID**, **siteID**, **application**, and **subtask** (**SOP** (if applicable)).
    a. For **Query Type**, select "Duplicates by Sample Identifier".
    b. **Choose Identifier Field to check** will automatically populate with the correct identifier if there is only one option, otherwise choose the appropriate identifier for sample type.
    c. For **Report Records?**, select "Deletion Request".
    d. If there are multiple sample identifiers (e.g. sampleID and subsampleID), an additional field will appear for you to indicate which value you'd like to check against.
3. **<u>If there are any duplicates</u>**, the application will return a set of records on the screen.
    a. Sort the records by the sample identifier field so that duplicate records are ordered together.
    b. You must now determine which record is the erroneous record that should be **fixed** (i.e. ***edited***) or ***discarded*** (i.e. ***reported for deletion***).
    c. ***Records can be discarded or reported for deletion***:
        i. If every single column value is identical across the duplicate records set (excluding fulcrum_id or created_at date), you can simply report all but one record for deletion.

ii. If the duplicate record in question is a ***child***, you can simply discard it yourself without using Magpie to make a request by:

    1. Navigating to the record in Fulcrum.
    2. In a web browser, click on the “x” symbol to the left of the child record’s title.
    3. On a tablet, long pressing on a child record will bring up a “context menu”.
    4. Select the trash bin symbol.
    5. Select “yes” when prompted to delete the record.

__**NOTE**__: This can be reversed if you discard the changes to the record before saving

iii. If the duplicate record in question is a ***parent***:

    1. Select the duplicate record(s) in Magpie.
    2. Select an appropriate **Reason for Deletion** from the drop-down menu options.
    3. The **Deletion Explanation** field is a free-form text box to add more details to the request for HQ staff.
    4. Tap the blue “**Report Fulcrum Records for Deletion**” button.
    5. Headquarters staff will process the deletion request.

iv. ***Records can be fixed*** if all column values are *not* identical.

    1. If this is the case, you must determine whether the recorder accidentally selected or entered a metadata value that created the duplicate sample identifier (i.e. plotID, collectDate, coordinates, etc.).
        a. For example, an incorrect (but valid) plotID was selected for a record on the same date as another duplicate record with the sample plotID value.
        b. Or, an incorrect but technically valid date was selected for a record from the same plotID.
    2. Edit the erroneous records so that the sample identifier is not a duplicate value of another record’s sample identifier.

__**NOTE**__: Records upstream or downstream from the one that was fixed will not automatically be fixed and should be edited to match the newly fixed record (see next section, **SOP F.2**).

F.2 Prevent Creation of Orphan Records

For protocols with linked applications, where metadata are referenced across applications (e.g. a lab record references a field sampling event), changes or corrections to metadata that make up the unique identifier fields in upstream applications can result in "orphaned" records. Orphaned records are a data quality issue because data in "downstream" applications cannot be accurately traced to an "upstream" record. Changes to sampling locations (e.g. plotID, clipCellNumber, soil coordinates) and/or sampling dates (e.g. setDate, collectDate) are the most common source of orphan records, as these values are almost always used to construct a unique sample identifier and data edits are never automatically propagated across applications.

Upstream applications are usually the first place where important metadata are recorded by field staff (plotID, subplotID, sampling date, soil coordinate, clip cell number, sample type etc.). Many protocols generate a unique identifier, typically named "sampleID", that is used to link data across multiple SOPs or data tables. Identifiers that are not **exact matches** across applications result in "orphan" and "childless" records.

Downstream applications "inherit" metadata from upstream applications, most importantly the sample identifier. Data that were already entered in an upstream application are generally un-editable. This is to prevent users from entering the same metadata values multiple times in different locations, which results in difficult to resolve transcription errors.

Follow these steps to prevent an orphan record in a downstream application when making edits to a record in an upstream application:

1. Have the application user manual on hand (Fulcrum Manuals are currently linked via each protocol page on the SSL).
2. **For each edit to a sampling location or sampling date** in an upstream record (**Table 3**), write down the original sample identifier.
    a. Using the original sample identifier, locate the downstream record.
    b. Open the downstream record and update the metadata following instructions from the application's user manual.

Table 3. List of protocols that have linked applications. Applications to the left of a table cell are "upstream"; applications to the right are "downstream". Records in downstream applications can have orphan records if corresponding sample information in upstream records is altered. *(p-c) = application has a parent-child structure, child records inherit metadata from the parent and each child must be updated if sample identifier information in the parent is changed.

Upstream ←---→ Downstream

Protocol	**App 1**	**App 2**	**App 3**	**App 4**
Algae – Periphyton and Phytoplankton Sampling	(AOS) Algae - Field (p-c)	(AOS) Algae - Lab (p-c)		
Aquatic Plant, Bryophyte, Lichen, and Macroalgae Sampling	(AOS) Plant Biomass - Field	(AOS) Plants - Lab		
Plant Belowground Biomass Sampling	BBC: Field Sampling (p-c)	BBC: Lab Weighing (p-c)	BBC: Grind and Pool (p-c)	
Plant Belowground Biomass Sampling	BBC: Field Sampling (p-c)	BBC: Lab Dilution (p-c)		
Canopy Foliage Sampling	CFC: Field Sampling (p-c)	CFC: LMA		
Canopy Foliage Sampling	CFC: Field Sampling (p-c)	CFC: Chemistry Subsampling		
Coarse Downed Wood	CDW: Field Bulk Density (p-c)	CDW: Lab Bulk Density (p-c)		
Dust Particulate Mass	(TIS) Dust Particulate SET	(TIS) Dust Particulate Mass COLLECT		
Ground Beetle Sampling	BET: Field Sampling (p-c)	BET: Lab Processing (p-c)		
Measurement of Herbaceous Biomass	HBP: Field Sampling	HBP: Lab Masses (p-c)		
Litterfall and Fine Woody Debris	LTR: Trap Deployment	LTR: Field Sampling (p-c)	LTR: Lab Mass Data (p-c)	LTR: BGC Sub-Sampling
Plant Phenology	PHE: Field Setup (p-c)	PHE: Phenophase Observations (p-c)		
Plant Phenology	PHE: Field Setup (p-c)	PHE: Annual Measurements (p-c)		

Protocol	**App 1**	**App 2**	**App 3**	**App 4**
Small Mammal Sampling	MAM: Trap Setting	MAM: Trap Collection (p-c)		
Soil Biogeochemical and Microbial sampling	SLS: Field Sampling (p-c)	SLS: pH		
Soil Biogeochemical and Microbial sampling	SLS: Field Sampling (p-c)	SLS: Moisture		
Soil Biogeochemical and Microbial sampling	SLS: Field Sampling (p-c)	SLS: Metagenomic Pooling (p-c)		
Soil Biogeochemical and Microbial sampling	SLS: Field Sampling (p-c)	SLS: BGC Sub-Sampling		
Soil Biogeochemical and Microbial sampling	SLS: Field Sampling (p-c)	SLS: Nitrogen Transformations (p-c)		
Measurement of Vegetation Structure	VST: Plot Meta-data	VST: Mapping and Tagging	VST: Apparent Individuals (p-c)	
Measurement of Vegetation Structure	VST: Plot Meta-data	VST: Shrub Groups (p-c)		
Wet Deposition	(TIS_AIS) Wet Deposition SET	(TIS_AIS) Wet Deposition COLLECT		

SOP G Detecting Process Quality Issues

Context: *Collection QA/QC* (for protocols with bouts that are longer than 2 weeks) or *Post-Collection QA/QC* (***before*** the next bout begins).

NOTE: There is a companion spreadsheet that is housed on the SSL Data Management page that is referenced in this SOP and where the dynamic tables **Table 4 – Proc Qual Field** and **Table 5 – Proc Qual Lab** are located, these tables are not directly located in any protocol. The direct link to access the tables is here: Data Management Spreadsheet.

- Tables 4 and 5 are maintained by Science staff and provide the expected number of records for field and lab activities per bout
    - These activities are divided by sub-system and protocol and are to be used as a reference during quality control activities as described below.

G.1 Analyze Process Quality: Field Sampling Completeness

The goal of checking data for “Field Sampling Completeness” is to ensure that all data have been collected from the field *and* are present in the cloud database. Refer to **Box 2** for a more detailed walk-through.

1. From the Fulcrum application dashboard on a web browser:
    a. Click “View Data” for the application you are checking; you should be taken to a table-like view of the application’s data.
    b. If a table does not appear, click “Table Mode” in the top-right corner of the screen (next to the “+” add record button).
2. If “Clear All Filters” appears in blue text at the top of the table, click the “X” to remove filtering.
3. Most protocols have an “event” identifier (e.g. weekBoutBegan, boutType, eventID, yearBoutBegan, etc.)
    a. If it is not visible, click on the “Column Setup” (three vertical lines) button and search for an event field; check it to make the column appear.
    b. For protocols with no event identifier, filter by collectDate, yearBoutBegan, etc.
4. Filter the visible rows to the sampling bout/event you are reviewing.
    a. Click the down arrow of the event column, a window appears.
    b. Click “Select Specific Values”, search for and select an event identifier.
    c. Be sure to filter data by a single siteID, as this simplifies record counts and will match the information in **Table 4 (companion spreadsheet, tab 1).**

TIP: You can also filter via the main search bar by typing in any field value because it searches through all fields i.e. eventID (except for Record ID).

5. Only records with the specified event identifier will now appear in the web browser table.
6. **Evaluate the record count:**
    a. For a given bout, if **the application does not have child or grandchild records**:
        i. The total number of records will appear at the top-left above the table.
        ii. **Compare this number to what is expected in Table 4, columns E-H**, based on information in the relevant protocol and SOPs listed.
            1. **Column E** contains rules describing how many digital records should appear per sampling location.
            2. Columns F and G only need to be referenced if an application has nested child and/or grandchild records (see below).
            3. **Column H is the total expected number of parent records that should appear in the web browser.**
        iii. If your counts are **BELOW** the expectation, you may be missing records from the bout, missing physical samples from the field, or have incorrectly entered metadata that goes into the event identifier (e.g. yearBoutBegan, siteID, soil coordinate etc.)

        NOTE: You may be able to discover if records have incorrect metadata by sorting records based on the “created date”. Records collected in sequence and/or in the same time interval should be adjacent when sorted this way.

        iv. If your counts are **ABOVE** the expectation, there may be duplicate records in the dataset and/or too many samples were collected in the field.
    b. For a given bout, if **the application has child or grandchild records**:
        i. The total number of **parent records** will appear at the top-left above the table. **This number should equal the number in Column H** (**Table 4**)
        ii. For each parent record, open the record and view the number of child and grandchild records present.
            1. **View the number of child records** (**Table 4, Column F**).
            2. **If present, tally the number of grandchild records per child record** (**Table 4, Column G**).
        iii. If your counts are **BELOW** the expectation, you may be missing records from the bout and/or missing physical samples from the field.

iv. If your counts are **ABOVE** the expectation, there may be duplicate records in the dataset and/or too many samples were collected in the field.

G.2 Analyze Process Quality: Protocol Sampling Completeness

The goal of checking for "Protocol Sampling Completeness" is to ensure that all post-field sampling data have been collected, entered, and align with field sampling data. These checks only need to occur if protocols require additional data collection generated from lab procedures. Refer to **Box 2** for a more detailed example.

1. From the Fulcrum application dashboard on a web browser:
    a. Click "View Data" for the application you are checking; you should be taken to a table-like view of the application's data.
        i. If a table does not appear, click "Table Mode" in the top-right corner of the screen (next to the "+" add record button).

NOTE: *This tab will be referred to as the "**upstream**" application (e.g. Field Sampling).*

2. Open another tab in the web browser and locate the second application you will be checking.

NOTE: *This tab will be referred to as the "**downstream**" application (e.g. Lab Mass).*

3. Select "Clear All Filters" for both open tabs.
4. Filter the visible rows to the sampling bout/event you are reviewing in ***both tabs:***
    a. Click the down arrow of the event column, a window appears.
    b. Click "Select Specific Values", search for and select an event identifier.
    c. Be sure to filter data by a single siteID, as this simplifies record counts and will match the information in **Table 4.**

TIP: You can also filter via the main search bar by typing in any field value because it searches through all fields i.e. eventID (except for Record ID).

5. For a given bout:
    a. Note the number of upstream records there are after filtering.
    b. Note the number of downstream records there are after filtering.
6. Evaluate the record counts:
    a. **Table 5** (**companion spreadsheet, tab 2**) describes rules for how to count downstream records.
        i. These are rules rather than specific numeric values because not all sampling events result in a downstream record, but are instead based on various

conditions defined in a protocol (e.g. Lab Mass records are not created if no field sample was collected).

  ii. Applications and protocols not listed in **Table 5** do not need to have Process Quality checks performed.

b. In most cases, **the number of parent records in the downstream application should equal the number of parent records in the upstream application.**

  i. If your counts are **BELOW** the expectation, you may be missing records in the downstream app that should be present.

    1. Potential reasons for this include:

      a. Data have been collected, but a device was not fully synced.

      b. Data have been collected, but were recorded on paper and not entered.

      c. Critical upstream metadata may have been changed, causing a mismatch (e.g. siteID or an event identifier were changed in some way).

      d. Data have not been collected, recorded, or entered.

  ii. If your counts are ABOVE the expectation, there may be duplicate records present in the upstream or downstream app, or event identifiers may be incorrect (e.g. weekBoutBegan was manually adjusted improperly).

  iii. Workflows that deviate from these general guidelines are listed below. See **Table 5** for more details.

    1. Canopy Foliar Sampling (CFC).

    2. Soil Biogeochemistry and Microbe Sampling (SLS).

Box 2. Detailed example of conducting all Process Quality checks with Litterfall data in Fulcrum (Field and Lab data).

1. Open the "**LTR: Field Sampling**" data table in a web browser.
2. Open the "**LTR: Lab Mass**" data table in another web browser window.

TIP: The 'Save View' button can be used to save a particular filter set for quick viewing next time. Give your custom view a meaningful title and click 'Confirm'.

- Your saved view will be displayed in the application bar at the top of the screen underneath the application name titled 'Views'.

3. To assess ***Field Sampling Completeness*** (**SOP G.1**)
    a. **Organize data**:
        i. In "*Column Setup*" the first four columns are re-arranged as siteID, weekBoutBegan, setDate, and trapID.
        ii. The rows are sorted by trapID so that duplicates can be spotted.
    b. **Filter data:**
        i. In the **siteID** column: select one site, "**UNDE**"
        ii. The event identifier is "**weekBoutBegan"**, select a single value in the **weekBoutBegan** column.
    c. **Assess results:**
        i. UNDE is a forested site that uses 1600 m^2 plots, **Table 4** of the Data Management Protocol suggests that there should be two parent records per plotID and 40 records total (two traps per plot in two random subplots)
        ii. The web browser shows 40 records
            1. Are there any duplicate records inflating the count?
                a. Click on the trapID column options, then "*Select Specific Values*"
                b. This shows a count of the number of rows with the same trapID value
                c. All counts equal "1", so there are no duplicates present
                d. The correct number of records has been collected and synced
4. To assess ***Protocol Sampling Completeness*** (**SOP G.2**)
    a. **Table 4** of the Data Management Protocol suggests that LTR: Lab Mass records should appear if a Field Sampling record has a **trapCondition** of "OK" or "PF"
        i. **Organize data:**
            1. In the "Column Setup" of the "**LTR: Lab Mass**" application, the first three columns are re-arranged as siteID, weekBoutBegan, and setDate
        ii. **Filter data:**
            1. In the "**LTR: Field Sampling**" window, limit the visible records by the trapCondition column to those with "OK" or "PF"
            2. In the "**LTR: Lab Mass**" window, filter data to the same site and weekBoutBegan as is visible in "**LTR: Field Sampling**"
        iii. **Compare results across applications:**
            1. There are 40 records visible in the "**LTR: Field Sampling**" window
            2. There are 41 records visible in the "**LTR: Lab Mass**" window
            3. All expected records appear to be present from the Field and Lab portions, however there is one more record in the Lab app…
        iv. **Evaluate discrepancy:**
            1. There are more records in the "**LTR: Lab Mass**" app, start there
            2. Clicking on the **trapID** column options and "*Select Specific Values*" shows there is a single **trapID** with two entries, checking the box on this trapID filters rows down to these two records
            3. Further inspection of the records shows that one parent record has qaDrymass values and is a duplicate of the other record
            4. One record should be reported for deletion

SOP H Parser Error Resolution

It is inevitable that some data records will fail to pass the parser given the sheer volume of human-collected data NEON ingests. Fortunately, there are multiple systems in place to catch those errors before they are published on the NEON Data Portal. As discussed in Section 3.3 of this document, the Fulcrum applications have built in checks and validation requirements, and once data are set to load, the parser provides another round of validation requirements before data are ingested.

This SOP describes the Field Science staff role with regard to parser error resolution. These tasks comprise the final opportunity for Field Science to ensure data are complete and accurate.

H.1 Data Ingest Process and Statuses

Remember that the OS Parser attempts to load records from Fulcrum applications within a certain timeframe (loadDelay) on a nightly basis. The system communicates whether records were loaded successfully by updating the record status in Fulcrum.

1. LOAD_STATUS states:
    a. **PARSE_FAIL** = set by Parser, failed ingest
    b. **LOADED** = set by Parser, passed ingest and loaded into database
    c. **SKIP** = set by Canary, will not be pulled for ingest
    d. **NONE** = set by apps, generally means has not hit the load delay yet

NOTE: HQ staff members can set a record to have any load status and may need to do so to resolve certain parser errors.

2. Record Status Cycles:
    - **Complete** = set by user
        - Also set by Canary one week AFTER a SKIP cycle
        - This means Canary only sets to Complete IF the record was originally set to SKIP
    - **RECORD_ERROR**/**recordError** = set by Canary only.

NOTE: The Record Status might not equal “Complete” after a record fix because Field Science staff have control over the field and “reset” the Record Status from RECORD_ERROR to complete if desired. This means a successfully loaded record may have a record status of RECORD_ERROR.

H.2 Efficient Workflow for Addressing Parser Errors

The parser errors listed in the Canary reports (see below section **H.3**) can sometimes be overwhelming, and although all parser errors must eventually be resolved, it can be unclear where to start. Records can sometimes be flagged erroneously due to problems with linked records in *upstream* applications. For example, a sampleID error in a Field record will cause a "Sample does not exist" error in the related Lab record.

Identifying and fixing those 'high-priority' errors first should be the standard workflow. Remember that Fulcrum records are loaded per load group, so one error within one application for a load group may be holding up data from a completely different application.

The following order of operations is suggested when tackling parser errors:

1. Resolve all errors ***within a load group*** at the same time.
2. Within a load group, identify and prioritize errors ***for the data governed by one ingest workbook at a time*** (if more than one error per record or sample identifier).

NOTE: See **Table 5** below for definitions of ingest workbook validations.

3. In linked applications, ***resolve sampleID errors in upstream applications first (see Table 3).***
4. Resolve errors associated with ***samples sent to an external lab*** for analysis or an external facility for taxonomic identification, these are typically sampleID or barcode issues.

IMPORTANT: Errors should not persist for weeks and months on end. Create an Incident in ServiceNow and seek help if necessary.

H.3 Canary Tool - List of Parser Errors per NEON and per Domain

The Canary is a tool that holds and updates current parser errors daily at 10 AM (MST) in spreadsheet form. The Canary sends out a weekly email to domain staff on Monday at 10 AM (MST) as well.

To access the Canary spreadsheet go here: Canary. The Canary can also be found within The Aviary.

The following information is captured in Canary and should serve as the first place that Field Science staff view the current data errors:

- **messageDate** – today's date (or if viewing the weekly email, then Monday's date)
- **domain** – domainID
- **loadGroup** – name of data product per OS Parser
- **appOrIngest** – name of Fulcrum application
- **errorType** – error message reported by the OS Parser
- **errorField** – the field in the Fulcrum application with issue
- **errorValue** – the value in error, i.e., an identifier
- **errorMessage** – further details describing what failed and what identifiers (if applicable) are involved
- **fix** – more detailed instruction regarding what the errorMessage means and how to correct it
- **fulcrumRecordID** – direct link to the Fulcrum record with issue
- **dateErrorFirstReported** – this is the date the error was first reported

IMPORTANT: The Canary report displays the same error for the same single sample identifier in more than one way – i.e., multiple lines can exist in the spreadsheet for one single issue, but each line will contain different errorTypes.

For example (**Table 4**): A MAM: Trap Collection [PROD] record is causing an error because a single barcode is associated with more than one sampleID. The Canary report lists this single error in three different ways! Note that the errorValue is the same for all three.

Table 4. Example error messages from Canary report. The same error is reported three different ways.

errorType	errorValue	errorMessage
Changing Tag	D00000082539	Changing Tag from HARV.20220622.R6812.B to HARV.20220623.R6812.B not allowed for Sample[[tag=HARV.20220622.R6812.B,barcode=D00000082539,sampleClass=mam_pertrapnight_in.bloodSampleID]].
ReferenceCount	D00000082539	ReferenceCount is greater than one: [tag=HARV.20220623.R6812.B,barcode=D00000082539,sampleClass=mam_pertrapnight_in.bloodSampleID]. Fulcrum ID=8ed907c4-2d3a-4788-ab6a-a94c1c95bdb7.
has different tags	D00000082539	Uploaded sample with barcode 'D00000082539' has different tags. (HARV.20220623.R6812.B, HARV.20220622.R6812.B).

Do not let this confuse you. Consolidating these messages so they are all together may be useful (such as how they appear above). Choose whichever error message makes the most sense for you to digest and go from there.

H.4 General Workflow for Resolving Parser Errors

There are many resources for resolving parser errors that exist within the Observatory, and as such, there may be more than one avenue to resolve a parser error given the different levels of experience and familiarity with these types of issues amongst Field Science staff. This protocol should serve as a resource along with knowledge base articles, domain specific training, staff meetings, word-of-mouth, weekly meetings, asking for assistance via ServiceNow from HQ Science staff, etc.

The following table lists the type of error messages that may be found within Canary weekly reports and disseminates what they mean in more literal terms for field staff to resolve. Note some messages may be something only HQ Science staff can resolve but you may still be notified of the issue via the Canary.

Table 5. Description of parser error message language, what it means in more simple terms, and how to resolve.

Generic Error Message	Example message	Issue
ACTIVITY_END_DATE_VALIDATION	Validation failed for field 'collectDate' in function ACTIVITY_END_DATE_VALIDATION. The end date (2021-07-05T10:20-05:00[US/Central]) for the record is before the start date (2021-07-05T22:00-05:00[US/Central]).	HQ Science staff fix
Cannot add parent Sample	Cannot add parent Sample[[tag=HARV_010.E.20210707.IB.01,sampleClassSet=[bet_sorting_in.subsampleID.mam, bet_sorting_in.subsampleID.ib, bet_sorting_in.subsampleID.herp, bet_sorting_in.subsampleID.bet]]] to existing sample [uuid=8c48fe26-254d-4ce0-b035-84601e7b89c5,tag=HARV_010.E.20210707,barcode=A00000152098,sampleClass=bet_fielddata_in.sampleID].	Duplicate sampleID where one has already been ingested and its parent sampleID is not the same as the one in the current load
Changing Barcode	Changing Barcode from B00000153375 to A00000358651 not allowed for Sample[[uuid=4d486a7e-fc12-4d59-9b5b-c74276778a16, tag=LIRO.W2.20220606.RAW,barcode=B00000153375,sampleClass=swc_fieldData_in.rawSampleIDgwc]].	Duplicate barcode where both duplicates are in the current data upload: same barcode, different sample tags
Changing SampleClass	Changing SampleClass from apl_domainLab_in.herbariumSampleID to apc_voucher_in.herbariumSubsampleID not allowed for Sample[[uuid=f409145d-2ca2-415b-bc72-e86fa48f23aa,tag=BARC.20220629.AP2.P3.H,barcode=A00000173431,sampleClass=apl_domainLab_in.herbariumSampleID]].	Indicates the sampleClass ingested by the parser and what is acceptable in the ingest workbook differs from what is in the record. These are most common in shipment records and can arise when a sampleClass changes and was not propagated downstream
Changing Tag	Changing Tag from TOOK.INFLOW.20220516.FIL to TOOK.IN.20220516.FIL not allowed for Sample[[uuid=3d0e81a7-c00d-4ea3-bd34-98859024c506, tag=TOOK.INFLOW.20220516.FIL,barcode=A00000302356,sampleClass=swc_fieldData_in.filtSampleID]].	Changing tag from SAMPLE.TAG.X to SAMPLE.TAG.Y and sampleClass from sampleClassA to sampleClassB not allowed for sample with uuid ### and barcode ###

Generic Error Message	Example message	Issue
CONVERT_TO_UTC	Validation failed for field 'collectDate' in function CONVERT_TO_UTC. The incoming value '-0-0T' expected to be a date, but instead is a 'StringC' type.	HQ Science staff fix
DERIVE_FROM_SAMPLE_TREE	Validation failed for field 'collectDate' in function ACTIVITY_END_DATE_VALIDATION. The incoming value '' expected to be a date, but instead is a 'VoidC' type.	the voidC failure for dates can mean blank, or it can also mean a date in an unreadable format such as in Excel with "PM" in the time column HQ Science staff fix
GREATER_THAN	Validation failed for field 'height' in function GREATER_THAN. Comparison failed (0 > 0).	The value in this field does not fulfill the validation that the number for the field in question needs to be greater than a specified value
has different tags	Uploaded sample with barcode 'D00000012729' has different tags. (FSH.CARI.01.20180904.2.0001.DNA, FSH.TOOK.09.20220629.5.0001).	Duplicate sampleID where both duplicates are in the current data upload: same sample tag, different barcodes
LESS_THAN	Validation failed for field 'vdBaseBreakHeight' in function LESS_THAN. Comparison failed (2.4 < 2.4).	The value in this field does not fulfill the validation that the number for the field in question needs to be less than a specified value
LOV	Validation failed for field 'samplingProtocolVersion' in function LOV. LOV value 'NEON.DOC.014049vM' could not be validated due to missing activity end date.	Invalid LOV value. These most often need HQ Science Staff to fix
NAMED_LOCATION_TYPE	Validation failed for field 'plotID' in function NAMED_LOCATION_TYPE. Named location KONZ with type 'OS Plot - bet' is not allowed	HQ Science staff fix
ReferenceCount	ReferenceCount is greater than one: [tag=VST.CLBJ.03311.2021, sampleClass=yearPlant].	Duplicates: Multiple records with this sampleID are being uploaded in this load attempt
REGULAR_EXPRESSION	Validation failed for field 'plotID' in function MATCH_REGULAR_EXPRESSION. value 'PUUM_015.basePlot.bet' failed with pattern '^[A-Z]{4}'.	The parser is checking this value against a regular expression HQ Science staff fix
REQUIRE.	Validation failed for field 'dryMass' in function REQUIRE. Input Value is required	A required Fulcrum field is blank in a record. Required fields represent critical information for a data product. A data product is often less usable, or not useful, without all of the required data being present. The value in this field needs to be filled in.

Generic Error Message	Example message	Issue
REQUIRE_NULL	Validation failed for field 'sampleFate' in function REQUIRE_NULL. Input Value is not empty	The value in this field needs to be NULL (empty)
Sample already exists	Sample already exists in the database: [tag=PRIN.SS.20220621, sampleClass=swc.asi.sdg.amc].	Indicates that the database has already loaded a particular identifier and a duplicate record exists in Fulcrum with the same sample identifier
Sample does not exist	Sample does not exist: [tag=NEON.LTR.BART046852.20220607, sampleClass=ltr_fielddata_in.fieldSampleID]. Fulcrum ID=602fab18-7ae8-4116-828e-15f8965a7ea3.	Orphan sample: the sample ID is supposed to already be in the database from an earlier record, but it isn't.
_THAN		The parser is expecting the value in this field to be greater than/less than/equal to another field (the other field is not listed)
DERIVE_FROM_SAMPLE_TREE	Validation failed for field 'collectDate' in function ACTIVITY_END_DATE_VALIDATION. The incoming value '' expected to be a date, but instead is a 'VoidC' type.	the voidC failure for dates can mean blank, or it can also mean a date in an unreadable format such as in Excel with "PM" in the time column HQ Science staff fix

General workflow for troubleshooting and resolving parser errors:

1. Consult the Canary Parser Error Report or the weekly email notification from the Canary to view applicable parser errors from your domain and for the data product(s) you are responsible for.

   REMINDER: The Canary Parser Error Report will automatically refresh daily at 10 AM (MST) so this is the most current resource to use versus the Monday morning email.

2. Review the entire row of information in the report and make sure you understand what the message is telling you, see section Error! Reference source not found. for a brief outline of each column in the spreadsheet and remember one single error may appear multiple times in the spreadsheet, worded slightly differently.

   a. First you want to note the **errorType** – this will tell you whether it's an issue where a sample does not exist, a duplicate sample exits, there's an issue with a sample identifier, etc.

   b. Next you will need to note the **errorValue** in the Canary spreadsheet (column G), this most often will be a sampleID or sample barcode that is causing the error but could also be a field within the data that is throwing the error.

   c. The next most important section of the Canary report to note and understand is the **errorMessage** itself, column H. This will give you more context about what the issue is and if applicable to the error, the other sample identifier that you will need to check into.

3. Once you've established what the error message is telling you and which field or identifier you need to check, go to the Fulcrum application in which the record was created.
4. Enter the identifier (barcode, sampleID) directly into the search bar or search for the FulcrumID given in the Canary report.
   a. If there is a duplicate issue – i.e., the same barcode is associated with more than one sampleID – it will result in multiple Fulcrum records from your query.

   NOTE: To query the FulcrumID in Fulcrum, you need to click on the dropdown arrow under the field titled 'Record ID' and paste the FulcrumID into the search bar. If you do not see Record ID as an option, it is because you do not have it selected to view. To select it, click the three vertical lines next to the filter button and be sure Record ID has a blue checkmark next to it.

5. Review the identifier combination of i.e. sampleID (if applicable) + barcode (if applicable) + sampleClass, etc. or the value in the field.
   a. Correct the field value (if applicable).
   b. Determine what identifier combination is correct and fix the Fulcrum record.
   c. If you determine an incorrect identifier is already loaded into the database and the failing Fulcrum record is actually correct, then you will need to submit a Magpie update request, as you cannot edit a loaded record.
6. Once you believe you have corrected the error, return the next day to confirm the record has loaded.
   a. The next morning from the day you made the edit the OS Parser should try to pull the data again, so checking the Canary spreadsheet after 10 AM (MST) will tell you whether or not your edit was successful (if it no longer appears on the report).
   b. You can also check the Fulcrum record directly – the load_status will update to LOADED.
7. If you are unable to resolve a parser error on your own and you have exhausted all other resources, than you can create a ServiceNow incident outlining the error(s), what steps you've taken, where you are stuck or what you don't understand, and a HQ Science staff member will assist you.

Below are some case studies of various error messages and how to solve them. These are some of the most common parser errors seen:

1. In the MAM: Trap Collection [PROD] application there is a 'Changing Tag' (errorType) associated with barcode D00000082539 (errorValue), where it is associated with two different sampleID's, errorMessage of: Changing Tag from HARV.20220622.R6812.B to HARV.20220623.R6812.B not allowed for Sample[[tag=HARV.20220622.R6812.B,barcode=D00000082539,sampleClass=mam_pertrapnight_in.bloodSampleID]].
    a. This is saying that in Fulcrum this barcode D00000082539 is associated with both sampleID: HARV.20220622.R6812.B AND HARV.20220623.R6812.B
    b. Figure out the correct barcode associations for the sampleID's and correct the applicable Fulcrum record.
2. In the SLS: Nitrogen Transformations [PROD] application there is a 'Sample already exists' (errorType) associated with barcode B00000111234 (errorValue), where it is associated with two different sampleID's, errorMessage of: Sample already exists in the database: [tag=JORN_004-M-21-20220705-KCL, barcode=B00000111234, sampleClass=ntr_internalLab_in.kclSampleID]. Fulcrum ID=78e9179a-c003-4025-95cb-7db06135339c.
    a. This is saying that this barcode B00000111234 is associated with another sampleID in the NEON database (has already loaded). It is associated with: JORN_004-M-41-20220705-BM
    b. Figure out the correct barcode associations for the sampleID's and correct the applicable Fulcrum record and/or submit a Magpie update request if applicable.
3. In the LTR: Field Sampling [PROD] application there is a 'Sample already exists' (errorType) associated with sampleID NEON.LTR.SJER064707.20220811 (errorValue), where the sampleID has already loaded in the NEON database, errorMessage of: Sample already exists in the database: [tag=NEON.LTR.SJER064707.20220811, sampleClass=ltr_fielddata_in.fieldSampleID]. Fulcrum ID=117ee765-d6a4-4d0f-95fb-7f4331c3ddc6.
    a. This is saying the sampleID has already loaded either via a different Fulcrum record and this one is a duplicate, or something else may be happening.
    b. After review of this FulcrumID's full history (see Appendix B and C), it loaded previously and then was re-synched on a tablet that had not been opened and synched for a while, thus rolling the record's load_status back to NONE from LOADED.
        i. The OS Parser is trying to re-load this but since the record already loaded it is giving this error.
    c. This requires assistance from HQ Science staff because Field Science cannot adjust a Fulcrum records status.
    d. Submit a ServiceNow incident to resolve.

4. In the LTR: Lab Mass Data [PROD] application there is a 'Sample already exists' (errorType) associated with sampleID NEON.LTR.BART046852.20220607 (errorValue), where the sampleID has already loaded in the NEON database, errorMessage of: Sample does not exist: [tag=NEON.LTR.BART046852.20220607, sampleClass=ltr_fielddata_in.fieldSampleID]. Fulcrum ID=602fab18-7ae8-4116-828e-15f8965a7ea3.
    a. This is saying the sampleID does not exist in the *associated parent* LTR: Field Sampling application, because it likely has changed since it was originally created.
    b. Upon inspection, the collectionDate changed after the record was initially created and therefore changed the sampleID from NEON.LTR.BART046852.*20220607* to NEON.LTR.BART046852.*20220608.* However, this update was not propagated to the LTR: Lab Mass Data application.
    c. The Lab Mass Data record, along with the child records, need to be re-associated with the new, updated parent Field Sampling application by re-selecting it.
5. In the (AOS) Plants - Lab [PROD] application there is a 'LESS_THAN' (errorType) within the field: wetMass (errorField) stating that: (5407.33 <= 5000) (errorValue) associated with one specific Fulcrum record. The errorMessage states: Validation failed for field wetMass in function LESS_THAN_OR_EQUAL_TO. Comparison failed (5407.33 <= 5000).
    a. This is saying that you cannot have a value greater than 5000 for the wetMass and since the value input was 5407.33, it failed the parser validation.
    b. This is likely a typo, correct the wetMass value
6. In the (AOS) Reaeration [PROD application there is an 'expected to be a date' (errorType) within the field: loggerInWaterDateTime (errorField) with an errorValue of: 2022-07-05Tundefined associated with one specific Fulcrum record. The errorMessage states: Validation failed for field 'loggerInWaterDateTime' in function CONVERT_TO_UTC. The incoming value '2022-07-05Tundefined' expected to be a date, but instead is a 'StringC' type. Fulcrum ID=9fa12b47-357c-43d9-839f-48a2f66dac64.
    a. This is saying that the date string (format) is not recognized as valid for the field, hobo_in_water_ingest.
    b. This is something HQ Science staff will need to correct in the Fulcrum record by exporting the record as a spreadsheet, making the correction, and importing back into Fulcrum as a spreadsheet, a SN incident can be created to move the process forward.
7. In the (AOS) Dissolved Gas [PROD] application there is a 'Sample does not exist' error (errorType) with an errorValue of: MART.SS.20220906. The errorMessage states: Sample does not exist: [tag=MART.SS.20220906, sampleClassSet=[swc.sdg, swc.asi.sdg, sdg, asi.sdg, asi.sdg.amc, swc.asi.sdg.amc, sdg.amc, swc.sdg.amc]]. Fulcrum ID=9a568b22-fa77-40c5-8611-b125ab77efaa.

   a. This is saying that the sampleID: MART.SS.20220906 does not exist in the *upstream* (AOS) Water Chemistry [PROD] app, which creates the parent sampleID for this *downstream* application
   b. A likely scenario to cause this is a change in the sampling date.
   c. Upon review of the records in the (AOS) Water Chemistry app affiliated with the date 20220906 or something near that date, it appears the correct parent sampleID is MART.SS.20220907
   d. To correct, the record in the (AOS) Dissolved Gas app needs updated to reflect the correct date of 20220907.
      i. Edit the 'Date Processed' field which will propagate to the Reference Air Sample ID automatically.
      ii. To update the Water sampleID, click the child record and tap 'Select Parent Water Sample' to re-populate the field with the correct record with the correct date of 20220907.

H.5 Workflow for Resolving Shipping Parser Errors

Shipping samples to external labs is the last phase of a sample's chain of custody, and it is extremely important that data captured in the shipment receipt and manifest are accurate. Due to the sheer volume of samples shipped from NEON domains to external lab facilities, as well as the shipping application dependency upon all other applications to pull in accurate sample identifiers such as sampleID, barcode, sampleClass etc., it is not uncommon for shipping data issues to arise.

Shipping parser errors are most often related to sample identifiers (barcode/sampleID) not being found in the NEON database. This is usually caused by changes/edits made in the data collection upstream applications between when a shipment was created and sent, to when the OS Parser attempts to load the shipping data. The loadDelay for the shipping application is 45 days from record creation. Exacerbating the challenges to resolve shipping parser errors is the fact that the physical specimens are no longer in-house and thus cannot be physically checked.

However, Field Science staff members can and should be the first NEON staff to attempt to resolve shipping parser errors. If Field Science staff are unable to resolve the error on their own, then it is appropriate to reach out via ServiceNow for HQ Science staff assistance.

Follow the workflow below to troubleshoot and resolve shipping parser errors:

1. Consult the Canary Parser Error Report or the weekly email notification from the Canary to view applicable current shipping parser errors.

REMINDER: The Canary will automatically set the shipping records with issue to have a load Status of SKIP for one week (Mon to Mon) to allow for other records to load in the meantime. Keep this in mind when you are viewing the Shipping application data in Fulcrum.

2. Query the failed identifier in the Fulcrum Shipping: Shipment Review [PROD] Application i.e. the **errorValue** in the Canary spreadsheet (column G), most often will be a sampleID or sample barcode.
    a. It is easiest to type this value directly into the search bar.
    b. This will bring up the entire shipment record that has the failed identifier as a child record.
    c. Open the child record view of the record and find the identifier, hint: use CTRL+F for a pop-up search window which will highlight the identifier for a large shipment.
    d. View the identifier combination – i.e., sampleID (if applicable) + barcode (if applicable) + sampleClass, etc.
3. Open another Fulcrum browser and open the applicable Fulcrum application where the record for the sample was originally created.
    a. Query the sample identifier (the same identifier from the Canary report).
    b. Compare the identifier combination between this record and the shipping record – *they will not match!*
4. Now you will need to troubleshoot and do some digging to resolve the mismatch.
    a. Was the sampleID or barcode or sampleClass for the failing sample changed in the original data application record?
        i. Was there a Magpie update request made and fulfilled? Check Magpie Fulcrum application.
        ii. View the Fulcrum record history to see any changes made over time with the record (see **Appendix B** and **Appendix C** for how to view full record history).
    b. Was the sampleID or barcode or sampleClass for the failing sample changed in the shipping application after shipment?
        i. Was there a Magpie update request made and fulfilled? Check Magpie Fulcrum application.
        ii. View the Fulcrum record history to see any changes made over time with the record.
        iii. Checking the shipment email sent out by the Stork application can also be handy – query your email to view the original manifest and compare that to both Fulcrum records if necessary.

   iv. Is there a duplicate shipping record?
      1. There is potential for two shipping records to exist where one has the correct identifier combination and the other does not.
      2. One record could be loaded successfully and the other not.

   NOTE: If a duplicate shipping record is discovered, always make sure to request the errant record for Deletion either via a Magpie Deletion request (NOT LOADED) or a Magpie Update request for deletion (LOADED).

5. Once you have figured out where/why the change occurred between the sample and the shipment, there are two options for resolution:
   a. Directly edit the shipping record to correct it:
      i. First, delete the errant sample's child record.
      ii. Save and close the record.
      iii. Reopen the record and add the sample back in, making sure the identifiers are all correct.
      iv. Save and close the record.
   b. If you are unable to determine which sample identifier is correct and have exhausted all your in-house resources, create a ServiceNow Incident outlining the shipping error and the applicable sample identifiers in question for a HQ staff member to assist you.

Below are some case studies of various error messages and how to solve them. These are some of the most common parser errors seen for Shipping:

1. In the Shipping: Shipment Review [PROD] application there is a 'Sample does not exist' (errorType) associated with barcode B00000129130 (errorValue), with an errorMessage of: Sample does not exist: [tag=BARC.20220711.EPIPHYTON.1.CHL/PHEO.1, barcode=B00000129130, sampleClass=alg_domainLab_in.sampleIDchem]. Fulcrum ID=f96216fe-c8eb-4645-a158-687f1add85aa.
   a. This is saying the sampleID and barcode combination has not loaded into the NEON database yet.
   b. This type of error typically occurs when a sample is shipped before it has loaded or if the associated record within the application where the sampleID was created has a parser error as well.
      i. If the shipped sample(s) have not yet hit the load_delay (viewable on Magpie) than it is usually best to wait until they do. Otherwise, HQ Science staff can assess and if deemed necessary manually load the records.

   ii. If the record is failing ingest within the application where it was created than it must be corrected and loaded before the shipping record will load.

   iii. In either of these scenarios, the actual shipping record does not need a modification, it is the associated records in the *upstream* applications causing the issue.

2. In the Shipping: Shipment Review [PROD] application there are two 'has different tags' errors (errorType) associated with barcodes A00000131300 and A00000131303 (errorValue), with errorMessages of: Uploaded sample with barcode 'A00000131300' has different tags. (OSBS_063.20210603.IB.01, OSBS_063.20210520.IB.01) AND Uploaded sample with barcode 'A00000131303' has different tags. (OSBS_063.20210520.IB.01, OSBS_063.20210603.IB.01)
   a. These error messages are saying that the two barcodes selected within the shipping application record do not correspond with the sampleID and barcode combination that is loaded to the NEON database.
      i. This is likely due to a change made in the barcode and/or sampleID within the *upstream* application, in this case the BET: Lab Processing [PROD] app after the shipment record itself was created.
      ii. The change needs propagated into the shipping record as well i.e. these two barcode/sampleID combinations need removed from the shipping record and then added back in to match what is reflected in the BET: Lab Processing [PROD] application.

   <u>NOTE</u>: If you are unsure which barcode definitively should go with which sampleID it may be necessary to ask HQ Science staff for assistance. For instance, these samples were shipped to archive at ASU and a physical sample check may be necessary to confirm the correct combination.

3. In the Shipping: Shipment Review [PROD] application there is a 'Changing Barcode' (errorType) associated with A00000308847 (errorValue), with an errorMessage of: Changing Barcode from A00000308847 to A00000308830 not allowed for Sample [[tag=CARI.SS.20220802.RAW, barcode=A00000308847, sampleClass=swc_fieldData_in.rawSampleID]].
   a. This is saying that the barcode associated with CARI.SS.20220802.RAW in the shipping record is A00000308830 while the barcode loaded to the NEON database is A00000308847.
      i. The barcode needs updating in the shipping record to match what is in the NEON database.

   <u>NOTE</u>: This error appears in the Canary report in two different ways. It also appears as: 'has different barcodes' (errorType) associated with CARI.SS.20220802.RAW (errorValue), with an errorMessage of: Uploaded sample with tag 'CARI.SS.20220802.RAW' has different barcodes. (A00000308830, A00000308847).

4. In the Shipping: Shipment Review [PROD] application there is a 'NAMED_LOCATION_TYPE' (errorType) within the field destinationFacility (errorField) associated with 'Laragen Inc.' (errorValue), with an errorMessage of: Validation failed for field 'destinationFacility' in function NAMED_LOCATION_TYPE. 'Laragen Inc.' is not a recognized NEON named location. List of type codes: (('External Lab'|'DOMAIN')). Fulcrum ID=6c2d7334-530d-442b-9985-e96daedf015d.
    a. This is saying the external laboratory is not a value recognized in the named location list in the NEON database.
        i. This is a task for HQ Science staff to correct.

8 REFERENCES

Chapman, A.D. 2005. Principles of Data Quality – Report for the Global Biodiversity Information Facility 2004. Copenhagen: GBIF.

Fisher, C., Eitel, L., Chengalur-Smith, S., and R. Wang. Introduction to information quality. Cambridge, MA: MIT Information Quality Program.

Michener, W.K. and J.W. Brunt. 2000. Data Management Principles, Implementation and Administration. Ecological Data – Design, Management, and Processing (ed. W.K. Michener and J.W. Brunt), Blackwell Science Ltd, Oxford.

N.J. Van Lanen, C.M. White, J.A. Fogg, and M. F. McLaren. 2012. Integrated Monitoring In Bird Conservation Regions (IMBCR): Data Entry Protocol. Unpublished report. Rocky Mountain Bird Observatory, Brighton, CO, USA. URL: rmbo.org/v3/Portals/0/Documents/Science/Protocols/

Sebastian-Coleman, L. Measuring Data Quality for Ongoing Improvement: A Data Quality Assessment Framework. 2013. Elsevier Inc: Waltham, MA.

APPENDIX A DATA ENTRY TRAINING CHECKLIST

So, you are ready to begin entering ____________________ data.

(protocol)

Make sure you complete the following checklist before beginning. A permanent staff member or designated lead technician will need to sign off on your checklist before actual data entry can begin.

- ☐ Read through the Data Management protocol.
- ☐ Review Data Management Training Materials (Presentations, quizzes, worksheets).
- ☐ Read general Data Entry Application (Fulcrum) User Training Guide.
- ☐ Read relevant protocol-specific Fulcrum User Manual(s).
- ☐ Enter at least one datasheet, plot or subplot of data into the training data entry application (CERT) for the relevant protocol.
- ☐ Talk through the process of:
    - ☐ Annotating datasheets
    - ☐ NEON data entry webpage login, ability to navigate to protocol-specific apps
    - ☐ Appropriate remarks, use of commas and special characters
    - ☐ Manual data transcription DQC process
    - ☐ Where entered and checked datasheets get stored
    - ☐ How to report DQC bouts
    - ☐ Importance of catching and reporting duplicates
    - ☐ Use of the Magpie tool
    - ☐ Evaluating and resolving parser errors (may only be applicable to year-round staff)

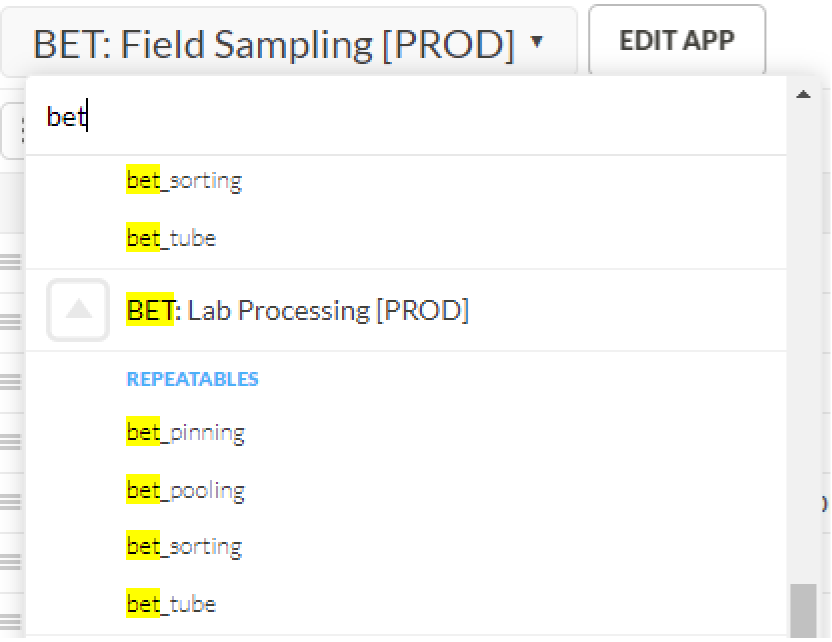
____________________________	_________	____________________________	_________
Trainee Signature	Date	Trainer Signature	Date

APPENDIX B HOW TO FILTER AND VIEW FULCRUM DATA IN THE WEBUI

This appendix describes how to set up filters, select specific columns, and how to efficiently view Fulcrum data via the webUI. A quick data review is much more efficient if you are only viewing those data of interest. Proper filtering will also set up your data for potential export into Excel for further investigation if needed, see **Appendix C**.

1. From the application dashboard (url: https://web.fulcrumapp.com/) query and click on the desired application. This brings you to the App Dashboard which will default to viewing the 'Record Activity'.
    a. Click the 'View Records' button, upper right-hand corner, far left icon.
    b. Make sure you are in 'Table View' (upper right-hand corner icon), not 'Map View'.
    c. To toggle between applications, tap on the application name bar at the top left-hand side of the screen and begin typing in the first few characters of the application name.

TIP: You can view a table of child records in the web browser as well. When using the search bar, child records, if applicable, will be displayed under 'REPEATABLES' for a given parent application.



2. Once an app is selected, the view defaults to showing all the records for a given application from your domain.
    a. Quick Filters are available on the left-hand side bar, which include dates and record status which may be useful filters.

   b. Click the ‘Toggle Quick Filters’ button to remove the Quick Filters bar (three horizontal lines with a dot to the left of the main search bar).

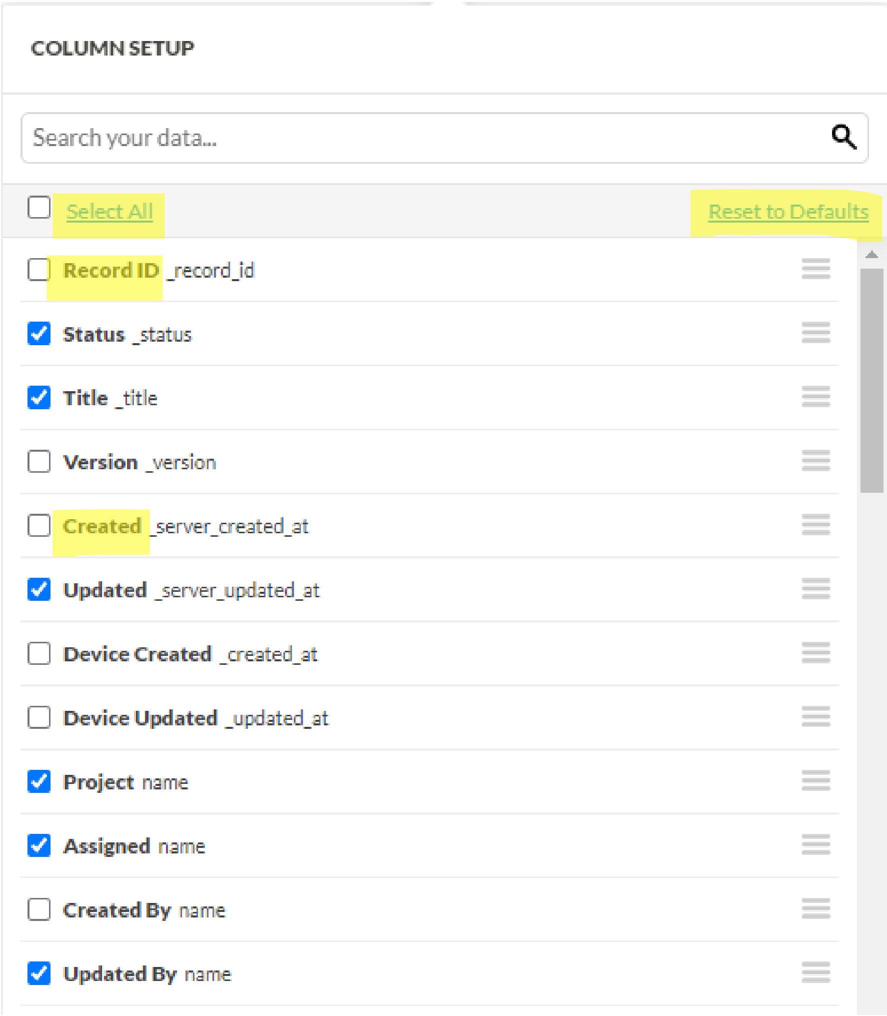
3. When first viewing records, Fulcrum has a default view comprising a subset of columns for a particular application. It is often necessary to add in more columns for effective data viewing and QC checking. Additionally, when exporting data (see **Appendix C**), only those columns selected will be in the downloaded file.

   a. Hit the ‘Column setup’ button (three vertical lines) to manually add in or remove columns that will display by checking the box to the left of the column name.

   b. There are options to ‘Select All’ or to ‘Reset to Defaults’ which can come in handy.

   c. **Record ID** is often very useful to have displayed, this is the FulcrumID.

   d. **Created** is when the record was synched and when it was created on a tablet, respectively, and those dates are important to know for QC.



COLUMN SETUP

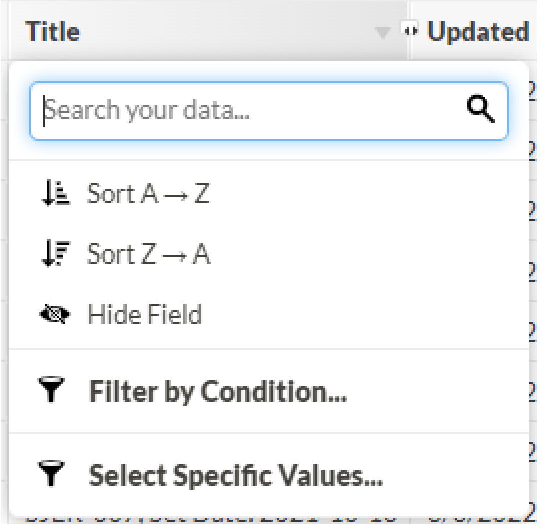
Search your data...

☐ Select All   Reset to Defaults

☐ **Record ID** _record_id
☑ **Status** _status
☑ **Title** _title
☐ **Version** _version
☐ **Created** _server_created_at
☑ **Updated** _server_updated_at
☐ **Device Created** _created_at
☐ **Device Updated** _updated_at
☑ **Project** name
☑ **Assigned** name
☐ **Created By** name
☑ **Updated By** name

   e. **LOAD_STATUS** is a default column. Make sure to keep this column visible, as it will tell you the current status of the record – i.e., NONE, SKIP, LOADED, PARSE_FAIL.

4. There are several options for filtering data, and Fulcrum has published a how-to article here: [Fulcrum instructions to filter data]
   a. The main search bar is useful for a particular identifier i.e. sampleID, barcode, siteID, etc.
      i. The search functionality will also query child records that may not be displayed within the parent record but will locate an identifier within that child record.
   b. Clicking on any column header will allow further filtering or sorting functionality:
      i. **Sort** alphabetically – A-Z or Z-A.
      ii. **Filter by Condition** – use logical operator.
      iii. **Select Range** – toggle range of values or type in directly.
      iv. **Select Specific Values** – type directly in search bar and/or manually check boxes.



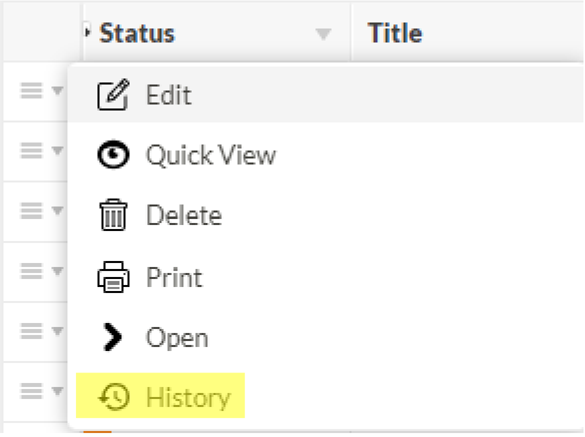
   c. The 'Custom Filters' button (to the left of the Column Setup button) allows for multiple levels of filters using logical operators.
   d. The 'Clear all filters' button is used to clear filters.

5. The 'Save View' button can be used to save a particular filter set for quick viewing next time. Give your custom view a meaningful title and click 'Confirm'.

   a. Your saved view will be displayed in the application bar at the top of the screen underneath the application name titled 'Views'.

6. An individual record can be edited, viewed, or popped out into another browser window by selecting the filter button (down arrow and three horizontal lines) ≡▾
    a. **Edit** – will fully open record to modify
    b. **Quick View** – will open record in a pop-up view
    c. **Print** – will open a printable pdf
    d. **Open** – will pop out the individual record for easy viewing in another browser window
    e. **History** – this will display all the changes made to a record over time and will pop out in another web browser

<u>NOTE</u>: The history of a record can be very useful for resolving parser errors as changes made to sampleID, taxon, or barcodes etc. after a record was initially created will often lead to downstream errors in lab or shipping applications.



APPENDIX C HOW TO EXPORT FULCRUM DATA IN THE WEBUI

This appendix describes how to export data from Fulcrum. You may need to export data for QC review or other reporting purposes. It is typically easier to export data from Fulcrum, then manipulate and filter those data in Excel compared to filtering and summarizing in Fulcrum via the webUI. In addition, it can be very helpful to download and view the full version history of a record. This can be especially true when the full version history of a record is needed to resolve an ingest failure.

Exporting Data From the Record Editor

1. From the application dashboard (***url***: https://web.fulcrumapp.com/), navigate and click on the "records" button of an application for which you wish to export data.

   This brings you to what is called the "Record Editor" – i.e., the location for entering records through a web browser.

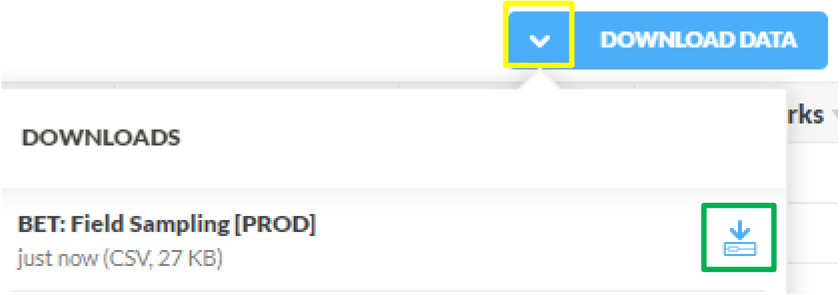
2. Make sure you have your desired filters selected to narrow down the subset of records you would like to export, see **Appendix B** for further details.

3. You may also customize which columns are exported using the "Column Setup" button (three vertical lines).

   NOTE: Only those columns selected within the "Column Setup" will be included in your data download!

4. After any custom filters are in place, click on the blue "DOWNLOAD DATA" button in the top right-hand corner of the Record Editor.

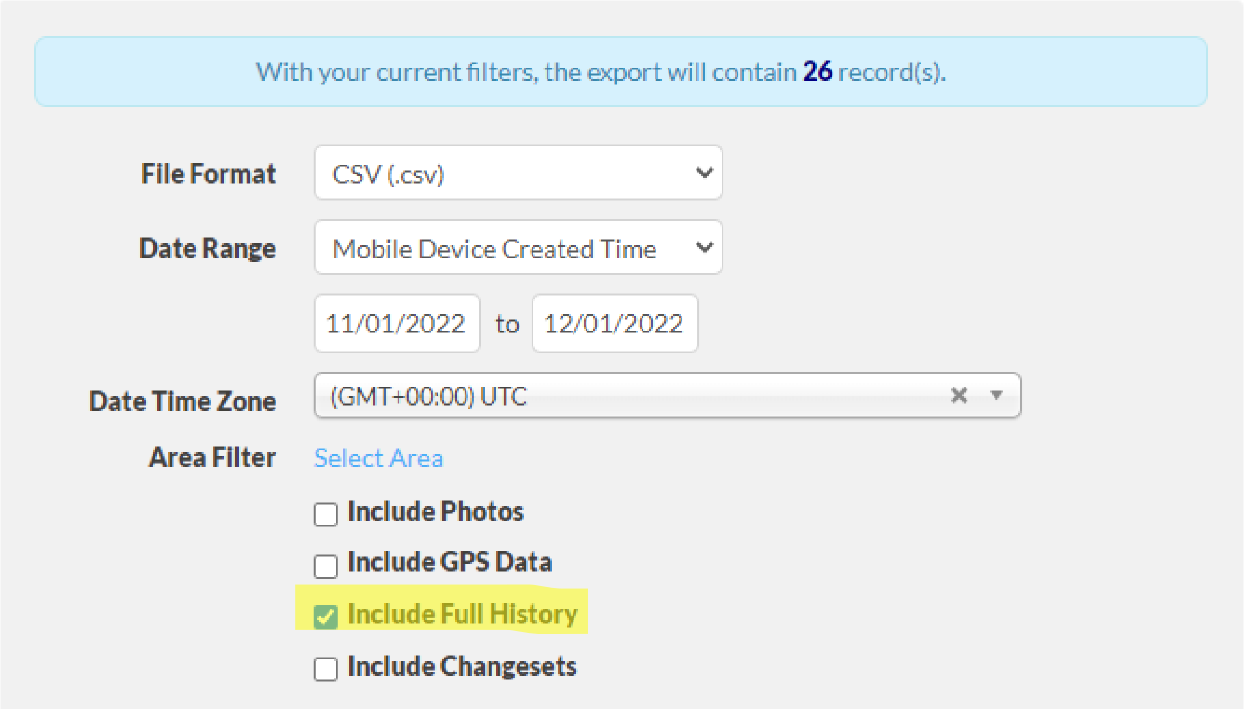
5. A new window will pop-up prompting you to select the desired file format.
    a. Both .CSV and .XLSX file formats can be opened in Microsoft Excel
        i. Select .CSV if you intend to pull your data into a software program such as R
        ii. Select .XLSX if you would like both parent and child records downloaded as one spreadsheet with multiple tabs
    b. 'Record Links' pulls in the FulcrumID via the record title or the record ID.
        i. Select *Title* for logistical referencing and *ID* for relational joining
    c. *If applicable*, select the type of 'Media Fields' to download
7. Click "Start" after selecting a file format, the window will then say "Processing".
    a. This process will take more time with increasing numbers of records and child records
8. Click the blue down arrow button next to the "DOWNLOAD DATA" button, then highlight and select the data you just downloaded.



9. The data will automatically save to the "Downloads" folder of your local computer.
    a. For applications with child and/or grandchild records, *the exported data will be divided into separate files* ("child" and "grandchild" are not keywords in the file names) UNLESS you have selected .XLSX in which there will be one file with multiple tabs:
        i. One **parent** file.
        ii. One or more **child** files (if applicable).
        iii. One or more **grandchild** files (if applicable).
    b. Rows in these separate files are related by several columns:
        i. **Parent** files will have a single unique identifier column named "_record_id"
        ii. **Child** and **grandchild** files will have three record identifier files "_child_record_id", "_record_id", and "_parent_id"
            1. In **Child** files, the "_record_id" and "_parent_id" columns will be equal to each other. Both of these fields correspond to the "_record_id" value in a parent file.
            2. In **Grandchild** files, the "_record_id" value equals the "_record_id" in the parent file. Confusingly, the "_parent_id" equals the "_child_record_id" in the child file (because the child records are the direct "parent" of the grandchild record).
        iii. Instructions for merging dozens of data columns across hundreds or thousands of child/grandchild records in Excel or other software are complex, error prone, and beyond the scope of this document. If you require merged records, data exported from the Magpie applications are delivered as a single, fully merged file.
            1. A tutorial on how to use and manipulate data in Excel is provided in the Field Science Training Center (link).

Exporting full version history for a record

1. From the application dashboard (***url***: https://web.fulcrumapp.com/), click on the '+' sign in the upper righthand corner of the screen, a menu will pop up with several options, select 'Export'.
2. **File Format** should be either CSV or Excel XLSX.
3. Select a **Date Range** – be sure to include the date the record was created.
4. For **Area Filter** be sure to **check the box 'Include Full History'.**
5. For **Apps**, select the application of interest, it is best practice to only select one at a time for download.
6. For **Projects**, select your domains Project ID.
7. Scroll back up to the top of the page where there will be a light blue box displaying the number of records that will be contained in your export.



With your current filters, the export will contain **26** record(s).

File Format CSV (.csv)

Date Range Mobile Device Created Time

11/01/2022 to 12/01/2022

Date Time Zone (GMT+00:00) UTC

Area Filter Select Area

☐ **Include Photos**

☐ **Include GPS Data**

☑ **Include Full History**

☐ **Include Changesets**

   a. Select NEXT.
   b. The Export Summary will be displayed, select CONFIRM.
   c. The Exports page will display downloads you've made over time, highlight the one of interest and select the Download icon to download your results locally.
   d. The downloaded file will have the same conventions as other Fulcrum downloads, with a parent file, and if applicable, "child" and "grandchild" files.

i. The parent file will give details such as version history, load status, error remarks, dates changes to the record were made, etc. Fields such as these may come in handy to pinpoint when a particular error arose and what the error message was.

ii. The child or grandchild file will generally give details such as the sample identifier – i.e., the sampleID or the barcode. It is then easy to see if sampleIDs or barcodes have changed over the lifetime of the record and could be leading to a parser error.

APPENDIX D SUPPLEMENTAL PARSER ERROR RESOLUTION WORKFLOWS

This appendix describes specific parser error resolution workflows, including a few protocol specific errors that may be encountered and ways to resolve.

D.1 Parser Error: Sample Does Not Exist

Problem

A record's sampleID should have a matching value in another "upstream" application but does not. Usually, this means a Lab record does not directly link to a Field record.

Causes

- Changes to metadata values used to construct the sampleID in upstream records, AFTER downstream records were created.
- Upstream records deleted but downstream records still exist.

Check For…

- (**Solution A**) SampleIDs that do not match ***exactly*** between upstream and downstream records.
- (**Solution B & C**) Missing records in the upstream app.
- (**Solution D**) All sampleIDs match between upstream and downstream records.

Solutions

Solution A

FULCRUM DOES NOT AUTOMATICALLY UPDATE DATA ACROSS LINKED RECORDS

If upstream record data are correct but downstream record data are not updated:

1. Open downstream record.
2. De-select the currently selected upstream record.
3. Re-select the upstream record. This action manually copies the updated upstream metadata into the downstream record.
4. Save the record (and re-sync if editing on a tablet).

Solution B

If the upstream record/sampleID no longer exists and should NOT exist:

1. Request downstream records for deletion.

Solution C

If the upstream record/sampleID no longer exists and SHOULD exist:

1. Create a new upstream record, if possible. This record's sampleID must match the downstream record's sampleID.
2. Recall that a new record will have a "fresh" load delay time, thus the parser error message will persist until this new record is ingested.
    a. Contact Science for assistance if the wait period is excessive – e.g., samples need to be shipped and records need to be manually loaded ahead of the load delay.

Solution D

The upstream record already exists, SHOULD exist, and SHOULD allow the downstream record to load (because the upstream record sampleID is correct). This is usually caused by one of the following scenarios:

1. **Load Delay Time**:
    a. Remember that every record has a load delay that starts counting down from the **"created_date".**
    b. If you just created a new record to match a downstream sampleID, you will see the parser error until the new record's load delay is reached.

    <u>**SOLUTION**</u>: wait until the load delay time is met.
2. **Other Record Parser Failures**:
    a. Recall that data are ingested in "sets" or "groups" of records per domain and per load group, from potentially multiple applications.
    b. A single error in a loading data set will cause all records to not ingest.
        i. ***Example***: 10 records have met the load delay time. There is one record with a parser error, therefore ALL 10 records will NOT load.

    <u>**SOLUTION**</u>: fix parser errors in other records.

D.2 Parser Error: _THAN (Fails Comparison)

<u>Problem</u>

A numeric value or date is compared to another value for validation or plausibility purposes. This error generally occurs when a reported value is outside an expected range or reported dates are out of an expected order (e.g. sample collection date is reported as occurring *before* the trap is set). Comparison may be made to an absolute value (e.g. basalStemDiameter < 30) or as a comparison to another measurement field (e.g. vdBaseHeight < vdApexHeight).

Causes

- Typographical error, value includes an extra digit or numbers have been transposed.
- Incorrect units used for measurement.
- Protocol confusion.
- SampleID errors in already loaded records.

Check For…

- (**Solution A**) Typographical error in values.
- (**Solution B**) Transposition of values across data fields.
- (**Solution C**) SampleID errors. ***Note****: Errors may be caused by other records.*
- (**Solution D**) Incomplete records.

Table 6. Table of common _THAN requirements by TOS protocol.

Data Product	Field	operator	comparisonValue
BET	identifiedDate	>	collectDate
VST	measurementHeight	<=	200
CDW, HBP, LTR, SLS	ovenEndDate	>	ovenStartDate
CDW, HBP, LTR	weighDate	>=	ovenEndDate
CDW, VST	ninety…Diameter	<=	max... Diameter
HBP, LTR, MOS, TCK	collectDate	>	setDate
HBP, LTR, SLS	ovenStartDate	>=	collectDate
VST	measurementHeight	<=	200
VST	vdBaseBreakHeight	<	vdApexBreakHeight
VST	vdApexHeight	>	vdBaseHeight
VST	basalStemDiameterMsrmntHeight	<	100
SLS	freshMassBoatMass	>	dryMassBoatMass
SLS	dryMassBoatMass	<	freshMassBoatMass

Solutions

Solution A

Typographical errors in values

1. If it is clear that the error is caused by a data entry error and the intended value is clear (e.g. measurementHeight = 1300 when standard measurementHeight = 130 or stemAzimuth = 725, a clear transposition of 275, based on pointID and subplotID):
   a. Open the record with the incorrect value.
   b. Edit the value.

    c. Save/Sync.

2. If it is not clear what caused the error:
    a. If the sample is still available, re-measure and edit the record accordingly, OR
    b. If the sample is not available and the field is not required, nullify after consulting with the protocol author, OR
    c. If the error cannot be resolved, and the field is required, delete the record (child), or request deletion through magpie (parent).

Solution B

Transposition of values across data fields

1. Look at both fields being compared (note that only the field causing the error will appear in the Parser error output). If it is clear that the intended values are switched:
    a. Open the record.
    b. Switch values to correct fields.
    c. Save/Sync.

Solution C

SampleID error in an already loaded record. Since the sampleID already exists in the database, the parser ignores internal, within record comparisons, and defaults to the value/date associated with the already loaded record.

1. Look up the fulcrum record that is causing the error.
2. Copy the sampleID.
3. Open the child sample view for the app (child record view).
4. Filter on sampleID.
5. If >1 record comes up and one or more have load_status = 'LOADED', review the loaded records, for parent / child sampleID mismatches.
6. Submit a Magpie L0 update request.

Solution D

Incomplete records: e.g., errorField = nullFieldValidation; errorValue = (0 > 0). This error occurs when a set of the expected data fields are all null.

1. Look up the fulcrum record that is causing the error.
2. If the record is a child record, use the child record view.
3. Filter data columns by Parent ID = fulcrumRecordID.

4. Left click data column, under 'Filter by Condition…', select 'Is blank'.
5. Repeat for all data columns.
6. Identify null records, containing no data.
7. Edit record if appropriate.
8. Delete if unresolvable, and the record is a child record.
9. Request deletion if the error is unresolvable and the record is a parent record.

<u>Protocol Specific Solutions</u>

***Vegetation Structure*: VST – nullFieldValidation (0 > 0)**

This field does not exist in Fulcrum, it is generated on ingest to check that each record actually has data in it. The requirement is 1 or more fields in the list below:

- stemDiameter
- basalStemDiameter
- vdApexHeight
- maxCrownDiameter
- ninetyCrownDiameter

Must be NON NULL if plant status is *not* one of the following values:

- Removed
- No longer qualifies
- Lost, burned
- Lost, herbivory
- Lost, presumed dead
- Lost, tag damaged
- Lost, fate unknown
- Downed

If plant status = live, standing dead, broken bole, or any of the live but damaged fields, there must be data in one of the data fields above.

D.3 Parser Error: ReferenceCount

<u>Problem</u>

There is a duplicate sample identifier within the dataset being uploaded. Sample identifier fields must be unique; if not, then the data associated with an identifier cannot be traced back to one individual sample, thereby breaking the sample chain. This error typically occurs across records (i.e. rows of data) rather than within a single record.

Note that the same rules apply for sample barcodes in that they must be unique, although this parser error currently only captures duplication of non-barcode sample identifier fields.

Causes

While some data entry applications can stop you from generating duplicate sampleIDs in child records, they can't do this for parent records.

Some common causes of this error are:

- An identical record for a sample was inadvertently created more than once.
- A sample was inadvertently created more than once at different times and/or by different people.
- In many Fulcrum applications, the sampleIDs are generated behind-the-scenes by concatenating different pieces of metadata that, when joined together, create a unique ID. A duplicated value in one of these metadata values can create a duplicate identifier.
- A sample identifier created in an upstream app was accidentally selected more than once in the downstream app in which the duplication error occurred.

Check For…

- (**Solution A**) Identical *records* in the same app, and all associated metadata values are correct
  Example: Data for a batch of samples is entered one sample at a time, and the data for one sample is accidentally entered more than once. In this instance, every value in the affected records is an exact match.
- (**Solution B**) Identical *sample identifiers* in more than one record in the same app, but the metadata associated with the records is different and values are correct.
  Example: A soil sample was inadvertently processed and measured for soil pH twice and the data were entered twice.
- (**Solution C**) Identical *sample identifiers* in the same app due to incorrect metadata values being selected.
  Example: For two different plots sampled on the same day, the same plotID was accidentally selected for two different records.
- (**Solution D**) Identical *sample identifiers* in the same app due to the incorrect sample identifier being selected from a drop-down list of available sample identifiers

Solutions

This error is generally straightforward to address because you know that the duplicate entry occurred within the specific dataset that the parser is attempting to load.

Solution A

If an identical record was accidentally created:

1. For parent records: Submit a deletion request for the record containing the duplicated identifier.

2. For child records: While these are typically caught by the app before you can save a parent record, it is still possible for a child record to be duplicated. To correct:
    a. Open the duplicated record.
    b. Delete the record from the app.
    c. Save the record (and re-sync if editing on a tablet).

Solution B

If a record containing the same sample identifier was accidentally created:

1. Only one of the records can be kept, so it is important to review all metadata values in the affected records and determine which record to keep.
2. For parent records: Submit a deletion request for the record to be deleted.
3. For child records: While these are typically caught by the app before you can save a parent record, it is still possible for a child record to be duplicated. To correct:
    a. Open the duplicated record.
    b. Delete the record from the app.
    c. Save the record (and re-sync if editing on a tablet).

Solution C

If a sample identifier was accidentally created in the app due to incorrect metadata values in the affected record:

1. If you can identify which record contains the incorrect sample identifier and you have the correct metadata values:
    a. Open the record.
    b. Modify the incorrect metadata value(s).
    c. Save the record (and re-sync if editing on a tablet).
2. If you can identify which record contains the incorrect sample identifier but cannot determine how to correct the metadata, then:
    a. Submit a deletion request (for parent records).
    b. Or delete the record (for child records only; be sure to save the changes!).
3. If you cannot identify which record contains the incorrect sample identifier, then all affected records may need to be deleted.
    a. Contact Science for further guidance.

Solution D

If a sample identifier created in an upstream app was accidentally selected more than once in the downstream app where the duplication occurred:

1. If you can identify which record contains the incorrect sample identifier and know how to fix it, then:
    a. Open the record.
    b. Modify the record by selecting the correct identifier, if the correct identifier exists.
    c. Save the record (and re-sync if editing on a tablet).
2. If the correct identifier does not appear as an option in the downstream app, it may be that the sample identifier was not created in the upstream app.
    a. Open the upstream app.
    b. If the correct sample identifier exists in the upstream record but still isn't appearing in the downstream app for selection, this could be because Fulcrum does not automatically update data across linked records. To fix:
        i. First check that the upstream data are synched.
        ii. Then, open the downstream record, de-select the current sampleID, then re-select the correct sampleID.
        iii. In some cases, you may also need to delete the information in the plot field first, and then re-select the plot before the correct sampleIDs will appear.
        iv. Save the record (and re-sync if editing on a tablet).
    c. If the correct sample identifier does not exist in the upstream app because of a data entry error in an upstream record:
        i. Open the upstream record that contains the error.
        ii. Fix the data entry error(s).
        iii. Save the record (and re-sync if editing on a tablet).
    d. If the correct sample identifier does not exist in the upstream app because the upstream record was never created, then create a record for the missing sample and save the record (and re-sync if editing on a tablet).
3. If you can identify which record contains the incorrect sample identifier but do not know how to fix it, then:
    a. Submit a deletion request (for parent records), OR
    b. Delete the record (for child records only; be sure to save the changes!).

4. If you cannot identify which record contains the incorrect sample identifier, then all affected records may need to be deleted.
    a. Contact Science for further guidance.

Protocol Specific Solutions

Here's a quick snapshot of common types of soil protocol-specific errors and possible solutions

Application	Common cause of error	Possible solutions
sls_soilpH, sls_soilMoisture	Duplicated measurement; accidentally selected upstream record from drop-down list more than once	B, C, D
sls_soilCoreCollection	Accidentally entered same sample more than once	A, B, C

D.4 Parser Error: Changing Tag

Problem

The Parser has detected duplicate barcode values in a data set. Barcodes are treated like sample identifiers on ingest and must therefore be unique across all records. One way duplicate barcodes arise is when the same barcode is associated with two or more different sampleIDs.

Example: ONE barcode (A00000007572) is associated with TWO different records where the sampleIDs = SITE_001-M-**5.5-38.5**-20180509 and SITE_001-M-**38.5-5.5**-20180509 (note the reversed values in bold). The Parser does not know to which sampleID the barcode belongs in reality.

Causes

Duplicate barcodes may be:

- In the same record,
- In different records within the ***same app***, or
- In different records within ***different apps.***
- Duplicate barcodes may variously be located within only ingested (LOAD_STATUS = "LOADED") and/or non-ingested (LOAD_STATUS = "NONE") records.

Solutions

Follow the suggestions outlined below to resolve these types of errors:

1. **Pay attention to the errorMessage text:** The errorMessage presents three useful pieces of information in the following format:
    a. "***barcode = x, tag = y, sampleClass = z***": The barcode noted here is used multiple times, use this to search for Fulcrum records.
    b. "***Changing tag from sampleID1 to sampleID2***": This tells you whether the barcode is associated with multiple different sampleIDs.

        i. ***Example***: "SITE_006-M-0.5-26.5-20180510 to SITE_006-M-26.5-0.5-20180510" are NOT the same sampleID, so one barcode is pointing to two different samples, potentially.

    c. "***sampleClass from ingestTable.A to ingestTable.B***": This information denotes whether the barcode is being used in different TYPES of samples – e.g., a soil sample and a beetle sample might have the same barcode value.

2. **Determine what the errors are with a desktop or laptop computer**
    a. Copy the barcode from step 1.a
    b. Analyze the text in step 1.b "Changing tag from..."
        i. ***If sampleID1 does NOT EQUAL sampleID2***, the same barcode has been used with two different sampleIDs
            1. ***Example***: "SITE_048-**O**-36.5-25.5-20180621 to SITE_048-**M**-36.5-25.5-20180621" are NOT matching sampleIDs.
    c. Analyze the text in step 1.c "sampleClass from..."
        i. **If "*ingestTable.A does NOT EQUAL ingestTable.B*",** the same barcode has been used with two different sampleIDs of different sampleTypes.
3. **Find all records with the same barcode**
    a. Navigate to the application noted in the "ingestTable" column.
    b. Determine if the application has a linked "upstream" application.
        i. *Example*: ingestTable = sls_soilMoisture_in; this is the Soil Moisture app and is linked to the upstream Soil Core Collection app.
    c. Paste the barcode into the search bar of the downstream app.
        i. Fulcrum will find the associated record even if the barcode is in a child record.
    d. Open another browser tab if there is an upstream app, and paste the barcode into the search bar of the upstream app.
4. **Determine the error**
    a. *Within the same application:*
        i. If a search for a single barcode returns more than one record, then the barcode is duplicated across multiple records. This situation is equivalent to having a duplicate sampleID.
            1. To fix, use a different barcode where the duplicate value appears.

ii. If a search for a single barcode returns one record, then check all barcode fields within the record.

1. CTRL+F and paste the barcode value.
2. If the same barcode is used across multiple barcode fields within the same record, update one or more barcode values.

b. *Across separate unrelated applications:*

i. The same barcode may have accidentally been used in two unrelated applications.

Example: Barcode = A000000001 is used in both Mosquitoes (Application 1) and Ticks (Application 2)

1. Determine the correct barcode associations for both sampleIDs, and update the barcode values accordingly

D.5 Parser Error: VERIFY_URL

Problem

The parser is using a **URL** created by Fulcrum to find an externally hosted file that it is not able to locate. URLs are constructed from file path information and the Fulcrum file path must exactly match the file path of the external file.

Causes

- The external file does not exist, and the Fulcrum file path is correct.
- The external file exists and:
  - There is an error in the external file path; the Fulcrum file path is correct.
  - The external file path is correct; there is an error in the Fulcrum file path.
  - There are errors in both the external file path and the Fulcrum file path.

Check For…

- (**Solution A**) Required files have been copied to the external location.
- (**Solution B**) Errors in external file paths. Look to ensure that:
  - The correct files are copied into the correct folders and subfolders.
  - Folder structure, folder names, file names, and file types conform to protocol specifications, including case-sensitivity.
- (**Solution C**) Errors in the Fulcrum file path.
  - Look for errors in input fields that Fulcrum uses to auto-create the file path (and URL).

Solutions

Solution A

Problem: Required files have not been copied to the external host location, and the parser cannot find file paths (URLs) that do not exist. To resolve:

1. Verify required files are organized properly. Use the 'Warbler' Shiny QAQC tool to organize and copy images based on Fulcrum file path data. Look to ensure that:
    a. The correct files are copied into the correct folders and subfolders.
    b. Folder structure, folder names, file names, and file types conform to protocol specifications, including case-sensitivity.
2. Copy properly organized files to the network location where they will automatically be transferred to the external hosting service.
    a. This location is specified in the protocol (i.e., the 'LAI_RAW_validated' folder in the domain-specific CI Dropbox).

Solution B

Problem: Required files have been copied to the external host location, and there is an error in one or more file paths. To resolve:

1. Verify that files have been organized into the correct folders and subfolders. Use the 'Warbler' Shiny QAQC tool to organize and copy images based on Fulcrum file path data.
2. Verify that the correct number of files exist in each terminal subfolder.
3. Fix any folder or file names that do not conform to naming conventions specified in the protocol.
4. Place a new copy of the corrected files to the network location where they will automatically be transferred to the external hosting service.
    a. This location is specified in the protocol (i.e., the 'LAI_RAW_validated' folder in the domain-specific CI Dropbox).

Solution C

Problem: There is an error in one or more Fulcrum file paths. To resolve:

1. View file paths in Fulcrum table format.
    a. View Child Records in Fulcrum since file path data are contained in child records.
2. Identify the record(s) with one or more errors in the file path.
3. Edit the record(s) to fix input data that are the source of the error(s).
4. Save record(s).

Protocol Specific Solutions

Digital Hemispherical Photos for Leaf Area Index

1. (**Solution A**) Use the Warbler Shiny QC tool to copy DHPs that have passed QC checks to the ‘**LAI_RAW_validated**’ folder the domain-specific CI Dropbox.
    a. Images copied to this location are automatically transferred to the NEON external file hosting service and a URL is created.
2. (**Solutions B** and **C**) Use the Warbler Shiny QC tool to ensure that DHPs are organized properly and can be matched to a Fulcrum record in the ***(TOS) Digital Hemispherical Photos – LAI [PROD]*** app.
    a. Resolve any problems identified by the Warbler QC tool.
        i. (If necessary) Fix folder and/or file names according to protocol guidance. Place a new copy of corrected files in the ‘**LAI_RAW_validated**’ folder.
        ii. (If necessary) Update and save image file path data in Fulcrum child records.
3. If the Warbler QC tool cannot run image QC tests and/or create organized folders of images:
    a. Submit an Incident to obtain assistance from HQ Science staff.