

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

**ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600**

UMI[®]



Université d'Ottawa • University of Ottawa

UNIVERSITY OF OTTAWA

SYSTEM SCIENCE

THESIS

**The Automatic Product Management System (APMS):
Integrated Business Process Management for a Small
Business Application**

**Student Name: Guowei Liang
Student No: 1763470**

Thesis Director: Professor Daniel Lane

August 13, 2002

© Guowei Liang, Ottawa, Canada, 2002



**National Library
of Canada**

**Acquisitions and
Bibliographic Services**

**395 Wellington Street
Ottawa ON K1A 0N4
Canada**

**Bibliothèque nationale
du Canada**

**Acquisitions et
services bibliographiques**

**395, rue Wellington
Ottawa ON K1A 0N4
Canada**

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-76602-0

Canada

Abstract

The purpose of this thesis is to analyze the business management process for small to medium size enterprises (SMEs) and to develop an automated product management system for integrated business processing. Due to the relatively small scale of operation of these companies, they experience difficulties in applying readily available tools in the area of product management. The target of the analysis is to develop a prototype software system for automatic product management. Moreover, this software developed for this thesis is applied to a real business case — that of *Inshore Fisheries*, a small scale integrated fishing operation in Atlantic Canada. The system consists of six components that are grouped into two categories of input and output analysis. The Input category includes production operations and customer ordering. The Output category includes four performance reports: (1) order invoices, (2) production worksheets, (3) inventory tracking, and (4) periodic income statements. The entire system is implemented using Visual Basic 6.0. The database is designed using Microsoft Access. The four performance reports in the output are designed using Microsoft Excel spreadsheets. Furthermore, a GUI integrates all components together to form a complete business processing system.

Based on the results of the application of this system to *Inshore Fisheries*, this software solution can be applied to other SMEs by making specific modifications but without revising the basic principle and system design of the software system. This system possesses a great amount of flexibility and extendibility for practical business process decision-making.

Acknowledgements

Special thanks, with a grateful heart, go to Professor Daniel Lane both for his professional knowledge of business management and for fishery industry operation. His wisdom and his experience provide this thesis a uncountable contribution. His support and supervision are essential for the entire research. Special thanks also go to Inshore Fishries, especially go to Yvon d'Entremont, business office manager, and company co-owner Jean-Guy d'Entremont. Without their help, the APMS has nowhere to put into practice. Without the feedback from them, APMS would end up to be a windbaggary. Special thanks further go to Professor Jean Michel Thizy. He provided us a very useful PC environment.

The thesis defense committee members also provide this thesis many helpful comments. They are: Professor Daniel Lane (Supervisor), Professor Jean Michel Thizy (Chairman), Professor Doug Angus (Examiner), and Professor Gilles Paquet (Examiner).

Table of Contents

1	Introduction.....	1
1.1	SMEs: The current situation.....	1
1.2	Objective and Model overview	3
1.3	Outline of the thesis	4
2	Literature Review	5
2.1	Canadian SMEs.....	5
2.2	Business Process Management (BPM) Software solutions: Spreadsheet/Database approaches	7
	2.2.1 Spreadsheet solutions.....	8
	2.2.2 Database solutions	9
	2.2.3 Excel vs. Database applications.....	9
2.3	BPM software solutions applications.....	10
	2.3.1 Banco BBA Creditanstalt S.A. (Banco BBA 2000)	10
	2.3.2 MemberWorks Canada Corporation (Rainier 2001).....	10
	2.3.3 Agris Corporation (Company Coach 2001).....	11
3	Methodology	12
3.1	The automatic product management system model	12
3.2	How the model works	12
	3.2.1 Using VB to construct the user GUI.....	13
	3.2.2 VBA – Macro Control inside the Microsoft Excel spreadsheets.....	16
3.3	Performance reporting	17
	3.3.1 Four output components	18
	3.3.2 The integration of output reports	20
3.4	Access Database	21
4	Case Analysis and Expected Results	23
4.1	Inshore Fishery Inc.	23
4.2	Motivation.....	23

4.3	Business Process Management Challenges	24
4.3.1	The management requirements.....	25
4.3.2	Inshore Office software.....	26
4.4	The database design of Inshore	26
4.5	The Inshore APMS Model	28
4.5.1	Input — Production.....	31
4.5.2	Input — Order.....	36
4.5.3	Output — Invoice	39
4.5.4	Output — WorkSheet	40
4.5.5	Output — Inventory.....	42
4.5.6	Output — Income Statement	43
4.6	Use of APMS Model	44
4.7	Implementation in Inshore	45
5	Research Issues	46
5.1	The selection of the spreadsheet software	46
5.2	Other BPM model applications	47
5.3	Model Scale – Local vs Internet	48
5.4	Model Implementation	48
6	Conclusion	49
6.1	Application to Other SMEs	49
6.2	Software Requirement	50
6.3	Extendibility	50
6.4	On-line order	51
7	Bibliography	52
Appendix A	55
The Invoice Statement	55
Appendix B	56
The Production Worksheet	56

Appendix C	58
The Inventory Report	58
Appendix D	59
The Income Statement	59
Appendix E	60
Source Code of the APMS	60

List of Figures

Figure 1.1: The Outline of the Principal Model -----	3
Figure 3.1: The VB Graphic User Interface (GUD) -----	14
Figure 3.2: The Flow Chart for VBA Code -----	17
Figure 3.3: The Relationship among Four Components -----	20
Figure 3.4: The Design of the Relationships -----	22
Figure 4.1: The Database Relationship of Inshore Fisheries -----	28
Figure 4.2: The Opening window -----	29
Figure 4.3: The VB Graphic User Interface (GUD) for Inshore Fisheries -----	30
Figure 4.4: The Production Window for Choosing a Species -----	31
Figure 4.5: Species Information Summary -----	32
Figure 4.6: Input Production Details -----	33
Figure 4.7: Production Process for Fresh -----	34
Figure 4.8: The Production Process for Frozen -----	35
Figure 4.9: The Production Process for Slabs -----	36
Figure 4.10: Order Window for Choosing a Customer -----	37
Figure 4.11: Customer Information -----	38
Figure 4.12: Invoice Pop-up Window -----	40
Figure 4.13: Work Sheet Pop-up Window -----	41
Figure 4.14: Inventory Pop-up Window -----	42
Figure 4.15: Income Statement Pop-up Window -----	43

List of Tables

Table 1: Sample Production Report Design	18
Table 2: Sample Invoice Statement Design	18
Table 3: Sample Inventory Report Design	19
Table 4: Sample Income Statement Design	19

1 INTRODUCTION

The purpose of this research is to describe the the development, design, testing and application of an automatic spreadsheet-based product management system to assist small and medium-sized business operations reporting and decision making.

1.1 SMEs: The current situation

SMEs (Small and Medium-sized Enterprises) are categorized by their firm size in terms of the number of employees in the organization- usually less than 250 employees in Europe or Canada, and less than 500 employees in the U.S.A. Most SMEs are independent and non-subsidiary operations. They may appear in the format of a sole proprietorship or a partnership and cover all the business sectors, e.g., production, finance, marketing, accounting, human resource management. The critical role in the development and success of SMEs is provided by progressive entrepreneurs who are willing to take more risks in order to get a better market opportunity.

Along with the growth of the economy, the market structure becomes more and more complex. As the first economist who studied successfully the dynamics of market structures, Schumpeter claimed that the growth of the economy is a result of entrepreneurial innovative behavior (Schumpeter 1962). Although many large corporations are specialized in formal R&D (research and development) and register many patents, they are not the sole source of innovation in the economy. The entrepreneurs in SMEs are also essential to economy development because they are more willing to take the risks in order to take advantage of the fleeting business opportunities. One research study found that small firms are more efficient than large ones in R&D (Acs and Audretsch 1990). Due to these innovative activities, SMEs drive market dynamics and stimulate the economic growth through continual birth, expansion, contraction, and death of firms (OECD 2000a).

Accompanying the globalization of the economy, SMEs are more and more exposed to international market competition. SMEs are “directly producing about 20 percent of OECD exports and about 35 percent of Asia’s exports” (OECD 1997, p. 7). Further, many SMEs have also established supply, sales, and promotion networks with some transnational corporations (TNCs). These outsourcing activities of the TNCs increase the market share of SMEs. Moreover, due to the dynamism of frequent entry and exit of SMEs, the international productivity and the overall economy will be developed continually.

In Canada, SMEs are a highly diversified group. Their activities are spread to every business sector, especially the service sector, e.g., wholesale and retail, restaurant and hotel, communications and the Internet. As the marketing plan of Corporations Directorate (Canada 2001) stated in 1995, SME businesses are the primary target markets in British Columbia, Alberta and Ontario (Kindra 1997). It is important to mention that 99% of firms in Canada are SMEs. Over 2.6 million SMEs account for 43% of private sector GDP and created 81% of net new jobs in last decade. The federal government in Canada has supported the growth of SMEs through many programs for SMEs that include tax reduction, human resource management, R&D, and the information highway.

Along with the continuous emergence of new technology and globalization of economy, the importance of economies of scale is declining. This fact provides more opportunity for smaller scale operations to be successful. But it also creates many business barriers to SMEs simultaneously. These include: low productivity, financial shortage, technology renewal, constrained managerial skillset, and new industry standards. The objective of most SMEs is to reap the maximum profit from the current market segmentation. They focus on promotion and sales of products and customer services. It is very easy to overlook the internal management. In order to extend their companies’ life cycle and to succeed opponents in competitive market situations, the successful SMEs should take advantage of existing technologies to improve the management skill and capabilities of internal business process.

1.2 Objective and Model overview

In this research, we explore and integrate up-to-date software technologies to assist an SME in production planning and standard reporting of business operations performance. This includes the capability of gathering and analyzing internal information in order to manage better.

The emphasis of this project is on the design of a performance-reporting package of the SME's business process management (BPM) system. From production to sales, the system records all production and customer entries and stores them in a database. Based on the production and the delivery of the orders, a supporting system is established to manage the inventory and report on aggregate economic performance on a periodic basis. Furthermore, all the information should be retrievable later under a simple request from the user and in an easy user select format that will enable users to query the database system. Based on these principles, the proposed research design of the business process management system is based on the following model overview:

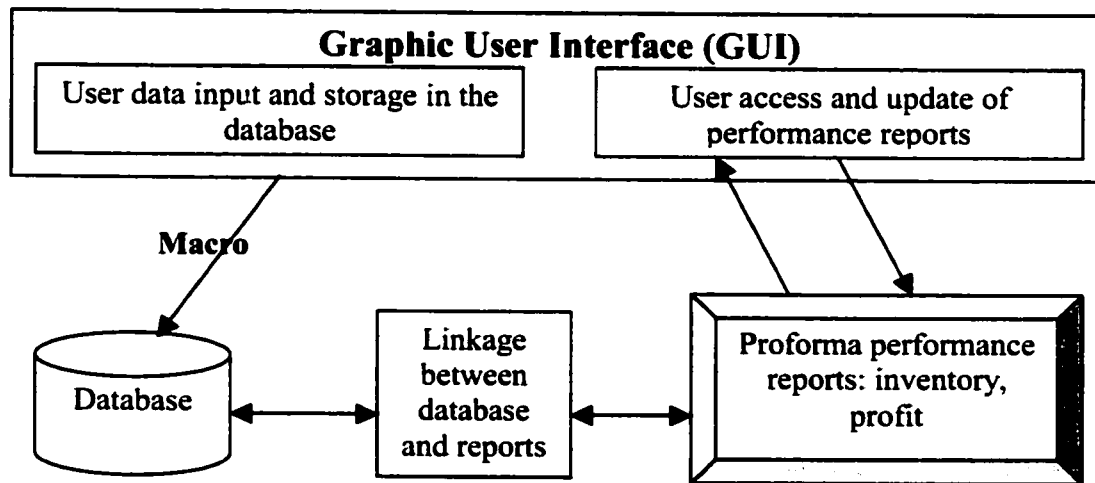


Figure 1.1: The Outline of the Principal Model

The BPM model allows users to provide business process information (such as production dynamics, inventory characteristics, revenue and cost components) in a structural manner to a database. The database provides the information necessary to develop *pro forma* company reports. The user interface allows company members to query the database, to make additions,

deletions and changes, and to request *pro forma* company reports automatically generated by the report writing functionality. The model design caters to specific company requirements and results.

By implementing the automation strategy through the system, it reduces human error. The system manages regular business process activities automatically, e.g. issuing invoices, preparing customers' orders, etc. These tasks release employees to carry out the operation more efficiently. By using the automation system, the employees can handle more business documents but use less time. Furthermore, due to using a "trustworthy system", employees don't have to double-check the documents.

1.3 Outline of the thesis

This thesis describes the development, design, testing and application of an automatic spreadsheet-based product management system for a small and medium-sized business operation. It outlines the motivation, the technologies, and real world application relevant to this research. The following section contains a literature review that takes a broad look at the related issues published in the main areas of research. Next, the research methodology is presented. A cornerstone of the research is the case study for the application and implementation of the research output to a small-scale integrated fish harvesting and processing operation in southwest Nova Scotia. Finally, in the last chapter of the thesis the expected results of this work are described and the extensions of the research are discussed briefly.

2 LITERATURE REVIEW

The following presents a brief discussion on the development of product management systems in SMEs and an analysis of system components. The literature review is divided into the following areas:

1. Production management systems in Canadian SMEs: a brief history of SMEs and their use of automatic business process solutions to date.
2. Business Software solutions: an introduction to the business applications of spreadsheets and databases and the differences between them.
3. Examples of case studies: 3 cases that describe the actual use of the cutting-edge technology to deal with the internal business process management problem.

2.1 Canadian SMEs

Since the mid-1970s, entrepreneurial activities among small and medium-sized enterprises increased. In 1980s and 1990s, we perceive a further rapid growth of SMEs in Canada. Both the numbers of the businesses and numbers of employees have increased dramatically (Baldson 2000) over this period. Moreover, these entrepreneurs are coming from different ethnic cultures and the proportion of participating females is increasing. Most Canadian SMEs are active in the service industry. The following lists some aspects of current SMEs business in Canada (Canada 1994):

- 99% of firms are SMEs and 53% of the labor market is in SMEs.
- 60% of SMEs are in the private sector and they account for 43% GDP of the private sector.
- 32% business profit and 29% gross sales are generated by SMEs.

Implementing the proper strategies makes SMEs successful and competitive. These strategies are related to the firms' activities like financing and technology. We can summarize the key successful factors (KSFs) as follows (Baldwin, Chandler, Le and Papailiadis 1994):

1. Flexibility of quick response to customer requirements, quality of products and customer service.
2. Rigorous TQM (Total Quality Management), just-in-time inventory control.
3. Marketing and operation management.
4. Skilled employees.
5. Fast capital turnover and use of limited capital.
6. Adaptability to new technology, new products, and new materials.
7. Government programs.

Among all these KSFs, “technology”, or in other words, R&D, is a measure that differentiates more successful from less successful SMEs. New technology appears in the market at an accelerating rate. In order to catch up to the pace of the information age and satisfy the dynamic market requirements, R&D is an ongoing issue. R&D provides SMEs with new cutting-edge technologies. Knowing the new technology is indispensable, for example, to entering into the foreign market.

There are four types of technology strategies for an SME:

improving own existing technology,

using other firms’ technology,

refining other firms’ technology, and

developing a new product (Baldwin, Chandler, Le and Papailiadis 1994).

Except for using others’ technology, the other three types involve the R&D process of the development of technology to different extents. Thus, in order to increase competitive advantage, R&D is necessary. R&D contains product innovation and process innovation. Both of these are crucial for an SME to survive in the severe market competition it must face.

In this research, the major focus is on the business process innovation and development. It explores aspects of products management using cutting-edge technology in the form of available business software systems. According to this innovation, SMEs can upgrade their capability of

internal management efficiency and thus increase their competitive advantage in the marketplace..

As mentioned in the first part of this thesis, developing management skills and applying technology potential are the weaknesses attributed to failed Canadian SMEs. Thus, how to take the full advantage of the existing technologies and integrate them into management operations is the key problem for most SMEs. Compared with large firms, SMEs are still in a 'sunrise stage'. A high entry rate of new SMEs is also associated with a high exit rate (WIPO 2001). This fact doesn't necessarily imply that the SMEs which failed to survive don't have a marketable product. Actually, it is a result of the incomplete and inexperienced management teams in these SMEs. On the other hand, the high exit rate is not an adverse indicator of the SME business. It is more like a learning process. The entrepreneurs obtain valuable management skills and experience from the failures. Further, as stated in Baldwin, Chandler, Le and Papailiadis (1994), R&D makes a difference for the more successful firms. R&D enables the SMEs to have the ability to adopt new technology. It enables the Canadian SMEs to grow continually.

2.2 Business Process Management (BPM) Software solutions: Spreadsheet/Database approaches

Along with the rapid popularization of the PC (Personal Computer), using computerized or electronic spreadsheet and databases to deal with data are already very common approaches to solving business process management problems. Especially in the SME business environment, entrepreneurs are familiar with computerized bookkeeping of employee information, inventory, customer invoices and so on.

A BPM software system is a solution for SMEs seeking a technology advantage. The basic requirement for BPM software is to automate the interaction of the customers, employees, business transactions, and the data (MetaStorm 2001). By using BPM software, it helps a SME to release resources from tedious paper work, and allows the business process to become more productive and more profitable. BPM software are business tools which enable users to automate the business transaction process by integrating some kinds of software that could be either customized software, or some common commercial business software such as spreadsheet

or database software designed for specific company needs. As a solution, users can obtain regular customized, professional business performance reports.

Spreadsheet solutions, such as those provided by the leaders in the field, Microsoft or Corel, have very powerful functionality. These spreadsheet and database software packages summarize and analyze information from different resources and then present them in user-friendly format to help decision makers (e.g., entrepreneurs) gain the business “big picture” for the company (Power 2000). Moreover, built-in templates or user-defined templates are capable of providing much convenience to users.

The following describes the basic characteristics of these software solutions using common spreadsheet and database platforms.

2.2.1 Spreadsheet solutions

Microsoft Excel is a spreadsheet program used to organize, manipulate, and chart data and to perform calculations. It is particularly good at managing tabular data. It also excels at creating, publishing, viewing and analyzing data for professional reporting purposes (Microsoft 1998). It has a set of rich analysis and formula tools that enable business users to do calculations much easier and simpler than before. For example, the pivot table and query functions (although not widely recognized and used in business applications) allow users to summarize and compare data quickly. The Excel worksheet-to-worksheet link, management and function wizards enable users to handle data applications more automatically.

After the introduction of Microsoft Excel 97 (JWalk & Associates 2001), Excel became a more intelligent business software package since it integrated the Visual Basic for Application (VBA) programming language inside the software allowing advanced users to develop automatic execution of multiple spreadsheet functions. Through VBA, and Object Linked and Embedded (OLE) functionality, other programming package applications can now communicate internally with other Microsoft software products, e.g., Excel passing information back and forth with the

Access database package. By using VBA, macros (short executable programs) can be manipulated by allowing the spreadsheet to perform some user-specific commands.

2.2.2 Database solutions

Microsoft Excel List and Microsoft Access both can serve as a PC-based relational Database Management System (DBMS). A DBMS is a program that lets one or more computer users create and access data in a database (Brydon 1997). DBMS is challenged by the following:

- It should support two industry standards: Standard Query Language (SQL) and Query By Example (QBE).
- Take advantage of VB and macros.
- The new object-oriented extensions need to encapsulate the concepts of state and behavior information.
- Must provide user-friendly graphic users' interface (GUI) such as the Access Database windows.

2.2.3 Excel vs. Database applications

Excel is not necessarily the perfect tool for complicated data architectures that require significant data input and validation. The following issues arise for considering development-using Excel (Powell 1995):

- **Data size:** if it is large (more than 10,000 records), database software like Access may be more appropriate.
- **Data Security:** Excel has a very low security level. Its security is only for an entire workbook, not for each record (row).
- **Multiple user requirements:** Excel is not designed for multi-user applications. For some business transactions processing involving many users in different locations, it may be better to use multi-user database management software.

- **Application objective:** Excel is effective at a combination of data analysis, statistical analysis, diagram plotting, report generation while database software is more suitable for intensive data entry and data validation.

2.3 BPM software solutions applications

Following on the more formal descriptions above regarding business software, some examples are necessary for further understanding software solutions applied to business process management. The following lists three cases to illustrate potential advantages of deploying new technologies.

2.3.1 *Banco BBA Creditanstalt S.A. (Banco BBA 2000)*

Banco BBA is a Brazilian investment bank that provides a wide range of wholesale financial services. As the global competition increases continuously, BBA recognized the importance of IT. It was challenged to reduce its IT cost without compromising its market position and market share. By using Microsoft Office 2000, Banco BBA Creditanstalt S.A. reduced 17% of the end-user costs in distributed environment and 9% of overall business costs. Before taking advantage of Office 2000, it had deployed Office 97 for a few years to analyze and report on a range of its financial services business process work. The new features of Office 2000 allowed Banco BBA to further reduce its operation expenses by increasing efficiency. GartnerGroup conducted end-user and IT surveys both before and after the deployment of Office 2000 to evaluate the impact on end-user productivity and satisfaction. The survey showed that the time saving for the each GartnerGroup category is as follows: casual learning & self-support reduced by 43%, peer-to-peer support reduced by 43%, and file and data management reduced by 9%.

2.3.2 *MemberWorks Canada Corporation (Rainier 2001)*

MemberWorks Incorporation is a direct marketing company located in Montréal, Canada (MemberWorks 2001). It designs innovative membership programs for its customer companies.

These programs help customers get more value by providing them with necessary information not generally available to the public. As part of its business nature, MemberWorks has to respond to customers very quickly. The information for the customers must also be reliable. These challenges required MemberWorks to have a integration of the existing warehouse automation systems in order to customize the configuration of membership benefits. Thus, it uses clustered SQL 2000 and Windows 2000 as its business process management solution. Further, by taking advantage of the web-based application which uses a built-in function in Visual Basic, different departments in different locations can create and modify member benefit configurations. This BPM software solution provides much more membership flexibility for MemberWorks; and the customized reports are generated much faster and more conveniently.

2.3.3 *Agris Corporation (Company Coach 2001)*

Agris Corporation, headquartered in Atlanta, develops software solutions for agricultural communities in U.S., Canada, South Africa, and Australia. It has a sales support department which is responsible for generating proposals using Corel's WordPerfect for the sales representatives. When it merged with Farmland Industries, they had many repeated "cut and paste" to be carried out because Farmland used the Excel spreadsheet package to produce the proposals. It took time to produce a complete proposal because the graphics and fonts did not match between these two pieces of software. As a consequence, it created friction between the sales department and administration. After a market search, they found a software, ProposalMaster, which is a sophisticated, automated, PC-based system for generating professional and customized proposals. ProposalMaster can match the software that Agris already had in place: Word, Excel, Access, and PowerPoint. ProposalMaster also interfaced with Agris' existing contact management software. After implementing ProposalMaster, Agris now produces a proposal much faster than before. It creates a better correlation between what sales representatives need and what the proposals provide. ProposalMaster has the flexibility to customize the proposal to meet the specific opportunity and address the particular client demand. As a result, Agris' productivity has increased.

3 METHODOLOGY

The following sections describe the technical implementation of a general BPM system model for an SME. This model has a GUI (Graphic User Interface) for users to choose the preferred business process analysis and reporting functions. Each of these business process functions is represented by a model component. Basically, there are six components which are grouped into two categories. These include the general business process reports for Production and Customer Order input reports, and Invoice, Inventory, Production Worksheet, and Income output reports (Figure 3.1).

3.1 The automatic product management system model

Consider a small to medium sized enterprise, which has to develop its business dynamics in order to maintain market position and competitive advantage through its production and sales processes. Due to its relatively small scale of business, the company cannot afford to have its own R&D team. It is critical, however, for the company to have an effective automatic operation and production management system. Thus, it has to deploy a suitable software solution to meet the requirements of its internal business process management. These requirements may include the real-time reporting of sales and production processes which are the key sources of company benefits and expenditures. The business reports are necessary because the company is required to keep the records of company transactions. Furthermore, the production and the sales processes affect the company ongoing products inventory positions. Finally, periodic reporting formal reports, e.g., the income statements, require that an aggregate review of business process activities be conducted over a defined period. A generic look at this automatic product management system model is described in the following subsection.

3.2 How the model works

Inside the model, we deal with business process model components as objects. For example, production, sales, inventory, and income are different objects. All these objects are linked into

the database, which is designed and implemented using the software of Microsoft Access[®]. The connection between the user interfaces and the database is made by taking advantage of the new technology of ActiveX Data Objects (ADO). ADO is a data access technology which combines all the best features of Data Access Objects (DAO), Open DataBase Connectivity (ODBC), and Remote Data Objects (RDO) (Wright 1998). Moreover, VBA is applied in the spreadsheets that are designed with the software of Microsoft Excel[®]. The spreadsheet is linked into the database using macros. The information of all the tables in the database is manipulated with the sequenced query language (SQL). This design allows us to develop a user-friendly Graphic User Interface (GUI) that helps the user to carry out all the functionalities of the model in a much more straightforward and easier than otherwise. These functions of course may include the data input and the information retrieval.

3.2.1 Using VB to construct the user GUI

The VB-GUI provides an interface for users to input the raw data. On the interface, it describes itself by some text that tells the user what and how to use this software. The user simply clicks on the appropriate menu buttons on the interface in order to fulfill different functions. Each button will bring the user to another pop-up window in which the user can enter data and then insert them into the database or retrieve the requested information from the database in report format. This interface should also allow easier access to the spreadsheet reports so that the user can view or print out the reports according to their preference. A basic form of the GUI showing the business process model functionality looks as in Figure 3.1.

In Figure 3.1, there are two major parts for the user interface: INPUT and OUTPUT. Each part has several functionalities; and every function is treated as an object in this model.

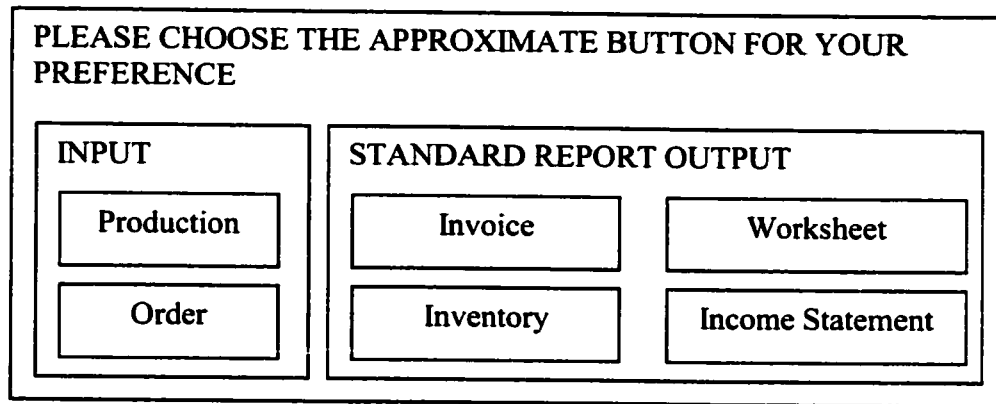


Figure 3.1: The VB Graphic User Interface (GUI)

- **INPUT:** This component includes two functions; one is for production (product output “Production” button) and the other is for customer order (“Order” button).
 - (1) “Production” activates another pop-up window to ask the user to select a specific product he/she wants to process. Users can update the information of that species from that pop-up window. After choosing the product, the user then can go step by step according to the instruction of the interface in order to enter all the production data of that particular production item. All these data are eventually stored in the database. Moreover, the number of the product item produced is added into the inventory.
 - (2) “Order” works in a similar way as “Production”. After clicking on the button of “Order”, another pop-up window will be activated. The user can update the customer’s personal information according to that window. In order to choose the correct customer among all entered in the database of the company’s customers, the user moves through the customer list. On selection, the user then activates an Excel spreadsheet to process the customer’s order and to print the order report form. Again, all the data will be inserted into the database for future retrieval.
- **OUTPUT:** There are four reports included in this function. Each of these reports is designed according to a standard *pro forma* statement format. These statements include

company reports on the customer order invoices, the company's production worksheet reports, the current inventory position of the product portfolio, and aggregate cost and earnings (income) statements. These reports arise from the GUI buttons of "Invoice", "WorkSheet", "Inventory", and "Income Statement", respectively, as shown in Figure 3.1. The user can modify and update the information directly from these reports. These reports are developed based on the linkages to the company input and database by macro programs using VBA.

(1) "Invoice" activates a pop-up window that presents the user with a combo box asking the user to choose from a list of invoice numbers recorded in the database. According to this invoice number selected, the program retrieves all the related information on a particular customer order from the database. This information may be previewed as a Microsoft Excel spreadsheet report. It may also be printed out as a report and sent to the customer as the order invoice informing the customer of the sales agreement, products purchased and payment required for this order.

(2) "WorkSheet" activates another pop-up window which also asks the user to pick a production number from a combo box of recorded database information. Through this production number, the user can also examine all related information presented on the Microsoft Excel spreadsheet. The spreadsheet further calculates the profit and the cost of the production activity. This spreadsheet is a final report for the company on the production activity.

(3) "Inventory" activates a different pop-up window, which asks the user to identify a particular product. After the user selects a product from a combo box list of all products, this user can preview the current inventory status of that product that is presented via an Excel spreadsheet.

(4) "Income Statement" also activates a distinct pop-up window which will ask the user to provide input on fixed expenses. With this information, the system combines the results from the production and order activities over a specified time period to construct the

Income Statement for the period. The income statement is presented in *pro forma* format on a separate Excel spreadsheet report.

3.2.2 VBA – Macro Control inside the Microsoft Excel spreadsheets

The application of Visual Basic programming language built around Microsoft Office business applications increases the power of this popular software to a great extent. Visual Basic for Application (VBA) allows programmers to design spreadsheet and database macros that execute tasks specified by user preference. VBA is a useful programming tool for integrating several Microsoft Office solutions together, such as integrating Excel and Access in this model. The following provides a descriptive example of how the VBA Macro will be used to control the spreadsheet and the database.

For the “Order” executable, there is a defined Excel spreadsheet template that matches the company’s order report form. This template has a built-in macro. This macro is controlled by a command button. It can transfer all the data about the customer’s request to the database using Microsoft ActiveX Data Objects (ADO) Library. For example, assume three customers request orders today. The first orders 25 different products, the second orders 38 different products, and the third orders 42 different products. Each customer has to click the command button, (for example “SUBMIT/SAVE”) when they finish filling out the order form. This “SUBMIT/SAVE” button is already assigned to a macro programming function within VBA. It activates the database file and transfers the information about the customer order just input to that file. Information for the three customers goes to that file one by one, that is, 25 of the first, 38 of the second, and 42 of the third. All 105 product request records are now stored in that database file. The VBA code for the “SUBMIT/SAVE” button is designed as following control flow diagram:

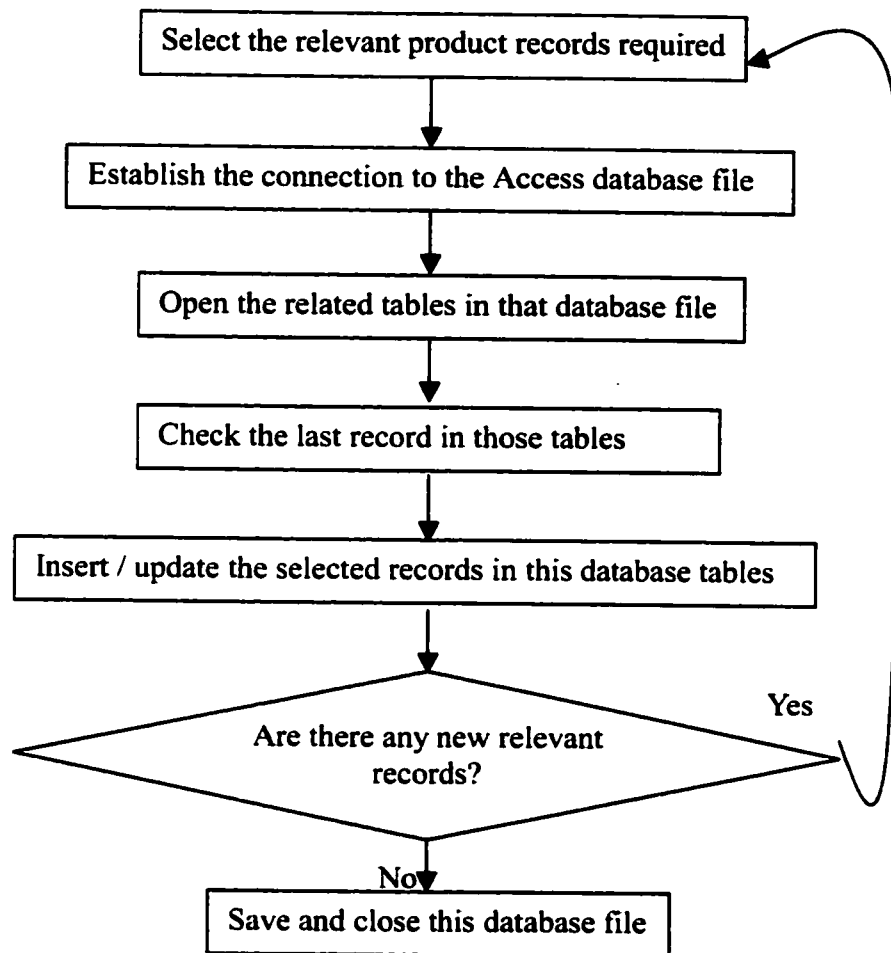


Figure 3.2: The Flow Chart for VBA Code

3.3 Performance reporting

In this research, there are four output spreadsheet components and associated reports: (1) production worksheet, (2) invoice statement, (3) inventory reports, and (4) income statement. They are all designed using the spreadsheet methods of Microsoft Excel. Moreover, these reports have to be integrated together using the database and the GUI input components. They all can be retrieved according to the output part from the GUI which is described briefly in Figure 3.1. They are also designed using the built-in cell functions provided by Excel which calculate cell values automatically and consistently, and thus eliminating the possibility for human error.

3.3.1 Four output components

Production. The Production worksheet, as its name suggests, records the basic data and report form information about the production process. This information includes the wage rate, the quantity produced for each individual product item, the cost of production, the production area, the exchange rate, e.g., between the US dollar and the Canada dollar (as required), and so on, in accordance with company procedures. Table 1 provides a view of a design for a simple production report.

Exchange rate (\$US vs. \$CAN): \$1.5018		Wage rate (\$CAN/hr): \$15.85		Production area Code: 4Xp	
Product items:	Item 1	Item 2	Item 3	Total	
Quantity (units)	50	69	71	190	
Expense (\$CAN)	\$5,000	\$6,900	\$7,100	\$19,000	

Table 1: Sample Production Report Design

Invoice. The Invoice statement is a worksheet for billing the customer. It keeps track of the order made in the input segment by the customer. All the information from the customer is shown on this worksheet. The customer name, the products ordered, the total amount due, the invoice number, and the date of ordering and payment are all recorded on this invoice statement. Further, the invoice report checks the remaining quantity of each product item in inventory as an available quantity for completing sales (Table 2).

Customer name: Minutes Food	Account No.: 50001		Invoice No. 62011		Ship VIA: R&R		Date: Aug. 21, 2001
Product items	Item 1	Avail.	Item 2	Avail.	Item 3	Avail.	Total
Quantity	80	100	67	200	53	169	200
Price (\$US)	9,000		7,900		7,100		24,000

Table 2: Sample Invoice Statement Design

Inventory. Inventory reports are used for keeping track of the volume of all product items. Knowing the inventory levels helps manager make decisions about whether to manufacture more

or not and which product item should be produced. Along with the production and the customer order, the amount of inventory moves up with new production and down with customer orders over time. Therefore, the business process management systems needs to include a report to keep track of inventory levels for all the product items in the firm's product portfolio. Further, the inventory status is used to check and verify the final invoice statement as to how much customers can order at maximum by product. If the customer requires more than the inventory, the system sends back a warning message to the user to inform about the inventory shortfall (Table 3).

Inventory Statement		Date: Apr 12, 2002			
Species	Item No.	Size	SO/B	Pkg	Qty
Product 1	CA2B1	8 to 12	B	5	589
Product 2	FL1S2	12 up	SO	10	500
Product 3	TU1S4	4 to 8	SO	20	734

Table 3: Sample Inventory Report Design

Income Statement. The last of the four output reports is the income statement. This report provides a *pro forma* statement for a given period of the profit or the loss for the company. The definition of the period is flexible but normally is done monthly, quarterly or yearly. The income statement is a consolidated performance measure of a collection of production activities (i.e., operating costs of the firm) and customer orders (i.e., operating revenues to the firm). The following table presents a simple example of the income statement report.

Income Statement	Year to Date Jan-Apr (in CAN\$)
Operating Revenue	\$900,000
Operating Cost	500,000
Other Expense	100,000
Profit before Tax	400,000
Tax (50 %)	200,000
Net Profit (Loss)	200,000

Table 4: Sample Income Statement Design

3.3.2 The integration of output reports

All business process spreadsheets whether input or output is related to each other to provide an integrated system of business reports. The production worksheets add the amount of every product into the inventory worksheet. The invoice statements take out the same amount from the inventory. The inventory also tells the invoice statement how much is remaining for future sales and triggers production decisions. The cost comes from the production, and the revenues from the invoice orders. Cost and revenues go together to calculate the final profit position of the firm.

The production worksheet has all the information of the production process of every product item. The quantity (Table 1) of each item produced adds to the inventory entries. The expense data items from the production worksheet (Table 1) also are linked to the income statement (Table 4). The Invoice statement asks how much is available (Table 2) for every product in advance of new orders by establishing a link to actual inventory. The total balance (Table 3) of the inventory is reduced by the quantity for each item (Table 2) in the invoice statement. The prices (Table 2) for each item that the customer orders contribute to the firm's revenue. These amounts go to the income statement as the revenue (Table 4). Figure 3.3 illustrates the component linkages of the system.

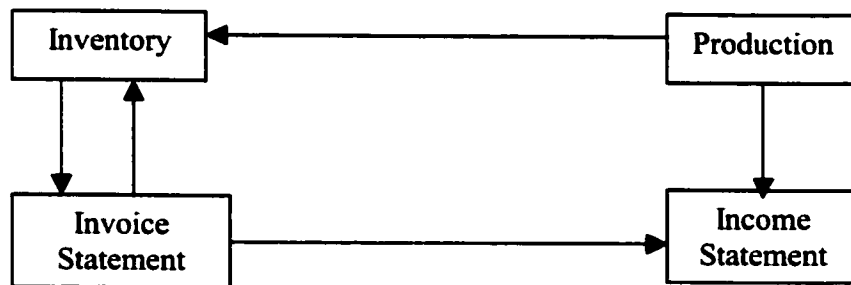


Figure 3.3: The Relationship among Four Components

3.4 Access Database

The database is designed and developed using with the popular Microsoft Access software as part of the Microsoft Office suite of business software. According to the information required by the six model components (2 input and 4 output modules), the database basically contains 9 tables which store the relevant data. These database tables are described below.

1. **Invoice Table:** it contains the records of customer orders.
2. **Production Table:** it stores the information for every production process such as date, exchange rate, and production area, etc.
3. **Inventory Table:** it records the amount of each frozen product item .
4. **Customer Information Table:** it keeps the information on the customers.
5. **Product information Table:** it stores the data for processing each product including workers' salary rates, packing costs, and overtime, etc.
6. **Frozen Products Table:** it stores the records of the frozen product for each production process.
7. **Fresh Products Table:** it stores the records of the fresh product for each production process.
8. **Other Products Table:** it stores the records of other products produced in the production process including by-products.
9. **Invoice and Customer Table:** it has the information which has the invoice number and the corresponding customer name.

These 9 tables are linked with defined relationships among them. The following diagram describes these general relationships (detailed explanations of the relationships between each table are referred to in section 4.3 of this thesis related to the case study). Figure 4.1 below summarizes the implied relationships among the various database tables.

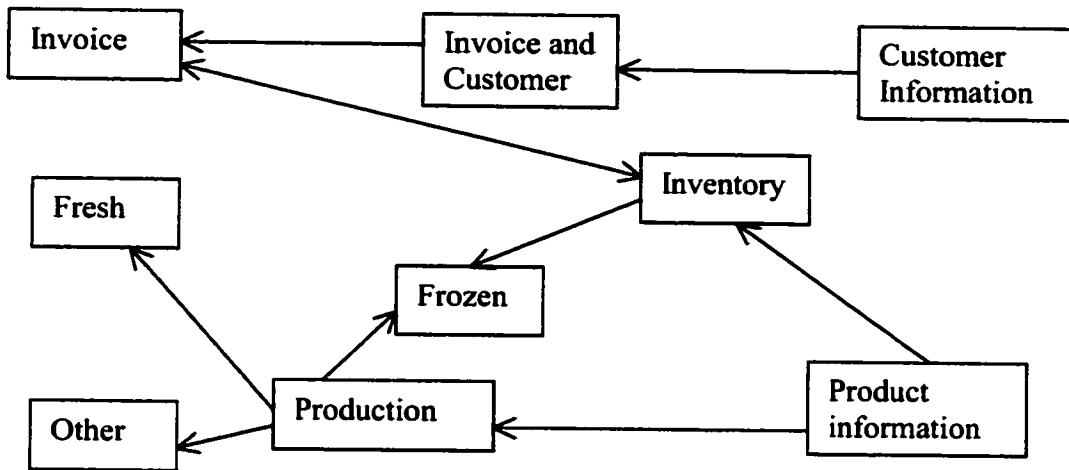


Figure 3.4: The Design of the Relationships

The following section presents details of the system design and use for a specific case study for a small vertically integrated fish harvest and processing firm in southwest Nova Scotia.

4 CASE ANALYSIS AND EXPECTED RESULTS

Chapter 3 described the model developed in the course of this research in general terms. This chapter presents a business case that applies this model practically for a real application. This case is used to test the validity and functionality of the automatic business processing model. The expected result is also presented in this case study.

4.1 Inshore Fishery Inc.

The fishing sector in Atlantic Canada is a significant economic activity for the region, especially for small to medium-sized enterprises (SMEs) that dominate the sector throughout the provinces of Nova Scotia, Newfoundland, New Brunswick, and Prince Edward Island. The SME business environment with its heavy reliance on nearby American markets is considerably dynamic. *Inshore Fisheries* of Pubnico, Nova Scotia is a third generation family-owned and operated fish processing and harvesting company. It has diversified into various areas in the fishing sector of southwestern Nova Scotia, including groundfish, invertbrates, and shellfish. *Inshore Fisheries* produces fresh and frozen fish products primarily for a strong customer base in the eastern seaboard of the United States. The company specializes in ground fish species (e.g., cod, haddock, flounder), but also shrimp and silver hake (Inshore 2001). *Inshore Fisheries* has five mobile gear vessels that vary between the sizes of 45 and 63 feet in length. The crew in each vessel is usually 3 to 5 people. *Inshore Fisheries* has a complete fish products line for fresh and frozen seafood products. They can deliver the packages of fresh fillets in groundfish of 10lb, 20lb, and 25lb.

4.2 Motivation

Inshore Fisheries, a small, family-owned and operated diversified fishing and processing company located in Nova Scotia's southwestern area, produces fresh and frozen fish products primarily for a customer base in the eastern seaboard of the United States. Improved business

performance in this competitive marketplace requires companies to be on top of every business transaction. For example, *Inshore* needs to know very clearly about product inventory, especially for frozen products, and to be prepared to respond quickly to customer demands for fresh fish products. The APMS model is designed to assist *Inshore* in its ability to respond quickly based on the potential customer sales and inventory and production possibilities.

At present, *Inshore Fisheries* operations are handled by a limited number of well-versed managers who have effective control over buying, selling, and production. During the peak business periods, decisions are made with rapid-fire urgency and very little documented information. Business reporting is done after the fact and as a result is a cumbersome task that requires automation.

The APMS system links production reports from handling and processing raw fish product to meet customer demands arising from product invoice lists, to frozen inventory products, and finally to reports on net operating income over a series of production, inventory, and customer invoices.

The APMS allows the production, inventory, invoice, and income statement reports to be linked. This means that changes in one business report (e.g., production worksheets) are immediately reflected and updated in related reports (e.g., product inventories). This system provides an easy means of providing work reports data and then automatically uses these data to generate business reports, e.g., an invoice report for a customer, a worksheet production statement, and an operating income statement report for managers.

The major motivation of developing this APMS system is to save time and minimize the possibility of human error through a standard reporting design.

4.3 Business Process Management Challenges

In order to improve the business performance in this competitive marketplace (US – Canada), it requires the company to be on top of every business transaction. During the active

fishing season (usually between June and December) *Inshore Fisheries* requires an efficient and effective way to deal with the daily barrage of information on product management from its customers, from its vessels, and from other vessels from whom it may be advantageous to purchase supplies of raw fish products to keep up inventories in the face of customer demand. For example, *Inshore Fisheries* needs to know very clearly about product inventory, especially for frozen products, and to be prepared to respond quickly to customer demands for fresh fish products. The APMS solution represents a very effective tool for what *Inshore Fisheries* needs to improve its production and customer order processes. The following section describes the management requirements and selection of particular business office software as well as the development of the APMS model specifically in response to the needs of the company.

4.3.1 The management requirements

Thus, *Inshore Fisheries* are required to develop their internal business process management and recording systems in order to meet the continuous business change in markets, exchange rates, supply and demand for its fish products. Accordingly, company managers are at an advantage if they find new and efficient ways to keep track of products inventory, client business lists, production control, and operating incomes and bottom line profits. This model can assist *Inshore Fisheries* in its ability to respond quickly based on the potential customer sales and inventory and production possibilities.

Inshore Fisheries' requirements for a business process management system include the functionality that allows the production, inventory, invoice, and income statement reports to be linked together. This means that changes in one business report (e.g., production worksheets) are immediately reflected and updated in related reports (e.g., product inventories). The model proposed here provides an easy means of providing work reports data and then automatically uses these data to generate business reports, e.g., an invoice report for a customer, a worksheet production statement, and an operating income statement report for managers. The user can use the GUI to retrieve these reports.

4.3.2 Inshore Office software

Currently for *Inshore's* office work, the staff uses the Corel Suite software package primarily to produce basic office reports, letters, etc. Driven by the staff and the awareness that much more of their analysis and report writing could be automated, they were seeking to develop more advanced business process systems.

Driven by its clientele and the rise of electronic communications, *Inshore* has experienced some difficulties in its compatibility and use of its office software systems. Consequently, they have been gradually shifting from Corel software to the more widely used Microsoft Office software.

Given current capabilities of *Inshore Fisheries*, it wanted to develop a system which has more business process management functionalities. The challenge for this research was to provide an easy to use software solution adapted to their specific needs in the fishing industry to match the way in which they currently do their work. As part of this discussion, the selection of an appropriate software platform took place as a precursor to development and implementation of the proposed model. After some deliberation, it was decided jointly that the Microsoft Office would be the most convenient platform on which to develop the system. Moreover, stimulated by the discussion, the company was enticed to move up its move to Microsoft Office.

4.4 The database design of Inshore

Before designing the full model and after the software issue was resolved, the database setup was required to be developed first. In order to maintain the compatibility of the different software applications, Microsoft Access was adopted primarily in order to make the database compatible with the other parts of this software that would also be using the business office products from Microsoft, as well as to take advantage of the software capabilities associated with the suite of Microsoft software.

As described in section 3.4 above, the database is composed of nine interrelated tables. Thus, *Inshore*'s database consists of 9 related tables designed specifically for this case problem as follows. Details of the following text description is also illustrated in the Microsoft Access Figure 4.1 found below.

1. **ComInfo Table:** contains the customer information. It has nine fields including the customer name, contact information, and the address of this customer.
2. **Inv_Com Table:** contains the information about matching the customer with the invoice. It has invoice No., date, customer, and ship 4 fields, where "customer" refers to the name in the table of ComInfo and the field ship means the shipping company name here.
3. **Invoice Table:** contains four fields, invoice No., item No., quantity, and unit price, where invoice No. relates to the invoice No. in the table of Inv_Com.
4. **Inventory Table:** consist of seven fields, species name, item No, size, skin on or boneless, package amount, quantity, and index. The field item No. relates to the field item No. in the table of Invoice. The field index here is used as a key of sorting all the items.
5. **Species_Info Table:** contains the basic production information for each of the species, such as the wage, cost, packing expense, and freight, where the field species relates to field species in the table of Inventory.
6. **Production Table:** contains 10 fields here, which includes production No., date, species, vessel name, processing area, rate of return, and three kinds of exchange rate. The field species relates to the field species in the table of Species_Info.
7. **Frozen Products Table:** contains only three field which are production No., item No., and weight. Item No. relates to the item No. in the table of Inventory. Production No. relates to the production No. in the table of Production.
8. **Fresh Products Table:** contains threes field also which are production No, size and weight. The production No. relates to the field of production No. in the table of Production.
9. **Slab Products Table:** contains 3 fields; they production No., the amount of slab in frozen, and in fresh. The production No. relates to the field of production No. in the table of Production.

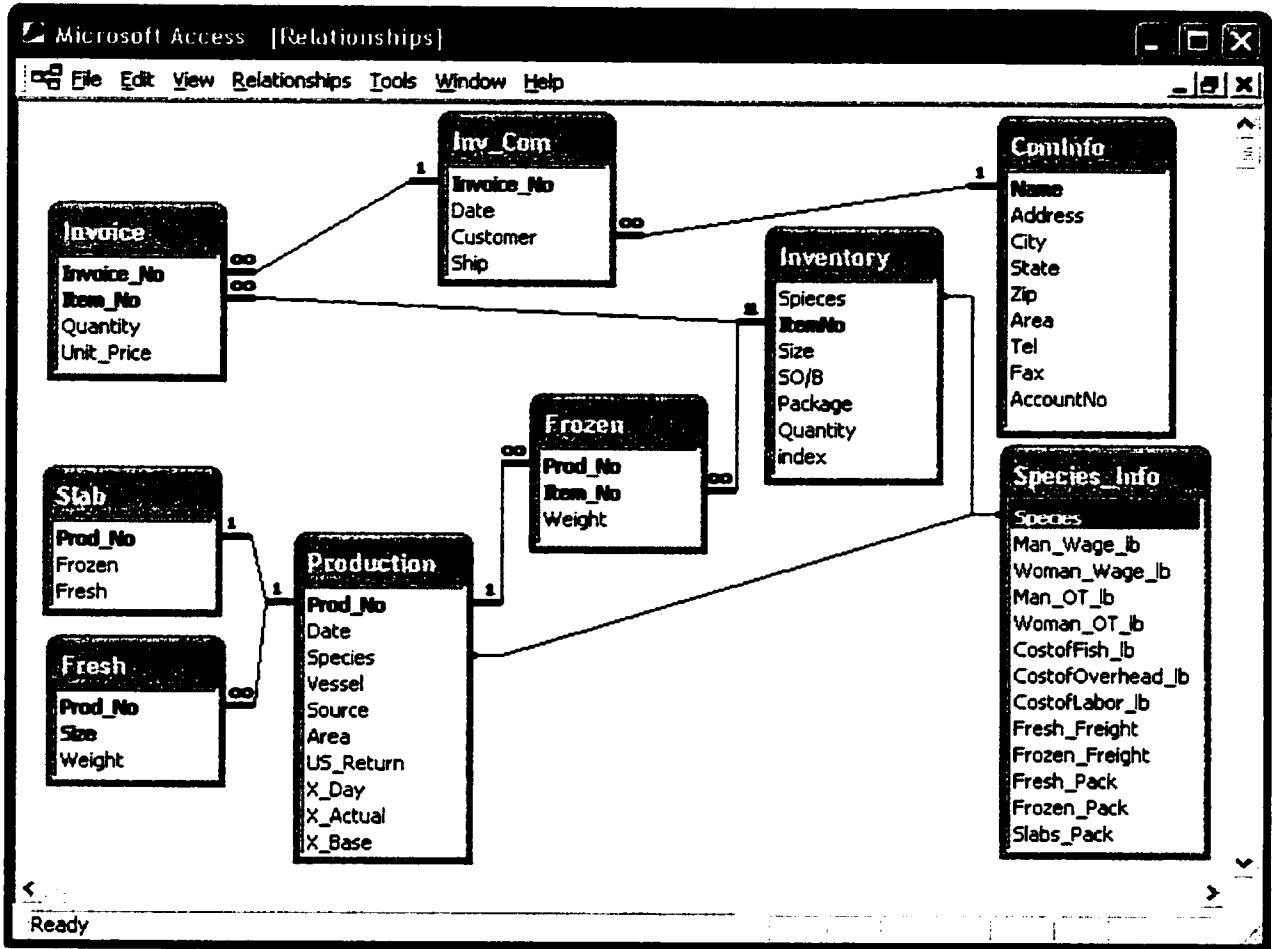


Figure 4.1: The Database Relationship of Inshore Fisheries

4.5 The Inshore APMS Model

In order to handle the problems mentioned above, a customized design software solution is required. Thus, the automatic product management system (APMS) was developed for this purpose. Since this model is being applied for *Inshore Fisheries* in this case, it needed to have an opening window which introduces the company name:

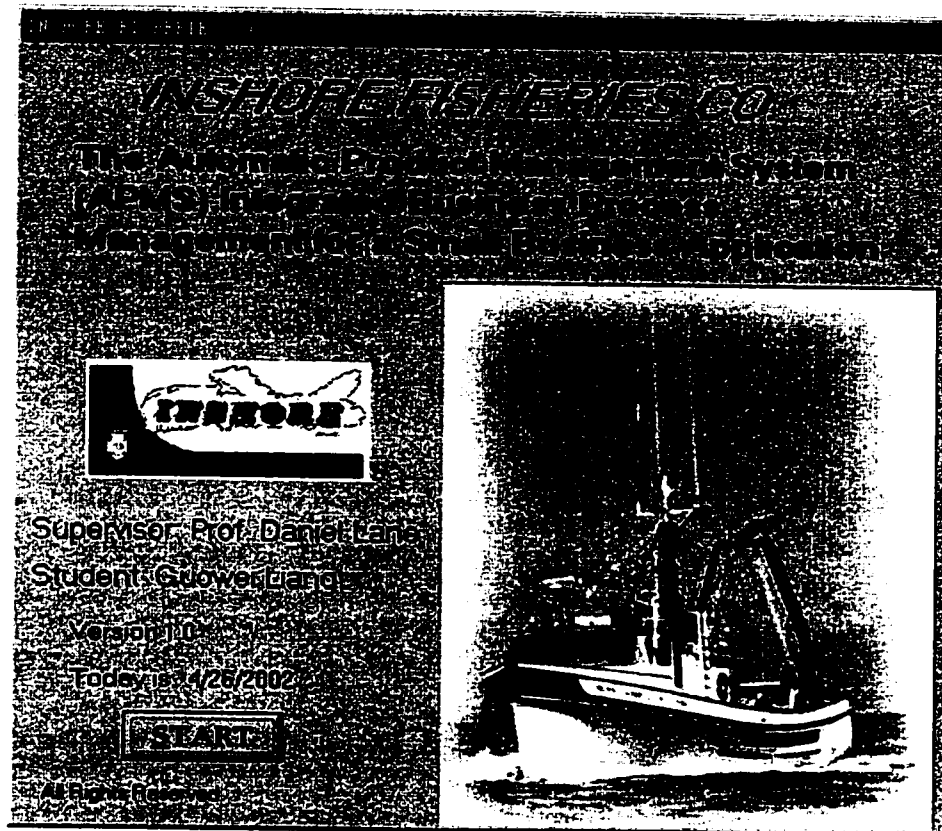


Figure 4.2: The Opening window

The Figure 4.2 above tells the user what this software is for and what it does generally.

As described in part 3.2.1, the system required a user-friendly GUI. This GUI must fulfill all the necessary business functions referred to in 3.2.1. Based on the idea illustrated in Figure 3.1, a customized GUI was developed using Visual Basic (Figure 4.3). Clicking on the command button “Start” from Figure 4.2 brings the user to this GUI. It includes two main parts: one is the input functions and the other is the output functions. Inside the input frame, it consists of two functions, both production and customer order. Users can choose either one by just clicking on the appropriate command button. Inside the output frame, it is composed of four functions which can generate four different operation performance reports. Further, this GUI remains on the user’s desktop of the computer unless the user clicks on the “EXIT” button at the end of the session. The functions for this application are described in detail in the following six subsections: two for input and four for output.

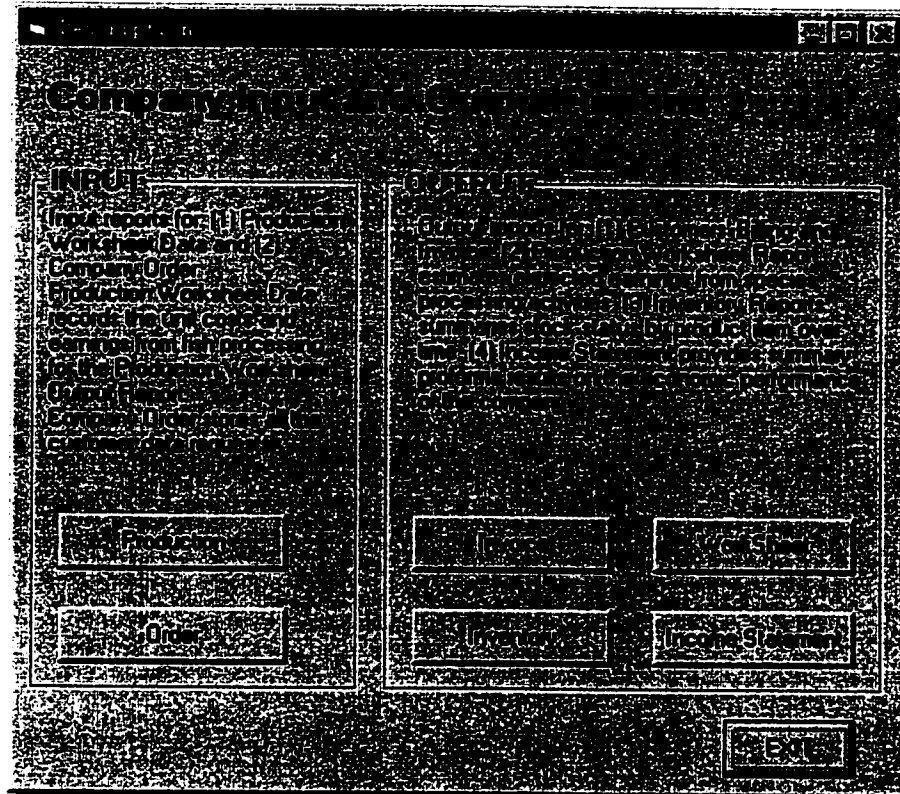


Figure 4.3: The VB Graphic User Interface (GUI) for Inshore Fisheries

There are seven command buttons in the above figure. The functions for these point and click buttons are defined as follows:

1. **Production:** a button for processing the production procedure.
2. **Order:** a button for customer order.
3. **Invoice:** a button for generating the performance report of customer order based on recorded customer invoice numbers.
4. **WorkSheet:** a button for generating the performance report of production based on recorded numbered production worksheets.
5. **Inventory:** a button for generating the performance report of inventory status by product.
6. **Income Statement:** a button for generating the *pro forma* performance report of the operation situation of the firm over a given time period.
7. **Exit:** close the application.

The following section itemizes the functionality of the *Inshore Fisheries* APMS.

4.5.1 Input — Production

According to the GUI shown in Figure 4.3, users can select either “Production” or “Order”. Then a series of pop-up windows will come up to ask the user to fill in the related information. After going through these windows, the data are stored in the linked database tables.

For illustration purposes, consider the process of using the Inshore model for the “Production” function. Figure 4.4 shows the pop-up window when clicking on “Production”. This window asks the user to choose the species of interest that will be produced. Each species (product) will have some basic required information such as freight cost, packing cost, and other processing costs as noted in the input frame of Figure 4.5 below. Users can modify this information if necessary. In order to view or update the data, the user needs simply to click on the button “View/Update Species Production Data” shown in Figure 4.4. This command button can trigger another pop-up window which shows all the fundamental information related to this species (Figure 4.5). Closing the window of Figure 4.5 will return the user back to the window of production (Figure 4.4).

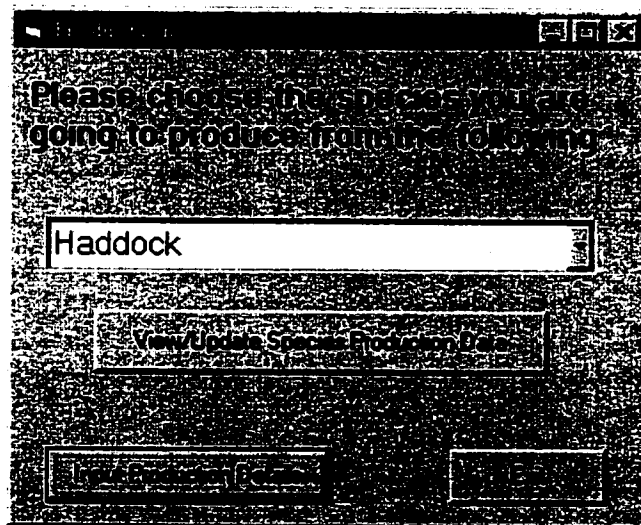


Figure 4.4: The Production Window for Choosing a Species

From Figure 4.4, there are three command buttons and a drop down combo box. These are:

- Combo Box: contains 7 species for users to choose. These species are restricted to: Catfish, Cod, Flounder, Haddock, Ocean Perch, Pollock, and Turbot.
- View/Update Species Production Data - allows the user to confirm that the data of the species information is correct, otherwise, it can modify or update the data.
- Input Production Details - it brings users to the Figure 4.5 window where they can process the production data.
- Exit: closes this window.

Field	Value
Species	Haddock
Wage Rate Men	0.1
Wage Rate Women	0.09
OT Men	0.25
OT Women	0.1
Fish Cost	0.35
Livehead Cost	0.11
Labor Cost	0.38
Fresh Feed Cost	0.06
Fresh Feed Cost	0.11
Fresh Fish Cost	0.09
Fresh Fish Cost	0.07
Stale Feed Cost	0.05

Figure 4.5: Species Information Summary

For Figure 4.5 below, there are two command buttons and an arrow button. These are:

- OK: it means all data in this window are valid; the user is returned to the production window (Figure 4.4)

- Exit: if there are no modifications or any updates on the window, it brings the user back to Figure 4.4.
- Arrow Button: it allows the user to go through all the species one by one.

After the user checks and/or updates all the information of the species that is ready to be processed, the user can proceed to the next step by clicking on “OK”. At this point, the user is returned main window “Production”, Figure 4.4. Now, the user must click on button “Input Production Details” (Figure 4.4) in order to process this species. Figure 4.6 shows the processing window for the chosen species. Users can then type the related information into these input text boxes. In order to maintain software integrity, the buttons of “Fresh”, “Frozen”, and “Slabs” in Figure 4.6 are initially “faded out”, i.e., no entries are permitted until a production number is provided from the outset.

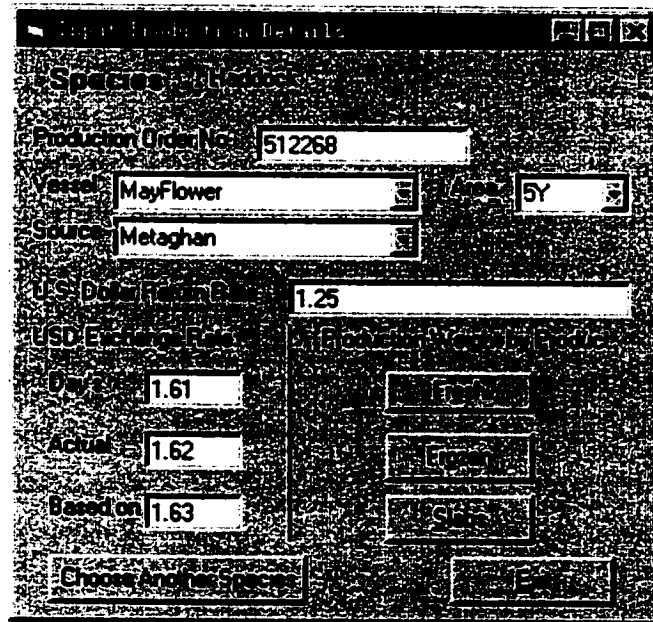


Figure 4.6: Input Production Details

The Inout Production Details menu contains five command buttons and three combo boxes (Figure 4.6). These are:

Production Weight data by product for the various categories of harvested fish including-

- Fresh: process fresh products.
- Frozen: process frozen products.
- Slab: process slab products.

In order to complete the entire production process, the user has to click on these three buttons one by one. They are illustrated below in Figure 4.7, Figure 4.8, and Figure 4.9, respectively .

- Choose Another Species: process another species other than the one which is currently identified at the top of the menu or go back to the previous window after the user finishes processing the current species.
- Exit: close this window without processing.

Provide fixed information about the company's production definition, i.e., vessel, area, source:

- Vessel Combo Box: list all the company's vessel names for single selection.
- Area Combo Box: list all the production areas for single selection.
- Source Combo Box: list all sources available for single selection.

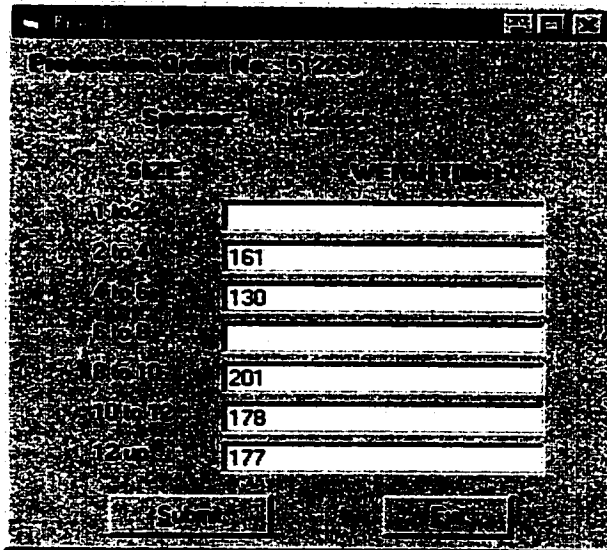


Figure 4.7: Production Process for Fresh

There are two command buttons in the Fresh window.

- Submit: submit all the amounts and insert into database.
- Exit: close this window without enter any information.

The screenshot shows a window titled "Production Order No. 01226". It contains two columns of data, each with a header "ITEMS" and "QUANTITY". The data is as follows:

ITEMS	QUANTITY	ITEMS	QUANTITY
HA1B1		HA21	167
HA1B2	312	HA22	
HA1B3		HA23	
HA1B4	166	HA24	200
HA1S1		HA25	
HA1S2		HA26	
HA1S3	245	HA27	245
HA1S4		HA28	
HA2B1	111	HA29	
HA2B2	222	HA30	
HA2B3		HA31	
HA2B4		HA32	169

At the bottom of the window, there are two buttons: "Submit" and "Exit".

Figure 4.8: The Production Process for Frozen

There are also two buttons in the Frozen window; they have the same function as the corresponding buttons in the Fresh window.

- Submit: submit and insert all the values into the database.
- Exit: close the window without saving any data.

For the Slabs window (shown in Figure 4.9), its two command buttons function in the same manner as the Frozen and Fresh windows above.

- Submit: submit and insert all the values into the database.
- Exit: close the window without saving any data.

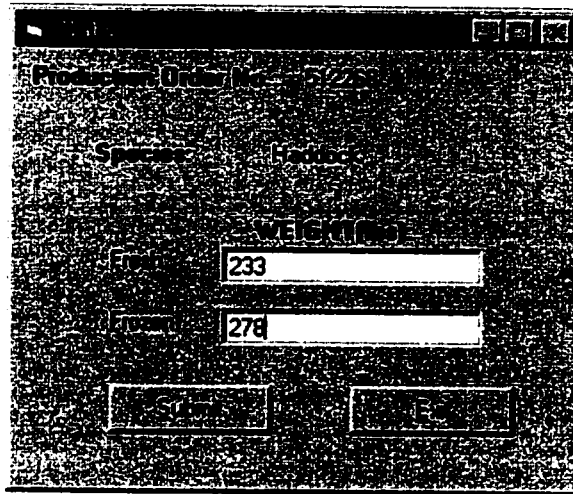


Figure 4.9: The Production Process for Slabs

After finishing the execution of the Fresh, Frozen, and Slabs production input tables, the production process is complete and the user can return to the main menu, e.g., Figure 4.3.

4.5.2 Input — Order

Next we review the input function “Order”. When the user clicks on the button “Order” from the GUI (Figure 4.3), the following window (Figure 4.10) pops up. This software can deal with three kinds of customers:

1. If the customer has previously ordered seafood products from *Inshore Fisheries*, then the customer is an “existing customer” and the user can retrieve the customer name and information in the combo box by clicking on the down arrow.
2. If the customer is a new customer who has never ordered any products from *Inshore Fisheries* before, this customer and tombstone information must be added to the contact information of *Inshore’s* customer database.
3. For an existing client, one may also want to update the contact information in the *Inshore’s* database because this customer has perhaps already changed its telephone number or even moved to another address.

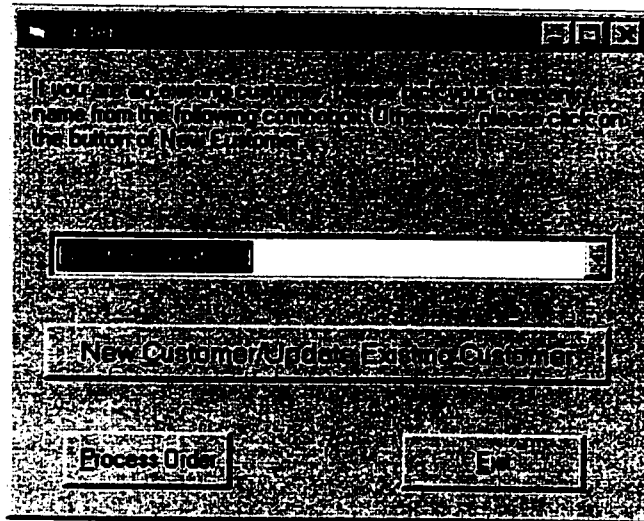


Figure 4.10: Order Window for Choosing a Customer

From the figure above, there are three command buttons in this Order window.

- **New Customer/Update Existing Customer:** for a new customer to enter some basic contact information or for an existing customer to update the contact information.
- **Process Order:** orders can be placed directly by clicking this button.
- **Exit:** close the window without doing anything.

For the customer belonged to the first category, “existing customer”, one can proceed directly to click the button “Process Order” after finding the name from the combo box. But for the other two kinds of customers (category 2 and 3 above), they need to click on the button “New Customer/Update Existing Customer”. This activates a new pop-up window (Figure 4.11) which shows the information requirements for the customer database. New customers can enter the contact information by clicking on the button “Add New Record”. On this pop-up window, the user can further review the other existing customers’ information using the arrow button.

Customer Information

New Customer: Existing Customer: Use the arrow buttons to view all customers one by one.

Company Name: Boston Seafood

Address: 280 Addison Ave.

City: Boston

State/Province: Massachusetts

Zip/Postal Code: MA7219

Area Code: 678 Fax No: 3331233

Telephone No: 3331234

Submit Exit

Figure 4.11: Customer Information

There are command buttons and an arrow button in this Customer information window.

- Add New Record: all the text boxes will become blank by clicking on this button. A new customer can thus be entered with the related information into these empty fields.
- Submit: after the customer data has been entered all the data for the customer is updated with the information in this window. The Submit button records these data into the corresponding tables in the database.
- Exit: close this window without any modification.
- Arrow button: allow the user to view all the customers one by one through the database table.

Clicking “Process Order” (Figure 4.10) activates a Microsoft Excel spreadsheet. This is a pre-defined spreadsheet template. The spreadsheet contains a company order form and a built-in button “SUBMIT / SAVE” which turns on the spreadsheet macro. This macro transfers all the

data from the Excel spreadsheet to the related tables in the database file. On this Excel spreadsheet, the user is required to fill in the cells that are highlighted. These include such data as: INVOICE NUMBER, INVOICE DATE, and Ship VIA fields. Further, before the user enters the quantity of any particular product item; one should look in advance at the Stock-on-Hand column which shows how many units of the product are remaining in the inventory. The quantity of the order must not exceed the storage amount in the inventory. After filling out this order, the user can simply click on the macro button “SUBMIT/SAVE” to complete the entire order process and transfer data to the database.

4.5.3 Output — Invoice

After the customer places the order, the customer can anticipate receiving the bill. *Inshore Fisheries* has to send this customer the invoice to demand payment. As a basic customer requirement, the software APMS has developed a function to generate a customized company invoice. Every invoice requires an invoice number corresponding to the invoice number in the order sheet. According to this invoice number, the APMS can retrieve all the related information about the customer order including the total amount of money that that customer should pay.

When *Inshore Fisheries* finishes the order for the customer, it needs to send the invoice to the customer as soon as possible in order to collect the bill. Thus, the operator of APMS clicks on the button of “Invoice” from the GUI (please refer to Figure 4.3). This activates a pop-up window which asks the user to pick up an invoice number from a drop-down combo box (Figure 4.12). Originally, the combo box of Invoice No and all three text boxes of Date, Customer, and Ship VIA are empty until the operator chooses the invoice number (e.g. 112211 in Figure 4.12) when the text boxes are filled with the corresponding data from that invoice. These data are retrieved from the database and are consistent with the invoice number.

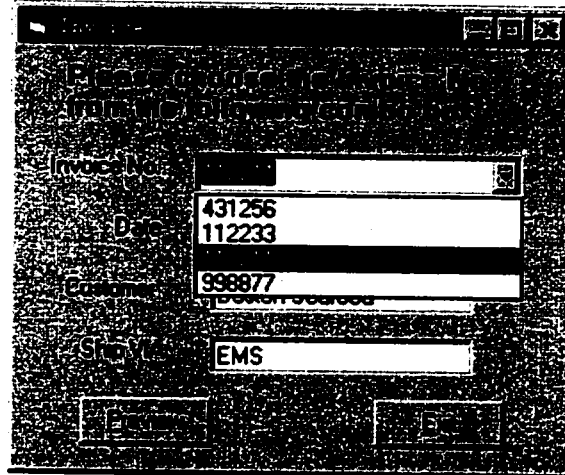


Figure 4.12: Invoice Pop-up Window

Before sending the invoice to the customer, the operator of APMS needs to take a look at the report in advance. This operator also wants to make sure everything on the invoice report is correct, especially the amount of payment. By clicking Preview shown on Figure 4.12, it brings the operator to a Microsoft Excel Spreadsheet which has all the information for the customer who just placed an order with an invoice number of 112211 (see also Appendix A for a full example of this output report). Moreover, the button of “Preview” is in nonclickable (“gray”) at the beginning. It becomes enabled only by choosing a invoice number from the combo box.

4.5.4 Output — WorkSheet

When a production process is completed, a report is required to keep track of the company internal management. There is a production number for each production process that is entered by the user. In order to generate the production report, or “WorkSheet”, the production number is very important. Based on this requirement of the users of the software solution, APMS has this function already built-in. The user of the APMS can retrieve all the information related to any particular production number and generate a WorkSheet report for that particular production run.

When *Inshore Fisheries* finishes the production process, the operator can then click on the button “WorkSheet” from the GUI (Figure 4.3). This activates another pop-up window (Figure

4.13). Similar to the pop-up window of the invoice (Figure 4.12), all the text boxes there are initially empty. This Work Sheet window asks the operator to select a production number from the drop-down combo box. After picking up a production number, e.g.,512268 in the case of Figure 4.13, the fields for Date, Vessel, Source, and Area are filled in automatically. These data are retrieved from the database which is built up when processing the product items. All these data are consistent with the unique production number.

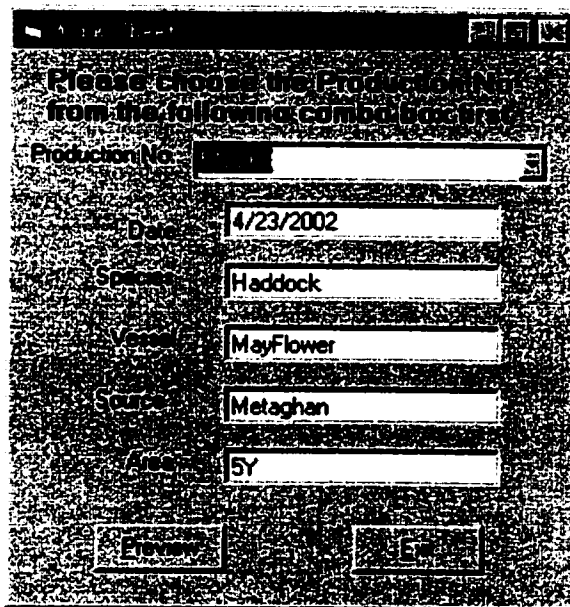


Figure 4.13: Work Sheet Pop-up Window

After choosing the production number, the operator can view the report by clicking on the . Preview button. Note that the “Preview” button is disabled when opening the Work Sheet Pop-up window. Choosing a production number will enable this button. Clicking on the “Preview” button activates a Microsoft Excel spreadsheet which contains all the related information about the production number 512268 (see also Appendix B for a full example of this output report). The operator can also further modify the values for any cell on this spreadsheet. When all the values are correct, this spreadsheet can be saved and printed as a final report for the production process.

4.5.5 Output — Inventory

Inventory is a critical factor for the product management inside a SME due to its small operation scale. A SME must keep its inventory at a proper level which is not too high or too low. If the inventory level were too high, it would occupy lots of space and thus take a great amount of cost to maintain all the product items. If the inventory level were too low, it would not have enough product items for the customer demand and thus reduce the profit of the company. Therefore, this requires that the product management software has a function to monitor the inventory level.

Inside the APMS, the inventory level goes down after the customer places an order and the inventory level goes up right after the production process has occurred. Thus, *Inshore Fisheries* has to monitor the inventory closely. It can do this by clicking on the button of “Inventory” on the GUI (Figure 4.3). This action brings the operator an inventory pop-up window (Figure 4.14). The inventory in the database is organized by different species. There are seven species in total. The operator of the software APMS can choose any species of interest from the drop-down combo box.

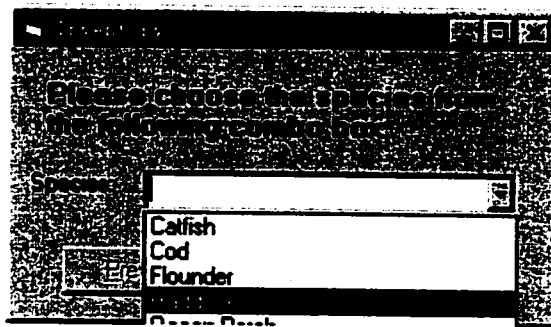


Figure 4.14: Inventory Pop-up Window

The selection of a species also enables the “Preview” button. The operator thus can go to preview the inventory level (see also Appendix C for a full example of this output report). This report shows the current amount of every product item for that species. The operator can make a decision on whether to produce more or not. And this operator can know how much to produce

for each item. Further, the inventory level can be import directly to the customer order spreadsheet which tells the operator how much is remaining in the inventory.

4.5.6 Output — Income Statement

Economic performance analysis is a very essential step for the product management of an SME. SMEs have tight budgets relative to larger companies. *Inshore Fisheries*, must have a short payment collection period and a short cash flow period in order to increase its competition capability. A strong financial performance is required for *Inshore Fisheries* in the ravaged fisheries sector. Thus, it needs a tool to monitor its economic performance situation very closely.

To meet this requirement, the APMS contains a built-in function to evaluate economic performance inside the software. The operator can observe a button called “Income Statement” on the software GUI (Figure 4.3). After clicking on this button, a pop-up window appears (Figure 4.15). There are a few text boxes on this window. Each of these text boxes represents an exogenous fixed input item related to the computation of the company’s income statement.

Item	Value
Other Revenue:	\$211.00
Miscellaneous Costs	\$145.00
License and Other Fixed Expense	\$134.00
Depreciation Expense	\$231.00
Repairs Expense	\$205.00
Insurance Expense	\$431.00
Interest Expense	\$100.00

Figure 4.15: Income Statement Pop-up Window

After the operator confirms all these monetary values are correct by clicking “OK”, the “Preview” button is enabled. This button will again activate the Microsoft Excel spreadsheet where the report of income statement is generated. All the numbers on the income statement pop-up window (Figure 4.15) are imported into this spreadsheet and calculated automatically. The final report is illustrated in Appendix D.

4.6 Use of APMS Model

The development of the APMS for Inshore Fisheries was made possible with the assistance of Yvon d’Entremont, Business Office manager, and company co-owner Jean-Guy d’Entremont of Pubnico, Nova Scotia. Their assistance and feedback throughout the period of software design and testing provided a valuable opportunity to observe the basis requirements of the company’s operations.

The company, through its officials demonstrated throughout a desire to become more efficient. Opportunities for increased efficiency presented themselves in the carrying out of basic business tasks, including data input and generation of summary and consolidated report writing, and the monitoring and tracking of regular business transactions including customer purchase requests, and production processing activities. While the company was very familiar with electronic processing to speed communications (e.g., FAX, email, cellular telephone and other wireless communication devices were in common usage around the office well as on board company vessels and down on the wharf), there was less familiarity with the possibilities to enhance business office performance and customer relations via enhanced use of business software.

This project has not yet been fully accepted by the company as a complete replacement for the routine, manual exercises that is has been designed to replace. However, as familiarity with the software grows, it is anticipated that the company will seek out still further opportunities to automate its business processing activities using similar, custom-designed software.

4.7 Implementation in Inshore

Considering the fact that the APMS system has not yet been accepted by *Inshore Fisheries*, it is important to think about the possible barriers to implementation.

Under the current system, the business operation manager has the complete control of all transactions and often must make many decisions quickly. While the APMS could assist in these decisions, the implementation of having to consult the computer to define transactions may be felt to be too disruptive.

As well, potential users may have difficulty understanding the software and therefore are immediately skeptical about its benefits. These factors taken into account suggest the need for a “hands-on” training workshop be implemented to assist *Inshore fisheries* users in answering questions and dealing with potential contingencies that may arise during software use.

5 RESEARCH ISSUES

This section describes several issues that need to be addressed during the course of this research project. These issues include the implementation of the BPM system for *Inshore* at the local office level, and the future direction of research including wide area, Intranet and Internet applications of the BPM toward a practical e-business system.

The following sub-sections discuss these issues in terms of: (1) the practice of Quattro Pro and Microsoft Excel 2000, (2) the application to other BPM models, (3) the potential Internet application, and (4) model implementation.

5.1 The selection of the spreadsheet software

Currently in the commercial market, we may find two major spreadsheet business solutions: Quattro Pro[®] and Microsoft Excel[®]. They both specialize on the data analysis and have a great deal of built-in formulas. For the spreadsheet solution in this case, we may have to make a trade-off between Quattro Pro 9 and Microsoft Office 2000.

Quattro Pro allows us to create professional spreadsheets for data management and analysis. It provides all the tools to produce tables, financial forms, lists, databases, charts, reports, or any other type of data-oriented document. Using Quattro Pro, we can create a notebook that contains a single spreadsheet or an entire project, complete with spreadsheets, reports, charts, and a slide show presentation.

Quattro Pro does have lots of advantages for data handling on any one particular spreadsheet. Unfortunately, in the course of this work, problems had been discovered in the use of Quattro Pro to handle multiple linking tasks that are required. Under the operating system (OS) of Windows (as used at *Inshore*), these multiple linking activities create some fatal problems, e.g.,

- Inability to save files after data modification or link update;
- Inability to open files with links to other notebooks;

- Computer crashes;
- Tedious updating of links when transferring the spreadsheet system between two machines.

Similar to Quattro Pro 9 described above, Microsoft Office 2000 is also a spreadsheet software with very strong business-oriented functionalities. Again, we use Excel in one of the most popular operating systems - Windows. Basically, all the problems in Quattro Pro that are listed above do not appear in Excel. Excel provides us with an easier linking environment.

Compared to Quattro Pro 9, Excel is much more compatible to Windows OS. Moreover, Excel provides a good working environment for customized spreadsheet model design as required for the *Inshore* project. It is easier to update all the references that connect to other cells and other sheets. It has a strong error and type checking function. Further, the latest version of Visual Basic for Applications is included with Excel 2000 although we did not use any VBA in this project. Visual Basic 6 provides Excel with features such as model input user forms and support for additional ActiveX controls than Quattro Pro.

All in all, given the problems that were experienced with Quattro Pro, Excel is much easier to expand and manipulate.

5.2 Other BPM model applications

Although the BPM model described here is a tailor-made model for *Inshore Fisheries*, the principle still applies to other companies, and other business sectors of the economy, especially where SMEs are involved.

As described in a generic BPM system descriptions in the third part of this thesis, this model does not necessarily focus on any particular business case. Actually, any SME company that has problems similar to those of *Inshore Fisheries* can take advantage of this software solution. In other words, if a company, or an independent subunit of the company which is going

to automate its business transaction process and willing to have the customized print-out of some performance reports can really think about this Automatic Product Management System (APMS).

5.3 Model Scale – Local vs Internet

The current APMS model presented here focuses only on Intranet-type applications, i.e., a private network established within an enterprise. It is not open to the public. It is only used by the employees of the enterprise on proprietary company data. Thus, it does have its constraints.. The customers have no access to this network.

An intranet only pays attention to the information communication among its own company network. It can be set up by connecting several local area networks (LANs) together or taking advantage of leased line in the wide area network (WAN). All the employees share the same resources inside this company. In the current information age, most Intranets are connecting to the outside Internet through one or more gateway computers. The people outside this company can obtain access to its network anywhere and anytime. This creates convenience for the customers because they can do the business by just clicking at their homes or offices through the Internet. As a consequence, this may increase the efficiency and thus bring in more business to the SME. Further, it saves lots of paperwork due to electronic business transaction processing.

5.4 Model Implementation

This model generally is suitable for the use in SMEs because it does not take up too many resources to implement and use. It only requires a PC which has Microsoft Excel and Access (contained as part of the Microsoft Office suite) installed. Further, it does not require too much computer skill or professional knowledge for the person who will ultimately control this software because it has a neat and straightforward interface. What the operator needs is to follow the software instruction step by step. Also because of its simplicity, it does have some constraints. For example, it is only compatible with Windows operating system. It does not fit into Unix or Macintosh.

6 CONCLUSION

For an SME, the product management is a very critical business procedure. An efficient product management system provides the dynamics for a company. It increases the vitality for the company in such a severe competitive market. It also can minimize the product maintenance cost and improve the profit position of the company's stakeholders.

This research focused on how to automate this business procedure. It also built a software prototype APMS (Automatic Product Management System). Aside from applying it to *Inshore Fisheries*, it was also shown that it could be applied more broadly into other SMEs. The functionality of the APMS is extendable and concise. According to the different requirements of different customer, the APMS can be further developed to a much more complex software by adding some other functions.

All in all, we believe that APMS is a simple software solution for SMEs that need assistance in business process management. We also believe the commercialization of APMS is a challenging but meaningful job when the software design takes into account the specific needs of the company.

6.1 Application to Other SMEs

The nature of SMEs is small-to-medium size business ventures. They do not have a large operational scale. They do not have a large amount of budget. They do not have too many resources. Usually, they cannot afford their own R&D team. On the other hand, just because they are not large, they are faced with market dynamics. They can reach a management decision quickly. They can change their operation very easily in order to meet the different economic situation. Based on these common characteristics of SMEs, APMS can easily be modified for use in each company. According to the different requirements of each company, APMS can make the consistent modifications which fit into this company's environment. For example, if a company is producing toys, APMS can just change the product items what are currently fish into toys. As

well, the database is modified based on the data that this toy company will generate. Although there are many subtle modifications for the software, the fundamental principal is still the same.

6.2 Software Requirement

The entire software was developed using the tools from Microsoft. The database was designed using Microsoft Access. The spreadsheet solution was Microsoft Excel. The macros inside the spreadsheet were developed using VBA. The coding language for the entire program was Visual Basic. The integrated development environment was VB 6.0.

This software is a standalone version so far which means someone has to install it on his/her own machine if he/she wants to use it. It is not a software which can be running on a server and providing service to all the client machines because it is not a network version. Furthermore, the current system has been designed and will run only on the Windows operating system, including Windows 98, 2000, ME, and XP.

6.3 Extendibility

APMS right now is a very simple software concept. It only fulfills some basic functionalities of a company: production process, order, invoice report, production report (worksheet), inventory report, and income statement. These functions may not satisfy a company's needs in a more globalized and competitive market. Due to the simplicity of APMS, it can be extended very easily. According to the further requirements of the customer, it can add functions to the APMS. For instance, APMS may satisfy the current economic environment when a company develops this software. Due to the uncertainty of the market, the economic situation may require that the product management reflect some other aspects of the company (e.g. cash flow statement) in the future. It is unreasonable to ask this SME to develop another software in order to meet this new requirement because it involves lots of investment of human power and cost. It can just add this function to the GUI and develop it using Visual Basic. Moreover, the database has to add some tables and fields which are consistent with the new functions.

6.4 On-line order

So far, the entire research has been focused on the internal usage of the company. We assumed that this software solution, APMS, would be used only by the employees inside this company. The operator of the APMS will take the order for the customer.

Considering the popularity of World Wide Web, it is appealing to place order forms on-line which would allow the customer to fill it out directly on the company's web page. This is also a competitive strategy which can increase the sales volume. Customers can easily access the web page of the company. It is convenient for them if they can just order on-line.

7 BIBLIOGRAPHY

Acs, Z.J., and Audretsch, D.B. 1990. *Innovation and Small Firms*. Cambridge, Mass: The MIT Press.

Balderson, D.W. 2000. *Canadian Entrepreneurship and Small Business Management*. (fourth edition), McGraw-Hill Ryerson.

Baldwin, J.R., William, C., Can Le, and Tom, P. 1994. *Strategies for Success: A Profile of Growing Small and Medium-sized Enterprises (GSMEs) in Canada*. Business and Labor Market Analysis Division, Statistics Canada.

Banco BBA. 2000. *Banco BBA Creditanstalt S.A. Office 2000 Case Study*. January.
<http://www.microsoft.com/technet/prodtechnol/office/case/bbastdy.asp>.

Brydon, M. 1997. *Microsoft Access Tutorials*.
<http://mis.bus.sfu.ca/tutorials/MSAccess/tutorials.html>.

Canada. 1994. *Growing Small Business*.p.3 – p.5. Industry Canada / Statistics Canada

Canada. 2001. *Corporations Directorate*. Industry Canada,
http://strategis.ic.gc.ca/sc_mrksv/corpdire/engdoc/homepage.html.

Chester, T. 1995. *Advanced Data Access with MS Excel*. Microsoft Co.
<http://www.microsoft.com/technet/excel/technote/xl302.asp?a=frame>.

Company Coach. 2001. *Agris Grows More Business Creating Quality Proposals Quickly*.
<http://www.companycoach.com/agris.htm>.

Inshore. 2001. *Inshore Fisheries Limited*. <http://www.auracom.com/~inshore/index.html>

JWalk & Associates. 2001. *Microsoft Excel*. <http://www.j-walk.com/ss/excel/>.

Kindra, G.S. 1997. *A survey to Assess the Views of SMEs regarding Business Formation Options*. WP 97-31, Faculty of Administration, University of Ottawa. 36p.

Lefebvre, É., Lefebvre, L. A. CIRANO and Polytech, É. 2000. *SMEs, Exports and Job Creation: A Firm-level Analysis*. No. 26, Micro-Economic Policy Analysis, Industry Canada, <http://strategis.ic.gc.ca/SSG/ra01789e.html>.

Lokken, C. 1995. *Creating a Custom UI for MS Excel Applications*. Microsoft Co. <http://www.microsoft.com/technet/excel/technote/xl209.asp?a=frame>.

Lyons, T. 2001. *Maytag International Refines Data Distribution with DI-Diver*. Dimensional Insight, Inc. February. <http://www.dssresources.com/cases/maytaginternational.htm>.

MemberWorks. 2001. *MemberWorks Incorporated*. <http://www.memberworks.com/company.htm>

Metastorm. 2001. *Metastorm corporation web site*. <http://www.metastorm.com/default.asp> , and http://www.metastorm.com/products/overview_index.asp

McDougall, G. and Swimmer, D. 1997. *Business Strategies of SMEs and large Firms in Canada*. WP 16, Micro-Economic Policy Analysis, Industry Canada, <http://strategis.ic.gc.ca/SSG/ra01578e.html>.

Microsoft. 1998. *MS Excel 2000 Product Enhancements Guide*. Microsoft Corporation, <http://www.microsoft.com/technet/prodtechnol/office/evaluate/ofcpeg.asp>.

OECD. 2000a. *Small and Medium-sized Enterprises: Local Strength, Global Reach*. June. http://www.oecd.org/publications/Pol_brief/2000/2000_02.pdf.

OECD. 2000b. *OECD Small and Medium Enterprise Outlook*. 222p.

OECD. 1997. *Globalization and Small and Medium Enterprises (SMEs)*, Vol. 1 (Synthesis Report) and Vol. 2 (Country Studies), Paris.

Powells, D.L. 1995. *MS Excel Development in the Real World*. Microsoft Co. <http://www.microsoft.com/technet/excel/technote/mmaexcel.asp?a=frame>.

Power, D.J. 2000. *A Brief History of Spreadsheets*. <http://www.dssresources.com/history/sshistoryv.html>.

Rainier. 2001. *Rainier Technology - Infrastructure Technologies Case Study: MemberWorks Windows 2000, SQL Server 2000, and Web-based solutions*. http://www.rainier.com/it_cs4.html

Schumpeter, J. A. 1962. *Capitalism, Socialism and Democracy*. (third edition), Harper Torchbooks, 431p.

UNCTAD. 2000. *TNC-SME Linkages for Development Issues-experiences-best practices*. 113p.

WIPO. 2001. *WIPO Milan Forum on Intellectual Property and Small and Medium-sized Enterprises*. www.wipo.org/eng/meetings/2001/ip_milan/pdf/ip_mil01_4a.pdf.

Wright, P. 1998. *Beginning Visual Basic 6*, p432, Wrox Press.

Appendix A

The Invoice Statement

This is a printed-screen of the invoice statement that is designed for *Inshore Fishery*. It contains all information described in Table 2: Sample Invoice Statement Design. After making every transaction, *Inshore* must record all the customer information, including the customer name, account number, date, ordered product items and the total balance and so on and so for. This is also the bill that will be sent to this customer in order to charge for the payment.

INSHORE FISHERIES LTD.						INVOICE STATEMENT		
TO Customer: Boston Seafood Ship VIA: EMS						INVOICE NUMBER 112211 INVOICE DATE 01/22/99 PAYMENT DUE DATE 04/22/99		
DESCRIPTION						UnitPrice	Total Price	BALANCE
SPECIES	SIZE	SO/B	PKG	Item#	QTY			\$
CATFISH	12 up	SO	15	CA3SS	34	\$ 180.00	\$ 6,120.00	\$ 6,120.00
COD	4 to 8	SO	5	C01S1	45	\$ 180.00	\$ 8,100.00	\$ 14,220.00
COD	8 to 12	B	5	C02B1	25	\$ 180.00	\$ 4,500.00	\$ 18,720.00
COD	12 up	SO	10	C03S2	12	\$ 180.00	\$ 2,160.00	\$ 20,880.00
FLOUNDER	4 to 8	B	20	FL1B4	11	\$ 180.00	\$ 1,980.00	\$ 22,860.00
FLOUNDER	12 up	B	5	FL3B1	15	\$ 180.00	\$ 2,700.00	\$ 25,560.00
HADDOCK	4 to 8	SO	10	HA1S2	26	\$ 180.00	\$ 4,680.00	\$ 30,240.00
OCEAN PENCE	8 to 12	B	15	OP2B5	22	\$ 180.00	\$ 3,960.00	\$ 34,200.00
POLLOCK	4 to 8	SO	15	P01S3	32	\$ 180.00	\$ 5,760.00	\$ 39,960.00
TURBOT	8 to 12	SO	20	TU2S4	111	\$ 180.00	\$ 19,980.00	\$ 59,940.00
TURBOT	12 up	SO	5	TU3S1	34	\$ 180.00	\$ 6,120.00	\$ 66,060.00
TURBOT	12 up	SO	15	TU3S3	22	\$ 180.00	\$ 3,960.00	\$ 70,020.00
CATFISH	4 to 8	B	15	CA1B5	25	\$ 180.00	\$ 4,140.00	\$ 74,160.00
CATFISH	4 to 8	SO	15	CA1S3	11	\$ 180.00	\$ 1,980.00	\$ 76,140.00
						Total	\$ 76,140.00	
						Credit		\$ 76,140.00
Please pay last amount in Balance column.								
Please detach portion below and send with your remittance.								
FROM: INSHORE FISHERIES LTD. Middle West Pubnico, Yarmouth Country, Nova Scotia BOX 2ND						INVOICE NUMBER 112211 PAYMENT DUE DATE 04/22/99 TOTAL AMOUNT DUE \$ 76,140.00		
MAKE ALL CHECKS PAYABLE TO INSHORE FISHERIES LTD.						ADDRESS QUESTIONS TO: Yves d'Entremont (902) 762-2522		
THANK YOU FOR YOUR BUSINESS!								

Note: All data presented in this Appendix have been disguised and do not reflect the actual business performance of *Inshore Fisheries*.

Appendix B

The Production Worksheet

This is printed-screen of the production worksheet which is designed for *Inshore Fishery*. It contains all information described in Table 1: Sample Production Report Design. After *Inshore* harvests the products, it sells fresh fish on direct order from customers and flash freezes other product and puts in inventory. During the transactions, especially the foreign (US primarily) business, the exchange rate is very important because it affects the profit in Canadian dollar directly. Therefore, Inshore sets a bottom line for the exchange rate before calculating expected profit. If the actual exchange rate goes under this bottom line, then they may not make a profit on the transaction.

INSHORE FISHERIES LTD. Work Sheet											
Date:	4/23/2002		Production No:	512208		Area:	5Y				
			Source:	Metaghan		Vessel:	MayFlower				
Wages:			Costs per lb:		Regular	Overtime					
	Men		\$0.10		\$0.25						
	Women		\$0.09		\$0.10						
Day's exchange	\$1.01 based on		\$1.03 exchange								
Actual exchange	\$1.02										
Returns:											
Fresh											
Based on Day											
Size	Weight (lbs)	US Return	US Amount	CDN Return	CDN Amount	Actual CDN					
2 to 4	161	\$ 1.25	\$ 201.25	\$ 2.04	\$ 328.04	\$ 328.03					
4 to 6	130	\$ 1.25	\$ 162.50	\$ 2.04	\$ 264.87	\$ 263.25					
8 to 10	201	\$ 1.25	\$ 251.25	\$ 2.04	\$ 409.54	\$ 407.03					
10 to 12	178	\$ 1.25	\$ 222.50	\$ 2.04	\$ 362.67	\$ 360.45					
12 up	177	\$ 1.25	\$ 221.25	\$ 2.04	\$ 360.64	\$ 358.43					
10 to 12		\$ -	\$ -	\$ -	\$ -	\$ -					
12 up		\$ -	\$ -	\$ -	\$ -	\$ -					
Fresh Totals			\$ 1,058.75		\$ 1,725.76	\$1,715.18					
Frozen											
Based on Day											
Species	Size	SO/B	Pkg	Item#	Qty	Weight (lbs)	US Return	US Amount	CDN Return	CDN Amount	Actual
Haddock	4 to 8	B	10	HA1B2	312	3120	\$ 1.25	\$ 3,900.00	\$ 2.04	\$ 6,357.00	\$ 6.3
Haddock	4 to 8	B	20	HA1B4	106	3320	\$ 1.25	\$ 4,150.00	\$ 2.04	\$ 6,764.50	\$ 6.7
Haddock	4 to 8	SO	15	HA1S3	246	3675	\$ 1.25	\$ 4,593.75	\$ 2.04	\$ 7,487.81	\$ 7.4
Haddock	8 to 12	B	5	HA2B1	111	555	\$ 1.25	\$ 693.75	\$ 2.04	\$ 1,130.81	\$ 1.1

(to be continued...)

(...continued)

Frozen											
Species	Size	SOB	Pkg	Item#	Qty	Weight (lbs)	US Return	US Amount	CDN Return	CDN Amount	Actual CDN
Haddock	4 to 8	8	10	HA1B2	312	3120	\$ 1.25	\$ 3,900.00	\$ 2.04	\$ 6,357.00	\$ 6,318.00
Haddock	4 to 8	8	20	HA1B4	166	3320	\$ 1.25	\$ 4,150.00	\$ 2.04	\$ 6,764.50	\$ 6,723.00
Haddock	4 to 8	SO	15	HA1S3	246	3875	\$ 1.25	\$ 4,593.75	\$ 2.04	\$ 7,467.81	\$ 7,441.88
Haddock	8 to 12	8	5	HA2B1	111	555	\$ 1.25	\$ 693.75	\$ 2.04	\$ 1,130.81	\$ 1,123.88
Haddock	8 to 12	8	10	HA2B2	222	2220	\$ 1.25	\$ 2,775.00	\$ 2.04	\$ 4,523.25	\$ 4,405.50
Haddock	8 to 12	SO	5	HA2S1	167	835	\$ 1.25	\$ 1,043.75	\$ 2.04	\$ 1,701.31	\$ 1,690.88
Haddock	8 to 12	SO	20	HA2S4	200	4000	\$ 1.25	\$ 5,000.00	\$ 2.04	\$ 8,150.00	\$ 8,100.00
Haddock	12 up	8	15	HA3B3	246	3875	\$ 1.25	\$ 4,593.75	\$ 2.04	\$ 7,467.81	\$ 7,441.88
Haddock	12 up	SO	20	HA3S4	169	3380	\$ 1.25	\$ 4,225.00	\$ 2.04	\$ 6,888.75	\$ 6,844.50
Frozen Total:						1837		\$ 30,975.00		\$ 50,489.25	\$ 50,179.50

Total Weight					
Category	Weight (lbs)	CDN Return	CDN Amount	Amount/lb	Actual CDN
Fresh-Frozen	25627	\$ 1.87	\$ 48,037.81	\$ 1.87	\$ 51,894.68
Slabs I	511	\$ 1.87	\$ 957.87	\$ 1.87	\$ 1,034.78
		\$ 1.87	\$ -		\$ -
Total Returns			\$ 48,995.68		\$ 52,929.46

Costs				Reference	
Item	Weight (lbs)	Costs per lb	Total Cost	Frozen	Weight (lbs)
Cost of fish	25627	\$0.35	\$ 8,969.45		233
Overhead	25627	\$0.11	\$ 2,818.97		278
Labour	25627	\$0.38	\$ 9,738.26		
Freight Costs					
Fresh	847	\$0.06	\$ 50.82		
Frozen	24780	\$0.11	\$ 2,725.80		
Pack Costs					
Fresh	847	\$0.09	\$ 76.23		
Frozen	24780	\$0.07	\$ 1,734.60		
Slabs	511	\$0.05	\$ 25.55		
Fish Totals		\$ 1.22	\$ 20,139.88		

Nonfish Costs			
Item	Weight (lbs)	Costs per lb	Total Cost
Nonfish Costs	25627	\$ 0.67	\$ 17,170.23

Total Returns				
Item	Weight (lbs)	Value per lb	Value	Actual CDN
Returns	25627	\$ 1.91	\$ 48,995.68	\$ 52,929.46
Cost	25627	\$ 1.02	\$ 26,139.88	\$ 26,139.88
Profit		\$ 0.89	\$ 22,855.80	\$ 26,789.57

Slabs	
Item	Weight (lbs)
Frozen	233
Fresh	278
Total	511
Yield	1.99 X

Per: _____

Note: All data presented in this Appendix have been disguised and do not reflect the actual business performance of *Inshore Fisheries*.

Appendix C

The Inventory Report

This is a printed-screen of the inventory report that is designed for *Inshore Fisheries*. It contains all information described in Table 3: Sample Inventory Report Design. Inshore keeps the inventory of all the product items in order to keep track of the availability for next customer order. It can serve as a measure which tells when it is the time to produce which product item. It also records the production date, ordered date and the customer name.

INSHORE FISHERIES LTD.					
Inventory Statement				Date: 4/26/2002	
Species	ItemNo	Size	SO/B	Pkg	Qty
1 HADDOCK	HA1B1	4 to 8	B	5	631
2 HADDOCK	HA1B2	4 to 8	B	10	749
3 HADDOCK	HA1B3	4 to 8	B	15	900
4 HADDOCK	HA1B4	4 to 8	B	20	513
5 HADDOCK	HA1S1	4 to 8	SO	5	335
6 HADDOCK	HA1S2	4 to 8	SO	10	288
7 HADDOCK	HA1S3	4 to 8	SO	15	593
8 HADDOCK	HA1S4	4 to 8	SO	20	327
9 HADDOCK	HA2B1	8 to 12	B	5	428
10 HADDOCK	HA2B2	8 to 12	B	10	537
11 HADDOCK	HA2B3	8 to 12	B	15	599
12 HADDOCK	HA2B4	8 to 12	B	20	138
13 HADDOCK	HA2S1	8 to 12	SO	5	305
14 HADDOCK	HA2S2	8 to 12	SO	10	365
15 HADDOCK	HA2S3	8 to 12	SO	15	419
16 HADDOCK	HA2S4	8 to 12	SO	20	321
17 HADDOCK	HA3B1	12 up	B	5	343
18 HADDOCK	HA3B2	12 up	B	10	336
19 HADDOCK	HA3B3	12 up	B	15	589
20 HADDOCK	HA3B4	12 up	B	20	321
21 HADDOCK	HA3S1	12 up	SO	5	138
22 HADDOCK	HA3S2	12 up	SO	10	493
23 HADDOCK	HA3S3	12 up	SO	15	337
24 HADDOCK	HA3S4	12 up	SO	20	485

INSHORE FISHERIES LTD.

Middle West Pubnico,
Yarmouth Country, Nova Scotia
BOX 2MO

ADDRESS QUESTIONS TO
Yvon d'Entremont
(902) 762-2522

Note: All data presented in this Appendix has been disguised and does not reflect the actual business performance of *Inshore Fisheries*.

Appendix D

The Income Statement

This is a printed-screen of the income statement that is designed for *Inshore Fishery*. It contains all information described in Table 4: Sample Income Statement Design. This statement measures the profitability of the business after a certain period. It shows an overall performance situation of the company.

Income Statement INSHORE FISHERIES LTD. For The Period Ending: 23-Apr-02 (all numbers in Canadian dollars)		
	Year to Date	
	Amount	% of Revenue
Operating Revenue		
Fresh	\$ 3,702.30	2.27%
Frozen	\$ 157,417.25	96.62%
Slab	\$ 1,589.25	0.98%
Other	\$ 211.00	0.13%
Total Operating Revenue	\$ 162,919.80	100.00%
Operating Costs		
Cost of Fish	\$ 27,659.80	16.98%
Overhead	\$ 3,693.08	2.27%
Freight	\$ 3,604.68	2.21%
Packing	\$ 5,606.32	3.45%
Subtotal	\$ 40,563.88	24.89%
Miscellaneous Costs	\$ 145.00	0.09%
Labour Costs	\$ 30,000.00	18.42%
Total Operating Cost	\$ 70,708.96	43.40%
Net Operating Income	\$ 92,210.84	56.60%
Other Expenses		
License & Other Fixed	\$ 134.00	0.08%
Depreciation	\$ 231.00	0.14%
Repairs	\$ 205.00	0.13%
Insurance	\$ 431.00	0.26%
Interest Expense	\$ 100.00	0.06%
Total Other Expenses	\$ 1,101.00	0.68%
Total Costs	\$ 71,809.96	44.08%
Profit (before income taxes)	\$ 20,400.84	12.52%
Income Tax Payable (50%)	\$ 10,200.42	6.26%
NET PROFIT (LOSS)	\$ 10,200.42	6.26%
Date	23/Apr/02	
Prepared by		

Note: All data presented in this Appendix have been disguised and do not reflect the actual business performance of *Inshore Fisheries*.

Appendix E

Source Code of the APMS

This appendix presents the entire source code of the APMS model developed and applied for the Inshore Fisheries case study. The code is written in Visual Basic Version 6.0 from Microsoft.

The code contains approximately 2600 lines of code. Code functionality includes tasks for the user interface GUI and the linkages of input and output of the system. The executable file has size of 320KByte. The code runs on a Windows 98, ME, 2000, or XP operating systems.

The code also contains documentation throughout to describe the functionality of each frame and window. Further details and access to this source code are available from the author.

```

1 'Code for start frame
2 Option Explicit
3
4 Private Sub cmdStart_Click()
5     frmDescription.Show vbModal, Me
6     Load frmDescription
7     frmDescription.Show
8     Unload Me
9 End Sub
10
11 Private Sub Form_Load()
12     lblToday.Caption = "Today is: " & Date
13 End Sub
14
15 'Code for the frame of Description
16 Option Explicit
17 Public cn As ADODB.Connection
18
19 Private Sub cmdExit_Click()
20     cn.Close
21     Set cn = Nothing
22
23     End
24 End Sub
25
26 Private Sub cmdIncome_Click()
27     frmIncome.Show vbModal, Me
28 End Sub
29
30 Private Sub cmdInven_Click()
31     frmInventory.Show vbModal, Me
32 End Sub
33
34 Private Sub cmdInvoice_Click()
35     frmInvoice.Show vbModal, Me
36 End Sub
37
38 Private Sub cmdOrder_Click()
39     frmOrder.Show vbModal, Me
40 End Sub
41
42 Private Sub cmdProd_Click()
43     frmProd.Show vbModal, Me
44 End Sub
45
46 Private Sub cmdWS_Click()
47     frmWS.Show vbModal, Me
48 End Sub
49
50 Private Sub Form_Load()
51     Set cn = New ADODB.Connection
52     'cn.Open "Provider =
53     Microsoft.Jet.OLEDB.4.0;Data
54     Source=C:\Documents and
55     Settings\Guowei\Desktop\ADMS\ADMS.mdb
56     ;Persist Security Info=False"
57     cn.Open "Provider =
58     Microsoft.Jet.OLEDB.4.0;Data
59     Source=ADMS.mdb;Persist Security
60     Info=False"
61 End Sub
62
63
64 'Code for frame of Production
65 Option Explicit
66 Public sSpecies As String
67 'sSpecies
68 'Set sSpecies = "hello wei"
69 'Private Sub cmbSpecies_Change()
70 ' If cmbSpecies.DataChanged = True Then
71 '     sSpecies = cmbSpecies.Index
72 ' End If
73 'End Sub
74 Public bChange As Boolean
75
76 Private Sub cmbSpecies_Click()
77     If cmbSpecies.DataChanged = True Then
78         sSpecies = cmbSpecies.Text
79     End If
80 End Sub
81
82 Private Sub cmdExit_Click()
83     Unload Me
84 End Sub
85
86 Private Sub cmdProcess_Click()
87     'Dim iProdTemp As Variant

```

```

88 'iProdTemp = Shell("c:\Program
89 Files\Microsoft Office\Office\EXCEL " &
90 "D:\EEEE\Sys\Thesis\ADMS\WorkSheetTempl
91 ate.xlt", 1)
92 'ActivateProd
93 Unload Me
94 frmProdInput.Show 'vbModal, Me
95 End Sub
96
97 Private Sub cmdSpecies_Click()
98 Unload Me
99 frmSpecies.Show 'vbModal, Me
100 End Sub
101
102 Private Sub Form_Load()
103 'bChange = False
104 cmbSpecies.AddItem ("Catfish")
105 cmbSpecies.AddItem ("Cod")
106 cmbSpecies.AddItem ("Flounder")
107 cmbSpecies.AddItem ("Haddock")
108 cmbSpecies.AddItem ("OceanPerch")
109 cmbSpecies.AddItem ("Pollock")
110 cmbSpecies.AddItem ("Turbot")
111
112 If bChange = True Then
113     cmbSpecies.Text = sSpecies
114 Else
115     sSpecies = cmbSpecies.Text
116 End If
117 End Sub
118
119
120 'Code for frame of Species Production Data
121 Option Explicit
122
123 Private Sub cmdExit_Click()
124 Unload Me
125 End Sub
126
127 Private Sub cmdOk_Click()
128 dataSpecies.Recordset.MoveLast
129 dataSpecies.Refresh
130 Unload Me
131 frmProd.Show 'vbModal, Me
132 'frmProd.cmbSpecies.Text =
133 frmProd.sSpecies
134 End Sub
135
136 Private Sub Form_Load()
137 dataSpecies.ToolTipText = "Use the arrow
138 key to switch species"
139 txtSpecies.ToolTipText = "Name of the
140 Species"
141 'txtArea.ToolTipText = "Name of that Area"
142 'txtVessel.ToolTipText = "Name of the
143 Vessel"
144 'txtSource.ToolTipText = "Where the fish
145 come from"
146 txtManWage.ToolTipText = "Canadian
147 dollars per pound"
148 txtWomanWage.ToolTipText = "Canadian
149 dollars per pound"
150 txtManOT.ToolTipText = "Canadian dollars
151 per pound"
152 txtWomanOT.ToolTipText = "Canadian
153 dollars per pound"
154 txtFishCost.ToolTipText = "Canadian
155 dollars per pound"
156 txtOHCost.ToolTipText = "Canadian
157 dollars per pound"
158 txtLaborCost.ToolTipText = "Canadian
159 dollars per pound"
160 txtFreshFreight.ToolTipText = "Canadian
161 dollars per pound"
162 txtFrozenFreight.ToolTipText = "Canadian
163 dollars per pound"
164 txtFreshPack.ToolTipText = "Canadian
165 dollars per pound"
166 txtFrozenPack.ToolTipText = "Canadian
167 dollars per pound"
168 txtSlabPack.ToolTipText = "Canadian
169 dollars per pound"
170
171 Select Case frmProd.sSpecies
172 Case "Catfish"
173     dataSpecies.Recordset.MoveFirst
174 Case "Cod"
175     dataSpecies.Recordset.MoveNext

```

```

176     Case "Flounder"
177         dataSpecies.Recordset.Move 2
178     Case "Haddock"
179         dataSpecies.Recordset.Move 3
180     Case "OceanPerch"
181         dataSpecies.Recordset.Move 4
182     Case "Pollock"
183         dataSpecies.Recordset.Move 5
184     Case "Turbot"
185         dataSpecies.Recordset.MoveLast
186
187 End Select
188 frmProd.bChange = True
189 End Sub
190
191
192 'Code for frame of Input Production Details
193 Option Explicit
194 Public exist As Boolean
195 Private Sub cmdBack_Click()
196     Unload Me
197     frmProd.Show 'vbModal, Me
198 End Sub
199
200 Private Sub cmdExit_Click()
201     Unload Me
202 End Sub
203
204 Private Sub cmdFresh_Click()
205     'Unload Me
206     frmFresh.Show 'vbModal, Me
207 End Sub
208
209 Private Sub cmdFrozen_Click()
210     'Unload Me
211     frmFrozen.Show 'vbModal, Me
212 End Sub
213
214 Private Sub cmdSlabs_Click()
215     'Unload Me
216     frmSlab.Show 'vbModal, Me
217 End Sub
218
219 Private Sub Form_Load()
220     'Dim cn As Connection
221     Dim rs1 As Recordset
222     Dim rs2 As Recordset
223     Dim rs3 As Recordset
224     DimNewItem As String
225
226     'Set cn = New Connection
227     'cn.Open
228     "Provider=Microsoft.Jet.OLEDB.4.0;Data
229     Source=ADMS.mdb;Persist Security
230     Info=False"
231
232     Set rs1 = New Recordset
233     rs1.Open "Vessel", frmDescription.cn,
234     adOpenForwardOnly, adLockReadOnly
235     rs1.MoveFirst
236     Do Until rs1.EOF
237         frmProdInput.cboVessel.AddItem
238         (rs1("Vessel_Name"))
239     rs1.MoveNext
240 Loop
241     rs1.Close
242
243     Set rs2 = New Recordset
244     rs2.Open "Source", frmDescription.cn,
245     adOpenForwardOnly, adLockReadOnly
246     rs2.MoveFirst
247     Do Until rs2.EOF
248         frmProdInput.cboSource.AddItem
249         (rs2("Source_Name"))
250     rs2.MoveNext
251 Loop
252     rs2.Close
253
254     Set rs3 = New Recordset
255     rs3.Open "Area", frmDescription.cn,
256     adOpenForwardOnly, adLockReadOnly
257     rs3.MoveFirst
258     Do Until rs3.EOF
259         frmProdInput.cboArea.AddItem
260         (rs3("Area_Name"))
261     rs3.MoveNext
262 Loop
263     rs3.Close

```

```

264 'cn.Close
265
266 Set rs1 = Nothing
267 Set rs2 = Nothing
268 Set rs3 = Nothing
269 'Set cn = Nothing
270
271 cmdFresh.Enabled = False
272 cmdFrozen.Enabled = False
273 cmdSlabs.Enabled = False
274 'lblSpecies.Caption = "Species: " &
275 frmProd.sSpecies
276 val_species.Caption = frmProd.sSpecies
277 exist = False
278 End Sub
279
280 Private Sub txtProduction_KeyDown(KeyCode
281 As Integer, Shift As Integer)
282     cmdFresh.Enabled = True
283     cmdFrozen.Enabled = True
284     cmdSlabs.Enabled = True
285 End Sub
286
287
288 'Code for frame of Fresh
289 Option Explicit
290
291 Private Sub cmdExit_Click()
292     Unload Me
293 End Sub
294
295 Private Sub cmdSubmit_Click()
296
297 Dim cmd As String
298 Dim fconnection As Connection
299 Dim tmp As Boolean
300 tmp = False
301
302 Set fconnection = New Connection
303 fconnection.Open
304 "Provider=Microsoft.Jet.OLEDB.4.0;Data
305 Source=ADMS.mdb;Persist Security
306 Info=False"
307 If Text1.Text <> "" Then
308     If frmProdInput.exist = False Then
309         cmd = "insert into Production values ("
310         cmd = cmd + """" +
311         frmProdInput.txtProduction.Text + """"
312         cmd = cmd + ", " + "Date()"
313         cmd = cmd + ", "" +
314         frmProdInput.val_species.Caption + """"
315         cmd = cmd + ", "" +
316         frmProdInput.cboVessel.Text + """"
317         cmd = cmd + ", "" +
318         frmProdInput.cboSource.Text + """"
319         cmd = cmd + ", "" +
320         frmProdInput.cboArea.Text + """"
321         cmd = cmd + ", " + frmProdInput.txtRate.Text
322         cmd = cmd + ", " +
323         frmProdInput.txtDayX.Text
324         cmd = cmd + ", " +
325         frmProdInput.txtActualX.Text
326         cmd = cmd + ", " +
327         frmProdInput.txtBaseX.Text
328         cmd = cmd + ")""
329         fconnection.Execute cmd
330         frmProdInput.exist = True
331     End If
332
333     cmd = "insert into Fresh values ("
334     cmd = cmd + """" + val_prod_no.Caption + """"
335     cmd = cmd + ", "" + Label1.Caption + """"
336     cmd = cmd + ", " +
337     cmd = cmd + Text1.Text
338     cmd = cmd + ")""
339     fconnection.Execute cmd
340     tmp = True
341 End If
342
343 If Text2.Text <> "" Then
344     If frmProdInput.exist = False Then
345         cmd = "insert into Production values ("
346         cmd = cmd + """" +
347         frmProdInput.txtProduction.Text + """"
348         cmd = cmd + ", " + "Date()"
349         cmd = cmd + ", "" +
350         frmProdInput.val_species.Caption + """"

```

```

351 cmd = cmd + ", " +
352 frmProdInput.cboVessel.Text + ""
353 cmd = cmd + ", " +
354 frmProdInput.cboSource.Text + ""
355 cmd = cmd + ", " +
356 frmProdInput.cboArea.Text + ""
357 cmd = cmd + ", " + frmProdInput.txtRate.Text
358 cmd = cmd + ", " +
359 frmProdInput.txtDayX.Text
360 cmd = cmd + ", " +
361 frmProdInput.txtActualX.Text
362 cmd = cmd + ", " +
363 frmProdInput.txtBaseX.Text
364 cmd = cmd + ")"
365 fconnection.Execute cmd
366 frmProdInput.exist = True
367 End If
368 cmd = "insert into Fresh values ("
369 cmd = cmd + "" + val_prod_no.Caption + ""
370 cmd = cmd + ", " + Label2.Caption + ""
371 cmd = cmd + ", "
372 cmd = cmd + Text2.Text
373 cmd = cmd + ")"
374 fconnection.Execute cmd
375 tmp = True
376 End If
377
378 If Text3.Text <> "" Then
379 If frmProdInput.exist = False Then
380 cmd = "insert into Production values ("
381 cmd = cmd + "" + frmProdInput.txtProduction.Text
382 + ""
383 cmd = cmd + ", " + "Date()"
384 cmd = cmd + ", " +
385 frmProdInput.val_species.Caption + ""
386 cmd = cmd + ", " + frmProdInput.cboVessel.Text
387 + ""
388 cmd = cmd + ", " + frmProdInput.cboSource.Text
389 + ""
390 cmd = cmd + ", " + frmProdInput.cboArea.Text +
391 ""
392 cmd = cmd + ", " + frmProdInput.txtRate.Text
393 cmd = cmd + ", " + frmProdInput.txtDayX.Text
394 cmd = cmd + ", " + frmProdInput.txtActualX.Text

```

```

395 cmd = cmd + ", " + frmProdInput.txtBaseX.Text
396 cmd = cmd + ")"
397 fconnection.Execute cmd
398 frmProdInput.exist = True
399 End If
400 cmd = "insert into Fresh values ("
401 cmd = cmd + "" + val_prod_no.Caption + ""
402 cmd = cmd + ", " + Label3.Caption + ""
403 cmd = cmd + ", "
404 cmd = cmd + Text3.Text
405 cmd = cmd + ")"
406 fconnection.Execute cmd
407 tmp = True
408 End If
409
410 If Text4.Text <> "" Then
411 If frmProdInput.exist = False Then
412 cmd = "insert into Production values ("
413 cmd = cmd + "" +
414 frmProdInput.txtProduction.Text + ""
415 cmd = cmd + ", " + "Date()"
416 cmd = cmd + ", " +
417 frmProdInput.val_species.Caption + ""
418 cmd = cmd + ", " +
419 frmProdInput.cboVessel.Text + ""
420 cmd = cmd + ", " +
421 frmProdInput.cboSource.Text + ""
422 cmd = cmd + ", " +
423 frmProdInput.cboArea.Text + ""
424 cmd = cmd + ", " + frmProdInput.txtRate.Text
425 cmd = cmd + ", " +
426 frmProdInput.txtDayX.Text
427 cmd = cmd + ", " +
428 frmProdInput.txtActualX.Text
429 cmd = cmd + ", " +
430 frmProdInput.txtBaseX.Text
431 cmd = cmd + ")"
432 fconnection.Execute cmd
433 frmProdInput.exist = True
434 End If
435 cmd = "insert into Fresh values ("
436 cmd = cmd + "" + val_prod_no.Caption + ""
437 cmd = cmd + ", " + Label4.Caption + ""
438 cmd = cmd + ", "

```

```

439 cmd = cmd + Text4.Text
440 cmd = cmd + ")"
441 fconnection.Execute cmd
442 tmp = True
443 End If
444
445 If Text5.Text <> "" Then
446 If frmProdInput.exist = False Then
447 cmd = "insert into Production values ("
448 cmd = cmd + "" +
449 frmProdInput.txtProduction.Text + ""
450 cmd = cmd + ", " + "Date()"
451 cmd = cmd + ", " +
452 frmProdInput.val_species.Caption + ""
453 cmd = cmd + ", " +
454 frmProdInput.cboVessel.Text + ""
455 cmd = cmd + ", " +
456 frmProdInput.cboSource.Text + ""
457 cmd = cmd + ", " +
458 frmProdInput.cboArea.Text + ""
459 cmd = cmd + ", " + frmProdInput.txtRate.Text
460 cmd = cmd + ", " + frmProdInput.txtDayX.Text
461 cmd = cmd + ", " + frmProdInput.txtActualX.Text
462 cmd = cmd + ", " + frmProdInput.txtBaseX.Text
463 cmd = cmd + ")"
464 fconnection.Execute cmd
465 frmProdInput.exist = True
466 End If
467 cmd = "insert into Fresh values ("
468 cmd = cmd + "" + val_prod_no.Caption + ""
469 cmd = cmd + ", " + Label5.Caption + ""
470 cmd = cmd + ", "
471 cmd = cmd + Text5.Text
472 cmd = cmd + ")"
473 fconnection.Execute cmd
474 tmp = True
475 End If
476
477 If Text6.Text <> "" Then
478 If frmProdInput.exist = False Then
479 cmd = "insert into Production values ("
480 cmd = cmd + "" + frmProdInput.txtProduction.Text
481 + ""
482 cmd = cmd + ", " + "Date()"
483 cmd = cmd + ", " +
484 frmProdInput.val_species.Caption + ""
485 cmd = cmd + ", " + frmProdInput.cboVessel.Text
486 + ""
487 cmd = cmd + ", " + frmProdInput.cboSource.Text
488 + ""
489 cmd = cmd + ", " + frmProdInput.cboArea.Text +
490 ""
491 cmd = cmd + ", " + frmProdInput.txtRate.Text
492 cmd = cmd + ", " + frmProdInput.txtDayX.Text
493 cmd = cmd + ", " + frmProdInput.txtActualX.Text
494 cmd = cmd + ", " + frmProdInput.txtBaseX.Text
495 cmd = cmd + ")"
496 fconnection.Execute cmd
497 frmProdInput.exist = True
498 End If
499 cmd = "insert into Fresh values ("
500 cmd = cmd + "" + val_prod_no.Caption + ""
501 cmd = cmd + ", " + Label6.Caption + ""
502 cmd = cmd + ", "
503 cmd = cmd + Text6.Text
504 cmd = cmd + ")"
505 fconnection.Execute cmd
506 tmp = True
507 End If
508
509 If Text7.Text <> "" Then
510 If frmProdInput.exist = False Then
511 cmd = "insert into Production values ("
512 cmd = cmd + "" +
513 frmProdInput.txtProduction.Text + ""
514 cmd = cmd + ", " + "Date()"
515 cmd = cmd + ", " +
516 frmProdInput.val_species.Caption + ""
517 cmd = cmd + ", " +
518 frmProdInput.cboVessel.Text + ""
519 cmd = cmd + ", " +
520 frmProdInput.cboSource.Text + ""
521 cmd = cmd + ", " +
522 frmProdInput.cboArea.Text + ""
523 cmd = cmd + ", " + frmProdInput.txtRate.Text
524 cmd = cmd + ", " +
525 frmProdInput.txtDayX.Text

```

```

526 cmd = cmd + ", " +
527 frmProdInput.txtActualX.Text
528 cmd = cmd + ", " +
529 frmProdInput.txtBaseX.Text
530 cmd = cmd + ")"
531 fconnection.Execute cmd
532 frmProdInput.exist = True
533 End If
534 cmd = "insert into Fresh values ("
535 cmd = cmd + "" + val_prod_no.Caption + ""
536 cmd = cmd + ", " + Label7.Caption + ""
537 cmd = cmd + ", "
538 cmd = cmd + Text7.Text
539 cmd = cmd + ")"
540 fconnection.Execute cmd
541 tmp = True
542 End If
543
544 If tmp = True Then
545 frmProdInput.exist = True
546 frmProdInput.txtProduction.Enabled = False
547 frmProdInput.cboArea.Enabled = False
548 frmProdInput.cboSource.Enabled = False
549 frmProdInput.cboVessel.Enabled = False
550 frmProdInput.txtActualX.Enabled = False
551 frmProdInput.txtBaseX.Enabled = False
552 frmProdInput.txtDayX.Enabled = False
553 frmProdInput.cmdFresh.Enabled = False
554 frmProdInput.txtRate.Enabled = False
555 End If
556 Unload Me
557 End Sub
558
559 Private Sub Form_Load()
560     val_prod_no.Caption =
561     frmProdInput.txtProduction.Text
562     val_species.Caption = frmProd.sSpecies
563     'cmdSubmit.Enabled = False
564 End Sub
565
566 Private Sub Text1_Change()
567     'cmdSubmit.Enabled = True
568 End Sub
569
570
571 'Code for frame Frozen
572 Option Explicit
573 Dim lab_array(24) As Label
574
575 Private Sub cmdExit_Click()
576 Unload Me
577 End Sub
578
579 Private Sub cmdSubmit_Click()
580 Dim tmp2 As Integer
581 Dim cmd As String
582 Dim fconnection As Connection
583 Dim rs As Recordset
584 'Dim piconnection As Connection
585 Dim tmp As Boolean
586 tmp = False
587 Set fconnection = New Connection
588 fconnection.Open
589 "Provider=Microsoft.Jet.OLEDB.4.0;Data
590 Source=ADMS.mdb;Persist Security
591 Info=False"
592 If Text1.Text <> "" Then
593 If frmProdInput.exist = False Then
594 cmd = "insert into Production values ("
595 cmd = cmd + "" +
596 frmProdInput.txtProduction.Text + ""
597 cmd = cmd + ", " + "Date()"
598 cmd = cmd + ", " +
599 frmProdInput.val_species.Caption + ""
600 cmd = cmd + ", " +
601 frmProdInput.cboVessel.Text + ""
602 cmd = cmd + ", " +
603 frmProdInput.cboSource.Text + ""
604 cmd = cmd + ", " +
605 frmProdInput.cboArea.Text + ""
606 cmd = cmd + ", " + frmProdInput.txtRate.Text
607 cmd = cmd + ", " +
608 frmProdInput.txtDayX.Text
609 cmd = cmd + ", " +
610 frmProdInput.txtActualX.Text
611 cmd = cmd + ", " +
612 frmProdInput.txtBaseX.Text
613 cmd = cmd + ")"

```

```

614 fconnection.Execute cmd
615 frmProdInput.exist = True
616 End If
617 cmd = "insert into Frozen values ("
618 cmd = cmd + "" + val_prod_no.Caption + ""
619 cmd = cmd + ", " + Label1.Caption + ""
620 cmd = cmd + ", "
621 cmd = cmd + Text1.Text
622 cmd = cmd + ")"
623 fconnection.Execute cmd
624 Set rs = New Recordset
625 cmd = "select * from Inventory where spieces
626 = ""
627 cmd = cmd + frmProd.sSpecies + "" and ""
628 cmd = cmd + "ItemNo = ""
629 cmd = cmd + lab_array(0).Caption + ""
630 rs.Open cmd, fconnection,
631 adOpenForwardOnly, adLockReadOnly
632 rs.MoveFirst
633 tmp2 = rs("Quantity")
634 rs.Close
635 Set rs = Nothing
636 Text25.Text = 0
637 Text25.Text = Text1.Text + tmp2
638 cmd = "update Inventory set Quantity ="
639 cmd = cmd + Text25.Text
640 cmd = cmd + " where Spieces = ""
641 cmd = cmd + frmProd.sSpecies + "" and ""
642 cmd = cmd + "ItemNo = ""
643 cmd = cmd + lab_array(0).Caption + ""
644 fconnection.Execute cmd
645 tmp = True
646 End If
647
648 If Text2.Text <> "" Then
649 If frmProdInput.exist = False Then
650 cmd = "insert into Production values ("
651 cmd = cmd + "" +
652 frmProdInput.txtProduction.Text + ""
653 cmd = cmd + ", " + "Date()"
654 cmd = cmd + ", " +
655 frmProdInput.val_species.Caption + ""
656 cmd = cmd + ", " +
657 frmProdInput.cboVessel.Text + ""
658 cmd = cmd + ", " +
659 frmProdInput.cboSource.Text + ""
660 cmd = cmd + ", " +
661 frmProdInput.cboArea.Text + ""
662 cmd = cmd + ", " + frmProdInput.txtRate.Text
663 cmd = cmd + ", " +
664 frmProdInput.txtDayX.Text
665 cmd = cmd + ", " +
666 frmProdInput.txtActualX.Text
667 cmd = cmd + ", " +
668 frmProdInput.txtBaseX.Text
669 cmd = cmd + ")"
670 fconnection.Execute cmd
671 frmProdInput.exist = True
672 End If
673 cmd = "insert into Frozen values ("
674 cmd = cmd + "" + val_prod_no.Caption + ""
675 cmd = cmd + ", " + Label2.Caption + ""
676 cmd = cmd + ", "
677 cmd = cmd + Text2.Text
678 cmd = cmd + ")"
679 fconnection.Execute cmd
680 Set rs = New Recordset
681 cmd = "select * from Inventory where spieces
682 = ""
683 cmd = cmd + frmProd.sSpecies + "" and ""
684 cmd = cmd + "ItemNo = ""
685 cmd = cmd + lab_array(1).Caption + ""
686 rs.Open cmd, fconnection,
687 adOpenForwardOnly, adLockReadOnly
688 rs.MoveFirst
689 tmp2 = rs("Quantity")
690 rs.Close
691 Set rs = Nothing
692 Text25.Text = 0
693 Text25.Text = Text2.Text + tmp2
694 cmd = "update Inventory set Quantity ="
695 cmd = cmd + Text25.Text
696 cmd = cmd + " where Spieces = ""
697 cmd = cmd + frmProd.sSpecies + "" and ""
698 cmd = cmd + "ItemNo = ""
699 cmd = cmd + lab_array(1).Caption + ""
700 fconnection.Execute cmd
701 tmp = True

```

```

702 End If
703
704 If Text3.Text <> "" Then
705 If frmProdInput.exist = False Then
706 cmd = "insert into Production values ("
707 cmd = cmd + "" +
708 frmProdInput.txtProduction.Text + ""
709 cmd = cmd + ", " + "Date()"
710 cmd = cmd + ", " +
711 frmProdInput.val_species.Caption + ""
712 cmd = cmd + ", " +
713 frmProdInput.cboVessel.Text + ""
714 cmd = cmd + ", " +
715 frmProdInput.cboSource.Text + ""
716 cmd = cmd + ", " +
717 frmProdInput.cboArea.Text + ""
718 cmd = cmd + ", " + frmProdInput.txtRate.Text
719 cmd = cmd + ", " +
720 frmProdInput.txtDayX.Text
721 cmd = cmd + ", " +
722 frmProdInput.txtActualX.Text
723 cmd = cmd + ", " +
724 frmProdInput.txtBaseX.Text
725 cmd = cmd + ")"
726 fconnection.Execute cmd
727 frmProdInput.exist = True
728 End If
729 cmd = "insert into Frozen values ("
730 cmd = cmd + "" + val_prod_no.Caption + ""
731 cmd = cmd + ", " + Label3.Caption + ""
732 cmd = cmd + ", "
733 cmd = cmd + Text3.Text
734 cmd = cmd + ")"
735 fconnection.Execute cmd
736 Set rs = New Recordset
737 cmd = "select * from Inventory where spieces
738 = "
739 cmd = cmd + frmProd.sSpecies + " and "
740 cmd = cmd + "ItemNo = "
741 cmd = cmd + lab_array(2).Caption + ""
742 rs.Open cmd, fconnection,
743 adOpenForwardOnly, adLockReadOnly
744 rs.MoveFirst
745 tmp2 = rs("Quantity")
746 rs.Close
747 Set rs = Nothing
748 Text25.Text = 0
749 Text25.Text = Text3.Text + tmp2
750 cmd = "update Inventory set Quantity="
751 cmd = cmd + Text25.Text
752 cmd = cmd + " where Spieces = "
753 cmd = cmd + frmProd.sSpecies + " and "
754 cmd = cmd + "ItemNo = "
755 cmd = cmd + lab_array(2).Caption + ""
756 fconnection.Execute cmd
757 tmp = True
758 End If
759
760 If Text4.Text <> "" Then
761 If frmProdInput.exist = False Then
762 cmd = "insert into Production values ("
763 cmd = cmd + "" +
764 frmProdInput.txtProduction.Text + ""
765 cmd = cmd + ", " + "Date()"
766 cmd = cmd + ", " +
767 frmProdInput.val_species.Caption + ""
768 cmd = cmd + ", " +
769 frmProdInput.cboVessel.Text + ""
770 cmd = cmd + ", " +
771 frmProdInput.cboSource.Text + ""
772 cmd = cmd + ", " +
773 frmProdInput.cboArea.Text + ""
774 cmd = cmd + ", " + frmProdInput.txtRate.Text
775 cmd = cmd + ", " +
776 frmProdInput.txtDayX.Text
777 cmd = cmd + ", " +
778 frmProdInput.txtActualX.Text
779 cmd = cmd + ", " +
780 frmProdInput.txtBaseX.Text
781 cmd = cmd + ")"
782 fconnection.Execute cmd
783 frmProdInput.exist = True
784 End If
785 cmd = "insert into Frozen values ("
786 cmd = cmd + "" + val_prod_no.Caption + ""
787 cmd = cmd + ", " + Label4.Caption + ""
788 cmd = cmd + ", "
789 cmd = cmd + Text4.Text

```

```

790 cmd = cmd + ")"
791 fconnection.Execute cmd
792 Set rs = New Recordset
793 cmd = "select * from Inventory where spieces
794 = ""
795 cmd = cmd + frmProd.sSpecies + "" and ""
796 cmd = cmd + "ItemNo = ""
797 cmd = cmd + lab_array(3).Caption + ""
798 rs.Open cmd, fconnection,
799 adOpenForwardOnly, adLockReadOnly
800 rs.MoveFirst
801 tmp2 = rs("Quantity")
802 rs.Close
803 Set rs = Nothing
804 Text25.Text = 0
805 Text25.Text = Text4.Text + tmp2
806 cmd = "update Inventory set Quantity ="
807 cmd = cmd + Text25.Text
808 cmd = cmd + " where Spieces = ""
809 cmd = cmd + frmProd.sSpecies + "" and ""
810 cmd = cmd + "ItemNo = ""
811 cmd = cmd + lab_array(3).Caption + ""
812 fconnection.Execute cmd
813 tmp = True
814 End If
815
816 If Text5.Text <> "" Then
817 If frmProdInput.exist = False Then
818 cmd = "insert into Production values ("
819 cmd = cmd + "" +
820 frmProdInput.txtProduction.Text + ""
821 cmd = cmd + ", " + "Date()"
822 cmd = cmd + ", " +
823 frmProdInput.val_species.Caption + ""
824 cmd = cmd + ", " +
825 frmProdInput.cboVessel.Text + ""
826 cmd = cmd + ", " +
827 frmProdInput.cboSource.Text + ""
828 cmd = cmd + ", " +
829 frmProdInput.cboArea.Text + ""
830 cmd = cmd + ", " + frmProdInput.txtRate.Text
831 cmd = cmd + ", " +
832 frmProdInput.txtDayX.Text

```

```

833 cmd = cmd + ", " +
834 frmProdInput.txtActualX.Text
835 cmd = cmd + ", " +
836 frmProdInput.txtBaseX.Text
837 cmd = cmd + ")"
838 fconnection.Execute cmd
839 frmProdInput.exist = True
840 End If
841 cmd = "insert into Frozen values ("
842 cmd = cmd + "" + val_prod_no.Caption + ""
843 cmd = cmd + ", " + Label5.Caption + ""
844 cmd = cmd + ", " +
845 cmd = cmd + Text5.Text
846 cmd = cmd + ")"
847 fconnection.Execute cmd
848 Set rs = New Recordset
849 cmd = "select * from Inventory where spieces
850 = ""
851 cmd = cmd + frmProd.sSpecies + "" and ""
852 cmd = cmd + "ItemNo = ""
853 cmd = cmd + lab_array(4).Caption + ""
854 rs.Open cmd, fconnection,
855 adOpenForwardOnly, adLockReadOnly
856 rs.MoveFirst
857 tmp2 = rs("Quantity")
858 rs.Close
859 Set rs = Nothing
860 Text25.Text = 0
861 Text25.Text = Text5.Text + tmp2
862 cmd = "update Inventory set Quantity ="
863 cmd = cmd + Text25.Text
864 cmd = cmd + " where Spieces = ""
865 cmd = cmd + frmProd.sSpecies + "" and ""
866 cmd = cmd + "ItemNo = ""
867 cmd = cmd + lab_array(4).Caption + ""
868 fconnection.Execute cmd
869 tmp = True
870 End If
871
872 If Text6.Text <> "" Then
873 If frmProdInput.exist = False Then
874 cmd = "insert into Production values ("
875 cmd = cmd + "" +
876 frmProdInput.txtProduction.Text + ""

```

```

877 cmd = cmd + ", " + "Date()"
878 cmd = cmd + ", "" +
879 frmProdInput.val_species.Caption + ""
880 cmd = cmd + ", "" +
881 frmProdInput.cboVessel.Text + ""
882 cmd = cmd + ", "" +
883 frmProdInput.cboSource.Text + ""
884 cmd = cmd + ", "" +
885 frmProdInput.cboArea.Text + ""
886 cmd = cmd + ", " + frmProdInput.txtRate.Text
887 cmd = cmd + ", " +
888 frmProdInput.txtDayX.Text
889 cmd = cmd + ", " +
890 frmProdInput.txtActualX.Text
891 cmd = cmd + ", " +
892 frmProdInput.txtBaseX.Text
893 cmd = cmd + ")"
894 fconnection.Execute cmd
895 frmProdInput.exist = True
896 End If
897 cmd = "insert into Frozen values ("
898 cmd = cmd + "" + val_prod_no.Caption + ""
899 cmd = cmd + ", "" + Label1.Caption + ""
900 cmd = cmd + ", "
901 cmd = cmd + Text6.Text
902 cmd = cmd + ")"
903 fconnection.Execute cmd
904 Set rs = New Recordset
905 cmd = "select * from Inventory where spieces
906 = ""
907 cmd = cmd + frmProd.sSpecies + "" and ""
908 cmd = cmd + "ItemNo = ""
909 cmd = cmd + lab_array(5).Caption + ""
910 rs.Open cmd, fconnection,
911 adOpenForwardOnly, adLockReadOnly
912 rs.MoveFirst
913 tmp2 = rs("Quantity")
914 rs.Close
915 Set rs = Nothing
916 Text25.Text = 0
917 Text25.Text = Text6.Text + tmp2
918 cmd = "update Inventory set Quantity ="
919 cmd = cmd + Text25.Text
920 cmd = cmd + " where Spieces = ""

921 cmd = cmd + frmProd.sSpecies + "" and ""
922 cmd = cmd + "ItemNo = ""
923 cmd = cmd + lab_array(5).Caption + ""
924 fconnection.Execute cmd
925 tmp = True
926 End If
927
928 If Text7.Text <> "" Then
929 If frmProdInput.exist = False Then
930 cmd = "insert into Production values ("
931 cmd = cmd + "" +
932 frmProdInput.txtProduction.Text + ""
933 cmd = cmd + ", " + "Date()"
934 cmd = cmd + ", "" +
935 frmProdInput.val_species.Caption + ""
936 cmd = cmd + ", "" +
937 frmProdInput.cboVessel.Text + ""
938 cmd = cmd + ", "" +
939 frmProdInput.cboSource.Text + ""
940 cmd = cmd + ", "" +
941 frmProdInput.cboArea.Text + ""
942 cmd = cmd + ", " + frmProdInput.txtRate.Text
943 cmd = cmd + ", " +
944 frmProdInput.txtDayX.Text
945 cmd = cmd + ", " +
946 frmProdInput.txtActualX.Text
947 cmd = cmd + ", " +
948 frmProdInput.txtBaseX.Text
949 cmd = cmd + ")"
950 fconnection.Execute cmd
951 frmProdInput.exist = True
952 End If
953 cmd = "insert into Frozen values ("
954 cmd = cmd + "" + val_prod_no.Caption + ""
955 cmd = cmd + ", "" + Label7.Caption + ""
956 cmd = cmd + ", "
957 cmd = cmd + Text7.Text
958 cmd = cmd + ")"
959 fconnection.Execute cmd
960 Set rs = New Recordset
961 cmd = "select * from Inventory where spieces
962 = ""
963 cmd = cmd + frmProd.sSpecies + "" and ""
964 cmd = cmd + "ItemNo = ""

```

```

965 cmd = cmd + lab_array(6).Caption + ""
966 rs.Open cmd, fconnection,
967 adOpenForwardOnly, adLockReadOnly
968 rs.MoveFirst
969 tmp2 = rs("Quantity")
970 rs.Close
971 Set rs = Nothing
972 Text25.Text = 0
973 Text25.Text = Text7.Text + tmp2
974 cmd = "update Inventory set Quantity ="
975 cmd = cmd + Text25.Text
976 cmd = cmd + " where Spieces = "
977 cmd = cmd + frmProd.sSpecies + " and "
978 cmd = cmd + "ItemNo = "
979 cmd = cmd + lab_array(6).Caption + ""
980 fconnection.Execute cmd
981 tmp = True
982 End If
983
984 If Text8.Text <> "" Then
985 If frmProdInput.exist = False Then
986 cmd = "insert into Production values ("
987 cmd = cmd + "" +
988 frmProdInput.txtProduction.Text + ""
989 cmd = cmd + ", " + "Date()"
990 cmd = cmd + ", " +
991 frmProdInput.val_species.Caption + ""
992 cmd = cmd + ", " +
993 frmProdInput.cboVessel.Text + ""
994 cmd = cmd + ", " +
995 frmProdInput.cboSource.Text + ""
996 cmd = cmd + ", " +
997 frmProdInput.cboArea.Text + ""
998 cmd = cmd + ", " + frmProdInput.txtRate.Text
999 cmd = cmd + ", " +
1000 frmProdInput.txtDayX.Text
1001 cmd = cmd + ", " +
1002 frmProdInput.txtActualX.Text
1003 cmd = cmd + ", " +
1004 frmProdInput.txtBaseX.Text
1005 cmd = cmd + ")")
1006 fconnection.Execute cmd
1007 frmProdInput.exist = True
1008 End If

```

```

1009 cmd = "insert into Frozen values ("
1010 cmd = cmd + "" + val_prod_no.Caption + ""
1011 cmd = cmd + ", " + Label8.Caption + ""
1012 cmd = cmd + ", "
1013 cmd = cmd + Text8.Text
1014 cmd = cmd + ")")
1015 fconnection.Execute cmd
1016 Set rs = New Recordset
1017 cmd = "select * from Inventory where spieces = "
1018 cmd = cmd + frmProd.sSpecies + " and "
1019 cmd = cmd + "ItemNo = "
1020 cmd = cmd + lab_array(7).Caption + ""
1021 rs.Open cmd, fconnection, adOpenForwardOnly,
1022 adLockReadOnly
1023 rs.MoveFirst
1024 tmp2 = rs("Quantity")
1025 rs.Close
1026 Set rs = Nothing
1027 Text25.Text = 0
1028 Text25.Text = Text8.Text + tmp2
1029 cmd = "update Inventory set Quantity ="
1030 cmd = cmd + Text25.Text
1031 cmd = cmd + " where Spieces = "
1032 cmd = cmd + frmProd.sSpecies + " and "
1033 cmd = cmd + "ItemNo = "
1034 cmd = cmd + lab_array(7).Caption + ""
1035 fconnection.Execute cmd
1036 tmp = True
1037 End If
1038
1039 If Text9.Text <> "" Then
1040 If frmProdInput.exist = False Then
1041 cmd = "insert into Production values ("
1042 cmd = cmd + "" + frmProdInput.txtProduction.Text +
1043 ""
1044 cmd = cmd + ", " + "Date()"
1045 cmd = cmd + ", " +
1046 frmProdInput.val_species.Caption + ""
1047 cmd = cmd + ", " + frmProdInput.cboVessel.Text +
1048 ""
1049 cmd = cmd + ", " + frmProdInput.cboSource.Text +
1050 ""
1051 cmd = cmd + ", " + frmProdInput.cboArea.Text + ""
1052 cmd = cmd + ", " + frmProdInput.txtRate.Text

```

```

1053 cmd = cmd + ", " + frmProdInput.txtDayX.Text
1054 cmd = cmd + ", " + frmProdInput.txtActualX.Text
1055 cmd = cmd + ", " + frmProdInput.txtBaseX.Text
1056 cmd = cmd + ")"
1057 fconnection.Execute cmd
1058 frmProdInput.exist = True
1059 End If
1060 cmd = "insert into Frozen values ("
1061 cmd = cmd + "" + val_prod_no.Caption + ""
1062 cmd = cmd + ", " + Label9.Caption + ""
1063 cmd = cmd + ", "
1064 cmd = cmd + Text9.Text
1065 cmd = cmd + ")"
1066 fconnection.Execute cmd
1067 Set rs = New Recordset
1068 cmd = "select * from Inventory where spieces
1069 = ""
1070 cmd = cmd + frmProd.sSpecies + "" and ""
1071 cmd = cmd + "ItemNo = ""
1072 cmd = cmd + lab_array(8).Caption + ""
1073 rs.Open cmd, fconnection,
1074 adOpenForwardOnly, adLockReadOnly
1075 rs.MoveFirst
1076 tmp2 = rs("Quantity")
1077 rs.Close
1078 Set rs = Nothing
1079 Text25.Text = 0
1080 Text25.Text = Text9.Text + tmp2
1081 cmd = "update Inventory set Quantity ="
1082 cmd = cmd + Text25.Text
1083 cmd = cmd + " where Spieces = ""
1084 cmd = cmd + frmProd.sSpecies + "" and ""
1085 cmd = cmd + "ItemNo = ""
1086 cmd = cmd + lab_array(8).Caption + ""
1087 fconnection.Execute cmd
1088 tmp = True
1089 End If
1090
1091 If Text10.Text <> "" Then
1092 If frmProdInput.exist = False Then
1093 cmd = "insert into Production values ("
1094 cmd = cmd + "" +
1095 frmProdInput.txtProduction.Text + ""
1096 cmd = cmd + ", " + "Date()"
1097 cmd = cmd + ", "" +
1098 frmProdInput.val_species.Caption + ""
1099 cmd = cmd + ", "" +
1100 frmProdInput.cboVessel.Text + ""
1101 cmd = cmd + ", "" +
1102 frmProdInput.cboSource.Text + ""
1103 cmd = cmd + ", "" +
1104 frmProdInput.cboArea.Text + ""
1105 cmd = cmd + ", " + frmProdInput.txtRate.Text
1106 cmd = cmd + ", " +
1107 frmProdInput.txtDayX.Text
1108 cmd = cmd + ", " +
1109 frmProdInput.txtActualX.Text
1110 cmd = cmd + ", " +
1111 frmProdInput.txtBaseX.Text
1112 cmd = cmd + ")"
1113 fconnection.Execute cmd
1114 frmProdInput.exist = True
1115 End If
1116 cmd = "insert into Frozen values ("
1117 cmd = cmd + "" + val_prod_no.Caption + ""
1118 cmd = cmd + ", " + Label10.Caption + ""
1119 cmd = cmd + ", "
1120 cmd = cmd + Text10.Text
1121 cmd = cmd + ")"
1122 fconnection.Execute cmd
1123 Set rs = New Recordset
1124 cmd = "select * from Inventory where spieces
1125 = ""
1126 cmd = cmd + frmProd.sSpecies + "" and ""
1127 cmd = cmd + "ItemNo = ""
1128 cmd = cmd + lab_array(9).Caption + ""
1129 rs.Open cmd, fconnection,
1130 adOpenForwardOnly, adLockReadOnly
1131 rs.MoveFirst
1132 tmp2 = rs("Quantity")
1133 rs.Close
1134 Set rs = Nothing
1135 Text25.Text = 0
1136 Text25.Text = Text10.Text + tmp2
1137 cmd = "update Inventory set Quantity ="
1138 cmd = cmd + Text25.Text
1139 cmd = cmd + " where Spieces = ""
1140 cmd = cmd + frmProd.sSpecies + "" and ""

```

```

1141 cmd = cmd + "ItemNo = "
1142 cmd = cmd + lab_array(9).Caption + ""
1143 fconnection.Execute cmd
1144 tmp = True
1145 End If
1146
1147 If Text11.Text <> "" Then
1148 If frmProdInput.exist = False Then
1149 cmd = "insert into Production values ("
1150 cmd = cmd + "" +
1151 frmProdInput.txtProduction.Text + ""
1152 cmd = cmd + ", " + "Date()"
1153 cmd = cmd + ", " +
1154 frmProdInput.val_species.Caption + ""
1155 cmd = cmd + ", " +
1156 frmProdInput.cboVessel.Text + ""
1157 cmd = cmd + ", " +
1158 frmProdInput.cboSource.Text + ""
1159 cmd = cmd + ", " +
1160 frmProdInput.cboArea.Text + ""
1161 cmd = cmd + ", " + frmProdInput.txtRate.Text
1162 cmd = cmd + ", " +
1163 frmProdInput.txtDayX.Text
1164 cmd = cmd + ", " +
1165 frmProdInput.txtActualX.Text
1166 cmd = cmd + ", " +
1167 frmProdInput.txtBaseX.Text
1168 cmd = cmd + ")
1169 fconnection.Execute cmd
1170 frmProdInput.exist = True
1171 End If
1172 cmd = "insert into Frozen values ("
1173 cmd = cmd + "" + val_prod_no.Caption + ""
1174 cmd = cmd + ", " + Label11.Caption + ""
1175 cmd = cmd + ", "
1176 cmd = cmd + Text11.Text
1177 cmd = cmd + ")
1178 fconnection.Execute cmd
1179 Set rs = New Recordset
1180 cmd = "select * from Inventory where spieces
1181 = "
1182 cmd = cmd + frmProd.sSpecies + " and "
1183 cmd = cmd + "ItemNo = "
1184 cmd = cmd + lab_array(10).Caption + ""

1185 rs.Open cmd, fconnection,
1186 adOpenForwardOnly, adLockReadOnly
1187 rs.MoveFirst
1188 tmp2 = rs("Quantity")
1189 rs.Close
1190 Set rs = Nothing
1191 Text25.Text = 0
1192 Text25.Text = Text11.Text + tmp2
1193 cmd = "update Inventory set Quantity = "
1194 cmd = cmd + Text25.Text
1195 cmd = cmd + " where Spieces = "
1196 cmd = cmd + frmProd.sSpecies + " and "
1197 cmd = cmd + "ItemNo = "
1198 cmd = cmd + lab_array(10).Caption + ""
1199 fconnection.Execute cmd
1200 tmp = True
1201 End If
1202
1203 If Text12.Text <> "" Then
1204 If frmProdInput.exist = False Then
1205 cmd = "insert into Production values ("
1206 cmd = cmd + "" +
1207 frmProdInput.txtProduction.Text + ""
1208 cmd = cmd + ", " + "Date()"
1209 cmd = cmd + ", " +
1210 frmProdInput.val_species.Caption + ""
1211 cmd = cmd + ", " +
1212 frmProdInput.cboVessel.Text + ""
1213 cmd = cmd + ", " +
1214 frmProdInput.cboSource.Text + ""
1215 cmd = cmd + ", " +
1216 frmProdInput.cboArea.Text + ""
1217 cmd = cmd + ", " + frmProdInput.txtRate.Text
1218 cmd = cmd + ", " +
1219 frmProdInput.txtDayX.Text
1220 cmd = cmd + ", " +
1221 frmProdInput.txtActualX.Text
1222 cmd = cmd + ", " +
1223 frmProdInput.txtBaseX.Text
1224 cmd = cmd + ")
1225 fconnection.Execute cmd
1226 frmProdInput.exist = True
1227 End If
1228 cmd = "insert into Frozen values ("

```

```

1229 cmd = cmd + "" + val_prod_no.Caption + ""
1230 cmd = cmd + ", " + Label12.Caption + ""
1231 cmd = cmd + ", "
1232 cmd = cmd + Text12.Text
1233 cmd = cmd + ")"
1234 fconnection.Execute cmd
1235 Set rs = New Recordset
1236 cmd = "select * from Inventory where spieces
1237 = "
1238 cmd = cmd + frmProd.sSpecies + " and "
1239 cmd = cmd + "ItemNo = "
1240 cmd = cmd + lab_array(11).Caption + ""
1241 rs.Open cmd, fconnection,
1242 adOpenForwardOnly, adLockReadOnly
1243 rs.MoveFirst
1244 tmp2 = rs("Quantity")
1245 rs.Close
1246 Set rs = Nothing
1247 Text25.Text = 0
1248 Text25.Text = Text12.Text + tmp2
1249 cmd = "update Inventory set Quantity ="
1250 cmd = cmd + Text25.Text
1251 cmd = cmd + " where Spieces = "
1252 cmd = cmd + frmProd.sSpecies + " and "
1253 cmd = cmd + "ItemNo = "
1254 cmd = cmd + lab_array(11).Caption + ""
1255 fconnection.Execute cmd
1256 tmp = True
1257 End If
1258
1259 If Text13.Text <> "" Then
1260 If frmProdInput.exist = False Then
1261 cmd = "insert into Production values ("
1262 cmd = cmd + "" +
1263 frmProdInput.txtProduction.Text + ""
1264 cmd = cmd + ", " + "Date()"
1265 cmd = cmd + ", " +
1266 frmProdInput.val_species.Caption + ""
1267 cmd = cmd + ", " +
1268 frmProdInput.cboVessel.Text + ""
1269 cmd = cmd + ", " +
1270 frmProdInput.cboSource.Text + ""
1271 cmd = cmd + ", " +
1272 frmProdInput.cboArea.Text + ""
1273 cmd = cmd + ", " + frmProdInput.txtRate.Text
1274 cmd = cmd + ", " +
1275 frmProdInput.txtDayX.Text
1276 cmd = cmd + ", " +
1277 frmProdInput.txtActualX.Text
1278 cmd = cmd + ", " +
1279 frmProdInput.txtBaseX.Text
1280 cmd = cmd + ")"
1281 fconnection.Execute cmd
1282 frmProdInput.exist = True
1283 End If
1284 cmd = "insert into Frozen values ("
1285 cmd = cmd + "" + val_prod_no.Caption + ""
1286 cmd = cmd + ", " + Label13.Caption + ""
1287 cmd = cmd + ", "
1288 cmd = cmd + Text13.Text
1289 cmd = cmd + ")"
1290 fconnection.Execute cmd
1291 Set rs = New Recordset
1292 cmd = "select * from Inventory where spieces
1293 = "
1294 cmd = cmd + frmProd.sSpecies + " and "
1295 cmd = cmd + "ItemNo = "
1296 cmd = cmd + lab_array(12).Caption + ""
1297 rs.Open cmd, fconnection,
1298 adOpenForwardOnly, adLockReadOnly
1299 rs.MoveFirst
1300 tmp2 = rs("Quantity")
1301 rs.Close
1302 Set rs = Nothing
1303 Text25.Text = 0
1304 Text25.Text = Text13.Text + tmp2
1305 cmd = "update Inventory set Quantity ="
1306 cmd = cmd + Text25.Text
1307 cmd = cmd + " where Spieces = "
1308 cmd = cmd + frmProd.sSpecies + " and "
1309 cmd = cmd + "ItemNo = "
1310 cmd = cmd + lab_array(12).Caption + ""
1311 fconnection.Execute cmd
1312 tmp = True
1313 End If
1314
1315 If Text14.Text <> "" Then
1316 If frmProdInput.exist = False Then

```

```

1317 cmd = "insert into Production values ("
1318 cmd = cmd + "" +
1319 frmProdInput.txtProduction.Text + ""
1320 cmd = cmd + ", " + "Date()"
1321 cmd = cmd + ", " +
1322 frmProdInput.val_species.Caption + ""
1323 cmd = cmd + ", " +
1324 frmProdInput.cboVessel.Text + ""
1325 cmd = cmd + ", " +
1326 frmProdInput.cboSource.Text + ""
1327 cmd = cmd + ", " +
1328 frmProdInput.cboArea.Text + ""
1329 cmd = cmd + ", " + frmProdInput.txtRate.Text
1330 cmd = cmd + ", " +
1331 frmProdInput.txtDayX.Text
1332 cmd = cmd + ", " +
1333 frmProdInput.txtActualX.Text
1334 cmd = cmd + ", " +
1335 frmProdInput.txtBaseX.Text
1336 cmd = cmd + ")"
1337 fconnection.Execute cmd
1338 frmProdInput.exist = True
1339 End If
1340 cmd = "insert into Frozen values ("
1341 cmd = cmd + "" + val_prod_no.Caption + ""
1342 cmd = cmd + ", " + Label14.Caption + ""
1343 cmd = cmd + ", "
1344 cmd = cmd + Text14.Text
1345 cmd = cmd + ")"
1346 fconnection.Execute cmd
1347 Set rs = New Recordset
1348 cmd = "select * from Inventory where spieces
1349 ="
1350 cmd = cmd + frmProd.sSpecies + "" and ""
1351 cmd = cmd + "ItemNo ="
1352 cmd = cmd + lab_array(13).Caption + ""
1353 rs.Open cmd, fconnection,
1354 adOpenForwardOnly, adLockReadOnly
1355 rs.MoveFirst
1356 tmp2 = rs("Quantity")
1357 rs.Close
1358 Set rs = Nothing
1359 Text25.Text = 0
1360 Text25.Text = Text14.Text + tmp2

1361 cmd = "update Inventory set Quantity ="
1362 cmd = cmd + Text25.Text
1363 cmd = cmd + " where Spieces ="
1364 cmd = cmd + frmProd.sSpecies + "" and ""
1365 cmd = cmd + "ItemNo ="
1366 cmd = cmd + lab_array(13).Caption + ""
1367 fconnection.Execute cmd
1368 tmp = True
1369 End If
1370
1371 If Text15.Text <> "" Then
1372 If frmProdInput.exist = False Then
1373 cmd = "insert into Production values ("
1374 cmd = cmd + "" +
1375 frmProdInput.txtProduction.Text + ""
1376 cmd = cmd + ", " + "Date()"
1377 cmd = cmd + ", " +
1378 frmProdInput.val_species.Caption + ""
1379 cmd = cmd + ", " +
1380 frmProdInput.cboVessel.Text + ""
1381 cmd = cmd + ", " +
1382 frmProdInput.cboSource.Text + ""
1383 cmd = cmd + ", " +
1384 frmProdInput.cboArea.Text + ""
1385 cmd = cmd + ", " + frmProdInput.txtRate.Text
1386 cmd = cmd + ", " +
1387 frmProdInput.txtDayX.Text
1388 cmd = cmd + ", " +
1389 frmProdInput.txtActualX.Text
1390 cmd = cmd + ", " +
1391 frmProdInput.txtBaseX.Text
1392 cmd = cmd + ")"
1393 fconnection.Execute cmd
1394 frmProdInput.exist = True
1395 End If
1396 cmd = "insert into Frozen values ("
1397 cmd = cmd + "" + val_prod_no.Caption + ""
1398 cmd = cmd + ", " + Label15.Caption + ""
1399 cmd = cmd + ", "
1400 cmd = cmd + Text15.Text
1401 cmd = cmd + ")"
1402 fconnection.Execute cmd
1403 Set rs = New Recordset

```

```

1404 cmd = "select * from Inventory where spieces
1405 = ""
1406 cmd = cmd + frmProd.sSpecies + "" and ""
1407 cmd = cmd + "ItemNo = ""
1408 cmd = cmd + lab_array(14).Caption + ""
1409 rs.Open cmd, fconnection,
1410 adOpenForwardOnly, adLockReadOnly
1411 rs.MoveFirst
1412 tmp2 = rs("Quantity")
1413 rs.Close
1414 Set rs = Nothing
1415 Text25.Text = 0
1416 Text25.Text = Text15.Text + tmp2
1417 cmd = "update Inventory set Quantity ="
1418 cmd = cmd + Text25.Text
1419 cmd = cmd + " where Spieces = ""
1420 cmd = cmd + frmProd.sSpecies + "" and ""
1421 cmd = cmd + "ItemNo = ""
1422 cmd = cmd + lab_array(14).Caption + ""
1423 fconnection.Execute cmd
1424 tmp = True
1425 End If
1426
1427 If Text16.Text <> "" Then
1428 If frmProdInput.exist = False Then
1429 cmd = "insert into Production values ("
1430 cmd = cmd + "" +
1431 frmProdInput.txtProduction.Text + ""
1432 cmd = cmd + ", " + "Date()"
1433 cmd = cmd + ", "" +
1434 frmProdInput.val_species.Caption + ""
1435 cmd = cmd + ", "" +
1436 frmProdInput.cboVessel.Text + ""
1437 cmd = cmd + ", "" +
1438 frmProdInput.cboSource.Text + ""
1439 cmd = cmd + ", "" +
1440 frmProdInput.cboArea.Text + ""
1441 cmd = cmd + ", " + frmProdInput.txtRate.Text
1442 cmd = cmd + ", " +
1443 frmProdInput.txtDayX.Text
1444 cmd = cmd + ", " +
1445 frmProdInput.txtActualX.Text
1446 cmd = cmd + ", " +
1447 frmProdInput.txtBaseX.Text

```

```

1448 cmd = cmd + ")"
1449 fconnection.Execute cmd
1450 frmProdInput.exist = True
1451 End If
1452 cmd = "insert into Frozen values ("
1453 cmd = cmd + "" + val_prod_no.Caption + ""
1454 cmd = cmd + ", " + Label16.Caption + ""
1455 cmd = cmd + ", "
1456 cmd = cmd + Text16.Text
1457 cmd = cmd + ")"
1458 fconnection.Execute cmd
1459 Set rs = New Recordset
1460 cmd = "select * from Inventory where spieces
1461 = ""
1462 cmd = cmd + frmProd.sSpecies + "" and ""
1463 cmd = cmd + "ItemNo = ""
1464 cmd = cmd + lab_array(15).Caption + ""
1465 rs.Open cmd, fconnection,
1466 adOpenForwardOnly, adLockReadOnly
1467 rs.MoveFirst
1468 tmp2 = rs("Quantity")
1469 rs.Close
1470 Set rs = Nothing
1471 Text25.Text = 0
1472 Text25.Text = Text16.Text + tmp2
1473 cmd = "update Inventory set Quantity ="
1474 cmd = cmd + Text25.Text
1475 cmd = cmd + " where Spieces = ""
1476 cmd = cmd + frmProd.sSpecies + "" and ""
1477 cmd = cmd + "ItemNo = ""
1478 cmd = cmd + lab_array(15).Caption + ""
1479 fconnection.Execute cmd
1480 tmp = True
1481 End If
1482
1483 If Text17.Text <> "" Then
1484 If frmProdInput.exist = False Then
1485 cmd = "insert into Production values ("
1486 cmd = cmd + "" +
1487 frmProdInput.txtProduction.Text + ""
1488 cmd = cmd + ", " + "Date()"
1489 cmd = cmd + ", "" +
1490 frmProdInput.val_species.Caption + ""

```

```

1491 cmd = cmd + ", " +
1492 frmProdInput.cboVessel.Text + ""
1493 cmd = cmd + ", " +
1494 frmProdInput.cboSource.Text + ""
1495 cmd = cmd + ", " +
1496 frmProdInput.cboArea.Text + ""
1497 cmd = cmd + ", " + frmProdInput.txtRate.Text
1498 cmd = cmd + ", " +
1499 frmProdInput.txtDayX.Text
1500 cmd = cmd + ", " +
1501 frmProdInput.txtActualX.Text
1502 cmd = cmd + ", " +
1503 frmProdInput.txtBaseX.Text
1504 cmd = cmd + ")"
1505 fconnection.Execute cmd
1506 frmProdInput.exist = True
1507 End If
1508 cmd = "insert into Frozen values ("
1509 cmd = cmd + "" + val_prod_no.Caption + ""
1510 cmd = cmd + ", " + Label17.Caption + ""
1511 cmd = cmd + ", "
1512 cmd = cmd + Text17.Text
1513 cmd = cmd + ")"
1514 fconnection.Execute cmd
1515 Set rs = New Recordset
1516 cmd = "select * from Inventory where spieces
1517 ="
1518 cmd = cmd + frmProd.sSpecies + " and "
1519 cmd = cmd + "ItemNo = "
1520 cmd = cmd + lab_array(16).Caption + ""
1521 rs.Open cmd, fconnection,
1522 adOpenForwardOnly, adLockReadOnly
1523 rs.MoveFirst
1524 tmp2 = rs("Quantity")
1525 rs.Close
1526 Set rs = Nothing
1527 Text25.Text = 0
1528 Text25.Text = Text17.Text + tmp2
1529 cmd = "update Inventory set Quantity ="
1530 cmd = cmd + Text25.Text
1531 cmd = cmd + " where Spieces = "
1532 cmd = cmd + frmProd.sSpecies + " and "
1533 cmd = cmd + "ItemNo = "
1534 cmd = cmd + lab_array(16).Caption + ""
1535 fconnection.Execute cmd
1536 tmp = True
1537 End If
1538
1539 If Text18.Text <> "" Then
1540 If frmProdInput.exist = False Then
1541 cmd = "insert into Production values ("
1542 cmd = cmd + "" +
1543 frmProdInput.txtProduction.Text + ""
1544 cmd = cmd + ", " + "Date()"
1545 cmd = cmd + ", " +
1546 frmProdInput.val_species.Caption + ""
1547 cmd = cmd + ", " +
1548 frmProdInput.cboVessel.Text + ""
1549 cmd = cmd + ", " +
1550 frmProdInput.cboSource.Text + ""
1551 cmd = cmd + ", " +
1552 frmProdInput.cboArea.Text + ""
1553 cmd = cmd + ", " + frmProdInput.txtRate.Text
1554 cmd = cmd + ", " +
1555 frmProdInput.txtDayX.Text
1556 cmd = cmd + ", " +
1557 frmProdInput.txtActualX.Text
1558 cmd = cmd + ", " +
1559 frmProdInput.txtBaseX.Text
1560 cmd = cmd + ")"
1561 fconnection.Execute cmd
1562 frmProdInput.exist = True
1563 End If
1564 cmd = "insert into Frozen values ("
1565 cmd = cmd + "" + val_prod_no.Caption + ""
1566 cmd = cmd + ", " + Label18.Caption + ""
1567 cmd = cmd + ", "
1568 cmd = cmd + Text18.Text
1569 cmd = cmd + ")"
1570 fconnection.Execute cmd
1571 Set rs = New Recordset
1572 cmd = "select * from Inventory where spieces
1573 ="
1574 cmd = cmd + frmProd.sSpecies + " and "
1575 cmd = cmd + "ItemNo = "
1576 cmd = cmd + lab_array(17).Caption + ""
1577 rs.Open cmd, fconnection,
1578 adOpenForwardOnly, adLockReadOnly

```

```

1579 rs.MoveFirst
1580 tmp2 = rs("Quantity")
1581 rs.Close
1582 Set rs = Nothing
1583 Text25.Text = 0
1584 Text25.Text = Text18.Text + tmp2
1585 cmd = "update Inventory set Quantity ="
1586 cmd = cmd + Text25.Text
1587 cmd = cmd + " where Spieces = "
1588 cmd = cmd + frmProd.sSpecies + " and "
1589 cmd = cmd + "ItemNo = "
1590 cmd = cmd + lab_array(17).Caption + ""
1591 fconnection.Execute cmd
1592 tmp = True
1593 End If
1594
1595 If Text19.Text <> "" Then
1596 If frmProdInput.exist = False Then
1597 cmd = "insert into Production values ("
1598 cmd = cmd + "" +
1599 frmProdInput.txtProduction.Text + ""
1600 cmd = cmd + ", " + "Date()"
1601 cmd = cmd + ", " +
1602 frmProdInput.val_species.Caption + ""
1603 cmd = cmd + ", " +
1604 frmProdInput.cboVessel.Text + ""
1605 cmd = cmd + ", " +
1606 frmProdInput.cboSource.Text + ""
1607 cmd = cmd + ", " +
1608 frmProdInput.cboArea.Text + ""
1609 cmd = cmd + ", " + frmProdInput.txtRate.Text
1610 cmd = cmd + ", " +
1611 frmProdInput.txtDayX.Text
1612 cmd = cmd + ", " +
1613 frmProdInput.txtActualX.Text
1614 cmd = cmd + ", " +
1615 frmProdInput.txtBaseX.Text
1616 cmd = cmd + ")")
1617 fconnection.Execute cmd
1618 frmProdInput.exist = True
1619 End If
1620 cmd = "insert into Frozen values ("
1621 cmd = cmd + "" + val_prod_no.Caption + ""
1622 cmd = cmd + ", " + Label19.Caption + ""
1623 cmd = cmd + ", "
1624 cmd = cmd + Text19.Text
1625 cmd = cmd + ")")
1626 fconnection.Execute cmd
1627 Set rs = New Recordset
1628 cmd = "select * from Inventory where spieces
1629 = "
1630 cmd = cmd + frmProd.sSpecies + " and "
1631 cmd = cmd + "ItemNo = "
1632 cmd = cmd + lab_array(18).Caption + ""
1633 rs.Open cmd, fconnection,
1634 adOpenForwardOnly, adLockReadOnly
1635 rs.MoveFirst
1636 tmp2 = rs("Quantity")
1637 rs.Close
1638 Set rs = Nothing
1639 Text25.Text = 0
1640 Text25.Text = Text19.Text + tmp2
1641 cmd = "update Inventory set Quantity ="
1642 cmd = cmd + Text25.Text
1643 cmd = cmd + " where Spieces = "
1644 cmd = cmd + frmProd.sSpecies + " and "
1645 cmd = cmd + "ItemNo = "
1646 cmd = cmd + lab_array(18).Caption + ""
1647 fconnection.Execute cmd
1648 tmp = True
1649 End If
1650
1651 If Text20.Text <> "" Then
1652 If frmProdInput.exist = False Then
1653 cmd = "insert into Production values ("
1654 cmd = cmd + "" +
1655 frmProdInput.txtProduction.Text + ""
1656 cmd = cmd + ", " + "Date()"
1657 cmd = cmd + ", " +
1658 frmProdInput.val_species.Caption + ""
1659 cmd = cmd + ", " +
1660 frmProdInput.cboVessel.Text + ""
1661 cmd = cmd + ", " +
1662 frmProdInput.cboSource.Text + ""
1663 cmd = cmd + ", " +
1664 frmProdInput.cboArea.Text + ""
1665 cmd = cmd + ", " + frmProdInput.txtRate.Text

```

```

1666 cmd = cmd + ", " +
1667 frmProdInput.txtDayX.Text
1668 cmd = cmd + ", " +
1669 frmProdInput.txtActualX.Text
1670 cmd = cmd + ", " +
1671 frmProdInput.txtBaseX.Text
1672 cmd = cmd + ")"
1673 fconnection.Execute cmd
1674 frmProdInput.exist = True
1675 End If
1676 cmd = "insert into Frozen values ("
1677 cmd = cmd + "" + val_prod_no.Caption + ""
1678 cmd = cmd + ", "" + Label20.Caption + ""
1679 cmd = cmd + ", "
1680 cmd = cmd + Text20.Text
1681 cmd = cmd + ")"
1682 fconnection.Execute cmd
1683 Set rs = New Recordset
1684 cmd = "select * from Inventory where spieces
1685 = ""
1686 cmd = cmd + frmProd.sSpecies + "" and ""
1687 cmd = cmd + "ItemNo = ""
1688 cmd = cmd + lab_array(19).Caption + ""
1689 rs.Open cmd, fconnection,
1690 adOpenForwardOnly, adLockReadOnly
1691 rs.MoveFirst
1692 tmp2 = rs("Quantity")
1693 rs.Close
1694 Set rs = Nothing
1695 Text25.Text = 0
1696 Text25.Text = Text20.Text + tmp2
1697 cmd = "update Inventory set Quantity ="
1698 cmd = cmd + Text25.Text
1699 cmd = cmd + " where Spieces = ""
1700 cmd = cmd + frmProd.sSpecies + "" and ""
1701 cmd = cmd + "ItemNo = ""
1702 cmd = cmd + lab_array(19).Caption + ""
1703 fconnection.Execute cmd
1704 tmp = True
1705 End If
1706
1707 If Text21.Text <> "" Then
1708 If frmProdInput.exist = False Then
1709 cmd = "insert into Production values ("
1710 cmd = cmd + "" +
1711 frmProdInput.txtProduction.Text + ""
1712 cmd = cmd + ", " + "Date()"
1713 cmd = cmd + ", "" +
1714 frmProdInput.val_species.Caption + ""
1715 cmd = cmd + ", "" +
1716 frmProdInput.cboVessel.Text + ""
1717 cmd = cmd + ", "" +
1718 frmProdInput.cboSource.Text + ""
1719 cmd = cmd + ", "" +
1720 frmProdInput.cboArea.Text + ""
1721 cmd = cmd + ", " + frmProdInput.txtRate.Text
1722 cmd = cmd + ", " +
1723 frmProdInput.txtDayX.Text
1724 cmd = cmd + ", " +
1725 frmProdInput.txtActualX.Text
1726 cmd = cmd + ", " +
1727 frmProdInput.txtBaseX.Text
1728 cmd = cmd + ")"
1729 fconnection.Execute cmd
1730 frmProdInput.exist = True
1731 End If
1732 cmd = "insert into Frozen values ("
1733 cmd = cmd + "" + val_prod_no.Caption + ""
1734 cmd = cmd + ", "" + Label21.Caption + ""
1735 cmd = cmd + ", "
1736 cmd = cmd + Text21.Text
1737 cmd = cmd + ")"
1738 fconnection.Execute cmd
1739 Set rs = New Recordset
1740 cmd = "select * from Inventory where spieces
1741 = ""
1742 cmd = cmd + frmProd.sSpecies + "" and ""
1743 cmd = cmd + "ItemNo = ""
1744 cmd = cmd + lab_array(20).Caption + ""
1745 rs.Open cmd, fconnection,
1746 adOpenForwardOnly, adLockReadOnly
1747 rs.MoveFirst
1748 tmp2 = rs("Quantity")
1749 rs.Close
1750 Set rs = Nothing
1751 Text25.Text = 0
1752 Text25.Text = Text21.Text + tmp2
1753 cmd = "update Inventory set Quantity ="

```

```

1754 cmd = cmd + Text25.Text
1755 cmd = cmd + " where Spieces = "
1756 cmd = cmd + frmProd.sSpecies + " and "
1757 cmd = cmd + "ItemNo = "
1758 cmd = cmd + lab_array(20).Caption + ""
1759 fconnection.Execute cmd
1760 tmp = True
1761 End If
1762
1763 If Text22.Text <> "" Then
1764 If frmProdInput.exist = False Then
1765 cmd = "insert into Production values ("
1766 cmd = cmd + "" +
1767 frmProdInput.txtProduction.Text + ""
1768 cmd = cmd + ", " + "Date()"
1769 cmd = cmd + ", " +
1770 frmProdInput.val_species.Caption + ""
1771 cmd = cmd + ", " +
1772 frmProdInput.cboVessel.Text + ""
1773 cmd = cmd + ", " +
1774 frmProdInput.cboSource.Text + ""
1775 cmd = cmd + ", " +
1776 frmProdInput.cboArea.Text + ""
1777 cmd = cmd + ", " + frmProdInput.txtRate.Text
1778 cmd = cmd + ", " +
1779 frmProdInput.txtDayX.Text
1780 cmd = cmd + ", " +
1781 frmProdInput.txtActualX.Text
1782 cmd = cmd + ", " +
1783 frmProdInput.txtBaseX.Text
1784 cmd = cmd + ")""
1785 fconnection.Execute cmd
1786 frmProdInput.exist = True
1787 End If
1788 cmd = "insert into Frozen values ("
1789 cmd = cmd + "" + val_prod_no.Caption + ""
1790 cmd = cmd + ", " + Label22.Caption + ""
1791 cmd = cmd + ", "
1792 cmd = cmd + Text22.Text
1793 cmd = cmd + ")""
1794 fconnection.Execute cmd
1795 Set rs = New Recordset
1796 cmd = "select * from Inventory where spieces
1797 = "

```

```

1798 cmd = cmd + frmProd.sSpecies + " and "
1799 cmd = cmd + "ItemNo = "
1800 cmd = cmd + lab_array(21).Caption + ""
1801 rs.Open cmd, fconnection,
1802 adOpenForwardOnly, adLockReadOnly
1803 rs.MoveFirst
1804 tmp2 = rs("Quantity")
1805 rs.Close
1806 Set rs = Nothing
1807 Text25.Text = 0
1808 Text25.Text = Text22.Text + tmp2
1809 cmd = "update Inventory set Quantity ="
1810 cmd = cmd + Text25.Text
1811 cmd = cmd + " where Spieces = "
1812 cmd = cmd + frmProd.sSpecies + " and "
1813 cmd = cmd + "ItemNo = "
1814 cmd = cmd + lab_array(21).Caption + ""
1815 fconnection.Execute cmd
1816 tmp = True
1817 End If
1818
1819 If Text23.Text <> "" Then
1820 If frmProdInput.exist = False Then
1821 cmd = "insert into Production values ("
1822 cmd = cmd + "" +
1823 frmProdInput.txtProduction.Text + ""
1824 cmd = cmd + ", " + "Date()"
1825 cmd = cmd + ", " +
1826 frmProdInput.val_species.Caption + ""
1827 cmd = cmd + ", " +
1828 frmProdInput.cboVessel.Text + ""
1829 cmd = cmd + ", " +
1830 frmProdInput.cboSource.Text + ""
1831 cmd = cmd + ", " +
1832 frmProdInput.cboArea.Text + ""
1833 cmd = cmd + ", " + frmProdInput.txtRate.Text
1834 cmd = cmd + ", " +
1835 frmProdInput.txtDayX.Text
1836 cmd = cmd + ", " +
1837 frmProdInput.txtActualX.Text
1838 cmd = cmd + ", " +
1839 frmProdInput.txtBaseX.Text
1840 cmd = cmd + ")""
1841 fconnection.Execute cmd

```

```

1842 frmProdInput.exist = True
1843 End If
1844 cmd = "insert into Frozen values ("
1845 cmd = cmd + """" + val_prod_no.Caption + """"
1846 cmd = cmd + ", " + Label23.Caption + """"
1847 cmd = cmd + ", "
1848 cmd = cmd + Text23.Text
1849 cmd = cmd + ")"
1850 fconnection.Execute cmd
1851 Set rs = New Recordset
1852 cmd = "select * from Inventory where spieces
1853 = ""
1854 cmd = cmd + frmProd.sSpecies + "" and ""
1855 cmd = cmd + "ItemNo = ""
1856 cmd = cmd + lab_array(22).Caption + """"
1857 rs.Open cmd, fconnection,
1858 adOpenForwardOnly, adLockReadOnly
1859 rs.MoveFirst
1860 tmp2 = rs("Quantity")
1861 rs.Close
1862 Set rs = Nothing
1863 Text25.Text = 0
1864 Text25.Text = Text23.Text + tmp2
1865 cmd = "update Inventory set Quantity = "
1866 cmd = cmd + Text25.Text
1867 cmd = cmd + " where Spieces = ""
1868 cmd = cmd + frmProd.sSpecies + "" and ""
1869 cmd = cmd + "ItemNo = ""
1870 cmd = cmd + lab_array(22).Caption + """"
1871 fconnection.Execute cmd
1872 tmp = True
1873 End If
1874
1875 If Text24.Text <> "" Then
1876 If frmProdInput.exist = False Then
1877 cmd = "insert into Production values ("
1878 cmd = cmd + """" +
1879 frmProdInput.txtProduction.Text + """"
1880 cmd = cmd + ", " + "Date()"
1881 cmd = cmd + ", " +
1882 frmProdInput.val_species.Caption + """"
1883 cmd = cmd + ", " +
1884 frmProdInput.cboVessel.Text + """"
1885 cmd = cmd + ", " +
1886 frmProdInput.cboSource.Text + """"
1887 cmd = cmd + ", " +
1888 frmProdInput.cboArea.Text + """"
1889 cmd = cmd + ", " + frmProdInput.txtRate.Text
1890 cmd = cmd + ", " +
1891 frmProdInput.txtDayX.Text
1892 cmd = cmd + ", " +
1893 frmProdInput.txtActualX.Text
1894 cmd = cmd + ", " +
1895 frmProdInput.txtBaseX.Text
1896 cmd = cmd + ")"
1897 fconnection.Execute cmd
1898 frmProdInput.exist = True
1899 End If
1900 cmd = "insert into Frozen values ("
1901 cmd = cmd + """" + val_prod_no.Caption + """"
1902 cmd = cmd + ", " + Label24.Caption + """"
1903 cmd = cmd + ", "
1904 cmd = cmd + Text24.Text
1905 cmd = cmd + ")"
1906 fconnection.Execute cmd
1907 Set rs = New Recordset
1908 cmd = "select * from Inventory where spieces
1909 = ""
1910 cmd = cmd + frmProd.sSpecies + "" and ""
1911 cmd = cmd + "ItemNo = ""
1912 cmd = cmd + lab_array(23).Caption + """"
1913 rs.Open cmd, fconnection,
1914 adOpenForwardOnly, adLockReadOnly
1915 rs.MoveFirst
1916 tmp2 = rs("Quantity")
1917 rs.Close
1918 Set rs = Nothing
1919 Text25.Text = 0
1920 Text25.Text = Text24.Text + tmp2
1921 cmd = "update Inventory set Quantity = "
1922 cmd = cmd + Text25.Text
1923 cmd = cmd + " where Spieces = ""
1924 cmd = cmd + frmProd.sSpecies + "" and ""
1925 cmd = cmd + "ItemNo = ""
1926 cmd = cmd + lab_array(23).Caption + """"
1927 fconnection.Execute cmd
1928 tmp = True

```

```

1929 End If
1930
1931 If tmp = True Then
1932 frmProdInput.exists = True
1933 frmProdInput.txtProduction.Enabled = False
1934 frmProdInput.cboArea.Enabled = False
1935 frmProdInput.cboSource.Enabled = False
1936 frmProdInput.cboVessel.Enabled = False
1937 frmProdInput.txtActualX.Enabled = False
1938 frmProdInput.txtBaseX.Enabled = False
1939 frmProdInput.txtDayX.Enabled = False
1940 frmProdInput.cmdFrozen.Enabled = False
1941 frmProdInput.txtRate.Enabled = False
1942 End If
1943 Unload Me
1944 End Sub
1945
1946 Private Sub Form_Load()
1947 'Dim lab_array(24) As Label
1948 Dim cmd As String
1949 Dim fconnection As Connection
1950 Dim rs As Recordset
1951 Dim tmp As Boolean
1952 Dim i As Integer
1953 tmp = False
1954 Set fconnection = New Connection
1955 fconnection.Open
1956 "Provider=Microsoft.Jet.OLEDB.4.0;Data
1957 Source=ADMS.mdb;Persist Security
1958 Info=False"
1959 cmd = "select * from Inventory where spieces
1960 = ""
1961 cmd = cmd + frmProd.sSpecies + ""
1962 Set rs = New Recordset
1963 rs.Open cmd, fconnection,
1964 adOpenForwardOnly, adLockReadOnly
1965 rs.MoveFirst
1966
1967 val_prod_no.Caption =
1968 frmProdInput.txtProduction.Text
1969 val_species.Caption = frmProd.sSpecies
1970 cmdSubmit.Enabled = False
1971 'Label1.Caption = rs("ItemNo")
1972 'rs.MoveNext
1973 'Label2.Caption = rs("ItemNo")
1974 'rs.MoveNext
1975 'Label3.Caption = rs("ItemNo")
1976 'rs.MoveNext
1977 'Label4.Caption = rs("ItemNo")
1978 'rs.MoveNext
1979 'Label5.Caption = rs("ItemNo")
1980 'rs.MoveNext
1981 'Label6.Caption = rs("ItemNo")
1982 'rs.MoveNext
1983 'Label7.Caption = rs("ItemNo")
1984 'rs.MoveNext
1985 'Label8.Caption = rs("ItemNo")
1986 'rs.MoveNext
1987 'Label9.Caption = rs("ItemNo")
1988 'rs.MoveNext
1989 'Label10.Caption = rs("ItemNo")
1990 'rs.MoveNext
1991 'Label11.Caption = rs("ItemNo")
1992 'rs.MoveNext
1993 'Label12.Caption = rs("ItemNo")
1994 'rs.MoveNext
1995 'Label13.Caption = rs("ItemNo")
1996 'rs.MoveNext
1997 'Label14.Caption = rs("ItemNo")
1998 'rs.MoveNext
1999 'Label15.Caption = rs("ItemNo")
2000 'rs.MoveNext
2001 'Label16.Caption = rs("ItemNo")
2002 'rs.MoveNext
2003 'Label17.Caption = rs("ItemNo")
2004 'rs.MoveNext
2005 'Label18.Caption = rs("ItemNo")
2006 'rs.MoveNext
2007 'Label19.Caption = rs("ItemNo")
2008 'rs.MoveNext
2009 'Label20.Caption = rs("ItemNo")
2010 'rs.MoveNext
2011 'Label21.Caption = rs("ItemNo")
2012 'rs.MoveNext
2013 'Label22.Caption = rs("ItemNo")
2014 'rs.MoveNext
2015 'Label23.Caption = rs("ItemNo")
2016 'rs.MoveNext

```

```

2017 'Label24.Caption = rs("ItemNo")
2018 'rs.MoveNext
2019
2020 Set lab_array(0) = Label1
2021 Set lab_array(1) = Label2
2022 Set lab_array(2) = Label3
2023 Set lab_array(3) = Label4
2024 Set lab_array(4) = Label5
2025 Set lab_array(5) = Label6
2026 Set lab_array(6) = Label7
2027 Set lab_array(7) = Label8
2028 Set lab_array(8) = Label9
2029 Set lab_array(9) = Label10
2030 Set lab_array(10) = Label11
2031 Set lab_array(11) = Label12
2032 Set lab_array(12) = Label13
2033 Set lab_array(13) = Label14
2034 Set lab_array(14) = Label15
2035 Set lab_array(15) = Label16
2036 Set lab_array(16) = Label17
2037 Set lab_array(17) = Label18
2038 Set lab_array(18) = Label19
2039 Set lab_array(19) = Label20
2040 Set lab_array(20) = Label21
2041 Set lab_array(21) = Label22
2042 Set lab_array(22) = Label23
2043 Set lab_array(23) = Label24
2044
2045 'Do Until rs.EOF
2046 'frmProdInput.cboVessel.AddItem
2047 (rs("Vessel_Name"))
2048 'rs.MoveNext
2049 'Loop
2050 For i = 0 To 23
2051 lab_array(i).Caption = rs("ItemNo")
2052 rs.MoveNext
2053 Next i
2054 rs.Close
2055 fconnection.Close
2056 Set rs = Nothing
2057 Set fconnection = Nothing
2058 End Sub
2059
2060 Private Sub Text1_Change()
2061 cmdSubmit.Enabled = True
2062 End Sub
2063
2064 Private Sub Text2_Change()
2065 cmdSubmit.Enabled = True
2066 End Sub
2067
2068 Private Sub Text3_Change()
2069 cmdSubmit.Enabled = True
2070 End Sub
2071
2072 Private Sub Text4_Change()
2073 cmdSubmit.Enabled = True
2074 End Sub
2075
2076 Private Sub Text5_Change()
2077 cmdSubmit.Enabled = True
2078 End Sub
2079
2080 Private Sub Text6_Change()
2081 cmdSubmit.Enabled = True
2082 End Sub
2083
2084 Private Sub Text7_Change()
2085 cmdSubmit.Enabled = True
2086 End Sub
2087
2088 Private Sub Text8_Change()
2089 cmdSubmit.Enabled = True
2090 End Sub
2091
2092 Private Sub Text9_Change()
2093 cmdSubmit.Enabled = True
2094 End Sub
2095
2096 Private Sub Text10_Change()
2097 cmdSubmit.Enabled = True
2098 End Sub
2099
2100 Private Sub Text11_Change()
2101 cmdSubmit.Enabled = True
2102 End Sub
2103
2104 Private Sub Text12_Change()

```

```

2105 cmdSubmit.Enabled = True
2106 End Sub
2107
2108 Private Sub Text13_Change()
2109 cmdSubmit.Enabled = True
2110 End Sub
2111
2112 Private Sub Text14_Change()
2113 cmdSubmit.Enabled = True
2114 End Sub
2115
2116 Private Sub Text15_Change()
2117 cmdSubmit.Enabled = True
2118 End Sub
2119
2120 Private Sub Text16_Change()
2121 cmdSubmit.Enabled = True
2122 End Sub
2123
2124 Private Sub Text17_Change()
2125 cmdSubmit.Enabled = True
2126 End Sub
2127
2128 Private Sub Text18_Change()
2129 cmdSubmit.Enabled = True
2130 End Sub
2131
2132 Private Sub Text19_Change()
2133 cmdSubmit.Enabled = True
2134 End Sub
2135
2136 Private Sub Text20_Change()
2137 cmdSubmit.Enabled = True
2138 End Sub
2139
2140 Private Sub Text21_Change()
2141 cmdSubmit.Enabled = True
2142 End Sub
2143
2144 Private Sub Text22_Change()
2145 cmdSubmit.Enabled = True
2146 End Sub
2147
2148 Private Sub Text23_Change()
2149 cmdSubmit.Enabled = True
2150 End Sub
2151
2152 Private Sub Text24_Change()
2153 cmdSubmit.Enabled = True
2154 End Sub
2155
2156
2157 'Code for frame of Slab
2158 Option Explicit
2159
2160 Private Sub cmdExit_Click()
2161 Unload Me
2162 End Sub
2163
2164 Private Sub cmdSubmit_Click()
2165 Dim cmd As String
2166 Dim fconnection As Connection
2167 'Dim pconnection As Connection
2168 Dim tmp As Boolean
2169 tmp = False
2170 Set fconnection = New Connection
2171 fconnection.Open
2172 "Provider=Microsoft.Jet.OLEDB.4.0;Data
2173 Source=ADMS.mdb;Persist Security
2174 Info=False"
2175 If Text1.Text <> "" Then
2176 If frmProdInput.exist = False Then
2177 cmd = "insert into Production values ("
2178 cmd = cmd + """" + frmProdInput.txtProduction.Text
2179 + """"
2180 cmd = cmd + ", " + "Date()"
2181 cmd = cmd + ", "" +
2182 frmProdInput.val_species.Caption + """"
2183 cmd = cmd + ", "" + frmProdInput.cboVessel.Text
2184 + """"
2185 cmd = cmd + ", "" + frmProdInput.cboSource.Text
2186 + """"
2187 cmd = cmd + ", "" + frmProdInput.cboArea.Text +
2188 """"
2189 cmd = cmd + ", " + frmProdInput.txtRate.Text
2190 cmd = cmd + ", " +
2191 frmProdInput.txtDayX.Text

```

```

2192 cmd = cmd + ", " +
2193 frmProdInput.txtActualX.Text
2194 cmd = cmd + ", " +
2195 frmProdInput.txtBaseX.Text
2196 cmd = cmd + ")"
2197 fconnection.Execute cmd
2198 frmProdInput.exist = True
2199 End If
2200 cmd = "insert into Slab values ("
2201 cmd = cmd + """" + val_prod_no.Caption + """"
2202 cmd = cmd + ", " + Text1.Text
2203 cmd = cmd + ", "
2204 cmd = cmd + Text2.Text
2205 cmd = cmd + ")"
2206 fconnection.Execute cmd
2207 tmp = True
2208 End If
2209
2210 If tmp = True Then
2211 frmProdInput.exist = True
2212 frmProdInput.txtProduction.Enabled = False
2213 frmProdInput.cboArea.Enabled = False
2214 frmProdInput.cboSource.Enabled = False
2215 frmProdInput.cboVessel.Enabled = False
2216 frmProdInput.txtActualX.Enabled = False
2217 frmProdInput.txtBaseX.Enabled = False
2218 frmProdInput.txtDayX.Enabled = False
2219 frmProdInput.cmdSlabs.Enabled = False
2220 frmProdInput.txtRate.Enabled = False
2221 End If
2222 Unload Me
2223 End Sub
2224
2225 Private Sub Form_Load()
2226 val_prod_no.Caption =
2227 frmProdInput.txtProduction.Text
2228 val_species.Caption = frmProd.sSpecies
2229 cmdSubmit.Enabled = False
2230 End Sub
2231
2232 Private Sub Text1_Change()
2233 cmdSubmit.Enabled = True
2234 End Sub
2235
2236 Private Sub Text2_Change()
2237 cmdSubmit.Enabled = True
2238 End Sub
2239
2240
2241 'Code for frame of Order
2242 Option Explicit
2243 Public customerName As String
2244
2245 Private Sub cmdExit_Click()
2246 Unload Me
2247 End Sub
2248
2249 Private Sub cmdNewCustomer_Click()
2250 Unload Me
2251 frmNewCustomer.Show 'vbModal, Me
2252 End Sub
2253
2254 Private Sub cmdProcess_Click()
2255 'Dim iOrderTemp As Variant
2256 'iOrderTemp = Shell("c:\Program
2257 Files\Microsoft Office\Office\EXCEL " &
2258 "D:\EEE\Sys\Thesis\ADMS\InvProTemplate.x
2259 lt", 1)
2260 customerName = DataCombo1.Text
2261 ActivateInvoice
2262 End Sub
2263
2264 Public appADMS As Excel.Application
2265 Public wbADMS As Excel.Workbook'
2266
2267 Sub ActivateInvoice()
2268 Dim shtInvoice As Excel.Worksheet
2269 Dim cn As ADODB.Connection
2270 Dim rs As ADODB.Recordset
2271 Dim cmd As String
2272 Dim i As Integer
2273
2274 On Error Resume Next
2275 Set appADMS = GetObject(,
2276 "Excel.Application")
2277 If Err.Number <> 0 Then
2278 Set appADMS =
2279 CreateObject("Excel.Application")

```

```

2280 End If
2281 Err.Clear
2282 On Error GoTo 0
2283 Set wbADMS =
2284 appADMS.Workbooks.Open(App.Path &
2285 "\InvProTemplate.xlt")
2286 wbADMS.Application.Visible = True
2287 'wbADMS.Window(1).Visible = True
2288 Set shtInvoice = wbADMS.Sheets(1)
2289 shtInvoice.Cells(6, 3).Value =
2290 frmOrder.DataCombo1.Text
2291
2292 Set cn = New Connection
2293 cn.Open "Provider =
2294 Microsoft.Jet.OLEDB.4.0;Data
2295 Source=C:\Documents and
2296 Settings\Guowei\Desktop\ADMS\ADMS.mdb
2297 ;Persist Security Info=False"
2298 Set rs = New Recordset
2299 cmd = "select * from Inventory order by
2300 index"
2301 rs.Open cmd, cn, adOpenForwardOnly,
2302 adLockReadOnly
2303 rs.MoveFirst
2304 i = 17
2305 Do While Not rs.EOF
2306 shtInvoice.Cells(i, 9).Value =
2307 rs("Quantity")
2308 i = i + 1
2309 rs.MoveNext
2310 Loop
2311 rs.Close
2312 Set rs = Nothing
2313 End Sub
2314
2315
2316 'Code for the frame of Customer Information
2317 Option Explicit
2318
2319 Private Sub cmdAdd_Click()
2320 Adodc1.Recordset.AddNew
2321 End Sub
2322
2323 Private Sub cmdExit_Click()
2324 Unload Me
2325 End Sub
2326
2327 Private Sub cmdSummit_Click()
2328 Adodc1.Recordset.MoveLast
2329 Adodc1.Refresh
2330 Unload Me
2331 frmOrder.Show 'vbModal, Me
2332 End Sub
2333
2334
2335 Sub ExcelToAccess()
2336 ' ExcelToAccess Macro
2337 ' Macro recorded 1/12/2002 by Guowei'
2338 ' Keyboard Shortcut: Ctrl+a
2339 '
2340 Dim r As Long
2341 Dim cn As ADODB.Connection
2342 Dim rsInvoice As ADODB.Recordset
2343 Dim rsInvCom As ADODB.Recordset
2344 Dim rsInventory As ADODB.Recordset
2345 Dim rs As ADODB.Recordset
2346 Dim tmpInvoice As Integer
2347 Dim tmpInventory As Integer
2348 Dim tmp As String
2349 Dim temItemNo As String
2350 Dim sqlCmd As String
2351
2352 ' connect to the Access database
2353 Set cn = New ADODB.Connection
2354 cn.Open
2355 "Provider=Microsoft.Jet.OLEDB.4.0;Data
2356 Source=C:\Documents and
2357 Settings\Guowei\Desktop\ADMS\ADMS.mdb
2358 ;Persist Security Info=False"
2359
2360 ' open a recordset
2361 Set rsInvoice = New ADODB.Recordset
2362 rsInvoice.Open "Invoice", cn, adOpenKeyset,
2363 adLockOptimistic, adCmdTable ' all records in a
2364 table
2365 Set rsInvCom = New ADODB.Recordset

```

```

2366     rsInvCom.Open "Inv_Com", cn, adOpenKeyset,
2367 adLockOptimistic, adCmdTable ' all records in a
2368 table
2369     Set rsInventory = New ADODB.Recordset
2370     rsInventory.Open "Inventory", cn,
2371 adOpenKeyset, adLockOptimistic,
2372 adCmdTable ' all records in a table
2373
2374     rsInvCom.AddNew
2375     rsInvCom.Fields("Invoice_No") =
2376 Range("L" & 6).Value
2377     rsInvCom.Fields("Date") = Range("L" &
2378 7).Value
2379     rsInvCom.Fields("Customer") = Range("C"
2380 & 6).Value
2381     rsInvCom.Fields("Ship") = Range("C" &
2382 7).Value
2383     rsInvCom.Update
2384
2385     r = 17 ' the start row in the worksheet
2386     Do While Len(Range("C" & r).Formula) >
2387 0 ' repeat until first empty cell in column A
2388         With rsInvoice
2389
2390             If Range("H" & r).Value <> "" Then
2391                 tmpInvoice = Range("H" & r).Value
2392                 tmpItemNo = Range("G" & r).Value
2393                 ' add values to each field in the
2394 record
2395                 .AddNew ' create a new record
2396                 .Fields("Invoice_No") = Range("L" &
2397 6).Value
2398                 .Fields("Item_No") = Range("G" &
2399 r).Value
2400                 .Fields("Quantity") = Range("H" &
2401 r).Value
2402                 .Fields("Unit_Price") = Range("J" &
2403 r).Value
2404                 .Update ' stores the new record
2405 sqlCmd = "select * from Inventory where
2406 ItemNo = ""
2407                 sqlCmd = sqlCmd + tmpItemNo +
2408 """"
2409                 Set rs = New ADODB.Recordset
2410
2411                 rs.Open sqlCmd, cn,
2412 adOpenForwardOnly, adLockReadOnly
2413                 rs.MoveFirst
2414                 tmpInventory = rs("Quantity")
2415                 rs.Close
2416                 Set rs = Nothing
2417                 tmp = tmpInventory - tmpInvoice
2418                 sqlCmd = "update Inventory set
2419 Quantity ="
2420                 sqlCmd = sqlCmd + tmp
2421 = ""
2422                 sqlCmd = sqlCmd + tmpItemNo +
2423 """"
2424                 cn.Execute sqlCmd
2425             End If
2426         End With
2427         r = r + 1 ' next row
2428     Loop
2429
2430     rsInvoice.Close
2431     Set rsInvoice = Nothing
2432     rsInvCom.Close
2433     Set rsInvCom = Nothing
2434     rsInventory.Close
2435     Set rsInventory = Nothing
2436     cn.Close
2437     Set cn = Nothing
2438 End Sub
2439
2440
2441 'Code for the frame of Invoice
2442 Option Explicit
2443 Public tmpNo As String
2444 'Public cn As Connection
2445 Public rs As Recordset
2446 Public appADMS As Excel.Application
2447 Public wbADMS As Excel.Workbook
2448 Dim tmp As String
2449
2450 Private Sub cmbInvoiceNo_Click()
2451     Dim cmd As String
2452     tmpNo = cmbInvoiceNo.Text
2453     If tmpNo <> "" Then

```

```

2454     cmdPreview.Enabled = True
2455     Set rs = New Recordset
2456     cmd = "select * from Inv_Com where
2457 Invoice_No = ""
2458     cmd = cmd + tmpNo
2459     cmd = cmd + """"
2460     rs.Open cmd, frmDescription.cn,
2461 adOpenForwardOnly, adLockReadOnly
2462     txtDate.Text = rs("Date")
2463     txtCustomer.Text = rs("Customer")
2464     txtShip.Text = rs("Ship")
2465     rs.Close
2466     Set rs = Nothing
2467 End If
2468 End Sub
2469
2470 Private Sub cmdExit_Click()
2471     Unload Me
2472 End Sub
2473
2474 Private Sub cmdPreview_Click()
2475     Dim rs_Invoice As Recordset
2476     Dim rs_Inventory As Recordset
2477     Dim cmd As String
2478     Dim i As Integer, j As Integer, rowNo As
2479 Integer
2480     Dim cellValue As Variant
2481     Dim shtInvoice As Excel.Worksheet
2482
2483     On Error Resume Next
2484     Set appADMS = GetObject(,
2485 "Excel.Application")
2486     If Err.Number <> 0 Then
2487         Set appADMS =
2488 CreateObject("Excel.Application")
2489     End If
2490     Err.Clear
2491     On Error GoTo 0
2492     Set wbADMS =
2493 appADMS.Workbooks.Open(App.Path &
2494 "\InvoiceTemplate.xlt")
2495     wbADMS.Application.Visible = True
2496     Set shtInvoice = wbADMS.Sheets(1)
2497
2498     shtInvoice.Cells(5, 11) =
2499 cmbInvoiceNo.Text
2500     shtInvoice.Cells(5, 3) = txtCustomer.Text
2501     shtInvoice.Cells(6, 3) = txtShip.Text
2502     shtInvoice.Cells(6, 11) = txtDate.Text
2503
2504     cmd = "select * from Invoice where Invoice_No
2505 = ""
2506     cmd = cmd +
2507 frmInvoice.cmbInvoiceNo.Text
2508     cmd = cmd + """"
2509
2510     Set rs_Invoice = New Recordset
2511
2512     rs_Invoice.Open cmd, frmDescription.cn,
2513 adOpenForwardOnly, adLockReadOnly
2514     rs_Invoice.MoveFirst
2515     i = 16
2516
2517     'rowNo = rs_Invoice.RecordCount
2518     'shtInvoice.Cells(3, 2) = rowNo
2519
2520     Set rs_Inventory = New Recordset
2521
2522     Do Until rs_Invoice.EOF
2523         tmp = rs_Invoice("item_no")
2524         cmd = "select * from Inventory where
2525 ItemNo = ""
2526         cmd = cmd + tmp
2527         cmd = cmd + """"
2528
2529         rs_Inventory.Open cmd,
2530 frmDescription.cn, adOpenForwardOnly,
2531 adLockReadOnly
2532         rs_Inventory.MoveFirst
2533
2534         shtInvoice.Cells(i, 2) =
2535 rs_Inventory("Spieces")
2536         shtInvoice.Cells(i, 3) =
2537 rs_Inventory("Size")
2538         shtInvoice.Cells(i, 4) =
2539 rs_Inventory("SO/B")
2540         shtInvoice.Cells(i, 5) =
2541 rs_Inventory("Package")

```

```

2542     shtInvoice.Cells(i, 6) =
2543 rs_Invoice("Item_No")
2544     shtInvoice.Cells(i, 7) =
2545 rs_Invoice("Quantity")
2546     shtInvoice.Cells(i, 9) =
2547 rs_Invoice("Unit_Price")
2548     i = i + 1
2549     rs_Invoice.Close
2550     rs_Invoice.MoveNext
2551 Loop
2552
2553 rowNo = i + 2
2554 j = rowNo
2555 For j = rowNo To 48 Step 1
2556     shtInvoice.Rows(rowNo).Delete
2557
2558     'rowNo = rowNo + 1
2559 Next j
2560 Set rs_Invoice = Nothing
2561 rs_Invoice.Close
2562 Set rs_Invoice = Nothing
2563
2564 End Sub
2565
2566 Private Sub Form_Load()
2567     'Set cn = New Connection
2568     'cn.Open "Provider =
2569 Microsoft.Jet.OLEDB.4.0;Data
2570 Source=C:\Documents and
2571 Settings\Guowe\i\Desktop\ADMS\ADMS.mdb
2572 ;Persist Security Info=False"
2573     cmdPreview.Enabled = False
2574
2575     Set rs = New Recordset
2576     rs.Open "Inv_Com", frmDescription.cn,
2577 adOpenForwardOnly, adLockReadOnly
2578     rs.MoveFirst
2579     Do Until rs.EOF
2580         frmInvoice.cmbInvoiceNo.AddItem
2581 (rs("Invoice_No"))
2582         rs.MoveNext
2583     Loop
2584     rs.Close
2585     Set rs = Nothing

```

```

2586 End Sub
2587
2588
2589 'Code for the frame of WorkSheet
2590 Option Explicit
2591 Public tmp_prod_no As String
2592 'Public cn As Connection
2593 Public rs As Recordset
2594 Public appADMS As Excel.Application
2595 Public wbADMS As Excel.Workbook
2596 Dim tmp As String
2597
2598 Private Sub cmbProdNo_Click()
2599     Dim cmd As String
2600     tmp_prod_no = cmbProdNo.Text
2601     If tmp_prod_no <> "" Then
2602         cmdPreview.Enabled = True
2603         Set rs = New Recordset
2604         cmd = "select * from production where
2605 prod_no = ""
2606         cmd = cmd + tmp_prod_no
2607         cmd = cmd + """"
2608         rs.Open cmd, frmDescription.cn,
2609 adOpenForwardOnly, adLockReadOnly
2610         txtArea.Text = rs("Area")
2611         txtDate.Text = rs("Date")
2612         txtSpecies.Text = rs("Species")
2613         txtVessel.Text = rs("Vessel")
2614         txtSource.Text = rs("Source")
2615     End If
2616 End Sub
2617
2618 Private Sub cmdExit_Click()
2619     Unload Me
2620 End Sub
2621
2622 Private Sub cmdPreview_Click()
2623     Dim rs_info As Recordset
2624     Dim rs_slab As Recordset
2625     Dim rs_frozen As Recordset
2626     Dim rs_fresh As Recordset
2627     Dim rs_Invoice As Recordset
2628     Dim cmd As String
2629     Dim i As Integer, j As Integer

```

```

2630 Dim rowNo As Integer
2631 Dim cellValue As Variant
2632 Dim temp As String
2633
2634 cmd = "select * from species_info where species
2635 = ""
2636 cmd = cmd + frmWS.txtSpecies.Text
2637 cmd = cmd + """"
2638 Set rs_info = New Recordset
2639 rs_info.Open cmd, frmDescription.cn,
2640 adOpenForwardOnly, adLockReadOnly
2641
2642 Dim shtInvoice As Excel.Worksheet
2643 On Error Resume Next
2644 Set appADMS = GetObject(,
2645 "Excel.Application")
2646 If Err.Number <> 0 Then
2647 Set appADMS =
2648 CreateObject("Excel.Application")
2649 End If
2650 Err.Clear
2651 On Error GoTo 0
2652 Set wbADMS =
2653 appADMS.Workbooks.Open(App.Path &
2654 "\worksheettemplate.xls")
2655 wbADMS.Application.Visible = True
2656 Set shtInvoice = wbADMS.Sheets(1)
2657
2658 cellValue = frmWS.txtDate.Text
2659 shtInvoice.Cells(5, 2).Value = cellValue
2660 cellValue = frmWS.cmbProdNo.Text
2661 shtInvoice.Cells(5, 5).Value = cellValue
2662 cellValue = frmWS.txtArea.Text
2663 shtInvoice.Cells(5, 8).Value = cellValue
2664 cellValue = frmWS.txtSource.Text
2665 shtInvoice.Cells(6, 5).Value = cellValue
2666 cellValue = frmWS.txtVessel.Text
2667 shtInvoice.Cells(6, 8).Value = cellValue
2668
2669 cellValue = rs("x_day")
2670 shtInvoice.Cells(13, 2).Value = cellValue
2671 cellValue = rs("x_base")
2672 shtInvoice.Cells(13, 4).Value = cellValue
2673 cellValue = rs("x_actual")
2674 shtInvoice.Cells(15, 2).Value = cellValue
2675 cellValue = rs_info("Man_Wage_lb")
2676 shtInvoice.Cells(9, 3).Value = cellValue
2677 cellValue = rs_info("Woman_Wage_lb")
2678 shtInvoice.Cells(10, 3).Value = cellValue
2679 cellValue = rs_info("Man_OT_lb")
2680 shtInvoice.Cells(9, 4).Value = cellValue
2681 cellValue = rs_info("Woman_OT_lb")
2682 shtInvoice.Cells(10, 4).Value = cellValue
2683 cellValue = rs_info("CostofFish_lb")
2684 shtInvoice.Cells(70, 3).Value = cellValue
2685 cellValue = rs_info("CostofOverhead_lb")
2686 shtInvoice.Cells(71, 3).Value = cellValue
2687 cellValue = rs_info("CostofLabor_lb")
2688 shtInvoice.Cells(72, 3).Value = cellValue
2689 cellValue = rs_info("Fresh_Freight")
2690 shtInvoice.Cells(74, 3).Value = cellValue
2691 cellValue = rs_info("Frozen_Freight")
2692 shtInvoice.Cells(75, 3).Value = cellValue
2693 cellValue = rs_info("Fresh_Pack")
2694 shtInvoice.Cells(77, 3).Value = cellValue
2695 cellValue = rs_info("Frozen_Pack")
2696 shtInvoice.Cells(78, 3).Value = cellValue
2697 cellValue = rs_info("Slabs_Pack")
2698 shtInvoice.Cells(80, 3).Value = cellValue
2699
2700 cmd = "select * from slab where Prod_No = ""
2701 cmd = cmd + frmWS.cmbProdNo.Text
2702 cmd = cmd + """"
2703 Set rs_slab = New Recordset
2704 rs_slab.Open cmd, frmDescription.cn,
2705 adOpenForwardOnly, adLockReadOnly
2706 cellValue = rs_slab("Frozen")
2707 shtInvoice.Cells(70, 7).Value = cellValue
2708 cellValue = rs_slab("Fresh")
2709 shtInvoice.Cells(71, 7).Value = cellValue
2710
2711 cmd = "select * from frozen where Prod_No = ""
2712 cmd = cmd + frmWS.cmbProdNo.Text
2713 cmd = cmd + """"
2714 i = 33
2715 Set rs_frozen = New Recordset
2716 rs_frozen.Open cmd, frmDescription.cn,
2717 adOpenForwardOnly, adLockReadOnly

```

```

2718 rs_frozen.MoveFirst
2719 Set rs_Inventory = New Recordset
2720 Do Until rs_frozen.EOF
2721     tmp = rs_frozen("item_no")
2722     cmd = "select * from inventory where ItemNo
2723 = ""
2724     cmd = cmd + tmp
2725     cmd = cmd + """"
2726
2727     rs_Inventory.Open cmd, frmDescription.cn,
2728 adOpenForwardOnly, adLockReadOnly
2729     rs_Inventory.MoveFirst
2730
2731     cellValue = rs("species")
2732     shtInvoice.Cells(i, 1).Value = cellValue
2733     cellValue = rs_Inventory("size")
2734     shtInvoice.Cells(i, 2).Value = cellValue
2735     cellValue = rs_Inventory("SO/B")
2736     shtInvoice.Cells(i, 3).Value = cellValue
2737     cellValue = rs_Inventory("Package")
2738     shtInvoice.Cells(i, 4).Value = cellValue
2739     cellValue = tmp
2740     shtInvoice.Cells(i, 5).Value = cellValue
2741     cellValue = rs_frozen("weight")
2742     shtInvoice.Cells(i, 6).Value = cellValue
2743     cellValue = rs("us_return")
2744     shtInvoice.Cells(i, 8).Value = cellValue
2745     i = i + 1
2746     rs_Inventory.Close
2747     rs_frozen.MoveNext
2748 Loop
2749
2750 cmd = "select * from fresh where prod_no
2751 = ""
2752 cmd = cmd + frmWS.cmbProdNo.Text
2753 cmd = cmd + """"
2754 i = 20
2755 Set rs_fresh = New Recordset
2756 rs_fresh.Open cmd, frmDescription.cn,
2757 adOpenForwardOnly, adLockReadOnly
2758 rs_fresh.MoveFirst
2759 Do Until rs_fresh.EOF
2760     cellValue = rs_fresh("size")
2761     shtInvoice.Cells(i, 1).Value = cellValue
2762     cellValue = rs_fresh("weight")
2763     shtInvoice.Cells(i, 2).Value = cellValue
2764     cellValue = rs("us_return")
2765     shtInvoice.Cells(i, 3).Value = cellValue
2766     i = i + 1
2767     rs_fresh.MoveNext
2768 Loop
2769
2770 End Sub
2771
2772 Private Sub Form_Load()
2773     'Set cn = New Connection
2774     'cn.Open "Provider =
2775 Microsoft.Jet.OLEDB.4.0;Data
2776 Source=C:\Documents and
2777 Settings\Guowei\Desktop\ADMS\ADMS.mdb
2778 ;Persist Security Info=False"
2779     cmdPreview.Enabled = False
2780
2781     Set rs = New Recordset
2782     rs.Open "production", frmDescription.cn,
2783 adOpenForwardOnly, adLockReadOnly
2784     rs.MoveFirst
2785     Do Until rs.EOF
2786         frmWS.cmbProdNo.AddItem
2787 (rs("prod_no"))
2788         rs.MoveNext
2789     Loop
2790     rs.Close
2791     Set rs = Nothing
2792 End Sub
2793
2794 Sub SavetoDB()
2795 '
2796 ' SavetoDB Macro
2797 ' Macro recorded 2/27/2002 by Guowei'
2798 ' Keyboard Shortcut: Ctrl+s'
2799     Dim cn As ADODB.Connection
2800     Dim rsIncomeVar As ADODB.Recordset
2801     Dim tmp As Double
2802     'Dim sqlCmd As String
2803
2804     ' connect to the Access database
2805     Set cn = New ADODB.Connection

```

```

2806     cn.Open
2807     "Provider=Microsoft.Jet.OLEDB.4.0;Data
2808     Source=C:\Documents and
2809     Settings\Guowe\ Desktop\ADMS\ADMS.mdb
2810     ;Persist Security Info=False"
2811
2812     ' open a recordset
2813     Set rsIncomeVar = New ADODB.Recordset
2814     rsIncomeVar.Open "IncomeVar", cn,
2815     adOpenKeyset, adLockOptimistic,
2816     adCmdTable ' all records in a table
2817
2818     rsIncomeVar.MoveFirst
2819     tmp = rsIncomeVar("FreshRev") +
2820     Range("G" & 28).Value
2821     rsIncomeVar.Fields("FreshRev") = tmp
2822     'rsInvCom.Fields("Invoice_No") =
2823     Range("L" & 6).Value
2824     tmp = rsIncomeVar("FrozenRev") +
2825     Range("L" & 58).Value
2826     rsIncomeVar.Fields("FrozenRev") = tmp
2827     tmp = rsIncomeVar("SlabRev") +
2828     Range("F" & 63)
2829     rsIncomeVar.Fields("SlabRev") = tmp
2830     tmp = rsIncomeVar("FishCost") +
2831     Range("D" & 70)
2832     rsIncomeVar.Fields("FishCost") = tmp
2833     tmp = rsIncomeVar("Overhead") +
2834     Range("D" & 71)
2835     rsIncomeVar.Fields("Overhead") = tmp
2836     tmp = rsIncomeVar("Freight") + Range("D"
2837     & 74) + Range("D" & 75)
2838     rsIncomeVar.Fields("Freight") = tmp
2839     tmp = rsIncomeVar("Packing") +
2840     Range("D" & 77) + Range("D" & 78) +
2841     Range("D" & 80)
2842     rsIncomeVar.Fields("Packing") = tmp
2843     tmp = rsIncomeVar("Labour") + Range("D"
2844     & 72)
2845     rsIncomeVar.Fields("Labour") = tmp
2846     rsIncomeVar.Update
2847
2848     rsIncomeVar.Close
2849     Set rsIncomeVar = Nothing
2850 End Sub
2851
2852
2853 'Code for the frame of Inventory
2854 Option Explicit
2855 Public sSpecies As String
2856 'Public cn As Connection
2857 Public rs As Recordset
2858 Public appADMS As Excel.Application
2859 Public wbADMS As Excel.Workbook
2860
2861 Private Sub cmbSpecies_Click()
2862     If cmbSpecies.DataChanged = True Then
2863         sSpecies = cmbSpecies.Text
2864     End If
2865     If sSpecies <> "" Then
2866         cmdPreview.Enabled = True
2867     End If
2868 End Sub
2869
2870 Private Sub cmdPreview_Click()
2871     Dim cmd As String
2872     Dim i As Integer, j As Integer
2873     Dim rowNo As Integer
2874     'Dim cellRange As String
2875     Dim cellValue As Variant
2876     'Dim cellPlace As String
2877     'Dim cellName As String
2878     'Dim totalRows As Integer
2879     'Dim nameExcel As String
2880     Dim temp As String
2881     Dim shtInvoice As Excel.Worksheet
2882     Dim ssName As String
2883     On Error Resume Next
2884     Set appADMS = GetObject(,
2885     "Excel.Application")
2886     If Err.Number <> 0 Then
2887         Set appADMS =
2888         CreateObject("Excel.Application")
2889     End If
2890     Err.Clear
2891     On Error GoTo 0

```

```

2892 Set wbADMS =
2893 appADMS.Workbooks.Open(App.Path &
2894 "inventory.xls")
2895 wbADMS.Application.Visible = True
2896 'wbADMS.Window(1).Visible = True
2897 Set shtInvoice = wbADMS.Sheets(1)
2898 'shtInvoice.Cells(6, 3) =
2899 frmOrder.DataCombo1.Text
2900
2901 'Dim saveCursor
2902 'saveCursor = Me.MousePointer
2903 'Me.MousePointer = vbHourglass
2904 cmd = "select * from inventory where
2905 Spieces = "
2906 cmd = cmd + ""
2907 cmd = cmd + sSpecies
2908 cmd = cmd + ""
2909 Set rs = New Recordset
2910 rs.Open cmd, frmDescription.cn.
2911 adOpenForwardOnly, adLockReadOnly
2912 'oleExcel.CreateEmbed "", "Excel.Sheet.8"
2913 On Error GoTo 0
2914 rowNo = 7
2915 i = 5
2916 'ssName = oleExcel.object.Name
2917 'oleExcel.object.Worksheets(1).Row = 6
2918 'oleExcel.object.Worksheets(1).Col = 24
2919 Do While Not rs.EOF
2920 'rowNo = rowNo + 1
2921 For j = 0 To i
2922 cellValue = rs(j)
2923
2924 'oleExcel.object.Worksheets(1).Cells(rowNo, j
2925 + 1).Value = cellValue
2926 shtInvoice.Cells(rowNo, j + 3).Value =
2927 cellValue
2928 Next
2929 rowNo = rowNo + 1
2930 rs.MoveNext
2931 Loop
2932 'Set appADMS.Worksheets =
2933 oleExcel.object.Worksheets(1)
2934 'appADMS.Visible = True
2935 rs.Close
2936 Set rs = Nothing
2937 'Exit Sub
2938 'OLEError:
2939 ' MsgBox "An OLE error occurred,
2940 probably because Excel is not installed on this
2941 computer."
2942 ' Unload Me
2943 End Sub
2944
2945 Private Sub cmdExit_Click()
2946 'cn.Close
2947 Unload Me
2948 End Sub
2949
2950 Private Sub Form_Load()
2951 'bChange = False
2952 cmbSpecies.AddItem ("Catfish")
2953 cmbSpecies.AddItem ("Cod")
2954 cmbSpecies.AddItem ("Flounder")
2955 cmbSpecies.AddItem ("Haddock")
2956 cmbSpecies.AddItem ("Ocean Perch")
2957 cmbSpecies.AddItem ("Pollock")
2958 cmbSpecies.AddItem ("Turbot")
2959 'Set cn = New Connection
2960 'cn.Open "Provider =
2961 Microsoft.Jet.OLEDB.4.0;Data
2962 Source=ADMS.mdb;Persist Security
2963 Info=False"
2964
2965 'If bChange = True Then
2966 'cmbSpecies.Text = sSpecies
2967 'Else
2968 'sSpecies = cmbSpecies.Text
2969 'End If
2970 cmdPreview.Enabled = False
2971 End Sub
2972
2973
2974 'Code for the frame of Income Statement
2975 Option Explicit
2976 'Public tmp_prod_no As String
2977 'Dim cn As Connection
2978 Dim appADMS As Excel.Application
2979 Dim wbADMS As Excel.Workbook

```

```

2980 Dim tmp As String
2981
2982 Private Sub cmdExit_Click()
2983     Unload Me
2984 End Sub
2985
2986 Private Sub cmdOk_Click()
2987     'dataIncome.Recordset.MoveLast
2988     dataIncome.Recordset.Update
2989     dataIncome.Refresh
2990     cmdPreview.Enabled = True
2991 End Sub
2992
2993 Private Sub cmdPreview_Click()
2994     Dim rsInFix As Recordset
2995     Dim rsInVar As Recordset
2996     Dim shtInvoice As Excel.Worksheet
2997
2998     On Error Resume Next
2999     Set appADMS = GetObject(,
3000 "Excel.Application")
3001     If Err.Number < 0 Then
3002         Set appADMS =
3003 CreateObject("Excel.Application")
3004     End If
3005     Err.Clear
3006     On Error GoTo 0
3007     Set wbADMS =
3008 appADMS.Workbooks.Open(App.Path &
3009 "\IncomeTemplate.xlt")
3010     wbADMS.Application.Visible = True
3011     Set shtInvoice = wbADMS.Sheets(1)
3012
3013     Set rsInFix = New Recordset
3014     rsInFix.Open "IncomeFix",
3015 frmDescription.cn, adOpenForwardOnly,
3016 adLockReadOnly
3017     rsInFix.MoveFirst
3018     Set rsInVar = New Recordset
3019     rsInVar.Open "IncomeVar",
3020 frmDescription.cn, adOpenForwardOnly,
3021 adLockReadOnly
3022     rsInVar.MoveFirst
3023
3024     shtInvoice.Cells(11, 5) =
3025 rsInVar("FreshRev")
3026     shtInvoice.Cells(12, 5) =
3027 rsInVar("FrozenRev")
3028     shtInvoice.Cells(13, 5) = rsInVar("SlabRev")
3029     shtInvoice.Cells(18, 5) =
3030 rsInVar("FishCost")
3031     shtInvoice.Cells(19, 5) =
3032 rsInVar("Overhead")
3033     shtInvoice.Cells(20, 5) = rsInVar("Freight")
3034     shtInvoice.Cells(21, 5) = rsInVar("Packing")
3035     shtInvoice.Cells(26, 5) = rsInVar("Labour")
3036
3037     shtInvoice.Cells(14, 5) =
3038 rsInFix("OtherRev")
3039     shtInvoice.Cells(24, 5) = rsInFix("MisCost")
3040     shtInvoice.Cells(33, 5) = rsInFix("License")
3041     shtInvoice.Cells(34, 5) =
3042 rsInFix("Depreciation")
3043     shtInvoice.Cells(35, 5) = rsInFix("Repairs")
3044     shtInvoice.Cells(36, 5) =
3045 rsInFix("Insurance")
3046     shtInvoice.Cells(37, 5) = rsInFix("Interest")
3047
3048     rsInFix.Close
3049     Set rsInFix = Nothing
3050     rsInVar.Close
3051     Set rsInVar = Nothing
3052 End Sub
3053
3054 Private Sub Form_Load()
3055     'Set cn = New Adodb.Connection
3056     'cn.Open "Provider =
3057 Microsoft.Jet.OLEDB.4.0;Data
3058 Source=C:\Documents and
3059 Settings\Guowei\Desktop\ADMS\ADMS.mdb
3060 ;Persist Security Info=False"
3061
3062     cmdPreview.Enabled = False
3063     txtOtherRev.ToolTipText = "The revenue
3064 other than fresh, frozen, or slab"
3065     lblOther.ToolTipText = "The revenue other
3066 than fresh, frozen, or slab"
3067 End Sub

```