

# MULTIMEDIA, MEETINGS, RECORDING, RETRIEVAL AND PLAYBACK

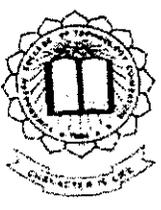
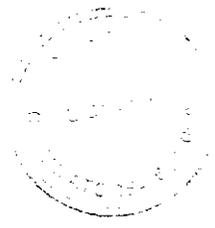
R-945

PROJECT REPORT

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS  
FOR THE AWARD OF THE DEGREE OF  
MASTER OF SCIENCE IN  
APPLIED SCIENCE-SOFTWARE ENGINEERING  
OF BHARATHIYAR UNIVERSITY, COIMBATORE.

Submitted by

Mr. Y.RAGHUVVEER 9837S0057



Under the Guidance of  
MR.K.SIVAN ARUL SELVAN, M.Sc(Phy), M.C.A., PGD.P.M.I.R., MISTE..  
LECTURER, CSE DEPARTMENT  
KUMARAGURU COLLEGE OF TECHNOLOGY, COIMBATORE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
KUMARAGURU COLLEGE OF TECHNOLOGY  
COIMBATORE :- 641006  
APRIL 2003

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
KUMARAGURU COLLEGE OF TECHNOLOGY  
COIMBATORE:- 641006

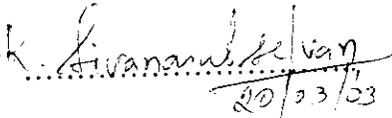
CERTIFICATE

This is to certify that the project entitled  
**MULTIMEDIA, MEETINGS, RECORDING, RETRIEVAL, AND PLAYBACK**

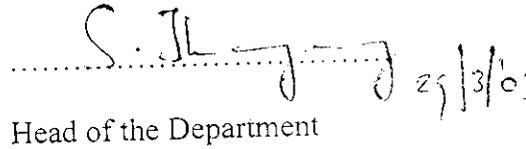
has been submitted by

MR.Y.RAGHUVVEER 98SE17

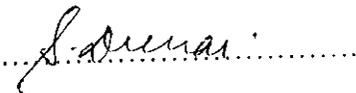
In the partial fulfillment of the award of the degree of  
Master of Science in Applied Science- Software Engineering of  
Bharathiyar University, Coimbatore.  
During the academic year 2002-2003

  
.....  
20/03/03

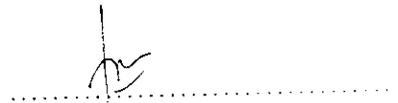
Guide

  
..... 29/3/03  
Head of the Department

Certified that the candidate was examined by us in the Project Work Viva Voce  
Examination held on ..05-04-2003 And the University Register Number was  
.....983750057.....

  
.....

Internal Examiner

  
.....

External Examiner

## DECLARATION

I hereby declare that the project entitled **Multimedia, Meetings, Recording, Retrieval and Playback**, submitted to Bharathiyar University as the project work for Master of Science in Applied Science-Software Engineering Degree, is a good record of original work done by me under the supervision and guidance of Mr.K.Sivan Arul Seivan, Lecturer and Dr.S.Thangaswamy, Professor and Head of the Department of Computer Science and Engineering Department, Kumaraguru College of Technology, Coimbatore and this project work has not formed the basis for the award of Degree/ Diploma/ Associateship/ Fellowship or similar title to any candidate of any university.

Place            Coimbatore  
Date             04/04/2003

  
Y.RAGHUVVEER

*DEDICATED...*

*...TO MY BELOVED PARENTS*

***CERTIFICATE***

## CERTIFICATE

### TO WHOMSOEVER IT MAY CONCERN

This is a bonafide certificate for the successful completion of the project work titled “**Multimedia, Meetings, Recording, Retrieval and Playback (MMRRP)**” done by **Mr. Raghuv eer Y**, student of Computer Science and Engineering Dept., Kumaraguru College of Technology, from Nov 2002 to Mar 2003 as a part of his curriculum for the degree of Master of Science (Applied Science - Software Engineering) in our organization.

The student completed all the study required for this project on his own, discussed his findings with the engineers in the company, learnt new skills and implemented the project. This has turned out to be a very useful project for the company, and we expect to use it companywide on a regular basis. We take this opportunity to congratulate him and wish him all the very best for a bright and promising future.



**K.T. Kumar**  
**Senior Project Engineer**  
**External Project Guide**



**Ganesh Devaraj, Ph.D.**  
**Managing Director**

# ***ACKNOWLEDGEMENT***

I extend my sincere thanks to my beloved Principal **Dr.K.K. Padmanabhan**, B.Sc (Engg),M.Tech.,Ph.D, for his kind support in the successful completion of this project.

I sincerely thank **Dr.S.Thangaswamy**, Ph.D., Head of the Department Computer Science and Engineering, for his encouragement and providing moral support for completing my project successfully.

I thank **Mrs.S.Devaki** B.E, MS., Assistant Professor of the Department Computer Science and Engineering and Course Coordinator of Master of Science Applied Science-Software Engineering for her valuable support for completing the project successfully.

I thank **Mr.K.Sivan Arul Selvan**, M.Sc(phy)., M.C.A.,PGD.P.M.I.R., MISTE.. Lecturer of the Department Computer Science and Engineering for his valuable support and guidance at various levels for completing the project successfully.

I thank my **Parents** for their financial and moral support, without whom I would be nowhere. I also express my sincere gratitude and thanks to my intimate friends who helped me a lot in bringing this project to a successful completion.

I extend my sincere thanks to **Dr.Ganesh Devaraj**, Ph.D., Managing Director and CEO of Soliton Automation India Private Limited, Coimbatore for giving me an opportunity to work on this project in his company. I also thank him for his very valuable suggestions for developing my project into a successful one.

I extend my sincere thanks to **Mr.Anand Prasad Chinnaswamy**, Director and CTO of Soliton Automation India Private Limited, Coimbatore for his very valuable suggestions for developing my project into a successful one.

I extend my sincere thanks to **Mr.K.T.Kumar** - Senior Project Engineer,  
**Dr.VineetGupta** - Project Manager, **Ms.T.Sasikala** - Senior Project Engineer,  
**Mr.V.M.Subramaniam** - Network Administrator, **Mr. T.V.Gokul Dass** - Executive  
Technical Communication, **Mr.R.Gopalakrishnan** - Applications Engineer of Soliton  
Automation India Private Limited, Coimbatore for their very valuable suggestions for  
developing my project into a successful one.

I extend my sincere thanks to all the members of Soliton Automation India Private  
Limited, Coimbatore for their valuable suggestions and support for developing my project  
into a successful one.

# ***CONTENTS***

# CONTENTS

	PAGE NO
SYNOPSIS	
1 INTRODUCTION	
1.1 MEETING IN THE COMPANY	01
1.2 SOLITON AUTOMATION	03
2 SYSTEM REQUIREMENTS	
2.1 PROBLEM STATEMENT	07
2.2 FUNCTIONS TO BE PROVIDED	07
2.3 PROJECT PLAN	07
2.3.1 LIFE CYCLE MODE	07
2.3.2 DEVELOPMENT SCHEDULE	08
2.3.3 SOURCES OF INFORMATION	09
3 PROGRAMMING LANGUAGES AND DEVELOPMENT TOOLS	
3.1 PROGRAMMING LANGUAGES AND DEVELOPMENT TOOLS	10
3.1.1 VC++	10
3.1.2 GOLD WAVE	19
4 SOFTWARE DETAILS	
4.1 PRODUCT SUMMARY AND FUNCTION SPECIFICATION	21
4.2 REQUIREMENTS	24

4.3	DFD	25
5	ISSUES FACED DURING IMPLEMENTATION	26
6	SAMPLE FORMS AND TEST REPORT	
6.1	DESCRIPTION	27
7	CONCLUSION	30
8	REFERENCES	
8.1	WEBSITE	31
8.2	BOOKS	31

***SYNOPSIS***

The project "Multimedia, Meetings, Recording, Retrieval and Playback " has been developed in Soliton Automation India Private Limited, Coimbatore for recording the meetings that take place once a week in the company. This system effectively replaces the regular procedure of jotting down information and maintaining the information for future review in written format.

The major tools used in this project are VC++, GoldWave, Microphone and Web camera. The application is developed using VC++ and GoldWave is used for editing the needed sound file in the required format.

The various operations that can be performed in the system are Recording, Playing the part of the file needed with the help of GoldWave, copying the file to the location needed, deleting the file and to display the properties of the file that is selected.

# ***INTRODUCTION***

## 1.1 Meetings in the Company

Meetings in the company are very important. They help the people in the company to express and transfer information among the group members and other officials within the company. It is a usual practice to have meetings once in a week. Generally there is a practice in the company that all the meeting details are to be taken down manually and entered in to the computer. They are maintained in the system for future review. This is an old traditional process. It is referred as "Minutes of Meeting".

### Disadvantages

- Misinterpretation of information can be a common problem in this method.
- At times critical information might be lost.

In order to reduce these problems the MMRRP (Multimedia, Meetings, Recording, Retrieval and Playback) was planned. The main features of this MMRRP is that the total meeting details is recorded and with the help of yet another software GoldWave the recorded audio is converted into the specified format and edited wherever needed.

The **Sound Recorder** in the windows has a problem that it will record only for a specific period of time. Unless clicked manually, extending the time by manual intervention. Hence, this system was developed.

**Tape Recorder** has a disadvantage with respect to the time of recording, which is usually 90minutes. But a technical session or meeting may go well beyond that. So the person has to change the cassette which is again by manual intervention. This product was also intended to have a video recording. In the next version video will be added to the application.

Next option may be **Camcorder**. Unnecessary waste of money since webcams are available for a lower rate for capturing the video.

This software may well be used in future for multimedia conferencing between various branches and have a log (record) of the meeting. To go for professional software would increase the cost incurred by the company and may not be customized to company's need. Further customization of the professional software by that company may incur a large expenditure which may not be wise for to invest.

## **1.2 Introduction to Soliton Automation India Pvt. Ltd.**

**Soliton Automation India Pvt. Ltd.**, a high technology company based at Coimbatore, India, was started with the aim of becoming a leader in the development of test and measurement and industrial automation software and systems. Soliton Automation specializes in providing comprehensive test & measurement and industrial automation solutions that result in enhanced quality and productivity for our customers around the world. Soliton is one of the few companies in India to offer test & measurement solutions based on the latest Virtual Instrumentation technologies. Today Soliton Automation is India's No.1 provider of Virtual Instrumentation solutions.

### **Milestones & Achievements**

Soliton Automation is committed to developing original and innovative solutions for its customers. This approach has resulted in high levels of customer satisfaction and led to international recognition. It has established a solid presence in the domestic and international Virtual Instrumentation market with its award-winning LabVIEW expertise. The wide acceptance of Soliton's capabilities is reflected in the impressive list of customers and the string of international awards:

- Customers and end-users include industry leaders like DaimlerChrysler, IBM and Nortel Networks in the U.S., and Ford, GE Medical Systems, Exide, Pricol, and others in India.
- In 1999 Soliton became the first Indian company to exhibit in NIWeek, the worldwide conference on Measurement and Automation.
- In 2000, Soliton's entry in the NIWeek 2000 Best Application Contest - 'Large Scale Battery Test System' was selected the 'Most Outstanding Application of Measurement & Automation' becoming the first Indian company to win the most prestigious international Virtual Instrumentation award.
- In 2001, our entry in the NIWeek 2001 Best Application Contest - 'Networked Furnace Monitoring system, won for Soliton the overall best application award for the second year in a row. Soliton became the first company in the world to win the awards twice and in successive years.

- In 2001, Soliton signed an agreement with Cal-Bay Systems, Inc., a Silicon Valley company specializing in Virtual Instrumentation Systems Integration Services to establish the 'Cal-Bay Soliton Development Center', a dedicated Virtual Instrumentation Software Development Center for Cal-Bay at Coimbatore.

## **Technology**

### **Virtual Instrumentation**

Virtual Instrumentation, a concept pioneered by National Instruments, is today the preferred platform for building measurement and automation systems. Virtual Instrumentation leverages the high performance and low cost of today's computers for building powerful automation systems very cost-effectively. Our expertise with the software and hardware products from National Instruments allows us to build customized automation systems for our customers extremely quickly. The major benefits of Virtual Instrumentation systems over traditional instrumentation are

- Higher performance and lower cost by using high volume PC components
- Highly integrated and customized to the requirement
- Modular, reusable, and easy to reconfigure as requirements change
- Built in networking and communications capabilities for enterprise connectivity
- Continuously upgradeable with the latest technologies
- Highly scaleable for future expandability

### **LabVIEW**

LabVIEW, the flagship product from National Instruments, is the leading high-productivity software development platform for Virtual Instrumentation. LabVIEW is the primary development platform at Soliton to build feature-rich, customized automation systems to meet the short development cycles required by our customers.

## **Customized Solutions**

### **Test & Measurement Automation**

- Production Testing System
- Prototype Testing System
- Automated Testing Equipment

### **Industrial Automation**

- HMI/SCADA
- Distributed Data Acquisition & Control
- Process Monitoring Systems

### **Machine Vision & Image Processing**

- Vision based Automation – Calibration, Control, Assembly, Positioning etc.
- Vision based Sorting, Fault Inspection
- Vision based Dimensioning

### **Virtual Instrumentation Software**

Software development using –

- LabVIEW, LabWindows™/CVI, Measurement Studio, Lookout™, and HiQ™ from National Instruments
- C/C++, Visual Basic, and Visual C++

### **Device / Instrument Drivers**

Driver development using –

- LabVIEW, LabWindows/CVI, C/C++

### **e-Factory**

- Instrumenting Production Lines to Monitor Production and Quality
- Networking Production and Quality Monitoring Systems

- Online Production and SPC reports
- Interfacing Factory Floors with ERP Systems

### **Energy Saving System**

- Micro-controller or computer controlled variable speed motor drives for energy saving.

Soliton's product development efforts have achieved significant breakthroughs in developing state-of-the-art products in energy saving, machine vision and data acquisition. The energy saving device for textile autoconers saves up to 60% energy. New products are being developed, of which the Energy Saving System for Humidification Plants in Textile Mills will lead to considerable savings in energy consumption.

### **Industries Served**

- Automotive
- Medical Equipment
- Electronics & Semiconductor
- Battery Manufacturing
- Textile

***SYSTEM  
REQUIREMENTS***

## **2.1 Problem Statement**

To develop a system for recording company meetings and playback.

## **2.2 Functions to be provided**

To Record Meetings

To Playback when ever needed

To Copy a file

To Delete a file

To Display the File Properties

## **2.3 Project Plan**

### *2.3.1 Life Cycle Mode*

System study

Learning VC++ basics

Designing

Integrating

Testing

Validating

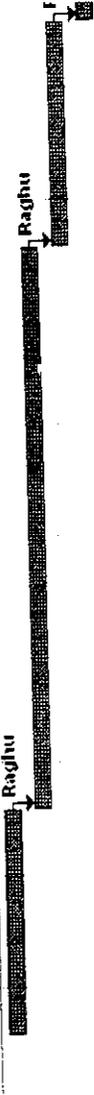
Implementation via Intranet

Implementation via Internet

### *2.3.2 Development Schedule*

System study

Task Name	Duration	Start	Finish
System study	2 wks	Thu 11/21/02	Wed 12/4/02
Learning VC++ basics	5 wks	Thu 12/5/02	Wed 1/8/03
Designing	2 wks	Thu 1/9/03	Wed 1/22/03
Integrating	2 wks	Thu 1/23/03	Wed 2/5/03
Testing	2 wks	Thu 2/6/03	Wed 2/19/03
Validating	1 wk	Thu 2/20/03	Wed 2/26/03
Implementation via Intranet	1 wk	Thu 2/27/03	Wed 3/5/03
Implementation via Internet	2 wks	Thu 3/6/03	Wed 3/19/03
Documentation	1 wk	Thu 3/20/03	Wed 3/26/03





Microsoft Project - schedule

Edit View Insert Format Tools Project Window Help

Task Name	Duration	Start	Finish
System study	2 wks	Thu 11/21/02	Wed 12/4/02
Learning VC++ basics	5 wks	Thu 12/5/02	Wed 1/8/03
Designing	2 wks	Thu 1/9/03	Wed 1/22/03
Integrating	2 wks	Thu 1/23/03	Wed 2/5/03
Testing	2 wks	Thu 2/6/03	Wed 2/19/03
Validating	1 wk	Thu 2/20/03	Wed 2/26/03
Implementation via Intranet	1 wk	Thu 2/27/03	Wed 3/5/03
Implementation via Internet	2 wks	Thu 3/6/03	Wed 3/19/03
Documentation	4 days	Thu 3/20/03	Tue 3/25/03

### *2.3.3 Sources of Information*

Dr. Ganesh Devaraj	Managing Director & CEO
Mr. Anand Prasad Chinnaswamy	Director & CTO
Mr. K.T. Kumar	Senior Project Engineer
Dr. Vineet Gupta	Project Manager
Mr. M.Senthil Kumar	Project Engineer
Mr. T.V.GokulDass	Executive-Technical Communication

***PROGRAMMING  
LANGUAGES AND  
DEVELOPMENT  
TOOLS***

## 3.1 Programming Languages and Development Tools

### 3.1.1 VC++

#### *MFC Introduction*

MFC is a large and extensive C++ class hierarchy that makes Windows application development significantly easier. MFC is compatible across the entire Windows family. As each new version of Windows comes out, MFC gets modified so that old code compiles and works under the new system. MFC also gets extended, adding new capabilities to the hierarchy and making it easier to create complete applications.

The advantage of using MFC and C++ - as opposed to directly accessing the Windows API from a C program-is that MFC already contains and encapsulates all the normal "boilerplate" code that all Windows programs written in C must contain. Programs written in MFC are therefore much smaller than equivalent C programs.

The best part about using MFC is that it does all of the hard work for you. The hierarchy contains thousands and thousands of lines of correct, optimized and robust Windows code. Many of the member functions that you call invoke code that would have taken you weeks to write yourself. In this way MFC tremendously accelerates your project development cycle.

#### *Designing a Program*

The three main objects to write a program.

An application object which initializes the application and hooks it to Windows. The application object handles all low-level event processing.

A window object that acts as the main application window.

A static text object which will hold the static text label or needed information.



Every program that you create in MFC will contain the first two objects. The third object is unique to its application. Each application will define its own set of user interface objects that display the application's output as well as gather input from the user.

### *Various VC++ Functions and Classes Used In MMRRP*

The **CDialog** class is the base class used for displaying dialog boxes on the screen. Dialog boxes are of two types: modal and modeless. A modal dialog box must be closed by the user before the application continues. A modeless dialog box allows the user to display the dialog box and return to another task without canceling or removing the dialog box.

In the implementation (.CPP) file that defines the member functions for your class, start the message map with the **BEGIN\_MESSAGE\_MAP** macro, then add macro entries for each of your message-handler functions.

Each **CCmdTarget**-derived class in your program must provide a message map to handle messages. Use the **DECLARE\_MESSAGE\_MAP** macro at the end of your class declaration.

Use the **END\_MESSAGE\_MAP** macro to end the definition of your message map.

The **GetDlgItemText** function retrieves the title or text associated with a control in a dialog box.

The **OnPaint** Prepares the container for painting, gets the control's client area, then calls the control class's **OnDraw** method.

The **IsIconic** function determines whether the specified window is minimized (iconic).

The **SendMessage** function sends the specified message to a window or windows. It calls the window procedure for the specified window and does not return until the window procedure has processed the message.

The **GetDlgItem** function retrieves a handle to a control in the specified dialog box.

The **EnableWindow** function enables or disables mouse and keyboard input to the specified window or control. When input is disabled, the window does not receive input such as mouse clicks and key presses. When input is enabled, the window receives all input.

The **IMPLEMENT\_DYNCREATE** macro with the **DECLARE\_DYNCREATE** macro to enable objects of **CObject**-derived classes to be created dynamically at run time. The framework uses this ability to create new objects dynamically

The **CString** object consists of a variable-length sequence of characters. **CString** provides functions and operators using a syntax similar to that of Basic. Concatenation and comparison operators, together with simplified memory management, make **CString** objects easier to use than ordinary character arrays.

Retrieves the window's text. The second version of this method allows you to store the text in a **BSTR**. If the text is successfully copied, the return value is **TRUE**; otherwise, the return value is **FALSE**.

The **SetWindowText** function changes the text of the specified window's title bar (if it has one). If the specified window is a control, the text of the control is changed. However, **SetWindowText** cannot change the text of a control in another application.

The **OnOK** function is called when the user clicks the OK button that is the button with an ID of **IDOK**. Override this member function to perform the OK button action. If the dialog box includes automatic data validation and exchange, the default implementation of this member function validates the dialog-box data and updates the appropriate variables in your application.

If you implement the OK button in a modeless dialog box, you must override the **OnOK** member function and call **DestroyWindow** from within it. Don't call the base-class member function, because it calls **EndDialog**, which makes the dialog box invisible but does not destroy it.

**OnSetFocus** Checks that the control is in-place active and has a valid control site, then informs the container the control has gained focus.

The **CEdit** class provides the functionality of a Windows edit control. An edit control is a rectangular child window in which the user can enter text.

**RemoveAll** removes all the elements from this array. If the array is already empty, the function still works.

The **Lock** method blocks other clients from modifying the variables stored in the **Application** object, ensuring that only one client at a time can alter or access the **Application** variables. If you do not call the **Unlock** method explicitly, the server unlocks the locked **Application** object when the .asp file ends or times out.

The **Remove** method removes the element from the head of the list and returns a pointer to it. You must ensure that the list is not empty before calling **RemoveHead**. If the list is empty, then the Debug version of the Microsoft Foundation Class Library asserts. Use **IsEmpty** to verify that the list contains elements.

The **WAVEFORMATEX** structure defines the format of waveform-audio data. Only format information common to all waveform-audio data formats is included in this structure. For formats that require additional information, this structure is included as the first member in another structure, along with the additional information.

## *Members of WAVEFORMATEX*

### **wFormatTag**

Waveform-audio format type. Format tags are registered with Microsoft Corporation for many compression algorithms. A complete list of format tags can be found in the MMREG.H header file.

### **nChannels**

Number of channels in the waveform-audio data. Monaural data uses one channel and stereo data uses two channels.

### **nSamplesPerSec**

Sample rate, in samples per second (hertz), that each channel should be played or recorded. If **wFormatTag** is WAVE\_FORMAT\_PCM, then common values for **nSamplesPerSec** are 8.0 kHz, 11.025 kHz, 22.05 kHz, and 44.1 kHz. For non-PCM formats, this member must be computed according to the manufacturer's specification of the format tag.

### **nAvgBytesPerSec**

Required average data-transfer rate, in bytes per second, for the format tag. If **wFormatTag** is WAVE\_FORMAT\_PCM, **nAvgBytesPerSec** should be equal to the product of **nSamplesPerSec** and **nBlockAlign**. For non-PCM formats, this member must be computed according to the manufacturer's specification of the format tag.

Playback and record software can estimate buffer sizes by using the **nAvgBytesPerSec** member.

### **nBlockAlign**

Block alignment, in bytes. The block alignment is the minimum atomic unit of data for the **wFormatTag** format type. If **wFormatTag** is **WAVE\_FORMAT\_PCM**, **nBlockAlign** should be equal to the product of **nChannels** and **wBitsPerSample** divided by 8 (bits per byte). For non-PCM formats, this member must be computed according to the manufacturer's specification of the format tag.

Playback and record software must process a multiple of **nBlockAlign** bytes of data at a time. Data written and read from a device must always start at the beginning of a block. For example, it is illegal to start playback of PCM data in the middle of a sample (that is, on a non-block-aligned boundary).

### **wBitsPerSample**

Bits per sample for the **wFormatTag** format type. If **wFormatTag** is **WAVE\_FORMAT\_PCM**, then **wBitsPerSample** should be equal to 8 or 16. For non-PCM formats, this member must be set according to the manufacturer's specification of the format tag. Note that some compression schemes cannot define a value for **wBitsPerSample**, so this member can be zero.

### **cbSize**

Size, in bytes, of extra format information appended to the end of the **WAVEFORMATEX** structure. This information can be used by non-PCM formats to store extra attributes for the **wFormatTag**. If no extra information is required by the **wFormatTag**, this member must be set to zero. Note that for **WAVE\_FORMAT\_PCM** formats (and only **WAVE\_FORMAT\_PCM** formats), this member is ignored.

The **ZeroMemory** function fills a block of memory with zeros.

The **MMCKINFO** structure contains information about a chunk in a RIFF file.

### *Members of MMCKINFO*

#### **ckid**

Chunk identifier.

#### **cksize**

Size, in bytes, of the data member of the chunk. The size of the data member does not include the 4-byte chunk identifier, the 4-byte chunk size, or the optional pad byte at the end of the data member.

#### **fccType**

Form type for "RIFF" chunks or the list type for "LIST" chunks.

#### **dwDataOffset**

File offset of the beginning of the chunk's data member, relative to the beginning of the file.

#### **dwFlags**

Flags specifying additional information about the chunk. It can be zero or the following flag:

#### **MMIO\_DIRTY**

The length of the chunk might have changed and should be updated by the `mmioAscend` function. This flag is set when a chunk is created by using the `mmioCreateChunk` function.

The `mmioAscend` function ascends out of a chunk in a RIFF file descended into with the `mmioDescend` function or created with the `mmioCreateChunk` function.

The **mmioWrite** function writes a specified number of bytes to a file opened by using the **mmioOpen** function.

The **mmioRead** function reads a specified number of bytes from a file opened by using the **mmioOpen** function.

The **mmioFOURCC** macro converts four characters into a four-character code. This macro does not check whether the four-character code it returns is valid.

The **IDirectSound3DBuffer::GetMode** method retrieves the current operation mode for 3-D sound processing.

**AfxMessageBox** displays a message box on the screen. The first form of this overloaded function displays a text string pointed to by *lpzText* in the message box and uses *nIDHelp* to describe a Help context. The Help context is used to jump to an associated Help topic when the user presses the Help key. The second form of the function uses the string resource with the ID *nIDPrompt* to display a message in the message box. The associated Help page is found through the value of *nIDHelp*. If the default value of *nIDHelp* is used (-1), the string resource ID, *nIDPrompt*, is used for the Help context.

The **PostQuitMessage** function indicates to the system that a thread has made a request to terminate (quit). It is typically used in response to a **WM\_DESTROY** message.

The **InitInstance** is a derived class of **CWinApp**. Windows allows several copies of the same program to run at the same time. Application initialization is conceptually divided into two sections: one-time application initialization that is done the first time the program runs, and instance initialization that runs each time a copy of the program runs, including the first time. The framework's implementation of **WinMain** calls this function.

Override **InitInstance** to initialize each new instance of your application running under Windows. Typically, you override **InitInstance** to construct your main window object and set the **CWinThread::m\_pMainWnd** data member to point to that window.

**ON\_THREAD\_MESSAGE** Indicates which function will handle a user-defined message. **ON\_THREAD\_MESSAGE** must be used instead of **ON\_MESSAGE** when you have a **CWinThread** class. User-defined messages are any messages that are not standard Windows **WM\_MESSAGE** messages. There should be exactly one **ON\_THREAD\_MESSAGE** macro statement in your message map for every user-defined message that must be mapped to a message-handler function.

### 3.1.2 Gold Wave

#### Features

- Converting files from one format to the other.
- Select a part of file and play.
- Various Functions for dealing with the recorded sound file.

In the application MMRRP the output sound file is recorded in the WAV format. So in order to convert in to the needed format the sound file is opened and it is saved in the needed format.

The sound file can be edited at any location and it can be alone played. It is a process which helps the project manager or any person related to hear the needed part instead of the full one.

#### Function Commands for modifying a sound file

*Doppler* Dynamically changes pitch.

*Dynamics* Distorts, compresses or limits output levels.

*Echo* Produces an echo.

*Expand* Compresses, limits, or expands dynamic range.

*/Compress*

*Filter* Displays a submenu listing several filter related commands.

*Flange* Creates a number of flanging related audio effects.

*Interpolate* Uses linear interpolation to smooth out samples between the start and finish markers. Use this command on a tiny selection to remove a pop or click.

*Invert* Turns the sound upside-down.

*Mechanize* Adds a mechanical characteristic.

*Offset* Changes the dc offset.

*Pitch* Changes or transposes the pitch (frequency).

*Reverse* Reverses the selection so that it plays backwards.

*Silence* Silences (mutes) the selection.

*Stereo* Displays a submenu showing stereo related effects.

*Time warp* Changes the speed and stretches or compresses the length of the selection.

*Volume* Displays a submenu listing several volume related commands.

*Playback* Changes the rate at which the entire sound is played.

*Resample* Changes the entire sound so that it can be played at a different sampling rate.

***SOFTWARE  
DETAILS***

## 4.1 Product Summary and Function Specification

### MMRRP

The system that i developed will help in recording the Audio of all that is spoken during the meeting. There various buttons that are present on the exe of the system are as follows.

- Record
- Play
- Copying a File
- Deleting a File
- File Properties
- Quit

There is a single field in the menu called the Enable.

When the application is started then all the buttons in the exe that are listed are in the disable mode. The operator i order to perform any operation in the application he has to first click the enable button and make all the buttons active or enable them.

### Record

When the record button is clicked then the text in the button is automatically set to “Stop”. This is a special feature in the project because when the operator thinks of stopping the recording that is currently going on then he can just click on the Stop button. On doing this the sound file recording is stopped and the sound file is placed in the location where the exe file is located. And this leaves the content in the button again to “Record”.

## **Play**

When the play button is clicked then the text in the button is automatically set to "Stop". On doing so the sound file that is been recorded is linked to the application and then it is played. There is a special feature in this application that is when the file playing has reached the end of file then the text in the button is automatically replace to "Play". Otherwise when the operator wants to stop the application then he has to just click on the "Stop" button then this terminates the application and automatically sets the text in the button to "Play".

## **Copying a File**

On clicking the button on the application it displays a dialog box. The dialog box contains the following buttons.

- Ok
- Cancel
- Browse

In the dialog box there are mainly two edit boxes which are labeled "Source" and "Target". In this the Source location it consist of the name of the file which is to be copied. In the target location it consist of the path where the file has to be stored. When the operator clicks the on the "Browse" button then the operation is performed and the end result is displayed in the message box that is whether the file has been copied successfully.

## **Deleting a File**

On clicking the button on the application it displays a dialog box. The dialog box contains the following buttons.

- Ok
- Cancel
- Browse

In the dialog box there is one edit box which is labeled "Source". In this the Source location it consist of the name of the file which is to be deleted. The delete operation can be performed in the intranet. When the operator clicks the on the "Browse" button then the operation is performed and the end result is displayed in the message box that is whether the file has been deleted successfully.

### **File Properties**

On clicking this button in the application the "Browse" window is opened and the operator is needed to select the file name for which he has to know the details regarding the file like the extension, path and other details. When he selects the file and on clicking "Ok" button in the browse window the needed details are displayed in the message box.

### **Quit**

On clicking this button the application is aborted .

### **Storing Recorded Sound File**

The sound file which is been stored is in the same location where the application exe is located. After the sound file is stored then it should be renamed so that there is a record maintained. If the file is not renamed and when the application is executed again for recording a new meeting then the old sound file is deleted and the new file overwritten.

## 4.2 Requirements

### Hardware

Pentium II and above  
6.4 GB hard disk capacity  
64 MB RAM and above  
Microphone MA-369  
Kodak DVC 323

### Software

VC ++  
Gold Wave

## 4.3 DFD

General Application Structure

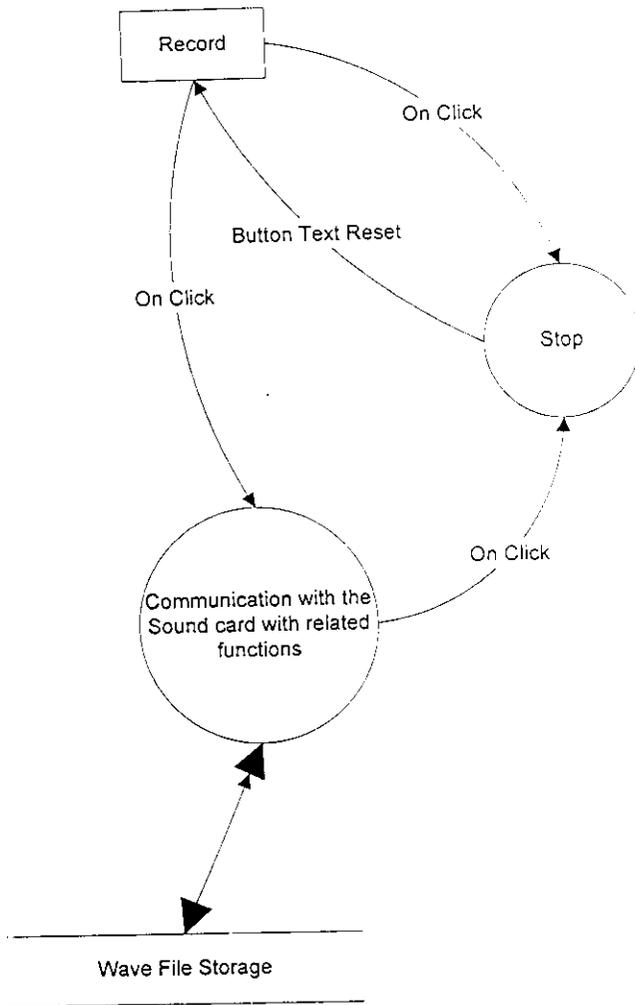
Recording a File

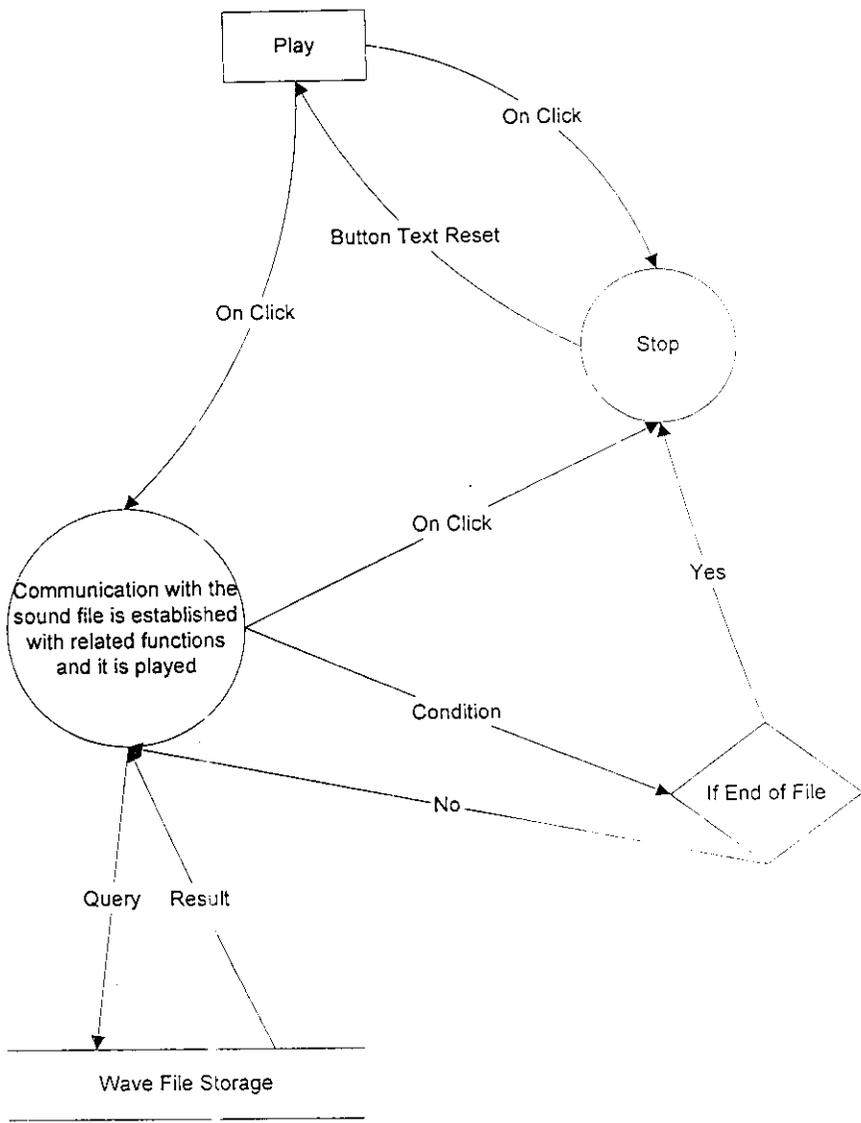
Playing the Recorded File

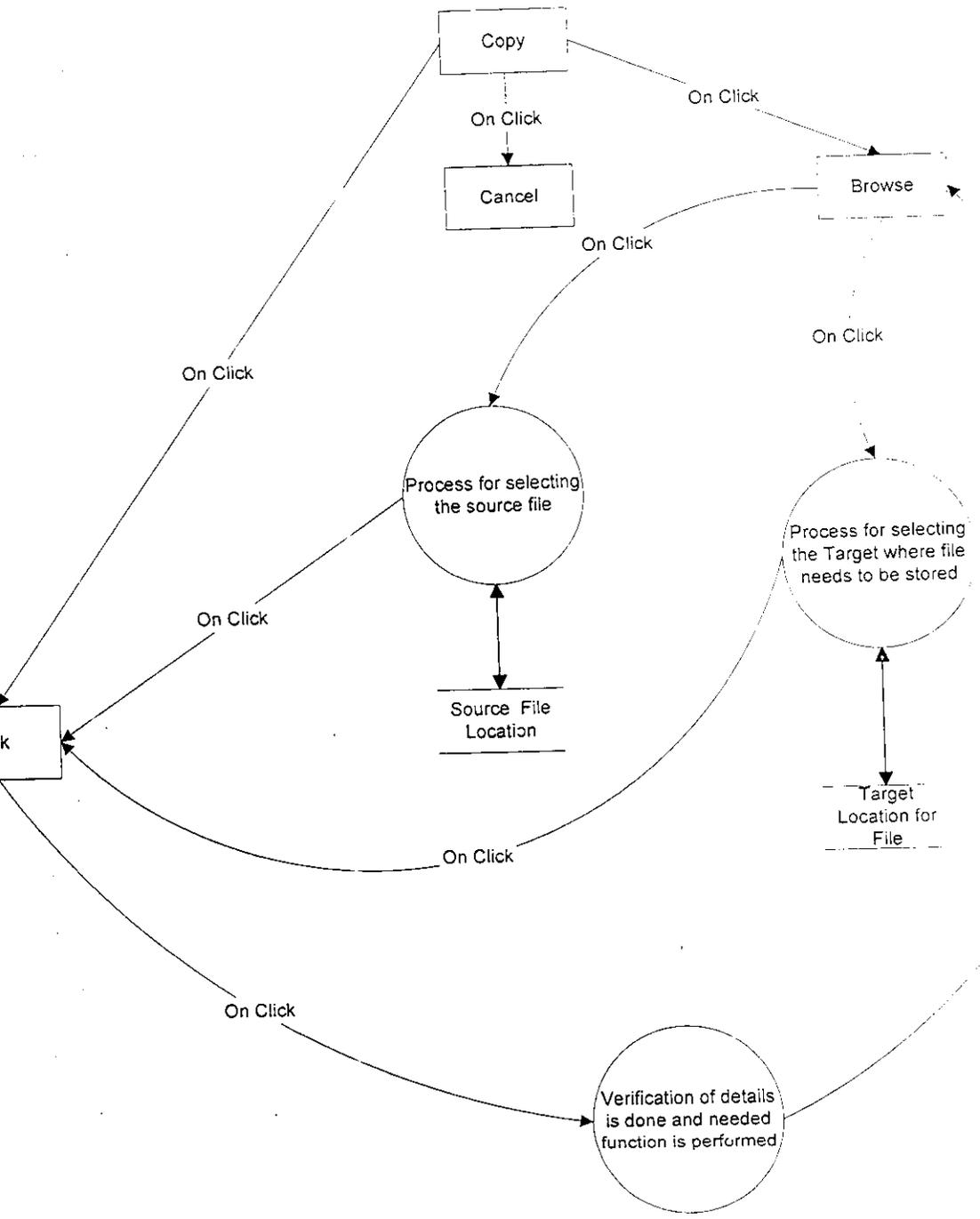
Copying the File

Deleting the File

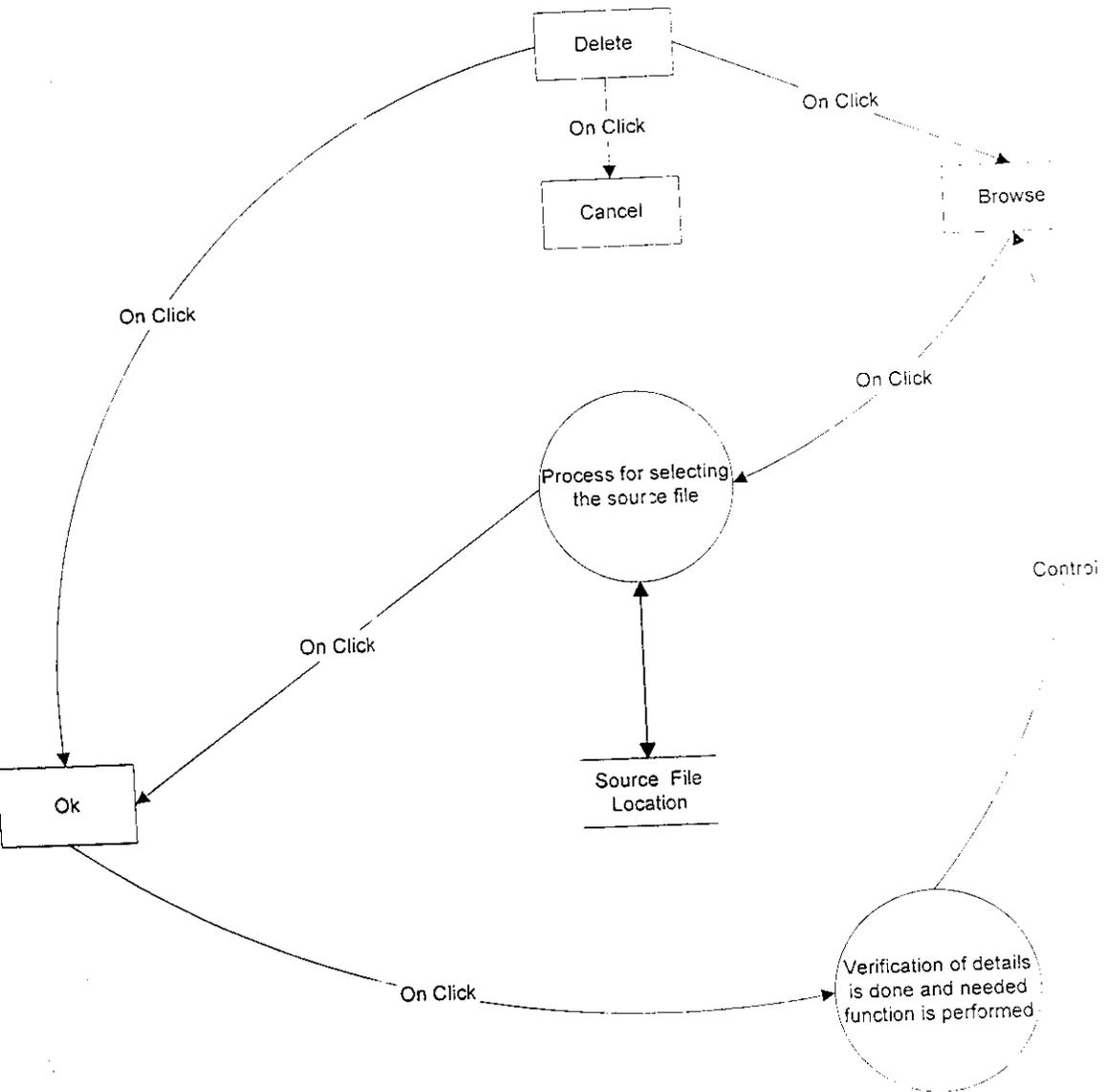
Viewing File Properties

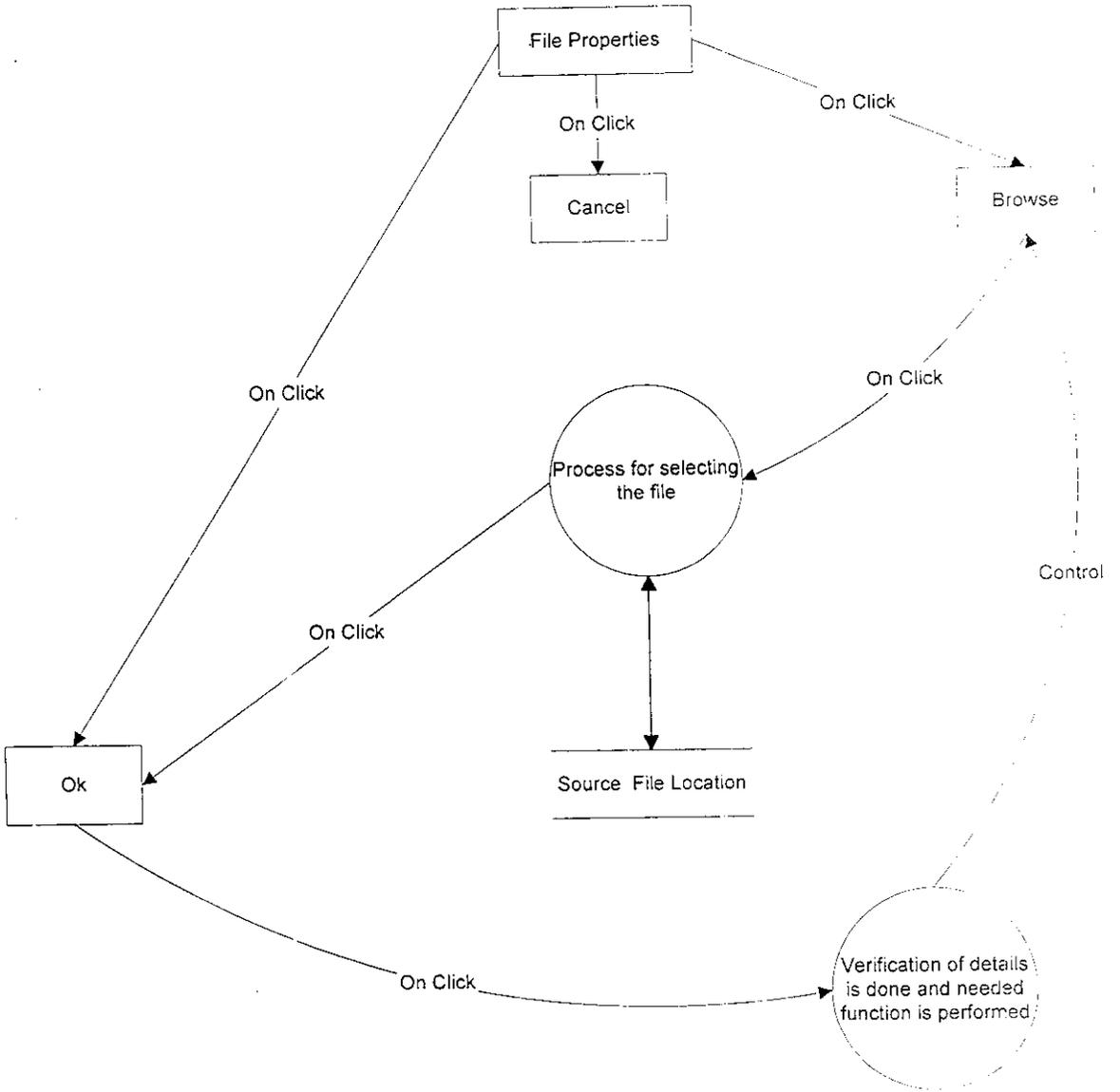


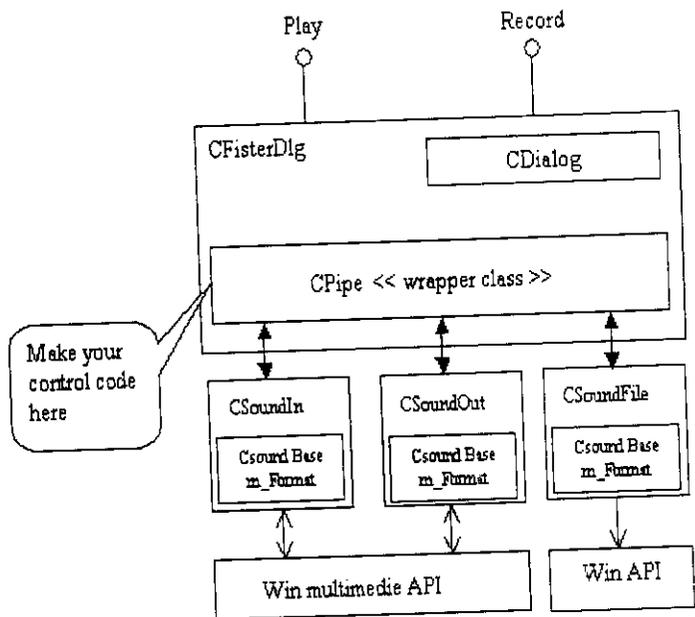




Control







This is an application that shows you how to play and record sound under windows. It uses the old multimedia API, a better solution may be to use DirectSound.

Start with the two functions in CFisterDlg called OnPlay and OnRecord. Follow them down to the depth you need to use the classes.

CSoundIn is a wrapper class that will let you retrieve sound from the soundcard. The main functions are Start() and Stop()

CSoundOut is a wrapper class that will let you play sound on the soundcard. The main functions are Start() and Stop()

CSoundFile is a wrapper class of a single wave file, it can either be a file reader or a file writer object. See the constructor.

CSoundBase is a very small class that encapsulates the wave format.

CBuffer is a very small class that encapsulates a simple one dimensional buffer.

The project has a number of different callback functions:

One callback function makes the Play button change its label to stop when it has finished playing the file. CDialogDlg inherits CPipe and overloads a function that CPipe can call when it has finished playing the wave file.

Another callback function makes it possible for CSoundIn to callback to CPipe when it has filled the input buffer. Thus CPipe can give CSoundIn a new buffer to fill.

A clone of the above principle is also used in CSoundOut, which enables it to callback to the owner when it is finished playing the sound in a given buffer.

## **5 Issues faced during Implementation**

While recording the meetings with MMRRP, noise is a major issue which affects the quality of sound. With the help of another application software called Gold Wave the noise was removed from the sound file. This problem was solved in about 3 weeks.

When the exe file was send to a remote system for recording the particular conference then it popped up a dialog box stating that there were few DLL files were missing. On installing the DLL files the application was working as needed.

***SAMPLE FORMS AND  
TEST REPORT***

## **6.1 Description Sample Screens and Test Report**

The various sample views that are present in the application MMRRP are shown in the following pages.



ALS

Audio



Soliton  
the engineers



Quit

Play

Record

Copy File

Delete File

File Properties

ALS

Audio

Enable



Record

Play

Quit

File Properties

Delete File

Copy File



ACS

Audio



Quit

Play

Record

Copy File

Delete File

File Properties

ACS Audio



Stop

Play

Quit

File Properties

Delete File

Copy File



Audio



**Soliton**  
for engineering

Record

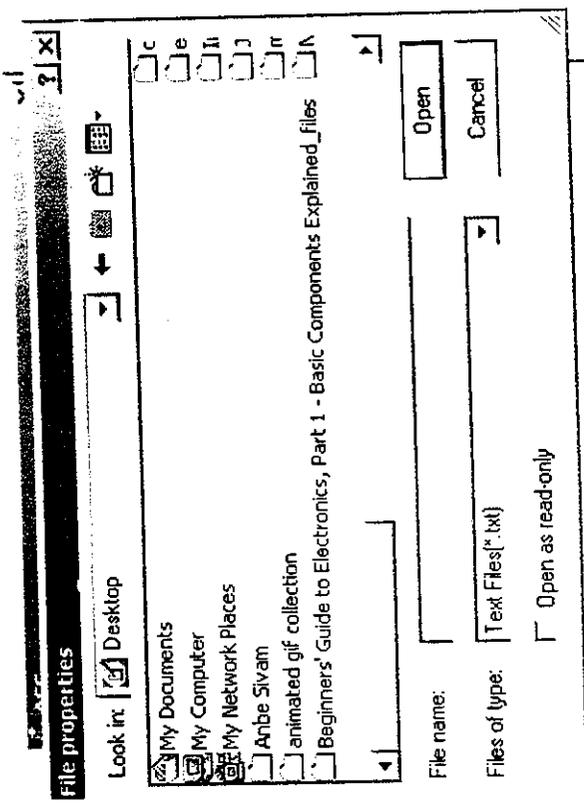
Stop

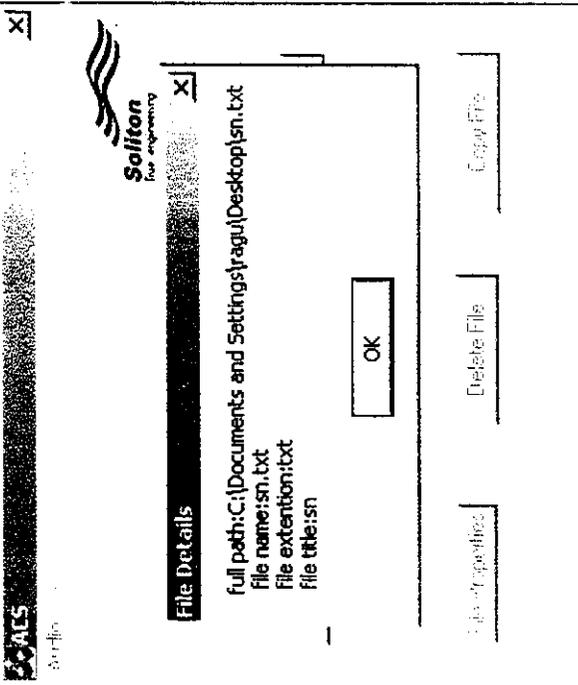
Quit

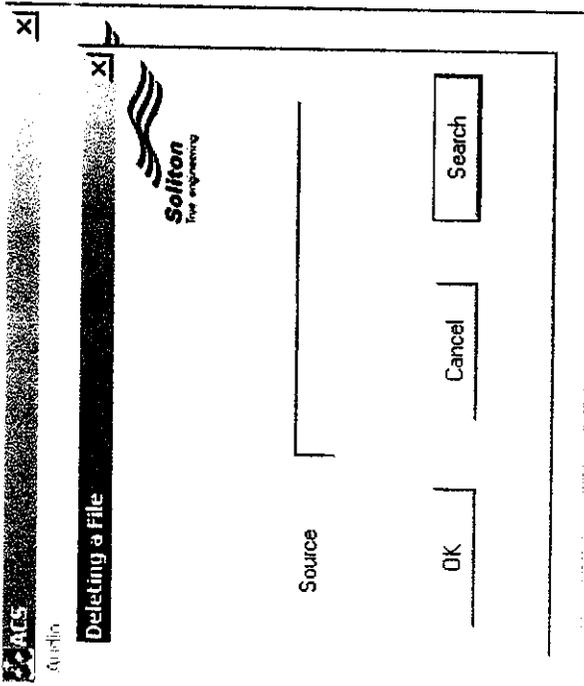
File Properties

Delete File

Copy File







Alt Copying a File



Source \_\_\_\_\_

Target \_\_\_\_\_

MMRRP Main Page

*Buttons Available*

*Record*

*Play*

*Quit*

*Delete File*

*Copy File*

*File Properties*

Button Name	Input	Desired Output	Actual Output	Comments
Record	On click	The button text changes to Stop and the Audio file is recorded	Same operation was performed	Verified
Play	On click	The button text changes to Stop and the Audio file is played	Same operation was performed	Verified
Quit	On Click	Exit the system	Same operation was performed	Verified
Copy a File	On click	The dialog box asking for source and target file will be displayed and after entering them	Same operation was performed	Verified

		and clicking on OK button then the File is copied and a message box is displayed		
Delete a File	On click	The dialog box asking for source file will be displayed and after entering it and clicking on OK button then the File is deleted and a message box is displayed	Same operation was performed	Verified
File Properties	On click	The dialog box asking for source file for which details are to be displayed after entering it and clicking on OK button then the File properties are listed displayed	Same operation was performed	Verified

***CONCLUSION***

The project MMRRP was successfully implemented in Soliton Automation India Private Limited. It is used for recording the meetings that take place in the company.

In the future the same application will be modified in such a way that it helps in recording the video from various office branches located in different parts of the country. The next future enhancement in the system is incorporation of facilities like virtual conferencing.

## ***REFERENCES***

## 8.1 Website

1. [www.microsoft.com](http://www.microsoft.com)
2. [www.codeguru.com](http://www.codeguru.com)
3. [www.google.com](http://www.google.com) a search engine
4. [www.misha.com](http://www.misha.com) for Microphone

## 8.2 Books

1. Name Visual C++ Programming  
Author Yeshwanth Kanetkar  
Page No Full Book  
Year of Publication 1998
2. Name Visual C++ Handbook  
Author SSI press  
Page No From 160 onwards  
Year of Publication 1998
3. Name Mastering Visual C++ 6  
Author Michael Young  
Page No 1279 on wards  
Year of Publication 1998

NOTE : MMRRP – Multimedia, Meetings, Recordings, Retrieval, and Playback