

Developer's Guide on Reconciliation Project

Developed By: Daryl Bullock

Date: 14 November 2001

Table of Contents

Section 1	Project Summary	ii
Section 2	General Information	
	1.1 Database Used for Application.....	1
	1.1 Database Version.....	1
	1.1 Operating System.....	1
	1.1 Server Name(s).....	1
	1.1 Forms and Reports Versions.....	1
	1.1 Test and Production Database Names.....	1
	1.1 Indexes.....	1
	1.1 Data Entry Form Procedure	
Section 3	Form Procedures	
	2.1 Southeast Trip Totals.....	1
	2.2 All Records.....	2
	2.3 All Records By Boat.....	2
Section 4	Reconciliation Data Entry Forms Operating Instructions	
	3.1 Southeast Trip Data Entry Form.....	3
	3.2 Southeast Trip Totals.....	5
	3.3 Reconciliation Form.....	6
	3.4 Meta Data.....	8
	3.5 Reconciliation All Records.....	9
	3.6 Reconciliation By Vessel.....	10
Section 5	Reports	
	4.1 Trip Comparison.....	11
Appendix A	Entity Relationship Diagram	12
Appendix B	PL/SQL Coding	
	Trip Total.....	13
	All Records.....	14
	All Records By Vessel.....	15
Appendix C	Monthly Script Processing	17

Reconciliation Summary

The reconciliation project compares highly migratory species trip data stored in the NMFS Southeast Science Center (SEFSC) to dealer trip data stored in the NMFS Northeast Science Center (NEFSC). This project will also compare SEFSC to data stored in the database of the Atlantic Coastal Cooperative Statistics Program (ACCSP), which stores the Florida and North Carolina data. The ultimate goal of the reconciliation project is to ensure vessel trip data is accurate and complete between the state, NMFS and ACCSP agencies. The reconciliation application will perform data comparisons between southeast and northeast data centers. Then North Carolina and Florida data in the ACCSP database to data in the southeast database. Those comparisons are then stored in tables on the NMFS database, shark, in a reconciliation table and a log file table. The reconciliation system will also help to identify records known as *split trips*. These are trips where the fisherman sells a portion of his trip in one state and the other portion of his trip in another state. This system will help to identify duplicated data between state and federal agencies, data reported in one system, but not another and also help to identify discrepancies in records, such as landed weights, species landed, state landed or gears used.

General Information

Database: Oracle

Database Version: 8i Release 8.1.7.0.0

Operating System: Sun Solaris 5.6

Name of Server: Urchin

Forms Version: 4.0 (Client Server) 6i (Internet)

Reports Version: 2.5 (Client Server) 6i (Internet)

Tables Involved: *See Appendix A (ERD)*

Test Database: Urchin

Production Database: Shark (*scripts and tables storage*)

Indexes/Table_Type: I_Nmfs_Cd_Log_Dt_State_Cd - Se_Ne_Recon_T - Normal

Pl/Sql Procedures: Within the Reconciliation data entry forms, there are 3 procedures: *Data_In*, which produces the information displayed in the **Trip Totals** screen.

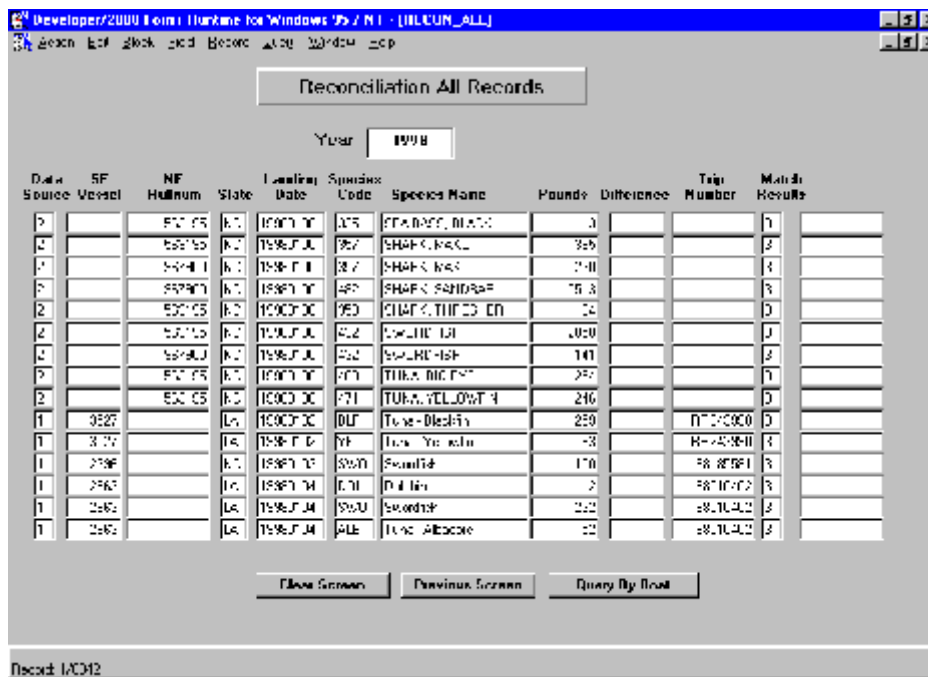
The screenshot shows a Windows 95 window titled 'Developer/2000 Forms Runtime In Windows 95 / NT - (BPO0008_TOT)'. The main window content is a form titled 'SouthEast Trip Totals'. The form contains several input fields and a table.

Trip Number	Vessel ID	State	State Name	Log Date	Gear
9809927	1238	MA	Massachusetts	1998/12	LL

Species Code	Species Name	Weight	Totals
01	Dolphin	100	
02	Striped Grouper	10	
03	Wahoo	100	
04	Yellowtail Snapper	100	

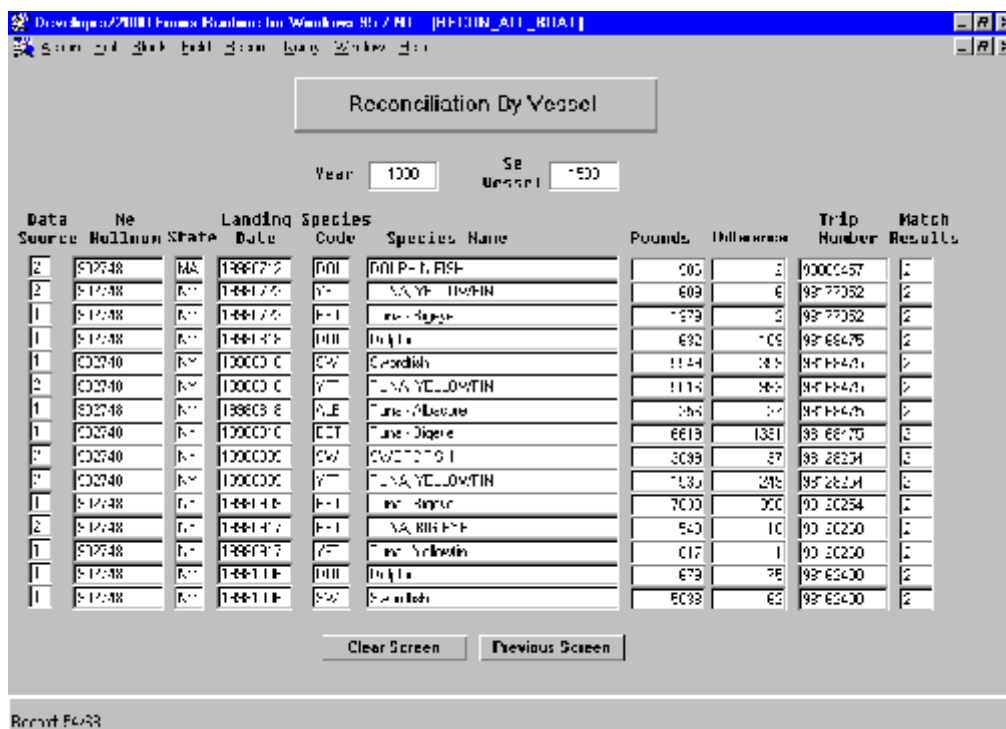
At the bottom of the form, there are two buttons: 'CLEAR SCREEN' and 'PREVIOUS WINDOW'.

The *a_In* info in the Reco



Recon_All_Dat produces the information displayed in the Reconciliation All records screen

All By display info in the screen



and *Data_In_boat* lays the information in the Reconciliation By Vessel screen.

Note - See Appendix B for PL/SQL coding.

2

Reconciliation Data Entry Forms Operating Instructions

The following are operating instructions on the use of the data entry reconciliation form. Upon entering the url: <http://urchin.ssp.nmfs.gov:8000/dev60html/serecon.htm>, a Oracle Forms and Reports Server page with a hotlink of *Longline Data Entry Form* will appear. Clicking on this link will cause the loading of a java applet necessary to run Oracle forms. Then a page which asks for a userid, password and database name appears. After entering the appropriate data, the first form appears, *Southeast Trip Data Entry Form*.

Note: *Userid and passwords can be obtained from Daryl Bullock - F/ST1*

Southeast Trip Data Entry Form

Southeast Trip Data Entry Form

Trip Number	District	Tid	Gear	Trip Date	HOURS FISHED	STATE	STATE NAME
1-84884	007	HI		1998117		A	Louisiana
Dealer	Number of Sets	Port Code	Days Fishing	Location	Data Source		
66		02		0	C		

Species Code		Species Name									
-		Tuna - Yellowfin									
W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	
W12	W13	W14	W15	W16	W17	W18	W19	W20	W21	W22	
W23	W24	W25	W26	W27	W28	W29	W30	W31	W32		

Buttons: ENTER QUERY, EXECUTE QUERY, GET FOR SCREEN, DELETE RECORD, Trip Totals Screen, Reconciliation Screen, EXIT FORM

The
i
s
to
the
east
omesti
gline
This
stores
from
out

form
linked
south
dls_d
c_lon
table.
table
data
weigh
sheet

information. Upon opening the form, its immediate mode of operation is in data entry mode. Therefore at this point, records can be entered into the system.

Entering a record into the database:

A record is entered into the dls_longline table by SEFSC personnel. A record consist of trip, state, date, species along with individual weights. Enter the appropriate data in each field and press the *tab or enter key* on your keyboard or *mouse click* in each field in order to move to the next field. When entering a state code, the appropriate state name will automatically appear. When entering a species code, the species name will automatically appear. Enter individual weight values. When all values are entered, hit the *down arrow key* to commit the record to the database and enter the next record.

Querying records in the database:

The Southeast Trip Data Entry Form can search records in the dls_domestic_longline according to a specified criteria. In order to search all records in the database, mouse click on the **Enter Query** button and immediately hit the **Execute Query** button. This will display all records in the database. In order to view each record, press the *down arrow key* on your keyboard. If there is a particular search criteria that is used, ie a trip number, the query will display all the data for a particular trip number.

FYI: If you wish to query a record and have already started entering the search criteria, but realize the form is in insert mode, press the **Enter Query** button. A forms prompt box will appear, asking, “Do you want to save the changes you have made?” Answer “No” to this box so that the form will not save erroneous data to the database. The form should now be in query mode and you can enter your search criteria.

Hint: Also, a hint that you are in query mode, you will see a **Cancel Query** label on one of the buttons. If you are in insert mode, that same button will change labels and read, **Exit Form**.

Deleting a record from the database:

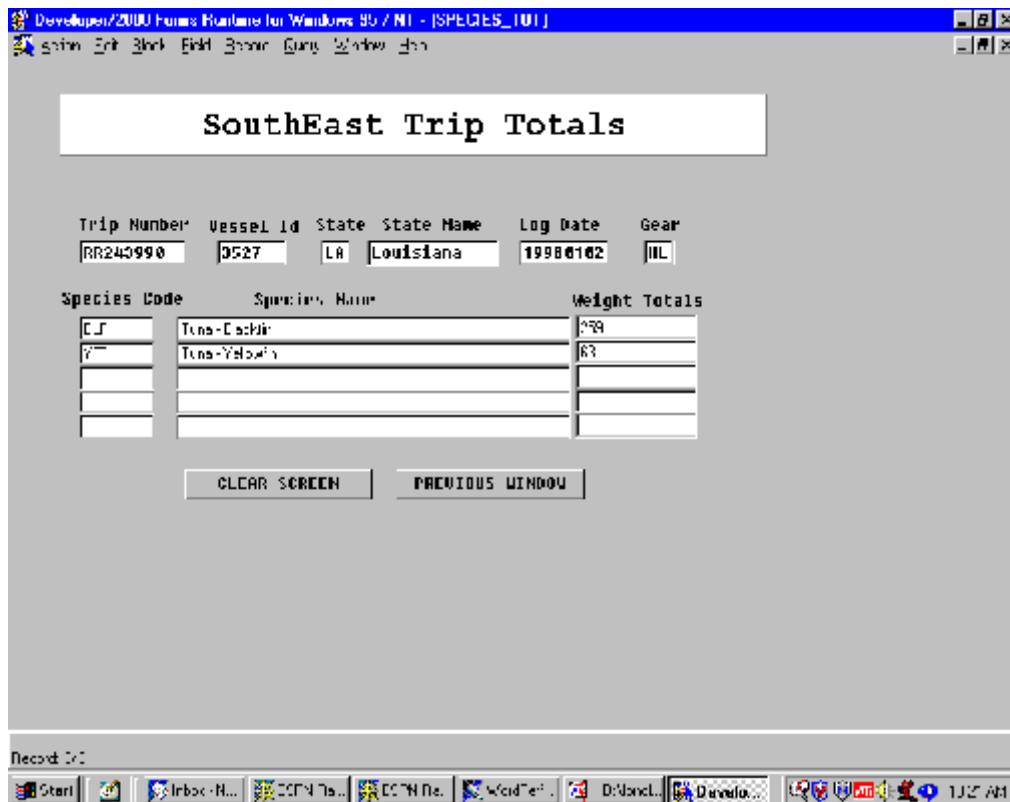
Deletion can only be accomplished one record at a time. Each record contains a maximum of 32 weights. To delete a record from the dls_domestic_longline database, the authorized person must first query the record for deletion. Once that record is displayed, hit the **Delete Record** button. A forms prompt will appear asking, “Are you sure you want to delete this record?” If this is the correct record marked for deletion, answer “Yes” to this box. This record is now deleted from the database. If you are completely finished with all necessary modifications, additions or deletions, click the **Exit Form** button.

Updating a record in the database:

To update a record, first query the record. Once that record is displayed, change the data needing the update and press the *down arrow key*. This record is automatically updated.

If you wish to see species totals by trip, click the *Trip Totals Screen* button.

Southeast Trip Totals Screen



Note: This screen is query only mode. Records can not be edited

The purpose of this screen is to display the species totals per trip. This screen shows the vessel identification number, state landed for a trip, the log or landing date, the gear used on that trip and all the species with their totals that were landed on that trip.

Querying trip totals:

To query trip totals on this screen, enter the trip number in the appropriate field and press the *tab key* and the species code, species name and species totals are displayed.

Clear Screen button will clear any data from the screen and allow the entry of another trip number.

Previous Screen button returns the user to the **Southeast Trip Data Entry Screen**. The last record displayed on that screen will remain on that screen, until the *clear screen* or *enter query* buttons is pressed.

Reconciliation Data Entry Screen

The screenshot shows a Windows 95/NT desktop environment with a window titled "Developer/2000 Forms Runtime for Windows 95 / NT - [INCCOM]". The window displays the "Reconciliation Data Entry Form".

The form contains the following fields:

Data Source	Se Trip Number	Se Vessel ID	Ne Hull Num	State Code
1	HR24389J	3127		LA

Species Code	Species Name	Priority
SLP	Luna-Echlin	388

Landing Date	Case Code	Accep Res Num	Reconciled Date	Match Results
19980702	HI			F

Below the form are two rows of buttons:

- Row 1: ENTER QUERY, EXECUTE QUERY, DELETE RECORD, COMMIT RECORD
- Row 2: Meta Data Screen, Display All Records, Southeast Trip Screen, EXIT FORM

The Windows taskbar at the bottom shows the Start button, several open applications (including "In Lax - N...", "bSPH - Re...", "Woodpeil...", "D.V. 99...", "Drawin..."), and the system clock showing 11:35 AM.

The purpose of this form is to display records that have gone through the matching process of merging records from the southeast and northeast databases according to landing dates, species and vessel identification. From this, the southeast trip number, southeast vessel id, northeast hull numbers, state of landing, landing date(log_date) and the date the records were matched and added to reconciled table.

Querying records:

Querying reconciled (matched) records is done in the same manner as before. To query the entire database, press the ***Enter Query*** button and then ***Execute Query*** button. To use a particular search criteria, enter the criteria in the appropriate field, ie a particular trip number or vessel identification or hull number. Another example, to find all records that matched exactly, press the ***enter query*** button, place a "1" in the *match results* field and hit ***Execute Query***. To query records where the data originated from the northeast, place a "2" in the data source field and then execute query.

Updating a record:

To update a record, query that record and then make the necessary change and hit the ***Commit Record*** button . The changes are then updated in the database.

Deleting a record:

To remove a record from the database, query the record and then press the ***Delete Record*** button. The forms prompt will ask, "*Are you sure you want to delete this record?*" Answer the prompt accordingly.

To view matched records that had discrepancies such a weight difference, the larger of the two weights is place in the reconciliation table and the lower weight is placed in the meta data table. Unmatched trips are identified as *match_result* = "3." To view changes or deletions made to the reconciliation table, press the ***Meta Data Screen*** button.

Recon Meta Data Record Screen

Developer/2000 Forms Runtime for Windows 95 / NT [RECON_MFTA]

Recon Meta Data Table

Trip Number: 311000 Se Vessel Id: 2143 No Nullinum: 33353

Field Name: Fours Old Value: 172 Modified Value:

Status Type: Change Date: User Name:

Comments: The netbar weight was the greater of the two records. The error of the weight was corrected in the meta data table.

ENTER QUERY EXECUTE QUERY PREVIOUS SCREEN

Record 1 of 1

Start | Intra II | FSN Rn | FSN Rn | WinEnt | D:\ | Develop... | 11:01 AM

This screen displays all records that had some type of discrepancy with records in the reconciled table such as the weight. The rule for reconciliation is that the larger value is accepted in the reconciled table and the smaller value is added to the meta data table. Also, if an update is made to the reconciled table, the meta data table stores the previous value and the current value in order to track all transactions to a record including changes to trip header variables as well as deleted unmatched records.

Note: Only the comments field can be modified.

Querying records:

Records are queried in the same manner as before. To query the entire database, press the *Enter Query* button and then press the *Execute Query* button. To query particular records, enter the search criteria and press *execute query*.

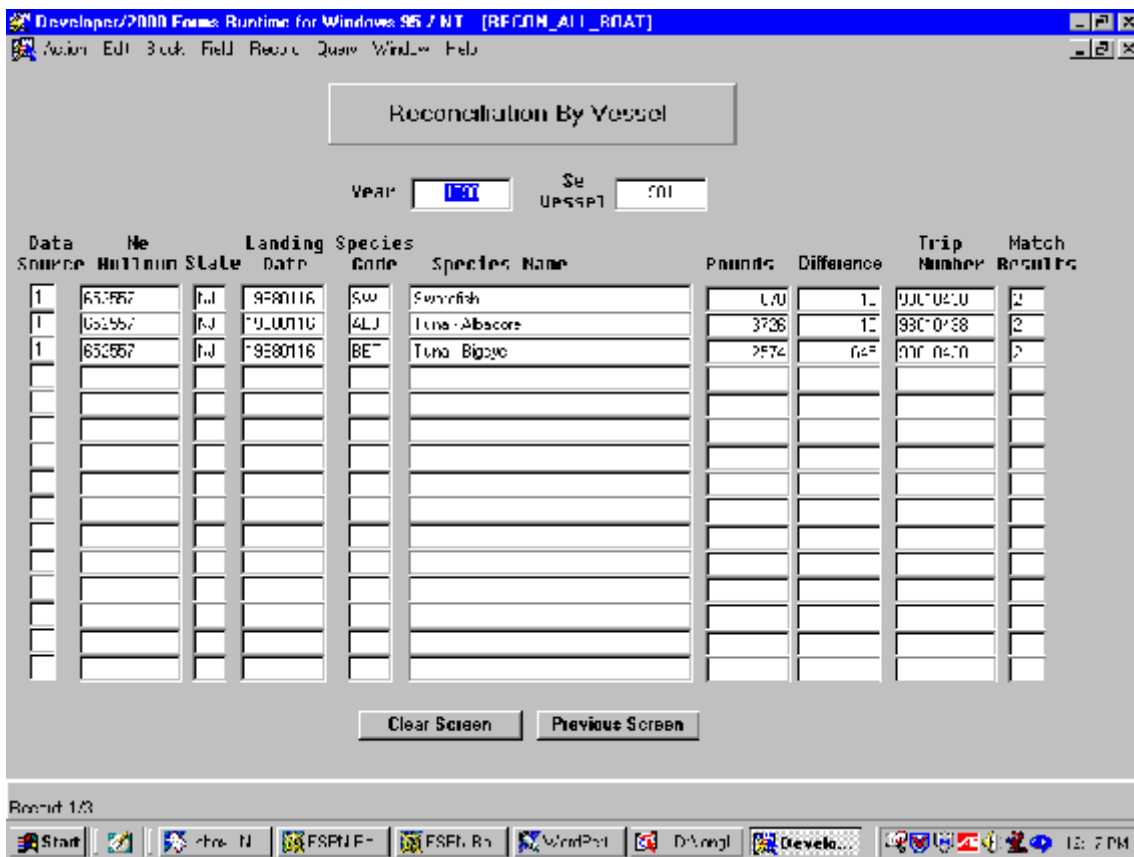
The *Previous Screen* button will return the user to the Reconciliation Data Entry Form.

The *Record* button brings up a display that either weight do not match in reconcil

Data Source	SE Vessel	NE Hullnum	State	Landing Date	Species Code	Species Name	Pounds	Difference	Trip Number	Match Results
1	583		FL	1997-11-06	077	FAW-COASTAL	2		58 7547P	0
1	585		FL	1997-11-06	000	SW-CLASH	7501		58 8150P	0
1	57	2417	FL	1997-11-06	001	SEA-BED	400	11	58 1148	0
1	57	2417	FL	1997-11-06	01F	FAW-COASTAL	2756	17	58 1148P	0
1	57	2420	FL	1997-11-06	021	LAND-SCAPE	2072	240	58 1148	0
1	57		FL	1997-11-06	001	SEA-BED	772		58 9411	0
1	57		FL	1997-11-06	01F	FAW-COASTAL	168		58 9411	0
1	57		FL	1997-11-06	021	LAND-SCAPE	273		58 9411	0
1	57		FL	1997-11-06	077	FAW-COASTAL	2321		58 9411	0
1	57		FL	1997-11-06	00H	SEA-BED	7		58 9411	0
2		2417A	FL	1997-11-06	000	SEA-BED, SEA-BED	5			0
2		2417A	FL	1997-11-06	021	SEA-BED, SEA-BED	5			0
2		2417A	FL	1997-11-06	027	SEA-BED, SEA-BED	7			0
2		2420A	FL	1997-11-06	024	SEA-BED, SEA-BED	258			0
2		2420A	FL	1997-11-06	027	SEA-BED, SEA-BED	270			0

Display All button brings screen that all records that had a landed discrepancy or have a g record in the ed table.

To Query records from this screen, enter the year to view and hit the tab-key or click on the first record of the data source field. All records in the reconciled table are then displayed. Any records having a landings weight discrepancy between the southeast and northeast record, a difference in the weight values is displayed. This screen also allows the user to view all records in a given year and look up whether a record has a possible split trip causing incorrect weight values or incorrectly identified species.



9

The Query By Boat button brings up a screen that displays

all records by year and southeast vessel identification number. This screen is similar to the All Records Screen.

To query records, enter the year in the appropriate field then hit the tab key and enter the southeast vessel number and then hit the tab key or mouse click into the data source field. The query will result in all records pertaining to that vessel. Weight discrepancies and differences are displayed, along with any vessels not having a matching record from the southeast or northeast. This screen also allows the user to view vessel landings in a given year that may have a split trips. This screen may help to identify incorrect landing dates or incorrect vessel identification numbers.


Reconciliation Reports

Developer/2000 Reports Designer for Windows 95 / NT [main: Preview]

File Edit Window Help

Prev Next First Last Page: Print Mail Close New

Date 16 NOV 2001
Page 1



SETRC ST1 IMS Trip Comparison
For The State Of Massachusetts

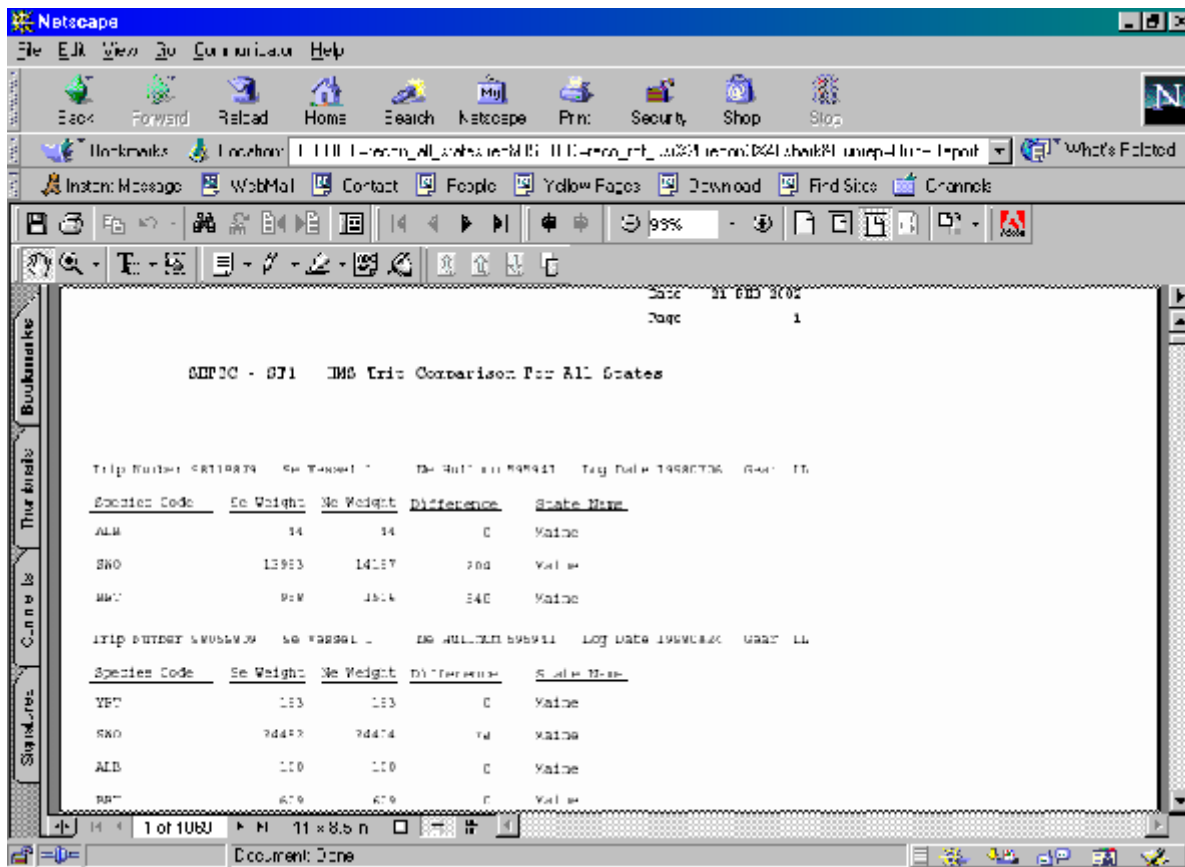
Trip Number 98141801 Se Vessel 1 No Hull No 49507 Log Date 19981118 Gear 1

Species Code	Se Weight	Ne Weight	Difference
ALB	396	447	49
EDL	140	119	11
Y37	3076	3076	0
E37	5335	5335	0
Y40	511	521	41

Trip Number 98129677 Se Vessel 1176 No Hull No Log Date 19980312 Gear GK

Species Code	Se Weight	Ne Weight	Difference
Y30	5076		
Y37	507		

The Highly Migratory Trip Comparison report lists all species landed on each trip, per state, along with the comparison of southeast and northeast landings per species and any weight differences. This report provides a complete list of all trips included in each database by state, species and dates. Unmatched trips are those that lack either a southeast vessel identification number or northeast hull number variable. These trips can then be quickly compared against values listed in the difference columns. This will help to resolve discrepancies attributed to split trips.



This Highly Migratory Trip Comparison report displays all records from all states and lists all landings per species landed on each trip within each state by date.

Appendix B Procedures

Southeast Trip Total Procedure

```
PROCEDURE DATA_IN(trip_number IN VARCHAR2) IS
  CURSOR curs1 IS
    SELECT trip_number, a.species_code, common_name, sum(nvl(weight1,0) + nvl(weight2,0) +
      nvl(weight3,0) + nvl(weight4,0) + nvl(weight5,0) + nvl(weight6,0) +
      nvl(weight7,0) + nvl(weight8,0) + nvl(weight9,0) + nvl(weight10,0) +
      nvl(weight11,0) + nvl(weight12,0) + nvl(weight13,0) + nvl(weight14,0) +
      nvl(weight15,0) + nvl(weight16,0) + nvl(weight17,0) + nvl(weight18,0) +
      nvl(weight19,0) + nvl(weight20,0) + nvl(weight21,0) + nvl(weight22,0) +
      nvl(weight23,0) + nvl(weight24,0) + nvl(weight25,0) + nvl(weight26,0) +
      nvl(weight27,0) + nvl(weight28,0) + nvl(weight29,0) + nvl(weight30,0) +
      nvl(weight31,0) + nvl(weight32,0)) wts
  FROM dls_longline a, dls_species_code b
  WHERE a.species_code = b.dls_species_code AND
        trip_number = :longline_tot.trip_number
  GROUP BY trip_number, a.species_code, common_name;

BEGIN
  OPEN curs1;
  LOOP
    FETCH curs1 INTO :species_tots.trip_number, :species_tots.species_code,
      :species_tots.species_name, :species_tots.weight_tot;
    IF curs1%NOTFOUND THEN
      NULL;
    ELSE
      NEXT_RECORD;
    END IF;
  EXIT WHEN curs1%NOTFOUND;
  END LOOP;
  CLOSE curs1;
  END data_in;
```

Reconciliation All Records Procedure

```

PROCEDURE recon_all_data_in(year IN VARCHAR2) IS
  CURSOR curs1 IS
    SELECT data_source, se_vessel_id, ne_hullnum, state_code, landing_date, s
    species_code, species_name, pounds, se_trip_number, accsp_rec_num,
    match_results
  FROM recon_se_ne
  WHERE substr(landing_date,1,4) = :recon_se_ne2.year AND match_results IN ('2','3')
  ORDER BY landing_date, se_vessel_id, species_name;

  CURSOR se_curs IS
    SELECT data_source, se_vessel_id, ne_hullnum, state_code, landing_date, species_code,
    species_name, pounds, se_trip_number, accsp_rec_num, match_results
  FROM recon_se_ne
  WHERE substr(landing_date,1,4) = :recon_se_ne2.year AND match_results IN ('2','3') AND
    species_code in ('ALB','470','BET','469','BLF','464','BON','033','DOL','105',
    'LTA','468','SKJ','466','SWO','432','WAH','472','YFT','471')
  ORDER BY landing_date, se_vessel_id, species_name;

BEGIN
  IF GET_APPLICATION_PROPERTY(USERNAME) in ('ABERTOLI','JCRAMER') THEN
    OPEN se_curs;
    LOOP
      FETCH se_curs INTO :recon_se_ne2.data_source, :recon_se_ne2.se_vessel_id,
        :recon_se_ne2.ne_hullnum, :recon_se_ne2.state_code,
        :recon_se_ne2.landing_date, :recon_se_ne2.species_code,
        :recon_se_ne2.species_name, :recon_se_ne2.pounds,
        :recon_se_ne2.se_trip_number, :recon_se_ne2.accsp_rec_num,
        :recon_se_ne2.match_results;

      IF se_curs%NOTFOUND THEN
        NULL;
      ELSE
        BEGIN
          SELECT ABS(se_weight - ne_weight)
          INTO :recon_se_ne2.difference
          FROM se_ne_recon_t a
          WHERE :recon_se_ne2.match_results = '2' AND
            a.se_vessel = :recon_se_ne2.se_vessel_id AND
            a.se_state_code = :recon_se_ne2.state_code AND
            a.se_log_date = :recon_se_ne2.landing_date AND
            a.trip_number = :recon_se_ne2.se_trip_number AND
            a.se_species_code = :recon_se_ne2.species_code;
        EXCEPTION
          WHEN NO_DATA_FOUND THEN
            NULL;
        END;
        NEXT_RECORD;
      END IF;
      EXIT WHEN se_curs%NOTFOUND;
    END LOOP;
    CLOSE se_curs;
  
```

ELSE

14

```
OPEN curs1;
  LOOP
    FETCH curs1 INTO :recon_se_ne2.data_source, :recon_se_ne2.se_vessel_id,
                    :recon_se_ne2.ne_hullnum, :recon_se_ne2.state_code,
                    :recon_se_ne2.landing_date, :recon_se_ne2.species_code,
                    :recon_se_ne2.species_name, :recon_se_ne2.pounds,
                    :recon_se_ne2.se_trip_number, :recon_se_ne2.accsp_rec_num,
                    :recon_se_ne2.match_results;

    IF curs1%NOTFOUND THEN
      NULL;
    ELSE
      BEGIN
        SELECT ABS(se_weight - ne_weight)
        INTO :recon_se_ne2.difference
        FROM se_ne_recon_t a
        WHERE :recon_se_ne2.match_results = '2' AND
              a.se_vessel = :recon_se_ne2.se_vessel_id AND
              a.se_state_code = :recon_se_ne2.state_code AND
              a.se_log_date = :recon_se_ne2.landing_date AND
              a.trip_number = :recon_se_ne2.se_trip_number AND
              a.se_species_code = :recon_se_ne2.species_code;
      EXCEPTION
        WHEN NO_DATA_FOUND THEN
          NULL;
      END;
      NEXT_RECORD;
    END IF;
  EXIT WHEN curs1%NOTFOUND;
END LOOP;
CLOSE curs1;
END IF;
END recon_all_data_in;
```

All Records By Vessel Procedure

```
PROCEDURE all_data_in_byboat(year IN VARCHAR2, se_vessel_id IN VARCHAR2) IS
  CURSOR boat_curs IS
    SELECT data_source, ne_hullnum, state_code, landing_date, species_code,
           species_name, pounds, se_trip_number, accsp_rec_num, match_results
    FROM recon_se_ne
    WHERE substr(landing_date,1,4) = :recon_se_ne3.year AND
          se_vessel_id = :recon_se_ne3.se_vessel_id AND match_results IN ('2','3') AND
          species_code in ('ALB','470','BET','469','BLF','464','BON','033','DOL','105',
                           'LTA','468','SKJ','466','SWO','432','WAH','472','YFT','471')
    ORDER BY landing_date, species_name;
```

```
BEGIN
  OPEN boat_curs;
  LOOP
    FETCH boat_curs INTO :recon_se_ne3.data_source, :recon_se_ne3.ne_hullnum,
                        :recon_se_ne3.state_code, :recon_se_ne3.landing_date,
                        :recon_se_ne3.species_code, :recon_se_ne3.species_name,
                        :recon_se_ne3.pounds, :recon_se_ne3.se_trip_number,
                        :recon_se_ne3.accsp_rec_num, :recon_se_ne3.match_results;

    IF boat_curs%NOTFOUND THEN
      NULL;
    ELSE
      BEGIN
        SELECT ABS(se_weight - ne_weight)
        INTO :recon_se_ne3.differences
        FROM se_ne_recon_t a
        WHERE :recon_se_ne3.match_results = '2' AND
              a.se_vessel = :recon_se_ne3.se_vessel_id AND
              a.se_state_code = :recon_se_ne3.state_code AND
              a.se_log_date = :recon_se_ne3.landing_date AND
              a.trip_number = :recon_se_ne3.se_trip_number AND
              a.se_species_code = :recon_se_ne3.species_code;
      EXCEPTION
        WHEN NO_DATA_FOUND THEN
          NULL;
      END;
      NEXT_RECORD;
    END IF;
  EXIT WHEN boat_curs%NOTFOUND;
  END LOOP;
  CLOSE boat_curs;
END all_data_in_byboat;
```

Appendix C Monthly Script Processing

REM Copied from dls8, this script is pulling all trips and matching records
REM Updates state code in NEHMS table in order to do match from view
REM Also updates ne gears to match se gear codes.

```
UPDATE nehms
set state_code = 'CT'
where substr(port,1,2) = '07';
```

```
UPDATE nehms
set state_code = 'DE'
where substr(port,1,2) = '08';
```

```
UPDATE nehms
set state_code = 'ME'
where substr(port,1,2) = '22';
```

```
UPDATE nehms
set state_code = 'MD'
where substr(port,1,2) = '23';
```

```
UPDATE nehms
set state_code = 'MA'
where substr(port,1,2) = '24';
```

```
UPDATE nehms
set state_code = 'NH'
where substr(port,1,2) = '32';
```

```
UPDATE nehms
set state_code = 'NJ'
where substr(port,1,2) = '33';
```

```
UPDATE nehms
set state_code = 'NY'
where substr(port,1,2) = '35';
```

```
UPDATE nehms
set state_code = 'NC'
where substr(port,1,2) = '36';
```

```
UPDATE nehms
set state_code = 'RI'
where substr(port,1,2) = '42';
```

```
UPDATE nehms
set state_code = 'VA'
where substr(port,1,2) = '49';
```


REM NJ gave me problems with too many rows. Converting ne gear codes to SE

update nehms

```
set segear = decode(negear,'110','GN','100','GN','020','HL','066','HL',
                    '170','PT','030','HP','031','HP','010','LL','040','LL',
                    '120','PS','121','PS','122','PS','123','PS','124','PS',
                    '020','RR','180','TP','181','TP','182','TP',
                    '183','TP','184','TP','185','TP','186','TP',
                    '187','TP','188','TP','189','TP','190','TP',
                    '191','TP','192','TP','193','TP','194','TP',
                    '195','TP','196','TP','197','TP','198','TP',
                    '199','TP','200','TP','201','TP','202','TP',
                    '203','TP','204','TP','205','TP','206','TP',
                    '207','TP','208','TP','209','TP','210','TP',
                    '211','TP','212','TP','213','TP','214','TP',
                    '215','TP','216','TP','217','TP',
                    '050','TR','051','TR','052','TR','053','TR',
                    '054','TR','055','TR','056','TR','057','TR',
                    '058','TR','059','TR', null)
```

where segear is null;

CREATE or REPLACE view se_ne_recon_v

```
AS SELECT a.trip_number trip, a.gear_code gear, to_char(a.vessel_id) vessel,
          a.species_code species, b.nmfs_species_code nmfs_code,
          a.state_code, to_char(a.log_date) log_date,
          sum(nvl(weight1,0) + nvl(weight2,0) + nvl(weight3,0) + nvl(weight4,0) +
              nvl(weight5,0) + nvl(weight6,0) + nvl(weight7,0) + nvl(weight8,0) +
              nvl(weight9,0) + nvl(weight10,0) + nvl(weight11,0) + nvl(weight12,0) +
              nvl(weight13,0) + nvl(weight14,0) + nvl(weight15,0) + nvl(weight16,0) +
              nvl(weight17,0) + nvl(weight18,0) + nvl(weight19,0) + nvl(weight20,0) +
              nvl(weight21,0) + nvl(weight22,0) + nvl(weight23,0) + nvl(weight24,0) +
              nvl(weight25,0) + nvl(weight26,0) + nvl(weight27,0) + nvl(weight28,0) +
              nvl(weight29,0) + nvl(weight30,0) + nvl(weight31,0) + nvl(weight32,0)) weights,
          to_char(c.cg_nmbr) matching_vessel, 'southeast' dbase
FROM dls_longline a, dls_species_code b, se_cg9899 c
WHERE a.species_code in ('SWO','YFT','BET','SKJ','ALB','BON','BLF',
                        'LTA','WAH','DOL')
AND a.species_code = b.dls_species_code
AND a.vessel_id = c.dls_id(+)
GROUP BY a.trip_number, a.vessel_id, a.species_code, b.nmfs_species_code,
         b.common_name, a.gear_code, a.state_code, a.log_date, c.cg_nmbr
UNION ALL
SELECT '0000' trip, segear gear, a.hullnum vessel,
       substr(a.nespp4,1,3) species, b.nmfs_cd, state_code,
       year||month||day log_date, sum(splndlb) weights,
       to_char(c.dls_id) matching_vessel, 'northeast' dbase
```

```

FROM nehms a, ne_species b, se_cg9899 c
WHERE a.nespp4 = b.nespp4 AND a.hullnum = to_char(c.cg_nمبر)
GROUP BY a.hullnum, substr(a.nespp4,1,3), b.nmfs_cd, year||month||day,
        segear, state_code, c.dls_id;

```

18

REM Cursor set up to load 1 record at a time in to recon and meta table.

DECLARE

CURSOR curs1 IS

```

        SELECT trip_number, segear, negear, se_vessel, ne_hullnum,
               se_species_code, se_species_name, ne_species_code,
               ne_species_name, se_nmfs_code, ne_nmfs_code, se_state_code,
               ne_state_code, state_name, se_log_date, ne_landing_date,
               se_weight, ne_weight
        FROM se_ne_recon_t;

```

```

strip_number    varchar2(10);
ssegear         varchar2(2);
snegear         varchar2(3);
sse_vessel      varchar2(5);
sne_hullnum     varchar2(10);
sse_species_code varchar2(3);
sse_species_name varchar2(30);
sne_species_code varchar2(3);
sne_species_name varchar2(30);
sse_nmfs_code   varchar2(4);
sne_nmfs_code   varchar2(4);
sse_state_code  varchar2(2);
sne_state_code  varchar2(2);
sstate_name     varchar2(30);
sse_log_date    varchar2(8);
sne_landing_date varchar2(8);
sse_weight      number(6);
sne_weight      number(6);

```

BEGIN

OPEN curs1;

LOOP

```

        FETCH curs1 INTO strip_number, ssegear, snegear, sse_vessel,
               sne_hullnum, sse_species_code, sse_species_name, sne_species_code,
               sne_species_name, sse_nmfs_code, sne_nmfs_code, sse_state_code,
               sne_state_code, sstate_name, sse_log_date, sne_landing_date,
               sse_weight, sne_weight;

```

```

        IF curs1%ROWCOUNT = 0 THEN
            close curs1;
            RAISE NO_DATA_FOUND;

```

```
END IF;
```

```
IF curs1%NOTFOUND THEN
```

```
  EXIT;
```

```
END IF;
```

```
IF sse_weight = sne_weight AND sse_log_date = sne_landing_date AND  
sse_nmfs_code = sne_nmfs_code THEN
```

```
  INSERT INTO recon_se_ne(se_vessel_id, ne_hullnum, state_code,  
    landing_date, gear_code, species_code, species_name, pounds,  
    se_trip_number, data_source, match_results, reconciled_date)  
  VALUES (sse_vessel, sne_hullnum, sse_state_code, sse_log_date,  
    ssegear, sse_species_code, sse_species_name, sse_weight,  
    strip_number, '1', '1', sysdate);
```

19

```
ELSIF
```

```
  sse_log_date = sne_landing_date AND  
  sse_nmfs_code = sne_nmfs_code AND sse_weight > sne_weight THEN
```

```
    INSERT INTO recon_se_ne(se_vessel_id, ne_hullnum, state_code,  
      landing_date, gear_code, species_code, species_name,  
      pounds, se_trip_number, data_source, match_results,  
      reconciled_date)  
    VALUES (sse_vessel, sne_hullnum, sse_state_code, sse_log_date,  
      ssegear, sse_species_code, sse_species_name, sse_weight,  
      strip_number, '1', '2', sysdate);
```

```
/* This section puts the ne weight into the meta data table because it is the  
  lesser of the weight values */
```

```
  INSERT INTO recon_meta(trip_number, se_vessel_id, ne_hullnum,  
    species_name, field_name, old_value, comments)  
  VALUES (strip_number, sse_vessel, sne_hullnum,  
    sne_species_name, 'Pounds', sne_weight,
```

```
    'The southeast weight was the greater of the two records. The lesser of the weight values is  
  enter in the meta data table.');
```

```
ELSIF
```

```
  sse_log_date = sne_landing_date AND  
  sse_nmfs_code = sne_nmfs_code AND sse_weight < sne_weight THEN
```

```
    INSERT INTO recon_se_ne(se_vessel_id, ne_hullnum, state_code,  
      landing_date, gear_code, species_code, species_name,  
      pounds, se_trip_number, data_source, match_results,  
      reconciled_date)  
    VALUES (sse_vessel, sne_hullnum, sse_state_code, sse_log_date,  
      ssegear, sse_species_code, sne_species_name, sne_weight,  
      strip_number, '2', '2', sysdate);
```

```
/* This section puts the se weight in the meta data table because it is the  
  lesser of the weight values in the meta table. */
```

```
  INSERT INTO recon_meta(trip_number, se_vessel_id, ne_hullnum,  
    field_name, species_name, old_value, comments)  
  VALUES (strip_number, sse_vessel, sne_hullnum,  
    'Pounds', sse_species_name, sse_weight,
```

```
    'The northeast weight was the greater of the two records. The lesser of the weight values is  
  enter in the meta data table.');
```

```
/* SE data exists, without northeast data to match. */
```

```
ELSIF
```

```
sne_landing_date IS NULL THEN
```

```
INSERT INTO recon_se_ne(se_vessel_id, state_code, landing_date,  
gear_code, species_code, species_name, pounds,  
se_trip_number, data_source, match_results)
```

```
VALUES (sse_vessel, sse_state_code, sse_log_date, ssegear,  
sse_species_code, sse_species_name, sse_weight,  
strip_number, '1', '3');
```

20

```
/* NE data exists, without southeast data to match. */
```

```
ELSIF
```

```
sse_log_date IS NULL THEN
```

```
INSERT INTO recon_se_ne(ne_hullnum, state_code, landing_date,  
species_code, species_name, pounds, data_source,  
match_results)
```

```
VALUES (sne_hullnum, sne_state_code, sne_landing_date,  
sne_species_code, sne_species_name, sne_weight, '2',  
'3');
```

```
END IF;
```

```
EXIT WHEN curs1%NOTFOUND;
```

```
END LOOP;
```

```
CLOSE curs1;
```

```
END;
```

```
/
```

```
show errors;
```

```
commit;
```

