Environmental Compliance: Acquisition, Storage, and Analysis of Waste Oil Data

An Interactive Qualifying Project
Submitted to the faculty of
Worcester Polytechnic Institute
in partial fulfillment of the requirements for the
Degree of Bachelor of Science

Submitted by:

Patrick Brodeur
Renée Lanza
Elizabeth Morris
Edward Osowski

Sponsoring Agency:

United States Coast Guard
Office of Vessel Activities (COMDT CG-543), in conjunction with the
Office of Investigations and Casualty Analysis (COMDT CG-545)

Submitted to:

Project Advisor: Mustapha Fofana, WPI Professor
Project Co-advisor: John Orr, WPI Professor
On-Site Liaison: LCDR Chaning Burgess, United States Coast Guard (CG-5432)

Date: 15 December 2011

Abstract

In 2007, the United States Coast Guard in conjunction with the Department of Justice began implementing a Special Waste Oil Monitoring System on vessels that have repeatedly violated environmental regulations regarding oil-waste. Until now, the process for analyzing this data has been both slow and laborious. To address this, a system for data acquisition, storage and analysis was created. If implemented, this system will reduce analysis time from three months down to two weeks.

Executive Summary

The improper disposal of waste oil from vessels significantly contributes to marine pollution. Typically this waste oil is mixed with water, which can then either remain on the vessel until it reaches port or be treated on the vessel and then be expelled into the ocean, provided that it abides by the 15 parts-per-million legal maximum concentration set forth by MARPOL (International Convention for the Prevention of Pollution from Ships). Some vessels illegally dump waste oil into the ocean to avoid disposal costs, which can easily add up to tens of thousands of dollars in a year for a medium-sized vessel. If evidence of illegal activities is found while a vessel is in US waterways, the United States Department of Justice and the United States Coast Guard will investigate the allegations and after a trial may mandate the implementation of a Special Waste Oil Monitoring System (SWOMS) that records waste-stream data about waste oil discharge. There are currently two shipping companies who have agreed to develop and implement a SWOMS aboard some of their vessels.

All ships are required to keep manual records of all oil storage, discharges, and incinerations in an oil record book (ORB) on the ship. Finding inconsistencies in these records is one way that vessels are caught polluting. The Coast Guard's Office of Vessel Activities (COMDT CG-543) and the Office of Investigations and Casualty Analysis (COMDT CG-545) in Washington, DC, are tasked with processing this data in all environmental crimes cases, as well as SWOMS data in cases of repeat offending vessels or companies with the system installed. SWOMS is currently used to monitor ships on probationary periods to make sure they are holding up to the MARPOL standards. As SWOMS becomes more standardized and effective, it may be implemented on a larger scale. Currently, the system for reviewing these data has not been standardized, making data analysis difficult for Coast Guard personnel. The goal of this project is to develop a system for the acquisition, storage, and analysis of Special Waste Oil Monitoring Systems (SWOMS) and oil record book (ORB) data. This would allow the Coast Guard to more effectively track violations and enforce environmental compliance laws, ultimately reducing oceanic pollution.

The first objective was to learn about oil management systems on ships, including the physical components as well as the standard conventions for recording information in the ORB. The next step was to understand how to verify compliance within the ORB and SWOMS data. This included identifying what constitutes a discrepancy or anomaly. Interviews with Coast Guard personnel revealed that items to flag include a significant difference between SWOMS and ORB data or between two subsequent days within the SWOMS or ORB data alone.

The next objective was to understand the current data submission process and to develop a proposal for a new standard environmental compliance plan to the Department

of Justice to use for future non-compliance cases. This would include the implementation of SWOMS as well as methods of data transmission, timelines of data submission in addition to the format of the data submitted. Current methods and formats do not allow for easy analysis, proving the need for a standard submission process. Originally this was to be accomplished by creating a template for the ORB data and an additional one for the SWOMS data. However, attempts to create a template for the ORB made it quickly apparent that the book's format was incompatible with the table layout required by a database. The sounding log is already laid out in a table, and would simply have to be entered into a standardized spreadsheet. The SWOMS data are emailed in a consistent format which allowed for the development of an executable program to automatically parse the data into a standardized spreadsheet, organized in order to be easily comparable to the sounding log spreadsheet. Figure 1 shows the current and proposed data transmission methods.

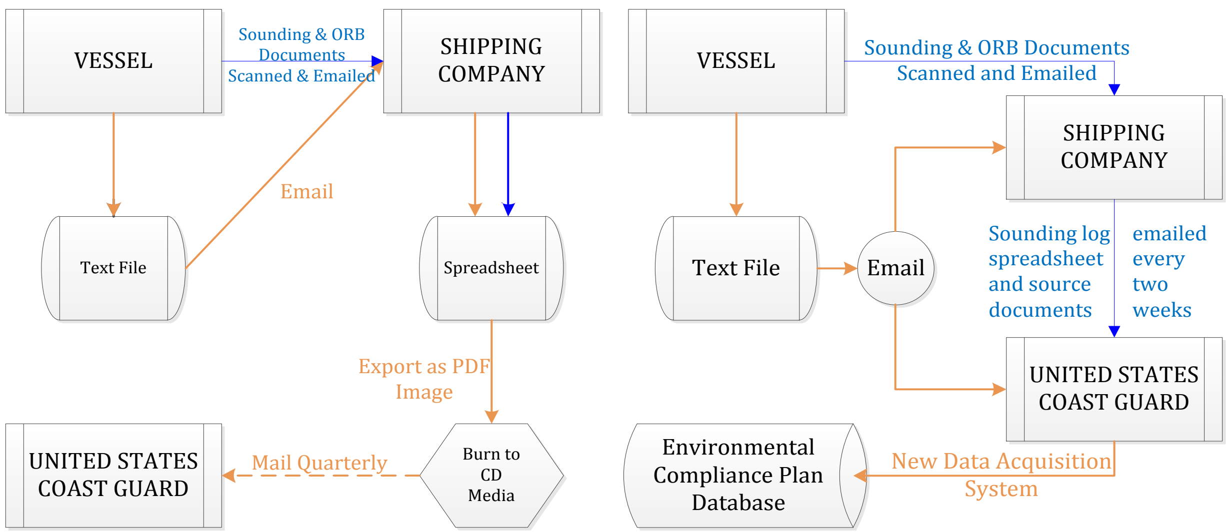


Figure 1: Data Transmission Path, Current (to the left) and Proposed (to the right)

The final objective was the development of the database. The primary goal was to compare manually recorded tank data recorded in the sounding log book against the automatically recorded SWOMS data, as well as to analyze trends within the SWOMS data itself. The database includes tables for sounding book data, SWOMS data, a list of vessels, and a list of tanks for each vessel. Users can manually add or modify data using the forms within the database. They can also use the database to analyze data and generate printable reports highlighting discrepancies. Once these steps were completed, it was crucial to test the database to find any errors and to determine whether the user interface was user-friendly. The team and the members of the Coast Guard who will be using the database assisted in testing and improving the database. Once this was completed, a user manual was written in order to assist the users to understand the database and maximize its potential.

The completion of this project is a great improvement for the Coast Guard in its mission to prevent marine pollution. Together, this standard submission process and database constitute a standardized system for data analysis that will enable a great increase in the efficiency of the Coast Guard personnel who analyze this waste-stream data. This will effectively help the Coast Guard to enforce maritime environmental compliance laws, reducing the amount of oil-waste pollution in the world's oceans.

Authorship

The members of this IQP team attest that this report is the result of equal work shared across all team members. We believe that each team member deserves equal credit for the work contained herein. The following describes major written contributions from each member:

Patrick Brodeur focused on the technical elements of data transmission into the database. He was in charge of creating the SWOMS and Tank ID submission templates for the database, as well as writing VBA scripts and a program to convert received plain-text emails into a database-readable format. He was in charge of writing the user manual for Outlook/Exchange integration for the minimally-manage mailbox. He researched environmental regulations and wrote portions of the background section on Databases. He also designed several flowcharts.

Renée Lanza focused on shipboard waste management systems and data analysis. She was in charge of creating the sounding submission template for the database. She researched current waste management systems and several non-compliance cases. She was in charge of determining the format and data contained in the reports generated by the Environmental Compliance Plan (ECP) Database. She was also tasked with database testing and wrote significant portions of the Results and Analysis. She also assisted in an audit for the Department of Justice.

Elizabeth Morris focused on shipboard environmental management systems. She researched ship classifications, specifications for oil water separators and tanks, Port State Control, and environmental management systems on shipping vessels. She also wrote significant portions of Appendix B and the ECP Database User Manual. Lisa also performed extensive reviews of first-draft materials.

Edward Osowski focused on researching shipping and added the relevant appendices. His major project contribution is the coding and development of the ECP Database. He also wrote large portions of the ECP Database sections in the report. He also worked on formatting and organizing the references.

Other written portions were largely team-written in group meetings. These portions reflect creative input from all members of the team. In addition, each team member has peer-reviewed others' sections.

Acknowledgements

Though this project was a team effort, it would not have been possible without the help of numerous people. First, we would like to thank the United States Coast Guard (USCG) and Worcester Polytechnic Institute (WPI) for giving us with the opportunity to participate in this project. Throughout this project we were able to use knowledge gained in the classroom at WPI towards achieving a solution for a real world problem. In particular, we would like to thank our ID 2050 instructor Professor Creighton Peet, who prepared us to participate in this Interactive Qualifying Project.

We would especially like to thank our liaison LCDR Chaning Burgess for providing us with both a challenging and interesting project for us to work on as well as his help along the way. In addition, we would like to thank LT Sharmine Jones, who helped organize many of the aspects of our experience at United States Coast Guard Headquarters. We are confident that our time at USCG Headquarters would not have been the same without these individuals.

We would also like to thank Mr. Shahzad Aziz of USCG Headquarters for his technical expertise and assistance in our project. He helped us to realize challenges and guided some of our thinking towards solutions. Shahzad's help was much appreciated and very valuable to this project.

Also of great importance to our project was Ken Olsen of USCG Headquarters. His sharing of his experiences in high profile environmental crimes cases helped us to create an analysis portion to the database to address concerns that he had seen first-hand. Without his knowledge of shipboard waste systems and shipboard monitoring systems as well as previous environmental crimes, our project would not have been the same. Much of his input and experiences led to certain analyses within the database.

Outside of the USCG Headquarters but still within the Coast Guard, we would like to thank CAPT Rich Sanders of the Coast Guard Academy and CDR Paul Lattanzi. These two individuals helped organize a trip for our group to an oil tanker with USCG ship inspectors before our arrival in Washington DC. This experience helped us to understand many of the problems and systems involved in this project and gave us valuable insight into how compliance is verified from a ship inspector's point of view.

Lastly, we would like to thank our advisors, Professor Mustapha Fofana and Professor John Orr, without whom we could not have completed this project. They have been a part of this project since it was first assigned and have been of great help all along the way, especially while in Washington DC. We are most appreciative of all the work and time they have contributed to our project while simultaneously advising several other projects here in Washington and back in Worcester.

Table of Contents

List of Figures

List of Tables

Glossary

Ballast Tank: Large compartment within a ship's hull used to regulate ship buoyancy. Typically these chambers are filled with sea water when a ship unloads its cargo and the sea water is discharged when loading new cargo. Without ballast tanks, ships can become unstable and potentially capsize.

Bilge Tank: The lowest compartment on the ship where the two sides meet. This area holds any water drips down to the bottom of the ship. The water often contains oil.

Environmental Compliance Plan (ECP): Clearly defined guidelines and responsibilities to ensure compliance with all environmental regulations.

MARPOL: Short for "Marine Pollution." It refers to the International Convention for the Prevention of Pollution from Ships. It is a set of international regulations that govern shipboard wastes, including the regulation that oil waste cannot exceed 15 parts-per-million.

Oil Record Book (ORB): A book onboard the ship that *must* contain all records of oil-waste discharge and levels. Failure to keep an accurate oil record book is a crime. The Chief Engineer must sign the book after every discharge stating that the data are accurate.

Oily Water Separator (OWS): A system that separates large amounts of oil from oily water.

PPM: Parts per Million (of oil), a measure of concentration

Port State Control (PSC): Port State Control is the inspection of foreign ships in national ports to ensure that ships are complying with international regulations. In the United States, the Coast Guard assumes the role of Port State Control.

Sounding Log: A written log of level and volume soundings taken for all the tanks in the engine room.

Special Waste Oil Monitoring System (SWOMS): A system that monitors all waste-oil discharges as well as the flows of waste-oil in different areas of the ships and keeps records of all incinerations. It is considered an accessory to the oil water separator. This system is not on all ships, and supplies the same information as well as a little bit extra as the oil record book.

Standardized Submission Process: A universal plan for the data to get from the vessel to the Coast Guard's database for analysis. This includes all elements in the acquisition of the data including templates and transmission methods.

Acronyms

APPS: Act to Prevent Pollution by Ships

CSV: Comma Separated Values

CWA: Clean Water Act

DAPS: Data Acquisition and Processing System

DHS: Department of Homeland Security

DoJ: Department of Justice

IMO: International Maritime Organization

IOPP: International Oil Pollution Prevention [Certificate]

MARPOL: "Marine Pollution" – Refers to the International Convention for the Prevention of Pollution from Ships

MOU: Memorandum of Understanding

NOA: Notice of Arrival

NOD: Notice of Departure

ORB: Oil Record Book

OWS: Oily Water Separator

PSC: Port State Control

PSCO: Port State Control Officer

PWSA: Ports and Waterways Safety Act

SWOMS: Special Waste Oil Monitoring System

UNEP: United Nations Environment Programme

USCG: United States Coast Guard

VBA: Visual Basic for Applications

1 Introduction

The majority of this planet is covered by oceans, which are teeming with wildlife in delicate ecosystems. However, many people disregard the effects of polluting these oceans. Marine pollution is the entry of any contaminants or foreign materials into oceanic areas, which brings about harmful effects. Pollutants can include oil spills and waste released from vessels. Vessels that release great quantities of pollutants into the oceans show a blatant disregard for the marine environment which poses a risk to the quantity and biodiversity of marine life around the globe.

There has been a great deal of progress made in regards to prosecuting vessels found to be in non-compliance with pollution regulations. Due to technological advances, marine pollution from vessels is more detectable, making environmental crime cases easier to prove than in the past. However, increases in global marine traffic have the potential to increase the number of environmental crimes at sea. The Coast Guard, in conjunction with the Department of Justice, has the responsibility to catch and prosecute vessels that are not in compliance. When shipping companies are caught committing environmental crimes, such as dumping waste into the ocean, oftentimes they are assessed a fine and put on a probationary period. For some vessels, data from their environmental management systems are monitored by the United States Coast Guard during this period. It would be beneficial for the Coast Guard to have a more advanced system to analyze the data from these vessels in order to verify that their discharges are in accordance with current regulations.

Prior to the development of electronic monitoring, vessels that illegally dumped wastes would have to be caught actively dumping oil wastes or would have to be found making obvious record-keeping errors. The installation of a monitoring system on all repeat-offending vessels holds vessels accountable at all times. It is evident that this installation helps to prevent repeat offenders, but there were still a few complications in the system to be worked out. The sheer quantity of this data means that it is still difficult to enforce environmental regulations on vessels during probationary periods. In addition, the data collected by the United States Coast Guard are also difficult to analyze. As there is no efficient method to analyze the data, it is difficult for Coast Guard personnel to analyze the data for anomalies, discrepancies, tampering, and non-compliant activities. Despite the need for a cohesive system to compile and analyze data from the monitoring systems, little had been done to establish such a system due to the fact that these monitoring systems are relatively new.

The goal of this project is to develop a means to compile and analyze data from shipboard environmental management systems for the Coast Guard's Environmental Compliance Program. To accomplish this, we need to determine how data are collected with the current system; create or find a way to transmit these data into a Coast Guard database system; and then create that system and a method to analyze the data in it. In order to make the best possible system, ways to improve the current data collection system in addition to creating a database to collect and analyze these data must be identified. Another component of this project is to create a standard submission plan for future Department of Justice MARPOL cases. These developments are necessary because protection of the environment is an important mission of the Coast Guard. This system will allow the United States Coast Guard to better detect, monitor and prevent future pollution crimes.

2 Literature Review and Background

Marine pollution occurs when harmful effects result from the entry of chemicals, particles, industrial, agricultural and residential waste, or the spread of invasive organisms into the ocean. Types of marine pollution include overfishing, trawling the ocean bottom, oil spills, ballast water taken up at sea and released in port, sewage, gray water, oily bilge, and hazardous and solid waste (Katsioulidus, 2010; Milton, 2006).

This chapter provides an overview of marine shipping and the pollution caused by it, as well as the current systems used to prevent and reduce such pollution. We will then discuss the shipping industry, the Coast Guard's involvement in shipping, environmental regulations as they pertain to the shipping industry, enforcement of maritime environmental regulations, waste treatment on ships, as well as current environmental and database management systems. Finally we will discuss two example court cases pertaining to waste oil environmental crimes, and how those relate to the larger problem of developing a system to effectively compile and review the data from shipboard environmental management systems.

2.1 Marine Shipping

While shipping has always been an important part of commerce both in the United States and abroad, the globalization of the world economy over the last 50 years has led to an unprecedented growth in shipping, with a current world fleet of over 50,000 vessels (Round Table of International Shipping Organizations, 2010). This has also led to an increase in marine pollution.

2.1.1 Worldwide Shipping Traffic

As the world economy has become more connected, international shipping has grown significantly. This is because it is still seen as the most cost-effective and practical method to transport large-volume and large-weight goods among countries separated by oceans (Nguyen, 2010, p. 22). In 2007, there were more than 8 billion short tons (1 short ton = 2000 lbs = 907.2 kg) of international cargos transported by sea at a value of more than 7.7 trillion US dollars, a significant increase from the 4.6 billion short tons and 2.2 trillion dollars in 1995, as shown in Table 1 which shows world and US exports by sea.

Table 1: World and US exports by value and weight (Nguyen, 2010, p. 22)

	World Exports		U.S. Exports	
Year	Value (billions US $)	Weight (millions of short tons)	Value (billions US $)	Weight (millions of short tons)
1995	2,252.00	4,651.00	228.00	475.00
1996	2,354.00	4,758.00	238.00	451.00
1997	2,422.00	4,953.00	225.00	432.00
1998	2,243.00	5,631.00	192.00	405.00
1999	2,354.00	5,863.00	182.00	400.00
2000	3,027.00	5,963.00	199.00	415.00
2001	2,901.00	5,984.00	199.00	399.00
2002	2,979.00	5,891.00	191.00	384.00
2003	3,646.00	5,948.00	206.00	373.00
2004	4,551.00	6,598.00	234.00	416.00
2005	2,590.00	6,893.00	263.00	402.00
2006	6,301.00	7,122.00	308.00	434.00
2007	7,723.00	8,032.00	375.00	467.00

As the world's single largest economy and one bordered by oceans on multiple sides, seaborne trade accounts for a significant portion of US commerce. In 2007, shipping accounted for 78% of the weight and 45% of the value of US international trade (Nguyen, 2010). The reason the value is low relative to the weight is that ship-borne trade is primarily in low value-per-ton commodities such as grain, oil, or mass-produced merchandise. While the US has many ports all along its coastline, the three biggest by amount of goods handled are South Louisiana, Houston, and New York/New Jersey (US Army Corps of Engineers, 2009). Figure 2 shows the flow of trade through the world's oceans, with deeper red signifying more vessels traveling along a route, as well as major straights and canals being marked.

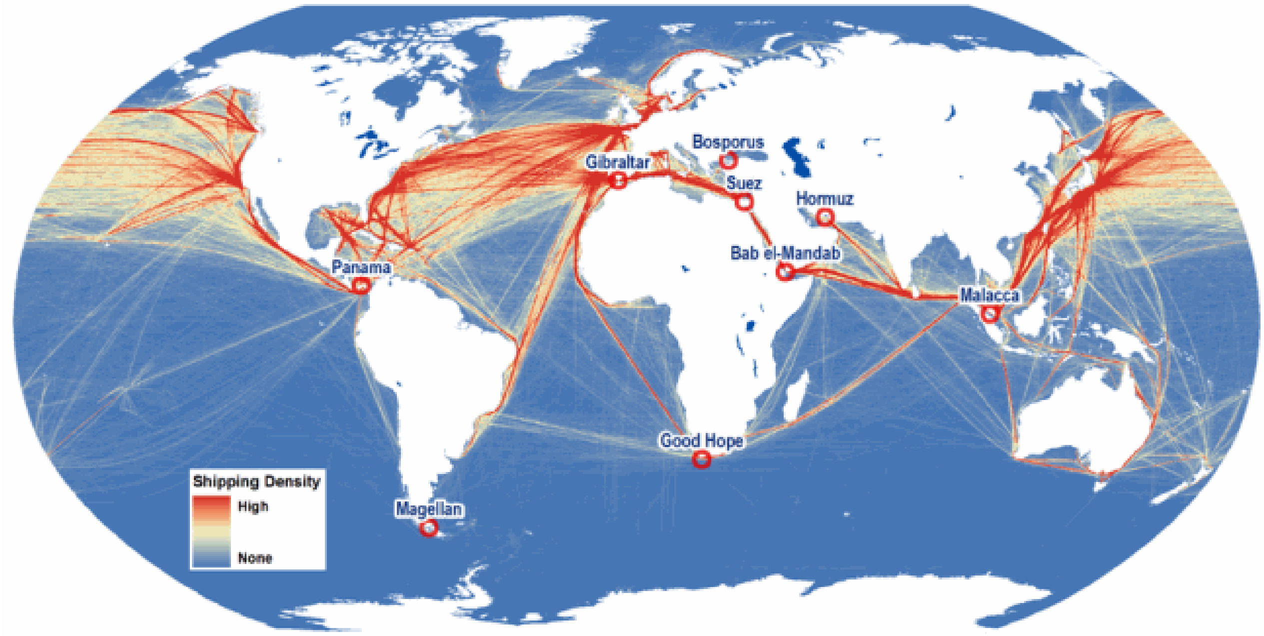


Figure 2: International Maritime Shipping Routes (Kochi International Marina, 2011)

As can be seen in the figure, a significant amount of seaborne trade routes start or terminate at US ports and additionally US vessels use the Panama canal to facilitate easy trade between the East and West Coast.

2.1.2 International Maritime Organization (IMO) Vessel Identification

The large number of shipping vessels could have the potential to cause a great deal of confusion without a unique identifier. However, as of 1996 the IMO identification number scheme has become mandatory for all ships (International Maritime Organization, 2011). As part of this scheme, each vessel is assigned a permanent number for identification. The number refers to the portion of the hull that encloses the machinery space, so it will remain constant even through changes in the flag nation, company, or even name of the vessel. The number will never even be reassigned to another vessel. It consists of the three letters "IMO" followed by a seven-digit number that is assigned to all ships by the company IHS Fairplay once the keel is laid during construction. Vessels are required to permanently mark their IMO number in a visible place, either on the ship's hull or superstructure, as well as mark it internally.

2.1.3 Vessel Classification

There are several classifications of ships in the world merchant fleet (Round Table, 2011). These ships are all involved in world commerce and trade, whether through the transport of raw goods, the transport of people, or by supporting pre-existing industries at sea such as offshore oilrig workers. The various types are indicated on the next pages in Table 2. The pictures show an example of that vessel classification and the descriptions are to the right of the images.

Table 2: Vessel Classification

Vessel Classification	**Description**
Container Ship 	Typically carries manufactured goods and products in standard shipping containers
Bulk Carrier 	Transports raw materials and can be recognized by the hatches raised above deck levels covering the large cargo holds
Tanker 	Primarily carries crude oil, other petroleum products or chemicals in its cargo spaces, which are designed for holding liquids NOTE: Tankers are similar in appearance to bulk carriers; however, a tanker's deck is flush and is covered with product pipelines and vents

Ferry  	Usually performs short journeys for passengers, cars, and commercial vehicles
Cruise ship 	Luxurious ship that goes on long journeys
Specialist ships 	Vessels such as anchor handling and supply vessels for the offshore oil industry, salvage tugs, ice breakers, and research vessels

As the table shows, there are many different types of vessels. These vessels all must follow environmental regulations to some extent. Larger vessels such as tankers and container ships must abide to stricter regulations depending on the overall size of the ship. Smaller vessels still have to adhere to strict regulations but in a different manner. For example, all vessels made in the United States as well as cruise ships that call upon U.S. ports often must submit their plans and blueprints to the United States Coast Guard's Marine Safety Center, where the blueprints are checked for safety features as well as whether the vessels abide to all codes. There are many types of vessels that navigate the ocean, and all must abide by environmental and safety regulations.

2.1.4 Tanks within a Ship

Tanks allow for the storage of various substances on vessels, including waste and oil. Tanks are highly relevant to this project because tracking the amounts of oil contained in the tanks is a substantial portion of tracking the movement of oil throughout the whole vessel to ensure compliance with environmental regulations. The layout of tanks on various ships is different. The layout differs since the numbers of tanks as well as the types of tanks differ from ship to ship. However, to give an idea of a potential layout, Figure 3 below shows a diagram of the tanks contained in the Merchant Marine training vessel T/V Kings Pointer. Of specific importance in this figure are the waste, fuel, and lube oil tanks. These show where the various types of oil are stored on this particular ship.

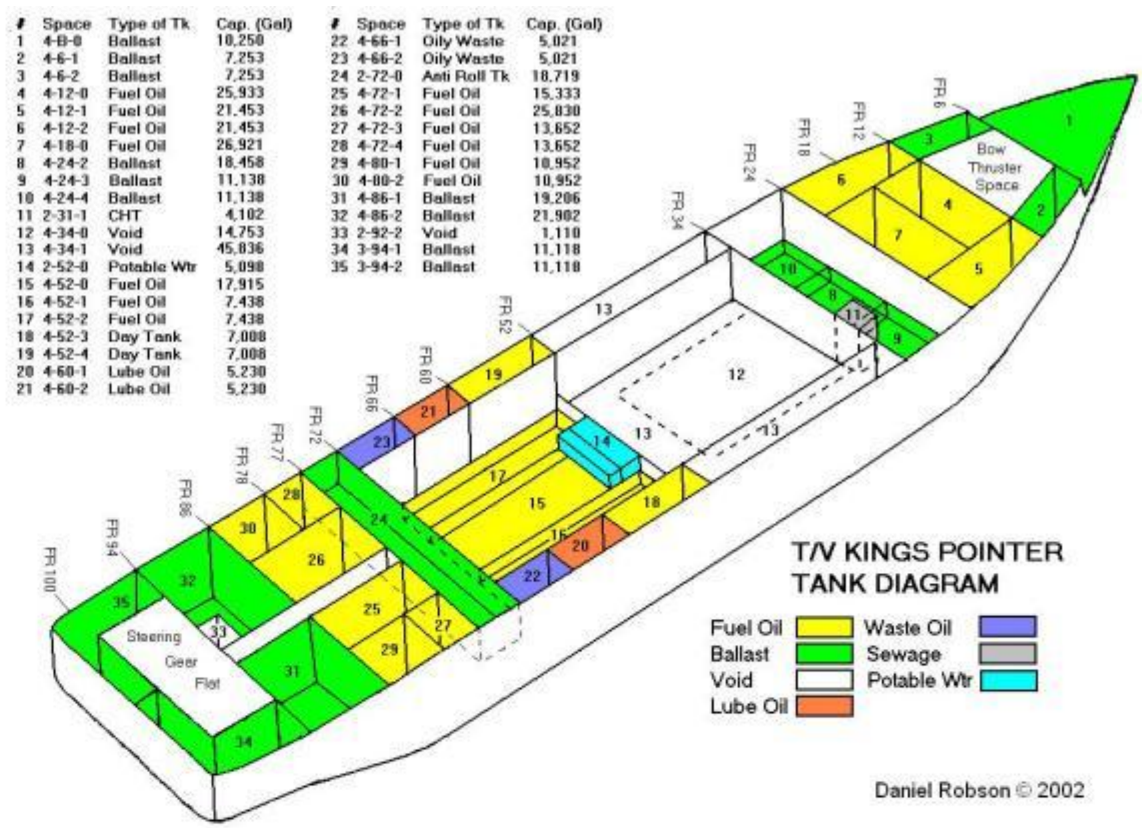


Figure 3: T/V Kings Pointer Tank Diagram (Robson, 2002)

By looking at the diagram, it is evident that there is a lot of stored oil on these ships within many tanks. All of the yellow, orange, and purple tanks contain some consistency of oil. The oil within these tanks is systematically monitored to abide by international marine pollution regulations, as will be discussed later on.

2.1.5 Coast Guard Involvement

As it is the Coast Guard's duty to protect and regulate US coastal areas, they take a major role in the safety, navigation, and law enforcement of shipping in US waters (US Coast Guard, 2011B)(For more information on the United States Coast Guard See Appendix A. United States Coast Guard). In order to facilitate marine safety and environmental protection, they have formed partnerships with several civilian commercial entities in order to better coordinate safety for both people and the environment.

Since 1995, the US Coast Guard has worked with American Waterway Operators to improve safety for the tugboat, towboat and barge industry (US Coast Guard, 2011A). This partnership is the first of its kind between the USCG and any segment of the US maritime industry, using effective analysis, open dialogue and non-regulatory solutions to address a multitude of issues including crew fatalities, tank barge spills, and safe operations during dangerous weather. In 1996 the Coast Guard signed an agreement with the Passenger Vessel Association to improve communication and the working relationship between the USCG and the domestic vessel passenger industry in order to promote safety and protect the environment. In 1997 the USCG started a partnership with the International Council of Cruise Lines, now called Cruise Lines International Association (CLIA), in order to complement already existing international conventions in regards to cruise ship safety to make the USCG more efficient at working with the international passenger vessel industry.

In 1998, the USCG (2011A) signed agreements with the International Association of Independent Tanker Owners (INTERTANKO) and with the Baltic and International Maritime Council (BIMCO). The INTERTANKO agreement works on generic safety issues with the tank vessel/maritime oil and chemical industry as well as works to prevent environmental damage from tank vessel incidents. The partnership between the USCG and BIMCO works to prevent environmental damage from commercial vessel incidents. In 2001 the USCG signed an agreement with the Chamber of Shipping of America with several goals including minimizing non-essential communications during critical operations, improving overall communications between members of a bridge team, and enhancing maritime safety and quality of the environment by working on the issue of crew endurance. The agreements the Coast Guard has made with these organizations, in addition to the Coast Guard's other work, have done much to protect the environment from a variety of issues including shipboard waste.

2.2 Environmental Regulations

It has been recognized globally that environmental regulations are necessary to preserve a healthy and safe world. The global community has become very cognizant of the need to protect the global environment, as demonstrated by the signing of the Kyoto Protocol (Kyoto Protocol Takes Effect, 2005), MARPOL (International Maritime Organization, 2011, p.94), and the Basel Convention (Conference of the Parties to the Basel Convention, 2010). These and other reforms were talked about as early as the 1950's (e.g. International Convention for the Prevention of Pollution of the Sea by Oil, a.k.a. OILPOL) (International Maritime Organization, 2011, p.9) and have been since updated, for example with the additions of annexes VI and VII of MARPOL (Butt, 2007). Currently the United States has signed onto these and other international reforms (International Maritime Organization, 2011, p.104; United Nations, 2011), but has not ratified them. Signing onto these reforms indicates the United States' approval; however the United States has not agreed to be legally bound to these regulations. Instead, the United States maintains its own set of policies that are nearly equivalent (e.g. US Congress, 2000). It is the duty of the United States Coast Guard to uphold these regulations (Office of the Law Revision Counsel, US House of Representatives, 2011).

2.2.1 US Coast Guard Environmental Enforcement

In accordance with the missions of the Coast Guard set forth in the Homeland Security Act (2002), one of its missions is to enforce marine environmental protection. The Coast Guard is tasked with the enforcement of laws on the water, much as police are charged to uphold the law on land (Chief of Staff, USCG, 1997). The USCG must "ensure an effective response to actual or threatened pollution incidents in order to minimize damage to the public and the marine environment… [and] enforce applicable pollution laws and regulations" (ch.2, p.1). The Coast Guard may also conduct an investigation into criminal activities on the water through the Coast Guard Investigative Service. The prosecution of these crimes must be passed through the appropriate geographic office of the Department of Justice (DoJ), or through the DoJ Environmental Crimes Section in Washington, DC.

2.2.2 MARPOL/APPS

MARPOL, short for "marine pollution", is more correctly known as the International Convention for the Prevention of Pollution from Ships (International Maritime Organization, 2011, p.94). In the United States, the Act to Prevent Pollution from Ships is "An act to implement the Protocol of 1978 Relating to the International Convention for the Prevention of Pollution from Ships, 1973, and for other purposes"(US Congress, 2000, p.1). MARPOL, in brief, is an international treaty signed by all major shipping nations and over one hundred additional countries around the world (International Maritime Organization, 2011, pp.100-117). Figure 4 below shows a world map with all MARPOL signatory nations highlighted in green.

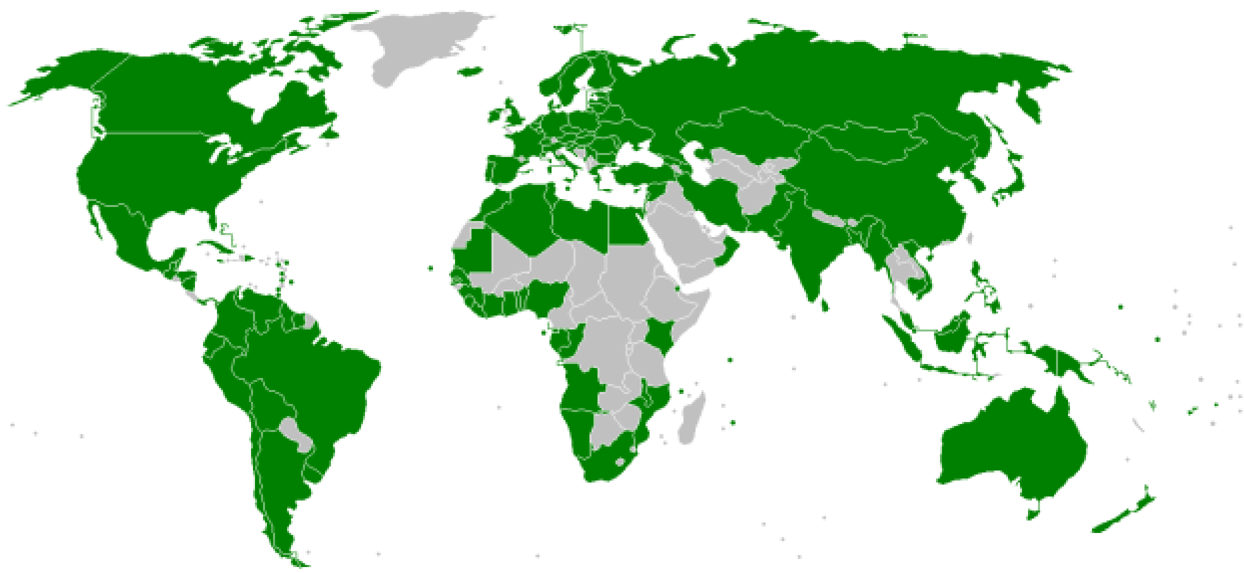


Figure 4: MARPOL Signatory Nations. Source: Wikimedia Commons, 2008.

As can be seen by the map, most nations with a significant coastal area have signed on to MARPOL, with the exception of Greenland, Saudi Arabia, and eastern African nations.

The current MARPOL treaty is divided into six annexes that each covers a certain subset of shipboard wastes (International Maritime Organization, 2011, p.155).

Annex I—Annex I of MARPOL addresses oils and oily bilge water that is generated during normal operation of most marine vessels (Butt, 2007). Annex I specifies that any discharge with an oil concentration in excess of 15 parts per million is not allowed and is subject to corrective action (Lin, 2006). With respect to this project, Annex I is the most relevant considering that the system the team will work with is checking for this particular type of release.

Annex I is the culmination of previous international efforts, bolstered by additional reforms, drawn mostly from OILPOL (OECD, 2003, p.10). It has a two-fold way of preventing oil pollution. First, there are discharge regulations, as previously mentioned. Annex I allows certain limits of oil in water, but in concentrations low enough that oil doesn't leave a sheen on the water (p.13). It also specifies special areas where no oil discharge is allowed, like the Mediterranean Sea, the Baltic Sea, the Antarctic region, and other areas where there is not a sufficient ocean current to disperse oil discharge (p.15).

In order to achieve these very low discharge levels, ships compliant with MARPOL must have certain equipment on board. Specifically, each ship must have an oily water tank, at least one bilge pump, an oil-water separator, and a means to measure outbound oil levels and redirect discharge back through the system if the oil concentration is too great (OECD, 2003, p.18). The elements of this system must be operated and maintained properly in order to avoid further oil spills within the ship, as well as to avoid contaminating the discharge water with more oil. Furthermore, oil sludge must be entirely separated and disposed of either by incineration or by disposal at a port.

The final element of Annex I is its structural requirements (OECD, 2003, p.19). First, ballast tanks were required to change to either a segregated ballast system, or use a crude oil washing system. This was a departure from the former system in which significant amounts of oil could get drained to the sea when combined cargo/ballast tanks were emptied. The other major structural change mandated by MARPOL was the addition of a double hull requirement (p.23). Existing tankers were given until their 25th year of operation to either upgrade or employ an alternative system such as putting segregated ballast between the cargo and hull on 30% of the ship's bottom and sides or using a hydrostatically balanced load system (pp.23-24).

Annex II—Annex II addresses "noxious liquid substances" which are typically toxic chemicals that pose a severe risk to life (US Congress, 2000). This includes chemicals like n-Butyl benzene, lauric acid, triethanolamin, triisopropylated phenyl phosphates, and other hazardous, usually volatile substances (Höfer, 1999). These and other chemicals are known to cause health problems, from minor irritation to cancer or death.

Annex III—Annex III "covers all waste that requires special treatment and/or disposal and includes chemicals used in photo-processing, dry cleaning, print shop waste, fluorescent and mercury vapor bulbs and batteries" (Butt, 2007, p.594).

Annex IV—Annex IV relates to sewage and waste water, restricting the dumping of raw sewage to 12 miles from shore and banning the dumping of treated waste within 4 miles from shore (Butt, 2007). This is important because of the bio-hazardous nature of human waste and its danger to marine life.

Annex V—Annex V applies to garbage, or solid wastes (US Congress, 2000). Here, the United States deviates from MARPOL in that it allows US Navy ships the latitude to dispose of additional wastes as long as they comply with less-stringent standards. This provision is in place for vessels that are incapable of reasonably complying with the full MARPOL standards, or for vessels that cannot comply without impairing mission capabilities.

Annex VI—Annex VI is the newest addition, added to United States law by the Maritime Pollution Prevention Act of 2008 (Office of the Law Revision Counsel, US House of Representatives, 2008). According to an EPA press release, this annex regulates emissions from category three diesel engines used primarily on "container ships, tankers, cruise ships, and bulk carriers" (Milbourn, 2008, p.1). The purpose of Annex VI is to limit nitrogen and sulfur-based combustion byproducts that contribute to atmospheric damage and acid rain.

2.2.3 Clean Water Act (CWA) & Ports and Waterways Safety Act (PWSA)

The Clean Water Act is the common name for the legislation that regulates surface water pollution in domestic waterways. The Clean Water Act is meant to “sharply reduce direct pollutant discharges into waterways, finance municipal wastewater treatment facilities, and manage polluted runoff” (EPA Watershed Academy, 2008, p.1). The overall purpose of the Clean Water Act is to maintain the environmental integrity of United States waterways and ensure that they are clean enough for human use.

The CWA can be broken into a few distinct sections. It establishes water quality and monitoring standards (EPA Watershed Academy, 2008, p.2), antidegradation policies, a permit program for high-concentration polluters, runoff pollution control, restrictions on federal pollution, and provisions to provide money to states for water pollution control efforts. The CWA also includes criminal punishments for violations of the regulations set forth (US Congress, 2002).

The Coast Guard comes in to enforce the Clean Water Act, where it is assigned to “engage in such research, studies, experiments, and demonstrations as…appropriate, relative to the removal of oil from any waters and to the prevention, control, and elimination of oil and hazardous substances pollution” (US Congress, 2002, §104, p.10). This ties in well to the Ports and Waterways Safety Act (2002), which gives the Coast Guard the task to supervise vessel operation in order to:

> [R]educe the possibility of vessel or cargo loss, or damage to life, property or the marine environment and ensure that the handling of dangerous articles and substances on the structures in, on, or immediately adjacent to the navigable waters of the United States is conducted in accordance with established standards and requirements (§1221).

The challenges involved in maintaining safe waterways fall on the Coast Guard. The monitoring of the environment and enforcement of law is shared between the Coast Guard and the states in which they are operating (US Congress, 2002). This project is an extension of the Coast Guard’s task of monitoring and defending the marine environment.

2.3 Port State Control

Port State Control (PSC) is the inspection of foreign ships in national ports to check the condition of the ship, its equipment, and its crew, for compliance with international regulations (International Maritime Organization, 2001, p.1). In the United States, Port State Control is run by the Coast Guard but all other countries with major ports have a Port State Control. The flag state is the state under whose laws the vessel is registered or licensed. It is primarily the responsibility of the flag state to make sure the ship is following these regulations, including those relating to inspection, certification, and the issuance of safety and pollution prevention documents. PSC provides a second opportunity to catch substandard ships. There are several regional PSC organizations and agreements. Many of these organizations have signed Memoranda of Understanding (MoU), spanning all of the oceans. The U.S. has not signed a MoU; however they still inspect vessels to make sure that they are compliant with international regulations, as well as U.S. laws.

2.3.1 Notice of Arrival

U.S. and foreign vessels are required to submit a Notice of Arrival (NOA) or Notice of Departure (NOD) before they may enter or depart ports within U.S. waters (Minerals Management, p.1). The NOA requires information about the vessel and its crew, passengers, voyage, and cargo. The NOA must be submitted according to the specifications in Table 3 below for vessels that are not towing vessels.

Table 3: Specifications for NOA Submissions Based on Voyage Time

If your voyage time is—	You must submit an NOA—
(i) 96 hours of more; or	At least 96 hours before entering the port or place of destination; or
(ii) Less than 96 hours	Before departure, but at least 24 hours before entering the port or place of destination.

If ships do not submit the NOA in time, they will not be allowed into the port.

2.3.2 Ship Inspections

A Port State Control Officer (PSCO) has the authority to carry out PSC ship inspections (International Maritime Organization, 2001, p.3). The PSCO should be an experienced officer qualified as a master or chief engineer with appropriate seagoing experience; or he should be educated in a maritime related field and have specialized training; or he should be a qualified administrative officer with experience and training in performing inspections (p.7). PSCOs are required to undergo initial training, as well as periodic seminars. Below, in Figure 5, is a picture of a United States Coast Guard ship inspector on the job. In addition to picturing the inspector in his work, the image also depicts a machine room area.

Figure 5: Inspector on Board a Ship (Sherman, 2009)

PSC may inspect any foreign ship included under international regulations that enters their port (International Maritime Organization, 2001, p.1). Before even boarding the vessel, PSCOs look at the general appearance of the vessel in order to judge its state of maintenance (p.5). Upon boarding, the PSCO is to examine the vessel's relevant certificates and documents. If this initial examination leads the PSCO to believe that the ship is under a good standard of maintenance, the inspection will primarily be confined to reported or observed deficiencies. However, a more detailed inspection will be performed if this examination leads to the discovery of clear grounds. Clear grounds is any evidence that shows that the ship, its equipment, or its crew do not meet up with relevant regulations; or that crew members are unfamiliar with shipboard procedures regarding ship safety or pollution prevention (p.3). This can include:

- Missing or deficient safety, pollution prevention, or navigational equipment;
- Invalid or missing certificates;
- Missing, incomplete, or falsely maintained documentation;
- Serious hull or structural deterioration or deficiencies that threaten the integrity of the ship;
- Evidence that the master or crew is unable to perform essential shipboard operations, or evidence that these operations have not been performed;
- Evidence that crew members cannot communicate with each other or with other persons on board;
- False distress alerts that have not been followed up with proper cancellation procedures;

- Receipt of a report containing information that a ship appears to be substandard (p.6).

The main focuses of a more detailed inspection are the ship's structural and equipment requirements, guidelines for discharge requirements as are detailed in Annexes I and II of MARPOL 73/78, guidelines for ship operations, minimum manning standards, and guidelines for PSC related to the International Safety Management (ISM) Code (International Maritime Organization, 2001, p.10). The MARPOL discharge regulations outlined in Annex I are the most relevant to this project. This portion of the inspection will include checking the IOPP Certificate and the equipment listed in the attached Record of Construction and Equipment (p.46). The oil record book must also be investigated (See Section 2.4.1 Oil Record Books). In addition to the port state report on deficiencies, a report should be supplemented with the statements of the PSCO or other observers of the pollution; reports of analyses of samples taken; photographs of an oil slick; or copies of relevant pages of the ORB, log-books, discharge recordings, etc.

2.3.3 Consequences for Non-Compliance

If a ship is regarded to be substandard, then this information is submitted to the port state (country where it is currently at port) (International Maritime Organization, 2001, p.34). Once this information is received, the authorities are to begin to investigate the matter and take corrective action in order to safeguard the ship and its crew and to eliminate any threat of harm to the marine environment (p.35). In the case of deficiencies that are clearly hazardous to safety or the environment, the PSCO must ensure that the hazard is removed before the ship is allowed to proceed to sea, which may include detention or a formal prohibition for a ship to continue an operation (p.36).

If the port state determines that the sailing of the ship would present a danger to the ship or the crew, or an unreasonable threat to the marine environment, the port state may detain the vessel until it ceases to present this danger (International Maritime Organization, 2001, p. 3). This may occur regardless of the normal schedule of departure of the ship. When deciding whether or not to detain a vessel, the PSCO assesses the documentation on the ship as well as whether the ship has the required manpower (p.41). The PSCO assesses whether the ship is able to:

- Navigate safely;
- Safely handle, carry, and monitor the condition of the cargo;
- Operate the engine-room safely;
- Maintain proper propulsion and steering;
- Fight fires effectively in any part of the ship if necessary;
- Abandon ship speedily and safely and effect rescue if necessary;
- Prevent pollution of the environment;
- Maintain adequate watertight integrity;

- Communicate in distress situations if necessary; and
- Provide safe and healthy conditions on board.

If the ship is unable to perform any of these, the ship should be strongly considered for detention.

If this occurs, the authorities must notify any maritime, consular, and/or diplomatic representatives of the flag state in the area of the ship, and request for them to initiate or cooperate with investigations (International Maritime Organization, 2001, p.35). If the deficiencies cannot be remedied in the port of inspection, the Port State authority may allow the ship to proceed to the nearest repair yard available (p.36).

2.4 Current Waste Management

There are many different options for disposing of waste. The ships have the option to unload wastes at ports, which in turn makes more waste for different port cities to deal with. The other option is that ships treat waste en-route with on-board treatments (e.g. compactors, incinerators, etc.) (Butt, 2007). Treatment and methods of dealing with shipboard waste reduce the amount of pollutants put into the ocean.

2.4.1 Oil Record Books

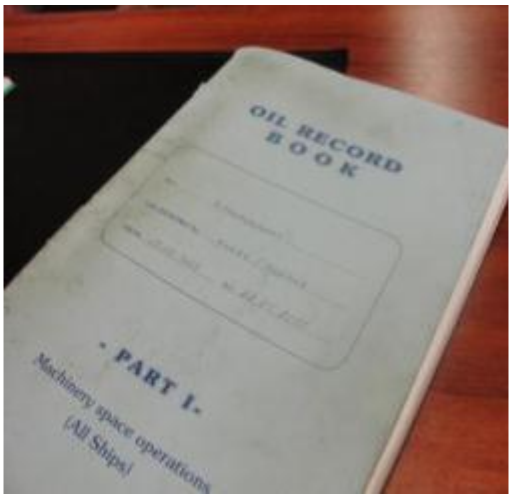


Figure 6: Oil Record Book

In accordance with MARPOL *Annex I—Oil*, all vessels must document the discharge and disposal of any onboard oily water and waste in an oil record book like the one pictured to the left in Figure 6 (Butt, 2007).

The book contains all of the discharges of bilge water and concentrations as well as the disposals of oily wastes. It is illegal to alter or falsify the recordings into the oil record book to be in accordance with discharge regulations. The discharge concentration is not allowed to exceed 15 PPM of oil (Lin, 2006). In addition, the ship is required to keep the ORBs for the past three years on the ship and must be able to show them if instructed. However, information in oil record books is not limited to information about the discharges. The first few pages state the required compliance from the ship as well as expectations of the oil record book. This is so that no single ship can state ignorance since all ships are required to keep an ORB. The first pages are word for word the regulations set forth by MARPOL. Following those pages are the instructions of how entries are expected to be labeled as well as what is required within different entries. Following the instructions are several pages of examples of entries in the correct format for reference. The way an ORB is completed is universal so that all ship inspectors in any country can easily read them. There are nine coded categories of information that is to be recorded in the oil record book. The categories with their corresponding letter code are as follows:

A. Ballasting or cleaning of oil fuel tanks
B. Discharge of dirty ballast or cleaning water from oil fuel tanks referred to under section (A)
C. Collection and disposal of oil residues (Sludge and other oil residues)*
D. Non-automatic discharge overboard or disposal otherwise of bilge water which has accumulated in machinery spaces*

E. Automatic discharge overboard or disposal otherwise of bilge water which has accumulated in machinery spaces*
F. Condition of the oil filtering equipment
G. Accidental or other exceptional discharge of oil
H. Bunkering of fuel or bulk lubricating oil
I. Additional operational procedures and general remarks

The codes that the group is most concerned with as it relates to the project are C, D, and E. The information is entered in with the letter code and item number (for more information, see Appendix E: ORB Key). Below in Figure 7, is an example oil record book entry. Note from left to right the date, the letter code, item number, and required signature. Those items will be within every entry. The other details of the entry depend on what code the entry is and what the item number demands.

The oil record book requires an entry whenever the oil water separator (OWS) is run. If the OWS is not run, but there is a decrease in volume of sludge or oily bilge water, then the ORB must indicate how it was removed.

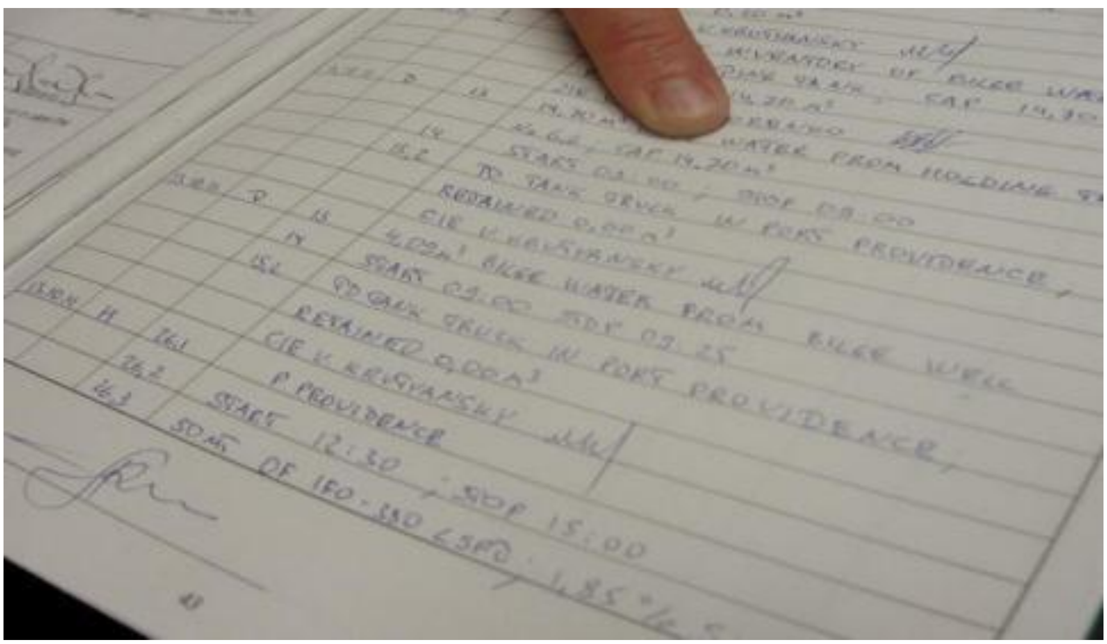


Figure 7: Oil Record Book Entry on a Ship

In the event that it was pumped off the ship, then it is the responsibility of the ship to denote that within the ORB and keep the receipt for the pumping. Figure 7, above, shows a record book entry for sludge being pumped off the ship at board. Figure 8, below, shows the receipt for this pumping.

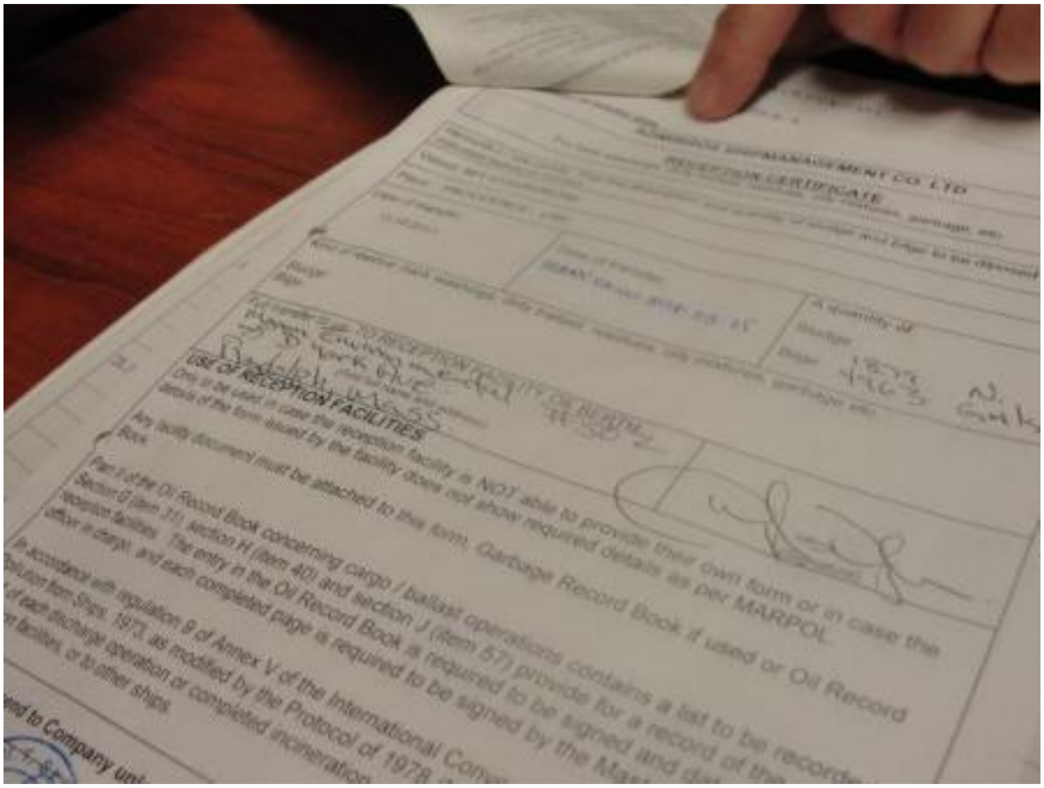


Figure 8: Receipt for Sludge Pumping

Currently, the United States uses the False Statement Act to prosecute those who falsify oil record book entries to cover up illegal discharges in the ocean (Berg, 2009). The problem with this is that if the False Statement Act is used to prosecute an offender that dumped in areas outside of U.S. jurisdiction, it typically does not hold. However, if it is found that there was a failure to keep an accurate oil record book, then regardless of in what country's jurisdiction the violation took place, any false entries found within the ORB are violations of MARPOL. There are several cases in which failure to keep accurate records or false records have led to prosecutions by the United States Department of Justice (for more information, see Section 2.6).

2.4.1.1 Engine Room Tank Sounding Log Book

In addition to the oil record book, ships are required to keep a written log of soundings of all the tanks in the engine room, including the waste water tank, fuel oil, and diesel oil service tanks (Raunekk, 2010). Soundings measure the level and volume for each tank. These entries are recorded daily. The engine and deck officers are required to sign off on each entry. Both of these officers, as well as the chief engineer, are required to sign off on each page. These signatures signify a written contract, but are not held to the same legal standards as the ORB.

2.4.2 On-board Waste Treatment

To minimize the amount of waste to dispose of on land or dump into the ocean, ships can take certain prevention measures on-board. There are several ways in which waste can be treated onboard a ship including compactors, comminuters, pulpers, shredders, incinerators as well as sewage treatment plants (Butt, 2007).

Most large ships have:

- A few incinerators
- A sewage treatment plant
- Compactors
- Can or glass crushers.

Table 4 lists examples of the different types of on-board waste management and explains what each method does to deal with different types of waste. It can be seen that certain methods are more ideal for some materials and wastes than others.

Table 4: Types of On-Board Waste Management

Compactors	Reduces the volume of solid waste, which makes this a good solution for ships that have limited with space.
Comminuters	Lacerates food bits into even smaller particles that can be released overboard.
Pulpers	Makes paper maché out of paper and cardboard, which makes the waste dischargeable.
Incinerators	Burns garbage that is unable to be recycled and is under MARPOL Annex V, which does not include hazardous wastes and the majority of plastics. The ash from incinerators is scooped out afterwards and is then sometimes thrown overboard.
Shredders	Grinds things that are unable to be treated with the previous methods (bone, metal, plastic, and glass).
Sewage treatment plants	Different types of treatment plants are used to treat wastewater. Types of plants can include membrane bioreactors (MBRs) and advanced water purification systems (AWP).

A ship may dispose of oil-waste using an incinerator (see Figure 9 one the next page) use the Oil Water Separator, or hold the oil waste on the ship until port to be pumped off.

Figure 9: Incinerator used for burning sludge

Incinerators, like the one pictured above in Figure 9, are commonly used to burn waste oil in the compartment shown. There is another compartment within this system that burns solid waste.

2.4.3 Oil Water Separator

Wastes that need to be treated include the oily residues that are a result of lubricating moving parts of the engines that are treated via an oil-water separator. On all ships exceeding 400 gross tons, which includes the majority of merchant marine vessels, a waste oil-water separator (OWS) is required in accordance with MARPOL (Lin, 2006). The OWS separates the oil content from the bilge water. Bilge water with an oil concentration of less than 15 ppm is legal to be discharged into the ocean in accordance with MARPOL. An oil water separator is pictured in Figure 10 below.

Figure 10: Oil Water Separator

The system needs to be manually turned on and run. Oily bilge water will run through the system. The silver disk above the box on the right in Figure 10 is the sensor to test the concentration of the bilge water. If it is over the desired maximum concentration of 15 PPM then the alarm sounds and the outflow valve shuts. Below in Figure 11, is a picture of the outflow valve.

Figure 11: Outflow Valve

Only water with less than 15 PPM oil is able to go into the top white pipe in Figure 11. If the connection between that top white and green valve looks like it has been opened often then it is usually a red flag that there has been tampering. In the event that it looks like the connection has been opened frequently, the USCG will open it up and see if there is oil inside the white pipe. If there is oil in the white pipe, then the OWS does not work or the sensor was tampered with. More recent OWS have a seal on this junction and it is evident when it is tampered with. There are many signs on and around the OWS and other pertinent systems to reduce non-compliance.

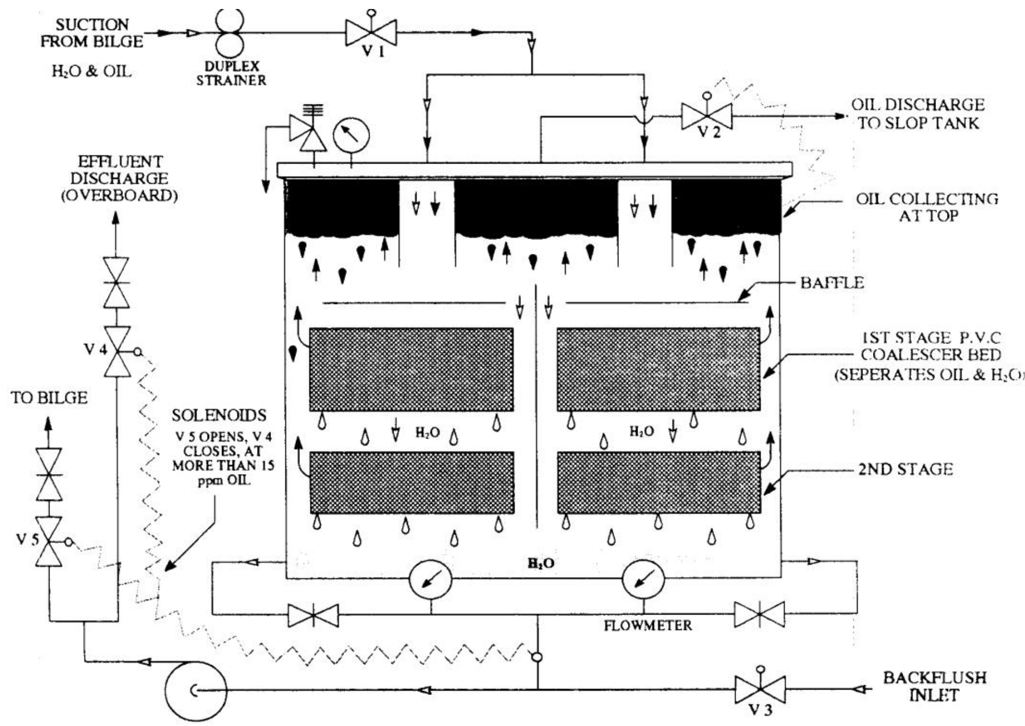


Figure 12: Schematic of an Oil-Water Separator (Haynes, 1996, p. 1).

The Schematic of an oil-water separator (as shown in Figure 12) shows the water being pumped from the bilge into a strainer. After going through the strainer and a valve, the oily water then flows into a large tank. The tank contains several zones that have oleophilic-hydrophobic polypropylene coalescer beads (Haynes, 1996). The term “oleophilic-hydrophobic” means oil loving and water repelling. This helps attract the oil and lets less concentrated water proceed through the separator. The oil droplets gather on these beads until eventually the droplets build up to be large enough to break off and float to the top of the tank (Haynes, 1996). The oil on the surface is skimmed off the top, and the remaining oily water, which is less concentrated with oil, then proceeds down the tank. This oily water goes through an oil content monitor, which tests the oil content. The water either goes back to the bilge, where it will be drawn to go through this process again or discharged as an effluent if the content is less that 15ppm in accordance with MARPOL.

2.4.4 Pollution Prevention System Specifications

The Marine Environmental Protection Committee (MEPC) outlined the guidelines for pollution prevention equipment in the machinery spaces of ships based on the regulations outlined in MARPOL Annex I (for more information, see Section 2.2.2 MARPOL/APPS) (Guidelines, 1992). There are two relevant regulations, MEPC.60(33) and MEPC.107(49). MEPC.60(33) is an older regulation that was adopted in 1992, and refers to older ships with older equipment. MEPC.107(49) is a newer regulation that was adopted in 2003 due to the advancement of technology that had occurred, and refers to newer ships, or older ships whose equipment has been upgraded (Revised Guidelines, 2003). These regulations intend to provide a uniform interpretation of the requirements of MARPOL Annex I, Regulation 14: to assist administrations to determine the appropriate design, construction, and operational parameters for pollution prevention equipment, to define test and performance requirements for pollution prevention equipment, and to provide guidance for installation requirements (International Maritime Organization, 2006, p.66).

MEPC.60(33) applies to installations fitted to ships on or after April 30, 1994 (Guidelines, 1992). It requires the installation of pollution prevention equipment, including 15ppm oil filtering equipment and 15ppm bilge alarms. This alarm can be seen in Figure 13. It also requires oil filtering equipment, which may include any combination of a separator, filter, or coalesce, as well as a single unit, the OWS, which is designed to produce oil content less than 15ppm. The oil filtering equipment is to function automatically, but provisions should be made for emergency manual control. The response time of the meter, which is the time that elapses between an alteration in the sample and the updated reading shown on the meter, should not exceed twenty seconds. The meter should also be fitted with an alarm device which can be set to operate automatically either to alert to crew or to operate the control valves whenever the oil content exceeds 15ppm or if the meter should ever fail to function. All the pictures in the above section (2.4.3 Oil Water Separator) are of a MEPC.60(33) oil water separator.

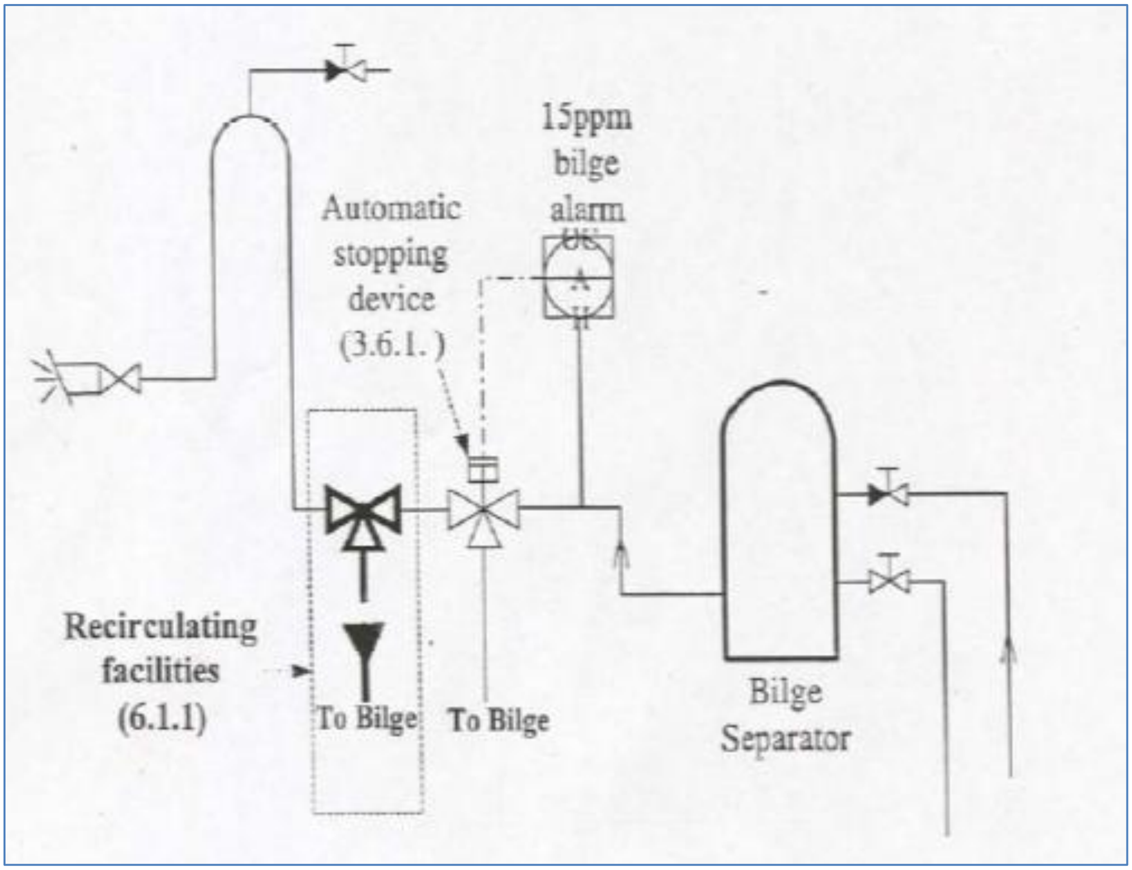


Figure 13: The OWS with 15PPM Alarm (Revised Guidelines, 2003).

MEPC.107(49) supersedes the recommendations of MEPC.60(33) (Revised Guidelines, 2003). This applies to system installations fitted to ships on or after January 1, 2005. The MEPC.107(49) has the same base requirements as MEPC.60(33); however it includes an additional system which must record the date, time, alarm status, and operating status of the 15ppm OWS. It must store data for at least eighteen months, and should be able to display or print a protocol for official inspections as required. Unlike the other systems on the ships, the inner-workings of this recording system are specifically not allowed to be changed in any way by the engineers on board the ship, so any access requires a seal to be broken. To see the different inspection protocols as they related to the type of OWS, see Appendix F: OWS Inspection Protocol.

2.5 Monitoring System Operation and Compliance on Vessels

Environmental protection is an important mission of the Coast Guard, and it is one that has been greatly affected by technological changes. In the past, this monitoring was done manually, and there was really no way to efficiently track ships that repeatedly disregarded the environmental regulations. With the advent of modern tank-monitoring systems came the opportunity to use computer systems to automatically monitor the waste disposed of by ships.

2.5.1 Current System

In the current system to prevent excessive waste entering the ocean from ships, the Coast Guard uses oil record books to evaluate and analyze the amount of waste that a ship may have discharged into the ocean or waterway. These books are manual logs kept on the ships that document all shipboard oil transfer and discharge, machinery space operations, the ballasting and cleaning of oil fuel tanks, disposal of oily residues, and discharge overboard or disposal of bilge water (Allain, 2010, p.73) (for more information, see Section 2.4.3 Oil Water Separator). When ships come into port, the Coast Guard is in charge of routinely reviewing the pollution prevention requirements that are contained in MARPOL Annex I. During these examinations, the ships check the oil record books for irregularities, which usually include out of order dates, missing pages, repetitive entries, or significant differences in recorded tank levels over time (p.74). In addition to checking the oil record books, the Coast Guard is also responsible for testing the waste management equipment on the ships, particularly the OWS if it is indicated in the oil record book that the specific ship regularly uses it. It is possible for ships to not use the OWS and only dispose of oil wastes at port via pumping.

There are various levels of MARPOL Annex I violations (Allain, 2010, p.74). Lesser violations include discrepancies on the IOPP certificate or missing signatures in the oil record book. These violations can be quickly corrected, and do not merit significant follow-up. However, more serious violations can result in the detention of the vessel (p.75). When it appears that these serious violations are willfully done, the Coast Guard pursues criminal charges. An example of a more serious violation is large quantities of sludge or oily bilge water being removed with no pumping receipt or OWS run. This would mean that the removed contents were potentially disposed of illegally and would warrant more investigation.

One of the consequences for repeated infractions is the installation of the Special Waste Oil Monitoring System (SWOMS) on the vessels (District of Connecticut, 2010). SWOMS is designed to electronically monitor and record all waste oil generation, as well as processing in the engine room. These data are automatically recorded at least every hour and are then electronically sent to shore side offices. Each ship is required to have a User

Manual in order to deal with potential problems with the SWOMS, although the ability to make manual changes is severely limited in order to prevent tampering.

Ionia Management, a Greek shipping company, was one of the companies charged with implementing procedures that require the comparison of data produced by the SWOMS with other shipboard waste management records, as well as procedures for maintenance and testing of the SWOMS (District of Connecticut, 2010). SWOMS training was also included in the charges and is now a requirement for all seafarers before they may join a vessel in the Ionia fleet (for more information on the Ionia Management Case, see Section 2.6.2 Ionia Management Case, p.32).

2.5.2 Problems with the Current System

There are several problems with the current system. The Coast Guard has long relied on manual inspections of vessels in order to check for environmental compliance and has only recently begun to use these digital systems. A manual inspection occurs when a PSCO boards a vessel and physically reviews the records while the vessel is in port. Reliance on manual inspections is time-consuming and is often not effective for regularly testing individual vessels. The SWOMS is an improvement, allowing for SWOMS data analysis to be performed in addition to manual inspections in order to more thoroughly check for day-to-day compliance. However, the Coast Guard currently does not have a good way to efficiently analyze the data that it is receiving from these ships. There is a significant amount of data to sift through, so the task remains very tedious and time-consuming as it must be completed manually. The data are often supplied to the Coast Guard via CDs in various file formats, complicating the analysis. This physical transaction means that the transmission of the data is slow. Some of the data are stored in real-time on the internet, but the Coast Guard is unable to credibly verify, or sometimes even access, these data due to system firewalls and incompatibilities of the systems.

In 2008, Ionia Management oversaw the installation of SWOMS on two of its ships that were being prosecuted (District of Connecticut, 2010). The system was incompatible with the existing Ionia communication system, which meant that the data could not be transferred properly. In addition, one of these ships was in West African waters, which further delayed the installation. Ionia was forced to purchase a new communications system for both of these vessels. Onboard evaluations on these ships showed some discrepancies between the data gathered by the SWOMS and the actual tank soundings, most likely because of the need for further calibration or timing of sampling, which is complicated by the pitching and rolling of ships on the water.

2.5.3 SWOMS Data Transmission Methods

Currently, there are two ways in which the data from the monitoring systems is able to be transmitted from the ship to the United States Coast Guard. One company has this data directly streamed to the internet, which the Coast Guard cannot access at USCG headquarters due to firewalls and internet security settings. A current method that one company practices has been able to avoid any interference with the Coast Guard firewalls and internet security settings and increase the overall accessibility of the data. The company uses the method of sending emails containing the data to the shipping company, whereupon the shipping company compiles CDs of all of the data. Since the method of sending CDs is the only process that the United States Coast Guard regularly uses, this is the method that was looked into most carefully. Details of the specific path of the data from the vessel to the United States Coast Guard are documented in Section 5.1.1: Data Transmission Method Analysis.

2.5.4 Financial Cost of Monitoring Systems

An important factor for ships in determining whether or not they will dispose of their waste properly is the cost. Since correct disposal of wastes can add up to be very expensive, there is a lack of incentive for commercial ships to properly dispose of their waste. The costs of complying with MARPOL are detailed in Table 5. Referring to the table, it shows that the record keeping of MARPOL compliance is one of the more expensive aspects of the environmental compliance costs. Many violations are in the record keeping of this information. The various sizes of ships have different estimated costs of compliance. The ships are measured in terms of their dead weight tonnage (DWT). As can be expected, larger ships have a greater cost of compliance.

Table 5: Costs of Complying with MARPOL. Source: OECD, 2003

Environmental Compliance Costs (USD)	66000 DWT Containership (4800 TEU)	150000 DWT Bulk Carrier	280000 DWT Oil Tanker
MARPOL Capital Costs (new/replacement cost, assumed equipment life span of 15 yrs.)			
Oily-Water Separator	10000	10000	10000
15 ppm. Monitor	1000	1000	1000
Incinerator	45000	45000	45000
Annex VI equipment (proposed)	50000	50000	50000
sub-total	56000	56000	56000
Capital Costs per year	4655	4655	4655
Capital Costs per day	13	13	13
Capital Costs per day (w/ Annex VI equipment)	24	24	24
Other MARPOL Fixed Costs (per year)			
Filters	2000	2000	2500
Maintenance OWS	1000	1300	1600
Maintenance OWS system pipes, valves and tanks	1530	1000	780
Maintenance Incinerator	1000	1000	1000
Maintenance Annex VI	1500	1500	1500
Record keeping	14700	14700	18000
Training	2850	2850	3600
MARPOL fixed costs per year	23080	22850	27480
MARPOL fixed costs per day	63	63	75
MARPOL fixed costs per day (w/ Annex VI)	67	67	79

There are high capital costs for the pollution prevention equipment, which includes the OWS, 15 ppm monitor, and incinerator. There are additional costs for the maintenance of this equipment, as well as for record keeping and training of employees. All of this adds up to a high cost of compliance, which makes illegal behavior look more appealing.

2.6 Non-Compliance Case Studies

When ships are caught participating in illegal behaviors that violate the MARPOL regulations in the waters of the United States and it is deemed a serious enough violation, then a case may go before the Department of Justice Environmental and Natural Resources Division. This includes cases where the violation did not occur in the United States but was discovered in the United States (e.g. falsified records). The United States Coast Guard, Office of Investigations and Casualty Analysis (COMDT CG-545) leads an investigation in cooperation with the DOJ to bring evidence to the case. In the event that the shipping company does not win the case, there is either a plea bargain or sentence. Many MARPOL violation cases end with a sentencing to a probationary period in which the shipping companies or specific ships are watched closely in all operations related to MARPOL. Some specific Department of Justice cases, like the ones mentioned below, have ended with verdicts that mandate environmental compliance systems like SWOMS (Special Waste Oil Monitoring System).

2.6.1 Clipper Trojan Case

One of the more relevant cases that involved breaking environmental regulations was that of the chemical carrier Clipper Trojan which can be seen Figure 14.

Figure 14: Clipper Trojan (Boat Nerd, 2007)

The Clipper Marine Services reached a settlement with the United State Department of Justice in response to the actions of certain crewmembers on the vessel (Clipper Group, 2007). After extensive negotiations, the United States Department of Justice dismissed 8 of the 11 counts, and Clipper Marine Services plead guilty to the 3 remaining counts. As a result, Clipper Marine Services agreed to pay $3.25 million and pay $1.5 million to the National Fish and Wildlife Foundation instead of the original $16.5 million fine.

The violations of CMS's Clipper Trojan included the Chief Engineer entering false information into the oil record book, which ultimately led to the installation of the Special Waste Oil Monitoring Systems (SWOMS) (Clipper Group, 2007). This system monitors the oil concentrations in the bilge waters. As a result of the negotiations with the Department of Justice, Clipper Marine Services agreed to cooperate with the United States Coast Guard in the development of

Special Waste Oil Monitoring Systems (SWOMS). Since the installation of the systems and the negotiations with the D.O.J., the U.S. Coast Guard and Independent Consultants have examined and/or audited many of Clipper's vessels, noting significant environmental management system improvements.

2.6.2 Ionia Management Case

The Greek shipping company, Ionia Management, was fined $4.9 million dollars for falsifying records and the obstruction of justice (DoJ, 2007). The company, a repeat offender, falsified records in the oil record book to hide waste oil being illegally dumped overboard from the M/T Kriton into international waters. The M/T Kriton can be seen below in Figure 15.

Figure 15: M/T Kriton (Shipspotting, 2010)

The shipping company was to appoint a "Special Master" in charge of monitoring the ship's records. In addition, another term of the sentence was that no ships under Ionia Management are permitted in U.S. ports without first installing a Special Waste Oil Monitoring System. The investigation showed at least 968 tons of oil-waste that was not accurately accounted for in the ship's oil record book. Not only did Ionia falsify oil record books, but the company also failed to submit accurate environmental compliance checklists to the U.S. Coast Guard, which was part of the company's probation from a previous 2004 conviction. U.S. Attorney O'Connor went on record to say, referring to this particular case: "We hope and expect that this prosecution and the stiff sentence imposed sends a clear message to all who intend to pollute the world's waters that such a conduct will not be tolerated" (p.1). The convictions in these cases are to warn other companies as to why they should follow the regulations set forth to minimize illegal ocean dumping.

Currently, Ionia Management submits evidence of their environmental compliance from the Special Waste Oil Monitoring Systems within 40 days of the end of the month for which the data was collected (District of Connecticut, 2010). The company installed a fully functional SWOMS system on the M/T Fidias that was fully commissioned on April 18, 2009. Since the appointed Special Master's first hearing in December 2008, the M/T Fidias and M/T Theo T have submitted waste oil generation data on a monthly basis.

Since this company was involved in several cases of non-compliance, there are extensive measures mandated for the company to follow as a result of their actions in violation of MARPOL.

2.7 Databases

Databases are the basis on which most modern data storage and processing systems rely. At its core, a database is "a collection of logically related data stored together in one or more files" (Geraci et al., 1990, p.23). This definition allows for a significant degree of latitude in what is classified as a database. From a computer science point of view, a database could be as simple as a single sequential-access file, or as complex as the 12-million-object database in use by Alexa Web Services (Amazon Web Services, 2011).

2.7.1 Database Management Systems

A database management system is "[a] computer system involving hardware, software, or both that provides a systematic approach to creating, storing, retrieving and processing information stored in a database" (Geraci, et al., 1990, p.23). These systems come in a variety of forms, and can vary significantly in their abilities. The most common interface seen is the simple web query, for example, when searching for products on a website.

There are many database management systems available on the market. The three major players in the market are Oracle, IBM, and Microsoft, which together accounted for roughly 85% of total market share in relational database management software in 2006 (Gartner, Inc., 2006). The most readily available system to this project team is Microsoft Access, which was chosen by the Coast Guard as the platform for this project. Access is well-suited for this project due to its availability and its non-reliance on web servers during the development stage.

2.7.2 Access Database & Interface

Microsoft Access 2007 has an interface similar to that of the rest of the Office 2007 suite. It offers full control over creating and modifying all the various parts of a database. In addition, Access offers ways to more easily create graphical user interfaces (GUI), which often need to be created separately with other database systems. There are template interfaces that are highly customizable and the particular setup of any element can be modified through graphical menus, or by directly modifying the Visual Basic for Applications (VBA) code.

Access databases involve the use of several related objects. The first and most basic is the table, which is where the actual data are stored. Each entry into the database is called a record, and each record can have multiple fields, which are represented by the rows and columns of the database. The next object is a form, which is what the developer uses to set up the GUI or "front-end" of the database where the user can enter, modify and view data in the tables. In order to organize that data, queries can be used to select data from the tables and forms based on certain parameters or calculations. This built in functionality is often easier than coding in VBA to get the same result for the user. Lastly, queries can be used to generate reports, which can be used to organize the data in a more professional and readable form for reporting and printing results.

With the ability to code or write queries for different "events", designers can define behaviors that the database will take when actions like mouse clicks, screen changes, or other events occur. The database core can thereby monitor the users' interactions with the database and update fields on the interface or change internal information actively during users' sessions.

The other major advantage is offline development. Many database systems rely on dynamically generated web documents for their interfaces, as well as ties to database files on servers that store the actual data. With an Access database, during development the data can all be stored locally up to 2 gigabytes, a significant amount of data and more than enough space for development purposes. Access integrates with Microsoft SQL server, and can be converted from an offline database to a live, networked database without major modifications.

2.7.3 Example Database

The most familiar large-scale database is probably the one that powers the e-commerce giant, Amazon.com. Amazon has a full storefront that includes product information, comparison tools, a payment system, and other user-oriented features. The site's pages are loaded with dynamic content that is "pulled" in real time from its database.

When shopping, typically one might search for a specific product, or browse through a category of products. Amazon offers a simple search form that takes keywords and searches the product listing for those words. The search form gives users the ability to enter in a query and the results are displayed on the page below. Users can also typically choose additional criteria through which a search can be narrowed.

If a shopper wants more information about a product, they can go to its product page, which is again dynamically generated. The web server automatically uses many different pieces of information to create a page full of relevant information. The server queries the database and shows a product image and product information. When the shopper finds what they are looking for and are ready to check out, they are prompted to enter in information.

A frequent shopper may already have their information in the database, and simply needs to choose which address to ship to. Otherwise, new users are presented with a form to fill out identification and billing information. Once past that stage, customers are presented with a digital receipt, and the option to print their receipt. If they do want print their receipts, customers can open a different page which is a report detailing their transaction. The report view is designed for efficient viewing and printing. At this point, the database has provided data, received data, and processed a transaction. This same type of system, although differently-implemented, will be used to acquire, store, and analyze ship data to produce reports that will help the Coast Guard to better track repeat-offending vessels.

2.8 Summary

It is important to consider how MARPOL regulates ship generated oil-waste, as violations to this treaty are what often lead to mandating the installation of Special Waste Oil Monitoring Systems (SWOMS). Many ships with SWOMS have been repeat offenders, and the monitoring systems are to deter shipping companies from tampering with records. The transmission of SWOMS data to a database will dictate how quickly the Coast Guard will be able to monitor waste composition and disposals. Therefore, the transmission of data as well as which data are being recorded will be used to determine the feasibility of developing a database to reference ships' environmental compliance. Looking at the process to review current SWOMS data will give insight into how we can make the data more readily available for Coast Guard monitoring. Based on information collected in interviews and direct observation, we must determine how the Special Waste Oil Monitoring System (SWOMS) works. We will then consider the opinions of individuals currently tasked with reviewing SWOMS data to gain a deeper understanding of what a database should include beyond the data collected directly by SWOMS. This will allow for the creation of a database that the Coast Guard can then use to effectively monitor pollution.

Overall, there are many things that can yield non-compliant behavior as seen previously in this chapter. However, it is very difficult to determine whether a vessel is adhering to its probation and abiding by all MARPOL regulations. To do this, a comprehensive system that stores, analyzes, and reports data from these vessels must be developed. This system will look at several different factors to determine whether the vessel is in compliance with MARPOL and its probation. The next chapter will discuss how the development process of this system will be carried out.

3 Goals and Deliverables

The goal of this project is to develop a system to receive, store, compile and analyze data from shipboard environmental management systems. The system of data collection and analysis is designed to facilitate investigations into environmental crimes committed by repeat offenders. In order to achieve this goal, we produced four deliverables to give to the Coast Guard. These deliverables are as follows:

1. A standardized method for sending Special Waste Oil Monitoring System (SWOMS), sounding log book, and oil record book (ORB) data to the Coast Guard;
2. An Environmental Compliance Plan Database that extracts, compiles, and organizes data output from shipboard environmental management systems;
3. A method to analyze environmental management data and highlight inconsistencies;
4. A "user manual" for the Environmental Compliance Plan System that includes information on how a user would navigate the database and perform specific tasks as well as information on how to modify the software as needed in the future.

4 Methodology

4.1 Shipboard Oil Waste Management

In order to comprehend how to properly read and analyze data about oil waste streams on vessels, it is important to learn about the current system used to track shipboard oil waste. Since this project focuses on poor oil-waste management, it is necessary to establish what constitutes acceptable versus inadequate oil-waste management. In order to better understand this waste management, additional information is researched, including:

- How oil moves through a ship
- What specific equipment/documents ship inspectors examine
- How oil record books are used to track oil waste

4.1.1 Oil Movement Through Ships

Understanding the causes of oil accumulation on ships and how this waste oil is processed helps in determining where and how oil management aboard ships can fail to meet established standards. To accomplish this, archival background research is done regarding these systems, supplemented by field research aboard an oil tanker (see Appendix G: Ship Tour Summary for a detailed account). In order to understand the data received from oil-waste monitoring on ships, the systems on the ship that are referred to in oil record books (ORBs) and Special Waste Oil Monitoring Systems (SWOMS) must be researched. These systems include:

- Bilge Tanks
- Sludge Tanks
- Oil Water Separator
- Incinerator
- Bilge Pump

Throughout the research and observation of these systems, two specific factors were considered: (1) How this system is intended to be run and (2) How it can be tampered with in order to bypass oil treatment systems.

4.1.2 Oil Record Book Analysis

Oil record books are an essential element in oil waste monitoring and are often used as evidence in cases of non-compliance. It is important to determine the organization of ORBs in order to recognize anomalies within them. To be able to identify what constitutes an anomaly within the ORB, several tasks are completed:

- Interview with ship inspectors

- Thorough analysis of the ORB codes and items
- Looking through ORBs

Speaking with United States Coast Guard ship inspectors is helpful for understanding the oil record book. They are able to go through the entries of an oil record book onboard a vessel with the chief engineer present and explain the different codes and item numbers (See Section 2.4.1 for more information). Once familiarized with the ORB and its organization, looking through example ORB entries is easier and more informative.

4.2 Standardized Submission Process

One of the deliverables of this project is to create a plan for a standardized method of sending Special Waste Oil Monitoring System (SWOMS), sounding log book, and oil record book (ORB) data to the Coast Guard. This is done with the creation of standardized formats of the data, being templates in this case, as well as determining a method and timeline for the data submission. This standardized submission process will allow for smooth integration of data into the database, but is able to stand alone and will still increase efficiency of data analysis.

4.2.1 Create Templates for Standard Submission Process

In order to integrate data into the database, it is necessary to use a standardized form that allows uniform data to be entered into the database. The need for this uniformity was solved through the development of data submission templates. The templates are designed to match the data entry requirements of the Environmental Compliance Plan Database. One template is designed for the incoming data that is sent through automatically generated report emails from the SWOMS. Another template is designed to allow for the entry of ORB data into the database. These templates are designed to easily import data into corresponding tables in the database.

An additional template takes the variations in ship design into account. This specific template relates generic tank identifiers found in the SWOMS and ORB templates (e.g. "Tank01") to specific tanks aboard a given ship (e.g. starboard bilge tank). The use of standard templates simplifies imports of SWOMS data into the Environmental Compliance Plan Database.

4.2.2 Determine Data Submission Method and Timeline

An important part of the standardized submission process is the ability to receive the data. The method for receiving the data that is most appealing to the Coast Guard is email. Emails would need to be sent from the vessels onboard environmental management system to the Coast Guard as well as emails containing the oil record book, sounding book, and any completed data spreadsheets required of the vessel.

Once data are received by the Coast Guard, part of the SWOMS data collection process is data conversion. SWOMS onboard a vessel converts sensor data into usable data points and transmits the collected information via email. These emails are in plain text, which makes them very conducive to basic read/write operations. In order to convert the emails to a format readily readable by the database management software (Access), a parsing program is created that separates the data in a standardized email into the same comma-separated-value format used in the SWOMS template previously mentioned.

In addition to the method and the data parsing that takes place, a timeline is needed for the data submission. Research into the current timeline of data submission and the drawbacks of that timeline are needed. This helps to determine a better timeline for receiving all data from the SWOMS as well as the vessel or shipping company.

4.3 Verifying Non-Compliance

Tracking non-compliant activity in vessels on probationary periods is an important aspect to the database. This means that it is very important to determine how to identify non-compliance based on numerical trends. To accomplish this, the SWOMS data are studied, both in the format currently sent in by the shipping companies as well as their initial format. In studying the data, the meaning of the numerical values and the significance of their changes over time are examined thoroughly. This knowledge, combined with the ORB research, helps to determine what inconsistencies between SWOMS and ORB data are deemed unacceptable.

4.3.1 Special Waste Oil Monitoring System Data

Currently, the SWOMS data spreadsheets that are sent by the shipping companies tend to be hard to interpret even for those who are familiar with oil-waste systems. Learning how the current method of analysis is used to make conclusions from the data can help to determine what changes are necessary to make the layout of the data user-friendly for auditors. In looking at the data sets, there were several questions to be answered:

- Are the data clearly labeled?
- Are the columns in a logical order?
- Is all of the given information necessary?
- Is there information missing that might be necessary?

Answers to these questions can help to identify a layout that will better portray the data's meaning. In addition, these answers can determine whether there is information that should be omitted or added to what is currently recorded.

4.3.2 Criteria of Unacceptable Deviation

One of the uses of the database being created is to cross-reference the SWOMS data with the ORB. If any anomalies are detected when cross-referencing the two data sets, an alert is set off. Interviews with auditors and investigators are used to determine how large of a deviation in data should set off an alert. These interviews also help determine other data trends to look for and what kinds of reports the database needs to produce.

4.4 Database Construction

To create a database for this project, there were several elements to the process. The outline of the database and plans need to be developed followed by the actual creation of the database. Once the database is created, testing and troubleshooting can take place. The following sections describe the procedure for creating the database.

4.4.1 Planning for Development

Using the templates as a guideline for the data structure, the next step was to begin the design of the database. Access was chosen as the platform of this project by the Coast Guard. Access is well-suited for this project due to its availability and its non-reliance on web servers during the development stage (See Section 2.7.2 for additional information on Access). This database will be able to collect, organize and display SWOMS and sounding book data, as well as a list of all vessels that use SWOMS and all tanks on those vessels and allow the user to modify this data where appropriate. Additionally, the database needs to be able to analyze that data for anomalies. By taking user input along with built-in calculations and queries, the database will be able to generate reports that show anomalies and discrepancies in this data broken down by multiple variables in order to assist Coast Guard analysts in rapidly assessing whether a vessel is in compliance.

4.4.2 Development of the Database

The first step in this plan is to create the tables to store all the data, which are based on the previously discussed templates. Once the tables are completed, forms to display the data stored in them will be created so the user can view the data. Additional forms will have to be created to allow the user to add more data as well as modify current data. Then a main menu form will be created to link the various forms together. An additional form to allow the user to generate reports will be the last form to be constructed, followed by the reports themselves. Once all the forms and reports are constructed, the queries that will be used to display data on the forms and to generate data for the reports will have to be created.

4.4.3 Testing

Once all queries are written database testing will be begun. Extensive testing will be conducted to check for any errors in the code and queries, as well as to check for missing features. Additional error checking will be written into the code to account for issues discovered during testing, and all other features will be either fixed or cut from the database based on feedback and results of the testing.

4.5 User Manual

In conjunction with the development of the data transmission system and database, a user manual is written. First, the manual explains how to use the parsing program. It then addresses the database, giving step-by-step instructions for how to navigate the database in order to add, edit, and analyze the data. Finally, the manual addresses future maintenance of the parsing program and the database. The code of the parsing program and the queries of the database are explained such that a technical person should be able to perform any necessary troubleshooting or incorporate additional functions.

4.6 Summary

Interviews, analysis, and case studies were used in order to meet the overall goal of creating a data acquisition, storage, and analysis system. The database is made to meet the needs of the US Coast Guard. At the end of this project, the Coast Guard will be given a thorough standard submission proposal for future non-compliance cases as well as a fully-functional database that is able to store and analyze SWOMS data with the capability to cross-reference with oil record book data. Any potential modifications or uses for this database that became apparent throughout the course of this project are also included as future recommendations.

5 Results and Analysis

The goal of this project was to develop a database and system that would be able to receive, store, compile and analyze data from shipboard environmental management systems. To achieve this goal, the current system of data analysis was studied which included the transmission method of the data as well as the format of the SWOMS data and the oil record book that are sent from shipping companies to the Coast Guard. Once there was extensive knowledge of the current system's capabilities and drawbacks it was determined what was needed to make a more efficient system. To do this, templates for the database as well as any manual data entry were created in addition to a new standard submission process for the SWOMS data. This new process includes new methods, timelines, and ways of receiving and extracting data from the SWOMS data transmissions. In addition to this a database was created that is able to store, organize and analyze all data received from SWOMS transmissions. Achieving this goal included four specific deliverables. These deliverables are a standardized method for sending SWOMS and ORB data to the Coast Guard, a database that extracts, compiles, condenses, and compares data output from shipboard environmental management systems, a method to analyze environmental management system data and highlight inconsistencies, as well as a user manual for all technical elements of the system. This chapter will thoroughly illustrate the steps taken to produce these deliverables.

5.1 Current System for Data Analysis

The first element that was looked at is how the system for the current data analysis works. In determining how the current system for receiving SWOMS data from ships, key problems can be identified to work with in order to make it more favorable for both the shipping companies and the United States Coast Guard.

5.1.1 Data Transmission Method Analysis

Currently, the SWOMS data travel through many hands before it gets to the United States Coast Guard. In addition, both of the two companies that currently implement SWOMS do it a little differently. One company has this data directly streamed to an internet site to which the Coast Guard cannot access this data from the USCG headquarters. With SWOMS being a newer system and each company's environmental compliance plan being different, submitting the data directly to the Coast Guard may not have been a legal requirement for the company. Since the Coast Guard currently is not able to easily access the information from that company, this study of data transmission is particular to the other shipping company that currently has four ships with fully functional SWOMS and sends the data to the United States Coast Guard regularly. The general path of the data for the specific company studied is portrayed below in Figure 16:

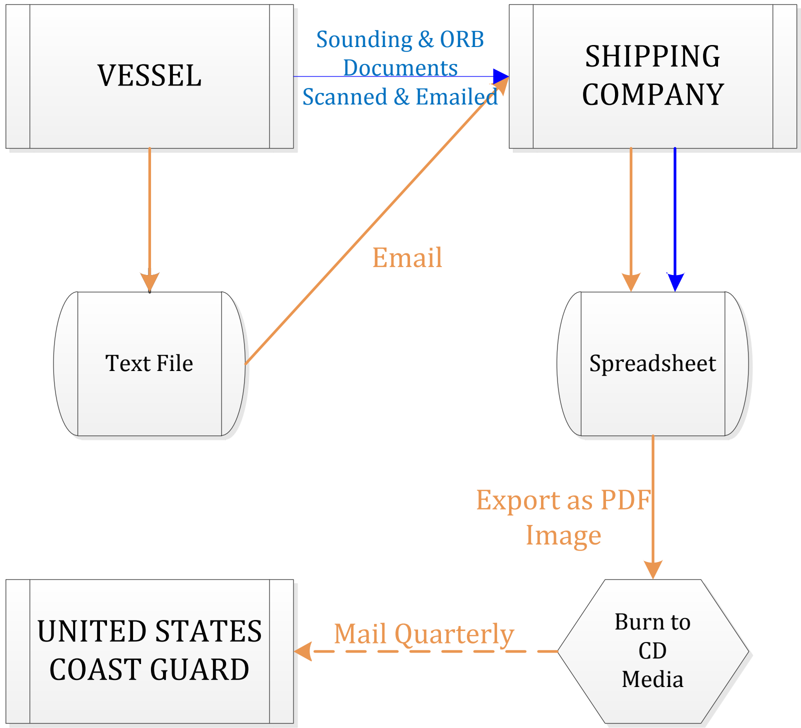


Figure 16: The Current Data Transmission Process

Figure 16 shows the path of the SWOMS data from the ship to the United States Coast Guard. The data do not directly go from the ship to the Coast Guard. In fact, they are sent in emails as a text file once a day to the shipping company. At the end of each month, the shipping company compiles the data from the text file emails into a spreadsheet with data for the entire month. The company does this for each ship with a SWOMS onboard. The spreadsheets are then converted to PDFs, which are compiled onto a CD with a scanned copy of the oil record book for the specific time period. The CDs contain one month of data, but are sent quarterly. This means several CDs arrive at the Coast Guard at once. The problem with this is that cumbersome amounts of data are being given at once, and the Coast Guard has a short window of time to act legally to extend probation if in fact the data proves noncompliance.

5.1.1.1 CD Contents

The CDs sent to the Coast Guard contain significant amounts of data at once but are organized specifically. When a CD is loaded and opened, there is a folder for the month. Within the folder for that month are separate folders for each individual ship as can be seen in Figure 17 below.

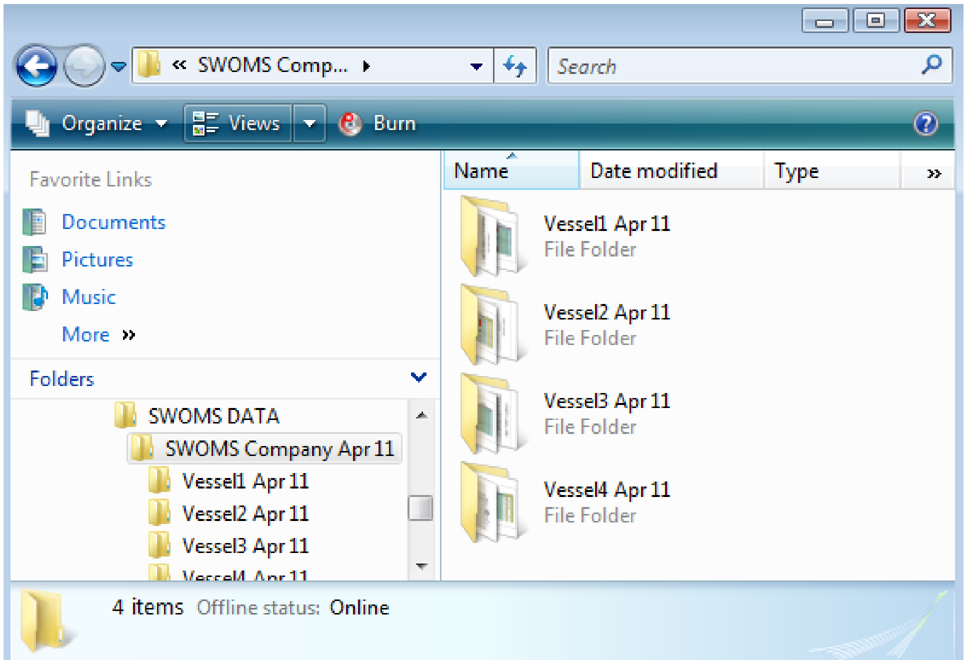


Figure 17: Contents of April Folder

In the left menu of Figure 17, it can be seen that the title of the folder that contains the ship folders is "SWOMS Company Apr11" and within that folder are the four folders pictured to the right. Within each folder is the SWOMS data along with the scanned images of the oil record book for each individual ship. Figure 18, below, shows the contents of each vessel folder included on the CD.

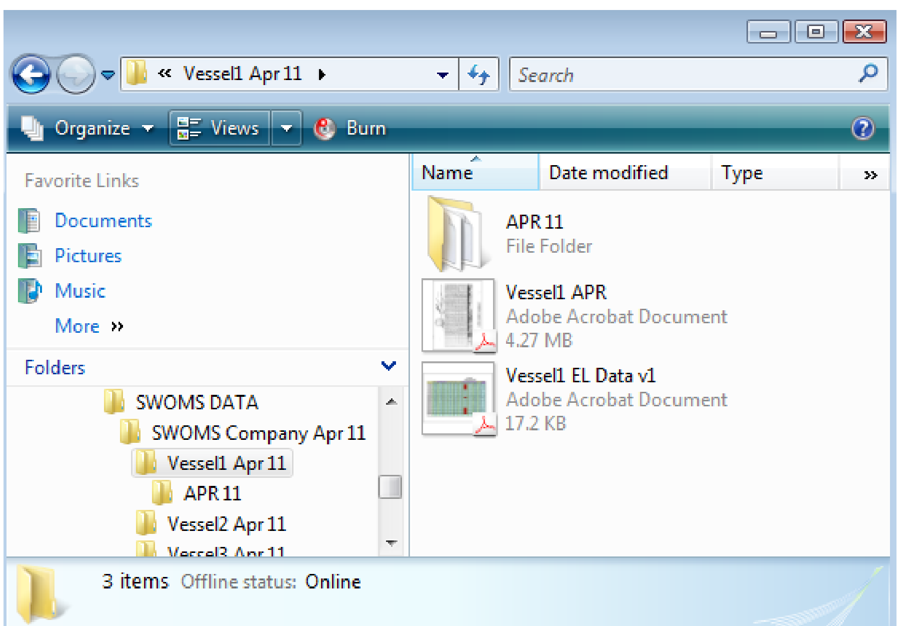
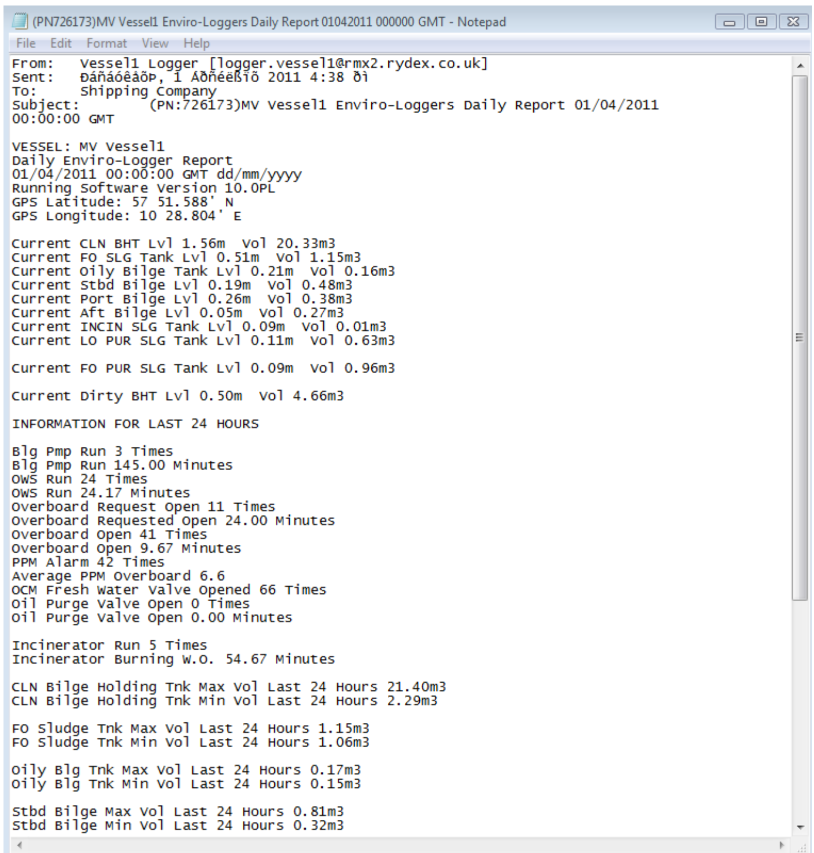


Figure 18: Contents of Vessel Folder

Within the folder of the vessel is a PDF of the ORB for the denoted month, the PDF spreadsheet of the SWOMS for the month, as well as another folder containing emails. Currently, the analysts look at the SWOMS spreadsheet by hand in the event of an anomaly or an unexplained trend. If looking at the ORB and the SWOMS spreadsheet still does not give the analysts the answer they are looking for, they will look at the individual emails. The individual emails are sent each day in the same format. These emails contain important information such as the location of the ship, the amount of time the OWS was run, the number of times the bilge alarm was sounded, the instantaneous levels and volumes for all tanks that SWOMS monitors, as well as the maximum and minimum tank volumes over the past 24 hours. An example of one of these emails can be seen below, in Figure 19.



(PN726173)MV Vessel1 Enviro-Loggers Daily Report 01042011 000000 GMT - Notepad

File Edit Format View Help

```
From:    Vessel1 Logger [logger.vessel1@rmx2.rydex.co.uk]
Sent:    Ðáñáóêåõþ, 1 Áðñéëßïõ 2011 4:38 ðì
To:      Shipping Company
Subject:        (PN:726173)MV Vessel1 Enviro-Loggers Daily Report 01/04/2011
00:00:00 GMT

VESSEL: MV Vessel1
Daily Enviro-Logger Report
01/04/2011 00:00:00 GMT dd/mm/yyyy
Running Software Version 10.0PL
GPS Latitude: 57 51.588' N
GPS Longitude: 10 28.804' E

Current CLN BHT Lvl 1.56m  Vol 20.33m3
Current FO SLG Tank Lvl 0.51m  Vol 1.15m3
Current Oily Bilge Tank Lvl 0.21m  Vol 0.16m3
Current Stbd Bilge Lvl 0.19m  Vol 0.48m3
Current Port Bilge Lvl 0.26m  Vol 0.38m3
Current Aft Bilge Lvl 0.05m  Vol 0.27m3
Current INCIN SLG Tank Lvl 0.09m  Vol 0.01m3
Current LO PUR SLG Tank Lvl 0.11m  Vol 0.63m3

Current FO PUR SLG Tank Lvl 0.09m  Vol 0.96m3

Current Dirty BHT Lvl 0.50m  Vol 4.66m3

INFORMATION FOR LAST 24 HOURS

Blg Pmp Run 3 Times
Blg Pmp Run 145.00 Minutes
OWS Run 24 Times
OWS Run 24.17 Minutes
Overboard Request Open 11 Times
Overboard Requested Open 24.00 Minutes
Overboard Open 41 Times
Overboard Open 9.67 Minutes
PPM Alarm 42 Times
Average PPM Overboard 6.6
OCM Fresh Water Valve Opened 66 Times
Oil Purge Valve Open 0 Times
Oil Purge Valve Open 0.00 Minutes

Incinerator Run 5 Times
Incinerator Burning W.O. 54.67 Minutes

CLN Bilge Holding Tnk Max Vol Last 24 Hours 21.40m3
CLN Bilge Holding Tnk Min Vol Last 24 Hours 2.29m3

FO Sludge Tnk Max Vol Last 24 Hours 1.15m3
FO Sludge Tnk Min Vol Last 24 Hours 1.06m3

Oily Blg Tnk Max Vol Last 24 Hours 0.17m3
Oily Blg Tnk Min Vol Last 24 Hours 0.15m3

Stbd Bilge Max Vol Last 24 Hours 0.81m3
Stbd Bilge Min Vol Last 24 Hours 0.32m3
```

Figure 19: Example of SWOMS Email

The email shows the amount of information sent from the SWOMS to the shipping company. These emails contain more information than is currently displayed in the SWOMS spreadsheets (for more information, see Section 5.1.2 Special Waste Oil Monitoring System Data Format & Analysis).

5.1.1.2 Problems with the Current Transmission Method

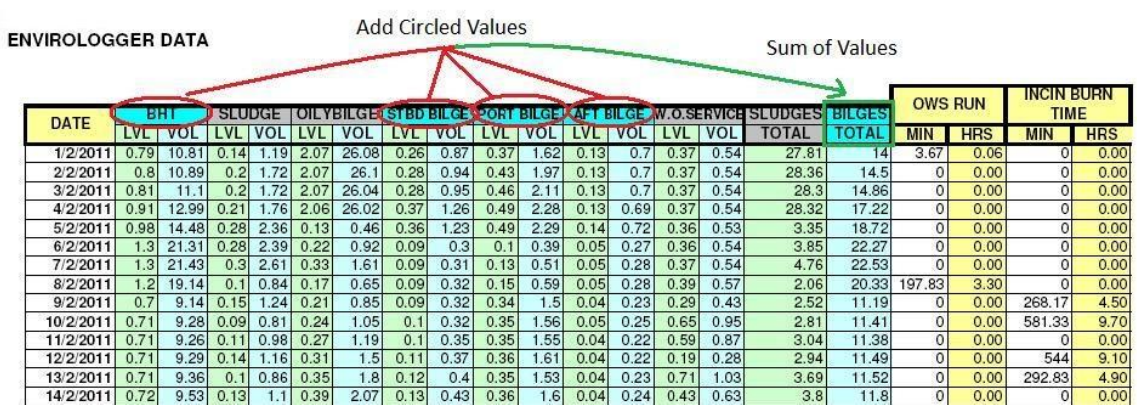
Interviews and analyses of the transmission method showed that there are a few problems in the current system that need to be revisited. The main problem with the current transmission method is the time that it takes for the information to get from the ship to the Coast Guard. There is the potential for it to be over three months between the time the data was originally documented and the time it is finally able to be analyzed at the Coast Guard, which is problematic considering the short window of opportunity for prosecuting criminals. This is something that was taken into consideration in the development of a new "standard submission process".

5.1.2 Special Waste Oil Monitoring System Data Format & Analysis

As was mentioned in the previous section, the Special Waste Oil Monitoring System data are received in two forms: the raw data within the emails and the spreadsheet PDF. Currently, the format of the data that is most frequently referred to is the spreadsheets. However, the analysis of the SWOMS spreadsheets is tedious and often produces inconclusive results regarding ship compliance.

5.1.2.1 Special Waste Oil Monitoring System Spreadsheet Analysis

The first step of analyzing the SWOMS data is to verify that the sludge and bilge totals are accurate. In order to do this, the volumes of the various tanks must be summed together. To calculate the bilges total, the volumes of the bilge holding tank (BHT), starboard bilge tank (STDB Bilge), port bilge tank, and aft bilge tank must be summed up as is shown in Figure 20 below.



DATE	BHT		SLUDGE		OILYBILGE		STBD BILGE		PORT BILGE		AFT BILGE		W.O.SERVICE		SLUDGES	BILGES	OWS RUN		INCIN BURN TIME	
	LVL	VOL	LVL	VOL	LVL	VOL	LVL	VOL	LVL	VOL	LVL	VOL	LVL	VOL	TOTAL	TOTAL	MIN	HRS	MIN	HRS
1/2/2011	0.79	10.81	0.14	1.19	2.07	26.08	0.26	0.87	0.37	1.62	0.13	0.7	0.37	0.54	27.81	14	3.67	0.06	0	0.00
2/2/2011	0.8	10.89	0.2	1.72	2.07	26.1	0.28	0.94	0.43	1.97	0.13	0.7	0.37	0.54	28.36	14.5	0	0.00	0	0.00
3/2/2011	0.81	11.1	0.2	1.72	2.07	26.04	0.28	0.95	0.46	2.11	0.13	0.7	0.37	0.54	28.3	14.86	0	0.00	0	0.00
4/2/2011	0.91	12.99	0.21	1.76	2.06	26.02	0.37	1.26	0.49	2.28	0.13	0.69	0.37	0.54	28.32	17.22	0	0.00	0	0.00
5/2/2011	0.98	14.48	0.28	2.36	0.13	0.46	0.36	1.23	0.49	2.29	0.14	0.72	0.36	0.53	3.35	18.72	0	0.00	0	0.00
6/2/2011	1.3	21.31	0.28	2.39	0.22	0.92	0.09	0.3	0.1	0.39	0.05	0.27	0.36	0.54	3.85	22.27	0	0.00	0	0.00
7/2/2011	1.3	21.43	0.3	2.61	0.33	1.61	0.09	0.31	0.13	0.51	0.05	0.28	0.37	0.54	4.76	22.53	0	0.00	0	0.00
8/2/2011	1.2	19.14	0.1	0.84	0.17	0.65	0.09	0.32	0.15	0.59	0.05	0.28	0.39	0.57	2.06	20.33	197.83	3.30	0	0.00
9/2/2011	0.7	9.14	0.15	1.24	0.21	0.85	0.09	0.32	0.34	1.5	0.04	0.23	0.29	0.43	2.52	11.19	0	0.00	268.17	4.50
10/2/2011	0.71	9.28	0.09	0.81	0.24	1.05	0.1	0.32	0.35	1.56	0.05	0.25	0.65	0.95	2.81	11.41	0	0.00	581.33	9.70
11/2/2011	0.71	9.26	0.11	0.98	0.27	1.19	0.1	0.35	0.35	1.55	0.04	0.22	0.59	0.87	3.04	11.38	0	0.00	0	0.00
12/2/2011	0.71	9.29	0.14	1.16	0.31	1.5	0.11	0.37	0.36	1.61	0.04	0.22	0.19	0.28	2.94	11.49	0	0.00	544	9.10
13/2/2011	0.71	9.36	0.1	0.86	0.35	1.8	0.12	0.4	0.35	1.53	0.04	0.23	0.71	1.03	3.69	11.52	0	0.00	292.83	4.90
14/2/2011	0.72	9.53	0.13	1.1	0.39	2.07	0.13	0.43	0.36	1.6	0.04	0.24	0.43	0.63	3.8	11.8	0	0.00	0	0.00

Figure 20: Finding the SWOMS Bilge Total

Notice that the values that need to be summed are not placed next to each other, meaning that summing the values is not intuitive. Similarly, to calculate the sludge total, the volumes of the sludge tank, the oily bilge tank, and the waste oil service tank (W.O. Service) must be summed up as is shown in Figure 21 below. Once these totals have been calculated, the values can be compared with the given total values.

ENVIROLOGGER DATA

Values Summed

DATE	BHT		SLUDGE		OILYBILGE		STBD BILGE		PORT BILGE		AFT BILGE		W.O.SERVICE		SLUDGES	BILGES	OWS RUN		INCIN BURN TIME	
	LVL	VOL	LVL	VOL	LVL	VOL	LVL	VOL	LVL	VOL	LVL	VOL	LVL	VOL	TOTAL	TOTAL	MIN	HRS	MIN	HRS
1/2/2011	0.79	10.81	0.14	1.19	2.07	26.08	0.26	0.87	0.37	1.62	0.13	0.7	0.37	0.54	27.81	14	3.67	0.06	0	0.00
2/2/2011	0.8	10.89	0.2	1.72	2.07	26.1	0.28	0.94	0.43	1.97	0.13	0.7	0.37	0.54	28.36	14.5	0	0.00	0	0.00
3/2/2011	0.81	11.1	0.2	1.72	2.07	26.04	0.28	0.95	0.46	2.11	0.13	0.7	0.37	0.54	28.3	14.86	0	0.00	0	0.00
4/2/2011	0.91	12.99	0.21	1.76	2.06	26.02	0.37	1.26	0.49	2.28	0.13	0.69	0.37	0.54	28.32	17.22	0	0.00	0	0.00
5/2/2011	0.98	14.48	0.28	2.36	0.13	0.46	0.36	1.23	0.49	2.29	0.14	0.72	0.36	0.53	3.35	18.72	0	0.00	0	0.00
6/2/2011	1.3	21.31	0.28	2.39	0.22	0.92	0.09	0.3	0.1	0.39	0.05	0.27	0.36	0.54	3.85	22.27	0	0.00	0	0.00
7/2/2011	1.3	21.43	0.3	2.61	0.33	1.61	0.09	0.31	0.13	0.51	0.05	0.28	0.37	0.54	4.76	22.53	0	0.00	0	0.00
8/2/2011	1.2	19.14	0.1	0.84	0.17	0.65	0.09	0.32	0.15	0.59	0.05	0.28	0.39	0.57	2.06	20.33	197.83	3.30	0	0.00
9/2/2011	0.7	9.14	0.15	1.24	0.21	0.85	0.09	0.32	0.34	1.5	0.04	0.23	0.29	0.43	2.52	11.19	0	0.00	268.17	4.50
10/2/2011	0.71	9.28	0.09	0.81	0.24	1.05	0.1	0.32	0.35	1.56	0.05	0.25	0.65	0.95	2.81	11.41	0	0.00	581.33	9.70
11/2/2011	0.71	9.26	0.11	0.98	0.27	1.19	0.1	0.35	0.35	1.55	0.04	0.22	0.59	0.87	3.04	11.38	0	0.00	0	0.00
12/2/2011	0.71	9.29	0.14	1.16	0.31	1.5	0.11	0.37	0.36	1.61	0.04	0.22	0.19	0.28	2.94	11.49	0	0.00	544	9.10
13/2/2011	0.71	9.36	0.1	0.86	0.35	1.8	0.12	0.4	0.35	1.53	0.04	0.23	0.71	1.03	3.69	11.52	0	0.00	292.83	4.90
14/2/2011	0.72	9.53	0.13	1.1	0.39	2.07	0.13	0.43	0.36	1.6	0.04	0.24	0.43	0.63	3.8	11.8	0	0.00	0	0.00
15/2/2011	0.09	0.66	0.11	0.95	0.32	1.52	0.15	0.5	0.38	1.67	0.04	0.23	0.55	0.8	3.27	3.06	42.17	0.70	565.67	9.40
16/2/2011	0.09	0.66	0.13	1.09	0.27	1.23	0.18	0.59	0.38	1.67	0.05	0.25	0.39	0.57	2.89	3.17	0	0.00	0	0.00
17/2/2011	0.09	0.67	0.13	1.06	0.26	1.17	0.24	0.81	0.39	1.72	0.04	0.23	0.82	1.19	3.42	3.43	0	0.00	608.5	10.10

Figure 21: Finding the SWOMS Sludge Total

Once the totals have been verified, the next area to focus on is the effect of running the oil water separator (OWS) or the incinerator, as well as to look into any significant decreases or increases in the sludge and bilge totals. Therefore, attention is moved from the left "level/volume" section to the right "totals/run time" section. This is highlighted in Figure 22 below.

ENVIROLOGGER DATA

DATE	BHT		SLUDGE		OILYBILGE		STBD BILGE		PORT BILGE		AFT BILGE		W.O.SERVICE		SLUDGES	BILGES	OWS RUN		INCIN BURN TIME	
	LVL	VOL	LVL	VOL	LVL	VOL	LVL	VOL	LVL	VOL	LVL	VOL	LVL	VOL	TOTAL	TOTAL	MIN	HRS	MIN	HRS
1/2/2011	0.79	10.81	0.14	1.19	2.07	26.08	0.26	0.87	0.37	1.62	0.13	0.7	0.37	0.54	27.81	14	3.67	0.06	0	0.00
2/2/2011	0.8	10.89	0.2	1.72	2.07	26.1	0.28	0.94	0.43	1.97	0.13	0.7	0.37	0.54	28.36	14.5	0	0.00	0	0.00
3/2/2011	0.81	11.1	0.2	1.72	2.07	26.04	0.28	0.95	0.46	2.11	0.13	0.7	0.37	0.54	28.3	14.86	0	0.00	0	0.00
4/2/2011	0.91	12.99	0.21	1.76	2.06	26.02	0.37	1.26	0.49	2.28	0.13	0.69	0.37	0.54	28.32	17.22	0	0.00	0	0.00
5/2/2011	0.98	14.48	0.28	2.36	0.13	0.46	0.36	1.23	0.49	2.29	0.14	0.72	0.36	0.53	3.35	18.72	0	0.00	0	0.00
6/2/2011	1.3	21.31	0.28	2.39	0.22	0.92	0.09	0.3	0.1	0.39	0.05	0.27	0.36	0.54	3.85	22.27	0	0.00	0	0.00
7/2/2011	1.3	21.43	0.3	2.61	0.33	1.61	0.09	0.31	0.13	0.51	0.05	0.28	0.37	0.54	4.76	22.53	0	0.00	0	0.00
8/2/2011	1.2	19.14	0.1	0.84	0.17	0.65	0.09	0.32	0.15	0.59	0.05	0.28	0.39	0.57	2.06	20.33	197.83	3.30	0	0.00
9/2/2011	0.7	9.14	0.15	1.24	0.21	0.85	0.09	0.32	0.34	1.5	0.04	0.23	0.29	0.43	2.52	11.19	0	0.00	268.17	4.50
10/2/2011	0.71	9.28	0.09	0.81	0.24	1.05	0.1	0.32	0.35	1.56	0.05	0.25	0.65	0.95	2.81	11.41	0	0.00	581.33	9.70
11/2/2011	0.71	9.26	0.11	0.98	0.27	1.19	0.1	0.35	0.35	1.55	0.04	0.22	0.59	0.87	3.04	11.38	0	0.00	0	0.00
12/2/2011	0.71	9.29	0.14	1.16	0.31	1.5	0.11	0.37	0.36	1.61	0.04	0.22	0.19	0.28	2.94	11.49	0	0.00	544	9.10
13/2/2011	0.71	9.36	0.1	0.86	0.35	1.8	0.12	0.4	0.35	1.53	0.04	0.23	0.71	1.03	3.69	11.52	0	0.00	292.83	4.90
14/2/2011	0.72	9.53	0.13	1.1	0.39	2.07	0.13	0.43	0.36	1.6	0.04	0.24	0.43	0.63	3.8	11.8	0	0.00	0	0.00
15/2/2011	0.09	0.66	0.11	0.95	0.32	1.52	0.15	0.5	0.38	1.67	0.04	0.23	0.55	0.8	3.27	3.06	42.17	0.70	565.67	9.40
16/2/2011	0.09	0.66	0.13	1.09	0.27	1.23	0.18	0.59	0.38	1.67	0.05	0.25	0.39	0.57	2.89	3.17	0	0.00	0	0.00
17/2/2011	0.09	0.67	0.13	1.06	0.26	1.17	0.24	0.81	0.39	1.72	0.04	0.23	0.82	1.19	3.42	3.43	0	0.00	608.5	10.10
18/2/2011	0.09	0.68	0.15	1.25	0.26	1.14	0.3	1.02	0.38	1.67	0.05	0.26	0.7	1.02	3.41	3.63	0	0.00	0	0.00

Figure 22: SWOMS Area of Focus

Once attention is focused onto that specific area of the spreadsheet, the changing of the levels can be more easily observed. When the OWS runs, there should be a decrease in volume somewhere else, as the system is turning oily waste into a mostly water substance that can be expelled overboard. When looking at Figure 23, below, this can be seen.

ENVIROLOGGER DATA			OWS RUN		INCIN BURN TIME	
DATE	SLUDGES TOTAL	BILGES TOTAL	MIN	HRS	MIN	HRS
1/2/2011	27.81	14	3.67	0.06	0	0.00
2/2/2011	28.36	14.5	0	0.00	0	0.00
3/2/2011	28.3	14.86	0	0.00	0	0.00
4/2/2011	28.32	17.22	0	0.00	0	0.00
5/2/2011	3.35	18.72	0	0.00	0	0.00
6/2/2011	3.85	22.27	0	0.00	0	0.00
7/2/[illegible]	4.76	22.53	0	0.00	0	0.00
8/2/20[illegible]	2.06	20.33	197.83	3.30	0	0.00
9/2/2011	2.52	11.19	0	0.00	268.17	4.50
10/2/2011	2.81	11.41	0	0.00	581.33	9.70
11/2/2011	3.04	11.38	0	0.00	0	0.00

Figure 23: Analyzing the Effects of Running the OWS

As can be seen in Figure 23, the sludge total decreased from 4.76 to 2.06 cubic meters and the bilge total decreased from 22.53 to 20.33 cubic meters when the OWS was run for 197.83 minutes (3.3 hours). Decreases in the sludge or bilge totals should be noted with a reason in the ORB. Records of the OWS and incinerator running should be shown in the SWOMS and ORB data; however things such as a transfer between tanks, discharge, or evaporation will only be recorded in the ORB data.

5.1.2.2 Problems with the SWOMS Spreadsheets

Since people that do not do the analyzing currently compile this data, the format of these spreadsheets is not favorable for analysis purposes. The colors tend to be distracting and the columns are not set up in a logical order. This problem has been previously noted, which is why one of the deliverables for this project is to produce an outline of standardized submission formats for future cases. The solution to this problem will be addressed in Section 5.2.

5.1.3 Oil Record Book Analysis

In vessels without SWOMS, the oil record book is the only standard way of documenting oil waste produced in machinery spaces. In a standard ship inspection, errors in the oil record books are most often the evidence of MARPOL violations. In rare cases, there may be first hand observation of means of illegal dumping, like a bypass hose in plain view, or an anonymous whistleblower report sent into the Coast Guard. To get a better idea of how oil record book analysis works, the team was involved in a Department of Justice environmental crimes case similar to the kinds that result in the implementation of SWOMS (Since this is an ongoing case, details that could identify the case, including the type of suspected incompliance, may not be stated). The following sections explain how the analysis in terms of the oil record book takes place (for more information on ORBs, see Section 2.4.1: Oil Record Books or Appendix E: ORB Key).

5.1.3.1 Assessing the Violation & Comparative Analysis

All MARPOL violations that end up being brought to the Department of Justice have some sort of evidence. In order to make an appropriate analysis, understanding where and how the violation was detected is important. Where the violation is found gives insight into which oil record book entries to focus on. For example, if the violation is found to be that one tank was emptied in one day, then all oil record books entries regarding that tank should be looked at with the utmost scrutiny. Once the violation is assessed and the analyst knows what to look for, the analyst will compare the ORB with another monitoring system or another source. For the SWOMS vessels, if a violation is detected, their monitoring system data will be compared to the oil record book and any discrepancies will be noted. For vessels without a SWOMS, noticing discrepancies in tank levels and suspecting violations of MARPOL is a bit more difficult. In some instances, vessels will send information about their systems (tank levels, OWS runs, and incinerator runs) back to their headquarters. In other instances, a vessel may be equipped with the most updated Oil Water Separator (built to MEPC.107(49) standards) which stores a memory of the runs for 18 months at a time. In a case where the discrepancy may be a tampered oil content meter, looking into the memory can prove to be useful evidence in a case. The comparative analysis portion is important evidence in Department of Justice cases because differences between two sources of information referencing the same system points towards one being falsified. Unfortunately, many cases do not have two sources to reference making ORB analysis more difficult.

5.1.3.2 System Balance

In the event that there are not two sources (oil record book, email transmissions, OWS memory card, etc.) to cross reference to be able to determine whether there has been a violation, then one must conduct a system balance. The oil record book is designed to work very much like a mass balance in which all oil accumulation from machinery spaces must be accounted for. If a system balance needs to be conducted, an analyst may go through the oil record book and in their own notes keep a detailed account of all the tank levels and volumes and how they change with different run times or transfers. Each week in the ORB, there is a Code C, item number 11 in which the retained amounts in all tanks must be listed as well as capacities. If there is oil unaccounted for or large changes in tank levels with no notation in the ORB between weeks, then this can immediately denote a MARPOL violation since falsifying the ORB is a crime against MARPOL. Though the mass balance approach may be more difficult and tedious, it is often the only option when there is no other source of information to reference, such as emails or monitoring systems.

5.1.3.3 Identifying Discrepancies in the Oil Record Book vs. SWOMS

Comparing oil record books with data from the SWOMS is important to ensure that the probationary vessels are adhering to the terms set by their probations. Therefore, any discrepancies in the SWOMS data must be checked against the oil record book. Currently, this is challenging due to the fact that the analyst must note the date of the anomaly and then scroll through an oil record book document that can sometimes be as long as 70 pages to look for that date and then find specific information within the handwritten entries. These challenges are addressed later in this chapter in Section 5.2: Templates for Database Integration.

5.1.3.4 Challenges in Oil Record Book Analysis

Oil record book analysis is often difficult and tedious. Challenges in this analysis primarily come from the fact that the oil record book was designed to record a wealth of different kinds of information in one place. Since this book is universal for all vessels and is designed for humans to document many different types of entries, it is designed to be easy for a person to input information, but much more difficult for a person unfamiliar with that specific vessel to extract information from. Not only do all the entries contain different types of information, but all the records are handwritten. This can complicate the analysis process. These record books are great as a universal tool for all vessels and for human record keeping, but are not as good for being referenced at a later date.

5.2 Templates for Database Integration

One of the deliverables of this project is to propose a standard plan for the DoJ to use in future sanctions. This includes a format for data to be sent in, specifications regarding which data should be included in the data sent, as well as a standard plan for sending the data in to the United States Coast Guard. The following sections describe the templates created as part of this standard plan. These templates are a response to some of the drawbacks that were addressed in the previous and current submission formats. The SWOMS Data template is a template for the database where as the Sounding Log Book Spreadsheet Template is a template for a spreadsheet to be filled out in a specific manner. The Tank Identification Template is a template for the database. Therefore, two out of the three of the templates listed in the following sections are templates for the database where as one is a spreadsheet template to be filled out by a person. The two things considered most in the development of the new templates for data submission are (1) Database readability and (2) Friendliness of human analysis (for more information on the USCG, see Appendix A. United States Coast Guard)

5.2.1 SWOMS Data Template

Several of the problems with the current SWOMS spreadsheets were expressed previously in this chapter (see Section 5.1.2.2: Problems with the SWOMS Spreadsheets). Most notable of these problems is that the format of these spreadsheets is unfavorable for analytical purposes. A new template for the data was drafted to address all the previous concerns, in addition to formatting the data for database integration. This new template is not to be filled out by humans and is the template for how the data will be sorted in the database. Since human viewing of the data will be filtered and formatted in reports generated by the database, a human-friendly format for the data analysis was a consideration rather than the primary influence. Also, since this template with be populated with data using a program, the ease of filling it out manually was not considered since any manual entry will be done via a form within the Environmental Compliance Plan Program System.

5.2.1.1 Advantages of the New Templates

The new template includes more information than was previous displayed in the SWOMS spreadsheets. The template was designed to contain all of the information that SWOMS records in addition to the current information sent such as the current geographic location of the vessel, which has always been recorded but is not currently listed in SWOMS data tables that the shipping companies send to the Coast Guard. Having all of the possible information in one spreadsheet will make it easier to track trends.

The new template is laid out effectively containing all information within the emails sent from the ship to the shipping. The SWOMS template consists of 51 columns per data entry (32 of which give room for up to 16 tanks, which is more than most vessels have). The template contains the following information:

- IMO Number
- Date (dd/mm/yyyy)
- Time (UTC; 00:00)
- Latitude & Longitude
- Bilge Pump Runs & Run Time
- OWS Runs & Run Time
- Overboard Opened & Open Time
- PPM Alarms
- Average PPM Overboard
- OCM Freshwater Valve Opened & Open Time
- Oil Purge Valve Opened & Open Time
- Incinerator Runs & Run Time
- Individual Tank Levels & Volumes

The information given in the newer version of the template is much more detailed than the previous data spreadsheets sent by the shipping company, making it easier to identify potential anomalies. For example, if the PPM Alarm is sounded several times or the OCM Freshwater Valve has been opened multiple times, there may be an issue. Below, in Figure 24, is an image of a SWOMS template populated with real ship data. The column with the IMO number was removed from the figure to protect the vessel's identity. This view shows the vast amount of data collected and stored. Further on in this section are more zoomed in screenshots of the template.

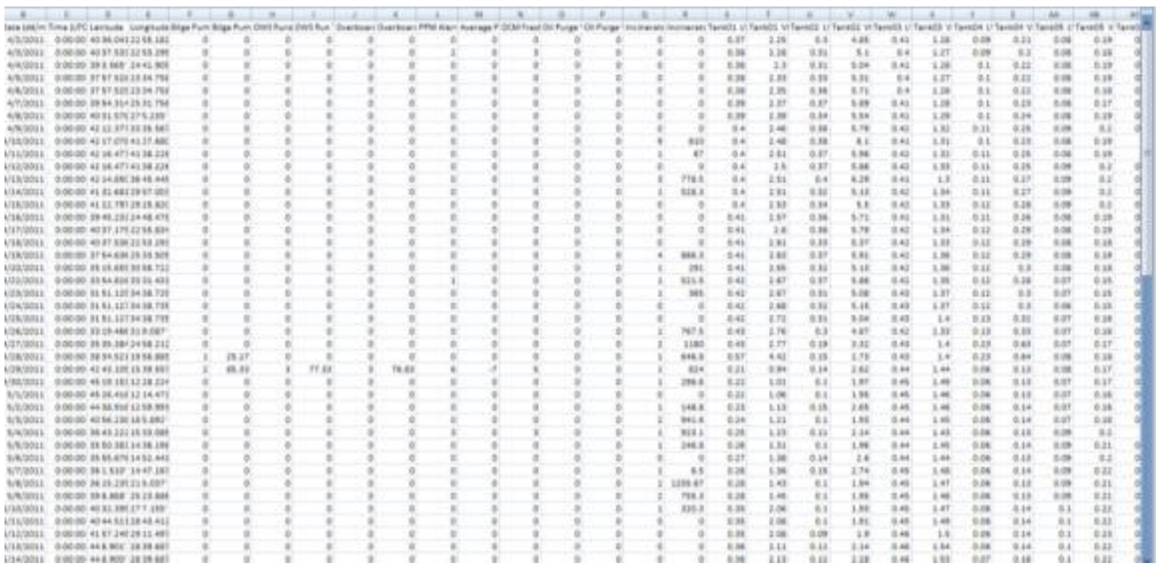


Figure 24: Populated SWOMS Template

This view shows just how many data are being collected in these templates. It also highlights how different the template is compared to the older spreadsheet. In a closer look at the template, as in Figure 25, the identifying information can be seen on the left.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	IMO Numb	Date (dd/m	Time (UTC	Latitude	Longitude	Bilge Pum	Bilge Pum	OWS Runs	OWS Run T	Overboar	Overboar	PPM Alarr	Average P	OCM Fresl	Oil Purg
2	[redacted]	4/2/2011	0:00:00	40 36.041	22 55.182	0	0	0	0	0	0	0	0	0	
3	[redacted]	4/3/2011	0:00:00	40 37.533	22 53.299	0	0	0	0	0	0	2	0	3	
4	[redacted]	4/4/2011	0:00:00	39 3.565'	24 41.909	0	0	0	0	0	0	0	0	0	
5	[redacted]	4/5/2011	0:00:00	37 57.528	23 34.758	0	0	0	0	0	0	0	0	0	
6	[redacted]	4/6/2011	0:00:00	37 57.529	23 34.758	0	0	0	0	0	0	0	0	0	
7	[redacted]	4/7/2011	0:00:00	39 54.314	25 31.756	0	0	0	0	0	0	0	0	0	
8	[redacted]	4/8/2011	0:00:00	40 31.578	27 5.235'	0	0	0	0	0	0	0	0	0	
9	[redacted]	4/9/2011	0:00:00	42 12.377	33 35.567	0	0	0	0	0	0	0	0	0	

Figure 25: SWOMS Template—Identifier Information and Alarms

Each record is identified with the IMO number. The IMO is the unique identifier for the ship and will follow the ship if it changes ownership or moves to another flag state (for more information on IMO numbers, see Section 2.1.2: International Maritime Organization (IMO) Vessel Identification). This will be useful when the data are imported into the database. All date, time, and location information is directly next to the identifier to show time and location of each entry. The columns to the right of these identifiers, such as the alarms and valve notifications, summarize the 24-hour data collection period. This information includes the number of times something happened as well as the duration of the activity during the 24-hour data collection. Towards the right of this alarms section is the instantaneous tank level and volume section that can be seen in Figure 26.

Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC
Incinerato	Incinerato	Tank01 L\	Tank01 V	Tank02 L\	Tank02 V	Tank03 L	Tank03 V	Tank04 L	Tank04 V	Tank05 L	Tank05 V	Tank0
0	0	0.37	2.25	0.3	4.85	0.41	1.28	0.09	0.21	0.08	0.19	0
0	0	0.38	2.28	0.31	5.1	0.4	1.27	0.09	0.2	0.08	0.18	0
0	0	0.38	2.3	0.31	5.04	0.41	1.28	0.1	0.22	0.08	0.19	0
0	0	0.38	2.33	0.33	5.31	0.4	1.27	0.1	0.22	0.08	0.19	0
0	0	0.38	2.35	0.36	5.71	0.4	1.26	0.1	0.22	0.08	0.18	0
0	0	0.39	2.37	0.37	5.89	0.41	1.28	0.1	0.23	0.08	0.17	0
0	0	0.39	2.39	0.34	5.54	0.41	1.29	0.1	0.24	0.08	0.19	0
0	0	0.4	2.46	0.36	5.79	0.42	1.32	0.11	0.25	0.09	0.2	0

Figure 26: SWOMS Template—Instantaneous Tank Data

On the right side of the SWOMS template, as can be seen above in Figure 26, are the instantaneous tank measurements. The levels and volumes of each tank are recorded instantaneously and sent within the SWOMS emails. This information is what will be directly compared to the ORB information. The template includes all the recorded information and is organized in a format that is readily importable into the database.

5.2.1.2 Database Integration with the SWOMS Template

This template for the SWOMS data was created with the intent of database integration. The template was constructed to make sense to a human but also to a computer so that a computer can analyze the information and report anomalies. This standardized template will make it so that the import into the database will be easily carried out with minimal errors, if any. The format regarding the tank names such as "Tank01" was done specifically to ease analysis between the SWOMS data and the ORB data. The standards for the identities behind the tank names will be addressed in Section 5.2.3: Tank Identification Template.

5.2.2 ORB Data Template

Currently, analyzing oil record books against the data from the SWOMS is important in ensuring that the probationary vessels are in compliance with MARPOL regulations. Therefore, any discrepancies in the SWOMS data must be checked against the oil record book to indicate whether the discrepancies are indeed evidence of noncompliant activity. With the current handwritten version of the oil record book and the desire to incorporate the ORB data into the Environmental Compliance Plan Database, a method to enter the data into a digital template to be compared against the SWOMS data is necessary. For there to be any integration of information from the oil record book into the Environmental Compliance Plan Program System, there needs to be a manual rewriting of information from the oil record book or sounding books into the templates that the Coast Guard will provide shipping companies.

5.2.2.1 Preliminary ORB Template

The initial solution to this dilemma was to manually transcribe the ORB entries into a spreadsheet that was organized by code and item number. The original thought was that it would the same as the current ORB but in a digital format, as can be seen below in Figure 27.

	A	B	C	D	E	F	G	H	I
1	IMO Numl	Date (mm/dd/yyyy)	Code	11.1 Colle	11.2 Capa(	11.3 Retai	12. Amoui	12. Tank Ei	12. Retained
2	[redacted]	4/1/2011	C				0.23	M/E Scave	0
3	[redacted]	4/2/2011	D						
4	[redacted]	4/2/2011	C						
5	[redacted]								

Figure 27: Preliminary ORB Template

The reasoning behind this preliminary template for the ORB was that it would be a digital version of the oil record book. It would have all entries listed by code and item number and would be readable by the Environmental Compliance Plan Program Database. This can be seen in Figure 27, where the entries are organized by code and the specific item numbers for that entry are completed. In an effort to manually transcribe some of the ORB data into this template, the amount of time spent transcribing the information into the template as well as other concerns about how to directly compare this data to the SWOMS were realized. A re-evaluation of the approach towards how to incorporate the ORB data into the system for analysis resulted in a new idea to instead use the sounding log book.

5.2.2.2 Using the Sounding Log Book to Compare to SWOMS Data

A sounding log book is a supplementary document in which all the tank levels and volumes are documented and signed by the Engine Officer (for more information on sounding log books, see Section 2.4.1.1 Engine Room Tank Sounding Log Book). When viewing a sounding log book, the information contained matches that of the instantaneous tank measurements that SWOMS produces. This makes it a good source of data to compare to the SWOMS data within the database. A sounding log book can be seen below, in Figure 28. The log book is a tabular representation of data within the ORB but in the format of the SWOMS data. It only includes the current level and volume of specific tanks each day. Therefore, it does not include the tank transfer data or oil water separator (OWS) data that the oil record book contains.

Figure 28: A Sounding Log Book

In order to be able to make a direct comparison to the SWOMS data, a template was created for the sounding log book that would require manual input of the data from the sounding log book into an Excel spreadsheet. The template for the sounding book data can be seen below, in Figure 29. The template is an exact digital replica of the actual sounding log book but with a column to enter the time if given. This field will become more important when the SWOMS data switches to an hourly recording.

	B	C	D	E	F	G	H	I	J	K	
1	Date (mm/dd/yyyy)	Time (UTC; 00:00)	Tank01 LV	Tank01 V(	Tank02 LV	Tank02 V(	Tank03 LV	Tank03 V(	Tank04 LV	Tank04 V(	Tan
2	4/1/2011		1.560	20.740	0.560	5.260	0.220	0.200	0.900	1.130	
3	4/2/2011		1.560	20.740	0.820	8.920	0.220	0.200	1.200	1.590	
4	4/3/2011		1.560	20.740	0.900	10.130	0.220	0.200	1.100	1.440	
5	4/4/2011		1.560	20.740	1.100	13.280	0.220	0.200	1.100	1.440	
6	4/5/2011		1.560	20.740	1.180	14.400	0.220	0.200	1.100	1.440	
7	4/6/2011		0.530	4.730	1.720	22.370	0.220	0.200	1.170	1.570	
8	4/7/2011		1.900	25.500	0.520	4.730	0.220	0.200	1.170	1.570	
9	4/8/2011		1.900	25.500	0.540	4.990	0.220	0.200	1.170	1.570	
10	4/9/2011		0.700	6.830	0.620	6.060	0.220	0.200	0.900	1.130	

Figure 29: Sounding Log Book Template

Comparing the SWOMS data to a sounding log book is more beneficial than using the oil record book entries because it is comparing the same types of data, which are volumes and levels of tanks. The analysis will be more straightforward and will have the ability to produce a conclusion right away as to whether something should be looked at more carefully or whether the vessel is operating in compliance with all MARPOL regulations.

5.2.2.3 Potential Problems with Using the Sounding Log Book

Using a vessel's transcribed sounding log book to compare to the SWOMS data, rather than comparing it the transcribed ORB data to compare to the SWOMS data, is a better solution to the challenge of comparing the vessel's records to the monitoring system's records. It is a better solution because it is comparing the tank levels and volumes recording by the monitoring system to tank levels and volumes that have been recorded on the vessel by individuals. This makes for a direct comparison that can help conclude whether the ship is keeping accurate records. However, there are a few concerns that may need to be further considered in the event of the future implementation of this system. The most notable concern is that although the sounding log book is signed and maintained, it does not have the same legal weight as the ORB. In the event that a shipping company knows that anomalies are being flagged via the sounding log book, they may falsify the sounding book in hopes that no one will look at the ORB and discover inconsistencies. This is a concern that will be addressed in recommendations.

5.2.3 Tank Identification Template

One problem in this entire process is that vessels use different names for their tanks than SWOMS. In addition to this, various vessels may refer to the same tank with different names as well. This element is established to ensure that the correct tanks within the database are compared during the computerized analysis process. To ensure that this takes place, a template is created to identify all tanks. The template would only need to be filled out once. Pictured below, in Figure 30, is the template to be filled out.

	A	B	C	D	
1	IMO Numl	Tank01 Type	Tank01 Desc.	Tank01 Capac	Tank02
2	[redacted]	Bilge	Clean BHT	32.21	
3					

Figure 30: Tank Identification Template

The template relates the type of tank, a description, and the capacity with the "TankXX" and the IMO number. This identification makes for direct comparison as well as information about the tanks that is important for analysis.

5.2.3.1 Entering Tank Identification Information

Tank identification information will be entered into the database via an electronic form. Using the form will populate the tank identification table within the database that is the exact format as the template above. In order to avoid mistakes in the tank information, it is suggested that a new vessel not be added into the database without one SWOMS email and one sounding log book sent. This is because the tank identities are governed by the order in which they are reported within the SWOMS emails. They will automatically be generated into the SWOMS template in that order. The next task is to record and identify the tanks in the order they are recorded within the SWOMS emails. This will be the order that tanks should be entered into the Tank Identification Form on the database and also the order in which the tanks should be listed on the sounding log book template. This order of tanks will need to be sent to the shipping companies with the template for the sounding log so that all the tanks labels correspond to the same tanks in the sounding log data as in the SWOMS data for correct comparison. Below in Figure 31 is the template from within the database that correlates with the tank identification sheet.

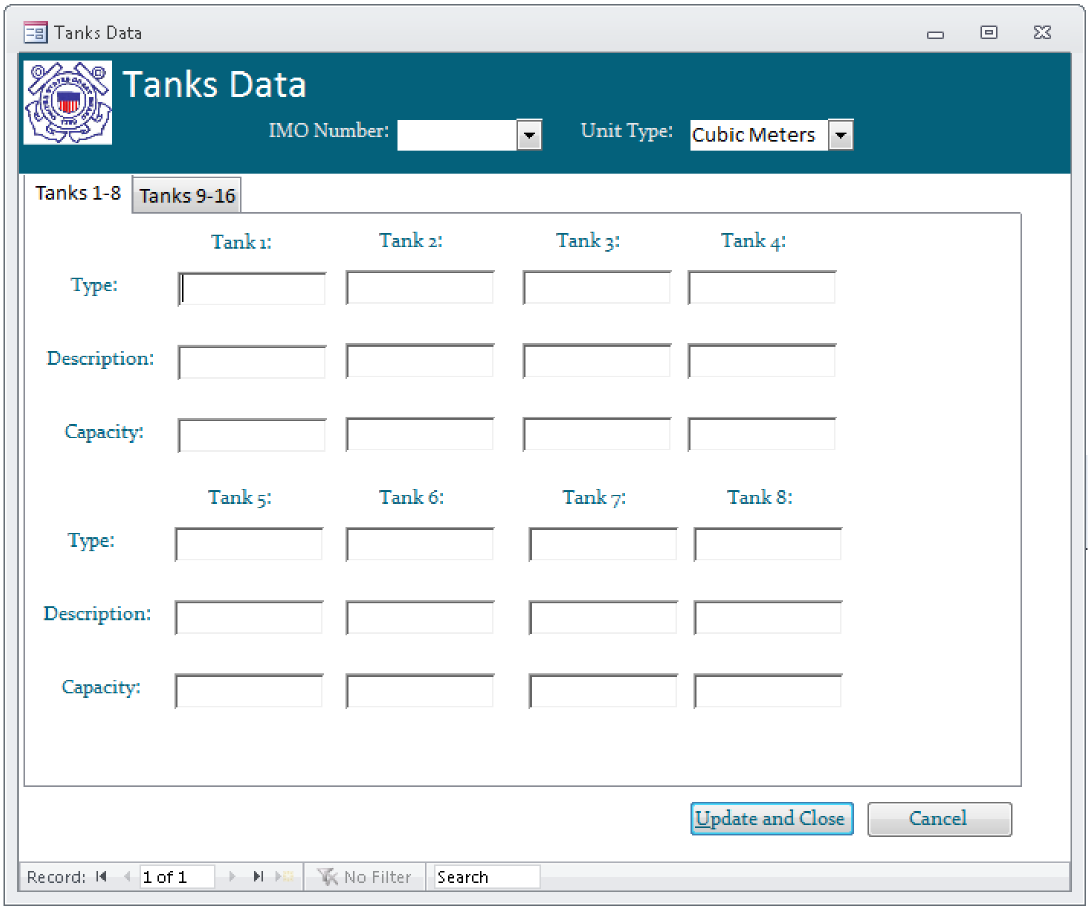


Figure 31: Tank ID Form

The above form allows for the initial entering as well as future editing of tank information. This makes it possible to be able to enter the tank type and description and then add in capacities once known.

5.3 New Standard Transmission Method

The current transmission method was outlined previously in this chapter. In summary, the ships send the data to the shipping company, who send the data to the Coast Guard on a quarterly basis. Currently, the data comes in larges quantities and by the time it is received and analyzed, a significant amount of time has already passed. The problem with this is that the Coast Guard has a limited allotted time in which they can make a case for a violation detected in order to extend the vessel's probation. In order to address these concerns, as well as accommodate the amount of data being sent, there is a proposal for a new data transmission method. Below, in Figure 32, the previous transmission methods are compared to the proposed transmission methods.

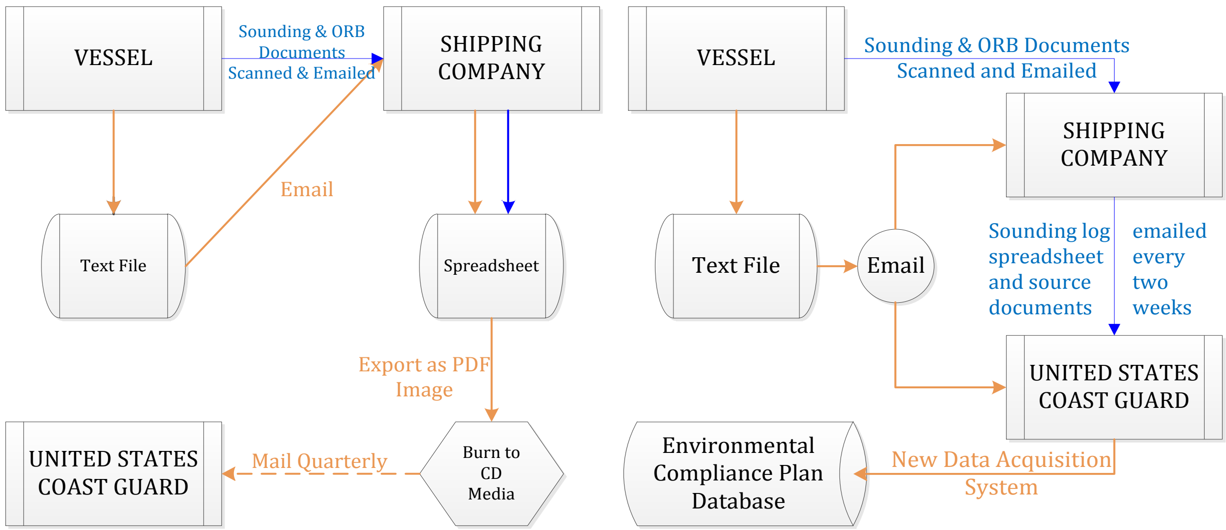


Figure 32: Data Transmission Path, Current (to the left) and Proposed (to the right)

5.3.1 Mandate that Emails Replace CDs

Since the current method of sending data on CDs that come via mail is insufficient, it is proposed that the data be sent via email. CDs come in to the Coast Guard long after the data are logged, and then there are further time losses because a person needs to manually extract the information from the CDs. After retrieving data from the CDs, a person would then also need to import the data to the database before analysis can begin. Email is significantly faster and will be able to accommodate the future projection of establishing a real-time system. Sending the information via email directly from the ship to the USCG would benefit all parties. The shipping companies would not be responsible for formatting the data from the emails into a spreadsheet. In addition, the Coast Guard would get the data more timely as well as in the format desired. This system will also incorporate SWOMS data into the Environmental Compliance Plan database more quickly.

5.3.2 Mandate that Direct Emails from SWOMS Replace the PDF files

The format of the current SWOMS spreadsheets is not compatible with the database to store the information. If the format of the SWOMS data is not changed, it would need to be manually entered because the data are currently sent in PDF image format. Currently the SWOMS sends emails to the shipping company with all of the information that is currently in the SWOMS spreadsheets sent on the CDs. The Coast Guard would be receiving the emails that SWOMS currently sends to the shipping companies and this would make the

Coast Guard also accountable for compiling the data into a spreadsheet format, rather than the shipping companies. This would take a burden off of the shipping companies and would give the Coast Guard the information needed as well as the ability to sort and format the data to the Coast Guard's preference.

5.3.3 Mandate Bi-Weekly Data Submission

To help address the concern of receiving the data more quickly and the proposed email submission, it is also proposed to mandate data submissions via email every two weeks. Receiving the data every two weeks will help catch any violations in a timely manner and will allow for adequate time to take legal measures if necessary. This also will help to not exceed the 5MB email limit at the United States Coast Guard. Required every two weeks will be a copy of the sounding logs and the oil record books that will correspond with the dates for the previous two weeks. It is recommended that the completion of the sounding log template be done by the shipping company.

5.3.4 Minimally-Managed Exchange Mailbox

As part of the automation process, it was decided that a minimally-managed, shared mailbox would be used as a destination for incoming data. The mailbox acts much like a "no-reply" email address, but administrators will perform first-time setup for each new ship that gets added to the system. The process requires human control at the beginning of new environmental compliance plans in order to correctly establish data associations. Once set into motion, the system will be able to process SWOMS data with only a few mouse clicks, a little bit more for sounding data, and a short import process to bring data into the database.

The process starts when the first SWOMS email comes in from a particular ship. In addition to other setup required for the database such as the tank identification form, the first SWOMS email data will need to be manually updated with the vessel's IMO number. The initial email for each vessel will be assigned a dummy IMO number, which a user will need to change once for each new vessel. Every successive entry from a vessel will automatically match the IMO number on record.

When emails arrive in the shared mailbox, database users can run an Outlook macro from their personal mailbox on a selected group of SWOMS emails. The macro's script is run on the selection, and processes emails together as a group. The script first checks their subject lines for key words that identify them as a "daily report" email. If an email is identified as a daily report, it will be marked with a "completed" flag (check mark) and the script will initiate the parsing process. The email is converted to plaintext and saved as a temporary text file while leaving the original email intact. In order to reduce overhead on the Exchange server, users will need to manually archive old SWOMS emails to Outlook storage files. The VBA macro also creates a batch file that runs the parsing program on each temporary text file, deletes the text file, and finally deletes itself.

5.3.5 Program to Convert SWOMS Data into a Database-Readable Format

As previously described, the data currently coming in to the Coast Guard is not well-suited for direct input to the database. The CDs require manual insertion and retrieval of the data. The individual text files cannot be directly imported, and the PDF spreadsheet is virtually useless for programmatic manipulation due to the formatting changes that occur during PDF conversion. One solution for importing the data is to use a program to automatically convert the text documents into a standard, easily-manipulated format. A program, "swomsparser" was written as part of this project that parses the text documents into a single comma-delimited spreadsheet, also known as a "comma-separated values" file, or "CSV". A standard CSV file has many limitations in that it does not support any of the formulas, formatting, colors, or even column width information that is normal for any modern spreadsheet.

However, the CSV format takes up a fraction of the size of an Excel file since it lacks any of those above mentioned features. This is preferable since it reduces overhead, especially for large amounts of data. The relative ease of converting the text documents to CSVs for importation into the database means that a plaintext email is usable with little processing. The speed benefit of this method is twofold; email can arrive significantly faster than physical mail services, and the data importation system can be automated so as to remove the manual extraction of data from CDs. In addition, this should minimize the risk of any human error in extensive data input. This system should allow for significant expandability, allowing additional ships to be added without changing the processing elements, provided that the incoming data matches the newly developed templates that were described earlier. Furthermore, the simplicity of this system allows its use with basic data output from the embedded systems one would expect in a SWOMS unit.

5.4 Constructing the Database

5.4.1 Planning

The first step in writing any program is to come up with a plan for how the different parts will interact with each other. Using the flowchart shown in Figure 33, a plan for the database was created.

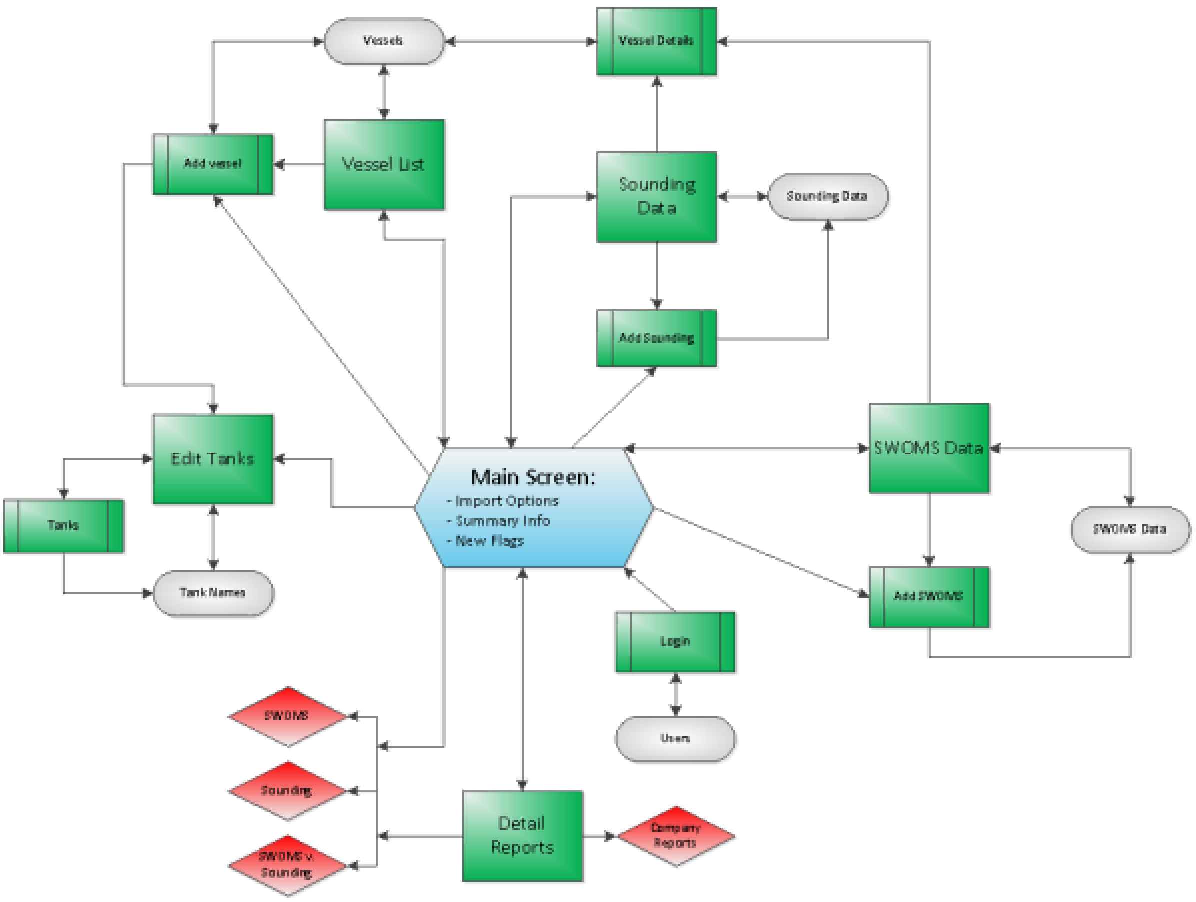


Figure 33: Access Database Flow

The Main Screen form is in blue to represent that it is the primary form, which is the form that opens when you start the program; otherwise the forms are green, reports red, and tables to store data are in white. The forms are where the user enters, modifies and views the data stored in the tables. The reports use Access queries to analyze the data in the tables and display a list of anomalies and additional relevant data in printable form. The arrows between the different objects indicate which objects interact and where the queries will be used to gather data to be displayed or calculated. The tables for storing data are all related to the list of vessels by the unique ship IMO number as shown in Figure 34.

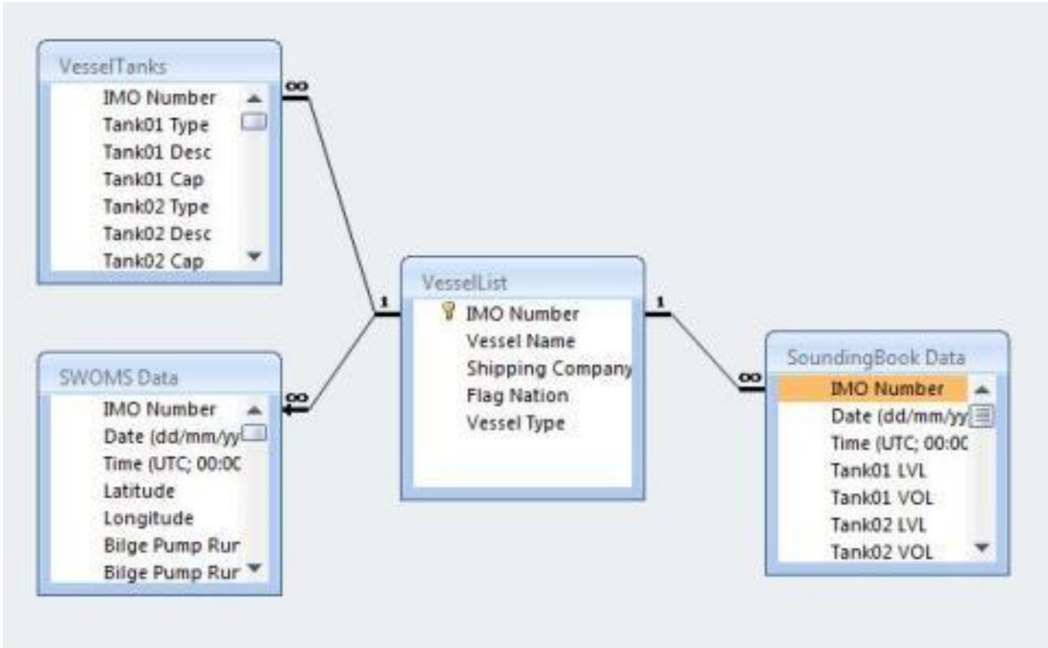


Figure 34: Access table relationships

This means that you cannot add data for a vessel that is not already in the database to avoid any errors from having incomplete data sets. This layout will be elaborated on in the following section.

5.4.2 Development

With this plan, actual construction of the database using Microsoft Access could begin. Microsoft Access was chosen because of its availability as well as its ability to carry out all of the functions required. Using the flowchart as a guide, the various forms, reports and tables were added and populated with controls. The tables created for the database are:

- SWOMS data
- Sounding Book data
- Vessel List
- Tanks Data

Using the format of the Excel templates for entering data, the tables were given the requisite fields in order to contain all the necessary data. The creation of all database templates is detailed previously in Section 5.2: Templates for Database Integration. The SWOMS data and sounding book data tables contain all fields from the templates, and records for all the ships for all dates that have been added. The vessel list table simply lists every vessel in the database. The tanks data table lists the type, description, and volume for each tank for each vessel in the system. The forms were then designed in order to edit and view the data from the tables in a user-friendly manner. The various forms were generated using Access's built in defaults, and then additional controls such as buttons and combo boxes were added and named according to what the user would use them for. The forms in the database are:

- Main Menu
- Add SWOMS data/Add Sounding Book data
- View SWOMS/View Sounding
- Add Vessel
- Edit Tanks
- Vessel List
- Vessel Details
- Detail Reports

The main menu form was designed to be the primary user interface and as such is the first form to open when you open the database. In order to provide the user access to all the various forms an organized layout was developed with command buttons that use VBA code to navigate between the main form and all the others. The most important part of the main menu however, is that it is where the user generates basic anomaly reports using the data. A combo box stores the list of IMO numbers in the database which the user selects, along with typing a start and end date into a pair of text boxes that are formatted to only accept dates in the dd/mm/yy format.

The forms for adding SWOMS and sounding book data were created so Coast Guard employees could manually add data to the database if that was ever necessary. Both forms use a series of text boxes which are linked to fields in their respective tables which allow a new record to be created when the form is closed using Access' built-in data entry methods. In order to better organize the add SWOMS form, the Access tab control was used to create two tabs, one for the basic SWOMS data and one for tank levels and volumes. Since the sounding book does not contain that first set of data, tabs were unnecessary for that form.

While the reports that filter for anomalies are the most important way for viewing SWOMS and sounding data, it also important for the user to be able to view the unfiltered data in its entirety. Two forms were created in order to do this, one for each set of data. It was decided that viewing all the data stored in the table at once was impractical, so basic queries that filtered the data for a specific IMO number and date range were created. To display the data from these queries, the two forms were modified to have a split view, with command buttons at the top and a datasheet underneath to display the data, but not modify it. This datasheet was made by creating text boxes for each field in the query, which access then automatically duplicates for each record. Much like on the main form, a combo box for IMO number, and two text boxes for the start and end dates were created to provide the user input to filter the data.

In order to allow the Coast Guard to easily add new vessels to the system, a New Vessel form was added to the database, which allows the user to quickly add a new vessel to the database that will now be part of the SWOMS system. There are fields for IMO number, vessel name, shipping company name, the flag nation of the vessel, and the vessel type. Only the IMO number and vessel name are required fields, as the IMO number is the vessel's unique identifier, and the name is important for users who don't have the IMO numbers memorized to identify the vessel. However the remaining fields, while important for the Coast Guard to know, can be entered at a later date without affecting the use of the database and so are not required when the user enters this data. When the user finishes this form they are taken straight to the edit tanks form in order to add tanks to this new vessel.

The edit tanks form allows the user to add, modify, or delete tank information. As the number of tanks on a vessel varies, it was decided the database would have the capability to hold information on up to 16 tanks and it would be up to the Coast Guard to make sure that the data are accurate for each tank. Therefore this form was created with two tabs having text boxes linked to all the fields in the tanks data table. A combo box in the form header is set up to select an IMO number, and then when the user changes data in the text boxes it will change the record in the tanks data table for that IMO number

The last forms for viewing and editing data are the Vessel List and Vessel details forms. The vessel list form uses a basic query to select the data in the vessel list table and display it on a form much like the SWOMS and sounding data forms, with buttons and other selectors on top, with a datasheet underneath. The vessel details form can be accessed from the SWOMS and sounding forms and allows the user to see the data displayed in Vessel List, but for the currently selected vessel on those other forms, as well as allowing the user to modify that data with the exception of the IMO number which cannot be changed.

Lastly, there is the Detail Reports form. It was constructed to allow the user greater control over how data are displayed in the reports. It allows for the user to select a vessel and date range like the main menu, but also allows the user to select certain tanks or other data fields to be highlighted in the report to make it easier to focus on certain fields and data values. In order to select the list of tanks for a vessel, a new table was added to the database that takes just the tank names from the tanks data table and puts them in a new table so they are in a format compatible with the list box. It was originally intended that the user would be able to organize reports by company as well, but this feature ended up being too much to implement in the time allotted.

Once the forms were completed, the next step was to add basic functionality such as the ability to move between the various forms using command buttons, as well being able to view the data stored in the database. The command buttons and other controls were constructed using built-in Access tools for designing the user interface. All of this basic functionality was implemented by writing VBA code for the various events that could happen to controls such as a button being pressed or the value of a text box being changed such as the example code in Figure 35.

```
Option Explicit
Option Compare Database
Dim User As String
'this menu allows access to the data viewing forms, as well as links to add data or to generate a report based on basic parameter

Private Sub cmdCompare_Click() 'open comparison report
    If (Not IsNull(Me.cmbIMO)) And txtEnd.Value > txtStart.Value Then
    DoCmd.OpenForm "DetailReports", acNormal
    Form_DetailReports.cmbIMO.Value = Me.cmbIMO.Value
    Form_DetailReports.txtStart.Value = Me.txtStart.Value
    Form_DetailReports.txtEnd.Value = Me.txtEnd.Value
    Form_DetailReports.Visible = False
    DoCmd.OpenReport "SWOMSvsSounding", acViewReport, "", "", , acNormal
    Else
    MsgBox ("Missing IMO Number or Incorrect Date Range")
    End If
End Sub

Private Sub cmdCPrint_Click() 'print combined report
    If (Not IsNull(Me.cmbIMO)) And txtEnd.Value > txtStart.Value Then
    DoCmd.OpenForm "DetailReports", acNormal
    Form_DetailReports.cmbIMO.Value = Me.cmbIMO.Value
    Form_DetailReports.txtStart.Value = Me.txtStart.Value
    Form_DetailReports.txtEnd.Value = Me.txtEnd.Value
    Form_DetailReports.Visible = False
    Report_SWOMSVsSounding.Printer.Orientation = acPRORLandscape
    DoCmd.OpenReport "SWOMSvsSounding", acViewNormal, , , acHidden
    Else
    MsgBox ("Missing IMO Number or Incorrect Date Range")
    End If
End Sub

Private Sub cmdDelete_Click()
```

Figure 35: Code from the Main Menu Form

In order to display and edit the data from the tables, additional work was needed. Most of this code was also implemented using VBA, although basic queries were used to display the SWOMS and Sounding data by taking the data from the table and displaying them as datasheets on their respective forms. With data manipulation and viewing implemented, the final step was to write queries to calculate the anomalies in the data.

These queries are much more complex than the ones written to display data and therefore were coded in SQL which Access can then use to make a query. The comparisons between records of the same type for both SWOMS and Sounding compared the change in the percentage of capacity of the various tanks. The initial queries take the data from the tables and compare that to the capacity for that tank as stored in the tank data table and then calculate the percentage. Then, using the data from that query, another query can take those percentages and calculate the day to day change in percent capacity. If the change is greater than a 30% decrease, it is highlighted as a possible anomaly. Any other changes in the data besides large decreases are not important to highlight as they are not likely to be anomalies caused by the vessel crew tampering. In addition to the day to day comparison queries, there is also a query to calculate the difference between SWOMS and Sounding book data for each day. As this data should be the same or very close, this smaller margin of error is what is checked for. If the error is greater than 5%, it is displayed in the report.

The database can create three kinds of reports: one for the day to day change in SWOMS data, an equivalent report for sounding book data, and a third report for comparing entries from SWOMS and sounding to each other. These reports are populated with data from the previously discussed series of access queries that gather data from the various tables and sort, organize and make calculations on that data to determine where inconsistencies have occurred. The first two reports are identical except for which data they display. After the title section of the report are two tables, one for tanks 1-8, and another for tanks 9-16, which show the greater than 30% anomalies in the day to day data, the first report doing so with the SWOMS tank data, and the second with the sounding book data. Below those first two tables are two more which give a breakdown on additional relevant information collected by SWOMS such as OWS and Incinerator runs, and the number of PPM alarms for each date. In order to save space, any day where none of these actions were run is omitted from the report. The third report is the one which displays the results of the 5% difference query between the two data sets. It also has a title section followed by the same setup of tables, the difference being that these top two tables show the difference between the two data values for the same date, while the other two tables show the difference between the current date and the previous one.

5.4.3 Capabilities

After going through this development process the end result was a database that has the capability to allow the user to add, view and modify data. In addition the database has the capability to perform a comprehensive analysis of the SWOMS and sounding book data to find anomalies. When a user first opens the database the main menu opens. The main menu form gives access to the other forms in the database, as well as having the ability to generate quick reports based on an individual vessel and a date range. From the main menu, the user can access the forms for manually adding data to the database. The most important of these is the New Vessel form, which is used to add new vessels to the database and is shown in Figure 36.

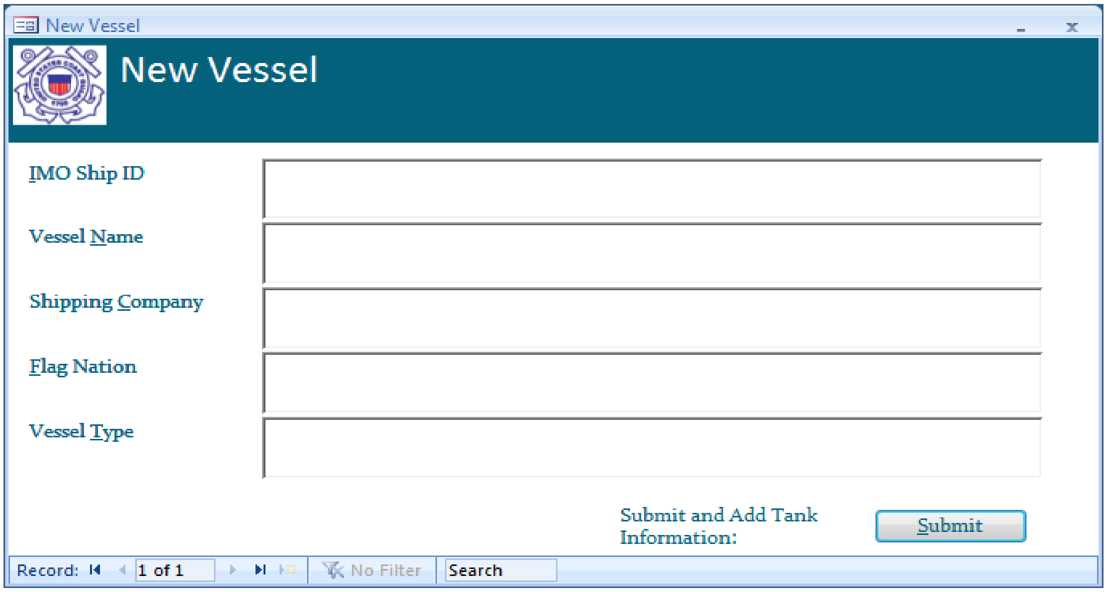


Figure 36: New Vessel Form

This form asks for some basic information about the vessel, most importantly its IMO number, and then has the user enter the tank data for that new vessel. This tank data form, shown in Figure 37, can also be accessed from the Main Menu and other forms, and can be used to edit the tanks of existing ships as well as adding data for a new one.

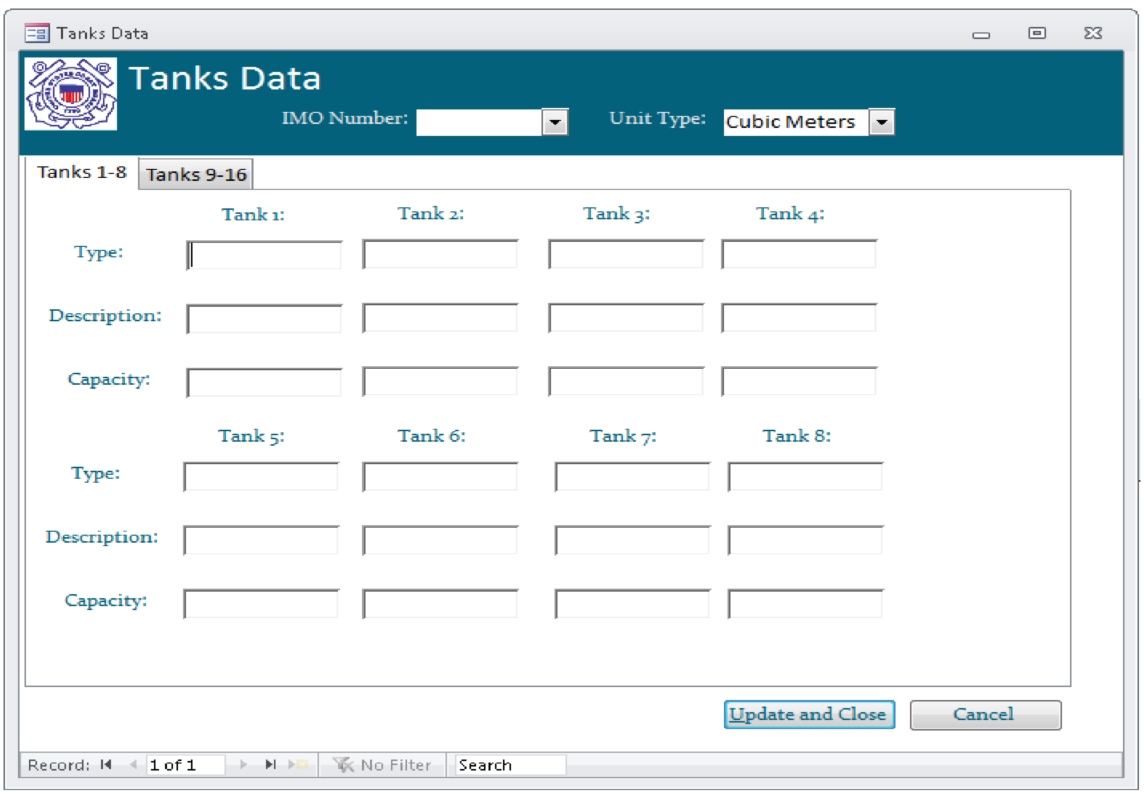


Figure 37: Tanks Data Form

Additionally, there are forms for adding new SWOMS data and new Sounding data. Both forms are similar, except that the SWOMS form (see Figure 38) has more inputs for the user to enter.

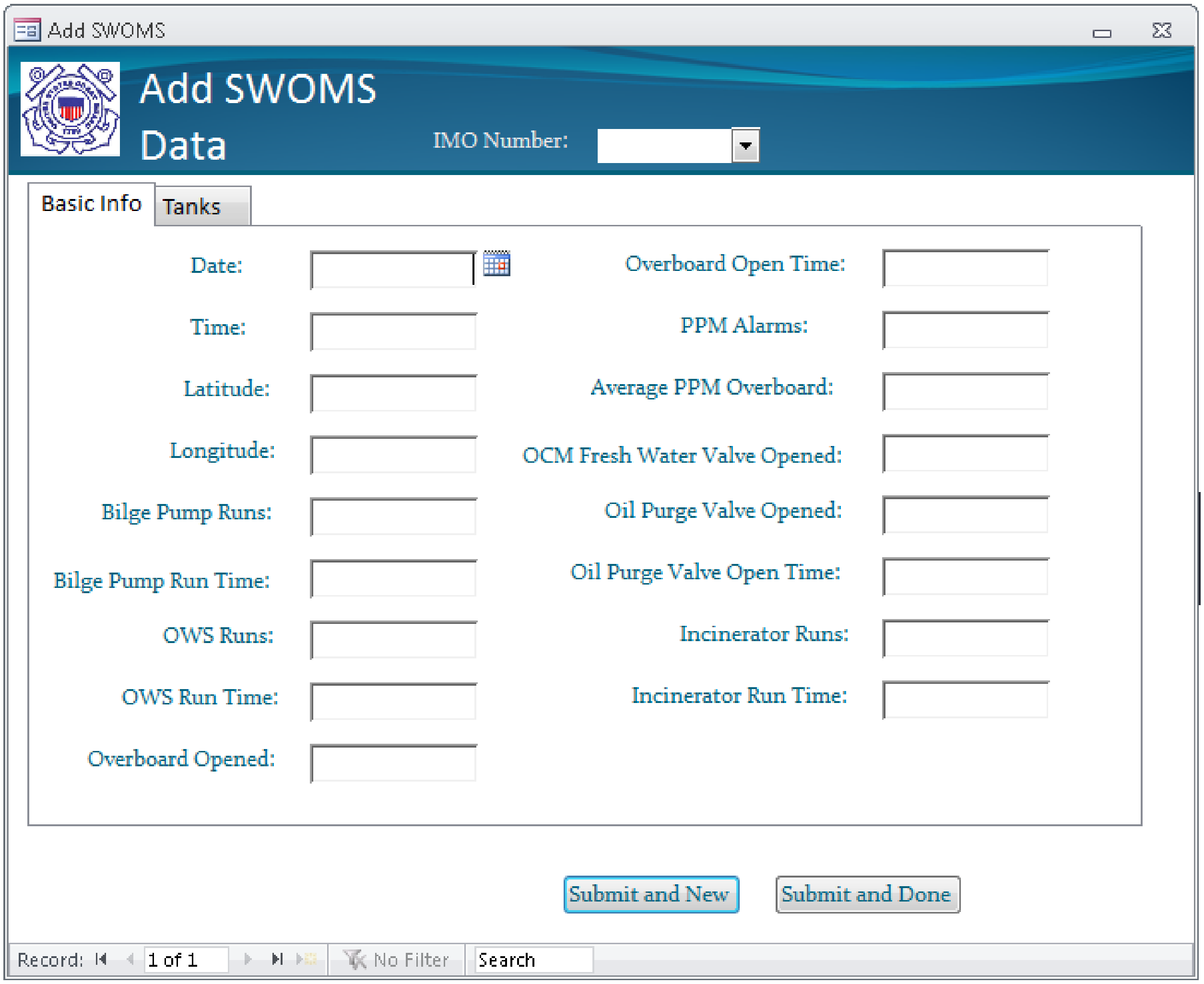


Figure 38: Add SWOMS Data Form

Once the user has finished filling in the form, he has the option to either submit that entry and finish, or to add it and then automatically clear the form allowing the user to start a new record. These two forms would primarily be used for entering a small amount of data received in a manner different then the usual transmission method, as it would be tedious to enter large amounts of data this way. All the forms previously discussed are pop-ups that leave the main menu or the previous window still open and visible to the user.

The database also allows users to delete SWOMS and sounding data once it is no longer needed in the database. The user selects an IMO number and date range, and then presses a button to delete either SWOMS or sounding to delete the data as seen in Figure

39. After a message double-checking the user wants to do this, the data are permanently deleted from the database.

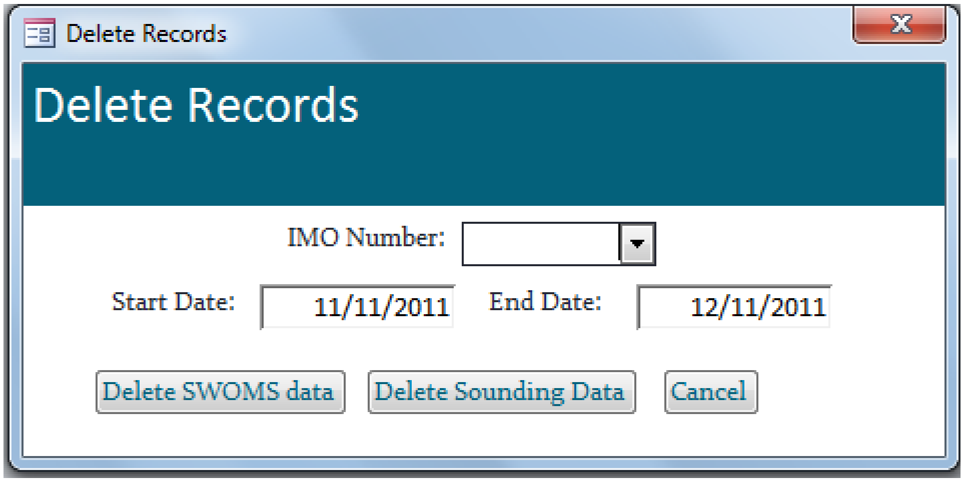
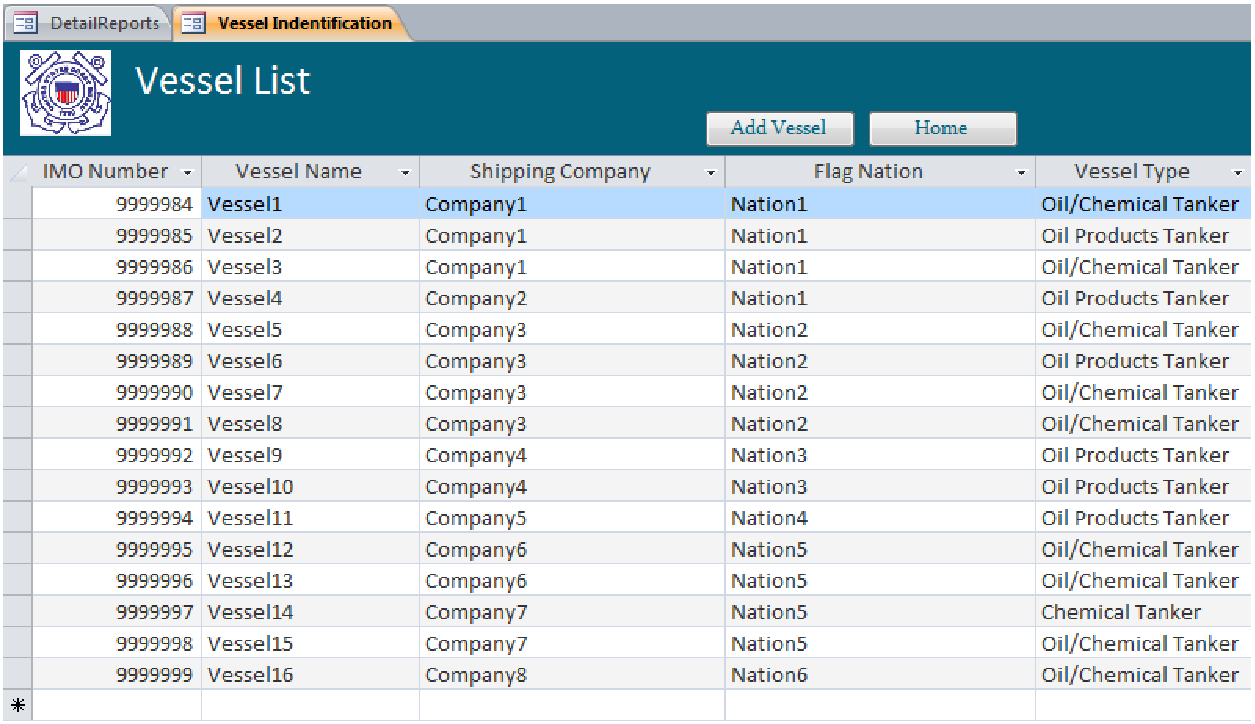


Figure 39: Delete Records Form

There are two forms in the database that allow for both the viewing and modifying of data, Vessel Details and Edit Tanks. The Vessel Details form has all the details of a specific vessel that are stored in the vessel table. As all of this data except the IMO number could be subject to change the form allows all other fields to be edited. The edit tanks form allows the user to view all currently stored data on the tanks for a specific vessel. It allows for the modification of any of these fields as well as the addition or removal of tanks as necessary as this could also be subject to change. Both of these forms are pop-ups as well that are opened from other forms in Access.

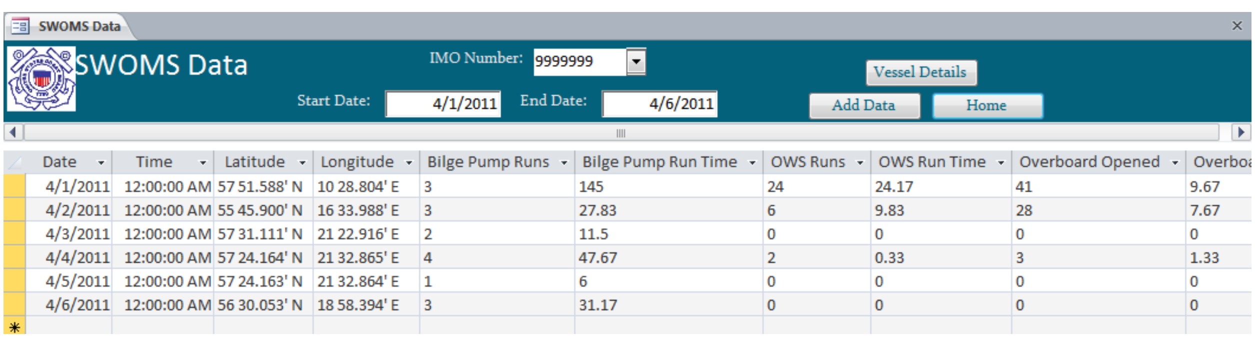
From the Main Menu, the user can also switch to the other main forms for viewing data and modifying reports. The first of these is the vessel list form which simply takes the vessel list table and makes it viewable in a form as shown in Figure 40.



IMO Number	Vessel Name	Shipping Company	Flag Nation	Vessel Type
9999984	Vessel1	Company1	Nation1	Oil/Chemical Tanker
9999985	Vessel2	Company1	Nation1	Oil Products Tanker
9999986	Vessel3	Company1	Nation1	Oil/Chemical Tanker
9999987	Vessel4	Company2	Nation1	Oil Products Tanker
9999988	Vessel5	Company3	Nation2	Oil/Chemical Tanker
9999989	Vessel6	Company3	Nation2	Oil Products Tanker
9999990	Vessel7	Company3	Nation2	Oil/Chemical Tanker
9999991	Vessel8	Company3	Nation2	Oil/Chemical Tanker
9999992	Vessel9	Company4	Nation3	Oil Products Tanker
9999993	Vessel10	Company4	Nation3	Oil Products Tanker
9999994	Vessel11	Company5	Nation4	Oil Products Tanker
9999995	Vessel12	Company6	Nation5	Oil/Chemical Tanker
9999996	Vessel13	Company6	Nation5	Oil/Chemical Tanker
9999997	Vessel14	Company7	Nation5	Chemical Tanker
9999998	Vessel15	Company7	Nation5	Oil/Chemical Tanker
9999999	Vessel16	Company8	Nation6	Oil/Chemical Tanker

Figure 40: Vessel List

There is also a link to add a vessel so the user can view the current list to determine if adding a new vessel is needed. There are also forms for displaying both the SWOMS and sounding book data in a similar fashion. However, as the related tables contain far more records then the Vessel List, it is impractical to show all of the data at once. There are combo boxes that allow the user to first select a specific IMO number, and then a start date and end date so they can view data within a certain date range as shown in the SWOMS Data Table in Figure 41.



Date	Time	Latitude	Longitude	Bilge Pump Runs	Bilge Pump Run Time	OWS Runs	OWS Run Time	Overboard Opened	Overboa
4/1/2011	12:00:00 AM	57 51.588' N	10 28.804' E	3	145	24	24.17	41	9.67
4/2/2011	12:00:00 AM	55 45.900' N	16 33.988' E	3	27.83	6	9.83	28	7.67
4/3/2011	12:00:00 AM	57 31.111' N	21 22.916' E	2	11.5	0	0	0	0
4/4/2011	12:00:00 AM	57 24.164' N	21 32.865' E	4	47.67	2	0.33	3	1.33
4/5/2011	12:00:00 AM	57 24.163' N	21 32.864' E	1	6	0	0	0	0
4/6/2011	12:00:00 AM	56 30.053' N	18 58.394' E	3	31.17	0	0	0	0

Figure 41: SWOMS Data Table

Additionally both forms have a link to the Vessel Details form in order to view the data for the specific ship that the user is currently viewing data from. The last, and perhaps most important, form is the Detail Reports form shown in Figure 42.

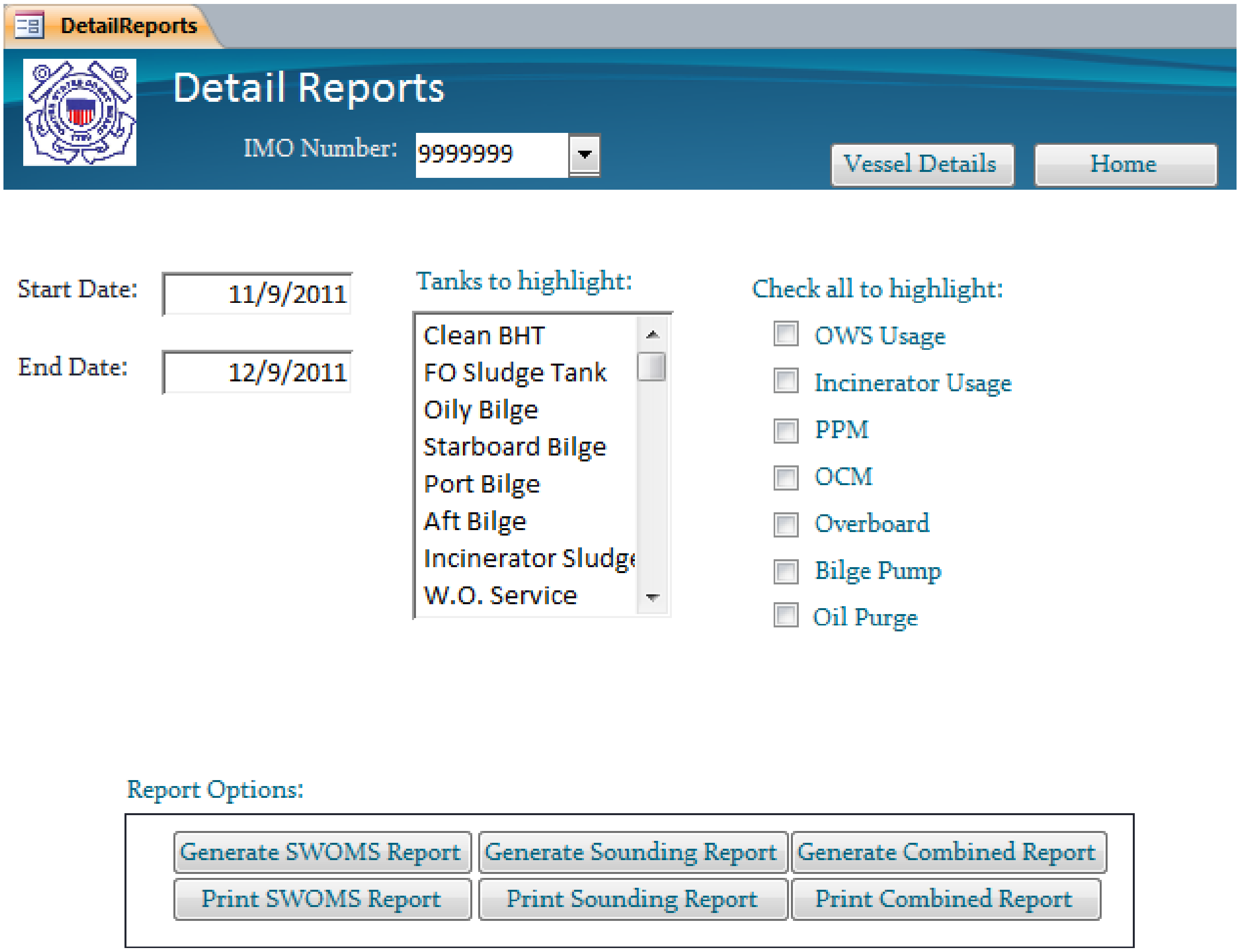


Figure 42: Detail Reports Form

This form allows the user to select any or all of the parameters shown in the reports, and then create a report based on a shorter and more focused data set. In addition to a date range, the user can also select which tanks they would like to see, as well as use of the OWS, Incinerator, OCM, and other data recorded by SWOMS. By default all of these are selected like they would be if a quick report was generated from the main menu and then the user can deselect whatever is extraneous.

5.5 Testing and Analysis

To ensure that this database system is dependable, the system went through extensive testing. Once all queries and reports were complete, the testing was able to begin. Each report was generated for fifteen vessels to determine any errors or elements that needed fixing. Of these fifteen vessels tested, four vessels are actual vessels under the program with real data. The other nine vessels are not real vessels in the program and the data are fake. These extra vessels were to test things such as having the most tanks possible, hourly data, if the data are the same for the sounding and the SWOMS data, as well as if there are extreme differences between the sounding and the SWOMS data.

5.5.1 Error Messages

The following section addresses several error messages and problems noticed or obtained during testing. These error messages and issues were fixed before the finalization of the database.

5.5.1.1 Date Range Error

Date ranges are an important part in determining what data to look at in the reports generated by the database. To create reports, an IMO and a date range is required. The following errors are related to entering date ranges.

5.5.1.2 Inaccurate Date Range

The first error that was determined was the data range error. The system was checked to see what would happen if the start data of the data range was more recent than the end data of the data range. This produced an error that can be seen below in Figure 43. This proved that there needed to be an error checking function to determine that the date range is logical, meaning that the end date is after the start date.

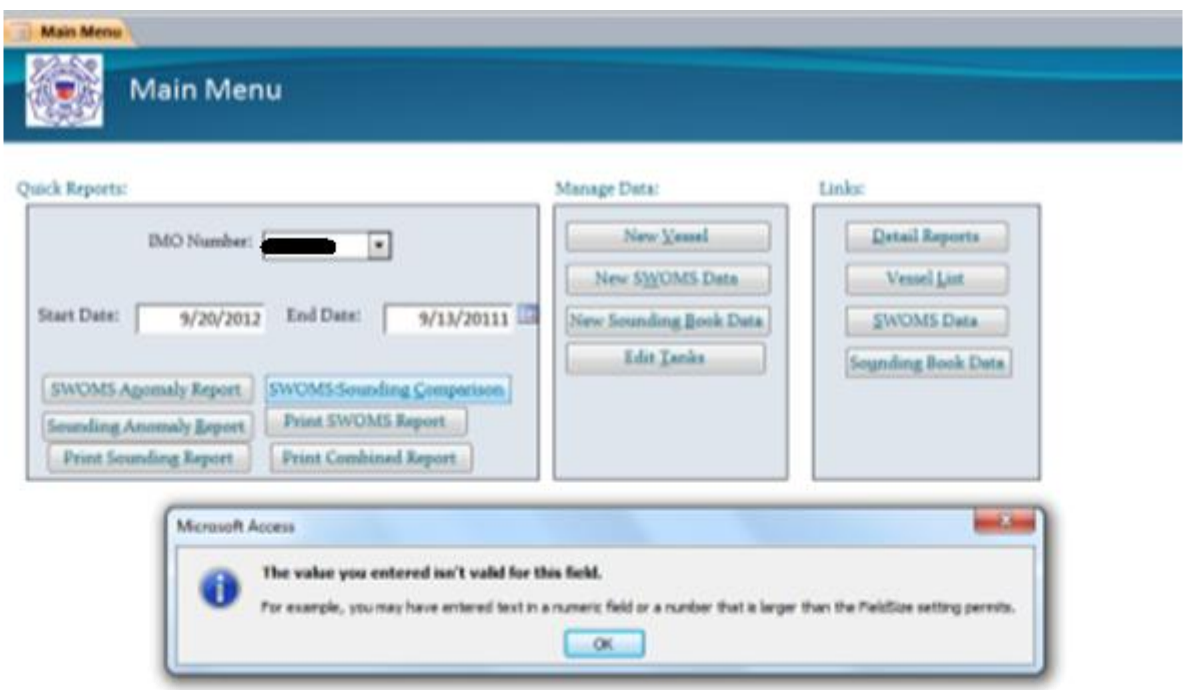


Figure 43: Data Range Error

5.5.1.3 Date Range Omitted

When the date range is omitted, a report will still be generated. This report will be a blank report with no data. Below, Figure 44 shows the main menu "Quick Reports" section without the data range completed.

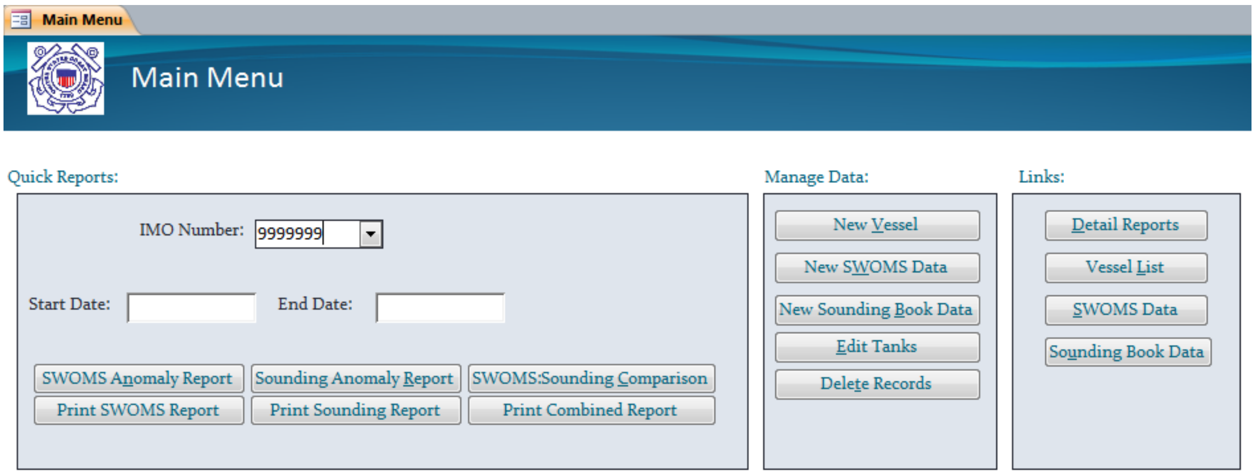


Figure 44: Date Range Omitted

Without a date range, the report will be generated as a report with no data. Figure 45 below shows a blank report that was the result of generating a report with no date range.

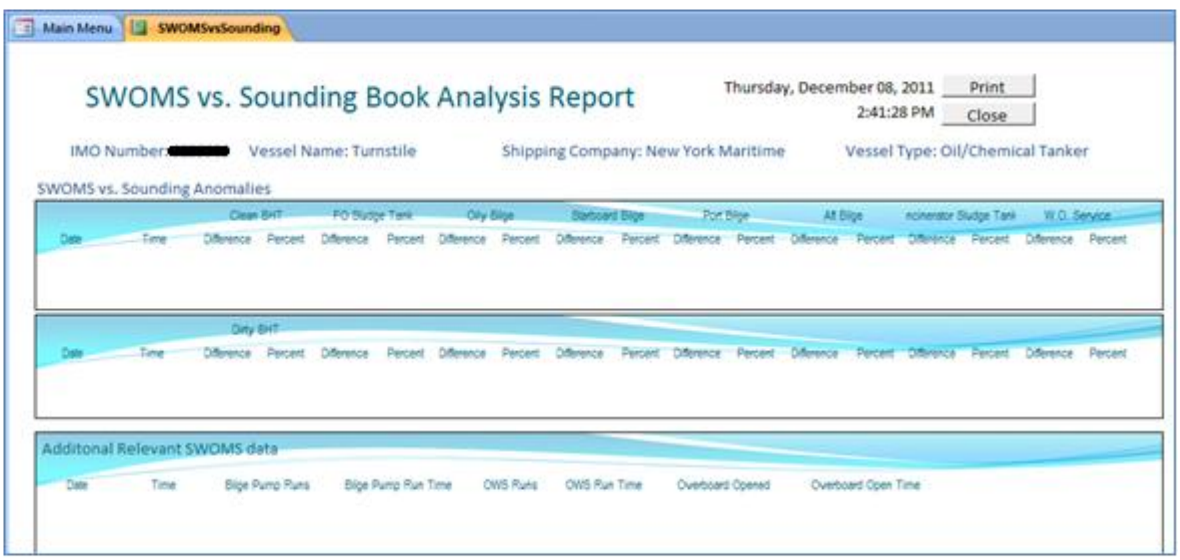


Figure 45: Blank Report with No Date Range

5.5.1.4 Close Button on Reports

When generating reports for different vessels it was confusing as to how to get out of the report. Closing the report by right-clicking on the tab, as can be seen in Figure 46, seems like it could delete useful information. Also, only a person fairly comfortable with Access might know that this is an option to close the report. These concerns led to the idea that inserting a “Close Report” button was necessary.

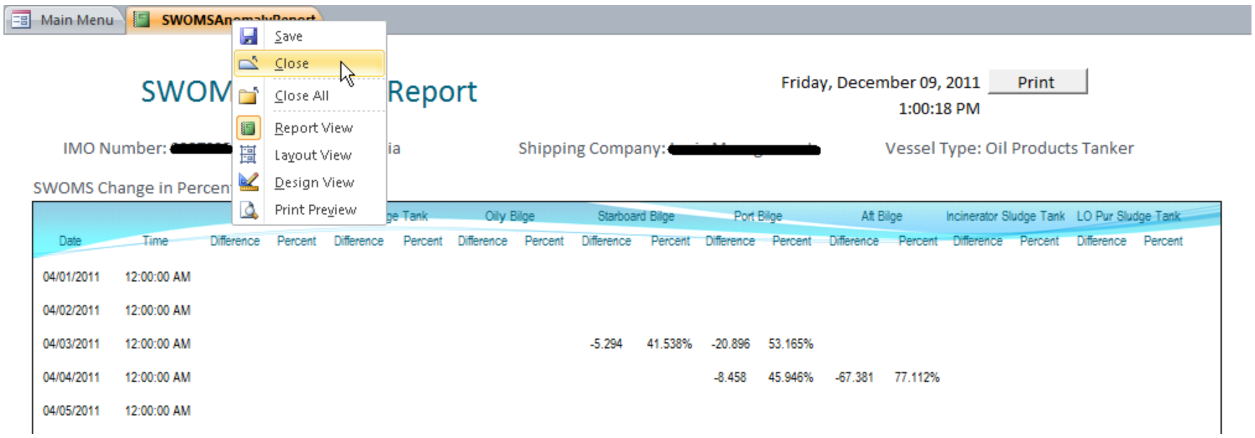


Figure 46: Closing Reports

Therefore, a close button was added to all three reports below the print button as shown in Figure 47. This button closes the report and, like the print button, is hidden when the report is printed.

Figure 47: Report with Close Button Inserted

5.5.2 Comprehensive Database Testing

To ensure that there is confidence in the product being given to the Coast Guard the database underwent comprehensive testing in addition to troubleshooting. This mostly applied to the generated reports. The testing began when all reports seemed to be in working form. There are fifteen vessels, of which 11 are fake vessels with fake data, and 4 are real vessels using actual data collected. The three separate reports can be generated for these vessels. So to test the reliability, each report was generated for every vessel.

5.5.2.1 Testing Trial 1

The testing was done in trials. The first trial was to determine any potential problems. The first comprehensive testing trial showed the result listed in Table 6 below.

Table 6: Trial 1 of Comprehensive Testing

Test Vessel	SWOMS	Notes	Sounding	Notes	Comparison	Notes
1	N		**N**	Blank	Y	
2	Y		Y		Y	
3	Y		Y		Y	
4	Y		Y		Y	
5	Y		Y		Y	
6	Y		**N**	Blank	Y	
7	Y		Y		Y	
8	Y		Y		Y	After conducting all reports to this point Access crashed
9	Y		**N**	Blank	Y	
10	Y		**N**	Blank	Y	
11	Y		Y		Y	
12	Y		Y		Y	
13	Y		**N**	Excessive loading time then Blank report	Y	
14	Y		**N**	Blank	Y	
15	Y		Y		Y	
All Vessels	%Success	93.33	%Success	60	%Success	100

The results showed that the only report that was 100% reliable was the SWOMS vs. Sounding Comparison Report. Otherwise, the SWOMS Report had a 93.33% success rate, only failing in one vessel; the Sounding report proved a 60% success rate, failing in a total of 6 vessels. Most concerning of these results was that the green highlighting in the left-hand column represents the real vessels, of which 4 out of 4 of the Sounding Reports failed. Though this test proved very successful with the SWOMS Report and the Comparison Report, the testing showed that there were still some errors to work out as far as the Sounding Reports went. In each of the Sounding Report failures, a blank screen would show up with the header "Sounding Report" but no data or formatting.

In response to the results of the first trial of comprehensive testing, modifications were made to the queries used to generate the reports. These changes simplified the way the queries decided which records to compare, and no longer looked at the time field to determine which records would be compared.

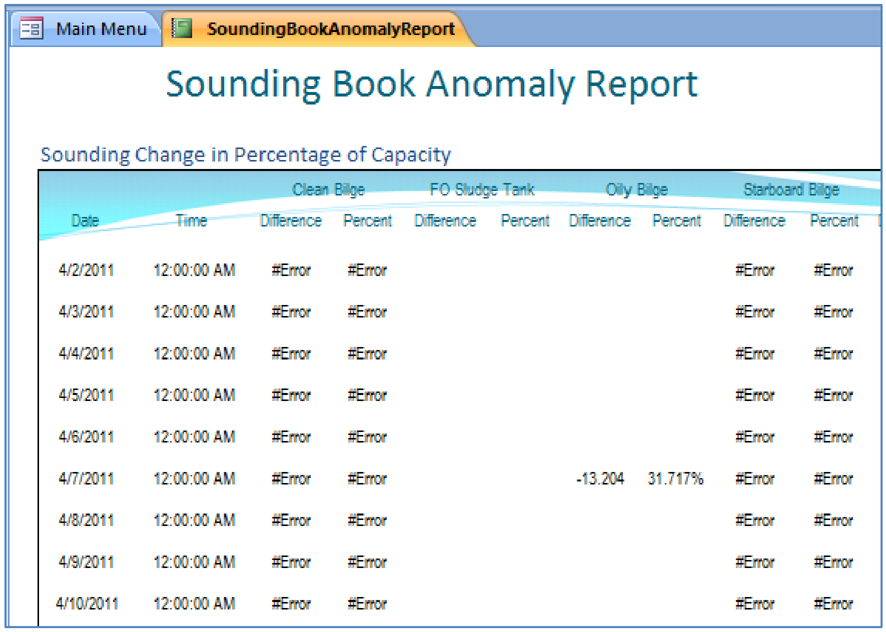
5.5.2.2 Testing Trial 2

After some modifications were made to some of the queries within the database, the second trial of testing was ready to occur. This trial focused on improvement in all of the report failures discovered in the first trial. The second trial proved to do just that and the percent success increased in all categories except for the SWOMS vs. Sounding Comparison Report which was already at a 100% success rate. The data for Trial 2 of comprehensive testing can be seen below in Table 7.

Table 7: Trial 2 of Comprehensive Testing

Test Vessel	SWOMS	Notes	Sounding	Notes	Comparison	Notes
1	Y		Y		Y	
2	Y		Y		Y	
3	Y		Y		Y	
4	Y		Y		Y	
5	Y		Y		Y	
6	Y		Y		Y	
7	Y		Y		Y	
8	Y		Y		Y	
9	Y		**N**	#Error	Y	
10	Y		**N**	#Error	Y	
11	Y		Y		Y	
12	Y		Y		Y	
13	Y		**N**	#Error	Y	
14	Y		**N**	#Error	Y	
15	Y		Y		Y	Identical SWOMS and Sounding data
All Vessels	%Success	**100.00**	%Success	73.33	%Success	**100.00**
Improvement since Trial 1	% Improvement	6.67	% Improvement	13.33	% Improvement	0.00

Though there were still a few report failures, Trial 2 proved to be an improvement still. For example, the success rate rose to 100% for the SWOMS Report making two reports with 100% success rates. Even though the Sounding Report still had four “failures”, the success rate rose by 13.33% to a 73.33% success rate from a 60% success rate and the problem was no longer a failure to generate report. Instead, in two tanks of the vessels highlighted in yellow an error message, “#Error”, would show up instead of tank calculations all down the column. An image of this error message can be seen below in Figure 48.



Main Menu | SoundingBookAnomalyReport

Sounding Book Anomaly Report

Sounding Change in Percentage of Capacity

		Clean Bilge		FO Sludge Tank		Oily Bilge		Starboard Bilge	
Date	Time	Difference	Percent	Difference	Percent	Difference	Percent	Difference	Percent
4/2/2011	12:00:00 AM	#Error	#Error					#Error	#Error
4/3/2011	12:00:00 AM	#Error	#Error					#Error	#Error
4/4/2011	12:00:00 AM	#Error	#Error					#Error	#Error
4/5/2011	12:00:00 AM	#Error	#Error					#Error	#Error
4/6/2011	12:00:00 AM	#Error	#Error					#Error	#Error
4/7/2011	12:00:00 AM	#Error	#Error			-13.204	31.717%	#Error	#Error
4/8/2011	12:00:00 AM	#Error	#Error					#Error	#Error
4/9/2011	12:00:00 AM	#Error	#Error					#Error	#Error
4/10/2011	12:00:00 AM	#Error	#Error					#Error	#Error

Figure 48: Sounding Book "#Error"

It was observed was that it was Tank01 and Tank04 for three out of the four reported failures. The “#Error” inferred that it could be a calculation error. After further analysis of the problem, it was noted that the query had an error anywhere that the tank volume was zero. In order to solve this problem the queries were modified so that a very small amount was added to the divisor. By adding this value the query never divided by zero, and yet the amount was small enough that it did not affect any of the calculations when the divisor was not zero.

5.5.2.3 Testing Trial 3

After additional modifications were made to some of the queries to solve the error message discovered in Trial 2, the third trial of testing was ready to occur. This trial focused on improvement in the error messages discovered in the first and second trials. The third trial proved to do just that, and the percent success was 100% in all categories. The data for Trial 3 of comprehensive testing can be seen below in Table 8.

Table 8: Trial 3 of Comprehensive Testing

Test Vessel	SWOMS	Notes	Sounding	Notes	Comparison	Notes
1	Y		Y		Y	
2	Y		Y		Y	
3	Y		Y		Y	
4	Y		Y		Y	
5	Y		Y		Y	
6	Y		Y		Y	
7	Y		Y		Y	
8	Y		Y		Y	
9	Y		**Y**		Y	
10	Y		**Y**		Y	
11	Y		Y		Y	
12	Y		Y		Y	
13	Y		**Y**		Y	
14	Y		**Y**		Y	
15	Y		Y		Y	
All Vessels	%Success	**100.00**	%Success	**100.00**	%Success	**100.00**
Improvement since Trial 1	% Improvement	0.00	% Improvement	26.67	% Improvement	0.00

This testing trial showed significant improvement for the success of the Sounding Report, jumping from 73% success to 100% success due to modifications in the queries to address errors. With all of the reports being generated 100% accurately, there was another trial to ensure consistency. However, during this round of testing, it was discovered that some of the anomalies generated had unexpected values. Upon further inspection it was discovered that these were in fact incorrect anomalies generated due to logic errors in some of the queries. In order to solve this all of the queries were worked through to find any errors. During this process, a more logical way for the queries to run was discovered. Implementing the new way to run the queries resulted in a more robust analysis that solved the errors as well as made it less likely for the queries to generate additional errors in the future. For further information on how the queries work, see Appendix K: User Manual.

5.5.2.4 Testing Trials 4 & 5

The fourth and fifth testing trials were run to ensure consistency in the reports generated as well as confidence in the program as a whole. These testing trials proved that all error messages previously found in the reports were addressed, as well as that once the reports reached a 100% success rate for a trial, the success rate was maintained for every successive test trial. All reports had a 100% success rate for being generated without any errors for the fourth and fifth trials.

5.5.2.5 Conclusion of Comprehensive Testing

The testing proved several errors in the queries for both the SWOMS Report and the Sounding Report. However, these errors did not appear for all vessels, and so without this comprehensive testing in which each report was generated for each test vessel, some of these errors could have gone unnoticed and caused problems in the future. The test vessels that were created for this purpose had different types of data including one vessel in which the Sounding and SWOMS data were identical to prove no return in discrepancies on the Comparison Report.

The SWOMS Report had one error in the first test trial and then all successive trials proved a 100% success rate. Figure 49 shows the success rates of the SWOMS Reports as they relate to the trial number.

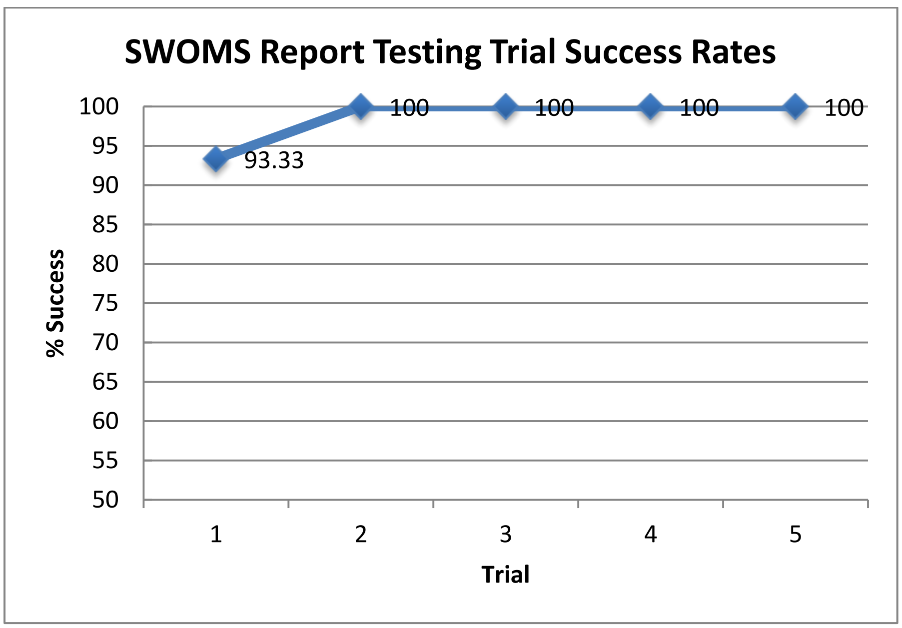


Figure 49: SWOMS Testing Trial Success

As it can be seen from the graph above, the modifications made after the first testing trials sufficiently addressed the error within the SWOMS query and report generation. This report has 4 straight trials of 100% success, which proves that this report is very reliable.

The Sounding Report, on the other hand, had several errors in the first few test trials. The errors were similar to that with the SWOMS Report in the first trial. The successive problems with the reports dealt with how the queries were dealing with tank levels of zero. Doing the calculations with the zeros was providing error messages. The queries were rewritten to address this issue. Figure 50 shows the success rates of the Sounding Reports as they relate to the trial number.

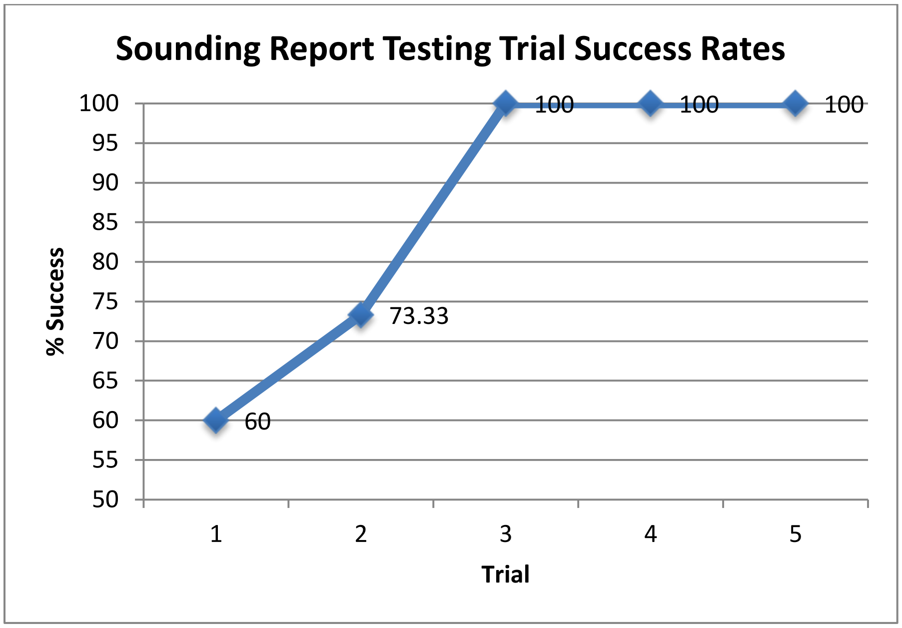


Figure 50: Sounding Report Testing Trial Success

As can be seen above in the graph above, the modifications made to the code behind the queries that were being calculated solved many of the error messages that were being displayed. These problems were only discovered after extensive testing, as these errors only came up in some vessels. This being said, the comprehensive testing trials were most important in ensuring the reliability of the Sounding Report.

After this comprehensive testing, all reports have shown to be reliable. All bugs and error messages were addressed until a 100% success rate was achieved for two or more successive testing trials. The Comparison Report proved to be 100% successful in all testing trials. The testing done with all of these reports proves the reliability of the system.

5.6 User Manual

A user manual was written in conjunction with the development of the data transmission system and database. The first section of the manual includes information on how to use the parsing program. The next gives detailed instructions for how to navigate the database in order to add, edit, and analyze the data. Finally, the manual gives information about the coding aspects of the parsing program and the database in order to allow for future maintenance or troubleshooting. The code for these items is explained such that a technical person will be able to perform any necessary troubleshooting or incorporate additional functions. To see the user manual, refer to Appendix .

5.7 Summary

In summary, after studying the current process for data transmission and analysis, it was very evident that a new process was needed. The new proposed process aims to improve the areas in which the current process is lacking, such as better means for analysis and more standardization. With the new process, there are standardized templates for recording the SWOMS and sounding book data, as well as a standard method for transmitting this data to the Coast Guard. With the parser program and Exchange mailbox, gathering this data will be far easier. With the Access database completed, the Coast Guard has one program and interface for gathering, viewing and analyzing all the necessary data. With the considerable testing that has gone into this software, the Coast Guard can be confident that they have an accurate and robust product.

6 Recommendations and Conclusions

For this project, several deliverables were set forth to help achieve the overall goal. These deliverables were all completed as specific items to be given to United States Coast Guard Headquarters, Office of Vessel Activities (CG-543) and Office of Investigations and Casualty Analysis (CG-545). The research and analysis conducted in order to complete these deliverables are explained in the previous chapter. The conclusions that were determined throughout the development of these deliverables, as well as specific recommendations, are presented below.

6.1 Recommendations for Implementation

This project produced a standardized submission process for future environmental compliance plans as well as a database, which can store and analyze the data received as terms of the environmental compliance plan. The following are recommendations pertaining to the implementation of the created system.

6.1.1 Standardized Submission Process

Through extensive research of the current data received, the acquisition of the data, and the current analysis process, it was determined that a standardized submission process is necessary. This means that the data would have to be submitted as mandated by the DoJ. The SWOMS data would be received in automatically-generated daily emails sent by the onboard system. The sounding log spreadsheet, along with the scanned sounding log and ORB would have to be submitted via email every two weeks. Having deadlines for data submissions and templates to be submitted creates very specific responsibilities for the vessels and the shipping companies and will hold them accountable to meet these standards. These more frequent data submissions will also allow for quicker turn-around time for data analyses that could potentially prove illegal behaviors and warrant legal action. Since there is a quicker turn-around time of these analyses, legal action will be able to be taken before vessels' probations end. In addition, a standardized process would guarantee uniform data that would be easier for analysts to interpret. It will also allow quantifiable analysis to be performed the same way for all vessel data and will ensure that all SWOMS data are compatible with the Environmental Compliance Plan Database.

6.1.2 Environmental Compliance Database

Research regarding the previous storage and analysis methods for SWOMS data revealed a definite need for a better system. The immediate solution was to create a database with the ability to acquire, store, and analyze all SWOMS data sent to the Coast Guard. The time required to perform an analysis of the SWOMS data is now minutes instead of the several hours that it previously took. The database can also generate various analytical reports: for just the SWOMS data, for just the sounding data, or for comparisons between the two. This gives the users many options as to what kind of an analysis they would like to see. For clarity, even if the Coast Guard does not choose to implement the Standardized Submission Plan, the database will be able to work with the existing SWOMS data for generating SWOMS analysis reports.

6.2 Recommendations for Future Work

The research and development of the ECP database and standardized transmission process has led to the creation of several recommendations which should improve the overall efficiency of the database and analysis beyond the functionality achieved during the course of this project.

6.2.1 Initial Vessel Implementation Procedures

In the event that the Coast Guard chooses to use the new standard submission process designed, then it is recommended that the initial implementation of the process onto new vessel is done carefully. The installation of SWOMS on board a vessel has previously taken several months due to issues such as vessel location or incompatibility with pre-installed systems on board. Even after the SWOMS equipment is installed and working, further time is required to calibrate the readings. This delay undermines the Coast Guard's efforts to check the compliance of the vessel during the beginning of its environmental compliance plan. Requiring the SWOMS system to be online and functional within one month would discourage non-compliance more quickly. Once the SWOMS is fully working, the vessel can begin automatically sending the system's data to the Coast Guard using emails. The Coast Guard is then able to automatically import data from these emails into the database using specialized software.

The database uses generic identifiers for the individual tanks (e.g. "Tank01"), which could lead to confusion for comparison of the SWOMS tank data with the sounding log book tank data. Taking these details into consideration, we recommend that the Coast Guard be responsible for filling out a tank identification template once they receive the first SWOMS email from each vessel. The template would list each tank, including its type, description, and capacity, in the order presented in the automatically-generated emails. From there, the Coast Guard could request that each vessel completes their sounding book in the same sequence that the tank identification template indicates in order to decrease confusion regarding data entry into the sounding book template. An alternative would be for each vessel to fill out the handwritten sounding book as the vessel desires but to have them fill out the sounding log book spreadsheet template to be sent to the Coast Guard in the sequence noted on the tank identification template. Either way, the Coast Guard will be responsible for sending the Tank Identification Spreadsheet along with the Sounding Spreadsheet Template to the shipping company. The vessels will be responsible for filling out the template in the indicated order regardless of the order of the handwritten sounding book. It is suggested that the Coast Guard pays special attention to the first several data submissions to make sure that the tank comparisons are accurate. If the SWOMS vs. Sounding Report consistently indicates a high percentage of deviation, it may simply be indicative of an incorrectly ordered sounding log spreadsheet.

Potential falsification of sounding log books is also a concern when attempting to verify compliance using that data. In order to ensure that ships are complying with MARPOL regulations and the terms of their probation, audits should randomly occur throughout the first year following the installation of SWOMS. These audits should examine the SWOMS, oil record book, and sounding book data more thoroughly than the database-generated reports. This should minimize the number of non-compliant vessels that are able to avoid detection.

6.2.2 Database / User Interaction

In order to reduce user confusion regarding how to use the database to enter data or generate analytical reports, we recommend that the database include clear instructions in addition to the existing user manual. This could include having a help menu, instructions on the various forms, tooltips, etc. Even with these additions, we recommend that the Coast Guard implements a policy requiring users of this Environmental Compliance Plan database to undergo a brief training period before they may enter or edit information. The functionality of the database is reliant on the data being entered accurately and in the correct format and location, meaning that errors could potentially cause inaccurate analyses or even malfunctions in the database itself.

An additional safeguard against inaccurate or incomplete data being entered into the database would be the ability to track which users have modified data, identify which data were changed during a session, identify when data were changed, and restrict which users are able to modify certain information. Access has many of these features built-in frameworks for, which we were unable to fully implement in time.

Additional features that we would have liked to add, but were unable to include the ability for the user to generate reports by company, as well as have the reports reformat if you choose to hide data. Both features would allow for more useful report generation and should be added given the opportunity. Allowing the user to select multiple IMO numbers on the detail reports form, with the ability to filter the list of IMO numbers by a company if they wanted, would be the simplest way to allow report generation by company as the existing queries and reports should be able to handle extra data that way, without the need to generate another set of reports and queries to generate company-wide data. In order to better compact reports if the user chooses to hide data would simply require VBA code that would move the next text box to the left if the previous one is hidden, which while simple in concept is time-consuming to code which is why we were unable to implement in time for the completion of this project. There are other formatting issues as well, as sometimes the reports do not expand properly with large amounts of data, which can result in some data being cut off when viewing reports in access. The reports will still print out properly, but additional work should be done to make it easier for the reports to be viewed without the need for printing. Additionally, there are probably ways to make the queries and VBA code more efficient which would assist in making report generation a faster process without the need to upgrade the processing power of Coast Guard computers or upgrading to Access 2010, though both would help with performance issues if there is the opportunity to upgrade.

6.2.3 Improvements on Automation

One goal for this project was ease of use for those individuals using this system. Increased automation of the data acquisition process would decrease the need for manual data entry, thereby reducing the risk of inaccurate data entry and lessening the labor involved in processing data. This automation process has been started with the development of the swomsparser program and the email rules and scripting that automatically parse the SWOMS data into database-usable form.

The Outlook automation allows the Coast Guard to receive emails from ships and add their contents into the ECP database. Further improvement is possible here. From empirical data, it is clear that there is little chance for error in data formatting coming from the SWOMS. On the other hand, it would not significantly impact processing time to perform better data verification. This may reduce the risk of misreading entries, for example when two pieces of data are labeled the same but with different units. In addition, sending read receipts when emails are received would give shipping companies proof that the Coast Guard receives their data. This would help in the event of a dispute in which the Coast Guard claimed that the data was not received from the vessel, as these receipts would be able to prove whether the company actually sent the data to the Coast Guard.

In addition to data acquisition, users are highly involved in the data analysis verification. The automation routines cannot account for all variables, make the judgment call to say whether a vessel is in violation or not, or verify data entry errors. One specific recommendation we have for this limitation is to link to the ORB PDF file from within the database, allowing the user to immediately access the hand-written ORB data upon the discovery of inconsistencies in the data. This could save the user the trouble of locating the PDF manually, and reduce overall analysis time.

A system for ORB data inclusion and sounding book data would also be beneficial to the system. Presently, some work has been done for standardizing the sounding book data for database import, but there is no way to automate the acquisition process. Adding ORB data would require the automation program/script to be able to identify which ship the ORB data matches up with, and it would then need to properly synchronize this with the database. For associating sounding book data, the program/script would need to verify that the data was accurate before importing to the database. Since sounding book data are entered by hand and sent manually, there is a chance that emails will be sent with incorrect subjects or with improper formatting. This is the major reason that human interaction is necessary in acquiring this data now.

6.2.4 Exchange/Outlook Integration

A major element in this project is the integration of Outlook and an Exchange mailbox for receiving data from vessels and from shipping companies. Although a public Exchange folder was created, public folders are not well-suited to automation. A working prototype of the system was tested in a personal mailbox, and should be portable to a service account with only a few changes to the Outlook VBA coding. Since public folders are not capable of providing the automation that is desired, it is recommended that a virtualized server be set up with an Exchange shared/service account running Outlook.

It is recommended that the Coast Guard IT group be contacted to assign a designated maintenance sub-administrator within CG-543 and/or CG-545 that could make modifications to code elements as necessary. This person would need to have privileges to modify the VBA code running in Outlook on the virtualized server.

6.2.5 Parsing Program Modifications

C programming is low-level by nature. This language was chosen entirely because of personal familiarity. Another language may be able to perform the same tasks as are performed with the swomsparser program with improved file handling. Whether rewritten in another language or using C, the swomsparser program should have a way to modify its data collection and output formatting. This could be done using a simple configuration file, for example a window “ini” file, or by using registry entries, or even by designing the program to use very many command-line arguments. There are multiple ways to go about this task. Another improvement would be the creation of an optional GUI that would allow for more expanded use. The swomsparser in its current implementation is designed to achieve one specific task consistently. There are many additional features that could be added in the future if necessary, though for the purpose of this project there was not enough time for further development.

6.2.6 Future Database Expansion and Modifications

An important consideration for the database is its longevity, which will increasingly become an issue as the size of the database continues to grow. The Coast Guard hopes to increase the frequency of the entries from daily records to hourly ones. They also hope to one day be able to implement this SWOMS system on all ships rather than just non-compliant ships, and use it as a preventative measure rather than as a punitive one. The database in its current form simply will not be able to hold that much information. In order to increase the longevity of this database, we recommend dividing the tables into more manageable sizes, to circumvent the maximum level of records per table. An archiving system would also help to reduce the overhead. Ultimately, our recommendation is to use this database as the basis for a larger and more powerful database, one which is greater than the scope of this IQP allows. This database would optimally be located on an enterprise server, which would allow use of a web interface to make changes. This would also mean that the data are stored in a "cloud" rather than stored on a computer's hard drive.

In the shorter term, something that should also be considered is having tables linked to the files generated by the parsing program. This would allow, with some modifications to the parser and outlook programs, the ability for SWOMS data to be put into one large file by the parser, which is linked to the SWOMS data table in the database. This means that as soon as an email is received and processed by outlook, the data would go into this master SWOMS file, which would then automatically update the table in the database, making SWOMS data available in real-time in Access. This would remove manual uploading by users and make it even easier to analyze this data.

6.3 Conclusion

Prior to the completion of this project, the systematic analysis of waste-oil data from ships was laborious and time-consuming. The completion of this database and standard submission plan should greatly improve the Coast Guard's ability to verify the compliance of vessels with oil-waste regulations. Having developed deadlines for data submission as well as a very specific format of the data, the Department of Justice will have clean cut guidelines for future environmental compliance plans. These specific guidelines for future environmental compliance plans will allow for easy integration of new vessels added to the program into the database. The standard submission plan will make for easy analysis since the SWOMS data and the sounding logbook data will be in the format needed for the database in a shared folder. This will allow for all Coast Guard personnel with access to the folder to easily upload data to the database. This will make for a seamless import of the SWOMS and sounding data into the database for immediate analysis. With several different kinds of reports that the database is able to generate, the Coast Guard may look at the overall status of the vessel and then look into potential problems with more detail. Being able to receive, store, and analyze this data within one system will allow for quicker analyses that will help to expedite legal actions to extend probations if necessary. There is still room for expansion, but the initial steps have been put into place.

References

Allain, R. (2010). LCDR. Proceedings of the Marine Safety & Security Council, (4) Retrieved from http://www.uscg.mil/proceedings/Winter2008-09/articles/73_Allain_MARPOL%20Annex%20I.pdf

Amazon Web Services. (2011). *AWS case study: Alexa.* Retrieved 09/25, 2011, from http://aws.amazon.com/solutions/case-studies/alexa/

Barrett, E. J. (1998). Commandant Notice. Retrieved on 9/26/11 from http://www.uscg.mil/directives/cim/16000-16999/CIM_16478_1B.pdf

Berg, Nicholas H. (2010). Bringing It All Back Home: The Fifth and Second Circuits Allow Domestic Prosecutions for Oil Record Book Violations on Foreign-Flagged Ships. Tulane Maritime Law Journal. 34(1), 253. Retrieved on 9/14, 2011 from https://litigation-essentials.lexisnexis.com/webcd/app?action=DocumentDisplay&crawlid=1&doctype=cite&docid=34+Tul.+Mar.+L.+J.+253&srctype=smi&srcid=3B15&key=876d766ae34b18ca56a9f755ee4b23de

Boat Nerd. (2007). *Clipper Trojan.* Retrieved 09/29, 2011, from http://www.boatnerd.com/news/newsthumbs/images-07-4/ClipperTrojan11-11-07-jm.jpg

Butt, N. (2007). The impact of cruise ship generated waste on home ports and ports of call: A study of Southampton. *Marine Policy, 31*(5), 591-598. doi: 10.1016/j.marpol.2007.03.002

Chief of Staff, USCG. (1997). *Criminal enforcement of environmental laws.* Retrieved 09/11, 2011, from http://www.winston.com/ftp/Coast%20Guard%20Criminal%20Enforcement%20of%20Environmental%20Laws%2007-30-97.pdf

Clipper Group. (2007). *Clipper reaches agreement with US Department of Justice in oily water case.* Retrieved 09/09, 2011, from http://www.clipper-group.com/web/cmsresources.nsf/filenames/ClipperPressRelease_ClipperTrojan_19-06-08.pdf/$file/ClipperPressRelease_ClipperTrojan_19-06-08.pdf

Conference of the Parties to the Basel Convention. (2010). Basel Convention on the Control of Transboundary Movements of Hazardous Wastes and their Disposal. Retrieved from http://www.basel.int/text/documents.html

Department of Justice. (2007, December 14). *Tanker Company Fined $4.9 Million For Falsifying Records and Obstruction of Justice* [Press release]. Retrieved on 9/14, 2011 from http://www.justice.gov/opa/pr/2007/December/07_tax_1012.html

District of Connecticut. (2010). Retrieved from http://protect.theinfo.org/pacer/ecf.ctd/04112209348.pdf

EPA Watershed Academy. (2008). *Introduction to the Clean Water Act.* Retrieved 09/18, 2011, from http://www.epa.gov/owow/watershed/wacademy/acad2000/cwa/rightindex.htm

Gartner, Inc. (2006). *Gartner says worldwide relational database market increased 8 percent in 2005* [Press Release]. Retrieved 9/25/11 from http://www.gartner.com/it/page.jsp?id=493002

Geraci, A. K., Barbera, J., Katki, F., Lane, J. B., McMonegal, L., Meyer, B., et al. (1990). *IEEE Standard Glossary of Data Management Terminology.* The Institute of Electrical and Electronics Engineers, Inc. doi:10.1109/IEEESTD.1990.94601

Guidelines and Specifications for Pollution Prevention Equipment for Machinery Space Bilges of Ships, Resolution MEPC.60(33) (1992). Retrieved from http://www5.imo.org/SharePoint/blastDataHelper.asp/data_id%3D15634/60%2833%29.pdf

Haynes, W. (1996). *T.S. patriot state engineering manual* (Manual Massachusetts Maritime Academy). Retrieved on 9/24/11 from http://weh.maritime.edu/campus/tsps/manual/BilgeBal.html

Höfer, T. (1999). Review articles: Tankships in the marine environment. *Environmental Science and Pollution Research, 6*(1), 21-28. doi:10.1007/BF02987117

International Conference on Marine Pollution: International Convention For The Prevention of Pollution from Ships. (1973). *International Legal Materials, 6*(12), 1319-1444.

International Maritime Organization. (2001). Procedures for port state control. London: International Maritime Organization. Retrieved 11/4/11

International Maritime Organization. (2006). MARPOL consolidated edition 2006: articles, protocols, annexes, unified interpretations of the International Convention for the Prevention of Pollution from Ships, 1973, as modified by the Protocol of 1978 relating thereto. London: International Maritime Organizion. Retrieved 12/27/11

International Maritime Organization. (2011). IMO identification number scheme. Retrieved 11/30, 2011, from http://www.imo.org/OurWork/Safety/Implementation/Pages/IMO-identification-number-scheme.aspx Katsioloudis, P. J. (2010). Green ships: Keeping oceans blue. *Technology Teacher, 69*(5), 5-9. Retrieved on 9/24/11 from http://search.ebscohost.com/login.aspx?direct=true&db=8gh&AN=47758683&site=ehost-live

Kochi international marina. (2011). Retrieved 09/20, 2011, from http://kochimarina.blogspot.com/2011/02/features-of-kochi-international-marina.html

Kyoto protocol takes effect. (2005). *Global Environmental Change Report, 17*, 1-1-3. Retrieved from http://ezproxy.wpi.edu/login?url=http://search.proquest.com/docview/219553619?accountid=29120

Lin, B., Lin, C., & Jong, T. (2007). Investigation of strategies to improve the recycling effectiveness of waste oil from fishing vessels. *Marine Policy, 31*(4), 415-420. doi: 10.1016/j.marpol.2007.01.004

Milbourn, C. (2008). New law bolsters U.S. efforts to make ocean-going ships cleaner [Press Release]. *EPA.* Retrieved 9/15/11 from http://yosemite.epa.gov/opa/admpress.nsf/6424ac1caa800aab85257359003f5337/f1e6594e8e04fdd88525748e0069fb1f!OpenDocument

Minerals management. 33 C.F.R. § 106.201 (2011). Retrieved from GPO Catalog of U.S. Government Publications database.

Montgomery, C. (2006). Polluting ship goes unpunished despite crew member's report. Times Colonist, April 18, pp. A2.

Nguyen, L. (2010). *Freight transportation: Global highlights.* Bureau of Transportation Statistics. Retrieved 9/9/11 from http://www.bts.gov/publications/freight_transportation/pdf/entire.pdf

Office of the Law Revision Counsel, US House of Representatives. Coast Guard Program, U.S.C. 33U.S.C. §1953 (2011). Retrieved from http://uscode.house.gov/uscode-cgi/fastweb.exe?getdoc+uscview+t33t36+1115+63++%28Coast%20Guard%20Program%29

Office of the Law Revision Counsel, US House of Representatives. Maritime Pollution Prevention Act, U.S.C. 1901U.S.C. 1901 (2008). Retrieved from http://uscode.house.gov/uscode-cgi/fastweb.exe?getdoc+uscview+t33t36+1096+11++(Maritime Pollution Prevention Act)

Organisation for Economic Cooperation and Development (OECD). (2003). Cost Savings from Non-Compliance with International Environmental Regulations in the Maritime Sector. Retrieved 9/24/11 from http://www.oecd.org/dataoecd/4/26/2496757.pdf

Raunekk. (2010). The most important engine room documents a ship cannot sail without. Retrieved 11/10, 2011, from http://www.marineinsight.com/marine/the-most-important-engine-room-documents-a-ship-cannot-sail-without-3/

Revised Guidelines and Specifications for Pollution Prevention Equipment for Machinery Space Bilges of Ships, Resolution MEPC.107(49) (2003). Retrieved from http://www.gea-westfalia.no/arch/_img/9042813.pdf

Robson, D. (2002). T/V Kings Pointer Tank Diagram. In United States Merchant Marine Academy. Retrieved October 27, 2011, from http://www.usmma.edu/waterfront/kingspointer/Ships%20Publications/Tank%20Diagram.jpg

Round Table of international shipping organizations. (2010). Shipping facts. Retrieved 09/12, 2011, from http://www.marisec.org/shippingfacts/home/

Sherman, T. (2009), Federal authorities crack down on sea-borne oil polluters. Star-Ledger, Retrieved from http://www.nj.com/news/index.ssf/2009/05/federal_authorities_crack_down.html

Shipping Spotting. (2010). *MT Kriton.* Retrieved 09/29, 2011, from http://www.shipspotting.com/gallery/photo.php?lid=1157131

US Army Corps of Engineers. (2009). *US port rankings by cargo tonnage.* Retrieved 09/06, 2011, from http://aapa.files.cms-plus.com/Statistics/2009US_PORTRANKINGS_BY_CARGO_TONNAGE.pdf

US Coast Guard. (2011A). *Prevention through people partnerships.* Retrieved 09/15, 2011, from http://www.uscg.mil/hq/cg5/cg5211/partner.asp

US Coast Guard. (2011B). *U.S. Coast Guard: United States Department of Homeland Security.* Retrieved 09/02, 2011, from http://www.uscg.mil/

US Congress. Act to Prevent Pollution from Ships, U.S.C. 33U.S.C. §1901 (2000). Retrieved from http://epw.senate.gov/atppfs.pdf

US Congress. Clean Water Act, U.S.C. 33U.S.C. §1251. (2002). Retrieved from http://epw.senate.gov/water.pdf

Appendix A. United States Coast Guard

The United States Coast Guard is a branch of the U.S. armed forces. It lies under the Department of Homeland Security (DHS), along with FEMA and the U.S. Secret Service (US Coast Guard, 2011B). The mission of the USCG is to ensure maritime safety, coastal security, and environmental policy compliance in maritime regions. This is carried out in myriad ways, including saving lives at sea, protecting the environment, maintaining navigational tools, and defending our coastal borders. The USCG has a significant amount of manpower, as well as financial support from Congress which it needs to complete the diverse and massive amount of missions it is responsible for. As of 2011, there were over 40,000 men and women on active duty, in addition to the over 8,000 civilians serving the USCG. The 2011 budget allocated $10.1 billion to the USCG.

This project was completed under the Assistant Commandant for Marine Safety, Security and Stewardship (CG-5) seen in Figure 51 below (US Coast Guard, 2011B).

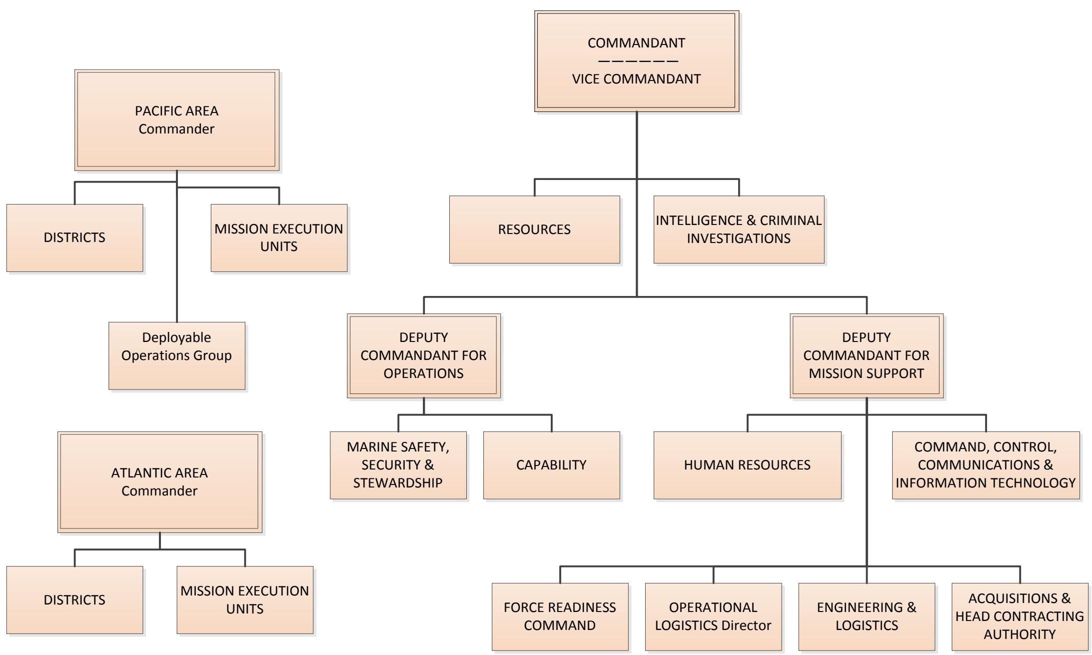


Figure 51: Organization of the United States Coast Guard (US Coast Guard, 2011B)

The program, complemented by the actions of the Office of Vessel Activities (COMDT CG-543), creates and enforces regulations dealing with invasive species, illegal dumping in the ocean, as well as preventing oil and chemical spills. See Figure 52 for an organizational chart for CG-5.

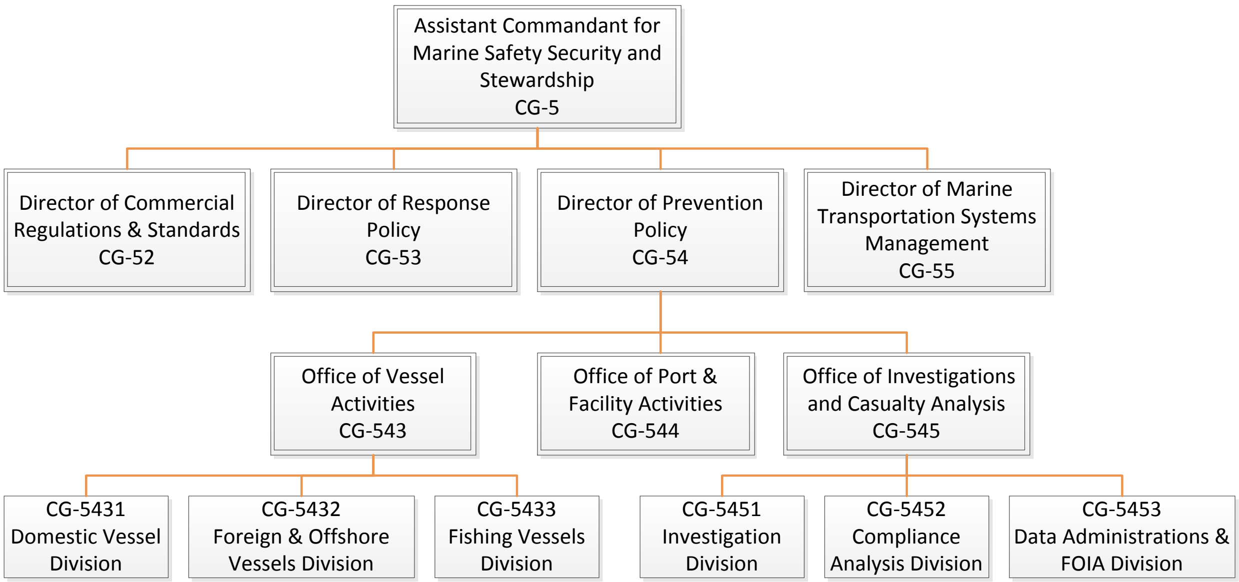


Figure 52: Organization of CG-5

Our project lies in the Office of Vessel Activities (COMDT CG-543). Much of the work they do relates to establishing policy and guidance for the enforcement of international treaties and domestic regulations by developing advanced strategies that cultivate the implementation of marine safety, security and environmental protection standards for all vessel operations and compliance Their mission is to "eliminate the operation of substandard vessels in US waters" by promoting commercial vessel safety, security, and environmental compliance programs (US Coast Guard, 2011B).. CG-543 is responsible for establishing policy to enforce the international and domestic standards. The Office of Vessel Activities (COMDT CG-543) employs 46 Military, 23 Civilians, 2 Reservists, and 16 Contracted Employees.

The project is also done in conjunction with the Office of Investigations and Analysis (COMDT CG-545). The office contains three divisions: Investigations Division (CG-5451), Analysis Division (CG-5452), and the Data Administration and FOIA Division (CG-5453) (US Coast Guard, 2011B). Their mission is to "lead the Coast Guard's investigation program to promote safety, protect the environment, and to prevent future accidents." This is the office that is in charge of managing the pollution investigation programs.

We were also assisted by the Marine Safety Center. The Marine Safety Center (MSC) supports the people and objectives of the Marine Safety, Security, and Environmental Protection programs through the verification of compliance with technical standards for the design, construction, alteration and repair of commercial vessels. The MSC is an independent Headquarters command unit that was established in 1986 by consolidating the Coast Guard Merchant Marine Technical offices located in New York, New Orleans, Cleveland and San Francisco. The MSC's primary mission is the review and approval of plans for the design, construction, alteration and repair of U.S. and foreign flag commercial vessels subject to the U.S. laws, regulations and international standards. The MSC has a current complement of 28 Officers, 23 Civilians, 2 Reservists, and 16 Contract Employees.

Appendix B: How This Project Qualifies as an IQP

WPI expects their students to learn how science and technology are embedded in the fabric of society. The Interactive Qualifying Project (IQP) accomplishes this by having students address a problem that lies at the intersection of science and technology with society and addresses human needs. The goal of this particular IQP was to create a standardized system for the Coast Guard that would acquire, store, and analyze data collected from ships. The system is required to analyze digitally generated data as well as manually recorded data. The analysis process includes translating the oil-waste data from a human-readable format into a database-readable format for examination, and then back into a human-readable format clearly highlighting any discrepancies and inconsistencies in the data. This will greatly help the men and women of the Coast Guard monitor what vessels are releasing into the ocean which should also help to discourage ship pollution. The use of technology to facilitate a more automated and efficient analysis process for the Coast Guard is what makes this project an IQP.

Appendix C: Interview Protocol

USCG #1: Environmental Compliance

Interview Protocol

Introduction:

You have been selected to speak with us today because you have been identified as a professional with a significant knowledge relevant to our project. The project's final goal is to develop a system for environmental waste oil release tracking. In the pursuit of this goal, the project team will be collecting additional information as to the background of the project, the needs of users of the eventual system, and any other information that would be useful in the course of the project.

Interviewee Background

Current position: ______________________________

Highest degree: ____________________________________

Field of study: ____________________________________

Relevant previous work (brief):

Questions for SWOMS Reviewers:

- How often do you review SWOMS data?
- What things do you look for when reviewing the data?
- What are the most important aspects of the data?
- When reviewing the data, do you have ORB readings?
- What sorts of incompliancy can you conclude from reviewing the data?
- How do you receive the data? (Ex. CD)
- How often does the data have to be sent in? How often does it usually come in?
- How long do you spend looking at one set of data?
- Do you respond to the ships with concerns? The shipping companies?
- Do you only respond when there is a major compliancy issue?
- Generally, how does the review process work?
- What do you dislike about the current review process?
- What do you like about the current review process?
- What would you like see in a database to help with the review process?

Appendix D: Shipping Data

Table 9: 25 Largest Container Shipping Companies Worldwide

Rank	Carrier	Company	Country of owner	Fleet size (TEUs in thousands)	Ships
1	APM-Maersk Line	A. P. Moller-Maersk Group	Denmark	2,035	539
2	Mediterranean Shiping Company	Mediterranean Shipping Company S.A.	Switzerland	1,509	406
3	CMA CGM	CMA CGM Group	France	1,021	359
4	Evergreen Line	Evergreen Marine	Taiwan	589	160
5	APL	American President Lines	Singapore	550	139
6	Hapag-Lloyd	Hapag-Lloyd	Germany	469	116
7	COSCO Container Line	China Ocean Shipping Company	China	466	144
8	CSCL	China Shipping Container Lines	China	461	140
9	NYK	Nippon Yusen Kaisha	Japan	413	109
10	Hanjin Shipping	Hanjin Shipping	South Korea	409	92
11	MOL	Mitsui O.S.K. Lines	Japan	343	93
12	K Line	Kawasaki Kisen Kaisha	Japan	331	90
13	OOCL	Orient Overseas Container Line	Hong Kong	329	71
14	Hamburg Süd Group	Wallenius Wilhelmsen Logistics	Sweden	326	108
15	Yang Ming Line	Mediterranean Shipping Company S.A.	Switzerland	314	78
16	CSAV Group	Compañía Sud Americana de Vapores	Chile	294	89
17	Zim	Zim Integrated Shipping Services	Israel	271	88
18	Hyundai M.M.	Hyundai Merchant Marine	South Korea	266	52
19	PIL	Pacific International Lines	Singapore	191	107
20	UASC	United Arab Shipping Company	Kuwait	171	45
21	Wan Hai Lines	Wan Hai Lines	Taiwan	130	68
22	IRIS Lines	Islamic Republic of Iran Shipping Lines	Iran	108	65
23	Orient Overseas International Line	Orient Overseas International Limited	Hong Kong	100	33
24	Sea Consortium	Sea Consortium	Singapore	57	52
25	RCL	Regional Container Lines	Thailand	51	39

KEY: TEUs = twenty-foot equivalent units. One 20-foot container equals one TEU, and one 40-foot container equals two TEUs.

Table 10: US Seaborne Regional Trade by Volume

TABLE I: U.S. SEABORNE TRADE BY WORLD REGION 2009 - 2008 Metric Tons, 000s									
REGION/COUNTRY	EXPORTS			IMPORTS			TOTAL TRADE		
	2009	2008	Change	2009	2008	Change	2009	2008	Change
AFRICA	23,505	25,468	-7.7%	124,285	140,955	-11.8%	147,790	166,423	-11.2%
North Africa	12,801	15,066	-15.0%	34,240	36,813	-7.0%	47,041	51,879	-9.3%
Sub-Sahara Africa	10,704	10,403	2.9%	90,045	104,142	-13.5%	100,749	114,545	-12.0%
ASIA	189,198	172,477	9.7%	92,091	121,861	-24.4%	281,289	294,338	-4.4%
ASEAN	29,962	28,182	6.3%	18,937	21,525	-12.0%	48,899	49,707	-1.6%
China	65,610	46,827	40.1%	45,806	64,050	-28.5%	111,416	110,877	0.5%
Japan	40,505	47,273	-14.3%	7,923	12,299	-35.6%	48,428	59,572	-18.7%
South Korea	24,278	24,990	-2.9%	9,267	10,686	-13.3%	33,545	35,676	-6.0%
Taiwan	12,587	13,290	-5.3%	4,420	5,823	-24.1%	17,007	19,113	-11.0%
South Asia	15,920	11,683	36.3%	5,359	7,220	-25.8%	21,279	18,903	12.6%
AUSTRALASIA	4,231	4,766	-11.2%	4,817	6,577	-26.8%	9,048	11,343	-20.2%
Australia	3,420	3,971	-13.9%	3,789	5,247	-27.8%	7,209	9,218	-21.8%
CARIBBEAN	17,371	20,026	-13.3%	32,714	38,087	-14.1%	50,084	58,113	-13.8%
Jamaica	1,713	3,279	-47.7%	4,245	6,306	-32.7%	5,958	9,585	-37.8%
CENTRAL AMERICA	18,850	17,525	7.6%	7,655	9,041	-15.3%	26,505	26,566	-0.2%
Guatemala	4,352	3,945	10.3%	3,116	3,555	-12.3%	7,468	7,500	-0.4%
EUROPE	77,660	96,632	-19.6%	63,791	70,689	-9.8%	141,452	167,321	-15.5%
European Union	72,621	89,147	-18.5%	56,403	70,561	-20.1%	129,024	159,708	-19.2%
FORMER USSR	2,638	3,814	-30.8%	43,190	42,450	1.7%	45,828	46,264	-0.9%
Russia	1,592	2,157	-26.2%	35,132	31,494	11.6%	36,725	33,651	9.1%
NEAR EAST	23,543	28,399	-17.1%	94,605	125,342	-24.5%	118,148	153,741	-23.2%
Saudi Arabia	2,587	3,082	-16.1%	52,659	76,082	-30.8%	55,246	79,164	-30.2%
NORTH AMERICA	46,534	63,726	-27.0%	125,237	147,162	-14.9%	171,771	210,888	-18.5%
Canada	21,501	39,727	-45.9%	55,023	64,463	-14.6%	76,524	104,190	-26.6%
Mexico	25,033	23,999	4.3%	70,212	82,697	-15.1%	95,245	106,696	-10.7%
SOUTH AMERICA	47,366	52,306	-9.4%	161,330	181,985	-11.3%	208,696	234,291	-10.9%
Brazil	16,929	18,416	-8.1%	28,672	33,009	-13.1%	45,600	51,425	-11.3%
Venezuela	5,871	6,846	-14.2%	70,928	79,326	-10.6%	76,798	86,172	-10.9%
US GLOBAL TOTAL	451,822	485,822	-7.0%	749,716	891,894	-15.9%	1,201,538	1,377,716	-12.8%

Table 11: US Seaborne Regional Trade by Value

TABLE II: U.S. SEABORNE TRADE BY WORLD REGION 2009-2008 Millions of Current U.S. Dollars									
REGION/COUNTRY	EXPORTS			IMPORTS			TOTAL TRADE		
	2009	2008	Change	2009	2008	Change	2009	2008	Change
AFRICA	$17,847	$21,792	-18.1%	$58,508	$103,402	-43.4%	$76,355	$125,194	-39.0%
North Africa	$7,024	$8,231	-14.7%	$14,973	$26,650	-43.8%	$21,996	$34,881	-36.9%
Sub-Sahara	$10,823	$13,561	-20.2%	$43,535	$76,752	-43.3%	$54,359	$90,313	-39.8%
ASIA	$134,757	$154,003	-12.5%	$389,262	$494,871	-21.3%	$524,019	$648,874	-19.2%
ASEAN	$29,614	$33,980	-12.8%	$50,454	$61,530	-18.0%	$80,068	$95,510	-16.2%
China	$44,865	$44,843	0.1%	$210,630	$250,784	-16.0%	$255,496	$295,626	-13.6%
Japan	$25,224	$33,407	-24.5%	$67,311	$102,873	-34.6%	$92,534	$136,280	-32.1%
South Korea	$15,892	$19,305	-17.7%	$23,222	$30,364	-23.5%	$39,114	$49,668	-21.2%
Taiwan	$8,336	$11,071	-24.7%	$15,392	$21,975	-30.0%	$23,727	$33,046	-28.2%
South Asia	$10,627	$11,133	-4.5%	$20,211	$24,355	-17.0%	$30,838	$35,488	-13.1%
AUSTRALASIA	$11,505	$14,225	-19.1%	$7,548	$11,000	-31.4%	$19,053	$25,225	-24.5%
Australia	$9,948	$12,554	-20.8%	$5,156	$7,981	-35.4%	$15,104	$20,535	-26.4%
CARIBBEAN	$14,658	$18,813	-22.1%	$11,077	$17,481	-36.6%	$25,734	$36,294	-29.1%
Trinidad & Tobago	$1,572	$1,769	-11.1%	$5,090	$8,934	-43.0%	$6,662	$10,702	-37.7%
CENTRAL AMERICA	$14,828	$18,728	-20.8%		$13,068	-100.0%	$14,828	$31,796	-53.4%
Honduras	$2,955	$4,293	-31.2%	$3,041	$3,776	-19.5%	$5,997	$8,069	-25.7%
EUROPE	$80,753	$114,636	-29.6%	$135,518	$186,045	-27.2%	$216,270	$300,681	-28.1%
European Union	$76,969	$108,954	-29.4%	$127,782	$185,794	-31.2%	$204,751	$294,748	-30.5%
FORMER USSR	$5,644	$10,200	-44.7%	$21,786	$34,248	-36.4%	$27,430	$44,448	-38.3%
Russia	$3,929	$7,125	-44.9%	$17,146	$24,451	-29.9%	$21,075	$31,576	-33.3%
MIDDLE EAST	$27,020	$37,626	-28.2%	$45,983	$94,089	-51.1%	$73,003	$131,715	-44.6%
Saudi Arabia	$7,754	$9,603	-19.3%	$21,787	$54,162	-59.8%	$29,540	$63,765	-53.7%
NORTH AMERICA	$15,734	$22,437	-29.9%	$43,516	$70,819	-38.6%	$59,250	$93,256	-36.5%
Canada	$4,293	$7,510	-42.8%	$15,252	$24,328	-37.3%	$19,545	$31,838	-38.6%
Mexico	$11,436	$14,921	-23.4%	$28,257	$46,485	-39.2%	$39,693	$61,406	-35.4%
SOUTH AMERICA	$43,777	$58,847	-25.6%	$71,193	$114,013	-37.6%	$114,969	$172,860	-33.5%
Brazil	$13,114	$17,598	-25.5%	$17,331	$25,983	-33.3%	$30,445	$43,582	-30.1%
Venezuela	$6,723	$9,765	-31.1%	$27,913	$51,096	-45.4%	$34,636	$60,861	-43.1%
US GLOBAL TOTAL	$367,520	$471,536	-22.1%	$795,279	$1,152,327	-31.0%	$1,162,799	$1,623,863	-28.4%

Appendix E: ORB Key

The most relevant oil record book entry categories that this project is concerned with is letters C,D, and E for "Machinery Spaces Operations". Below lists the details needed in these entries **FOR TANKERS AND NON-TANKERS ALIKE** as well as examples of these entries.

C. Collection and disposal of oil residues (Sludge and other oil residues)

11. Collection of oil residues
    Quantities of oil residues (sludge and other oil residues) retained on board. The quantity should be recorded weekly: (This means that the quantity must be recorded weekly if the voyage lasts longer than 1 week)

    .1 - Identity of tank(s)

    .2 - Capacity of tank(s) in m^3, gal., or bbl.

    .3 - Total quantity of retention in m^3, gal., or bbl.

12. Methods of disposal of residue
    State quantity of oil residues disposed of, the tank(s) emptied and the quantity of the contents retained in m^3, gal., or bbl.

    .1 - To reception facilities (identify port)

    .2 - Transferred to another (other) tank(s) (Indicate tank(s) and the total content of tank(s));

    .3 - Incinerated (indicate total time of operation);

    .4 - Other method (state which).

D. Non-automatic discharge overboard or disposal otherwise of bilge water which has accumulated in machinery spaces

13. Quantity discharged or disposed of (in m^3, gal., or bbl.)

14. Time of discharge or disposal (starts and stop)

15. Method of discharge or disposal:

    .1 - Through 15 PPM equipment (state position at start and end)

    .2 - To reception facilities (identify port);

    .3 - Transfer to slop tank or holding tank (indicate tank(s); state the total quantity retained in tank(s) in m^3, gal., or bbl.).

E. Automatic discharge overboard or disposal otherwise of bilge water which has accumulated in machinery spaces

16. Time and position of ship at which the system has been put into automatic mode of operation for discharge overboard, through 15 PPM equipment.
17. Time when the system has been put into automatic mode of operation for transfer of bilge water to holding tank (identify tank).
18. Time when the system has been put into manual operation.

An example of a CODE D entry is seen in Figure 53 below:

Name of Ship M/V NOT AN OIL TANKER
Official Number 413567

CARGO/BALLAST OPERATIONS (Oil Tanker) / MACHINERY SPACE OPERATIONS

Date	Code	Item	Record of Operations/signature of officers in charge
			EXAMPLE: BILGE WATER DISPOSAL # 1
05/06/2006	D	13	14 m³ bilge water
	D	14	Start 1000 - Stop 1200
	D	15.3	To collecting tank, 14 m³ transferred,
			30 m³ retained in tank.
			Z.L. Hughes
			EXAMPLE: BILGE WATER DISPOSAL # 2
05/06/2006	D	13	14 m³ bilge water
	D	14	Start 0000 - Stop 0300
	D	15.1	50°00' N x 29°58'W - Start
			49°56' N x 30°00'W - Stop
			K. Brennan

Figure 53: Coded ORB Entry

Appendix F: OWS Inspection Protocol

Task 7.0 Conduct Machinery Examination

Step		Action	Ref
7.16	O	Examine oil and HAZMAT. • Fuel oil and bulk lubricating oil discharge containment • Prohibited oil spaces	33CFR 155 320 33CFR 155.470
7.17	O	Examine oily water separating equipment, bilge alarm, and bilge monitor.	MARPOL 73/78 Annex 1/16 33CFR 155.380 PCV Policy Letter 01-06

If oily water separator built to . . .	Then continue with step . . .
MEPC.107(49)	7.17a
MEPC.60(33) or earlier standard	7.17b

7.17a Oily Water Separator (built to MEPC.107(49))

- Conduct review of 15 parts per million (ppm) bilge monitoring/alarm records
- Verify oily water monitoring/bilge alarm equipment designed to store data for up to 18 months & able to display or print a protocol
- Verify recorded items: date, time, alarm status, and operating status of the 15 ppm separator
- Compare above entries against existing ORB entries for nonconformities
- Verify 15-ppm monitor/bilge alarm sealed
- Verify 15-ppm oily water monitors or bilge alarms have been calibrated. (To be completed only by an authorized equipment testing company)
- Verify valid IOPP certificate accompanied by the manufacturer's calibration certificate as proof (Manufacturer's calibration certificates cannot be older than five years)
- No further testing is needed unless tampering or malfunctioning is suspected. The entire alarm unit may be replaced by a calibrated 15 ppm alarm. A bilge alarm should not be accepted as compliant if it is over five years old unless it has been calibrated as discussed above

52

Figure 54: Ship inspection protocols for Systems Using MEPC.107(49)

Task 7.0 Conduct Machinery Examination

Step	Action	Ref
7.17b	Oily Water Separator (built to MEPC.60(33) or earlier standard)	MARPOL 73/78 Annex 1/16
	If the OWS is approved in accordance with Resolution MEPC.60(33) or earlier standard, the following examination guidance is recommended as a supplement to the guidance contained in NVIC 8 83 and NVIC 6 94, Change 1.	33CFR 155.380 PCV Policy Letter01-06

- Identify crewmembers responsible for the operation of the OWS based on the Safety Management System or by asking the Chief Engineer
- During the operational test, observe and determine their competency with the equipment and associated piping
- Consult the manufacturer's operations manual for operating the OWS and OCM and follow any relevant procedures provided
- Witness operational test for at least 15 20 minutes
- Verify fluid entering the OWS for processing comes directly from the bilge holding tank or rose box and is not diluted by open sea or fresh water connections
- Verify no dilution of the processed oily water sample line to the OCM. The OCM outlet fluid should be visible
- If the vessel uses a source tank to supply oily water to the OWS, verify the source tank level drops proportionately in comparison to the capacity of the OWS for the period of time the equipment was run
- Verify the OWS effluent is visibly clean
- Verify that reasonable quantities of consumable filter elements, coalescing media, recording paper, etc., if applicable
- Verify that OWS manufacturer's recommended spare parts onboard
- Examine OWS for signs of unapproved modifications bypasses, etc.

53

Figure 55: Ship inspection protocols for Systems Using MEPC.60(33)

Appendix G: Ship Tour Summary

14 October 2011

STAVRODOMI (Tanker), East Providence

(Ship Info: http://www.marinetraffic.com/ais/shipdetails.aspx?mmsi=256068000)

On this trip we learned a lot about how waste oil is dealt with on the ships in addition to learning more about the inspection side of things and really how ship inspectors are able to find non-compliance. Before we left the Coast Guard office, they talked us through some preliminary information and gave us some background information. For example, we were told that every foreign-flagged ship doing business with the US is usually checked once a year at the least, and the inspection takes about 4-5 hours. A US merchant marine shipping vessel is required to undergo a 3-day-long in-depth inspection once per year. We went through some documents, a couple of which they allowed us to keep (including a sample oil record book).

As soon as we got onto the ship, the pollution prevention measures became clearly evident, as can be seen in Figure 56 below.

Figure 56: "Avoid Pollution" in Writing in a Public Place on the Ship

When we boarded the ship, we immediately went up to meet the captain of the ship, introduce ourselves and state our business. The USCG inspectors that brought us onto the ship did all the talking and introduced us to the captain. They instructed us that one should treat a foreign vessel as though it is a foreign country. It was in the captain's office where we were introduced to the chief engineer of the ship, who led the remainder of the tour.

G1. Oil-Water Separator

The first stop of the tour was down at the oil-water separator (see Figure 57). We were told that there are two major categories of OWS. There are the older models and the newer models. The newer models have a display as well as an 18-month memory card. However, the vessel we were on has one of the older systems. The USCG inspectors with us were pretty positive that a SWOMS can only work with the newer OWS.

Figure 57: The Oil Water Separator

The chief engineer explained to us how the system works. The system needs to be manually turned on and run. Oily bilge water will run through the system, and the silver disk above the box on the right (pictured in Figure 58) is the sensor to test the concentration of the bilge water. If it is over the desired maximum concentration of 15 ppm then the alarm sounds and the outflow valve shuts.

Figure 58: The Valve and Pipe Leaving the OWS.

Pictured in Figure 58 above is the outflow valve that closes. Only water with less than 15 ppm oil is able to go into the white pipe. The USCG ship inspectors said that if the connection between that white and green pipe look like it has been opened a lot, that is usually a red flag that there has been tampering. In the event that it looks like the valve has been opened frequently, the USCG will open it up and see if there is oil inside the white pipe. If there is oil in the white pipe, then the OWS does not work or the sensor was tampered with. The chief engineer said that more recent OWS have a seal on this junction and it is evident when it is tampered with. The message pictured in Figure 59 was painted on the wall next to the OWS.

Figure 59: Example on Board Instruction for Oil Waste Safety

G2. Incinerator

Our next stop after the OWS was the room that contains the incinerators and the sludge tank. The incinerator on this ship had not been used in 10 months. The open part pictured in Figure 60 below is where the oil would normally be burned. There was also another compartment higher up for solid waste.

Figure 60: Incinerator

G3. Oil Record Books

Next, we went to the office of the chief engineer to see examples of a real oil record book (see Figure 61). The first oil record book we looked at was blank.

Figure 61: Oil Record Book Sample

However, there was also a completed oil record book on board. According to regulations, an oil record book dating back three years must be stored on the ship at all times. We were able to find an oil pumping entry in which the ship had waste oil pumped off the ship (see Figure 62).

Figure 62: Example ORB entry

The entry in Figure 62 was for the oil pumped off the ship. Since it was not done via the OWS, there needs to be a receipt from a recognized pumping company to prove that it was pumped off the ship by an acceptable means. We looked at the entry in the ORB and then the receipt, shown in Figure 63, to verify the amounts and the duration of the pumping.

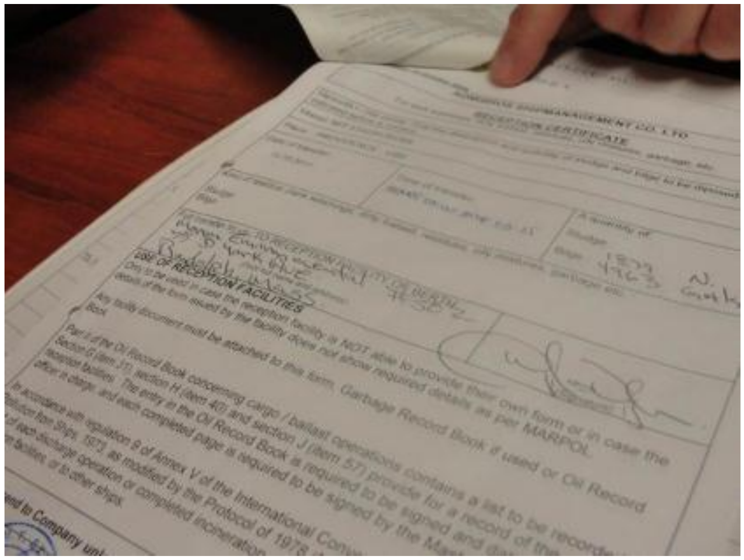


Figure 63: Receipt for pumping.

The receipt matched up with the amounts in the ORB and the duration of pumping was approximately the same. All of the compliance of the ships during these inspections is checked on the spot by hand.

Figure 64: The hookups to pump sludge off of the vessel.

When the waste is pumped off, it is done via the hookups pictured in Figure 64 above. All ships have the same hookups so that it is easy for waste removal companies to be able to pump from any shipping vessel.

G4. Non-Compliance

Most non-compliant activity on shipping vessels is isolated to individuals. An example story we were told was about an inspection that USCG inspectors had done in which they asked the crew to run the OWS but no one on board knew how. The ORB had indicated that the OWS had been run one week prior to the inspection. When the USCG inspectors asked repeatedly for someone to run it and no one knew how, it became evident that something was wrong. After 6 hours of working the situation, the crew member instructed to run the OWS a week prior admitted that he had actually removed oil from the storage tank and put it into the sludge tank because he was unaware of how to run the OWS and was ashamed to ask for help.

On many ship inspections, the crewmembers are instructed to run the systems if it is indicated in the ORB that these systems were used. If the crew is unable to run the systems, it is usually a red flag for non-compliant activity. If the numbers on receipts do not match up with the ORB amounts, then that is another red flag. A major factor in catching non-compliant ships is the interaction between the crew and the inspectors. Another form of non-compliance that we were informed of is forged documents. Apparently years back, in some areas of Africa, forgers would sell fake receipts for “pumped oil”.

G5. Supplemental Materials

The USCG inspectors gave us the inspection checklist for OWS, a United States blank oil record book, as well as an OWS inspection training PowerPoint. They gave us their contact information so that we are able to contact them with any questions during the course of this project. They also informed us about one of their older inspectors who now works at the Marine Safety Center at USCG Headquarters in DC.

Appendix H: SWOMS Parser Program

In order to process SWOMS data delivered via email, a multi-step system was implemented that utilizes Microsoft Outlook rules, VBA (Visual Basic for Applications) scripting, and a program written specifically for parsing the data contained in the SWOMS emails. The first step in the data acquisition process is the transmission of the SWOMS email from a vessel to the Coast Guard.

Once sent from the vessel, a SWOMS email is transmitted by way of the internet to the Coast Guard's network. As it enters, the email is scanned for malicious code and, if clean, is allowed through the network gateway. From there, the email is routed to a Coast Guard Exchange (email) server and arrives at the inbox of the specified address. In this case, the addressee is a special mailbox monitored using Outlook running on a computer system. Once the email arrives in the inbox, Outlook starts processing the new email.

Outlook has a system to process rules, which can be simple or complex routines to organize emails and do basic processing. Upon first arrival, an Outlook rule strips the message of any embedded category data, which is a property of emails often used for sorting and categorizing. A second rule is run that has a much more complex set of tasks. Rule 2 is set to only run on emails that do not have the "Processed" category set. This allows the rule to be run later, in the event that there is a failure and data must be re-acquired. This rule initiates the SWOMS parsing process by moving the email to a separate folder, marking it as read, and setting the category to "Processed" in order to indicate that the email does not need to be processed again. The last command in the rule starts a VBA script, where major processing is started.

The VBA script prepares the emails for processing by the SWOMS parsing program described further in this section. The email is first converted to the plaintext format to avoid any formatting tags in the data that could cause conflicts when analyzing the text. Next, a new text file containing this text is written to a network folder. The VBA script then creates and runs a Windows batch file, concluding the VBA portion of processing. At this point, Outlook is no longer performing any processes on the email and it is freed to process the next email.

The batch file generated by the VBA script is processed using the Windows command interpreter. First, it executes the custom parsing program, "swomsparser", and provides it with the plaintext version of the email that was written to disk by the VBA script. The batch file then waits for the swomsparser program to finish processing the data. Once the data are processed, the batch file removes the temporary text file and removes itself, leaving behind no intermediate files.

The swomsparser program uses the plain-text output of the VBA script that is fed into it from the batch file. These emails are formatted as shown below in Figure 65. In its current form, these data are fairly easy for humans to read, and it is well-labeled.

```
VESSEL: vessel1
Daily Enviro-Logger Report
21/09/2010 00:00:00 GMT dd/mm/yyyy
Running Software Version 8.12FS
GPS Latitude: 40 55.972' N
GPS Longitude: 18 6.714' E

Current BHT Lvl 0.39m  Vol 2.45m3
Current Sludge Tank Lvl 0.09m  Vol 1.90m3
Current Oily Bilge Tank Lvl 0.37m  Vol 1.12m3
Current Stbd Bilge Lvl 0.38m  Vol 1.32m3
Current Port Bilge Lvl 0.06m  Vol 0.12m3
Current Aft Bilge Lvl 0.59m  Vol 4.05m3
Current W.O. Service Tank Lvl 0.84m  Vol 1.13m3

INFORMATION FOR LAST 24 HOURS

Blg Pmp Run 0 Times
Blg Pmp Run 0.00 Minutes
OWS Run 0 Times
OWS Run 0.00 Minutes
Overboard Request Open 0 Times
Overboard Requested Open 0.00 Minutes
Overboard Open 0 Times
Overboard Open 0.00 Minutes
PPM Alarm 0 Times
Average PPM Overboard 0.0
OCM Fresh Water Valve Opened 0 Times
Oil Purge Valve Open 0 Times
Oil Purge Valve Open 0.00 Minutes

Incinerator Run 2 Times
Incinerator Burning W.O. 900.17 Minutes
```

Figure 65: Example text from SWOMS Email

It is also notable that there are definite beginnings and endings to different numeric values found in the data. For example, tank levels are enveloped by “Lvl ” and ‘m’, and tank volumes are between the delimiters “Vol ”and “m3”. Furthermore, other information may be found between a label and a line break (e.g. longitude, average PPM overboard) or between a label and a “space” character (e.g. “Blg Pmp Run ”, “Incinerator Burning W.O. ”). This allows the program to identify and parse different pieces of data that are included in the database.

The final program developed has been designed to make future changes fairly simple from a software development point of view, although more could have been done to make the system more adaptable. The largest constraint on the parsing program is the rigidity of its data structure. It expects to find data with specific labels and delimiters that are hard-coded into the program. This means that after a change to the code, the "swomsparser" program must be fully recompiled before it is used. Before describing the parsing elements of the program, it is important to note how data are eventually stored and used.

The program, "swomsparser", will create a .CSV file for each new vessel name that is encountered. It will also interact with another table (currently hard-coded as "SWOMS_TABLE.csv") that will be synchronized with the database automatically. Figure 66 below shows how the program behaves when writing a piece of data to the tables. Note that it will always write data to the vessel table, but will only write data to the SWOMS_DATA table if it has a non-default IMO number.

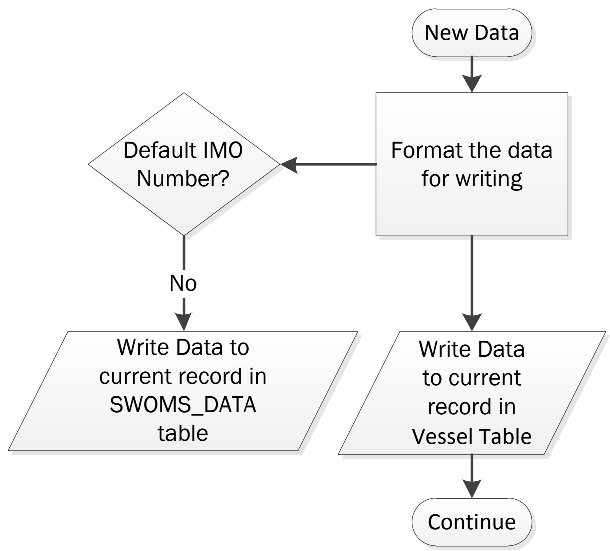


Figure 66: Writing Data to Output Files

The first line of each vessel's individual table is reserved to hold the ship's IMO number and the ship's name. The major difference between the two tables is that instead of an IMO number and vessel name, the SWOMS table has a header including the labels that match up with the table in the database. This allows the database to identify which data columns in the SWOMS table correspond to the columns found within the database, and so allows this table to be directly imported into the database. Every line in the body of the vessel table and in the body of the SWOMS table follows the data format necessary to seamlessly interface with the database.

The program flowchart can be seen later on in this section in Figure 69. At its startup, swomsparser will first verify that it has been given at least one command line parameter. The program is designed to ignore any parameters beyond the first so the program will give no errors or modified output if it is given more than one parameter. If no parameter is given, the program will briefly print an error message to the screen and then exit. Next, the program ensures that the input file is valid and begins processing the input data. The program identifies the vessel name and looks for an existing table that corresponds to the vessel's data. If no table is found, it generates one, inserting a dummy "0000000" IMO number along with the vessel name on the first line of the file. If a matching vessel-specific table is found, the program will use the IMO number found on the first line of the vessel-specific table.

Once the vessel-specific output file is established, the program also opens the SWOMS table file for adding the data. If there is no IMO number found, the program will only write records to the vessel-specific table. Otherwise, each file-writing operation is performed twice: once for the vessel table and once for the SWOMS table. Each email becomes a single record on each table. Next, the program searches for the email log date.

The Ionia SWOMS logs the date in the international DD/MM/YYYY format, but the database needs to use the United States preferred MM/DD/YYYY. First, the search algorithm seeks out lines that contain a '/'. If a '/' is found, the program will check the beginning of the line for the sequence "##/##/####". If it doesn't find this string, it will keep searching line by line using the same method until it hits the end of the input file. When the date is found, it is read into a variable in the US format and then written into the output file(s). Time is recorded on the same line, and is read "blindly" by the program. The program expects the date and time line to be of the format "DD/MM/YYYY hh:mm:ss" and does not verify the time data, but just accepts that the last eight characters on the line are the time.

After date and time are written to the output file(s), the program reads and writes latitude, longitude, and 24-hour data one at a time. To find these, a function within the program named "getData" is called that finds and returns the data to a buffer string. This function writes a string of data found between two specified delimiters to a buffer string, and returns a position value pointing to the start of the line where the data string was found. In order to use getData, the calling function needs to give it the following parameters:

- A pointer to a buffer string for output
- The start delimiter that comes immediately before the relevant data string
- The maximum length of the string the function may read in
- A pointer to the file to search through (this pointer refers to the file-name parameter that is called when invoking the program)
- The end delimiter that comes immediately after the data string
- A Boolean "continuous-search" flag

The function getData initially defaults to give "0" as its output, in case it cannot find the specified string. It then checks for the continuous flag. If it is set to non-continuous, the program will start the search from the top of the input file. Otherwise, it will continue the search from the line following where the last search left off. This is useful in the event that there are two identical start delimiters (e.g. “Blg Pmp Run x Times” and “Blg Pmp Run x.xx Minutes”). The program proceeds to check each line for the target starting delimiter. Once this search term is found, the program will then read between the start delimiter and the end delimiter. If the end of the line is reached before the end delimiter, it will use the end of line instead. If there is a problem with the end delimiter passed to the function or if the string would exceed the maximum length specified, then the function will give an error. The buffer string will remain set to “0” and getData will return the file position of the beginning of that line to the calling function. Otherwise, the temporary “0” buffer spring will be overwritten by the data string, which will then be written to the output file, and the file position of the beginning of that line will be returned.

The swomsparser program will process all of the 24-hour data using the getData function. The program uses a loop to read data and then write it, one item at a time. This process is shown in the code shown in Figure 67 below.

```
i=0;
while(1)
{
// Acquire data from input file
switch(i)
{
case  0: getData(tmpStr, "Latitude: ", 30, inFile, D_NEWLINE, 0); break;
case  1: getData(tmpStr, "Longitude: ", 30, inFile, D_NEWLINE, 0); break;
case  2: getData(tmpStr, "Blg Pmp Run ", 30, inFile, D_SPACE, 0); break;
case  3: getData(tmpStr, "Blg Pmp Run ", 30, inFile, D_SPACE, 1); break;
case  4: getData(tmpStr, "OWS Run ", 30, inFile, D_SPACE, 0); break;
case  5: getData(tmpStr, "OWS Run ", 30, inFile, D_SPACE, 1); break;
case  6: getData(tmpStr, "Overboard Open ", 30, inFile, D_SPACE, 0); break;
case  7: getData(tmpStr, "Overboard Open ", 30, inFile, D_SPACE, 1); break;
case  8: getData(tmpStr, "PPM Alarm ", 30, inFile, D_SPACE, 0); break;
case  9: getData(tmpStr, "Average PPM Overboard ", 30, inFile, D_SPACE, 0); break;
case 10: getData(tmpStr, "OCM Fresh Water Valve Opened ", 30, inFile, D_SPACE, 0); break;
case 11: getData(tmpStr, "Oil Purge Valve Open ", 30, inFile, D_SPACE, 0); break;
case 12: getData(tmpStr, "Oil Purge Valve Open ", 30, inFile, D_SPACE, 1); break;
case 13: getData(tmpStr, "Incinerator Run ", 30, inFile, D_SPACE, 0); break;
case 14: getData(tmpStr, "Incinerator Burning W.O. ", 30, inFile, D_SPACE, 0); break;
default: goto tank_data;
}
```

Figure 67: Loop to Parse 24-Hour Data

As can be seen in the code shown above, it uses a series of "case" entries to keep track of which piece of data should read/write next. The last "case" is the default case, where it ends the loop and then begins getting tank data.

The tank data are organized in such a way that multiple pieces of information are contained in a single line. The getData function automatically advances to the line following the position value that is returned. This return value is useful when multiple searches must occur on one line because it allows the program to return to the beginning of the line instead of going to the next. An example of this is seen when getting tank level and volume data from the same line as shown below in Figure 68.

```
pos = getData(tmpStr, "Lvl ", 30, inFile, D_M, 0);
fprintf(outFile, "%s,", tmpStr);
if(strcmp(imoNumber,"0000000") != 0) fprintf(bigOutFile, "%s,", tmpStr);
fsetpos(inFile, &pos);
getData(tmpStr, "Vol ", 30, inFile, D_M, 1);
fprintf(outFile, "%s,", tmpStr);
if(strcmp(imoNumber,"0000000") != 0) fprintf(bigOutFile, "%s,", tmpStr);
```

Figure 68: Multiple getData Searches Within a Single Line

The code snippet above sets the return value of the previous getData call (located at the beginning of the line where the string was found) as the position where the current getData call will begin to search. This means that when getData is called, it will search the line again, rather than skipping the additional data contained at the end of the line.

The next stage is getting the tank data. This process is divided into three "phases". Phase 1 gets the first tank's data, phase 2 gets data for tanks 2-15, and phase 3 gets data for the last possible tank, tank 16. The data itself is retrieved in the same manner in each phase, but the first and last tanks have special requirements. The first tank's level is searched for starting at the top of the file, so the getData function is called with a "non-continuous" flag. Phase 2 is a simple loop that performs continuous getData function calls throughout. Phase 3 differs from the individual loops of phase 2 in that its output is concluded with a newline character. Phase 3 "closes" the current record, and prepares the output file(s) for a new record entry.

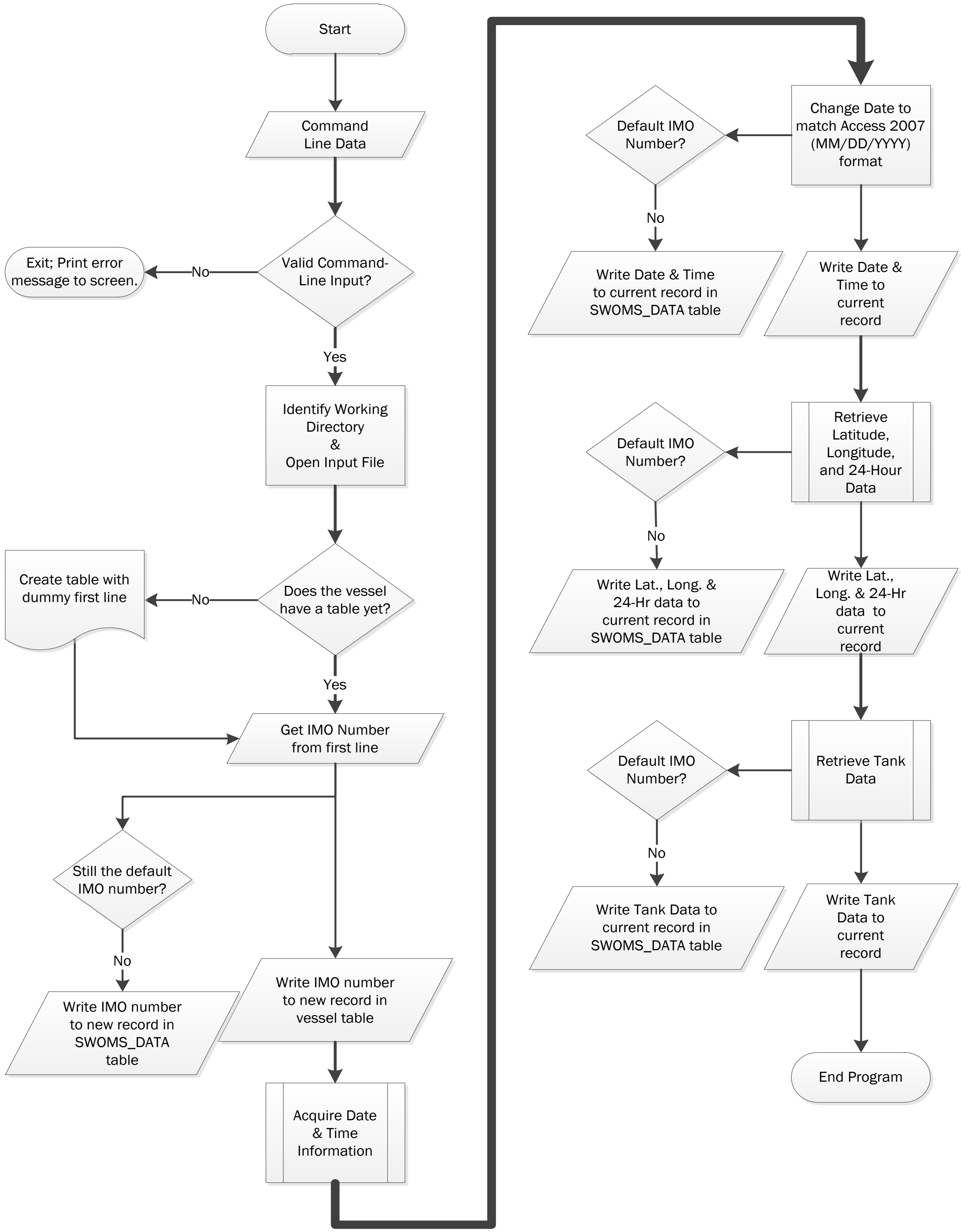


Figure 69: SWOMS Parser Program Flowchart

Below is the code, written in C, for the swomsparser program. Consult the user manual for instructions for building and rebuilding the program from this code.

```
/*****************
Patrick Brodeur w/ WPI IQP team:
    ¤ Reneé Lanza     ¤ Lisa Morris      ¤ Eddie Osowski
Single-file txt-to-csv parser
Purpose-built to read SWOMS data from email dumps into a CSV.
Written during November/December 2011
*****************/

/*******************
    Include Files
*******************/
#include <windows.h>    // Windows includes; supersedes the others.
#include <stdio.h>
#include <string.h>
#include <time.h>

/***************
    Constants
***************/
#define L_DATETIME 11
#define D_NEWLINE 1
#define D_SPACE  2
#define D_M 3
#define CTS 1
#define NOT_CTS 0
#define BIGTABLE "SWOMS_DATA" // Change this to set the name of the table for syncing with DB

/*************************
    Function Prototypes
*************************/
fpos_t getData( char* outputStr, const char* sLabel, unsigned int maxLength, FILE* searchFile, unsigned int nDelimiter, unsigned char cts);
void getData2( char* outputStr, const char* sLabel, FILE* searchFile, char delimiter);
void textToFileName( char* outputStr );

/*******************
    MAIN Function
*******************/
int main(int argc, char *argv[])    // Setup for command-line input
{
    // Make sure there's an argument or else Windows (or  *nix) will cry.
    if(argv[1] == NULL)
    {
        printf("Critical error: Must have input file on command line.\n");
        return 0;
    }

    /** Test for valid input **/
    if(fopen(argv[1], "r") == NULL)
    {
        printf("Must provide a valid input file\n");
        return 0;
    }

    /*********************
        Local Variables
    *********************/
    FILE *inFile, *outFile, *bigOutFile; // Pointers to the input file and output files
    FILE *SWOMSLOCK, *VTABLELOCK; // Pointers to the .LOCK files

    char tmpStr[100]; // Buffer string
    char outPath[175], outLock[180]; // Output directory path
    char imoNumber[] = "0000000";
    char* strPtr; // General pointer for string ops
```

```c
    int i,j; // Iterators
    char date[L_DATETIME];
    char time[L_DATETIME];
    clock_t timeOut;

    /*****************
        Processing
    *****************/

    /** Open the input file **/
    strcpy(outPath, argv[1]); // Move into local variables
//printf("DEBUG: Input File is %s\n",outPath);
    inFile = fopen(outPath, "r"); // Open input file for reading
//printf("DEBUG: inPath is %s\n", outPath);


    /** Find working directory and establish output file **/
    getData(tmpStr, "VESSEL: ", 20, inFile, D_NEWLINE, 0);   // Retrieve vessel's name from the input file
    textToFileName(tmpStr); // Remove prohibited characters
    strPtr = strrchr(outPath,'\\'); // Extract the path

    if(strPtr == NULL) strcpy(outPath, ".\\"); // If no path given, use working directory
    else outPath[strlen(outPath)-strlen(strPtr)+1] = 0; // Keep trailing slash
    strcat(outPath, tmpStr); // Add vessel name as the filename
    strcpy(outLock, outPath);
    strcat(outPath, ".csv"); // Tack on CSV extension so excel, etc. will recognize it
//printf("DEBUG: outPath is %s\n", outPath);

    /** Write first line if the file doesn't exist **/
    if( fopen(outPath, "r") == NULL)
    {
        outFile = fopen(outPath, "w");  // Create and open output file for writing
        getData(tmpStr, "VESSEL: ", 20, inFile, D_NEWLINE, 0); // Retrieve vessel name again
        fprintf(outFile, "0000000,%s\n", tmpStr); // Put the first line in (IMO Number, Shipname)
        fclose(outFile);
    }

/** Lock system: Lock down the vessel table **/
    j=0;
    printf("LOCKING: \n%s\n", outLock);
    strcat(outLock, ".LOCK");
    VTABLELOCK = fopen(outLock,"r");
    while(VTABLELOCK != NULL)
    {
        fclose(VTABLELOCK);
        j++;
        if(j>90)
        {
            printf("\nSWOMSPARSER FAILED. FILE:\n%s\n",argv[1]);
            bigOutFile = fopen("ErrorLog.txt", "a");
            fprintf(bigOutFile,"Failed to lock Vessel table. Lock file: %s.\n", outLock);
            return 0;
        }
        if(j>=60) remove(outLock);
        printf("Failed to lock (%d). Retrying", j);
        if(j==30) printf("Warning: Attempting Lock Removal on Retry #60\n");
        timeOut = clock() + CLOCKS_PER_SEC;
        i=0;
        while (clock() < timeOut)
        {
            if (clock() >= timeOut - ((10-i) * CLOCKS_PER_SEC / 10))
            {
                putchar('.');
                i++;
            }
        }
        putchar('\n');
```

```c
        VTABLELOCK = fopen(outLock,"r");
    }

    VTABLELOCK = fopen(outLock,"w");
    fputs("LOCKED FOR EDITING",VTABLELOCK);
    fclose(VTABLELOCK);
    printf("LOCKED: \n%s\n\n", outLock);


    /** Retrieve the IMO number from line 1 **/
    outFile = fopen(outPath, "r");                                  // Open for reading
    fgets(tmpStr, 100, outFile);                                    // Read in first line

    // Copy up until the first comma
    i=0; imoNumber[0]=tmpStr[0];
    while (imoNumber[i] != ',')
    {
        i++;
        imoNumber[i] = tmpStr[i];
    }
    imoNumber[i] = 0;
//printf("DEBUG: IMO NUMBER is %s\n", imoNumber);
    fclose(outFile);


    /** Lock system: Lock down the SWOMSTABLE table **/
    j=0;
    printf("LOCKING: \n%s\n", BIGTABLE".LOCK");
    SWOMSLOCK = fopen(BIGTABLE".LOCK","r");
    while(SWOMSLOCK != NULL)
    {
        fclose(SWOMSLOCK);
        j++;
        if(j>90)
        {
            printf("\nSWOMSPARSER FAILED. FILE:\n%s\n",argv[1]);
            bigOutFile = fopen("ErrorLog.txt", "a");
            fprintf(bigOutFile,"Failed to lock SWOMS table. IMO: %s.\n", imoNumber);
            return 0;
        }
        if(j>=60) remove(BIGTABLE".csv");
        printf("Failed to lock (%d). Retrying\n", j);
        if(j==30) printf("Warning: Attempting Lock Removal on Retry #60\n");
        timeOut = clock() + CLOCKS_PER_SEC;
        i=0;
        while (clock() < timeOut)
        {
            if (clock() == timeOut - ((10-i) * CLOCKS_PER_SEC / 10))
            {
                putchar('.');
                i++;
            }
        }
        putchar('\n');
        SWOMSLOCK = fopen(BIGTABLE".LOCK","r");
    }
    SWOMSLOCK = fopen(BIGTABLE".LOCK","w");
    fputs("LOCKED FOR EDITING",SWOMSLOCK);
    fclose(SWOMSLOCK);
    printf("LOCKED: \n%s\n\n", BIGTABLE".LOCK");


    /** Prepare individual and cumulative tables for appending data **/
//printf("DEBUG: Output File -- %s\n", outPath);
    outFile = fopen(outPath, "a");
    if(strcmp(imoNumber,"0000000") != 0) bigOutFile = fopen(BIGTABLE".csv", "a+");
```

```
    /** Add IMO Number **/
    fprintf(outFile, "%s,", imoNumber);
    if(strcmp(imoNumber,"0000000") != 0) fprintf(bigOutFile, "%s,",imoNumber);


    /** Add Date and Time**/
    // Date... the hard part.
    date[0]=0;
    while ( date[0] != tmpStr[3] )
    {
        // See if we've hit EOF yet...
        if( fgets(tmpStr, 100, inFile) == NULL )
        {
            printf("Error: Could not find date\n");
            return 0;                                  // Returns function call, breaking the while loop and skipping the rest.
        }
        strPtr = strstr(tmpStr, "/");                  // Find a forward slash.

        // See if the line goes like this: /##/####
        if( strPtr != NULL && strPtr[0] == '/' &&
            ((strPtr[1] >= 48) && (strPtr[1] <= 57)) && ((strPtr[2] >= 48) && (strPtr[2] <= 57)) && (strPtr[3] == '/') &&
            ((strPtr[4] >= 48) && (strPtr[4] <= 57)) && ((strPtr[5] >= 48) && (strPtr[5] <= 57)) &&
            ((strPtr[6] >= 48) && (strPtr[6] <= 57)) && ((strPtr[7] >= 48) && (strPtr[7] <= 57)) )
        {
        // If it looks like a date, format it in MM/DD/YYYY
            date[0]=tmpStr[3]; date[1]=tmpStr[4]; date[3]=tmpStr[0]; date[4]=tmpStr[1];
            date[6]=tmpStr[6]; date[7]=tmpStr[7]; date[8]=tmpStr[8]; date[9]=tmpStr[9];
            date[2]='/'; date[5]='/';
            date[10]=0;                                // nul-terminate date string
//printf("DEBUG: Date is %s\n",date);
        }
    }
    // Write date to files
    fprintf(outFile, "%s,",date);
    if(strcmp(imoNumber,"0000000") != 0) fprintf(bigOutFile, "%s,",date);

    // Time, the part that comes right after the date.
    for(i=0; i<8; i++)
    {
        time[i]=tmpStr[11+i];
    }
    time[8] = 0;
//printf("DEBUG: Time is %s\n", time);

    // Write time to files
    fprintf(outFile, "%s,",time);
    if(strcmp(imoNumber,"0000000") != 0) fprintf(bigOutFile, "%s,",time);



    /** Get longitude, latitude, and all 24-hour data **/
    i=0;
    while(1)
    {
        // Acquire data from input file
        switch(i)
        {
        case  0: getData(tmpStr, "Latitude: ", 30, inFile, D_NEWLINE, 0); break;
        case  1: getData(tmpStr, "Longitude: ", 30, inFile, D_NEWLINE, 0); break;
        case  2: getData(tmpStr, "Blg Pmp Run ", 30, inFile, D_SPACE, 0); break;
        case  3: getData(tmpStr, "Blg Pmp Run ", 30, inFile, D_SPACE, 1); break;
        case  4: getData(tmpStr, "OWS Run ", 30, inFile, D_SPACE, 0); break;
        case  5: getData(tmpStr, "OWS Run ", 30, inFile, D_SPACE, 1); break;
        case  6: getData(tmpStr, "Overboard Open ", 30, inFile, D_SPACE, 0); break;
        case  7: getData(tmpStr, "Overboard Open ", 30, inFile, D_SPACE, 1); break;
```

```
        case  8: getData(tmpStr, "PPM Alarm ", 30, inFile, D_SPACE, 0); break;
        case  9: getData(tmpStr, "Average PPM Overboard ", 30, inFile, D_SPACE, 0); break;
        case 10: getData(tmpStr, "OCM Fresh Water Valve Opened ", 30, inFile, D_SPACE, 0); break;
        case 11: getData(tmpStr, "Oil Purge Valve Open ", 30, inFile, D_SPACE, 0); break;
        case 12: getData(tmpStr, "Oil Purge Valve Open ", 30, inFile, D_SPACE, 1); break;
        case 13: getData(tmpStr, "Incinerator Run ", 30, inFile, D_SPACE, 0); break;
        case 14: getData(tmpStr, "Incinerator Burning W.O. ", 30, inFile, D_SPACE, 0); break;
        default: goto tank_data;
        }

        // Put the data into the output files
        fprintf(outFile, "%s,", tmpStr);
        if(strcmp(imoNumber,"0000000") != 0) fprintf(bigOutFile, "%s,", tmpStr);    // Only add to bigOutFile if the IMO number is non-zeros
//printf("DEBUG: Case: %d\t Value: %s\n", i, tmpStr);
        i++;
    }

    tank_data:
    /** Get current tank data **/
    // Phase 1: Read in first tank
    fseek(inFile,0L,SEEK_SET);
    getData2(tmpStr,"Lvl ",inFile,'m');
    fprintf(outFile, "%s,", tmpStr);
    if(strcmp(imoNumber,"0000000") != 0) fprintf(bigOutFile, "%s,", tmpStr);
    getData2(tmpStr,"Vol ",inFile,'m');
    fprintf(outFile, "%s,", tmpStr);
    if(strcmp(imoNumber,"0000000") != 0) fprintf(bigOutFile, "%s,", tmpStr);

    // Phase 2: Loop through tanks 2-15
    for(i=2; i<=15; i++)
    {
        getData2(tmpStr, "Lvl ", inFile, 'm');
        fprintf(outFile, "%s,", tmpStr);
        if(strcmp(imoNumber,"0000000") != 0) fprintf(bigOutFile, "%s,", tmpStr);
        getData2(tmpStr, "Vol ", inFile, 'm');
        fprintf(outFile, "%s,", tmpStr);
        if(strcmp(imoNumber,"0000000") != 0) fprintf(bigOutFile, "%s,", tmpStr);
    }

    // Phase 3: Read in tank 16 and prepare for next record
    getData2(tmpStr, "Lvl ", inFile, 'm');
    fprintf(outFile, "%s,", tmpStr);
    if(strcmp(imoNumber,"0000000") != 0) fprintf(bigOutFile, "%s,", tmpStr);
    getData2(tmpStr, "Vol ", inFile, 'm');
    fprintf(outFile, "%s\n", tmpStr);
    if(strcmp(imoNumber,"0000000") != 0) fprintf(bigOutFile, "%s\n", tmpStr);

    fclose(outFile);
    if(strcmp(imoNumber,"0000000") != 0) fclose(bigOutFile);


    /** Lock system: Remove locks. **/
    if(strcmp(imoNumber,"0000000") != 0)
    {
        fclose(SWOMSLOCK);
        remove(BIGTABLE".LOCK");
    }
    fclose(VTABLELOCK);
    remove(outLock);

    return 0; // Successful exit.

}  // End main

/**************************
    Additional Functions
**************************/
```

```c
// getData: Extracts bounded data from the text file being parsed, with a return value pointing just above where it finds data
fpos_t getData( char* outputStr, const char* sLabel, unsigned int maxLength, FILE* searchFile, unsigned int nDelimiter, unsigned char cts)
{
    // Local Variables
    unsigned int i;
    char tmpStr[100];   // Avoid segfaults; allocate the memory
    char *strPtr;
    fpos_t foundposition;

    // Default to "0" in case the program fails to find the search
    outputStr[0]='0';
    outputStr[1]=0;

    // Processing
    // Reset file position to beginning
    if(!cts) fseek(searchFile, 0L, SEEK_SET);

    while ( 1 )
    {
        // Check if we hit EOF
        fgetpos(searchFile, &foundposition);
        if( fgets(tmpStr, 100, searchFile) == NULL )
            {
//printf("Error: Could not find text \"%s\"\n", sLabel);
                return foundposition; // Returns function call, breaking the while loop and skipping the rest.
//printf("Last Line Read: %s\n", tmpStr);
//printf("Output: %s\n", outputStr);
            }

        // Point to the section of tmpStr that has the data we care about
        strPtr = strstr(tmpStr, sLabel);

        // If found, stop searching.
        if (strPtr != NULL) break;  // Stop searching and continue on
    }

    // Update output string to contain the relevant data, up until the chosen delimiter
    for( i = 0; i < maxLength; i++ ) outputStr[i] = strPtr[i+strlen(sLabel)];
    switch (nDelimiter)
    {
    case D_NEWLINE:
        {
            i=0; while((outputStr[i] != '\n') && (outputStr[i] !=  '\r'))
            {
                i++; // Error if the i gets too high
                if(i==maxLength){ printf("Error: Bad Delimiter or Max Length\n\tNewline. Search Term: ---%s---\n", sLabel); return foundposition;}
            }
            outputStr[i]=0; break;
        }
    case D_SPACE:
        {
            i=0;
            while((outputStr[i] != ' ') && (outputStr[i] != '\n') && (outputStr[i] !=  '\r'))
            {
                i++; // Error if the i gets too high
                if(i==maxLength){ printf("Error: Bad Delimiter or Max Length\nSpace. Search Term: ---%s---\n", sLabel); return foundposition;}
            }
            outputStr[i]=0; break;
        }
    case D_M:
        {
            i=0;
            while((outputStr[i] != 'm') && (outputStr[i] != '\n') && (outputStr[i] !=  '\r'))
```

```c
            {
                i++; // Error if the i gets too high
                if(i==maxLength){ printf("Error: Bad Delimiter or Max Length\n'm'. Search Term: ---%s---\n", sLabel); return foundposition;}
            }
            outputStr[i]=0; break;
        }
    default:printf("Error: Bad Delimiter Value\n\tSearch Term: ---%s---\n", sLabel);
    }
//printf("Last Line Read: %s\n", tmpStr);
//printf("Output: %s\n", outputStr);
    return foundposition;
}

// textToFileName: Replaces any illegal characters with characters that are filename-friendly
void textToFileName( char* outputStr )
{
    int i; // Iterator
    //for( i=0; i<strlen(outputStr); i++ ) outputStr[i] = outputStr[i];
    for ( i=0; i < strlen(outputStr); i++ )
        {
        switch (outputStr[i])
            {
                // Remove all invalid characters
                case '\\': case '/': case '?': case '*': case '\"': case '<': case '>': case '|':
                    outputStr[i] = '-'; break;
                case ' ':
                    outputStr[i] = '_'; break;
                default: break;
            }
        }
    outputStr[i] = 0; // End string with nul character

    return;
} // End textToFilename

// getData: Extracts bounded data from the text file being parsed, with a return value pointing just above where it finds data
void getData2( char* outputStr, const char* sLabel, FILE* searchFile, char delimiter)
{
    // Local Variables
    char tmpStr[100];   // Avoid segfaults; allocate the memory
    unsigned int i=0, j=0;

    // Default to "0" in case the program fails to find the search
    outputStr[0]='0';
    outputStr[1]=0;

    // Processing
    j = strlen(sLabel);
    while ( 1 )
    {
        for (i=0; i <  j; i++) tmpStr[i] = tmpStr[i+1];
        tmpStr[j-1] = fgetc(searchFile);
        tmpStr[j] = 0;

//printf("%s", tmpStr);
//while(getchar() != '\n');
        if(strcmp(tmpStr, sLabel) == 0) break;
        if(tmpStr[j-1] == EOF)
        {
//printf("EOF Error\n");
            return;
        }
    }

    outputStr[0] = fgetc(searchFile);
```

```
    i=0; while ( (outputStr[i] != delimiter) && (outputStr[i] != '\n') && (outputStr[i] !=  '\r') && (outputStr[i] !=  EOF))
    {
        i++;
        outputStr[i] = fgetc(searchFile);
    }
    outputStr[i] = 0;

    i=0; while(outputStr[i] != 0)
    {
        switch(outputStr[i])
        {
        // as long as it's numeric or a decimal point, we don't care
        case '0': case '1': case '2': case '3': case '4': case '5':
        case '6': case '7': case '8': case '9': case '.': /*printf("%c\n",outputStr[i]); */break;

        // if it's not a number or decimal point, output "0"
        default: outputStr[0] = '0'; outputStr[1] = 0; /*printf("Set to Zero\n");*/ return;
        }
        i++;
    }

    return;
}

    // End Program Source.
```

Appendix I: Access Database Code

What follows is the VBA code for the database, followed by the SQL code for the queries. The VBA code is fully documented with comments, while the SQL code is explained in more detail in the user manual, Appendix K.

VBA Code:

```
Option Compare Database
'this form allows the user to enter a new record into the sounding book table and then either exit or
continue to add more records

Private Sub cmdNew_Click() 'submits current record and resets the form to add a new one
    If cmbIMO <> 0 Then
    Dim IMO As Long
    IMO = cmbIMO
    DoCmd.GoToRecord , , acNewRec
    cmbIMO.Value = IMO
    txtDate.SetFocus
    End If
End Sub

Private Sub cmdSubmit_Click() 'submits record and closes the form
    DoCmd.Close , ""
End Sub

Option Compare Database
'this form allows the user to add records to the SWOMS data table

Private Sub cmdNew_Click() 'submits record and resets the form to add another
    Dim IMO As Long
    IMO = cmbIMO
    DoCmd.GoToRecord , , acNewRec
    cmbIMO.Value = IMO
    txtDate.SetFocus
End Sub

Private Sub cmdSubmit_Click() 'submits record and closes the form
    DoCmd.Close , ""
End Sub

Option Compare Database
Dim msg As Integer 'gets value from message box
'this form allows the user to delete data from the SWOMS and Sounding tables

Private Sub cmd_SWOMS_Click()
    If (Not IsNull(Me.cmbIMO)) And txtEnd.Value > txtStart.Value Then 'if correct date range and IMO then
run
        msg = MsgBox("Are you sure you want to delete these records? This cannot be undone.", vbYesNo)
'double-check that user wants to delete
        If msg = 6 Then 'value for pressing Yes
        DoCmd.OpenQuery "Delete SWOMS", acNormal, acEdit
        End If
    Else
    MsgBox ("Missing IMO Number or Incorrect Date Range") 'message if incorrect date or IMO
    End If
End Sub

Private Sub cmdCancel_Click()
    DoCmd.Close
End Sub

Private Sub cmdSound_Click()
```

```
    If (Not IsNull(Me.cmbIMO)) And txtEnd.Value > txtStart.Value Then 'if correct date range and IMO then
run
        msg = MsgBox("Are you sure you want to delete these records? This cannot be undone.", vbYesNo)
'doubke-check user wants to delete
        If msg = 6 Then
            DoCmd.OpenQuery "Delete Sounding", acNormal, acEdit
        End If
    Else
    MsgBox ("Missing IMO Number or Incorrect Date Range") 'message if incorrect date or IMO
    End If
End Sub

Option Compare Database
'this form allows the user to select any and all details for analysis and then can either generate a
preview of the report
'or print it

Private Sub cmbIMO_AfterUpdate() 'sets the tanks list to the currently selected IMO numbeer
    Me.lstTanks.Requery
End Sub

Private Sub cmbIMO_LostFocus() 'sets focus on first textbox on form, since combo box is in header
    txtStart.SetFocus
End Sub

Private Sub cmdComb_Click() 'open combined report if correct date range and select IMO
    If (Not IsNull(Me.cmbIMO)) And txtEnd.Value > txtStart.Value Then
    DoCmd.OpenReport "SWOMSvsSounding", acViewReport
    Else
    MsgBox ("Missing IMO Number or Incorrect Date Range")
    End If
End Sub

Private Sub cmdCPrint_Click() 'print combined report if parameters selected
    If (Not IsNull(Me.cmbIMO)) And txtEnd.Value > txtStart.Value Then
    Report_SWOMSVsSounding.Printer.Orientation = acPRORLandscape
    DoCmd.OpenReport "SWOMSvsSounding", acViewNormal, , , acHidden
    Else
    MsgBox ("Missing IMO Number or Incorrect Date Range")
    End If
End Sub

Private Sub cmdHome_Click() 'opens main menu
    DoCmd.Close , ""
    DoCmd.OpenForm "Main Menu", acNormal, "", "", , acNormal
End Sub

Private Sub cmdMPrint_Click() 'print swoms report if correct parameters
    If (Not IsNull(Me.cmbIMO)) And txtEnd.Value > txtStart.Value Then
    Report_SWOMSAnomalyReport.Printer.Orientation = acPRORLandscape
    DoCmd.OpenReport "SWOMSAnomalyReport", acViewNormal, , , acHidden
    Else
    MsgBox ("Missing IMO Number or Incorrect Date Range")
    End If
End Sub

Private Sub cmdSound_Click() 'open sounding report if correct parameters
    If (Not IsNull(Me.cmbIMO)) And txtEnd.Value > txtStart.Value Then
    DoCmd.OpenReport "SoundingBookAnomalyReport", acViewReport
    Else
    MsgBox ("Missing IMO Number or Incorrect Date Range")
    End If
End Sub

Private Sub cmdSPrint_Click() 'print sounding report if correct parameters
    If (Not IsNull(Me.cmbIMO)) And txtEnd.Value > txtStart.Value Then
    Report_SoundingBookAnomalyReport.Printer.Orientation = acPRORLandscape
    DoCmd.OpenReport "SoundingBookAnomalyReport", acViewNormal, , , acHidden
```

```
    Else
    MsgBox ("Missing IMO Number or Incorrect Date Range")
    End If
End Sub

Private Sub cmdSWOMS_Click() 'open swoms report if correct parameters
    If (Not IsNull(Me.cmbIMO)) And txtEnd.Value > txtStart.Value Then
    DoCmd.OpenReport "SWOMSAnomalyReport", acViewReport
    Else
    MsgBox ("Missing IMO Number or Incorrect Date Range")
    End If
End Sub

Private Sub cmdVessel_Click() 'open the details form for the current vessel
    If Not IsNull(Me.cmbIMO) Then
    DoCmd.OpenForm "VesselDetails", acNormal, "", "", , acNormal
    Form_VesselDetails.txtIMO.Value = Me.cmbIMO.Value 'sets IMO number on details form using value in
combo box
      If Not IsNull(txtIMO) Then
      Dim db As Database
      Dim rs As Recordset
      Dim tmpUser As String

      Set db = CurrentDb
      Set rs = db.OpenRecordset("Select * FROM VesselList " & "WHERE [IMO Number] = " &
Form_VesselDetails.txtIMO.Text)
      'set rest of fields based on IMO number
      Form_VesselDetails.txtName = rs![Vessel Name]
      Form_VesselDetails.txtComp = rs![Shipping Company]
      Form_VesselDetails.txtFlag = rs![Flag Nation]
      Form_VesselDetails.txtType = rs![Vessel Type]

      rs.Close

      Set rs = Nothing
      Set db = Nothing
      End If
    End If
End Sub

Private Sub Form_Open(Cancel As Integer) 'sets focus on IMO box and refreshes the tank list
    cmbIMO.SetFocus
    lstTanks.Requery
End Sub

Option Explicit
Option Compare Database
Dim User As String
'this menu allows access to the data viewing forms, as well as links to add data or to generate a report
based on basic parameters

Private Sub cmdCompare_Click() 'open comparison report if parameters are correct
    If (Not IsNull(Me.cmbIMO)) And txtEnd.Value > txtStart.Value Then
    DoCmd.OpenForm "DetailReports", acNormal 'sets values on detail reports form so query can gather data
properly
    Form_DetailReports.cmbIMO.Value = Me.cmbIMO.Value
    Form_DetailReports.txtStart.Value = Me.txtStart.Value
    Form_DetailReports.txtEnd.Value = Me.txtEnd.Value
    Form_DetailReports.Visible = False
    DoCmd.OpenReport "SWOMSvsSounding", acViewReport, "", "", , acNormal 'opens report
    Else
    MsgBox ("Missing IMO Number or Incorrect Date Range")
    End If
End Sub

Private Sub cmdCPrint_Click() 'print combined report if paramters correct
    If (Not IsNull(Me.cmbIMO)) And txtEnd.Value > txtStart.Value Then
```

```
    DoCmd.OpenForm "DetailReports", acNormal 'sets values on detail reports form so query can gather data
properly
    Form_DetailReports.cmbIMO.Value = Me.cmbIMO.Value
    Form_DetailReports.txtStart.Value = Me.txtStart.Value
    Form_DetailReports.txtEnd.Value = Me.txtEnd.Value
    Form_DetailReports.Visible = False
    Report_SWOMSVsSounding.Printer.Orientation = acPRORLandscape 'set printing to landscape
    DoCmd.OpenReport "SWOMSvsSounding", acViewNormal, , , acHidden 'prints report
    Else
    MsgBox ("Missing IMO Number or Incorrect Date Range")
    End If
End Sub

Private Sub cmdDelete_Click() 'opens delete records form
    DoCmd.OpenForm "DeleteRecords", acNormal
End Sub

Private Sub cmdDetail_Click() 'open detailed reports
    DoCmd.Close , ""
    DoCmd.OpenForm "DetailReports", acNormal, "", "", , acNormal
End Sub

Private Sub cmdMPrint_Click() 'print SWOMS report if the parameters are correct
    If (Not IsNull(Me.cmbIMO)) And txtEnd.Value > txtStart.Value Then
    DoCmd.OpenForm "DetailReports", acNormal 'sets values on detail reports form so query can gather data
properly
    Form_DetailReports.cmbIMO.Value = Me.cmbIMO.Value
    Form_DetailReports.txtStart.Value = Me.txtStart.Value
    Form_DetailReports.txtEnd.Value = Me.txtEnd.Value
    Form_DetailReports.Visible = False
    Report_SWOMSAnomalyReport.Printer.Orientation = acPRORLandscape 'sets printing to landscape
    DoCmd.OpenReport "SWOMSAnomalyReport", acViewNormal, , , acHidden 'prints report
    Else
    MsgBox ("Missing IMO Number or Incorrect Date Range")
    End If
End Sub

Private Sub cmdNewSWOMS_Click() 'open new swoms form
    DoCmd.OpenForm "Add SWOMS", acNormal, "", "", , acNormal
End Sub

Private Sub cmdNewVessel_Click() 'open new vessel form
    DoCmd.OpenForm "NewVessel", acNormal, "", "", , acNormal
End Sub

Private Sub cmdReport_Click() 'open SWOMS anomaly report if IMO and dates are correct
    If (Not IsNull(Me.cmbIMO)) And txtEnd.Value > txtStart.Value Then
    DoCmd.OpenForm "DetailReports", acNormal 'sets values on detail reports form so query can gather data
properly
    Form_DetailReports.cmbIMO.Value = Me.cmbIMO.Value
    Form_DetailReports.txtStart.Value = Me.txtStart.Value
    Form_DetailReports.txtEnd.Value = Me.txtEnd.Value
    Form_DetailReports.Visible = False
    DoCmd.OpenReport "SWOMSAnomalyReport", acViewReport, "", "", , acNormal
    Else
    MsgBox ("Missing IMO Number or Incorrect Date Range")
    End If
End Sub

Private Sub cmdSound_Click() 'opens add sounding form
    DoCmd.Close , ""
    DoCmd.OpenForm "Add Sounding", acNormal, "", "", , acNormal
End Sub

Private Sub cmdSounddata_Click() 'opens the sounding data form
    DoCmd.Close , ""
    DoCmd.OpenForm "SoundingBookData", acNormal, "", "", , acNormal
End Sub
```

```vba
Private Sub cmdSounding_Click() 'opens sounding report if parameters are correct
    If (Not IsNull(Me.cmbIMO)) And txtEnd.Value > txtStart.Value Then
    DoCmd.OpenForm "DetailReports", acNormal 'sets values on detail reports form so query can gather data
properly
    Form_DetailReports.cmbIMO.Value = Me.cmbIMO.Value
    Form_DetailReports.txtStart.Value = Me.txtStart.Value
    Form_DetailReports.txtEnd.Value = Me.txtEnd.Value
    Form_DetailReports.Visible = False
    DoCmd.OpenReport "SoundingBookAnomalyReport", acViewReport, "", "", , acNormal
    Else
    MsgBox ("Missing IMO Number or Incorrect Date Range")
    End If
End Sub

Private Sub cmdSPrint_Click() 'print sounding book report if parameters are correct
    If (Not IsNull(Me.cmbIMO)) And txtEnd.Value > txtStart.Value Then
    DoCmd.OpenForm "DetailReports", acNormal 'sets values on detail reports form so query can gather data
properly
    Form_DetailReports.cmbIMO.Value = Me.cmbIMO.Value
    Form_DetailReports.txtStart.Value = Me.txtStart.Value
    Form_DetailReports.txtEnd.Value = Me.txtEnd.Value
    Form_DetailReports.Visible = False
    Report_SoundingBookAnomalyReport.Printer.Orientation = acPRORLandscape 'set printing to landscape
    DoCmd.OpenReport "SoundingBookAnomalyReport", acViewNormal, , , acHidden 'print report
    Else
    MsgBox ("Missing IMO Number or Incorrect Date Range")
    End If
End Sub

Private Sub cmdSWOMS_Click() 'open swoms data
    DoCmd.Close , ""
    DoCmd.OpenForm "SWOMSData", acNormal, "", "", , acNormal
End Sub

Private Sub cmdTank_Click() 'open tank form
    DoCmd.OpenForm "TanksData", acNormal, "", "", , acNormal
End Sub

Private Sub cmdVessel_Click() 'open vessel list
    DoCmd.Close , ""
    DoCmd.OpenForm "VesselList", acNormal, "", "", , acNormal
End Sub

Private Sub Form_Load() 'set printer margins so reports will print properly
    'Print margins, Left margin
    Application.SetOption "Left Margin", 0.2
    'Print margins, Right margin
    Application.SetOption "Right Margin", 0.2
    'Print margins, Top margin
    Application.SetOption "Top Margin", 0.5
    'Print margins, Bottom margin
    Application.SetOption "Bottom Margin", 0.5

End Sub

Option Compare Database
'This form allows the user to enter a new vessel into the database and then leads them into the tank form
to enter that data as well

Private Sub cmdSubmit_Click() 'updates vessel table and opens tank form with IMO of new vessel selected
    txtID.SetFocus
    cmdSubmit.Enabled = False
    If (Not IsNull(Me.txtID)) And (Not IsNull(Me.txtName)) And (Not IsNull(Me.txtCompany)) And (Not
IsNull(Me.txtFlag)) And (Not IsNull(Me.txtType)) Then
        Dim strSQL As String
        DoCmd.SetWarnings False 'disable warnings
        strSQL = "INSERT INTO VesselTanks ([IMO Number]) VALUES (" & txtID.Value & ")"
```

```
        DoCmd.RunSQL strSQL
        DoCmd.Close , ""
        DoCmd.OpenForm "TanksData", acNormal, "", "", , acNormal
        Form_TanksData.cmbIMO.Requery
        Form_TanksData.cmbIMO.Value = Form_NewVessel.txtID.Value
        Form_TanksData.t1t.SetFocus
    Else
        MsgBox ("Missing Information")
    End If
End Sub

Private Sub txtType_LostFocus() 'puts focus on command add in footer if you hit tab after entering last field
    cmdSubmit.Enabled = True
    cmdSubmit.SetFocus
End Sub

Option Compare Database
'this form allows the user to view Sounding Book Data for a specific IMO number based on a date range

Private Sub cmbIMO_AfterUpdate() 'updates table based on IMO number
    DoCmd.Requery
End Sub

Private Sub cmdAdd_Click() 'opens form to add data
    DoCmd.OpenForm "Add Sounding", acNormal, "", "", , acNormal
End Sub

Private Sub cmdHome_Click() 'closes and returns to main menu
    DoCmd.Close , ""
    DoCmd.OpenForm "Main Menu", acNormal, "", "", , acNormal
End Sub

Private Sub cmdVessel_Click() 'opens vessel details form and populates it for the currently selected IMO number
    If Not IsNull(Me.cmbIMO) Then
    DoCmd.OpenForm "VesselDetails", acNormal, "", "", , acNormal
    Form_VesselDetails.txtIMO.Value = Me.cmbIMO.Value
      If Not IsNull(txtIMO) Then 'prevents it from running if no IMO is selected
      Dim db As Database
      Dim rs As Recordset
      Dim tmpUser As String

      Set db = CurrentDb
      Set rs = db.OpenRecordset("Select * FROM VesselList " & "WHERE [IMO Number] = " & Form_VesselDetails.txtIMO.Text)
      'set rest of fields based on IMO number
      Form_VesselDetails.txtName = rs![Vessel Name]
      Form_VesselDetails.txtComp = rs![Shipping Company]
      Form_VesselDetails.txtFlag = rs![Flag Nation]
      Form_VesselDetails.txtType = rs![Vessel Type]

      rs.Close

      Set rs = Nothing
      Set db = Nothing
      End If

    End If
End Sub

Private Sub txtStart_AfterUpdate() 'resets data based on date
    Me.Requery
End Sub

Private Sub txtEnd_AfterUpdate() 'resets data based on date
    Me.Requery
End Sub
```

```
Option Compare Database
'this form allows the user to view SWOMS data for a specific vessel within a selected date range

Private Sub cmbIMO_AfterUpdate() 'updates based on current IMO
    Me.Requery
End Sub

Private Sub cmdAdd_Click() 'opens the add swoms form
    DoCmd.OpenForm "Add SWOMS", acNormal, "", "", , acNormal
End Sub

Private Sub cmdHome_Click() 'returns to the main menu
    DoCmd.Close , ""
    DoCmd.OpenForm "Main Menu", acNormal, "", "", , acNormal
End Sub

Private Sub cmdVessel_Click() 'opens vessel details form and populates it based on currently selected IMO
    If Not IsNull(Me.cmbIMO) Then
    DoCmd.OpenForm "VesselDetails", acNormal, "", "", , acNormal
    Form_VesselDetails.txtIMO.Value = Me.cmbIMO.Value
      If Not IsNull(txtIMO) Then 'prevents it from running if no IMO selected
      Dim db As Database
      Dim rs As Recordset
      Dim tmpUser As String

      Set db = CurrentDb
      Set  rs  =  db.OpenRecordset("Select  *  FROM  VesselList  "  &  "WHERE  [IMO  Number]  =  "  &
Form_VesselDetails.txtIMO.Text)
      'set rest of fields based on IMO number
      Form_VesselDetails.txtName = rs![Vessel Name]
      Form_VesselDetails.txtComp = rs![Shipping Company]
      Form_VesselDetails.txtFlag = rs![Flag Nation]
      Form_VesselDetails.txtType = rs![Vessel Type]

      rs.Close

      Set rs = Nothing
      Set db = Nothing
      End If
    End If

End Sub

Private Sub txtStart_AfterUpdate() 'updates table
    Me.Requery
End Sub

Private Sub txtEnd_AfterUpdate() 'updates table
    Me.Requery
End Sub

Option Compare Database
Dim gallons As Boolean 'keep track whether displaying in gallons or not
'this form allows the user to view and edit the tank information stored for the vessels

Private Sub cmbIMO_AfterUpdate() 'updates the form based on currently selected IMO number
    Dim db As Database
    Dim rs As Recordset
    Dim tmpUser As String

    Set db = CurrentDb
    Set rs = db.OpenRecordset("Select * FROM VesselTanks " & "WHERE [IMO Number] = " & Me.cmbIMO.Value)
    'populate Tanks Data form with appropiate tanks based on selected IMO number
    Me.t1t = rs![Tank01 Type]
    Me.t1d = rs![Tank01 Desc]
    Me.t1c = rs![Tank01 Cap]
    Me.t2t = rs![Tank02 Type]
```

```
    Me.t2d = rs![Tank02 Desc]
    Me.t2c = rs![Tank02 Cap]
    Me.t3t = rs![Tank03 Type]
    Me.t3d = rs![Tank03 Desc]
    Me.t3c = rs![Tank03 Cap]
    Me.t4t = rs![Tank04 Type]
    Me.t4d = rs![Tank04 Desc]
    Me.t4c = rs![Tank04 Cap]
    Me.t5t = rs![Tank05 Type]
    Me.t5d = rs![Tank05 Desc]
    Me.t5c = rs![Tank05 Cap]
    Me.t6t = rs![Tank06 Type]
    Me.t6d = rs![Tank06 Desc]
    Me.t6c = rs![Tank06 Cap]
    Me.t7t = rs![Tank07 Type]
    Me.t7d = rs![Tank07 Desc]
    Me.t7c = rs![Tank07 Cap]
    Me.t8t = rs![Tank08 Type]
    Me.t8d = rs![Tank08 Desc]
    Me.t8c = rs![Tank08 Cap]
    Me.t9t = rs![Tank09 Type]
    Me.t9d = rs![Tank09 Desc]
    Me.t9c = rs![Tank09 Cap]
    Me.t10t = rs![Tank10 Type]
    Me.t10d = rs![Tank10 Desc]
    Me.t10c = rs![Tank10 Cap]
    Me.t11t = rs![Tank11 Type]
    Me.t11d = rs![Tank11 Desc]
    Me.t11c = rs![Tank11 Cap]
    Me.t12t = rs![Tank12 Type]
    Me.t12d = rs![Tank12 Desc]
    Me.t12c = rs![Tank12 Cap]
    Me.t13t = rs![Tank13 Type]
    Me.t13d = rs![Tank13 Desc]
    Me.t13c = rs![Tank13 Cap]
    Me.t14t = rs![Tank14 Type]
    Me.t14d = rs![Tank14 Desc]
    Me.t14c = rs![Tank14 Cap]
    Me.t15t = rs![Tank15 Type]
    Me.t15d = rs![Tank15 Desc]
    Me.t15c = rs![Tank15 Cap]
    Me.t16t = rs![Tank16 Type]
    Me.t16d = rs![Tank16 Desc]
    Me.t16c = rs![Tank16 Cap]

    rs.Close

    Set rs = Nothing
    Set db = Nothing

End Sub

Private Sub cmbUnit_AfterUpdate() 'allows the user to switch between viewing capacity in cubic meters or
gallons
    'check what value to switch to and make sure that if the same thing is selected twice the math isn't
redone
    If cmbUnit.Value = "Gallons" And gallons = False Then 'converts to gallons
        t1c.Value = t1c.Value * 264.172
        t2c.Value = t2c.Value * 264.172
        t3c.Value = t3c.Value * 264.172
        t4c.Value = t4c.Value * 264.172
        t5c.Value = t5c.Value * 264.172
        t6c.Value = t6c.Value * 264.172
        t7c.Value = t7c.Value * 264.172
        t8c.Value = t8c.Value * 264.172
        t9c.Value = t9c.Value * 264.172
        t10c.Value = t10c.Value * 264.172
        t11c.Value = t11c.Value * 264.172
```

```
        t12c.Value = t12c.Value * 264.172
        t13c.Value = t13c.Value * 264.172
        t14c.Value = t14c.Value * 264.172
        t15c.Value = t15c.Value * 264.172
        t16c.Value = t16c.Value * 264.172
    End If
    If cmbUnit.Value = "Cubic Meters" And gallons = True Then 'converts to cubic meters
        t1c.Value = t1c.Value / 264.172
        t2c.Value = t2c.Value / 264.172
        t3c.Value = t3c.Value / 264.172
        t4c.Value = t4c.Value / 264.172
        t5c.Value = t5c.Value / 264.172
        t6c.Value = t6c.Value / 264.172
        t7c.Value = t7c.Value / 264.172
        t8c.Value = t8c.Value / 264.172
        t9c.Value = t9c.Value / 264.172
        t10c.Value = t10c.Value / 264.172
        t11c.Value = t11c.Value / 264.172
        t12c.Value = t12c.Value / 264.172
        t13c.Value = t13c.Value / 264.172
        t14c.Value = t14c.Value / 264.172
        t15c.Value = t15c.Value / 264.172
        t16c.Value = t16c.Value / 264.172
    End If
    If cmbUnit.Value = "Gallons" Then
        gallons = True
    Else
        gallons = False
    End If

End Sub

Private Sub cmdCancel_Click() 'closes without updating
    DoCmd.Close , ""
End Sub

Private Sub cmdClose_Click() 'update any and all changed values for the tanks
    Dim strSQL As String
    DoCmd.SetWarnings False 'disable warnings
    If gallons = True Then 'resets to cubic meters if necessary to prevent data in table from changing
        t1c.Value = t1c.Value / 264.172
        t2c.Value = t2c.Value / 264.172
        t3c.Value = t3c.Value / 264.172
        t4c.Value = t4c.Value / 264.172
        t5c.Value = t5c.Value / 264.172
        t6c.Value = t6c.Value / 264.172
        t7c.Value = t7c.Value / 264.172
        t8c.Value = t8c.Value / 264.172
        t9c.Value = t9c.Value / 264.172
        t10c.Value = t10c.Value / 264.172
        t11c.Value = t11c.Value / 264.172
        t12c.Value = t12c.Value / 264.172
        t13c.Value = t13c.Value / 264.172
        t14c.Value = t14c.Value / 264.172
        t15c.Value = t15c.Value / 264.172
        t16c.Value = t16c.Value / 264.172
        gallons = False
    End If
    If Not IsNull(Me.cmbIMO) Then 'updates tank data for selected IMO
    'type
    strSQL = "UPDATE VesselTanks SET [Tank01 Type]= '" & Me!t1t & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank02 Type]= '" & Me!t2t & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank03 Type]= '" & Me!t3t & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
```

```
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank04 Type]= '" & Me!t4t & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank05 Type]= '" & Me!t5t & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank06 Type]= '" & Me!t6t & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank07 Type]= '" & Me!t7t & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank08 Type]= '" & Me!t8t & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank09 Type]= '" & Me!t9t & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank10 Type]= '" & Me!t10t & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank11 Type]= '" & Me!t11t & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank12 Type]= '" & Me!t12t & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank13 Type]= '" & Me!t13t & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank14 Type]= '" & Me!t14t & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank15 Type]= '" & Me!t15t & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank16 Type]= '" & Me!t16t & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    'description
    strSQL = "UPDATE VesselTanks SET [Tank01 Desc]= '" & Me!t1d & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank02 Desc]= '" & Me!t2d & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank03 Desc]= '" & Me!t3d & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank04 Desc]= '" & Me!t4d & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank05 Desc]= '" & Me!t5d & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank06 Desc]= '" & Me!t6d & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank07 Desc]= '" & Me!t7d & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank08 Desc]= '" & Me!t8d & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank09 Desc]= '" & Me!t9d & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
```

```
    strSQL = "UPDATE VesselTanks SET [Tank10 Desc]= '" & Me!t10d & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank11 Desc]= '" & Me!t11d & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank12 Desc]= '" & Me!t12d & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank13 Desc]= '" & Me!t13d & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank14 Desc]= '" & Me!t14d & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank15 Desc]= '" & Me!t15d & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank16 Desc]= '" & Me!t16d & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    'capacity
    strSQL = "UPDATE VesselTanks SET [Tank01 Cap]= '" & Me!t1c & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank02 Cap]= '" & Me!t2c & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank03 Cap]= '" & Me!t3c & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank04 Cap]= '" & Me!t4c & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank05 Cap]= '" & Me!t5c & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank06 Cap]= '" & Me!t6c & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank07 Cap]= '" & Me!t7c & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank08 Cap]= '" & Me!t8c & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank09 Cap]= '" & Me!t9c & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank10 Cap]= '" & Me!t10c & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank11 Cap]= '" & Me!t11c & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank12 Cap]= '" & Me!t12c & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank13 Cap]= '" & Me!t13c & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank14 Cap]= '" & Me!t14c & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselTanks SET [Tank15 Cap]= '" & Me!t15c & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
```

```
    strSQL = "UPDATE VesselTanks SET [Tank16 Cap]= '" & Me!t16c & "'" & " WHERE [IMO Number] = " &
Me!cmbIMO.Value
    DoCmd.RunSQL strSQL
    'tank names
    strSQL = "UPDATE TankNames SET [Tank Name]= '" & Me!t1d & "'" & " WHERE [IMO Number] = " & Me!cmbIMO &
" AND [Tank Number] = '" & 1 & "'"
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE TankNames SET [Tank Name]= '" & Me!t2d & "'" & " WHERE [IMO Number] = " & Me!cmbIMO &
" AND [Tank Number] = '" & 2 & "'"
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE TankNames SET [Tank Name]= '" & Me!t3d & "'" & " WHERE [IMO Number] = " & Me!cmbIMO &
" AND [Tank Number] = '" & 3 & "'"
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE TankNames SET [Tank Name]= '" & Me!t4d & "'" & " WHERE [IMO Number] = " & Me!cmbIMO &
" AND [Tank Number] = '" & 4 & "'"
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE TankNames SET [Tank Name]= '" & Me!t5d & "'" & " WHERE [IMO Number] = " & Me!cmbIMO &
" AND [Tank Number] = '" & 5 & "'"
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE TankNames SET [Tank Name]= '" & Me!t6d & "'" & " WHERE [IMO Number] = " & Me!cmbIMO &
" AND [Tank Number] = '" & 6 & "'"
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE TankNames SET [Tank Name]= '" & Me!t7d & "'" & " WHERE [IMO Number] = " & Me!cmbIMO &
" AND [Tank Number] = '" & 7 & "'"
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE TankNames SET [Tank Name]= '" & Me!t8d & "'" & " WHERE [IMO Number] = " & Me!cmbIMO &
" AND [Tank Number] = '" & 8 & "'"
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE TankNames SET [Tank Name]= '" & Me!t9d & "'" & " WHERE [IMO Number] = " & Me!cmbIMO &
" AND [Tank Number] = '" & 9 & "'"
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE TankNames SET [Tank Name]= '" & Me!t10d & "'" & " WHERE [IMO Number] = " & Me!cmbIMO
& " AND [Tank Number] = '" & 10 & "'"
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE TankNames SET [Tank Name]= '" & Me!t11d & "'" & " WHERE [IMO Number] = " & Me!cmbIMO
& " AND [Tank Number] = '" & 11 & "'"
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE TankNames SET [Tank Name]= '" & Me!t12d & "'" & " WHERE [IMO Number] = " & Me!cmbIMO
& " AND [Tank Number] = '" & 12 & "'"
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE TankNames SET [Tank Name]= '" & Me!t13d & "'" & " WHERE [IMO Number] = " & Me!cmbIMO
& " AND [Tank Number] = '" & 13 & "'"
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE TankNames SET [Tank Name]= '" & Me!t14d & "'" & " WHERE [IMO Number] = " & Me!cmbIMO
& " AND [Tank Number] = '" & 14 & "'"
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE TankNames SET [Tank Name]= '" & Me!t15d & "'" & " WHERE [IMO Number] = " & Me!cmbIMO
& " AND [Tank Number] = '" & 15 & "'"
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE TankNames SET [Tank Name]= '" & Me!t16d & "'" & " WHERE [IMO Number] = " & Me!cmbIMO
& " AND [Tank Number] = '" & 16 & "'"
    DoCmd.RunSQL strSQL
    End If

    DoCmd.Close , ""
End Sub

Option Compare Database
'This form allows the user to view and edit details of a ship, though they are not allowed to change the
IMO number

Private Sub cmdClose_Click() 'closes and updates data
    Dim strSQL As String
    DoCmd.SetWarnings False 'disable warnings
    If Not IsNull(Me.txtIMO) Then
    strSQL = "UPDATE VesselList SET [Vessel Name]= '" & Me!txtName & "' WHERE [IMO Number] = " & Me!txtIMO
    DoCmd.RunSQL strSQL
```

```
    strSQL = "UPDATE VesselList SET [Shipping Company]= '" & Me!txtComp & "' WHERE [IMO Number] = " &
Me!txtIMO
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselList SET [Flag Nation]= '" & Me!txtFlag & "' WHERE [IMO Number] = " & Me!txtIMO
    DoCmd.RunSQL strSQL
    strSQL = "UPDATE VesselList SET [Vessel Type]= '" & Me!txtType & "' WHERE [IMO Number] = " & Me!txtIMO
    DoCmd.RunSQL strSQL
    End If
    Form_VesselList.Requery
    DoCmd.Close , ""
End Sub

Private Sub cmdTanks_Click() 'opens tank form and sets it up based on IMO number of details form
     DoCmd.OpenForm "TanksData", acNormal, "", "", , acNormal
     Form_TanksData.cmbIMO.Value = Form_VesselDetails.txtIMO.Value 'opens tank form with correct IMO
     Dim db As Database
    Dim rs As Recordset
    Dim tmpUser As String

    Set db = CurrentDb
    Set rs = db.OpenRecordset("Select * FROM VesselTanks " & "WHERE [IMO Number] = " &
Form_TanksData.cmbIMO.Value)
    'populate Tanks Data form with appropiate tanks based on selected IMO number
    Form_TanksData.t1t = rs![Tank01 Type]
    Form_TanksData.t1d = rs![Tank01 Desc]
    Form_TanksData.t1c = rs![Tank01 Cap]
    Form_TanksData.t2t = rs![Tank02 Type]
    Form_TanksData.t2d = rs![Tank02 Desc]
    Form_TanksData.t2c = rs![Tank02 Cap]
    Form_TanksData.t3t = rs![Tank03 Type]
    Form_TanksData.t3d = rs![Tank03 Desc]
    Form_TanksData.t3c = rs![Tank03 Cap]
    Form_TanksData.t4t = rs![Tank04 Type]
    Form_TanksData.t4d = rs![Tank04 Desc]
    Form_TanksData.t4c = rs![Tank04 Cap]
    Form_TanksData.t5t = rs![Tank05 Type]
    Form_TanksData.t5d = rs![Tank05 Desc]
    Form_TanksData.t5c = rs![Tank05 Cap]
    Form_TanksData.t6t = rs![Tank06 Type]
    Form_TanksData.t6d = rs![Tank06 Desc]
    Form_TanksData.t6c = rs![Tank06 Cap]
    Form_TanksData.t7t = rs![Tank07 Type]
    Form_TanksData.t7d = rs![Tank07 Desc]
    Form_TanksData.t7c = rs![Tank07 Cap]
    Form_TanksData.t8t = rs![Tank08 Type]
    Form_TanksData.t8d = rs![Tank08 Desc]
    Form_TanksData.t8c = rs![Tank08 Cap]
    Form_TanksData.t9t = rs![Tank09 Type]
    Form_TanksData.t9d = rs![Tank09 Desc]
    Form_TanksData.t9c = rs![Tank09 Cap]
    Form_TanksData.t10t = rs![Tank10 Type]
    Form_TanksData.t10d = rs![Tank10 Desc]
    Form_TanksData.t10c = rs![Tank10 Cap]
    Form_TanksData.t11t = rs![Tank11 Type]
    Form_TanksData.t11d = rs![Tank11 Desc]
    Form_TanksData.t11c = rs![Tank11 Cap]
    Form_TanksData.t12t = rs![Tank12 Type]
    Form_TanksData.t12d = rs![Tank12 Desc]
    Form_TanksData.t12c = rs![Tank12 Cap]
    Form_TanksData.t13t = rs![Tank13 Type]
    Form_TanksData.t13d = rs![Tank13 Desc]
    Form_TanksData.t13c = rs![Tank13 Cap]
    Form_TanksData.t14t = rs![Tank14 Type]
    Form_TanksData.t14d = rs![Tank14 Desc]
    Form_TanksData.t14c = rs![Tank14 Cap]
    Form_TanksData.t15t = rs![Tank15 Type]
    Form_TanksData.t15d = rs![Tank15 Desc]
    Form_TanksData.t15c = rs![Tank15 Cap]
```

```
    Form_TanksData.t16t = rs![Tank16 Type]
    Form_TanksData.t16d = rs![Tank16 Desc]
    Form_TanksData.t16c = rs![Tank16 Cap]

    rs.Close

    Set rs = Nothing
    Set db = Nothing

End Sub

Private Sub Form_Open(Cancel As Integer) 'populates form based on the IMO number set by another form
    If Not IsNull(txtIMO) Then 'runs if IMO number is selected
    Dim db As Database
    Dim rs As Recordset
    Dim tmpUser As String

    Set db = CurrentDb
    Set rs = db.OpenRecordset("Select * FROM VesselList " & "WHERE [IMO Number] = '" & Me.txtIMO.Text &
"'")
    'set rest of fields based on IMO number
    Me.txtName = rs![Vessel Name]
    Me.txtComp = rs![Shipping Company]
    Me.txtFlag = rs![Flag Nation]
    Me.txtType = rs![Vessel Type]

    rs.Close

    Set rs = Nothing
    Set db = Nothing
    End If
End Sub

Option Compare Database
'this form allows users to view all vessels within the database

Private Sub cmdAdd_Click() 'opens new vessel form
    DoCmd.OpenForm "New Vessel", acNormal, "", "", , acNormal
End Sub

Private Sub cmdHome_Click() 'closes and returns to main
    DoCmd.Close , ""
    DoCmd.OpenForm "Main Menu", acNormal, "", "", , acNormal
End Sub

'the following 5 sections open the vessel details form for whatever vessel the user double-clicks the
field of
Private Sub txtComp_DblClick(Cancel As Integer)
    DoCmd.OpenForm "VesselDetails", acNormal, "", "", , acNormal
    Form_VesselDetails.txtIMO.Value = Me.txtIMO.Value
      If Not IsNull(txtIMO) Then
      Dim db As Database
      Dim rs As Recordset
      Dim tmpUser As String

      Set db = CurrentDb
      Set rs = db.OpenRecordset("Select * FROM VesselList " & "WHERE [IMO Number] = " &
Form_VesselDetails.txtIMO.Text)
      'set rest of fields based on IMO number
      Form_VesselDetails.txtName = rs![Vessel Name]
      Form_VesselDetails.txtComp = rs![Shipping Company]
      Form_VesselDetails.txtFlag = rs![Flag Nation]
      Form_VesselDetails.txtType = rs![Vessel Type]

      rs.Close

      Set rs = Nothing
      Set db = Nothing
```

```
      End If
End Sub

Private Sub txtFlag_DblClick(Cancel As Integer)
    DoCmd.OpenForm "VesselDetails", acNormal, "", "", , acNormal
    Form_VesselDetails.txtIMO.Value = Me.txtIMO.Value
      If Not IsNull(txtIMO) Then
      Dim db As Database
      Dim rs As Recordset
      Dim tmpUser As String

      Set db = CurrentDb
      Set   rs   =   db.OpenRecordset("Select   *   FROM   VesselList   "   &   "WHERE   [IMO   Number]   =   "   &
Form_VesselDetails.txtIMO.Text)
      'set rest of fields based on IMO number
      Form_VesselDetails.txtName = rs![Vessel Name]
      Form_VesselDetails.txtComp = rs![Shipping Company]
      Form_VesselDetails.txtFlag = rs![Flag Nation]
      Form_VesselDetails.txtType = rs![Vessel Type]

      rs.Close

      Set rs = Nothing
      Set db = Nothing
      End If
End Sub

Private Sub txtIMO_DblClick(Cancel As Integer)
   DoCmd.OpenForm "VesselDetails", acNormal, "", "", , acNormal
    Form_VesselDetails.txtIMO.Value = Me.txtIMO.Value
      If Not IsNull(txtIMO) Then
      Dim db As Database
      Dim rs As Recordset
      Dim tmpUser As String

      Set db = CurrentDb
      Set   rs   =   db.OpenRecordset("Select   *   FROM   VesselList   "   &   "WHERE   [IMO   Number]   =   "   &
Form_VesselDetails.txtIMO.Text)
      'set rest of fields based on IMO number
      Form_VesselDetails.txtName = rs![Vessel Name]
      Form_VesselDetails.txtComp = rs![Shipping Company]
      Form_VesselDetails.txtFlag = rs![Flag Nation]
      Form_VesselDetails.txtType = rs![Vessel Type]

      rs.Close

      Set rs = Nothing
      Set db = Nothing
      End If
End Sub

Private Sub txtType_DblClick(Cancel As Integer)
    DoCmd.OpenForm "VesselDetails", acNormal, "", "", , acNormal
    Form_VesselDetails.txtIMO.Value = Me.txtIMO.Value
      If Not IsNull(txtIMO) Then
      Dim db As Database
      Dim rs As Recordset
      Dim tmpUser As String

      Set db = CurrentDb
      Set   rs   =   db.OpenRecordset("Select   *   FROM   VesselList   "   &   "WHERE   [IMO   Number]   =   "   &
Form_VesselDetails.txtIMO.Text)
      'set rest of fields based on IMO number
      Form_VesselDetails.txtName = rs![Vessel Name]
      Form_VesselDetails.txtComp = rs![Shipping Company]
      Form_VesselDetails.txtFlag = rs![Flag Nation]
      Form_VesselDetails.txtType = rs![Vessel Type]
```

```
      rs.Close

      Set rs = Nothing
      Set db = Nothing
      End If
End Sub

Private Sub txtvessel_DblClick(Cancel As Integer)
    DoCmd.OpenForm "VesselDetails", acNormal, "", "", , acNormal
    Form_VesselDetails.txtIMO.Value = Me.txtIMO.Value
      If Not IsNull(txtIMO) Then
      Dim db As Database
      Dim rs As Recordset
      Dim tmpUser As String

      Set db = CurrentDb
      Set rs = db.OpenRecordset("Select * FROM VesselList " & "WHERE [IMO Number] = " &
Form_VesselDetails.txtIMO.Text)
      'set rest of fields based on IMO number
      Form_VesselDetails.txtName = rs![Vessel Name]
      Form_VesselDetails.txtComp = rs![Shipping Company]
      Form_VesselDetails.txtFlag = rs![Flag Nation]
      Form_VesselDetails.txtType = rs![Vessel Type]

      rs.Close

      Set rs = Nothing
      Set db = Nothing
      End If
End Sub

Option Compare Database
Dim lngRed As Long

Private Sub Report_Open(Cancel As Integer) 'changes text of fields selected by user to red
    lngRed = RGB(255, 0, 0)
    If Form_DetailReports.chkBilge = -1 Then
        [Bilge Pump Runs_Label].ForeColor = lngRed
        [Bilge Pump Runs].ForeColor = lngRed
        [Bilge Pump Run Time_Label].ForeColor = lngRed
        [Bilge Pump Run Time].ForeColor = lngRed
    End If
    If Form_DetailReports.chkOWS = -1 Then
        [OWS Runs_Label].ForeColor = lngRed
        [OWS Runs].ForeColor = lngRed
        [OWS Run Time_Label].ForeColor = lngRed
        [OWS Run Time].ForeColor = lngRed
    End If
    If Form_DetailReports.chkOver = -1 Then
        [Overboard Opened_Label].ForeColor = lngRed
        [Overboard Opened].ForeColor = lngRed
        [Overboard Open Time_Label].ForeColor = lngRed
        [Overboard Open Time].ForeColor = lngRed
    End If
End Sub

Option Compare Database
Dim lngRed As Long

Private Sub Report_Open(Cancel As Integer) 'changes text of fields selected by user to red
    lngRed = RGB(255, 0, 0)
    If Form_DetailReports.chkPPM = -1 Then
        [PPM Alarms_Label].ForeColor = lngRed
        [PPM Alarms].ForeColor = lngRed
        [Average PPM Overboard_Label].ForeColor = lngRed
        [Average PPM Overboard].ForeColor = lngRed
    End If
    If Form_DetailReports.chkOCM = -1 Then
```

```
        [OCM Fresh Water Valve Opened_Label].ForeColor = lngRed
        [OCM Fresh Water Valve Opened].ForeColor = lngRed
    End If
    If Form_DetailReports.chkOil = -1 Then
        [Oil Purge Valve Opened_Label].ForeColor = lngRed
        [Oil Purge Valve Opened].ForeColor = lngRed
        [Oil Purge Valve Open Time_Label].ForeColor = lngRed
        [Oil Purge Valve Open Time].ForeColor = lngRed
    End If
    If Form_DetailReports.chkInc = -1 Then
        [Incinerator Runs_Label].ForeColor = lngRed
        [Incinerator Runs].ForeColor = lngRed
        [Incinerator Run Time_Label].ForeColor = lngRed
        [Incinerator Run Time].ForeColor = lngRed
    End If
End Sub

Option Compare Database
Dim lngRed As Long

Private Sub Report_Open(Cancel As Integer)
    Dim IMO As Long
    lngRed = RGB(255, 0, 0)
    IMO = Form_DetailReports.cmbIMO.Value
    'sets labels on report to proper tank names
    lbl1.Caption = Nz(DLookup("[Tank01 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl2.Caption = Nz(DLookup("[Tank02 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl3.Caption = Nz(DLookup("[Tank03 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl4.Caption = Nz(DLookup("[Tank04 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl5.Caption = Nz(DLookup("[Tank05 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl6.Caption = Nz(DLookup("[Tank06 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl7.Caption = Nz(DLookup("[Tank07 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl8.Caption = Nz(DLookup("[Tank08 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    'changes text of fields selected by user to red
        If Form_DetailReports.lstTanks.Selected(0) Then
                lbl1.ForeColor = lngRed
                [Diff_Label].ForeColor = lngRed
                [Diff].ForeColor = lngRed
                [Percent_Label].ForeColor = lngRed
                [Percent].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(1) Then
                lbl2.ForeColor = lngRed
                [Diff2_Label].ForeColor = lngRed
                [Diff2].ForeColor = lngRed
                [Percent2_Label].ForeColor = lngRed
                [Percent2].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(2) Then
                lbl3.ForeColor = lngRed
                [Diff3_Label].ForeColor = lngRed
                [Diff3].ForeColor = lngRed
                [Percent3_Label].ForeColor = lngRed
                [Percent3].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(3) Then
                lbl4.ForeColor = lngRed
                [Diff4_Label].ForeColor = lngRed
                [Diff4].ForeColor = lngRed
                [Percent4_Label].ForeColor = lngRed
                [Percent4].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(4) Then
                lbl5.ForeColor = lngRed
                [Diff5_Label].ForeColor = lngRed
                [Diff5].ForeColor = lngRed
                [Percent5_Label].ForeColor = lngRed
                [Percent5].ForeColor = lngRed
```

```
        End If
        If Form_DetailReports.lstTanks.Selected(5) Then
                lbl6.ForeColor = lngRed
                [Diff6_Label].ForeColor = lngRed
                [Diff6].ForeColor = lngRed
                [Percent6_Label].ForeColor = lngRed
                [Percent6].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(6) Then
                lbl7.ForeColor = lngRed
                [Diff7_Label].ForeColor = lngRed
                [Diff7].ForeColor = lngRed
                [Percent7_Label].ForeColor = lngRed
                [Percent7].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(7) Then
                lbl8.ForeColor = lngRed
                [Diff8_Label].ForeColor = lngRed
                [Diff8].ForeColor = lngRed
                [Percent8_Label].ForeColor = lngRed
                [Percent8].ForeColor = lngRed
        End If
End Sub

Option Compare Database

Private Sub Report_Open(Cancel As Integer)
    Dim IMO As Long
    Dim lngRed As Long
    lngRed = RGB(255, 0, 0)
    IMO = Form_DetailReports.cmbIMO.Value
    'sets labels on report to proper tank names
    lbl9.Caption = Nz(DLookup("[Tank09 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl10.Caption = Nz(DLookup("[Tank10 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl11.Caption = Nz(DLookup("[Tank11 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl12.Caption = Nz(DLookup("[Tank12 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl13.Caption = Nz(DLookup("[Tank13 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl14.Caption = Nz(DLookup("[Tank14 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl15.Caption = Nz(DLookup("[Tank15 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl16.Caption = Nz(DLookup("[Tank16 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    'hide if no tanks have values
    If  lbl9.Caption  =  ""  And  lbl10.Caption  =  ""  And  lbl11.Caption  =  ""  And  lbl12.Caption  =  ""  And
lbl13.Caption = "" And lbl14.Caption = "" And lbl15.Caption = "" And lbl16.Caption = "" Then
        Me.Visible = False
        Exit Sub
    Else
        Me.Visible = True
    End If
    'changes text of fields selected by user to red
        If Form_DetailReports.lstTanks.Selected(8) Then
                lbl9.ForeColor = lngRed
                [Diff9_Label].ForeColor = lngRed
                [Diff9].ForeColor = lngRed
                [Percent9_Label].ForeColor = lngRed
                [Percent9].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(9) Then
                lbl10.ForeColor = lngRed
                [Diff10_Label].ForeColor = lngRed
                [Diff10].ForeColor = lngRed
                [Percent10_Label].ForeColor = lngRed
                [Percent10].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(10) Then
                lbl11.ForeColor = lngRed
                [Diff11_Label].ForeColor = lngRed
                [Diff11].ForeColor = lngRed
                [Percent11_Label].ForeColor = lngRed
```

```
                [Percent11].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(11) Then
                lbl12.ForeColor = lngRed
                [Diff12_Label].ForeColor = lngRed
                [Diff12].ForeColor = lngRed
                [Percent12_Label].ForeColor = lngRed
                [Percent12].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(12) Then
                lbl13.ForeColor = lngRed
                [Diff13_Label].ForeColor = lngRed
                [Diff13].ForeColor = lngRed
                [Percent13_Label].ForeColor = lngRed
                [Percent13].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(13) Then
                lbl14.ForeColor = lngRed
                [Diff14_Label].ForeColor = lngRed
                [Diff14].ForeColor = lngRed
                [Percent14_Label].ForeColor = lngRed
                [Percent14].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(14) Then
                lbl15.ForeColor = lngRed
                [Diff15_Label].ForeColor = lngRed
                [Diff15].ForeColor = lngRed
                [Percent15_Label].ForeColor = lngRed
                [Percent15].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(15) Then
                lbl16.ForeColor = lngRed
                [Diff16_Label].ForeColor = lngRed
                [Diff16].ForeColor = lngRed
                [Percent16_Label].ForeColor = lngRed
                [Percent16].ForeColor = lngRed
        End If
End Sub

Option Compare Database

Private Sub cmdClose_Click() 'close report
    DoCmd.Close
End Sub

Private Sub cmdPrint_Click() 'set to landscape and print
    Me.Printer.Orientation = acPRORLandscape
    DoCmd.PrintOut
End Sub

Private Sub Report_Open(Cancel As Integer) 'sets header to proper vessel data
    Dim IMO As Long
    IMO = Form_DetailReports.cmbIMO.Value
    lblIMO.Caption = "IMO Number: " & IMO
    lblname.Caption = "Vessel Name: " & DLookup("[Vessel Name]", "[VesselList]", "[IMO Number]=" & IMO)
    lblComp.Caption = "Shipping Company: " & DLookup("[Shipping Company]", "[VesselList]", "[IMO Number]="
& IMO)
    lblType.Caption = "Vessel Type: " & DLookup("[Vessel Type]", "[VesselList]", "[IMO Number]=" & IMO)
End Sub

Option Compare Database

Private Sub Report_Open(Cancel As Integer)
    Dim IMO As Long
    Dim lngRed As Long
    lngRed = RGB(255, 0, 0)
    IMO = Form_DetailReports.cmbIMO.Value
    'sets labels on report to proper tank names
```

```
    lbl1.Caption = Nz(DLookup("[Tank01 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl2.Caption = Nz(DLookup("[Tank02 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl3.Caption = Nz(DLookup("[Tank03 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl4.Caption = Nz(DLookup("[Tank04 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl5.Caption = Nz(DLookup("[Tank05 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl6.Caption = Nz(DLookup("[Tank06 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl7.Caption = Nz(DLookup("[Tank07 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl8.Caption = Nz(DLookup("[Tank08 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    'changes text of fields selected by user to red
    If Form_DetailReports.lstTanks.Selected(0) Then
            lbl1.ForeColor = lngRed
            [Diff_Label].ForeColor = lngRed
            [Diff].ForeColor = lngRed
            [Percent_Label].ForeColor = lngRed
            [Percent].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(1) Then
            lbl2.ForeColor = lngRed
            [Diff2_Label].ForeColor = lngRed
            [Diff2].ForeColor = lngRed
            [Percent2_Label].ForeColor = lngRed
            [Percent2].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(2) Then
            lbl3.ForeColor = lngRed
            [Diff3_Label].ForeColor = lngRed
            [Diff3].ForeColor = lngRed
            [Percent3_Label].ForeColor = lngRed
            [Percent3].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(3) Then
            lbl4.ForeColor = lngRed
            [Diff4_Label].ForeColor = lngRed
            [Diff4].ForeColor = lngRed
            [Percent4_Label].ForeColor = lngRed
            [Percent4].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(4) Then
            lbl5.ForeColor = lngRed
            [Diff5_Label].ForeColor = lngRed
            [Diff5].ForeColor = lngRed
            [Percent5_Label].ForeColor = lngRed
            [Percent5].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(5) Then
            lbl6.ForeColor = lngRed
            [Diff6_Label].ForeColor = lngRed
            [Diff6].ForeColor = lngRed
            [Percent6_Label].ForeColor = lngRed
            [Percent6].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(6) Then
            lbl7.ForeColor = lngRed
            [Diff7_Label].ForeColor = lngRed
            [Diff7].ForeColor = lngRed
            [Percent7_Label].ForeColor = lngRed
            [Percent7].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(7) Then
            lbl8.ForeColor = lngRed
            [Diff8_Label].ForeColor = lngRed
            [Diff8].ForeColor = lngRed
            [Percent8_Label].ForeColor = lngRed
            [Percent8].ForeColor = lngRed
        End If
End Sub

Option Compare Database
```

```
Private Sub Report_Open(Cancel As Integer)
    Dim IMO As Long
    Dim lngRed As Long
    lngRed = RGB(255, 0, 0)
    IMO = Form_DetailReports.cmbIMO.Value
    'sets labels on report to proper tank names
    lbl9.Caption = Nz(DLookup("[Tank09 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl10.Caption = Nz(DLookup("[Tank10 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl11.Caption = Nz(DLookup("[Tank11 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl12.Caption = Nz(DLookup("[Tank12 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl13.Caption = Nz(DLookup("[Tank13 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl14.Caption = Nz(DLookup("[Tank14 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl15.Caption = Nz(DLookup("[Tank15 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl16.Caption = Nz(DLookup("[Tank16 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    'hides if no values
    If lbl9.Caption = "" And lbl10.Caption = "" And lbl11.Caption = "" And lbl12.Caption = "" And
lbl13.Caption = "" And lbl14.Caption = "" And lbl15.Caption = "" And lbl16.Caption = "" Then
        Me.Visible = False
        Exit Sub
    Else
        Me.Visible = True
    End If
    'changes text of fields selected by user to red
     If Form_DetailReports.lstTanks.Selected(8) Then
                lbl9.ForeColor = lngRed
                [Diff9_Label].ForeColor = lngRed
                [Diff9].ForeColor = lngRed
                [Percent9_Label].ForeColor = lngRed
                [Percent9].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(9) Then
                lbl10.ForeColor = lngRed
                [Diff10_Label].ForeColor = lngRed
                [Diff10].ForeColor = lngRed
                [Percent10_Label].ForeColor = lngRed
                [Percent10].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(10) Then
                lbl11.ForeColor = lngRed
                [Diff11_Label].ForeColor = lngRed
                [Diff11].ForeColor = lngRed
                [Percent11_Label].ForeColor = lngRed
                [Percent11].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(11) Then
                lbl12.ForeColor = lngRed
                [Diff12_Label].ForeColor = lngRed
                [Diff12].ForeColor = lngRed
                [Percent12_Label].ForeColor = lngRed
                [Percent12].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(12) Then
                lbl13.ForeColor = lngRed
                [Diff13_Label].ForeColor = lngRed
                [Diff13].ForeColor = lngRed
                [Percent13_Label].ForeColor = lngRed
                [Percent13].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(13) Then
                lbl14.ForeColor = lngRed
                [Diff14_Label].ForeColor = lngRed
                [Diff14].ForeColor = lngRed
                [Percent14_Label].ForeColor = lngRed
                [Percent14].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(14) Then
                lbl15.ForeColor = lngRed
```

```
                [Diff15_Label].ForeColor = lngRed
                [Diff15].ForeColor = lngRed
                [Percent15_Label].ForeColor = lngRed
                [Percent15].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(15) Then
                lbl16.ForeColor = lngRed
                [Diff16_Label].ForeColor = lngRed
                [Diff16].ForeColor = lngRed
                [Percent16_Label].ForeColor = lngRed
                [Percent16].ForeColor = lngRed
        End If
End Sub

Option Compare Database

Private Sub Report_Open(Cancel As Integer)
    Dim IMO As Long
    Dim lngRed As Long
    lngRed = RGB(255, 0, 0)
    IMO = Form_DetailReports.cmbIMO.Value
    'sets labels to proper tank names
    lbl1.Caption = Nz(DLookup("[Tank01 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl2.Caption = Nz(DLookup("[Tank02 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl3.Caption = Nz(DLookup("[Tank03 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl4.Caption = Nz(DLookup("[Tank04 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl5.Caption = Nz(DLookup("[Tank05 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl6.Caption = Nz(DLookup("[Tank06 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl7.Caption = Nz(DLookup("[Tank07 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl8.Caption = Nz(DLookup("[Tank08 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    'changes text of fields selected by user to red
    If Form_DetailReports.lstTanks.Selected(0) Then
                lbl1.ForeColor = lngRed
                [Diff_Label].ForeColor = lngRed
                [Diff].ForeColor = lngRed
                [Percent_Label].ForeColor = lngRed
                [Percent].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(1) Then
                lbl2.ForeColor = lngRed
                [Diff2_Label].ForeColor = lngRed
                [Diff2].ForeColor = lngRed
                [Percent2_Label].ForeColor = lngRed
                [Percent2].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(2) Then
                lbl3.ForeColor = lngRed
                [Diff3_Label].ForeColor = lngRed
                [Diff3].ForeColor = lngRed
                [Percent3_Label].ForeColor = lngRed
                [Percent3].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(3) Then
                lbl4.ForeColor = lngRed
                [Diff4_Label].ForeColor = lngRed
                [Diff4].ForeColor = lngRed
                [Percent4_Label].ForeColor = lngRed
                [Percent4].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(4) Then
                lbl5.ForeColor = lngRed
                [Diff5_Label].ForeColor = lngRed
                [Diff5].ForeColor = lngRed
                [Percent5_Label].ForeColor = lngRed
                [Percent5].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(5) Then
                lbl6.ForeColor = lngRed
```

```
                [Diff6_Label].ForeColor = lngRed
                [Diff6].ForeColor = lngRed
                [Percent6_Label].ForeColor = lngRed
                [Percent6].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(6) Then
                lbl7.ForeColor = lngRed
                [Diff7_Label].ForeColor = lngRed
                [Diff7].ForeColor = lngRed
                [Percent7_Label].ForeColor = lngRed
                [Percent7].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(7) Then
                lbl8.ForeColor = lngRed
                [Diff8_Label].ForeColor = lngRed
                [Diff8].ForeColor = lngRed
                [Percent8_Label].ForeColor = lngRed
                [Percent8].ForeColor = lngRed
        End If
End Sub

Option Compare Database

Private Sub Report_Open(Cancel As Integer)
    Dim IMO As Long
    Dim lngRed As Long
    lngRed = RGB(255, 0, 0)
    IMO = Form_DetailReports.cmbIMO.Value
    'sets labels to proper tank names
    lbl9.Caption = Nz(DLookup("[Tank09 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl10.Caption = Nz(DLookup("[Tank10 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl11.Caption = Nz(DLookup("[Tank11 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl12.Caption = Nz(DLookup("[Tank12 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl13.Caption = Nz(DLookup("[Tank13 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl14.Caption = Nz(DLookup("[Tank14 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl15.Caption = Nz(DLookup("[Tank15 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    lbl16.Caption = Nz(DLookup("[Tank16 Desc]", "[VesselTanks]", "[IMO Number]=" & IMO), "")
    'hides if vessel doesnt have any of these tanks
    If lbl9.Caption = "" And lbl10.Caption = "" And lbl11.Caption = "" And lbl12.Caption = "" And
lbl13.Caption = "" And lbl14.Caption = "" And lbl15.Caption = "" And lbl16.Caption = "" Then
        Me.Visible = False
        Exit Sub
    Else
        Me.Visible = True
    End If
    'changes text of fields selected by user to red
     If Form_DetailReports.lstTanks.Selected(8) Then
                lbl9.ForeColor = lngRed
                [Diff9_Label].ForeColor = lngRed
                [Diff9].ForeColor = lngRed
                [Percent9_Label].ForeColor = lngRed
                [Percent9].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(9) Then
                lbl10.ForeColor = lngRed
                [Diff10_Label].ForeColor = lngRed
                [Diff10].ForeColor = lngRed
                [Percent10_Label].ForeColor = lngRed
                [Percent10].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(10) Then
                lbl11.ForeColor = lngRed
                [Diff11_Label].ForeColor = lngRed
                [Diff11].ForeColor = lngRed
                [Percent11_Label].ForeColor = lngRed
                [Percent11].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(11) Then
```

```
                lbl12.ForeColor = lngRed
                [Diff12_Label].ForeColor = lngRed
                [Diff12].ForeColor = lngRed
                [Percent12_Label].ForeColor = lngRed
                [Percent12].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(12) Then
                lbl13.ForeColor = lngRed
                [Diff13_Label].ForeColor = lngRed
                [Diff13].ForeColor = lngRed
                [Percent13_Label].ForeColor = lngRed
                [Percent13].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(13) Then
                lbl14.ForeColor = lngRed
                [Diff14_Label].ForeColor = lngRed
                [Diff14].ForeColor = lngRed
                [Percent14_Label].ForeColor = lngRed
                [Percent14].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(14) Then
                lbl15.ForeColor = lngRed
                [Diff15_Label].ForeColor = lngRed
                [Diff15].ForeColor = lngRed
                [Percent15_Label].ForeColor = lngRed
                [Percent15].ForeColor = lngRed
        End If
        If Form_DetailReports.lstTanks.Selected(15) Then
                lbl16.ForeColor = lngRed
                [Diff16_Label].ForeColor = lngRed
                [Diff16].ForeColor = lngRed
                [Percent16_Label].ForeColor = lngRed
                [Percent16].ForeColor = lngRed
        End If
End Sub

Option Compare Database

Private Sub cmdClose_Click() 'close report
    DoCmd.Close
End Sub

Private Sub cmdPrint_Click() ' set printing to landscape and prints report
    Me.Printer.Orientation = acPRORLandscape
    DoCmd.PrintOut
End Sub

Private Sub Report_Open(Cancel As Integer) 'sets header to proper vessel information
    Dim IMO As Long
    IMO = Form_DetailReports.cmbIMO.Value
    lblIMO.Caption = "IMO Number: " & IMO
    lblname.Caption = "Vessel Name: " & DLookup("[Vessel Name]", "[VesselList]", "[IMO Number]=" & IMO)
    lblComp.Caption = "Shipping Company: " & DLookup("[Shipping Company]", "[VesselList]", "[IMO Number]="
& IMO)
    lblType.Caption = "Vessel Type: " & DLookup("[Vessel Type]", "[VesselList]", "[IMO Number]=" & IMO)
End Sub

Option Compare Database

Private Sub cmdClose_Click() 'close report
    DoCmd.Close
End Sub

Private Sub cmdPrint_Click() 'set to landscape and print report
    Me.Printer.Orientation = acPRORLandscape
    DoCmd.PrintOut
End Sub
```

```
Private Sub Report_Open(Cancel As Integer) 'set header to proper vessel information
    Dim IMO As Long
    IMO = Form_DetailReports.cmbIMO.Value
    lblIMO.Caption = "IMO Number: " & IMO
    lblname.Caption = "Vessel Name: " & DLookup("[Vessel Name]", "[VesselList]", "[IMO Number]=" & IMO)
    lblComp.Caption = "Shipping Company: " & DLookup("[Shipping Company]", "[VesselList]", "[IMO Number]="
& IMO)
    lblType.Caption = "Vessel Type: " & DLookup("[Vessel Type]", "[VesselList]", "[IMO Number]=" & IMO)
End Sub
```

SQL Queries:

Capacity Query:

```
SELECT VesselTanks.[IMO Number], VesselTanks.[Tank01 Cap], VesselTanks.[Tank02 Cap], VesselTanks.[Tank03
Cap], VesselTanks.[Tank04 Cap], VesselTanks.[Tank05 Cap], VesselTanks.[Tank06 Cap], VesselTanks.[Tank07
Cap], VesselTanks.[Tank08 Cap], VesselTanks.[Tank09 Cap], VesselTanks.[Tank10 Cap], VesselTanks.[Tank11
Cap], VesselTanks.[Tank12 Cap], VesselTanks.[Tank13 Cap], VesselTanks.[Tank14 Cap], VesselTanks.[Tank15
Cap], VesselTanks.[Tank16 Cap]
FROM VesselTanks
WHERE (((VesselTanks.[IMO Number])=[Forms].[DetailReports].[cmbIMO]));
```

Delete SWOMS:

```
DELETE [SWOMS Data].[IMO Number], [SWOMS Data].[Date (dd/mm/yyyy)]
FROM [SWOMS Data]
WHERE ((([SWOMS Data].[IMO Number])=[Forms].[DeleteRecords].[cmbIMO]) AND (([SWOMS Data].[Date
(dd/mm/yyyy)]) Between [Forms].[DeleteRecords].[txtStart] And [Forms].[DeleteRecords].[txtEnd]));
```

Sorted Query:

```
SELECT [SWOMS Analysis].[Date (dd/mm/yyyy)], [SWOMS Analysis].[Time (UTC; 00:00:00)], [SWOMS
Analysis].[Bilge Pump Runs], [SWOMS Analysis].[Bilge Pump Run Time], [SWOMS Analysis].[OWS Runs], [SWOMS
Analysis].[OWS Run Time], [SWOMS Analysis].[Overboard Opened], [SWOMS Analysis].[Overboard Open Time]
FROM [SWOMS Analysis]
WHERE ((([SWOMS Analysis].[Bilge Pump Runs])<>"0")) OR ((([SWOMS Analysis].[Bilge Pump Run Time])<>"0"))
OR ((([SWOMS Analysis].[OWS Runs])<>"0")) OR ((([SWOMS Analysis].[OWS Run Time])<>"0")) OR ((([SWOMS
Analysis].[Overboard Opened])<>"0")) OR ((([SWOMS Analysis].[Overboard Open Time])<>"0"))
ORDER BY [SWOMS Analysis].[Date (dd/mm/yyyy)], [SWOMS Analysis].[Time (UTC; 00:00:00)];
```

Sorted Query 2:

```
SELECT [SWOMS Analysis].[Date (dd/mm/yyyy)], [SWOMS Analysis].[Time (UTC; 00:00:00)], [SWOMS
Analysis].[PPM Alarms], [SWOMS Analysis].[Average PPM Overboard], [SWOMS Analysis].[OCM Fresh Water Valve
Opened], [SWOMS Analysis].[Oil Purge Valve Opened], [SWOMS Analysis].[Oil Purge Valve Open Time], [SWOMS
Analysis].[Incinerator Runs], [SWOMS Analysis].[Incinerator Run Time]
FROM [SWOMS Analysis]
WHERE ((([SWOMS Analysis].[PPM Alarms])<>"0")) OR ((([SWOMS Analysis].[Average PPM Overboard])<>"0")) OR
((([SWOMS Analysis].[OCM Fresh Water Valve Opened])<>"0")) OR ((([SWOMS Analysis].[Oil Purge Valve
Opened])<>"0")) OR ((([SWOMS Analysis].[Oil Purge Valve Open Time])<>"0")) OR ((([SWOMS
Analysis].[Incinerator Runs])<>"0")) OR ((([SWOMS Analysis].[Incinerator Run Time])<>"0"));
```

SWOMS % CAP:

```
SELECT [SWOMS Analysis].[Date (dd/mm/yyyy)], [SWOMS Analysis].[Time (UTC; 00:00:00)], IIf(Nz([Capacity
Query].[Tank01 Cap])=0,Null,((Nz([SWOMS Analysis].[Tank01 VOL])/Nz([Capacity Query].[Tank01 Cap])))) AS
[Tank 1], IIf(Nz([Capacity Query].[Tank02 Cap])=0,Null,((Nz([SWOMS Analysis].[Tank02 VOL])/Nz([Capacity
Query].[Tank02 Cap])))) AS [Tank 2], IIf(Nz([Capacity Query].[Tank03 Cap])=0,Null,((Nz([SWOMS
Analysis].[Tank03 VOL])/Nz([Capacity Query].[Tank03 Cap])))) AS [Tank 3], IIf(Nz([Capacity Query].[Tank04
Cap])=0,Null,((Nz([SWOMS Analysis].[Tank04 VOL])/Nz([Capacity Query].[Tank04 Cap])))) AS [Tank 4],
IIf(Nz([Capacity Query].[Tank05 Cap])=0,Null,((Nz([SWOMS Analysis].[Tank05 VOL])/Nz([Capacity
Query].[Tank05 Cap])))) AS [Tank 5], IIf(Nz([Capacity Query].[Tank06 Cap])=0,Null,((Nz([SWOMS
Analysis].[Tank06 VOL])/Nz([Capacity Query].[Tank06 Cap])))) AS [Tank 6], IIf(Nz([Capacity Query].[Tank07
Cap])=0,Null,((Nz([SWOMS Analysis].[Tank07 VOL])/Nz([Capacity Query].[Tank07 Cap])))) AS [Tank 7],
IIf(Nz([Capacity Query].[Tank08 Cap])=0,Null,((Nz([SWOMS Analysis].[Tank08 VOL])/Nz([Capacity
Query].[Tank08 Cap])))) AS [Tank 8], IIf(Nz([Capacity Query].[Tank09 Cap])=0,Null,((Nz([SWOMS
Analysis].[Tank09 VOL])/Nz([Capacity Query].[Tank09 Cap])))) AS [Tank 9], IIf(Nz([Capacity Query].[Tank10
Cap])=0,Null,((Nz([SWOMS Analysis].[Tank10 VOL])/Nz([Capacity Query].[Tank10 Cap])))) AS [Tank 10],
IIf(Nz([Capacity Query].[Tank11 Cap])=0,Null,((Nz([SWOMS Analysis].[Tank11 VOL])/Nz([Capacity
```

```
Query].[Tank11 Cap])))) AS [Tank 11], IIf(Nz([Capacity Query].[Tank12 Cap])=0,Null,((Nz([SWOMS Analysis].[Tank12 VOL])/Nz([Capacity Query].[Tank12 Cap])))) AS [Tank 12], IIf(Nz([Capacity Query].[Tank13 Cap])=0,Null,((Nz([SWOMS Analysis].[Tank13 VOL])/Nz([Capacity Query].[Tank13 Cap])))) AS [Tank 13], IIf(Nz([Capacity Query].[Tank14 Cap])=0,Null,((Nz([SWOMS Analysis].[Tank14 VOL])/Nz([Capacity Query].[Tank14 Cap])))) AS [Tank 14], IIf(Nz([Capacity Query].[Tank15 Cap])=0,Null,((Nz([SWOMS Analysis].[Tank15 VOL])/Nz([Capacity Query].[Tank15 Cap])))) AS [Tank 15], IIf(Nz([Capacity Query].[Tank16 Cap])=0,Null,((Nz([SWOMS Analysis].[Tank16 VOL])/Nz([Capacity Query].[Tank16 Cap])))) AS [Tank 16], [SWOMS Analysis].[Tank01 VOL], [SWOMS Analysis].[Tank02 VOL], [SWOMS Analysis].[Tank03 VOL], [SWOMS Analysis].[Tank04 VOL], [SWOMS Analysis].[Tank05 VOL], [SWOMS Analysis].[Tank06 VOL], [SWOMS Analysis].[Tank07 VOL], [SWOMS Analysis].[Tank08 VOL], [SWOMS Analysis].[Tank09 VOL], [SWOMS Analysis].[Tank10 VOL], [SWOMS Analysis].[Tank11 VOL], [SWOMS Analysis].[Tank12 VOL], [SWOMS Analysis].[Tank13 VOL], [SWOMS Analysis].[Tank14 VOL], [SWOMS Analysis].[Tank15 VOL], [SWOMS Analysis].[Tank16 VOL]
FROM [SWOMS Analysis], [Capacity Query]
ORDER BY [SWOMS Analysis].[Date (dd/mm/yyyy)];
```

SWOMS Analysis:

```
SELECT [SWOMS Data].[IMO Number], [SWOMS Data].[Date (dd/mm/yyyy)], [SWOMS Data].[Time (UTC; 00:00:00)], [SWOMS Data].Latitude, [SWOMS Data].Longitude, [SWOMS Data].[Bilge Pump Runs], [SWOMS Data].[Bilge Pump Run Time], [SWOMS Data].[OWS Runs], [SWOMS Data].[OWS Run Time], [SWOMS Data].[Overboard Opened], [SWOMS Data].[Overboard Open Time], [SWOMS Data].[PPM Alarms], [SWOMS Data].[Average PPM Overboard], [SWOMS Data].[OCM Fresh Water Valve Opened], [SWOMS Data].[Oil Purge Valve Opened], [SWOMS Data].[Oil Purge Valve Open Time], [SWOMS Data].[Incinerator Runs], [SWOMS Data].[Incinerator Run Time], [SWOMS Data].[Tank01 VOL], [SWOMS Data].[Tank02 VOL], [SWOMS Data].[Tank03 VOL], [SWOMS Data].[Tank04 VOL], [SWOMS Data].[Tank05 VOL], [SWOMS Data].[Tank06 VOL], [SWOMS Data].[Tank07 VOL], [SWOMS Data].[Tank08 VOL], [SWOMS Data].[Tank09 VOL], [SWOMS Data].[Tank10 VOL], [SWOMS Data].[Tank11 VOL], [SWOMS Data].[Tank12 VOL], [SWOMS Data].[Tank13 VOL], [SWOMS Data].[Tank14 VOL], [SWOMS Data].[Tank15 VOL], [SWOMS Data].[Tank16 VOL]
FROM [SWOMS Data]
WHERE ((([SWOMS Data].[IMO Number])=[Forms].[DetailReports].[cmbIMO]) AND (([SWOMS Data].[Date (dd/mm/yyyy)]) Between [Forms].[DetailReports].[txtStart] And [Forms].[DetailReports].[txtEnd]))
ORDER BY [SWOMS Data].[Date (dd/mm/yyyy)], [SWOMS Data].[Time (UTC; 00:00:00)];
```

SWOMS Change:

```
SELECT [SWOMS % CAP].[Date (dd/mm/yyyy)], [SWOMS % CAP].[Time (UTC; 00:00:00)], ([SWOMS % CAP].[Tank 1]-(SELECT TOP 1 [Tank 1] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([SWOMS % CAP].[Tank 1]-Q.[Tank 1])<-.3))) AS [Percent], ([SWOMS % CAP].[Tank01 VOL]-(SELECT TOP 1 [Tank01 VOL] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND ([SWOMS % CAP].[Tank 1]-(SELECT TOP 1 [Tank 1] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([SWOMS % CAP].[Tank 1]-Q.[Tank 1])<-.3))))) AS Diff, ([SWOMS % CAP].[Tank 2]-(SELECT TOP 1 [Tank 2] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([SWOMS % CAP].[Tank 2]-Q.[Tank 2])<-.3))) AS Percent2, ([SWOMS % CAP].[Tank02 VOL]-(SELECT TOP 1 [Tank02 VOL] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND ([SWOMS % CAP].[Tank 2]-(SELECT TOP 1 [Tank 2] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([SWOMS % CAP].[Tank 2]-Q.[Tank 2])<-.3))))) AS Diff2, ([SWOMS % CAP].[Tank 3]-(SELECT TOP 1 [Tank 3] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([SWOMS % CAP].[Tank 3]-Q.[Tank 3])<-.3))) AS Percent3, ([SWOMS % CAP].[Tank03 VOL]-(SELECT TOP 1 [Tank03 VOL] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND ([SWOMS % CAP].[Tank 3]-(SELECT TOP 1 [Tank 3] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([SWOMS % CAP].[Tank 3]-Q.[Tank 3])<-.3))))) AS Diff3, ([SWOMS % CAP].[Tank 4]-(SELECT TOP 1 [Tank 4] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([SWOMS % CAP].[Tank 4]-Q.[Tank 4])<-.3))) AS Percent4, ([SWOMS % CAP].[Tank04 VOL]-(SELECT TOP 1 [Tank04 VOL] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND ([SWOMS % CAP].[Tank 4]-(SELECT TOP 1 [Tank 4] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([SWOMS % CAP].[Tank 4]-Q.[Tank 4])<-.3))))) AS Diff4, ([SWOMS % CAP].[Tank 5]-(SELECT TOP 1 [Tank 5] FROM [SWOMS % CAP] AS Q WHERE
```

```
(DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date
(dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([SWOMS % CAP].[Tank 5]-Q.[Tank 5])<-.3))) AS Percent5, ([SWOMS
% CAP].[Tank05 VOL]-(SELECT TOP 1 [Tank05 VOL] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS %
CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date
(dd/mm/yyyy)]) AND ([SWOMS % CAP].[Tank 5]-(SELECT TOP 1 [Tank 5] FROM [SWOMS % CAP] AS Q WHERE
(DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date
(dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([SWOMS % CAP].[Tank 5]-Q.[Tank 5])<-.3))))) AS Diff5, ([SWOMS
% CAP].[Tank 6]-(SELECT TOP 1 [Tank 6] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date
(dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)])
AND (([SWOMS % CAP].[Tank 6]-Q.[Tank 6])<-.3))) AS Percent6, ([SWOMS % CAP].[Tank06 VOL]-(SELECT TOP 1
[Tank06 VOL] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date
(dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND ([SWOMS %
CAP].[Tank 6]-(SELECT TOP 1 [Tank 6] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date
(dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)])
AND (([SWOMS % CAP].[Tank 6]-Q.[Tank 6])<-.3))))) AS Diff6, ([SWOMS % CAP].[Tank 7]-(SELECT TOP 1 [Tank 7]
FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -
1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([SWOMS % CAP].[Tank 7]-Q.[Tank
7])<-.3))) AS Percent7, ([SWOMS % CAP].[Tank07 VOL]-(SELECT TOP 1 [Tank07 VOL] FROM [SWOMS % CAP] AS Q
WHERE (DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS %
CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND ([SWOMS % CAP].[Tank 7]-(SELECT TOP 1 [Tank 7] FROM
[SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1)
AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([SWOMS % CAP].[Tank 7]-Q.[Tank 7])<-
.3))))) AS Diff7, ([SWOMS % CAP].[Tank 8]-(SELECT TOP 1 [Tank 8] FROM [SWOMS % CAP] AS Q WHERE
(DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date
(dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([SWOMS % CAP].[Tank 8]-Q.[Tank 8])<-.3))) AS Percent8, ([SWOMS
% CAP].[Tank08 VOL]-(SELECT TOP 1 [Tank08 VOL] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS %
CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date
(dd/mm/yyyy)]) AND ([SWOMS % CAP].[Tank 8]-(SELECT TOP 1 [Tank 8] FROM [SWOMS % CAP] AS Q WHERE
(DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date
(dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([SWOMS % CAP].[Tank 8]-Q.[Tank 8])<-.3))))) AS Diff8, ([SWOMS
% CAP].[Tank 9]-(SELECT TOP 1 [Tank 9] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date
(dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)])
AND (([SWOMS % CAP].[Tank 9]-Q.[Tank 9])<-.3))) AS Percent9, ([SWOMS % CAP].[Tank09 VOL]-(SELECT TOP 1
[Tank09 VOL] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date
(dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND ([SWOMS %
CAP].[Tank 9]-(SELECT TOP 1 [Tank 9] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date
(dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)])
AND (([SWOMS % CAP].[Tank 9]-Q.[Tank 9])<-.3))))) AS Diff9, ([SWOMS % CAP].[Tank 10]-(SELECT TOP 1 [Tank
10] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)])
= -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([SWOMS % CAP].[Tank 10]-Q.[Tank
10])<-.3))) AS Percent10, ([SWOMS % CAP].[Tank10 VOL]-(SELECT TOP 1 [Tank10 VOL] FROM [SWOMS % CAP] AS Q
WHERE (DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS %
CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND ([SWOMS % CAP].[Tank 10]-(SELECT TOP 1 [Tank 10] FROM
[SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1)
AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([SWOMS % CAP].[Tank 10]-Q.[Tank
10])<-.3))))) AS Diff10, ([SWOMS % CAP].[Tank 11]-(SELECT TOP 1 [Tank 11] FROM [SWOMS % CAP] AS Q WHERE
(DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date
(dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([SWOMS % CAP].[Tank 11]-Q.[Tank 11])<-.3))) AS Percent11,
([SWOMS % CAP].[Tank11 VOL]-(SELECT TOP 1 [Tank11 VOL] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS
% CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date
(dd/mm/yyyy)]) AND ([SWOMS % CAP].[Tank 11]-(SELECT TOP 1 [Tank 11] FROM [SWOMS % CAP] AS Q WHERE
(DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date
(dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([SWOMS % CAP].[Tank 11]-Q.[Tank 11])<-.3))))) AS Diff11,
([SWOMS % CAP].[Tank 12]-(SELECT TOP 1 [Tank 12] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS %
CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date
(dd/mm/yyyy)]) AND (([SWOMS % CAP].[Tank 12]-Q.[Tank 12])<-.3))) AS Percent12, ([SWOMS % CAP].[Tank12
VOL]-(SELECT TOP 1 [Tank12 VOL] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date
(dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)])
AND ([SWOMS % CAP].[Tank 12]-(SELECT TOP 1 [Tank 12] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS %
CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date
(dd/mm/yyyy)]) AND (([SWOMS % CAP].[Tank 12]-Q.[Tank 12])<-.3))))) AS Diff12, ([SWOMS % CAP].[Tank 13]-
(SELECT TOP 1 [Tank 13] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)],
Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([SWOMS %
CAP].[Tank 13]-Q.[Tank 13])<-.3))) AS Percent13, ([SWOMS % CAP].[Tank13 VOL]-(SELECT TOP 1 [Tank13 VOL]
FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -
1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND ([SWOMS % CAP].[Tank 13]-(SELECT TOP
1 [Tank 13] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date
(dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([SWOMS %
```

```
CAP].[Tank 13]-Q.[Tank 13])<-.3))))) AS Diff13, ([SWOMS % CAP].[Tank 14]-(SELECT TOP 1 [Tank 14] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([SWOMS % CAP].[Tank 14]-Q.[Tank 14])<-.3))) AS Percent14, ([SWOMS % CAP].[Tank14 VOL]-(SELECT TOP 1 [Tank14 VOL] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND ([SWOMS % CAP].[Tank 14]-(SELECT TOP 1 [Tank 14] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([SWOMS % CAP].[Tank 14]-Q.[Tank 14])<-.3))))) AS Diff14, ([SWOMS % CAP].[Tank 15]-(SELECT TOP 1 [Tank 15] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([SWOMS % CAP].[Tank 15]-Q.[Tank 15])<-.3))) AS Percent15, ([SWOMS % CAP].[Tank15 VOL]-(SELECT TOP 1 [Tank15 VOL] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND ([SWOMS % CAP].[Tank 15]-(SELECT TOP 1 [Tank 15] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([SWOMS % CAP].[Tank 15]-Q.[Tank 15])<-.3))))) AS Diff15, ([SWOMS % CAP].[Tank 16]-(SELECT TOP 1 [Tank 16] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([SWOMS % CAP].[Tank 16]-Q.[Tank 16])<-.3))) AS Percent16, ([SWOMS % CAP].[Tank16 VOL]-(SELECT TOP 1 [Tank16 VOL] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND ([SWOMS % CAP].[Tank 16]-(SELECT TOP 1 [Tank 16] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([SWOMS % CAP].[Tank 16]-Q.[Tank 16])<-.3))))) AS Diff16
FROM [SWOMS % CAP];
```

SWOMS Data Query:

```
SELECT [SWOMS Data].[IMO Number], [SWOMS Data].[Date (dd/mm/yyyy)], [SWOMS Data].[Time (UTC; 00:00:00)], [SWOMS Data].Latitude, [SWOMS Data].Longitude, [SWOMS Data].[Bilge Pump Runs], [SWOMS Data].[Bilge Pump Run Time], [SWOMS Data].[OWS Runs], [SWOMS Data].[OWS Run Time], [SWOMS Data].[Overboard Opened], [SWOMS Data].[Overboard Open Time], [SWOMS Data].[PPM Alarms], [SWOMS Data].[Average PPM Overboard], [SWOMS Data].[OCM Fresh Water Valve Opened], [SWOMS Data].[Oil Purge Valve Opened], [SWOMS Data].[Oil Purge Valve Open Time], [SWOMS Data].[Incinerator Runs], [SWOMS Data].[Incinerator Run Time], [SWOMS Data].[Tank01 LVL], [SWOMS Data].[Tank01 VOL], [SWOMS Data].[Tank02 LVL], [SWOMS Data].[Tank02 VOL], [SWOMS Data].[Tank03 LVL], [SWOMS Data].[Tank03 VOL], [SWOMS Data].[Tank04 LVL], [SWOMS Data].[Tank04 VOL], [SWOMS Data].[Tank05 LVL], [SWOMS Data].[Tank05 VOL], [SWOMS Data].[Tank06 LVL], [SWOMS Data].[Tank06 VOL], [SWOMS Data].[Tank07 LVL], [SWOMS Data].[Tank07 VOL], [SWOMS Data].[Tank08 LVL], [SWOMS Data].[Tank08 VOL], [SWOMS Data].[Tank09 LVL], [SWOMS Data].[Tank09 VOL], [SWOMS Data].[Tank10 LVL], [SWOMS Data].[Tank10 VOL], [SWOMS Data].[Tank11 LVL], [SWOMS Data].[Tank11 VOL], [SWOMS Data].[Tank12 LVL], [SWOMS Data].[Tank12 VOL], [SWOMS Data].[Tank13 LVL], [SWOMS Data].[Tank13 VOL], [SWOMS Data].[Tank14 LVL], [SWOMS Data].[Tank14 VOL], [SWOMS Data].[Tank15 LVL], [SWOMS Data].[Tank15 VOL], [SWOMS Data].[Tank16 LVL], [SWOMS Data].[Tank16 VOL], [SWOMS Data].[IMO Number]
FROM [SWOMS Data]
WHERE ((([SWOMS Data].[Date (dd/mm/yyyy)]) Between [Forms].[SWOMSData].[txtStart] And [Forms].[SWOMSData].[txtEnd]) AND (([SWOMS Data].[IMO Number])=[Forms].[SWOMSData].[cmbIMO]))
ORDER BY [SWOMS Data].[Date (dd/mm/yyyy)];
```

SWOMS VS Sounding:

```
SELECT [SWOMS Analysis].[Date (dd/mm/yyyy)], [SWOMS Analysis].[Time (UTC; 00:00:00)], [SWOMS Analysis].[Tank01 VOL]-(SELECT TOP 1 [Tank01 VOL] FROM [Sounding Analysis] WHERE DateDiff("d", [SWOMS Analysis].[Date (dd/mm/yyyy)], [Sounding Analysis].[Date (dd/mm/yyyy)]) = 0 AND [SWOMS Analysis].[Time (UTC; 00:00:00)] = [Sounding Analysis].[Time (UTC; 00:00:00)] AND ((([SWOMS Analysis].[Tank01 VOL]-[Sounding Analysis].[Tank01 VOL])/[SWOMS Analysis].[Tank01 VOL]) >.05 OR (([Sounding Analysis].[Tank01 VOL]-[SWOMS Analysis].[Tank01 VOL])/[Sounding Analysis].[Tank01 VOL]) >.05)) AS Diff, IIf([Diff]>0,([Tank01 VOL]-([Tank01 VOL]-[Diff]))/[Tank01 VOL],(([Tank01 VOL]+(-1*[Diff]))-[Tank01 VOL])/([Tank01 VOL]+(-1*[Diff]))) AS [Percent], [SWOMS Analysis].[Tank02 VOL]-(SELECT TOP 1 [Tank02 VOL] FROM [Sounding Analysis] WHERE DateDiff("d", [SWOMS Analysis].[Date (dd/mm/yyyy)], [Sounding Analysis].[Date (dd/mm/yyyy)]) = 0 AND [SWOMS Analysis].[Time (UTC; 00:00:00)] = [Sounding Analysis].[Time (UTC; 00:00:00)] AND ((([SWOMS Analysis].[Tank02 VOL]-[Sounding Analysis].[Tank02 VOL])/[SWOMS Analysis].[Tank02 VOL]) >.05 OR (([Sounding Analysis].[Tank02 VOL]-[SWOMS Analysis].[Tank02 VOL])/[Sounding Analysis].[Tank02 VOL]) >.05)) AS Diff2, IIf([Diff2]>0,([Tank02 VOL]-([Tank02 VOL]-[Diff2]))/[Tank02 VOL],(([Tank02 VOL]+(-1*[Diff2]))-[Tank02 VOL])/([Tank02 VOL]+(-1*[Diff2]))) AS Percent2, [SWOMS Analysis].[Tank03 VOL]-(SELECT TOP 1 [Tank03 VOL] FROM [Sounding Analysis] WHERE DateDiff("d", [SWOMS Analysis].[Date (dd/mm/yyyy)], [Sounding Analysis].[Date (dd/mm/yyyy)]) = 0 AND [SWOMS Analysis].[Time (UTC; 00:00:00)] = [Sounding Analysis].[Time (UTC; 00:00:00)] AND ((([SWOMS Analysis].[Tank03 VOL]-[Sounding Analysis].[Tank03 VOL])/[SWOMS Analysis].[Tank03 VOL]) >.05 OR
```

```
(([Sounding Analysis].[Tank03 VOL]-[SWOMS Analysis].[Tank03 VOL])/[Sounding Analysis].[Tank03 VOL]) >.05)) AS Diff3, IIf([Diff3]>0,([Tank03 VOL]-([Tank03 VOL]-[Diff3]))/[Tank03 VOL],(([Tank03 VOL]+(-1*[Diff3]))-[Tank03 VOL])/([Tank03 VOL]+(-1*[Diff3]))) AS Percent3, [SWOMS Analysis].[Tank04 VOL]-(SELECT TOP 1 [Tank04 VOL] FROM [Sounding Analysis] WHERE DateDiff("d", [SWOMS Analysis].[Date (dd/mm/yyyy)], [Sounding Analysis].[Date (dd/mm/yyyy)]) = 0 AND [SWOMS Analysis].[Time (UTC; 00:00:00)] = [Sounding Analysis].[Time (UTC; 00:00:00)] AND ((([SWOMS Analysis].[Tank04 VOL]-[Sounding Analysis].[Tank04 VOL])/[SWOMS Analysis].[Tank04 VOL]) >.05 OR (([Sounding Analysis].[Tank04 VOL]-[SWOMS Analysis].[Tank04 VOL])/[Sounding Analysis].[Tank04 VOL]) >.05)) AS Diff4, IIf([Diff4]>0,([Tank04 VOL]-([Tank04 VOL]-[Diff4]))/[Tank04 VOL],(([Tank04 VOL]+(-1*[Diff4]))-[Tank04 VOL])/([Tank04 VOL]+(-1*[Diff4]))) AS Percent4, [SWOMS Analysis].[Tank05 VOL]-(SELECT TOP 1 [Tank05 VOL] FROM [Sounding Analysis] WHERE DateDiff("d", [SWOMS Analysis].[Date (dd/mm/yyyy)], [Sounding Analysis].[Date (dd/mm/yyyy)]) = 0 AND [SWOMS Analysis].[Time (UTC; 00:00:00)] = [Sounding Analysis].[Time (UTC; 00:00:00)] AND ((([SWOMS Analysis].[Tank05 VOL]-[Sounding Analysis].[Tank05 VOL])/[SWOMS Analysis].[Tank05 VOL]) >.05 OR (([Sounding Analysis].[Tank05 VOL]-[SWOMS Analysis].[Tank05 VOL])/[Sounding Analysis].[Tank05 VOL]) >.05)) AS Diff5, IIf([Diff5]>0,([Tank05 VOL]-([Tank05 VOL]-[Diff5]))/[Tank05 VOL],(([Tank05 VOL]+(-1*[Diff5]))-[Tank05 VOL])/([Tank05 VOL]+(-1*[Diff5]))) AS Percent5, [SWOMS Analysis].[Tank06 VOL]-(SELECT TOP 1 [Tank06 VOL] FROM [Sounding Analysis] WHERE DateDiff("d", [SWOMS Analysis].[Date (dd/mm/yyyy)], [Sounding Analysis].[Date (dd/mm/yyyy)]) = 0 AND [SWOMS Analysis].[Time (UTC; 00:00:00)] = [Sounding Analysis].[Time (UTC; 00:00:00)] AND ((([SWOMS Analysis].[Tank06 VOL]-[Sounding Analysis].[Tank06 VOL])/[SWOMS Analysis].[Tank06 VOL]) >.05 OR (([Sounding Analysis].[Tank06 VOL]-[SWOMS Analysis].[Tank06 VOL])/[Sounding Analysis].[Tank06 VOL]) >.05)) AS Diff6, IIf([Diff6]>0,([Tank06 VOL]-([Tank06 VOL]-[Diff6]))/[Tank06 VOL],(([Tank06 VOL]+(-1*[Diff6]))-[Tank06 VOL])/([Tank06 VOL]+(-1*[Diff6]))) AS Percent6, [SWOMS Analysis].[Tank07 VOL]-(SELECT TOP 1 [Tank07 VOL] FROM [Sounding Analysis] WHERE DateDiff("d", [SWOMS Analysis].[Date (dd/mm/yyyy)], [Sounding Analysis].[Date (dd/mm/yyyy)]) = 0 AND [SWOMS Analysis].[Time (UTC; 00:00:00)] = [Sounding Analysis].[Time (UTC; 00:00:00)] AND ((([SWOMS Analysis].[Tank07 VOL]-[Sounding Analysis].[Tank07 VOL])/[SWOMS Analysis].[Tank07 VOL]) >.05 OR (([Sounding Analysis].[Tank07 VOL]-[SWOMS Analysis].[Tank07 VOL])/[Sounding Analysis].[Tank07 VOL]) >.05)) AS Diff7, IIf([Diff7]>0,([Tank07 VOL]-([Tank07 VOL]-[Diff7]))/[Tank07 VOL],(([Tank07 VOL]+(-1*[Diff7]))-[Tank07 VOL])/([Tank07 VOL]+(-1*[Diff7]))) AS Percent7, [SWOMS Analysis].[Tank08 VOL]-(SELECT TOP 1 [Tank08 VOL] FROM [Sounding Analysis] WHERE DateDiff("d", [SWOMS Analysis].[Date (dd/mm/yyyy)], [Sounding Analysis].[Date (dd/mm/yyyy)]) = 0 AND [SWOMS Analysis].[Time (UTC; 00:00:00)] = [Sounding Analysis].[Time (UTC; 00:00:00)] AND ((([SWOMS Analysis].[Tank08 VOL]-[Sounding Analysis].[Tank08 VOL])/[SWOMS Analysis].[Tank08 VOL]) >.05 OR (([Sounding Analysis].[Tank08 VOL]-[SWOMS Analysis].[Tank08 VOL])/[Sounding Analysis].[Tank08 VOL]) >.05)) AS Diff8, IIf([Diff8]>0,([Tank08 VOL]-([Tank08 VOL]-[Diff8]))/[Tank08 VOL],(([Tank08 VOL]+(-1*[Diff8]))-[Tank08 VOL])/([Tank08 VOL]+(-1*[Diff8]))) AS Percent8, [SWOMS Analysis].[Tank09 VOL]-(SELECT TOP 1 [Tank09 VOL] FROM [Sounding Analysis] WHERE DateDiff("d", [SWOMS Analysis].[Date (dd/mm/yyyy)], [Sounding Analysis].[Date (dd/mm/yyyy)]) = 0 AND [SWOMS Analysis].[Time (UTC; 00:00:00)] = [Sounding Analysis].[Time (UTC; 00:00:00)] AND ((([SWOMS Analysis].[Tank09 VOL]-[Sounding Analysis].[Tank09 VOL])/[SWOMS Analysis].[Tank09 VOL]) >.05 OR (([Sounding Analysis].[Tank09 VOL]-[SWOMS Analysis].[Tank09 VOL])/[Sounding Analysis].[Tank09 VOL]) >.05)) AS Diff9, IIf([Diff9]>0,([Tank09 VOL]-([Tank09 VOL]-[Diff9]))/[Tank09 VOL],(([Tank09 VOL]+(-1*[Diff9]))-[Tank09 VOL])/([Tank09 VOL]+(-1*[Diff9]))) AS Percent9, [SWOMS Analysis].[Tank10 VOL]-(SELECT TOP 1 [Tank10 VOL] FROM [Sounding Analysis] WHERE DateDiff("d", [SWOMS Analysis].[Date (dd/mm/yyyy)], [Sounding Analysis].[Date (dd/mm/yyyy)]) = 0 AND [SWOMS Analysis].[Time (UTC; 00:00:00)] = [Sounding Analysis].[Time (UTC; 00:00:00)] AND ((([SWOMS Analysis].[Tank10 VOL]-[Sounding Analysis].[Tank10 VOL])/[SWOMS Analysis].[Tank10 VOL]) >.05 OR (([Sounding Analysis].[Tank10 VOL]-[SWOMS Analysis].[Tank10 VOL])/[Sounding Analysis].[Tank10 VOL]) >.05)) AS Diff10, IIf([Diff10]>0,([Tank10 VOL]-([Tank10 VOL]-[Diff10]))/[Tank10 VOL],(([Tank10 VOL]+(-1*[Diff10]))-[Tank10 VOL])/([Tank10 VOL]+(-1*[Diff10]))) AS Percent10, [SWOMS Analysis].[Tank11 VOL]-(SELECT TOP 1 [Tank11 VOL] FROM [Sounding Analysis] WHERE DateDiff("d", [SWOMS Analysis].[Date (dd/mm/yyyy)], [Sounding Analysis].[Date (dd/mm/yyyy)]) = 0 AND [SWOMS Analysis].[Time (UTC; 00:00:00)] = [Sounding Analysis].[Time (UTC; 00:00:00)] AND ((([SWOMS Analysis].[Tank11 VOL]-[Sounding Analysis].[Tank11 VOL])/[SWOMS Analysis].[Tank11 VOL]) >.05 OR (([Sounding Analysis].[Tank11 VOL]-[SWOMS Analysis].[Tank11 VOL])/[Sounding Analysis].[Tank11 VOL]) >.05)) AS Diff11, IIf([Diff11]>0,([Tank11 VOL]-([Tank11 VOL]-[Diff11]))/[Tank11 VOL],(([Tank11 VOL]+(-1*[Diff11]))-[Tank11 VOL])/([Tank11 VOL]+(-1*[Diff11]))) AS Percent11, [SWOMS Analysis].[Tank12 VOL]-(SELECT TOP 1 [Tank12 VOL] FROM [Sounding Analysis] WHERE DateDiff("d", [SWOMS Analysis].[Date (dd/mm/yyyy)], [Sounding Analysis].[Date (dd/mm/yyyy)]) = 0 AND [SWOMS Analysis].[Time (UTC; 00:00:00)] = [Sounding Analysis].[Time (UTC; 00:00:00)] AND ((([SWOMS Analysis].[Tank12 VOL]-[Sounding Analysis].[Tank12 VOL])/[SWOMS Analysis].[Tank12 VOL]) >.05 OR (([Sounding Analysis].[Tank12 VOL]-[SWOMS Analysis].[Tank12 VOL])/[Sounding Analysis].[Tank12 VOL]) >.05)) AS Diff12, IIf([Diff12]>0,([Tank12 VOL]-([Tank12 VOL]-[Diff12]))/[Tank12 VOL],(([Tank12 VOL]+(-1*[Diff12]))-[Tank12 VOL])/([Tank12 VOL]+(-1*[Diff12]))) AS Percent12, [SWOMS Analysis].[Tank13 VOL]-(SELECT TOP 1 [Tank13 VOL] FROM [Sounding Analysis] WHERE DateDiff("d", [SWOMS Analysis].[Date (dd/mm/yyyy)], [Sounding Analysis].[Date (dd/mm/yyyy)]) = 0 AND [SWOMS Analysis].[Time (UTC; 00:00:00)] = [Sounding Analysis].[Time (UTC; 00:00:00)] AND ((([SWOMS Analysis].[Tank13 VOL]-[Sounding Analysis].[Tank13 VOL])/[SWOMS Analysis].[Tank13 VOL]) >.05 OR (([Sounding Analysis].[Tank13 VOL]-[SWOMS Analysis].[Tank13 VOL])/[Sounding Analysis].[Tank13 VOL]) >.05)) AS Diff13, IIf([Diff13]>0,([Tank13 VOL]-([Tank13 VOL]-[Diff13]))/[Tank13 VOL],(([Tank13 VOL]+(-1*[Diff13]))-[Tank13 VOL])/([Tank13 VOL]+(-1*[Diff13]))) AS Percent13, [SWOMS
```

```
Analysis].[Tank14 VOL]-(SELECT TOP 1 [Tank14 VOL] FROM [Sounding Analysis] WHERE DateDiff("d", [SWOMS
Analysis].[Date (dd/mm/yyyy)], [Sounding Analysis].[Date (dd/mm/yyyy)]) = 0 AND [SWOMS Analysis].[Time
(UTC; 00:00:00)] = [Sounding Analysis].[Time (UTC; 00:00:00)] AND (((([SWOMS Analysis].[Tank14 VOL]-
[Sounding Analysis].[Tank14 VOL])/[SWOMS Analysis].[Tank14 VOL]) >.05 OR (([Sounding Analysis].[Tank14
VOL]-[SWOMS Analysis].[Tank14 VOL])/[Sounding Analysis].[Tank14 VOL]) >.05)) AS Diff14,
IIf([Diff14]>0,([Tank14 VOL]-([Tank14 VOL]-[Diff14]))/[Tank14 VOL],(([Tank14 VOL]+(-1*[Diff14]))-[Tank14
VOL])/([Tank14 VOL]+(-1*[Diff14]))) AS Percent14, [SWOMS Analysis].[Tank15 VOL]-(SELECT TOP 1 [Tank15 VOL]
FROM [Sounding Analysis] WHERE DateDiff("d", [SWOMS Analysis].[Date (dd/mm/yyyy)], [Sounding
Analysis].[Date (dd/mm/yyyy)]) = 0 AND [SWOMS Analysis].[Time (UTC; 00:00:00)] = [Sounding Analysis].[Time
(UTC; 00:00:00)] AND ((([SWOMS Analysis].[Tank15 VOL]-[Sounding Analysis].[Tank15 VOL])/[SWOMS
Analysis].[Tank15 VOL]) >.05 OR (([Sounding Analysis].[Tank15 VOL]-[SWOMS Analysis].[Tank15
VOL])/[Sounding Analysis].[Tank15 VOL]) >.05)) AS Diff15, IIf([Diff15]>0,([Tank15 VOL]-([Tank15 VOL]-
[Diff15]))/[Tank15 VOL],(([Tank15 VOL]+(-1*[Diff15]))-[Tank15 VOL])/([Tank15 VOL]+(-1*[Diff15]))) AS
Percent15, [SWOMS Analysis].[Tank16 VOL]-(SELECT TOP 1 [Tank16 VOL] FROM [Sounding Analysis] WHERE
DateDiff("d", [SWOMS Analysis].[Date (dd/mm/yyyy)], [Sounding Analysis].[Date (dd/mm/yyyy)]) = 0 AND
[SWOMS Analysis].[Time (UTC; 00:00:00)] = [Sounding Analysis].[Time (UTC; 00:00:00)] AND ((([SWOMS
Analysis].[Tank16 VOL]-[Sounding Analysis].[Tank16 VOL])/[SWOMS Analysis].[Tank16 VOL]) >.05 OR
(([Sounding Analysis].[Tank16 VOL]-[SWOMS Analysis].[Tank16 VOL])/[Sounding Analysis].[Tank16 VOL]) >.05))
AS Diff16, IIf([Diff16]>0,([Tank16 VOL]-([Tank16 VOL]-[Diff16]))/[Tank16 VOL],(([Tank16 VOL]+(-
1*[Diff16]))-[Tank16 VOL])/([Tank16 VOL]+(-1*[Diff16]))) AS Percent16
FROM [SWOMS Analysis]
WHERE ((((SELECT TOP 1 [Tank01 VOL] FROM [Sounding Analysis] WHERE DateDiff("d", [SWOMS Analysis].[Date
(dd/mm/yyyy)], [Sounding Analysis].[Date (dd/mm/yyyy)]) = 0  AND [SWOMS Analysis].[Time (UTC; 00:00:00)] =
[Sounding Analysis].[Time (UTC; 00:00:00)]AND((([SWOMS Analysis].[Tank01 VOL]-[Sounding Analysis].[Tank01
VOL])/[SWOMS Analysis].[Tank01 VOL]) >.05 OR (([Sounding Analysis].[Tank01 VOL]-[SWOMS Analysis].[Tank01
VOL])/[Sounding Analysis].[Tank01 VOL]) >.05))) Is Not Null)) OR ((((SELECT TOP 1 [Tank02 VOL] FROM
[Sounding Analysis] WHERE DateDiff("d", [SWOMS Analysis].[Date (dd/mm/yyyy)], [Sounding Analysis].[Date
(dd/mm/yyyy)]) = 0  AND [SWOMS Analysis].[Time (UTC; 00:00:00)] = [Sounding Analysis].[Time (UTC;
00:00:00)]AND((([SWOMS Analysis].[Tank02 VOL]-[Sounding Analysis].[Tank02 VOL])/[SWOMS Analysis].[Tank02
VOL]) >.05 OR (([Sounding Analysis].[Tank02 VOL]-[SWOMS Analysis].[Tank02 VOL])/[Sounding
Analysis].[Tank02 VOL]) >.05))) Is Not Null)) OR ((((SELECT TOP 1 [Tank03 VOL] FROM [Sounding Analysis]
WHERE DateDiff("d", [SWOMS Analysis].[Date (dd/mm/yyyy)], [Sounding Analysis].[Date (dd/mm/yyyy)]) = 0
AND [SWOMS Analysis].[Time (UTC; 00:00:00)] = [Sounding Analysis].[Time (UTC; 00:00:00)]AND((([SWOMS
Analysis].[Tank03 VOL]-[Sounding Analysis].[Tank03 VOL])/[SWOMS Analysis].[Tank03 VOL]) >.05 OR
(([Sounding Analysis].[Tank03 VOL]-[SWOMS Analysis].[Tank03 VOL])/[Sounding Analysis].[Tank03 VOL])
>.05))) Is Not Null)) OR ((((SELECT TOP 1 [Tank04 VOL] FROM [Sounding Analysis] WHERE DateDiff("d", [SWOMS
Analysis].[Date (dd/mm/yyyy)], [Sounding Analysis].[Date (dd/mm/yyyy)]) = 0  AND [SWOMS Analysis].[Time
(UTC; 00:00:00)] = [Sounding Analysis].[Time (UTC; 00:00:00)]AND((([SWOMS Analysis].[Tank04 VOL]-[Sounding
Analysis].[Tank04 VOL])/[SWOMS Analysis].[Tank04 VOL]) >.05 OR (([Sounding Analysis].[Tank04 VOL]-[SWOMS
Analysis].[Tank04 VOL])/[Sounding Analysis].[Tank04 VOL]) >.05))) Is Not Null)) OR ((((SELECT TOP 1
[Tank05 VOL] FROM [Sounding Analysis] WHERE DateDiff("d", [SWOMS Analysis].[Date (dd/mm/yyyy)], [Sounding
Analysis].[Date (dd/mm/yyyy)]) = 0  AND [SWOMS Analysis].[Time (UTC; 00:00:00)] = [Sounding
Analysis].[Time (UTC; 00:00:00)]AND((([SWOMS Analysis].[Tank05 VOL]-[Sounding Analysis].[Tank05
VOL])/[SWOMS Analysis].[Tank05 VOL]) >.05 OR (([Sounding Analysis].[Tank05 VOL]-[SWOMS Analysis].[Tank05
VOL])/[Sounding Analysis].[Tank05 VOL]) >.05))) Is Not Null)) OR ((((SELECT TOP 1 [Tank06 VOL] FROM
[Sounding Analysis] WHERE DateDiff("d", [SWOMS Analysis].[Date (dd/mm/yyyy)], [Sounding Analysis].[Date
(dd/mm/yyyy)]) = 0  AND [SWOMS Analysis].[Time (UTC; 00:00:00)] = [Sounding Analysis].[Time (UTC;
00:00:00)]AND((([SWOMS Analysis].[Tank06 VOL]-[Sounding Analysis].[Tank06 VOL])/[SWOMS Analysis].[Tank06
VOL]) >.05 OR (([Sounding Analysis].[Tank06 VOL]-[SWOMS Analysis].[Tank06 VOL])/[Sounding
Analysis].[Tank06 VOL]) >.05))) Is Not Null)) OR ((((SELECT TOP 1 [Tank07 VOL] FROM [Sounding Analysis]
WHERE DateDiff("d", [SWOMS Analysis].[Date (dd/mm/yyyy)], [Sounding Analysis].[Date (dd/mm/yyyy)]) = 0
AND [SWOMS Analysis].[Time (UTC; 00:00:00)] = [Sounding Analysis].[Time (UTC; 00:00:00)]AND((([SWOMS
Analysis].[Tank07 VOL]-[Sounding Analysis].[Tank07 VOL])/[SWOMS Analysis].[Tank07 VOL]) >.05 OR
(([Sounding Analysis].[Tank07 VOL]-[SWOMS Analysis].[Tank07 VOL])/[Sounding Analysis].[Tank07 VOL])
>.05))) Is Not Null)) OR ((((SELECT TOP 1 [Tank08 VOL] FROM [Sounding Analysis] WHERE DateDiff("d", [SWOMS
Analysis].[Date (dd/mm/yyyy)], [Sounding Analysis].[Date (dd/mm/yyyy)]) = 0  AND [SWOMS Analysis].[Time
(UTC; 00:00:00)] = [Sounding Analysis].[Time (UTC; 00:00:00)]AND((([SWOMS Analysis].[Tank08 VOL]-[Sounding
Analysis].[Tank08 VOL])/[SWOMS Analysis].[Tank08 VOL]) >.05 OR (([Sounding Analysis].[Tank08 VOL]-[SWOMS
Analysis].[Tank08 VOL])/[Sounding Analysis].[Tank08 VOL]) >.05))) Is Not Null)) OR ((((SELECT TOP 1
[Tank09 VOL] FROM [Sounding Analysis] WHERE DateDiff("d", [SWOMS Analysis].[Date (dd/mm/yyyy)], [Sounding
Analysis].[Date (dd/mm/yyyy)]) = 0  AND [SWOMS Analysis].[Time (UTC; 00:00:00)] = [Sounding
Analysis].[Time (UTC; 00:00:00)]AND((([SWOMS Analysis].[Tank09 VOL]-[Sounding Analysis].[Tank09
VOL])/[SWOMS Analysis].[Tank09 VOL]) >.05 OR (([Sounding Analysis].[Tank09 VOL]-[SWOMS Analysis].[Tank09
VOL])/[Sounding Analysis].[Tank09 VOL]) >.05))) Is Not Null)) OR ((((SELECT TOP 1 [Tank10 VOL] FROM
[Sounding Analysis] WHERE DateDiff("d", [SWOMS Analysis].[Date (dd/mm/yyyy)], [Sounding Analysis].[Date
(dd/mm/yyyy)]) = 0  AND [SWOMS Analysis].[Time (UTC; 00:00:00)] = [Sounding Analysis].[Time (UTC;
00:00:00)]AND((([SWOMS Analysis].[Tank10 VOL]-[Sounding Analysis].[Tank10 VOL])/[SWOMS Analysis].[Tank10
VOL]) >.05 OR (([Sounding Analysis].[Tank10 VOL]-[SWOMS Analysis].[Tank09 VOL])/[Sounding
```

```
Analysis].[Tank10 VOL]) >.05))) Is Not Null)) OR ((((SELECT TOP 1 [Tank11 VOL] FROM [Sounding Analysis] WHERE DateDiff("d", [SWOMS Analysis].[Date (dd/mm/yyyy)], [Sounding Analysis].[Date (dd/mm/yyyy)]) = 0 AND [SWOMS Analysis].[Time (UTC; 00:00:00)] = [Sounding Analysis].[Time (UTC; 00:00:00)]AND((([SWOMS Analysis].[Tank11 VOL]-[Sounding Analysis].[Tank11 VOL])/[SWOMS Analysis].[Tank11 VOL]) >.05 OR (([Sounding Analysis].[Tank11 VOL]-[SWOMS Analysis].[Tank11 VOL])/[Sounding Analysis].[Tank11 VOL]) >.05))) Is Not Null)) OR ((((SELECT TOP 1 [Tank12 VOL] FROM [Sounding Analysis] WHERE DateDiff("d", [SWOMS Analysis].[Date (dd/mm/yyyy)], [Sounding Analysis].[Date (dd/mm/yyyy)]) = 0 AND [SWOMS Analysis].[Time (UTC; 00:00:00)] = [Sounding Analysis].[Time (UTC; 00:00:00)]AND((([SWOMS Analysis].[Tank12 VOL]-[Sounding Analysis].[Tank12 VOL])/[SWOMS Analysis].[Tank12 VOL]) >.05 OR (([Sounding Analysis].[Tank12 VOL]-[SWOMS Analysis].[Tank12 VOL])/[Sounding Analysis].[Tank12 VOL]) >.05))) Is Not Null)) OR ((((SELECT TOP 1 [Tank13 VOL] FROM [Sounding Analysis] WHERE DateDiff("d", [SWOMS Analysis].[Date (dd/mm/yyyy)], [Sounding Analysis].[Date (dd/mm/yyyy)]) = 0 AND [SWOMS Analysis].[Time (UTC; 00:00:00)] = [Sounding Analysis].[Time (UTC; 00:00:00)]AND((([SWOMS Analysis].[Tank13 VOL]-[Sounding Analysis].[Tank13 VOL])/[SWOMS Analysis].[Tank13 VOL]) >.05 OR (([Sounding Analysis].[Tank13 VOL]-[SWOMS Analysis].[Tank13 VOL])/[Sounding Analysis].[Tank13 VOL]) >.05))) Is Not Null)) OR ((((SELECT TOP 1 [Tank14 VOL] FROM [Sounding Analysis] WHERE DateDiff("d", [SWOMS Analysis].[Date (dd/mm/yyyy)], [Sounding Analysis].[Date (dd/mm/yyyy)]) = 0 AND [SWOMS Analysis].[Time (UTC; 00:00:00)] = [Sounding Analysis].[Time (UTC; 00:00:00)]AND((([SWOMS Analysis].[Tank14 VOL]-[Sounding Analysis].[Tank14 VOL])/[SWOMS Analysis].[Tank14 VOL]) >.05 OR (([Sounding Analysis].[Tank14 VOL]-[SWOMS Analysis].[Tank14 VOL])/[Sounding Analysis].[Tank14 VOL]) >.05))) Is Not Null)) OR ((((SELECT TOP 1 [Tank15 VOL] FROM [Sounding Analysis] WHERE DateDiff("d", [SWOMS Analysis].[Date (dd/mm/yyyy)], [Sounding Analysis].[Date (dd/mm/yyyy)]) = 0 AND [SWOMS Analysis].[Time (UTC; 00:00:00)] = [Sounding Analysis].[Time (UTC; 00:00:00)]AND((([SWOMS Analysis].[Tank15 VOL]-[Sounding Analysis].[Tank15 VOL])/[SWOMS Analysis].[Tank15 VOL]) >.05 OR (([Sounding Analysis].[Tank15 VOL]-[SWOMS Analysis].[Tank15 VOL])/[Sounding Analysis].[Tank15 VOL]) >.05))) Is Not Null)) OR ((((SELECT TOP 1 [Tank16 VOL] FROM [Sounding Analysis] WHERE DateDiff("d", [SWOMS Analysis].[Date (dd/mm/yyyy)], [Sounding Analysis].[Date (dd/mm/yyyy)]) = 0 AND [SWOMS Analysis].[Time (UTC; 00:00:00)] = [Sounding Analysis].[Time (UTC; 00:00:00)]AND((([SWOMS Analysis].[Tank16 VOL]-[Sounding Analysis].[Tank16 VOL])/[SWOMS Analysis].[Tank16 VOL]) >.05 OR (([Sounding Analysis].[Tank16 VOL]-[SWOMS Analysis].[Tank16 VOL])/[Sounding Analysis].[Tank16 VOL]) >.05))) Is Not Null))
ORDER BY [SWOMS Analysis].[Date (dd/mm/yyyy)], [SWOMS Analysis].[Time (UTC; 00:00:00)], [SWOMS Analysis].[Tank01 VOL];
```

Delete Sounding:

```
DELETE [SoundingBook Data].[IMO Number], [SoundingBook Data].[Date (dd/mm/yyyy)]
FROM [SoundingBook Data]
WHERE ((([SoundingBook Data].[IMO Number])=[Forms].[DeleteRecords].[cmbIMO]) AND (([SoundingBook Data].[Date (dd/mm/yyyy)]) Between [Forms].[DeleteRecords].[txtStart] And [Forms].[DeleteRecords].[txtEnd]));
```

Sounding % CAP:

```
SELECT [Sounding Analysis].[Date (dd/mm/yyyy)], [Sounding Analysis].[Time (UTC; 00:00:00)], IIf(Nz([Capacity Query].[Tank01 Cap])=0,Null,((Nz([Sounding Analysis].[Tank01 VOL])/Nz([Capacity Query].[Tank01 Cap])))) AS [Tank 1], IIf(Nz([Capacity Query].[Tank02 Cap])=0,Null,((Nz([Sounding Analysis].[Tank02 VOL])/Nz([Capacity Query].[Tank02 Cap])))) AS [Tank 2], IIf(Nz([Capacity Query].[Tank03 Cap])=0,Null,((Nz([Sounding Analysis].[Tank03 VOL])/Nz([Capacity Query].[Tank03 Cap])))) AS [Tank 3], IIf(Nz([Capacity Query].[Tank04 Cap])=0,Null,((Nz([Sounding Analysis].[Tank04 VOL])/Nz([Capacity Query].[Tank04 Cap])))) AS [Tank 4], IIf(Nz([Capacity Query].[Tank05 Cap])=0,Null,((Nz([Sounding Analysis].[Tank05 VOL])/Nz([Capacity Query].[Tank05 Cap])))) AS [Tank 5], IIf(Nz([Capacity Query].[Tank06 Cap])=0,Null,((Nz([Sounding Analysis].[Tank06 VOL])/Nz([Capacity Query].[Tank06 Cap])))) AS [Tank 6], IIf(Nz([Capacity Query].[Tank07 Cap])=0,Null,((Nz([Sounding Analysis].[Tank07 VOL])/Nz([Capacity Query].[Tank07 Cap])))) AS [Tank 7], IIf(Nz([Capacity Query].[Tank08 Cap])=0,Null,((Nz([Sounding Analysis].[Tank08 VOL])/Nz([Capacity Query].[Tank08 Cap])))) AS [Tank 8], IIf(Nz([Capacity Query].[Tank09 Cap])=0,Null,((Nz([Sounding Analysis].[Tank09 VOL])/Nz([Capacity Query].[Tank09 Cap])))) AS [Tank 9], IIf(Nz([Capacity Query].[Tank10 Cap])=0,Null,((Nz([Sounding Analysis].[Tank10 VOL])/Nz([Capacity Query].[Tank10 Cap])))) AS [Tank 10], IIf(Nz([Capacity Query].[Tank11 Cap])=0,Null,((Nz([Sounding Analysis].[Tank11 VOL])/Nz([Capacity Query].[Tank11 Cap])))) AS [Tank 11], IIf(Nz([Capacity Query].[Tank12 Cap])=0,Null,((Nz([Sounding Analysis].[Tank12 VOL])/Nz([Capacity Query].[Tank12 Cap])))) AS [Tank 12], IIf(Nz([Capacity Query].[Tank13 Cap])=0,Null,((Nz([Sounding Analysis].[Tank13 VOL])/Nz([Capacity Query].[Tank13 Cap])))) AS [Tank 13], IIf(Nz([Capacity Query].[Tank14 Cap])=0,Null,((Nz([Sounding Analysis].[Tank14 VOL])/Nz([Capacity Query].[Tank14 Cap])))) AS [Tank 14], IIf(Nz([Capacity Query].[Tank15 Cap])=0,Null,((Nz([Sounding Analysis].[Tank15 VOL])/Nz([Capacity Query].[Tank15 Cap])))) AS [Tank 15], IIf(Nz([Capacity Query].[Tank16 Cap])=0,Null,((Nz([Sounding Analysis].[Tank16 VOL])/Nz([Capacity Query].[Tank16 Cap])))) AS [Tank 16], [Sounding Analysis].[Tank01 VOL], [Sounding Analysis].[Tank02 VOL], [Sounding Analysis].[Tank03 VOL], [Sounding Analysis].[Tank04 VOL], [Sounding Analysis].[Tank05 VOL], [Sounding Analysis].[Tank06 VOL], [Sounding Analysis].[Tank07 VOL], [Sounding Analysis].[Tank08 VOL], [Sounding Analysis].[Tank09 VOL], [Sounding Analysis].[Tank10 VOL], [Sounding Analysis].[Tank11 VOL],
```

```
[Sounding Analysis].[Tank12 VOL], [Sounding Analysis].[Tank13 VOL], [Sounding Analysis].[Tank14 VOL],
[Sounding Analysis].[Tank15 VOL], [Sounding Analysis].[Tank16 VOL]
FROM [Sounding Analysis], [Capacity Query]
ORDER BY [Sounding Analysis].[Date (dd/mm/yyyy)];
```

Sounding Analysis:

```
SELECT [SoundingBook Data].[IMO Number], [SoundingBook Data].[Date (dd/mm/yyyy)], [SoundingBook
Data].[Time (UTC; 00:00:00)], [SoundingBook Data].[Tank01 VOL], [SoundingBook Data].[Tank02 VOL],
[SoundingBook Data].[Tank03 VOL], [SoundingBook Data].[Tank04 VOL], [SoundingBook Data].[Tank05 VOL],
[SoundingBook Data].[Tank06 VOL], [SoundingBook Data].[Tank07 VOL], [SoundingBook Data].[Tank08 VOL],
[SoundingBook Data].[Tank09 VOL], [SoundingBook Data].[Tank10 VOL], [SoundingBook Data].[Tank11 VOL],
[SoundingBook Data].[Tank12 VOL], [SoundingBook Data].[Tank13 VOL], [SoundingBook Data].[Tank14 VOL],
[SoundingBook Data].[Tank15 VOL], [SoundingBook Data].[Tank16 VOL]
FROM [SoundingBook Data]
WHERE ((([SoundingBook Data].[IMO Number])=[Forms].[DetailReports].[cmbIMO]) AND (([SoundingBook
Data].[Date (dd/mm/yyyy)]) Between [Forms].[DetailReports].[txtStart] And
[Forms].[DetailReports].[txtEnd]))
ORDER BY [SoundingBook Data].[Date (dd/mm/yyyy)], [SoundingBook Data].[Time (UTC; 00:00:00)];
```

Sounding Change:

```
SELECT [Sounding % CAP].[Date (dd/mm/yyyy)], [Sounding % CAP].[Time (UTC; 00:00:00)], ([Sounding %
CAP].[Tank 1]-(SELECT TOP 1 [Tank 1] FROM [Sounding % CAP] AS Q WHERE (DateDiff("d", [Sounding %
CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date
(dd/mm/yyyy)]) AND (([Sounding % CAP].[Tank 1]-Q.[Tank 1])<-.3))) AS [Percent], ([Sounding % CAP].[Tank01
VOL]-(SELECT TOP 1 [Tank01 VOL] FROM [Sounding % CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date
(dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date
(dd/mm/yyyy)]) AND ([Sounding % CAP].[Tank 1]-(SELECT TOP 1 [Tank 1] FROM [Sounding % CAP] AS Q WHERE
(DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([Sounding %
CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([Sounding % CAP].[Tank 1]-Q.[Tank 1])<-.3))))) AS
Diff, ([Sounding % CAP].[Tank 2]-(SELECT TOP 1 [Tank 2] FROM [Sounding % CAP] AS Q WHERE (DateDiff("d",
[Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([Sounding % CAP].[Date
(dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([Sounding % CAP].[Tank 2]-Q.[Tank 2])<-.3))) AS Percent2,
([Sounding % CAP].[Tank02 VOL]-(SELECT TOP 1 [Tank02 VOL] FROM [Sounding % CAP] AS Q WHERE (DateDiff("d",
[Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([Sounding % CAP].[Date
(dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND ([Sounding % CAP].[Tank 2]-(SELECT TOP 1 [Tank 2] FROM [Sounding
% CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND
([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([Sounding % CAP].[Tank 2]-Q.[Tank
2])<-.3))))) AS Diff2, ([Sounding % CAP].[Tank 3]-(SELECT TOP 1 [Tank 3] FROM [Sounding % CAP] AS Q WHERE
(DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([Sounding %
CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([Sounding % CAP].[Tank 3]-Q.[Tank 3])<-.3))) AS
Percent3, ([Sounding % CAP].[Tank03 VOL]-(SELECT TOP 1 [Tank03 VOL] FROM [Sounding % CAP] AS Q WHERE
(DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([Sounding %
CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND ([Sounding % CAP].[Tank 3]-(SELECT TOP 1 [Tank 3]
FROM [Sounding % CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date
(dd/mm/yyyy)]) = -1) AND ([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([Sounding %
CAP].[Tank 3]-Q.[Tank 3])<-.3))))) AS Diff3, ([Sounding % CAP].[Tank 4]-(SELECT TOP 1 [Tank 4] FROM
[Sounding % CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) =
-1) AND ([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([Sounding % CAP].[Tank 4]-
Q.[Tank 4])<-.3))) AS Percent4, ([Sounding % CAP].[Tank04 VOL]-(SELECT TOP 1 [Tank04 VOL] FROM [Sounding %
CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND
([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND ([Sounding % CAP].[Tank 4]-(SELECT TOP 1
[Tank 4] FROM [Sounding % CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date
(dd/mm/yyyy)]) = -1) AND ([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([Sounding %
CAP].[Tank 4]-Q.[Tank 4])<-.3))))) AS Diff4, ([Sounding % CAP].[Tank 5]-(SELECT TOP 1 [Tank 5] FROM
[Sounding % CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) =
-1) AND ([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([Sounding % CAP].[Tank 5]-
Q.[Tank 5])<-.3))) AS Percent5, ([Sounding % CAP].[Tank05 VOL]-(SELECT TOP 1 [Tank05 VOL] FROM [Sounding %
CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND
([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND ([Sounding % CAP].[Tank 5]-(SELECT TOP 1
[Tank 5] FROM [Sounding % CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date
(dd/mm/yyyy)]) = -1) AND ([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([Sounding %
CAP].[Tank 5]-Q.[Tank 5])<-.3))))) AS Diff5, ([Sounding % CAP].[Tank 6]-(SELECT TOP 1 [Tank 6] FROM
[Sounding % CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) =
-1) AND ([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([Sounding % CAP].[Tank 6]-
Q.[Tank 6])<-.3))) AS Percent6, ([Sounding % CAP].[Tank06 VOL]-(SELECT TOP 1 [Tank06 VOL] FROM [Sounding %
CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND
```

```
([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND ([Sounding % CAP].[Tank 6]-(SELECT TOP 1
[Tank 6] FROM [Sounding % CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date
(dd/mm/yyyy)]) = -1) AND ([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([Sounding %
CAP].[Tank 6]-Q.[Tank 6])<-.3))))) AS Diff6, ([Sounding % CAP].[Tank 7]-(SELECT TOP 1 [Tank 7] FROM
[Sounding % CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) =
-1) AND ([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([Sounding % CAP].[Tank 7]-
Q.[Tank 7])<-.3))) AS Percent7, ([Sounding % CAP].[Tank07 VOL]-(SELECT TOP 1 [Tank07 VOL] FROM [Sounding %
CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND
([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND ([Sounding % CAP].[Tank 7]-(SELECT TOP 1
[Tank 7] FROM [Sounding % CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date
(dd/mm/yyyy)]) = -1) AND ([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([Sounding %
CAP].[Tank 7]-Q.[Tank 7])<-.3))))) AS Diff7, ([Sounding % CAP].[Tank 8]-(SELECT TOP 1 [Tank 8] FROM
[Sounding % CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) =
-1) AND ([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([Sounding % CAP].[Tank 8]-
Q.[Tank 8])<-.3))) AS Percent8, ([Sounding % CAP].[Tank08 VOL]-(SELECT TOP 1 [Tank08 VOL] FROM [Sounding %
CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND
([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND ([Sounding % CAP].[Tank 8]-(SELECT TOP 1
[Tank 8] FROM [Sounding % CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date
(dd/mm/yyyy)]) = -1) AND ([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([Sounding %
CAP].[Tank 8]-Q.[Tank 8])<-.3))))) AS Diff8, ([Sounding % CAP].[Tank 9]-(SELECT TOP 1 [Tank 9] FROM
[Sounding % CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) =
-1) AND ([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([Sounding % CAP].[Tank 9]-
Q.[Tank 9])<-.3))) AS Percent9, ([Sounding % CAP].[Tank09 VOL]-(SELECT TOP 1 [Tank09 VOL] FROM [Sounding %
CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND
([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND ([Sounding % CAP].[Tank 9]-(SELECT TOP 1
[Tank 9] FROM [Sounding % CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date
(dd/mm/yyyy)]) = -1) AND ([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([Sounding %
CAP].[Tank 9]-Q.[Tank 9])<-.3))))) AS Diff9, ([Sounding % CAP].[Tank 10]-(SELECT TOP 1 [Tank 10] FROM
[Sounding % CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) =
-1) AND ([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([Sounding % CAP].[Tank 10]-
Q.[Tank 10])<-.3))) AS Percent10, ([Sounding % CAP].[Tank10 VOL]-(SELECT TOP 1 [Tank10 VOL] FROM [Sounding
% CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND
([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND ([Sounding % CAP].[Tank 10]-(SELECT TOP
1 [Tank 10] FROM [Sounding % CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date
(dd/mm/yyyy)]) = -1) AND ([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([Sounding %
CAP].[Tank 10]-Q.[Tank 10])<-.3))))) AS Diff10, ([Sounding % CAP].[Tank 11]-(SELECT TOP 1 [Tank 11] FROM
[Sounding % CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) =
-1) AND ([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([Sounding % CAP].[Tank 11]-
Q.[Tank 11])<-.3))) AS Percent11, ([Sounding % CAP].[Tank11 VOL]-(SELECT TOP 1 [Tank11 VOL] FROM [Sounding
% CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND
([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND ([Sounding % CAP].[Tank 11]-(SELECT TOP
1 [Tank 11] FROM [Sounding % CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date
(dd/mm/yyyy)]) = -1) AND ([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([Sounding %
CAP].[Tank 11]-Q.[Tank 11])<-.3))))) AS Diff11, ([Sounding % CAP].[Tank 12]-(SELECT TOP 1 [Tank 12] FROM
[Sounding % CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) =
-1) AND ([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([Sounding % CAP].[Tank 12]-
Q.[Tank 12])<-.3))) AS Percent12, ([Sounding % CAP].[Tank12 VOL]-(SELECT TOP 1 [Tank12 VOL] FROM [Sounding
% CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND
([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND ([Sounding % CAP].[Tank 12]-(SELECT TOP
1 [Tank 12] FROM [Sounding % CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date
(dd/mm/yyyy)]) = -1) AND ([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([Sounding %
CAP].[Tank 12]-Q.[Tank 12])<-.3))))) AS Diff12, ([Sounding % CAP].[Tank 13]-(SELECT TOP 1 [Tank 13] FROM
[Sounding % CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) =
-1) AND ([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([Sounding % CAP].[Tank 13]-
Q.[Tank 13])<-.3))) AS Percent13, ([Sounding % CAP].[Tank13 VOL]-(SELECT TOP 1 [Tank13 VOL] FROM [Sounding
% CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND
([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND ([Sounding % CAP].[Tank 13]-(SELECT TOP
1 [Tank 13] FROM [Sounding % CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date
(dd/mm/yyyy)]) = -1) AND ([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([Sounding %
CAP].[Tank 13]-Q.[Tank 13])<-.3))))) AS Diff13, ([Sounding % CAP].[Tank 14]-(SELECT TOP 1 [Tank 14] FROM
[Sounding % CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) =
-1) AND ([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([Sounding % CAP].[Tank 14]-
Q.[Tank 14])<-.3))) AS Percent14, ([Sounding % CAP].[Tank14 VOL]-(SELECT TOP 1 [Tank14 VOL] FROM [Sounding
% CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND
([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND ([Sounding % CAP].[Tank 14]-(SELECT TOP
1 [Tank 14] FROM [Sounding % CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date
(dd/mm/yyyy)]) = -1) AND ([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([Sounding %
CAP].[Tank 14]-Q.[Tank 14])<-.3))))) AS Diff14, ([Sounding % CAP].[Tank 15]-(SELECT TOP 1 [Tank 15] FROM
```

```
[Sounding % CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([Sounding % CAP].[Tank 15]-Q.[Tank 15])<-.3))) AS Percent15, ([Sounding % CAP].[Tank15 VOL]-(SELECT TOP 1 [Tank15 VOL] FROM [Sounding % CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND ([Sounding % CAP].[Tank 15]-(SELECT TOP 1 [Tank 15] FROM [Sounding % CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([Sounding % CAP].[Tank 15]-Q.[Tank 15])<-.3))))) AS Diff15, ([Sounding % CAP].[Tank 16]-(SELECT TOP 1 [Tank 16] FROM [Sounding % CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([Sounding % CAP].[Tank 16]-Q.[Tank 16])<-.3))) AS Percent16, ([Sounding % CAP].[Tank16 VOL]-(SELECT TOP 1 [Tank16 VOL] FROM [Sounding % CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND ([Sounding % CAP].[Tank 16]-(SELECT TOP 1 [Tank 16] FROM [Sounding % CAP] AS Q WHERE (DateDiff("d", [Sounding % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([Sounding % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([Sounding % CAP].[Tank 16]-Q.[Tank 16])<-.3))))) AS Diff16
FROM [Sounding % CAP];
```

SoundingBook Data Query:

```
SELECT [SoundingBook Data].[IMO Number], [SoundingBook Data].[Date (dd/mm/yyyy)], [SoundingBook Data].[Time (UTC; 00:00:00)], [SoundingBook Data].[Tank01 LVL], [SoundingBook Data].[Tank01 VOL], [SoundingBook Data].[Tank02 LVL], [SoundingBook Data].[Tank02 VOL], [SoundingBook Data].[Tank03 LVL], [SoundingBook Data].[Tank03 VOL], [SoundingBook Data].[Tank04 LVL], [SoundingBook Data].[Tank04 VOL], [SoundingBook Data].[Tank05 LVL], [SoundingBook Data].[Tank05 VOL], [SoundingBook Data].[Tank06 LVL], [SoundingBook Data].[Tank06 VOL], [SoundingBook Data].[Tank07 LVL], [SoundingBook Data].[Tank07 VOL], [SoundingBook Data].[Tank08 LVL], [SoundingBook Data].[Tank08 VOL], [SoundingBook Data].[Tank09 LVL], [SoundingBook Data].[Tank09 VOL], [SoundingBook Data].[Tank10 LVL], [SoundingBook Data].[Tank10 VOL], [SoundingBook Data].[Tank11 LVL], [SoundingBook Data].[Tank11 VOL], [SoundingBook Data].[Tank12 LVL], [SoundingBook Data].[Tank12 VOL], [SoundingBook Data].[Tank13 LVL], [SoundingBook Data].[Tank13 VOL], [SoundingBook Data].[Tank14 LVL], [SoundingBook Data].[Tank14 VOL], [SoundingBook Data].[Tank15 LVL], [SoundingBook Data].[Tank15 VOL], [SoundingBook Data].[Tank16 LVL], [SoundingBook Data].[Tank16 VOL]
FROM [SoundingBook Data]
WHERE ((([SoundingBook Data].[IMO Number])=[Forms].[SoundingBookData].[cmbIMO]) AND (([SoundingBook Data].[Date (dd/mm/yyyy)]) Between [Forms].[SoundingBookData].[txtStart] And [Forms].[SoundingBookData].[txtEnd]))
ORDER BY [SoundingBook Data].[Date (dd/mm/yyyy)];
```

Tank Names Query:

```
SELECT TankNames.[Tank Name]
FROM TankNames
WHERE (((TankNames.[IMO Number])=[Forms].[DetailReports].[cmbIMO]));
```

Appendix J: Data Transmission Proposal

SWOMS Data Transmission Formats and Methods for Future Department of Justice Cases

Proposal

Submitted by:

Patrick Brodeur

Renée Lanza

Elizabeth Morris

Edward Osowski

Sponsoring Agency:

United States Coast Guard

Office of Vessel Activities (COMDT CG-543), in conjunction with the

Office of Investigations and Casualty Analysis (COMDT CG-545)

Submitted to:

On-Site Liaison**:**

LCDR Chaning Burgess

Date: 18 November 2011

Table of Contents

J1. Summary

The proposed standardized plan for data format and transmission will make the jobs of the men and women currently involved much easier. In addition, the implementation of this plan would make for easy integration into the ECP database that will provide more in depth analyses in quicker time than any human analysis could be done. In short, the standardized plan is as follows:

1. Company is charged with a violation of MARPOL by DOJ
2. Implementation of SWOMS
    a. SWOMS installed on vessel
    b. First email sent from SWOMS to USCG
    c. Manually run parsing program on email, input IMO number
    d. Fill out tank identification sheet in order of tank listed in SWOMS email
    e. Send tank id sheet back to shipping company with sounding log book template—sounding log book template to be filled out in the order specified in tank id sheet.
    f. First sounding log book template send back
3. Regular submissions
    a. Via email every day to the USCG
    b. Every 2 weeks
        i. Sounding log book template
        ii. ORB scanned and sent in
4. First year random audits
    a. Generate reports on all ships and investigate ORB randomly within the first year

J2. Introduction

Currently, the SWOMS data travels through many hands before it gets to the United States Coast Guard. Not only does it travel through many steps before reaching the Coast Guard, but every company does things a little different. This study of data transmission is particular to one shipping company that currently has four ships under the SWOMS program. The current method and formats of data transmission are known as a result of extensive interviews the Environmental Compliance Plan Program Manager and a Coast Guard Marine Investigator. The general path of the data is portrayed in the image below in

Figure 70.

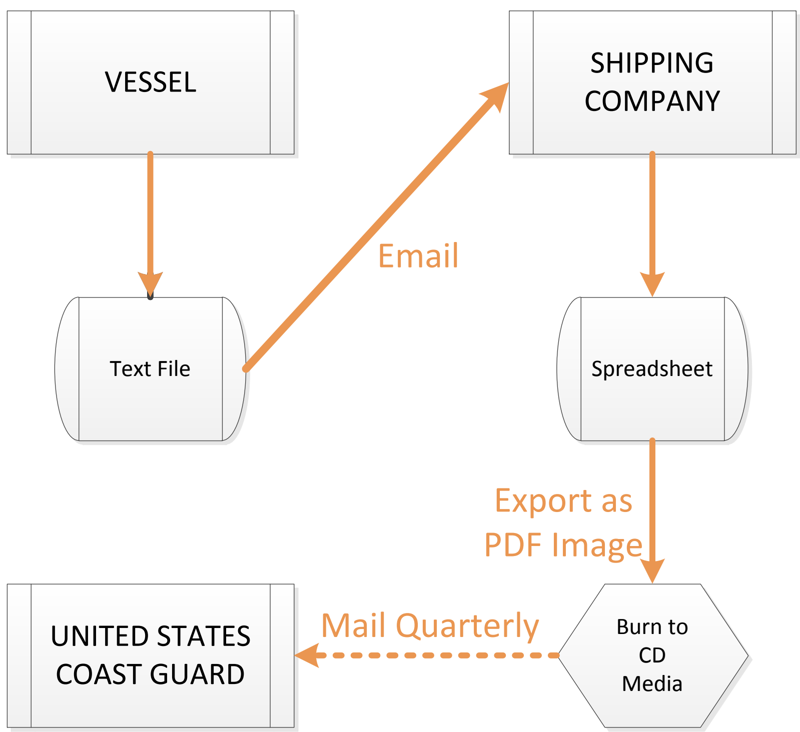


Figure 70: Shows the Data Transmission Process

Figure 70 shows the path of the SWOMS data from the ship to the United States Coast Guard. It can be seen that it does not directly go from the ship to the Coast Guard. In fact, the data are sent in emails as a text file once a day to the shipping company. At the end of each month, the shipping company compiles the data from the text file emails into a spreadsheet with data for the entire month. The company does this for every ship with a SWOMS onboard. The spreadsheets are converted to PDFs and are compiled onto a CD with a scanned copy of the oil record book for every ship for the specific time period. The CDs contain one month of data for all ships, but are sent quarterly. This means several CDs will arrive to the Coast Guard at once. The problem with this is that cumbersome amounts of data are being given at once, and the Coast Guard has a short window of time to act legally to extend a vessel's probation if in fact the data proves incompliance. The following sections outline a proposed new standardized plan for data entry, transmission, and timeline.

J3. Problems Identified in the Current System

There were a few problems in the current data transmission process that led to the proposal of a new standard process. Since the amount of ships that submit the information is still few, these problems have been manageable. Through thorough analysis of the current procedures as well as speaking with USCG employees, it was determined that for a more wide-spread expansion of the SWOMS, these problems needed to be addressed. Problems that were addressed in the design of this new standardized submission plan are as follows:

- Data being submitted via CD
- Data being submitted quarterly
- Data being submitted in different formats
- Data being received not compatible with a database
- Current system does not allow for a direct comparison between the ORB data and the SWOMS
- Data would go to shipping companies then to USCG

These concerns are some of the concerns that were addressed in the creation of a new system to be proposed. Receiving the data quarterly on CDs makes for large quantities of data coming in at once as fast at mail can get it from the shipping companies to the USCG. This would make for large amounts of data to be analyzed and sometimes in the time it would take to receive and analyze the data it was too late to take legal action. In addition to the data being submitted in large quantities via the mail, the SWOMS data are being submitted in different formats from vessel to vessel. This can be fine for human analysis but in the switch to a computer analysis program, this needed to be changed. In addition, human analysis also needs to be conducted to compare the ORB to the SWOMS currently. This tends to be tedious and time-consuming. Instead, the proposed data submissions will be able to help in electronic comparison between the sounding book and the SWOMS. All of these problems in the current system have been addressed in the creation of the standardized system.

J4. Proposed System

J4.1 New Standard Transmission Method

The current transmission method was outlined previously in this report. In summary, the ships send the data to the shipping company, who send the data to the Coast Guard on a quarterly basis. Currently, the data comes in larges quantities and by the time it is received and analyzed, a significant amount of time has already passed. The problem with this is that the Coast Guard has an allotted time in which they must make a case for a violation detected to extend the vessel's probation. In order to address these concerns, as well as accommodate the amount of data being sent, there is a proposal for a new data transmission method. The following are recommendations to optimize the SWOMS submission process as well as the implementation of the templates outlined in the later sections.

J4.1.1 Mandate that Emails Replace CDs

Since the current method of sending data on CDs that come via mail is insufficient, it is proposed that the data be sent via email. CDs come in to the Coast Guard long after the data are logged, and then there are further time losses because a person needs to manually extract the information from the CDs. After retrieving data from the CDs, a person would then also need to import the data to the database before analysis can begin. Email is significantly faster and will be able to accommodate the future projection of establishing a real-time system. Sending the information via email directly from the ship to the USCG would benefit all parties. The shipping companies would not be responsible for formatting the data from the emails into a spreadsheet. In addition, the Coast Guard would get the data more timely as well as in the format desired. This system will also incorporate SWOMS data into the Environmental Compliance Plan database more quickly.

J4.1.2 Mandate that Emails from SWOMS Replace the PDF files

The format of the current SWOMS spreadsheets is not compatible with the database to store the information. If the format of the SWOMS data is not changed, it would need to be manually entered because the data are currently sent in PDF format. Currently the SWOMS sends emails to the shipping company with all of the information that is currently in the SWOMS spreadsheets sent on the CDs. The Coast Guard would be receiving the emails that SWOMS sends to the shipping companies and would be accountable for compiling the data into the database. This would take a burden off of the shipping companies and would give the Coast Guard the information needed.

J4.1.3 Mandate Bi-Weekly Data Submission

To help address the concern of receiving the data more quickly and the proposed email submission, it is also proposed to mandate data submissions via email every two weeks. Receiving the data every two weeks will help catch any violations in a timely manner and will allow for adequate time to take legal measures if necessary. This also will help to not exceed the 5MB email limit at the United States Coast Guard. Required every two weeks will be a copy of the sounding logs and the oil record books that will correspond with the dates for the previous two weeks

J4.1.4 Mandate Use of Proposed Templates

Using the proposed templates will make for a uniform format of all SWOMS data, making it easier for analysis purposes. This will also be good for the integration of SWOMS and sounding book data into the database that will store and analyze it. Templates are addressed in the next section.

J4.1.5 Minimally-Managed Exchange Mailbox

As part of the automation process, it was decided that a minimally-managed, shared mailbox would be used as a destination for incoming data. The mailbox acts much like a "no-reply" email address, but administrators will perform first-time setup for each new ship that gets added to the system. The process requires human control at the beginning of new environmental compliance plans in order to correctly establish data associations. Once set into motion, the system will operate on existing ships without human intervention.

The process starts when the first SWOMS email comes in from a particular ship. In addition to other setup required for the database, the first SWOMS email will need to be manually passed through the parsing program described below. In addition, the first email will be used to add on to the Outlook rule for automatic processing. Once the first entry is created by the parsing program, with a human-inputted IMO number, every successive entry will automatically match that IMO number. After the first entry goes into the ship-specific CSV table, the administrator can also add it to the master linked-list table.

For subsequent entries, the process for SWOMS data are entirely automated. Emails arrive in the shared mailbox and are processed using a VBA macro. First, the email is checked for attachments and moved to a separate folder if it contains a CSV document (assumed to be SWOMS data). If there are no attachments, the email is converted to plaintext to remove HTML tags, moved to another folder, and saved to a time-and-date-stamped text file. The text file is then processed to add it to its ship-specific CSV document, and the new entry is copied to the master link-list CSV file. Once parsing is completed, the text file will be deleted, but the original email will still be retained. In order to reduce overhead on the Exchange server, the mailbox should be set up to auto-archive these emails.

J4.1.6 Program to Convert SWOMS Data into a Database-Readable Format

As previously described, the data coming in to the Coast Guard is not well-suited for direct input to the database. The CDs require manual insertion and retrieval of the data. The individual text files cannot be directly imported, and the PDF spreadsheet is virtually useless for programmatic manipulation. The solution for importing the data are to use a program to automatically convert the text documents into a standard, easily-manipulated format. A program was written as part of this project that parses the text documents into a single comma-delimited spreadsheet, also known as a "comma-separated values" file, or "CSV". A standard CSV file has many limitations in that it does not support any of the formulas, formatting, colors, or even column width information that is normal for any modern spreadsheet.

However, the CSV format takes up a fraction of the size of an excel file since it lacks any of those above mentioned features. This is preferable since it reduces overhead, especially for large amounts of data. The relative ease of converting the text documents to CSVs for importation into the database means that a plaintext email is usable with little processing. The speed benefit of this method is twofold; email can arrive significantly faster than physical mail services, and the data importation system can be automated so as to remove the manual extraction of data from CDs. In addition, this should minimize the risk of any human error in data input. This system should allow for significant expandability, allowing additional ships to be added without changing the processing elements provided that the incoming data matches the newly developed templates. Furthermore, the simplicity of this system allows its use with basic data output from the embedded systems one would expect in a SWOMS unit.

J4.2 Templates for Database Integration

One of the deliverables of this project is to propose a more standard plan for DoJ to use in future sanctions. This includes a format in which data are to be sent, which data are to be included in the data sent, as well as a standard plan for sending the data in to the United States Coast Guard. These templates are a response to some of the drawbacks that were addressed in the previous and current submission formats. The two things considered most in the development of the new templates for data submission are (1) Database readability and (2) More human analysis friendly.

J4.2.1 SWOMS Data Template

Several of the problems with the current SWOMS spreadsheets were expressed. Most notable of these problems is that the format of these spreadsheets is unfavorable for analytical purposes. A new template for the data was drafted to address all the previous concerns, in addition to formatting the data for database integration. Since human viewing of the data will be filtered and formatted in reports generated by the database, a human-friendly format for the data was a consideration rather than the primary influence.

J4.2.2 Advantages of the New Templates

The new template includes more information than was previous displayed in the SWOMS spreadsheets. The templates were designed to contain all of the information that SWOMS records, including elements such as location which were not previously shown. Having all of the possible information in one spreadsheet will make it easier to track trends. For example, if a ship is not moving and there seems to be no activity, it can potentially be assumed that the ship is at port and has not been accumulating waste to process. In Figure 71 below is a section of the SWOMS template.

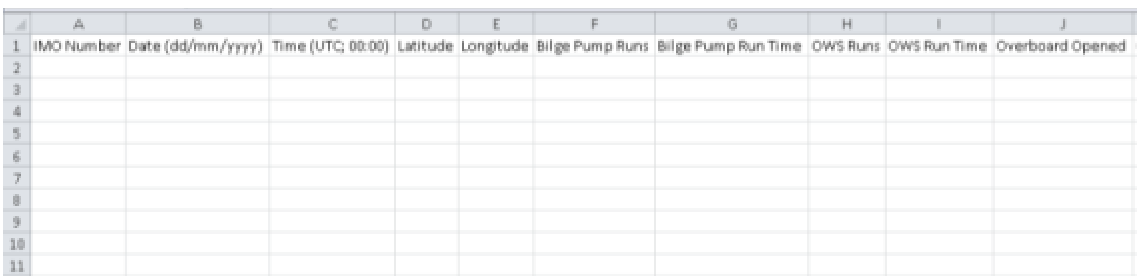


Figure 71: SWOMS Template

The new template is laid out effectively containing all information within the emails sent from the ship to the shipping. The SWOMS template consists of 51 columns per data entry (32 of which give room for up to 16 tanks, which is more than most vessels have). The template contains the following information:

- IMO Number
- Date (dd/mm/yyyy)
- Time (UTC; 00:00)
- Latitude & Longitude
- Bilge Pump Runs & Run Time
- OWS Runs & Run Time
- Overboard Opened & Open Time
- PPM Alarms
- Average PPM Overboard
- OCM Freshwater Valve Opened & Open Time
- Oil Purge Valve Opened & Open Time
- Incinerator Runs & Run Time
- Individual Tank Levels & Volumes

The information given in the newer version of the spreadsheet is much more detailed, making it easier to identify potential anomalies. If the PPM Alarm is sounded several times or the OCM Freshwater Valve has been opened multiple times, there may be an issue. Below, in Figure 72, is an image of a SWOMS template populated with real ship data.

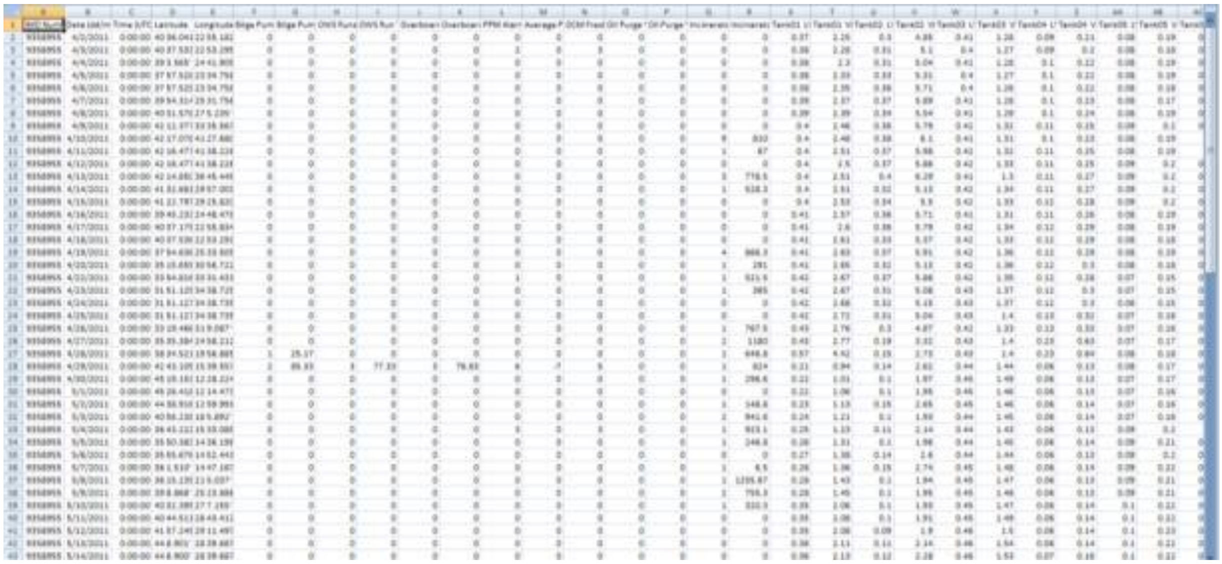


Figure 72: Populated SWOMS Template

This view shows just how much data are being collected in these templates. It also highlights how different the template is compared to the older spreadsheet. In a closer look at the template, as in Figure 73, the identifying information can be seen on the left.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	IMO Numb	Date (dd/m	Time (UTC	Latitude	Longitude	Bilge Pum	Bilge Pum	OWS Runs	OWS Run	Overboar	Overboar	PPM Alarr	Average P	OCM Fresl	Oil Purg
2	9358955	4/2/2011	0:00:00	40 36.041	22 55.182	0	0	0	0	0	0	0	0	0	
3	9358955	4/3/2011	0:00:00	40 37.533	22 53.295	0	0	0	0	0	0	2	0	3	
4	9358955	4/4/2011	0:00:00	39 3.565'	24 41.909	0	0	0	0	0	0	0	0	0	
5	9358955	4/5/2011	0:00:00	37 57.528	23 34.758	0	0	0	0	0	0	0	0	0	
6	9358955	4/6/2011	0:00:00	37 57.529	23 34.758	0	0	0	0	0	0	0	0	0	
7	9358955	4/7/2011	0:00:00	39 54.314	25 31.756	0	0	0	0	0	0	0	0	0	
8	9358955	4/8/2011	0:00:00	40 31.578	27 5.235'	0	0	0	0	0	0	0	0	0	
9	9358955	4/9/2011	0:00:00	42 12.377	33 35.567	0	0	0	0	0	0	0	0	0	

Figure 73: SWOMS Template—Identifier Information and Alarms

Each record is identified with the IMO number. This will be useful when the data are imported into the database since each vessel has a unique IMO number. All date, time, and location information is directly next to the identifier to show time and location of each entry. The columns to the right of these identifiers, such as the alarms and valve notifications, summarize the 24-hour data collection period. This information includes the number of times something happened as well as the duration of the activity during the 24 hour data collection. Towards the right of this alarms section is the instantaneous tank level and volume section that can be seen in Figure 74.

Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC
Incinerato	Incinerato	Tank01 L\	Tank01 V	Tank02 L\	Tank02 V	Tank03 L	Tank03 V	Tank04 L	Tank04 V	Tank05 L	Tank05 V	Tank0
0	0	0.37	2.25	0.3	4.85	0.41	1.28	0.09	0.21	0.08	0.19	0
0	0	0.38	2.28	0.31	5.1	0.4	1.27	0.09	0.2	0.08	0.18	0
0	0	0.38	2.3	0.31	5.04	0.41	1.28	0.1	0.22	0.08	0.19	0
0	0	0.38	2.33	0.33	5.31	0.4	1.27	0.1	0.22	0.08	0.19	0
0	0	0.38	2.35	0.36	5.71	0.4	1.26	0.1	0.22	0.08	0.18	0
0	0	0.39	2.37	0.37	5.89	0.41	1.28	0.1	0.23	0.08	0.17	0
0	0	0.39	2.39	0.34	5.54	0.41	1.29	0.1	0.24	0.08	0.19	0
0	0	0.4	2.46	0.36	5.79	0.42	1.32	0.11	0.25	0.09	0.2	0

Figure 74: SWOMS Template---Instantaneous Tank Data

On the right side of the SWOMS template, as can be seen above in Figure 26, are the instantaneous tank measurements. The levels and volumes of each tank are recorded instantaneously and sent within the SWOMS emails. This information is what will be directly compared to the ORB information. The template includes all the recorded information and is organized in a format that is readily importable into the database.

J4.2.3 Database Integration with the SWOMS Template

This template for the SWOMS data was created with the intent of database integration. The template was constructed to make sense to a human but also to a computer so that a computer can analyze the information and report anomalies. This standardized template will make it so that the import into the database will be easily carried out with minimal, if any errors. The format regarding the tank names such as "Tank01" was done specifically to ease analysis between the SWOMS data and the ORB data. The standards for the identities behind the tank names will be addressed in Section 0.

J4.2.4 ORB Data Template

Currently, analyzing oil record books against the data from the SWOMS is important in ensuring that the probationary vessels are in compliance with MARPOL regulations. Therefore, any discrepancies in the SWOMS data must be checked against the oil record book to indicate whether the discrepancies are indeed evidence of noncompliant activity. With the current handwritten version of the oil record book and the desire to incorporate the ORB data into the Environmental Compliance Plan Database, a method to enter the data into a digital template to be compared against the SWOMS data is necessary.

J4.2.5 Preliminary ORB Template

The initial solution to this dilemma was to manually transcribe the ORB entries into a spreadsheet that was organized by code and item number. The original thought being that it would the same as the current ORB but in a digital format, as can be seen below in Figure 75.

	A	B	C	D	E	F	G	H	I
1	IMO Numl	Date (mm/dd/yyyy)	Code	11.1 Colle	11.2 Capac	11.3 Retai	12. Amoui	12. Tank E	12. Retained
2	9327035	4/1/2011	C				0.23	M/E Scave	0
3	9327035	4/2/2011	D						
4	9327035	4/2/2011	C						
5	9327035								

Figure 75: Preliminary ORB Template

The reasoning behind this preliminary template for the ORB is to be a digital version of the oil record book to have all codes and items distinguishable from each other to help with visual analysis purposes as well as to be readable by the database created. In an effort to manually transcribe some of the ORB data into this template, the burden of how lengthy the process would take to transcribe the information as well as other concerns in how to directly compare this data to the SWOMS was realized. A re-evaluation of the approach towards the ORB data resulted in a new idea to instead use the sounding log book.

J4.2.6 Using the Sounding Log Book to Compare

When viewing a sounding log book, the information contained matches that of the instantaneous tank measurements that SWOMS produces. This makes it a good source of data to compare to the SWOMS data within the database. A sounding log book is a supplementary document in which all the tank levels and volumes are documented and signed by the Engine Officer. A sounding log book can be seen below, in Figure 76. The log book is a tabular representation of data within the ORB but in the format of the SWOMS data. It only includes tanks.

Figure 76: A Sounding Log Book

In order to make a direct comparison to the SWOMS data, a template to be manually filled out including this sounding log data is created. The template for the sounding book data can be seen below, in Figure 77. The template is an exact digital replica of the sounding log book but with a column to enter the time if given. This field will become more important when the SWOMS data switches to an hourly recording.

	B	C	D	E	F	G	H	I	J	K	
1	Date (mm/dd/yyyy)	Time (UTC; 00:00)	Tank01 L\	Tank01 V(	Tank02 L\	Tank02 V(	Tank03 L\	Tank03 V(	Tank04 L\	Tank04 V(	Tar
2	4/1/2011		1.560	20.740	0.560	5.260	0.220	0.200	0.900	1.130	
3	4/2/2011		1.560	20.740	0.820	8.920	0.220	0.200	1.200	1.590	
4	4/3/2011		1.560	20.740	0.900	10.130	0.220	0.200	1.100	1.440	
5	4/4/2011		1.560	20.740	1.100	13.280	0.220	0.200	1.100	1.440	
6	4/5/2011		1.560	20.740	1.180	14.400	0.220	0.200	1.100	1.440	
7	4/6/2011		0.530	4.730	1.720	22.370	0.220	0.200	1.170	1.570	
8	4/7/2011		1.900	25.500	0.520	4.730	0.220	0.200	1.170	1.570	
9	4/8/2011		1.900	25.500	0.540	4.990	0.220	0.200	1.170	1.570	
10	4/9/2011		0.700	6.830	0.620	6.060	0.220	0.200	0.900	1.130	

Figure 77: Sounding Log Book Template

Comparing the SWOMS data to a sounding log book is more beneficial because it's comparing the same types of data. The analysis will be more straightforward and will have the ability to produce a conclusion right away as to whether something should be looked at more carefully or if it is in the clear.

J4.2.7 Potential Problems with Using the Sounding Log Book

Using a vessel's transcribed sounding log book to compare SWOMS to rather than comparing it the transcribed ORB data is a much better solution to the challenge of being able to digitally analyze ORB data against the SWOMS data. However, there are a few concerns that may need to be further considered in the event of the future implementation of this system. The most notable concern is that although the sounding log book is signed and maintained, it does not have the same legal weight as the ORB. In the event that a shipping company knows that anomalies are being flagged via the sounding log book, they may falsify the sounding book in hopes that no one will look at the ORB and discover inconsistencies. This is a concern that will be addressed in recommendations.

J4.2.8 Tank Identification Template

One problem in this entire process is that vessels refer to their tanks differently than SWOMS. In addition to this, various vessels within the same fleet may refer to the same tank with different names as well. This element is established to ensure that the correct tanks within the database are compared during the computerized analysis process. To make sure that this takes place, a template is created to identify all tanks. The template would only need to be filled out once. Pictured below, in Figure 78, is the template to be filled out.

	A	B	C	D	
1	IMO Numl	Tank01 Type	Tank01 Desc.	Tank01 Capac	Tank02
2	[redacted]	Bilge	Clean BHT	32.21	
3					

Figure 78: Tank Identification Template

The template identifies the type of tank, a description, and the capacity with the "Tank01" and the IMO number. This identification makes for direct comparison as well as information about the tanks that is important for analysis.

J4.2.9 Entering Tank Identification Information

Tank identification information will be entered into the database via an electronic form. Using the form will populate the tank identification table within the database that is the exact format as the template above. It is suggested that the vessel is not entered into the database without 1 SWOMS email and 1 sounding log book sent. The tank identities are governed by the order in which they are reported within the SWOMS emails. They will automatically be generated into the SWOMS template in that order. The next task it to record the order and identify the tanks in order to send to the shipping companies with the template to the sounding log so that all the tanks labels correspond to the same tanks. Below in Figure 79 is the form from within the database that correlates with the tank identification sheet.

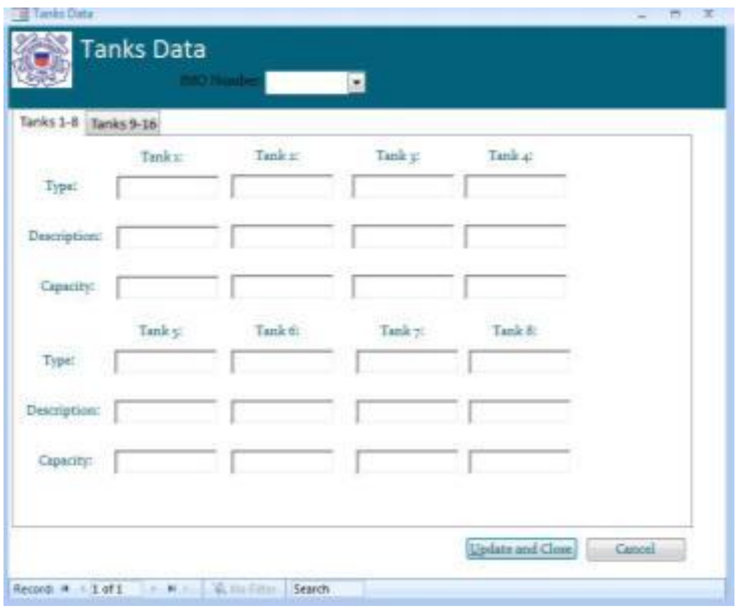


Figure 79: Tank ID Form

The above form allows for the initial entering of data as well as future editing of tank information. This makes it possible to be able to enter the tank type and description and then add in capacities once known.

J4.3 Conclusion

The proposed standardized plan for data format and transmission will make the jobs of the men and women currently involved much easier. In addition to making their jobs easier, the implementation of this plan would make for easy integration into the ECP database that will provide more in depth analyses in quicker time than any human analysis could be done. The templates are able to be used without this specific transmission process but would call for more manual interaction with the data and system as well as more human interaction. The changes to email as well as more frequent data submission are to aid in the eventual transition to hourly data collection. This will allow for less data overload and make for the least human interaction possible after the initial set up for each individual vessel.

J5. Recommendations

J5.1 Conduct Random Audits

To address the potential issue of companies falsifying the sounding logs and not the oil record books, it is proposed that random audits of a 1-to-2 week interval be conducted for ships at random during the first year of probation under these terms. This audit should include the analysis reports available in the database, as well as an extensive comparison between the SWOMS, sounding log book, and the ORB.

J5.2 SWOMS Implementation

In order to make sure that the integration of the data into the database is seamless, there is a recommendation for extensive detail oriented in the initial implementation of a SWOMS on a new ship. A one month period to set up the SWOMS as well as get all the supplementary materials in order is recommended. Once the first SWOMS email is sent to the Coast Guard, it will need to be manually run through the program to identify the vessel and emails with the unique IMO number. Once an initial SWOMS email arrives, someone at the USCG should fill out the tank identification form within the database as well as within the tank identification template. The filled out tank identification template along with the sounding logbook template is then to be sent to the vessel to fill out and sent in via email on a two week basis as well. Special attention to detail in the initiation set-up of all templates and database integration will help yield accurate detailed reports from the database. Failure to do so can result in mismatching tanks being compared or identification information within the database being wrong.

Appendix K: User Manual

Environmental Compliance: Acquisition, Storage, and Analysis of Waste Oil Data

User Manual

This guide is intended to be used in conjunction with the data transmission system and database developed by WPI students for the storage and analysis of shipboard waste data.

Developed For:

United States Coast Guard

Office of Vessel Activities (COMDT CG-543), in conjunction with the

Office of Investigations and Casualty Analysis (COMDT CG-545)

Prepared & Edited by:

Patrick Brodeur

Renée Lanza

Elizabeth Morris

Edward Osowski

Table of Contents

K1. Parsing Program

The parsing program “swomsparser” runs automatically after a combination of Outlook rules and VBA code is run on the incoming SWOMS email. The program should run without any user interaction. As subject-line IMO number retrieval has not implemented, the program uses dummy IMO numbers. The program first finds the vessel name within the automatically generated SWOMS email, and then looks for a vessel-specific CSV file named with the vessel name. If this file does not exist, the program will generate a new file, the first line of which will contain a dummy IMO number (“0000000”) and the vessel’s name, and then it will append the single record containing the data parsed from the email into that file. Otherwise, the program will append a single record containing the data parsed from the email to vessel-specific table, copying the IMO number contained in the first line. If that IMO number is verified to be an actual IMO number (i.e. not “0000000”), the program will proceed to add that same record to a master table containing SWOMS data (SWOMS_Table.csv) which is automatically monitored by the database.

In the case that the user desires to manually run the program on a specific file, there are two methods that can be used. The first is to use the Windows Command Prompt (see Figure 80). This method is preferable because it will display any relevant error messages in the case that the command is incorrectly typed.

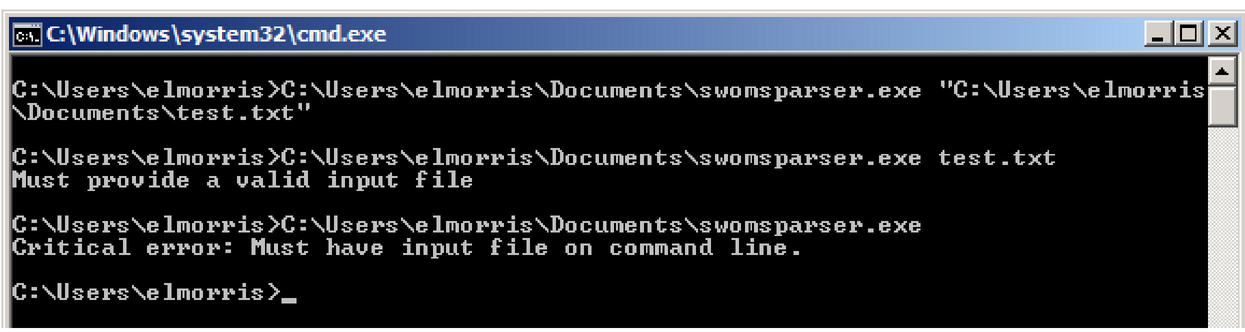


Figure 80: Run swomsparser.exe Via cmd.exe

In order to open the command prompt, go to the Start Menu, select “Run,” type in “cmd,” and then click “OK.” In the command prompt, type the path to the executable (e.g. “C:\Users\elmorris\Documents\swomsparser.exe” not including the quotes) followed by a space and then the path to the text file to be processed (e.g. “C:\Users\elmorris\Documents\textfile.txt” which may or may not include quotes). If this is entered correctly, the program should run without issue, and the program will not print any text in response, giving a fresh prompt. If the file that is referenced does not exist, or is not in the location indicated, the program will print the error message “Must provide a valid input file.” If no argument is given after the path to the executable, the program will print the error message “Critical error: Must have input file on command line.” If no error messages are shown, then the only way for the user to ensure that the file was parsed is to open the vessel-specific CSV file, which is created in the same directory as the input file.

The second method is to use the same command inputs as before, but to enter them directly into the "Run" dialogue box (see Figure 81) rather than the Windows Command Prompt.

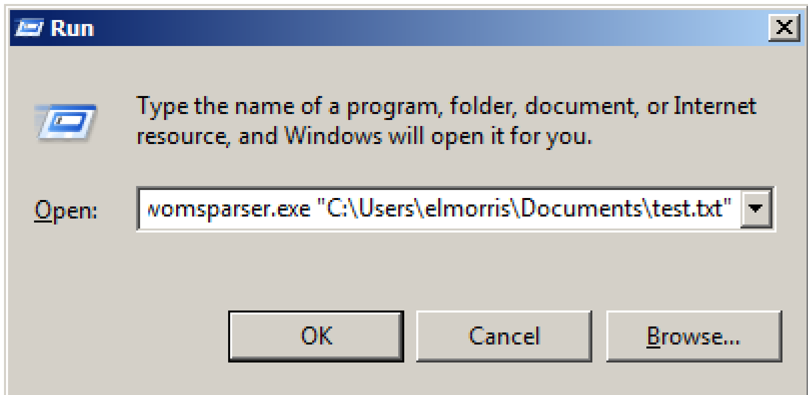


Figure 81: Run swomsparser.exe Via the Run Dialogue

This is a less helpful method because any error messages will only briefly flash on the screen before the window exits. Just as with system functions like "ping", the program will close its command interpreter window as soon as it has completed processing. This will not give the user enough time to read any error messages that are displayed.

K2. Database

This section of the user manual details the navigation and use of the database. The database has a combination of dropdown menus, text entry boxes, buttons, and check boxes for navigation. In addition to clicking on each of these items in order to select them, the database also has the capability to use keyboard shortcuts for navigation. Tab-navigation is enabled, which means that pressing the Tab key will select the next clickable element on each page. The buttons on some of the pages have a single letter underlined. If that key is pressed at the same time as the Alt key, it will perform the equivalent of clicking on that button. In order to open the database, simply navigate to the folder location where the database is saved, and open the Access database file just like any other file, it has the .accdb file extension.

K2.1 Importing Data

In order to add a new spreadsheet already containing data to the database, the user will have to use the Access tools for importing data. In order to import data, click the

External Data tab in the Access toolbar and select the Text File option contained within the Import & Link section (see Figure 82).

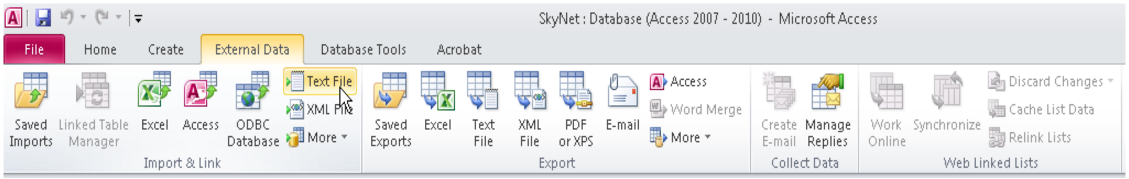


Figure 82: Access Toolbar to Import Data

In the window that opens hit browse to select the file to import. Select "Append a copy of the records to the table" and then either "SWOMS data" or "SoundingBook Data" depending on what is being imported. Finally, click OK (see Figure 83).

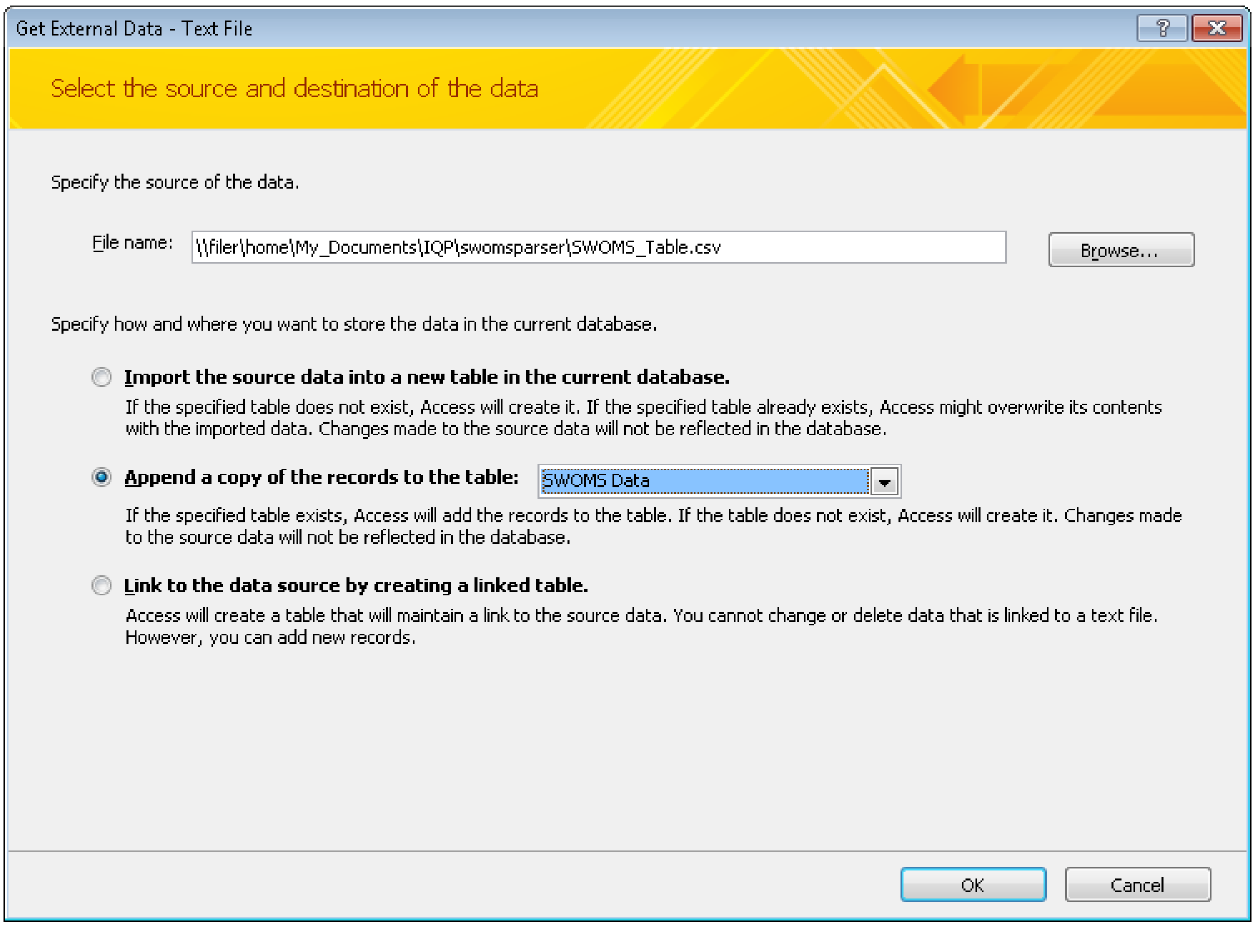


Figure 83: Text File Import Source and Destination

The Import Text Wizard will then open, on the first screen make sure Delimited is selected, and then click next. On the next screen, make sure that Comma is selected and that the "first Row Contains Field Names" box is checked, then click finish (see Figure 84). This will have to be done for all new data being added to the database.

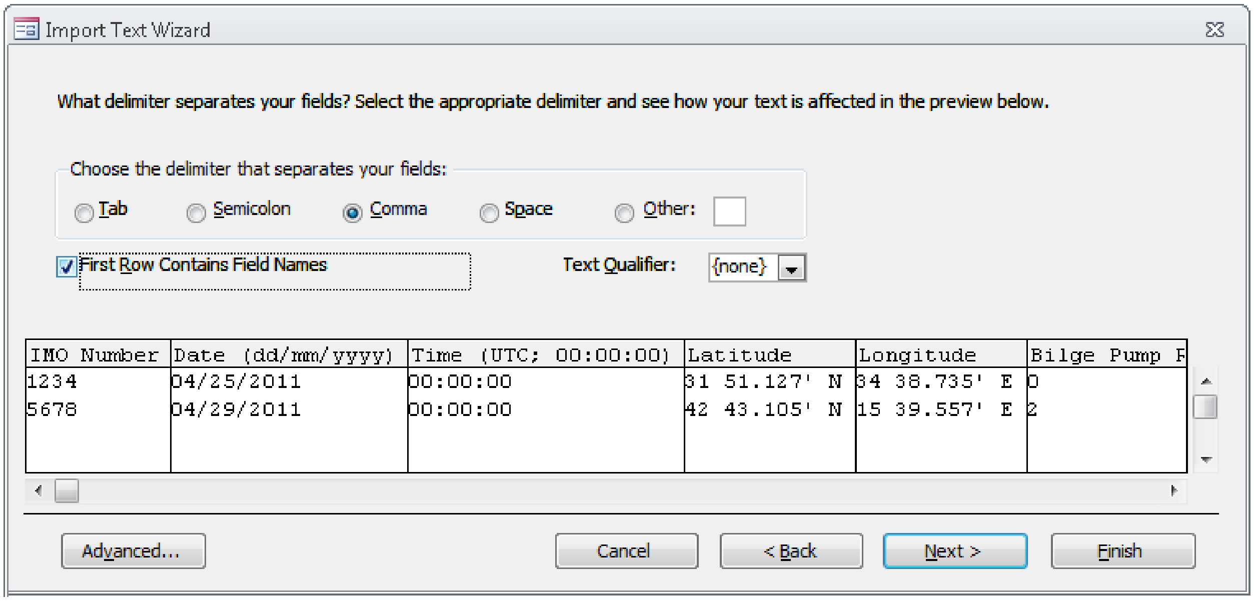


Figure 84: Import Data Delimiter Options

K2.2 Main Menu

Upon opening the database, the user is brought to the Main Menu (see Figure 85). This screen can be accessed by clicking the "Home" button located on most screens. It contains several buttons giving several different options. Nearly every aspect of the database can be directly accessed from this menu.

Figure 85: Main Menu

From within the leftmost section titled "Quick Reports", the user can generate quick reports for an individual vessel based on a date range (see Section K2.5.1).

From within the center section titled "Manage Data", the user can access the following data entry forms which are used to enter new data into the database:

- "New Vessel", which brings up the New Vessel form (see Section K2.3.1);
- "New SWOMS Data", which brings up the Add SWOMS Data form (see Section K2.3.2);
- "New Sounding Book Data", which brings up the Add Sounding Book Data form (see Section K2.3.3);
- "Edit Tanks," which brings up a window that allows the user to edit the tanks for a specific vessel (see Section K2.3.4); and
- "Delete Records," which brings up a window that allow the user to delete SWOMS or sounding book records (see Section K2.3.6)

From within the rightmost section titled "Links," the following options are accessible:

- "Detail Reports," which allows the user to generate detailed reports, which are more selective in the data displayed (see Section K2.5.2);
- "Vessel List," which displays a table with a list of the vessels in the database along with some other relevant information (see Section K2.4.1);
- "SWOMS Data," which links to the table containing the raw SWOMS data (see Section K2.4.2); and
- "Sounding Book Data," which links to the table containing the raw sounding book data (see Section K2.4.3).

K2.3 Data Entry and Modification

K2.3.1 New Vessel

The "New Vessel" form (see Figure 86), which is accessible through the Main Menu, is used to enter a new vessel. The form should appear as a pop-up window. This form must be filled out for each new vessel before any other data for that vessel can be added (including tank data, SWOMS data, and sounding data).

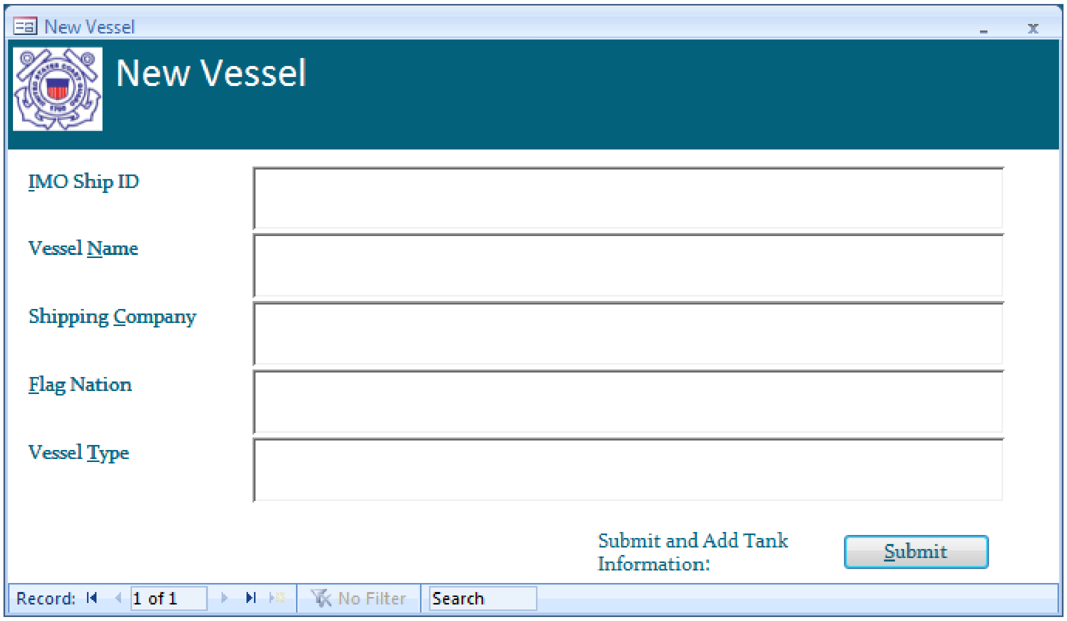


Figure 86: New Vessel form

This form provides spaces to enter the IMO Ship ID (the IMO Number), the Vessel Name, the Shipping Company, the Flag Nation, and the Vessel Type (e.g. Oil Tanker). Upon completion of the form, the user must click the "Submit" button at the lower right-hand corner in order to enter the information into the Vessel List in the database. This button will be disabled except for when the user hits tab from the Vessel Type text box. This is in order to prevent vessel data from being incorrectly uploaded to the tables. In addition, if the user tabs through to the submit button without filling in all fields, a message will pop up telling them to fill them in as all fields are required. In order to cancel the entry, simply close the window.

Clicking "Submit" should bring up the Tanks Data form (see Section K2.3.4), which allows the user to add tanks to a vessel. This form should be filled out by a knowledgeable Coast Guard employee based on the order of the tanks in the SWOMS emails.

K2.3.2 Add SWOMS Data

The "Add SWOMS Data" form, which can be accessed by selecting the "New SWOMS Data" button on the Main Screen, is used to submit a new entry of SWOMS data. It should appear as a pop-up window. At the top of this window is a drop-down menu that allows the user to select the relevant IMO number. This form has two tabbed sections:

- The first tabbed section (titled "Basic Info", see Figure 87) allows the user to enter basic information, including the date, time, location, and any shipboard system information.

- The second tabbed section (titled "Tanks", see Figure 88) allows the user to enter the instantaneous tank levels and volumes that relate to the date and time on the Basic Info tab. The tanks are referred to by their numerical assignments, as is set up in the Tanks Data form (see Section K2.3.4).

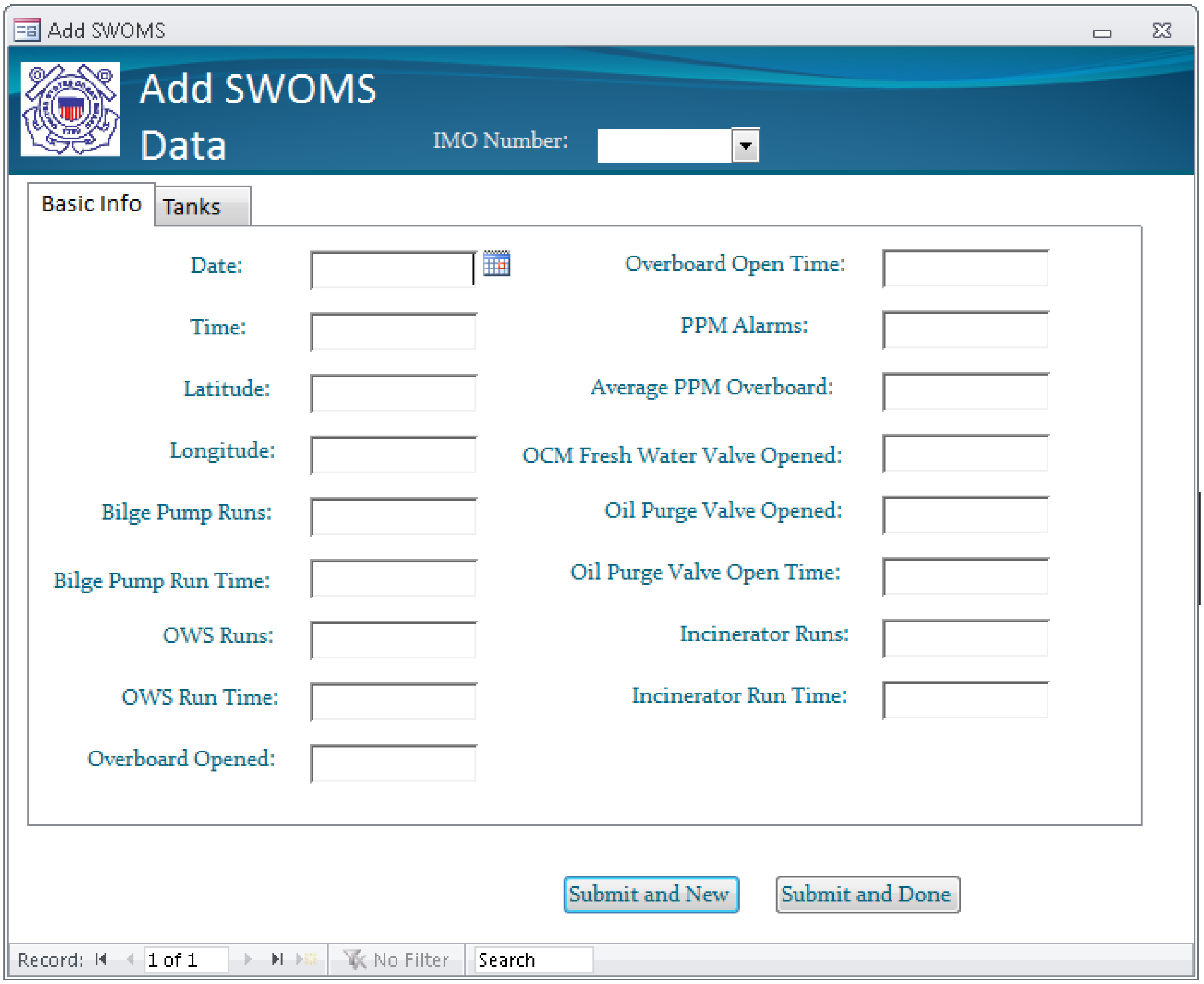


Figure 87: Add SWOMS Data form – Basic Info tab

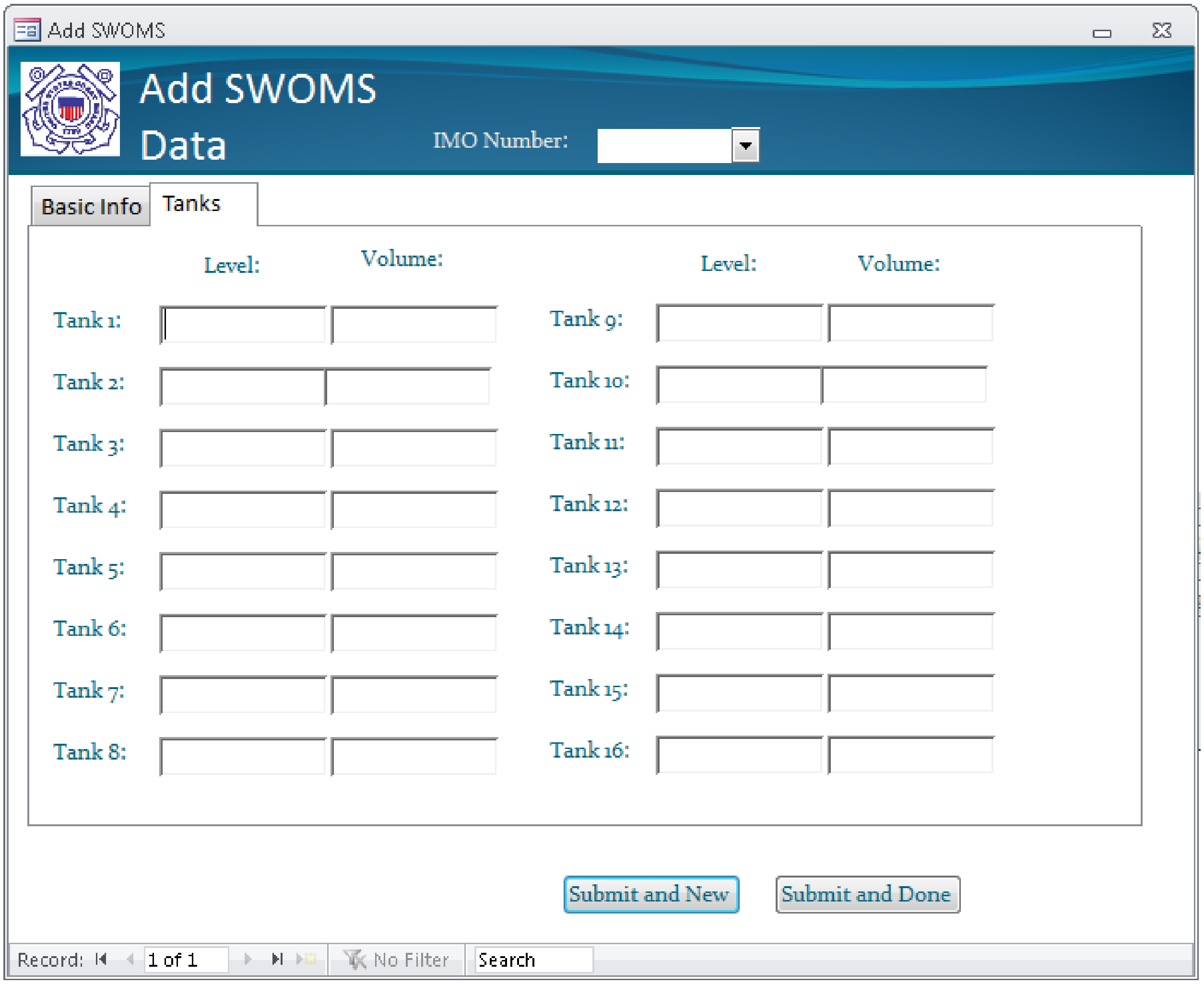


Figure 88: Add SWOMS Data form – Tanks tab

In order to enter the information upon completion of the form, the user may click the "Submit and New" button or the "Submit and Done" button, both of which are located in the lower right-hand corner. The "Submit and New" button will submit the data and then reset (clear) the form, allowing a new entry to be filled out, whereas the "Submit and Done" button will submit the data and then close the window.

K2.3.3 Add Sounding Book Data

The "Add Sounding Book Data" form (see Figure 89), which can be accessed by clicking the "New Sounding Book Data" button on the Main Screen, is used to submit a new entry of sounding book data. It should appear as a pop-up window.

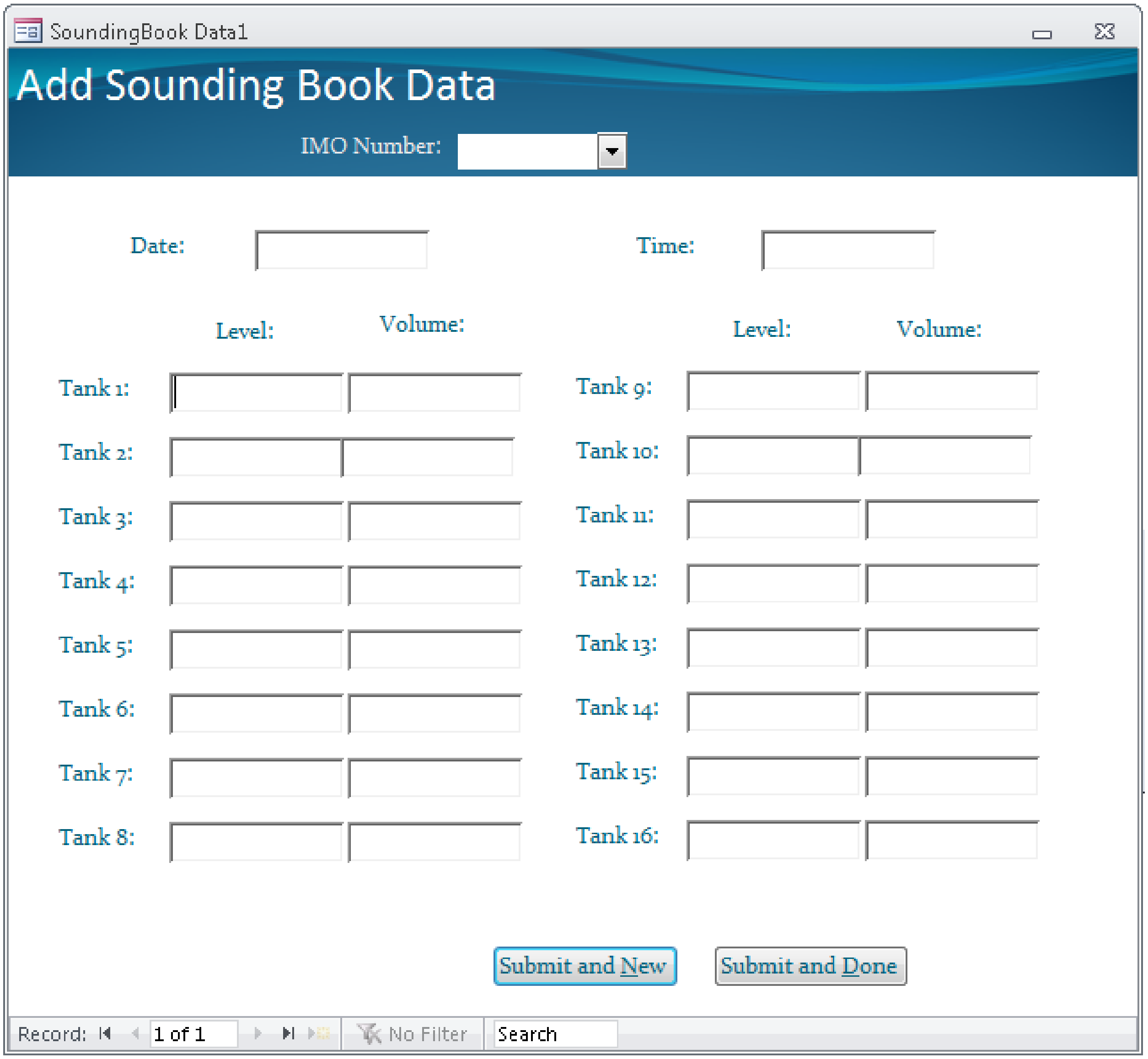


Figure 89: Add Sounding Book Data form

At the top of this window is a drop-down menu that allows the user to select the desired IMO number. This form contains spaces where the user can enter the date, time, and the instantaneous tank levels and volumes.

In order to enter the information upon completion of the form, the user may click the "Submit and New" button or the "Submit and Done" button, both of which are located in the lower right-hand corner. The "Submit and New" button will submit the data and then reset (clear) the form, allowing a new entry to be filled out, whereas the "Submit and Done" button will submit the data and then close the window.

K2.3.4 Modify Tanks Data

The "Tanks Data" form (see Figure 90), which can be accessed by clicking the "Edit Tanks" button on the Main Screen, is used to add or edit the list of tanks for each vessel.

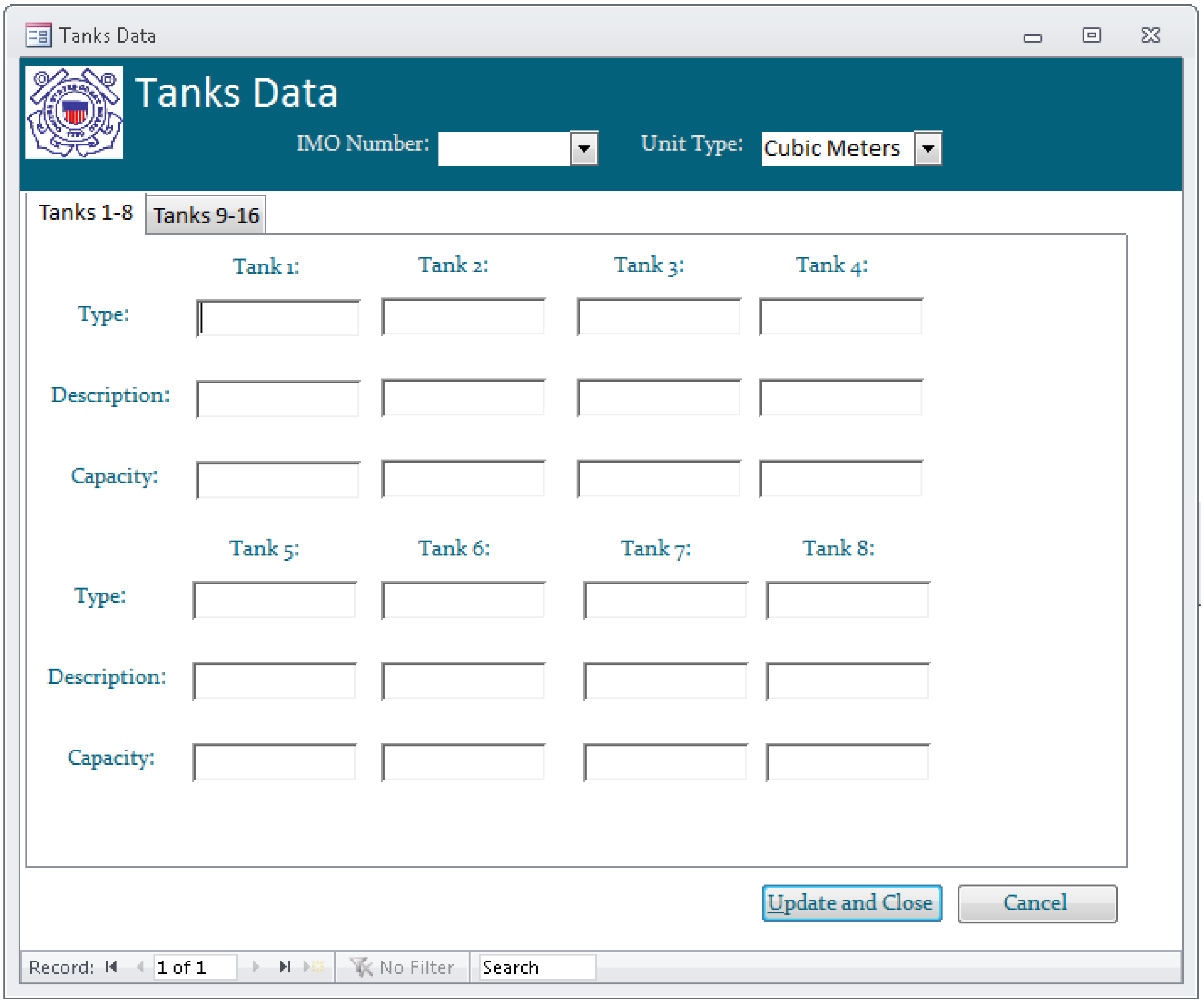


Figure 90: Tanks Data form

At the top of the form is a dropdown menu to select the IMO number of the desired vessel. There is also a dropdown menu that allows the user to choose what units (cubic meters or gallons) to use for the capacity of the tank. This form also includes spaces to enter the Type (Bilge or Sludge), Description (the tank name, e.g. "WO Service" or "Starboard Bilge"), and Capacity for up to 16 tanks (note the tabbed entry on the top of the form allowing access to Tanks 9-16). This form should be filled out by a knowledgeable Coast Guard employee in the same order that the tanks appear in the SWOMS emails.

Upon completion of this form, the user should click the "Update and Close" button, which will automatically enter the data into the database. Alternatively, click "Cancel" in order to close the form. This will lose any changes entered into the form.

K2.3.5 Vessel Details

The "Vessel Details" form (see Figure 91), which can be accessed through the "Detail Reports" form (see Section K2.5.2), as well as the "SWOMS Data and "Sounding Book Data" forms, is used to view and edit the existing vessel information. In order to open this form from the Detail Reports, simply select the desired IMO number from the dropdown button, and then click the "Vessel Details" button. This form is almost identical to the "New Vessel" form (see Section K2.3.1). It contains spaces to view the IMO Number and to view or modify the Vessel Name, the Shipping Company, the Flag Nation, and the Vessel Type.

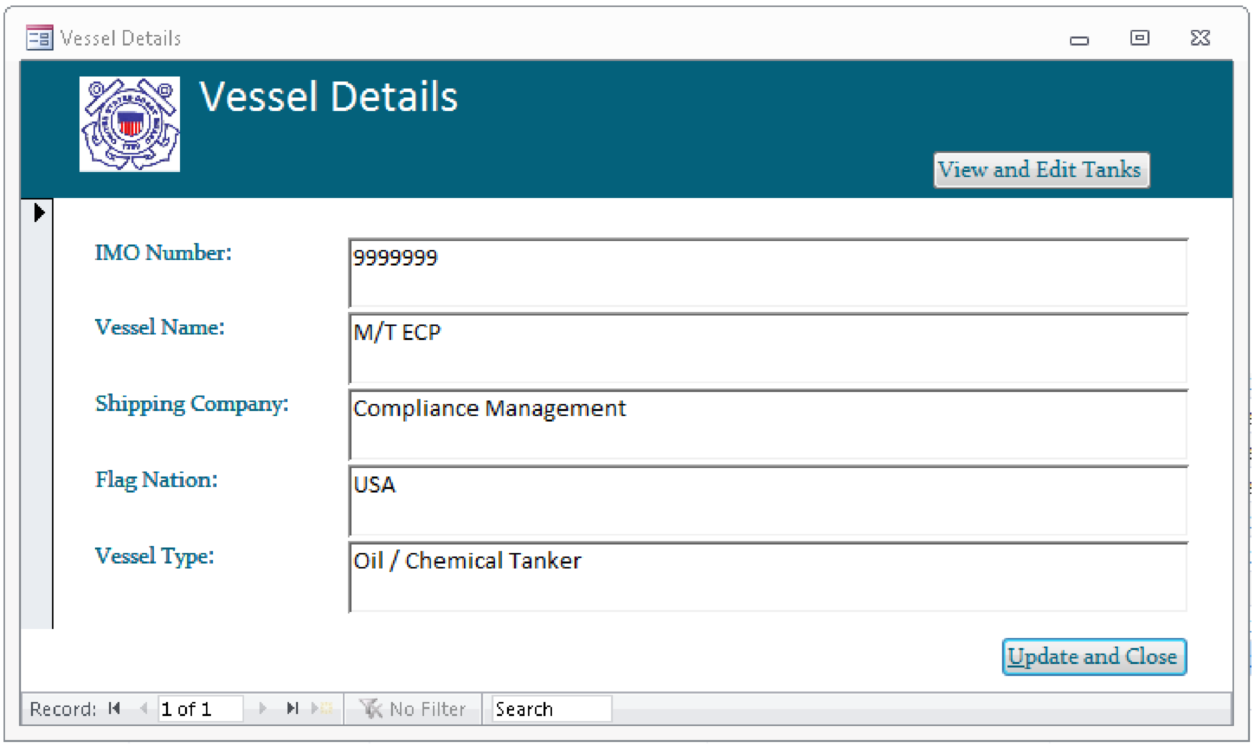


Figure 91: Vessel Details form

Once the user is satisfied with the information, click the "Update and Close" button at the bottom of the screen. This will submit the information to the database, replacing the previous information. Alternatively, the user can click the "View and Edit Tanks" button at the top of the screen in order to bring up the "Tanks Data" form (see Section K2.3.4).

K2.3.6 Delete Records

The "Delete Records" form (see Figure 92), which can be accessed through the Main Menu, is used to delete SWOMS or sounding book data records contained in the database.

This form includes a dropdown menu to choose the IMO Number, spaces to fill out a date range, and buttons to "Delete SWOMS data", "Delete Sounding Data", and to "Cancel."

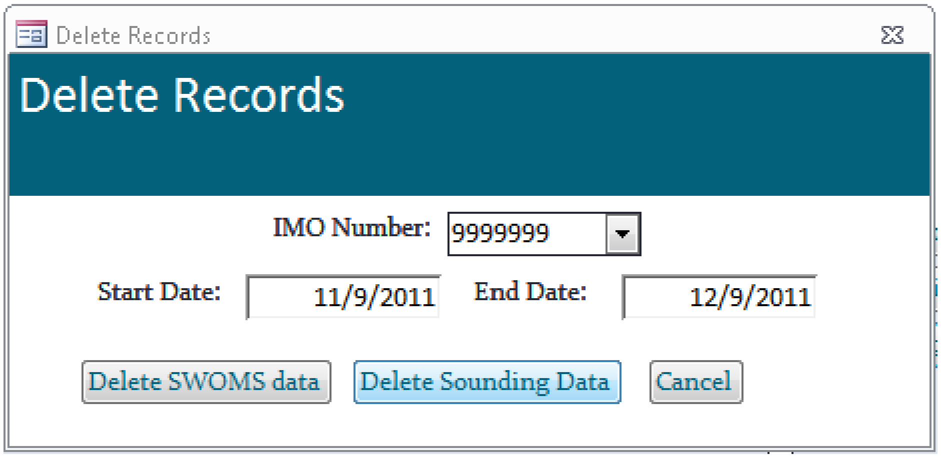


Figure 92: Delete Records form

Clicking "Delete SWOMS Data" will delete all of the SWOMS data for the selected IMO Number within the specified date range. Clicking "Delete Sounding Data" will delete all of the Sounding Data for the selected date range. Clicking either of these will bring up a dialog window asking if the user is sure about deleting the records (see Figure 93).

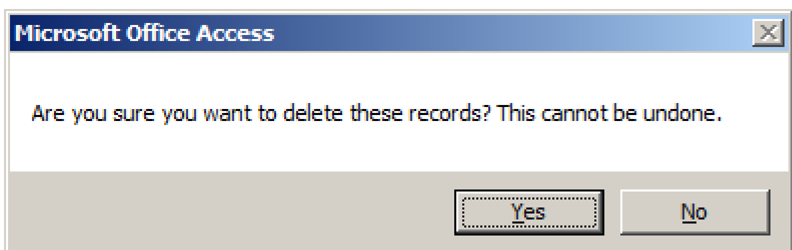


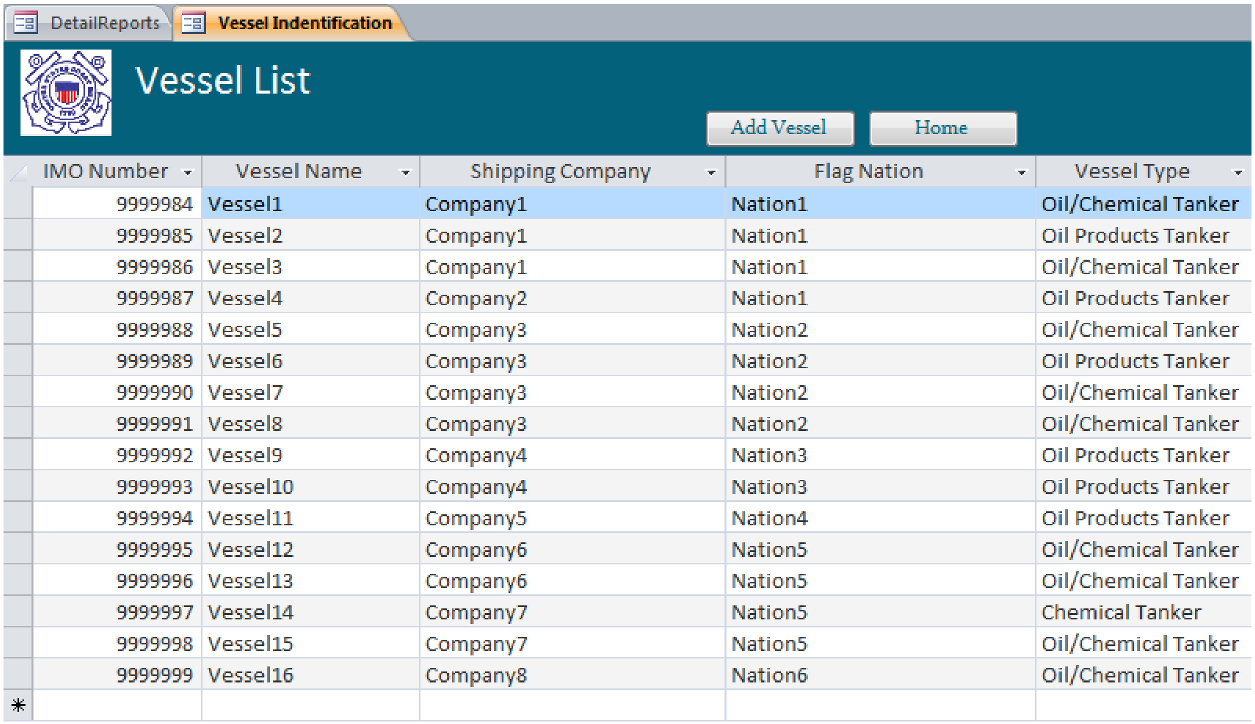
Figure 93: Delete Records Warning

Clicking "Yes" will delete the data and "No" will cancel the operation. This data deletion is permanent and cannot be undone.

K2.4 Data Viewing

K2.4.1 Vessel List

The "Vessel List" table (see Figure 94), which can be accessed by clicking the "Vessel List" button on the Main Screen, is used to view the list of vessels included in the database. The table includes columns for the IMO number, the Vessel Name, the Shipping Company, the Flag Nation, and the Vessel Type.



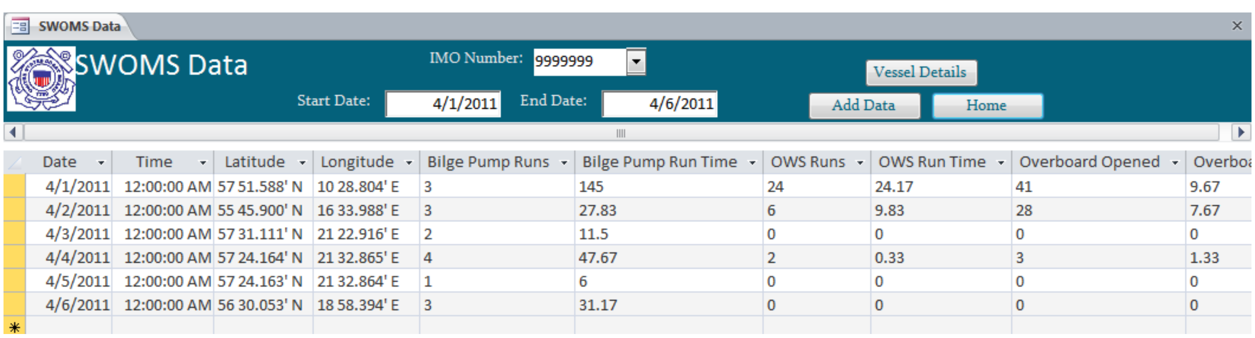
IMO Number	Vessel Name	Shipping Company	Flag Nation	Vessel Type
9999984	Vessel1	Company1	Nation1	Oil/Chemical Tanker
9999985	Vessel2	Company1	Nation1	Oil Products Tanker
9999986	Vessel3	Company1	Nation1	Oil/Chemical Tanker
9999987	Vessel4	Company2	Nation1	Oil Products Tanker
9999988	Vessel5	Company3	Nation2	Oil/Chemical Tanker
9999989	Vessel6	Company3	Nation2	Oil Products Tanker
9999990	Vessel7	Company3	Nation2	Oil/Chemical Tanker
9999991	Vessel8	Company3	Nation2	Oil/Chemical Tanker
9999992	Vessel9	Company4	Nation3	Oil Products Tanker
9999993	Vessel10	Company4	Nation3	Oil Products Tanker
9999994	Vessel11	Company5	Nation4	Oil Products Tanker
9999995	Vessel12	Company6	Nation5	Oil/Chemical Tanker
9999996	Vessel13	Company6	Nation5	Oil/Chemical Tanker
9999997	Vessel14	Company7	Nation5	Chemical Tanker
9999998	Vessel15	Company7	Nation5	Oil/Chemical Tanker
9999999	Vessel16	Company8	Nation6	Oil/Chemical Tanker

Figure 94: Vessel List table

The "Add Vessel" button in the top-right corner of the screen goes to the "New Vessel" form (see Section K2.3.1). The "Home" button in the top-right-hand corner of the screen returns to the Main Menu. Double-clicking on any ship in this list will bring up the Vessel Details form for that vessel (see Section K2.3.5), allowing the user to directly edit the information displayed in the table.

K2.4.2 SWOMS Data

The "SWOMS Data" form (see Figure 95 for a portion), which can be accessed by clicking the "SWOMS Data" button on the Main Screen, shows the SWOMS data included in the database. The form includes the date, time, location, any shipboard system information (e.g. OWS runs), and the instantaneous tank levels and volumes.



Date	Time	Latitude	Longitude	Bilge Pump Runs	Bilge Pump Run Time	OWS Runs	OWS Run Time	Overboard Opened	Overboa
4/1/2011	12:00:00 AM	57 51.588' N	10 28.804' E	3	145	24	24.17	41	9.67
4/2/2011	12:00:00 AM	55 45.900' N	16 33.988' E	3	27.83	6	9.83	28	7.67
4/3/2011	12:00:00 AM	57 31.111' N	21 22.916' E	2	11.5	0	0	0	0
4/4/2011	12:00:00 AM	57 24.164' N	21 32.865' E	4	47.67	2	0.33	3	1.33
4/5/2011	12:00:00 AM	57 24.163' N	21 32.864' E	1	6	0	0	0	0
4/6/2011	12:00:00 AM	56 30.053' N	18 58.394' E	3	31.17	0	0	0	0

Figure 95: SWOMS Data table

Upon opening the form, no data appears. In order to populate the table with data, select an IMO number from the dropdown menu at the top of the screen and enter the desired date range by filling in the Start and End Dates just below that dropdown menu.

The "Vessel Details" button in the top-right-hand corner of the screen brings up the "Vessel Details" form (see Section K2.3.5), which can be used to modify the details of the vessel, and also gives a link to view and edit the tank data (see Section K2.3.4). The "Add Data" button in the top-right-hand corner of the screen brings up the "Add SWOMS Data" form (see Section K2.3.2) to manually add SWOMS data. The "Home" button in the top-right-hand corner of the screen returns to the Main Menu.

K2.4.3 Sounding Book Data

The "SoundingBook Data" form (see Figure 96 for a portion), which can be accessed by clicking the "Sounding Book Data" button on the Main Screen, shows the sounding data included in the database. The table includes the date, time, location, and instantaneous tank levels and volumes.

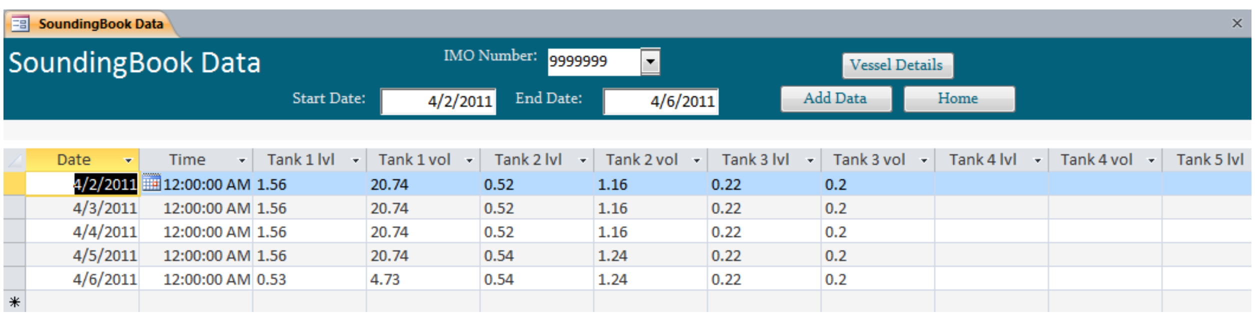


Figure 96: Sounding Book Data form

Upon opening the form, no data appears. In order to populate the form with data, select an IMO number from the dropdown menu at the top of the screen, and enter the desired date range by filling in the Start and End Dates in the spaces just below that dropdown menu.

The "Vessel Details" button in the top-right-hand corner of the screen brings up the "Vessel Details" form (see Section K2.3.5), which can be used to modify the details of the vessel, and also gives a link to view and edit the tank data (see Section K2.3.4). The "Add Data" button in the top-right-hand corner of the screen brings up the "Add Sounding Book Data" form (see Section K2.3.3) to manually add sounding data. The "Home" button in the top-right-hand corner of the screen returns to the Main Menu.

K2.5 Generating Reports

K2.5.1 Quick Reports Using the Main Menu

The Main Menu gives the option to generate reports solely based on the IMO number and a date range. The ability to do this is contained in the Quick Reports, which is the left-most section of the Main Menu (see Figure 85, p.181). In order to do this, select the IMO number from the dropdown menu, and then enter the desired date range in the Start and End Date spaces. The following reports can then be generated or printed:

- SWOMS Anomaly Report, which looks within the SWOMS data for anomalies (see Section K2.5.3);
- SWOMS: Sounding Comparison, which compares the SWOMS data to the sounding data, looking for anomalies (see Section K2.5.5); and
- Sounding Anomaly Report, which looks within the sounding data for anomalies (see Section K2.5.4).

Note that Access will be unable to generate a report if a report of the same type is already open. Also while reports generate all the necessary data sometimes information gets cut off while reviewing the report in Access. Therefore it is recommended for most analysis to print the reports or export them as PDFs as discussed in section 2.5.6.

K2.5.2 Detail Reports

More detailed reports can be generated or printed using the "Detail Reports" form (see Figure 97), which can be accessed through the Main Menu.

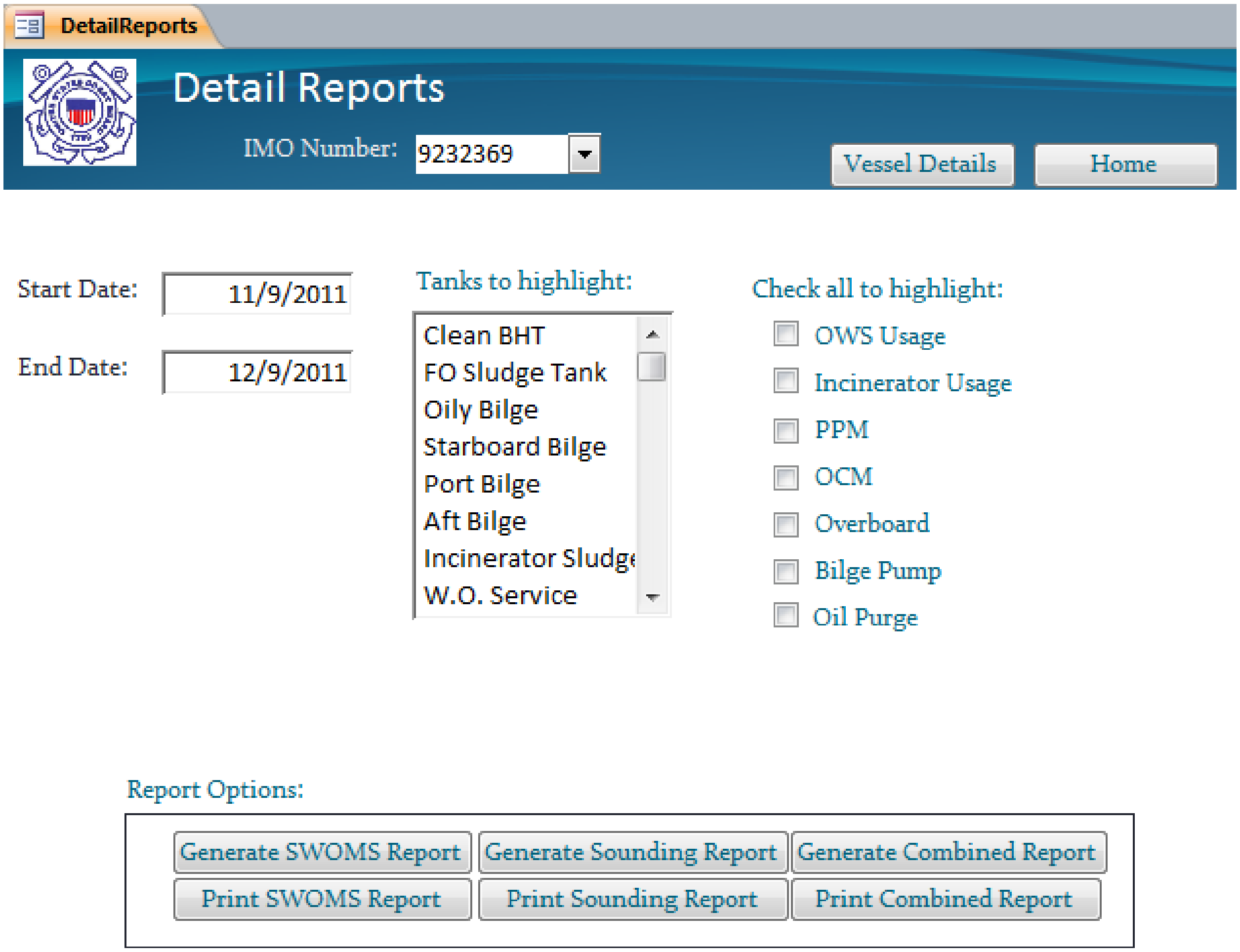


Figure 97: Detail Reports form

The following types of reports can be generated:

- SWOMS Anomaly Report, which looks within the SWOMS data for anomalies (see Section K2.5.3);
- Sounding Anomaly Report, which looks within the sounding data for anomalies (see Section K2.5.4); and
- Combined Report, which compares the SWOMS data to the sounding data, looking for anomalies (see Section K2.5.5).

Note that Access will be unable to generate a report if a report of the same type is already open.

These detailed reports are essentially the same as the quick reports, except that they give the user the opportunity to exclude unwanted information, giving the user the ability to select which tanks and what SWOMS system information is included.

Once on the “Detail Reports” form, select the IMO number for the vessel from the dropdown menu at the top of the screen, and then enter the desired date range in the Start and End Date spaces. To select which tanks to generate data for, highlight the desired tank descriptions from the list of tanks. To select or deselect a tank, simply click on its name in the list. In order to select which shipboard environmental system information is included, check the items in the list to the right of the tanks list (i.e. OWS Usage, Incinerator Usage, PPM, OCM, Overboard, Bilge Pump, and Oil Purge). The checked items will be included.

K2.5.3 SWOMS Anomaly Report

The SWOMS Anomaly Report (see Figure 98), which can be generated using Quick Reports (see Section K2.5.1) or Detail Reports (see Section K2.5.2), looks for anomalies within the SWOMS data.

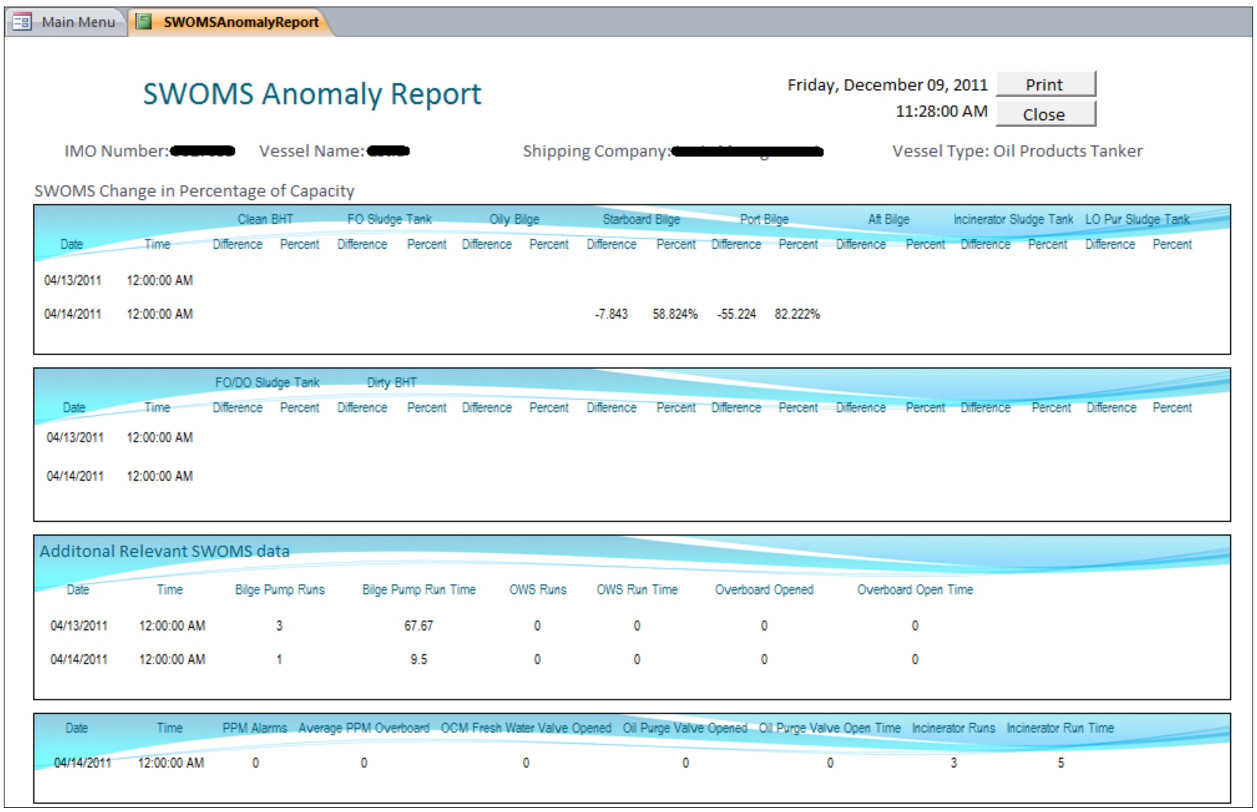


Figure 98: SWOMS Report for Two Days of Data

Contained within the header for this report is the date and time that the report is generated, as well as information taken from the Vessel Tanks table (see Section K2.3.4) including the IMO Number, Vessel name, Company, and Vessel Type. Located in the upper right-hand corner is a button that will allow the user to print the report as well as one for closing it. Printing the report in this method will not give the user any preview of what will be printed.

Located below the header, the upper portion of this report, entitled "SWOMS Change," focuses on finding the anomalies within the SWOMS data. This finds the days where the tank volume decreases by over 30% of the total tank capacity and prints the volumetric difference as well as the percentage difference as compared to the previous day for each of these days. It is divided up into two sections, one above the other, which look at 8 tanks at a time to prevent the report from being too wide.

The lower portion, entitled "Additional Relevant SWOMS data" looks through the basic shipboard SWOMS data in order to find when the on-board systems were run, such as the OWS (Oil Water Separator) or bilge pump, to facilitate quick interpretation of the data. This portion is also divided into two sections, one above the other.

K2.5.4 Sounding Anomaly Report

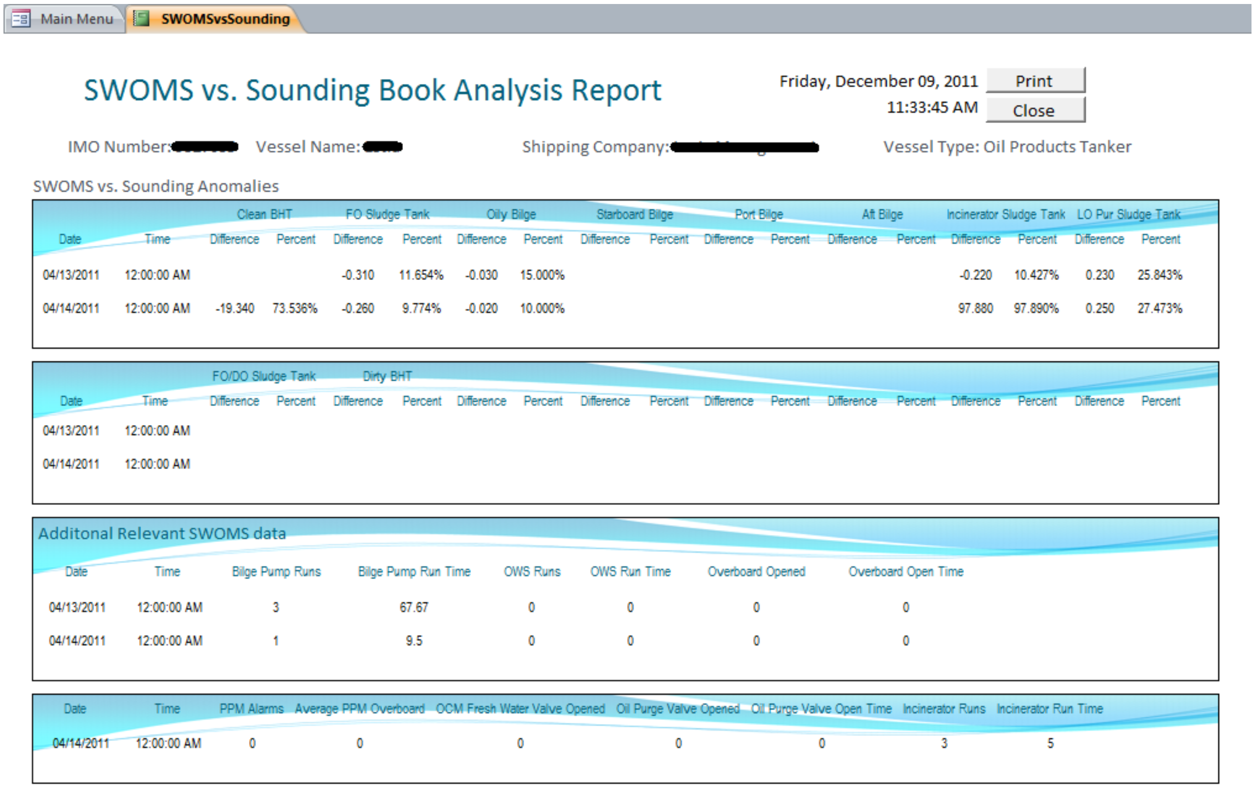
The Sounding Book Anomaly Report, which can be generated using Quick Reports (see Section K2.5.1) or Detail Reports (see Section K2.5.2), looks for anomalies within the sounding data. This report has the same format as the SWOMS report (see Figure 98). Contained within the header for this report is the date and time that the report is generated, as well as information taken from the Vessel Tanks table (see Section K2.3.4) including the IMO Number, Vessel name, Company, and Vessel Type. Located in the upper right-hand corner is a button that will allow the user to print the report and another for closing it.

Located below the header, the upper portion of this report, entitled "Sounding Book Change Subreport," focuses on finding the anomalies within the Sounding data. This finds the days where the tank volume decreases by greater than 30% of the total tank capacity and prints the volumetric difference as well as the percentage difference as compared to the previous day for each of these days. It is divided up into two sections, one above the other, which look at 8 tanks at a time to prevent the report from being too wide.

The lower section, entitled "Additional Relevant SWOMS data" looks through the basic shipboard SWOMS data in order to find when the on-board systems were run, such as the OWS (Oil Water Separator) or bilge pump, to facilitate quick interpretation of the data. This portion is also divided into two sections, one above the other.

K2.5.5 Combined Report

The Combined Report (see Figure 99), which can be generated using Quick Reports (see Section K2.5.1) or Detail Reports (see Section K2.5.2), looks for substantial differences between the SWOMS and sounding data.


Main Menu | SWOMSvsSounding

SWOMS vs. Sounding Book Analysis Report

Friday, December 09, 2011 Print
11:33:45 AM Close

IMO Number:[redacted] Vessel Name:[redacted] Shipping Company:[redacted] Vessel Type: Oil Products Tanker

SWOMS vs. Sounding Anomalies

		Clean BHT		FO Sludge Tank		Oily Bilge		Starboard Bilge		Port Bilge		Aft Bilge		Incinerator Sludge Tank		LO Pur Sludge Tank	
Date	Time	Difference	Percent	Difference	Percent	Difference	Percent	Difference	Percent	Difference	Percent	Difference	Percent	Difference	Percent	Difference	Percent
04/13/2011	12:00:00 AM			-0.310	11.654%	-0.030	15.000%							-0.220	10.427%	0.230	25.843%
04/14/2011	12:00:00 AM	-19.340	73.536%	-0.260	9.774%	-0.020	10.000%							97.880	97.890%	0.250	27.473%

		FO/DO Sludge Tank		Dirty BHT													
Date	Time	Difference	Percent	Difference	Percent	Difference	Percent	Difference	Percent	Difference	Percent	Difference	Percent	Difference	Percent	Difference	Percent
04/13/2011	12:00:00 AM																
04/14/2011	12:00:00 AM																

Additonal Relevant SWOMS data

Date	Time	Bilge Pump Runs	Bilge Pump Run Time	OWS Runs	OWS Run Time	Overboard Opened	Overboard Open Time
04/13/2011	12:00:00 AM	3	67.67	0	0	0	0
04/14/2011	12:00:00 AM	1	9.5	0	0	0	0

Date	Time	PPM Alarms	Average PPM Overboard	OCM Fresh Water Valve Opened	Oil Purge Valve Opened	Oil Purge Valve Open Time	Incinerator Runs	Incinerator Run Time
04/14/2011	12:00:00 AM	0	0	0	0	0	3	5

Figure 99: Combined Report for Two Days of Data

Contained within the header for this report is the date and time that the report is generated, as well as information taken from the Vessel Tanks table (see Section K2.3.4) including the IMO Number, Vessel name, Company, and Vessel Type.

Located in the upper right-hand corner is a button that will allow the user to print the report and another for closing it. Located below the header, the top section of this report, entitled “SWOMS vs. Sounding Anomalies,” focuses on comparing the SWOMS and sounding book data and looking for anomalies. This finds the days where the tank volume from SWOMS is off by more than five percent from the sounding book.

The lower section, entitled “Additional Relevant SWOMS data” looks through the basic shipboard SWOMS data in order to find when the on-board systems were run, such as the OWS (Oil Water Separator) or bilge pump, to facilitate quick interpretation of the data.

K2.5.6 Printing or Exporting Reports

There are several ways to print these reports. If the user would like to print them without first viewing them they can simply click the print report button for any of the three reports on either the Main Menu or the Detail Reports form (see Section K2.5.2). On the Main Menu these buttons are located in the quick reports section underneath their respective generate report button. On the Detail Reports form, they are located in the report options underneath the generate report buttons. If the user would like to review a report before printing they can use the generate report buttons to generate a report, and then print it from the button on the report (this will not print the command buttons).

In order to export the report to a PDF, simply select the Acrobat tab in the Access toolbar when a report is open and click create PDF (see figure). Access will prompt the user for a save location and name for the file and then when save is pressed, will create and save a pdf of the report in the specified location.

K3. Parsing Program Maintenance

The parsing program will need to be updated to match any future design templates or the addition of more command line arguments. The program was finally compiled using Microsoft Visual C++ (VC++), a component of Microsoft Visual Studio 2010. If changes are made to the source code, it is recommended that the program get recompiled using the same compiler to avoid issues. However, the program is written entirely in C. By removing dependency on windows.h and replacing that with stdlib.h, the program should properly compile from a Linux shell. The code used to create this program is well-commented and should be easily understood and modified by a programmer.

K3.1 Rebuild swomsparser Executable from Source Code

This section details how to recompile the parsing program "swomsparser.exe" from its source code using Microsoft Visual Studio 2010. This process requires the files "targetver.h" and "swomsparser.cpp". Any code modification should be implemented by someone familiar with the coding language C or C++.

K3.1.1 Create Project

The first step towards recompiling this program is to create a new project within Visual Studio. This can be done by using the keyboard shortcut Ctrl-Shift-N, or by clicking on File > New > Project. This brings up a window allowing the user to make specifications for the project (see Figure 100).

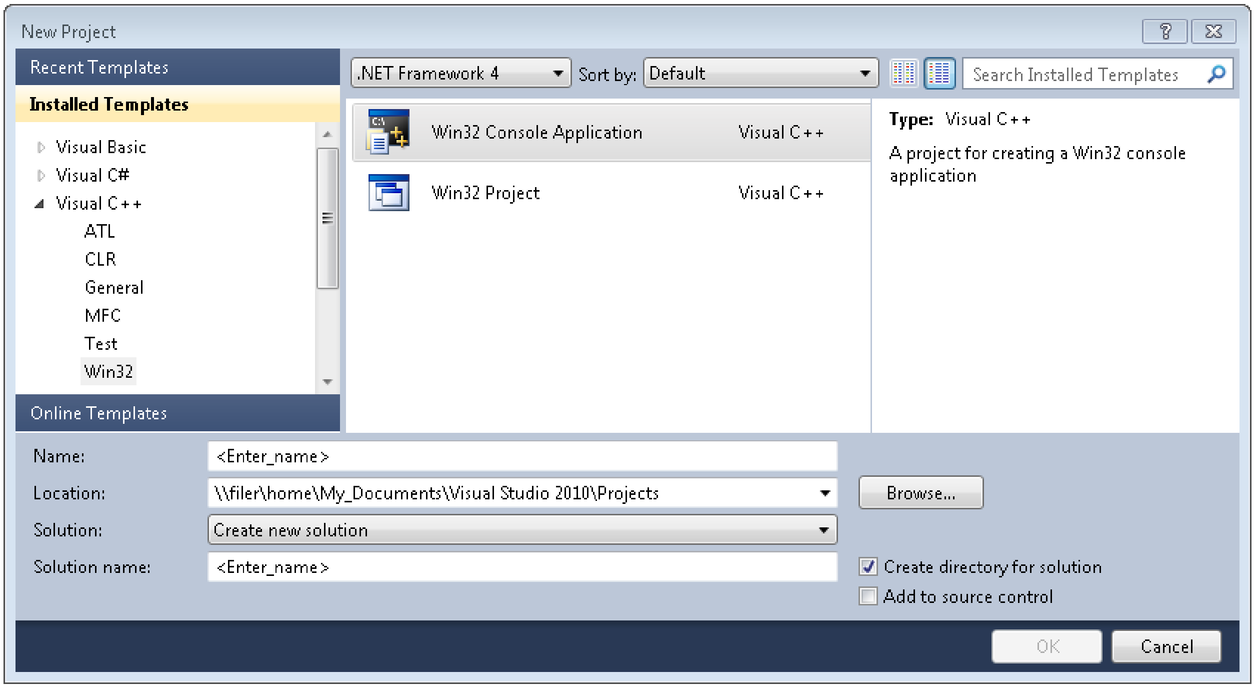


Figure 100: New Project dialog

The type of project is a Win32 Console Application coded in Visual C++. In order to select this option, click on Visual C++ > Win32 in the Installed Templates section of the window. Ensure that Win32 Console Application is selected in the main section of the window. Next, enter a name for the project. "Create directory for solution" should be checked and "Add to source control" should be unchecked.

Once satisfied, click OK, which should bring up the Application Wizard. Click Next to advance to the Application Settings section (see Figure 101).

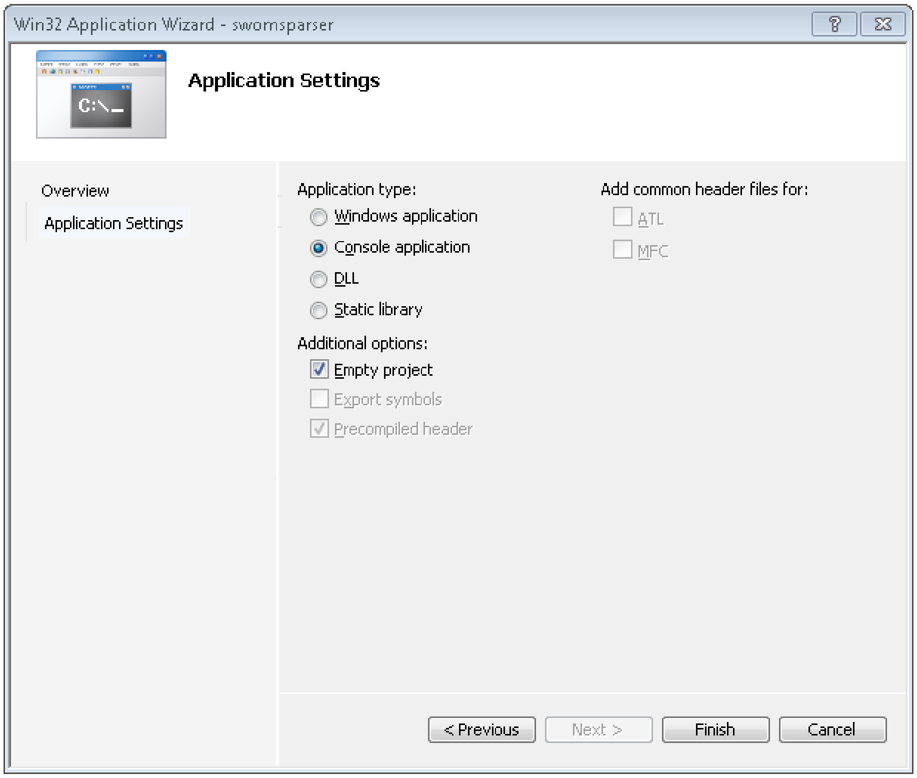


Figure 101: Application Settings

In the Application Settings, set the Application type to Console Application and check Empty Project under Additional Options. Click Finish to complete the creation of the project.

K3.1.2 Set Up Project

The Solution Explorer (see Figure 102) is an interface for viewing and managing the code contained in the projects. If the Solution Explorer interface is not automatically visible, use keyboard shortcut Ctrl-Alt-L or click on View > Solution Explorer.

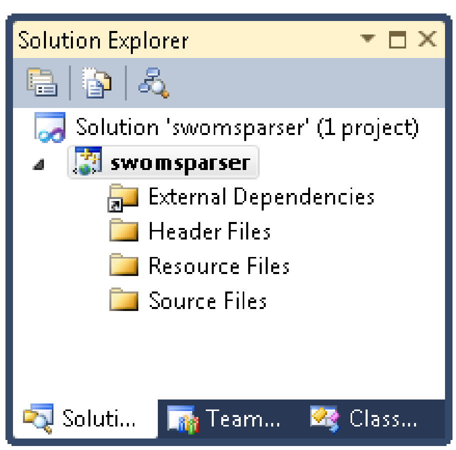


Figure 102: Solution Explorer

Right click on the project name in the Solution Explorer interface and click on “Open Folder in Windows Explorer”. Once the folder is open, copy the code files “targetver.h” and “swomsparser.cpp” into that folder. Once this is complete, the folder can be closed.

The Solution Explorer can then be used to add the files to the project. To add the header file “targetver.h”, right click on Header Files, then click on Add > Existing Item. Select “targetver.h” and then click Add. To add the source code “swomsparser.cpp”, right click on Source Files, then click on Add > Existing Item. Select “swomsparser.cpp” and then click Add. At this point, the Explorer should look like Figure 103.

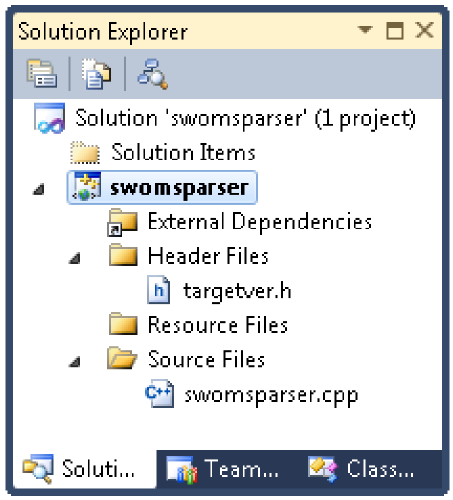


Figure 103: Solution Explorer Including Files

In order to set the properties for the project, right click on the project name in the Solution Explorer and click on Properties (see Figure 104).

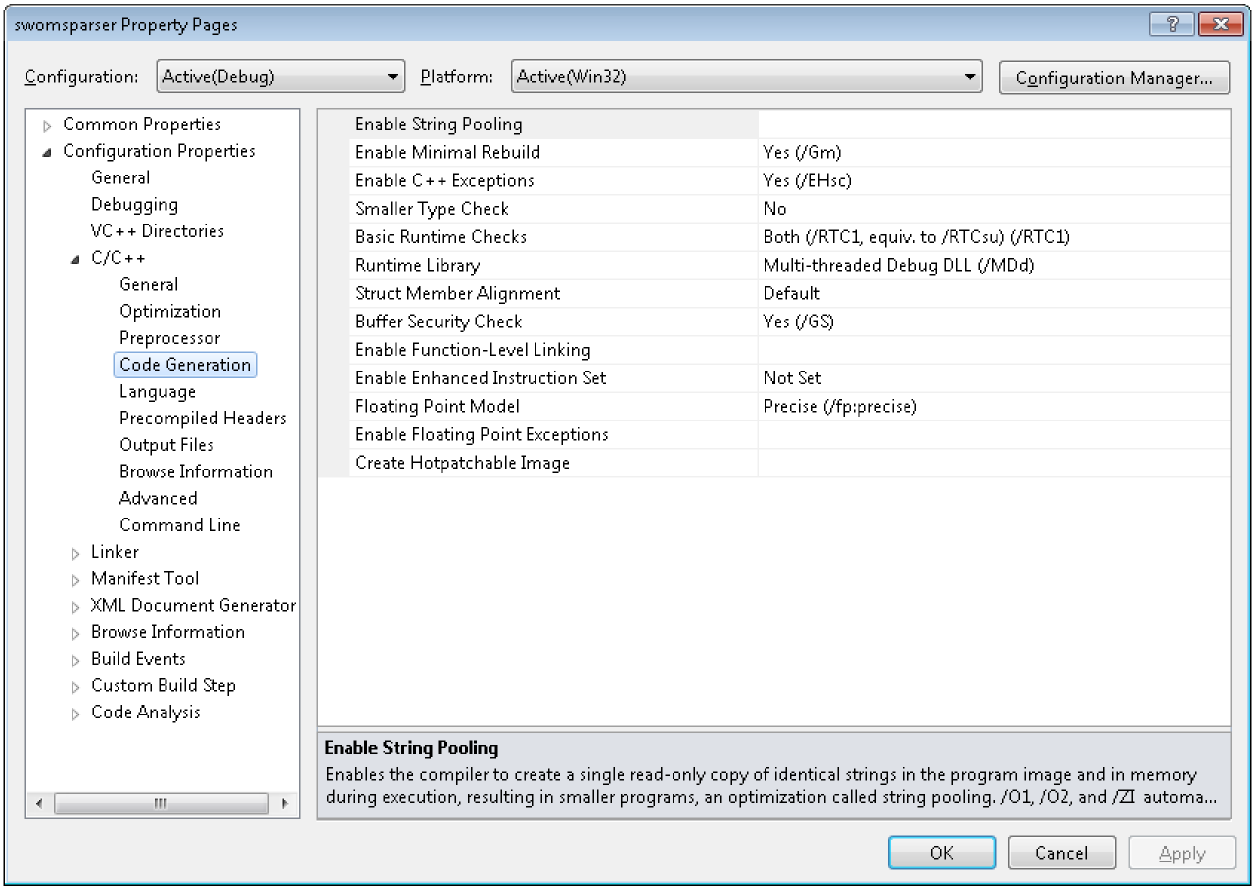


Figure 104: Project Properties

Under Configuration Properties > C/C++ > Code Generation, set the Runtime Library property to "Multi-threaded (/MT)". Under Configuration Properties > C/C++ > Preprocessor, add ";_CRT_SECURE_NO_WARNINGS" to the end of the pre-existent list of Preprocessor Definitions. Then, click "OK". After these changes are made, save the project using keyboard shortcut Ctrl-S or by clicking on File > Save.

K3.1.3 Modify and Build Code

To open the code, double-click on "swomsparser.cpp" from under the Source Files in the Solution Explorer. This should bring up the code in a window that will allow editing. After making any changes, build the code using Ctrl-Shift-B or by clicking on Build > Build Solution. The output of the build should show up within its own interface with the results of the build and any warnings. Figure 105 below shows a successful build output.

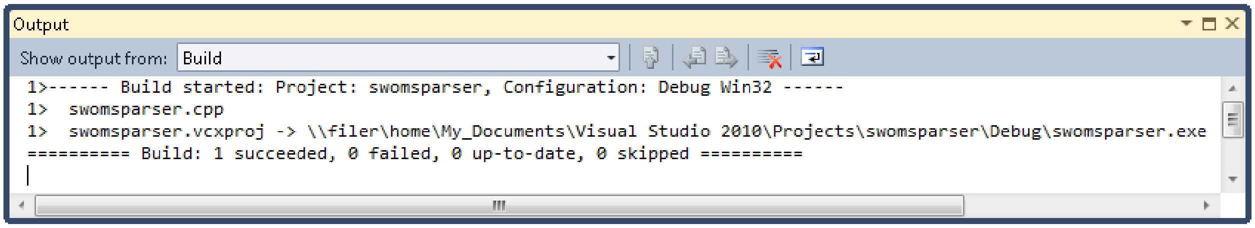


Figure 105: Build Output

K4. Access Database Maintenance

The database will need to be updated for any future expansions and modifications of the environmental compliance programs. Additionally if there are ever any problems or errors with the current database, a user familiar with Access will have to look at the code to make the necessary changes or fixes. The visual basic code, which is used for basic form functionality as well as the basics of some of the data updating and report generation, is fully documented and commented within the code. If there are any errors with this code, a user can look to the comments for a full explanation of how the various functions work, and therefore they will not be explained here. What follows are descriptions of all of the Access queries and the relevant SQL code, so that if future modifications are necessary it should be simple to understand what the queries currently do.

Capacity Query:

The capacity query simply takes all the tank capacities from the tank data table, and compiles them based on the user selected IMO number from the detail reports form.

Delete SWOMS and Delete Sounding:

The delete SWOMS query selects all the records from the SWOMS data table based on an IMO number and date range on the delete records form, and deletes those records. The Delete Sounding query does the same.

SWOMS Analysis and Sounding Analysis:

The SWOMS analysis query simply gathers all the data from the SWOMS table required for analysis for a user selected IMO number and date range so that this information can be easily used by the other queries. The Sounding Analysis query does the same for the sounding book data.

Sorted Query and Sorted Query 2:

The two sorted queries gather the fields from the SWOMS Analysis that are not related to the tank volumes. These are gathered separately so they can be easily used to generate data for the two Sorted Data subreports.

SWOMS % CAP and Sounding % CAP:

This query calculates the percentage of the total tank capacity each tank is at for every record in SWOMS Analysis. It takes the tank capacity from the capacity query and calculates the percent capacity using this formula:

IIf(Nz([Capacity Query].[Tank01 Cap])=0,Null,((Nz([SWOMS Analysis].[Tank01 VOL])/Nz([Capacity Query].[Tank01 Cap]))))

This ensures that Access does not divide by zero and accounts for possible null values with the Nz() method. The query has a field for each of these calculations, one for each tank. This query also collects all the tank volumes because in order for SWOMS change to work properly it needs to gather all its data from a single query. The Sounding query does the same but for Sounding Analysis data.

SWOMS Change and Sounding Change:

The SWOMS Change query takes the calculated capacities as well as the volumes from the SWOMS % CAP query, and does further calculations with this data. It first calculates the change in percentage between two records using this formula:

([SWOMS % CAP].[Tank 1]-(SELECT TOP 1 [Tank 1] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([SWOMS % CAP].[Tank 1]-Q.[Tank 1])<-.3)))

This formula selects the current record, as well as using "SELECT TOP 1" to select the previous record, and checks whether the date is for the previous day. It then checks that the difference between the two percentages is greater than 30%, which is the threshold for anomalies that should be highlighted. The query then calculates the difference between the tank volume for the same two records, if the dates are concurrent and the percentage change is greater than 30% using this formula:

([SWOMS % CAP].[Tank01 VOL]-(SELECT TOP 1 [Tank01 VOL] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND ([SWOMS % CAP].[Tank 1]-(SELECT TOP 1 [Tank 1] FROM [SWOMS % CAP] AS Q WHERE (DateDiff("d", [SWOMS % CAP].[Date (dd/mm/yyyy)], Q.[Date (dd/mm/yyyy)]) = -1) AND ([SWOMS % CAP].[Date (dd/mm/yyyy)] >Q.[Date (dd/mm/yyyy)]) AND (([SWOMS % CAP].[Tank 1]-Q.[Tank 1])<-.3)))))

Both of these are then repeated for all 16 tanks and Sounding Change does the same.

SWOMS Data Query and SoundingBook Data Query:

These two queries simply gather the data from their respective tables, based on an IMO number and date range on the SWOMSData and SoundingBook data forms respectively, so that this data can be properly displayed on the forms.

SWOMS VS Sounding:

This query takes the data from the SWOMS Analysis and Sounding Analysis queries and calculates the difference between these values. Using this formula, the query calculates the difference in volume from the two queries:

[SWOMS Analysis].[Tank01 VOL]-(SELECT TOP 1 [Tank01 VOL] FROM [Sounding Analysis] WHERE DateDiff("d", [SWOMS Analysis].[Date (dd/mm/yyyy)], [Sounding Analysis].[Date (dd/mm/yyyy)]) = 0 AND [SWOMS Analysis].[Time (UTC; 00:00:00)] = [Sounding Analysis].[Time (UTC; 00:00:00)] AND ((([SWOMS Analysis].[Tank01 VOL]-[Sounding Analysis].[Tank01 VOL])/[SWOMS Analysis].[Tank01 VOL]) >.05 OR (([Sounding Analysis].[Tank01 VOL]-[SWOMS Analysis].[Tank01 VOL])/[Sounding Analysis].[Tank01 VOL]) >.05))

Then using the values from this first formula, the query calculates whether this difference is greater than five percent with this formula:

IIf([Diff]>0,([Tank01 VOL]-([Tank01 VOL]-[Diff]))/[Tank01 VOL],(([Tank01 VOL]+(-1*[Diff]))-[Tank01 VOL])/([Tank01 VOL]+(-1*[Diff])))

This formula check whether the difference is greater than zero or not and calculates the percent difference accordingly. The query then does the same for all 16 tanks.

Appendix L: Exchange 2007 Mailbox Setup

Environmental Compliance: Acquisition, Storage, and Analysis of Waste Oil Data

Exchange 2007 Mailbox Setup

This guide is intended to be used in conjunction with the data transmission system and database developed by WPI students for the storage and analysis of shipboard waste data.

Developed For:
United States Coast Guard
Office of Vessel Activities (COMDT CG-543), in conjunction with the
Office of Investigations and Casualty Analysis (COMDT CG-545)

Prepared & Edited by:
Patrick Brodeur
Renée Lanza
Elizabeth Morris
Edward Osowski

Table of Contents

Table of Figures

L1. Introduction

This guide is intended to be used in conjunction with the data transmission system and ECP Database developed by WPI students for the storage and analysis of shipboard waste data. It offers multiple ways of implementing the Outlook/Exchange integration component of the data transmission system, including a test setup using a personal mailbox, a semi-automatic system using a public folder and a macro, and an automated system using a public folder and script running on the Exchange server.

Within this document, some technical terminology is used and there are references to a data transmission system and the "ECP Database". As briefly stated above, this document is meant to be used with the system developed by WPI students in the Fall of 2011 for the acquisition, storage, and analysis of SWOMS (Special Waste Oil Monitoring System) and sounding log data from vessels on environmental compliance plans (ECP's). The specific elements in this user manual refer to the Outlook/Exchange components of the standard data submission plan included in the system. Please refer to section 5.3 New Standard Transmission Method.

Some helpful terms:

- swomsparser – program developed as part of the project for parsing data out of SWOMS reports
- Outlook – the Microsoft Office program for email and other organization tasks
- Exchange – the Microsoft server software that handles email, calendars, and various other communication information
- Visual Basic for Applications (VBA) – scripting language used by Microsoft Office to programmatically perform many tasks; used here to perform actions on emails stored on a Microsoft Exchange server

L2. Outlook Test Setup

In order to test the SWOMS email submission system, a personal mailbox was set up using an Outlook rule and VBA script. This setup may be used to demonstrate or test the behavior of the data transmission system from receiving a SWOMS email to the data's storage as spreadsheets. The similarity between this setup and that on the server lends itself to allow testing on a local machine without making experimental modifications to the script on the server.

L2.1 Creating a File Structure

Before any Outlook/Exchange modifications are made, the working directories should be created. The code modules use a file tree similar to the one shown below in Figure 106. In this screenshot, the folder is on the 'M' drive of the computer. The way the parsing program is written, it will always create and modify the SWOMS_DATA spreadsheet in the same folder as itself but will create vessel-specific tables in the same folder as the input (source) data.

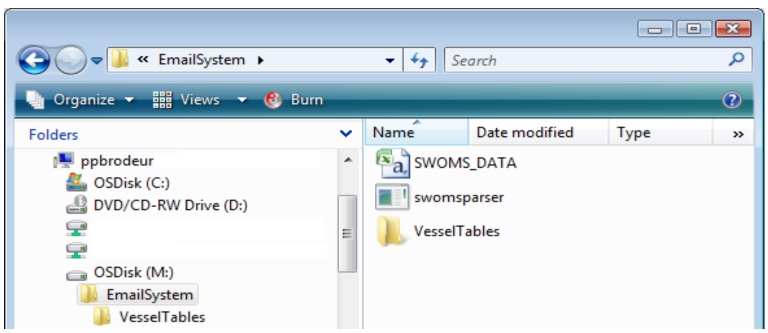


Figure 106: File Structure Example

The VBA script included uses the 'U' drive, which by default maps to a user's personal file share. The default structure in the included script uses "U:\EmailSystem\" as its primary working directory and "VesselTables\" as its vessel table subdirectory. In this setup, vessel-specific tables will be placed into the "VesselTables" directory, and the swomsparser program and SWOMS_DATA table will remain in the "EmailSystem" folder. More on the VBA script options with regard to folder structure may be found later in this document.

L2.2 Creating the Outlook Rule

The next stage is to create Outlook rules to process emails when they arrive. Before making the rule, two tasks must be accomplished. The first of these pre-rule tasks is to create a folder to store the sample SWOMS emails. This folder can be named anything memorable to the user. The second pre-rule task is to create a placeholder script as described below in 0 L2.2.1 Create Script Placeholder.

L2.2.1 Create Script Placeholder

To create and add the script placeholder, open the Visual Basic for Applications editor (see Figure 107). This can be done by pressing the keyboard shortcut Alt-F11 or by going to Tools > Macro > Visual Basic Editor in the top menu.

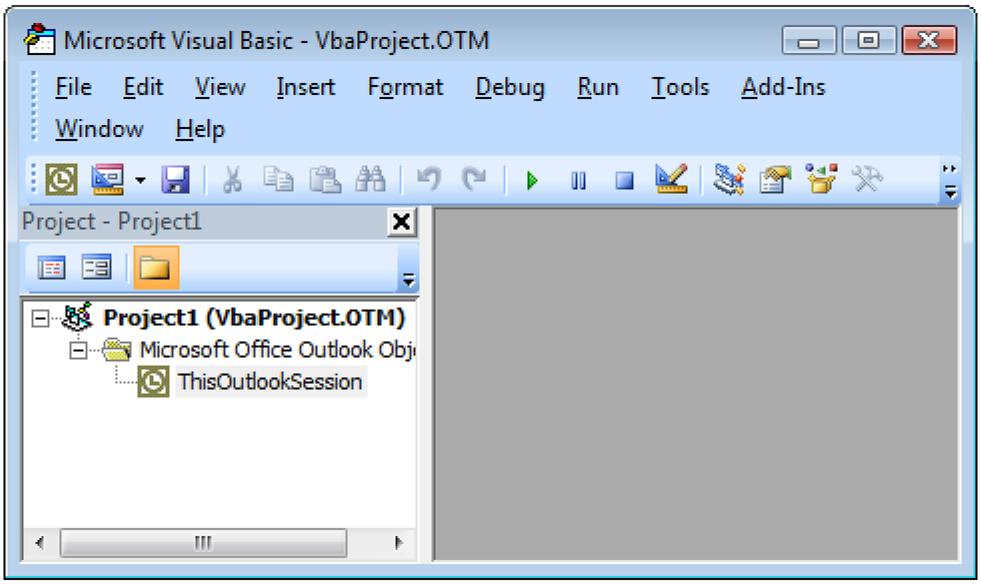


Figure 107: Opened VBA Editor

With the VBA editor opened, the "ThisOutlookSession" code module can then be opened. On the left-hand side, expand Project1 and its subfolder. "ThisOutlookSession" should appear in this list, as shown above in Figure 107. Double click to open it, and the following screen (Figure 108) should appear:

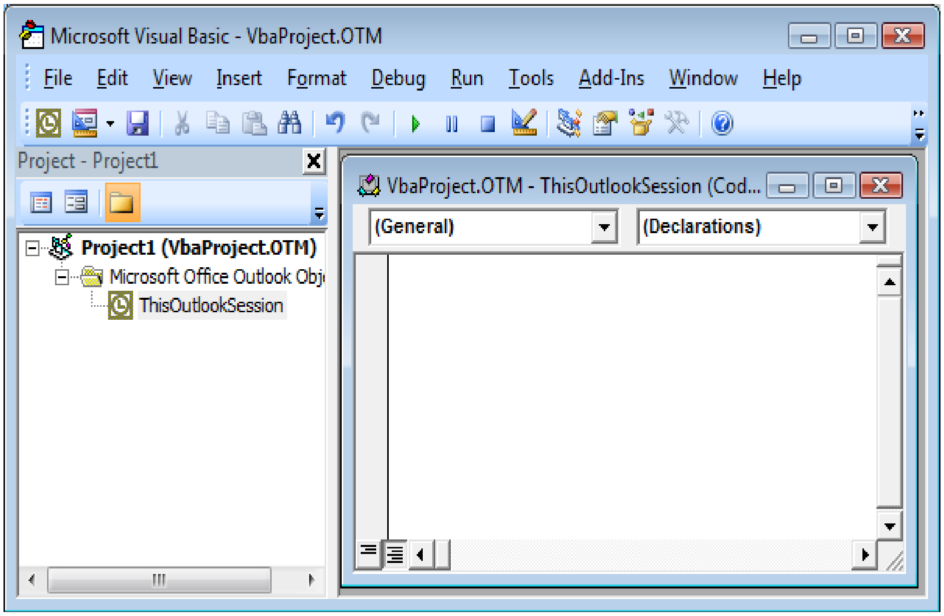


Figure 108: VBA Editor with ThisOutlookSession Opened

This code module should be empty by default. If there is any code already in place, verify with a system administrator that it is safe and belongs there. Any code with an unknown source is a significant security risk.

In the VBA editor's menu, click on Insert > Procedure. It should open up a simple dialog window (see Figure 109). Fill in the name, select "Sub" for the type parameter, and select "Public" for the scope parameter.

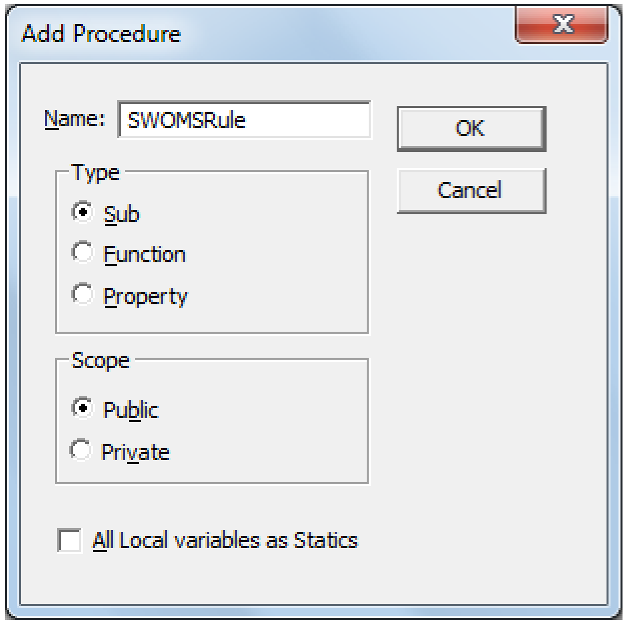


Figure 109: Add Procedure Dialog

The Add Procedure box will add in the start and end lines of the SWOMS rule. Modify the code to match the lines shown below in SWOMS Rule Code Segment 1 and save the script. The placeholder will now be a valid script to be called by a rule. The VBA editor window may be left open for later or closed.

```
Public Sub SWOMSRule(currentMail As Outlook.MailItem)
    Debug.Print "Script Run Successfully"
End Sub
```

Figure 110: SWOMS Rule Code Segment 1

L2.2.2 Making the Rule

To begin making the rule, go to Tools > Rules & Alerts from the menu bar. Click "New Rule…" to begin, and follow the steps below Figure 111.

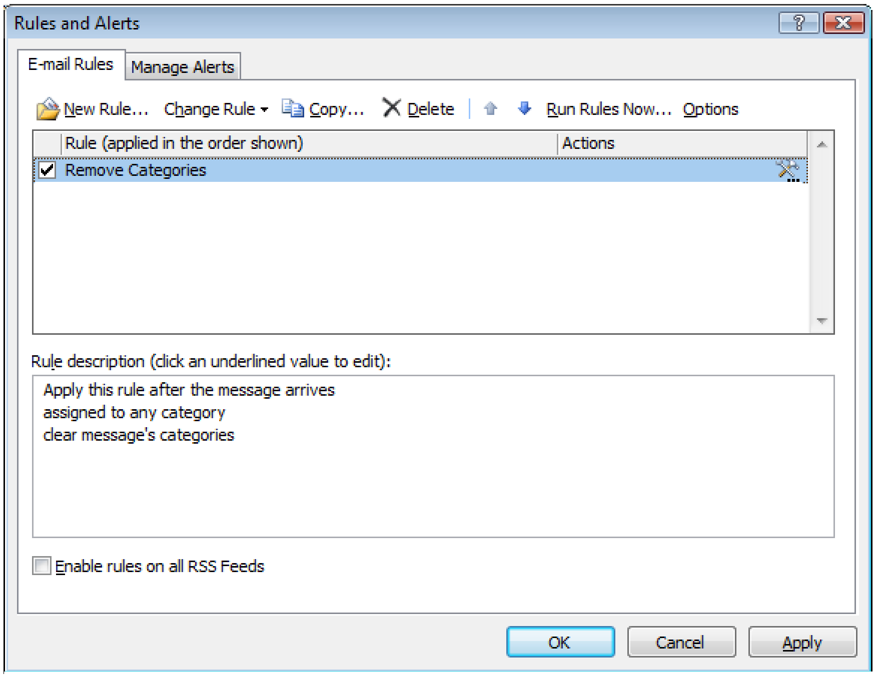


Figure 111: Rules and Alerts Main Screen

1. Select "Check messages when they arrive" under the "Start from a blank rule" heading and click "Next".
2. No conditions must be set, click "Next". Outlook may ask for confirmation; yes, this rule should be run on every email that is received.
3. Check the "run a script" box.
4. Click on "a script" and select the VBA script created earlier.
5. Click "Next".
6. The script checks if emails are marked as complete so no further exclusions must be made using a rule, so the exclusions page can be skipped.
7. On the final page, give the rule a descriptive name and keep "Turn on this rule" checked.
8. Click "Finish" to be returned to the Rules and Alerts dialog.

At this point, Outlook will pop up a warning message saying, "This rule is a client-only rule, and will process only when Outlook is running." As the warning indicates, this setup will only work when a user has Outlook running. For testing purposes, this is not an issue.

L2.3 Adding Visual Basic Script

Re-open the Visual Basic for Applications editor. This can be done by pressing the keyboard shortcut Alt-F11 or by going to Tools > Macro > Visual Basic Editor in the top menu bar. At this point, two options are possible. The first option is to replace the previously-added code in ThisOutlookSession with the code in the accompanying file "ThisOutlookSession.txt". For a better understanding of what is happening in the script, or if the accompanying file is missing, continue reading this section. Modifications may be necessary before the code works properly, depending on the computer's specific setup.

L2.3.1 Adding Code

In ThisOutlookSession, remove the "Debug.Print" line added earlier; it isn't necessary. Following the first line ("Public Sub…"), add the lines shown below in SWOMS Rule Code Segment 2. This section declares variables and sets the working directory (EmailSystem in the earlier example) and the vessel table subdirectory.

```
Dim vFile, vFile2, workingDir, vslTblSubdir As String
Dim i As Integer

workingDir = "U:\EmailSystem\"
vslTblSubdir = "VesselTables\"
```

Figure 112: SWOMS Rule Code Segment 2

SWOMS Rule Code Segment 3 below shows the "If" conditional that tests the email for the keywords that it is a SWOMS email, and also checks to make sure there isn't a "Completed" flag. The three lines following the "if" statement set the mail item to be plaintext and mark it as read.

```
If (    InStr(currentMail.Subject, "Daily Report") <> 0) And _
        currentMail.FlagStatus <> olFlagComplete Then

        currentMail.BodyFormat = olFormatPlain
        currentMail.UnRead = False
        currentMail.Save
```

Figure 113: SWOMS Rule Code Segment 3

SWOMS Rule Code Segment 4 creates a temporary text file with a randomly-generated number as part of its name and copies the body of the email into the text file. This text file will be processed by the swomsparser program.

```
        ' Random name for temporary text file
        vFile = vslTblSubdir & "TMP" & (Rnd * 1000) & ".txt"
        ' Put body of the email into a temporary text file
        Open (workingDir & vFile) For Output As #1
        Print #1, currentMail.Body
        Close #1
```

Figure 114: SWOMS Rule Code Segment 4

The next code segment creates a batch file that uses the Windows command line interpreter to run the swomsparser program and clean up the text file and itself. This section is lengthy, but comments (shown in green) explain what the code is doing.

```
        ' Create batch file to process data and clean up intermediate files
        ' Random name for temporary batch file
        vFile2 = workingDir & "TMP" & (Rnd * 1000) & ".bat"
        Open vFile2 For Output As #2
        ' Go to the directory for the SWOMS parser

        If (Left(workingDir, 2) = "\\") Then
                ' If it's a network directory then make a pushd drive map
                Print #2, "pushd " & workingDir
        Else
                ' Otherwise use the directory as-is
                Print #2, Left(workingDir, 2)
                    Print #2, "cd " & workingDir
        End If
        ' Run the parser & wait for completion
        Print #2, "start /wait swomsparser " & ".\" & vFile
        ' Delete intermediates
        Print #2, "del .\" & vFile       ' Instruction to delete input file
        Print #2, "del %0"               ' Instruction to delete self
        Close #2
```

Figure 115: SWOMS Rule Code Segment 5

The last section of code before the "End Sub" statement is seen below in Code Segment 6. The line beginning with "Shell" runs the batch file, and the rest of the code sets the email to "Complete". The script is complete at this point, and the rule can be run.

```
        ' Run the batch script outlined above
        Shell (vFile2)

        currentMail.FlagStatus = olFlagComplete
        currentMail.Save
End If
```

Figure 116: SWOMS Rule Code Segment 6

L3. Implementation on Public Folder

The ideal solution to the email integration system is to have a shared Outlook folder (a public folder) that automatically runs the parsing program on emails as they come in. This functionality was not able to be completed, but a script was written to be implemented when it is possible. See section 0 for details.

L3.1 Creating Public Folder

> A public folder for the ECP Database has already been created at:
> \HQS\HQS-PF-fldr-CG-5\HQS-PF-fldr-CG-543\CGECP
> Do not create another public folder unless necessary.

Before anything else can be set up, a public folder must be created. This can be done by going to File > New > Folder. In the Create New Folder dialog (see Figure 117), fill in the folder name, leave it set to "Mail and Post Items", and then select where to place the folder.

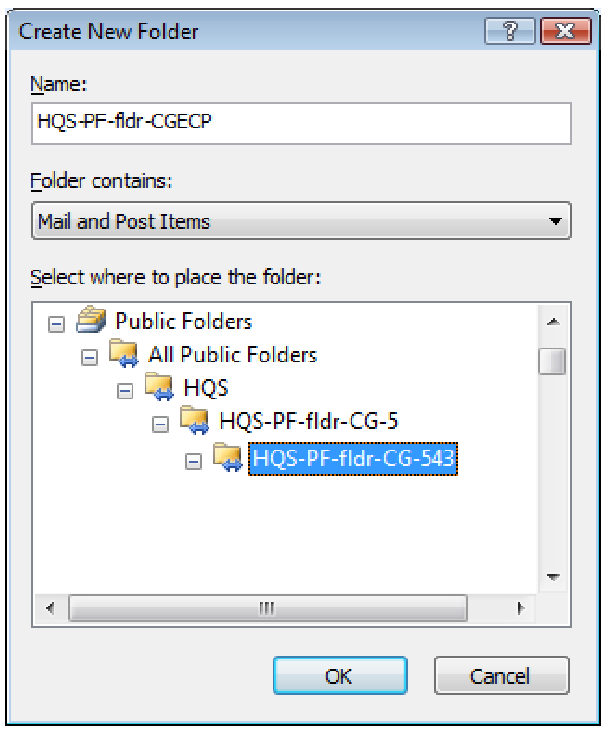


Figure 117: Adding Public Folder

Once the public folder is created, there should be separate subfolders created to hold processed SWOMS emails and document (sounding log, ORB) emails. Within these folders, it may be desirable to create subfolders on a per-ship basis, though this is probably unnecessary. The main folder will serve as triage and error control, where emails will enter and be directed and/or processed either automatically or by user intervention.

L3.2 Outlook Macro (Semi-Automated)

Partial automation is achieved through the use of an Outlook Macro. First, the macro must be added into Outlook's VBA project. Sections 0 and 0 are two different options for putting the macro's code into the VBA project, and section 0 describes how to practically use the macro.

L3.2.1 By Importing VBA Module

Importing the VBA macro as a VBA module is the easiest way to get it up and running immediately. Currently, the macro is set to use a shared folder on one of the Coast Guard servers. Modifications may be made to adjust for differences in how the file system is set up. At the time of this writing, no modification should be necessary for this macro to run "out of the box".

Open the Visual Basic for Applications editor. This can be done by pressing the keyboard shortcut Alt-F11 or by going to Tools > Macro > Visual Basic Editor in the top menu bar. Right click on "Project1" on the left-hand side and choose "Import File". Select the accompanying module file, "SWOMSMacro.bas". Continue to section 0.

L3.2.2 Adding Code

Open the VBA editor. Go to Insert > Module to add a new VBA script module to the VBA project. Next, add SWOMS Macro Code Segment 1. This sets up the name of the script and includes variable definitions. Except for SWOMS Macro Code Segment 1 and SWOMS Macro Code Segment 10, indentation will be shifted one tab to the left for readability.

```
Public Sub SWOMSMacro()
        Dim vFile, vFile2, workingDir, vslTblSubdir As String
        Dim currentMail As MailItem
        Dim currentItem, currentSelection As Object
        Dim i As Integer
        Dim answer As VbMsgBoxResult
```

Figure 118: SWOMS Macro Code Segment 1

SWOMS Macro Code Segment 2 sets up the directories, as shown before for the SWOMS Rule. This time the working directory is pointing to a network folder.

```
workingDir = "\\Hqs-nas-t-001\cg-5\CG-54\CG-543\CG-5432\" _
        & "ECP - Environmental Compliance Program Master File\EmailSystem\"
vslTblSubdir = "VesselTables\"
```

Figure 119: SWOMS Macro Code Segment 2

The next code segment is unique to the macro. SWOMS Macro Code Segment 3 makes a list of all emails that are currently selected. It then checks the number, and advises the user that performing large sets of emails may be very slow. The script asks the user to confirm that they want to continue, and will then either quit or continue.

```
Set currentSelection = Application.ActiveExplorer.Selection
If (currentSelection.Count > 100) Then
        answer = MsgBox("Running this macro on a large number of emails (" _
                & currentSelection.Count & ") may take a long time and could" _
                & "cause Outlook to become unstable. " & vbCrLf & vbCrLf _
                & "Are you sure you want to continue?", vbYesNo, "Warning")
        If answer = vbNo Then Exit Sub
End If
```

Figure 120: SWOMS Macro Code Segment 3

The next section of code creates the first part of the batch file. Instead of making the batch file at once as done previously, the macro adds to it up until the end.

```
' Establish batch file
' Random name for temporary batch file
vFile2 = workingDir & "TMP" & (Rnd * 1000) & ".bat"
Open vFile2 For Output As #2
' Go to the directory for the SWOMS parser
If (Left(workingDir, 2) = "\\") Then
        Print #2, "pushd " & workingDir
Else
    Print #2, Left(workingDir, 2)
    Print #2, "cd " & workingDir
End If
```

Figure 121: SWOMS Macro Code Segment 4

Once the batch file is established, the macro begins going through what is selected. The code below in SWOMS Macro Code Segment 5 tells the script to only run the rest of it if the selected object is an email. “For Each” means to go through every selected email.

```
i = 0
For Each currentItem In currentSelection
    If currentItem.Class = olMail Then
```

Figure 122: SWOMS Macro Code Segment 5

Again, indentation is being shifted for readability. The next section warns the user if the program hits a 225-email limit. This is imposed for stability and to avoid hitting a server-side restrictions that limits emails processed to 249.

```
' Server restricts max emails processed in this way to 249 per call
i = i + 1
If i >= 225 Then
        MsgBox ("This macro may only be run on 225 messages at once." & _
                vbCrLf & "Not all selected emails were processed.")
        GoTo closeandfinish
End If
```

Figure 123: SWOMS Macro Code Segment 6

Similarly to in the SWOMS RULE, SWOMS Macro Code Segment 7 checks that the email is a SWOMS email and sets the same settings (plaintext, marked as read).

```
Set currentMail = currentItem
If ((InStr(currentMail.Subject, "Daily Report") <> 0) And _
                currentMail.FlagStatus <> olFlagComplete) Then
        currentMail.BodyFormat = olFormatPlain
        currentMail.UnRead = False
        currentMail.Save
```

Figure 124: SWOMS Macro Code Segment 7

The macro then goes through the same process as before to produce a text file containing the body of the email.

```
        vFile = "VesselTables\" & "TMP" & (Rnd * 1000) & ".txt" ' Random name for temporary text file
        ' Put body of the email into a temporary text file
        Open (workingDir & vFile) For Output As #1
        Print #1, currentMail.Body
        Close #1
```

Figure 125: SWOMS Macro Code Segment 8

As mentioned previously, the macro adds onto the batch file. SWOMS Macro Code Segment 9 shows the addition of two lines: one to start swomsparser, and the other to delete the temporary text file.

```
' Add to batch file to process data and clean up intermediate files
' Run the parser on the email, wait until it finishes
Print #2, "start /wait swomsparser " & ".\" & vFile
Print #2, "del .\" & vFile   ' Instruction to delete input file
```

Figure 126: SWOMS Macro Code Segment 9

At this point the email needs to be marked completed. SWOMS Macro Code Segment 10 returns to one-left-shifted indentation and shows the end of the two "if" statements and the "Next" statement associated with the "For Each" loop.

```
                        currentMail.FlagStatus = olFlagComplete
                        currentMail.Close (olSave)
                        End If
                End If
        Next
```

Figure 127: SWOMS Macro Code Segment 10

The last section completes the batch file and runs it, ending the macro. The batch file will go through every email's text dump running the swomsparser on it.

```
closeandfinish:
        Print #2, "del %0"
        Close #2
        Shell (vFile2) ' Run the batch script outlined above
End Sub
```

Figure 128: SWOMS Macro Code Segment 11

L3.2.3 Using Macro

Using the macro is as easy as selecting emails and running the macro. The macro can be run by selecting emails, then going to Tools > Macros and selecting the SWOMSMacro.

To make it easier, a quick button can be added to the toolbar. Right click in the "whitespace" on any part of the toolbar (see Figure 129).

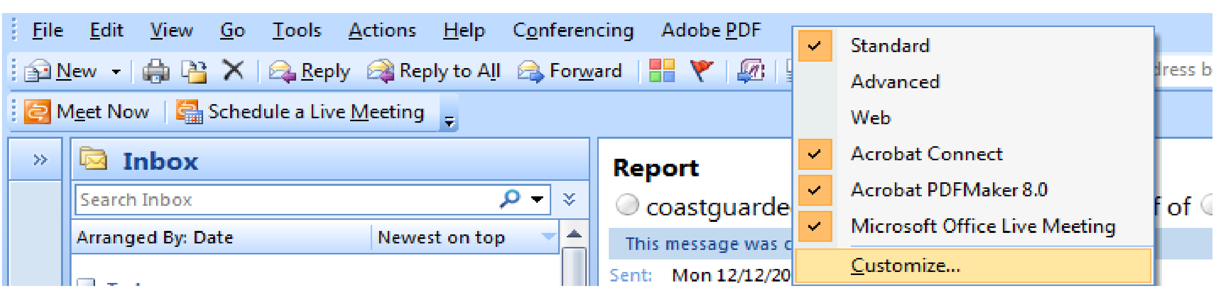


Figure 129: Menu to Customize Toolbar

Select "Customize…" and a dialog will open. Switch to the "Commands" tab, and then select "Macros" in the menu on the left. The window should look as it does in Figure 130.

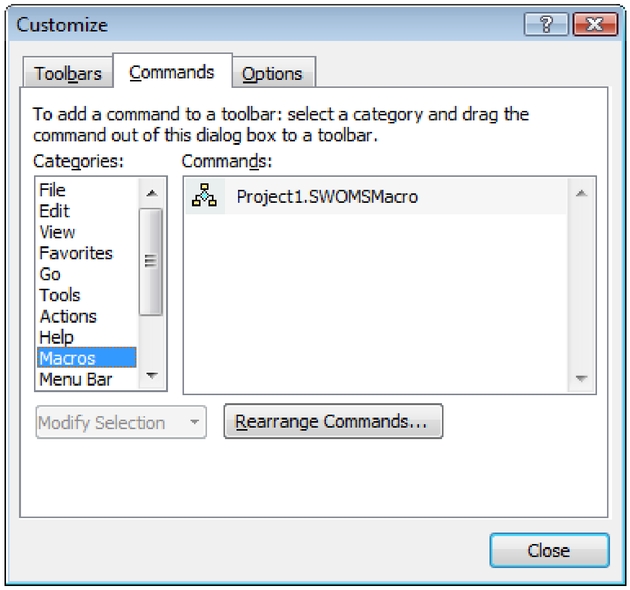


Figure 130: Add Macro to Toolbar

To add the SWOMSMacro to the toolbar, simply click the SWOMSMacro from the commands window and drag it to a toolbar. There should now be a button in the toolbar to run the macro. Figure 131 shows the macro after it has been dragged into the menu bar.

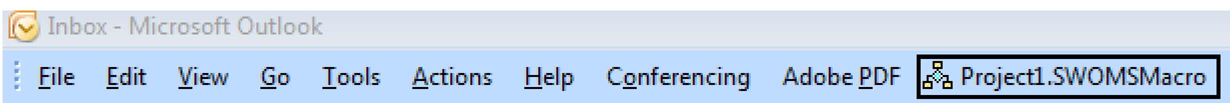


Figure 131: Macro Button in Toolbar

If at any point, the user would like to make any changes to the Macro button (including the name) or would like to remove the button, simply click the "Rearrange Commands" button in the Customization window that was originally used to add the Macro to the toolbar (see Figure 130). Select the toolbar that the Macro has been dragged onto under the Toolbars option (in this instance, it would be located under the Menu Bar), then select the Macro in the list to the left of the window, and then click on "Modify Selection" (see Figure 132).

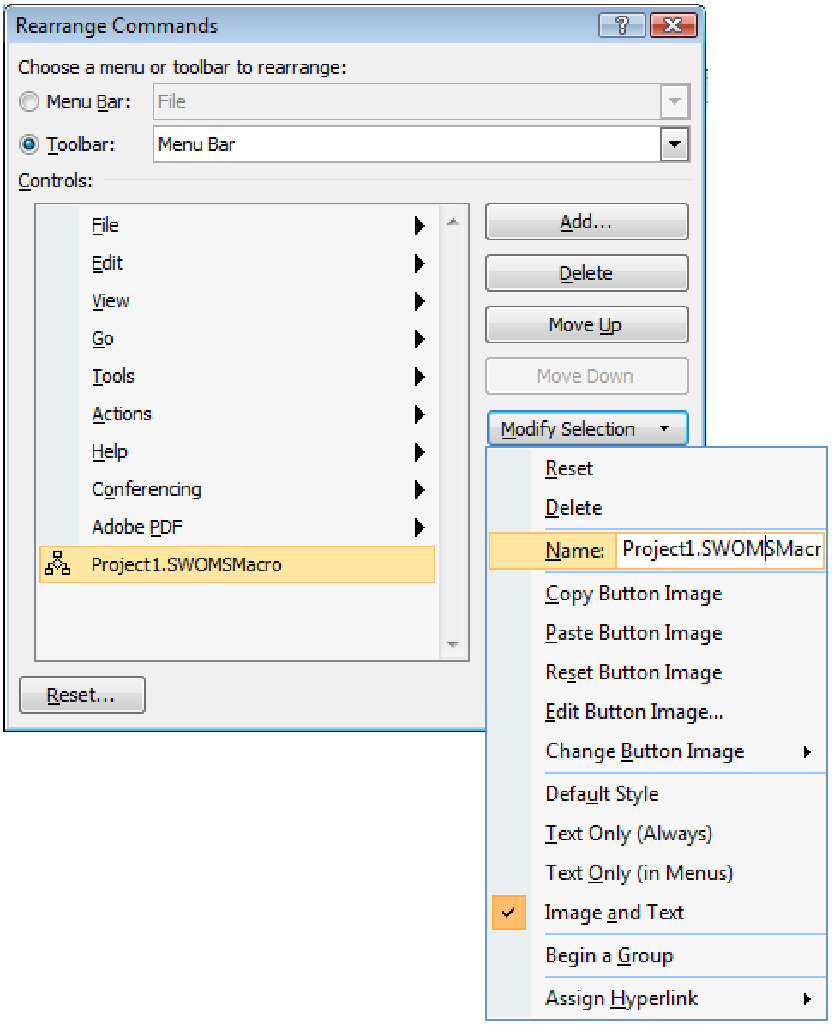


Figure 132: Modify Selection Menu

Once done, close the Customization window. At this point, simply select emails and click the button in order to automatically run the macro.

Note that the macro should not be used while a folder script is active, or else data collisions could occur, causing errors. Also included in the Mod_SWOMSMacro.bas file is a function and macro for removing "lock" files created by the swomsparser program. Normally, swomsparser will automatically remove the lock files that it creates. If program execution is halted before that point then the locks will still be present, preventing any future instances of the program from running and using the files. The ManualUnlock macro can help to remove the lock files.

Using the macro is as easy as selecting emails and running the macro. The macro can be run by selecting the emails, then going to Tools > Macros and selecting the

SWOMSMacro. To make it easier, a quick button can be added to the toolbar. Right click in the “whitespace” on any part of the toolbar. Select “Customize Toolbar” and a dialog will open. On the left-hand side, find “Macros” in the dropdown. Select SWOMSMacro and click the “Add” button. There should now be a button in the toolbar to run the macro. Simply select emails, and click the button and it should work automatically.

Note that the macro should not be used while a folder script is active, or else data collisions could occur, causing errors. Also included in the Mod_SWOMSMacro.bas file is a function and macro for removing “lock” files created by the swomsparser program. Normally, swomsparser will automatically remove the lock files that it creates. If program execution is halted before that point then the locks will still be left, preventing any future instances of the program from running and using the files.

L3.3 Exchange Script (Automated)

Setting up the Exchange script for full automation is the same process as it is done for the L2. Outlook Test Setup, but must be done on a service/shared account. Integration using public folders is an older communication service that is being phased out by Microsoft and is limited in its capabilities. The public folder administration tools, including the folder assistant, do not provide any way to run a script. In addition, running a VBA script to do any significant processing tasks is considered too much of a drain on a typical Exchange server to be practical from a stability standpoint.

The solution to this issue is to run an Outlook client on either a dedicated physical server or on a virtualized server. The Outlook client would be logged into a shared or “service” account on the Exchanger server and would be able to run the script in the same way as it would run for a personal user’s setup. The mailbox associated with the service account could still be set up to be viewable and changeable by personnel who use the system. Administration could be done by remote desktop sessions into the dedicated server, and would allow full control over changes to the VBA scripting and rules within the client.

Appendix M: Final Presentation

Final Presentation presented to:

Advisors:

Prof. Mustapha Fofana

Prof. John Orr

Liaison:

LCDR Chaning Burgess, CG-5432

Personnel of:

CG-543, Office of Vessel Activities

CG-545, Office of Investigations and Casualty Analysis

Date Presented: December 13, 2011