



**Regional Register VID Updating System  
User's Manual and Technical Reference**

by Murray Nicholson, Barry Mooney  
Associates  
Forum Fisheries Agency  
May, 1991

**FFA Report 91/74**

**PACIFIC ISLANDS FORUM FISHERIES AGENCY**  
P.O.BOX 629 HONIARA SOLOMON ISLANDS  
TELEPHONE (677) 21124  
FAX (677) 23995  
WEB <http://www.ffa.int>

## TABLE OF CONTENTS

ACKNOWLEDGEMENT .....	2
1. Background.....	1
2. Introduction.....	2
2.1 The Initial Study.....	2
2.2 SPC's Register Table.....	4
2.3 Some Additional Observations About the SPC REGISTER table.....	6
2.4 The Matching System.....	6
3. The Programs.....	6
3.1 Program Flow.....	7
3.2 How Records are Marked.....	8
3.3 Results of the Automated Vessel Identification Search.....	8
3.4 Conclusions and Recommendations.....	9
4. Technical Reference.....	10
4.1 The Directory Structures.....	10
4.2 The Rscan Program.....	10
4.2.1 Program Function Overview.....	12
4.2.2 Include Files.....	12
4.2.3 C Source File Descriptions.....	12
4.2.4 Pro*C Source File Descriptions.....	12
4.2.5 Program Calling Hierarchy.....	14
4.2.6 Error Messages.....	14
4.2.7 Using REGISTER_FFA as Input to rscan.....	15
4.2.8 Specifying a Range of boat Numbers for rscan.....	15
4.3 The Rlogs Program.....	15
4.3.1 Program Function Overview.....	17
4.3.2 Include Files.....	17
4.3.3 C Source File Descriptions.....	17
4.3.4 Pro*C Source File Descriptions.....	17
4.3.5 Program Calling Hierarchy.....	18
4.3.6 Error Messages.....	18
4.4 SQL Scripts.....	19
4.4.1 List of SQL Scripts and their Function.....	19
APPENDIX I - Data Flow Diagram .....	21

APPENDIX II - Database Fields.....	22
APPENDIX III - Query Matrix and Confidence Levels.....	26
APPENDIX IV - Sample RSCAN log file.....	28
APPENDIX V - Sample RLOGS log file.....	29
APPENDIX VI - Summary of Shell File Scripts.....	32

## **ACKNOWLEDGEMENT**

Funding for this Vessel Identity Enhancement project was provided by the International Centre for Ocean Development (ICOD). This funding support is gratefully acknowledged.

## 1. Background

The Fisheries Management Information System (FMIS) is a regional database at the Forum Fisheries Agency (FFA) headquarters in Honiara, Solomon Islands. FMIS contains several subsystems which include the Regional Register, Logsheet System and Mapping.

The FMIS resides on the Agency HP9000/845 mini computer with 16 M bytes RAM, 1.5 G bytes disc storage, running HP-UX version 7 and Oracle version 6.

The Logsheet subsystem contains data on fishing vessel activity within the region which is obtained from monthly logsheets. These logsheets are completed by the fishing master of a vessel and contain two types of data, *header* and *detail*.

The header data contains information regarding the vessel itself and the fishing trip. This includes the vessel name, country registration number, gross registered tonnage, number of crew, departure port, return port, departure date, return date and country where fishing activity occurred.

The detail data corresponds to activity line items on the logsheet form and includes date, location in latitude and longitude, and catch by species.

There are essentially only three types of logsheets which correspond to the three different fishing methods: longline, pole and line and purse seine. Each logsheet contains data which is relevant to that fishing method.

The logsheets are stored in three separate pairs of header and detail tables, known as *trip* and *raw*, respectively. The link between these table pairs is identical for all three fishing methods and is a composite key of **boat number (or vessel id)**, **reporting country** and **first date of fishing**. This first date of fishing is the same as the first date in the corresponding detail table's set of logsheets.

The South Pacific Commission (SPC) in Noumea, New Caledonia, undertakes the data entry of the majority of logsheets and regularly sends this data to FFA on magnetic tape where the data is loaded into the FFA database. As SPC is continually updating its historical data as well as adding new data, the full data is sent to FFA. The database systems at FFA and SPC are nearly identical.

The FFA maintains a Regional Register subsystem which contains current and historical information regarding fishing vessels, each one having a unique vessel id number. This subsystem consists of a set of tables containing data on all vessels licensed to fish in the South Pacific. The only two tables relevant to this process are the current vessel table and the history vessel table where each vessel has only one record in the current table but may have multiple records in the history table.

Unfortunately, this vessel id number cannot be used directly by vessel captains when they submit their logsheets for data entry by SPC. Instead, the vessel id is extracted from the Regional Register at the time of data entry by matching on several attribute fields.

These attributes are stored in the SPC table REGISTER and in the FFA tables VCRNT (current vessel data) and VHIST (historical vessel data). The radio call sign, one of the more useful Regional Register attributes, is noticeably absent from SPC's REGISTER table.

## 2. Introduction

SPC does its best to extract the correct vessel id but often does not find a match. We have discovered that they may not consult the historical table which could assist in finding matches. In cases such as this, SPC is forced to create a *boat* number to be able to link the logsheets back to SPC's own REGISTER table. FFA has tried working with SPC to ameliorate their process of vessel id extraction and have sent the FFA Senior Analyst Programmer to assist. However, no satisfactory routines were developed and the inconsistency of vessel id presents great difficulties, especially when attempting to extract activity for a specific vessel, an essential function of the FFA surveillance programme.

A process was deemed necessary which would update, where feasible, the *boat* field in both header and detail logsheet tables for each fishing method. FFA finally decided to bring in a consultant well versed in Oracle and C to perform this task. The objectives of the consultancy were:

- to maximise the number of logsheet data sets (header and detail) which are linked to the Regional Register by using extensive vessel id searches, and
- to report cases of partial vessel id matches that could possibly be linked to the Regional Register by further, manual investigations.

### 2.1 The Initial Study

A study was made of the problems in identifying vessels in the FMIS. Two main programs and a host of database scripts were written to examine and correct vessel id numbers in the SPC logsheet data. An exhaustive search of SPC data for matching vessels in the FFA Regional Register showed that approximately half the vessels referred to by SPC are either not in the Regional Register or cannot be recognized. Use of a radio call sign field in the SPC logsheet data would greatly assist vessel identification.

The problems with trying to match vessel names from the logsheet with one existing in the Regional Register are caused by several factors including:

- Names find various spellings on the logsheet, especially when translated from Japanese or Taiwanese to English;
- The vessel data (name, registration number, radio call sign, etc) may change during the course of the year, and FFA is not always or immediately notified of these changes: they simply appear on the newest logsheet;
- The radio call sign has never been included as part of the logsheet data, and seems to be one of the more stable data fields;
- The old registration number in the historical table is sometimes over-written with the value from the current table, VCRNT. When backup copies of VCRNT and VHIST were examined, there appeared to be only a few of these cases.
- The attributes used for matching, other than the *boat* and *vid* fields, are not required to be non-null:

<u>Attribute</u>	<u>SPC Name</u>	<u>FFA Name</u>
Vessel ID	BOAT (Not Null)	VID (Not Null)
SPC VID	VID	VID
Vessel Name	VESSEL	VNAME
Registration	REG_NO	VREGN
Vessel Type <sup>1</sup>	GR	VTYPE
Gross Reg. Tonnes	GRT	GRT
Country "Flag"	FL	FLAG

<sup>1</sup> Also known as "Gear" or "Fishing Method"

FFA id numbers consist of four digits plus a "check" fifth digit - meaning that one new vessel id for every multiple of 10 is generated. *Vid* numbers currently range from 10 to the 27,000's. SPC *boat* numbers are not restricted to this range.

When SPC tries to find the correct vessel, it identifies four levels of match between REGISTER and Regional Register records, using the a flag called *vdeg* with the following meanings:

<u>Vdeg value</u>	<u>Meaning</u>
Null	Neither name nor registration match
1	Vessel name and registration match
2	Vessel name and type match

### 3 Registration and type match

When there is no match (*vdeg* is null), SPC generates a new *boat* number between 90,000 and 99,999. At first filled completely with one boat every increment of 100, this range is now being refilled with smaller increments from 90,000 on up.

According to SPC, data prior to 1987 and not found in the Regional Register were given *boat* numbers between 60,000 and 79,999. The 90,000 range is used for post 1987 data.

#### 2.2 SPC's Register Table

The REGISTER table was studied further in relation to the FFA Regional Register, with these results:

- 50.0% of REGISTER records have a *vid* number, and of these,
  - 81.4% have *vdeg* at a level 1 match
  - 11.7% have *vdeg* at a level 2 match
  - 6.9% have *vdeg* at a level 3 match

This implies that only half the vessels are being identified in the Regional Register. To confirm this, the database was queried to find the number of vessel names and registration numbers in REGISTER which did not occur in the Regional Register at all. The answer was quickly obtained (using the "outer join" query; see the Section 2.4 SQL Scripts "count\_no\_nm.sql" script example):

	<b>Not in RR Current</b>	<b>Not in RR History</b>
REGISTER Vessel Name	51.7%	78.5%
REGISTER Registration	38.4%	67.5%

These figures are based on exact matches and could no doubt be lowered somewhat by accounting for spelling mismatches.

A few other statistics can be given to reflect the severity of the problem:

- 791 REGISTER records (between *boat* numbers 63184 and 65330) have coded vessel names, for example "JP Registration Number";
- 779 REGISTER records have null registrations;
- 168 records with *boat* numbers above 90,000 are "TW" and "KR" flag longline vessels with coded names like "NMFS....", a US designation;
- 242 records with *boat* numbers in the 90,000 range have a registration number consisting of all space characters;

- For a small number of records, (23 at the time of this report), SPC uses several vessel type codes which differ from the three standard ones:

**Standard designations:**

<u>SPC</u>	<u>FFA</u>	<u>Meaning</u>
L	LL	Longline vessel
P	PL	Pole and Line vessel
P	PS	Purse Seine vessel

**Others used by SPC:**

G	Gill Net
M	Carrier (Mother Ship)
T	Troll (NZ and US designations)

- 137 REGISTER records between 19000 and 19993 (inside the range currently in use by FFA *vid* numbers) are not Regional Register *vid* numbers and violate the SPC "one per 100" rule. For example, 19531, 19535, and 19537. Many records show a good match with proper *vid* numbers in the FFA Regional Register but still require further manual investigation.



## 2.3 Some Additional Observations About the SPC REGISTER table

- 32 REGISTER records occupy the range 80,000 to 80330. However, according to the SPC documentation, this range is not used.
- 2 REGISTER records are not actual vessels, but *documentation*:
  - 60000 "...Old TAB file vessels..."
  - 78000 "Freq/L type vessels"
- Certain logsheet data *boat* numbers are not in REGISTER at all:

### Longline

99000, 99090

### Pole and Line

60007, 60179, 60361, 61247, 61453, 65264

- 17 REGISTER records have registration numbers with trailing spaces, possibly old data:

19601, 19685, 19719, 19725, 19753, 19915, 61400, 61284, 61133, 61147  
60442, 60272, 60302, 60227, 60243, 60263, 60159

## 2.4 The Matching System

The data flow diagram in Appendix I shows the approach used to search the Regional Register and update the logsheet data. As explained, this data is stored in three pairs of tables: one pair for each vessel type. Within each pair there is a one to many relation between the header (*trip*) table and the detail (*raw*) table.

The first process, **rscan**, makes a copy of the REGISTER table called REGISTER\_FFA and creates six additional fields. The added fields include a recommended vessel id number, *vid\_new*, which contains the suggested *vid* from the Regional Register, if possible. The second process, **rlogs**, reads this table and updates logsheet data tables for the given vessel type using the recommended *vid*. This way, the logsheet *boat* numbers that were assigned values above 60,000 are updated with their correct *vid* numbers from the Regional Register. Appendix II contains a listing of all fields for the REGISTER, REGISTER\_FFA, VCRNT and VHIST tables.

## 3. The Programs

Programs were written in C and in SQL\*Plus to assist in updating the FFA logsheet data. These are called by a set of Unix shell scripts that reside in the **/mnt/log/bin**

directory on the Agency HP 9000 mini computer.

As explained, the sequence of scripts create a new, temporary register table and matches SPC *boat* and FFA *vid* numbers wherever possible. Each match is assigned a confidence level from 1 to 24, 50 to 57 or 99, where 1 is a perfect match on vessel name, registration, vessel type, country and gross tonnage. A list of confidence levels and their meaning is provided in Appendix III.

Reports of partial matches are generated during this process for careful consideration and investigation. However, it will involve a great deal of time and research -- a full time project for 3 to 6 months for one FFA staff member.

### 3.1 Program Flow

The scripts that perform these tasks have been consolidated into a main script called **log\_upd.sh**, also in the **/mnt/log/bin** directory and performs the following tasks by logging into Oracle in the ADMIN account:

- It first runs **rscan.sh** which creates a new table, REGISTER\_FFA, and creates the necessary indexes. It is fully described in the section 3.3. The important fields in the REGISTER\_FFA table are *log\_update*, *match*, *trip*, *raw*, *new\_vid*. It attempts to match the SPC boat field with the FFA vid field, and puts a rating of confidence in the match (a number from 1 to 99) in the *match* field, and the new boat number to be used in the updates in the *new\_vid* field. Appendix III shows the query matrix used to find vessel matches and the associated confidence levels.
- It then runs **count\_logs.sh** to update the fields *trip* and *raw* in the REGISTER\_FFA table with a count of the number of records in the logsheet header (*trip*) and detail (*raw*) tables, respectively.
- Next, it runs **rscan\_rpt.sh** to produce a report on the numbers of logsheet records affected by each confidence level, for each fishing method.
- It then calls the two SQL start files **set\_f\_l** and **set\_f\_s** to update the *log\_update* field for either an entire group of records with the specified confidence level or individually selected records. It flags these records as being ready for update of their *boat* field by placing a code in the *log\_update* field of "F" (Free to Update).
- Finally, it runs **rlogs.sh**, which performs the update of *boat* numbers for the logsheet header records whose corresponding record in the REGISTER\_FFA table have been marked as "F" (Free to Update) and changes this mark to "Y" (Yes, Updated).

If further investigations reveal that a match chosen for a given record is correct and that the confidence level should be higher, the user can rerun the SQL start files **set\_f\_l** and **set\_f\_s** as well as **rlogs.sh**, causing further updates to the logsheet data as explained above.

For high confidence levels (eg. 1, 3, 4, 5, 50, 51, 52, 55), the entire level could be marked for logsheet update by calling **set\_f\_l**. For lower confidence levels, further investigations are needed to confirm the choice. Once this has been done, the **set\_f\_s** routine can be used to individually mark records in REGISTER\_FFA.

If the **rscan** process is interrupted for some reason, it can be restarted using the **resume\_rs.sh** script.

### 3.2 How Records are Marked

The field *log\_update* in REGISTER\_FFA controls all attempts to update the logsheet *boat* field. When REGISTER\_FFA is created, this field is set to "N", (No update). The two SQL start files **set\_f\_l** and **set\_f\_s** can be used to set *log\_update* to "F", (Free to Update), for a confidence level specified by the user and the REGISTER\_FFA *match* field, or for an individual record that meets all of the following conditions:

- *boat* is not equal to *new\_vid*;
- *gear* is one of 'L', 'P', 'S';
- logsheet *trip* and *raw* counts are non zero, AND
- *log\_update* is set to 'N'.

After updating logsheet *boat* fields, the **rlogs.sh** program changes the *log\_update* flag from "F" to "Y" (Yes, Updated). The user may manually override the above rules by directly setting the *log\_update* flag to any value. However, **rlogs.sh** always checks the above four conditions and will not attempt to find logsheet records to update unless they are met.

### 3.3 Results of the Automated Vessel Identification Search

The report script **rscan\_rpt.sh** was used to generate a set of tables that summarize the results of finding the best matching vessel id for each record.

About 47% of the 4802 records in REGISTER failed to show any match at all with RR. About 40% of records showed a match on all five attribute fields with the remainder showing degrees of partial matching. The spell matching program was able to recommend that the vessel name was indeed the one sought in about 50% of the cases in which it was called upon.

If logsheet *boat* numbers were updated according to all levels of match, close to 10% of logsheet records would be affected. In reality, matches at the higher confidence numbers must be checked manually before being allowed to be updated.

21 records were found to give multiple matches, code **99**. These occur due to the existence of vessels in the Regional Register which have several identical attribute fields, including name, and yet are distinct vessels. Some produces a confidence level of 2 which means that records match on all but registration number - even this is possible for completely distinct vessels. This result is a close enough match to warrant further investigation.

### 3.4 Conclusions and Recommendations

The present effort has been able to increase the total number of vessels linked to the Regional Register by only about 3%. Several hundred partial matches have also been made more specific by the use of the five attribute fields. Unrecognized vessels represent about 33% of the logsheet data.

The reason for some of the non-matching records has now been explained above. It is hard to believe that data entry errors alone are responsible for the large number of SPC vessel names and registration numbers that are simply not in the Regional Register.

In the long term, additional data entry and tracking procedures will have to have to be agreed on by SPC and FFA. The Regional Register needs to be updated to include the latest changes to vessels which appear on the logsheets. The SPC could also attempt to catch incorrect vessel attributes is at the time of the first logsheet data entry by including more rigorous checks and possibly including the radio call sign attribute with logsheet data. This would greatly ease the problem of vessel identification. The FFA's new policy of a yearly registration requirement for vessels (including a colour photo showing the radio call sign) will significantly assist the FFA in maintaining a correct register of vessels.

The **rscan** program used here is not able to find a vessel if both its name and registration number together are incorrect or garbled in some way. Additional matches in the Regional Register could probably be obtained by using a spelling algorithm on the registration number character string like that used with the vessel names.

For example in the case of *boat 430*, REGISTER reports a *reg\_no* of **BSA-530** and the Regional Register has **BS-A-530** for a vessel with *vid 430*. The name matching algorithm would have to be modified to handle registration numbers, and concepts such as aligning digits and separators such as dashes would have to be addressed.

The present query matrix method begins with the most restrictive queries and moves to the more inclusive. It is possible that the opposite approach might be more productive for records that show no match at all, that is, begin with more general queries, then use a spelling algorithm to search through the lists obtained for matching name and registration numbers.

In the immediate future there is a need to create a SQL\*Form to be used in conjunction with the REGISTER\_FFA, REGISTER, VCRNT and VHIST tables. This form would present records from the various tables on a single multi-block screen. It would help the user

decide if a match suggested by **rscan** is valid. If the match was valid it could mark the appropriate REGISTER\_FFA record as "free" for update, and could even perform the update of the affected logsheet *boat* numbers.

At present, most of the computing time is spent going through a large number of queries for the non-matching records. The number of queries could be reduced by first creating temporary tables containing indexed lists of all vessel names and registration numbers not in the Regional Register. This can be done quickly by using a SQL "outer join". Before attempting a class of query that involved either name or registration attributes, the program would check the temporary table. If the attribute was in the table, it would skip that class of query or try another strategy, eg. using the SQL\*Plus "like" function or string comparison functions.

The need for such speed improvements may diminish as the percentage of vessels recognized in the SPC data increases.

## 4. Technical Reference

This section goes into more detail as far as program execution, directory structures, Makefiles, and other technical information. It is presented for the use of system developers and software maintenance.

### 4.1 The Directory Structures

The files for the logsheet update system are contained in the following directories:

- /mnt/log/bin - Unix shell scripts
- /mnt/log/dat - log output files
- /mnt/log/doc - documentation text files
- /mnt/log/rscan - rscan program source and library files
- /mnt/log/rlogs - rlogs program source and library files
- /mnt/log/sql - SQL\*Plus "start" files

### 4.2 The **Rscan** Program

Program: Executable file is **rscan**.

Script File: rscan.sh executes the **rscan** program and sends its output to **/mnt/log/dat/rscan.log**. A sample log file is shown in Appendix IV.

**Purpose:** reads SPC REGISTER table to find best possible match for each vessel in the Regional Register.

**Usage:** rscan <user/password> n1|n1/n2|all|resume [trace]

where

**user/password:** The oracle account user name and password. This account needs select permission on ADMIN.REGISTER, ADMIN.VCRNT, ADMIN.VHIST.

**n1:** a single boat number.

**n1/n2:** a range of boat numbers.

**all:** the word "all" scans the whole table.

**resume:** the word "resume" resumes a scan of whole table that was interrupted.

**trace:** optional trace printing flag used in development and debug.

**Input:** ORACLE tables ADMIN.REGISTER, ADMIN.VCRNT, ADMIN.VHIST.

**Output:**ORACLE table REGISTER\_FFA. Informative and error messages go to standard output. NOTE: Any existing REGISTER\_FFA table is destroyed and should be renamed if there is a need to save it.

*Examples:*

**rscan user/password all**  
recreates REGISTER\_FFA in <user> account

**rscan user/password resume**  
resumes an <all> scan

**rscan user/password 450/455 t**  
does a trace run on the boat numbers from 450 to 455

#### 4.2.1 Program Function Overview

The program **rscan** creates an index on the *boat* field of the REGISTER\_FFA table and names it R\_FFA\_B. It will process a 4800 REGISTER table in about 2 hours with default priority.

Source code for this program is in **/mnt/log/rscan** with compilation controlled with Makefile. The Makefile is invoked by entering "go" in the **rscan** directory. A printout of all **rscan** source files can be obtained by using the **rscan\_list.sh** script.

#### 4.2.2 Include Files

These library files are in the **/mnt/log/rscan** directory.

defines.h: constant definitions  
structures.h: C structure declarations

#### 4.2.3 C Source File Descriptions

rscan.c:contains MAIN and controls the **rscan** process: checking command line arguments, logging on, scanning REGISTER table, computing statistics and logging off.

query.c:controls which query will be made next to the database and looks at results that come back. Computes intermediate statistics, gives details on any multiple matches, calls vessel name spelling routine if required and fills elements of output structure for inserting into output REGISTER\_FFA table.

utility.c:contains vessel name spelling comparison functions

#### 4.2.4 Pro\*C Source File Descriptions

rscan0.pc:contains basic Oracle utility routines: logon, logoff, commit, rollback and sql\_error.



rscan1.pc:controls fetching records from REGISTER table and loading attributes into a structure for passing on to other functions.

rscan2.pc:creates dynamic queries for searching VCRNT and VHIST. Returns match information and loads it into a structure.

rscan3.pc:creates and indexes the REGISTER\_FFA table.

rscan4.pc: inserts a record into REGISTER\_FFA.

rscan5.pc:in the case of a <resume> run, returns the *boat* number of the last REGISTER\_FFA record committed.

#### 4.2.5 Program Calling Hierarchy

The following list shows the functions that are contained in the various **rscan** programs and utilities. The indentation of the function names shows which function is called by which, not the order of execution. Source file names are shown at the right with ".c" extensions for C code files and ".pc" extensions for Pro\*C files.

<u>C Function Name</u>	<u>File Name</u>
main	rscan.c
log_on	rscan0.pc
check_start_end	rscan.c
do_scan	rscan.c
build_new_reg	rscan3.pc
find_max_record	rscan5.pc
get_register	rscan1.pc
open_cursor_r1	rscan1.pc
fetch_cursor_r1	rscan1.pc
close_cursor_r1	rscan1.pc
do_queries	query.c
get_rr_count	rscan2.pc
score_names	utility.c
parse_name	utility.c
compare_names	utility.c
put_ffa_reg	rscan4.pc
commit	rscan0.pc
log_off	rscan0.pc

#### 4.2.6 Error Messages

All Pro\*C functions call the function **sql\_error** which in turn calls the **roll\_back** and **log\_off** functions in the file rscan0.pc. In addition, the **rscan** Pro\*C embedded SQL statements have error trapping which produce error messages such as the following:

ORACLE ERROR: <Pro\*C function name>: <SQL statement summary>

followed by the text of the Oracle ORA type error message. If such an error occurs, the program forces a rollback, logs off Oracle and terminates. The program normally does a commit after every 100 records processed so the rollback would take you back to the last 100 committed.

**WARNING:** If **rscan** is run from an account other than ADMIN, then subsequent runs of one of the shell script files will not be able to find your REGISTER\_FFA table since all shell scripts are set up to use the ADMIN account.

#### 4.2.7 Using REGISTER\_FFA as Input to **rscan**

The program **rscan** can read the REGISTER\_FFA table as input and create another REGISTER\_FFA table as output. However, to do this, the input REGISTER table must be owned by ADMIN since **rscan** looks for the table ADMIN.REGISTER. First, rename or drop the existing REGISTER table and its indexes. Then rename REGISTER\_FFA to REGISTER and finally, run **rscan** by selecting the ????? option of the FMIS Logsheet Update Menu.

#### 4.2.8 Specifying a Range of *boat* Numbers for **rscan**

The program **rscan** will accept a single *boat* number or a range of *boat* numbers as an argument. The REGISTER\_FFA table would then consist of only this subset of the original REGISTER table. Remember that **rscan** drops the existing REGISTER\_FFA table each time you do this.

### 4.3 The **Rlogs** Program

Program: Executable file is **rlogs**

Script File: **rlogs.sh** executes the **rlogs** program and sends its output to **/mnt/log/dat/rlogs.log**. A sample log file is shown in Appendix V.

Purpose: reads the REGISTER\_FFA table to find which logsheet table *boat* numbers will be updated to *vid* values found by the **rscan** program.

**Usage:** **rlogs** <user/password> [trace]

where:

**user/password:** The oracle account user name and password. This account needs select permission on ADMIN.REGISTER, ADMIN.VCRNT, ADMIN.VHIST.

**trace:** optional trace printing flag used in development and debug.

**Input:** oracle table REGISTER\_FFA

**Output:** updates REGISTER\_FFA and *boat* number in logsheet data tables:

L\_TRIP\_FFA  
P\_TRIP\_FFA  
S\_TRIP\_FFA  
L\_RAW\_FFA  
P\_RAW\_FFA  
S\_RAW\_FFA

Informative messages and error messages are directed to standard output. The *log\_update* column of REGISTER\_FFA is changed from "F" to "Y" for those records where logsheets were updated with new *boat* numbers.

#### *Examples*

##### **rlogs user/password t**

runs the program to update REGISTER\_FFA with trace on, and debug output to standard output

In a timed test running at default priority and without other users locking the logsheet tables, **rlogs** updated all confidence level "1" logsheets in about 45 minutes. This represented a total of 1435 *trip* records and 39,899 *raw* records.

#### 4.3.1 Program Function Overview

Source code for this program is in the **/mnt/log/rlogs** directory. Compilation is controlled with the Makefile. This Makefile is invoked by entering "go" from the **rlogs** directory. A listing of **rlogs** source files can be obtained by using the **rlogs\_list.sh** script.

#### 4.3.2 Include Files

These reside in the local **/mnt/log/rlogs** directory.

defines.h: constant definitions

structures.h: C structure declarations

#### 4.3.3 C Source File Descriptions

rlogs.c: contains MAIN and controls the **rlogs** process: checking command line arguments, logging on, scanning REGISTER\_FFA table, computing statistics and logging off.

#### 4.3.4 Pro\*C Source File Descriptions

rlogs0.pc:contains basic Oracle utility routines: logon, logoff, commit, rollback and sql\_error.

rlogs1.pc:Retrieves records from the table REGISTER\_FFA and loads attributes into a structure to be passed to other functions. Also contains a routine which verifies whether records were or were not updated.

rlogs2.pc:updates the *boat* field in appropriate *raw* logsheet table records.

rlogs3.pc:updates the *boat* field in appropriate *trip* logsheet table records.

rlogs4.pc:updates the *log\_update* flag in REGISTER\_FFA.

#### 4.3.5 Program Calling Hierarchy

The following list shows the functions that make up the rlogs program. The level of indentation shows which function calls which, not the order of execution. Source file name are shown at the right with ".c" extensions for C source code files and ".pc" extensions for Pro\*C source files.

<u>C Function</u>	<u>File Name</u>
main	rlogs.c
log_on	rlogs0.pc
do_scan	rlogs.c
get_register_ffa	rlogs1.pc
open_cursor_r1	rlogs1.pc
fetch_cursor_r1	rlogs1.pc
close_cursor_r1	rlogs1.pc
update_raw	rlogs2.pc
update_trip	rlogs3.pc
update_ffa_current	rlogs4.pc
commit	rlogs0.pc
rollback	rlogs0.pc
check_no_update	rlogs1.pc
log_off	rlogs0.pc

#### 4.3.6 Error Messages

All **rlogs** Pro\*C embedded SQL statements have error trapping which produce error messages like the following. The Pro\*C functions call the function **sql\_error** which in turn calls the **roll\_back** and **log\_off** functions in the file rlogs0.pc.

ORACLE ERROR: <Pro\*C function name>: <SQL statement summary>

followed by the text of the Oracle ORA type error message. If such an error occurs, the

program forces a rollback, logs off Oracle and terminates. The program normally does a commit after every completed transaction, so only the last transaction is lost. A transaction in this instance consists of updating a given set of *trip* and *raw* records that share the same *boat* number and updating the corresponding REGISTER\_FFA *log\_update* field from "F" to "Y".

#### 4.4 SQL Scripts

The following section contains a list of SQL\*Plus scripts used in this project. Some of these are useful diagnostic tools and others are scripts to create indexes used in the project. They are START files, meaning they must be invoked by using the **start** command from within Oracle, or from the Unix system prompt with the following syntax:

```
sqlplus username/password @start_file
```

##### 4.4.1 List of SQL Scripts and their Function

ch\_index.sql creates indexes on "vname" and "vregn" fields in Regional Register tables VCRNT and VHIST:

Index	Index Name
VCRNT(VNAME)	VC_VN
VCRNT(VREGN)	VC_RN
VHIST(VNAME)	VH_VN
VHIST(VREGN)	VH_RN

count\_logs.sql counts total **trip** and **raw** logsheet records for each record in REGISTER\_FFA table. Updates the fields *trip\_count* and *raw\_count*.

count\_no\_nm.sql: example of use of outer join to count all vessel names that are not in Regional Register.

rscan\_rpt.sql: generates six tables that summarize log sheet records affected by "rscan" queries of REGISTER table.

set\_f\_l.sql: sets *log\_update* field in REGISTER\_FFA to "F" for a specified confidence or query level.

**Usage:** sqlplus user/password @set\_f\_l <level\_no>  
where <level\_no> is the confidence or query level.

set\_f\_s.sql: sets *log\_update* column in REGISTER\_FFA to "F" for a record with specified *boat* number.

**Usage:** sqlplus user/password @set\_f\_s <boat\_no>

where <boat\_no> is the value of *boat* for that record.

show\_ra.sql:shows the record from REGISTER\_FFA for the specified *boat* number.

show\_rb.sql:shows a record from REGISTER given a *boat* number.

show\_rf.sql:shows a record from REGISTER\_FFA given a *vid* number.

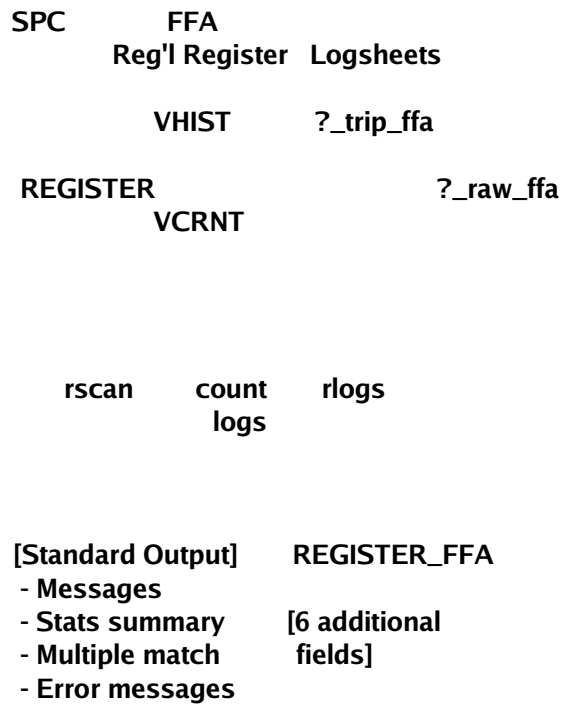
show\_rg.sql:shows a record from REGISTER given a *vid* number.

show\_rr.sql:shows records in Regional Register VCRNT and VHIST tables given a *vid* number. Only a subset of the many attributes are displayed.

tr\_index.sql:script used to create an index on "boat" for the "trip" and "raw" logsheet tables:

Index	Index Name
L_TRIP_FFA(BOAT)	L_T_B
P_TRIP_FFA(BOAT)	P_T_B
S_TRIP_FFA(BOAT)	S_T_B
L_RAW_FFA(BOAT)	L_R_B
P_RAW_FFA(BOAT)	P_R_B
S_RAW_FFA(BOAT)	S_R_B

## APPENDIX I - Data Flow Diagram



## APPENDIX II - Database Fields

Oracle description of tables used by the "rscan" program

**Table : REGISTER (SPC)**

Name	Null?	Type
-----	-----	-----
BOAT	NOT NULL	NUMBER(5)
VESSEL	NOT NULL	CHAR(30)
REG_NO		CHAR(12)
RDATE		DATE
FL		CHAR(2)
GR		CHAR(1)
GRT		NUMBER(4)
OLD		NUMBER(5)
FR		CHAR(2)
VID		NUMBER(5)
VDEG		CHAR(1)

**Table : REGISTER\_FFA (newly created)**

Name	Null?	Type
-----	-----	-----
BOAT	NOT NULL	NUMBER(5)
VESSEL	NOT NULL	CHAR(30)
REG_NO		CHAR(12)
RDATE		DATE
FL		CHAR(2)
GR		CHAR(1)
GRT		NUMBER(4)
OLD		NUMBER(5)
FR		CHAR(2)
VID		NUMBER(5)
VDEG		CHAR(1)
NEW_VID		NUMBER(5)
RR_TABLE		CHAR(1)
MATCH		NUMBER(2)
LOG_UPDATE		CHAR(1)
TRIP_COUNT		NUMBER(5)
RAW_COUNT		NUMBER(5)



[Additional fields in **bold**].

**Tables : VCRNT and VHIST (FFA)**

Name	Null?	Type
VID	NOT NULL	NUMBER(5)
VNAME	NOT NULL	CHAR(30)
VREGN		CHAR(12)
VRADIO		CHAR(8)
DDATE	NOT NULL	DATE
RPERIOD		CHAR(5)
FLAG		CHAR(2)
VTYP		CHAR(2)
STDG		CHAR(1)
BASCNT1		CHAR(2)
BASPORT1		CHAR(10)
BASCNT2		CHAR(2)
BASPORT2		CHAR(10)
OWNID		NUMBER(5)
CHRTID		NUMBER(5)
VMASID		NUMBER(5)
FMASTID		NUMBER(5)
APPLID		NUMBER(5)
CREWNO		NUMBER(3)
HULL		CHAR(2)
CONYR		CHAR(4)
CONPLC		CHAR(10)
CONCNT		CHAR(2)
FAREA		CHAR(6)
GRT		NUMBER(7,2)
LENGTH		NUMBER(7,2)
POWER		NUMBER(5)
FUEL		NUMBER(5)
RFLAG		CHAR(1)
STRTP1		CHAR(2)
STRCP1		NUMBER(5)
STRTM1		NUMBER(3)
STRTP2		CHAR(2)
STRCP2		NUMBER(5)
STRTM2		NUMBER(3)
FRZTP1		CHAR(2)
FRZRT1		NUMBER(4)
FRZTM1		NUMBER(3)
FRZTP2		CHAR(2)
FRZRT2		NUMBER(4)

FRZTM2  
BTSTP1  
BTSCP1  
BTSTP2  
BTSCP2  
NETLEN  
NETDEP  
SSACT1  
SSACT2

NUMBER(3)  
CHAR(2)  
NUMBER(4)  
CHAR(2)  
NUMBER(4)  
NUMBER(5)  
NUMBER(4)  
CHAR(30)  
CHAR(30)

### APPENDIX III - Query Matrix and Confidence Levels

The following table shows the query combinations used to search for a vessel given some or all of the five attributes shown. In the case of Gross Registered Tonnage (GRT), a range of plus and minus 5% about the nominal value in the Regional Register was considered a match. Records where GRT is recorded as zero were taken to be null (not available). Except where there is no match at all (level 0), the higher the level code the fewer the fields, or the weaker the match. In the table, an "X" indicates a match on the field; blank indicates no match.

#### CONFIDENCE

<u>LEVEL</u>	<u>NAME</u>	<u>REG NO.</u>	<u>TYPE</u>	<u>FLAG</u>	<u>GRT +/-5%</u>
0					
1		X	X X	X	X
2		X		X	X X
3			X X	X	X
4		X	X X	X	
5		X	X	X	X
6			X X	X	
7		X	X X		X
8		X		X	X
9			X X		X
10		X	X X		
11		X		X	
12			X X		
13*		X		X	X
14		X			X X
15			X	X	X
16		X	X	X	
17		X			X
18			X	X	
19		X	X		X
20		X			X
21			X		X
22		X	X		
23		X			
24			X		

50 = Matches at level 3 and name spelling is similar to that in Regional Register

51 = Matches at level 6 and name spelling is similar to that in Regional Register

52 = Matches at level 9 and name spelling is similar to that in Regional Register

53 = Matches at level 12 and name spelling is similar to that in Regional Register

54 = Matches at level 15 and name spelling is similar to that in Regional Register

55 = Matches at level 18 and name spelling is similar to that in Regional Register

56 = Matches at level 21 and name spelling is similar to that in Regional Register  
57 = Matches at level 24 and name spelling is similar to that in Regional Register  
99 = A confidence level of 1 through 24, but where the query returned more than one matching record from the Regional Register.

\* Query level 13 was not used because it tended to produce false matches. This is due to the existence of different vessels that actually match on name, vessel type and country. If a match was obtained on fields other than vessel name, then a comparison of the spelling between names was made. If seen to be really the same name, the record was flagged with one of the "50" codes above.

## APPENDIX IV - Sample RSCAN log file

Starting rscan Program: Logging on to oracle

Creating and Indexing REGISTER\_FFA output table: Scanning  
input REGISTER table.

At confidence level code 11 found 2 Regional Register matches  
for this REGISTER record:

Boat	Vessel Registration	VType	Flag
19307	EBISU 8 ON2-419		L J

more "multiple match" type records if found

### Statistics on Scan of REGISTER Table

Matches	Total	LL	PL	PS	Other
All	1915	1587	135	193	0
Name + Regist.	0	0	0	0	0
Name + Type	63	51	4	7	1
Regist. + Type	0	0	0	0	0
Other Matches	517	383	35	99	0
None	2286	1677	543	42	24
Multiple Match	21	18	3	0	0
	----	----	---	---	--
Totals	4802	3716	720	341	25
	====	====	===	===	==

Total records processed = 4802

Start time: Mon May 6 14:26:27 1991  
Finish time: Mon May 6 16:21:21 1991  
Elapse time: 01:54:54 hh:mm:ss (6894 sec)

Logging off oracle: Normal Completion of rscan Program

-----  
[End of sample log file]

### Comments on Sample RSCAN log file

The REGISTER table records that returned a multiple match from Regional Register are logged between the pairs of bars as in the example above. The Statistics table

presents a brief summary of the matches obtained, total records processed and elapsed time. Categories such as the **Name + Registration** show no matches in this example because it is actually quite low (confidence level 22) in the query matrix. If a match is made, it occurs well before this and with more attributes.

**Registration + Type** is also relatively low on the query list (confidence level 12) and shows no matches here. The tables generated by the **rscan\_rpt.sh** script are much more detailed than the above table.

#### APPENDIX V - Sample RLOGS log file

Starting rlogs Program: Logging on to oracle.

Scanning REGISTER\_FFA table.

#### Statistics on Scan of REGISTER\_FFA Table

Record Type	Total	LL	PL	PS
Qualify for Update	296	263	7	26
Actually Updated	296	263	7	26
Trip Records Updated	1435	1056	145	234
Raw Records Updated	39899	27753	1435	10711

Start time: Thu May 9 12:08:56 1991  
 Finish time: Thu May 9 12:48:20 1991  
 Elapse time: 00:39:24 hh:mm:ss (2364 sec)

Logging off oracle: Normal Completion of rlogs Program

-----  
 [End of sample log file]

#### Comments on Sample RLOGS log file:

Rows in the statistics table are as follows:

"Qualify for Update":those REGISTER\_FFA records marked as being "Free to Update" where "boat" is not equal to "new\_vid" and logsheet counts

are non zero.

"Actually Updated":number of committed transactions.

"Trip Records Updated": "trip" table records with "boat" numbers updated

"Raw Records Updated": "raw" table records with "boat" numbers updated



### **Additional RLOGS Messages:**

If not all REGISTER\_FFA records marked with "F" are updated then the following message will appear:

Logsheets data for <n> records marked with 'F' in REGISTER\_FFA not updated.

REGISTER\_FFA Records must meet these criteria before logsheets will be updated:

- new vid is not equal to existing *boat* number;
- both *trip* and *raw* logsheet data must exist for the record;
- the gear type is 'L', 'P' or 'S'.

This informs the user that <n> records where "F" was requested could not have their logsheet *boat* numbers changed because it did not meet one of the above criteria.

## APPENDIX VI - Summary of Shell File Scripts

count\_logs.sh: updates REGISTER\_FFA with "trip" and "raw" logsheet counts.

resume\_rs.sh: continues and interrupted "rscan" "all" run.

rlogs.sh: updates logsheet "trip" and "raw" table "boat" number.

rlogs\_list.sh: prints "rlogs" program source files.

rscan.sh: generates REGISTER\_FFA table to find best "vid" numbers.

rscan\_list.sh: prints "rscan" program source files.

rscan\_rpt.sh: generates six report tables from REGISTER\_FFA.