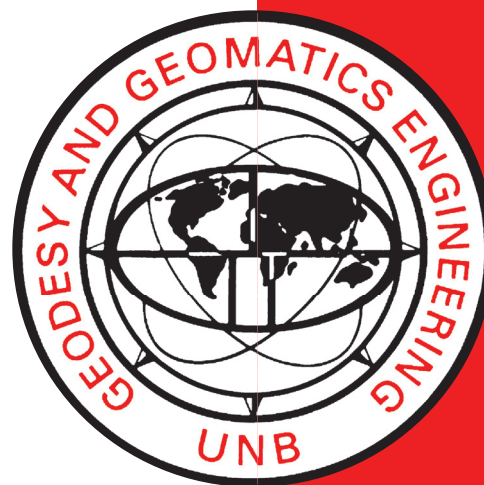


U. S. COAST GUARD MARINE ACCIDENT DATA BASE

**T. HOTTENDORFF
M. V. HULLMEINE**

August 1992



**TECHNICAL REPORT
NO. 159**

PREFACE

In order to make our extensive series of technical reports more readily available, we have scanned the old master copies and produced electronic versions in Portable Document Format. The quality of the images varies depending on the quality of the originals. The images have not been converted to searchable text.

U.S. COAST GUARD MARINE ACCIDENT DATA BASE

Thomas Hottendorff
Marcia V. Hullmeine

Department of Surveying Engineering
University of New Brunswick
P.O. Box 4400
Fredericton, N.B.
Canada
E3B 5A3

August 1992

PREFACE

This technical report is a reproduction of a report submitted by participants of the University of New Brunswick—University of Hannover (Germany) exchange program. The research was supervised by Mark E. Doucette of Watermark, Fredericton, and data was provided by Alex MacDonald of Strategic Ventures Corporation, Fredericton. Student support was provided by the German government's German Academic Exchange Service (DAAD), and by Dr. David E. Wells through a grant from the Natural Sciences and Engineering Research Council of Canada.

As with any copyrighted material, permission to reprint or quote extensively from this report must be received from the author. The citation to this work should appear as follows:

Hottendorff, T., and M.V. Hullmeine (1992). "U.S. Coast Guard marine accident data base." Department of Surveying Engineering Technical Report No. 159, University of New Brunswick, Fredericton, New Brunswick, Canada, 105 pp.

Abstract

Over the last years different systems of Electronic Navigational Charts (ENC) have been developed. Different companies distributed different mapped areas with their packages, but the official installation of ENCs as a navigational tool is far from complete. To get a picture of where installation of ENCs should have priority, this study on the Marine Accident Database (MAD) of the U.S. Coastguard was done. The goal was to input the data into different data management systems, access and visualize it. To accomplish this task several computer systems and software programmes were used: a PC running inFocus (a combination of QUIKMap and FoxPro) and a SUN workstation running INGRES (a relational database).

Towards the end of the study statistics were derived of different aspects of the accidents compiled in the MAD, not only concerning their geographical distribution, but also their more detailed circumstances.

Table of Contents

I. The Marine Accident Database (MAD)

1. Introduction

2. Data Processing

II. The Software

1. Zenith PC

a) inFOcus

b) Quattro Pro IV

2. Sun Workstation

a) INGRES

III. Auxiliary Programmes in C

IV. Data Transfer

V. Analysis of the Data

VI. Conclusion

VII. Appendices

Appendix 1: U.S. Coastguard Accident Form and Letter

Appendix 2: INHOUSE CASMAIN Data Dictionary

Appendix 3: Tables for inFOcus and INGRES

Appendix 4: Graphs of the Data and Derived Statistics

Appendix 5: Source Codes of Auxiliary Programmes in C

I. The Marine Accident Database (MAD)

1. Introduction

The Marine Accident Database (MAD) was established by the U.S. Coastguard in the early 80's. It is a compilation of marine accidents in U.S.waters (all vessels) and in foreign waters (US vessels only), (for more detailed information see part 2 of the form CG-2692 in appendix 1).

The database consists of two files, one for the vessel casualties (CAS), the other for personal casualties (PCAS). For the purpose of this study only the vessel file CAS was used. The CAS file comprised 64980 records (~38MByte) as of May 20, 1992. The records are organized in 60 entries of 10 character wide fields that follow mostly the INHOUSE CASMAIN Data Dictionary, Nov. 1991 by the Office of Marine Safety, Security and Environmental Protection, Marine Investigation Division, Marine Safety Evaluation Branch, G-MMI-3 (see appendix 2).

A quality control against the database has only been performed through 1989, which explains the poor quality of the data since then. Much data is missing and case numbers are duplicated which makes unique data retrieval very difficult. On the other hand one gets much redundant data, because there is a record for each vessel that was damaged in the accident, so there are for example 10 records of an accident that involved 10 vessels although all other data are the same.

Some data was of no use at all for us because there were no entries concerning the location of the accidents. Others were referenced in coordinates of which we had no origins, e.g. river mileposts (see chapter V).

2. Data Processing

The purpose of this project has been to set up the database provided by the US Coast Guard for a GIS and to implement it into a relational database such as INGRES. Since the original data file is not suitable for a relational database some changes and additions had to be made. Most importantly we had to find a way to reduce the redundancy in the database. To accomplish

this task we decided to split up the original file into four database files or tables. The three tables `weather`, `accident` and `location` contain only data that refers to the accident itself. The `vessel` table comprises the data describing the vessels damaged in the accident. Due to the nature of the data the latter table contains considerably more records than the former ones. The transformation of the original file to these four tables was done in four steps.

The first step was to read in the records of the original file field by field and to store them in comma delimited ASCII files. Four small C-programs called `location.c`, `accident.c`, `weather.c` and `vessel.c` have been written for this task. Additionally, two fields were added to each record, a `record#` field which contains a distinct number for each record in order to improve the accessibility and a field containing a distinct number for each accident. This field is yet to be filled.

In a second step the file `location.dat` was imported to INGRES and all distinct combinations of the fields `case_nr`, `waterbody`, `latitude`, `longitude` and `rivermp` were extracted. These fields were copied into an ASCII file called `position`. We obtained 38255 records with different locations.

In the next step a program called `accsearch.c` compares the files `position` and `location.dat` and creates an accident number for each record in `position`. At the same time the field `acc_nr` in `location` is being filled and a look-up table is being created containing record number and accident number. This look-up table is subsequently used to fill the accident number fields in the other three data files.

Finally all records having duplicate accident numbers had to be removed from the files `location.dat`, `accident.dat` and `weather.dat`. Three small C-programs called `uniloc.c`, `uniacc.c` and `uniwthr.c` were written to do that. Simultaneously, all records without any location were removed from the files. The resulting number of records in the `vessel` table is now 61079.

Some deficiencies of our approach have to be pointed out. Since we are using the location given in each record to discriminate between accidents there

might be some blunders in the current set-up. To achieve a more reliable solution for instance the date of the accident could be taken into account to discriminate between accidents.

Furthermore there are still many mistakes and input errors in the original data which could result in a faulty analysis. A quality control should be performed that checks the database for inconsistencies. Plotting the accidents for instance revealed quite a few input mistakes. Checking the coordinates against the waterbody could help to detect and correct these errors. Also, the empty fields should be filled with a value since these fields could otherwise result in errors.

The four files described above form the core of the database that was to be created. Some smaller tables that translate the codes given in the core files have been created using the data dictionary (see Appendix).

II. The Software

To work with the data of the MAD we were able to use two different computer systems: a Zenith PC (80386 CPU, 33 MHz) and a SUN Workstation. This chapter will shortly describe what we did with certain software packages, it is not a complete description of what can be done with them. For that kind of information please refer to the appropriate manuals.

The main task was to import the raw data (in ASCII format) into database management systems. In order to do that it was not only necessary to get familiar with the systems but also to manipulate the format of the original data in so far as to be able to use it with the programmes. This was done with several auxiliary programmes written in C (see chapter III).

1. Zenith PC

a) inFOcus, Beta Version 2.5

The inFOcus package consists of the database Fox Pro and the mapping program QUIKMap and was developed by Earth & Ocean Research, Dartmouth, N.S. We used inFOcus to manage the data on the PC rather

than using its facility as a GIS. The problem here was also that we did not have access to a worldmap but only to a map of North America.

The original data was split up into several tables (see appendix 3) by auxiliary programmes (see chapter III) in order to have more flexibility analysing the data and to avoid too much redundant data. The `accident` table was created to be the master or reference file that contained all the accident numbers, the date and how many vessels were involved in the casualty. The `location` table is the reference as to where the casualty occurred. Almost all the accidents had a district specification, about half had coordinates, only a minute part didn't have any information at all. The third and largest table is the `vessel` table that includes all information known about the vessels involved in the accidents, their identification, specifications and damages. The second part of that table describes the causes and the nature of the accidents. The last large table is the weather table that includes all the information about the weather at the time of the accident and the period of day, e.g. fog at night.

With the VIEWS module it was possible to organize the tables in views that allow to query about more than one table at a time. This proved to be very convenient especially to convert codes from the original data to text that explained the codes so the user didn't have to deal with the for him meaningless abbreviations (the text was taken from the data dictionary, see appendix 2).

Only simple queries can be asked in inFOcus, graphing is not possible at all without another optional module that we didn't have access to, so the whole set-up is more or less a mapping tool.

If a table contains geographical coordinates it is possible to convert the database into a map database which can then be used to display the data in a map. QUIKMap provides the user with a sufficient basemap for North America and has also a zoom-in option. Symbols can be chosen from an existing library or can be created by the user. Because of the time constraint we chose the first method.

Making the accidents around North America visible on a map served as a first estimate of where the most accidents occurred. What was more, it showed that there were more input errors (refer to letter from the US Coastguard, appendix 1) than expected: groundings in the middle of Greenland and in the middle of the Atlantic. Another 'feature' were accidents that seemed to follow a stream, only they were translated by a certain distance.

b) Quattro Pro IV

Quattro Pro is a spreadsheet package that we used to visualize the data and the statistics. Simple ASCII files can be used to derive bar graphs, rotated bar graphs and pie charts. Examples of the output can be found in appendix 4. For further discussion of the contents of these graphs refer to chapter V.

2. SUN Workstation

a) INGRES 6.3

INGRES is a relational database management system that was used to organize the data in a similar way to the inFOCUS approach. The division into tables is the same (see appendix 3). In INGRES we could ask the same queries, we also used it to derive the statistics needed for the visualization in Quattro Pro.

Although the query language is a little bit closer to English, the need for somebody familiar with the program is still vital. Since SQL (the command language used by INGRES) is much more flexible than the query language of FoxPro it is possible to ask much more complex questions.

The possibility for more complex queries and especially that to count entries, calculate averages etc. turned out to be most useful for analysing the data in the required way and coming up with statistics to put into Quattro Pro. Only with the next version of INGRES it will be possible to directly output graphs and histograms from the derived statistics.

III. Auxiliary Programmes in C

To input the raw data from the original file CAS into INGRES and later inFOcus, it had to be slightly converted into a form readable by the programmes. This was done by auxiliary programmes written in C. The programmes `accident.c`, `location.c`, `vessel.c` and `weather.c` converted the original data into four respective files suitable for input into the database management systems (see appendix 3 for the tables and appendix 5 for the source codes in C).

Other minor programmes were used to manipulate the files to transfer them from one system to the other, like erasing the blanks that were not readable by one system, chopping files into handy pieces to send them (via network or floppy disk) from one machine to another or merging them.

Another task accomplished with auxiliary programmes was the separation of accidents. Since the original CAS file had one record for each vessel, means had to be found to list the database by accident. This was done by introducing an accident number that is the same for all vessels involved in the same accident. We distinguished the accidents by searching for a distinct combination of case number, waterbody, coordinates and river mileposts. This was necessary because the case number provided by the coastguard turned out to be not unique after 1989. To have an overall reference a record number was added to the data to be able to reference an entry throughout all the tables.

IV. Data Transfer

Most of the data transfer was done manually, which means the data from one system was converted into a compact ASCII, then copied onto a floppy disk and imported to the other system.

This method was rather timeconsuming but could not be avoided with the current set-up. If it would be decided to continue this project this is one of the most important points to be improved.

V. Analysis of the Data

The procedure we followed to analyse the data was to create the appropriate ASCII files using INGRES on the SUN station and then to transfer the data to Quattro Pro onto the PC.

The first consideration in analysing the data had to be whether all the accidents could be visualized through coordinates or at least associated with a district (see chapter II.1.a). The next step was to find out if the database is a global one or not. This question was answered by the US Coastguard in a letter (see appendix 1). If a US vessel is involved in a casualty, it is recorded world-wide, whereas foreign vessels are only recorded in US waters. The consequence of this procedure is that the database is really representative only in US waters.

Because of the poor quality of the original database, we had to uniquely identify the different accidents. To produce statistics and graphs criteria had to be found to classify the data, such as location of accident, cause and nature of accident and type of vessel.

In the graph 'Nature of Accidents vs. Number' (see appendix 4) it is interesting to notice that the vast majority of the accidents were groundings and collisions. These are situations where an ENC would be able to send out audible or visible warnings as to the fact that the vessel is close to a too shallow depth or to another vessel. Further analysis of the data is recommended. The groundings and collisions should be correlated with the causes of the accidents.

In order to evaluate how the different natures of the accidents are distributed among different types of vessels the data was split up accordingly as can be seen in the graphs 'Accidents by Type of Vessel'. The pie chart shows that the accidents are rather evenly distributed among freighters, fishing boats, tow boats and tankers. In case of the tow boats the large number of barges has to be taken into account. Since there is an database record for each damaged barge we have a higher percentage here than we would have expected. The bar graph gives further insight into the nature of the accident. While fishing boats make up the vast majority of

material failures, fires, explosions and foundering, collisions and groundings are rather suffered by freighters, tow boats and tankers. A pie chart with percentages for each type of vessel appears to be a useful tool for further investigation.

VI. Conclusion

It was very interesting to get to know several database management systems but it must also be said that the incompatibility of the different systems created a time consuming problem; very often one was busy converting existing data into another format in order to use it on another machine.

The analysis and visualization of the data does not seem to be efficient with the present set-up. To use all the different components one needs considerably too much time to get to know the programmes and the data transfer has to be done manually or with auxiliary programmes. Moreover the present set-up of inFOcus is far from complete. The powerful tools to create reports haven't been exploited yet. On the other hand the program turned out to be rather slow since a large amount of data has to be managed. Other deficiencies of the inFOcus package are that it is not possible to create reports in QUIKMap by pointing at a symbol when information about a related file is desired. One would have to create a combined database file in inFOcus which is not very practical when a large amount of data is used. Another approach to this problem could be to tag the symbol in QUIKMap and then to display a report in inFOcus. Unfortunately, inFOcus does not provide the means to create macros, so that this approach requires a trained user and quite some time.

One recommendation would be the purchase of the whole FoxPro Database Package or at least the manual in order to be able to use all functions, such as the command line interpreter and programming facilities.

Appendix 1:

U.S. Coastguard Accident Form and Letter

U.S. Department
of Transportation
**United States
Coast Guard**



10
Commandant
United States Coast Guard

Washington, DC 20593
Staff Symbol: G-MMI-3/24
Phone: 202-267-1417
16732

JUN 23 1991

Mr. Thomas Hottendorff
University of New Brunswick
Department of Surveying Engineering
Post Office Box 4400
Fredericton, N.B.
Canada E3B 5A3

Dear Mr. Hottendorff:

This letter is in reply to your letter received by the Marine Investigation Division on June 18, 1992. In answer to your questions:

1. Could you indicate the units used for the length, gross, tonnage, windspeed and visibility?
Length = linear feet.
Gross tonnage = a unit of internal capacity for ships equal to 100 cubic feet.
Windspeed = mph.
Visibility = in tenths of miles, ex. 1.6 = 1 mile and 6 tenths.
2. Is the date given in each record the actual date of the accident? Yes
3. What are the criteria used to collect the data? Does it depend on the damage? Refer to the enclosed accident form CG-2692, part 2,
4. Did you only collect from accidents where American ships were involved? If a foreign vessel is involved in a casualty in U.S. waters we collect the data. Refer to the enclosed accident form CG-2692, part 2.
5. The newer records show duplicate case numbers for apparently different accidents. Why is this and what do you use as unique key? Quality control against the database has only been performed through calendar year 1989. Quality control usually corrects this. Vessels with the same casualty date and case number are involved in the same casualty the vast majority of the time. A vessels VIN is unique.
6. We mapped all records that have co-ordinates. Those show some discrepancies such as groundings in the middle of Greenland or in the middle of the Atlantic. Why is that? Data entry mistakes.

7. Some accidents are mapped off a river but they seem to follow the river. Is it sufficient to just translate these accidents to the river? Probably, but to be sure we would have to look at each individual case on microfilm.

8. Can you suggest a way to convert river mileposts to co-ordinates? Is there a database where the mileposts are coordinated? I would locate the milepost on a U.S. Army Corp of Engineers chart and then look at the grid for the corresponding latitude and longitude. I know of no database that coordinates mileposts with latitude and longitude.

9. We trust this information is suitable for your needs.

Sincerely,



S. S. SHEEK
Commander, U.S. Coast Guard
Chief, Marine Safety Evaluation Branch
By direction of the Commandant

Encl: CG-Form-2692

INSTRUCTIONS
FOR COMPLETION OF FORM CG-2692
REPORT OF MARINE ACCIDENT, INJURY OR DEATH
AND FORM CG-2692A, BARGE ADDENDUM

WHEN TO USE THIS FORM

1. This form satisfies the requirements for written reports of accidents found in the Code of Federal Regulations for vessels, Outer Continental Shelf (OCS) facilities, mobile offshore drilling units (MODUs), and diving. The kinds of accidents that must be reported are described in the following instructions.

VESSELS

2. A vessel accident must be reported if it occurs upon the navigable waters of the U.S., its territories or possessions; or whenever an accident involves a U.S. vessel wherever the accident may occur. (Public vessels and recreational vessels are excepted from these reporting requirements.) The accident must also involve one of the following (ref. 46 CFR 4.05-1):

A. All accidental groundings and any intentional grounding which also meets any of the other reporting criteria or creates a hazard to navigation, the environment, or the safety of the vessel;

B. Loss of main propulsion or primary steering, or an associated component or control system, the loss of which causes a reduction of the maneuvering capabilities of the vessel. Loss means that systems, component parts, subsystems, or control systems do not perform the specified or required function;

C. An occurrence materially and adversely affecting the vessel's seaworthiness or fitness for service or route including but not limited to fire, flooding, failure or damage to fixed fire extinguishing systems, lifesaving equipment or bilge pumping systems;

D. Loss of life;

E. Injury causing any person to be incapacitated for a period in excess of 72 hours.

F. An occurrence not meeting any of the above criteria but resulting in damage to property in excess of \$25,000. Damage cost includes the cost of labor and material to restore the property to the condition which existed prior to the casualty, but it does not include the cost of salvage, cleaning, gas freeing, drydocking or demurrage.

MOBILE OFFSHORE DRILLING UNITS

3. MODUs are vessels and are required to report an accident that results in any of the events listed by Instruction 2-A through 2-F for vessels. (Ref. 46 CFR 4.05-1, 46 CFR 109.411)

OCS FACILITIES

4. All OCS facilities (except mobile offshore drilling units) engaged in mineral exploration, development or production activities on the Outer Continental Shelf of the U.S. are required by 33 CFR 145.30 to report accidents resulting in:

A. Death;

B. Injury to 5 or more persons in a single incident;

C. Injury causing any person to be incapacitated for more than 72 hours;

D. Damage affecting the usefulness of primary lifesaving or firefighting equipment;

E. Damage to the facility in excess of \$25,000 resulting from a collision by a vessel;

F. Damage to a floating OCS facility in excess of \$25,000.

5. Foreign vessels engaged in mineral exploration, development or production on the U. S. Outer Continental Shelf, other than vessels already required to report by Instructions 2 and 3 above, are required by 33 CFR 145.303 to report casualties that result in any of the following:

A. Death;

B. Injury to 5 or more persons in a single incident;

C. Injury causing any person to be incapacitated for more than 72 hours.

DIVING

6. Diving casualties include injury or death that occurs while using underwater breathing apparatus while diving from a vessel or OCS facility.

A. **COMMERCIAL DIVING.** A dive is considered commercial if it is for commercial purposes from a vessel required to have a Coast Guard certificate of inspection, from an OCS facility or in its related safety zone or in a related activity, at a deepwater port or in its safety zone. Casualties that occur during commercial dives are covered by 46 CFR 197.486 if they result in:

1. Loss of life;
2. Injury causing incapacitation over 72 hours;
3. Injury requiring hospitalization over 24 hours.

In addition to the information requested on this form, also provide the name of the diving supervisor and, if applicable, a detailed report on gas embolism or decompression sickness as required by 46 CFR 197.410(a)(9).

Exempt from the commercial category are dives for:

1. Marine science research by educational institutions;
2. Research in diving equipment and technology;
3. Search and Rescue controlled by a government agency.

B. **ALL OTHER DIVING.** Diving accidents not covered by Instruction (6-A) but involving vessels subject to Instruction (2), **VESSELS**, must be reported if they result in death or injury causing incapacitation over 72 hours.
(Ref. 46 CFR 4.03-1(c)).

HAZARDOUS MATERIALS

7. When an accident involves hazardous materials, public and environmental health and safety require immediate action. As soon as any person in charge of a vessel or facility has knowledge of a release or discharge of oil or a hazardous substance, that person is required to immediately notify the U. S. Department of Transportation's National Response Center (telephone toll-free 800-424-8802 - in the Washington, D.C., area call 202-426-2675). Anyone else knowing of a pollution incident is encouraged to use the toll-free telephone number to report it. If etiologic (disease causing) agents are involved, call the U. S. Public Health Service's Center for Disease Control in Atlanta, Ga. (telephone 404-633-5313). (Ref. 42 USC 9603; 33 CFR 153; 49 CFR 171.5)

COMPLETION OF THIS FORM

8. This form should be filled out as completely and accurately as possible. Please type or print clearly. Fill in all blanks that apply to the kind of accident that has occurred. If a question is not applicable, the abbreviation "NA" should be entered in that space. If an answer is unknown and cannot be obtained, the abbreviation "UNK" should be entered in that space. If "NONE" is the correct response, then enter it in that space.

9. When this form has been completed, deliver or mail it as soon as possible to the Coast Guard Marine Safety or Marine Inspection Office nearest to the location of the casualty or, if at sea, nearest to the port of first arrival.

10. Amplifying information for completing the form:

A. **Block 15 - "LOCATION"** - Latitude and longitude to the nearest tenth of a minute should always be entered except in those rivers and waterways where a mile marker system is commonly used. In these cases, the mile number to the nearest tenth of a mile should be entered. If the latitude and longitude, or mile number, are unknown, reference to a known landmark or object (buoy, light, etc.) with distance and bearing to the object is permissible. Always identify the body of water or waterway referred to.

B. **Tug or towboat with tow** - Tugs or towboats with tows under their control should complete all applicable portions of the CG-2692. SECTION II should be completed if a barge causes or sustains damage or meets any other reporting criteria. If additional barges require reporting, the "Barge Addendum," CG-2692A, may be used to provide the information for the additional barges.

C. **Moored/Anchored Barge** - If a barge suffers a casualty while moored or anchored, or breaks away from its moorage, and causes or sustains reportable damages or meets any other reporting criteria, enter the location of its moorage in Block (1) of the CG-2692 and complete the form except for Blocks (2) through (13). The details will be entered in SECTION II for one barge and on the "Barge Addendum," CG-2692A, for additional barges.

D. **SECTION III - Personnel Accident Information** - SECTION III must be completed for a death or injury. In addition, applicable portions of SECTIONS I, II and IV must be completed. If more than one death or injury occurs in a single incident, complete one CG-2692 for one of the persons injured or killed, and attach additional CG-2692's, filling out Blocks (1) and (2) and SECTION III for each additional person.

NOTICE: The information collected on this form is routinely available for public inspection. It is needed by the Coast Guard to carry out its responsibility to investigate marine casualties, to identify hazardous conditions or situations and to conduct statistical analysis. The information is used to determine whether new or revised safety initiatives are necessary for the protection of life or property in the marine environment.

DEPARTMENT OF TRANSPORTATION U. S. COAST GUARD CG-2692 (Rev. 7-86)

REPORT OF MARINE ACCIDENT, INJURY OR DEATH

RCS No. G-MMI-4017

UNIT CASE NUMBER

SECTION I. GENERAL INFORMATION

1. Name of Vessel or Facility 2. Official No. 3. Nationality 4. Call Sign 5. USCG Certificate of Inspection issued at: 6. Type (Towing, Freight, Fish, Drill, etc.) 7. Length 8. Gross Tons 9. Year Built 10. Propulsion (Steam, diesel, gas turbine...) 11. Hull Material (Steel, Wood...) 12. Draft (ft. - in.) FWD. AFT. 13. If Vessel Classed, By Whom: (ABS, LLOYDS, DNV, BV, etc.) 14. DATE (of occurrence) 15. TIME (Local) 16. Location (See Instruction No. 10A) 17. Estimated Loss or Damage TO: VESSEL \$ CARGO \$ OTHER \$ 18. Name, Address & Telephone No. of Operating CO. 19. Name of Master or Person in Charge USCG License 20. Name of Pilot USCG License State License 19a. Street Address (City, State, Zip Code) 19b. Telephone Number 20a. Street Address (City, State, Zip Code) 20b. Telephone Number

21. Casualty Elements Check as many as needed and explain in Block 44. NO. OF PERSONS ON BOARD DEATH-HOW MANY? MISSING-HOW MANY? INJURED-HOW MANY? HAZARDOUS MATERIAL RELEASED OR INVOLVED (Identify Substances and amount in Block 44.) OIL SPILL-ESTIMATE AMOUNT: CARGO CONTAINER LOST/DAMAGED COLLISION (Identify other vessel or object in Block 44.) GROUNDING FLOODING; SWAMPING WITHOUT SINKING CAPSIZING (with or without sinking) FOUNDERING OR SINKING HEAVY WEATHER DAMAGE FIRE EXPLOSION COMMERCIAL DIVING CASUALTY ICE DAMAGE DAMAGE TO AIDS TO NAVIGATION STEERING FAILURE MACHINERY OR EQUIPMENT FAILURE ELECTRICAL FAILURE STRUCTURAL FAILURE WAKE DAMAGE FIREFIGHTING OR EMERGENCY EQUIPMENT FAILED OR INADEQUATE (Describe in Block 44.) LIFESAVING EQUIPMENT FAILED OR INADEQUATE (Describe in Block 44.) BLOW OUT (Petroleum exploration/production) OTHER (Specify)

22. Conditions A. Sea or River Conditions (wave height, river stage, etc.) B. WEATHER CLEAR RAIN SNOW FOG OTHER (Specify) C. TIME DAYLIGHT TWILIGHT NIGHT D. VISIBILITY GOOD FAIR POOR E. DISTANCE (miles) (of visibility) F. AIR TEMPERATURE (F) G. WIND SPEED & DIRECTION H. CURRENT SPEED & DIRECTION

23. Navigation Information MOORED, DOCKED OR FIXED ANCHORED UNDERWAY OR DRIFTING SPEED AND COURSE 24. Last Port Where Bound 24a. Time and Date of Departure 25. FOR TOWING ONLY 25a. NUMBER OF VESSELS TOWED Empty Loaded Total 25b. TOTAL H.P. OF TOWING UNITS 25c. MAXIMUM SIZE OF TOW WITH TOW-BOAT(S) Length Width 25d. (Describe in Block 44.) PUSHING AHEAD TOWING ASTERN TOWING ALONGSIDE MORE THAN ONE TOW-BOAT ON TOW

SECTION II. BARGE INFORMATION

26. Name 26a. Official Number 26b. Type 26c. Length 26d. Gross Tons 26e. USCG Certificate of Inspection issued at: 26f. Year Built 26g. SINGLE SKIN DOUBLE SKIN 26h. Draft FWD AFT 26i. Operating Company 26j. Damage Amount BARGE \$ CARGO \$ 26k. Describe Damage to Barge

REVERSE OF CG-2692 (Rev. 7-86)

SECTION III. PERSONNEL ACCIDENT INFORMATION

27. Person Involved <input type="checkbox"/> MALE or <input type="checkbox"/> FEMALE <input type="checkbox"/> DEAD <input type="checkbox"/> INJURED <input type="checkbox"/> MISSING	27a. Name (Last, First, Middle Name) 15 27b. Address (City, State, Zip Code)	27c. Status <input type="checkbox"/> CREW <input type="checkbox"/> PASSENGER <input type="checkbox"/> OTHER (Specify)
---	--	--

28. Birth Date	29. Telephone No. ()	30. Job Position	31. (Check here if off duty) <input type="checkbox"/>
----------------	--------------------------	------------------	--

32. Employer (If different from Block 18., fill in Name, Address, Telephone No.)

33. Person's Time	YEAR(S)	MONTH(S)	34. Industry of Employer (Towing, Fishing, Shipping, Crew Supply, Drilling, etc.)
A. IN THIS INDUSTRY -	_____	_____	35. Was the Injured Person Incapacitated 72 Hours or More? <input type="checkbox"/> YES <input type="checkbox"/> NO
B. WITH THIS COMPANY -	_____	_____	
C. IN PRESENT JOB OR POSITION -	_____	_____	
D. ON PRESENT VESSEL/FACILITY -	_____	_____	
E. HOURS ON DUTY WHEN ACCIDENT OCCURRED -	_____	_____	
			36. Date of Death

37. Activity of Person at Time of Accident

38. Specific Location of Accident on Vessel/Facility

39. Type of Accident (Fall, Caught between, etc.)

40. Resulting Injury (Cut, Bruise, Fracture, Burn, etc.)

41. Part of Body Injured

42. Equipment Involved in Accident

43. Specific Object, Part of the Equipment in Block 42., or Substance (Chemical, Solvent, etc.) that directly produced the injury.

SECTION IV. DESCRIPTION OF CASUALTY

44. Describe How Accident Occurred, Damage and Recommendations for Corrective Safety Measures. (Attach Additional Sheets if necessary).

45. Witness (Name, Address, Telephone No.)

46. Witness (Name, Address, Telephone No.)

SECTION V. PERSON MAKING THIS REPORT

47. Name (PRINT) (Last, First, Middle)	47b. Address (City, State, Zip Code)	47c. Title
47a. Signature		47d. Telephone No. ()
		47e. Date

FOR COAST GUARD USE ONLY

REPORTING OFFICE:

APPARENT CAUSE

INVESTIGATOR (Name)

DATE

APPROVED BY (Name)

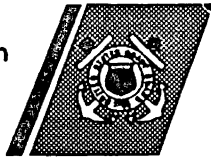
DATE

Appendix 2:

INHOUSE CASMAIN Data Dictionary

U.S. Department
of Transportation

United States
Coast Guard



Commandant
Coast Guard Headquarters
17

2100 Second Street S.W
Washington, DC 20593
Staff Symbol: (G-MIM-3
Phone: (202) 267-2824

5230.6
FOIA 92-0052
JAN 29 1992

Strategic Ventures Corporation
Attn: Ms. Pam Klein
225 Eglinton Street
Fredericton, New Brunswick E3B 2W2

Dear Ms. Klein:

This is in response to your request of January 9, 1992,
requesting a copy of the Vessel Casualty (CAS) and Personnel
Casualty (PCAS) data files.

The specification of the enclosed tape is in ASCII format, 6250
BPI and unlabelled.

File - CAS
Block Size - 10
Rec. Length - 600
No. of Records - 63,297

File - PCAS
Block Size - 10
Rec. Length - 360
No. of Records - 18,928

In response to your request, my staff has calculated the cost of
providing the information to be \$157.75. When making payment
your check, draft or money order should be made payable to the
Treasury of the United States and sent to the following address:

Commandant (G-MIM-3)
U.S. Coast Guard
2100 2nd St., S.W.
Washington, DC 20593
Attn: Ms. Denise Williams

Please reference this letter when forwarding payment.

Should you have any questions concerning the data, please contact
Mr. Pettin at (202) 267-1425. Any questions concerning the tape
copy should be referred to Ms. D. Williams at (202) 267-2824.

Sincerely

A handwritten signature in cursive script, appearing to read "M. E. Schremp".

M. E. SCHREMP
Chief, Systems Support Branch
U.S. Coast Guard
By direction of the Commandant

Encl: (1) Computer Tape
(2) Tape Dump
(3) Documentation

INHOUSE
CASMAIN Data Dictionary
November 1991

Office of Marine Safety, Security and Environmental Protection

MARINE INVESTIGATION DIVISION
Marine Safety Evaluation Branch
G-MMI-3

19

READ THIS BEFORE ATTEMPTING TO USE THIS DATA DICTIONARY !!!

The Office of Marine Safety, Security and Environmental Protection continually receives requests for marine casualty information. We respond with reports (CASMAIN database queries) that contain a wide range of marine casualty data.

The Coast Guard uses the CASMAIN file database within the confines of Coast Guard (G-MMI-3) vessel definitions. Users unfamiliar with these definitions run the risk of compromising their research results. Thus, their results may not compare favorably with that of the Coast Guard's.

This data dictionary is primarily for 'Inhouse Use' but may be used by individuals who require an understanding of coded values appearing in reports. When analyzing our data you may note various vessels with the same case number. This implies the vessels are associated with the same casualty.

Warning!!! DO NOT use these codes. There are certain codes that were designed specifically for G-MMI analytical use only. Less than a complete understanding of these codes will yield erroneous results. These codes are as follows:

<u>File "CAS"</u>		<u>File "PCAS"</u>
CASEYR	IND	CASENAME
TDAM	INSPECT	
ODAM	COIDATE	
OTHERD	IMSO	
OTHERI	VISIBLTY	

Coast Guard casualty and pollution databases are not currently linked together. Pollution information may be requested by contacting CDR Lentsch (G-MEP-2) at (202) 267-0440. Questions concerning this data dictionary may be directed to Mr. T. Pettin at (202) 267-1424.

The CASMAIN database may be purchased. Inquiries should be addressed to G-MIM-3 as follows:

Commandant (G-MIM-3)
U. S. Coast Guard
2100 2nd St., S.W.
Washington, DC 20503-1000

ATTENTION: Ms. D. Williams
(202) 267-2396

20
TABLE OF CONTENTS

VESSEL information (post 1980 data)	Page
Complete list of available vessel codes	4-5
Day, Night, etc.	7
Weather	7
Body of Water	8-11
Wind Direction	12
Nature of Casualty	12-14
Casualty Cause	15-17
Sea, River Conditions	17
Type of Investigation Undertaken	18
Configuration of Towboats	18
Nationality of Vessel	18-22
Service	23
Vessel Use	23-24
Hull Material	24
Propulsion	24
Hull Design	25
Bureau Classifying Vessel	25
Licensing Status of Pilot Aboard Vessel	25
Manning Status	25
Vessel Afloat, Moored, Docked, etc.	26
Seaworthiness of Vessel Following Casualty	26

PERSONNEL information (post 1980 data)	Page
Complete list of available personnel codes	6
Manning Status of Personnel On Vessel	25
Nature of Accident: Fall, etc.	27-28
Nature of Injury: Burn, etc.	28-29
Part of Body Injured	29
Result of Injury: Death, Injury, Missing, etc.	29
Cause of Injury	30
Location of Person on Vessel	31
Activity of Person on Vessel	31
Marine Safety Offices	32

Regarding fields in the VESSEL file "CAS"

Field	Length	Field Definition	
CASE	C10	CASE NUMBER OF VESSEL CASUALTY	
CASEYR	* C2	CALENDAR YEAR CASUALTY OCCURRED	
VIN	C8	VESSEL DOCUMENTATION NUMBER	
VIN1-2	U2	FIRST 2 CHARACTERS OF VIN	
NUMVSDAM	U3	NUMBER OF VSLS DAMAGED IN CASE	
CAUSE7	C7	SECOND CAUSE NATURE 3, DRUG FACTOR	
CASDATE	U6	DATE VESSEL CASUALTY OCCURRED	
CY	U2	CALENDAR YEAR OF CASUALTY	
MONTH	U2	MONTH OF CASUALTY	
PERIODAY	C1	DAY, NIGHT, ETC.	
WEATHER	C2	WEATHER	
WATER	C6	BODY WATER CASUALTY OCCURR/IN	+ district
WATER1-2	U2	DISTRICT BODY WATER OF CASUALTY	
WATER3	C1	THIRD CHARACTER OF WATER	
WATER4	C1	FOURTH CHARACTER OF WATER	
WATER5	C1	FIFTH CHARACTER OF WATER	
LATITUDE	C10	LATITUDE	
LONGITUD	C10	LONGITUDE	
REPTYP	C6	ROUTINE LETTER XMIT, FORM 2692	
TDAM	* U10	TOTAL DAM ENTIRE CASUALTY CASE	
OFFICE	C3	MSO OFFICE INVESTIGATING CASE	
MILEPOST	C5	RIVER MILEPOST	
CAUSE6	C7	FIRST CAUSE OF NATURE3	
ENTERDBY	C10	USCG EMPLOYEE RECORDING CASE	
WINDDIR	C3	WIND DIRECTION	
WIND	C3	WIND SPEED	
VISIBLTY	C4 5	VISIBILITY	
INVSLS	U3	NUMBER OF VSLS INVOLVED IN CASE	
NATURE1	C6	FIRST NATURE OF CASUALTY	
NATURE1A	C3	FIRST THREE CHARACTERS OF NATURE1	
NATURE2	C6	SECOND NATURE OF CASUALTY	
NATURE2A	C3	FIRST THREE CHARACTERS OF NATURE2	
NATURE3	C6	THIRD NATURE OF CASUALTY	
NATURE3A	C3	FIRST THREE CHARACTERS OF NATURE3	
CAUSE1	C7	FIRST CAUSE OF NATURE1	
CAUSE1A	C1	FIRST CHARACTER OF CAUSE1	
CAUSE2	C7	SECOND CAUSE OF NATURE1	
CAUSE2A	C1	FIRST CHARACTER OF CAUSE2	

* DO NOT USE THESE FIELDS!!!

con't.

Regarding fields in the VESSEL file "CAS"

Field	Length	Field Definition
CAUSE3	C7	THIRD CAUSE OF NATURE1
CAUSE3A	C1	FIRST CHARACTER OF CAUSE3
CAUSE4	C7	FIRST CAUSE OF NATURE2
CAUSE5	C7	SECOND CAUSE NATURE2
SEACON	C4	SEA CONDITIONS
CONFIG	C3	TOW CONFIGURATION
VSLNAME	C10	FIRST TEN CHARACTERS OF VSLNAME
FLAG	C2	FLAG OF VESSEL
YRBUILT	U2	YEAR VESSEL BUILT
SERVICE	C4	TYPE OF VESSEL: MODU, BARGE
ABC	C1	HOW SEAWORTHINESS WAS AFFECTED
USE	C4	HOW VESSEL WAS BEING USED
LENGTH	U4	LENGTH
GTON	U6	GROSS TONNAGE
HULL	C2	HULL MATERIAL
PROP	C2	TYPE OF VESSEL PROPULSION
HORSEPWR	U6	HORSEPOWER
DESIGN	C4	HULL DESIGN
VDAM	U10	VESSEL DAMAGE
CDAM	U10	CARGO DAMAGE
ODAM	* U10	OTHER CASE DAM BESIDES V/C DAM
CREWD	Z3	CREW DEATH
PASSD	Z3	PASSENGER DEATH
OTHERD	* Z3	OTHER DEATHS OCCURR/IN CASE
CREWI	Z3	CREW INJURY
PASSI	Z3	PASSENGER INJURY
OTHERI	Z3	OTHER INJURIES IN CASUALTY
VSLSTATE	C2	AFLOAT, AT ANCHOR, ETC.
PIC	C4	PERSON IN CHARGE OF VSL MOVEMT
SOCIETY	C3	VESSEL SOCIETY CLASSING VESSELS
OPERCO	C10	NAME OF COMPANY OPERATING VSL
PILOT	C4	VERIFICATION OF LICENSED PILOT
VSLNAMEA	C10	SECOND TEN CHARACTERS OFVSLNAME
VSLNAMEB	C10	THIRD TEN CHARACTERS OF VSLNAME
VSLNAMEC	C10	FOURTH TEN CHARACTERS OFVSLNAME
IND	* C1	INSPECTION INDICATOR: Y, N, U
INSPECT	* C10	VESSEL INSPECTION VERIFICATION
COIDATE	* C6	INSPECT CERT ISSUE DATE: YYMMDD
IMSO	* C3	MSO ISSUING CERTIF OF INSPECT

* DO NOT USE THESE FIELDS!!!

The following Vsl Inspection Fields haven't been used since 1986:

IND : Y=Yes vsl inspected N=No U=Unknown

IMSO : MSO issuing the Certificate of Inspection,
(see field codes for 'OFFICE' codes)

COIDATE : Date Certificate of Inspection was issued (YYMMDD)

INSPECT : Has vessel been inspected? Y=Yes N=No U=Unknown

Regarding fields in the PERSONNEL file "PCAS"

Field	Length	Field Definition
CASE1	C10	CASE NUMBER OF INJURY/DEATH
CASENAME	* C20	COMBINATION OF CASE AND LNAME
LNAME	C10	FIRST TEN LETTERS PER LAST NAME
FNAME	C10	FIRST TEN LETTERS PER. 1ST NAME
BIRTH	U6	BIRTHDATE
STATUS	C2	POSITION PERSON HELD ON VESSEL
NATACCID	C10	NATURE PERSONNEL ACCIDENT: FALL
NATINJ	C10	NATURE OF INJURY/DEATH: BURN
BODPART	C10	PART OF BODY INJURED
RESULT	C4	RESULT INJ/DEATH: MISSING-NVC
CAUSE1	C5	PRIMARY CAUSE OF INJURY/DEATH
CAUSE2	C5	SECONDARY CAUSE OF INJURY/DEATH
OFFICE	C3	MSO OFFICE INVESTIGATING
PLOC	C4	LOCATION OF PERSON ON VESSEL
ACTIVITY	C2	TYPE OF ACTVTY PERSON UNDERTKNG
WATER	C6	BODY WATER CASUALTY OCCURR/IN
WATER1-2	U2	DISTRICT BODY WATER OF CASUALTY
WATER3	C1	THIRD CHARACTER OF WATER
WATER4	C1	FOURTH CHARACTER OF WATER
WATER5	C1	FIFTH CHARACTER OF WATER
YRBUILT	U4	YEAR VESSEL BUILT
CASDATE	U6	DATE PERSONNL CASUALTY OCCURRED
CY	U2	CALENDAR YEAR OF CASE
INDTIME	* C7	INDUSTRY TIME OF PERSONNEL
COTIME	* C7	COMPANY TIME OF PERSONNEL
VSLNAME	C10	FIRST TEN CHARACTERS OF VSLNAME
OPERCO	C10	NAME OF COMPANY OPERATING VSL
VIN	C10	VESSEL DOCUMENTATION NUMBER
VIN1-2	U2	FIRST TWO CHARACTERS OF VIN
FLAG	C2	FLAG OF VESSEL
SERVICE	C4	SERVICE OF VESSEL: MODU, FISH
USE	C4	HOW VSL WAS USED: FERRY, CHEM
DESIGN	C4	DESIGN OF VSL: BARGE, CONV
LENGTH	U4	LENGTH OF VESSEL
GTON	U6	GROSS TONS OF VESSEL
DUTYTIME	C3	NUMBER HOURS PERSON WAS ON DUTY
LSTNAME	* C10	SECOND TEN LETTERS VICTIMS NAME
FSTNAME	* C10	SECOND TEN LETTERS PER.1ST NAME
VSLNAMEA	C10	SECOND TEN CHARACTERS VSLNAME
VSLNAMEB	C10	THIRD TEN CHARACTERS OF VSLNAME
VSLNAMEC	C10	FOURTH TEN CHARACTER OF VSLNAME

* DO NOT USE THESE FIELDS!!!

24
BEGIN "CAS" Retrieval Codes

PERIOD OF THE DAY	CODE	DATA LIMIT
"Day"	"D"	"PERIODAY"
"Night"	"N"	"
"Twilight"	"T"	"
"Ongoing"	"O"	"
"Unknown"	"U"	"

WEATHER CONDITION

WEATHER CONDITION	CODE	"WEATHER"
"Clear"	"CL"	"
"Snow"	"SN"	"
"Partly Cloudy"	"PC"	"
"Fog"	"FG"	"
"Overcast"	"OC"	"
"Rain"	"RN"	"
"Drizzle"	"DR"	"
"Hazy"	"HZ"	"
"Unknown"	"UN"	"
"Not applicable"	"NA"	"
"Not otherwise categorized"	"NC"	"

WATER1-2
99-Genl,
District#

WATER3
A-Atlantic Ocean
P-Pacific Ocean
G-Gulf of Mexico
F-Specfc Foreign
L-Great Lakes
X-Other

WATER4
O-Ocean, GT 3 mi
C-Coastal LE 3 mi
I-Internal
S-Sea
X-General/Other

WATER5
B-Bays/Sounds
H-Harbors
L-Lakes
R-Rivers
X-General/Oth
N-North
S-South

BODY OF WATER	CODE	DATA LIMIT
"North Atlantic Ocean"	"99AON"	"WATER"
"South Atlantic Ocean"	"99AOS"	"
"North Pacific Ocean"	"99PON"	"
"South Pacific Ocean"	"99POS"	"
"Indian Ocean"	"99XOI"	"
"Arctic Ocean"	"99XOA"	"
"Mediterranean Sea"	"99XSM"	"
"Gulf of Mexico"	"99GOX"	"
"Caribbean Sea"	"99XSC"	"
"Panama Canal"	"99FRP"	"
"Suez Canal"	"99FRS"	"
"Unknown"	"99UNK"	"
"Not specified, Foreign"	"99NEC"	"
"Atlantic Ocean (1st dist)"	"01AOXN"	"WATER"
"Cstl wtrs offshr (1st dist)"	"01ACXN"	"
"Internal waters NOC (1st dist)"	"01AIXN"	"
"Block Island Sound"	"01AOBB"	"
"Boston Harbor"	"01AIHB"	"
"Buzzards Bay"	"01AIBB"	"
"Cape Cod Bay"	"01AOBC"	"
"Cape Cod Canal"	"01AIRC"	"
"Casco Bay"	"01AIBC"	"
"Massachusetts Bay"	"01AOBM"	"
"Muscongus Bay"	"01AIBM"	"
"Nantucket Sound"	"01ACBN"	"
"Narragansett Bay"	"01AIBN"	"
"Penobscot Bay"	"01AIBP"	"
"Piscataqua river"	"01ARIP"	"
"Portland Harbor/River"	"01AIHP"	"
"Vineyard Sound"	"01AIBV"	"
"Internal Waters NOC (2nd dist)"	"02XIXN"	"WATER"
"Upper Mississippi River"	"02XIRU"	"
"Lower Miss River, starts at 507"	"02XIRL"	"
"Allegheny River"	"02XIRA"	"
"Arkansas River"	"02XIRR"	"
"Clinch River"	"02XIRN"	"
"Cumberland River"	"02XIRC"	"
"Green River"	"02XIRG"	"
"Illinois River"	"02XIRI"	"
"Kanawha River"	"02XIRK"	"
"Missouri River"	"02XIRM"	"
"Monongahela River"	"02XIRX"	"
"Ohio River"	"02XIRO"	"
"St. Croix River"	"02XIRS"	"
"Tennessee River"	"02XIRT"	con't

BODY OF WATER	CODE	DATA LIMIT
"Atlantic Ocean (3rd dist)"	"03AOXN"	"WATER"
"Cstl wtrs offshr (3rd dist)"	"03ACXN"	"
"Internal wtrs NOC (3rd dist)"	"03AIXN"	"
"Arthur Kill"	"03AIRA"	"
"Connecticut River"	"03AIRC"	"
"Delaware Bay"	"03AIBD"	"
"Delaware River"	"03AIRD"	"
"East River"	"03AIRE"	"
"Hudson River (battery-41 00N)"	"03AIRH"	"
"Hudson River (N of 41 00)"	"03AIRN"	"
"Inter cstl waterway"	"03AIXI"	"
"Kill Van Kull"	"03AIRK"	"
"Long Island Sound"	"03AIBI"	"
"New London Harbor"	"03AIHN"	"
"New York Harbor, lower bay"	"03AIBL"	"
"New York Harbor, upper bay"	"03AIBU"	"
"Newark Bay-Hcknsck R & Passaic R"	"03AIBN" ?	" ←
"Atlantic Ocean (5th dist)"	"05AOXN"	"WATER"
"Cstl Water Offshr (5th dist)"	"05ACXN"	"
"Internal Waters NOC (5th dist)"	"05AIXN"	"
"Baltimore Harbor"	"05AIHB"	"
"Cape Fear River"	"05AIRF"	"
"Chesapeake & Delaware Canal"	"05AIRC"	"
"Chesapeake Bay added 8/30/91"	"05AIBC"	"
"Elizabeth River"	"05AIRE"	"
"Inter Cstl Waterway"	"05AIXI"	"
"James Rv(Inc Norfolk/Nwprt Nws)"	"05AIRJ"	" ←
"Potomac River"	"05AIRP"	"
"Atlantic Ocean (7th dist)"	"07AOXN"	"WATER"
"Cstl Wtrs Arl Offsh (7th dist)"	"07ACXN"	" ←
"Internal waters NOC (7th dist)"	"07AIXN"	"
"Charleston Harbor"	"07AIHC"	"
"Inter Cstl Wtrwy"	"07AIXI"	"
"Krause Lagoon (St. Croix)"	"07AILK"	"
"Limetree Bay (St. Croix)"	"07AIBL"	"
"Ponce Hbr (Bahia de Ponce, SJ)"	"07AIHP"	" ←
"Port Canaveral"	"07AIHK"	"
"Port Everglades"	"07AIHE"	"
"Port of Jacksonville"	"07AIHJ"	"
"Port of Miami"	"07AIHM"	"
"San Juan Hbr(Bahia de Sn Jn)"	"07AIHS"	"
"Savannah River"	"07AIRS"	"
"St. Johns River"	"07AIRJ"	"
"St. Marys River"	"07AIRM"	"
"St. Thomas River"	"07AIHT"	"
"Tampa Bay"	"07AIBT"	"
"Gulf of Mexico (7th Dist)"	"07GOXF"	"
"Cstl Wtrs Offsh (7th Dist)"	"07GCXF"	"
"Inter Cstl Wtrwy"	"07GIXI"	con't

BODY OF WATER	CODE	DATA LIMIT
"Gulf of Mexico (8th dist)"	"08GOXN"	"WATER"
"Cstl Wtrs Offshr (8th dist)"	"08GCXN"	"
"Internal Wtrs NOC (8th dist)"	"08GIXN"	"
"Alabama River"	"08GIRA"	"
"Atchafalaya River"	"08GIRQ"	"
"Barataria Bay"	"08GIBB"	"
"Bayou LaFourche"	"08GIBL"	"
"Berwick Bay"	"08GIBE"	"
"Corpus Christi (Ship Chnl & Hbr)"	"08GIRR"	" ←
"Galveston Bay"	"08GIBG"	"
"Gulf Outlet-Mississippi River)"	"08GIRG"	"
"Houma Channel"	"08GIRO"	"
"Houston Ship Channel"	"08GIRC"	"
"Inter Cstl Wtrwy"	"08GIXI"	"
"Lake Ponchartrain"	"08GILP"	"
"Lower Miss. River (M507)"	"08GIRL"	"
"Mississippi Sound"	"08GCBM"	"
"Mobile Bay"	"08GIBM"	"
"Mobile River"	"08GIRM"	"
"Pass A Loutre"	"08GIRP"	"
"Pensacola Bay"	"08GIBP"	"
"Port Allen route"	"08GIRZ"	"
"Sabine/Neches river"	"08GIRB"	"
"South Pass-AHP to GOM"	"08GIRH"	" ←
"Southwest Pass-AHP to GOM"	"08GIRS"	" ←
"Timbalier Bay"	"08GIBT"	"
"Tombigbee River"	"08GIRT"	"
"Vermillion Bay"	"08GIBV"	"
"Internal waters NOC (9th dist)"	"09XIXN"	"WATER"
"Chicago Ship Canal"	"09XIRH"	"
"Detroit River"	"09XIRD"	"
"Illinois River"	"09XIRI"	"
"Lake Erie"	"09LXLE"	"
"Lake Huron"	"09LXLH"	"
"Lake Michigan"	"09LXLM"	"
"Lake Ontario"	"09LXLO"	"
"Lake St. Clair"	"09LILS"	"
"Lake Superior"	"09LXLS"	"
"Maumee River" added 3-5-87	"09XIRMU"	"
"Rogue River" added 3-5-87	"09XIRRU"	"
"Minnesota River"	"09X1RMN"	"
"Saginaw River"	"09XIRS"	"
"St. Clair River"	"09XIRC"	"
"St. Lawrence River"	"09XIRL"	"
"St. Mary's River"	"09XIRM"	"
"Welland Canal"	"09XIRW"	con't

BODY OF WATER

28

CODE

DATA LIMIT

"Pacific Ocean (11th dist)"	"11POXN"	"WATER"
"Cstl Wtrs Offshr (11th dist)"	"11PCXN"	"
"Internal waters NOC (11th dist)"	"11PIXN"	"
"Port of LA/LB"	"11PIHL"	"
"San Diego Harbor"	"11PIHS"	"
"San Pedro Bay"	"11PIHP"	"
"Pacific Ocean (12th dist)"	"12POXN"	"WATER"
"Cstl Wtrs Offshr (12 dist)"	"12PCXN"	"
"Internal waters NOC (12 dist)"	"12PIXN"	"
"San Francisco Bay"	"12PIBS"	"
"Sacramento river"	"12PIRS"	"
"Pacific Ocean (14th dist)"	"14POXN"	"WATER"
"Cstl Wtrs Offshr (14th dist)"	"14PCXN"	"
"Internal waters NOC (14th dist)"	"14PIXN"	"
"Honolulu Harbor"	"14PIHH"	"
"Pearl Harbor"	"14PIHP"	"
"Pacific Ocean (13th dist)"	"13POXN"	"WATER"
"Cstl Wtrs Offshr (13th dist)"	"13PCXN"	"
"Internal waters NOC (13th dist)"	"13PIXN"	"
"Columbia River"	"13PIRC"	"
"Gray's Harbor"	"13PIHG"	"
"Lake Washington SC/Lake Union"	"13LXLU"	"
"Puget Sound"	"13PIBP"	"
"Snake River"	"13PIRS"	"
"Strait of Juan De Fuca"	"13PCXJ"	"
"Willamette river"	"13PIRW"	"
"Pacific Ocean (17th dist)"	"17POXN"	"WATER"
"Cstl Wtrs Offshr (17th dist)"	"17PCXN"	"
"Internal waters NOC (17th dist)"	"17PIXN"	"
"Dixon Entrance"	"17POXD"	"
"Prince William Sound"	"17PIBW"	"
"Bering Sea"	"17XSXB"	"
"Bristol Bay"	"17XOBB"	"
"Norton Sound"	"17XOBN"	"
"Bering Straits"	"17XOXB"	"
"Chukchi Sea"	"17XOXC"	"
"Beaufort Sea"	"17XOXE"	"
"Chatham Strait"	"17PIRC"	"
"Frederick Sound"	"17PIRF"	"
"Stephens Passage"	"17PIRS"	"
"Sitka Sound"	"17POBS"	"
"Lynn Canal"	"17PIRL"	"
"Clarence Strait"	"17PIRA"	"

WIND DIRECTION	CODE	DATA LIMIT
"North (000.0 Deg)"	"N"	"WIND"
"North northeast (022.5 Deg)"	"NNE"	"
"Northeast (045.0 Deg)"	"NE"	"
"East northeast (067.5 Deg)"	"ENE"	"
"East (090.5 Deg)"	"E"	"
"East southeast (112.5 Deg)"	"ESE"	"
"Southeast (135.0 Deg)"	"SE"	"
"South southeast (157.5 Deg)"	"SSE"	"
"South (180.0 Deg)"	"S"	"
"South southwest (202.5 Deg)"	"SSW"	"
"Southwest (225.0 Deg)"	"SW"	"
"West southwest (247.5 Deg)"	"WSW"	"
"West (270.0 Deg)"	"W"	"
"West northwest (292.5 Deg)"	"WNW"	"
"Northwest (315.0 Deg)"	"NW"	"
"North northwest (337.5 Deg)"	"NNW"	"
"Variable"	"VAR"	"

NATURE OF CASUALTY	CODE	DATA LIMIT
"Allision"	"ALLIS"	"NATURE1+"
"Barge Breakaway"	"BRGBWY"	"
"Capsizing"	"CAPSIZ"	"
"Collision, Meeting"	"COLMTG"	"
"Collision, Crossing"	"COLCRS"	"
"Collision, Overtaking"	"COLOTK"	"
"Collision, Special circumstance"	"COLSPC"	"
"Collision, w/ice"	"COLICE"	"
"Collision, w/aid to navigation"	"COLATN"	"
"Collision, Submerged object"	"COLSUO"	"
"Collision, Floating object"	"COLFLO"	"
"Collision, Bridge"	"COLBDG"	"
"Collision, Pier/Dock"	"COLDOC"	"
"Collision, Offshore drlng unit"	"COLMOD"	"
"Collision, Fixed object NOC"	"COLFNC"	"
"Collision, NEC"	"COLNEC"	"
"Collision, Unknown"	"COLUNK"	"
"Collision, w/dike, lock or dam"	"COLLDM"	"
"Disappearance, w/trace"	"GONETR"	"
"Disappearance, wo/trace"	"GONENT"	"
"Explosion, Cargo - no fire"	"EXPCGN"	con't

NATURE OF CASUALTY	CODE	DATA LIMIT
"Explosion, Mach space - no fire"	"EXPMSN"	"NATURE1+"
"Explosion, Pressure vel-no fire"	"EXPPVN"	"
"Explosion, Pumproom-no fire"	"EXPPRN"	"
"Explosion, Boiler-no fire"	"EXPBNF"	"
"Explosion, Fuel-no fire"	"EXPFUN"	"
"Explosion, Cargo-fire"	"EXPCGF"	"
"Explosion, Mach space-fire"	"EXPMSF"	"
"Explosion, Pressure val-fire"	"EXPPVF"	"
"Explosion, Pumproom-fire"	"EXPPRF"	"
"Explosion, Boiler-fire"	"EXPBOF"	"
"Explosion, Fuel-fire"	"EXPFUF"	"
"Explosion, NEC"	"EXPNEC"	"
"Explosion, Unknown"	"EXPUNK"	"
"Fire, Vessel furnishing"	"FIRFUR"	"
"Fire, Vessel cargo, freight"	"FIRCFT"	"
"Fire, Machinery space"	"FIRMCS"	"
"Fire, Pumproom"	"FIRPMR"	"
"Fire, Vessel structure"	"FIRSTR"	"
"Fire, Vessel fuel"	"FIRVFU"	"
"Fire, Electrical"	"FIRELC"	"
"Fire, Vessel cargo, fuel"	"FIRCFU"	"
"Fire, Vessel cargo, HAZMAT"	"FIRCHZ"	"
"Fire, NEC"	"FIRNEC"	"
"Grounding, accidental"	"GRNDGA"	"
"Grounding, Intl w/damage-hazard"	"GRNDGI"	"
"Matl Failure, main eng/motor"	"MATMEN"	"
"Matl Failure, boiler"	"MATBOL"	"
"Matl Failure, main steam sys"	"MATMSS"	"
"Matl Failure, aux steam sys"	"MATASS"	"
"Matl Failure, feed and condens sys"	"MATFCS"	"
"Matl Failure, cooling water sys"	"MATCWS"	"
"Matl Failure, fuel oil supply"	"MATFOS"	"
"Matl Failure, lube oil supply"	"MATLOS"	"
"Matl Failure, main generator"	"MATMGN"	"
"Matl Failure, aux generator"	"MATAGN"	"
"Matl Failure, elec control systems"	"MATECS"	"
"Matl Failure, elec dis sys"	"MATEDS"	"
"Matl Failure, hyd contl sys"	"MATHCS"	"
"Matl Failure, phuem contl sys"	"MATPCS"	"
"Foundering, sinking"	"FNDRNG"	"
"Flooding, w/out sinking"	"FLDING"	con't ✓✓

NATURE OF CASUALTY	CODE	DATA LIMIT
"Matl Failure-Bilge Sys"	"MATBLG"	"NATURE1+"
"Matl Failure-Reduction Gear"	"MATRED"	"
"Matl Failure-Shaft System"	"MATSFT"	"
"Matl Failure-Propeller"	"MATPRO"	"
"Matl Failure-Cargo Hndlng-Tnkr"	"MATCGT"	"
"Matl Failure-Cargo Hndlng-Frt"	"MATCGF"	"
"Matl Failure-Salt Water Sys"	"MATSWS"	"
"Matl Failure-Venting System"	"MATVNT"	"
"Matl Failure-Inert Gas System"	"MATIGS"	"
"Matl Failure-Crude Oil Wshng Sys"	"MATCOW"	"
"Matl Failure-Navigation Eqpt"	"MATNAV"	" ✓
"Matl Failure-Ground Tackle"	"MATGTK"	"
"Matl Failure-Lifesaving Eqpt"	"MATLSG"	"
"Matl Failure-Firefighting Eqpt"	"MATFFG"	"
"Matl Failure-Pers Protect Eqpt"	"MATPPE"	"
"Matl Failure-Hull, Structural"	"MATHST"	"
"Matl Failure-Hull, Deterioration"	"MATHDT"	"
"Matl Failure-NEC"	"MATNEC"	"
"Steering Sys Fail, Contl Sys"	"SSFCSS"	"
"Steering Sys Fail, Rdr and Shaft"	"SSFRAS"	"
"Steering sys fail, aux pwr sply"	"SSFAPS"	"
"Steering sys fail, NEC"	"SSFNEC"	"
"Cargo, loss or damage"	"CARGLD"	"
"Disabled"	"DISABL"	"
"Wake Damage"	"WAK"	"
"Swamping"	"SWAMP"	"
"Weather Damage"	"WTHRDM"	"
"Well Blowout"	"WELBLO"	"

CAUSE OF CASUALTY	CODE	DATA LIMIT
"Unknown"	"UNKNOWN"	"CAUSE1+"
"Unknown"	"PUNKNOW"	
"P Bypassed avail safety devices"	"PBPSASD"	"
"P Inattention to duty"	"PINATT"	"
"P Intoxication (Alcohol/Drugs)"	"PDRUNK"	"
"P Calculated risk"	"PCALRSK"	"
"P Carelessness"	"PCRLSNS"	"
"P Error in Judgment"	"PERRJDG"	"
"P Lack of Knowledge"	"PLCKKNO"	"
"P Lack of Training"	"PLCKTNG"	"
"P Lack of Experience"	"PLCKEXP"	"
"P Operator Error"	"POPERER"	"
"P Fatigue"	"PTIRED"	"
"P Smoking"	"PSMOKD"	"
"P Open Flame"	"POPNFL"	"
"P Stress"	"PSTRESS"	"
"P Physical Impairment"	"PPHYSIM"	"
"P Psychological Impairment"	"PPSYCHO"	"
"P Failed comple w/rule,reg,pro"	"PFALRUL"	"
"P Inadequate Supervision"	"PINADSP"	"
"P Improper Casualty Control pro"	"PIMPCCP"	"
"P Improper safety precautions"	"PIMPSFP"	"
"P Failed to acct for crnt/wx"	"PFALACW"	"
"P Failed to acct for tide/riv sg"	"PFALATR"	"
"P Failed to ascertain position"	"PFALPOS"	"
"P Failed to use avail nac equip"	"PFALANE"	"
"P Failed to use charts and pubs"	"PFALCAP"	"
"P Failed to use radiotelephone"	"PFALRTE"	"
"P Relied on floating ATON"	"PRELFAN"	"
"P Failed to yield right of way"	"PFALTY"	"
"P Failed to est passing agreement"	"PFALEPA"	"
"P Failed to keep to right of chnl"	"PFALKRC"	"
"P Failed to proceed at safe speed"	"PFALSPD"	"
"P Failed to stop"	"PFALSTP"	"
"P Failed to keep proper lookout"	"PFALKPL"	"
"P Improper/faulty lights/shapes"	"PIMPFLT"	"
"P Improper/missing whistle signals"	"PIMPMWS"	"
"P Improper maintenance"	"PIMPMNT"	"
"P Used defective equipment"	"PDEFEQT"	"
"P Design creteria exceeded"	"PDSGCEX"	"
"P Service conditions exceeded"	"PSVCCEX"	con't

CAUSE OF CASUALTY

CODE DATA LIMIT

CAUSE OF CASUALTY	CODE	DATA LIMIT
"P Preventive maint not done"	"PPVTMNT"	"CAUSE1+"
"P Improper loading"	"PIMPLOD"	"
"P Improper cargo stowage"	"PIMPCGS"	"
"P Improper securing/rigging"	"PIMPSCR"	"
"P Improper mooring/towing"	"PIMPOT"	"
"V Inadequate lighting"	"VINADLT"	"
"V Inadequate improper NAV equip 8/14/91"	"VINADNE"	"
"V Inadequate stability"	"VINADST"	"
"V Inadequate lifesaving equip"	"VINADLE"	"
"V Inadequate firefighting equip"	"VINADFF"	"
"V Inadequate/missing guarding"	"VINADMG"	"
"V Inadequate controls"	"VINADCT"	"
"V Inadequate displays"	"VINADDS"	"
"V Inadequate horsepower"	"VINHRSP"	"
"V Faulty NAV (Running) lights"	"VFLTNLT"	"
"V Insufficient fuel"	"VINFUEL"	"
"V Failed materials, electrical"	"VFLDMEL"	"
"V Failed materials, mechanical"	"VFLDMME"	"
"V Failed materials, structural"	"VFLDMST"	"
"V Failed materials, other"	"VFLDMOT"	"
"V Failed fastenings"	"VFLDFST"	"
"V Improper riveting"	"VIMPRIV"	"
"V Improper welding"	"VIMPWEL"	"
"V Brittle fracture"	"VBRIFFRA"	"
"V Fatigue fracture"	"VFATFRA"	"
"V Stress fracture"	"VSTRFRA"	"
"V Auxiliary power failure"	"VAUXFAL"	"
"V Temperature stress"	"VTMPSTR"	"
"V Corrosion"	"VCORROS"	"
"V Vibration"	"VVIBRAT"	"
"V Normal wear"	"VNORMLW"	"
"V Static electricity"	"VSTATEL"	"
"V Inadequate lubrication"	"VINALUB"	"
"V Steering failure"	"VSTRFAL"	"
"V Propulsion failure"	"VPROFAL"	"
"V Fouled propeller"	"VFOUPRO"	"
"V Cargo shift"	"VCGOSHF"	"
"V Dragging anchor"	"VDRGANC"	"
"E Adverse weather"	"EADVWTH"	"
"E Adverse current/sea condition"	"EADVCRT"	con't

CAUSE OF CASUALTY

34

CODE

DATA LIMIT

"E Debris"		"EDEBRIS"	"CAUSE1+"
"E Suction bank/bottom/vsl"		"ESUCBBV"	"
"E Ice"		"EICE"	"
"E Lightning"		"ELITNIN"	"
"E Shoaling"		"ESHOAL"	"
"E Submerged object"		"ESUBOBJ"	"
"E Channel not maintained"		"ECHNMNT"	"
"E Uncontrollable channel hazard"		"EUNCCHZ"	"
"E Unmarked channel hazard"		"EUNMCHZ"	"
"E Hzrdus bridge/dock/pier lctn"		"EHZBDPL"	"
"E Hzrdus bridge/dock/pier config"		"EHZBDPC"	"
"E Inadequate bridge/dock/pier ID"		"EINBDPI"	"
"E Inadequate equip available"		"EINADEA"	"
"E Inadequate reg, rule, proc, plcy"		"EINADGP"	"
"M Inadequate equip available"	8/14/91	"MINADEA"	"
"M Inadequate WX info avail"	8/14/91	"MINADWI"	"
"M Inadequate manning"	8/14/91	"MINADMN"	"
"M Faulty design"		"MFLTDSG"	"
"M Sys maint functions improper"		"MSYSMNT"	"
"M Improper AID location"		"MIMADLO"	"
"M Inadequate AID display/type"		"MINAIDD"	"
"M Inadequate AID maintenance"		"MINAIDM"	"
"M Inadequate statutory/reg rqmnts"		"MINREGS"	"
"M Inadequate owner/op safety prog"		"MINSFT"	"
"Not elsewhere classified"		"NEC"	"
"Result of previous nature"		"PREVNAT"	"
"Vandalism"		"NVANDAL"	"

SEA CONDITIONS

CODE

DATA LIMIT

"Calm	0-1'	"	"CALM"	"SEACON"
"Choppy	1-3'	"	"CHOP"	"
"Moderate	4-12'	"	"MOD"	"
"Heavy	12-20'	"	"HVY"	"
"Very Heavy ...	20-40'	"	"VHVY"	"
"Precipitous ..	GT-40'	"	"PREC"	"
"Strong current"			"SCRT"	"
"Freshet/Flooding"			"FRFL"	"
"Flood tide"			"FLTD"	"
"Ebb tide"			"EBBT"	"
"Ice"			"ICE"	"
"Not specified"			"NA"	"
"Unknown"			"UNK"	"

TYPE OF CASE	CODE	DATA LIMIT
"Closed to file"	"CLOSED"	"REPTYP"
"Form CG-2692"	"CG2692"	"
"Ltr of transmittal, routine"	"LTRROT"	"
"Ltr of transmittal, formal"	"LTRFOR"	"
"Narrative, routine"	"NARROT"	"
"Narrative, formal"	"NARFOR"	"
"Marine board"	"MARBRD"	"
"National Trans Safety Board"	"NTSB"	"

TOWBOAT CONFIGURATION	CODE	DATA LIMIT
"Pushing ahead"	"PAH"	"CONFIG"
"Towing astern"	"TAS"	"
"Towing alongside"	"TAL"	"
"More than one towboat"	"MTO"	"
"Not applicable"	"NA"	"

VESSEL REGISTRY	CODE	DATA LIMIT
"United States"	"US"	"FLAG"
"Greece"	"GR"	"
"Liberia"	"LI"	"
"Panama"	"PN"	"
"United Kingdom"	"UK"	"
"Afars and Issas"	"FT"	"
"Afghanistan"	"AF"	"
"Albania"	"AL"	"
"Algeria"	"AG"	"
"American Samoa"	"AQ"	"
"Angola"	"AO"	"
"Antigua"	"AC"	"
"Argentina"	"AR"	"
"Australia"	"AS"	"
"Austria"	"AU"	"
"Bahamas"	"BF"	"
"Bahrain"	"BA"	"
"Bangladesh"	"BG"	"
"Barbados"	"BB"	"
"Belgium"	"BE"	con't

VESSEL REGISTRY	CODE	DATA LIMIT
"St. Lucia"	"ST"	"FLAG"
"St. Pierre Miquelon"	"SB"	"
"St. Vincent"	"VC"	"
"Senegal"	"SG"	"
"Seychelles Island"	"SE"	"
"Sierra Leone"	"SL"	"
"Singapore"	"SN"	"
"Somali Republic"	"SO"	"
"South Africa"	"SF"	"
"Spain"	"SP"	"
"Spanish Sahara"	"SS"	"
"Sp. Terr. of N. Morocco"	"ME"	"
"Sri Lanka (Ceylon)"	"CE"	"
"Sudan"	"SU"	"
"Surinam"	"NS"	"
"Sweden"	"SW"	"
"Switzerland"	"SZ"	"
"Syria"	"SY"	"
"Taiwan (Formosa)"	"TW"	"
"Tanzania"	"TZ"	"
"Thailand"	"TH"	"
"Togo"	"TO"	"
"Tonga"	"TN"	"
"Trinidad and Tobago"	"TD"	"
"Tunisia"	"TS"	"
"Turkey"	"TU"	"
"Turks and Caicos Islands"	"TK"	"
"Union Soviet Soc. Repub."	"UR"	"
"United Arab Emirates"	"TC"	"
"Uruguay"	"UY"	"
"Venezuela"	"VE"	"
"Vietnam, North"	"VN"	"
"Vietnam, South"	"VS"	"
"Virgin Islands"	"VO"	"
"Western Samoa"	"WS"	"
"Yemen (Aden)"	"YS"	"
"Yemen (Sana)"	"YE"	"
"Yugoslavia"	"YO"	"
"Zaire"	"CG"	"

VESSEL SERVICE	CODE	DATA LIMIT
"Freight"	"FRT"	"SERVICE"
"Public Freight"	"PFRT"	"
"Tank"	"TNK"	"
"Public Tank"	"PTNK"	"
"Passenger"	"PASS"	"
"Pleasure"	"PLEZ"	"
"Tug/Tow"	"TOW"	"
"Research"	"RES"	"
"School"	"SCOL"	"
"Industrial"	"IND"	"
"OTEC"	"OTEC"	"
"Unclassified"	"UNC"	"
"Public Unclassified"	"POTH"	"
"Mobile Offshore Drlng"	"MODU"	"
"Fixed Facility"	"PTFM"	"
"Fishing"	"FISH"	"
"Offshore Supply"	"OSV"	"
"Other"	"OTH"	"

VESSEL USE	CODE	DATA LIMIT
"Lash"	"LASH"	"USE"
"Break Bulk"	"BBLK"	"
"Offshore Supply"	"OSV"	"
"Towing"	"TOW"	"
"Ferry Vessel"	"FERY"	"
"Bulk Solid Cargo"	"BSLD"	"
"Production Platform"	"PROF"	"
"Process Facility"	"PROC"	"
"Cable Layer"	"CABL"	"
"Bulk Oil"	"OIL"	"
"Work Platform - General"	"WORK"	"
"Container"	"CONT"	"
"Roll on/Roll off"	"RORO"	"
"Unclassified"	"UNC"	"
"Combination"	"COMB"	"
"Gas Carrier"	"GAS"	"
"Ore-Bulk Oil"	"OBO"	"
"Offshore Transfer"	"DWP"	"
"Dredge"	"DRDG"	"
"Passenger"	"PSGR"	con't

VESSEL USE	CODE	DATA LIMIT
"Vessel is indescribable"	"NA"	"USE"
"Bulk Liquid Chemicals"	"CHEM"	"
"Crewboat"	"CREW"	"
"Mobile Drilling Unit"	"MODU"	"
"Drill Unit"	"DRIL"	"
"Energy General, Conv"	"ENER"	"
"Purse Seiner"	"PURS"	"
"Hook and Line"	"HOOK"	"
"Trawler"	"TRLR"	"
"Fishing, General"	"FSHG"	"
"Fishing Dredge"	"DRAG"	"
"Traps and Pots"	"TRAP"	"
"Bottom Shell Fishing"	"BTSF"	"
"Recreation"	"WREK"	"
"Research"	"RSCH"	"
"Other"	"OTH"	"
"Unknown"	"UNK"	"

VESSEL CONSTRUCTION	CODE	DATA LIMIT
"Aluminum"	"AL"	"HULL"
"Bronze"	"BZ"	"
"Concrete (Ferro-Cement)"	"CC"	"
"Copper"	"CU"	"
"Plastic (incl fiberglass)"	"PL"	"
"Iron"	"FE"	"
"Ductile Iron"	"DI"	"
"Steel"	"ST"	"
"Hi Strength"	"HS"	"
"Iron Nickel"	"IN"	"
"Stainless"	"SS"	"
"Wood"	"WD"	"
"Not Classified"	"NC"	"

VESSEL PROPULSION	CODE	DATA LIMIT
"Diesel Direct"	"DD"	"PROP"
"Diesel Electric"	"DE"	"
"Diesel Reduction"	"DR"	"
"Diesel, NEC"	"DN"	"
"Electric Motor"	"EM"	"
"Gas Engine"	"GE"	"
"Gas Turbine"	"GT"	"
"Sail"	"SA"	"
"Steam, General"	"SG"	"
"Steam Reciprocating"	"SR"	"
"Steam Turbine"	"ST"	"
"Steam Turboelectric"	"SE"	"
"Not self-propelled"	"NA"	"
"Self-prop, not classified"	"NC"	"
"Unknown"	"NK"	"

DESIGN OF VESSEL	CODE	DATA LIMIT
"Conventional Hull"	"CONV"	"DESIGN"
"Air Cushion Vehicle"	"ACV"	"
"Hydrofoil"	"HYD"	"
"Submersible"	"SUB"	"
"Integrated Tug/Barge"	"ITB"	"
"Semisubmersible Rig"	"SSUB"	"
"Tension Leg Rig"	"TLEG"	"
"Unpowered Barge"	"BRGE"	"
"Artificial Island"	"AISL"	"
"Unclassified"	"UNC"	"
"Captured Air Bubble"	"CAB"	"
"STILETTO Catamaran"	"ZOOM"	"
"Mono Hulled Sailboat (J-24)"	"SLUG"	"
"Multi-hulled Displacement"	"MHD"	"
"Jackup Rig"	"JKUP"	"

BUREAU CLASSIFYING VESSELS	CODE	DATA LIMIT
"American Bureau of Shipping"	"ABS"	"SOCIETY"
"Bureau Veritas"	"BV"	"
"Det Norske Veritas"	"DNV"	"
"Germanischer Lloyds"	"DUELL"	"
"Lloyds Register of Shipping"	"LLOYDS"	"
"Nippon Kaiji Kyokai (Japan)"	"NKK"	"
"Registrano Italiano Navale"	"RIN"	"
"Not Classified"	"NC"	"
"Unknown"	"UNK"	"

PILOT STATUS	CODE	DATA LIMIT
"None"	"NONE"	"PILOT"
"State Licensed"	"STLI"	"
"Federal Licensed"	"FDLI"	"
"State and Federal Licensed"	"SFLI"	"
"Not Classified"	"NC"	"
"Foreign Pilot"	"FGNP"	"

MANNING STATUS	CODE	DATA LIMIT
"Licensed Individual"	"LDID"	"PIC"
"Unlicensed/Undocumented Pers"	"UNLD"	"
"Unmanned"	"UNMD"	"
"Unknown"	"UNK"	"
"Not Classified"	"NC"	"
"Not Applicable"	"NA"	"

UNDERWAY STATUS

40

CODE

DATA LIMIT

"Afloat, Undrwy Making Way"	"AU"	"VSLSTATE"
"Afloat, Undrwy Not Making Way"	"AN"	"
"Moored or Anchored"	"MO"	"
"Docked"	"DO"	"
"Drydocked"	"DD"	"
"Aground"	"GR"	"
"Jacked Up"	"JU"	"
"Fixed Facility"	"FF"	"
"Unknown"	"UN"	"

VESSEL CONDITION AFTER CASUALTY

"Was Vessel a Total Loss?"	"A"	"ABC"
"Was Seaworthiness Affected? Yes"	"B"	"
"Was Seaworthiness Affected? No"	"C"	"
"Unknown"	"D"	"

BEGIN "PCAS" RETRIEVAL CODES

See the CONTROL FILE information page for this file to obtain the complete list of retrievable fields.

TYPE OF CREWMEMBER	CODE	DATA LIMIT
"Deck Crew"	"DC"	"STATUS"
"Deck Officer"	"DO"	"
"Engine Crew"	"EC"	"
"Engineering Officer"	"EO"	"
"Government Employee"	"GE"	"
"Longshore/Harbor Worker"	"LS"	"
"Master"	"MS"	"
"Passenger"	"PA"	"
"Platform Work"	"PW"	"
"Steward Department"	"SD"	"
"Tankerman"	"TM"	"
"Visitor"	"VI"	"
"Unknown"	"UN"	"
"Not Otherwise Classified"	"NC"	"
NATURE OF THE ACCIDENT		
"Natural Causes"	"NAT CAU"	"NATACCID"
"Homicide"	"HOMICIDE"	"
"Suicide"	"SUICIDE"	"
"Attempted Suicide"	"ATT SUICID"	"
"Disappearance"	"VANISH"	"
"Slip/Fall-Stairs"	"SLPXFL STR"	"
"Slip/Fall-Gangway"	"SLPXFL GWY"	"
"Slip/Fall-Deck"	"SLPXFL DCK"	"
"Slip/Fall-NOC"	"SLPXFL NOC"	"
"Fall, Same Level"	"FL, 1 LVL"	"
"Fall Into Water"	"FL N2 WATR"	"
"Fall, Not Classified"	"FL NOC"	"
"Fall, Into Hold/Tank"	"FL HLDXTNK"	"
"Fall, Other Level"	"FL OTH LVL"	"
"Struck by Falling Object"	"FALNG OBJ"	"
"Struck by Flying Object"	"FLYING OBJ"	"
"Struck by Moving Object"	"MVNG OBJ"	"
"Struck by Vessel"	"HIT BY VSL"	"
"Struck by Object, NOC"	"SBO NOC"	"
"Exposure"	"EXPOSURE"	con't

NATURE OF THE ACCIDENT	CODE	DATA LIMIT
"Asphyxiation"	"SUFFOCATE"	"NATACCID"
"Diving Accident"	"DIVE CAS"	"
"Bumped Fixed Object"	"STR FX OBJ"	"
"Cargo Handling-NOC"	"CGO HNDLNG"	"
"Line Handling"	"LN HNDLNG"	"
"Operate Machinery"	"HAND TOOLS"	"
"Non-Electric Burn"	"BURN"	"
"Scalded"	"SCALDED"	"
"Electric Burn, Shock"	"ELEC BURN"	"
"Caught in Lines"	"CGT IN LNS"	"
"Pinched/Crushed"	"CRUSHED"	"
"Overexertion"	"OVEREXERT"	"
"Sprain/Strain-NOC"	"SPRNKSTRN"	"
"Cuts, Bruises-NOC"	"CUTXBUISE"	"
"Galley Accident"	"GALLEY ACC"	"
"Altercation"	"FIGHT"	"
"Unknown"	"UNKNOWN"	"
"Not Classified"	"NC"	"
"Hypothermia"	"HYPOTHERM"	"

NATURE OF THE INJURY

"Abrasion"	"ABRASION"	"NATINJ"
"Aggrevated Old Injury"	"AG OLD INJ"	"
"Blister"	"BLISTER"	"
"Bruise"	"BRUISE"	"
"Burn"	"BURN"	"
"Chemical Burn"	"CHEM BURN"	"
"Concussion"	"CONCUSSION"	"
"Crushed"	"CRUSHED"	"
"Cut"	"CUT"	"
"Drowning"	"DROWN"	"
"Electric Shock"	"ELEC SHOCK"	"
"Flash Burn"	"FLASH BURN"	"
"Fracture"	"FRACTURE"	"
"Hemorrhoid"	"HEMORRHOID"	"
"Hernia"	"HERNIA"	"
"Puncture"	"PUNCTURE"	con't

NATURE OF THE INJURY	43	CODE	DATA LIMIT
"Sprain"		"SPRAIN"	"NATINJ"
"Strain"		"STRAIN"	"
"Multiple"		"MULTIPLE"	"
"Unknown"		"UNKNOWN"	"
"Not Classified"		"NC"	"
INJURED PART			
"Ankle"		"ANKLE"	"BODYPART"
"Arm"		"ARM"	"
"Back"		"BACK"	"
"Chest"		"CHEST"	"
"Eye"		"EYE"	"
"Finger"		"FINGER"	"
"Foot"		"FOOT"	"
"Groin"		"GROIN"	"
"Hand"		"HAND"	"
"Head"		"HEAD"	"
"Hip"		"HIP"	"
"Knee"		"KNEE"	"
"Leg"		"LEG"	"
"Neck"		"NECK"	"
"Shoulder"		"SHOULDER"	"
"Stomach"		"STOMACH"	"
"Trunk"		"TRUNK"	"
"Multiple Injuries"		"MULTIPLE INJ"	"
"Unknown"		"UNKNOWN"	"
"Not Classified"		"NC"	"
RESULT OF PERSONNEL CASUALTY			
"Death, Vessel Casualty"		"DVC"	"RESULT"
"Death, No Vessel Casualty"		"DNVC"	"
"Injury, Vessel Casualty"		"IVC"	"
"Injury, No Vessel Casualty"		"INVC"	"
"Missing, Vessel Casualty"		"MVC"	"
"Missing, No Vessel Casualty"		"MNVC"	"

ACCIDENT CAUSE	CODE	DATA LIMIT
"Intoxication, Another"	"INTXA"	"CAUSE1+"
"Intoxication, Self"	"INTXS"	"
"Adverse Weather"	"WTHR"	"
"Carelessness, Another"	"CRLSO"	"
"Carelessness, Self"	"CRLSS"	"
"Chemical Reaction"	"CHEMR"	"
"Deck Cluttered"	"DCKC"	"
"Deck Slippery"	"DCKS"	"
"Equipment Failure"	"EQPF"	"
"Failure to use PFD * "	"FPFD"	"
"Failure to use PPE ** "	"FPPE"	"
"Improper Lighting"	"IMLT"	"
"Improper Loading/Storage"	"IMLS"	"
"Improper Maintenance"	"IMNT"	"
"Improper Supervision"	"ISPR"	"
"Improper Tools/Equipment"	"OTPE"	"
"Inadequate/Missing Guarding"	"IGRD"	"
"Inadequate/Missing Railing"	"IRAL"	"
"Inadequate Training"	"ITNG"	"
"Insufficient Ventilation"	"IVNT"	"
"Lack of Available PFD"	"MPFD"	"
"Lack of Available PPE"	"MPPE"	"
"Material Failure"	"MATF"	"
"Misuse of Tools/Equipment"	"MIST"	"
"Mooring Line Surge"	"MRGS"	"
"Narcotics (other than alcohol)"	"NARC"	"
"Overloading"	"OVRLD"	"
"Physical Factors, Another"	"PHSO"	"
"Physical Factors, Self"	"PHSS"	"
"Psychological Factors, Another"	"PSYCO"	"
"Psychological Factors, Self"	"PSYCS"	"
"Unsafe Movement, Another"	"UNSMA"	"
"Unsafe Movement, Self"	"UNSMS"	"
"Unsafe Practice, Another"	"UNSPA"	"
"Unsafe Practice, Self"	"UNSPS"	"
"Vessel Casualty"	"VSLC"	"
"Unknown"	"UNK"	"
"Not Elsewhere Classified"	"NEC"	"

* PFD - Personal Flotation Device

** PPE - Personal Protection Equipment

LOCATION INDIV ON VSL	CODE	DATA LIMIT
"Aft Area"	"AFT"	"PLOC"
"Bridge"	"BRDG"	"
"Cargo Holds"	"CRGH"	"
"Cargo Pump Room"	"CRGP"	"
"Cargo Tanks"	"CRGT"	"
"Deck Stores"	"DKST"	"
"Deck (Open)"	"DECK"	"
"Engine Room"	"ENG"	"
"Engineer Stores"	"ENGR"	"
"Fire Room"	"FIRM"	"
"Forespeak"	"FORP"	"
"Forward Area"	"FWD"	"
"Fuel Tanks"	"FUEL"	"
"Galley"	"GALY"	"
"Laundry"	"LNDY"	"
"Machinery Spaces"	"MACH"	"
"Masts, Booms, Rigging"	"RIGG"	"
"Mid-Body Area"	"AMID"	"
"Offices"	"OFC"	"
"Paint Locker"	"PNTL"	"
"Passageway"	"PSWY"	"
"Quarters"	"QTRS"	"
"Segregated Ballast Tank"	"SGBT"	"
"Shaft Alley"	"SHFT"	"
"Steering Space"	"STRG"	"
"Void/Cofferdam"	"VOID"	"
"Windless Room"	"WNDL"	"
"Not Elsewhere Classified"	"NEC"	"
"Unknown"	"UNK"	"

ACTIVITY OF PERSON	CODE	DATA LIMIT
"Deck Duty"	"DD"	"ACTIVITY"
"Drilling"	"DR"	"
"Engine Duty"	"ED"	"
"Fishing"	"FS"	"
"Handling Cargo"	"HC"	"
"Off Duty"	"OD"	"
"Passenger"	"PA"	"
"Steward Duty"	"SD"	"
"Unknown"	"UN"	"
"Not Otherwise Classified"	"NC"	"

MARINE SAFETY OFFICES (MSO)

(Field Code - OFFICE)

DIST

17 ANC - Anchorage, AK
 5 BAL - Baltimore, MD
 1 BOS - Boston, MA
 9 BUF - Buffalo, NY
 7 CHA - Charleston, SC
 9 CHI - Chicago, IL
 2 CIN - Cincinnati, OH
 9 CLE - Cleveland, OH
 8 COR - Corpus Christi, TX
 9 DET - Detroit, MI
 9 DUL - Duluth, MN
 8 GAL - Galveston, TX
 5 HMR - Hampton Roads, VA
 5 HMR - Portsmouth, VA
 14 HON - Guam, GU
 14 HON - Honolulu, HI
 14 HON - Kobe, Japan
 8 HOU - Houston, TX
 2 HUN - Huntington, WV
 7 JAC - Jacksonville, FL
 17 JUN - Juneau, AK
 11 LOS - Long Beach, CA
 11 LOS - Los Angeles, CA
 2 LOU - Louisville, KY
 2 MEM - Memphis, TN
 7 MIA - Miami, FL
 9 MIL - Milwaukee, WI
 8 MOB - Mobile, AL
 8 MOR - Morgan City, LA

DIST

2 NAS - Nashville, TN
 8 NEW - New Orleans, LA
 3 NYC - Albany, NY
 3 NYC - New London, NY
 3 NYC - New York City
 3 NYC - Rotterdam, NY
 2 PAD - Paducah, KY
 8 PAT - Port Arthur, TX
 3 PHI - Philadelphia, PA
 3 PIT - Pittsburgh, PA
 1 POM - Portland, ME
 13 POR - Portland, OR
 1 PRO - Providence, RI
 7 SAV - Savannah, GA
 11 SDC - San Diego, CA
 13 SEA - Seattle, WA
 13 SEA - Tacoma, WA
 12 SFC - Oakland, CA
 12 SFC - San Francisco
 7 SJP - San Juan
 2 SLM - St Louis MO
 14 SNG - Singapore
 9 STB - Sturgeon Bay, WI
 2 STP - St Paul, MN
 7 TAM - Tampa, FL
 9 TOL - Toledo, OH
 17 VAL - Valdez, AK
 5 WNC - Wilmington, NC

Note: The district numbers that appear above reflect the district configuration prior to the District reorganization that took place in 1988.

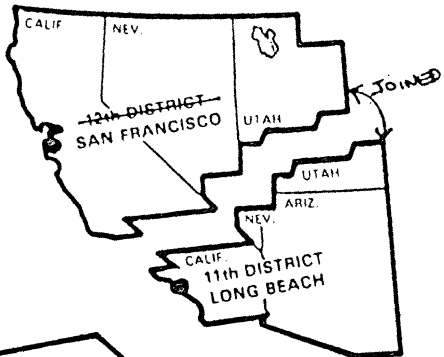
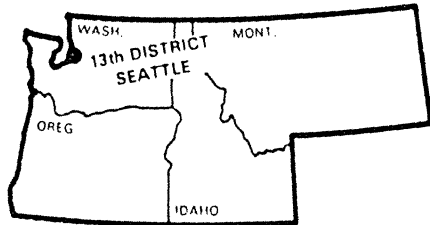
E
 . N
 D

VESSEL REGISTRY	CODE	DATA LIMIT
"Belize"	"BH"	"FLAG"
"Bermuda"	"BD"	"
"Bolivia"	"BL"	"
"Brazil"	"BR"	"
"British Indian Ocean Terr"	"IO"	"
"British Virgin Islands"	"VI"	"
"Brunei"	"BZ"	"
"Bulgaria"	"BU"	"
"Burma"	"BM"	"
"Burundi"	"BI"	"
"Cambodia"	"CB"	"
"Cameroon"	"CM"	"
"Canada"	"CA"	"
"Canal Zone"	"PQ"	"
"Cape Verde Islands"	"CV"	"
"Cayman Island"	"CJ"	"
"Chile"	"CI"	"
"China"	"CH"	"
"Coco Islands"	"CK"	"
"Columbia"	"CO"	"
"Conga"	"CF"	"
"Costa Rica"	"CS"	"
"Cuba"	"CU"	"
"Cyprus"	"CX"	"
"Czechoslovakia"	"CZ"	"
"Dahomey"	"DM"	"
"Denmark"	"DA"	"
"Dominica"	"DO"	"
"Dominican Republic"	"CR"	"
"Ecuador"	"EC"	"
"Egypt"	"EG"	"
"El Salvador"	"ES"	"
"Equatorial Guinea"	"EK"	"
"Ethiopia"	"ET"	"
"Faeroes Island"	"FO"	"
"Falkland Island"	"FA"	"
"Finland"	"FI"	"
"France"	"FR"	"
"French Antartic Lands"	"FS"	"
"French Guiana"	"FG"	con't

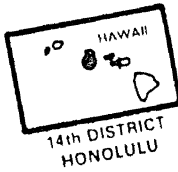
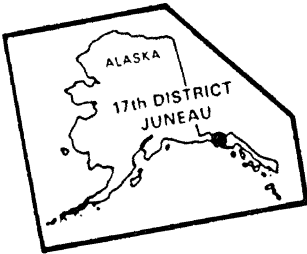
VESSEL REGISTRY	CODE	DATA LIMIT
"Gabon"	"GB"	"FLAG"
"Gambia"	"GA"	"
"Germany, East"	"GC"	"
"Germany, West"	"GE"	"
"Ghana"	"GH"	"
"Gibraltar"	"GI"	"
"Grand Fenwick (Duchy of)"	"PS"	"
"Greenland"	"GL"	"
"Grenada"	"GJ"	"
"Guadeloupa"	"GP"	"
"Guam"	"GQ"	"
"Guatemala"	"GT"	"
"Guinea"	"GV"	"
"Guyana"	"GY"	"
"Haiti"	"HA"	"
"Honduras"	"HO"	"
"Hong Kong"	"HK"	"
"Hungary"	"HU"	"
"Iceland"	"IC"	"
"Indonesia"	"ID"	"
"India"	"IN"	"
"Iran"	"IR"	"
"Iraq"	"IZ"	"
"Ireland"	"EI"	"
"Israel"	"IS"	"
"Italy"	"IT"	"
"Ivory Coast"	"IV"	"
"Jamaica"	"JM"	"
"Japan"	"JA"	"
"Jordan"	"JO"	"
"Kenya"	"KE"	"
"Korea, North"	"KN"	"
"Korea, South"	"KS"	"
"Kuwait"	"KU"	"
"Laos"	"LA"	"
"Lebanon"	"LE"	"
"Libya"	"LY"	"
"Luxembourg"	"LX"	"
"Macao"	"MC"	"
"Malagasy Republic"	"MA"	con't

VESSEL REGISTRY	CODE	DATA LIMIT
"Malaysia"	"MY"	"FLAG"
"Maldives"	"MV"	"
"Malta"	"MT"	"
"Martinique"	"MB"	"
"Mauritania"	"MR"	"
"Mauritius"	"MP"	"
"Mexico"	"MX"	"
"Mongolia"	"MG"	"
"Montserrat"	"MH"	"
"Monaco"	"MN"	"
"Morocco"	"MO"	"
"Mozambique"	"MZ"	"
"Nauru"	"NR"	"
"Nepal"	"NP"	"
"Netherlands"	"NL"	"
"Netherlands Antilles"	"NA"	"
"New Zealand"	"NZ"	"
"Nicaragua"	"NU"	"
"Nigeria"	"NI"	"
"Norway"	"NO"	"
"Oman"	"MU"	"
"Pakistan"	"PK"	"
"Palestine"	"YA"	"
"Papua New Guinea"	"PP"	"
"Paracel Islands"	"PF"	"
"Paraguay"	"PA"	"
"Peru"	"PE"	"
"Philippines"	"RP"	"
"Poland"	"PL"	"
"Portugal"	"PO"	"
"Portuguese Guinea"	"PU"	"
"Portuguese Timor"	"PT"	"
"Puerto Rico"	"RQ"	"
"Qatar"	"AQ"	"
"Reunion"	"RE"	"
"Romania"	"RO"	"
"Sao Tome E Principe"	"TR"	"
"Saudi Arabia"	"SA"	"
"St. Christopher-Nevis"	"SC"	"
"St. Helena"	"SH"	con't

**Pacific Area
COMPACAREA**



11th DISTRICT LONG BEACH
CALIF. NEV. ARIZ.

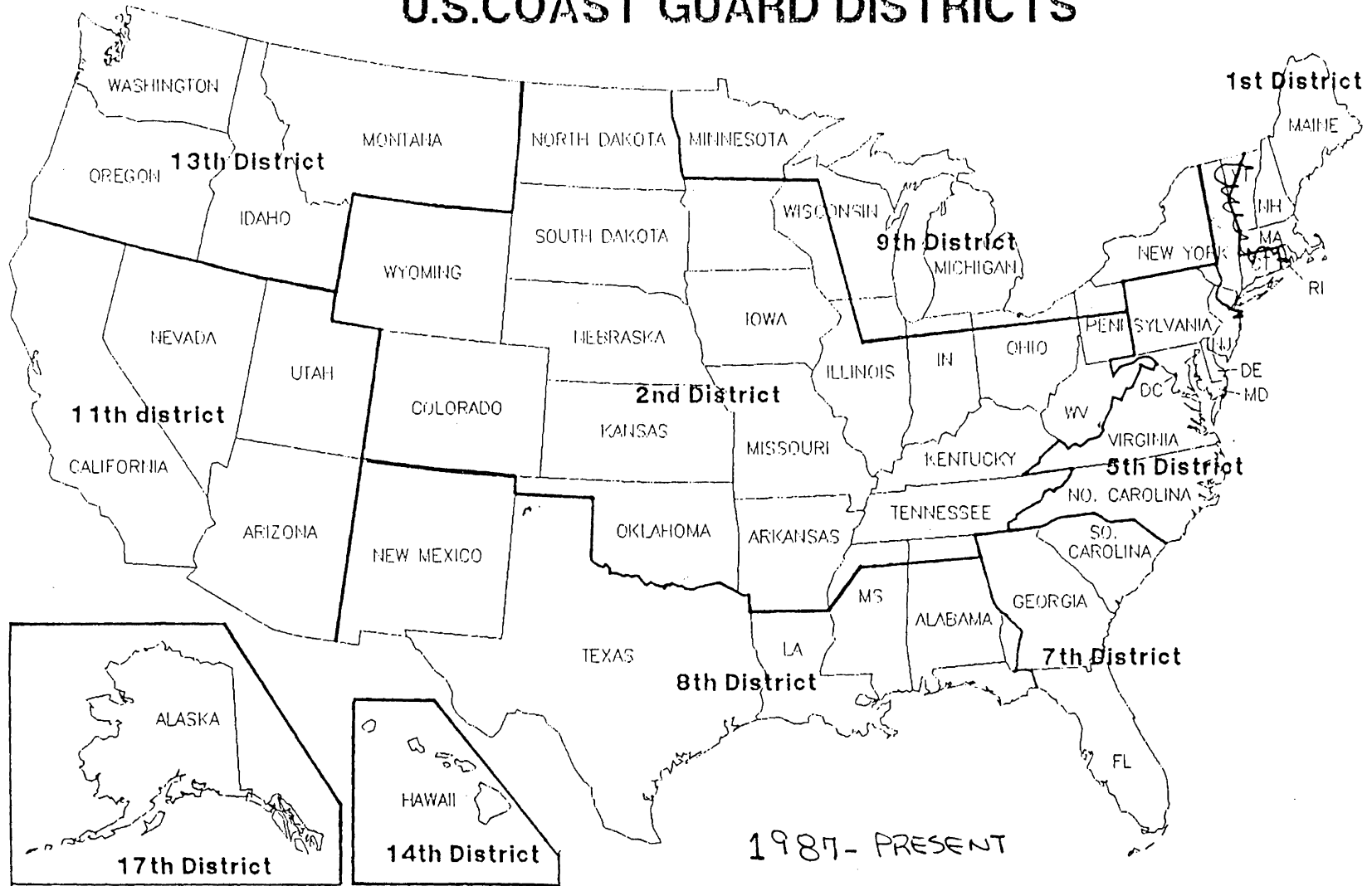


**Atlantic Area
COMLANTAREA**



- Prior to 1987 -

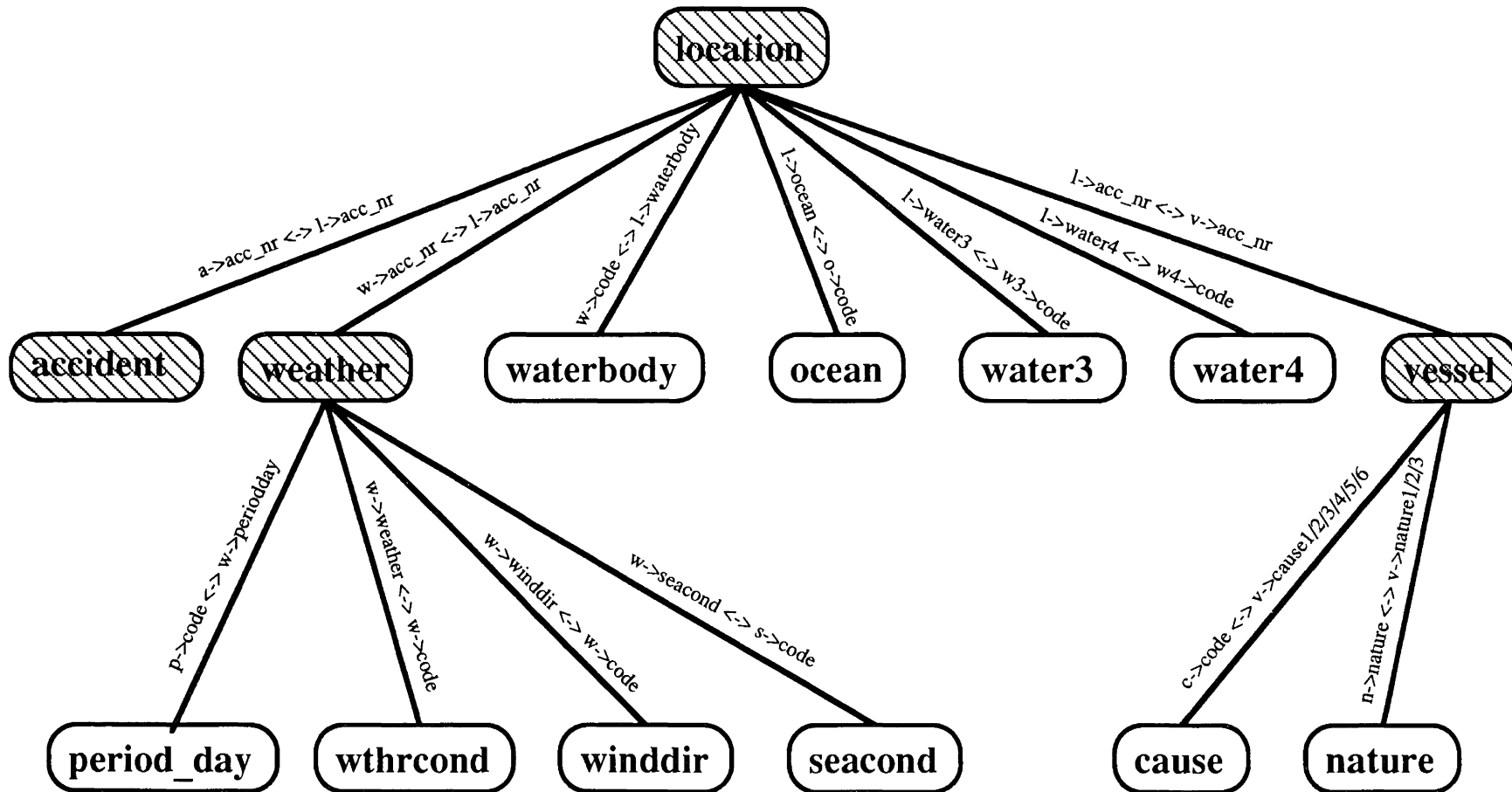
U.S. COAST GUARD DISTRICTS



Appendix 3:

Tables for inFOcus and INGRES

InFOcus Database Relations *



* The same set-up has been used in INGRES

___ Sat Jun 13 17:35:20 1992 _____

1> help accident, location, weather, vessel, cause, natcode, ocean, water3,
2> water4, period_day, sea_cond, wnddir, wthrcond, wtrbdy

Name: accident
Owner: iOot
Created: 11-jun-1992 11:00:00
Type: user table
Version: ING6.0

Column Information:

Column Name	Type	Length	Nulls	Defaults	Key Seq
record#	integer	4	yes	no	
acc_nr	integer	4	yes	no	
case_nr	c	10	yes	no	
dateacc	date		yes	no	
data_entry	date		yes	no	
numvslidam	integer	2	yes	no	
numvslinv	integer	2	yes	no	

Name: location
Owner: iOot
Created: 12-jun-1992 17:42:00
Type: user table
Version: ING6.0

Column Information:

Column Name	Type	Length	Nulls	Defaults	Key Seq
record#	integer	4	yes	no	
acc_nr	integer	4	yes	no	
case_nr	c	10	yes	no	
waterbody	c	7	yes	no	
district	integer	1	yes	no	
ocean	c	1	yes	no	
water3	c	1	yes	no	
water4	c	1	yes	no	
water5	c	1	yes	no	
latitude	float	8	yes	no	
longitude	float	8	yes	no	
rivermp	float	8	yes	no	

Name: weather
Owner: iOot
Created: 11-jun-1992 11:00:00
Type: user table
Version: ING6.0

Column Information:

Column Name	Type	Length	Nulls	Defaults	Key Seq
record#	integer	4	yes	no	
acc_nr	integer	4	yes	no	
case_nr	c	10	yes	no	
periodday	c	1	yes	no	
weather	c	2	yes	no	
wnddir	c	3	yes	no	

windspeed	c	3	yes	no
visibility	c	5	yes	no
seacord	c	4	yes	no

Name: vessel
 Owner: i0ot
 Created: 11-jun-1992 10:52:00
 Type: user table
 Version: ING6.0

Column Information:

Column Name	Type	Length	Nulls	Defaults	Key Seq
record#	integer	4	yes	no	
acc_nr	integer	4	yes	no	
case_nr	c	10	yes	no	
vessel_id	c	8	yes	no	
vesselname	c	40	yes	no	
flag	c	2	yes	no	
yearcons	integer	4	yes	no	
vslservice	c	4	yes	no	
vsluse	c	4	yes	no	
length	integer	4	yes	no	
tonnage	integer	4	yes	no	
material	c	2	yes	no	
propulsion	c	2	yes	no	
hp	integer	4	yes	no	
design	c	4	yes	no	
vslstate	c	2	yes	no	
towconfig	c	3	yes	no	
persincharge	c	4	yes	no	
society	c	6	yes	no	
company	c	10	yes	no	
pilotstatus	c	4	yes	no	
seaworthy	c	1	yes	no	
cause1	c	7	yes	no	
cause2	c	7	yes	no	
cause3	c	7	yes	no	
cause4	c	7	yes	no	
cause5	c	7	yes	no	
cause6	c	7	yes	no	
nature1	c	7	yes	no	
nature2	c	7	yes	no	
nature3	c	7	yes	no	

Name: cause
 Owner: i0ot
 Created: 13-jun-1992 17:27:00
 Type: user table
 Version: ING6.0

Column Information:

Column Name	Type	Length	Nulls	Defaults	Key Seq
code	c	7	yes	no	
text	c	80	yes	no	

Name: natcode

Owner: iOot
Created: 13-jun-1992 16:48:00
Type: user table
Version: ING6.0

Column Information:

Column Name	Type	Length	Nulls	Defaults	Key Seq
nature	c	6	yes	no	
text	c	80	yes	no	

Name: ocean
Owner: iOot
Created: 13-jun-1992 17:01:00
Type: user table
Version: ING6.0

Column Information:

Column Name	Type	Length	Nulls	Defaults	Key Seq
code	c	1	yes	no	
text	c	20	yes	no	

Name: water3
Owner: iOot
Created: 13-jun-1992 17:14:00
Type: user table
Version: ING6.0

Column Information:

Column Name	Type	Length	Nulls	Defaults	Key Seq
code	c	1	yes	no	
text	c	25	yes	no	

Name: water4
Owner: iOot
Created: 13-jun-1992 17:16:00
Type: user table
Version: ING6.0

Column Information:

Column Name	Type	Length	Nulls	Defaults	Key Seq
code	c	1	yes	no	
text	c	25	yes	no	

Name: period_day
Owner: iOot
Created: 13-jun-1992 16:54:00
Type: user table
Version: ING6.0

Column Information:

Column Name	Type	Length	Nulls	Defaults	Key Seq
code	c	1	yes	no	

text c 10 yes no

Name: sea_cond
Owner: iOot
Created: 13-jun-1992 16:58:00
Type: user table
Version: ING6.0

Column Information:

Column Name	Type	Length	Nulls	Defaults	Key Seq
code	c	4	yes	no	
text	c	20	yes	no	

Name: wnddir
Owner: iOot
Created: 13-jun-1992 17:23:00
Type: user table
Version: ING6.0

Column Information:

Column Name	Type	Length	Nulls	Defaults	Key Seq
code	c	3	yes	no	
text	c	25	yes	no	
direction	float	8	yes	no	

Name: wthrcond
Owner: iOot
Created: 13-jun-1992 17:21:00
Type: user table
Version: ING6.0

Column Information:

Column Name	Type	Length	Nulls	Defaults	Key Seq
code	c	2	yes	no	
text	c	25	yes	no	

Name: wtrbdy
Owner: iOot
Created: 13-jun-1992 17:18:00
Type: user table
Version: ING6.0

Column Information:

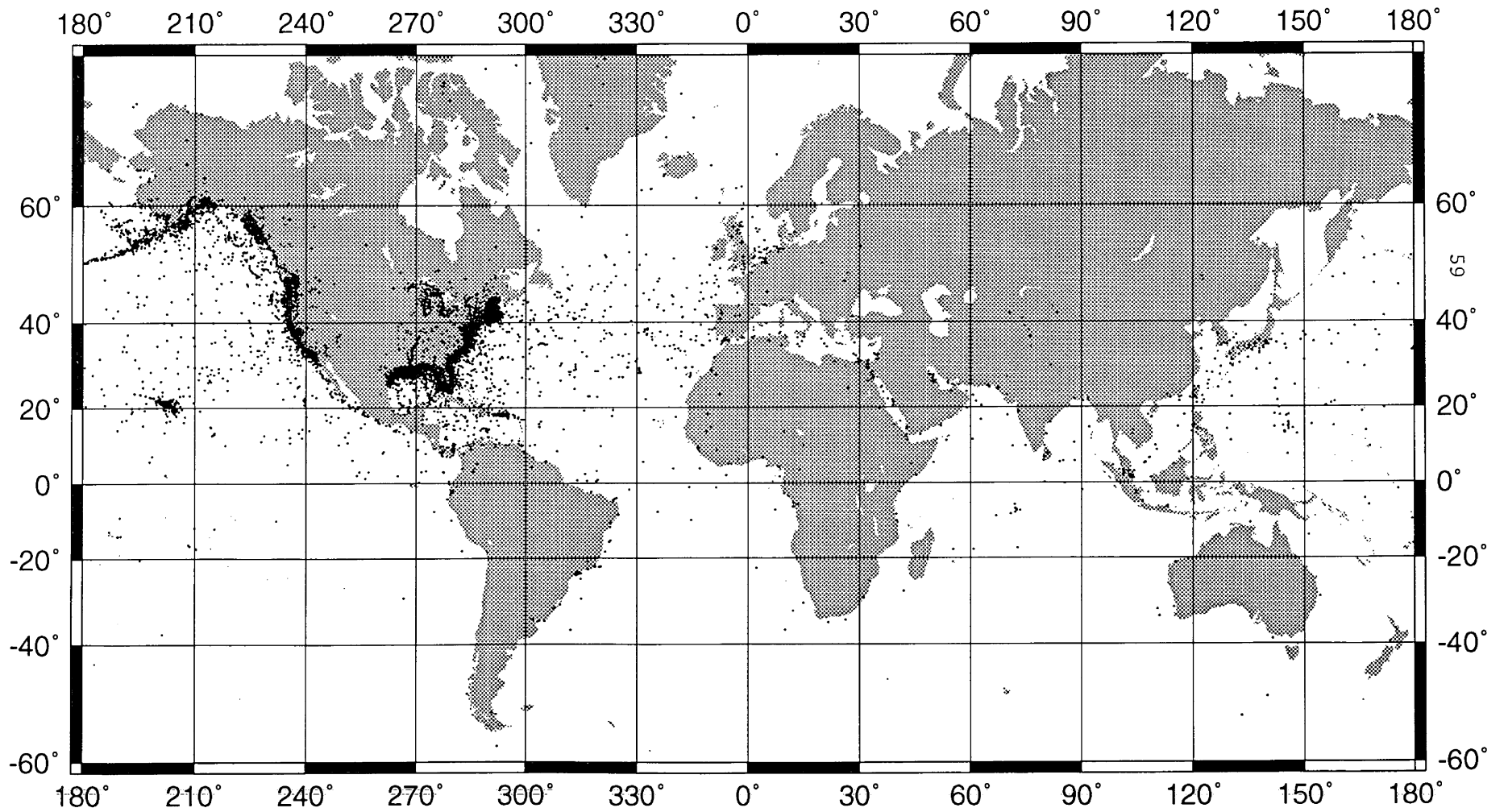
Column Name	Type	Length	Nulls	Defaults	Key Seq
code	c	7	yes	no	
text	c	80	yes	no	

End of Request

Appendix 4:

Graphs of the Data and Derived Statistics

CAS Accident Data

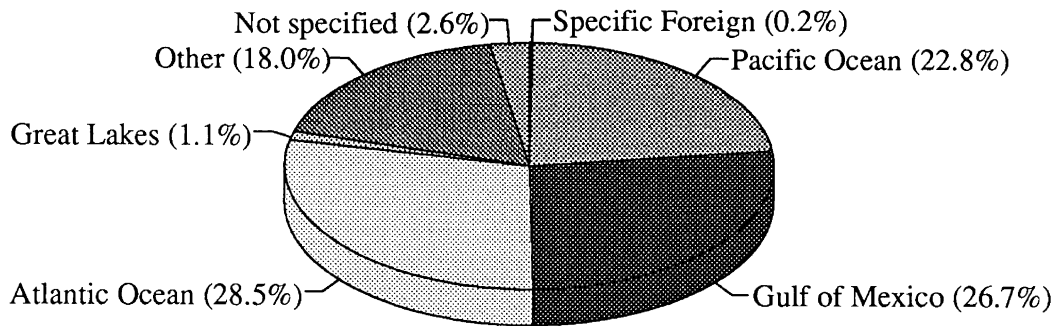


Distribution of Accidents

US Coast Guard Marine Accident Database

Data collected: 1982 - 1991

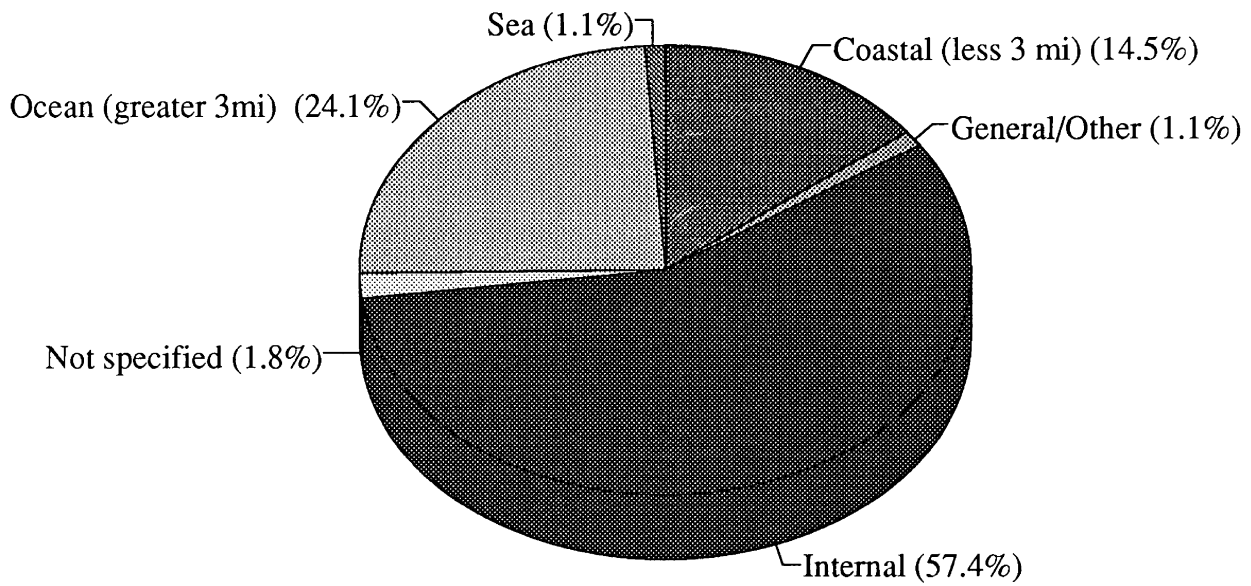
Total: 38141



Accident Distribution

US Coast Guard Marine Accident Database

Total: 37711

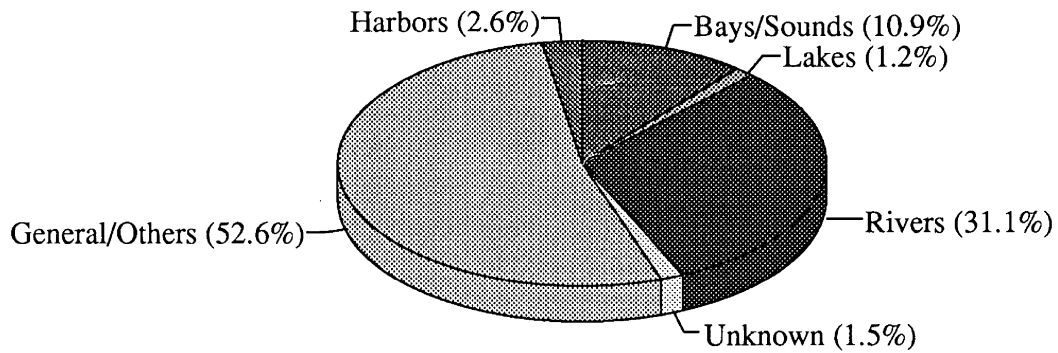


Accident Distribution

US Coast Guard Marine Accident Database

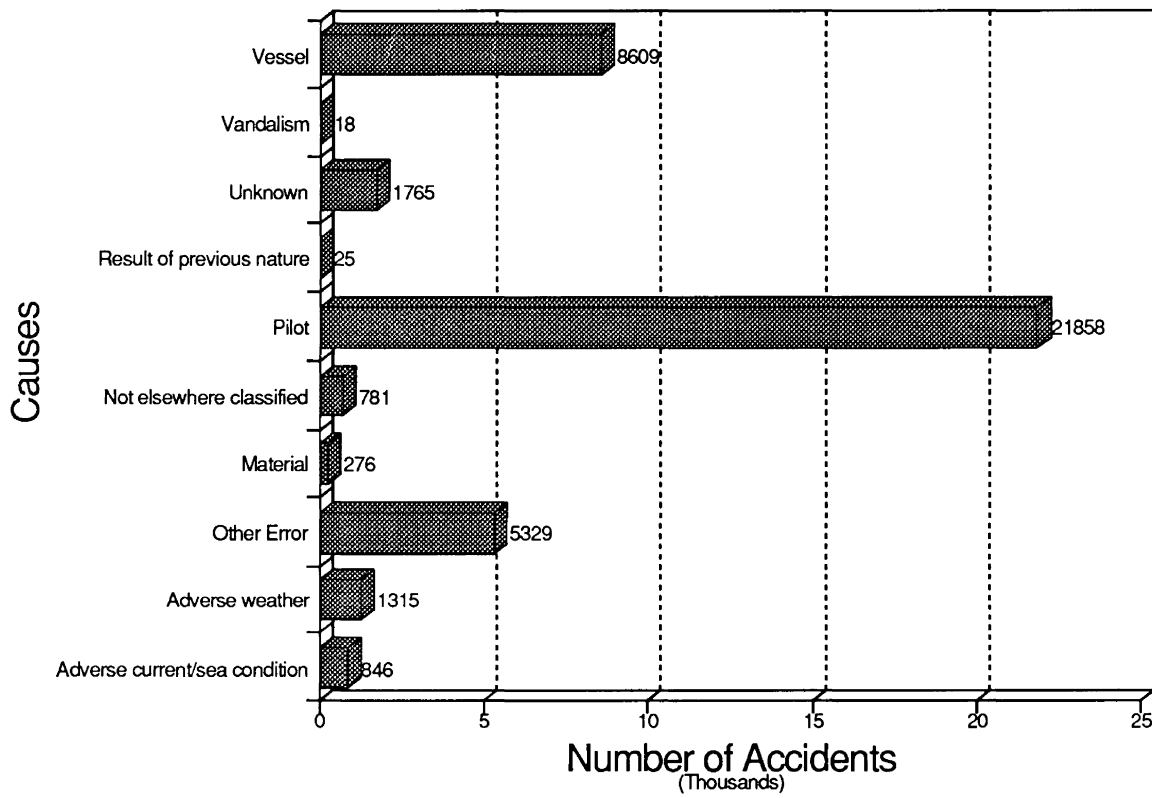
Data collected: 1982 - 1991

Total : 37263

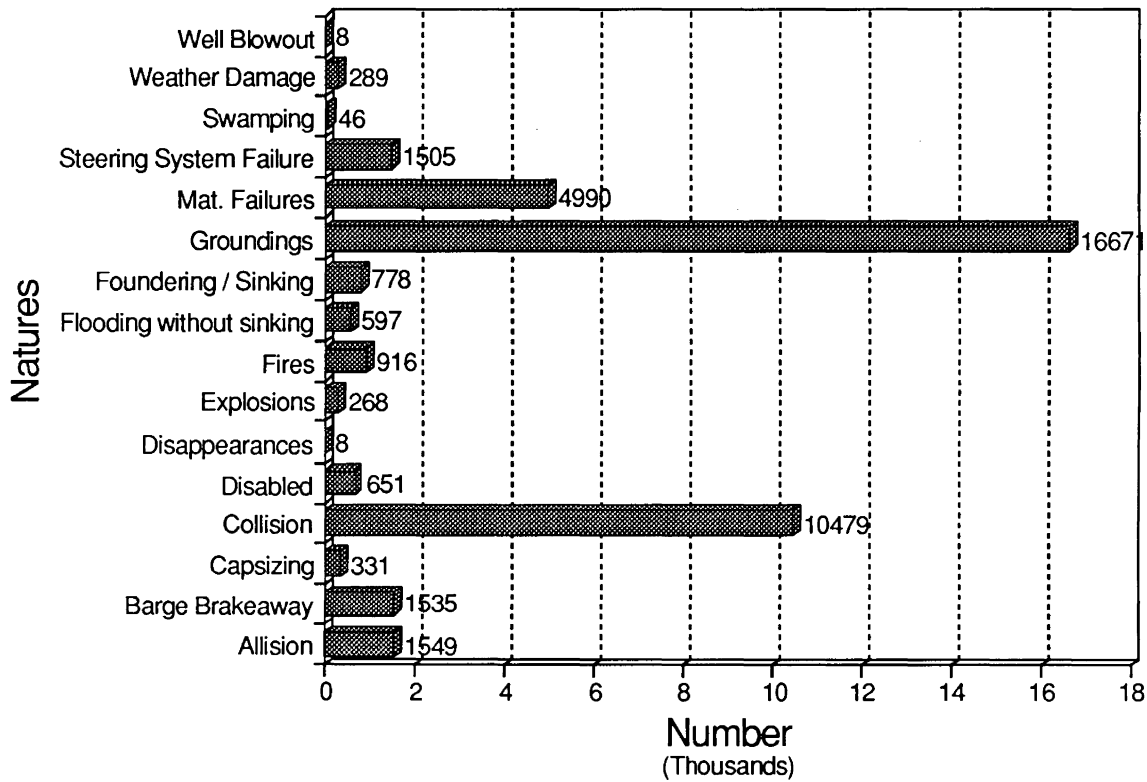


Causes vs. Accident

US Coast Guard Marine Accident Database

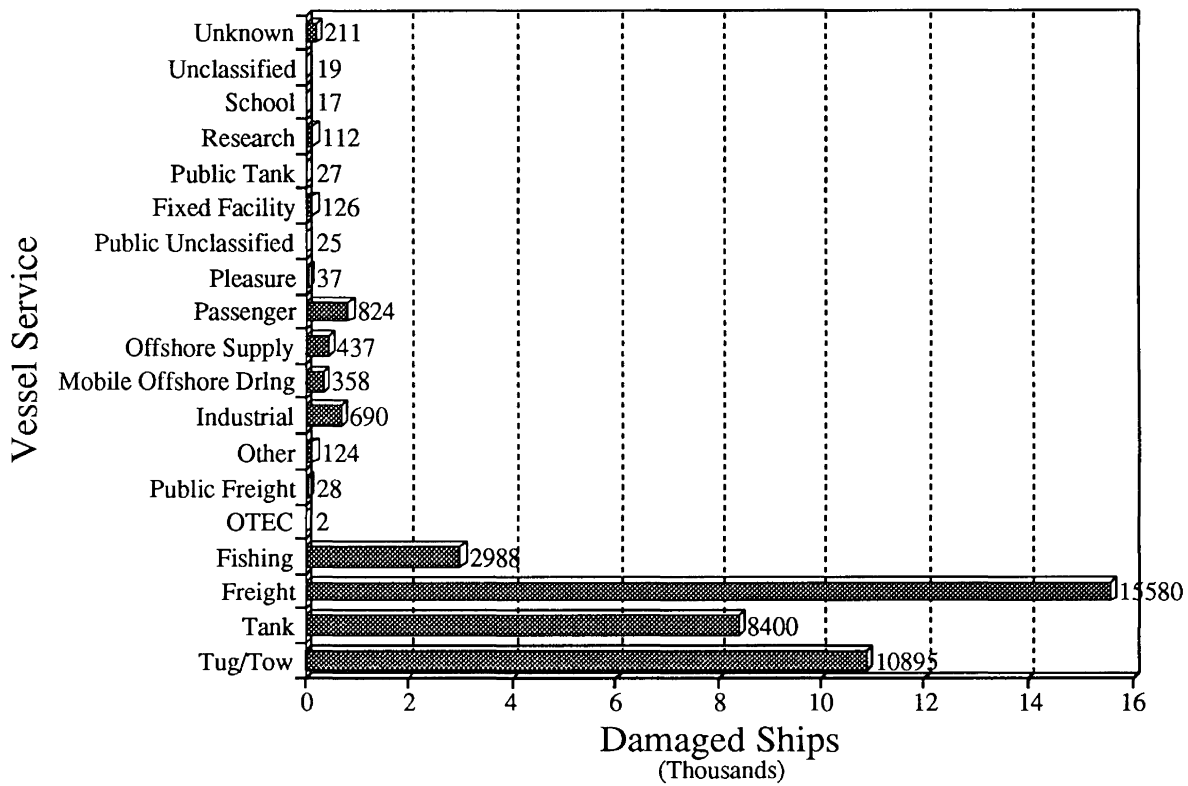


Nature of Accidents vs. Number US Coast Guard Marine Accident Database



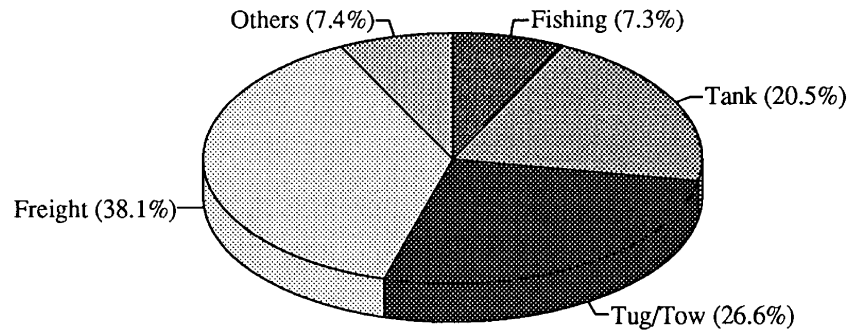
Vessel Service vs. Number

US Coast Guard Marine Accident Database



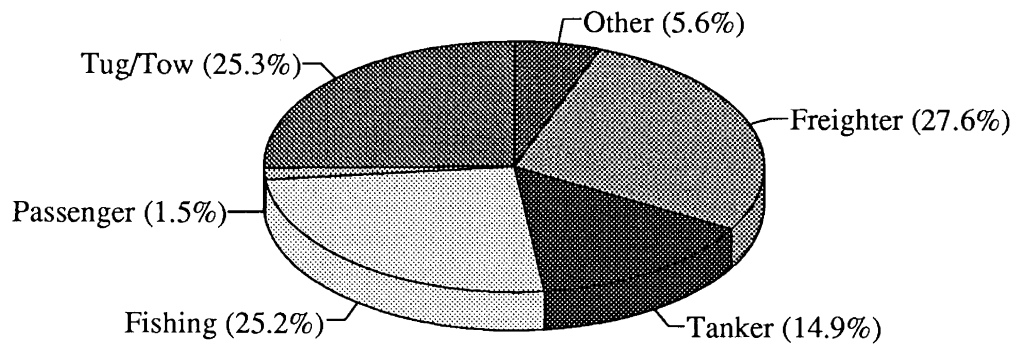
Vessel Service vs. Number

US Coast Guard Marine Accident Database



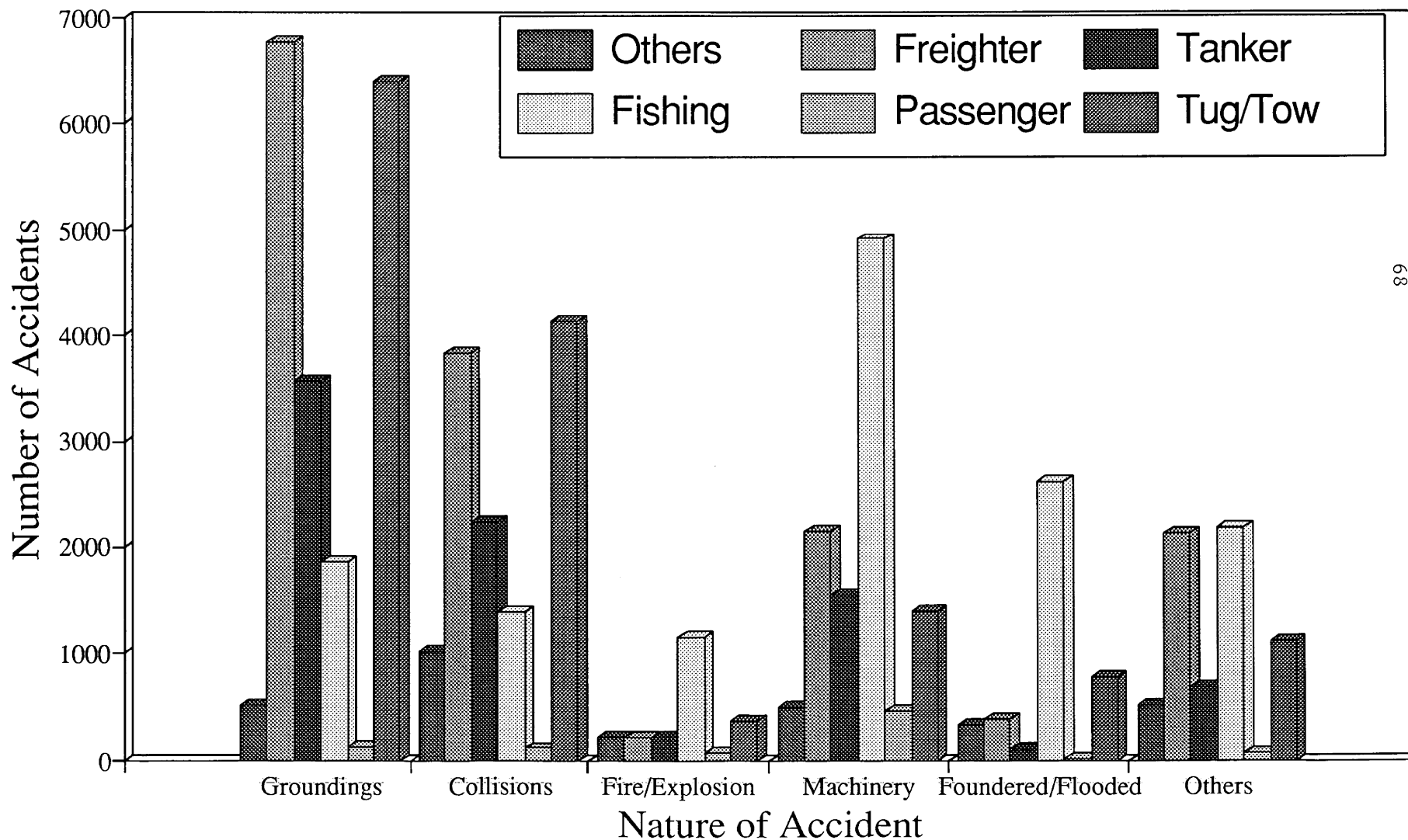
Accidents by Type of Vessel

US Coast Guard Marine Accident Database



Accidents by Type of Vessel

US Coast Guard Marine Accident Database



Appendix 5:

Source Codes of Auxiliary Programmes in C

program vessel.c

```
#include<stdio.h>
#include<string.h>

int del_blanks (string)
char string[];
{
    int i = 0;

    while (string[i++] != '\0');

    while ((--i >= 0) && !isgraph (string[i]));
    string[i+1] = '\0';
    return i;
}

main( argc, argv)
int argc;
char *argv[];
{
    FILE *inp;
    FILE *out;
    char CaseNr[11];
    char VesselID[9];
    int NumVsDam;
    char DrugFactor[11];
    char Date[11];
    char PeriodDay[2];
    char Weather[3];
    char WaterBody[7];
    char Latitude[11];
    char Longitude[11];
    char RoutineLetter[7];
    char TotDamage[11];
    char MSD_Office[4];
    char RiverMP[6];
    char Cause6[8];
    char Unknown[11];
    char WindDir[4];
    char WindSpeed[4];
    char Visibility[6];
    int NumVsInv;
    char Nature1[7];
    char Nature2[7];
    char Nature3[7];
    char Cause1[8];
    char Cause2[8];
    char Cause3[8];
    char Cause4[8];
    char Cause5[8];
    char SeaCond[5];
    char TowConfig[4];
    char VesselName[41];
    char Flag[3];
    int YearConstr;
    char VsIService[5];
    char Seaworthiness;
    char VsIUse[5];
    int VsILgth;
    int GrossTonnage;
    char HullMat[3];
}
```

```

char VsIProp[3];
int HorsePow;
char HullDesign[5];
int VsIDamage;
int CargoDam;
int OtherDam;
int CrewDeath;
int PassDeath;
int OtherDeath;
int CrewInj;
int PassInj;
int OtherInj;
char VsIState[3];
char PersonInCharge[5];
char Society[4];
char Company[11];
char PilotStatus[5];

char infield[11];
char lat, longit;

long int i, test, r = 0;
int Year, Month, Day;

if (argc != 3)
{
    fprintf (stderr, "Usage: %s inputfile outputfile \n", argv[0]);
    exit(-1);
}

if ((inp = fopen (argv[1], "r")) == NULL)
{
    fprintf (stderr, "Error: Cannot open %s for input. \n", argv[1]);
    exit(-1);
}

if ((out = fopen (argv[2], "w")) == NULL)
{
    fprintf (stderr, "Error: Cannot open %s for output. \n", argv[2]);
    exit(-1);
}

do
{
    for (i = 0; i < 60; i++)
    {
        test = fscanf (inp, "%10c", infield);
        infield[10] = '\0';
        Year = 92;
        Month = 6;
        Day = 8 ;
        switch (i)
        {
            case 0 : strncpy (CaseNr, infield, 9);
                    CaseNr[9] = '\0';
                    break;
            case 1 : strncpy (VesselID, infield, 8);
                    if (VesselID[0] == ' ') VesselID[0] = '\0';
                    else VesselID[8] = '\0';
                    break;
            case 4 : sscanf (infield, "%2d%2d%2d", &Year, &Month, &Day);
        }
    }
}

```

```

        sprintf (Date, "%2.2d/%2.2d/%2.2d", Month, Day, Year);
        break;
    case 2 : sscanf (infield, "%d", &NumVsDam);
        break;
    case 19 : sscanf (infield, "%d", &NumVsInv);
        break;
    case 23 : strncpy (Cause1, infield, 7);
        if (Cause1[0] == ' ') Cause1[0] = '\0';
        else Cause1[7] = '\0';
        break;
    case 24 : strncpy (Cause2, infield, 7);
        if (Cause2[0] == ' ') Cause2[0] = '\0';
        else Cause2[7] = '\0';
        break;
    case 25 : strncpy (Cause3, infield, 7);
        if (Cause3[0] == ' ') Cause3[0] = '\0';
        else Cause3[7] = '\0';
        break;
    case 26 : strncpy (Cause4, infield, 7);
        if (Cause4[0] == ' ') Cause4[0] = '\0';
        else Cause4[7] = '\0';
        break;
    case 27 : strncpy (Cause5, infield, 7);
        if (Cause5[0] == ' ') Cause5[0] = '\0';
        else Cause5[7] = '\0';
        break;
    case 14 : strncpy (Cause6, infield, 7);
        if (Cause6[0] == ' ') Cause6[0] = '\0';
        else Cause6[7] = '\0';
        break;
    case 20 : strncpy (Nature1, infield, 6);
        if (Nature1[0] == ' ') Nature1[0] = '\0';
        else Nature1[6] = '\0';
        break;
    case 21 : strncpy (Nature2, infield, 6);
        if (Nature2[0] == ' ') Nature2[0] = '\0';
        else Nature2[6] = '\0';
        break;
    case 22 : strncpy (Nature3, infield, 6);
        if (Nature3[0] == ' ') Nature3[0] = '\0';
        else Nature3[6] = '\0';
        break;
} /* end of switch */

strcpy (infield, " ");
} /* end of for */

fprintf (out, "%d,%d,%s,%s,%s,%d,", r++, 0, CaseNr, Date,
        "06/08/92", NumVsDam);
fprintf (out, "%d\n", NumVsInv );

} /* end of while */
while (EOF != test);

fclose (out);
fclose (inp);
}

#include<stdio.h>
#include<string.h>

```

program location.c

```
int del_blanks (string)
char string[];
{
    int i = 0;

    while (string[i++] != '\0');

    while ((--i >= 0) && !isgraph (string[i]));
    string[i+1] = '\0';
    return i;
}

main( argc, argv)
int argc;
char *argv[];
{
    FILE *inp;
    FILE *out;
    char CaseNr[11];
    char VesselID[9];
    int NumVsDam;
    char DrugFactor[11];
    char Date[6];
    char PeriodDay[2];
    char Weather[3];
    char WaterBody[8];
    char Latitude[11];
    char Longitude[11];
    char RoutineLetter[7];
    char TotDamage[11];
    char MSD_Office[4];
    float RiverMP;
    char Cause6[8];
    char Unknown[11];
    char WindDir[4];
    char WindSpeed[4];
    char Visibility[6];
    int NumVsInv;
    char Nature1[7];
    char Nature2[7];
    char Nature3[7];
    char Cause1[8];
    char Cause2[8];
    char Cause3[8];
    char Cause4[8];
    char Cause5[8];
    char SeaCond[5];
    char TowConfig[4];
    char VesselName[41];
    char Flag[3];
    int YearConstr;
    char VslService[5];
    char Seaworthiness;
    char VslUse[5];
    int VslLgth;
    int GrossTonnage;
    char HullMat[3];
    char VslProp[3];
    int HorsePow;
    char HullDesign[5];
    int VslDamage;
```

```

int CargoDam;
int OtherDam;
int CrewDeath;
int PassDeath;
int OtherDeath;
int CrewInj;
int PassInj;
int OtherInj;
char VsIState[3];
char PersonInCharge[5];
char Society[4];
char Company[11];
char PilotStatus[5];

char infield[11];
char lat, longit;

long int i, test, r = 0;
float latitude = 0, longitude = 0, latmin = 0, longmin = 0;
int latdeg = 0, longdeg = 0;

int district;

if (argc != 3)
{
    fprintf (stderr, "Usage: %s inputfile outputfile1 outputfile2 \n", argv[0]);
    exit(-1);
}

if ((inp = fopen (argv[1], "r")) == NULL)
{
    fprintf (stderr, "Error: Cannot open %s for input. \n", argv[1]);
    exit(-1);
}

if ((out = fopen (argv[2], "w")) == NULL)
{
    fprintf (stderr, "Error: Cannot open %s for output. \n", argv[2]);
    exit(-1);
}

do
{
    district = -1;
    RiverMP = 0.0;
    for (i = 0; i < 60; i++)
    {
        test = fscanf (inp, "%10c", infield);
        infield[10] = '\0';
        switch (i)
        {
            case 0 : strncpy (CaseNr, infield, 9);
                    CaseNr[10] = '\0';
                    break;

            case 7 : strncpy (WaterBody, infield, 7);
                    WaterBody[7] = '\0';
                    sscanf (WaterBody, "%2dZ*s", &district);
                    del_blanks (WaterBody);
                    break;

            case 8 : if (sscanf (infield, "%c %d %f", &lat, &latdeg,

```



```

        &latmin) >= 3)
    {
        latitude = latdeg + latmin / 60.0;
        if (lat == 'S') latitude = -1.0 * latitude;
    }
    else
        latitude = 0;
    break;
case 9 : if (sscanf (infield, "Z1C Zd Zf", &longit, &longdeg,
        &longmin) >= 3)
    {
        longitude = longdeg + longmin / 60.0;
        if (longit == 'W') longitude *= -1.0;
    }
    else
        longitude = 0;
    break;
case 13 : sscanf (infield, "Zf", &RiverMP);
    break;
} /* end of switch */

strcpy (infield, "          ");
} /* end of for */

fprintf (out, "Zd,Zd,Zs,Zs,Zd,Zc,Zc,Zc,Zc,Z.3f,Z.3f,Z.3f\n", r++, 0, CaseNr, WaterBody,
        district, WaterBody[2], WaterBody[3], WaterBody[4], WaterBody[5],
        latitude, longitude, RiverMP);

    latitude = longitude = 0;
    lat = longit = ' ';
} /* end of while */
while (EOF != test);

fclose (out);
fclose (inp);
}

#include<stdio.h>
#include<string.h>

#define RECLGTH 600

int del_blanks (string)
char string[];
{
    int i = 0;

    while (string[i++] != '\0');

    while ((--i >= 0) && !isgraph (string[i]));
    string[i+1] = '\0';
    return i;
}

main( argc, argv)
int argc;
char *argv[];
{
    FILE *inp;
    FILE *out;
    char CaseNr[11];

```

vessel.c

```

char VesselID[9];
int NumVsDam;
char DrugFactor[11];
int Date;
char PeriodDay;
char Weather[3];
char WaterBody[7];
char Latitude[11];
char Longitude[11];
char RoutineLetter[7];
char TotDamage[11];
char MSD_Office[4];
char RiverMP[6];
char Cause6[8];
char Unknown[11];
char WindDir[4];
char WindSpeed[4];
char Visibility[5];
int NumVsInv;
char Nature1[7];
char Nature2[7];
char Nature3[7];
char Cause1[8];
char Cause2[8];
char Cause3[8];
char Cause4[8];
char Cause5[8];
char SeaCond[5];
char TowConfig[4];
char VesselName[41];
char Flag[3];
int YearConstr;
char VslService[5];
char Seaworthiness;
char VslUse[5];
int VslLgth;
int GrossTonnage;
char HullMat[3];
char VslProp[3];
int HorsePow;
char HullDesign[5];
int VslDamage;
int CargoDam;
int OtherDam;
int CrewDeath;
int PassDeath;
int OtherDeath;
int CrewInj;
int PassInj;
int OtherInj;
char VslState[3];
char PersonInCharge[5];
char Society[7];
char Company[11];
char PilotStatus[5];

char inField[11];

long int i, test, l = 0;
long int NumCoord = 0;
long int NumWatBod = 0;

```

```

long int NoPos = 0;
char *comma;

if (argc != 3)
{
    fprintf (stderr, "Usage: %s inputfile outputfile \n", argv[0]);
    exit(-1);
}

if ((inp = fopen (argv[1], "r")) == NULL)
{
    fprintf (stderr, "Error: Cannot open %s for input. \n", argv[1]);
    exit(-1);
}

if ((out = fopen (argv[2], "w")) == NULL)
{
    fprintf (stderr, "Error: Cannot open %s for output. \n", argv[2]);
    exit(-1);
}

i = 0;

do
{
    for (i = 0; i < 60; i++)
    {
        test = fscanf (inp, "%10c", infield);
        infield[10] = '\0';
        switch (i)
        {
            case 23 : strncpy (Cause1, infield, 7);
                    if (Cause1[0] == ' ') Cause1[0] = '\0';
                    else Cause1[7] = '\0';
                    break;
            case 24 : strncpy (Cause2, infield, 7);
                    if (Cause2[0] == ' ') Cause2[0] = '\0';
                    else Cause2[7] = '\0';
                    break;
            case 25 : strncpy (Cause3, infield, 7);
                    if (Cause3[0] == ' ') Cause3[0] = '\0';
                    else Cause3[7] = '\0';
                    break;
            case 26 : strncpy (Cause4, infield, 7);
                    if (Cause4[0] == ' ') Cause4[0] = '\0';
                    else Cause4[7] = '\0';
                    break;
            case 27 : strncpy (Cause5, infield, 7);
                    if (Cause5[0] == ' ') Cause5[0] = '\0';
                    else Cause5[7] = '\0';
                    break;
            case 14 : strncpy (Cause6, infield, 7);
                    if (Cause6[0] == ' ') Cause6[0] = '\0';
                    else Cause6[7] = '\0';
                    break;
            case 20 : strncpy (Nature1, infield, 6);
                    if (Nature1[0] == ' ') Nature1[0] = '\0';
                    else Nature1[6] = '\0';
                    break;
            case 21 : strncpy (Nature2, infield, 6);
                    if (Nature2[0] == ' ') Nature2[0] = '\0';

```

```

        else Nature2[6] = '\0';
        break;
case 22 : strncpy (Nature3, infield, 6);
        if (Nature3[0] == ' ') Nature3[0] = '\0';
        else Nature3[6] = '\0';
        break;

case 0 : strncpy (CaseNr, infield, 9);
        break;
case 1: if (sscanf (infield, "%s", VesselID) == 0)
        strcpy (VesselID, " ");
        del_blanks (VesselID);
        break;
case 29 : if (sscanf (infield, "%s", TowConfig) == 0)
        strcpy (TowConfig, " ");
        del_blanks (TowConfig);
        break;
case 30 : strcpy (VesselName, infield);
        break;
case 31 : if (sscanf (infield, "%s", Flag) == 0)
        strcpy (Flag, " ");
        del_blanks (Flag);
        break;
case 32 : if (sscanf (infield, "%d", &YearConstr) == 0)
        YearConstr = 0;
        break;
case 33 : if (sscanf (infield, "%s", VslService) == 0)
        strcpy (VslService, " ");
        del_blanks (VslService);
        break;
case 34 : Seaworthiness = infield[0];
        if (Seaworthiness == ' ') Seaworthiness = '\0';
        break;
case 35 : if (sscanf (infield, "%s", VslUse) == 0)
        strcpy (VslUse, " ");
        del_blanks (VslUse);
        break;
case 36 : if (sscanf (infield, "%d", &VslLgth) == 0)
        VslLgth = 0;
        break;
case 37 : if (sscanf (infield, "%d", &GrossTonnage) == 0)
        GrossTonnage = 0;
        break;
case 38 : if (sscanf (infield, "%s", HullMat) == 0)
        strcpy (HullMat, " ");
        del_blanks (HullMat);
        break;
case 39 : if (sscanf (infield, "%s", VslProp) == 0)
        strcpy (VslProp, " ");
        del_blanks (VslProp);
        break;
case 40 : if (sscanf (infield, "%d", &HorsePow) == 0)
        HorsePow = 0;
        break;
case 41 : if (sscanf (infield, "%s", HullDesign) == 0)
        strcpy (HullDesign, " ");
        del_blanks (HullDesign);
        break;
case 51 : if (sscanf (infield, "%s", VslState) == 0)
        strcpy (VslState, " ");
        del_blanks (VslState);

```

```

        break;
    case 52 : if (sscanf (infield, "Zs", PersonInCharge) ==0)
                strcpy (PersonInCharge, " ");
                del_blanks (PersonInCharge);
                break;
    case 53 : if (sscanf (infield, "Zs", Society) ==0)
                strcpy (Society, " ");
                del_blanks (Society);
                break;
    case 54 : strcpy (Company, infield);
                del_blanks (Company);
                break;
    case 55 : if (sscanf (infield, "Zs", PilotStatus) ==0)
                strcpy (PilotStatus, " ");
                del_blanks (PilotStatus);
                break;
    case 56 : strcat (VesselName, infield);
                break;
    case 57 : strcat (VesselName, infield);
                break;
    case 58 : strcat (VesselName, infield);
                break;
} /* end of switch */

```

```

strcpy (infield, " ");
} /* end of for */

```

```

del_blanks (VesselName);
if ((comma = strchr (VesselName, 44)) != NULL)
    *comma = 32;
if ((comma = strchr (Company, 44)) != NULL)
    *comma = 32;

fprintf (out, "%d,%d,%s,%s", l++, 0, CaseNr, VesselID);

fprintf (out, "%s,%s,%d,%s,%s,%d,%d,%s,%s,%d,%s,%s,%s,%s,%s,%s",
        VesselName, Flag, YearConstr, VslService,
        VslUse, VslLgth,
        GrossTonnage, HullMat, VslProp, HorsePow, HullDesign,
        VslState, TowConfig,
        PersonInCharge, Society, Company, PilotStatus);
fprintf (out, "%c,%s,%s,%s,%s", Seaworthiness, Cause1, Cause2,
        Cause3, Cause4);
fprintf (out, "%s,%s,%s,%s,%s\n", Cause5, Cause6, Nature1,
        Nature2, Nature3);

```

```

} /* end of while */
while (EOF != test);

```

```

fclose (out);
fclose (inp);

```

```

}

```

```

#include<stdio.h>
#include<string.h>

```

```

int del_blanks (string)
char string[];

```

```

{
    int i = 0;

```

program weather.c

```

while (string[i++] != '\0');

while ((--i >= 0) && !isgraph (string[i]));
string[i+1] = '\0';
return i;
}

```

```

main( argc, argv)
int argc;
char *argv[];
{
FILE *inp;
FILE *out;
char CaseNr[11];
char VesselID[9];
int NumVsDam;
char DrugFactor[11];
char Date[6];
char PeriodDay[2];
char Weather[3];
char WaterBody[7];
char Latitude[11];
char Longitude[11];
char RoutineLetter[7];
char TotDamage[11];
char MSO_Office[4];
char RiverMP[6];
char Cause6[8];
char Unknown[11];
char WindDir[4];
char WindSpeed[4];
char Visibility[6];
int NumVsInv;
char Nature1[7];
char Nature2[7];
char Nature3[7];
char Cause1[8];
char Cause2[8];
char Cause3[8];
char Cause4[8];
char Cause5[8];
char SeaCond[5];
char TowConfig[4];
char VesselName[41];
char Flag[3];
int YearConstr;
char VslService[5];
char Seaworthiness;
char VslUse[5];
int VslLgth;
int GrossTonnage;
char HullMat[3];
char VslProp[3];
int HorsePow;
char HullDesign[5];
int VslDamage;
int CargoDam;
int OtherDam;
int CrewDeath;
int PassDeath;
int OtherDeath;

```

```

int CrewInj;
int PassInj;
int OtherInj;
char VsiState[3];
char PersonInCharge[5];
char Society[4];
char Company[11];
char PilotStatus[5];

char infield[11];

long int i, test, r = 0 ;
long int NumCoord = 0;
long int NumWatBod = 0;
long int NoPos = 0;

if (argc != 3)
{
    fprintf (stderr, "Usage: %s inputfile outputfile \n", argv[0]);
    exit(-1);
}

if ((inp = fopen (argv[1], "r")) == NULL)
{
    fprintf (stderr, "Error: Cannot open %s for input. \n", argv[1]);
    exit(-1);
}

if ((out = fopen (argv[2], "w")) == NULL)
{
    fprintf (stderr, "Error: Cannot open %s for output. \n", argv[2]);
    exit(-1);
}

do
{
    for (i = 0; i < 60; i++)
    {
        test = fscanf (inp, "%10c", infield);
        infield[10] = '\0';
        switch (i)
        {
            case 0 : strncpy (CaseNr, infield, 9);
                    break;
            case 5 : strncpy (PeriodDay, infield, 1);
                    PeriodDay[1] = '\0';
                    del_blanks (PeriodDay);
                    break;
            case 6 : strncpy (Weather, infield, 2);
                    del_blanks (Weather);
                    break;
            case 16 : strncpy (WindDir, infield, 3);
                    del_blanks (WindDir);
                    break;
            case 17 : strncpy (WindSpeed, infield, 3);
                    del_blanks (WindSpeed);
                    break;
            case 18 : strncpy (Visibility, infield, 5);
                    del_blanks (Visibility);
                    break;
            case 28 : strncpy (SeaCond, infield, 4);

```

```

                del_blanks (SeaCond);
                break;
            } /* end of switch */

            strcpy (infield, "          ");
        } /* end of for */

        fprintf (out, "%d,%d,%s,%s,%s,%s,%s,%s,%s\n", r++, 0,
                CaseNr, PeriodDay, Weather, WindDir, WindSpeed,
                Visibility, SeaCond);
    } /* end of while */
    while (EOF != test);

    fclose (out);
    fclose (inp);
}

```



```

#include <stdio.h>
#include <string.h>

main (argc, argv)
int argc;
char *argv[];
{
    int count = 0;
    char c;
    FILE *inp, *out;

    if (argc != 3)
    {
        fprintf (stderr, "\nUsage: Zs inputfile outputfile.", argv[0]);
        exit (-1);
    }

    if ((inp = fopen (argv[1], "r")) == NULL)
    {
        fprintf (stderr, "\nError: Cannot open Zs for input.", argv[1]);
        exit (-1);
    }

    if ((out = fopen (argv[2], "w")) == NULL)
    {
        fprintf (stderr, "\nError: Cannot open Zs for output.", argv[2]);
        exit (-1);
    }

    while ((c = fgetc (inp)) != EOF)
    {
        if (isctrl(c) && c != '\n')
        {
            fputc (' ', out);
            count++;
        }
        else
            fputc (c, out);
    }

    printf ("\n%d characters replaced with blanks !!", count);
    fclose (inp);
    fclose (out);
    return 0;
}

```

```

#include<stdio.h>
#include<string.h>

#define NUMREC 64980
#define RECLGTH 600

main (argc, argv)
int argc;
char *argv[];
{
    FILE *inp;
    FILE *out;

    char dummy1;
    char dummy2[RECLGTH];
    char north;
    char west;

    long int i = 0;
    long int k = 0;
    int latdeg;
    int longdeg;
    int j;

    float latmin;
    float longmin;
    float latitude, longitude;

    if(argc != 3)
    {
        fprintf (stderr, "Usage: Zs inputfile outputfile\n", argv[0]);
        exit(-1);
    }

    if ((inp = fopen(argv[1], "r")) == NULL)
    {
        fprintf (stderr, "Error: Cannot open Zs for input.\n", argv[1]);
        exit(-1);
    }

    if ((out = fopen(argv[2], "w")) == NULL)
    {
        fprintf (stderr, "Error: Cannot open Zs for output.\n", argv[2]);
        exit(-1);
    }

    while (i++ < NUMREC)
    {
        for (j = 0; j < 80; j++) dummy1 = getc (inp);
        fscanf (inp, "%1c%1d%1f%1c%1d%1f", &west, &latdeg, &latmin,
                &north, &longdeg, &longmin);

        latitude = latdeg + latmin / 60.0;
        longitude = longdeg + longmin / 60.0;

        for (j = 0; j < 500; j++) dummy1 = getc (inp);
        if ((north != ' ') && (west != ' '))
        {
            if (north == 'S')
            {

```

```
        latitude *= -1.0;
    }
    if (west == 'W')
    {
        longitude *= -1.0;
    }
    fprintf (out, " Z7.1f Z7.1f\n", longitude, latitude);
    k++;
    longitude = latitude = 0.0;
}
}

printf (" Zd records written!!\n", k);

fclose (inp);
fclose (out);
}
```

```

        else return -1;
    } /* end while */
    return c;
} /* end getfield */

```

```

int getlocation (row, inp)
struct location *row;
FILE *inp;
{
    int i = 0;
    char c;
    char *infield = " ";

    while ((c = getfield (infield, inp)) != EOF)
    {
        switch(i++)
        {
            case 0 : sscanf (infield, "%d", &(row->rec_nr));
                      break;
            case 1 : sscanf (infield, "%d", &(row->acc_nr));
                      break;
            case 2 : strcpy (row->case_nr, infield);
                      break;
            case 3 : strcpy (row->wb, infield);
                      break;
            case 4 : sscanf (infield, "%d", &(row->district));
                      break;
            case 5 : row->ocean = infield[0];
                      break;
            case 6 : row->w3 = infield[0];
                      break;
            case 7 : row->w4 = infield[0];
                      break;
            case 8 : row->w5 = infield[0];
                      break;
            case 9 : sscanf (infield, "%f", &(row->latitude));
                      break;
            case 11 : sscanf (infield, "%f", &(row->rmp));
                      break;
            case 10 : sscanf (infield, "%f", &(row->longitude));
                      break;
        } /* end switch */
        if (c == '\n') return c;
    } /* end while */
    return c;
} /* end getlocation */

```

```

main (argc, argv)
int argc;
char *argv[];
{
    struct lutable lut[38300];
    register struct location row;
    FILE *inp, *out;
    register long int i = 0, numkeys = 0;
    int notfound;

    if (argc != 3)
    {

```

```

#include<stdio.h>
#include<string.h>

struct lutable
{
    long int acc_nr;
    char case_nr[10];
};

struct location
{
    long int rec_nr;
    long int acc_nr;
    char case_nr[10];
    char wb[8];
    int district;
    char ocean;
    char w3;
    char w4;
    char w5;
    float latitude;
    float longitude;
    float rmp;
};

int del_blanks (string)
char string[];
{
    int i = 0;

    while (string[i++] != '\0');

    while ((--i >= 0) && !isgraph (string[i]));
    string[i+1] = '\0';
    return i;
};

int getfield (infield, inp)
char infield[];
FILE *inp;
{
    int end_field = 0;
    int i = 0;
    char c;

    strcpy (infield, " ");
    while (end_field == 0)
    {
        if ((infield[i] = fgetc(inp)) != EOF)
        {
            if (infield[i] == ',' || infield[i] == '\n')
            {
                c = infield[i];
                infield[i] = '\0';
                end_field = 1;
            }
            else
                i++;
        }
    }
}

```

```

    fprintf (stderr, "Usage: %s inputfile outputfile\n", argv[0]);
    exit(-1);
}

if ((inp = fopen (argv[1], "r")) == NULL)
{
    fprintf (stderr, "Error: Cannot open %s for input. \n", argv[1]);
    exit(-1);
}

if ((out = fopen (argv[2], "w")) == NULL)
{
    fprintf (stderr, "Error: Cannot open %s for output. \n", argv[2]);
    exit(-1);
}

numkeys = 0;
puts ("          BUSY !!");
while (getlocation (&row, inp) != EOF)
{
    printf ("          -> %d <-\r", row.rec_nr);
    notfound = 1;
    i = 0;
    while (i++ < numkeys && notfound != 0)
    {
        if (strcmp (row.case_nr, lut[i].case_nr) == 0)
            if (row.acc_nr == lut[i].acc_nr)
            {
                notfound = 0;
            }
    }

    if (notfound != 0)
    {
        lut[numkeys].acc_nr = row.acc_nr;
        strcpy (lut[numkeys].case_nr, row.case_nr);
        fprintf (out, "%d,%d,%s,%s,%d,%c,%c,%c,%c,%d,%d,%d,%d,%d,%d,%d,%d\n",
            row.rec_nr, row.acc_nr,
            row.case_nr, row.wb, row.district, row.ocean, row.w3, row.w4,
            row.w5, row.latitude, row.longitude * -1.0, row.rmp);
        numkeys++;
    }
}

fclose (inp);
fclose (out);
return 0;
}

```

```

#include<stdio.h>
#include<string.h>

struct lutable
{
    long int acc_nr;
    char case_nr[10];
};

struct accident
{
    long int rec_nr;
    long int acc_nr;
    char case_nr[10];
    char dateacc[9];
    char data_entry[9];
    int numvsldam;
    int numvslinv;
};

int del_blanks (string)
char string[];
{
    int i = 0;

    while (string[i++] != '\0');

    while ((--i >= 0) && !isgraph (string[i]));
    string[i+1] = '\0';
    return i;
};

int getfield (infield, inp)
char infield[];
FILE *inp;
{
    int end_field = 0;
    int i = 0;
    char c;

    strcpy (infield, " ");
    while (end_field == 0)
    {
        if ((infield[i] = fgetc(inp)) != EOF)
        {
            if (infield[i] == ',' || infield[i] == '\n')
            {
                c = infield[i];
                infield[i] = '\0';
                end_field = 1;
            }
            else
                i++;
        }
        else return -1;
    } /* end while */
    return c;
} /* end getfield */

```

```

int getaccident (row, inp)
struct accident *row;
FILE *inp;
{
    int i = 0;
    char c;
    char *infield = "      ";

    while ((c = getfield (infield, inp)) != EOF)
    {
        switch(i++)
        {
            case 0 : sscanf (infield, "%d", &(row->rec_nr));
                    break;
            case 1 : sscanf (infield, "%d", &(row->acc_nr));
                    break;
            case 2 : strcpy (row->case_nr, infield);
                    break;
            case 3 : strcpy (row->dateacc, infield);
                    break;
            case 4 : strcpy (row->data_entry, infield );
                    break;
            case 5 : sscanf (infield, "%d", &(row->numvsldam));
                    break;
            case 6 : sscanf (infield, "%d", &(row->numvslinv));
                    break;
        } /* end switch */
        if (c == '\n') return c;
    } /* end while */
    return c;
} /* end getlocation */

main (argc, argv)
int argc;
char *argv[];
{
    struct lutable lut[38300];
    register struct accident row;
    FILE *inp, *out;
    register long int i = 0, numkeys = 0;
    int notfound;
    int day, month, year;

    if (argc != 3)
    {
        fprintf (stderr, "Usage: %s inputfile outputfile\n", argv[0]);
        exit(-1);
    }

    if ((inp = fopen (argv[1], "r")) == NULL)
    {
        fprintf (stderr, "Error: Cannot open %s for input. \n", argv[1]);
        exit(-1);
    }

    if ((out = fopen (argv[2], "w")) == NULL)
    {
        fprintf (stderr, "Error: Cannot open %s for output. \n", argv[2]);
        exit(-1);
    }
}

```



```
while (getaccident (&row, inp) != EOF)
{
    printf ("                -> Zd <-\r", row.rec_nr);
    sscanf (row.dateacc, "%2d%*c%2d%*c%2d", &month, &day, &year);
    fprintf (out, "%d,%d,%s,%2.2d/%2.2d/%2.2d,06/08/92,%d,%d\n",
            row.rec_nr, row.acc_nr, row.case_nr, day, month, year,
            row.numvsldam, row.numvslinv);
}

fclose (inp);
fclose (out);
return 0;
}
```

```

#include<stdio.h>
#include<string.h>

struct lutable
{
    long int rec_nr;
    long int acc_nr;
    char case_nr[10];
};

struct vessel
{
    long int rec_nr;
    long int acc_nr;
    char case_nr[10];
    char vessel_id[9];
    char vesselname[41];
    char flag[3];
    int yearcons;
    char vsIService[5];
    char vsIuse[5];
    int length;
    int tonnage;
    char material[3];
    char propulsion[3];
    int hp;
    char design[5];
    char vsIstate[3];
    char towconfig[4];
    char persincharge[5];
    char society[7];
    char company[11];
    char pilotstatus[5];
    char seaworthy;
    char cause1[8];
    char cause2[8];
    char cause3[8];
    char cause4[8];
    char cause5[8];
    char cause6[8];
    char nature1[8];
    char nature2[8];
    char nature3[8];
};

int del_blanks (string)
char string[];
{
    int i = 0;

    while (string[i++] != '\0');

    while ((--i >= 0) && !isgraph (string[i]));
    string[i+1] = '\0';
    return i;
};

int getfield (infield, inp)
char infield[];

```

```

FILE *inp;
{
    int end_field = 0;
    int i = 0;
    char c;

    while (end_field == 0)
    {
        if ((infield[i] = fgetc(inp)) != EOF)
        {
            if (infield[i] == ',' || infield[i] == '\n')
            {
                c = infield[i];
                infield[i] = '\0';
                end_field = 1;
            }
            else
                i++;
        }
        else return -1;
    } /* end while */
    return c;
} /* end getfield */

int getvessel (row, inp)
struct vessel *row;
FILE *inp;
{
    int i = 0;
    char c;
    char *infield = ""

    while ((c = getfield (infield, inp)) != EOF)
    {
        switch(i++)
        {
            case 0 : sscanf (infield, "%d", &(row->rec_nr));
                    break;
            case 1 : sscanf (infield, "%d", &(row->acc_nr));
                    break;
            case 2 : strcpy (row->case_nr, infield);
                    break;
            case 3 : strcpy (row->vessel_id, infield);
                    del_blanks (row->vessel_id);
                    break;
            case 4 : strcpy (row->vesselname, infield);
                    del_blanks (row->vesselname);
                    break;
            case 5 : strcpy (row->flag, infield);
                    del_blanks (row->flag);
                    break;
            case 6 : sscanf (infield, "%d", &(row->yearcons));
                    break;
            case 7 : strcpy (row->vslservice, infield);
                    del_blanks (row->vslservice);
                    break;
            case 8 : strcpy (row->vsluse, infield);
                    del_blanks (row->vsluse);
                    break;
            case 9 : sscanf (infield, "%d", &(row->length));

```

```

        break;
    case 10 : sscanf (infield, "%d", &(row->tonnage));
              break;
    case 11 : strcpy (row->material, infield);
              del_blanks (row->material);
              break;
    case 12 : strcpy (row->propulsion, infield);
              del_blanks (row->propulsion);
              break;
    case 13 : sscanf (infield, "%d", &(row->hp));
              break;
    case 14 : strcpy (row->design, infield);
              del_blanks (row->design);
              break;
    case 15 : strcpy (row->vsistate, infield);
              del_blanks (row->vsistate);
              break;
    case 16 : strcpy (row->towconfig, infield);
              del_blanks (row->towconfig);
              break;
    case 17 : strcpy (row->persincharge, infield);
              del_blanks (row->persincharge);
              break;
    case 18 : strcpy (row->society, infield);
              break;
    case 19 : strcpy (row->company, infield);
              break;
    case 20 : strcpy (row->pilotstatus, infield);
              break;
    case 21 : row->seaworthy = infield[0];
              break;
    case 22 : strcpy (row->cause1, infield);
              break;
    case 23 : strcpy (row->cause2, infield);
              break;
    case 24 : strcpy (row->cause3, infield);
              break;
    case 25 : strcpy (row->cause4, infield);
              break;
    case 26 : strcpy (row->cause5, infield);
              break;
    case 27 : strcpy (row->cause6, infield);
              break;
    case 28 : strcpy (row->nature1, infield);
              break;
    case 29 : strcpy (row->nature2, infield);
              break;
    case 30 : strcpy (row->nature3, infield);
              break;
    } /* end switch */
    if (c == '\n') return c;
} /* end while */
return c;
} /* end getweather */

int getlut (pos, inp)
struct lutable *pos;
FILE *inp;
{
    char *fd = " ";
};

```

```

int i = 0;
char c;

while ((c= getfield (fd, inp)) != EOF)
{
    switch (i++)
    {
        case 0 : sscanf (fd, "%d", &(pos->rec_nr));
                break;
        case 1 : sscanf (fd, "%d", &(pos->acc_nr));
                break;
        case 2 : strcpy (pos->case_nr, fd);
                del_blanks (pos->case_nr);
                break;
    }
    if (c == '\n') return c;
} /* end while */
return c;
} /* end getpos */

main (argc, argv)
int argc;
char *argv[];
{
    struct lutable lut[64980];
    register struct vessel row;
    FILE *inp, *out1;
    FILE *inp2, *out2;
    register long int i = 0, num_pos = 0, found;

    if (argc != 4)
    {
        fprintf (stderr, "Usage: %s inputfile1 inputfile2 outputfile\n", argv[0]);
        exit(-1);
    }

    if ((inp = fopen (argv[1], "r")) == NULL)
    {
        fprintf (stderr, "Error: Cannot open %s for input. \n", argv[1]);
        exit(-1);
    }

    if ((inp2 = fopen (argv[2], "r")) == NULL)
    {
        fprintf (stderr, "Error: Cannot open %s for input. \n", argv[2]);
        exit(-1);
    }

    if ((out1 = fopen (argv[3], "w")) == NULL)
    {
        fprintf (stderr, "Error: Cannot open %s for output. \n", argv[3]);
        exit (-1);
    }

    while (getlut (&lut[num_pos++], inp) != EOF);
    printf ("\n          lut.loc loaded!!\n");

    while (getvessel (&row, inp2) != EOF)
    {
        if (lut[i].rec_nr == row.rec_nr)

```

```

    {
row.acc_nr = lut[i++].acc_nr;
fprintf (out1, "%d,%d,%s,%s,%s,%s,%d,%s,%s,%d,%d,%s,%s,%d,%s,%s,%s,%s,%s,%s,%c,%s,%s,%s,%s,%s,%s,%s,%s\n",
row.rec_nr, row.acc_nr,
row.case_nr, row.vessel_id, row.vesselname, row.flag,
row.yearcons, row.vsiservice, row.vsluse, row.length,
row.tonnage, row.material, row.propulsion, row.hp,
row.design, row.vslstate, row.towconfig, row.persincharge,
row.society, row.company, row.pilotstatus, row.seaworthy,
row.cause1, row.cause2, row.cause3, row.cause4, row.cause5,
row.cause6, row.nature1, row.nature2, row.nature3);
    }
}

fclose (inp2);
fclose (out1);
fclose (inp);
system ("logout");
return 0;
}

```

```

#include<stdio.h>
#include<string.h>

#define SIZE 700000

main (argc, argv)
int argc;
char *argv[];
{
    FILE *inp, *out[20];
    long int i = 0, j = 0, k = 0;
    char c, outfile[255];

    if (argc != 3)
    {
        fprintf (stderr, "\n\nUsage: %s inputfile outputfile !\n", argv[0]);
        exit (-1);
    }

    if ((inp = fopen (argv[1], "r")) == NULL)
    {
        fprintf (stderr, "\n\nERROR: Cannot open %s for input !", argv[1]);
        exit(-1);
    }

    strcpy (outfile, argv[2]);
    while (outfile[i++] != '\0');

    outfile[i-1] = k+49;
    outfile[i] = '\0';

    if ((out[k++] = fopen (outfile, "w")) == NULL)
    {
        fprintf (stderr, "\n\nERROR: Cannot open %s for output !",
                outfile);
        exit(-1);
    }

    while ((c = fgetc (inp)) != EOF)
    {
        if (fputc (c, out[k-1]) == '\n' && j >= SIZE)
        {
            fclose (out[k-1]);
            j = 0;
            outfile[i-1] = k+49;
            outfile[i] = '\0';
            if ((out[k++] = fopen (outfile, "w")) == NULL)
            {
                fprintf (stderr, "\n\nERROR: Cannot open %s for output !",
                        outfile);
                exit(-1);
            }
        }
        j++;
    }

    for (i = 0; i < k; i++)
        fclose (out[i]);
    fclose (inp);
    return (0);
}

```

```

#include <stdio.h>

main (argc, argv)
int argc;
char *argv[];
{
    FILE *inp[10], *out;
    char c;
    int i = 0;

    if (argc < 3)
    {
        fprintf (stderr, "Usage: Zs inputfile [inputfile] outputfile.\n",
            argv[0]);
        exit (-1);
    }

    for (i = 0; i < argc-2; i++)
    {
        if ((inp[i] = fopen (argv[i+1], "r")) == NULL)
        {
            fprintf (stderr, "Error: Cannot open Zs for input.\n", argv[i+1]);
            exit (-1);
        }
    }

    if ((out = fopen (argv[argc-1], "w")) == NULL)
    {
        fprintf (stderr, "Error: Cannot open Zs for output.\n", argv[argc-1]);
        exit (-1);
    }

    for (i = 0; i < argc-2; i++)
    {
        while ((c = fgetc(inp[i])) != EOF)
            fputc (c, out);
        fclose (inp[i]);
    }

    fclose (out);

    return 0;
}

```



```

#include<stdio.h>
#include<string.h>

struct position
{
    char wb[8];
    char case_nr[10];
    float lat;
    float longit;
    float rmp;
};

struct location
{
    long int recordnr;
    long int acc_nr;
    char case_nr[10];
    char wb[8];
    int dis;
    char ocean;
    char w3;
    char w4;
    char w5;
    float lat;
    float longit;
    float rmp;
};

int del_blanks (string)
char string[];
{
    int i = 0;

    while (string[i++] != '\0');

    while ((--i >= 0) && !isgraph (string[i]));
    string[i+1] = '\0';
    return i;
};

int getfield (infield, inp)
char infield[];
FILE *inp;
{
    int end_field = 0;
    int i = 0;
    char c;

    strcpy (infield, " ");
    while (end_field == 0)
    {
        if ((infield[i] = fgetc(inp)) != EOF)
        {
            if (infield[i] == '.' || infield[i] == '\n')
            {
                c = infield[i];
                infield[i] = '\0';
                end_field = 1;
            }
        }
    }
}

```

```

        else
            i++;
        }
        else return -1;
    } /* end while */
    return c;
} /* end getfield */

```

```

int getlocation (row, inp)
struct location *row;
FILE *inp;
{
    int i = 0;
    char c;
    char *infield = " ";

    while ((c = getfield (infield, inp)) != EOF)
    {
        switch(i++)
        {
            case 0 : sscanf (infield, "%d", &(row->recordnr));
                    break;
            case 1 : sscanf (infield, "%d", &(row->acc_nr));
                    break;
            case 2 : strcpy (row->case_nr, infield);
                    break;
            case 3 : strcpy (row->wb, infield);
                    del_blanks (row->wb);
                    break;
            case 4 : sscanf (infield, "%d", &(row->dis));
                    break;
            case 5 : row->ocean = infield[0];
                    break;
            case 6 : row->w3 = infield[0];
                    break;
            case 7 : row->w4 = infield[0];
                    break;
            case 8 : row->w5 = infield[0];
                    break;
            case 9 : sscanf (infield, "%f", &(row->lat));
                    break;
            case 10 : sscanf (infield, "%f", &(row->longit));
                    break;
            case 11 : sscanf (infield, "%f", &(row->rmp));
                    break;
        } /* end switch */
        if (c == '\n') return c;
    } /* end while */
    return c;
} /* end getlocation */

```

```

int getpos (pos, inp)
struct position *pos;
FILE *inp;
{
    char fd[13];
    int i = 0;
    char c;

    while ((c = getfield (fd, inp)) != EOF)

```

```

    {
    switch (i++)
    {
    case 0 : strncpy (pos->wb, fd, 8);
            del_blanks (pos->wb);
            break;
    case 1 : strncpy (pos->case_nr, fd, 10);
            del_blanks (pos->case_nr);
            break;
    case 2 : sscanf (fd, "%f", &(pos->lat));
            break;
    case 3 : sscanf (fd, "%f", &(pos->longit));
            break;
    case 4 : sscanf (fd, "%f", &(pos->rmp));
            break;
    }
    if (c == '\n') return c;
    } /* end while */
return c;
} /* end getpos */

main (argc, argv)
int argc;
char *argv[];
{
    struct position pos[40000];
    register struct location row;
    FILE *inp, *out1;
    FILE *inp2, *out2;
    register long int i = 0, num_pos = 0, found;

    if (argc != 4)
    {
        fprintf (stderr, "Usage: Zs inputfile1 inputfile2 outputfile\n", argv[0]);
        exit(-1);
    }

    if ((inp = fopen (argv[1], "r")) == NULL)
    {
        fprintf (stderr, "Error: Cannot open Zs for input. \n", argv[1]);
        exit(-1);
    }

    if ((inp2 = fopen (argv[2], "r")) == NULL)
    {
        fprintf (stderr, "Error: Cannot open Zs for input. \n", argv[2]);
        exit(-1);
    }

    if ((out1 = fopen (argv[3], "w")) == NULL)
    {
        fprintf (stderr, "Error: Cannot open Zs for output. \n", argv[3]);
        exit (-1);
    }

    if ((out2 = fopen ("lut.loc", "w")) == NULL)
    {
        fprintf (stderr, "Error: Cannot open lut.loc for output. \n");
        exit(-1);
    }
}

```

```

while (getpos (&(pos[num_pos]), inp2) != EOF)
{
    pos[num_pos].longit += -1.0;
    num_pos++;
}

while (getlocation (&row, inp) != EOF)
{
    printf ("                                -> Zd <-\r", row.recordnr);
    found = 0;
    i = 0;
    while (i++ < num_pos && found == 0)
    {
        if (strcmp (pos[i].case_nr, row.case_nr) == 0)
        {
            if (strcmp (pos[i].wb, row.wb) == 0)
            {
                if (pos[i].lat == row.lat)
                {
                    if (pos[i].longit == row.longit)
                    if (pos[i].rmp == row.rmp)
                    {
                        found = 1;
                        fprintf (out1, "Zd,Zd,Zs,Zs,Zd,Zc,Zc,Zc,Zc,Z.3f,Z.3f,Z.3f\n",
                            row.recordnr, i, row.case_nr, row.wb, row.dis,
                            row.ocean, row.w3, row.w4, row.w5, row.lat, row.longit,
                            row.rmp);

                        fprintf (out2, "Zd,Zd,Zs\n", row.recordnr, i,
                            row.case_nr);
                    } /* end if */
                }
            }
        }
    } /* end while */
} /* end while */

fclose (inp2);
fclose (out2);
fclose (out1);
fclose (inp);
}

```