2020

Desert Tortoise Monitoring

Data Management Plan

2 June 2020

Linda Allison
Desert Tortoise Monitoring Coordinator
US Fish and Wildlife Service
1340 Financial Blvd, Ste. 234
Reno, NV 89502

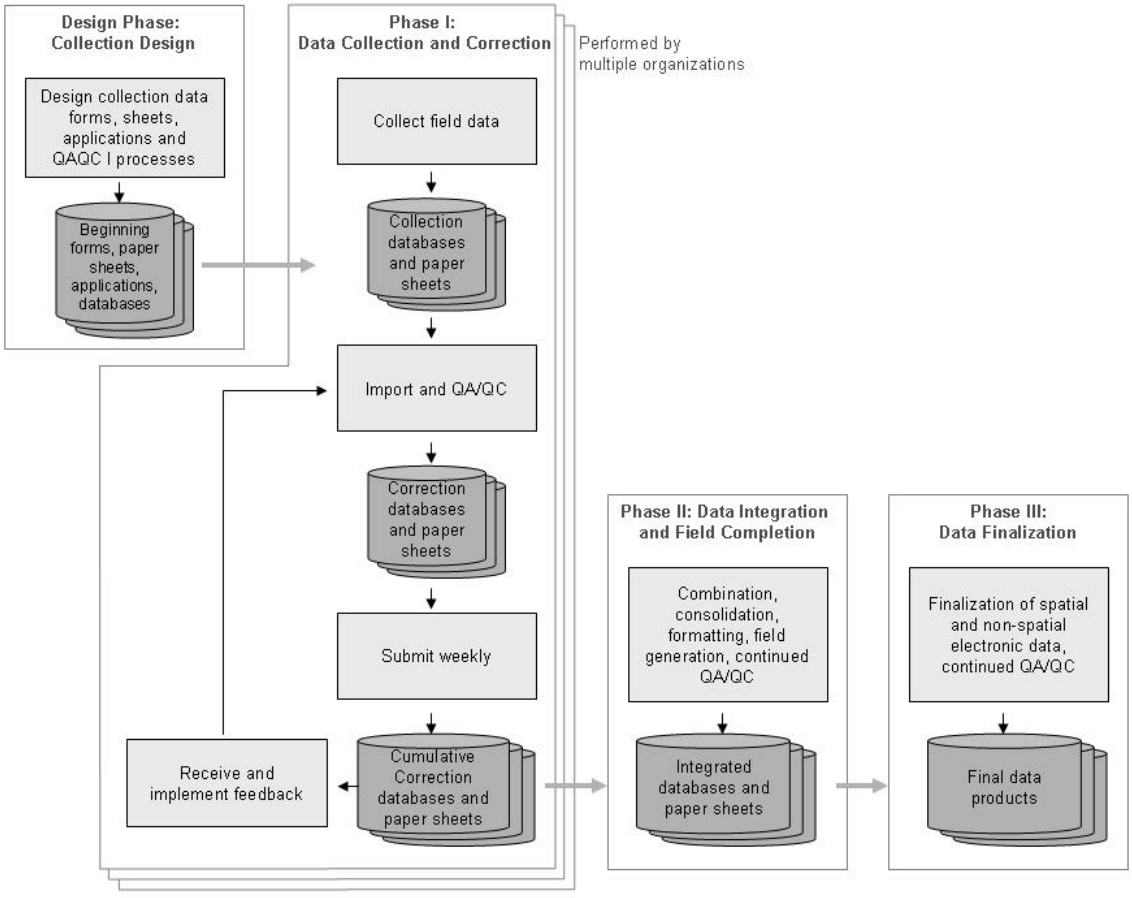
Melissa Brenneman
TopoWorks
110 Dry Run Drive
Noblesville, IN 46060

Table of Contents

Introduction

Desert tortoise monitoring data are organized into two periods of time when data are collected: Preseason (before actual tortoise monitoring begins) and Season (during tortoise monitoring). The preseason data include two core databases: Training Lines and Practice Transects. The season data include two core databases: Transects and G_0 Observations.

Data are collected by multiple survey organizations then combined and processed in a series of phases to create final database products. Data are collected in both electronic and hardcopy format. Collection data forms, sheets, applications and databases are designed. Data quality assurance and quality control (data QA/QC, also known as verification and validation) is performed during the data collection, data integration and data finalization phases. In addition, during the second phase of data processing, after combining data from separate groups, some attribute fields are added and all fields are formatted for final processing. Data finalization occurs in phases, with consolidation and resolution of data inconsistencies representing the first phase. Data are subsequently subject to preliminary analysis to confirm that the results and outputs are consistent with the field season narrative. Once this test and the annual reporting have been completed, the data bases and paper sheets are stored and maintained for ready access during the course of the long-term (25+ years) project. Electronic data are actively hosted and put in a format available for download from the internet. The following diagram describes the overall data flow.



General roles and responsibilities for each phase.

Phase	Responsible Parties
Design Phase: Collection System	GBI – Development of Collection database schemas for each database and their associated forms and checks according to U.S. Fish and Wildlife Service (USFWS) specifications. Includes any updating of database dictionaries that describe fields, field formats, acceptable values, etc. USFWS – Development of database dictionary for use of GBI in design phase. Design of paper data sheets as the hard copy version to match electronic field data. Setting specifications for Transect Tracking and Planning application to be developed by TopoWorks. TopoWorks (TW) – Development of Transect Tracking and Planning application.
Phase I: Data Collection and Correction	GBI – Development of scripts to import the populated Collection databases into their corresponding Correction databases and to automate some of the first level QA/QC checks. Survey Organizations – Responsible for ○ collecting data in the field on paper and electronic data collection devices (Nexus phones). ○ running the synchronization operations to integrate data collected on each individual Nexus phone into its appropriate single, populated Collection database (Preseason and Season). ○ using the import script to bring records from the Collection databases into their Correction databases, running first level QA/QC checks on their Correction databases, correcting errors, and delivering a weekly set of complete Collection and Correction databases, along with completed and scanned paper datasheets and extracted photos. The electronic data are submitted on a weekly basis for interim assessments (below) prior to final delivery. Includes addressing and correcting items in the interim assessments. Includes timely delivery of Phase I products to USFWS and Topoworks. USFWS – Performing interim assessments of each survey organization's populated Correction database. Verification of Phase I product delivery.
Phase II: Data Integration and Field Completion	TW – Compilation of electronic data submitted by Survey Organizations into a combined, well-organized set of (Preseason and Season) databases and associated photos and scanned data sheets. Includes review of surveyor products, combination, consolidation, formatting, generating additional data fields, and preparing Phase II products for data finalization.
Phase III: Data Finalization	TW – Creation of final Preseason and Season spatial and non-spatial products. Includes any additional processing steps beyond the Data Integration and Field Completion phase for standardization across all databases and supporting data sheets, creation of FGDC-compliant metadata, independent review of Integrated Data products from previous phases, generating the final set of data fields, performing QA/QC checks and corrections, identifying and resolving data integrity issues with USFWS, and finalizing spatial and non-spatial electronic data. The Preseason practice transects databases are only minimally reviewed and included as non-spatial products.

Types of QA/QC checks

As outlined in the following sections, each phase of data collection and processing involves performing QA/QC checks and corrections. Errors can enter the process at data collection, but also during later steps such as data editing and during processing steps. For this reason, many checks are performed in more than one phase of data processing. Following are the main types of QA/QC checks that will be performed.

Type	Description	Examples of Errors
Database relationships	Identifies orphans or deviations in the expected number of features related to another feature	Waypoints with no transect. More than 26 waypoints related to a transect
Field domains	Identifies values that are not within a specified range or set	Tortoise mcl > 400
Duplicate records	Identifies duplicate records	Duplicate transects
Multi-field attribute conditions	Identifies records that do not meet specific conditions for attribute values	Observer does not match the lead or follow for the last waypoint "Gsub0:visibility = Visible" but no behavior recorded
Spatial conditions	Identifies records that do not satisfy a spatial relationship	Transects that do not intersect their monitoring strata Observations that are more than 50 meters away from their related transect

Data are collected simultaneously by the two different members of each transect crew, and sequentially by the single telemetry technician working at G_0 sites. The two sets of data, paper and electronic are used to verify and correct one another. During each step of data processing, the electronic data are systematically reviewed (often using automated procedures). If a discrepancy is found in the electronic form, and a different entry is available on the paper form, the paper data take precedence. If an error is suspected on the paper data sheet, the crew recording the data is questioned and other evidence is considered to determine whether there was an error on the paper data sheet. If a discrepancy between the paper and electronic data forms cannot be corroborated one way or the other, the paper data sheet stands as the definitive data entry.

Design Phase: Collection System and Centrally Maintained Information

Electronic data collection system

GBI is responsible for developing the collection database schemas for each type of database (training lines, transects, and G_0) and their associated forms and checks according to USFWS specifications, which includes creation of database dictionaries that describe fields, field formats, acceptable values, etc.

Pendragon Collection database forms are built for Nexus phones based on database dictionaries for training lines, transects, and G_0. The forms constrain data entry for some fields and require data entry for others (see *Appendix A: Database Dictionaries* for fields describing required entry and Pendragon physical domains for constraining entry). Lookup tables for QAQC scripts should be updated before the field season to reflect the spatial coordinates for each stratum and G0 focal site (tables on page 43). Lookup tables for collection and correction databases should also be updated to reflect current surveyor names.

Photo collection system

The electronic data collection system is integrated with digital photo collection. To minimize data entry errors from individually naming then copying file names to the electronic and paper data systems, these photos are integrated into the database.

Paper data collection system

Paper datasheets are created by USFWS after completion of and in order to match the electronic data collection system.

Phase I: Data Collection and Correction

Survey organizations are responsible for collecting data in the field on paper and on Nexus phones. In addition to QA/QC steps associated with collected data, specialists prepare and maintain data collection hardware. The ability of others to evaluate and use the data also depends on the timing and format of databases and scanned paper datasheets that are delivered from Phase I. Much of this section details QAQC procedures, but their teams and the project rely on the specialists for many other associated activities.

Phase I includes implementing the synchronization operation to integrate data collected on each individual Nexus phone into the appropriate single, populated Collection database (training lines, transects, or G_0) for multiple organizations. In addition, they are responsible for correcting errors.

GBI is responsible for developing scripts to import the populated collection database into *Correction databases* and developing scripts to automate some of the first-level QA/QC checks.

Survey organization responsibilities include the following QA/QC steps:

- Importing the data from the populated Collection databases into the Correction databases.
- Running scripts to perform the required automated checks designed to identify common errors that can be best corrected by the survey organization. It is the survey organization's responsibility to perform an initial level of QA/QC and correct these errors (see specific checks below).
- Performing final, non-automated visual checks for errors. Examples of errors include odd times, incomplete fields, fields missing data, etc.
- Making and documenting corrections in an Errors table. Corrections to records should be clearly stated and include the fields and values that were changed. For example, instead of "corrected time", use "changed time to match datasheet." Unusual entries that are not corrected should also be documented in

the violations table with a resolution field value explaining why the entry could not be corrected. For example, in a case where a live tortoise was found, but its MCL was not recorded, for the resolution field use "MCL could not be recorded because tortoise retreated into burrow." It would also be helpful to include this comment in the "comments" field on the observations table. Time and date fields may be edited to reflect the correct time, but the TimeStamp fields populated by the Nexus phone should not be changed. Instead, in the Violations table, describe the origin of any anomalies in the TimeStamp fields, and indicate "exception allowed."

- Responding to and making corrections identified in the USFWS interim assessments of their data. The interim assessments are not automated and it is difficult to identify in advance all the possible corrections that may be required. However, examples from previous years include: incorrect times; sex record M or F, but comment field indicating uncertainty; mixing observers between teams; inconsistency in recording of tag numbers; G0 tortoises that are scored "not visible" but behavior is not "unknown",; incorrect numeric entries into text fields that cannot therefore limit numeric entry errors (e.g. waypoint "21" recorded as waypoint "21."); duplicate waypoints; etc.
- Delivering complete Phase I products, Collection and Correction databases (training lines, transects, and G0), for the next phase of data processing. Tables for training line and LSTS transect data will be packaged into a single Preseason database whereas all other transect data and all G0 data will be packaged in the Monitoring season database. Data collection and correction must be completed by survey organizations and the Monitoring season databases delivered to USFWS and Topoworks within 10 days of final data collection, no later than **5 June 2020**.

The following first level processes and QA/QC checks are described for each database (training lines, transect, and/or G_0), irrespective of whether the final product will be a preseason or monitoring season database.

Prepare laptop systems and software

At least 2 weeks before the beginning of training, QA/QC specialists should have the following system in place, prepared to install software provided by USFWS and the database developer:

Hardware:

- Intel Pentium Dual Core processor (2.0 GHz+) or newer
- at least 4GB of RAM
- at least 10 GB of free hard drive space

Software:

- Windows 7/ 32 bit or 64 bit with all updates installed
- Microsoft Access 2010
- ArcGIS Desktop 10 (ArcView level license) (for machines involved in transect planning)

NEXUS PHONE Preparation and Recovery from System Failure

The recovery procedure should only be performed by QA/QC personnel if restarting the phone does not resolve the issue. This method clears all the data from the device and then does a fresh install of the Pendragon software.

Data specialists are responsible for checking and conditioning the phones, then configuring the phones with the season's software and forms, and logging into Pendragon software on each phone with the phone's unique username and single password assigned to all phones in the project before handing over the phone for data collection. Conditioning procedures include cleaning the screen covers, recharging the phones, and identifying any units from the outset that do not hold charge well or have faulty accessories, such as charging cables. Initial preparations on all team phones can take several days, so leave sufficient time. Preparation of phones with the season's software and forms mirrors procedures used later to recover individual phones from system failure.

Nexus phone set-up and recovery procedures need a wireless internet connection (Wi-Fi), Android APK installer for Pendragon, and the username and password for our Pendragon account. “Sync” operations in this text refer to the interaction between the Nexus phone and the internet.

Setting up Nexus phones at the start of the season, also used for recovery from system failure

This procedure will delete all Pendragon data collected on the Nexus phone. Make sure you the phone is connected to internet before you perform this operation.

1. **Install Pendragon on the phone (If Pendragon forms is already installed then skip to step 2)**
   Make sure the Nexus phone is connected to the internet and then open the Chrome browser. Navigate to http://pendragonforms.com/support.html. Look for “Download Pendragon Forms APK v1.956A” link and click on it to download the latest Pendragon Forms APK. Tap on app drawer icon (the middle icon on the bottom of the phone, white circle with six dots), open “Downloads,” and then click on “PendragonForms-v1.956A.apk” to install the Pendragon application.

2. **Clear all the data, server address, username, and password from Pendragon software**
   On the Nexus phone, tap on the app drawer icon (a white circle with six dots on the bottom of the phone in the middle) then scroll down and open the phone settings by tapping the “gear” icon. Scroll down to “Apps” and tap it. Scroll down on the list of applications and click on “Pendragon Forms”. Click on “Storage (xx MB used in Internal Storage)” and then click the “CLEAR DATA” button. On the confirmation dialog “Delete App Data?” tap “Ok” to delete all the data from “Pendragon forms.”

3. **Sync Pendragon and load the collection database forms**
   Open the Pendragon application. You will be prompted to “Please enter your server address”. Make sure “Pendragon Account #” is selected and then enter “2951” and tap on ”Connect.” It will redirect to the Pendragon login screen. Enter the username and password for the current phone and tap “Ok”. The phone will automatically sync and download all the latest forms for the username.

4. **Test Phone Setup**
   Open the Pendragon application, click on the Forms button, then select the Transects_17 form. Click on the + sign to create a new record. Collect sample records to confirm that the perpendicular distance calculation and GPS grabs are working and that the crew names are correct and complete for your group. Make sure you delete these sample records on the Nexus phone before you hand it over to the crews. The device is now ready for field data collection.

Updating the Collection database on the Nexus phone

During training or early in the field season, small updates are often made to the collection database. Use the following procedure to replace the older version of the Pendragon database on the Nexus phone with the updated Pendragon database version. Also, any time the phones are synched to upload data, any database updates will automatically be downloaded as described below.

1. **Material to have at-hand**
   - The Nexus phone connected to the internet using Wi-Fi

2. **Install the newest version of the Pendragon forms.**
   - Open the Pendragon application on the Nexus phone. Tap on the “Sync” button to download the latest collection database forms to the phone.

Transfer of data from field crews to data specialists - Initiating first level QA/QC

Whenever paper data are transferred to the QA/QC specialist (also during training), the following procedures are applied:

While crews are present, the QAQC specialist should:

- Evaluate legibility of handwriting
- Confirm that the Nexus phone is functioning correctly
- Review paper sheets for any blank fields
- Check whether the transect drawing indicates standard or non-standard
    - If standard, check that this matches the associated field
    - If non-standard, check that this matches the associated field, and that at least one of the 3 "obstacles" fields has been used to indicate the reason for using a non-standard transect
- For transects with interruptions
    - Check that the transect was marked as non-standard
    - Check that interruption points and new transect records are marked on datasheets

    If any tortoises are indicated as "mcl_ greater_180=unknown", question the crew for any further, potentially discriminating information, and remind them that every effort should be made to collect information indicating "yes" or "no" for this field.

Importing Collection databases into Correction databases

To download the latest data into the collection database, open the FORMS32K.accdb database located in "C:\PendragonForms\" folder. After opening, navigate to the "External Data" tab on top and click on "ODBC Database". In the Dialog, select the first option to Import the source data and then select the "LDS_Database" Data Source Name from the "Machine Data Source" tab. Click the "Ok" button. A list of tables available in the online database is displayed. Select all the tables starting with vw_noimages and tables starting with form_id that have not already been selected and click "Ok". Access will copy all tables from the online database into the current database. This will complete the collection database download procedure.

Before running the QA/QC scripts for the first time make sure that the Correction database (Import_QAQC.mde) exists in the same folder on the machine ("C:\PendragonForms") as the Collection database. If not, copy the latest version of the Correction database to this folder.

Open the QA/QC scripts database by double-clicking "Import_Correction.mde" file.

Under "Forms" on the left-hand list of objects, double click on "2020_LDS_Import_QAQC" to open it. Click on "Import Raw Data" at the bottom of the form to import the Collection database into the Correction database. If the training database also needs to be imported, check the "Import Training Data" checkbox before clicking "Import Raw Data". The following automatic functions are executed as part of the import process without further user input:

Operations performed throughout all tables in all databases

- Null *string* values in the Pendragon Collection database are assigned "null" in the Correction database
- Null *numeric* values in the Pendragon Collection database are assigned -99 in the Correction database
- Null *date and time* values in the Pendragon Collection database are assigned 1:00AM in the Correction database

Training database (polystyrene tortoise model transects and observations)

- Populate trial_number, team_number, training_line_color, transect, training_date, tran_bearing fields in the child form from transect form.

Transect databases

- Populate transect number, stratum, team_num, group_, date_ fields in all child forms from transect form. For practice transects at the Large Scale Translocation Site, the stratum should be LSTS.

- Populate duplicate_tran in the Transects table Initiate with the value “N”. (QAQC specialists will subsequently run the duplicate check, then the duplicate_tran field would be changed to Y. Any transects that are duplicates will not be processed like the interrupted transects even though they have a decimal (.9) in their tran_num.
- Populate latitude and longitude values from gps grab in all forms
- Populate transect start and end time from Wpt 1 and 99.
- Populate drop off time and return time from Wpt 0 and 100.

G_0 database

- Populate start_time and end_time fields for G_0 table from first and last G_0 observations.
- Populate latitude and longitude values from gps grab in all forms
- Populate date, time_, G0_site, and group_ in all child forms from G_0 forms.

All errors encountered during the import process are logged into the file “LDS_Import_Log.txt” located in “C:\Program Files\Forms3”. Check this file after each import to see if any errors were encountered during the import process. This year photos will only be present in the collection database (not available in the correction database). A separate procedure will export photos from the collection database into jpg files.

If the database cannot be imported, there are two identified issues: 1) Presence of single- or double-quotes in the photo file or comments fields. These characters must be edited in the Collection database before it can be imported. 2) Comments string has too many characters. The string needs to be shortened in the Collection database before it can be imported.

Once data has been imported into the Correction database, the Collection database should be updated so that records in each table are marked as exported (exported = 1). Each time records are imported into the Correction database, the number of records between the Correction and Collection databases should reconcile, and this should be checked before executing a query to mark records as exported. To confirm that the databases are consistent, the data specialist should track the number of records they create or delete in the Correction database. After accounting for these records, the number of exported records in the Collection database should match the number of records in the Correction database. Once the records in each table are checked, a query is manually executed in the Collection database to indicate all records have been imported to the Correction database

- Click “View & Download”
- Click the “view/edit data” button for the table
- Click the SQL tab and enter (form_id_8000000## refers to the table id):
  UPDATE ‘form_id_8000000##’ SET ‘exported’ = 1 WHERE 1
- Click the “Go” button.
- This needs to be done for each table in the database. You can access the next table by exiting out of the current one, then clicking on the next table, or you can change the form id (form_id_8000000##) in the current query to match the form id of the next table.

Note: It is important to avoid importing data while crews are actively syncing in the field. Otherwise, it is possible that records that were not exported will be marked as though they were. Any records synched to the Collection database between confirming concordance with the Correction database and running queries to indicate exported = 1 will actually not be in the Correction database but will not be imported the next time Import Raw Data procedure is run.

Specific, automated QA/QC checks for errors and inconsistencies

The following describes the specific required scripts that will be executed during first level QA/QC. At a minimum these scripts should be run and errors addressed on a weekly basis.

After executing the QA/QC checks all errors encountered are logged into the Error tables. For subsequent QA/QC checks, all uncorrected errors will be logged again into the Errors table unless the error status field is marked as “exception allowed”, “unresolved” or is blank.. All errors from Transect database tables

(Transects, Waypoints, OppLiveObs, OppCarcObs, TranLiveObs and TranCarcObs) are logged into "Errors_Transects" table, all errors from G0 database tables (G0_Start, G0_Obs and G0_OppLiveObs) are logged into "Errors_G0" table and all errors from Training database tables (Tran_Train and Tran_Obs) are logged into "Errors_Training" table.

Training Database Checks

Training transects (training)

Tables included in check: Train_Tran.

- Checks for duplicate transects for the same team on the same day with the same trial_number and same line color.
  Error description: duplicate training transect.

Tortoise ID (training)

Tables included in check: Train_Obs.

- Checks for duplicate tortoise_id in same day, same team, same line_color. Error description: contains duplicate tortoise_id, trial_number, team_number, and training_line_color values

Observer name and position (training)

Tables included in check: Train_Tran, Train_Obs.

- Checks for observer_name and observer_position not matching lead and follow fields in the Train_Tran table.
  Error description: observer_name and observer_position do not match lead or follow in Train_Tran table.

Transect segment number (training)

Tables included in check: Train_Obs.

- Checks for transect_seg_num not matching calculated value from tran_bearing and start_post.
  Error description: transect_seg_num does not match tran bearing and start post.

Time (training)

Tables included in check: Train_Tran, Train_Obs.

- Checks for training_start_time not before training_end_time.
  Error description: training_start_time is after training_end_time
- Checks for observation_time not between training_start_time and training_end_time.
  Error description: observation_time is not between training_start_time and training_end_time

Radial Distance (training)

Tables included in check: Train_Obs.

- Checks for radial_distance_m with more than one decimal place.
  Error description: radial_distance_m has more than one decimal place

Bearing (training)

Tables included in check: Train_Obs.

- Checks for local_bearing not within 40 degrees of tran_bearing.
  Error description: local_bearing is not within 40 degrees of tran_bearing

Perpendicular distance (training)

Tables included in check: Train_Obs.

- Checks for perp_distance_m greater than radial_distance_m.

Error description: perp_dist_m is greater than radial_distance_m.

- Checks for perp_distance_m greater than 25 m.
  Error description: perp_distance_m is greater than 25 m.

Transects and G_0 Database checks

Database relationship checks

- Missing waypoints (transects)
  Tables included in check: Waypoints
  - For non-interrupted transects, identify transects that are missing drop off, start, end, or return waypoints. The Error record should be recorded for the Transects table, giving the tran_prime_key for the transect with the missing waypoint.
    Error description: missing waypoint xxx.
  - For interrupted transects (those with end_tran_part = Y, note: do not base this on the xxx.x format as that format is also used for duplicate walks of a transect), identify main transect (xxx) with missing drop off and start waypoint and for the last part (xxx.x) identify missing end and return waypoints. The Error record should be recorded for the Transects table, giving the tran_prime_key for the transect with the missing waypoint.
    Error description: missing waypoint xxx.

- Orphan records (transects and G_0)
  Tables included in check: Waypoints, OppCarcObs, OppLiveObs, TranCarcObs, TranLiveObs, G0_Obs
  - Check for orphan records created in sub-forms when the transect from the transects table has been deleted, G0_start record has been deleted, or training transect record has been deleted.
    Error description: missing parent record.

Duplicate record checks

- Duplicate transects (transects)
  Tables included in check: Transects
  - check for duplicate transect numbers.
    Error description: duplicate tran_num.
    Duplicate transects should be numbered by concatenating the original tran_num with ".9". If more than one duplicate is walked, continue by added extra decimal values of 9 (.99, .999, etc.). Duplicate transects that also have interruptions should be numbered first with the .9 for the duplicate transect, then a following decimal to describe the portion of the transect the record represents (e.g. 341.9, 341.91, 341.92).
- Duplicate waypoints (transects)
  Tables included in check: Waypoints
  - check for duplicate waypoints in the same transect.
    Error description: duplicate waypoint.

Multi-field attribute condition checks

- Lead and Follow (transects)
  Tables included in check: Transects
  - Check for transects that have the same observer name for 'observer1' and 'observer2' fields.
    Error description: observer1 and observer2 are the same.
  - Check for null 'observer1' and 'observer2' values;
    Error description 1: observer 1 is Null
    Error description 2: observer2 is Null.

- Time (transects, and G_0)
  Tables included in check: Transects, Waypoints, TranCarcObs, TranLiveObs, G0_Start, G0_Obs
  - Check all time values against these logical domains from database dictionary:

Table	time_field	Begin domain	End domain
Transects	do_time	4:00am	10:00am
Transects	tran_start_time	5:00am	10:00am
Transects	tran_end_time	8:00am	6:30pm
Transects	ret_do_time	8:00am	6:30pm
Waypoints	time_	5:00am	6:30pm
G0_Obs	time_	5:00am	6:30pm
TranCarcObs	time_	5:00am	6:00pm
TranLiveObs	time_	5:00am	6:00pm
G0_Start	tran_start_time	5:00am	10:00am
G0_Start	tran_end_time	8:00am	6:30pm

    Error description: [time_field] is not within domain [beg] to [end]. (Substitute the field name and domain values, Example: do_time is not within domain 4:00:00 AM to 10:00:00 AM.
  - For interrupted transect segments other than the first main transect record, do not apply start_time and end_time checks.

- Date (transects, and G0)
  Tables included in check: Transects, Waypoints, TranCarcObs, TranLiveObs, G0_Start,
  - Check all date values against logical domain from database dictionary
    Error description: [date_field] is not within domain [beg] to [end]. (Substitute the field name and domain values, Example: date is not within domain 3/01/2020 to 5/31/2020.
- MCL (transects and G_0)
  Tables included in check: OppCarcObs, OppLiveObs, TranCarcObs, TranLiveObs, G0_Obs
  - Check for mcl_mm greater than 180, but mcl_greater_180 not yes
    Error description: inconsistency between mcl_mm and mcl_greater_180.
  - Check for mcl_mm less than 180, but mcl_greater_180 not no
    Error description: inconsistency between mcl_mm and mcl_greater_180.
  - Check for mcl_mm greater than 0, but mcl_greater_180 is unknown.
    Error description: inconsistency between mcl_mm and mcl_greater_180.
  - Check for mcl_mm equal to 0 if temperature is less than 36°C.
    Error description: Mcl_mm is 0.
  - Check for carc_condition intact, but mcl_mm is Null.
    Error description: inconsistency between carc_condition and mcl_mm.
  - Check for carc_condition disarticulated, but mcl_mm is not Null.
    Error description: inconsistency between carc_condition and mcl_mm.
  - Check for tortoise location is not burrow or RockBurrow or SoilBurrow, and temperature is less than 36°C, but mcl_mm is null.
    Error description: inconsistency between tort_location and mcl_mm.

- Temperature (transects and G_0)
  Tables included in check: OppLiveObs, TranLiveObs
  - Check for inconsistency between temp_C and temp_greater_35C,
    Error Description: inconsistency between temp_C and temp_greater_35C
- Visibility (G_0)
  Tables included in check: G0_Obs
  - If visibility is "no" then behavior must be "unknown" (G_0).
    Error description: Inconsistency between visibility and behavior.

Spatial condition checks

- UTM zone
  Tables included in check: Waypoints, OppCarcObs, OppLiveObs, TranCarcObs, TranLiveObs, G0_Obs
  - Check for strata other than "BD" and "GB" having zone 12 or if gps zone and manual zone are missing.
    Error description1: gps_zone or manual_zone are 12, but stratum is not BD or GB.
    Error Description2: missing gps zone or manual zone.
- Easting and northing
  Tables included in check: Waypoints, TranCarcObs, TranLiveObs, G0_Obs
  - Check for easting or northing coordinates that fall outside of the assigned monitoring stratum. Coordinates for this check are in the Stratum_Info table of the database and should match coordinates in this plan (see *Spatial condition checks* for Phase II in this Plan for stratum and G0 coordinates).
    Error description1: easting or northing are not within stratum boundary.
    Error description2: easting or northing are not within G0 site boundaries.
- Missing location data
  Tables included in check: Waypoints, OppCarcObs, OppLiveObs, TranCarcObs, TranLiveObs, G0_Obs
  - Check records with missing automatic and manual gps coordinates (transects, G_0 observations).
    Error description: missing gps or manual easting or northing.

Procedures for addressing errors

Whether an error was identified automatically or by visual examination, at least one and possibly more error records must be created to document these inconsistencies and any resulting changes to the database. It is common to change a database only to later discover that the original data were unusual but not incorrect, or that a different correction is required. In these cases, revisions are straightforward if you have documented your edits, and are a headache if you have not.

Error tables and error records

All errors encountered are logged into either Errors_Transects, Errors_G0, or Errors_Training tables in the Correction database. Following are the fields in each of the error tables. All the error tables share "Common fields for all error tables," but also have specific fields (d1, d2, etc.) depending on the error table. Fields (a)-(f) and all specific fields are automatically populated by the scripts, whereas the (g)-(j) fields are manually entered by QA/QC specialists as they address each error.

Common fields for all error tables

a. ID – auto-number used to identify record
b. date – the date when the QA/QC scripts were run
c. table_name – table name of the error record.
d. fields specific to error table.
e. prime_key – primary key of the table with error record
f. error_desc – short description on type of error found
g. old_value – old incorrect value of the field
h. new_value – null, new correct value entered by QA/QC specialist

i. resolution – null, steps taken by QA/QC specialist to resolve the error
j. resolver – null, name of agency correcting the errors
k. error_status – null, status of error after correction

Specific fields in Transects Error table
d1. tran_date – date when the transect was walked
d2. tran_num – transect number for error record
d3. stratum – stratum for the error record
d4. team_num – team number for the error record
d5. wp_obs_num – waypoint or observation number depending on error table

Specific fields in G0 Error Table
d1. G0_date – date when the G0 observation was recorded
d2. G0_site– G0 site for the error record, null if not applicable
d3. group_ – group for the error record, null if not applicable
d4. tort_obs_num - tortoise ID or observation number depending on error table

Specific fields in Training Error table
d1. training_date – date when the training record was collected
d2. trial_num – trail number for error record
d3. team_num – team number for the error record
d4. training_line_color– training line color for the error record, null if not applicable
d5. transect - transect number for the error record.
d6. tortoise_id – tortoise_id for the error record, null if not applicable

Records with errors in the data tables can be identified using the information in table_name, prime_key, tran_num, stratum team_num, wp_obs_num and tran_date fields of the Errors table. If information is not available, the associated fields will be either null (text field), -99 (numeric field), or 1:00AM (time fields). As also described in the sequence of procedures below, if one of the above identifier fields such as date, tran_num, or stratum was part of the identified error, you will ultimately also edit these identifier fields in both the data table and the error table so they both reflect the correct information; these edits allow the error record to continue to match its associated and repaired record in the data tables.

Add error records for errors not identified by QA/QC scripts
For errors that were identified during systematic visual inspection of tables (see *Non-automated checks for errors and inconsistencies*, below), the error records are obviously not generated automatically in the Errors table; a manual error record will need to be created. The QAQC specialist will have to populate fields (b)-(k) and fields specific to the associated Error table for each record that is manually created. Care should be taken to enter the information accurately as these fields are critical for identifying the correct record.

Decide whether the identified error is correctable
To determine whether an identified error can be corrected, review the paper datasheets or contact crew members, if necessary. The two sets of data, paper and electronic, are used to verify and correct one another. During each step of QA/QC, the electronic data are systematically reviewed (often using automated procedures). If a discrepancy is found in the electronic form, and a different entry is available on the paper form the paper data take precedence. If an error is suspected on the paper data sheet, the crew recording the data should be questioned and any other evidence considered to determine whether an error was committed on the paper data sheet. If a discrepancy between the paper and electronic data forms cannot be corroborated one way or the other, the paper data sheet stands as the definitive data entry.

Correct errors in the QA/QC Database
If the error can be corrected, correct the erroneous record in the data table and prepare to use the new information to update the associated error record.

Update the error table resolution fields and the record's error_status field to explain action taken

After the error has been fixed or determined that it cannot be fixed, the old_value, new_value, resolution, resolver and status fields in the appropriate Errors table (Training, Transects, or G0 Error table) must be manually completed for the error record. Note that if one of the identifier fields such as date, tran_num, or stratum was part of the identified error (it is now reported as an "old_value"), you must edit these identifier fields in both the data and error tables to reflect the correct new_value; these edits allow the error record to continue to match its associated and repaired record in the data tables.

Whenever you make a determination on the error_status, you are a "resolver", and this field should contain the name of the person/agency addressing the error. The "resolution" column should clearly mention steps taken to correct the error or should explain why the error could not be corrected.

Error_status = Resolved

Any changes to data records should be represented by error records with an error_status value of "resolved". If you need to make a correction to a record for which there is no error record, then you must manually add the error record and use the resolution field to describe the correction, as described above. If you have thoroughly reviewed the database records and paper datasheets, as well as consulted the field monitors, and a correction is still not possible, you should mark the record as "exception allowed" or "unresolved" and use the resolution field to explain why the exception is being allowed or why the record cannot be resolved during QAQC I. The "old_value" columns, if automatically populated, should contain the original incorrect value of the field and "new_value" column should contain the correct value replaced. For example if the time field is corrected, the resolution should be "changed time to match the datasheet" and the "old_value" field should contain 6:00PM and "new_value" should contain 6:00AM.

Error_status = Unresolved

If the error cannot be corrected during QAQC I, but needs further evaluation during QAQC II or III, its error_status is "unresolved." You are still the "resolver" for QAQC I because you addressed the error, so the resolver field should contain your name or initials. Marking the status to "unresolved" will skip the error (and not log it again in error tables) in subsequent QA/QC checks. The resolution field should describe why the error_status is unresolved and provide any information that might help evaluation during QAQC II or III.

Error_status = Exception allowed

An exception may be allowed because a correction could not be made (i.e. the paper datasheet matched the electronic data or the field monitors could not explain the discrepancy) or because the record did not represent an error (i.e. the perpendicular distance really was greater than 25m, the tortoise retreated into a burrow, the tortoise was too small to tag).

It is especially important to describe why error records are being left as exceptions. For example, if an mcl_mm field is blank, it matches the paper datasheet, and could not be resolved by consulting with the field monitors, the resolution field should be "mcl_mm not measured, matched datasheet, team could not explain". Another example might be that one of the Nexus phone units failed and the data had to be entered into the electronic units later in the day. When this happens, there will be error records generated because the TimeStamp when the data was entered will not match the times entered from the paper datasheets. In these cases, the error records should be marked as "exception allowed" and the resolution field should note the device failure such as "Nexus phone failed, data was entered manually later in the day". If a crew forgets their Nexus phone and only records the data on the paper datasheets while walking the transect, these errors would also be marked as "exception allowed" but the resolution would be "team forgot Nexus phone, data was entered manually later in the day". All records marked as "exception allowed" should have an explanation in the

resolution field. Ideally, the associated records in the database table (Transects, Waypoints, etc.) would also have an explanation in the comments field. Make any explanations in database table comments fields as clear and concise as possible. They do not need to include all of the details that you might include in the error table resolution field. If you changed any part of a record, the associated error record should not be marked as “exception allowed”. Marking the status to “exception allowed” will skip the error (and not log it again in error tables) in subsequent QA/QC checks.

Error_status = Script error
In this case, an error record has been generated due to overly restrictive rules in an automated check. No edit to the original record is required, and if possible a revised version of the QAQC database will be created to avoid this mistake. If that happens, the error records can be removed from the Error table; the error records will not be regenerated the next time error scripts are run. The QAQC database has been revised in the past when interrupted transects are documented with new electronic records for each segment; the start time for the segments may be later than scripted, although the transect was started in the correct window of time. Alternatively, if the QAQC database is not revised, these resolutions are equivalent to “exception allowed,” signaling that no further review of this case is required. For this reason, it is important to use this resolution deliberately and to alert all other data management cooperators to this original issue. The QAQC database may not be revised simply to address a transmittered animal that has wandered beyond the scripted UTM polygon that was anticipated before the field season.

Update the paper data sheets to reflect changes and match the electronic database
If an error on the paper data sheet is corroborated, the QA/QC specialist identifying and determining the data error draws a single line through the erroneous data on the paper sheet, neatly prints in ink the correct data above the erroneous data, then initials and dates the correct data entry. This data correction is then input into the errors table of the digital database to maintain a record of the correction.

After the paper sheet has been scanned, it should no longer be edited by hand. Instead, all further corrections will be made electronically as Adobe “sticky notes.” These notes are anchored to the point where they are inserted, and automatically indicate the creation date. Although an identifier is added for the commenter, care should also be taken to type in the commenter’s initials at the start of the note. Necessary edits are documented by using language like, "Change field "zzzzz" from xxxx to yyyy." If multiple related changes need to be made, a general explanation is provided first: "Crew confirmed verbally that the tortoise was in a pallet, not a burrow. Change "tortoise_location" field from pallet to burrow. Change field "burrow_visibility" from high to null. Change field "tortoise_in_burrow_visibility" from high to null." The edited version should be given the same name as the original file so that it will automatically replace earlier versions of the scanned file (see “Deliver Paper Datasheets,” below)..

Non-automated processing steps and checks for errors and inconsistencies

The following lists describe checks for common errors that may not be identified by the automated checks, or that may have appeared in the process of making corrections for the automated checks. The lists are not exhaustive, but illustrate ways to methodically examine data tables for missing or inconsistent values.

When an error or inconsistency is found, it should be manually entered into the appropriate Error table, taking care to correctly identify the record by its TableName and TableRecord_ID fields. The error description should follow a similar format as the automated checks, and due to the types of checks that are possible manually, will often be of the form: Invalid [field name] (for instance, “Invalid team_num”, “Invalid observer_name”). Many of the non-automated checks are purposefully redundant of the automated checks above or involve checking for unusual entries in fields that are populated via a pull-down list. This provides a last review of the data to catch errors or inconsistencies that might have been introduced earlier in the QA/QC I process. Redundant checks are in italic below.

Checks based on format type and common to all tables

- Check that timestamps are valid by sorting them in ascending and descending order. **The TimeStamp fields are never edited, but descriptions of the cause for invalid times should be provided in the Violations table and the violation status should be 'exception allowed'.**
  Error description: Invalid TimeStamp
- Check numeric fields for errors (e.g. decimals after numbers, zero preceding number). Sort by field. These errors can occur where the field type is text, such as tortoise IDs (Transects and G0 forms), and Easting/Northing fields.
  Error description; Invalid number format
- Use the dropdown arrow for each text or categorical field to view unique entries looking for typos or values that are not consistent, such as observer names with typos, entries with extra punctuation, values that are abbreviations, etc.
  Error description; Invalid text entry
- Sort to view records with any comments, looking for indications that there are questions about field values.

Train_Tran Table

- Sort by total_time, look for unusually high or low numbers
- Sort by team_number, look for unusually high or low numbers
- Sort start_post descending, look for letters higher than "L"
- Sort transect_seg_num descending, look for numbers higher than 8
- *Sort training_date ascending and descending, look for dates outside of training sessions*
- View unique values for lead and follow, look for misspellings of observer names

Train_Obs Table

- View unique values for observer_name, look for misspellings of observer names
- *Sort tortoise_id in descending order, look for ID#s larger than the known highest ID#(288)*

Transects Table

- *Check if do_time, tran_start_time, tran_end_time and ret_do_time are null (1:00AM). Sort by time.*
- Sort by the combination of tran_standard, unplanned_modification, terrain_obstacles, subst_obstacles, and other_obstacles.
  - Check if tran_standard if null.
  - If tran_standard = N, confirm there are listed obstacles.
  - All unplanned_modification = Y records should also indicate tran_standard=N and obstacles should be indicated.
  - For interrupted transects, the Correction database should indicate tran_standard=N and should aggregate all listed obstacles from each segment of an interrupted transect in the main transect record (first segment) for that transect. The main record should also have the do_time, tran_start_time, tran_end_time, and ret_do_time as it applies to the entire transect. (Note that the Collection database differs from the Correction database in that each segment of an interrupted transect should list obstacles encountered on that segment.
- Make sure observer names are spelled consistently. View unique values for observer1 and observer2 fields.
- *Check for invalid entries in the date field. Sort dates ascending then descending.*
- *Check do_time, tran_start_time, tran_end_time and ret_do_time for accuracy. For example, do_time should be earlier than tran_start_time, etc.*
- Compare date and time fields with Nexus phone date/time stamp. Do not edit the TimeStamp fields, but if necessary, corrections can be made to the individual date and time fields.

Waypoints Table

- *Lead or Follow fields should be null only for waypoints 0, 99 and 100, or when a waypoint is taken at an interruption (end_part=Yes).*
- Check for waypoint numbers that are missing or invalid (duplicate, or with decimal places).

OppCarcObs Table

- Check for the accuracy of detection numbers. Sort by transect number, then detection number.
- *mcl_mm should be null for disarticulated carcass and mcl_mm should be > 0 for intact carcass.*
- Sex should be unknown for mcl_mm <180.

OppLiveObs Table

- Check for the accuracy of detection numbers. Sort by transect number, then detection number.
- Sort by neighboring fields location, burrow_visibility, tortoise_in_burrow_visiblity, tortoise_visibility. If tortoise location is burrow, burrow_visibility and tortoise_in_burrow_visibility should not be null. For other locations the tortoise_visibility field should not be null.
- If tortoise location is burrow, dist_to_burrow_m should be 0. Because distances for this field are estimated to the nearest meter, dist_to_burrow_m field should otherwise only be 0 if the tortoise is within half a meter of the burrow entrance. Note that a measured distance can be reported to one decimal place.
- Dist_to_burrow_m cannot be null (-99); check to see whether 100 (none seen within 15 m of the tortoise) is appropriate instead.
- The mcl_mm field should be null if the temperature is greater than 35°C.
- Sort descending on mcl_mm and look for unrealistically large values (check with crews for values over 320).)
- Sex should be unknown for mcl_mm <180.
- If new tag attached is “Yes” then new tag number should not be null and if existing tag is yes then existing tag number should not be null
- *If the tortoise is in the “Open” and temperature is less than 36°C then the mcl_mm, body_condition_score, nares_appearance, nares_discharge, and ticks should not be null.*
- Check for New tag number format. “FWxxxx” is correct. “FW-xxxx” is not.
- Sort on tort_not_handled, tort_not_handled_other, existing_tag, tag_attached, MCL, sex, and BCS fields and for null entries of the latter 4 fields, confirm that a reasonable explanation exists in the tort_not_handled or existing_tag fields.
    - If tag_attached is “no” then either existing_tag is “yes”, or tort_not_handled is not null.
    - tag_attached, MCL, sex, BCS, nares_appearance, nares_discharge are not null
    - if tag_attached, MCL, sex, BCS, nares_appearance, or nares_discharge are Unknown, tort_not_handled or tort_not_handled_other are not null

  Error description: inconsistency between conditions and data collection on tortoise
- Check whether the observation was made between tran_start_time and tran_end_time. If so, check with crew about why it was not correct to collect distance information.

TranCarcObs Table

- Check for the accuracy of detection numbers. Sort by transect number, then detection number.
- Look for radial distances with more than one decimal places. Radial distance should only be recorded to one decimal place.
- *If the carcass condition is Disarticulated then mcl_mm should be null.*
- *If the carcass condition is Intact then mcl_mm should not be null.*
- If existing tag is “Yes” then existing tag number should not be null.
- Sort descending on mcl_mm and look for unrealistically large values (check with crews for values over 320.)
- Sex should be unknown for mcl_mm <180. Crews may be able to identify males below this size, but do not accept these entries without conferring with crew.

TranLiveObs Table

- Check for the accuracy of detection numbers. Sort by transect number, then detection number.
- Radial distance should only be recorded to one decimal place.
- Sort by neighboring fields visible, location, burrow_visibility, tortoise_in_burrow_visiblity, tortoise_visibility. If tortoise location is burrow, burrow_visibility and tortoise_in_burrow_visibility should not be null. For other locations the tortoise_visibility field should not be null.
- If tortoise location is burrow, dist_to_burrow_m should be 0. Because distances for this field are estimated to the nearest meter, dist_to_burrow_m field should otherwise only be 0 if the tortoise is within half a meter of the burrow entrance. Note that a measured distance can be reported to one decimal place.
- Dist_to_burrow_m cannot be null (-99); check to see whether 100 (none seen within 15 m of the tortoise) is appropriate instead.
- If tortoise location is burrow, then tortoise_heading entries must include Profile, FacingIntoBurrow, or FacingOutOfBurrow. Other entries may be added, too, but one of these is required.
- Dist_to_burrow_m cannot be null (-99); check to see whether 100 (none seen within 15 m of the tortoise) is appropriate instead.
- The mcl_mm field should be null if the temperature is greater than 95°F.
- *If the tortoise is in the "Open" and temperature is less than 36°C then the mcl_mm, body_condition_score, nares_appearance, nares_discharge, and ticks should not be null.*
- Sort descending on mcl_mm and look for unrealistically large values (check with crews for values over 320.)
- Sex should be unknown for mcl_mm <180.
- Check for New tag number format. Should be "FWxxxx". Should not be "FW-xxxx".
- Sort on tort_not_handled, tort_not_handled_other, existing_tag, tag_attached, MCL, sex, and BCS fields and for null entries of the latter 4 fields, confirm that a reasonable explanation exists in the tort_not_handled or existing_tag fields.
    - If tag_attached is "no" then either existing_tag is "yes", or tort_not_handled is not null.
    - tag_attached, MCL, sex, BCS, nares_appearance, nares_discharge are not null
    - if tag_attached, MCL, sex, BCS, nares_appearance, or nares_discharge are Unknown, tort_not_handled or tort_not_handled_other are not null

  Error description: inconsistency between conditions and data collection on tortoise

G0_Start Table

- *Check for start_time or end_time null values (1:00AM). Sort ascending and descending.*

G0_Obs Table

- The burned field should be null for all sites.
- Sort by neighboring fields visible, location, burrow_visibility, tortoise_in_burrow_visiblity, tortoise_visibility. If tortoise location is burrow, burrow_visibility and tortoise_in_burrow_visibility should not be null. For other locations the tortoise_visibility field should not be null. Be alert for any indication that the tortoise was visible when first observed, but fled to a burrow. Frequently, the observer will collect data consistent with both locations instead of sticking the original setting.
- If more than one transmittered tortoise is in a burrow, the visibility of the inner tortoise(s) should reflect that they were blocked by the foremost one. There should not be 2 high visibility tortoises in one burrow, for instance.
- If visible=No and location is not burrow, it will usually be vegetation. Confirm that tortoise_visibility= Not Visible
- *If visible=No, behavior should be Unknown or rarely Digging or Moving.* If visible=Yes, pay attention to any records for which behavior is Unknown. Did the crew explain this in the comment? The comment should indicate the tortoise was responding to the observer before it was detected.

USFWS will perform interim assessments of each Survey organization's populated Correction database. This is a non-automated review of the data and is intended to provide timely feedback to reduce errors in future weeks of data collection. Because survey organizations submit appended correction databases each week, there is also opportunity to remedy errors identified by USFWS in the interim assessments. The interim assessments will focus on identification of non-script errors and inconsistencies (see above). It will also summarize start-, end-, and total time on transect, as well as changes in the shape of the detection curve over the field season.

Final processing steps: data backup and delivery

Each data specialist will work with a single Collection and a single Correction database. After all training is completed, QA/QC I is finalized on that database and the Collection and Correction databases are delivered. Both the training and field season databases have the same forms, so there is no difference in the Nexus phone version of the database nor in the Correction version of the database.

The Correction Database appends only new records each time the Collection Database is imported. The steps below ensure that 1) each iteration of the Collection and Correction databases are saved in a separate location from the active versions, 2) all saved databases are identified by the last date of information they contain (associated Collection and Correction databases will have the same date information), 3) the current Collection database to be uploaded always has the same name for recognition by the scripts in the Correction database.

The following steps describe the process of finalizing each updated version of a database after the data have been downloaded from Nexus phones and have gone through QA/QC procedures. After these steps, you are ready to repeat the process again with new data!

1. **Backup and deliver the Collection database**
   The Collection database should be backed up after any new data have been imported (Synced). Create a folder on external storage and rename it to "FORMS32K_YYYYMMDD ". Copy the "FORMS32K.accdb" file and manual photos folder ("manual_photos") from the "C:\Program Files\Forms3" folder on the machine used for syncing Nexus phones onto the new folder created on the external storage device and then rename the accdb file to "FORMS32K_ YYYYMMDD.accdb". Create a zip file from "FORMS32K_ YYYYMMDD "folder on external drive and rename the zip to "FORMS32K_ YYYYMMDD.zip". The date should correspond to the last instance of collected data. Do not rename the source file found in "C:\Program Files\Forms3". The zip version of the Collection database is delivered via Google Drive.

2. **Backup and deliver the Correction Database**
   The Correction databases should be backed up after each QA/QC session consisting of importing new data from the Collection database, running QA/QC checks and correcting all the errors found in the Errors table. To back up the database, copy the "Import_Correction.mde" file onto the external storage device and then rename the copy "Import_Correction_ YYYYMMDD.MDE". Create a zip file from this renamed MDE file, store it in the same location and rename it to "Import_Correction_ YYYYMMDD.zip". The date used for naming should match the date of the corresponding "FORMS…..mde" file. Do not rename the source file found in "C:\PendragonForms\"". The zip version of the Correction database should be delivered via Google Drive.

Note: All dates in file names should correspond to the date of the last instance of data collected in the "FORMS…accdb" file. For each set of collection data, two files should be sent to Topoworks and USFWS. Example: The FORMS32K.accdb file is appended to after transects are walked on 05/01/2020. When the QAQC session is complete on 05/04/2020 (checks have been run and errors corrected), the following two files would be sent to Topoworks and USFWS. They are both dated with 05/01/2020 even though they are not created on that date because the date represents the date of the data collected, not the date that the QAQC processes were completed.

- FORMS32K_ 20200501.zip
- Import_Correction_ 20200501.zip

At the time they deliver the final Preseason databases, the data specialist should designate the final Correction version using the naming convention above, suffixed by "_FinalTraining". Although other versions with this date may exist (covering the same time period), this version may have undergone subsequent editing and is the final submission by the survey team. If the time period covering all Preseason data all includes some field season data for one of the teams, those field season records should be removed from the final Preseason Correction database before it is delivered. Similarly, the final Field Season Correction database should not include any Preseason records.

In summary, do not remove any data from the single final Collection database. Two final Correction databases will be delivered; one containing only Preseason data and the other containing only Season data (Import_Correction_YYYYMMDD_FinalSeason.zip to distinguish it from any earlier versions of the same data records and to designate the final submission).

3. Deliver Paper Datasheets

Each week, QA/QC specialists send scanned versions of that week's paper datasheets to USFWS. The DPI Rate should be 150 with the setting of halftone black and white to pick up lightly hand written images. Copies of these scanned versions are kept by the QA/QC specialist to add necessary annotations when data errors and inconsistencies are addressed. The original paper datasheets themselves should be sent to USFWS as part of the final delivery of QAQC I products. Until final delivery of QAQC I products, electronic edits can be made to the scanned datasheets that were retained for QA/QC I. Datasheets that have been annotated after the initial scanned version should have the same name as the original so that the later version completely replaces any other version that has been shared with USFWS. Final data delivery should also include upload of any final edits to scanned data sheets. Digital scanned datasheet filenames and folder organization should follow the guidelines below.

Training Data: Training scanned datasheets should be organized into the following folders: 'TrainingLines_datasheets', and 'PracticeTransects_datasheets' Note that all G_0, regardless of when it was collected, will be packaged with the Monitoring Season data, not with Practice data..

TrainingLines_datasheets: This folder should contain scanned datasheet PDF files for the practice training lines. All sheets associated with a single date for a team in a specific trial should be grouped into one PDF file. Filenames should be composed of the Team number, Trial number, and Date (examples: Team21_Trial1_20200325.pdf, Team6_Trial2_20200402.pdf).

PracticeTransects_datasheets: This folder should contain scanned datasheet PDF files for the practice transects at the Large Scale Translocation Site (LSTS). All sheets associated with a single practice transect for a team should be grouped into one PDF file. Filenames should be composed of the Transect number, Stratum (LSTS), and Year (examples: 10_LSTS_2020.pdf, 14.1_LSTS_2020.pdf). For practice transects there are often duplicate walks of the same transect. Duplicate transects should be numbered with a concatenation of the main transect

number plus ".9". If more than one duplicate is walked, concatenate ".99". The scanned datasheet PDF files should contain the .9, .99, etc. suffixes.

Transect Data: Transect scanned datasheets should be organized into a folder named "Transects_datasheets". This folder should contain subfolders for each stratum abbreviation. Each subfolder should contain scanned datasheet PDF files for the transects walked in that stratum. All sheets associated with a single transect for a team should be grouped into one PDF file. Filenames should be composed of the Transect number, Stratum, and Year (examples for the 'AG' subfolder: 119_AG_2020.pdf, 1860_AG_2020.pdf; examples for the 'OR' subfolder: 188_OR_2020.pdf, 2399_OR_2020.pdf).

G0 Data: All G0 scanned datasheets should be organized into a folder named "G0_datasheets," irrespective of whether the data were collected during the training or field season portion of the project. This folder should contain subfolders for each G0 site abbreviation. Each subfolder should contain scanned datasheet PDF files for the G0 iterations in that G0 site. All sheets associated with a single G0 iteration for an observer should be grouped into one PDF file. Filenames should be composed of the G0 site and the Date of the iteration. If more than one observer performed iterations on the same day in the same G0 site, the files should be suffixed with the initials of the observer. (examples for the "CK" subfolder: CK_20200412_PA.pdf, CK_20200412_KH.pdf, CK_20200423.pdf; examples for the "SC" subfolder: SC_20200405_PA.pdf, SC_20200425_KS.pdf, SC_20200416.pdf). G0 site abbreviations for 2020 are as follows:

G0_site	G0_site_desc
CK	Chuckwalla
GB	Gold Butte
HW	Halfway Wash
IV	Ivanpah
OR	Ord-Rodman
RM	River Mountains
SC	Superior-Cronese

4. Deliver Extracted Photos

Each week, QA/QC specialists extract photos from the Collection database. These are sent on a weekly basis to USFWS and as a final complete set of photos to Topoworks as one of the QAQC 1 products.

Extracted photos should be named correctly (see below) and packaged in this folder structure:

a. G0 photos
    i. (G0_Start photos will all be together under the "G0 photos" root)
    ii. G0_Carcass_YY
b. Transect photos
    i. OppCarcObs_YY
    ii. OppLiveObs_YY
    iii. TranCarcObs_YY
    iv. TranLiveObs_YY
    v. Waypoints_YY
        1. xminus1
        2. xplus1

Each week photos will be extracted, named using an automated procedure, and stored in an associated folder. Photos should be in georeferenced format. Any photos taken on a transect but not

as part of the database must also be renamed using these conventions, and this name must be added to the paper datasheets and in the electronic database for the transect. The following naming conventions should be used for images.

Table Name	Field name	Naming convention	Example
Waypoints	photo_to_wp_xminus1	Tranxxx.x_st_wpww_to _minus1_yyyy	Tran234.0_wp03_to _minus1_2020.jpg
Waypoints	photo_to_wp_xplus1	Tranxxx.x__st_wpww_to _plus1_yyyy	Tran234.1_wp21_to _plus1_2020.jpg
TranLiveObs	photo_tort1	Tranxxx.x__st_TLzz_1_yyyy	Tran012.0_TL02_1_2020.jpg
TranLiveObs	photo_tort2	Tranxxx.x_st_TLzz_2_yyyy	Tran012.0_TL02_2_2020.jpg
TranCarcObs	photo_carc	Tranxxx.x_st_TCzz_yyyy	Tran245.0_TC12_2020.jpg
OppLiveObs	photo_tort	Tranxxx.x_st_OLzz_yyyy	Tran967.0_OL05_2020.jpg
OppCarcObs	photo_carc	Tranxxx.x_st_OCzz_yyyy	Tran009.0_OC01_2020.jpg
G0_Start	photo1	G0_site_yyyymmdd_FL_photo1	G0_SC_20200310_PF_photo1.jpg
G0_Start	photo2	G0_site_ yyyymmdd _FL_photo2	G0_GB_20200421_ZS_photo2.jpg
G0_Start	photo3	G0_site_ yyyymmdd _FL_photo3	G0_IV_20200323_DK_photo3.jpg
G0_Start	photo4	G0_site_ yyyymmdd _FL_photo4	G0_CK_20200513_AD_photo4.jpg
G0_Start	photo_carc	G0_site_ yyyymmdd _FL_photo_carc	G0_SC_20200513 _AD_photo_carc.jpg

xxx.x: transect number
st: stratum abbreviation
ww: waypoint number for current photo
zz: observation number
FL: first and last initials for the observer (as named in the database drop-down lists)
yyyy: 4-digit year; mm: 2-digit month; dd: 2-digit day

Phase I QA/QC for Preseason databases will be completed and delivered with scanned paper datasheets and photos to USFWS and Topoworks by **20 April 2020**.

Phase I QA/QC for Season databases will be completed and delivered with scanned paper datasheets and photos to USFWS and Topoworks within 10 days of the end of data collection or by **5 June 2020**. Original paper datasheets should be delivered to USFWS by this date as well.

Phase I products will be delivered by USFWS to Topoworks in two phases. The first two tables will be delivered after Preseason data have been collected and by **20 April 2020**. The latter 2 tables will be delivered at the end of the field season and by **5 June 2020.** In addition, USFWS will provide start and end dates for Season and Preseason work for each survey group separately.

- TortoiseLookUp_20 table for training products with these fields, current for 2020. Any updates to the final two fields based on the 2020 field season should be incorporated in these products for use in 2021
  - tortoise_ID (integer)
  - size (text)
  - training_line_color (text)
  - transect (text)
  - expected_distance (double)
  - location (text)
  - length_fromNEpoles (double)

- Train_Teams table populated by writing a query based on Train_Tran to create a single record for each team and trial. This table will have been created by USFWS in the process of reviewing training line data.
  - group_ (text 50 characters)
  - trial_number (integer)
  - team_num (integer)
  - observer1 (text 50 characters)
  - observer2 (text 50 characters)
- PlannedTransects table with these fields:
  - stratum (text)
  - tran_num (integer)
  - sel_20 (integer)
  - type_20 (text)
  - substat_obs_20 (text)
  - walked_20 (yes/no)
  - reason_replaced_20 (text)
  - substratum (text)
  - walk_desc_target (text)
- A G0_Siteinfo table with these fields:
  - G0_site (for the current year)
  - G0_site_desc (for the current year)
  - G0_group (for the current year)
  - covered_strata (for the current year)
  - group_ (for the current year)
  - minX (from this DMP page 43)
  - minY (from this DMP page 43)
  - maxX (from this DMP page 43)
  - maxY (from this DMP page 43)

Data Management Checklist (developed for working use based on above procedures)

- ☐ Accept data from crews
  - Paper datasheets: legibility, initials, names, blanks, standard/non-standard, interrupts, mcl_greater_180
  - Download/synchronize electronic data from Nexus phone
  - Move and rename manual photos from Nexus phone onto PC
- ☐ Backup the Collection (Pendragon) database and photos folder
- ☐ Import the Collection (Pendragon) database into the Correction (QAQC I) database
- ☐ Import manual photos into Correction database
- ☐ Perform automated QAQC checks
  - Execute QAQC scripts
  - When possible, correct the record in the corresponding table
  - Update the error record
    - Old_value, new_value, resolution, resolver, error_status
  - Update the paper datasheet, if needed
- ☐ Perform non-automated QAQC checks
  - Visually examine tables for missing or inconsistent values
  - Create error record, if necessary
  - When possible, correct the record in the corresponding table
  - Update the error record
    - Old_value, new_value, resolution, resolver, error_status
  - Update the paper datasheet, if needed
  - Implement any remaining corrections from USFWS interim assessment
- ☐ Backup the Correction (QAQC I) database after each QAQC session
- ☐ Scan paper datasheets (follow naming conventions in Data Management Plan)
- ☐ Use Transect Tracking database to assign the date that transects were walked
- ☐ Deliver data weekly
  - Upload Collection, Correction (QAQC I), and scanned datasheets
  - Prepare hardcopy paper datasheets for delivery to USFWS by 8 June 2020
  - Inform USFWS before any delay in weekly data delivery
- ☐ Review and respond to USFWS interim assessments
  - Implement all corrections for next delivery
  - Ensure crews have access to read assessments
  - Discuss recurring issues with crews

Reference documents:

- Data Management Plan (DMP)
- QAQC Specialist Training Materials
- Monitoring Handbook (for crew protocols only)

Phase II: Data Integration and Field Completion

TW is responsible for compiling electronic databases and scanned versions of paper data submitted by survey organizations into a combined, well-organized set of Preseason and Season databases, photos, and data sheets. Includes review of surveyor products, combination, consolidation, formatting, generating additional data fields, and delivering *Phase II products* for data finalization. The following section describes not only standards as above, but also the internal workflow which is implemented by the responsible organization.

On receipt of Phase I products:

- Verify with survey vendors and USFWS that final products have been delivered by survey vendors on or before 7 June.
- Verify that the set of scanned paper datasheet is complete
    - Compile and organize scanned datasheet files according to the specifications for the Data Collection and Correction phase.
        - Preseason datasheets should be compiled in 2 different folders named PracticeTransects_datasheets and TrainingLine_datasheets
        - Transects_datasheets should be compiled into subfolders named with stratum abbreviations
        - G0_datasheets should be compiled into subfolders named with site abbreviations
    - Verify that scanned datasheet file names match the conventions specified in the Data Collection and Correction phase.
- Verify that the set of extracted photos is complete
    - Compile and organize extracted photos according to the specifications for the Data Collection phase.
        - G0_Start photos should be compiled into a folder named “G0 photos”
        - Transect photos should be compiled into the following folders and subfolders:
            - OppCarcObs_YY
            - OppLiveObs_YY
            - TranCarcObs_YY
            - TranLiveObs_YY
            - Waypoints_YY
                - xminus1
                - xplus1
    - Verify that photo file names match the conventions specified in the Data Collection and Correction phase.
- Confer on a weekly basis with USFWS regarding progress toward data delivery deadlines for Phase 2 data handoff

Creation of Working Databases

1. Receive final QAQC I (Correction) databases.
    a. Preseason Data with populated tables related to:
        i. Training Lines
        ii. Practice Transects
    b. Monitoring Season with populated tables related to:
        i. Transects
        ii. G_o
2. Create working folders to hold
    a. the version of each of the above QAQC I Correction databases that are final and used to initiate Phase 2, and
    b. the corresponding version of the Collection databases (Pendragon format, names starting with Forms32 incorporating the same date as the associated Correction databases).

    These are 2 of the final Phase 2 products.

3. Starting with QAQC I correction databases, package appropriate tables into 4 separate databases, integrating records from all survey organizations into the same tables:
   a. Preseason Training lines
   b. Preseason Practice transects
   c. Season Transects
   d. Season G0
4. For each database in step 3, import *only* records from the associated Error tables with a status of "exception allowed", "unresolved" or "blank" (the latter is not anticipated but must be addressed if present). There should be no records with "resolved" errors.
5. Verify no records were lost during step 3, no cell values were truncated and no field names were truncated.

Steps for Processing the Transects Database (Season and Preseason)

Initial processing steps

1. Verify that the field names and cell format in each table match the tables of formats below under "Formats to Apply".
2. Create a Teams Lookup table. Using this lookup table, standardize observer names and case throughout all tables. Names are recorded as "First Last" with leading capitals (unless the formal name does not start with a capital). Remove any apostrophes. Any changes should be documented in the Errors table.
3. In the Transects table add a datasheet field and populate it with the path to the relative location and name of the associated scanned datasheet.
4. For all fields in all tables, remove null and -99 values and replace them using an Update query with true <Null> values, not zero length strings.
5. For all fields in all tables, replace Y and N with Yes and No, and U with Unknown.
6. Verify photos were extracted properly in Phase 1 and the file name is correct, then delete the photo Long Binary fields from the tables.
7. Add elev_m, final_easting, final_northing, final_zone, z12_easting, z12_northing, gps_grab_success fields to the obs and waypoints tables. Populate the fields as described below.
8. Populate gps_grab_success with values of Yes or No describing whether the GPS grab in the field was successful (if the team recorded manual coordinates or if the coordinates are changed during Phase I or Phase II, then gps_grab_success should be No)
9. Populate the final easting and northing coordinate fields.
   a. For points collected in zone 11: if manual coordinates were not collected, use the gps easting and northing fields to populate final_easting and final_northing. If manual coordinates were collected, use the manual easting and northing to populate final_easting and final_northing.
   b. For points collected in zone 12: if manual coordinates were not collected, use the gps easting and northing fields to populate z12_easting and z12_northing. If manual coordinates were collected, use the manual easting and northing fields to populate z12_easting and z12_northing.
      i. Note: Only records collected in zone 12 should have values in the z12 fields.
      ii. Note: At this point, the final_easting and final_northing fields will only contain values for points collected in zone 11. Later, during GIS processing the zone 11 coordinates for points collected in zone 12 will be populated.
   c. Verify and populate the final_zone field using the appropriate gps or manual zone.
10. Make the access database size more manageable by compressing the database after the photos have been removed. (Selecting "Compact on Close" in Access Options will ensure the database is kept at a reasonable size)

Address duplicate records

11. Perform duplicate record checks for the following:
    a. Duplicate transect numbers within the same stratum.
    b. Duplicate waypoint numbers on the same transect.

c. Duplicate transect observations.

Address orphan records

12. Check for transects, waypoints, and observations that do not have related records in the other tables. Correct, where possible. Document any orphans and corrections in the Errors table.

Documenting errors and corrections in the Errors table

13. If duplicates, orphans, missing records, or other anomalies are found:
    a. Review the error table to see if the error was already documented in Phase I. If so, do not add a new error record. If not, add a new error record. Populate the key fields. If the error cannot be resolved, Error Status should be exception allowed. If the error can be resolved, correct the error and mark Error Status as resolved (Only error_status = "resolved" from Phase 2 should be in the database; no resolved errors from Phase 1 should be present.),
    b. To correct errors: Identify and edit the appropriate record. Although a script may identify an inconsistency in an observation record, the edit may be required to the associated parent record. Because of this, also verify that the table and prime_key in the Errors table match the appropriate record that was edited.
    c. Verify the "old_value" field and update the "new_value" field with what was replaced in the database.
    d. The "resolution" field must contain a description on what was done. There is standardized wording for these resolutions (see Phase I: Procedures for addressing errors) to facilitate review of these tables and identification of duplicate errors.
    e. If the error cannot be fixed by comparing the entries on the paper datasheet or other techniques; Error Status should be marked exception allowed. The "resolution" field should be filled out with a reason why the record could not be updated.

GIS processing

14. For records collected in UTM Zone 12, convert coordinates to UTM Zone 11, WGS84 and populate final_easting and final_northing fields. All records should have zone 11 coordinate values in the final_easting and final_northing fields, while only records collected in zone 12 should have values in the z12_easting and z12_northing fields
15. Evaluate sources for an appropriate Digital Elevation Model (DEM) dataset. Download, process, and combine DEM datasets for a complete DEM encompassing the sample area.
16. For all observation and waypoint records, populate the elev_m field by extracting values from the DEM.

G0 Database

Initial processing steps

1. Check for null prime keys, resolve issues (delete; assign unique prime key, etc.)
2. Verify that the field names and cell formats in each table match the tables of formats below under "Formats to Apply".
3. Create a Teams Lookup table. Using this lookup table, standardize observer names and case throughout all tables. Names are recorded as "First Last" with leading capitals (unless the formal name does not start with a capital). Remove any apostrophes. Any changes should be documented in the Errors table.
4. Add a text field to the G0_Start table called "datasheet" and populate it with the path to the relative location and name of the associated scanned datasheet..
5. For all fields in all tables, remove "null" and "-99" values and replace them using an Update query with true <Null> values not zero length strings
6. For all fields in all tables, replace Y and N with Yes and No, and U with Unknown.
7. Verify photos were extracted properly during Phase 1 and the file paths in the database are correct, then delete the photo Long Binary fields from the tables.

8. Add elev_m, final_easting, final_northing, final_zone, gps_grab_success fields to the Obs tables. Populate the fields as described below.
9. Populate gps_grab_success with values of Yes or No describing whether the GPS grab in the field was successful (if the team recorded manual coordinates or if the coordinates are changed during Phase I or Phase II, then gps_grab_success should be No)
10. Populate the final easting and northing coordinates.
    a. For points collected in zone 11: if manual coordinates were not collected, use the gps easting and northing fields to populate final_easting and final_northing. If manual coordinates were collected, use the manual easting and northing to populate final_easting and final_northing.
    b. If any G0 points were collected in zone 12, add z12_easting, z12_northing and: if manual coordinates were not collected, use the gps easting and northing fields to populate z12_easting and z12_northing. If manual coordinates were collected, use the manual easting and northing fields to populate z12_easting and z12_northing.
        i. Note: Only records collected in zone 12 should have values in the z12 fields.
        ii. Note: At this point, the final_easting and final_northing fields will only contain values for points collected in zone 11. Later, during GIS processing the zone 11 coordinates for points collected in zone 12 will be populated.
11. Verify and populate the final_zone field using the appropriate gps or manual zone.

12. Make the access database size more manageable by compressing the database after the photos have been removed.

Address duplicate records

13. Perform duplicate record checks for the following:
    Duplicate G0 Start records and observations.

Address orphan records

14. Check for G0 start records and observations that do not have related records in the other tables. Correct, where possible. Document any orphans and corrections in the Errors table.

Documenting errors and corrections in the Errors table: see description in previous section

GIS processing

1. If any G0 records collected are collected in UTM Zone 12, convert coordinates to UTM Zone 11, WGS84 and populate final_easting and final_northing fields. All records should have zone 11 coordinate values in the final_easting and final_northing fields, while only records collected in zone 12 should have values in the z12_easting and z12_northing fields.
2.

For all observation records, populate the elev_m field by extracting values from the DEM. **Formats to Apply**

Ensure that fields have the appropriate storage format and decimal places. All data have Collection and Correction database formats, as indicated in the data dictionaries. The following fields are standardized to the indicated formats and decimal places in preparation for the final databases. The complete data dictionaries for those data are part of metadata for the final product. Fields not listed below can be whatever works best for the Data Integration phase. For photo file naming conventions, see “Photo Collection System” on page 7 in the Design Phase section of this document.

Training Line tables:

Table	Field	Format	Dec	Notes
Train_Tran	TimeStamp_	Date		
	trial_number	Short int	0	
	team_num	Short int	0	
	transect_seg_num	Short int	0	
	calc_transect_seg_ num	Short int	0	
	tran_bearing	Short int	0	
	training_date	Date		
	training_start_time	Date		
	training_end_time	Date		
	total_time	Double	2	
Train_Obs,	TimeStamp_	Date		
	trial_number	Short int	0	
	team_num	Short int	0	
	training_date	Date		
	observation_time	Date		
	tortoise_id	Short int	0	
	local_bearing	Short int	0	
	azimuth	Short int	0	
	radial_distance_m	Double	1	one decimal place
	bearing_radians	Double		
	azimuth_radians	Double		
	perp_distance_m	Double	1	full calculated value

G_0 tables:

Table	Field	Format	Dec	Notes
G0_Start,	date_	date		
	start_time	date		
	end_time	date		

Table	Field	Format	Dec	Notes
G0_Obs	date_	date		
	time_	date		
	final_easting	Long int	0	See above
	final_northing	Long int	0	See above
	gps_easting	Double		
	gps_northing	Double		
	manual _easting	Long int	0	
	manual _northing	Long int	0	
	elev_m	Short int	0	
	mcl_mm	Short int	0	

Transect tables (Preseason and Season):

Table	Field	Format	Dec	Notes
Transects	tran_num	Double	1	one decimal place
	TimeStamp_	Date		
	date_	Date		
	do_time	Date		
	tran_start_time	Date		
	tran_end_time	Date		
	ret_do_time	Date		
	team_num	Short int	0	
	trn_length_m	Long int	0	
Waypoints, TranCarcObs, TranLiveObs, OppCarcObs, OppLiveObs,	tran_num	Double	1	one decimal place
	wp_num	Short int	0	
	detection_number	Short int	0	opp_live/_carc, tran_live/_carc
	team_num	Short int	0	
	last_wp	Short int	0	
	TimeStamp_	Date		
	time_	Date		
	final_easting	Long int	0	Rounded integer UTM zone 11 values for all records
	final_northing	Long int	0	Rounded integer UTM zone 11 values for all records
	z12_easting	Long int	0	Rounded integer UTM zone 12 values for records collected in zone 12, others should be null
	z12_northing	Long int	0	Rounded integer UTM zone 12 values for records collected in zone 12, others should be null
	gps_easting	Double		
	gps_northing	Double		
	manual _easting	Long int	0	
	manual _northing	Long int	0	
	elev_m	Short int	0	
	tran_bearing	Short int	0	First, include values from tran_bearing_other field
	local_bearing	Short int	0	
	azimuth	Short int	0	
	radial_distance_m	Double	1	one decimal place
	perp_distance_m	Double		full calculated value
	dist_to_burrow_m	Double	1	one decimal place if measured in the field, integer if estimated
	temp_c	Short int	0	
	mcl_mm	Short int	0	

Non-automated checks for errors

Overall Database Checks

- Visually inspect database in design view to ensure field settings comply with specifications in the QAQC Plan.
- *Scan all tables and fields for visually unusual patterns or breaks in patterns that may reflect unexpected inconsistencies in the data.*
- *Sort ascending and descending number, date and time fields from all tables looking for values that are out of given ranges.*

Procedures for addressing errors

See Phase I version of this section on page 14.

Final Data Submission

Deliver the following Phase II products for data finalization:

1. Integrated and completed MS Access databases. Preseason submissions will consist of two Phase 2 databases (Training Lines, Practice Transects), Monitoring Season will consist of two Phase 2 databases (Transects, G0) as shown below with associated tables.

Collection Effort	Database	Tables
Preseason	Training Lines (polystyrene models)	Train_Tran Train_Obs Train_Teams Errors
	Practice Transects	Transects Waypoints TranLiveObs TranCarcObs OppLiveObs OppCarcObs Errors Site_Info Teams
Monitoring Season	Transects	Transects Waypoints TranLiveObs TranCarcObs OppLiveObs OppCarcObs Errors Site_Info Teams
	G_0	G0_Start G0_Obs Errors G0_Site_Info

2. Final QAQC I Collection Database delivered at the end of Phase I.
3. Final QAQC I Correction Databases delivered at the end of Phase I.
4. Extracted photos.
5. Scanned datasheets
6. Documentation of Phase II processes including processing steps involved during Phase II data integration, field completion, and QA/QC. Documentation should describe where steps

followed or deviated from the Data Management Plan and should describe major errors detected.

Phase II Data Integration, Field Completion, and QA/QC will be completed by Topoworks and delivered to USFWS by **20 July 2020**.

Phase III: Data Finalization

TopoWorks is responsible for creating final Season products, which includes any additional processing steps beyond the Data Integration phase for standardization across all databases and supporting data sheets and photos, creation of FGDC-compliant metadata, independent review of integrated data products from previous phases, generating the final set of data fields, performing third level QA/QC checks and corrections, identifying and resolving data integrity issues with USFWS, and finalizing spatial and non-spatial electronic data according to USFWS specifications.

Initial processing steps

- Add a unique identifier for QA/QC.
- Categorize error records to aid in error summary reporting.
- Create lookup tables.
- Rename database fields for consistency across all tables, if necessary.
- Delete fields that are no longer necessary (UnitID, calculated, exported).
- Standardize interrupted transect records making sure that appropriate times, obstacles, and comments from all parts of a single transect will be included in the final products.
- In the transects database, add and populate fields for waypoints describing the length of the segment between the last waypoint and the current waypoint, along with the total length of the transect at that waypoint. Add a field for each of the transect observations tables for transect length at the last waypoint.
- Update the transect datasheet field so that each datasheet filename is prefaced with its final relative location to aid in hyperlinking to the datasheets. Verify that the electronic datasheet exists and matches the filename in the datasheet field.
- Update the photo fields so that each photo filename is prefaced with its relative location to aid in hyperlinking to the photos. Verify that the electronic photo exists and matches the filename in the photo field.
- Make sure that any xxx_other field values (e.g. tran_bearing_other, existing_tag_color_other, tort_not_handled_other) are incorporated into their main field (e.g. tran_bearing, existing_tag_color, tort_not_handled). For observations where new tags were attached, populate color field with the 2020 tag color, then combine the other_color field with the color field.
- For non-standard transects, replace Null values for terr_obstacles, subs_obstacles, and other_obstacles with 'None' for more clarity.
- Standardize data values for terr_obstacles and subs_obstacles to remove extra semicolons (;;).
- Standardize interrupted transects (transects collected with multiple parts where teams stopped the transect, moved a distance, and restarted it) by associating all waypoints and observations with a single main transect record, ensuring that waypoints have consecutive numbers, ensuring that last_wp values for observations are correct, and ensuring that the main transect record represents the correct drop off, start, end, and return times for the entire transect. Add summary information for transects including: transect length, total transect live and carcass observations, total opportunistic live and carcass observations.
- Integrate each survey vendor's applicable transect tracking data fields (walked_xx, replaced_xx, reason_replaced_xx, replaced_by, replacement_for, and date_walked) into the PlannedTransects_all master table.
- Assign walk_desc_xx, substrat_obs_xx, substrat_rpt_xx, and reduction_obs_xx to walked Transects and Planned Transects (and submit for approval and updates from USFWS).
- Import and update StrataBnd spatial data from previous years.
- Ensure that fields have the appropriate storage format and decimal places as follows:

Training Line tables:

Table	Field	Format	Dec	Notes
Train_Tran	TimeStamp_	Date		
	trial_number	Short int	0	
	team_num	Short int	0	
	transect_seg_num	Short int	0	
	calc_transect_seg_num	Short int	0	
	tran_bearing	Short int	0	
	training_date	Date		
	training_start_time	Date		
	training_end_time	Date		
	total_time	Double	2	
Train_Obs,	TimeStamp_	Date		
	trial_number	Short int	0	
	team_num	Short int	0	
	training_date	Date		
	tran_bearing	Short int	0	
	observation_time	Date		
	tortoise_id	Short int	0	
	local_bearing	Short int	0	
	azimuth	Short int	0	
	radial_distance_m	Double	1	one decimal place
	bearing_radians	Double		
	azimuth_radians	Double		
	perp_distance_m	Double	1	one decimal place

G_0 tables:

Table	Field	Format	Dec	Notes
G0_Start, G0_Obs, G0_OppLiveObs	date_	date		
	start_time	date		
	end_time	date		
	time_	date		
	final_easting	Long int	0	See above
	final_northing	Long int	0	See above
	gps_easting	Double		
	gps_northing	Double		
	manual _easting	Long int	0	
	manual _northing	Long int	0	
	elev_m	Short int	0	
	mcl_mm	Short int	0	

Transect tables:

Table	Field	Format	Dec	Notes
Transects	tran_num	Double	1	one decimal place
	TimeStamp_	Date		
	date_	Date		
	do_time	Date		
	tran_start_time	Date		
	tran_end_time	Date		
	ret_do_time	Date		
	team_num	Short int	0	
	trn_length_m	Long int	0	
Waypoints, TranCarcObs, TranLiveObs, OppCarcObs, OppLiveObs,	tran_num	Double	1	one decimal place
	wp_num	Short int	0	
	detection_number	Short int	0	
	team_num	Short int	0	
	last_wp	Short int	0	
	TimeStamp_	Date		
	time_	Date		
	final_easting	Long int	0	Rounded integer UTM zone 11 values for all records
	final_northing	Long int	0	Rounded integer UTM zone 11 values for all records
	z12_easting	Long int	0	Rounded integer UTM zone 12 values for records collected in zone 12, others should be null
	z12_northing	Long int	0	Rounded integer UTM zone 12 values for records collected in zone 12, others should be null
	gps_easting	Double		
	gps_northing	Double		
	manual _easting	Long int	0	
	manual _northing	Long int	0	
	elev_m	Short int	0	
	tran_bearing	Short int	0	
	local_bearing	Short int	0	
	azimuth	Short int	0	
	radial_distance_m	Double	1	one decimal place
	perp_distance_m	Double	2	two decimal places
	temp_c	Short int	0	
	mcl_mm	Short int	0	

Specific QA/QC checks for errors and inconsistencies

The following describes the specific checks that will be performed during Final QA/QC.

Database relationship checks

- Check for transects with less than 10 or more than 30 waypoints.
- *Check for transects, waypoints, and observations that do not have related records in the other tables.*
- *Check for transects.observer1 and transects.observer2 without a related observer in the observer lookup table.*
- Check for transects, waypoints, and observations without a related stratum in the strata boundary dataset.
- *Check for G_0 Start records, Obs, and OppLiveObs that do not have related records in other tables.*

- *Check for training observations that do not have related training transect records.*

Field domain checks

The following describes tables and fields for which records are checked against their logical domain. Specific logical domains for each field are located in *Appendix A: Database Dictionaries*. Error description should be: [field_name] is not within domain or [field_name] is not within domain [low_value] to [high_value]. (Substitute the field name and domain values, Example: do_time is not within domain 4:00AM to 10:00AM.) The field value before correction should be written to the 'old_value' field on the Errors table.

Domain	Type	Table and fields applied to
Stratum	Coded	Transects, Waypoints, OppCarcObs, OppLiveObs, TranCarcObs, TranLiveObs: stratum
Tran_num (short)	Range	Transects, Waypoints, OppCarcObs, OppLiveObs, TranCarcObs, TranLiveObs: tran_num
Date	Range	Transects: date_; Transects, Waypoints, OppCarcObs, OppLiveObs, TranCarcObs, TranLiveObs: TimeStamp_; G0_Start, G0_Obs: date_
Group	Coded	Transects, G0_Start: group_; Train_Tran: group
Team_num (dbl)	Range	Transects: team_num; Train_Tran: team_number
Drop_Start_time	Range	Transects: do_time; Transects: tran_start_time; Train_Tran: training_start_time
End_Ret_time	Range	Transects: tran_end_time; Transects: ret_do_time; Train_Tran: training_end_time
Time	Range	Waypoints, OppCarcObs, OppLiveObs, TranCarcObs, TranLiveObs: time_; G0_Obs: time_; Train_Obs: observation_time
Observer_position	Coded	TranCarcObs, TranLiveObs: observer_position; Train_Obs: observer_position
Lead_Follow	Coded	Waypoints: lead; Waypoints: follow; TranLiveObs, TranCarcObs:observer
Wp_num (short)	Range	Waypoints: wp_num
Last_wp (short)	Range	TranCarcObs, TranLiveObs: last_wp
UTM_zone	Coded	Waypoints, OppCarcObs, OppLiveObs, TranCarcObs, TranLiveObs: zone_gps, zone_man ; G0_Obs: zone_gps, zone_man
Easting (long)	Range	Waypoints, OppCarcObs, OppLiveObs, TranCarcObs, TranLiveObs: final_easting; G0_Obs: final_easting
Northing (long)	Range	Waypoints, OppCarcObs, OppLiveObs, TranCarcObs, TranLiveObs: final_northing; G0_Obs: final_northing
Elevation (long)	Range	Waypoints, OppCarcObs, OppLiveObs, TranCarcObs, TranLiveObs: elev_m; G0_Obs: elev_m
Num_live_obs (short)	Range	OppLiveObs, TranLiveObs: detection_number
Num_carc_obs (short)	Range	OppCarcObs, TranCarcObs: detection_number

Domain	Type	Table and fields applied to
Bearing (short)	Range	TranCarcObs, TranLiveObs: tran_bearing, local_bearing; Train_Obs: local_bearing
Azimuth (short)	Range	TranCarcObs, TranLiveObs: azimuth; Train_Obs: azimuth
Radial_distance (dbl)	Range	TranCarcObs, TranLiveObs: radial_distance_m; Train_Obs: radial_dist
Perp_distance (dbl)	Range	TranCarcObs, TranLiveObs: perp_distance_m; Train_Obs: perp_distance_m
Yes_No	Coded	Waypoints, OppCarcObs, OppLiveObs, TranCarcObs, TranLiveObs: gps_grab_succes; OppLiveObs: new_tag_attached, tort_void; TranLiveObs: new_tag_attached, tort_void
Yes_No_Unknown	Coded	OppCarcObs, OppLiveObs, TranCarcObs, TranLiveObs: mcl_greater_180
Existing_tag_live	Coded	OppLiveObs, TranLiveObs: existing_tag
Existing_tag_carc	Coded	OppCarcObs, TranCarcObs:existing_tag
Carc_condition	Coded	OppCarcObs, TranCarcObs: carc_condition
Mcl_mm (short)	Range	OppCarcObs, OppLiveObs, TranCarcObs, TranLiveObs: mcl_mm
Sex	Coded	OppCarcObs, OppLiveObs, TranCarcObs, TranLiveObs:sex
G0_site	Code	G0_Start, G0_Obs: G0_site
Visibility	Code	TranLiveObs, OppLiveObs: burrow_visibility, tort_in_burrow_visibility, tort_visibility; G0_Obs: burrow_visibility, tort_in_burrow_visibility, tort_visibility
Location	Code	TranLiveObs, OppLiveObs: tort_location; G0_Obs: tort_location
Behavior	Code	G0_Obs: behavior
Visible	Code	G0_Obs: visible
Tort_num	Range	G0_Obs: tort_num
Training_Line_Color	Code	Train_Tran, Train_Obs: training_line_color
Training_Start_Post	Code	Train_Tran: start_post
Training_Seg_Num	Range	Train_Tran: transect_seg_num
Training_Transect	Code	Train_Tran, Train_Obs: transect
Training_Total_Time	Range	Train_Tran; total_time
Training_Tran_bearing	Code	Train_Tran: tran_bearing
Tortoise_Size	Code	Train_Obs: tortoise_size
Original_Observation	Code	Train_Obs: original_observation

Duplicate record checks

Table	Check
Transects	contains duplicate tran_num values
Transects	contains duplicate tran_prime_key values
Waypoints	contains duplicate tran_num and wp_num values
Waypoints	contains duplicate wp_prime_key values
OppCarcObs	contains duplicate tran_num and detection_number values
OppCarcObs	contains duplicate obs_key values
OppLiveObs	contains duplicate tran_num and detection_number values
OppLiveObs	contains duplicate obs_key values
TranCarcObs	contains duplicate tran_num and detection_number values
TranCarcObs	contains duplicate obs_key values
TranLiveObs	contains duplicate tran_num and detection_number values
TranLiveObs	contains duplicate obs_key values
G0_Start	contains duplicate G0_prime_key values
G0_Start	contains duplicate G0_site, date_, observer
G0_Obs	contains duplicate G0_obs_prime_key values
G0_Obs	contains duplicate G0_site, date_, time_, tort_num
Train_Tran	Contains duplicate transect, team_number, training_date, trial_number
Train_Obs	Contains duplicate tortoise_id, transect, team_number, training_date, trial_number

Multi-field attribute condition checks across multiple tables

- Check for waypoint or observation stratum values that do not match the associated transect stratum value.
- Check for waypoint or observation tran_num values that do not match the associated transect tran_num value.
- Check for transects, practice transects, or training transects where observer1 or observer2 and group do not match the observer and group in the observers lookup table.
- Check for transects, practice transects, or training transects where team_num, group, observer1 and observer2 do match the teams lookup table.
- Check for transect observations where the tran_prime_key and last_wp do not match an existing waypoint.
- *Check for waypoint or observations where the date from the TimeStamp field does not match the associated transect date (Transect) or G0_Obs where the date from the TimeStamp field does not match the associated Start date (G_0).* Do not edit the TimeStamp field, but check to see if there are errors in the date field. If there are not any errors, then the violation can remain as an exception allowed. The TimeStamp field should never be edited.
- *Check for waypoints or observations where time is before transect start_time or after transect end_time (Transect) or G0_Obs where time_ is before Start record start_time or after Start record end_time (G_0).*
- *Check for observations where time is before waypoint time for the last_wp or after time for next waypoint.*
- Check for observations where observer_position does not match the associated last waypoint lead or follow.
- *Check for waypoints 1 and 99 not within ± 30 minutes of transect do_time, tran_start_time, tran_end_time, and ret_do_time (respectively).*
- Check for G0_Obs where G0_site does not match G0_site in G0_Start.
- *Check for G0_Obs where date does not match date in G0_Start.*
- *Check for G_0 observations where time_ is not within ±30 minutes of the start and end times for the associated start record.*

- Check across all tables for null values in the following fields: TimeStamp, tran_num, date, team_num, do_time, tran_start_time, tran_end_time, ret_do_time, wp_num, final_easting, final_northing, time, elev_m, detection_number, last_wp, perp_distance_meters; G_0 - start_time, end_time, tort_num, Training -transect_seg_num, tran_bearing, training_date, training_start_time, training_end_time, total_time, tortoise_id, local_bearing, azimuth, radial_dist, perp_distance_m.
- Check across all tables for blank or null prime key values.
- Check across all tables for blank or null values in the following fields: stratum, group, observer1, observer2, carc_condition, carc_fate, sex, new_tag_attached, tort_void, observer, observer_position, G_0 - G0_site, burrow_visibility, tort_in_burrow_visibility, tort_visibility, tort_location, behavior, visible, in_burrow, tort_num, Training – training_line_color, start_post, transect, lead, follow, observer_name, observer_position, tortoise_size, original_observation.
- Check across all tables for blank values in comments field (need to be changed to null instead of blank).
- Check across waypoints and observations tables for gps_bluetooth is null but gps_grab_success is not No and gps_bluetooth is not null but gps_grab_success is not Yes.
- Check across waypoints and observations tables for gps_grab failed and easting_man, northing_man, or zone_man are blank.
- *Check for training observations where observer_position does not match the lead or follow from the training transect.*
- *Check for training observation local_bearing not within 40 degrees of tran_bearing.*
- *Check for training observation_time not between training_start_time and training_end_time.*

Multi-field transect attribute condition checks

- date_ does not match date from TimeStamp_
- group_ is Kiva and date is before or after logical domain
- group_ is Kiva and team_num is not within logical domain
- *Observer1 and Observer2 are the same*
- *do_time is after tran_start_time*
- *tran_start_time is after tran_end_time*
- *tran_end_time is after ret_do_time*
- Shape_Length is greater than 16,000

Multi-field Waypoint attribute condition checks

- *lead and follow are the same*
- *wp_num is 0, 99, or 100 but lead and follow are not null*
- segment_length is 0 and wp_num is not 1

Multi-field OppCarcObs attribute condition checks

- *existing_tag is yes, but existing_tag_num or tag_color is null*
- *existing_tag is no, but existing_tag_num and tag_color are not null*
- *carc_condition is intact, but mcl_mm is Null*
- *mcl_mm is 180 or greater but mcl_greater_180 is not yes*
- *mcl_mm is less than 180 but mcl_greater_180 is not no*

Multi-field OppLiveObs attribute condition checks

- *new_tag_attached is Yes but new_tag_num is null*
- *new_tag_attached is No but new_tag_num is not null*
- *existing_tag is yes, but existing_tag_num or tag_color is null*
- *existing_tag is no, but existing_tag_num and tag_color are not null*
- *existing_tag is Yes, but new_tag_attached is not No*
- *existing_tag is no and new_tag_attached is null, should be Yes or No*
- *mcl_mm is 180 or greater but mcl_greater_180 is not yes*
- *mcl_mm is less than 180 but mcl_greater_180 is not no*

- in_burrow is no and mcl_mm is null
- in_burrow is yes, but tort_location or tort_visibility are not null
- in_burrow is no, but burrow_visibility and tort_in_burrow_visibility are not null

Multi-field TranCarcObs attribute condition checks

- *perp_distance_m is greater than radial_distance_m*
- perp_distance_m does not equal radial_distance_m * (Sin(bearing_radians - azimuth_radians))
- *existing_tag is yes, but existing_tag_num or tag_color is null*
- *existing_tag is no, but existing_tag_num and tag_color are not null*
- *carc_condition is intact, but mcl_mm is Null*
- *mcl_mm is 180 or greater but mcl_greater_180 is not yes*
- *mcl_mm is less than 180 but mcl_greater_180 is not no*

Multi-field TranLiveObs attribute condition checks

- *perp_distance_m is greater than radial_distance_m*
- perp_distance_m does not equal radial_distance_m * (Sin(bearing_radians - azimuth_radians))
- *tag_attached is blank or null*
- *new_tag_attached is Yes but new_tag_num is null*
- *new_tag_attached is No but new_tag_num is not null*
- *existing_tag is yes, but existing_tag_num or tag_color is null*
- *existing_tag is no, but existing_tag_num and tag_color are not null*
- *existing_tag is Yes, but new_tag_attached is not No*
- *existing_tag is no and new_tag_attached is null, should be Yes or No*
- *mcl_mm is 180 or greater but mcl_greater_180 is not yes*
- *mcl_mm is less than 180 but mcl_greater_180 is not no*
- in_burrow is no and mcl_mm is null
- in_burrow is yes, but tort_location or tort_visibility are not null
- in_burrow is no, but burrow_visibility and tort_in_burrow_visibility are not null

Multi-field G0_Start attribute condition checks

- group_ is Kiva and date is before or after logical domain
- *start_time is after end_time*

Multi-field G0_Obs attribute condition checks

- *Check to ensure that burned is null for all G0 observations at all G0_sites*
- in_burrow is no and mcl_mm is null
- in_burrow is yes, but tort_location or tort_visibility are not null
- in_burrow is no, but burrow_visibility and tort_in_burrow_visibility are not null

Spatial condition checks

- *Check for transect observations not within 50 meters of their related transect line.*
- Check for opportunistic observations not within 600 meters of their related transect line.
- Check for transects that do not intersect their related stratum.
- Check easting and northing values against the following tables of values, which include a 3km buffer. Zone 12 minimum and maximum values were derived by re-projecting polygons for the 2 affected strata into Zone 12, not by converting the Zone 11 coordinates. Values below are appropriate for QA/QC I, but should be narrowed to use a 1km buffer for subsequent checks.

○

Stratum	MinX	MaxX	MinY	MaxY	MinX_z12	MaxX_z12	MinY_z12	MaxY_z12
AG	622378	694425	3666698	3715744				
BD	745282	782980	4079099	4132136	209973	4079152	250125	4133069
CK	572347	712582	3649506	3754226				
CM	629956	736642	3774572	3861252				
CS	667312	706312	4020612	4118031				
EV	672500	720671	3924000	3979347				
FE	629743	708222	3843000	3908253				
FK	409848	478341	3835816	3925343				
GB	737029	793114	4003540	4090808	198812	4002257	256475	4090988
IV	590016	664952	3853419	3948899				
JT	576456	662769	3724323	3770752				
LSTS	636500	653400	3944400	3966500				
MM	686754	751710	4061910	4093649				
OR	495338	552170	3818999	3859768				
PV	672500	720671	3884130	3939000				
PT	586840	662051	3752194	3778035				
SC	456658	563646	3851097	3921485				

G0 Site	G_0 Site Description	Min_easting	Max_easting	Min_northing	Max_northing
CK	Chuckwalla	630000	634200	3710000	3715000
CS	Coyote Springs	679669	683400	4037105	4049000
GB	Gold Butte	753988	757926	4042233	4046146
HW	Halfway	733487	738935	4076052	4082577
IV	Ivanpah	640600	647000	3906000	3911000
JT	Joshua Tree	567043	618822	3726790	3776272
MC2	Rodman-Sunshine Pk South	594140	599833	3786926	3791006
OR	Ord-Rodman	537421	542688	3833406	3836361
PM	Piute Mid	685211	689204	3918456	3923835
RM	River Mountains	685989	701400	3886709	4092795
SC4	Superior Cronese	538500	546200	3874000	3879500

Procedures for addressing errors

Please see the corresponding section under Phase 1 procedures (page 14).

Non-automated overall database checks

- *Visually inspect database in design view to ensure field settings comply with specifications.*
- *Scan all tables and fields for visually unusual patterns or breaks in patterns that may reflect unexpected inconsistencies in the data.*
- *Sort ascending and descending number, date and time fields from all tables looking for values that are out of given ranges.*
- *View unique entries in each field looking for typos or values that are not consistent.*

Final processing steps

- For the Season Transects and G_0 databases, generate a final unique ID number for each record (transects, waypoints, observations, start records, G_0 observations) that will be unique between all years of Season data. The ID is generated by prefixing a sequential ID number with the year of data collection and the type. For example, transects ID will be generated by concatenating "2020trn" with a sequential ID number.
- Verify and update summary information on the strata boundary feature class for transects selected, transects walked, km walked, group, sampling start date, sampling end date, total transect live and carcass observations, total opportunistic live and carcass observations, and other summary information to comply with USFWS specifications.
- Create metadata for preseason training lines, monitoring season transects, and monitoring season G_0 databases.
- Add the final ID from the G0_Start record to its associated observations to act as a single-field unique identifier in place of the G0_prime_key identifier created by the Nexus phone units.
- Generate summary report of general types of errors found, indicating the most common errors and those most important from the tortoise monitoring perspective. Some topics on which to report:
    - Categories of common/important errors in Violations table
    - Categories of unusual values (times, etc.) that were not adequately documented in Violations table
    - Fields that were most involved in "missing data" errors
- Create and deliver the following final internal user products (for possible later dissemination) and internal management products (for monitoring coordinator use) to USFWS (and others as specified by USFWS). Internal user products should not include any fields for photos, datasheets, comments, eastings or northings.
    - Preseason: Training Lines database, Practice Transects database, scanned datasheets, photos, and QA/QC documentation
    - Transect database, scanned datasheets, photos, and QA/QC documentation
    - G_0 database, scanned datasheets, photos, and QA/QC documentation
    - Error summary report
    - Additional content, as needed, resulting from Desert Tortoise Monitoring efforts

    The Preseason and Season database products will be provided in a variety of formats, where applicable, including:
        - ArcGIS File Geodatabase (Season Transects and G_0)
        - Stand alone Microsoft Access Database (not a geodatabase)
        - Microsoft Excel

Data Finalization must be completed by **30 September 2020** to enable timely analysis of data for final reporting.

Maintenance Phase: Data Dissemination and Long-Term Storage

The USFWS is responsible for formatting data for public access. Data are available at http://psw.databasin.org/galleries, where the "Species of Interest" gallery includes *Gopherus agassizii* Topoworks keeps copies of each year's Phase 3 products. The USFWS stores all copies of paper datasheets.

Spatial information will be provided for monitoring strata, transects, and tortoise locations. The following attributes will be provided. Non-spatial data will be provided for telemetry (G_0) sites as indicated.

Season Database	Table	Field/Attribute
Transects	Transects	year_
		tran_num
		stratum
		date_
		trn_length_m
		num_tranliveobs
		num_trancarcobs
	Waypoints	[none]
	TranLiveObs	year_
		tran_num
		easting
		northing
		z12_easting
		z12_northing
		Perp_distance_m
		sex
		mcl_mm
	TranCarcObs	year_
		tran_num
		easting
		northing
		z12_easting
		z12_northing
		perp_distance_m
		sex
		mcl_mm
	OppLiveObs	[none]
	OppCarcObs	[none]
	Stratum_Info	stratum
		stratum_desc
		transects_walked
		km_walked
		total_tran_live
		total_tran_carc
		group_
		sampling_start_date
		sampling_end_date

Season Database	Table	Field/Attribute
		area_sqkm
		area_acres
G0	G0_Start	[none]
	G0_Obs	date_
		G0_site
		Tort_num
		time_
		visible
	G0_Site_Info	G0_site_desc

Appendix B: Data dictionaries