8. FIELD DATA QUALITY ASSURANCE/QUALITY CONTROL (QAQC I)

Desert tortoise monitoring QA/QC is accomplished in three phases, of which the field data collection teams are responsible for the first phase, or QA/QC I. The goal of QA/QC I is to provide the most accurate, valid, and complete data to USFWS. Each and every crew member is responsible for QA/QC I. As a field monitor you are responsible for double-checking your data at each entry as well as reviewing it at the end of each transect for completeness and correctness before submitting it. At this point your team's QA/QC specialist receives your data and downloads it into a Collection database that contains all your fellow crew members' data. To keep an unaltered copy of this Collection database, it is imported into the Contractor database before edits are identified and made. The QA/QC specialist uses all tools (provided scripts and visual inspection) to search the data for inconsistencies and then corrects them. In the process of correcting the original field data, your QA/QC specialist may only need to review your paper data sheets. In other cases, the specialist will need your assistance in correcting errors in your data. The outline below details the individual objectives and standards each data collection team must meet. Field technicians and QA/QC specialists need to work together to meet the goal of these objectives: to deliver correct and complete data to benefit recovery of the desert tortoise.

Objective 1: Limit and correct data entry errors
Each field technician will 1) understand data required under every data field; 2) enter paper and electronic data in parallel; 3) use these data forms in tandem to check for completion and correctness at data entry, at transect completion, and when reviewing data from other teams; 4) understand each QA/QC check; and 5) understand checks that will be performed at later stages of QA/QC.

Objective 2: Initiate QAQC I by the QA/QC specialist
Each QA/QC specialist will implement a system of procedures for handling data between completion of data proofing by field technicians and through delivery to QA/QC II.

Objective 3: Execute QA/QC scripts
Each QA/QC specialist will be trained to 1) import field data, 2) understand conceptually how the scripts work, 3) understand each QAQC check, and 4) understand how and where errors are logged

Objective 4: Identify errors that are not addressed by QA/QC scripts
QA/QC specialists will be trained to 1) systematically work with each data table to visually identify errors.

Objective 5: Correct errors before data delivery
Each QA/QC specialist will be trained to 1) identify error records, 2) add error records for errors not identified by QA/QC scripts, 3) identify if error is correctable by contacting crew member or using paper data sheets, 4) correct errors, and 5) update the error table fields (old_value, new_value, resolution, resolver and the error_status field to explain action taken.

Objective 6: Database back up and delivery
Each QA/QC specialist will be trained to 1) back up database to external device daily, and 2) send the database to appropriate personnel by uploading to FTP sites.

Objective 7: RDA annual preparation and recovery from system failure
Each QA/QC specialist will be trained to recover RDA units in case of total system failure. This includes, 1) performing hard reset, 2) setting date and time, 3) syncing RDA and installing Pendragon forms software, 4) pairing GPS, 5) loading field data collection forms, and 6) testing the RDA setup.

Metrics: Each QA/QC specialist will be tested on their understanding of basic QA/QC process operation and ability to restore a faulty RDA. They will be given several RDA units to download data from, they will run QA/QC scripts, correct identified errors, identify additional data errors and correct them, backup the database and send it to appropriate personnel. In addition, they will be required to restore a faulty RDA. They will be re-trained if they are not proficient with the QA/QC process and restoring RDA. Continued delivery of poor quality data will result in the recommendation to USFWS that they not participate in monitoring.

Objective 1: Limit and Correct Data Entry Errors

1. **Crews proof their RDA and paper data sheets for missing entries every day.**

   In so doing they should find and complete any blank data fields assuming an answer can be found on the paper data sheet. If they cannot complete a blank data field it should be reported to the QA/QC specialist when data from their RDA is downloaded.

2. **Crews compare their RDA and paper data sheets for discrepancies.**

   If possible, discrepancies between the two should be resolved immediately. Any discrepancies that you are unable to resolve should be reported to your QA/QC specialist when data are transferred. These errors should be recorded in the discrepancy spread sheet so they do not continue to cause the QA/QC specialists to expend further effort.

3. **Crews will be apprised of identified issues so they can correct the way they enter data for future deliveries.**

   All resolvable, correctable, or fixable (synonyms used for emphasis!) data entry issues should be resolved before submitting the data to MDEP. With each passing day issues such as these become more and more difficult to fix.

   Crews should have face to face conversations with the QAQC specialist when their team has an identified issue. In addition, crews are responsible for ensuring they have discussed each weekly assessment provided by the FWS to the crew leaders. These assessments report on issues that are priorities for all field crews.

Objective 2: Initiate QAQC I by the QA/QC specialist

Accept paper and electronic sheets from field crews. While crews are present, evaluate legibility of handwriting and check the following information, which is more difficult to review in electronic form:

- Ensure that RDA is functioning correctly
- Review paper sheets for any blank fields
- Check whether drawing of transect 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
- Check whether any tortoises are indicated as "mcl_greater_180=unknown". If so, 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.

Run the error scripts (Objective 3)

Objective 3: Executing QA/QC Scripts

1. **Import Collection (Pendragon) database into the Contractor (QA/QC) Database**
   a. Open the QA/QC scripts database by double clicking “Import_QAQC.mde” file.
   b. Select the “Import data” tab on top if not already selected. Click on “Import Raw Data” to import the Collection database into the Contractor database. If training database also needs to be imported, check the "Import Training Data" checkbox before clicking "Import Raw Data". After importing, certain fields are automatically populated in the Contractor database. For instance, “tran_num” is populated in sub-forms; latitude and longitude are calculated from GPS grabs, etc.
   c. All the errors encountered during the import process are logged into (“LDS_Import_Log.txt”) file located in “C:\Program Files\Forms3”. This file must be checked after each import to see if any errors were encountered during import process

   NOTE: Before running the QA/QC scripts for the first time make sure that the Contractor database (Import_QAQC.mde) exists in the same folder as the Collection database (C:\Program Files\Forms3). If not, copy the latest version of the Contractor database in this folder.

2. **Understand how to execute QA/QC checks**
   a. To run the QA/QC checks select the “Transect Database Scripts” or “Training Database Scripts” tab on top. Select the desired checks to be performed or select "Select All" and all the checks will be selected. Click the “Run selected QA/QC checks” button to run selected checks.
   b. If updates are made to tran_num, stratum, team_num, group_ fields in Transects table or date, G0_site and group_ fields in G0_Start table, or trial_number, team_number, training_line_color, transect, training_date, transect_bearing fields in Tran_Train table the fields in related child subforms can be automatically updated by clicking the “Re-Calculate Fields” button on either “Transect Database Scripts” or “Training Database Scripts” tab.

3. **Understand Each QA/QC Check**
   a. Relationship Checks
      i. Missing Waypoints
      This checks if a xxx format transect is missing waypoints 0, 1, 99, 100 and if xxx.x format transect is missing waypoint 99.
      Tables included in check: *Waypoints*
      Error logged: *“ missing waypoint xxx”*

ii. Orphan Records

This checks for orphan records that get created when a parent record is deleted without first deleting the related child records.

Tables included in check: *Waypoints, OppCarcObs, OppLiveObs, TranCarcObs, TranLiveObs, g0_Obs, g0_OppLiveObs.*

Error Logged: *"missing parent record"*

b. Duplicate Checks

i. Transects

This checks for duplicate transect numbers in transect table.

Tables included in check: *Transect*

Error Logged: *"duplicate transect"*

ii. Waypoints for transect

This checks if duplicate waypoints are present for a transect.

Tables included in check: *Waypoints*

Error Logged: *"duplicate waypoint"*

c. Attribute Condition Checks

i. Lead and Follow

This checks if there are inconsistencies between observer1 and observer2. An error is logged if (a) observer1 or observer2 is missing or (b) if observer1 and observer2 are the same.

Tables included in check: *Transects.*

Error Logged: *"observer1 is Null" or "observer2 is Null" or "observer1 and observer2 are the same"*

ii. Existing Tag information

This checks if there are inconsistencies between existing tag and new tag. An error is logged if (a) existing tag is "yes" and tag attached is "yes" or (b) if existing tag is "unknown" and new tag attached is "yes" or (c)if existing tag is "no" and new tag attached is "no"

Tables included in check: *OppLiveObs, TranLiveObs, g0_OppLiveObs.*

Error Logged: *"inconsistency between existing tag and new tag attached"*

iii. Incorrect Date and Time

This checks for records with incorrect date and time. An error is logged if the date is not within 3/1/10 and 6/2/10 or times are not within the following:

Transects Table

1. if *do_time* is not between 4:00am and10:00am
2. if *tran_start_time* is not between 5:00am and10:00am
3. if *tran_end_time* is not between 8:00am and6:30pm
4. if *ret_do_time* is not between 8:00am and 6:30pm

Waypoints and G0_Obs Table

1. if *time_* is not between 5:00am and 6:30pm

TranCarcObs and TranLiveObs Tables

1. if *time_* is not between 5:00am and 6:00pm

G0_Start Table

1. if *start_time* is not between 5:00am and10:00am
2. if *end_time* is not between 8:00am and6:30pm

Tables included in check: *Transects, Waypoints, TranCarcObs, TranLiveObs, g0_Start and g0_Obs_0.*
Error Logged: *"time is not within domain" or "date is not within domain"*

iv. MCL Values

This checks for inconsistencies between mcl>=180, mcl_mm, mass_g, location and carc_condition fields.

1. mcl_mm and mcl_greater_180
   An error is logged if (a) mcl_mm > 180 and mcl_greater_180 is "no", or (b) mcl_mm < 180 and mcl_greater_180 is "yes", or (c) mcl_mm > 0 and mcl_greater_180 is "unknown"
2. if mcl_mm is 0 if temperature is less than 35C_
3. Carc_condition and mcl_mm
   An error is logged if (a) carc_condition is "Intact" and mcl_mm is "null" or (b) if carc_condition is "Disarticulated" and mcl_mm > 0
4. mcl_mm and mass_g
   An error is logged if (a) mcl_mm > 0 and mass_g <= 0 or (b) mcl_mm =< 0 and mass_g >0
5. location, mcl_mm and mass_g
   An error is logged if location is not burrow and mcl_mm or mass_g is null if temperature is less less than or equal to 35C.

Tables included in check: *OppCarcObs, OppLiveObs, TranCarcObs, TranLiveObs, g0_OppLiveObs.*
Error Logged: *"inconsistency between mcl_mm and mcl_greater_180", or "mcl_mm is 0" or "inconsistency between mcl_mm and mass_g" or*

"inconsistency between location, mcl_mm and mass_g" or "inconsistency between carc_condition and mcl_mm"

v. Temperature

This checks if there are any inconsistencies between temp>35 and actual temperature fields.

Tables included in check: OppLiveObs and TranLiveObs.

Error Logged: "inconsistency between temp_C and temp_greater_35C"

vi. Visibility

This checks inconsistency between visibility and behavior. An error is logged if invisible tortoise has behavior other than "unknown"

Tables included in check: g0_Obs.

Error Logged: "inconsistency between visibility and behavior"

vii. Burned

This checks if burned field is not null for sites other than CS and HW".

Tables included in check: G0_Obs

Error Logged: " site is CS or HW, but burned is null"or "site is not CS or HW, but burned is not null"

d. Spatial Condition Checks

i. UTM Zone

This checks for missing or incorrect UTM zone. An error is logged if sites other than "BD" and "GB" have UTM zone 12.

Tables included in check: Waypoints, OppCarcObs, OppLiveObs, TranCarcObs, TranLiveObs, g0_Obs, g0_OppLiveObs.

Error Logged: " gps_zone or manual_zone are 12, but stratum is not BD or GB"or "missing gps or manual zone"

ii. Easting and Northing

This checks if the spatial location of the record is within the strata boundary buffer. An error is logged if the co-ordinates are not within 3km buffer around the strata boundary.

Tables included in check: Waypoints, TranCarcObs, TranLiveObs, g0_Obs.

Error Logged: " easting or northing are not within stratum boundary"

iii. Missing Location Data

This checks for records missing GPS grab and hand entered co-ordinates. An error is logged if GPS grab as well as manual entry co-ordinates are null.
Tables included in check: *Waypoints, OppCarcObs, OppLiveObs, TranCarcObs, TranLiveObs, g0_Obs, g0_OppLiveObs.*
Error Logged: *" missing gps coordinates"*

e. Training Database Checks

i. Training Transects

This checks if a team has recorded a duplicate transect for the same trial number in training database.
Tables included in check: *Train_Tran*
Error Logged: *" duplicate training transect"*

ii. Tortoise ID

This checks if a team has recorded a duplicate tortoise in one day
Tables included in check: *Train_Obs*
Error Logged: *"duplicate TortoiseID"*

iii. Observer name and position

This checks if the observer name and observer position does not match lead and follow in training database.
Tables included in check: *Train_Tran, Train_Obs*
Error Logged: *" observer_name and observer_position do not match lead or follow in Train_Tran table*

iv. Transect Segment Number

This checks if the transect_seg_number is not matching the calculated value.
Tables included in check: *Train_Obs*
Error Logged: *" transect_seg_num does not match transect bearing and start post"*

v. Time

This checks the time values in training database and error is logged if (a) Training start time is not before training end time or (b) Observation time not between training start time and training end time
Tables included in check: *Train_Tran, Train_Obs*

Error Logged: *" training_start_time is after training_end_time" or " observation_time is not between training_start_time and training_end_time*

vi. Radial Distance

This checks if the radial distance has more than one decimal
Tables included in check: *Train_Obs*
Error Logged: *" radial_distance_m has more than one decimal place"*

vii. Bearing

This checks if the local bearing not within 40 degrees of transect bearing.
Tables included in check: *Train_Obs*
Error Logged: *"local bearing not within 40 degrees of tran bearing"*

viii. Perpendicular distance

This checks the following conditions on perpendicular distance. An error is logged if (a) if the perpendicular distance greater than radial distance or (b) If the perpendicular distance is greater than 25m
Tables included in check: *Train_Obs*
Error Logged: *" perp_dist_m is greater than radial_distance_m" or " perp_distance_m is greater than 25 m*

4. Understand How and Where Errors are Logged

After executing the QA/QC checks all errors encountered are logged into the Errors table. 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" for errors that cannot be resolved.

Objective 4: Identify errors that are not addressed by QA/QC scripts

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 Violations table being sure to correctly identify the record by its TableName and TableRecord_ID fields. Many of the non-automated checks are purposefully redundant of the automated checks to provide a last review of the data and to catch errors or inconsistencies that might have been introduced earlier in the QA/QC I process. Such redundant checks are shown in italic below.

Checks 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'. Until the status is changed to 'exception allowed', automated scripts will identify these errors each week.
- Check numeric fields for errors (e.g. decimals after numbers, zero preceding number). Sort by field.
- View unique entries in each text or categorical field looking for typos or values that are not consistent, such as observer names with typos, entries with extra punctuations, values that are abbreviations, etc. Unique entries for continuous numeric data such as radial distance, easting, northing, mass, etc. do not need to be viewed.

Transects_10 table

- *Check if do_time, tran_start_time, tran_end_time and ret_do_time are null (1:00AM). Sort by time.*
- Check if tran_standard field is null. Sort by field.
- Check for the accuracy of tran_standard field. Visual check of transect waypoints converted to lines in GIS software. Display by tran_standard. Only typical square 12km transects should be tran_standard = Y.
- 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 PDA date/time stamp. Do not edit the TimeStamp fields, but if necessary, corrections can be made to the individual date and time fields.

Waypoints_10 Table

- *Lead or Follow fields should be null only for waypoints 0, 99 and 100.*
- Check for waypoint numbers that are missing or invalid (skipped, duplicate, or with decimal places).

OppCarcObs_10 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.*

OppLiveObs_10 Table

- Check for the accuracy of detection numbers. Sort by transect number, then detection number.
- 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.
- The mcl_mm and mass fields should be null if the temperature is greater than 35°C.
- 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.

TranCarcObs_10 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.

TranLiveObs_10 Table

- Check for the accuracy of detection numbers. Sort by transect number, then detection number.
- Look for radial distances with more than one decimal place. Radial distance should only be recorded to one decimal place.
- 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.
- The mcl_mm and mass fields 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 and mass should not be null.*
- *Check if the existing tag is "No" and new tag attached is "No".*
- Sort ascending and descending on both mcl_mm and mass and look for unrealistic discrepancies.

- Check for New tag number format. Should be “FWxxxx”. Should not be “FW-xxxx”.

G0_Start_10 Table

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

G0_OppLiveObs_10 Table

- Check for the accuracy of detection numbers. Sort by transect number, then detection number.
- If the tortoise location is burrow, check if the burrow_visibility and tortoise_in_burrow_visibility should not be null. For other locations the tortoise_visibility field is not null.
- The mcl_mm and mass fields should be null if the temperature is greater than 35°C.
- 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.

G0_Obs_10 Table

- The burned field should be null for all sites except CS or HW.
- *If the tortoise is not visible the behavior can only be unknown.*
- If the tortoise is in open then behavior should not be unknown.
- If the tortoise location is burrow, check if the burrow_visibility and tortoise_in_burrow_visibility should not be null. For other locations the tortoise_visibility field should not be null.

Train_Tran_10 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 xx
- Sort transect_seg_num descending, look for numbers higher than xx
- *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
- sort end_post descending, look for letters higher than xx

Train_Obs_10 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#*

USFWS will perform a weekly assessment of each vendor's populated, corrected Contractor database. This is a non-automated review of the data and is intended to provide timely feedback in order to reduce errors in future weeks of data collection. Because vendors submit appended contractor databases each week, there is also opportunity to remedy errors identified by USFWS in the weekly assessment. The weekly assessment 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.

Objective 5: Correcting Errors before Data Delivery

1. **Identify Error Records**
   After the QA/QC checks are run, all the errors encountered are logged into the Errors table in Contractor database. Following are the fields present in the Errors table. Fields (a)-(k) are automatically populated by the scripts and fields (l)-(o) are manually entered by QA/QC specialist after correcting error.

   a. ID – auto-number used to identify record
   b. date – the date when the QA/QC scripts were run
   c. table_name – table name where error was found.
   d. prime_key – primary key for the table with error record (for transect table – tran_prime_key, for waypoints – wp_key, for OppCarcObs – OppCarcObs_key, etc.)
   e. tran_num – transect number for error record, -99 if not applicable
   f. stratum – stratum for the error record, null if not applicable
   g. team_num – team number for the error record, -99 if not applicable
   h. wp_obs_num – waypoint number or observation number depending on error table
   i. tran_date – date when the transect was walked
   j. error_desc – short description on type of error found
   k. old_value – old incorrect value of the field
   l. new_value – null, new correct value filled in while correcting errors
   m. resolution – null, resolution description filled in while correcting errors
   n. resolver – null, name of agency correcting the errors
   o. error_status – null, filled in while correcting errors

   The error records can be identified using the information in table_name, prime_key, tran_num, stratum team_num, wp_obs_num and tran_date fields in the Errors table. Some of the fields might not be available depending on the error source table.

2. **Add error records for errors not identified by QA/QC scripts**
   For errors that were identified during systematic visual inspection of tables, the error records will not be created automatically in the Errors table. In such cases a manual error record will need to be created. The QAQC specialist will have to populate fields (b)-(o) as above 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.

3. **Identify if Error is Correctable**
   After identifying the error record, a decision must be made if the error can be corrected or not. To determine if the 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.

4. **Correct Errors in the Contractor Database**
   If the error can be corrected, the error record identified should be corrected in the corresponding table in the Contractor database.

5. **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 Errors table must be manually filled in for the error record. The "resolution" column should clearly mention steps taken to correct the error or should explain why the error could not be corrected. 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. The resolver column should contain the name of the person/agency correcting the error. The error_status column should contain "resolved" if the error was corrected or "exception allowed" if the error cannot be corrected. Marking the status to "exception allowed" will ignore the error and it will not be logged again in the Errors table for subsequent QA/QC checks.

6. **Update the paper data sheets to reflect changes to match electronic database**
   If an error on the paper data sheet is corroborated, the QA/QC person 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 in order 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 date and identify the commenter. The commenter documents necessary edits by writing, "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." Rather than replace the original scanned file after edits are made, the edited version should be renamed to add the initials of the reviewer (see "Deliver Paper Datasheets," below).

Objective 6: Database Backup and Delivery

1. **Backup Collection (Pendragon) database**
   The collection database should be backed up on external storage device daily after any new data collected on all RDA's have been imported (hotsynced) into the Collection database.

   The collection database file "FORMS32K.MDB" is located in the "C:\Program Files\Forms3" folder on the machine used for hot-syncing RDA's. To back up the database, copy this file ("FORMS32K.MDB") onto the external storage device and then rename the file on the external storage device to "FORMS32K_ContractorName_MMDDYY.MDB". The date should correspond to the last instance of collected data. For instance, if the data were ready to upload on 8 April 2010 and represent data collected through the day before, the file name would include "040710". Do not rename the source file found in "C:\Program Files\Forms3".

2. **Deliver Collection Database**
   The Collection database backed up on the external storage device ("FORMS32K_ContractorName_MMDDYY.MDB") in the previous step should be uploaded to the MDEP FTP site. (see Step 5 below).

3. **Backup Contractor (QA/QC) Database**
   The Contractor database should be backed up after each QA/QC session. QA/QC session consists of importing new data from the Collection database, running QA/QC checks and then correcting all the errors found in the Errors table.

   The contractor database file "Import_QAQC.mde" is located in the "C:\Program Files\Forms3" folder on the machine used for hot-syncing RDA's. To back up the database, copy this file ("Import_QAQC.mde") onto the external storage device and then rename the file on the external storage device to "Import_QAQC_ContractorName_MMDDYY.MDE". The date used for naming should match the date of the corresponding "FORMS…..mde" file. Do not rename the source file found in "C:\Program Files\Forms3".

4. **Export the Contractor database into Access 2003 format**
   The following steps export the Contractor database tables into an Access 2003 format database using Access 2007.
   a. Create a new access database on your external hard drive.
      - Open Access 2007, select the Office 2007 blob and click "New"
      - On the right-hand side next to the file name, click on the Browse button (it looks like the familiar "folder" icon).
      - Select the external hard drive as the location where the new database will be saved and then select "Microsoft Office Access Databases (2002- 2003 format) (*.mdb)" for "Save As Type"
      - Enter the filename as " Import_QAQC_ContractorName_MMDDYY _v2003.mdb" and click OK button. Click the "Create" button to create an empty Access 2003 database on your external hard drive. The date used for naming should match the date of the corresponding "FORMS…..mde" file.

b. Import records from the contractor mde database that is on your c-drive.

- New table "Table1" will be opened by default, close it.
- Select "External Data" from the top tab bar.
- Within the Import group, click on "Access" and browse to the c-drive location of the contractor database ("Import_QAQC.mde") to identify the file from which data will be imported. Select the file and click "open". Note that you are importing from your working version of the contractor database, not from the version backed up on the external drive.
- Click "OK" on the dialog for "Select the source and destination of the data".
- If you see a Security notice click "Open".
- You should see an "Import Objects" Dialog with the Tables Tab selected. Click the "Select All" button on the right hand side to select all the tables and then click "OK".
- Click "Close" on "Save Import Steps" dialog.
- You should see all the tables imported into this new database.
- Close the database and upload it to the MDEP FTP site (see next step).

5. Deliver QAQC Databases (Access 2007 and 2003 versions)

The contractor databases on the external storage device ("Import_QAQC_ContractorName_MMDDYY.MDE" and "Import_QAQC_ContractorName_MMDDYY _v2003.MDB") should be uploaded to the Weekly_Data folder in the _GBI10 or _IWS10 or _Kiva10 folder on the MDEP FTP site. After the databases have been uploaded, send an email to appropriate personnel about data delivery. At a minimum, notice should be sent to Linda Allison and Doug Zeliff. The following is sufficient notice:

"FORMS32K_CONTRACTORNAME_MMDDYY.mdb and Import_QAQC_CONTRACTORNAME_MMDDYY.mde and Import_QAQC_CONTRACTORNAME_MMDDYY_V2003.mdb have been uploaded to the MDEP FTP site in this directory.

_CONTRACTORNAME10/weekly data"

USFWS has provided each group with details on accessing and uploading data to the FTP site.

6. Deliver Paper Datasheets

Each week, QA/QC specialists send scanned versions of that week's paper datasheets to MDEP. The DPI Rate should be 300 or higher 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 if further data errors and inconsistencies are addressed. The original datasheets themselves should be sent to MDEP when the QA/QC database for the associated transects is uploaded to the ftp site. Final data delivery should include any scanned data sheets that were annotated after the original version was uploaded to the ftp site. These edited files should have the reviewer's initials affixed at the end of the

file name that was otherwise built using the conventions below (e.g. 168_GB_2009_mb.pdf, where "mb" are the initials of the person(s) who added digital annotations).

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', 'PracticeTransects_datasheets', and 'PracticeG0_datasheets'.

TrainingLines_datasheets: This folder should contain scanned datasheet PDF files for the practice training lines. All sheets associated with a single transect 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 Transect (examples: Team21_Trial1_Green7.pdf, Team6_Trial2_Orange3.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_2009.pdf, 14.1_LSTS_2009.pdf).

PracticeG0_datasheets: This folder should contain scanned datasheet PDF files for the practice G0 iterations. All sheets associated with a single iteration for an observer on a day should be grouped into one PDF file (both the G0 observations sheet and the OppLiveObs sheet should be grouped into a single 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, one of the files should be suffixed with '_A' and the other with '_B'. (examples: PM_040208_A.pdf, PM_040208_B.pdf, PM_040308.pdf).

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_2009.pdf, 122_AG_2009.pdf; examples for the 'GB' subfolder: 168_GB_2009.pdf, 239_GB_2009.pdf).

G0 Data: G0 scanned datasheets should be organized into a folder named "G0_datasheets". 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, one of the files should be suffixed with '_A' and the other with '_B'. (examples for the 'CK' subfolder: CK_040208_A.pdf, CK_040208_B.pdf, CK_040308.pdf; examples for the PM subfolder: PM_040508_A.pdf, PM_040508_B.pdf, PM_040608.pdf).

Objective 7: RDA Preparation and Recovery from System Failure

The recovery procedure should only be performed by QA/QC personnel if soft reset as well as hard reset with system restore does not work. This method clears all the data from the RDA and PC and then does a fresh install of all software.

Approximately 5 days before training begins. QA/QC personnel are responsible for checking and conditioning RDAs, then configuring RDAs with the season's software and forms and linking each RDA to its associated laptop. Conditioning procedures include replacing the screen covers, replacing and attaching styluses as needed, recharging the RDAs and GPS units, and identifying any units from the outset that do not hold charge well or have faulty accessories, such as charging cables. The latter procedures (preparing RDAs with the season's software and forms) mirror those that can be used later to restore the RDA from system failure.

Both procedures involve interaction between the RDA and PC so that software on both can remove or distribute files (forms) that we use for monitoring work. "Hotsync" operations, used frequently below, refer to this interaction between the two devices and their shared software.

Installing and configuring RDA for the first time

If the RDA is being setup for the first time please follow the procedure outlined in the "Recovery from System Failure" section on next page.

Updating the RDA database

The following procedure should be used to replace the old working version of Pendragon database on the RDA with a new updated Pendragon database version.

1. **Material to have at-hand**
   - Have ready the RDA, hotsync cable, and an Aceeca CD.
   - Your machine should be loaded with Palm Desktop and Pendragon 5.1 software.
   - Under c:\Program Files on your PC, Pendragon is in the Forms3 folder, where you should also see FormsGPS.prc and Mathlib.zip.
2. **Erase all existing versions of Pendragon forms from your RDA**
   - Remove the current user from the Default Group list for the Pendragon database.
   - Sync the RDA to remove all current forms from the RDA
   - Verify that there are no old forms on the RDA. Open forms 5.1 and there should not be any forms in the list. Click on "Pendragon forms" on the top left. From the menu, select Delete form designs, after which it should either show you an empty list, or give you a fatal exception (it does this sometimes…). If there are any forms, select them and click delete for all forms.
3. **Install the newest version of the Pendragon forms.**
   - On your PC, copy the new database into the c:\Program Files\Forms3 folder. Rename any existing copies with an appropriate trailer, then rename the new version to FORMS32K.mdb.

- In Pendragon, open the database and add the user under “Users” and “Groups” (default group).
- Sync the RDA and you should have all the forms from the new database.

Recovery from System Failure

The following procedure should only be followed by QA/QC specialists in the following cases

1. Setting up a new RDA
2. If the RDA has regular fatal errors
3. If the RDA has a system failure and hard reset with data restore from cardbackup does not help.

This following procedure will delete all the data on RDA

1. **Hard Reset**
   Perform a Hard reset on the RDA to erase all existing data(see also page 39 in Chapter 4)

2. **RDA Setup**
   Follow on screen instructions to complete the RDA setup. Make sure it displays correct date and time, if not then change it to current date and time.
3. **Delete Palm Desktop Account from PC**
   On the PC, start the Palm desktop application by clicking start->Programs-> Palm Desktop and then Palm desktop. On the right hand top corner you will see Users dropdown. Click on the dropdown and select “Edit Users”. Click on the current RDA user and click “Delete”. Select “Yes” for deletion confirmation. After this, close the Palm Desktop application. This removes all the data associated with the current user on the PC..
4. **Setup RDA hotsync**
   On the RDA, use the Home window to get to and select “Hotsync” Push the center icon to perform hotsync. It will ask for a User account on the PC. Click “New” to create a new account and enter the name of the RDA.
5. **Install Pendragon Forms application onto the RDA**
   After the above hotsync is successfully completed, on PC click “Install Forms 5.1 on Palm OS Device” found in windows programs. (Start->programs->Pendragon Forms 5.1). Select the new user created in the previous step and click “Done”. Perform another HotSync on the RDA, which will distribute the Pendragon Forms application from the PC to the selected RDA. You should now see the Forms 5.1 on RDA.
6. **Install FormsGPS and Mathlib files onto the RDA.**
   On the PC, navigate to the “C:\Program Files\Forms3” folder, and double click on the “FormsGPS.prc” file. Select the new user created and click “Ok” and “Done” on the next dialogs. Follow the same procedure to install the “Mathlib.prc” file from the “Mathlib.zip” file. Perform a Hotsync on the RDA to install these files.

7. **Load forms for the tortoise monitoring database if needed**
   Open the Pendragon forms access database on the PC and make sure that the RDA username exists in the "Users List" and as a member in the "Default Group" within "Groups". If it does in both the locations then you should see all the monitoring forms on the RDA. If you add RDA username into “Users” or “Default Group”, perform Hotsync on RDA and you should see all the monitoring forms on the RDA.
8. **Pair Bluetooth GPS with RDA**
   Instructions for pairing can be found in Chapter 7 under “Using Bluetooth GPS and RDA Together.”
9. **Test RDA Setup**
   Test the RDA setup to confirm that all the forms and subforms are accessible and confirm that the perpendicular distance calculation and GPS grabs are working.