**FRAMBuilder 2.0**

**Program documentation & processing steps for preparing coded-wire tag data for Chinook FRAM base period calibration**

**The Chinook FRAM Base Period Workgroup, November 2019**



Contents

[**1.** **Background and purpose** 2](#_Toc24554132)

[**2.** **Overview of the process** 3](#_Toc24554133)

[**3.** **Required programs and data files** 4](#_Toc24554134)

[**4.** **Step 1: Select tag groups** 6](#_Toc24554135)

[**5.** **Step 2: Query RMIS for release/recovery data** 6](#_Toc24554136)

[**6.** **Step 3: Filter RMIS data for importing to CAS** 9](#_Toc24554137)

[**7.** **Step 4: Load filtered CWT data into CAS** 10](#_Toc24554138)

[**8.** **Step 5: Run FRAMBuilder** 13](#_Toc24554139)

[**9.** **Step 6: Export data** 21](#_Toc24554140)

[**10.** **FRAMBuilder’s growth function input file preparation feature** 22](#_Toc24554141)

[**11.** **Limitations to FRAMBuilder and opportunities for enhancement** 22](#_Toc24554142)

[**Appendix A. CTC Filter Database Recoveries Query Details/notes** 24](#_Toc24554143)

[**Appendix B. Overview of the FRAM-modified CAS database** 25](#_Toc24554144)

[**Appendix C. List of “Don’t Forget” external data pre-processing steps** 28](#_Toc24554145)

[**Appendix D. Updates for compatibility with 2019 CAS fishery strata** 30](#_Toc24554146)

1. **Background and purpose**

General purpose

Although the Regional Mark Processing Center’s (RMPC) Regional Mark Information System (RMIS) contains considerable information about the recovery of Chinook salmon with coded-wire tags (CWT), considerable processing must occur in order to translate this information into currency that’s meaningful within a FRAM base period (BP) calibration context. Firstly, individual tag groups must be associated with a specific FRAM model stock. Secondly, tags recovered at a particular location (indicated by RMIS location code), time, and using a particular gear, must be mapped to one of FRAM’s model fisheries and time steps. The FRAMBuilder program and workflow described here was developed to fulfill these needs, among others. For instance, the program, and companion FRAM-CAS database, was modified to facilitate the preparation of inputs for estimating the parameters of the von Bertalanffy growth functions used by FRAM.

The connection to CTC tools

Early in the development of FRAMBuilder and the overall CWT mapping workflow, the base period workgroup (BPW) identified distinct advantages/benefits to leveraging the Pacific Salmon Commission’s Chinook Technical Committee’s (CTC) CWT analysis tools (i.e., the Cohort Analysis System [CAS] mapping program and companion database) within a FRAM calibration context. The BPW ultimately decided to tie FRAMBuilder to the CTC world because this connection: (1) allows for the seamless integration of CTC ‘Auxiliary’ CWT files, agency-supplied/prepared files that supplement or correct known errors/gaps in RMIS’s CWT recovery information; (2) facilitates the efficient inclusion of screened/vetted CWT release groups (i.e., selected by CTC members with regional expertise) into the calibration database; and (3) increases the overlap in information driving models supporting the management decisions of the PSC, the Pacific Fishery Management Council (PFMC), and state–tribal co-managers. Additionally, given partial overlap in the fishery assessment units used by the CTC and in FRAM, the integration of CAS into the FRAM calibration workflow offered efficiency in the form of an initial stage of RMIS-to-FRAM mapping.

Document scope

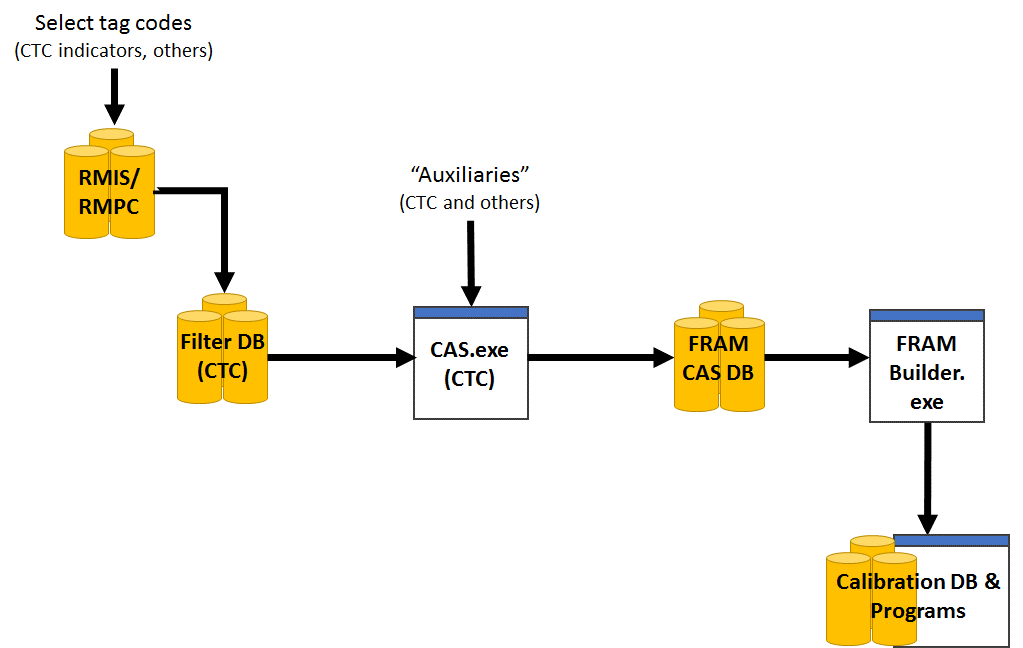
This user’s manual is meant to serve two purposes. Firstly, it provides a roadmap of the process that gets one from raw RMIS CWT release/recovery data to something useable in a FRAM calibration. Secondly, it provides basic documentation on the structure/function of the FRAMBuilder program, its companion FRAM-CAS database, and the ruleset it follows to get CWT recoveries from the initial CAS stage of mapping to a final FRAM fishery/time step state. As for the tools ‘borrowed’ from the CTC (i.e., CAS.exe), we provide only a brief sketch here and refer the reader to CTC resources for further documentation.

1. **Overview of the process**

In concept, the procedures to map an individual CWT recovery to a FRAM stock and fishery are straightforward: (a) in screening candidate codes, make a determination regarding which tags are suitable representatives for model stocks, and (2) given recovery details, such as RMIS location codes, gear codes, dates, etc., make a determination regarding the model fishery/time step to which the recovery belongs. In practice, however, this task is extremely difficult given that thousands of tag codes (= unique release groups) are available for consideration, resulting in hundreds of thousands of individual tag recoveries that must be mapped from one of tens of thousands of unique location-gear code combinations to one of FRAM’s seventy-two model fisheries. The FRAMBuilder workflow, although cumbersome at first glance, boils this seemingly insurmountable challenge down to a task that can be achieved by one person in a relatively short amount of time (i.e., assuming that candidate codes have been selected and auxiliary files have been acquired/prepared). It commences according to the following steps (Figure 1), each of which will be described in detail further below:

1. Select tag groups.
2. Query RMIS for release/recovery data.
3. Load RMIS query results into the CTC Filter database, and query it to create CAS input files.
4. Load tags into CAS (i.e., stage 1 of mapping – to CTC fishery strata).
5. Run FRAMBuilder (i.e., map/process recoveries).
6. Export data for any necessary post-FRAMBuilder processing to create calibration input files.

In addition to these steps, a handful of other functions can be invoked during step 5, depending on a user’s needs. These are also described further below. The remainder of this document is organized around each of these steps, where each subsection offers both ‘how to’ details and documentation on processing decisions, algorithms, etc. where necessary.



**Figure 2.1.** Relationships between the databases and programs used to construct FRAM base period calibration input files.

1. **Required programs and data files**

Data files

* **A list of tag codes:** A list of tag codes is needed for the purposes of querying RMIS (release/recovery), as well as for populating the CTC Filter database’s ‘STKCDS’ table.
* **CWT release data**: These are the raw release details for the chosen codes, acquired from RMIS via a ‘Tagged Releases’ query; query results are downloaded as a CSV, with the headings specified under Step 2 below.
* **CWT recovery data**: These are the raw recovery details for the selected codes, acquired from RMIS via a ‘Recoveries By Tag Code’ query; query results are downloaded as a CSV, with the headings specified under Step 2 below.
* **Auxiliary files (or ‘auxiliaries’):** These are text files (\*.csv or \*.txt), prepared by CTC members from a variety of agencies/jurisdictions, that contain supplementary CWT recovery information that is meant to augment (or revise) the CWT information acquired from RMIS for some stocks; these files are typically created on a stock/code basis and are necessary to ensure the calibration process includes the most accurate information. For example, CWT recoveries in escapement—a major anchor point for the type of backwards cohort reconstruction underlying FRAM calibration—are not available via RMIS for many Canadian stocks.

Databases (all Microsoft Access)

* **The CTC’s Filter Database:** This is a Microsoft Access database into which the RMIS release/recovery query results (above), combined with a tag list (‘STKCDS’) are loaded. Using two custom queries, this database returns RELEASES.txt and RECOVERIES.txt files which can be imported directly to the FRAM-CAS database.
* **A FRAM-modified CAS database (FRAM-CAS hereafter)**: This Access database is an adaptation of the CTC CAS database (final preseason 2019 version[[1]](#footnote-1)), which includes several tables (and added fields to existing tables) designed to (1) cross-walk CTC fishery strata to FRAM fisheries or (2) to house/contain mapped outputs for direct export/use in CAS.

Programs & companion files

* **FRAMBuilder 2.0:** Because FRAMBuilder is very much an interactive program subject to ad hoc changes/revisions to fulfill the BP team’s evolving needs, it hasn’t yet been developed into a distributed, fully compiled .exe file (i.e., ‘production mode’). Thus, the ‘program’ is actually a Microsoft Visual Studio solution (.sln) file that is operated within the development environment (i.e., Visual Studio, version 2008+). The code base and history of modifications can be found at: <https://github.com/jon-carey/FRAMBuilder>.
* **The CTC’s CAS (and dll)**: CAS\_1.9.exe and CASLib.dll (2019 versions)[[2]](#footnote-2)
* **Visual Studio, version 2008+**: To operate FRAMBuilder ‘in the environment’ you will need a compiler; Visual Studio Express for desktops is a good free option (if Professional isn’t on your list of programs).
* **Others**:Although they aren’t tied explicitly to the mapping procedures outlined here, there are both R and OpenBUGS programs that estimate parameters for growth functions from CWT length observations (i.e., mapped to FRAM fishery and size limit regulation) summarized by FRAMBuilder. Jon Carey also has an R program (and input template/files) that estimates missing recoveries for freshwater sport (Puget Sound, Willapa Bay tribs) and estuary sport (Willapa Bay).

1. **Step 1: Select tag groups**

Although the rationale surrounding the final decisions to include/exclude tag codes is beyond the scope of this document, here we outline the basic guidelines used to select the codes contained in the current calibration dataset. Firstly, we preferentially selected CWT codes associated with CTC ER indicator stocks given that CTC members with regional expertise have already screened what’s available to best represent natural and hatchery Chinook stocks within their jurisdiction. For stocks/regions beyond the CTC’s scope, selection was guided by the following criteria/considerations:

* For nearly all stocks, tag selection was limited to brood years (BY) 2005-2008; additional brood years were included for special calibration analyses (e.g., out-of-base procedures relied on brood years 2002-2004, growth functions); and, due to the ‘collapse years’, Sacramento/Central Valley fall Chinook include 2009 brood releases and omit 2005-06 broods.
* Only stocks belonging to the ‘5000’ series of marks (i.e., adipose-fin clipped) were included; unmarked fish could not be used due to their absence in (1) CWT samples for visually sampled fisheries and (2) mark-selective fishery catches.
* We generally avoided CWT release groups (1) from ‘experimental’ production groups (e.g., novel stock crosses), (2) with questionable warning flags (e.g., BKD outbreak), (3) that were released at stages earlier than the fingerling stage, and/or (4) that were released at locations with difficult/poor escapement enumeration (e.g., acclimation ponds).

For further detail on codes selected for particular stocks, please refer to the Chinook FRAM stock profile spreadsheet, available for download at: <https://github.com/jon-carey/FRAMBuilder>.

1. **Step 2: Query RMIS for release/recovery data**

Given a set of codes, the next task is to query RMIS for the necessary release and recovery information. The online system requires a user account and can be accessed by visiting [https://www.rmpc.org](https://www.rmpc.org/) and selecting ‘RMIS Standard Reporting.’ Release data are acquired via the ‘Releases: Tagged Releases’ query form, which requires your tag list. When pasting in your list of tag codes, it is important to ensure that any leading zeros are maintained, otherwise the system will not recognize the tag codes. After entering the list of tags (Figure 5.1) and clicking ‘Retrieve’, you’ll have to choose the type of report (CSV in email or browser is preferred; Figure 5.2), and then specify what fields you’d like to see in the query results. For the purposes of loading the Filter Database, there’s a specific User List that you’ll need to use (Note: you can copy/paste the list from here):

tag\_code\_or\_release\_id

species

run

brood\_year

release\_location\_code

first\_release\_date

last\_release\_date

cwt\_1st\_mark

cwt\_1st\_mark\_count

cwt\_2nd\_mark

cwt\_2nd\_mark\_count

non\_cwt\_1st\_mark

non\_cwt\_1st\_mark\_count

non\_cwt\_2nd\_mark

non\_cwt\_2nd\_mark\_count

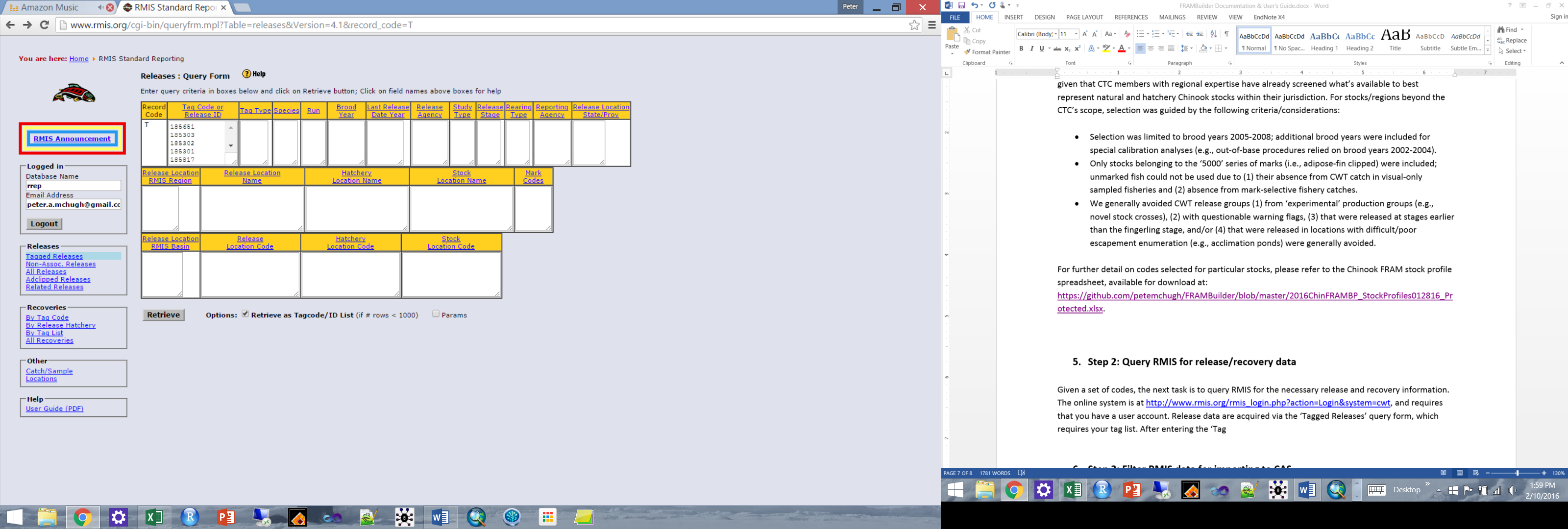
hatchery\_location\_code

stock\_location\_name

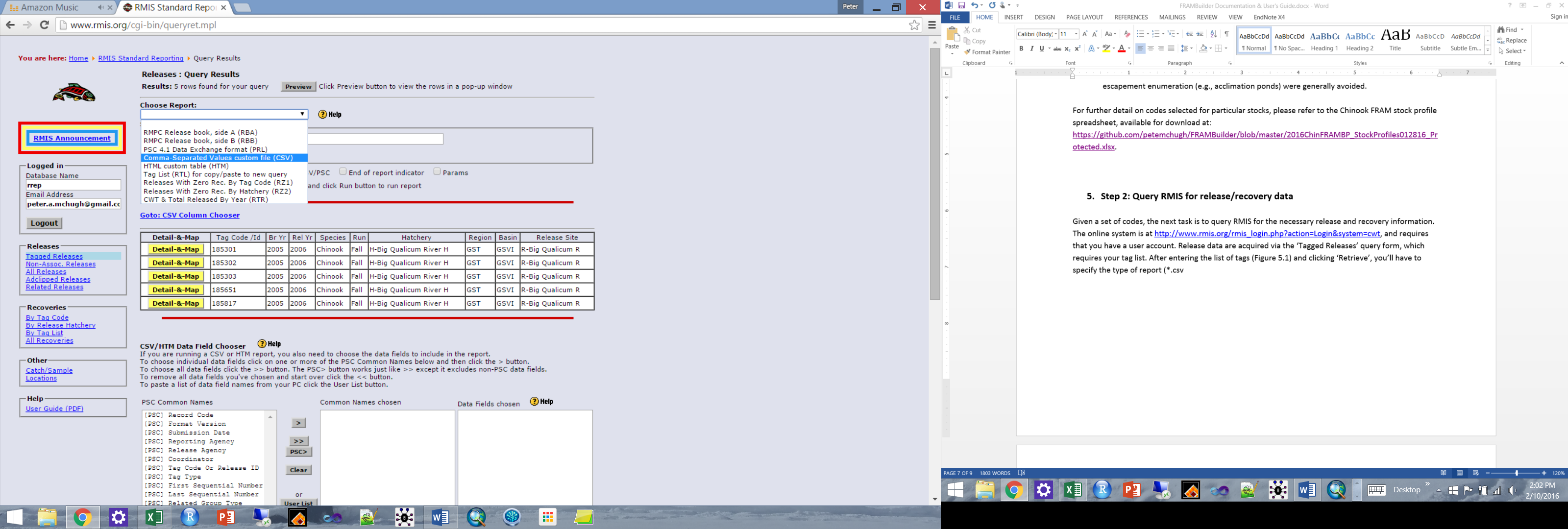
related\_group\_type

related\_group\_id

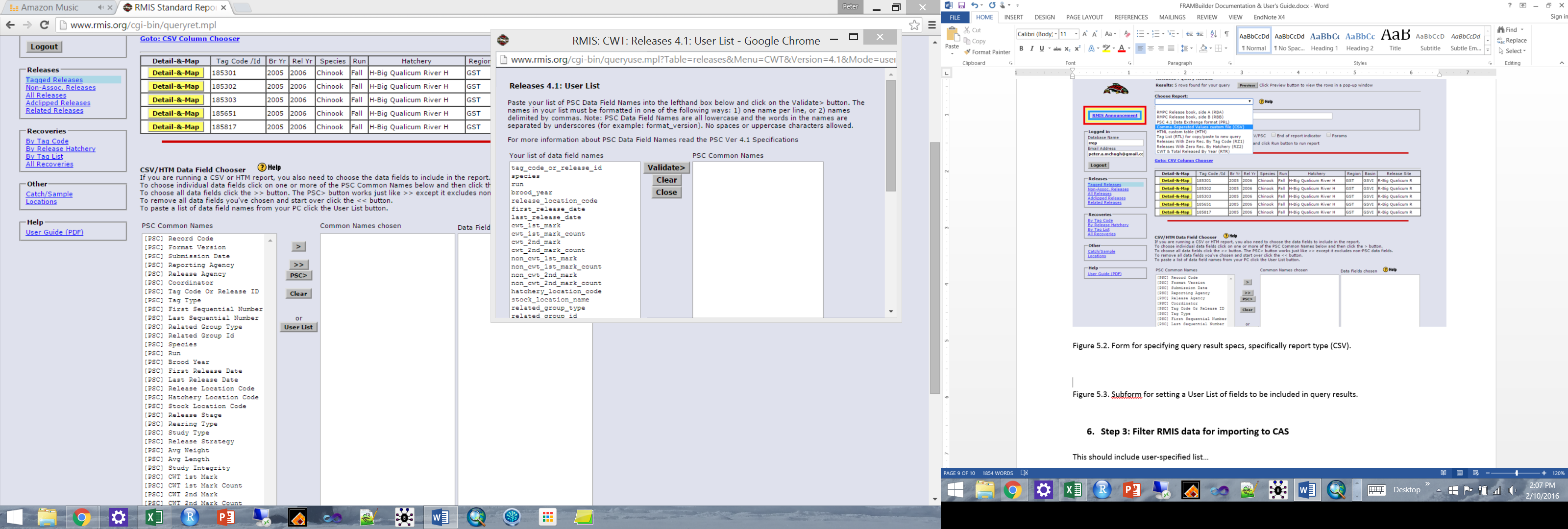
Upon validating the user list and clicking the ‘Run’ button, results will be returned to your screen/email; these, less the junk header (browser report only) text that reads ‘You are here: Home > RMIS Standard Reporting > Query Results > Report CSV’ can be copied/pasted into a text editor and saved for importing to the Filter Database.



**Figure 5.1.** RMIS’s ‘Tagged Releases’ query form, wherein the set of codes desired is specified.



**Figure 5.2.** Form for specifying query result specs, specifically report type (CSV).



**Figure 5.3.** Sub-form for setting a User List of fields to be included in query results.

The steps followed for running the custom ‘Releases: Tagged Releases’ query are now repeated for the ‘Recoveries: By Tag Code’ query, only this time, the custom User List will include the following fields:

recovery\_id

recovery\_date

period\_type

period

species

sex

length

length\_code

tag\_code

tag\_status

estimation\_level

recovery\_location\_code

fishery

estimated\_number

sample\_type

run\_year

recorded\_mark

catch\_sample\_id

detection\_method

sampled\_maturity

reporting\_agency

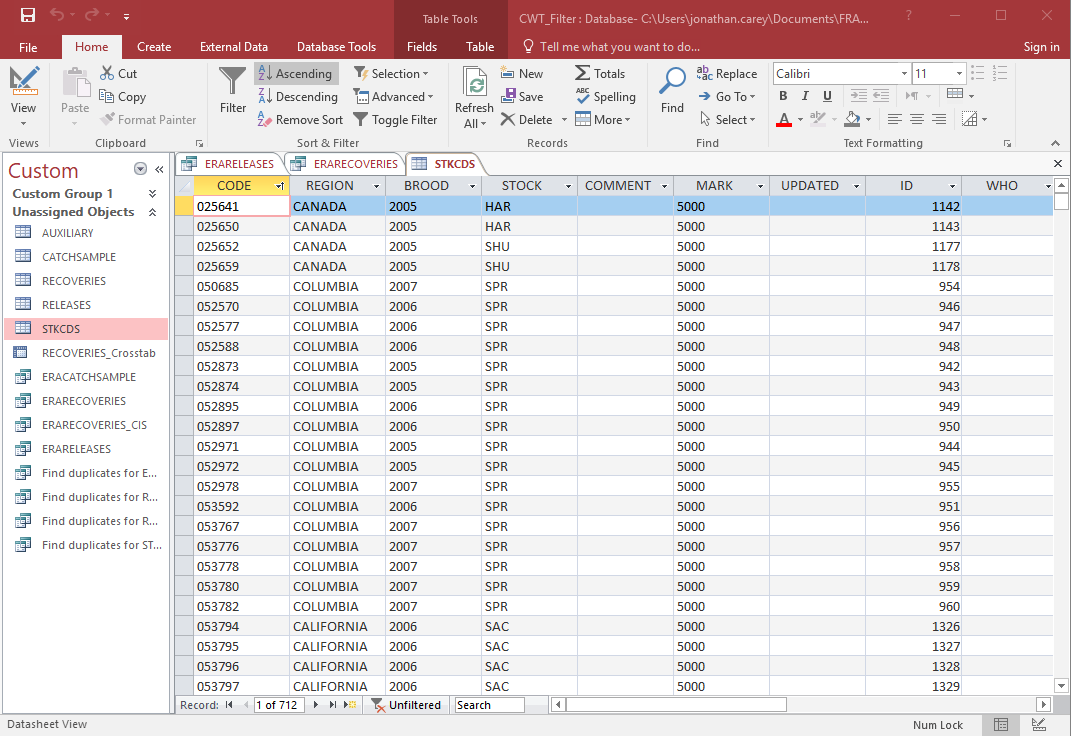
adclip\_selective\_fishery

Query results can now be saved as a CSV imported into the Filter Database. Be careful when opening/viewing these query results before importing into Filter Database; CSVs open readily in Excel and can be unintentionally reformatted upon opening (e.g., text to numeric will drop leading zeros on some codes).

1. **Step 3: Filter RMIS data for importing to CAS**

The release and recovery CSV files constructed in Step 2 can now be imported directly into the Filter Database. This is achieved via the Microsoft Access functions under External Data > Import & Link > Text File. This allows you to specify the appropriate Releases.CSV or Recoveries.CSV file in the ‘File Name field’ (browse and point to accordingly) and append records to the appropriate database table (RELEASES or RECOVERIES). To minimize potential error, it is generally advisable to start with a clean Filter Database for these steps.

After loading RMIS query results, the Filter Database’s ‘STKCDS’ table (Figure 6.1) must be populated with the tag codes contained in the release and recovery datasets. The three letter abbreviations in the STOCK field must follow existing conventions (i.e., be in the CAS ‘SpeciesStock’ table), otherwise records will be rejected [Note: although all FRAM stocks are already covered, new stocks can be added via steps contained in CTC CAS help files].

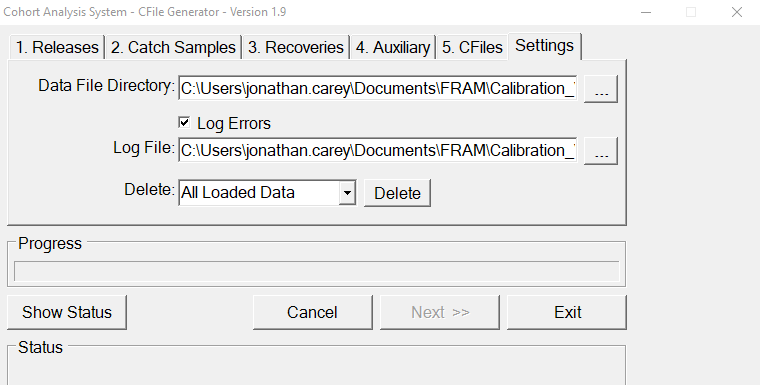
****

**Figure 6.1.** The Filter Database’s ‘STKCDS’ table, which requires tag code, jurisdiction, brood year, stock acronym, and marking status.

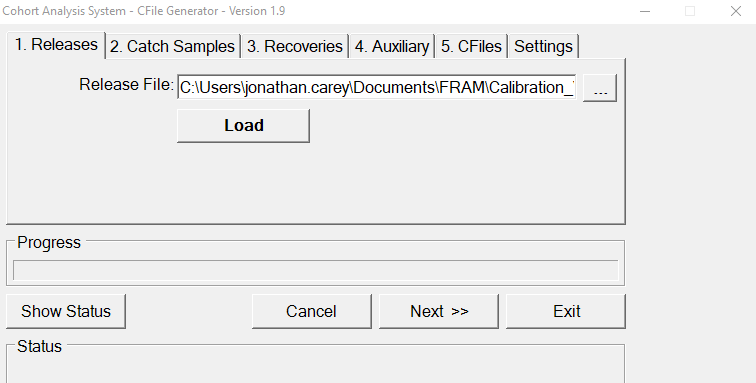
After the Filter Database is loaded with necessary content, these data are ‘filtered’ via two queries (releases, recoveries) that are saved as .txt files for importin to CAS via. The ‘ERARECOVERIES’ filter query eliminates (1) recovery observations that cannot be used in a quantitative analysis (i.e., RMPC ‘sample type’ = 5), (2) high seas fishery recoveries, and (3) some recoveries that are dealt with as auxiliaries (e.g., Canadian escapements), (4) the handful of recoveries in Alaska that occur outside of Southeast (i.e., location codes for all recoveries from Alaska should begin with ‘1M1…’); see Appendix A for SQL query details. The ‘ERARELEASES’ filter query, in contrast to recoveries, does minimal joining (STKCDS to Releases) and reformats the release data in prep for CAS import. Step 3 should yield two text files, ERARELEASES.TXT and ERARECOVERIES.TXT (you save under different names as you see fit).

1. **Step 4: Load filtered CWT data into CAS**

Using the CAS\_1.9.exe program (CAS program or CAS.EXE hereafter), the ERARELEASE.TXT and ERARECOVERIES.TXT files can now be imported into the FRAM-CAS database. Depending on the application, you may want to first clear out the entire FRAM-CAS database, using the CAS program (Figure 7.1). First, however, you must connect your CAS program to the FRAM-CAS database with which you’re working (you will be prompted to do so upon opening the program). Once connected, you can clear things out as needed (Figure 7.1) and then you’re ready to import filter database output releases (Figure 7.2), recoveries, and/or any auxiliary files (discussed further below). The procedure for each file type is the same, you first point the program to the appropriate release, recoveries, or auxiliary text file (.TXT or .CSV) and menu, then you click ‘Load’ and wait for the process to complete. Once CAS has finished loading the data, you should examine the running CAS error log (‘CASErrors.TXT’, this is written where the program occurs) and the ‘\*.BAD’ (a text file) associated with each input you’ve attempted to load—these will tell you which (if any) records were rejected and offer you clues regarding why this occurred.



**Figure 7.1.** The CAS program’s Settings menu, the location in which loaded data can be deleted.



**Figure 7.2.** The CAS program’s Releases loading menu.

Troubleshooting rejected releases and recoveries

Typically a handful, but sometimes many, records will be rejected during the FRAM-CAS database loading process. This can feel like the most cumbersome step of the FRAM-CAS database loading process, but with some patience and insight is relatively easy to work through. Common problems and solutions include (*solution in italics*):

1. For releases, this can arise because the stock code isn’t included in the ‘SpeciesStock’ database table (this shouldn’t happen, but if it does, see CTC help file guidance); *upon adding the new stock’s details, it should load correctly*.
2. For recoveries, the recovery location may not be parsing in such a way that can get a recovery from an RMIS location code all the way to a CTC fine-scale fishery; *in this case, you’ll have to manually add a parsed location (with fine-scale fishery mapping) to the FRAM-CAS database ‘FisheryLookup’ table*, *and reload the associated BAD records*; e.g.: RMIS Location Code: “3F10510 080122 R” parses to the fields in green, and you must supply the red info (Fishery = CTC Fishery Stratum, CWDBFishery = RMIS Fishery Code, Species = 1 for Chinook; gray fields can be left empty):

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Id | Fishery | CWDBFishery | StateProvince | WaterType | Species | Stock | Sector | Region | Area | Location | SubLocation | StartMonth | StartDay | EndMonth | EndDay | F17 | F18 | F19 |
|  | 1079 | 54 | 3 | F | 1 |  | 1 | 05 | 10 | 080122 | R |  |  |  |  |  |  |  |

The parsing rules from RMIS Location Code to field values are: Char(1) = StateProvince; Char(2) = WaterType; Char(3) = Sector; Char(4-6) = Region; Char(7-9) = Area; Char(10-16) = Location; Char(17-19) = Sub-Location.

1. In other cases, the recovery location might be parsing correctly, but a fishery really isn’t part of the FRAM (or CTC universe). In rare cases, for example, Chinook FRAM’s stocks have been recovered in net fisheries in Cook Inlet and Prince William Sound. *These rarities are not included* *and treated as though they’re ‘natural mortality’*.
2. On rare occasions, RMIS contains records with a valid sample type, but no ‘estimated number’ for a particular tag recovery. *In these cases, the user must decide what to do, i.e., to enter a value or omit altogether.* For example, if it’s likely a data error and in fact each fish represents an individual (e.g., censused escapement at a hatchery), then these records could be given an estimated number of 1.0 and reloaded. It is on the BP team, however, to make the call given whatever information is available (regional expertise/contacts, etc.).
3. A portion of the auxiliary data you’re supplying isn’t part of the tag set of interest. For example, a CTC member from Canada may supply escapements for all of their stocks for all years in a single file, whereas you’re only interested in 2005-2008 broods. *If this is the basis for rejection, there’s nothing more that you need to do*.

The good news is that, generally speaking, once a solution to a parsing problem or release rejection problem is resolved in the FRAM-CAS database, it will address all future instances subject to the same circumstances (and/or future reloads).

What are auxiliaries?

As noted above, so-called auxiliary CWT recovery files are also loaded into the FRAM-CAS database during this stage of the overall process, and are denoted as such in the ‘CWDBRecovery’ table (‘Auxiliary’ = T/F field). These files, which are supplied by regional experts who steward CWT recovery data for particular stocks or fisheries, are meant to either augment or correct the data acquired via RMIS for the tag groups in question. While the deficiencies/errors in RMIS content and/or the basis for auxiliary file creation are beyond the scope of this manual, these files are necessary to ensure that all observations of tagged model stock cohorts are correctly captured in the FRAM BP dataset. Because these datasets are routinely updated, it is important to acquire the latest[[3]](#footnote-3) versions from the appropriate CTC point(s) of contact prior to completing a full FRAM-CAS loading process. Finally, similar input files may need to be created for any non-CTC stocks necessitating similar supplementary or revisionary data (e.g., for unsampled freshwater sport fisheries in Puget Sound and Willapa Bay).

For further documentation on the CAS program and database, see the design specs document prepared by Wostman and Associates, Inc. for ADFG and the CTC, as well as the CAS help files located here: <https://github.com/jon-carey/FRAMBuilder/tree/master/CTC%20CAS%20Documentation>. For more information on the FRAM-CAS database, and more specifically how it has been modified to integrate FRAM functionality, see Appendix B.

1. **Step 5: Run FRAMBuilder**

Now that the FRAM-CAS database has been successfully loaded with the CWT data needed for running the BP calibration, we have just a few additional (albeit very important) steps to (1) complete the RMIS to FRAM fishery mapping work, (2) process the selected tags/broods for the selected stocks so they correspond to a single ‘super code’, and (3) format and write calibration input files to the FRAM-CAS database. This section covers the rationale and sequence of steps underlying this process.

Mapping/djustment rules

Before diving into the ‘how to’ details, here we summarize the main rules/processing tasks that FRAMBuilder executes to map CWT recoveries from CAS fishery purgatory to their final FRAM fishery/time step/age stratum (within code, see the subroutine ‘BGworker\_output\_DoWork’ in ‘OutputOptions.vb’ for further detail). Note that the following special mapping rules were revised in 2019 in accordance with updating from the 2013 CAS database and fishery structure to the 2019 CAS database and fishery structure. Whenever specific criteria invoke special processing rules, FRAMBuilder does the following:

1. Split CAS Puget Sound Areas 10/11/13 Net (Fishery = 2315) and Terminal Net (Fishery = 2316) into FRAM 10A, 10E, 10/11, 13A, and SPS Net fisheries; this is done using the RMIS recovery location code. Also assign 13C (Chambers) to freshwater net, consistent with TAMM treatment of the fishery.
2. Split CAS Puget Sound Areas 10/11/13 Sport (Fishery = 3311) and Terminal Sport (Fishery = 3315) into FRAM Areas 10, 10A, 10E, 11, and 13 Sport fisheries; this is done using the RMIS recovery location code. Recoveries are only assigned to 10A and 10E during July – September, consistent with fishery regulations. Note that due to suspected errors in data coding in RMIS, it is not possible to perfectly separate 10A/10E from general Area 10 in some base period years.
3. Assign recoveries mapped to CAS Puget Sound Area 8-2 Sport (Fishery = 3308) during the summer to FRAM 8D (Tulalip Bay) sport, as general Area 8 is closed to Chinook retention during the summer; this is done using the recovery month.
4. Assigned CAS Puget Sound Stilly/Sno Net (Fishery = 2313) and Terminal Net (Fishery = 2314) recoveries that occur in 8D (Tulalip Bay) to FRAM Tulalip Bay Net; this is done using the RMIS recovery location code.
5. Separate the KMZ component of CAS California Troll (Fishery = 1314) and assign to FRAM KMZ Troll; this is done using the RMIS recovery location code.
6. Separate the KMZ component of CAS California Sport (Fishery = 3331) and assign to FRAM KMZ Sport; this is done using the RMIS recovery location code.
7. Assign CAS Hood Canal Net (Fishery = 2318) and Terminal Net (Fishery = 2319) recoveries that occur in 12H (Hoodsport) to FRAM freshwater net, consistent with TAMM treatment of the fishery.
8. Correct Washington Area 1 Troll recoveries that have historically (incorrectly) been mapped (by CAS) to Oregon Area 3 Troll (Fishery = 1309).
9. Separate the combined Bellingham Bay (Areas 7BCD) treaty/non-treaty net fishery recoveries into separate treaty/non-treaty components. Unfortunately, this ‘processing rule’ can’t be fully implemented in code because a non-distinct (indiscernible treaty/non-treaty) fishery/gear code combination precludes assignment with certainty for a non-trivial percentage (>20%) of 7BCD recoveries. Thus, 7BCD net recoveries are pre-processed externally (see supplemental file ‘BellinghamBayEvaluation\_2019.xlsx’) to make individual tag ‘Tr/NT’ assignments based on fishery/gear codes in RMIS and/or annual treaty/non-treaty catch proportions (see Appendix C). Once this pre-processing step has been completed, FRAMBuilder will assign recoveries to the appropriate FRAM fishery based on the values entered in the ‘Tr\_NT’ field of the CWDBRecovery table.
10. Map recoveries assigned to CAS Washington Coast Net (Fishery = 2321) that occur in Area 4B to FRAM JDF net.
11. Map recoveries assigned to CAS Washington Area 2 Sport (Fishery = 3319) that occur in Willapa Bay and are of Willapa stock to freshwater sport (because the model does not contain a Willapa Bay sport fishery).
12. Map all freshwater fishery recoveries, i.e., sport, net, B10 sport, Col R Net, to the escapement fishery (fishery 74); this step facilitates calibration in ETRS units, as desired under the new BP framework.
13. Make and modify time step (TS) assignments as needed; TS assignments are made initially based on the calendar month in which a recovery occurred (month <4 & >=10 is TS1; 5&6 = TS2; 7-9 = TS3), but in several instances things will have to be shifted slightly to reflect the timing rules and biological assumptions built into the calibration’s cohort reconstruction. For instance, since fall Chinook are only allowed to mature in TS3, recoveries in freshwater sport/net/escapement during TS1/4 are assigned to TS3.
14. Determine the FRAM age equivalent for each fish. This is initially done as ‘age = run year - brood year’, with subsequent adjustments to reflect (a) fish having birthdays on Oct 1, (b) Willamette and Cowlitz/Kalama/Lewis spring Chinook (yearling releases) having ages treated as true age - 1 in FRAM[[4]](#footnote-4), and (c) min (age 2) and max (age 5) age constraints imposed by the model/cohort reconstruction.
15. Identify sublegal recoveries for exclusion, as they should not be used in the calculation of legal exploitation rates. This is done by identifying recoveries that have reported lengths (from RMIS) that are less than the lower bound of the size limit in the given fishery/time-step. For southern US and Alaskan fisheries, size limit regulations are in terms of total length. These are converted to fork length for modeling purposes, using the total length to fork length conversion from Conrad & Gutmann (1996). For fisheries where this conversion was applied, the lower bound of the size limit was set to 2.8 cm and based on the sum of the 95% prediction interval for the conversion (1.8 cm) and an estimated measurement error (1 cm). Size limit regulations for Canadian fisheries are in terms of fork length, thus the lower bound was set to 1 cm and represents only measurement error.

[*Note, whenever any of these special processing rules is invoked for a particular recovery, a record detailing related adjustments is added to the FRAM-CAS database’s ‘FRAM\_ProcessLog’ table*]

Merging & weighting

Beyond applying mapping/adjustment rules, a secondary function of FRAMBuilder is to merge CWT data for a given stock within selected BP broods and ultimately across broods to create a synthetic all-broods ‘super code’. The program is flexible and can accommodate a few different merging approaches. Within brood years, merging can be done either (1) on an unweighted basis (default) or (2) using user-specified weights which are generated on a stock/code/BY basis and added to the ‘FRAM\_Weights’ table. The first option treats all CWT release groups within a stock/BY as though they’re equivalent and simply lumps them together accordingly. Thus, if several hatcheries/groups make up a single FRAM stock, they’re essentially self-weighting within BYs as a function of the number of fish released and early marine survival. The latter option allows users to specify weights that up/down weight particular CWT codes/groups according to any rationale. For example, the Skagit spring yearling model stock is actually a composite fingerling/yearling stock for which user-supplied weights have been developed to achieve a 50:50 representation of the two life history variants in the pool of recovered CWTs.

Once codes are merged within BYs for a given stock, FRAMBuilder merges the within-BY-merged CWT recovery data across BYs using one of three approaches: (1) unweighted merging, (2) recoveries-weighted merging [the default], or (3) user-specified weighting[[5]](#footnote-5). The first option is equivalent to the default within-year method in that BYs are pooled as-is and thus self-weighting; this essentially means that BYs with better survival and/or abundance will have greater influence in the calibration dataset. The second and default option (‘recoveries weighted’) is designed to rescale recoveries (i.e., *R’*FAT-*b* = *Wb*\**R*FAT-*b*) within each brood year so that all BYs have equal influence on the calibration dataset, i.e., for brood year *b* the weighting/scaling factor (*Wb*) is computed as *Wb* = max(all *Rb*) / *Rb*, where *Rb* = *R*FAT-*b* and *Rb* and *R*FAT-*b* are the estimated recoveries total for brood year *b* and the fishery-age-time step-specific recoveries total within brood year *b*. As above, the last option permits the user to specify (again, within the ‘FRAM\_Weights’ database table) any weighting scheme that’s desired. This latter option might apply, for example, when there’s an anomalous year in the data series for a stock, exploitation or survival/abundance wise, that contains some useful information but needs to be down-weighted to avoid giving it undue influence on the final calibration dataset.

[*Sidebar Note, the default weighting scheme emulates what’s historically been assumed/done for Washington’s Puget Sound stocks within the PSC Chinook Model calibration procedure*]

Output file creation and database population

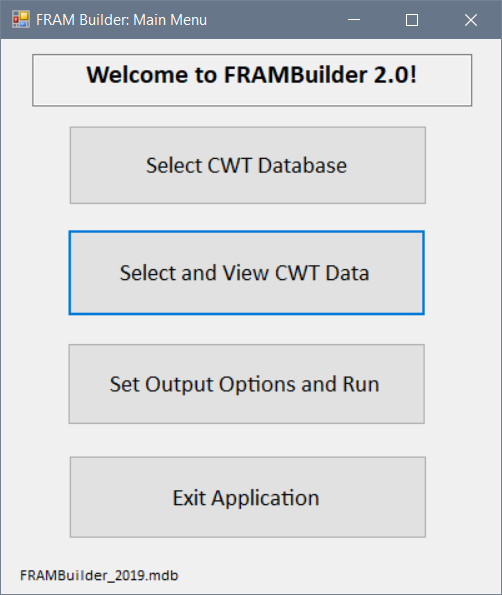
Once data processing is complete, FRAMBuilder writes several results to the FRAM-CAS database, with specific RunID attribution (a unique date-time based stamp). The main outputs of interest are:

1. The ‘FRAM-OUT\_CWTAll’ table; this contains the results for the processing run in a format (merged, etc.) that can be fed directly into the main calibration program[[6]](#footnote-6).
2. The ‘FRAM\_star\_CWT’ table; this contains mapped/summarized results for various stages in the processing sequence, i.e., mapped stock-fishery-age-TS totals by code (indicated by raw code ID), by BY (X*yy*.STK where *yy* and STK are the BY and 3-letter stock ID), and ‘super code’ (AB*mm*.STK, AB for ‘all broods’ and *mm* is the merging method ID and STK is as above)
3. The ‘FRAM\_ProcessLog’ table contains info about what special rules (listed above) were invoked in a given processing run for a particular recovery ID;
4. The ‘CWDBRecovery’ table, the main recovery table feeding the entire process, has a field (‘finalFmap’) that gets populated with the numerical ID for the fishery to which a given recovery was ultimately mapped; this result is quite useful for both error checking and post-run summarization (e.g., for other purposes).
5. If summary files for estimating growth functions are necessary (and this processing option is selected), FRAMBuilder will also write results to ‘FRAM\_GrowthData’;
6. If ‘old school’ outputs are needed (e.g., \*.CWT files, text based process logs, etc.), the checkbox allowing for this feature will need to be reactivated so that these text-based calibration inputs are created.

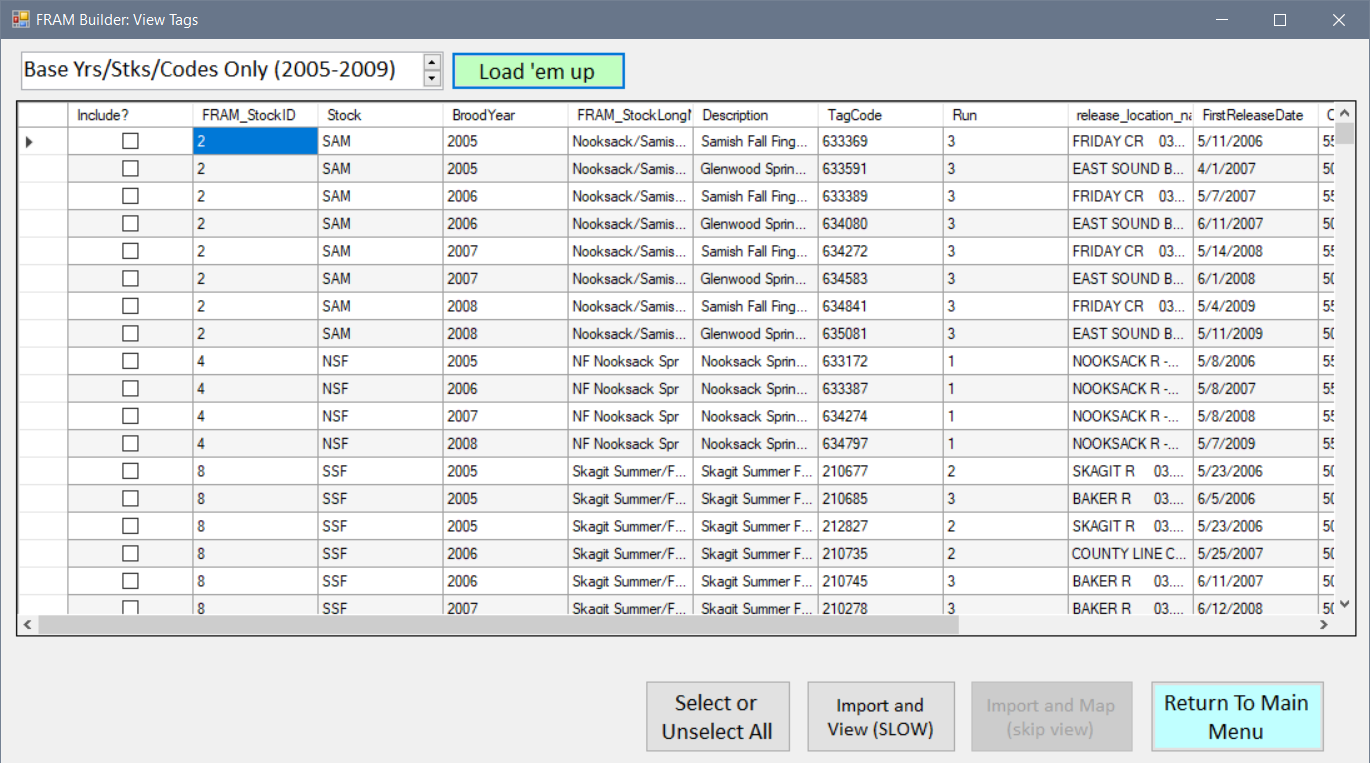
Running the full processing sequence

Now, with a clear understanding of what’s needed to get data from RMIS to FRAM-CAS in a state that’s ready for processing, it’s time to actually run the program. First, open the FRAMBuilder program (i.e., run it in debugging mode once you’ve opened the VB.NET solution[[7]](#footnote-7)) and connect it to your FRAM-CAS database by clicking ‘Select CWT Database’ when you reach the Welcome Screen (Figure 8.1).

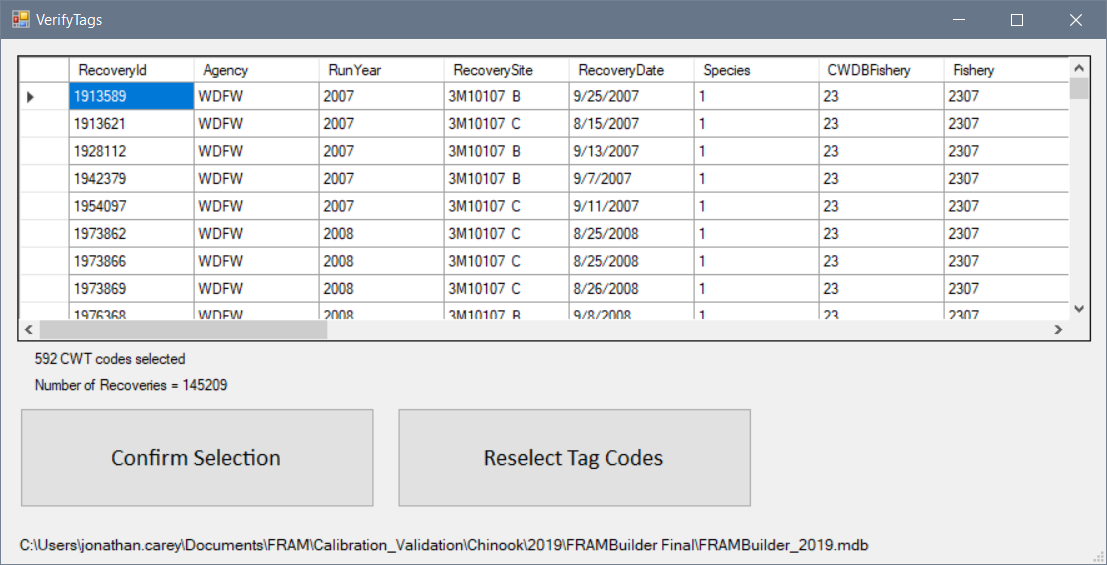
Second, once you’ve connected, click the second button (‘Select and View CWT Data’) on the welcome screen. This will take you to a form in which you’ll actually choose the specific CWT codes that you’re interested in mapping/processing for the calibration process (Figure 8.2). Upon selecting desired codes/stocks via this screen, you’ll get a chance to view raw recovery data (click ‘Import and view’ upon selecting) prior to initiating the processing sequence [*Note*, *this query-to-view step is rather slow (10s of seconds for a few 100K records), because things get bogged down query-wise on the MS Access side—be patient!*]. The next window (Figure 8.3) will show you the raw recoveries, which are now queued for processing; click ‘Confirm selection’ to return to the main menu, from where you’ll navigate to the processing menu by clicking ‘Set Output Options and Run’ (Figure 8.1).



**Figure 8.1.** FRAMBuilder welcome screen. Before doing anything you must select your database (uppermost button), from there the process follows sequentially down the list.



**Figure 8.2.** FRAMBuilder tag selection screen. Upon reaching this screen, the subset (if appropriate) of tags contained in the database desired for viewing/editing must be selected using the toggle list and ‘Load ‘em up’ button. Next, selecting particular codes is achieved by clicking check boxes (leftmost column, ‘Include?’) or by clicking select/unselect all. Thereafter, the FRAM-CAS database will be queried for the relevant content, which you can view prior to processing.



**Figure 8.3.** The view/verify codes screen, a quick stop between selecting codes and moving into processing the queued dataset. You can basically click ‘Confirm Selection’ as soon as you arrive here, or peruse data to your heart’s content.

Now you’re ready to set the specifications for the particular mapping and data processing/prep run, which you’ll do in the ‘Output Options’ form (Figure 8.4). For most runs, you’ll want to process things according to the defaults, which are already checked when the form loads. The output options include:

Merging and Mapping Options:

* ‘Create mapped table in DB’ [Default = True]. [note that this is in contrast to the old ‘\*.CWT’ calibration file format, which can still be created using a disabled feature]
* ‘Merge Codes w/in BY’ [Default = True].
* ‘Merge Codes b/n BY’ [Default = True].
* ‘Use db wts + rules for w/in BY merge?’ [Default = False].
* ‘Use db wts + rules for b/n BY merge?’ [Default = False].

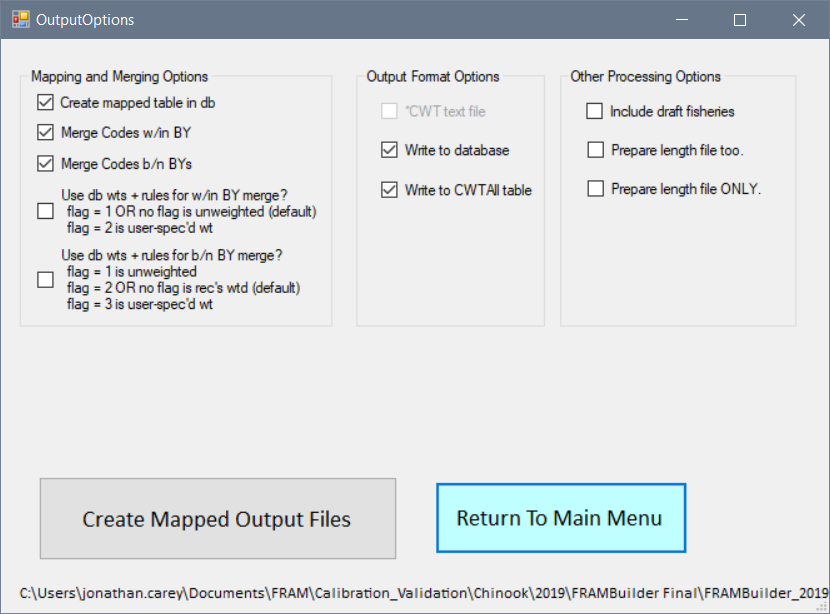
Output Format Options:

* ‘Write to database’ [Default = True]. [Note, this should generally be left checked, perhaps even locked, as without it and the \*.CWT out file format being deactivated, the mapping sequence will produce little of use]
* ‘Write to CWTAll table’ [Default = True]. [Note, this should generally be left checked, perhaps even locked, as without it and the \*.CWT out file format being deactivated, the mapping sequence will produce little of use]
* [**INACTIVE**] \*.CWT text file. [this is the old calibration program’s preferred format]

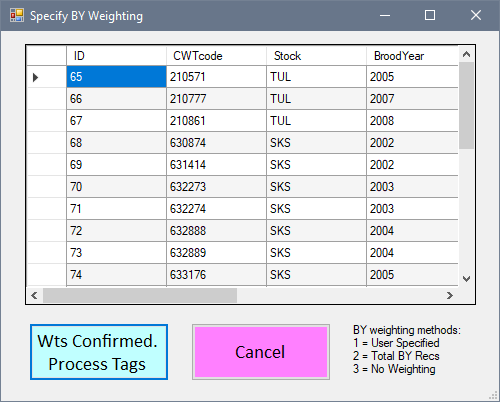
Other Processing Options:

* ‘Include draft fisheries’ [Default = False]. [Note, this was a field we added to inventory and explore possibilities of breaking apart a handful of fisheries, e.g., 3/4/4B troll into separate 3 and 4/4B fisheries; if it’s invoked, it includes both the original fishery mapping, plus the experimental mapping]
* ‘Prepare length file too’ [Default = False].
* ‘Prepare length file ONLY’ [Default = False]. [use this option if you just want to use FRAMBuilder to create summary files for estimating growth function parameters]

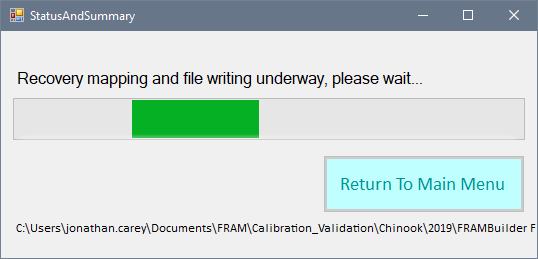
Once you’ve selected the specifications desired for the run, click ‘Create Mapped Output Files’ on the Output Options form to initiate the mapping/merging/output creation procedures described above. If you have checked either the ‘Use db wts + rules for w/in BY merge?’ or ‘Use db wts + rules for b/n BY merge?’ boxes, a dialog box will appear displaying the user specified flags and weights provided in the ‘FRAM\_Weights’ table of the database (Figure 8.5). To confirm these values, click the ‘Wts Confirmed. Process Tags’ button. Following this (or if you did not select either of the two above boxes) the view will shift to the ‘StatusAndSummary’ form screen (Figure 8.6) which will display a scrolling green bar until the run is complete, at which point it will shift to a status of ‘Processing complete’ (Figure 8.6). Note that the full processing sequence, although conceptually simple, may take several minutes when it’s run for the entire base period dataset at once; again, this is due to the slow nature of MS Access queries (of which there are multiple in this sequence) in VB.NET applications.

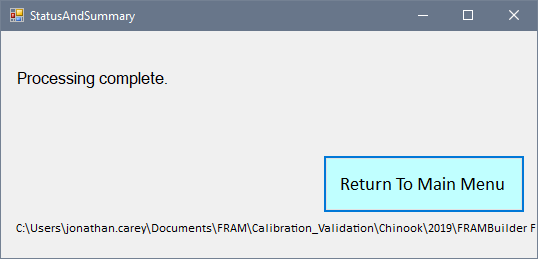


**Figure 8.4.** The set output specifications screen—the last stop before creating calibration inputs.



**Figure 8.5.** The weighting confirmation screen which displays tag-code-specific flags and weights included in the ‘FRAM\_Weights’ table of the database.





**Figure 8.6.** The ‘Be Patient I’m Working’ processing screen (top panel) and what you’ll see when things are complete (bottom panel).

1. **Step 6: Export data**

Aside from generating old-format (.CWT) text files and populating the FRAM-CAS database, FRAMBuilder does not produce any files extending beyond these stages. Thus, the end user must work directly within the FRAM-CAS database in order to acquire/use processing results in subsequent calibration procedures. The ‘FRAM-Out\_CWTAll’ and ‘FRAM\_star\_CWT’ tables contain the primary data of interest to the Main Calibration Program. To extract results for further use, first filter data on the FRAMBuilder run of interest (filter on ‘ID’ field) and simply copy-paste it into an Excel spreadsheet (or export in any format as desired). The current template for this is called ‘Calibration\_CWT\_Inputs; MM.DD.YY.xlsx’ and includes a few notes about the post-processing steps needed (see Appendix C for details) to generate a final dataset for input into the Main Calibration Program’s database. Otherwise, further review of processing results can be achieved by looking at the raw recovery level (‘CWDBRecovery’ table) or by viewing mapped code-, merged-within-BY-, or merged-across-broods-specific results in the ‘FRAM\_star\_CWT’, which correspond to the pieces that have been rolled up into the final composite ‘FRAM-Out-CWTAll’ table. Beyond these components, the length-at-age prep (‘FRAM\_GrowthData’) file can be exported for further processing/review in Excel, R, etc.

1. **FRAMBuilder’s growth function input file preparation feature**

In addition to creating input files for the Main Calibration Program, FRAMBuilder also has the ability to prepare files for use in estimating the model’s stock-specific von Bertalanffy growth functions (VBGFs). The approach used to estimate VBGF parameters requires (1) that fish lengths are standardized to a particular format (i.e., fork length), (2) an estimate of a fish’s age in months (not FRAM age) at the time of recovery, and (3) knowledge of size limit regulations for fishery-dependent recoveries, for each individual CWT recovery for which length has been measured. The first two requirements are fulfilled through a series of calculations/processing rules in code, whereas the latter is achieved through the help of an addition database table (FRAM\_SizeLimits) that is populated with the historic series of size limits for each time/area/run year fishery stratum. Because item #3 also has a mapping dependency (i.e., length limit in fishery), these processing steps are also implemented within the ‘BGworker\_output\_DoWork’ subroutine in ‘OutputOptions.vb’.

For further detail on the analysis methods, etc., refer to the PFMC analysis review document at <http://www.pcouncil.org/wp-content/uploads/2015/10/D2_Att2_FRAM_Growth_Meth_Nov2015BB.pdf>.

1. **Limitations to FRAMBuilder and opportunities for enhancement**

Like any program that’s been designed to automate and streamline an unwieldy and complex data processing task that necessarily requires some expert oversight and occasional ad hoc revisions, FRAMBuilder is far from a distributable, production-grade program. Rather, it’s a tool for streamlining one of the more cumbersome tasks that the BP calibration team has to complete in order to create a new base period dataset. Accordingly, there are (and will always be) opportunities for enhancing the program, both in basic ways that pertain to its current structure/algorithms and through the addition of new/different processing features. A few low-hanging fruits that may be worth considering are:

* Offer a means to do some of the post-processing required to replicate ‘FRAM-OUT\_CWTAll’ content for some stocks as required for surrogate or other purposes (e.g., NF Nooksack for both SF and NF; also in surrogate cases; see Appendix C for more on these).
* Automate, to the extent possible, the processing required to split 7BCD Net data into treaty and non-treaty components; the current external procedure, although straightforward, opens the door to errors and is easy to forget if new databases are being updated/created. This workflow could be modeled after the companion spreadsheet described in Appendix C.
* Automate the development of ‘user-supplied’ weights for within BY merging; as described in Appendix C in the case of Skagit spring fingerlings and yearlings, the process is more complicated than one might think (i.e., you’re working backwards to determine needed weights/scalars to achieve a target 50:50 representation in CWT recoveries).

Beyond these minor enhancements, IF calibration becomes a routine/updateable process, some bigger-picture enhancements worthy of consideration are:

* (higher priority) Update the FRAM to CAS linkage so that FRAMBuilder is not stuck in static ‘old CTC’ [currently 2019] fishery mappings; this could be achieved in a number of ways, ranging from relatively simple (e.g., update crosswalk table) to more complicated (e.g., integrating CAS.exe’s parsing/mapping code directly into FRAMBuilder to ‘cut out the middle man’); the former probably makes the most sense so that annually developed auxiliaries are readily usable in the FRAMBuilder workflow. (see also Appendix D notes on this subject)[[8]](#footnote-8)
* (low priority, given how infrequently calibrations occur) Marry FRAMBuilder directly to the Main Calibration Program and the FRAM-CAS content to the Calibration Database’s architecture (or the reverse). This would streamline things on the implementation front, which may not be a huge priority since calibrations occur infrequently.
* (low priority, given how infrequently calibrations occur) Identify ways (MS Access query structure, program architecture, or otherwise) to speed the program up. As noted previously, this isn’t worth losing much sleep since it’s not THAT slow in the grand scheme and calibrations occur intermittently at best. But something to ponder anyway…

**Appendix A. CTC Filter Database Recoveries Query Details/notes**

SELECT RECOVERIES.recovery\_id, RECOVERIES.recovery\_date, RECOVERIES.period\_type, RECOVERIES.period, RECOVERIES.species, RECOVERIES.sex, RECOVERIES.length, RECOVERIES.length\_code, RECOVERIES.tag\_code, RECOVERIES.tag\_status, RECOVERIES.estimation\_level, RECOVERIES.recovery\_location\_code, RECOVERIES.fishery, RECOVERIES.estimated\_number, RECOVERIES.sample\_type, RECOVERIES.run\_year, RECOVERIES.recorded\_mark, RECOVERIES.catch\_sample\_id, RECOVERIES.detection\_method, RECOVERIES.reporting\_agency

FROM RECOVERIES INNER JOIN STKCDS ON RECOVERIES.tag\_code = STKCDS.CODE

*# across all subsets, the ‘not like 1M1DF\*’ and ‘not like 1M2\*’ statements filter out AK recoveries outside of Southeast #*

*# across all subsets, sample\_type <> 5 excludes voluntary recoveries with no awareness #*

*# no high seas recoveries included for WA/ColR/OR/CA {164,878 recoveries} #*

WHERE (((RECOVERIES.recovery\_location\_code) Not Like '1M1DF\*' And (RECOVERIES.recovery\_location\_code) Not Like '1M2\*') AND ((RECOVERIES.fishery)<"60") AND ((RECOVERIES.sample\_type)<>"5") AND ((STKCDS.REGION) In ("WASH","COLUMBIA","OREGON",”CALIFORNIA”)))

*# no Canadian escapements (they’re entered as auxiliary data) {3,164 recoveries} #*

OR (((RECOVERIES.recovery\_location\_code) Not Like '1M1DF\*' And (RECOVERIES.recovery\_location\_code) Not Like '1M2\*') AND ((RECOVERIES.fishery)<"50") AND ((RECOVERIES.sample\_type)<>"5") AND ((RECOVERIES.reporting\_agency)="CDFO") AND ((STKCDS.REGION)="CANADA"))

*# [not relevant as FRAM contains no stocks originating from AK] For AK, include standard fisheries and marine trap (net) #*

OR (((RECOVERIES.recovery\_location\_code) Not Like '1M1DF\*' And (RECOVERIES.recovery\_location\_code) Not Like '1M2\*') AND ((RECOVERIES.fishery)<"49" Or (RECOVERIES.fishery)="94") AND ((RECOVERIES.sample\_type)<>"5") AND ((RECOVERIES.reporting\_agency) In ("ADFG","NMFS")) AND ((STKCDS.REGION)="ALASKA"))

*# recoveries of Canadian stocks outside of Canada {1,847 recoveries} #*

OR (((RECOVERIES.recovery\_location\_code) Not Like '1M1DF\*' And (RECOVERIES.recovery\_location\_code) Not Like '1M2\*') AND ((RECOVERIES.fishery)<"60") AND ((RECOVERIES.sample\_type)<>"5") AND ((RECOVERIES.reporting\_agency)<>"CDFO") AND ((STKCDS.REGION)="CANADA"))

# *[not relevant as FRAM contains no stocks originating from AK] #*

OR (((RECOVERIES.recovery\_location\_code) Not Like '1M1DF\*' And (RECOVERIES.recovery\_location\_code) Not Like '1M2\*') AND ((RECOVERIES.fishery)<"70" Or (RECOVERIES.fishery)="94") AND ((RECOVERIES.sample\_type)<>"5") AND ((RECOVERIES.reporting\_agency) Not In ("ADFG","NMFS")) AND ((STKCDS.REGION)="ALASKA"))

*# recoveries of Canadian stocks in SUS test fisheres {0 recoveries} #*

OR (((RECOVERIES.recovery\_location\_code) Not Like '1M1DF\*' And (RECOVERIES.recovery\_location\_code) Not Like '1M2\*') AND ((RECOVERIES.fishery)>="60" And (RECOVERIES.fishery)<"70") AND ((RECOVERIES.sample\_type)<>"5") AND ((RECOVERIES.reporting\_agency) Not In ("ADFG","NMFS","CDFO")) AND ((STKCDS.REGION)="CANADA"))

*# recoveries of SUS stocks in SUS test fisheres {68 recoveries} #*

OR (((RECOVERIES.recovery\_location\_code) Not Like '1M1DF\*' And (RECOVERIES.recovery\_location\_code) Not Like '1M2\*') AND ((RECOVERIES.fishery)>="60" And (RECOVERIES.fishery)<"70") AND ((RECOVERIES.sample\_type)<>"5") AND ((RECOVERIES.reporting\_agency) Not In ("ADFG","NMFS","CDFO")) AND ((STKCDS.REGION) In ("WASH","COLUMBIA","OREGON",“CALIFORNIA”)));

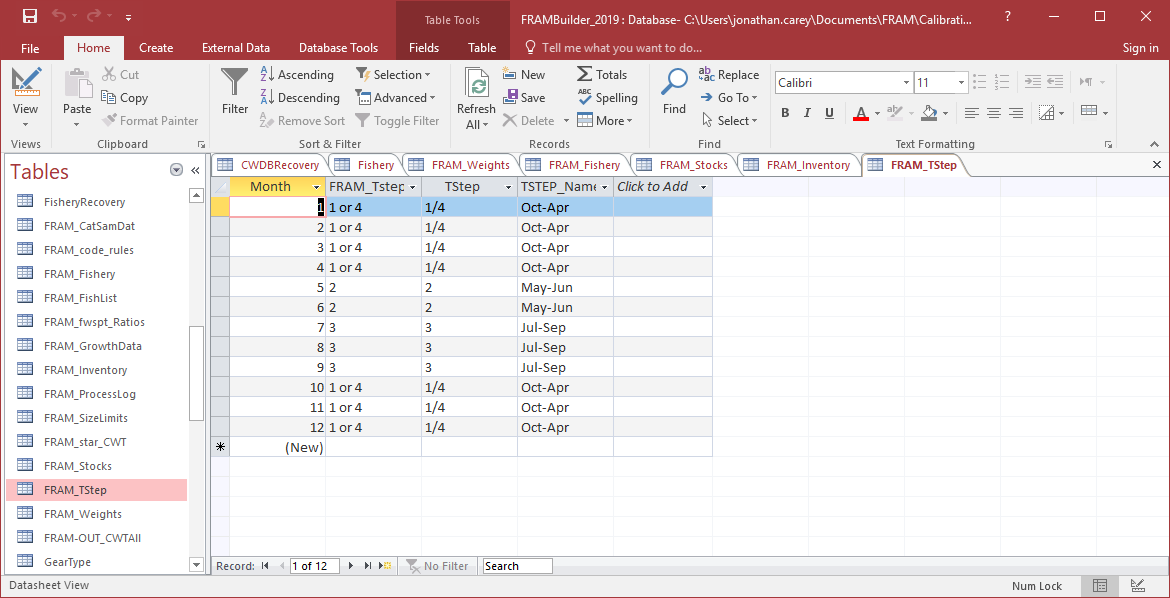
**Appendix B. Overview of the FRAM-modified CAS database**

The FRAM-CAS database (and filter database) is an adaptation of the PSC Chinook Technical Committee’s 2019 final preseason CAS database (available through the CTC sharepoint site [Analytical Work Group -> Exploitation Rate Analysis -> 2019], however, this requires a username and password – feel free to contact your friendly neighborhood CTC member for help). Thus, it’s rooted in the CTC’s fishery mappings in effect at that time (e.g., maybe escapement code = 4008 is a memory jogger for those who follow details of CTC fishery mappings). Beyond what that base CAS contained, several tables were added to fill FRAMBuilder’s mapping needs. All of these have a naming convention of ‘FRAM\_...’ so that they can be easily distinguished from the base CAS content (Figure B1). While the details associated with these additions and modifications are summarized here, the user is referred to the CTC’s CAS documentation for detail on the other database content/schema, etc., which is available at:  [<https://github.com/jon-carey/FRAMBuilder/tree/master/CTC%20CAS%20Documentation>](https://github.com/jon-carey/FRAMBuilder/blob/master/CTC%20CAS%20Documentation/CAS%20Design%20Spec.doc)

The main database tables that help FRAMBuilder in either the CAS-to-FRAM fishery or stock crosswalk/mapping process are:

1. ‘FRAM\_Fishery’ – this table provides the FRAM equivalent fishery or fisheries (there are some one-to-many cases to the CTC fishery strata) that records are initially mapped to when loaded using CAS.exe; this table is crucially important to FRAMBuilder’s ability to leverage CTC resources to get CWTs ultimately to FRAM fisheries. [*Note, if there’s ever a desire to update FRAM-CAS to a new set of CTC fishery mappings, a similar crosswalk will need to be built off of the ‘Fishery’ table in a newer version of CAS and code changes (within special rules) will also be required*].
2. ‘FRAM\_Stocks’ – this table is the primary means to collapse CTC (or other) stocks from finer stocks (e.g., specific hatcheries) into FRAM stock aggregates, as appropriate. The ‘FineStock’ field here corresponds to the ‘Stock’ field in both of the CAS’s ‘WireTagCode’ and ‘SpeciesStock’ tables; ‘Stock’ in ‘FRAM\_Stocks’ corresponds to higher-level aggregates to which ‘FineStocks’ will be collapsed (e.g., GRN, Green River is part of ‘SPS’, etc.); note also that this table provides (a) the numerical mapping of stock codes to numerical FRAM stock ID values and (b) a means to filter what’s accessed (‘Include’ field) during data viewing/processing via FRAMBuilder.
3. ‘FRAM\_Weights’ – this table contains the user-specified weights and flags (wnBYmeth 1 = unweighted, 2 = user specified; bnBYmeth 1 = unweighted, 2 = recoveries weighted, 3 = user specified) for any CWT codes that will be modified according to user-specified weights during either the within- or between-brood merging process. Whenever user-specified rules are desired, values must be provided for all codes or the procedure will crash; [*Note, this table only needs information for an individual stock that requires special weighting; defaults are applied if left alone for all other stocks/codes*]

Beyond these key mapping tables, FRAMBuilder’s key output tables are described in the main document and further identified visually in Figure B1. Note also, however, that the FRAM-CAS database tables ‘CWDBRecovery’ and ‘WireTagCode’, both base tables from the original CTC CAS database, include new fields that permit either processing or summarization. ‘CWDBRecovery’ includes the fields ‘finalFmap’ and ‘Tr\_NT’ which were described previously; ‘WireTagCode’ includes the Boolean field ‘FRAM\_OOB’ which is used to identify which codes/releases should be treated as Out-of-Base codes and ‘BP\_Notes’ which is available for adding notes indicating special attributes about a particular code; the ‘BP\_Stock’ field is available for additional notes but to date has not been used. Lastly, there are a handful of ‘FRAM\_...’ tables included that were either initially created for lookup purposes or to fulfill add-on needs. These include: (1) FRAM\_CatSamDat, (2) FRAM\_code\_rules, (3) FRAM\_fwspt\_ratios, and (4) FRAM\_TStep. And there’s a single table created by a make table query (Petes FRAM Stock Inventory, plus variations for different years) for inventory purposes called ‘FRAM\_Inventory’; this is deleted and recreated every time the query is run, so delete as you like.



**VERY IMPORTANT**; this is the table that contains the CTC -to-FRAM fishery crosswalk.

This is an unused table that houses RMIS Catch Sample data; at one point we were hopeful that we could use these data to derive base period catches for BP years.

This is an unused table that included rules for the automated generation of auxiliary recoveries, in conjunction with ‘FRAM\_fwspt\_Ratios’.

This table has fishery names, IDs, mat/mixed-mat, details for fisheries & draft fisheries.

This table is unused (see related note under FRAM code rules above).

This OUTPUT table gets populated with the mapped length-at-age file for use in analyses supporting calibration/BP growth function needs.

This is a table that’s created by a make-table query (‘Petes FRAM Stock Inventory’) that inventories observed tags by fishery, stock, time step, etc…it can be deleted at any time.

This OUTPUT table details records that invoked ‘special’ processing rules during the processing sequence.

**IMPORTANT** table containing a record of historic size limits for growth data processing.

**VERY IMPORTANT** table containing the mapping and aggregation details for individual tag codes, stocks, brood years, and ‘super codes’ that inform ‘FRAM-OUT\_CWTAll’ output.

**VERY IMPORTANT** table that (a) defines the linkage between CTC stock acronyms (fine stock) and FRAM stocks (stock & FRAM\_stockID), and (b) identifies whether or not particular stocks are accessible during processing (‘Include’).

This is an unused table that was initially created for lookup purposes.

**VERY IMPORTANT** table containing the user-specified weights for any within-BY or between-BY merging that’s to be done according to externally computed weights, scalars, or other modification ratios.

**VERY IMPORTANT** table containing what’s needed for the Calibration input file, formatted according to the specs required.

**Figure B1.** FRAM-related tables that have been added to the CAS database to support FRAMBuilder’s mapping/processing tasks.

**Appendix C. List of “Don’t Forget” external data pre-processing steps**

In no particular order, this is a list of a few processing/data modification steps that need to be taken before data can be fully ground through the FRAMBuilder mill.

1. There are a number of ‘post-FRAMBuilder’ processing steps that need to occur after the successful completion of a FRAMBuilder run but before the CWTAll and CWT\_OOB outputs get loaded into the Calibration support database. These steps are detailed in the accompanying spreadsheet titled ‘Calibration\_CWT\_Inputs; MM.DD.YY.xlsx’, available at <https://github.com/jon-carey/FRAMBuilder>. These steps include, but are not necessarily limited to:
   1. Duplicated stock data (NFNK Sp for SFNK Sp, SkagFF for Skag FYr, SPS FYr for UWAcc)
   2. JDF Surrogate CWT data (modified from STIL)
   3. Imputed Age 5 escapement
   4. Tulalip Net recoveries from TS2 pushed into TS3
   5. Modified 10E Net recoveries (to achieve 97% MPS contribution)
   6. Modified CWT recoveries to address unrealistic Age 5 BPERs
   7. Imputed WA North Coast recoveries in Grays Harbor Net
   8. White River OOB escapement tags apportioned into TS2 and TS3
   9. Modification to White R recovery in 8-1 sport with unrealistic expansion
2. Any auxiliary records (particularly those generated by the CTC) obtained in 2020 or beyond will potentially require reverting fishery mappings to the 2019 CAS fishery strata, for compatibility with the ‘FRAM\_Fishery’ lookup tabe.
3. To create an LCN stock base CWT file, a composite Oregon and Washington LRH stock CWT input files (releases, recoveries) needs to be duplicated and given dummy codes to be used to parameterize the LCN stock; this modification applies to both data downloaded from RMIS AND Auxiliary files involving LRH fish.
4. User-specified weights need to be generated to create the 50:50 merged Skagit spring fingerling/yearling stock (stock ID #12); an example of how this is done is contained in the supplied companion spreadsheet called ‘SkagitSp\_FingYrlWeighting\_Rnd7\_5.14.2019.xlsx’, which is available at <https://github.com/jon-carey/FRAMBuilder>. Note also that the ‘Stock’ and ‘FRAM\_StockID’ fields in ‘FRAM\_Stocks’ need to be concurrent for the two stocks (e.g., assign ‘Stock’ = SKS and ‘FRAM\_StockID’ = 12 for both).
5. Perhaps the most clunky (annoying?) external data preparation step (which only has to happen once, so quit complaining…), you will need to populate a field within ‘CWDBRecovery’ for all Areas 7BCD Net fishery recoveries so that Treaty/Non-treaty recoveries can be distinguished during processing [note, aside from Treaty Troll and 7BCD, no attempt is made to parse Treaty/NonTreaty within FRAMBuilder[[9]](#footnote-9)]. Unfortunately, this cannot be automated within FRAMBuilder because CAS does not get loaded with sufficient information to make the distinction, where such a distinction is possible, and a non-trivial percentage of 7BCD recoveries cannot be assigned even if CAS had all of the RMIS data on earth (~30% of recoveries receive a non-descript net fishery recovery code, one that’s not attributable to a fisher type); thus, within the external preparation spreadsheet, the original 7BCD data loaded into FRAM-CAS are removed, parsed out (by additional details, either supplementary RMIS data or catch fractions), and placed back into CWDB with a newly populated ‘Tr\_NT’ field. Note that recoveries that cannot be distinguished are split fractionally based on run year catch fractions (i.e., treaty % of landed, non-treaty % of landed) which necessarily means that those records have to be duplicated and have their associated recovery total split accordingly; this is all completed and illustrated in the companion spreadsheet file ‘BellinghamBayEvaluation\_2019.xlsx’ which can be downloaded at <https://github.com/jon-carey/FRAMBuilder>.

**Appendix D. Updates for compatibility with 2019 CAS fishery strata**

In 2019 work was completed to update both the FRAM-CAS database and the FRAMBuilder 2.0 application code for compatibility with the 2019 version of the CTC’s CAS database and fishery structure. The impetus for this update stemmed from the desire to re-query recovery data from RMIS in 2019, as there were numerous known (and likely numerous unknown) updates that had occurred since the data were previously queried in ~2014 (i.e., updates to Columbia River Summer Chinook escapement expansions, GMR updates to Stillaguamish escapement expansions, etc.). Additionally, it had been a number of years since the CTC’s auxiliary data had been updated. With that, the decision was made to start from scratch, building off of the CTC’s final preseason 2019 CAS database. The steps were as follows:

[Note: these steps are also detailed in the file ‘*FRAMBuilder DB Updates\_2019.xlsx,*’ which, in addition to the other files referenced below, can be downloaded at <https://github.com/jon-carey/FRAMBuilder>]

1. Obtain the complete list of base period tag codes from the most recently used version of FRAM-CAS database (in this case, ‘*2016 FRAMBuilder\_Rnd6.mdb*’).
2. Query RMIS for the necessary release and recovery data, per the instructions contained in Section 5 (Step 2) of the main body of this document.
3. Obtain a blank filter database, load in the release and recovery data, and update the ‘STKCDS’ tables as necessary. Run and export the ‘ERARELEASES’ and ‘ERARECOVERIES’ queries, per the instructions in Section 6 (Step 3) of the main body of this document.
4. Note that there are 64 recovery records in the resulting ‘ERARECOVERIES’ table that have sample\_type = 2, recovery\_location\_code that begins with ‘6F…’ and no estimated\_number. The estimated\_number for these records was updated to 1.
5. Obtain a copy of the 2019 final preseason CAS database, rename as desired (2019 FRAM-CAS herafter) and delete all records from the ‘CWDBRecovery’ and ‘WireTagCode’ tables.
6. Copy the following tables from the previous FRAM-CAS database (see step 1) into the 2019 FRAM-CAS database:

|  |  |  |
| --- | --- | --- |
| * FRAM\_CatSamDat |  | * FRAM\_GrowthData |
| * FRAM\_code\_rules |  | * FRAM\_Inventory |
| * FRAM\_Fishery |  | * FRAM\_ProcessLog |
| * FRAM\_FishList |  | * FRAM\_SizeLimits |
| * FRAM\_fwspt\_Ratios |  | * FRAM\_star\_CWT |
| * FRAM\_Stocks |  | * FRAM-OUT\_CWTAll |
| * FRAM\_Tstep |  | * RelLoc |
| * FRAM\_Weights |  |  |

1. Add the following record to the ‘Fishery’ table in the 2019 FRAM-CAS database (this record was identified as necessary during the iterative process of loading records into the database using CAS\_1.9.exe and examining the records that did not map correctly in the ‘.bad’ file):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Name** | **Description** | **Terminal** | **AgeIncrementStartYear** | **AgeIncrementMonthDay** | **SSMA\_TimeStamp** |
|  | CAL FS | CALIFORNIA FRESHWATER SPORT | TRUE |  |  | FALSE |

1. Add the following records to the ‘FisheryLookup’ table in the 2019 FRAM-CAS database (these records were identified as necessary during the iterative process of loading records into the database using CAS\_1.9.exe and examining the records that did not map correctly in the ‘.bad’ file):

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Fishery** | **CWDBFishery** | **StateProvince** | **WaterType** | **Species** | **Stock** | **Sector** | **Region** | **Area** | **Location** | **SubLocation** |
|  | 1314 | 10 | 6 | M | 1 |  | O |  |  | NHPS |  |
|  | 1314 | 10 | 6 | M | 1 |  | O |  |  | SFPP |  |
|  | 1314 | 10 | 6 | M | 1 |  | O |  |  | PAPS |  |
|  | 3331 | 40 | 6 | M | 1 |  | O |  |  | PSMB |  |
|  | 4008 | 50 | 6 | F | 1 | SAC |  |  |  |  |  |
|  | 4008 | 54 | 6 | F | 1 | SAC |  |  |  |  |  |
|  | 4008 | 56 | 6 | F | 1 | SAC |  |  |  |  |  |
|  | 4008 | 59 | 6 | F | 1 | SAC |  |  |  |  |  |
|  | 3332 | 46 | 6 | F | 1 |  |  |  |  |  |  |
|  | 2314 | 55 | 3 | F | 1 | STL | 1 | 03 | 08 | 050001 | R |
|  | 2213 | 23 | 2 | M | 1 |  | S | AI | M23A |  |  |

\* note that there are additional fields to the left of ‘SubLocation’ in the table, however, they are blank for each of these records, so they were omitted here.

1. Add the following record to the ‘RelLoc’ table in the 2019 FRAM-CAS database (this record was identified as necessary during the iterative process of loading records into the database using CAS\_1.9.exe and examining the records that did not map correctly in the ‘.bad’ file):

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **record\_code** | **format\_ version** | **submission\_date** | **reporting\_agency** | **release\_ location\_code** | **location\_type** | **release\_ location\_name** | **latitude** | **longitude** | **Rel\_ psc\_ basin** | **Rel\_ psc\_ region** | **epa\_reach** | **description** | **Rel\_ rmpc\_ region** | **Rel\_ rmpc\_ basin** | **RelMA** | **Rel MA Type** | **RelWRIA** | **Rel\_Loc\_Code\_Reg** | **Rel\_Loc\_Code\_State** |
| L | 4.1 | 20181108 | ODFW | 5F333 0300200001 | 4 | CLACKAMETTE COVE |  |  | WILL | LOCR | 1709001100100.00 | CLACKAMAS R. LAGOON ~1,500 FT UPSTREAM OF CONFLUENCE WITH WILLAMETTE R. PEN IN 1994. | LOCR | WILL | 1A | FW | 27.0000W | 14 | 4 |

Same record continued…

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **User\_ Comm** | **Field 1** | **Field 2** | **Field 3** | **Field 4** | **Field 5** | **Field 6** | **Field 7** | **Field 8** | **Field 9** | **Field 10** | **Field 11** | **Field 12** | **Field 13** | **Field 14** | **Field 15** | **Field 16** | **Field 17** | **Field 18** | **Field 19** | **Field 20** | **Field 21** | **Field 22** | **Field 23** | **Field 24** | **Field 25** | **Field 26** | **Field 27** | **Field 28** | **Field 29** | **Field 30** | **Field 31** |
|  |  |  |  |  |  |  |  | 0 |  |  | 0 |  | 0 |  | 0 |  | 0 | 0 |  | 0 | 0 | 0 |  |  |  |  |  |  |  |  |  |

1. Ensure that all records in the ‘SpeciesStock’ table of the previous FRAM-CAS database are present in the 2019 FRAM-CAS database. There will be some FRAM-specific stocks that will need to be added to the 2019 database. There may also be some records in the 2019 database that are not present in the previous FRAM-CAS database – these are stocks used by the CTC but not FRAM and can either remain in the table or be deleted.
2. Delete records in the ‘StockBrood’ table of the 2019 FRAM-CAS database and replace with the records contained in the ‘StockBrood’ table of the previous FRAM-CAS database.
3. Use the CAS program (in this case CAS\_1.9.exe) to load the releases and recoveries files that were produced by the filter database queries (‘*ERARELEASES\_5.2.19.txt’* & ‘*ERARECOVERIES\_5.2.19.csv*’; see steps 3 & 4 above).
4. Load in CAS auxiliary records produce by the CTC. Rather than loading using the CAS program, this was accomplished by querying the 2019 CAS database for all records in the ‘CWDBRecovery’ table with the ‘Auxiliary’ field equal to TRUE, then filtering the records to only those with tag codes used in the FRAM base period. These records were then pasted into the ‘CWDBRecovery’ table of the 2019 FRAM-CAS database (see file ‘*2019\_CAS\_Auxiliaries\_7.31.19.xlsx*’ for details).
5. Use the CAS program (in this case CAS\_1.9.exe) to load additional, non-CTC auxiliary recoveries including those that account for lack of sampling in WA freshwater sport fisheries (‘*WA\_FWspt\_CAS\_load.csv*’) and inter-dam loss (‘*IDL\_Aux\_CASload\_10.3.19.csv*’).
6. Beginning in 2019, WSH freshwater sport and escapement auxiliaries were handled via a CAS query rather than by generating and loading separate auxiliary records. Ensure that these expansions are properly accounted for (see WSH\_Aux tab in ‘*FRAMBuilder DB Updates\_2019.xlsx*’).
7. Update fields in the CWDBRecovery table to reflect those in the previous FRAM-CAS database:
   * There are three fields that will need to be deleted (the three rightmost columns: ERAAge, SSMA\_TimeStamp, AuxLoadID)
   * Two fields will need to be added (finalFmap, Tr\_NT)
8. Update fields in the WireTagCode table to reflect those in the previous FRAM-CAS database:
   * Three fields will need to be added (FRAM\_OOB, BP\_Stk, BP\_Notes – see previous FRAM-CAS database for tag code specific values
9. There are 14 White River tag recoveries that need to be removed from the CWDBRecovery table – see the ‘Removed White Tags’ table in the previous FRAM-CAS database for a list of RecoveryIDs.
10. Create a ‘dummy stock’ for LCN that is a combination of the WA and OR tule stocks (LRH, LCW, KAL, CWF, and WAS). See the ‘LCN\_WireTagCode’ and ‘LCN\_CWDBRecovery’ tabs in ‘‘*FRAMBuilder DB Updates\_2019.xlsx*’.
11. Assign Bellingham Bay net recoveries to treaty or non-treaty (see item 5 in Appendix C and ‘*BellinghamBayEvaluation\_2019.xlsx*’)

1. The FRAM-CAS fishery crosswalk adheres strictly to the CTC’s ‘fine scale’ fishery strata from 2019; any attempt to create an updated FRAM-CAS database will require an updated FRAM to CTC crosswalk (database table ‘FRAM\_Fishery’). Versions of FRAMBuilder databases and outputs created prior to 2019 were based on the final preseason CTC CAS database from 2013 and the ‘fine scale’ fishery strata contained therein. [↑](#footnote-ref-1)
2. Versions of FRAMBuilder databases and output created prior to 2019 used the 2013 version of CAS (CAS1.5\_No\_Restrictions.exe). [↑](#footnote-ref-2)
3. Because the CTC’s fine-scale fishery change routinely, if not updating to the current version of the CAS database, it may be necessary to modify auxiliaries to revert to the 2019 convention before using. [↑](#footnote-ref-3)
4. The origin of this decrementing is undocumented (precedes FRAMBuilder 2.0) and only applies to these two yearling release stocks; assumedly, this is because they’re yearling releases with maturation in TS1/4. [↑](#footnote-ref-4)
5. Note that user-specified between BY merging hasn’t been fully tested/vetted, despite the capacity to do so. [↑](#footnote-ref-5)
6. Note that some post-processing is currently needed to create the full 39/78 stock structure dataset (see Apendix C), which occurs in an excel file called ‘Calibration\_CWT\_Inputs; MM.DD.YY.xlsx’ and requires output from the ‘FRAM\_star\_CWT’ table. [↑](#footnote-ref-6)
7. A FRAMBuilder2.exe hasn’t yet been created because ongoing calibration, combined with new tag records (which occasionally introduce new exception errors), has necessitated minor code changes along the way. Note, however, that although this program was never envisioned to be a ‘production grade’ product, a distributable version can be easily created when things stabilize on the development front. [↑](#footnote-ref-7)
8. In 2019, work was completed to update the FRAM-CAS database and FRAMBuilder 2.0 application code for compatibility with the 2019 version of the CTC’s CAS database and fishery strata (See Appendix D for details). Versions of FRAM-CAS database, application code, and output data created prior to 2019 were based on the 2013 version of the CTC’s CAS database and fishery strata. [↑](#footnote-ref-8)
9. This processing decision was made because non-treaty net fisheries are either nonexistent/considerably diminished in many areas OR pooling recoveries between the two fisher types was necessitated because neither alone was adequate for base period purposes. [↑](#footnote-ref-9)