CERTIFICATION OF APPROVAL

CONVERSION & OPTIMIZATION OF PILOT PLANT DATABASE

by

Akmaruddin bin Jofri

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved:

[illegible]

Ms. Suhaila Badarol Hisham
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

December 2005

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

[illegible]

Akmaruddin Bin Jofri

ABSTRACT

A database system by *ICONICS* was used for Pilot Plant in Plant Process Control Laboratory of Universiti Teknologi Petronas. The Pilot Plant employs a Distributed Control System and simulates a basic process control system. Users of the database find it difficult to use the data to be utilized in other software such as Matlab because of its improper arrangement of data. Thus, main concern in this project is to convert the database file obtained from the UTP Pilot Plant to another file with better arrangement of data. An offline converter application has been successfully developed by using Visual Basic 6, which can convert the database to another smaller size file with the desired arrangement of data. The online converter was developed but could not be completed on time. Issues and problems regarding both offline and online converter application will be discussed thoroughly through out this report.

ACKNOWLEDGEMENTS

Praise to Allah, for giving me all the strength and courage to undergo this project and with His Will this project has progressed throughout this semester.

A special thanks to my supervisor, Ms. Suhaila Badarol Hisham, firstly, for awarding me this project and the most important thing, for helping, motivating and sharing her ideas with me. With her help, my path towards accomplishment of this project became clearer.

Big thanks to Mr. Azhar, the laboratory instructor, who has been helping me a lot in getting the information on the database and also for lending me the required documentations and manuals of the Pilot Plant without any hesitation or second thought.

My thanks also go to my friends, Aini, Lokman and Rosli, which are currently taking IT, for helping, advising and sharing their thoughts with me in the development of the program, especially the Visual Basic part. They have been both very inspirational and supportive throughout this project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ADO	–	ActiveX Data Object
CAL	–	Client Access Line
CSV	–	Comma Separated Values
DBMS	–	Database Management System
DCS	–	Distributed Control System
HMI	–	Human-Machine Interface
IT	–	Information Technology
MEDE	–	Microsoft® Excel Data Exchange
MS	–	Microsoft®
MSDE	–	Microsoft® Data Exchange
ODA	–	Open Data Access
ODBC	–	Open Database Connectivity
OLEDB	–	Object Linking and Embedding for Databases
OPC	–	OLE for Process Control
OEM	–	Original Equipment Manufacturer
SQL	–	Structured Query Language
UTP	–	Universiti Teknologi PETRONAS
VB	–	Visual Basic
VB.Net	–	Visual Basic.Net
VBA	–	Visual Basic for Applications

CHAPTER 1
INTRODUCTION

1.1 Background

UTP Plant Process Control Laboratory uses *ICONICS Plant Process Solution* Database Management System for the Pilot Plant. The *ICONICS Plant Process Solution* uses Microsoft Access for database storage. The data recorded in the database file was not arranged properly and user finds it difficult to utilize the data. The previous database by Honeywell has excellent features and ease of usability. It uses Microsoft Excel for spreadsheets and data storage, a well-known and simple software. This project focuses in solving the data arrangement problems, optimizing the current database system while identifying other arisen problems if any.

1.2 Problem Statement

User of the Pilot Plant database need the data inside the database file to be implemented in MATLAB for further analysis and utilization of the process data. The arrangement of data in database file generated by the Pilot Plant server is not suitable to be imported by MATLAB *(Please refer to Appendix D, Table 2 for example of initial data arrangement)*. MATLAB can import a collection data from Excess or Excel file and the data sequence from the first to the end must be in one column for the MATLAB to identify it as a set of data. But for current database file arrangement, the data were divided into 36 columns of data. Thus, user has to rearrange the data to only one column manually before exporting them to MATLAB *(Please Refer to Appendix D, Table 3 for example of desired data arrangement)*. Thus, a mechanism to rearrange the data automatically is required.

1.3 Objectives

1. To conduct study on the pilot plant database.
2. To propose a database conversion method for Pilot Plants that does not affect the performance of the system, can be developed on time, reliable and can solve the problem stated. Improvements on the performances of the system were encouraged.
3. To successfully develop and implement the proposed database conversion and optimization in the Pilot Plant DCS.

CHAPTER 2
LITERATURE REVIEW/ THEORY

With clear definitions of the problems and clear objectives to be achieved as discussed in the previous chapter, more literature review has to be done to gather all the required theory and information to be implemented in the project. Thus, this chapter will discuss on some important theory and subjects that are important to be reviewed and focused throughout this project.

2.1 Process Control System

A process basically describes how an input material is changed to be the desired output. Process control system deals with the system controlling the process or the changes from input to output. Controller continually reads data from a sensor and calculates final element adjustments to maintain a desired value.

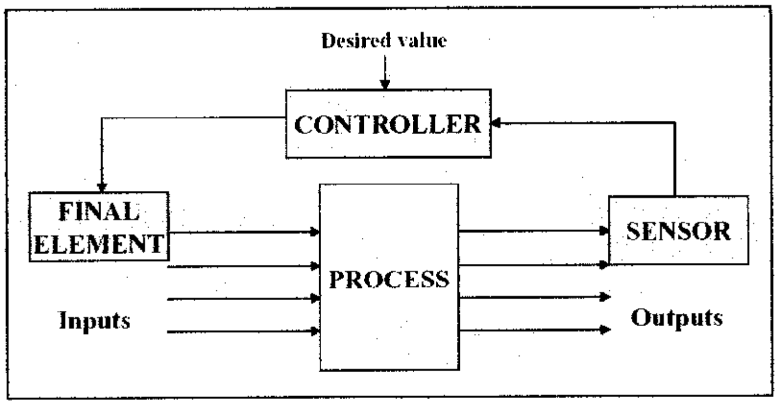


Figure 1   Basic Process Control System

2.2 Distributed Control System

Distributed control system (DCS) has become more important in process control system because of its capability in implanting a complex control system with stringent requirements involving fault tolerance and flexibility that will be of prevalence in the future [1]. A real time process control system consists of not only one or several loops, but may consist of hundreds or more of control loops.

DCS involve a set of control systems implemented in a distributed manner using an appropriate communication protocol. It is generally digital, and normally consists of field instruments, connected via wiring or busses to multiplexer/demultiplexers and A/D's or analog to digital and finally the Human-Machine Interface (HMI) or control consoles [2].

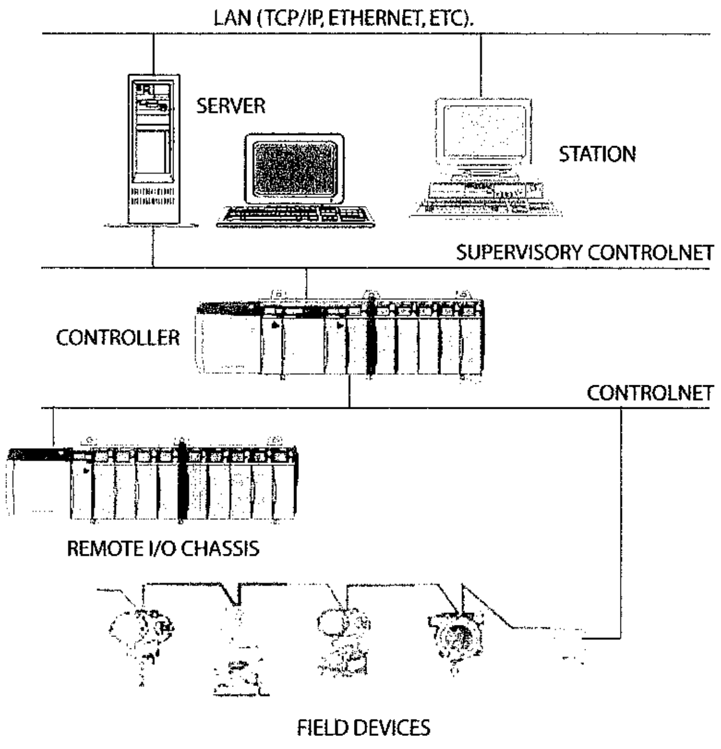


Figure 2 Distributed Control System Communication Network

2.3 Communication Networks

DCS is constructed in a structural and organized manner since it must have an excellent communications, interactions and controls among sensors, transmitters, controllers, servers, computers etc. Networks of DCS can be divided into four hierarchies or levels namely sensor networks, devices networks, control networks and the highest level, enterprise network. These networks use different protocols or languages that define how the data is transmitted and manipulated.

Sensor network is the simplest network in the DCS and was originally designed primarily for digital (on/off) interface. These busses target three specific areas: actuators (i.e., solenoid valves and motor starters); manufacturing automation; and sensors (i.e., limit switches and pushbuttons).

The second level is the device network that supports process automation, more complex transmitters, and valve actuators [3]. In device level protocols, data is floating point and status information is usually available.

Control level networks are the backbone for communications between I/O systems, controllers, operator stations and supervisory systems. This network deals with huge chunks of heterogeneous data and operates at high data rates.

Enterprise networks are a collection of LANs, Wide Area Networks (WANs), and a wide range of communication protocols. In these networks, several protocols are frequently in simultaneous use on the same physical connection. The data carried are diverse and can include everything from process and production data to e-mail, music, images or a wide range of business and financial transactions [4].

2.4 Database

All data and information are stored in the Enterprise Network gathered from the lower networks level after data conversion to a measurable and usable data. Data in DCS ranges from a simple plain text to videos and sounds. At this level, since PC based system is used, software being used can varies a lot. Rather than using special software like in the old days for data storage, current system implements Windows based operating system, a well known and open system, that provides easy integration with common software such as Microsoft Excel, Microsoft Access, Microsoft Word, AutoCAD, audio/video player, Internet Explorer, Lotus suit and many other applications [5]. With wide applications of software and PC tools, a lot of document types can be developed in a DCS. It only depends on the manufacturers or software developer in choosing the best solutions for data management and storage.

2.5 Honeywell PlantScape® Solution

Honeywell PlantScape® Solution manuals were reviewed to look at some of the system components for basic ideas on how the database was linked to the server.

In Honeywell PlantScape® Solution uses platform such as PlantScape Open Data Access, Microsoft Excel Data Exchange, Open Database Connectivity and OLE for Process Control [6].

2.5.1 PlantScape Open Data Access (ODA)

PlantScape ODA supports the exchange of real-time and historical data with other systems or applications.

2.5.2 Microsoft Excel Data Exchange (MEDE)

MEDE provides data exchange to Microsoft Excel Spreadsheet.

2.5.3 Open Database Connectivity (ODBC)

ODBC driver allows ODBC-compliant client applications such Crystal Reports, or Microsoft Access to retrieve PlantScape data.

2.5.4 OLE for Process Control (OPC)

OPC is an industry standard for exchange of process control system data. PlantScape Application Program Interface (API) allows user-written applications to run on either the server or over the network through.

These platforms for data exchange provide options for user to use various applications provided by Honeywell or user-written applications to retrieve or to write data to or from the database.

2.6 ICONICS TrendWorX^{TM}

Data collection and trending for ICONICS TrendWorX^{TM} is via TrendWorX32 SQL Server (TWXSQLSvr). TrendWorX32 SQL Server provides both a data collection/repository system and a data retrieval station for historical data to trending and reporting clients [7].

TWXSQLSvr uses ActiveX Data Objects (ADO), a database access mechanism. ADO provides Object Linking and Embedding for Databases (OLEDB) that can be used to program any database as long as there is a dedicated OLEDB Data Provider for that database. Applications based on ADO/OLEDB can potentially access any database regardless of the underlying storage media, file system and location. ADO Objects 2.1 in TWXSQLSvr provides Universal OLEDB ODBC Data Providers and also native OLEDB Providers which enable to access MS Access and SQL Server, MSDE and Oracle compliant databases.

CHAPTER 3
METHODOLOGY/ PROJECT WORK

To achieve the objectives of the project, proper methodology has to be planned and executed so that the project can be successfully finished within the limitations such as time, facilities and equipments. This chapter will discuss on list of methodology that will be carried out in the project.

3.1 Problem Statement and Objectives Identification

After the topic was awarded, the project background, problem statement and objectives of the project were identified to have full understanding and clear goals of the project. From the beginning of the project, the first problem emphasized was the improper arrangement of the numerical data inside the database that cause difficulties for user to use the data.

Thus, it was defined for the project that a solution has to be developed to improve and optimize the current database used for the aforementioned pilot plants for a more user-friendly layout and easier access to numerical experiment data. This is the main objective of the project, and from this objective, it will be the determinant for the success of this project. Several objectives also have been listed to extend the scope of the project as stated in Chapter 1. This project was evaluated based on these objectives.

3.2 Project Planning

With full understanding of the project, activities and work flow of the project were planned for achieving the desired goals. Gant Charts on Appendix A shows the planning that have been set for semester 1 and semester 2 of the project duration.

3.3 Literature Review/ Theory

Relevant materials and references were collected for gaining more information and understanding on the project such as books, journals, laboratory manuals, pilot plant operating manuals, websites and other suitable materials.

This project started with merely problem definitions and its objectives, which means it is a new project and no materials available for references. Thus, a lot of literature review and reading have to be done to gather information. The project was started with unclear motives and did not have specific focuses on which subject to be reviewed.

From the problem statement, the principles and operation of Distributed Control System (DCS) employed in the UTP Pilot Plant have to be explored and mastered. Thus, the literature review started with this subject. With knowledge in DCS, the solution could be generated to be compatible with the system and would not affect the system performances. Pilot Plant manuals, conference papers, books, slides and websites were reviewed to have the basic knowledge on DCS. The architecture of the UTP Pilot Plant itself were explored and learned.

The next focus is on the UTP Pilot Plant database system itself. The ICONICS catalogues and Honeywell manuals were reviewed. ICONICS TrendWorx ™ 32 is the database management system (DBMS) implemented currently in UTP Pilot Plant. From the catalogue, the information on the database architecture and application were obtained. Honeywell Plantscape Solutions was the previous DBMS used for the UTP Pilot Plant. Since the system has performed very well before, it might be useful to evaluate the system to find solution alternatives that can be implemented on the current system.

The literature review continues with focus on database software and application. Books on database development, database applications, and database programming were reviewed to obtain information on possible solutions and generating ideas for solving the problems. Database software such as Microsoft® Access, Microsoft® Excel, Oracle, Microsoft® SQL Server and others were reviewed and evaluated whether they can be used as a solution. Advantage and disadvantage for each application were weighed and justified for choosing the best solution.

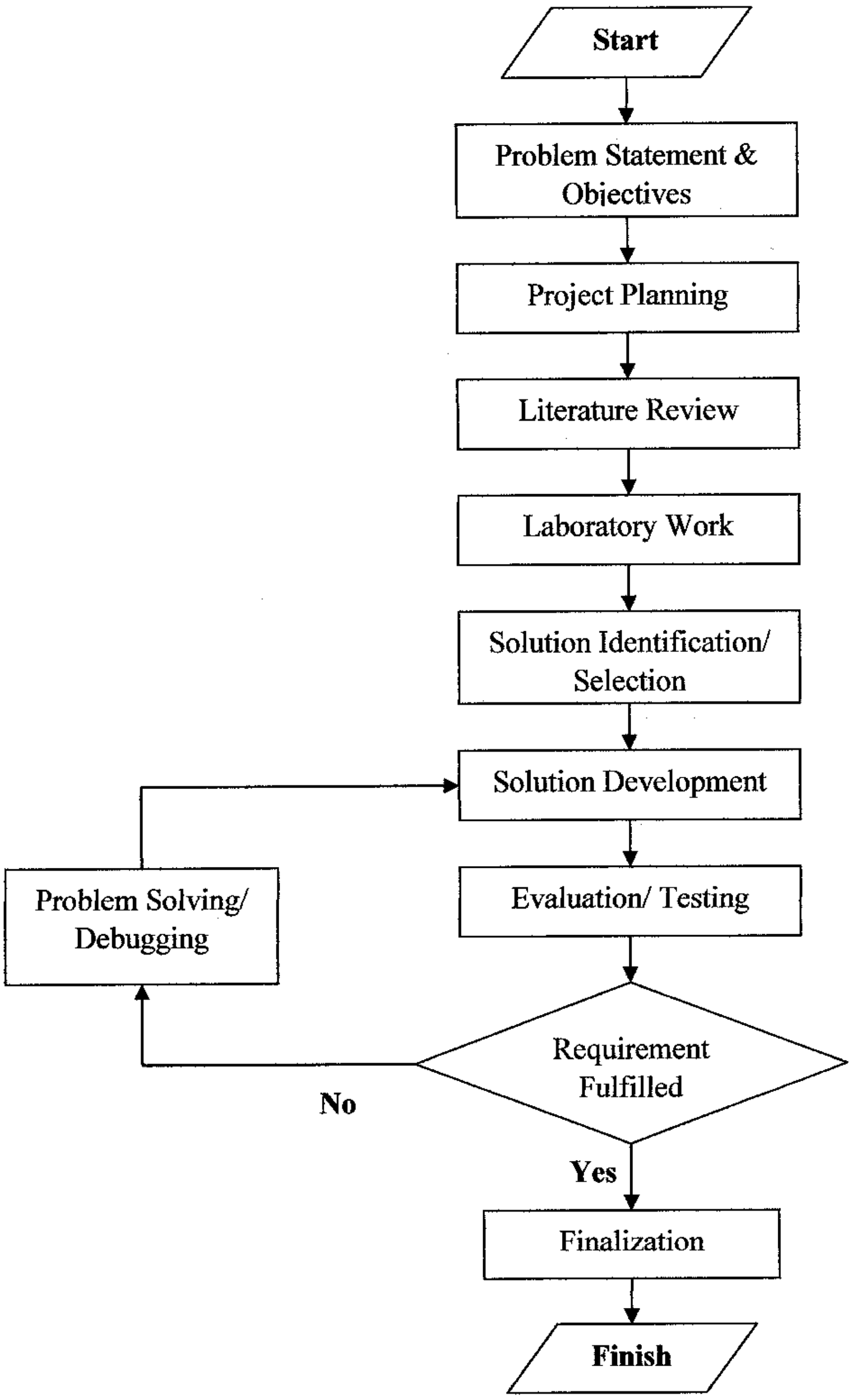


Figure 3 Methodology Flow Chart

3.4 Laboratory Work

Laboratory work was done for analysis, experiment observations and evaluations of the system. With hands-on experience on the pilot plant, the overall systems as well as problems were clearly understood and alternatives of solutions were generated. A suitable solution was chosen and developed.

Discussion was done with supervisor or laboratory technician on the problems and issues regarding the pilot plant database. All problems were listed down and verified with the early problem statement before deciding whether to extend the objectives or to stick with the previous scope of work. More details on laboratory work will be discussed in Chapter 5.

3.5 Solution Identification/ Selection

Solutions for arising problem before were identified. Several alternatives of the solutions were listed and justified. After advantageous and disadvantageous for each of the solutions were found out, the best solutions were developed for implementation. The solutions were developed offline before implemented on the real system.

After problems were identified, solutions were generated based on the findings and literature review. Solutions required were for implementation offline and online. Since the database storage and data processing problems could only be solved by online solutions, so the offline solutions was aimed for solving the main problem, which was the improper arrangement of data. The alternatives suggested were:

i. Visual Basic application for converting data.
ii. Conversion is done in the Microsoft Access itself.

The online solution was expected to solve all the listed problems including the storage and processing time problems. Because of the limitations of the ICONICS TrendWorx32, only these software can be used for database storage:

i. Microsoft® Access
ii. Microsoft® SQL Server 6.5 and 7.0
iii. Microsoft® Data Engine (MSDE)
iv. Oracle

The alternatives suggested for the online solution were:

i. Implementing Microsoft® SQL Server to increase the capability of the DBMS to manage file and distribute data.
ii. To improve the data acquisition and arrangement inside the Microsoft Access.
iii. To improve the ICONICS TrendWorx ™ 32 itself.
iv. To use other application to be export the data from MS Access to another file with the desired arrangement.

More on solutions selection will be discussed in Chapter 5.

3.6 Solution Development

After suitable solution was chosen, the application was developed. The development started with the offline converter. The online converter was planned to be developed after the offline converter had been completed. More time was allocated for online converter compared to offline converter. But because of arising problems and changed in project scope, more time was spent in development of the offline converter. Solution development will be discussed in more detail in Chapter 5.

3.7 Evaluation/ Testing

With this model, the solution was assessed and evaluated for improvement. Possible problems that may occur during implementation on the pilot plant were identified.

3.8 Problem Solving/ Debugging

After coming out with a reliable offline solution model, the project is to be continued with online solution model development for real time implementation on the pilot plant. The solution model will be developed starting on early of the next semester. Solution model will be tested, assessed and improved time after time. Arisen problems will be identified and solved. Better improvement on the solution is to be employed if possible.

3.9 Finalization

Solution is to be finalized by considering whether the objectives are achieved or not. Discussions and assessment of the final solution will be done for evaluating the successfulness of the project. Possible causes that lead to failure or incomplete outcome will be identified and commented for future references.

CHAPTER 4
RESULTS

This chapter will discuss achievements in the project. Each solution developed was discussed and evaluated.

4.1 UTP Pilot Plant Database

Database for UTP Pilot Plant is in MS Access format. There are 13 tables inside the database (*Refer to Appendix B, Figure 18*). Those tables are:

- Flowl,
- Flowl_Tags,
- Flow_Info,
- Flow_Notes,
- LevelVol,
- LevelVol_Tags,
- LevelVol_Info,
- LevelVol_Notes,
- Pressure1,
- Pressure1_Tags,
- Pressure1_Info,
- Pressure1_Notes,
- TemperatureVol,
- TemperatureVol_Tags,
- TemperatureVol_Notes,
- TemperatureVol_Info
- TWX_Global. Flowl,

Flow1, LevelVol, Pressure1 and TemperatureVol tables are the Data Tables that contain all the experimental data obtained from the plant activities. There are 185 fields in each table, which consist of Signal Index, Earliest Time, Latest Time, Record Modified, Fill Index and 180 fields for experimental data. In each row, there are 36 set of data and each data have information on Sample Time and Date, Sample Milliseconds, Sample Values, Sample Quality and Sample Modified. *(Refer to Appendix B, Figure 19).*

The Tags Tables (i.e. Flow1_Tags, LevelVol_Tags, TemperatureVol_Tags, Pressure1_Tags) provide the information on the input signal from the plant to DCS such as signal source, tag numbers, high and low limit of the signal and the descriptions. *(Refer to Appendix B, Figure 20).*

Notes Tables (i.e. Flow1_Notes, LevelVol_Notes, TemperatureVol_Notes, Pressure1_Notes) usually are empty. Any notes assign to tags indexes will be included in it. *(Refer to Appendix B, Figure 21).* The Information Tables (i.e. Flow1_Info, LevelVol_Info, TemperatureVol_Info, Pressure1_Info) contain the Start Time and End Time for particular experiment. *(Refer to Appendix B, Figure 22)*

Finally, the TWX_Global table provides the information on the tables available in the database. The table consists of 12 fields that contain the list of tables and other information such as Tables Number, Tables Group etc. *(Refer to Appendix B, Figure 23).*

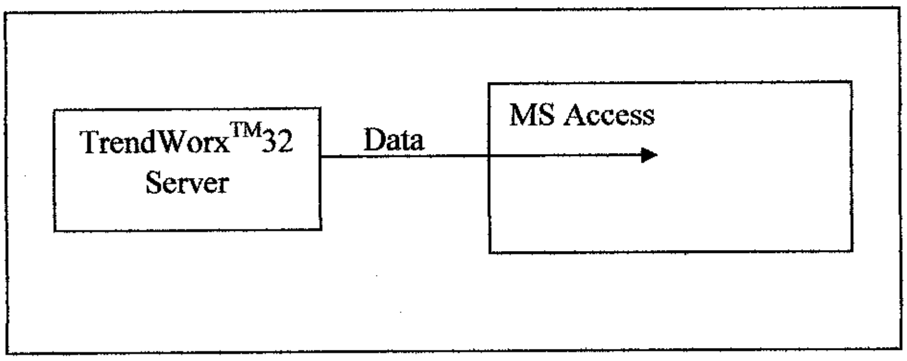
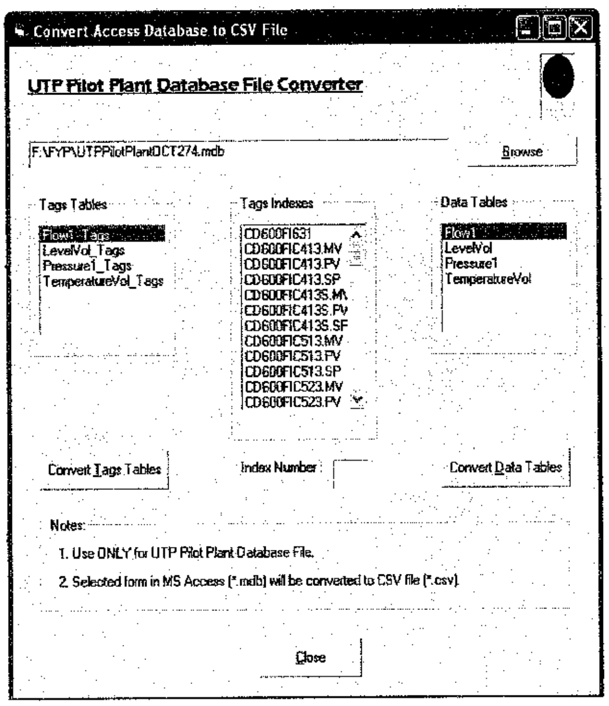


Figure 4   Current Database System

4.2 Offline Converter Application

After spending few months, the offline converter application was finally completed. Although most of the problems on the application have been solved, more improvements can be done.

4.2.1 Final Result



UTP Pilot Plant Database File Converter

F:\FYP\UTPPilotPlantOCT274.mdb

Tags Tables
- Flow1_Tags
- LevelVol_Tags
- Pressure1_Tags
- TemperatureVol_Tags

Tags Indexes
- CD600FI631
- CD600FIC413.MV
- CD600FIC413.PV
- CD600FIC413.SP
- CD600FIC413S.MV
- CD600FIC413S.PV
- CD600FIC413S.SF
- CD600FIC513.MV
- CD600FIC513.PV
- CD600FIC513.SP
- CD600FIC523.MV
- CD600FIC523.PV

Data Tables
- Flow1
- LevelVol
- Pressure1
- TemperatureVol

Notes:

1. Use ONLY for UTP Pilot Plant Database File.
2. Selected form in MS Access (*.mdb) will be converted to CSV file (*.csv).

Figure 5 Offline Converter Interface

4.2.2 Functionality

i. The converter will convert the Tags Table and Data Tables from MS Access Database to Comma Separated Value (CSV) format *(Refer to Appendix C, Figure 14).*

ii. Only the data from the selected Tags Indexes in the Data Tables will be converted.

iii. After the conversion, values in Data Tables will be rearranged as required.

(Refer to Appendix E, Figure 26 for Offline Application Functionality Flow Chart)

4.2.3 File Conversion

i. Final output (converted file) is in Comma Separated Value format. This file is text based file and can be opened in MS Excel *(Refer to Appendix C, Figure 24)*

ii. The data will be arranged and sorted in the Excel properly.

iii. The original data arrangement can be viewed by using the Note Pad.

iv. Since the original data is in text format, the file size is much smaller than the Excel file.

4.2.4 Error Handler

The converter has several methods in handling error made by user. With proper error handling, user will be alerted whenever error is detected and the program can be executed properly.

i. Error Message "No Tables Selected", only occur when "Convert Tag Tables" button is clicked and no file is selected for conversion or no table is selected in Tag Tables index list.

ii. Error Message "No Tags Indexes Selected", occur when "Convert Data Tables" button is clicked and no tag is selected in tags indexes list is selected or no file is selected for conversion or no table is selected in Tag Tables index list.

iii. Selected table in Tag Tables list will also select the appropriate table in Data Tables list. Vice versa, appropriate table in Tag Tables list will be selected when a table is selected in Data Table list. This is to avoid conversion of Data Table with wrong list of tags indexes since the tags indexes depend on the Tag Table selected.

iv. Ignore Error syntax was included in the VB coding so that the execution continues even if the database contain errors. In some cases, the database file contains errors caused by the system itself. Thus, without the Ignore Error syntax, the application could not perform the conversion when it encounters the error.

4.2.5 Problems/ Limitations

i. The converter cannot be used in Windows XP Operating System (or higher) unless Visual Basic 6 is installed since some components in the application are out dated.

ii. The converter can only be used for UTP Pilot Plant Database.

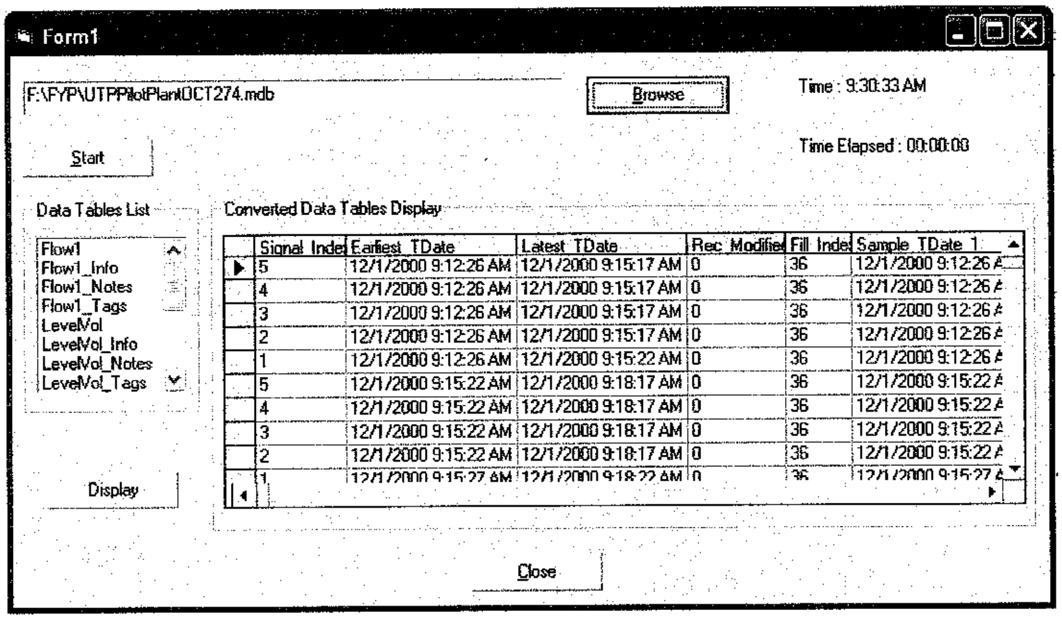
iii. When browsing for a file, canceling the "Open" window will cause error and the application will exit. No coding for solving the problems yet.

iv. UTP logo with file name "UTP_Logo_FYP.bmp" in bitmap format must be included in the same folder. If the file does not exist, the converter will encounter error and fail to start.

4.3 Online Converter Application

Since the attention is given more to offline converter application, online converter could not be completed on time. Many problems were encountered and not much time left before the project ends to finish the application.

4.3.1 Final Result


Form1

F:\FYP\UTPPilotPlantOCT274.mdb

Time : 9:30:33 AM

Time Elapsed : 00:00:00

Data Tables List

Flow1
Flow1_Info
Flow1_Notes
Flow1_Tags
LevelVol
LevelVol_Info
LevelVol_Notes
LevelVol_Tags

Converted Data Tables Display

	Signal Inde	Earliest TDate	Latest TDate	Rec Modifie	Fill Inde	Sample TDate 1
▶	5	12/1/2000 9:12:26 AM	12/1/2000 9:15:17 AM	0	36	12/1/2000 9:12:26 A
	4	12/1/2000 9:12:26 AM	12/1/2000 9:15:17 AM	0	36	12/1/2000 9:12:26 A
	3	12/1/2000 9:12:26 AM	12/1/2000 9:15:17 AM	0	36	12/1/2000 9:12:26 A
	2	12/1/2000 9:12:26 AM	12/1/2000 9:15:17 AM	0	36	12/1/2000 9:12:26 A
	1	12/1/2000 9:12:26 AM	12/1/2000 9:15:22 AM	0	36	12/1/2000 9:12:26 A
	5	12/1/2000 9:15:22 AM	12/1/2000 9:18:17 AM	0	36	12/1/2000 9:15:22 A
	4	12/1/2000 9:15:22 AM	12/1/2000 9:18:17 AM	0	36	12/1/2000 9:15:22 A
	3	12/1/2000 9:15:22 AM	12/1/2000 9:18:17 AM	0	36	12/1/2000 9:15:22 A
	2	12/1/2000 9:15:22 AM	12/1/2000 9:18:17 AM	0	36	12/1/2000 9:15:22 A
	1	12/1/2000 9:15:27 AM	12/1/2000 9:18:22 AM	0	36	12/1/2000 9:15:27 A

Figure 6 Online Converter Interface

Until this report was written, the online converter application could not perform any function yet. Only several parts of the coding were completed and the main coding for conversion has not being implemented yet.

4.3.2 Functionality Expected

i. The conversion can be done online.

ii. User can select data from which instrument or devices he wants to convert.

iii. New table is created for data from each instrument selected.

iv. The conversion will be done periodically. For example, for every 30 minutes.

v. The application will convert a part of the database for each period and continues with the next part after the specified period elapsed.

vi. A table viewer is available for user to view selected converted table in the file.

vii. A report regarding the conversion process and the file summary can be generated.

(Refer to Appendix E, Figure 27 for Online Application Functionality Flow Chart)

4.3.3 Functionality Achieved

i. Until now, the application has a timer and a counter for periodic conversion.

ii. A file browser to browse a file.

iii. A display window to view the converted table in the file.

iv. A Table list to display the available table to be viewed.

4.3.4 Unfinished Tasks

i. Coding for the conversion has not been completed yet.

ii. Coding for linking the Table list and the Table viewer has not been completed.

iii. Function to generate report and report window has not been included.

iv. Content formatting of the output/converted file has not been created.

v. The output/converted file format has not been decided yet whether to use Excel or Access database file.

4.3.5 Implementation Issues

i. The application will be done online on the system server. The conversion may slow the server processing and thus, affecting all other processes in the server such as data acquisition, database storing, data trending and data processing.

ii. The output file format will increase the storage space usage of the server. Although CVS file format is small in size, but for online conversion process, the format is not suitable to be used since it does not have a proper management for multiple data set and it could not handle large data size. MS Access is still the best choice since it can handle and manage medium size of data well but the size of the file has to be considered.

iii. The conversion will be done periodically because continuous conversion will use up more of the server processing capacity and thus slow the server processing time. But, the time period for each conversion must be done correctly since if the application does not manage to complete the conversion in the given time, then the conversion will still be continuous.

4.3.6 Recommendations

i. The application is to be applied remotely on other Workstation. The converter is connected with the main database file through the network. Thus, the workstation will provide the storage space for the converted database file and the conversion does not interrupt the server processing time.

ii. Since the data conversion is done in other workstation, MS Access can be used for data storing since it can manage multiple set of data well.

CHAPTER 5
DISCUSSIONS

This chapter will discuss more on problems encountered, approaches to problems, findings and discussions on overall of the project achievement for each of the problems.

5.1 Solution Development

From the beginning of the project until the end, many problems were encountered as the project progressed. Some of the problems could be solved immediately and some of them still have not been solved. These problems were to be discussed further next.

5.1.1 Laboratory Work

Not much laboratory work could be done since the lab was in used for students' experiments and for other FYP students who have more priority in using the plant for their project. Instead of doing the laboratory work, problems regarding the system were discussed with the technician of the plant. Several other problems related to the database storage, database processing time, data acquisition and trending were identified that are:

i. Slow Processing

For trending and graphing purposes, ICONICS GraphWorx^{TM}32 will obtain the data from the database to plot or display data. System normally utilize one centralize database. Several computer clients are connected to the database and those clients access the database at the same time to obtain the data.

Currently, there are three control rooms with fifteen computer clients. Only three clients can be used at a time to have an acceptable trending and graphing processing speed, since more clients accessing the database will decrease the trending and graphing time since collection of the data from the database is slower.

If the data could be stored in separated file rather in one database file, clients can access the only required data independently from other clients and the processing speed for trending and graphing will become much faster.

ii. Storage

When experiments are done using the pilot plant, the database will collect a huge collection of data and thus it requires a lot of computer storage space to cater the size. One of the issue is the database will collect all the data from all experiments even if some of the experiments fail and the data is not wanted. The unwanted data can not be deleted from the database until all the experiments are done.

Log and historian are normally first-in-first-out (FIFO) basis where old data will be overwritten by new ones A certain disk size is allocated for the data storage.

5.1.2 Solutions Selection

Selection criteria were listed. Based on these criteria, each alternatives listed were weighed and compared from each other. The selection considerations were:

i. The solution can at least solve the main problem.
ii. The solution does not affect the performance of the system.
iii. The solution does not bring other major problems to the system.
iv. The solution cost is reasonable.
v. The solution can be developed within the project timeframe.

Table 1 summarized the advantages and disadvantages of all the available solution alternatives.

Table 1 Advantages and Disadvantages of Available Solution Alternatives

Solutions	Advantages	Disadvantages
VB Offline Converter	• Well known VB programming. Easy to be developed. • Conversion time is short. • Final output size is smaller. • It does not effect the server performance	• Does not improve the database system performances. • Converted file lack of security.
Access Modification	• It does not need second application for conversion. The conversion will be done inside the Access itself.	• Only the modified Access file can be used in the plant. The previous file has to be replaced with the same empty file each time running the plant. • Increase of file size after each conversion. • Quite difficult to be developed.
VB Online Converter	• Well known VB programming. • Conversion can be done directly online.	• May effect the system performances • Does not solve other problems.
MS SQL Server	• Powerful database management system. • May solve all the problems	• Difficult to be developed. High IT expertise required. • High licensing cost.
TrendWorX^{TM}32 Modification	• May solve all the problems. • Maintain the system without any additional applications.	• Need vendor permission for modification. • Difficult to be modified. Intermediate IT expertise required

For the offline solution, the first alternative was selected, which is by developing Visual Basic to convert the data. By using the converter, the data would be converted from Access to Excel file. Advantages of using Access to Excel file converter:

i. Visual Basic programming is used. It is a well-known software and materials and examples can be easily obtained for references.

ii. The conversion time is short.

iii. Excel file is selected as final output because of its compatibility with Matlab and its small file size.

iv. Since it is not depend on the system, the converter can be used anywhere remotely from the server.

Disadvantages of using MS Access to MS Excel file converter:

i. Other problems were not solved since the converter did not affect the system at all.

ii. MS Excel may not be safe and lack of security aspects to be used as data storage. MS Access has better security aspects. It is better to keep the data in MS Access file than replacing it with MS Excel.

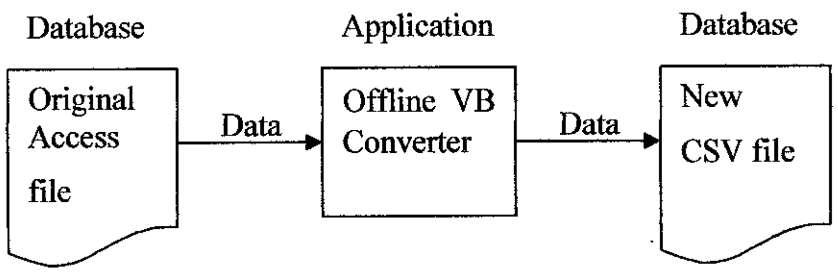


Figure 7 Offline Converter Application Flow Diagram

For the online solution, the last alternative was selected, which is by using other application to be connected to the MS Access for data arrangement and data distribution. This application will also be developed by using Visual Basic 6. The advantages of this solution are:

i. Since Visual Basic programming is also used for this application, the completed offline converter could be taken and extended to implement it online.

ii. The solution was easier compared to others and suitable for the developer's level of expertise. Thus, it was expected to be finished within the timeframe.

iii. Visual Basic has sufficient capability to develop the solution.

iv. Low in cost compared to others since it is tailored personally by the developer and does not require licensing payment.

Nevertheless, this solution is still not the best solution to be implemented. If the timeframe is longer and the level of the developer's expertise is higher, Microsoft SQL Server would be the best choice. The disadvantages of implementing the solution are:

i. The online converter would only solve the main problems.

ii. It is an additional application attached to the system and it may affect the server processing time.

iii. The application development is open ended. Thus, the functionality and reliability of the application is determined by the expertise level of the developer.

iv. Using available DBMS software such as Microsoft SQL Server will be much reliable and efficient.

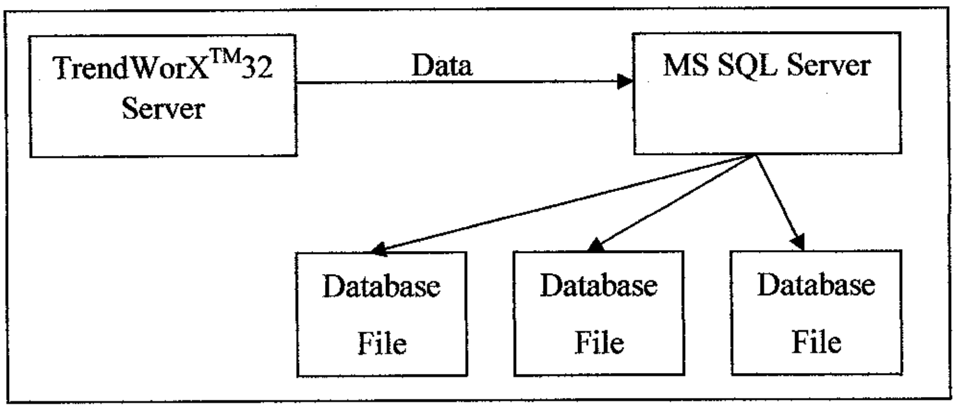


Figure 8 MS SQL Server Implementation

5.1.3 Application Development

5.1.3.1 Offline Solution

The application was developed with Visual Basic 6. At first, the latest version of Visual Basic, Visual Basic.Net was used but the programming language was difficult since it has a lot of changes form the previous version. The development was made stage by stage as follows:

i. <u>Connection</u>

Connection with other database software from VB can be done by several ways, either by using DAO connection or ADO connection. With both methods, the connection may be done by writing coding or linking it to the targeted file by creating Data Environments and select the database for connection.

MS Access is built on Microsoft Jet database engine and DAO data sets in Visual Basic can be used to open Jet database files (which have the extension *.mdb). ADO Data Control is an extension from DAO Data Control where it can handle more type of connections such as Oracle, SQL server, ODBC drivers, DTS packages and more.

Since the conversion only involves MS Access and MS Excel, then DAO Data Control was enough to be used in the programming. By using DAO Data Control, complexity could be avoided. The coding below will allow access to MS Access database through Microsoft Jet Engine and the data inside could be manipulated.

```
Dim db As Database

Dim rs As DAO.Recordset
```

ii. Conversion

During the early stage, the first aim was to make sure that the application can detect the Access file and list the tables inside it before converting the tables to Excel. The arrangement of data still does not have any changes at this stage. The coding below will turn the whole set of the selected table in MS Access to a single sheet in MS Excel.

```
Set objExcl = New Excel.Application

objExcl.Visible = True

objExcl.SheetsInNewWorkbook = 1

objExcl.Workbooks.Add

For I = 0 To rs.Fields.Count - 1

    objExcl.ActiveSheet.Cells(1, I + 1).Value = rs.Fields(I).Name
```

At this stage, the conversion time for the data tables was very long. It took more than half day to convert a table of data because of the size of the data that is too big. This problem might caused the application become unreliable and thus, not a suitable solution to be implemented.

Since the data inside a table is a collection from several inputs, the data could be separated into their respective inputs. Inside the table, the inputs are represented by index numbers. Thus, the conversion could be done by collecting the data only from an index at a time for conversion. This will shorten the conversion time and the data will be converted for an input at a time.

A code has to be written so that can detect the index that was to be converted and only that index was converted at a time. The application must stop when the index number changes. The coding below will search the tables for the desired index [13].

```
Do Until rs.Fields(0) = Ind

    rs.MoveNext

Loop
```

The index resides in the first column, which is denoted by `rs.Fields(0)`. `Ind` is the number of index provided by user. `rs.MoveNext` will tell the application to move from row to row until it finds the desired index before continuing with the conversion. Coding below will tell the application to perform the conversion only for the index number selected.

```
Do While rs.Fields(0) = Ind
```

iii. ***Data Arrangement***

After the connection had been established and the conversion was successfully done, the next aim was how to sort the data in Excel with proper arrangement. The required data to be fetched and rearranged from MS Access were identified. These data were Sample_TDate, Sample_MSec, and Sample_Value as discussed before. Thus, the method of rearranging these data was identified and the coding was implemented.

iv. ***Interface***

At first, the layout is simple and it can convert any of the tables available in the Access file as shown in Figure 9. It has the "browse" button for browsing file, "convert" button for conversion and "close" button to end the application. A text box was used to display the file name, a list box to display the tables available and the index number required by user to convert.

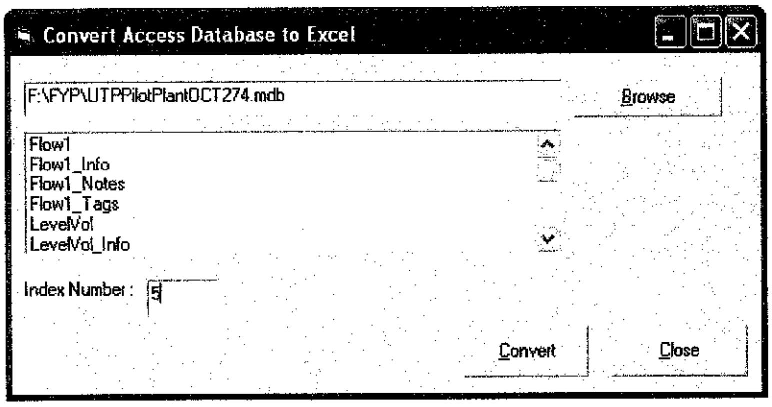


Figure 9
Converter Interface 1

Since in the previous stage all the tables are listed in one list, next stage would filter the tables and divide them to their respective group to be executed and converted differently. At this stage, four lists were made where each list will show the tables by groups that are Tags Tables, Info Tables, Notes Tables and Data Tables. Figure 10 shows the next version of the converter interface.

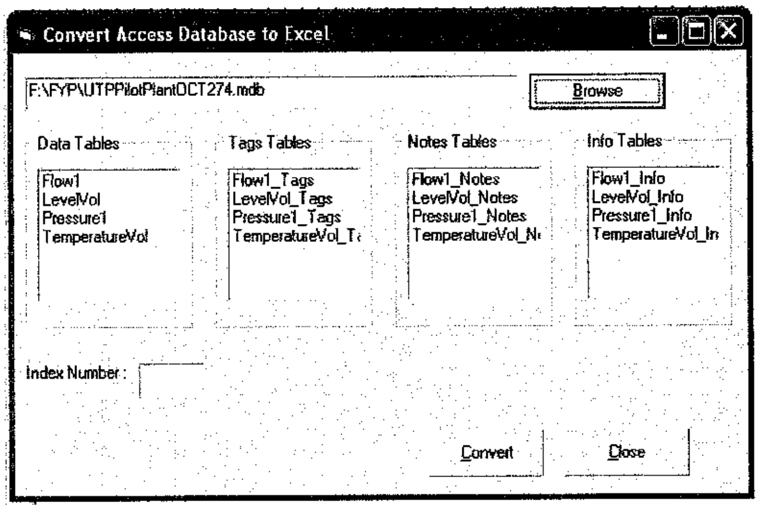


Figure 10
Converter Interface 2

Coding below will tell the list to filter and show only the data tables (which its name did not contain "Tags", "MSys", "Notes", "Info" and "TWX").

```
For Each TD In db.TableDefs
        If Right$(TD.Name, 4) <> "Tags" And _
        Left$(TD.Name, 4) <> "MSys" And _
        Right$(TD.Name, 5) <> "Notes" And _
        Right$(TD.Name, 4) <> "Info" And _
        Left$(TD.Name, 3) <> "TWX" Then List1.AddItem TD.Name
Next TD
```

The next task was to make the application to execute a suitable conversion method for each list. Since only the Data Tables need a new arrangement of data in the Excel file, the other list will perform the regular conversion method without new arrangement of data.

Three alternatives were found to solve the problem that are:

1. To make only a single "Convert" button. When the button is clicked, it will detect which table is selected and convert it. The button must be able to execute different function for respective table as discussed before. This will make the layout of the interface look good and easier usage for user. User will only have to select a table and click only one button to convert. The only drawback is the coding is quite difficult and complex to be done.

2. To provide four buttons for each list. Each button will only perform one function that suits the list. This method is easier but the layout of the application interface looks messy with a lot of buttons.

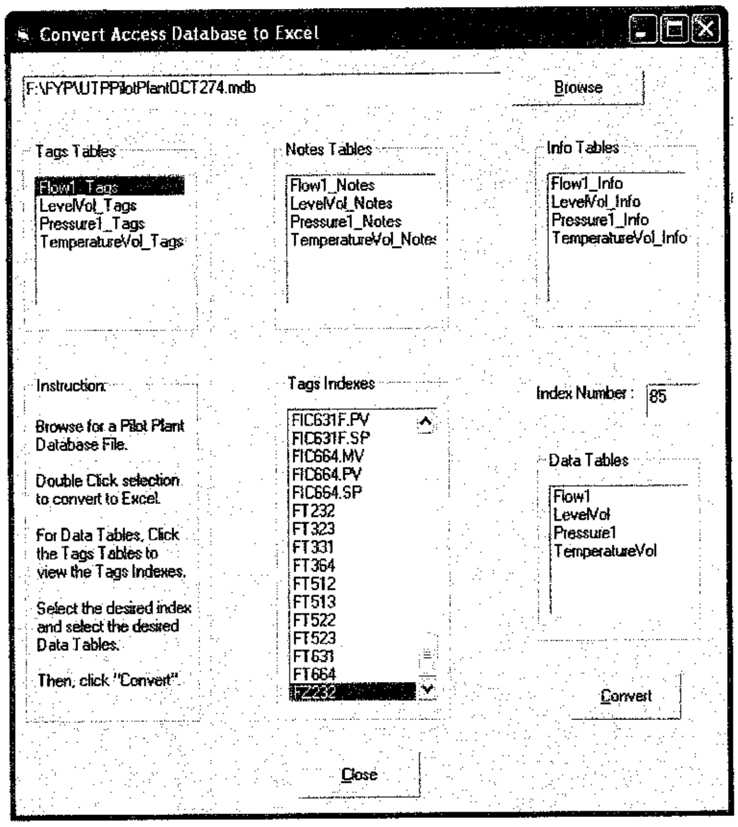
3. To provide a button for data conversion only. Other tables will be converted by other methods such as by double clicking it from the list. This method is the same like the second alternative but with only one button provided. The layout will be better and the coding is simple. The drawback of this alternative is the application will be inconvenient to user since double-clicking is an unfamiliar method to be used by user compared to by clicking a button.

The first alternative was selected because of its nicer layout and easier usage by user. Several problems had occurred. When the user selects the second table from a list, the first table selected was still being selected. When the user click the convert button, an error will occur because of two tables were selected at a time. The same thing happens when the next table was selected. The application needs to refresh after each selection. The refresh session when a new selection was made looks very annoying and uncomfortable to user.

Besides that, the coding for the button to execute a different function for each list was difficult and complex. Two functions have to be created to perform the conversion methods and the button will call the related functions by referring to the selected tables. For each list, a command to relate the list with a function has to be written.

Unfortunately, those coding were failed to be achieved and thus, causing the application failed to perform the desired conversion.

Since the implementation of the first alternative did not work, the third alternative was chosen next. Figure 11 shows the next improvement on the interface. The coding for this method was lengthy but easy to be done. Each list will have its own conversion command. Double-clicking the selected table would execute the conversion command. Since the selected table at each list did not effect other selections, this method avoids error for selecting two or more tables at a time. For Data Table conversion, a special button was provided because user has to type in the required index number first before conversion.



F:\FYP\UTPPilotPlantOCT274.mdb

Tags Tables: Flow1_Tags, LevelVol_Tags, Pressure1_Tags, TemperatureVol_Tags

Notes Tables: Flow1_Notes, LevelVol_Notes, Pressure1_Notes, TemperatureVol_Notes

Info Tables: Flow1_Info, LevelVol_Info, Pressure1_Info, TemperatureVol_Info

Instruction:

Browse for a Pilot Plant Database File.

Double Click selection to convert to Excel.

For Data Tables, Click the Tags Tables to view the Tags Indexes.

Select the desired index and select the desired Data Tables.

Then, click "Convert".

Tags Indexes: FIC631F.PV, FIC631F.SP, FIC664.MV, FIC664.PV, FIC664.SP, FT232, FT323, FT331, FT364, FT512, FT513, FT522, FT523, FT631, FT664, FZ232

Index Number: 85

Data Tables: Flow1, LevelVol, Pressure1, TemperatureVol

Figure 11 Converter Interface

At this stage, another list was created to list the available tags inside the tags tables. From this list, user will know how many and what tags available for conversion and the wanted tags or inputs could be selected to be converted. Coding below will list all the tags inside a tags table selected.

```
Set db = OpenDatabase(txtDbname.Text)
Set rs = db.OpenRecordset("select * from " & List2.Text)

List5.Clear
Do Until rs.EOF
       List5.AddItem rs.Fields(0)
       rs.MoveNext
Loop
```

At this stage, user could convert any table available in the database to Excel file. The application could perform the major task very well. The functionality of the application has to be improved to ease the user. Instructions were included to assist the user on how to use the converter since it quite hard for new user to use the application without proper instructions.

But, problems still occurred with the previous interface layout. When converting any Notes Tables, the application encounters error and terminated because the table cells is empty. Since from the database file Notes Tables are always empty, it was decided that the Notes Tables should be excluded from the selection. Info Tables list was also excluded from the list since it only contains the starting and ending time of the experiment. It was suggested that the information is included directly into the Excel when converting the Data Tables. Thus, there is no need to have separate conversion for the Info Tables.

The next problem was user might convert a wrong Data Tables from the index that has been selected. User has to select a Tags Tables first to list the index and select the respective Data Tables. If user selects different Data Tables, for example, a flow table and temperature tags, the application will still execute the conversion. The user would not obtain the data as desired.

From problems above, some improvement has been made. The Info Tables and Notes Tables were excluded from the application. So, only Data Tables and Tags Tables remained for conversion. For the second problem, when user selects a Tags Tables, the relevant Data Tables will also automatically be selected. This will avoid user making mistakes in selecting different Data Tables from the Tags Tables. The coding below will make sure that the Data Tables related with the Tags Tables will always be selected.

```
If List1.ListIndex = 0 Then List2.ListIndex = 0

If List1.ListIndex = 1 Then List2.ListIndex = 1

If List1.ListIndex = 2 Then List2.ListIndex = 2

If List1.ListIndex = 3 Then List2.ListIndex = 3
```

Since the type of tables to be converted were only Data Tables and Tags Tables, it will be suitable to put one more button for Tags Tables conversion. Thus, the new application will have two buttons, one for each type of tables. This will ease the user since both conversions will be done in the same manner, which is by clicking a button. So, the instruction could be excluded since user will not have any problem using the application without the instruction.

Some notes were included for reminder to anyone who runs the program that the application only used for UTP Pilot Plant Database File and it is used for converting the Access database file to Excel file. If other Access file is selected for conversion, either the applications will encounter error and terminated or the converted file is not in a proper format. Figure 5 in Chapter 4 shows the improved interface after those changes were implemented.

At this stage also, some formatting of the application was done. The title of the application was put at the top including a UTP logo indicating that the application is specially made for used in UTP Pilot Plant. The logo is in *.bmp format and must be in the same folder as the application. If not, then the application will encounter error and terminated.

v. ***Error Handling***

Sometimes, the file contains error and the applications will halt and exit when it encounters the errors. "On Error Resume Next" statement will solve this problem as shown below.

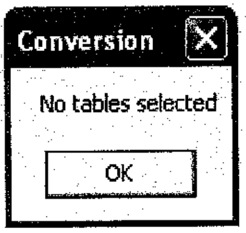
```
Do Until rs.Fields(0) = Ind
    rs.MoveNext
    On Error Resume Next
Loop
```

Evaluation was done as necessary after previous improvement. Some problems still occur. When user clicks the button but without selecting any of the tables or tags listed, the application will encounter error and terminated.

To solve the problem, some coding statements were added to make sure that the application does not exit when it encounters errors. When errors were encountered, it will automatically display a message box telling the user that either tag or table was not selected. Then, user will have to click the OK button before selecting the required item. The statement below will create a message box when the line is executed [14].

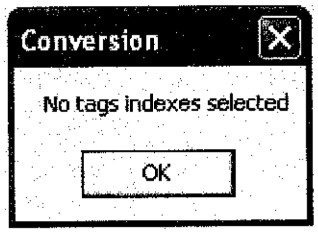
```
MsgBox "No tables selected"
```

Below is the figure of the message boxes that will appear when errors were encountered.


Conversion

No tables selected

Figure 12 Message Box "No tables selected"


Conversion

No tags indexes selected

Figure 13 Message Box "No tags indexes selected"

vi. ***File Formatting***

For a proper informative output, several data was generated and include in the converted file. The file name, date of conversion, date and time of the experiments, table name and tag name were provided to make it a proper formatted document. Coding below would write those information in the converted file.

```
objExcl.ActiveSheet.Cells(1, 1).Value = "File Name : " & Comm.FileTitle
objExcl.ActiveSheet.Cells(2, 1).Value = "Table Name : " & List1.Text
objExcl.ActiveSheet.Cells(3, 1).Value = "Tag Number : " & List5.Text
objExcl.ActiveSheet.Cells(4, 1).Value = "Start Time : " & rb.Fields(1)
objExcl.ActiveSheet.Cells(5, 1).Value = "End Time : " & rb.Fields(2)
```

"`Comm.FileTitle`" refers to the file name, "`List1.Text`" refers to the table name as selected in List1, "`List5.Text`" refers to the tag name as selected in List 5, while "`rb.Fields()`" and "`rb.Fields()`" refers to start time and end time respectively, where both were taken from the respective Info Tables from the selected data table.

vii. ***File Type***

Some comments were given by a lecturer after he tried the software. Instead of converting the file to Excel file, it would be better if to convert the Access file to Comma Separated Value format since the file saved in this format would be much smaller in size.

Comma Separated Value is a very basic, but very popular, technique for exporting data for plain text files. This format just stores one record on a separate line of a plain text file, and separates the values in the fields with commas [15]. Applications such as Microsoft Access and Microsoft Excel can read and convert *.csv files into tables. The coding for *.csv conversion is very much different from the previous conversion to Excel format, *.xls *(Refer to Appendix F, VB Coding: Coding 1).*

After changing the conversion to *.cvs file, some advantages were obtained that are:

a. Faster conversion time

   Conversion form Access to Excel file took about 10 minutes to complete for an index. But for conversion from Access to CVS file took almost immediate time to complete and the longest time for conversion would only take 2 minutes. Thus, great reduce of time were obtained.

b. Smaller file size.

   CVS file size saved after the conversion was only from 200 – 600 kilobytes per file compared to the Excel file size ranging from 400 – 1000 kilobytes per file.

c. Less coding

   Coding for the CVS conversion was much shorter since the conversion was much simpler.

Until this stage, the final output as described in previous chapter was obtained.

5.1.3.2 Online Solution

For the online solutions, the working application has not completed yet. Several errors were encountered and some of the coding was still not found to be implemented. Some requirements for the online solutions are:

A timer. It is needed to let the application performs the conversion at a regular time interval. Instead of converting the online file for each millisecond, it is better for the application to perform the conversion after several minutes to avoid access memory usage of the computer it resides on. A start and stop button are required to activate or deactivate the timer.

1. *To detect the changes of the database file during the plant running.* At some time intervals, the application will convert the data from online Access to

Excel. It must be able to detect the last data that it has converted and start a new conversion from the next data. It also must have a reference where it should stop the conversion.

2. *The conversion method.* The methods will be different from the offline conversion since an offline file have fixed data compared to online file that have an updated data for each milliseconds.

3. *Reports.* Reports of the conversion activity and the file output have to be generated for the activity summary and references to user. With the reports, the user will know whether the application performs the desired function or if there were any error encountered during the conversion.

4. *Outputs file display.* The display will show the output data that has been rearranged and converted in Excel file. From this display, user can observe the data that has been converted from time to time.

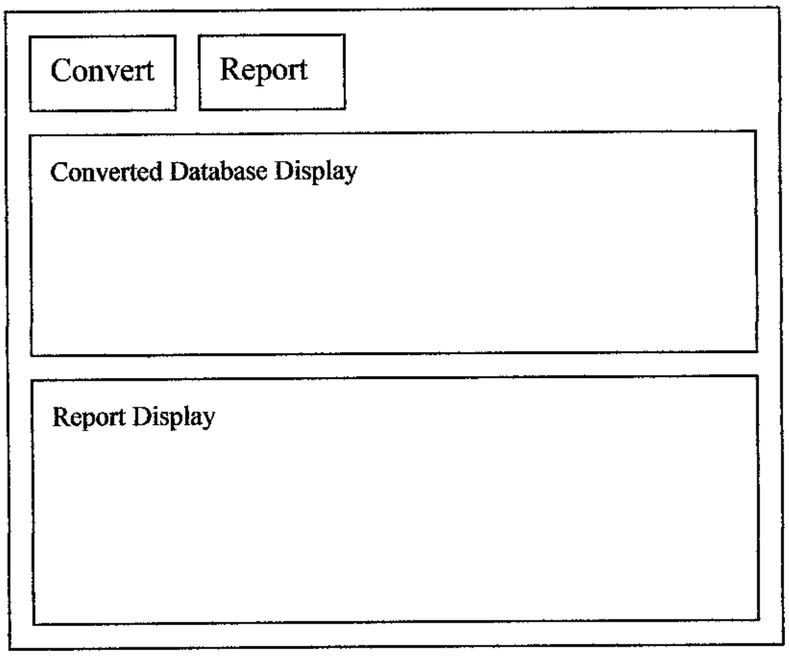


Figure 14   Proposed Online VB Application Interface

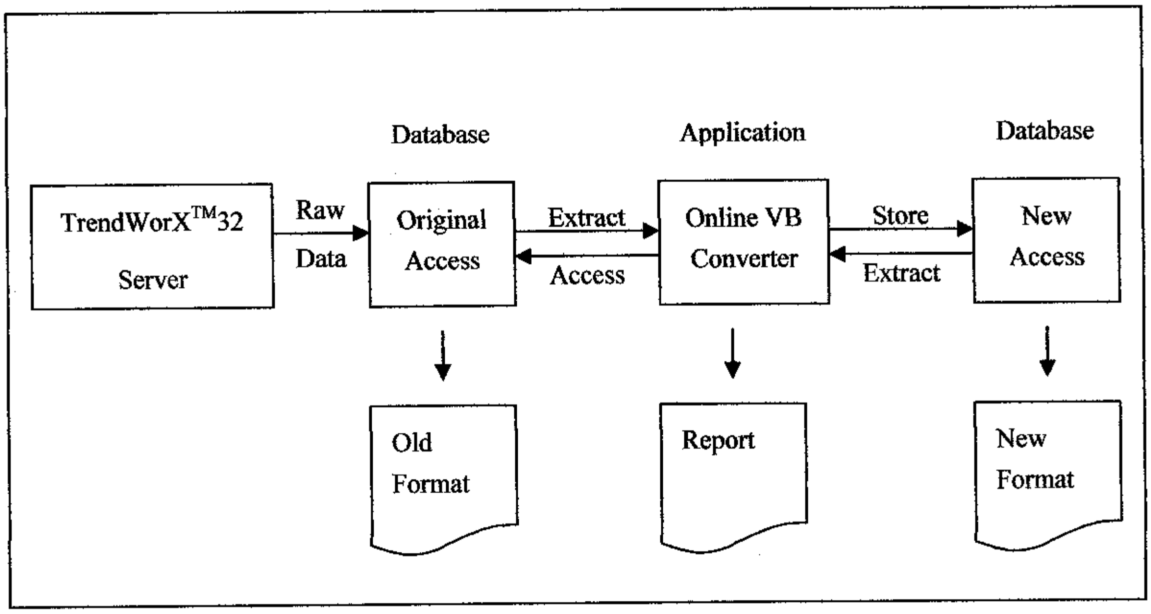


Figure 15 Online Converter Application Flow Diagram

The main concern for the online solutions is the time constraint to develop the application. Approximately three weeks of time remaining might not be enough for completing the application. Even if the application is completed, the application could not be implemented in the plant since the plant was shut down for repair because of some damages. There will be insufficient time for testing and improvement of the application.

The next worry is the online solution might cause the database system to run slower. Even the online solution is different software from the database system, but its interaction with the database file might cause delay in the writing process from the database system to the database file. Additional application to run in the same workstation as the database system might also cause the workstation to operate slower because of memory overload. Since the online solution does not have any improvement on the database system in term of its performances, it is considered that the offline solution is enough to do the conversion.

Still, the assumptions have not been proven yet. Testing and evaluation of the application online are necessary to verify the assumptions but that could only be done when the application is developed completely.

5.2 Project Scheduling/ Planning

On the early stage, offline solutions were expected to be completed during the first semester of the project. But, because of inexperienced in programming using Visual Basic.Net, the application could not even be developed during the first semester. Thus, the application development was started during the second semester by using Visual Basic 6.

Four weeks were given to finish the offline solution and the rest of the semester has been planned to finish the online solution. The offline solution managed to be completed within the planned scheduled. Since the offline solution was seen promising and could be improved more, more time was given to make the offline solution better.

For the online solutions, problem occurred when the plant was down and could not be used. Thus, integrating the software to the database could not be done in the plant. Many other critical issues related to the database system from implementing the online solution arose as discussed in previous section making the online solution development halted.

Thus, more focuses were given to improving the offline solution. The offline solution managed to be completed successfully and it could be used by the UTP Pilot Plant database user after this. The online solution has to be reviewed again whether its implementation is required or not.

CHAPTER 6
CONCLUSION AND RECOMMENDATIONS

From the previous discussions, this chapter will conclude and highlight the main points. Recommendations on further development of the project also will be discussed.

6.1 Conclusion

From previous discussion, we can conclude that:

1. The offline solution was developed successfully. Even though more improvements can be done, the converter can perform the desired tasks well.
2. More improvement can be done to offline solution and its functionality could be extended more than merely a converter.
3. The online solution could not be completed because of more focus is given in improving the offline solution.
4. The online solution has to be revised back whether to continue or change with other solution regarding the issues arisen from implementation of the solution.
5. The main problem was successfully solved and thus, the project main objective was successfully achieved.

6.2 Recommendations

1. *The offline converter functionality should be extended so it can be used for any type of database.*

   The converter could be programmed to convert any database to various types of format and with the desired arrangement of data for any purposes, not just for UTP Pilot Plant. Thus, it will increase functionality of the application, more reliable, and increase the commercial value.

   The converter conversion method can be extended with capability of putting several indexes in one file. This is for the purpose of importing multiple sets of data into the MATLAB instead of only one index from current conversion method.

2. *The offline converter could be linked to appropriate software.*

   Instead of just converting the data to a file, the converter may export the data directly to any appropriate application such as Matlab, MS Access, MS Word, Web Page or any others. This additional function will help users more than just converting the database.

3. *Students continuing this project can just focus on the online solution.*

   With current findings, it may help the student a lot to continue this project. Better solution may be developed and implemented online in the future.

4. *This project could be jointly done with Information Technology student.*

   As IT students, they have been exposed well to database and application development. They might help a lot on expertise and knowledge of database development and programming. EE students can give more focus on the development of the application instead of spending more time in learning about programming and database.

REFERENCES

[1] Juan R. Pimentel, Mario Salazar. Dependability of Distributed Control System Fault Tolerant Units, Page 1.

[2] http://en.wikipedia.org/w/wiki.phtml?title=Distributed_Control_System

[3] Jim Pinto. *Fieldbus - Conflicting "Standards" Emerge, but interoperability is Still Elusive,* 31 March 2000, San Diego, CA. USA

[4] http://www.hartcomm.org/technical/applications/network/compnet.html

[5] ABB Automation Technologies. Industrial *System 800xA Operations Overview.* 2005

[6] Honeywell PlantScape® Overview, Honeywell Inc. Release 320 – July 2000.

[7] ICONICS TrendWorX™ 32, OPC-Based Trending and Historical Client, Reference Guide, Release 6, ©ICONICS, Inc., 2000,

[8] *Database System Concepts*, 4th Edition, Abraham Silberschatz, Henry F. Korth, S. Sudarshan, Mc Graw-Hill Companies, Inc., 2002.

[9] *Learning Visual Basic.Net Through Applications*, First Edition, Clayton E. Crooks II, Charles River Media Inc., 2003.

[10] *Professional VB.Net 2003*, First Edition, I. Evjen, Bill, Wiley Publishing, Inc., 2004.

[11] 10 Projects You Can Do with Microsoft® SQL Server™ 7, First Edition, Karen Watterson, Bill Shadish and Garth Wells, John Wiley & Sons Inc., 2000.

[12] *Database Programming with Visual Basic 6,* Evangelos Petroutsos, First Edition, Sybex Inc. 2000.

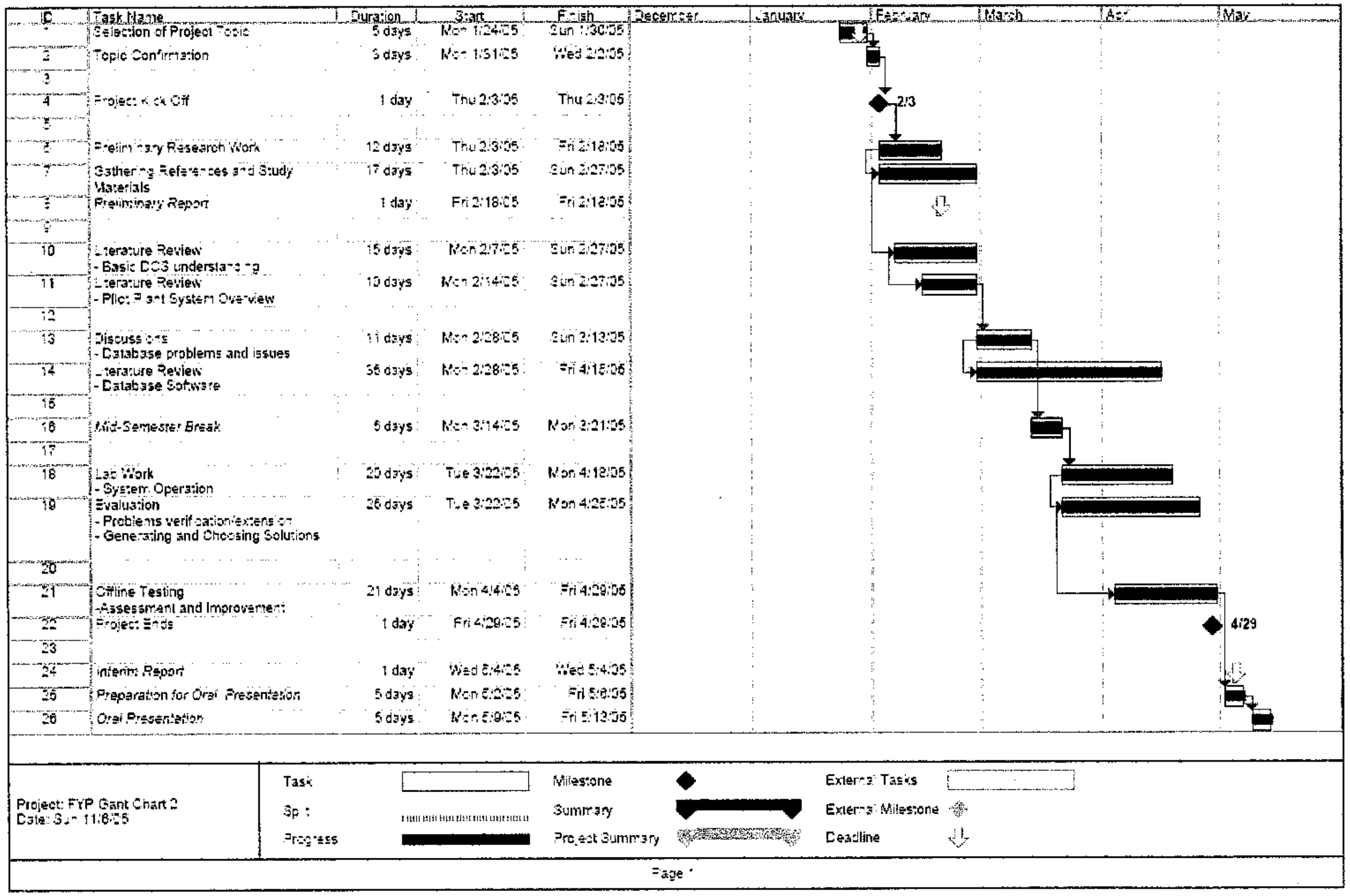
[13] *Visual Basic from the Ground up,* Gary Cornell, First Edition, McGraw-Hill, 1998.

[14] *Visual Basic 6 Unleashed, Professional Reference Edition,* Rob Thayer, et al., First Edition, Sam Publishing, 1999.

[15] *ADO Programming in Visual Basic 6,* Steven Holzner, First Edition, Prentice Hall, 1999.

APPENDICES

APPENDIX A GANT CHART



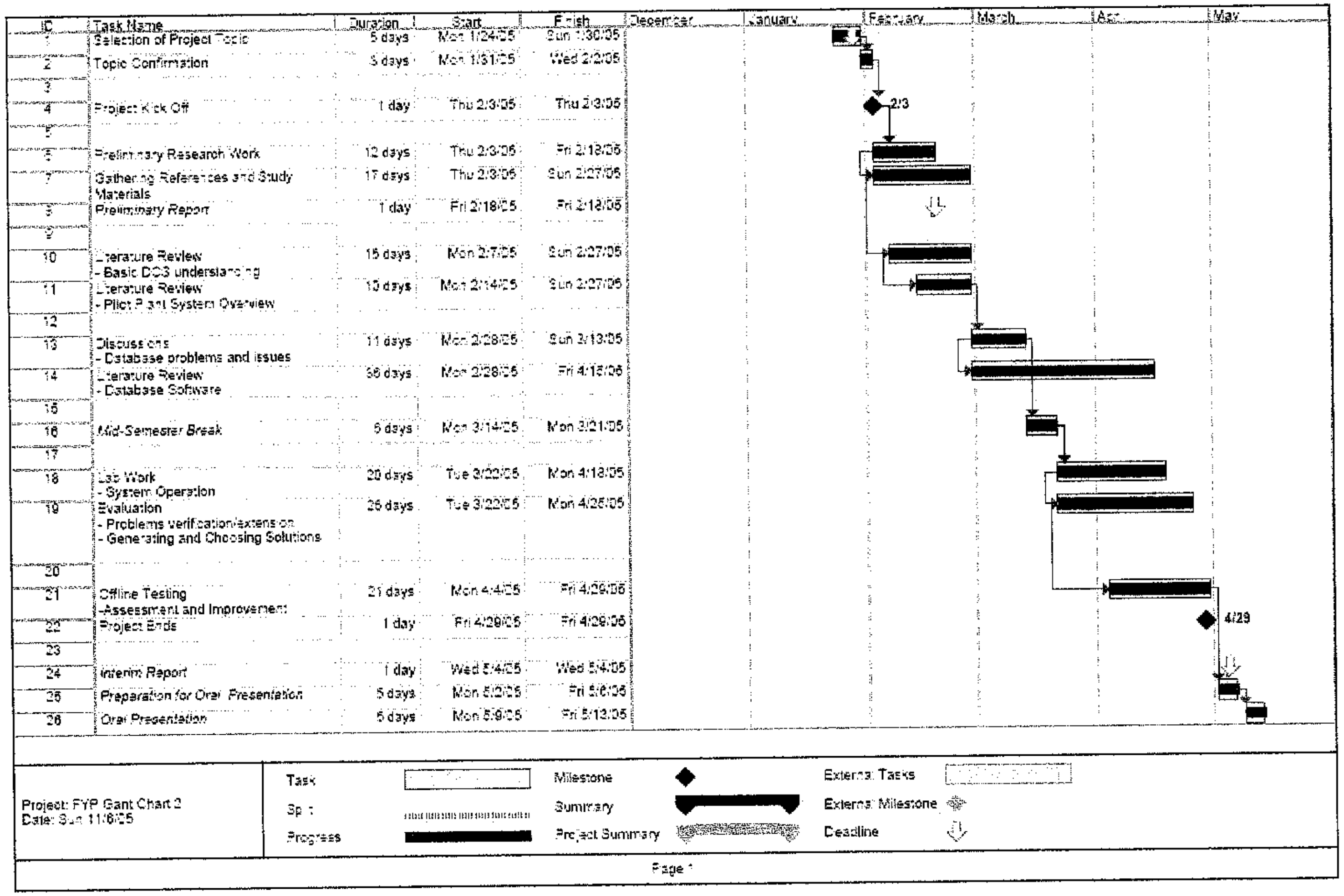
ID	Task Name	Duration	Start	Finish
1	Selection of Project Topic	5 days	Mon 1/24/05	Sun 1/30/05
2	Topic Confirmation	3 days	Mon 1/31/05	Wed 2/2/05
3				
4	Project Kick Off	1 day	Thu 2/3/05	Thu 2/3/05
5				
6	Preliminary Research Work	12 days	Thu 2/3/05	Fri 2/18/05
7	Gathering References and Study Materials	17 days	Thu 2/3/05	Sun 2/27/05
8	*Preliminary Report*	1 day	Fri 2/18/05	Fri 2/18/05
9				
10	Literature Review - Basic DCS understanding	15 days	Mon 2/7/05	Sun 2/27/05
11	Literature Review - Pilot Plant System Overview	10 days	Mon 2/14/05	Sun 2/27/05
12				
13	Discussions - Database problems and issues	11 days	Mon 2/28/05	Sun 3/13/05
14	Literature Review - Database Software	35 days	Mon 2/28/05	Fri 4/15/05
15				
16	*Mid-Semester Break*	5 days	Mon 3/14/05	Mon 3/21/05
17				
18	Lab Work - System Operation	20 days	Tue 3/22/05	Mon 4/18/05
19	Evaluation - Problems verification/extension - Generating and Choosing Solutions	26 days	Tue 3/22/05	Mon 4/25/05
20				
21	Offline Testing -Assessment and Improvement	21 days	Mon 4/4/05	Fri 4/29/05
22	Project Ends	1 day	Fri 4/29/05	Fri 4/29/05
23				
24	*Interim Report*	1 day	Wed 5/4/05	Wed 5/4/05
25	*Preparation for Oral Presentation*	5 days	Mon 5/2/05	Fri 5/6/05
26	*Oral Presentation*	5 days	Mon 5/9/05	Fri 5/13/05

Project: FYP Gant Chart 2
Date: Sun 11/6/05

Task	☐	Milestone	◆	External Tasks	☐
Split	┈	Summary	▬	External Milestone	◈
Progress	■	Project Summary	▨	Deadline	⇩

Page 1

Figure 16 Project Planner for 1st Semester



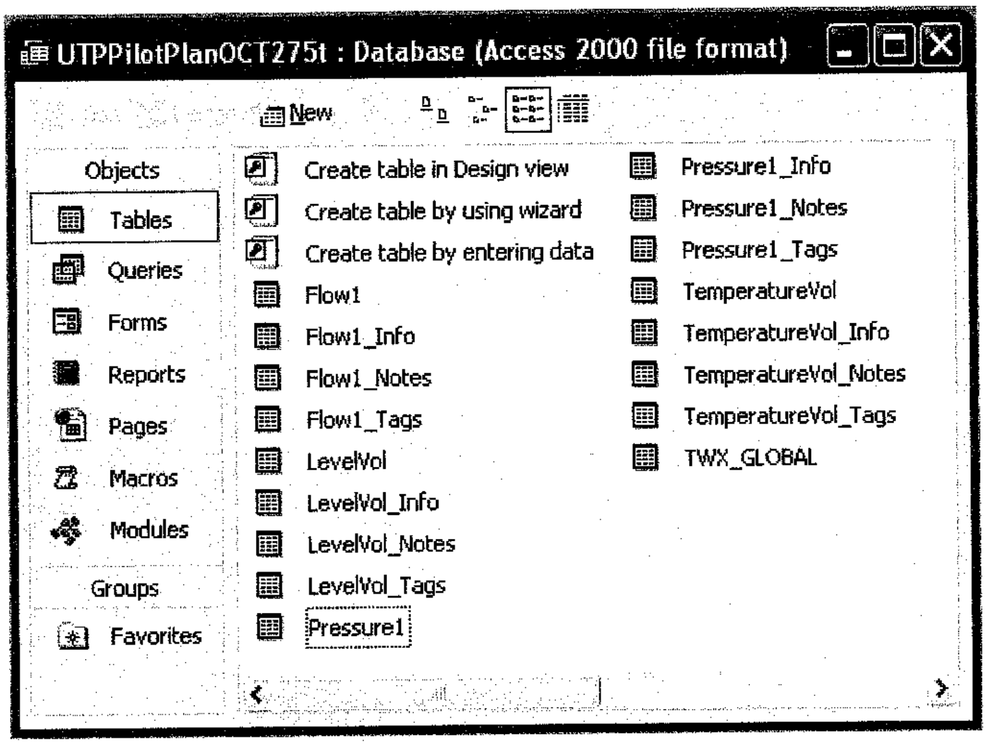
ID	Task Name	Duration	Start	Finish
1	Selection of Project Topic	5 days	Mon 1/24/05	Sun 1/30/05
2	Topic Confirmation	3 days	Mon 1/31/05	Wed 2/2/05
3				
4	Project Kick Off	1 day	Thu 2/3/05	Thu 2/3/05
5				
6	Preliminary Research Work	12 days	Thu 2/3/05	Fri 2/18/05
7	Gathering References and Study Materials	17 days	Thu 2/3/05	Sun 2/27/05
8	*Preliminary Report*	1 day	Fri 2/18/05	Fri 2/18/05
9				
10	Literature Review - Basic DCS understanding	15 days	Mon 2/7/05	Sun 2/27/05
11	Literature Review - Pilot Plant System Overview	10 days	Mon 2/14/05	Sun 2/27/05
12				
13	Discussions - Database problems and issues	11 days	Mon 2/28/05	Sun 3/13/05
14	Literature Review - Database Software	35 days	Mon 2/28/05	Fri 4/15/05
15				
16	*Mid-Semester Break*	6 days	Mon 3/14/05	Mon 3/21/05
17				
18	Lab Work - System Operation	20 days	Tue 3/22/05	Mon 4/18/05
19	Evaluation - Problems verification/extension - Generating and Choosing Solutions	25 days	Tue 3/22/05	Mon 4/25/05
20				
21	Offline Testing -Assessment and Improvement	21 days	Mon 4/4/05	Fri 4/29/05
22	Project Ends	1 day	Fri 4/29/05	Fri 4/29/05
23				
24	*Interim Report*	1 day	Wed 5/4/05	Wed 5/4/05
25	*Preparation for Oral Presentation*	5 days	Mon 5/2/05	Fri 5/6/05
26	*Oral Presentation*	5 days	Mon 5/9/05	Fri 5/13/05

Project: FYP Gant Chart 2
Date: Sun 11/6/05

Page 1

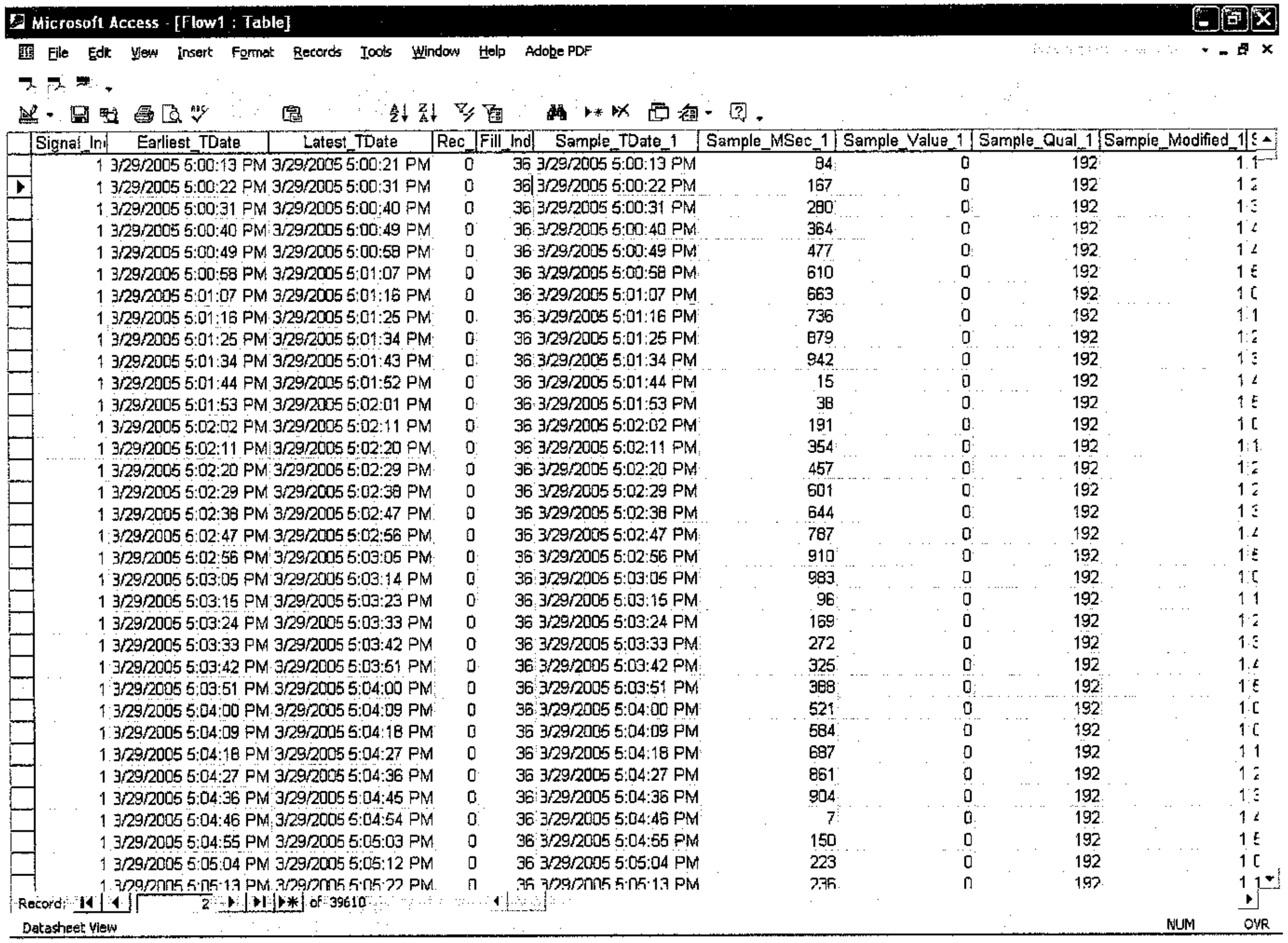
Figure 17 Project Planner for 2nd Semester

APPENDIX B UTP PILOT PLANT DATABASE



Create table in Design view
Create table by using wizard
Create table by entering data
Flow1
Flow1_Info
Flow1_Notes
Flow1_Tags
LevelVol
LevelVol_Info
LevelVol_Notes
LevelVol_Tags
Pressure1
Pressure1_Info
Pressure1_Notes
Pressure1_Tags
TemperatureVol
TemperatureVol_Info
TemperatureVol_Notes
TemperatureVol_Tags
TWX_GLOBAL

Figure 18 UTP Pilot Plant Database – Table list



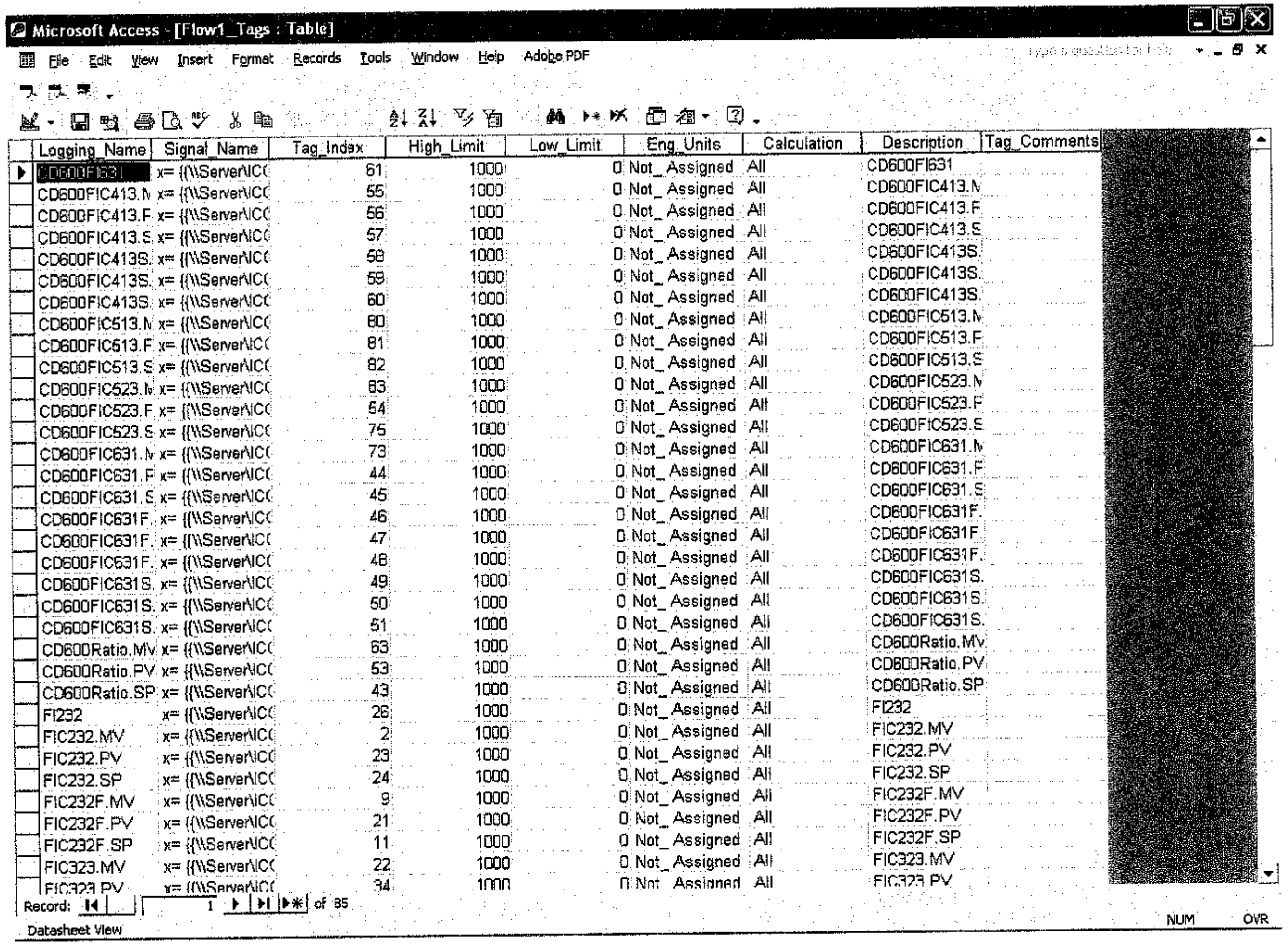
Microsoft Access - [Flow1 : Table]

Signal_Ini	Earliest_TDate	Latest_TDate	Rec	Fill_Ind	Sample_TDate_1	Sample_MSec_1	Sample_Value_1	Sample_Qual_1	Sample_Modified_1
1	3/29/2005 5:00:13 PM	3/29/2005 5:00:21 PM	0	36	3/29/2005 5:00:13 PM	84	0	192	1 1
1	3/29/2005 5:00:22 PM	3/29/2005 5:00:31 PM	0	36	3/29/2005 5:00:22 PM	167	0	192	1 2
1	3/29/2005 5:00:31 PM	3/29/2005 5:00:40 PM	0	36	3/29/2005 5:00:31 PM	280	0	192	1 3
1	3/29/2005 5:00:40 PM	3/29/2005 5:00:49 PM	0	36	3/29/2005 5:00:40 PM	364	0	192	1 4
1	3/29/2005 5:00:49 PM	3/29/2005 5:00:58 PM	0	36	3/29/2005 5:00:49 PM	477	0	192	1 4
1	3/29/2005 5:00:58 PM	3/29/2005 5:01:07 PM	0	36	3/29/2005 5:00:58 PM	610	0	192	1 5
1	3/29/2005 5:01:07 PM	3/29/2005 5:01:16 PM	0	36	3/29/2005 5:01:07 PM	663	0	192	1 0
1	3/29/2005 5:01:16 PM	3/29/2005 5:01:25 PM	0	36	3/29/2005 5:01:16 PM	736	0	192	1 1
1	3/29/2005 5:01:25 PM	3/29/2005 5:01:34 PM	0	36	3/29/2005 5:01:25 PM	879	0	192	1 2
1	3/29/2005 5:01:34 PM	3/29/2005 5:01:43 PM	0	36	3/29/2005 5:01:34 PM	942	0	192	1 3
1	3/29/2005 5:01:44 PM	3/29/2005 5:01:52 PM	0	36	3/29/2005 5:01:44 PM	15	0	192	1 4
1	3/29/2005 5:01:53 PM	3/29/2005 5:02:01 PM	0	36	3/29/2005 5:01:53 PM	38	0	192	1 5
1	3/29/2005 5:02:02 PM	3/29/2005 5:02:11 PM	0	36	3/29/2005 5:02:02 PM	191	0	192	1 0
1	3/29/2005 5:02:11 PM	3/29/2005 5:02:20 PM	0	36	3/29/2005 5:02:11 PM	354	0	192	1 1
1	3/29/2005 5:02:20 PM	3/29/2005 5:02:29 PM	0	36	3/29/2005 5:02:20 PM	457	0	192	1 2
1	3/29/2005 5:02:29 PM	3/29/2005 5:02:38 PM	0	36	3/29/2005 5:02:29 PM	601	0	192	1 2
1	3/29/2005 5:02:38 PM	3/29/2005 5:02:47 PM	0	36	3/29/2005 5:02:38 PM	644	0	192	1 3
1	3/29/2005 5:02:47 PM	3/29/2005 5:02:56 PM	0	36	3/29/2005 5:02:47 PM	787	0	192	1 4
1	3/29/2005 5:02:56 PM	3/29/2005 5:03:05 PM	0	36	3/29/2005 5:02:56 PM	910	0	192	1 5
1	3/29/2005 5:03:05 PM	3/29/2005 5:03:14 PM	0	36	3/29/2005 5:03:05 PM	983	0	192	1 0
1	3/29/2005 5:03:15 PM	3/29/2005 5:03:23 PM	0	36	3/29/2005 5:03:15 PM	96	0	192	1 1
1	3/29/2005 5:03:24 PM	3/29/2005 5:03:33 PM	0	36	3/29/2005 5:03:24 PM	169	0	192	1 2
1	3/29/2005 5:03:33 PM	3/29/2005 5:03:42 PM	0	36	3/29/2005 5:03:33 PM	272	0	192	1 3
1	3/29/2005 5:03:42 PM	3/29/2005 5:03:51 PM	0	36	3/29/2005 5:03:42 PM	325	0	192	1 4
1	3/29/2005 5:03:51 PM	3/29/2005 5:04:00 PM	0	36	3/29/2005 5:03:51 PM	368	0	192	1 5
1	3/29/2005 5:04:00 PM	3/29/2005 5:04:09 PM	0	36	3/29/2005 5:04:00 PM	521	0	192	1 0
1	3/29/2005 5:04:09 PM	3/29/2005 5:04:18 PM	0	36	3/29/2005 5:04:09 PM	584	0	192	1 0
1	3/29/2005 5:04:18 PM	3/29/2005 5:04:27 PM	0	36	3/29/2005 5:04:18 PM	687	0	192	1 1
1	3/29/2005 5:04:27 PM	3/29/2005 5:04:36 PM	0	36	3/29/2005 5:04:27 PM	861	0	192	1 2
1	3/29/2005 5:04:36 PM	3/29/2005 5:04:45 PM	0	36	3/29/2005 5:04:36 PM	904	0	192	1 3
1	3/29/2005 5:04:46 PM	3/29/2005 5:04:54 PM	0	36	3/29/2005 5:04:46 PM	7	0	192	1 4
1	3/29/2005 5:04:55 PM	3/29/2005 5:05:03 PM	0	36	3/29/2005 5:04:55 PM	150	0	192	1 5
1	3/29/2005 5:05:04 PM	3/29/2005 5:05:12 PM	0	36	3/29/2005 5:05:04 PM	223	0	192	1 0
1	3/29/2005 5:05:13 PM	3/29/2005 5:05:22 PM	0	36	3/29/2005 5:05:13 PM	236	0	192	1 1

Record: 2 of 39610

Datasheet View

Figure 19 UTP Pilot Plant Database File in MS Access – "Flow1" Table

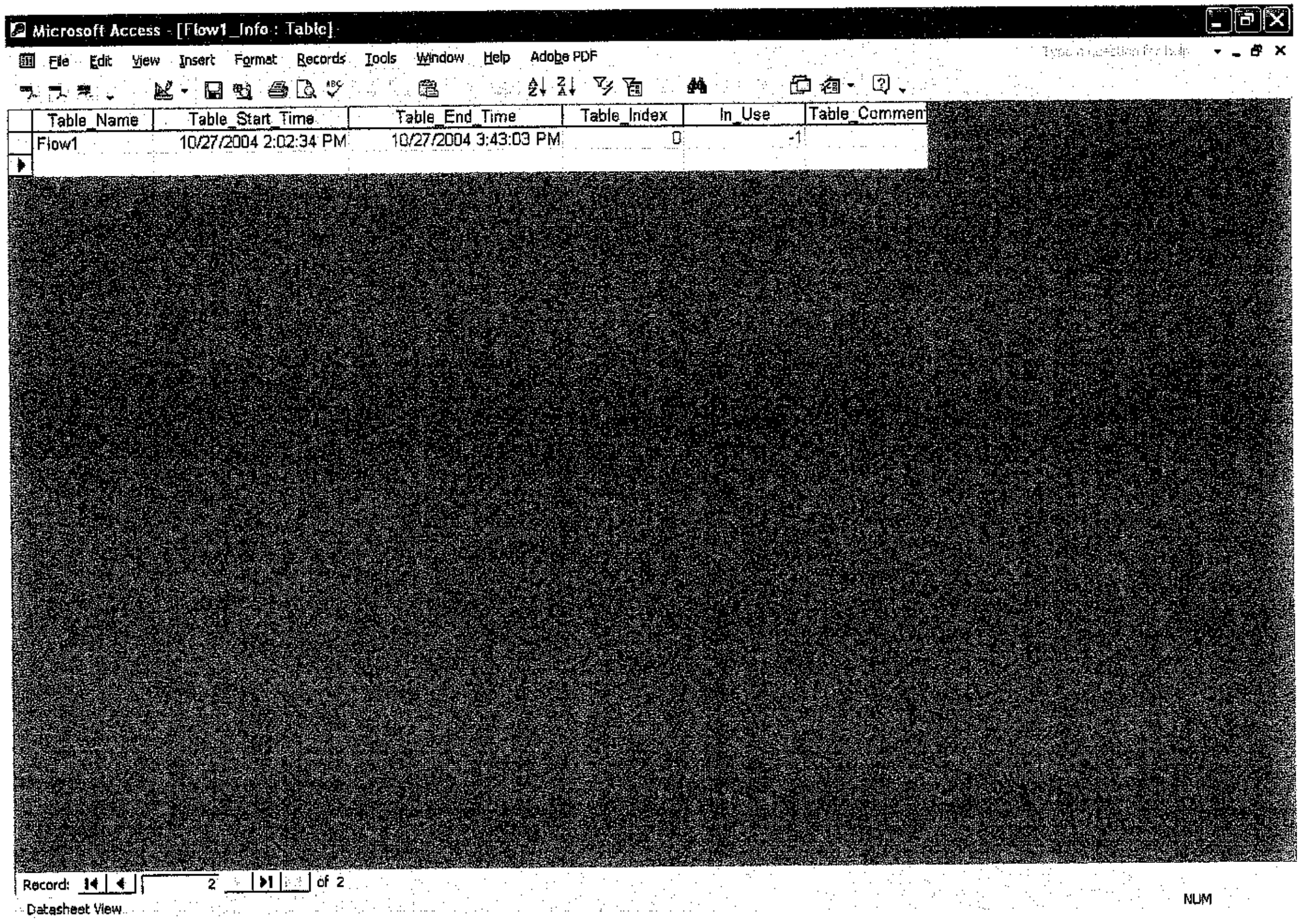

Microsoft Access - [Flow1_Tags : Table]

Logging_Name	Signal_Name	Tag_Index	High_Limit	Low_Limit	Eng_Units	Calculation	Description	Tag_Comments
CD600FI631	x= {{\\Server\IC(	61	1000	0	Not_Assigned	All	CD600FI631	
CD600FIC413.N	x= {{\\Server\IC(	55	1000	0	Not_Assigned	All	CD600FIC413.N	
CD600FIC413.F	x= {{\\Server\IC(	56	1000	0	Not_Assigned	All	CD600FIC413.F	
CD600FIC413.S	x= {{\\Server\IC(	57	1000	0	Not_Assigned	All	CD600FIC413.S	
CD600FIC413S.	x= {{\\Server\IC(	58	1000	0	Not_Assigned	All	CD600FIC413S.	
CD600FIC413S.	x= {{\\Server\IC(	59	1000	0	Not_Assigned	All	CD600FIC413S.	
CD600FIC413S.	x= {{\\Server\IC(	60	1000	0	Not_Assigned	All	CD600FIC413S.	
CD600FIC513.N	x= {{\\Server\IC(	80	1000	0	Not_Assigned	All	CD600FIC513.N	
CD600FIC513.F	x= {{\\Server\IC(	81	1000	0	Not_Assigned	All	CD600FIC513.F	
CD600FIC513.S	x= {{\\Server\IC(	82	1000	0	Not_Assigned	All	CD600FIC513.S	
CD600FIC523.N	x= {{\\Server\IC(	83	1000	0	Not_Assigned	All	CD600FIC523.N	
CD600FIC523.F	x= {{\\Server\IC(	54	1000	0	Not_Assigned	All	CD600FIC523.F	
CD600FIC523.S	x= {{\\Server\IC(	75	1000	0	Not_Assigned	All	CD600FIC523.S	
CD600FIC631.N	x= {{\\Server\IC(	73	1000	0	Not_Assigned	All	CD600FIC631.N	
CD600FIC631.F	x= {{\\Server\IC(	44	1000	0	Not_Assigned	All	CD600FIC631.F	
CD600FIC631.S	x= {{\\Server\IC(	45	1000	0	Not_Assigned	All	CD600FIC631.S	
CD600FIC631F.	x= {{\\Server\IC(	46	1000	0	Not_Assigned	All	CD600FIC631F.	
CD600FIC631F.	x= {{\\Server\IC(	47	1000	0	Not_Assigned	All	CD600FIC631F.	
CD600FIC631F.	x= {{\\Server\IC(	48	1000	0	Not_Assigned	All	CD600FIC631F.	
CD600FIC631S.	x= {{\\Server\IC(	49	1000	0	Not_Assigned	All	CD600FIC631S.	
CD600FIC631S.	x= {{\\Server\IC(	50	1000	0	Not_Assigned	All	CD600FIC631S.	
CD600FIC631S.	x= {{\\Server\IC(	51	1000	0	Not_Assigned	All	CD600FIC631S.	
CD600Ratio.MV	x= {{\\Server\IC(	63	1000	0	Not_Assigned	All	CD600Ratio.MV	
CD600Ratio.PV	x= {{\\Server\IC(	53	1000	0	Not_Assigned	All	CD600Ratio.PV	
CD600Ratio.SP	x= {{\\Server\IC(	43	1000	0	Not_Assigned	All	CD600Ratio.SP	
FI232	x= {{\\Server\IC(	26	1000	0	Not_Assigned	All	FI232	
FIC232.MV	x= {{\\Server\IC(	2	1000	0	Not_Assigned	All	FIC232.MV	
FIC232.PV	x= {{\\Server\IC(	23	1000	0	Not_Assigned	All	FIC232.PV	
FIC232.SP	x= {{\\Server\IC(	24	1000	0	Not_Assigned	All	FIC232.SP	
FIC232F.MV	x= {{\\Server\IC(	9	1000	0	Not_Assigned	All	FIC232F.MV	
FIC232F.PV	x= {{\\Server\IC(	21	1000	0	Not_Assigned	All	FIC232F.PV	
FIC232F.SP	x= {{\\Server\IC(	11	1000	0	Not_Assigned	All	FIC232F.SP	
FIC323.MV	x= {{\\Server\IC(	22	1000	0	Not_Assigned	All	FIC323.MV	
FIC323.PV	x= {{\\Server\IC(	34	1000	0	Not_Assigned	All	FIC323.PV	

Record: 1 of 85

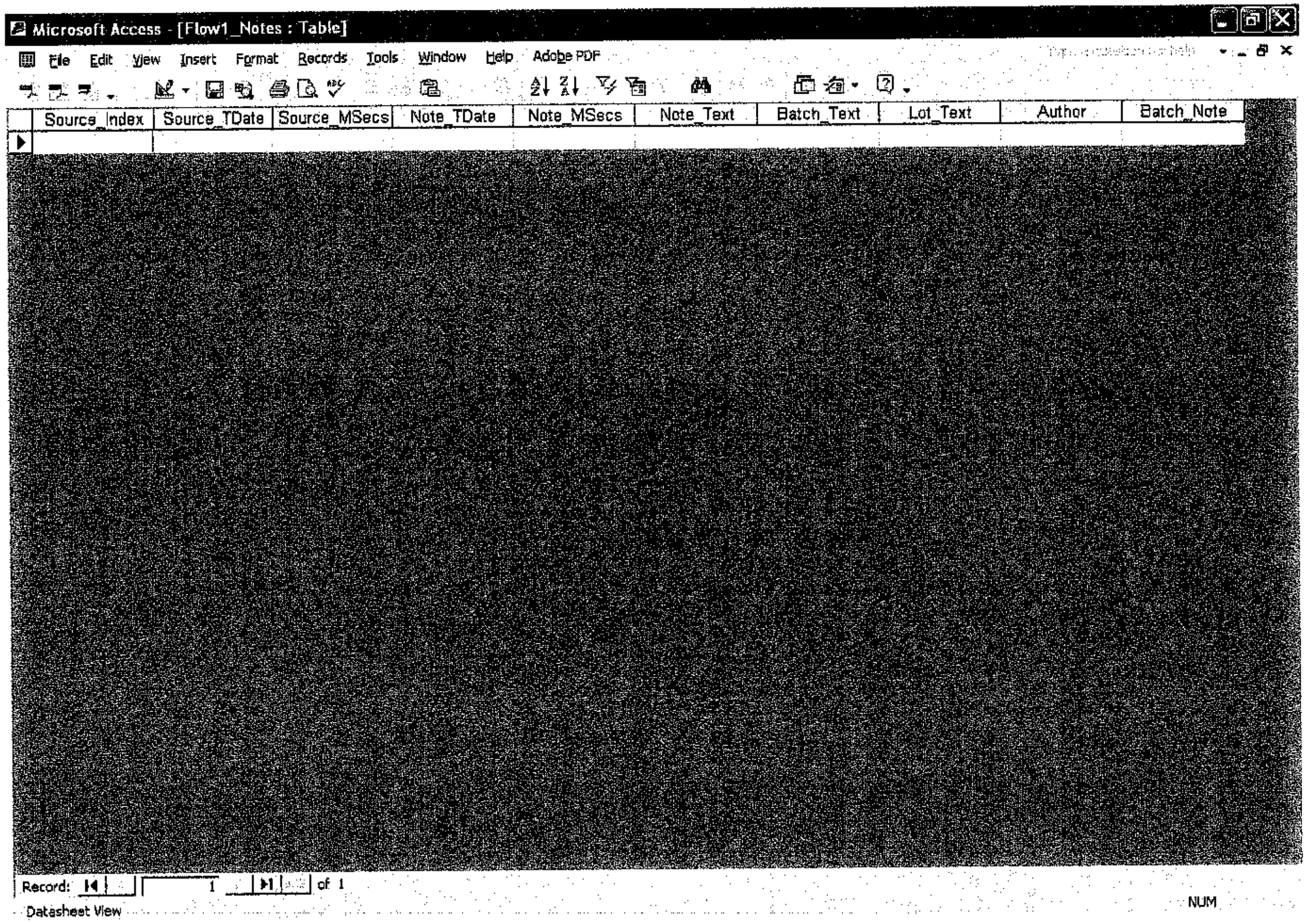
Datasheet View

Figure 20 UTP Pilot Plant Database File in MS Access – "Flow1_Tags" Table


Microsoft Access - [Flow1_Info : Table]
File Edit View Insert Format Records Tools Window Help Adobe PDF
Record: 2 of 2
Datasheet View
NUM

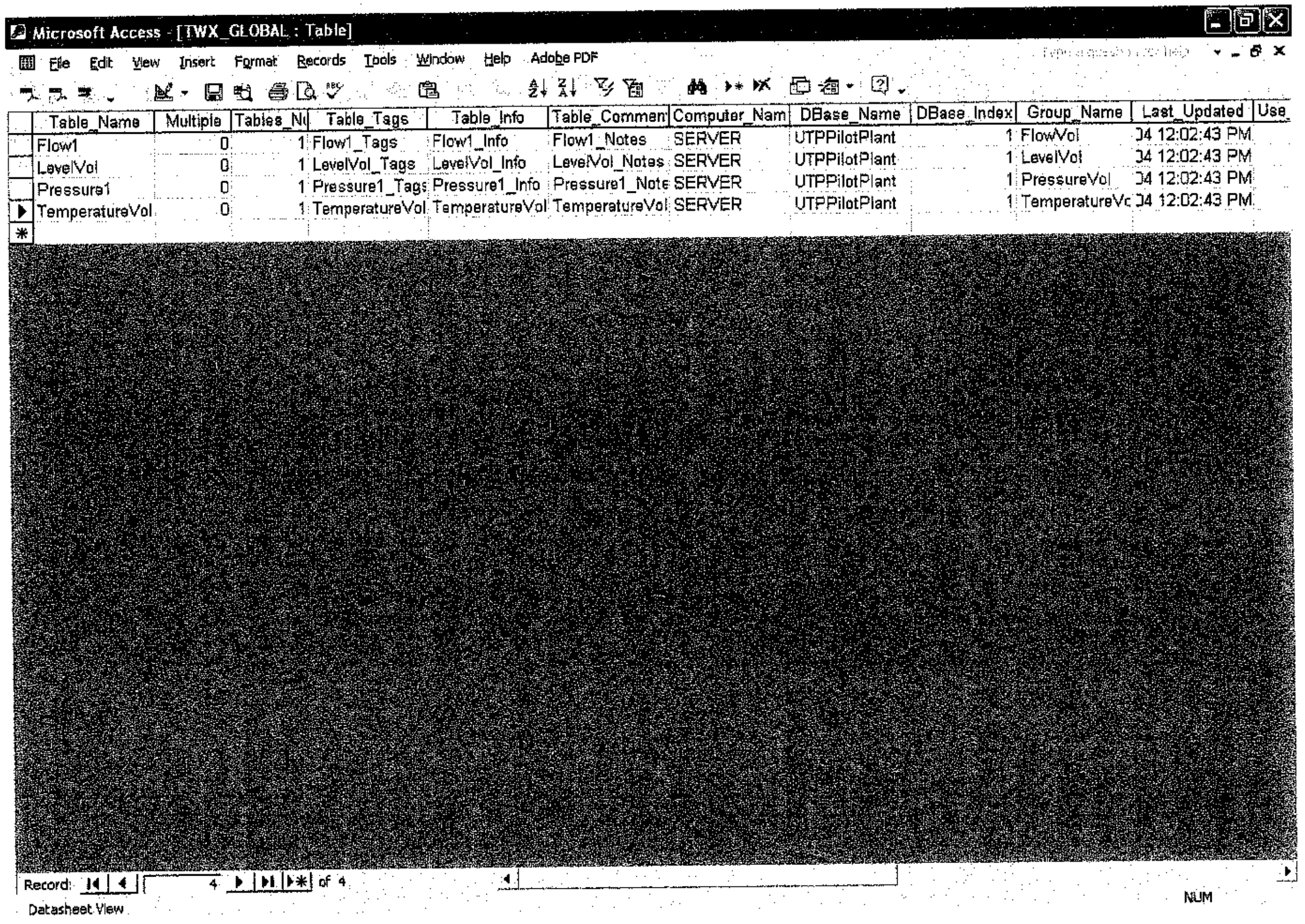
Table_Name	Table_Start_Time	Table_End_Time	Table_Index	In_Use	Table_Commen
Flow1	10/27/2004 2:02:34 PM	10/27/2004 3:43:03 PM	0	-1	

Figure 21 UTP Pilot Plant Database File in MS Access – "Flow1_Info" Table


Microsoft Access - [Flow1_Notes : Table]
File Edit View Insert Format Records Tools Window Help Adobe PDF
Record: 1 of 1
Datasheet View
NUM

Source_Index	Source_TDate	Source_MSecs	Note_TDate	Note_MSecs	Note_Text	Batch_Text	Lot_Text	Author	Batch_Note

Figure 22 UTP Pilot Plant Database File in MS Access – "Flow1_Notes" Table



Microsoft Access - [TWX_GLOBAL : Table]

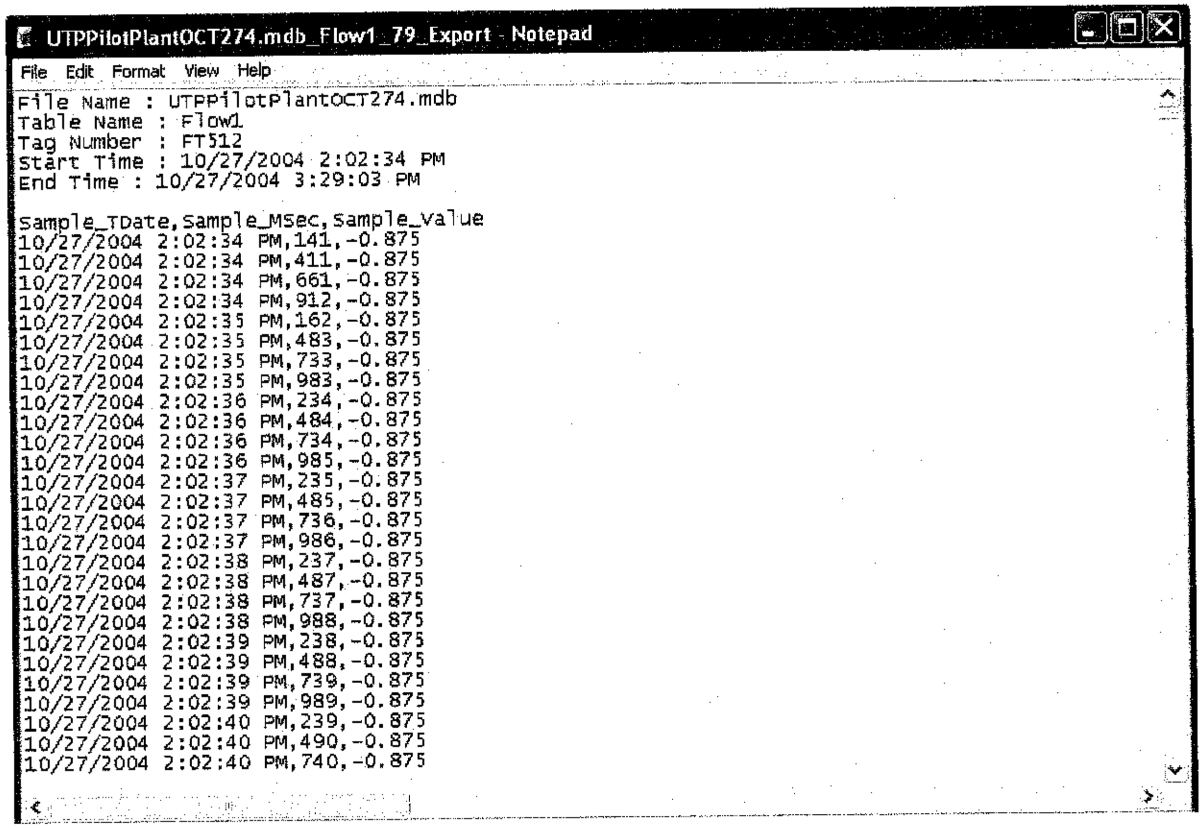
Table_Name	Multiple	Tables_Nu	Table_Tags	Table_Info	Table_Commen	Computer_Nam	DBase_Name	DBase_Index	Group_Name	Last_Updated	Use
Flow1	0	1	Flow1_Tags	Flow1_Info	Flow1_Notes	SERVER	UTPPilotPlant	1	FlowVol	34 12:02:43 PM	
LevelVol	0	1	LevelVol_Tags	LevelVol_Info	LevelVol_Notes	SERVER	UTPPilotPlant	1	LevelVol	34 12:02:43 PM	
Pressure1	0	1	Pressure1_Tags	Pressure1_Info	Pressure1_Note	SERVER	UTPPilotPlant	1	PressureVol	34 12:02:43 PM	
TemperatureVol	0	1	TemperatureVol	TemperatureVol	TemperatureVol	SERVER	UTPPilotPlant	1	TemperatureVc	34 12:02:43 PM	

Record: 4 of 4

Datasheet View

Figure 23 UTP Pilot Plant Database File in MS Access – "TWX_GLOBAL" Table

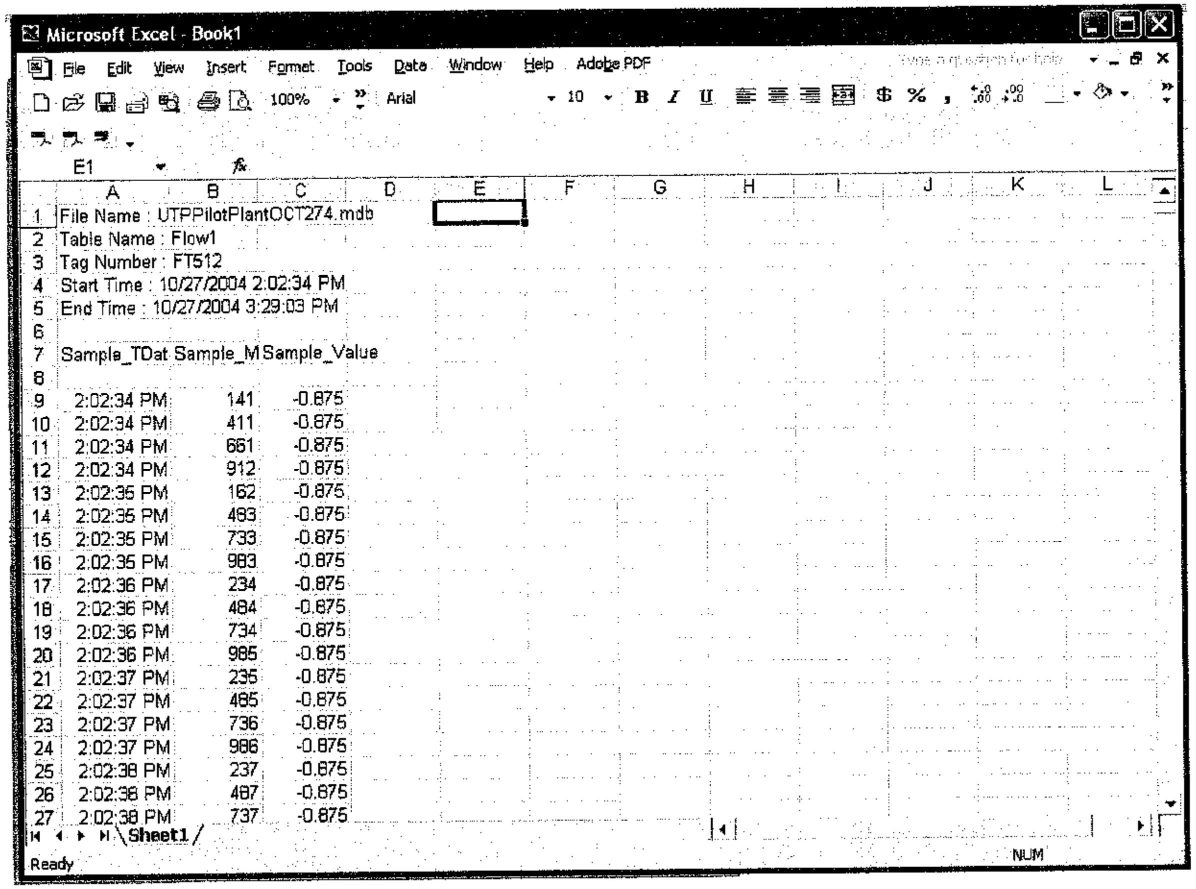
APPENDIX C CONVERTED DATABASE


UTPPilotPlantOCT274.mdb_Flow1_79_Export - Notepad
File Edit Format View Help

```
File Name : UTPPilotPlantOCT274.mdb
Table Name : Flow1
Tag Number : FT512
Start Time : 10/27/2004 2:02:34 PM
End Time : 10/27/2004 3:29:03 PM

Sample_TDate,Sample_MSec,Sample_Value
10/27/2004 2:02:34 PM,141,-0.875
10/27/2004 2:02:34 PM,411,-0.875
10/27/2004 2:02:34 PM,661,-0.875
10/27/2004 2:02:34 PM,912,-0.875
10/27/2004 2:02:35 PM,162,-0.875
10/27/2004 2:02:35 PM,483,-0.875
10/27/2004 2:02:35 PM,733,-0.875
10/27/2004 2:02:35 PM,983,-0.875
10/27/2004 2:02:36 PM,234,-0.875
10/27/2004 2:02:36 PM,484,-0.875
10/27/2004 2:02:36 PM,734,-0.875
10/27/2004 2:02:36 PM,985,-0.875
10/27/2004 2:02:37 PM,235,-0.875
10/27/2004 2:02:37 PM,485,-0.875
10/27/2004 2:02:37 PM,736,-0.875
10/27/2004 2:02:37 PM,986,-0.875
10/27/2004 2:02:38 PM,237,-0.875
10/27/2004 2:02:38 PM,487,-0.875
10/27/2004 2:02:38 PM,737,-0.875
10/27/2004 2:02:38 PM,988,-0.875
10/27/2004 2:02:39 PM,238,-0.875
10/27/2004 2:02:39 PM,488,-0.875
10/27/2004 2:02:39 PM,739,-0.875
10/27/2004 2:02:39 PM,989,-0.875
10/27/2004 2:02:40 PM,239,-0.875
10/27/2004 2:02:40 PM,490,-0.875
10/27/2004 2:02:40 PM,740,-0.875
```

Figure 24 Converted Database File in Comma Separated Values format (*.csv).



Microsoft Excel - Book1

	A	B	C
1	File Name : UTPPilotPlantOCT274.mdb		
2	Table Name : Flow1		
3	Tag Number : FT512		
4	Start Time : 10/27/2004 2:02:34 PM		
5	End Time : 10/27/2004 3:29:03 PM		
6			
7	Sample_TDat	Sample_M	Sample_Value
8			
9	2:02:34 PM	141	-0.875
10	2:02:34 PM	411	-0.875
11	2:02:34 PM	661	-0.875
12	2:02:34 PM	912	-0.875
13	2:02:35 PM	162	-0.875
14	2:02:35 PM	483	-0.875
15	2:02:35 PM	733	-0.875
16	2:02:35 PM	983	-0.875
17	2:02:36 PM	234	-0.875
18	2:02:36 PM	484	-0.875
19	2:02:36 PM	734	-0.875
20	2:02:36 PM	985	-0.875
21	2:02:37 PM	235	-0.875
22	2:02:37 PM	485	-0.875
23	2:02:37 PM	736	-0.875
24	2:02:37 PM	986	-0.875
25	2:02:38 PM	237	-0.875
26	2:02:38 PM	487	-0.875
27	2:02:38 PM	737	-0.875

File Edit View Insert Format Tools Data Window Help Adobe PDF
100%
Arial
10
E1
Sheet1
Ready
NUM

Figure 25 Converted Database File in MS Excel file (*.xls)

APPENDIX D DATA ARRANGEMENT CONVERSION EXAMPLE

Table 2 Data Arrangement for "Signal Index: 6" from "Sample 1" to "Sample 2" of 36 Samples from UTP Pilot Plant Database File

Signal_Index	Earliest_TDate	Latest_TDate	Rec_Modified	Fill_Index	Sample_TDate_1	Sample_MSec_1	Sample_Value_1	Sample_Qual_1	Sample_Modified_1	Sample_TDate_2	Sample_MSec_2	Sample_Value_2	Sample_Qual_2	Sample_Modified_2
6	10/27/2004 3:29:20 PM	10/27/2004 3:29:29 PM	0	36	10/27/2004 3:29:20 PM	928	100	192	1	10/27/2004 3:29:21 PM	178	100	192	1
6	10/27/2004 3:29:30 PM	10/27/2004 3:29:38 PM	0	36	10/27/2004 3:29:30 PM	21	100	192	1	10/27/2004 3:29:30 PM	271	100	192	1
6	10/27/2004 3:29:39 PM	10/27/2004 3:29:47 PM	0	36	10/27/2004 3:29:39 PM	104	100	192	1	10/27/2004 3:29:39 PM	354	100	192	1
6	10/27/2004 3:29:48 PM	10/27/2004 3:29:57 PM	0	36	10/27/2004 3:29:48 PM	197	100	192	1	10/27/2004 3:29:48 PM	447	100	192	1
6	10/27/2004 3:29:57 PM	10/27/2004 3:30:06 PM	0	36	10/27/2004 3:29:57 PM	270	100	192	1	10/27/2004 3:29:57 PM	520	100	192	1
6	10/27/2004 3:30:06 PM	10/27/2004 3:30:15 PM	0	36	10/27/2004 3:30:06 PM	343	100	192	1	10/27/2004 3:30:06 PM	593	100	192	1

Table 3 Data Arrangement for "Signal Index: 6" from "Sample 1" to "Sample 2" of 36 Samples from Converted Database File (Converted data from Table 1)

Sample_Tdate	Sample_Msec	Sample_Value
10/27/2004 3:29:20 PM	928	100
10/27/2004 3:29:21 PM	178	100
10/27/2004 3:29:21 PM	428	100
10/27/2004 3:29:30 PM	21	100
10/27/2004 3:29:30 PM	271	100
10/27/2004 3:29:30 PM	521	100
10/27/2004 3:29:39 PM	104	100
10/27/2004 3:29:39 PM	354	100
10/27/2004 3:29:39 PM	604	100
10/27/2004 3:29:48 PM	197	100
10/27/2004 3:29:48 PM	447	100
10/27/2004 3:29:48 PM	698	100
10/27/2004 3:29:57 PM	270	100
10/27/2004 3:29:57 PM	520	100
10/27/2004 3:29:57 PM	771	100
10/27/2004 3:30:06 PM	343	100
10/27/2004 3:30:06 PM	593	100
10/27/2004 3:30:06 PM	844	100

APPENDIX E FUNCTIONALITY FLOW CHART

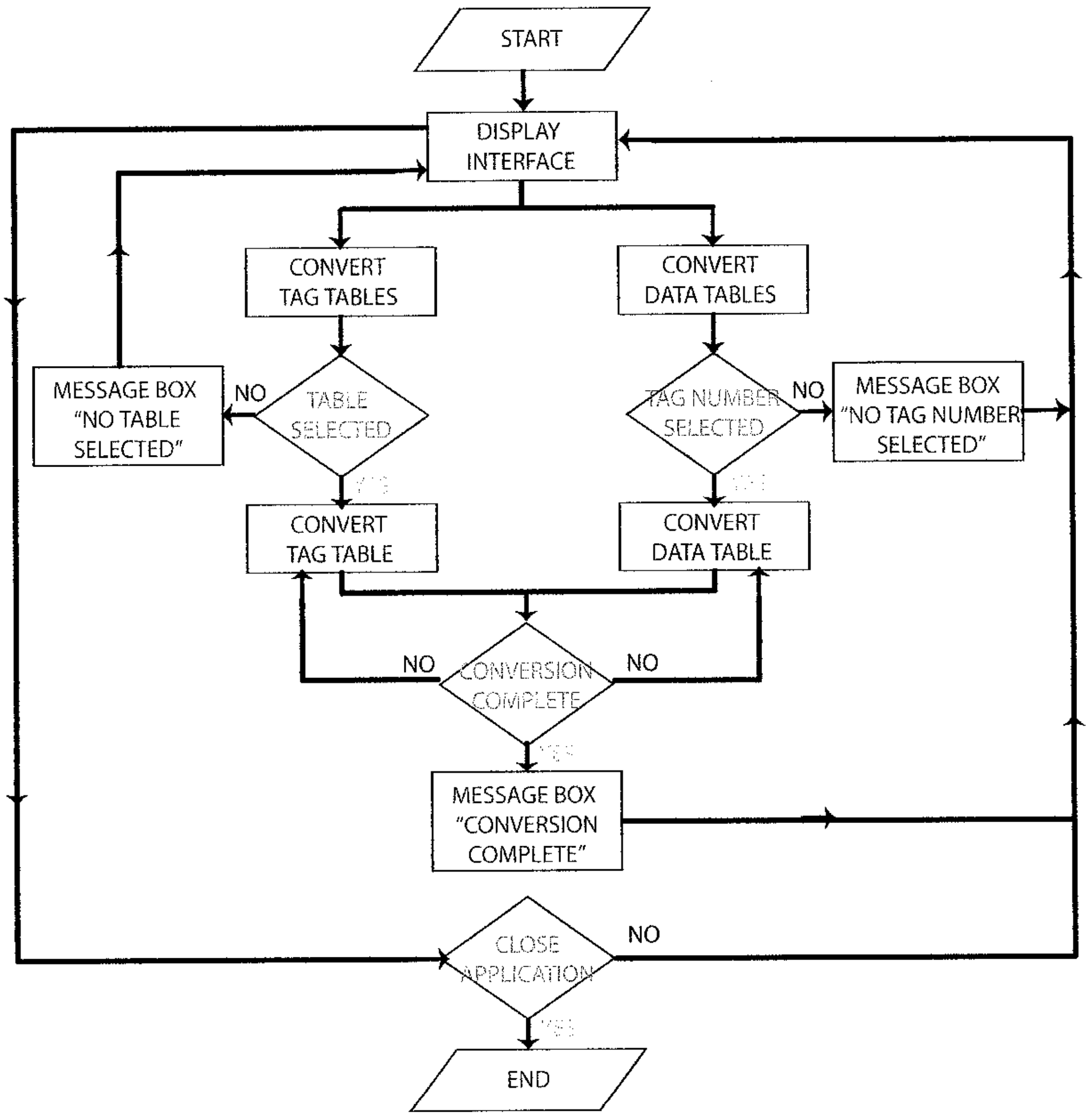


Figure 26 Offline Converter Application Functionality Flow Chart

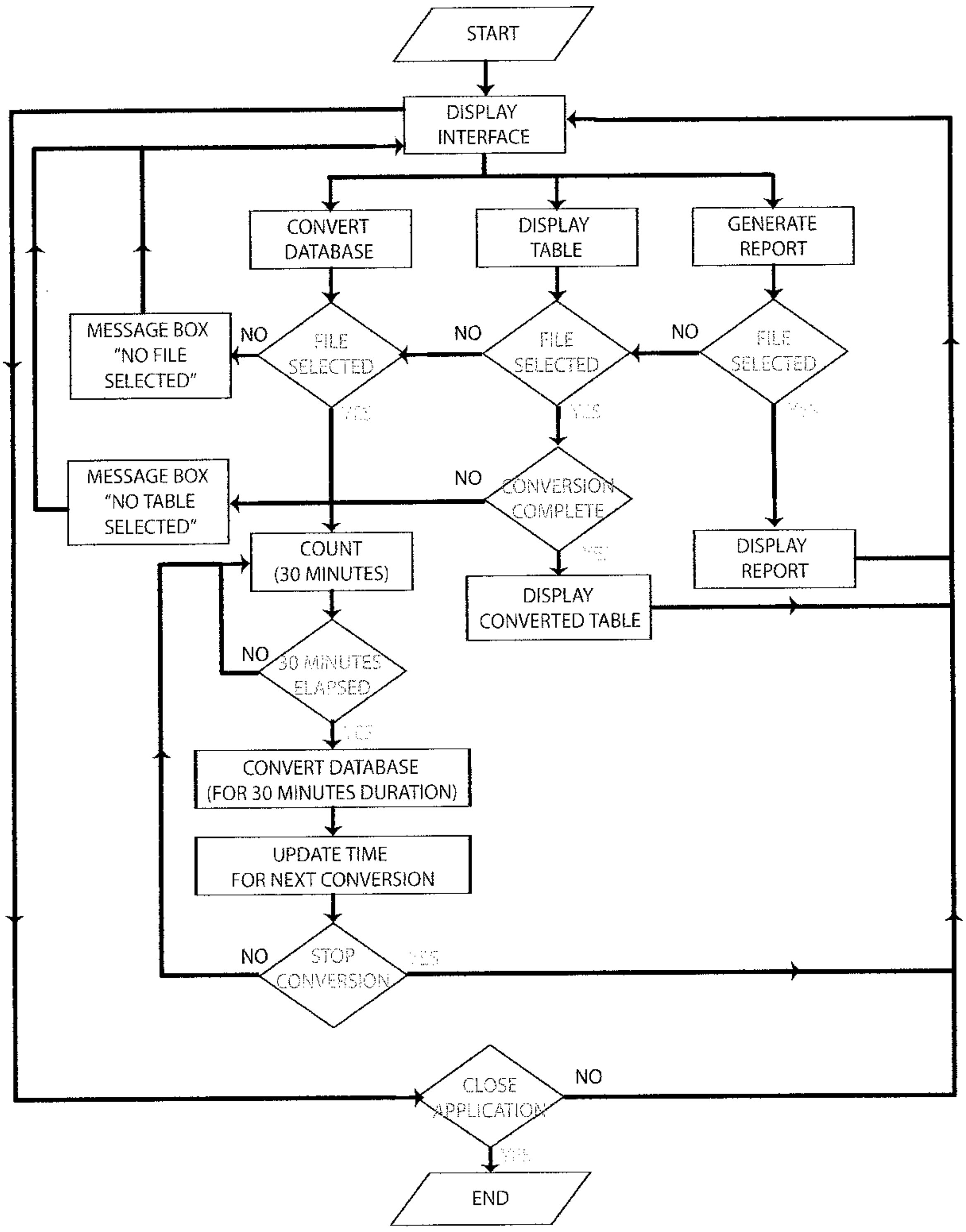


Figure 27 Online Converter Application Functionality Flow Chart

APPENDIX F VB CODINGS

Coding 1: Access to CVS format file Converter

```
Dim db As Database
Dim rs As DAO.Recordset

Private Sub cmdBrow_Click()
Comm.Filter = "Msaccess Database|*.mdb"
Comm.ShowOpen
txtDbname.Text = Comm.FileName
Set db = OpenDatabase(Comm.FileName)
List1.Clear
    ' List the table names.
    For Each TD In db.TableDefs
        ' Do not allow the system tables.
        If Right$(TD.Name, 4) <> "Tags" And _
            Left$(TD.Name, 4) <> "MSys" And _
            Right$(TD.Name, 5) <> "Notes" And _
            Right$(TD.Name, 4) <> "Info" And _
            Left$(TD.Name, 3) <> "TWX" Then _
            List1.AddItem TD.Name

        If Right$(TD.Name, 1) <> "1" And _
            Right$(TD.Name, 3) <> "Vol" And _
            Left$(TD.Name, 4) <> "MSys" And _
            Right$(TD.Name, 5) <> "Notes" And _
            Right$(TD.Name, 4) <> "Info" And _
            Left$(TD.Name, 3) <> "TWX" Then _
            List2.AddItem TD.Name
    Next TD
```

```
db.Close
End Sub
```

```
Private Sub cmdClose_Click()
End
End Sub
```

```
Private Sub Command1_Click()
Dim I, J, rtot
Dim db As Database, rs As DAO.Recordset, rb As _
DAO.Recordset
rtot = ""

Dim objExcl As Excel.Application
Dim info As String

info = List1.Text & "_Info"
If List5.ListIndex = -1 Then GoTo NoTags

Set db = OpenDatabase(txtDbname.Text)
Set rs = db.OpenRecordset("select * from " & List1.Text)
Set rb = db.OpenRecordset("select * from " & info)

Set objExcl = New Excel.Application
objExcl.Visible = True
objExcl.SheetsInNewWorkbook = 1
objExcl.Workbooks.Add

objExcl.ActiveSheet.Cells(1, 1).Value = "File Name : " & _
Comm.FileTitle
```

```
objExcl.ActiveSheet.Cells(2, 1).Value = "Table Name : " & _
List1.Text

objExcl.ActiveSheet.Cells(3, 1).Value = "Tag Number : " & _
List5.Text

objExcl.ActiveSheet.Cells(4, 1).Value = "Start Time : " & _
rb.Fields(1)

objExcl.ActiveSheet.Cells(5, 1).Value = "End Time : " & _
rb.Fields(2)

objExcl.ActiveSheet.Cells(7, 1).Value = "Sample_TDate"
objExcl.ActiveSheet.Cells(7, 2).Value = "Sample_MSec"
objExcl.ActiveSheet.Cells(7, 3).Value = "Sample_Value"

Dim Ind As Integer
Ind = CInt(Text1.Text) 'get input from Textbox labled
'"Text1". Convert text input to integer with CInt

Dim k, l
k = 0
l = 0

Do Until k = 36
J = 9 + k 'Row no 9 to row no k

    Do Until rs.Fields(0) = Ind
    rs.MoveNext
    On Error Resume Next
    Loop
```

```
Do While rs.Fields(0) = Ind
For I = (5 + 1) To (7 + 1) 'Wanted columns

objExcl.ActiveSheet.Cells(J, I - (4 + 1)).Value = _
rs.Fields(I) 'Enter value from Access to Excel

objExcl.ActiveSheet.Cells(J, 1).Value = _
Format(objExcl.ActiveSheet.Cells(J, 1).Value, "Long Time")
Next

objExcl.ActiveSheet.Cells(J, I - (4 + 1)).Value = rtot
'End of column = rtot
rs.MoveNext
J = J + 36 'j incremented by 36 (row)
Loop

k = k + 1 'k incremented by 1 (row)
l = l + 5 'l incremented by 5 (column)

rs.MoveFirst
Loop
MsgBox "Conversion complete"
Exit Sub

NoTags:
MsgBox "No tags indexes selected"
End Sub
```

```
Private Sub Command2_Click()

Dim I, J, rtot
Dim db As Database, rs As DAO.Recordset, rb As _
```

```
DAO.Recordset

rtot = ""

Dim objExcl As Excel.Application
Dim info As String

info = List1.Text & "_Info"

If List2.ListIndex = -1 Then GoTo NoTables

Set db = OpenDatabase(txtDbname.Text)
Set rs = db.OpenRecordset("select * from " & List2.Text)
Set rb = db.OpenRecordset("select * from " & info)

Set objExcl = New Excel.Application
objExcl.Visible = True
objExcl.SheetsInNewWorkbook = 1
objExcl.Workbooks.Add

objExcl.ActiveSheet.Cells(1, 1).Value = "File Name : " & _
Comm.FileTitle

objExcl.ActiveSheet.Cells(2, 1).Value = "Table Name : " & _
List2.Text

objExcl.ActiveSheet.Cells(3, 1).Value = "Start Time : " & _
rb.Fields(1)

objExcl.ActiveSheet.Cells(4, 1).Value = "End Time : " & _
rb.Fields(2)
```

```
For I = 0 To rs.Fields.Count - 1
objExcl.ActiveSheet.Cells(6, I + 1).Value = _
rs.Fields(I).Name
Next

J = 8
Do Until rs.EOF
For I = 0 To rs.Fields.Count - 1
objExcl.ActiveSheet.Cells(J, I + 1).Value = rs.Fields(I)
Next

objExcl.ActiveSheet.Cells(J, I + 1).Value = rtot
rs.MoveNext
J = J + 1
Loop

rs.MoveFirst
MsgBox "Conversion complete"
Exit Sub

NoTables:
MsgBox "No tables selected"
End Sub
```

```
Private Sub Form_Load()

Label3.Font.Name = "Eras Demi ITC"
Label3.Font.Size = 12
Label3.Font.Bold = False
Label3.Font.Underline = True
```

```
Picture1.Picture = LoadPicture("UTP_Logo_FYP.bmp")
Picture1.AutoSize = True
End Sub
```

```
Private Sub List1_Click()

If List1.ListIndex = 0 Then List2.ListIndex = 0
If List1.ListIndex = 1 Then List2.ListIndex = 1
If List1.ListIndex = 2 Then List2.ListIndex = 2
If List1.ListIndex = 3 Then List2.ListIndex = 3

End Sub
```

```
Private Sub List2_Click()

Dim I
Dim db As Database, rs As DAO.Recordset
Dim ctot(1 To 4)

Dim objExcl As Excel.Application
Set db = OpenDatabase(txtDbname.Text)
Set rs = db.OpenRecordset("select * from " & List2.Text)
List5.Clear
Text1.Text = ""

Do Until rs.EOF
I = 0
List5.AddItem rs.Fields(I)
rs.MoveNext
Loop
rs.MoveFirst
```

```
If List2.ListIndex = 0 Then List1.ListIndex = 0
If List2.ListIndex = 1 Then List1.ListIndex = 1
If List2.ListIndex = 2 Then List1.ListIndex = 2
If List2.ListIndex = 3 Then List1.ListIndex = 3

End Sub
```

```
Private Sub List5_Click()

Text1.Text = List5.ListIndex + 1

End Sub
```

```
Dim db As Database
Dim rs As DAO.Recordset
```

```
Private Sub cmdBrow_Click()
Comm.Filter = "Msaccess Database|*.mdb"
Comm.ShowOpen
txtDbname.Text = Comm.FileName

Set db = OpenDatabase(Comm.FileName)
List1.Clear
List2.Clear
List5.Clear

    ' List the table names.
    For Each TD In db.TableDefs
        ' Do not allow the system tables.
        If Right$(TD.Name, 4) <> "Tags" And _
            Left$(TD.Name, 4) <> "MSys" And _
            Right$(TD.Name, 5) <> "Notes" And _
            Right$(TD.Name, 4) <> "Info" And _
            Left$(TD.Name, 3) <> "TWX" Then _
            List1.AddItem TD.Name

        If Right$(TD.Name, 1) <> "1" And _
            Right$(TD.Name, 3) <> "Vol" And _
            Left$(TD.Name, 4) <> "MSys" And _
            Right$(TD.Name, 5) <> "Notes" And _
            Right$(TD.Name, 4) <> "Info" And _
            Left$(TD.Name, 3) <> "TWX" Then _
            List2.AddItem TD.Name
        Next TD
db.Close
End Sub
```

```
Private Sub cmdClose_Click()
End
End Sub
```

```
Private Sub Command2_Click()
Form1.Visible = True

Dim I, J
Dim db As Database, rs As DAO.Recordset, rb As _
DAO.Recordset

info = List1.Text & "_Info"
If List2.ListIndex = -1 Then GoTo NoTables

Set db = OpenDatabase(txtDbname.Text)
Set rs = db.OpenRecordset("select * from " & List2.Text)
Set rb = db.OpenRecordset("select * from " & info)

FileName = Comm.FileTitle & "_" & List2.Text & _
"_Export.csv"

Open FileName For Output As #1
Print #1, "File Name : " & Comm.FileTitle
Print #1, "Table Name : " & List2.Text
Print #1, "Start Time : " & rb.Fields(1)
Print #1, "End Time : " & rb.Fields(2)
Print #1, ""

For I = 0 To rs.Fields.Count - 1 'Wanted columns
    If I <> rs.Fields.Count - 1 Then
    'Enter value from Access to CSV
    Print #1, Trim(rs.Fields(I).Name) & ",";
    Else
    Print #1, Trim(rs.Fields(I).Name)
    End If
Next I

Do Until rs.EOF
        For J = 0 To rs.Fields.Count - 1
        If J <> rs.Fields.Count - 1 Then
        Print #1, Trim(rs.Fields(J)) & ",";
        Else
        Print #1, Trim(rs.Fields(J))
        End If
    Next J
    rs.MoveNext
Loop

    Close #1
    rs.MoveFirst
    MsgBox "Conversion complete"
Exit Sub
NoTables:
MsgBox "No tables selected"
End Sub
```

```
Private Sub Command3_Click()

Dim I, l
Dim db As Database, rs As DAO.Recordset, rb As _
DAO.Recordset

info = List1.Text & "_Info"
If List5.ListIndex = -1 Then GoTo NoTags

Set db = OpenDatabase(txtDbname.Text)
Set rs = db.OpenRecordset("select * from " & List1.Text)
Set rb = db.OpenRecordset("select * from " & info)

FileName = Comm.FileTitle & "_" & List1.Text & "_" & _
Text1.Text & "_Export.csv"

Open FileName For Output As #1
Dim Ind As Integer
Ind = CInt(Text1.Text)
'Get input from Textbox labled "Text1". Convert text input
'to integer with CInt

Print #1, "File Name : " & Comm.FileTitle
Print #1, "Table Name : " & List1.Text
Print #1, "Tag Number : " & List5.Text
Print #1, "Start Time : " & rb.Fields(1)
Print #1, "End Time : " & rb.Fields(2)
Print #1, ""
Print #1, "Sample_TDate,Sample_MSec,Sample_Value"

Do Until rs.Fields(0) = Ind
rs.MoveNext
```

```
On Error Resume Next
Loop
Do While rs.Fields(0) = Ind
        Do While l < 180
        For I = (5 + l) To (7 + l) 'Wanted columns
        If I <> (7 + l) Then
        'Enter value from Access to CSV
        Print #1, Trim(rs.Fields(I)) & ",";
        Else
        Print #1, Trim(rs.Fields(I))
        End If
        Next
        l = l + 5 'j incremented by 5 (column)
    Loop

    rs.MoveNext
    l = 0
Loop

rs.MoveFirst
Close #1
Form1.Visible = False
MsgBox "Conversion complete"
Exit Sub

NoTags:
MsgBox "No tags indexes selected"

End Sub
```

```
Private Sub Form_Load()
```

```
Label3.Font.Name = "Eras Demi ITC"
Label3.Font.Size = 12
Label3.Font.Bold = False
Label3.Font.Underline = True

Picture1.Picture = LoadPicture("UTP_Logo_FYP.bmp")
Picture1.AutoSize = True

End Sub
```

```
Private Sub List1_Click()

If List1.ListIndex = 0 Then List2.ListIndex = 0
If List1.ListIndex = 1 Then List2.ListIndex = 1
If List1.ListIndex = 2 Then List2.ListIndex = 2
If List1.ListIndex = 3 Then List2.ListIndex = 3

End Sub
```

```
Private Sub List2_Click()

Dim I
Dim db As Database, rs As DAO.Recordset
Dim objExcl As Excel.Application

Set db = OpenDatabase(txtDbname.Text)
Set rs = db.OpenRecordset("select * from " & List2.Text)

List5.Clear
Text1.Text = ""
```

```
Do Until rs.EOF
I = 0
List5.AddItem rs.Fields(I)
rs.MoveNext
Loop

rs.MoveFirst

If List2.ListIndex = 0 Then List1.ListIndex = 0
If List2.ListIndex = 1 Then List1.ListIndex = 1
If List2.ListIndex = 2 Then List1.ListIndex = 2
If List2.ListIndex = 3 Then List1.ListIndex = 3

End Sub
```

```
Private Sub List5_Click()

Text1.Text = List5.ListIndex + 1

End Sub
```