


1997

Microsoft Windows based database management system for roller dam gates

Timothy Paul Meyer
Iowa State University

Follow this and additional works at: <https://lib.dr.iastate.edu/rtd>

 Part of the [Civil Engineering Commons](#), and the [Structural Engineering Commons](#)

Recommended Citation

Meyer, Timothy Paul, "Microsoft Windows based database management system for roller dam gates" (1997). *Retrospective Theses and Dissertations*. 17280.

<https://lib.dr.iastate.edu/rtd/17280>

This Thesis is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

Microsoft Windows based database management system for roller
dam gates

by

Timothy Paul Meyer

A thesis submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

Major: Civil Engineering (Structural Engineering)

Major Professor: Dr. Lowell Greimann

Iowa State University

Ames, Iowa

1997

Graduate College
Iowa State University

This is to certify that the Master's thesis of
Timothy Paul Meyer
has met the thesis requirements of Iowa State University

Signatures redacted for privacy.

Dedicated to my parents Dale and Aileen for their support throughout
my educational pursuits.

And to my wife Amy for her support, companionship, and understanding
until now and forever.

TABLE OF CONTENTS

LIST OF FIGURES.....	vii
LIST OF TABLES.....	x
ACKNOWLEDGEMENTS	xi
CHAPTER 1. INTRODUCTION.....	1
1.1 Background	1
1.2 Objective	2
1.3 Overview	3
CHAPTER 2. PROGRAM DEVELOPMENT AND REQUIREMENTS	6
2.1 Introduction	6
2.2 Program Requirements	6
2.2.1 Program Requirements for Data Entry	6
2.2.2 Program Requirements for CI Calculation	7
2.2.3 Program Requirements for CI Results and Output.....	7
2.3 Program Development.....	8
2.3.1 The Uses of Oracle Power Objects 1.0 as a Development Tool	9
2.3.2 The Uses of Visual Basic 4.0 as a Development Tool	9
2.4 Structure of winREMR.....	10
2.4.1 Map Forms	10
2.4.2 Data Entry Forms	10
2.4.3 Data Entry Result Forms	10
2.4.4 Result Forms	11
2.4.5 System Forms	11
2.5 Development of winREMR	11
2.5.1 Front End Development using Visual Basic 4.0	14
2.5.1.1 Map Forms	16
2.5.1.2 Data Entry Forms	17
2.5.1.3 Data Entry Result Forms	17
2.5.1.4 Result Forms	18
2.5.1.5 System Forms	18
2.5.2 Database Development using Microsoft Access	19
CHAPTER 3. USERS MANUAL FOR winREMR 1.0.....	22
3.1 Introduction	22
3.2 Identifying Structure and Inspection Date for Structure.....	22
3.2.1 Identifying a New or Different Structure.....	22
3.2.1.1 Identifying a New Structure	23
3.2.1.2 Using the Default Structure.....	30
3.2.1.3 Saving a Structure as a Default Structure.....	31

3.2.2	Creating, Editing, and Deleting Form Dates	31
3.2.2.1	Creating a New Date	33
3.2.2.2	Editing a Previously Existing Date.....	34
3.2.2.3	Deleting a Previously Existing Date.....	35
3.3	Data Entry Forms, Data Entry Result Forms, and Result Forms.....	36
3.3.1	Choosing a Form	40
3.3.2	Filling out a Data Entry Form	42
3.3.2.1	Entering and Editing Data in a Text Entry Box	42
3.3.2.2	Using Buttons on Data Entry Forms	43
3.3.2.3	Using Drop Down List on Data Entry Forms.....	44
3.3.2.4	Using Record Indicators on Data Entry Form.....	44
3.3.3	Saving Data on a Data Entry Forms.....	45
3.3.3.1	Saving Data on Only One Data Entry Form.....	45
3.3.3.2	Saving Data on All Data Entry Forms.....	46
3.3.4	Opening and Closing Multiple Forms	46
3.3.4.1	Opening Multiple Forms	47
3.3.4.2	Closing Multiple Forms	47
3.3.5	Printing Result Forms or Data Entry Result Forms.....	48
3.3.5.1	Printing Only One Result Form or Data Entry Result Form.....	48
3.3.5.2	Printing All the Result Forms or Data Entry Result Forms.....	48
3.4	Calculating and Viewing CI.....	49
3.4.1	Calculating and Viewing CI in Data Entry Forms.....	49
3.4.1.1	Calculating CI for Distresses Listed on Active Data Entry Form.....	50
3.4.1.2	Calculating CI for all the Distresses Listed on all Data Entry Forms .	50
3.4.1.3	Viewing CI in Data Entry Forms.....	52
3.4.2	Viewing CI in the Result Forms.....	52
3.4.2.1	Viewing Component CI in the Result Form.....	53
3.4.2.2	Viewing the Sub-Component CI using the Result Forms	54
CHAPTER 4. INSTALLATION GUIDE FOR winREMR 1.0.....		56
4.1	Hardware Requirement and Recommendations	56
4.2	Installation Instructions for winREMR	56
CHAPTER 5. NEW FEATURES IN winREMR.....		58
5.1	Introduction	58
5.2	Changes to the Chose the Inspection Structure Method.....	58
5.3	Changes in the Data Entry Forms.....	59
5.4	Changes to the Condition Index Calculations	59
5.5	Changes to the Result Forms.....	60
CHAPTER 6. SUMMARY AND RECOMMENDATIONS.....		62
6.1	Summary	62
6.2	Recommendations	63

APPENDIX 1. ROLLER GATE COMPUTATIONAL MODULE.....	65
APPENDIX 2. OPERATING EQUIPMENT COMPUTATIONAL MODULE.....	75
APPENDIX 3 OTHER CONTRIBUTIONS TO THE PROJECTS.....	91
REFERENCES	94

LIST OF FIGURES

Figure 1.1	Typical Page of Field Inspection Data Form.....	5
Figure 2.1	Typical Map Form.....	11
Figure 2.2	Typical Data Entry Form.....	12
Figure 2.3	Typical Date Entry Result Form.....	13
Figure 2.4	Typical Result Form.....	14
Figure 2.5	Typical System Form.....	15
Figure 2.6	Form <i>frmdate</i>	19
Figure 2.7	Form <i>frmstart</i>	20
Figure 2.8	Form <i>frmcal</i>	20
Figure 3.1	Identify Inspection Form Structure Dialog Box.....	23
Figure 3.2	Civil Works Division Map	24
Figure 3.3	North Central Division Map.....	25
Figure 3.4	Rock Island District Part 1 Map.....	27
Figure 3.5	Rock Island District Part 2 Map.....	28
Figure 3.6	Save Default Structure Dialog Box.....	29
Figure 3.7	Rock Island District Lock and Dam #15 Map.....	29
Figure 3.8	Data Identification Window	32
Figure 3.9	Dialog Box for Creating New Date.....	33
Figure 3.10	Dialog Box for Deleting a Previously Existing Date	35
Figure 3.11	Typical Data Entry Form.....	37
Figure 3.12	Typical Data Entry Result Form.....	38

Figure 3.13	Typical Result Form.....	39
Figure 3.14	Data Entry Pages Sub Menu.....	41
Figure 3.15	Result Pages Sub Menu.....	41
Figure 3.16	Typical Text Entry Box.....	42
Figure 3.17	Typical Button.....	43
Figure 3.18	Typical Drop Down Pick List.....	44
Figure 3.19	Typical Record Indicator.....	44
Figure 3.20	File Sub Menu	46
Figure 3.21	Typical Save Data Dialog Box.....	47
Figure 3.22	Typical Data Entry Form with Condition Index Information.....	51
Figure 3.23	Typical Condition Index Bar Chart.....	53
Figure 3.24	Typical Sub-Distress Form.....	54
Figure 4.1	Run Dialog Box.....	57
Figure 4.2	Directory Dialog Box From Installation Program.....	57
Figure A1.1	REMR DBMS Edit Menu	66
Figure A1.2	RGCM Main Menu	67
Figure A1.3	Computational Module “What If” Scenario Menu.....	68
Figure A2.1	REMR DBMS Edit Menu	77
Figure A2.2	OECM Main Menu.....	77
Figure A2.3	OECM Summary Report Main Menu	78
Figure A2.4	OECM Available Power Transfer Numbers.....	79
Figure A2.5	OECM “What If” Main Menu.....	80

Figure A2.6	OECM Assembly List	80
Figure A2.7	OECM Gear Rack Assembly Distress List.....	81
Figure A2.8	OECM “What If” Noise, Jump, and Vibration Window For Gear Rack Assembly	82
Figure A2.9	OECM Save Data Window	82

LIST OF TABLES

Table 2.1	Name of Forms Contained within winREMR.....	16
Table 2.2	List of Tables Contained Within Database.....	21
Table A2.1	Data File Naming Table	85

ACKNOWLEDGEMENTS

I would like to acknowledge both David McKay and Stuart Foltz at the Construction Engineering Research Laboratories of the United States Army Corps of Engineers for their insight and suggestion while completing this project. Also, to Jim Stecker and Dr. Lowell Greimann for not only providing an invaluable resource, but also for providing the freedom to complete this project.

CHAPTER 1 INTRODUCTION

1.1 Background

For the last 100 years the U.S. Army Corps of Engineers has been constructing civil work lock and dam structures both on internal waterways and on the coast line. These structures have a variety of uses from hydro-electric generation to increasing the ability to navigate a waterway. With most of the civil work structures reaching their design life and the potential to replace the civil work projects limited, the need to extend the life of the structures is becoming more important. The U.S. Army Corps of Engineers has addressed this problem by developing the Repair, Evaluation, Maintenance, and Rehabilitation (REMR) Research Program. A part of the REMR research program developed methods to assess the condition of the lock or dam by taking various simple measurements from the structure. Rules have been developed to obtain a Condition Index (CI) for the structure using the measurement obtained from the structure.

The REMR research program depends on the use of a Database Management System (DBMS) which has been written for the U.S. Army Corps of Engineers to assist in storing field inspection data for future use. The current DBMS is used only to store data and requires a computational module to evaluate the data and assign a CI. Used in combination, the two software programs simplify the process of finding the CI for a structure.

Although the current DBMS is presently being used by the U.S. Army Corps of Engineers, it has drawbacks. First, the present software is written to run in MS-DOS, which Microsoft will probably discontinue supporting. Although the need for an upgrade has been seen by the Corps of Engineers, presently there are no formal plans to upgrade the DBMS

from MS-DOS to another operating system, thus limiting the life of the DBMS. Second, documentation of the present DBMS is poor at best. Not all commands available for use in the DBMS are mentioned in the documentation and those that are mentioned are not always explained completely. Also, no additional information on the present DBMS exists in the public domain. Third, the DBMS is prone to unreliable action, e.g., crashing for no apparent reason and giving unreproducible errors while running.

A research effort concentrating on the development of a window based DBMS for storing and analyzing data has been completed at Iowa State University (ISU). This DBMS is considered an experimental platform which stores all data entered into the system and generates various outputs. This DBMS was developed to work with the Condition Rating Procedures for Roller Dam Gate report (Greimann et al., 1995) and is capable of storing only the data developed using this report.

Being able to use a DBMS to store and analyze the data is essential. This assures that the data is saved for further use and also minimizes the human error in analyzing the data. Errors in the analysis of data could result in serious problems going unnoticed, possibly resulting in a failure of a roller gate. This failure could result in loss of pool, potentially stopping all navigation on that stretch of river until the pool is restored.

1.2 Objective

The purpose of the following research is to develop a new DBMS for use by the Army Corps of Engineers in its condition assessment program. The DBMS will be designed to run on a personal computer (PC) in Microsoft Windows. In addition, the software will be easy to

use, upgradable, and adaptable to all the structures investigated by the REMR research program.

1.3 Overview

The concept put forth in the development of a Windows based DBMS stems from various other projects for steel sheet pile (Greimann et al., 1989) (Greimann et al., 1990), miter lock gates (Greimann et al., 1990), sector gates (Greimann et al., 1991), tainter and butterfly valves (Greimann et al., 1994), tainter dam and lock gate (Greimann et al., 1994), REMR management systems' user's manual and rating software (Greimann et al., 1994), operating equipment (Greimann et al., 1996), and roller dam gate (Greimann et al., 1996). The program is also loosely modeled after an existing DBMS developed for the REMR program. Because of this similarity, the name chosen for this newly developed program is winREMR. Basic ideas, such as CI, distresses, inspection forms, rules for obtaining CI's, and others, are used as outlined in the roller dam gate report (Greimann et al., 1996).

A roller dam gate is used to control the upper pool elevation on a dam structure. Several different distresses are used to describe different aspects of the roller gates condition. This condition is represented by a CI which is a numerical scale that ranges from a low of 0 and a high of 100. The CI is a numerical measure used to rate the current condition of a distress. A combined CI is also used to describe the condition of the entire gate. This CI is developed using rules outlined in the Roller Gate Report (Greimann et al., 1996). These rules are based on inspection data taken during the field inspection.

During the field inspection, the roller gate is investigated using simple tools like a tape measure, dial gage, level, transit, etc... This tools are used to take simple measurements

that are recorded on the field inspection forms. A typical page of the inspection form can be seen in Figure 1.1. The entire inspection form is in the roller dam gate report (Greimann et al., 1996) along with instructions for completing the inspection itself. With this data and the rules for the roller dam gate, the CI for the gate can be achieved.

The winREMR program is used to store and analyze the data taken in the field inspection. The process of using winREMR is generally as follows. The names of the structure and sub-structure that have been inspected in the field is entered into the computer using a series of maps. The structure indicates the civil work project in question, e.g. Lock and Dam # 14 on the Mississippi River. The sub-structure indicates a component of the structure, e.g. Roller Dam Gate #4. The date of the field inspection is also entered. Once this is entered, the field inspection data can be entered into the program. The CI can be calculated from the field inspection data using the rules outlined in the roller gate report and incorporated into winREMR. Once the CI is calculated, it is presented to the user using a graphical format.

U.S. ARMY CORPS OF ENGINEERS
ROLLER DAM GATE STRUCTURE INSPECTION

PAGE 4

PAGES 4,5,6 COMPLETED UNDER
 (1)BULKHEADED OR (2)OUT OF OPERATION CONDITIONS
 (WITH GATE LIFTED OUT OF WATER): (no.) _____

INSPECTION DATE: _____ INSPECTED BY: _____
 CURRENT UPPER POOL ELEVATION: (ft.) _____
 CURRENT LOWER POOL ELEVATION: (ft.) _____

RACK ANCHORAGE DETERIORATION

	DRIVEN END	NON-DR. END
CONCRETE CRACKED OR SPALLED?	<u>Y N</u>	<u>Y N</u>
IS THERE EVIDENCE OF MOVEMENT?	<u>Y N</u>	<u>Y N</u>

RACK CONNECTIONS DETERIORATION

NUMBER OF BOLTS/NUTS:	DRIVEN END	NON-DR. END
CORRODED?	_____	_____
LOOSE?	_____	_____
MISSING?	_____	_____

RIM CONNECTIONS DETERIORATION

NUMBER OF BOLTS/NUTS/RIVETS:	DRIVEN END	NON-DR. END
CORRODED?	_____	_____
LOOSE?	_____	_____
MISSING?	_____	_____

RIM/RACK TOOTH DAMAGE

TYPE OF DAMAGE: CHIPPED (H), DEFORMED (D), PITTED (P),
 GOUGED (G), EXCESSIVE WEAR (W)

RIM			RACK		
TOOTH NUMBER	DAMAGE TYPE	%SURFACE AREA	TOOTH NUMBER	DAMAGE TYPE	%SURFACE AREA
DRIVEN END	(H,D,P,G,W)	_____	DRIVEN END	(H,D,P,G,W)	_____
NON-DRIVEN END	(H,D,P,G,W)	_____	NON-DRIVEN END	(H,D,P,G,W)	_____

RIM/RACK TRACK DAMAGE

IS THERE DEFORMATION OF THE	DRIVEN END	NON-DR. END
RIM TRACK?	<u>Y N</u>	<u>Y N</u>
RACK TRACK?	<u>Y N</u>	<u>Y N</u>

GUARD RAIL DAMAGE

IS THE GUARD RAIL DEFORMED?	DRIVEN END	NON-DR. END
IS DEFORMATION GREATER THAN 1-in.?	<u>Y N</u>	<u>Y N</u>
	<u>Y N</u>	<u>Y N</u>

Figure 1.1 Typical Page of Field Inspection Data Form

CHAPTER 2 PROGRAM DEVELOPMENT AND REQUIREMENTS

2.1 Introduction

winREMR was developed to be a possible replacement for the current version of REMR Tools. Since winREMR was designed to be a replacement, it had to maintain some of the same basic features contained within the current version of REMR Tools. It still must permit the user to enter and retrieve inspection form data and must calculate the CI from this data. It must also make the results of the calculation available to the user.

2.2 Program Requirements

Before winREMR was developed, the current version of REMR Tools was evaluated to identify areas in need of improvement. Once these areas were identified and requirements were developed for these areas, every attempt was made to implement these improvements into winREMR. The various requirements can be grouped into three different areas. These are:

- Program Input
- Calculation of the CI
- Program Output

2.2.1 Program Requirements for Data Entry

During the process of identifying improvements to entering of data, several possibilities came to mind. From the evaluation of these suggestions, the following list of requirements was developed for the program input.

1. The structure and sub-structure should be identified using a series of maps in which the user would click on the structure and sub-structure of interest.
2. The maps should be easy to understand and easy to use.
3. The data entry forms should be accessible to the user in any order that they chose.
4. The user should be able to view more than one data entry form at a time.
5. The data entry forms should contain various graphics and diagrams to aid the user in entering the data.
6. The data entry forms should be easy to use.

2.2.2 Program Requirements for CI Calculation

For the CI calculation section, two requirements were developed. These requirements are as follows:

1. The inclusion of the CI calculation within the actual database management system. This would make it possible to calculate the CI without requiring the user to exit the data entry section.
2. The ability to calculate the distress CI's on the data entry form directly. This gives the user the ability to evaluate one distress without entering data for all the distresses.

2.2.3 Program Requirements for CI Results and Output

The CI results and output section received the most suggestions for improvement. The main goal of this aspect of the program was to provide as much information to the user as they desire but still present the information in an easy to understand format. To accomplish this goal, the following requirements were developed.

1. The CI of the structure and the individual CI's should be presented using a bar chart that also contains the numerical values of the CI's.
2. Charts showing the different contributions of the sub-distress components to the combined distress CI should be available at the user's request.
3. The charts should be easy to understand and easy to use.
4. The charts should be accessible to the user in any order.
5. The user should be able to view more than one chart at a time.
6. The user should have the ability to print out all the charts that are available within the program.
7. The user should have the option to print out the inspection form data to obtain a hard copy of the data.
8. The user should also have the ability to print out a blank copy of the inspection forms.

2.3 Program Development

Since winREMR was designed to be an application that runs within a Windows environment and to provide easy-to-use database interaction, many different development tools were considered. In order to choose the best development tool, a few requirements were identified for the development tool. The requirements are as follows:

1. The development tool had to provide the ability to create an executable file that would operate on a different computer without the use of the development tool.
2. The development tool had to provide quick and easy access to an attached database with minimal effort from the developer.

3. The development tool had to provide a quick and easy way to develop Window applications.
4. The development tool had to be easy to operate and quick to learn.

2.3.1 The Uses of Oracle Power Objects 1.0 as a Development Tool

At the onset of the development of winREMR, Oracle Power Objects 1.0 (OPO) was chosen as the development tool. This tool combined the use of a relational database with the ability to quickly create links to various forms by simply dragging the contents from the database table to the form. At first this program seemed to work well, but with time difficulties developed. It was found that some of the features included in the program did not always work the way they were described to work. It was also found that some of the limitations included in the program were not clearly explained within the references. With these and various other problems that arose in the development of winREMR, OPO was dropped as the development tool approximately three months into development.

2.3.2 The Use of Visual Basic 4.0 as a Development Tool

The second development tool chosen was Visual Basic 4.0 (VB4). VB4 was chosen for two basic reasons. The first reason is that it fulfilled all the requirements outlined for the development tool. The second reason was that there were many resources available for VB4. It was believed that, with all these resources, the problems encountered while using OPO could be solved using VB4. This was found to be true. The one draw back which resulted in VB4 not being chosen as the first choice was that VB4's database connection functions were not as easy to use as the ones contained within OPO.

2.4 Structure of winREMR

The winREMR program has five main components. These components are listed below and explained in the following Section.

- Map Forms
- Data Entry Forms
- Data Entry Result Forms
- Result Forms
- System Forms

2.4.1 Map Forms

Map Forms are used by the user to identify a civil work structure. A typical Map Form can be viewed in Figure 2.1.

2.4.2 Data Entry Forms

Data Entry Forms are used to enter the field inspection data into the database. They provide an easy to use interface to the database. A typical Data Entry Form can be viewed in Figure 2.2.

2.4.3 Data Entry Result Forms

Data Entry Result Forms resemble the Data Entry Forms. They differ from the Data Entry Forms in that the user can not enter data into the Data Entry Result Forms. The Data Entry Result Forms are also used to view and print out the field inspection data. A typical Data Entry Result Form is shown in Figure 2.3.

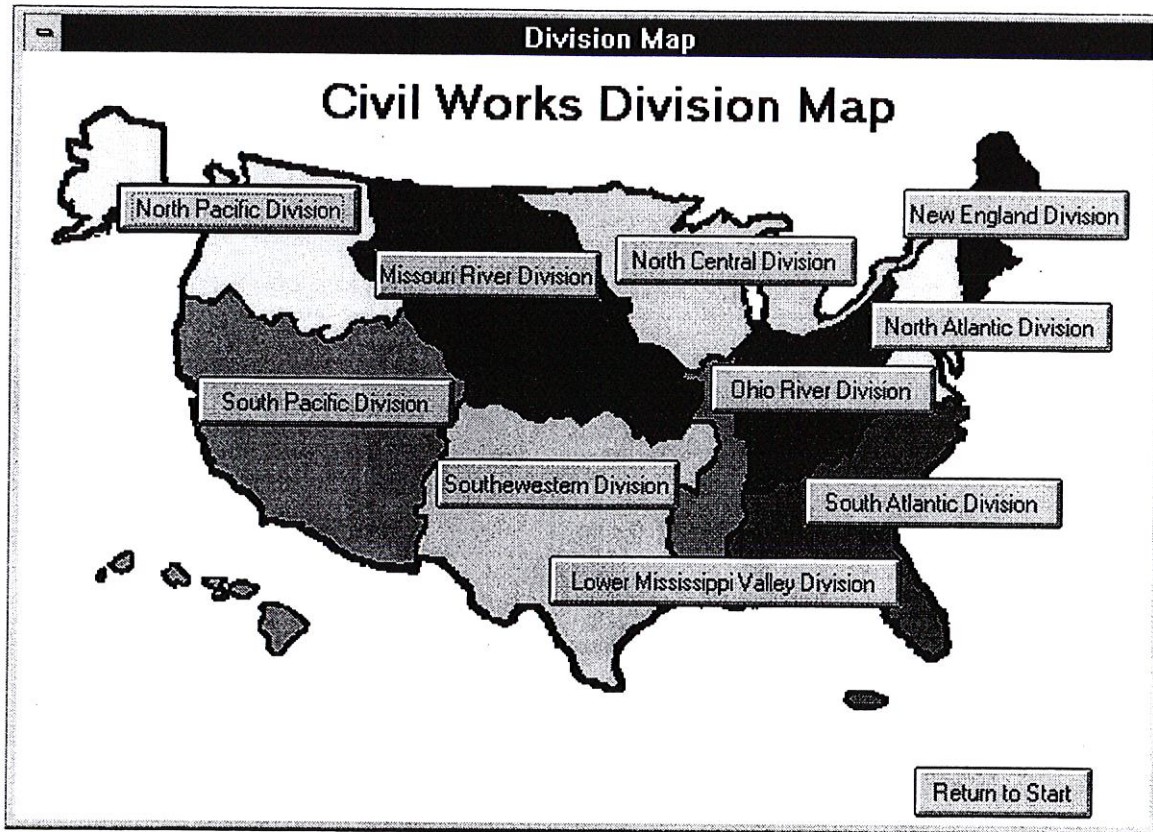


Figure 2.1 Typical Map Form

2.4.4 Result Forms

Result Forms are used to display the CI's calculated in the program. The Result Forms present this information as a bar chart. A typical Result Form is shown in Figure 2.4.

2.4.5 System Forms

System Forms contain all the forms that do not fall into the other four categories. The System Forms are used to perform miscellaneous functions throughout the program. One of the System Forms is shown in Figure 2.5.

2.5 Development of winREMR

The program winREMR was developed using the 16-bit version of VB4 for the front end of the database while using Microsoft Access for the actual database. The following



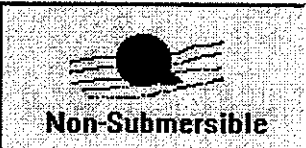
Roller Gate Inspection Form Page 1 Part 1	
Army Corps of Engineers Roller Lock Gate Structure Inspection	Page 1 Part 1
Name of the Civil Work Project	
Lock and Dam # 15	
Location of Civil Work Project (1. Body of Water, 2. Nearest Town)	
Rock Island	
First inspection data of Record	11/20/96
Complete List of Inspection Date(s)	
Date Page 3 was Completed (Normal Operations)	11/20/96
Inspected By:	
Date Pages 4,5,6 were completed:	11/22/96
Condition of inspection	Out of Operation 
Inspected By	
Date Page 7 was completed (Bulheaded)	12/12/96
Inspected By:	
Type of Gate:	Submersible
	
	<input type="button" value="Next Page"/>

Figure 2.2 Typical Data Entry Form

Page 1 Result Page			
Army Corps of Engineers		Page 1	
Roller Lock Gate Structure Inspection			
Name of the Civil Work Project			
<u>Lock and Dam # 15</u>			
Location of Civil Work Project (1. Body of Water, 2. Nearest Town)			
1. <u>Rock Island</u>	2. _____		
Gate identification Number: _____		First inspection data of Record <u>11/20/96</u>	
Complete List of Inspection Date(s)			
Date Page 3 was Completed (Normal Operations)		<u>11/20/96</u>	
Inspected By: _____			
Date Pages 4,5,6 were completed:		<u>11/22/96</u>	
Condition of inspection		<u>Out of Operation</u>	
Inspected By _____			
Date Page 7 was completed (Bulheaded)		<u>12/12/96</u>	
Inspected By: _____			
Type of Gate:	<u>Submersible</u>		
Number of Aprons:	<u>Lower apron only</u>		
Will interior of gate be accessed?	<u>Yes</u>		
Length of Gate (ft.)	<u>152</u>	Design Upper Pool Elevation: (ft.)	<u>125.9</u>
Gate Height (ft.)	<u>25</u>	Design Lower Pool Elevation: (ft.)	<u>113.6</u>
Drum Radius (in.)	<u>26</u>	Sill Elevation: (ft.)	<u>98.4</u>
Rim Radius (in.)	<u>26</u>		
Do you routinely bulkhead the Gate?	<u>No</u>		
If Yes, What year was the gate last bulkheaded?	_____		
Interval period: _____	Construction Date: <u>5/9/55</u>		

Figure 2.3 Typical Data Entry Result Form

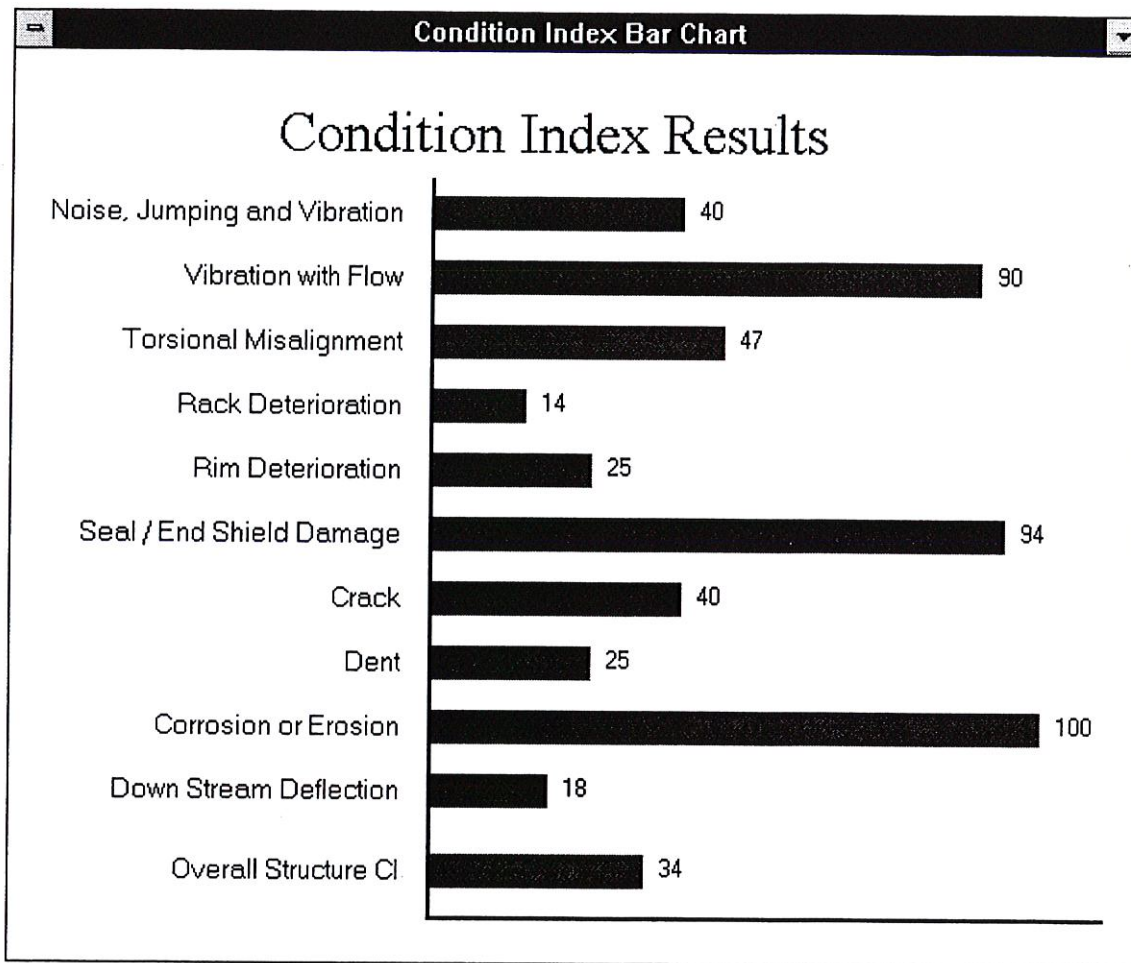


Figure 2.4 Typical Result Form

description of the program's structure assumes that the reader has a working knowledge of VB4 and Access. This Section is intended to give a developer general information on how the program was developed. This discussion is not intended to be a complete discussion of all the components contained within the program.

2.5.1 Front End Development using Microsoft Visual Basic 4.0

winREMR was develop as a Multiple-Document Interface (MDI) application. The MDI application was chosen to give the program the same feel as other MDI program, for

The screenshot shows a window titled "Date Identification" with the following content:

- Header: "Date Identification"
- Instruction: "Choose the appropriate Date"
- List of dates:
 - 1/21/97 (highlighted)
 - 11/20/96
 - 11/22/96
 - 11/24/96
 - 12/16/96
 - 1/8/97
- Buttons:
 - Edit Date
 - New Date
 - Delete Date

Figure 2.5 Typical System Form

example Microsoft Word. It was believed that this would reduce the learning curve for this program.

Like all VB4 application, winREMR contains many different forms with a total of 50 forms. The forms can be divided into five main categories;

- Map Forms
- Data Entry Forms
- Data Entry Result Forms
- Result Forms
- System Forms

The Table 2.1 gives the names of all the forms and the categories that they are contained within winREMR. For the following descriptions of the various categories, *Italics* indicate

Table 2.1 Name of Forms Contained within winREMR

Map Forms	Data Entry Form	Data Entry Display Form	Result Forms	System Forms
frmdiv	frmrg_p1p1	frmrgsp1	frmcichart	frmdate
frmrid1	frmrg_p1p2	frmrgsp2	frmnjvs	frmabout
frmrid2	frmrg_p1p3	frmrgsp3	frmseals	frmstart
frmrid15	frmrg_p2p1	frmrgsp4	frmdents	frmcal
frmncd	frmrg_p2p2	frmrgsp5	frmcrack	frmchangeagateid
	frmrg_p3p1	frmrgsp6	frmrackanch	frmdefault
	frmrg_p3p2	frmrgsp7	frm racks	frmsave
	frmrg_p4p1		frm rail	mdimain
	frmrg_p4p2		frm racktooth	
	frmrg_p4p3		frm rimconn	
	frmrg_p5p1		frm rims	
	frmrg_p5p2		frm rimtooth	
	frmrg_p6p1		frm vfs	
	frmrg_p6p2			
	frmrg_p7p1			
	frmdtseals			
	frmdtdent			
	frmdtcrack			
	frmdtdentint			

tools or forms in the program and `monotype` indicates methods or procedures for the tools or forms.

2.5.1.1 Map Forms

The main purpose of the Map Forms is to enable the user to choose a structure by clicking on *commandbuttons* contained within the map (Figure 2.1). The Map Forms contain a bitmap that displays the map. The *commandbuttons* are used to call the appropriate Form and also relay the user's choice to the database by using the *data* tool. Some of the information from the *commandbutton* clicks are stored in the `gate_default` table in the database. The various tables included in the database are mentioned in Section 2.5.2. The Map Forms are setup so that they are not sizable and cannot be minimized or closed.

2.5.1.2 Data Entry Forms

The Data Entry Forms are used to enter the data into the database (Figure 2.2). They contain many *textboxes* which are linked directly to the database using the *data* tools. They also contain *comboboxes* to aid the user in choosing data. Some the information entered by the user is entered using the *picturebox*. Most of the *pictureboxes* were used to simulate a *commandbutton* with a bitmap on it. This was done by having two different bitmaps for each *picturebox* and using the `onclick` method to toggle between the two different bitmaps. The toggling of these images changes a hidden *textbox* that is linked to the database. The other *pictureboxes* used in the Data Entry Forms contain bitmaps that help clarify the information needed on the Form. When the Form is loaded, the *data* tools are instructed to use the `findFirst` command to locate the first occurrence of the information that the user has chosen. The needed information is read from the `gate_default` table from the database.

2.5.1.3 Data Entry Result Forms

The Data Entry Result Forms contain the same information that the Data Entry Forms contain but do not allow the user to enter or edit the data (Figure 2.3). These Forms are used only to display the information in a matter similar to the field inspection data forms contained within the roller gate report. These Forms are designed to be printed with or without the field inspection data. The contents of these Forms is almost entirely *labels* with a *data* tool to link the *labels* to the database.

2.5.1.4 Result Forms

The Result Forms are designed to display the results from the calculations using bar charts (Figure 2.4). Some of the bars on the bar chart are made using the *shape* tool and

choosing a rectangle while others are made using the *picturebox*. The *picturebox* is used to represent bars on the graph which gives the user the option to click the bar which displays a window with more information. The *shape* tool was used where further information was not available. The main reason for the use of the two different tools was that the *picturebox* had an `onclick` method that could be used to call another Form while the *shape* tools did not. An additional reason for using the two different tools is that the two different tools look slightly different. This should help the user distinguish between the two different types and inform them if more information is available. The values for the bars in the bar charts are found using data from the database and are calculated using the `onactivate` method of the Forms. Some of the values used on the Result Forms are read directly from the database. An example of this is the Form *frmcalCI*. The distress CI's and the combined structure CI come directly from the database without any calculations on this Form.

2.5.1.5 System Forms

This is a broad group of Forms that contains all of the Forms that do not fall into the other four categories. One of these Forms is *frmdate* (Figure 2.6). This Form uses a *dblist* to list all the dates available to the user for the structure and sub-structure that they have identified with the Map Forms. This Form also includes code to delete a date or to add a new date. Some of the other Forms are used to provide information to the user at certain times within the program. Two of these forms are *frmstart* (Figure 2.7) which gives the user the disclaimer and *frmcal* (Figure 2.8) which tells the user how much of the calculations is completed.

Date Identification

Choose the appropriate Date

1/21/97

11/20/96

11/22/96

11/24/96

12/16/96

1/8/97

Edit Date

New Date

Delete Date

Figure 2.6 Form *frmdate*

2.5.2 Database Development using Microsoft Access

The database that contained the data was created in Microsoft Access. Access was chosen because VB4 had features already built in to link to an Access database. This built in ability came through the use of the Microsoft Jet Engine. The database includes 10 tables. A list of the tables and a description of the contents is found in Table 2.2. Each table contains two fields, `gate_id` and `date1`, that are used to determine the structure and the date that correspond to the information as well as the other fields needed to store all the information.

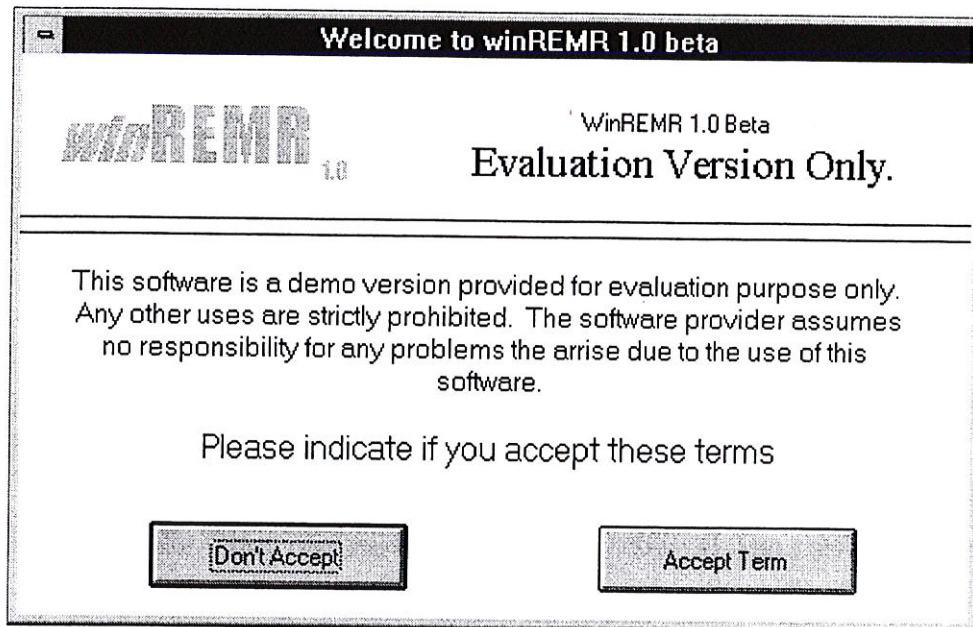


Figure 2.7 Form *frmstart*

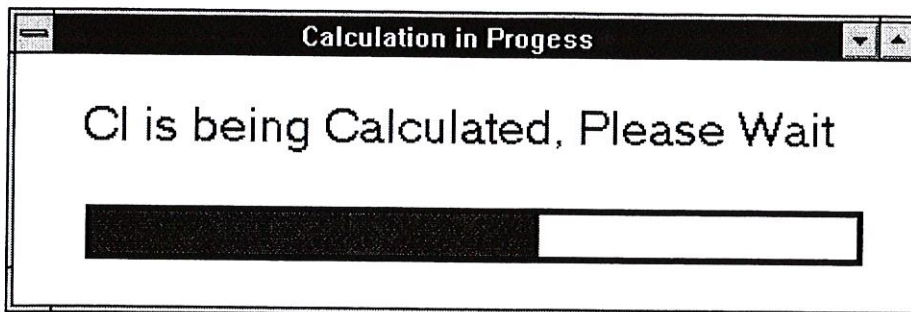


Figure 2.8 Form *frmcal*

Table 2.2 List of Tables Contained Within Database

rg_p1	Contained Data From Page 1 of Data Entry Forms
rg_p2	Contained Data From Page 2 of Data Entry Forms
rg_p3	Contained Data From Page 3 of Data Entry Forms
rg_p4	Contained Data From Page 4 of Data Entry Forms
rg_p5	Contained Data From Page 5 of Data Entry Forms
rg_p6	Contained Data From Page 6 of Data Entry Forms
rg_p7	Contained Data From Page 7 of Data Entry Forms
rgdent	Contained Data Form Dents, Page 5 of Data Entry Forms
rgseal	Contained Data Form Seal Damage, Page 5 of Data Entry Forms
rgcrack	Contained Data Form Cracks, Page 5 of Data Entry Forms
gate_default	Contained Data on Default used by the Program

CHAPTER 3 USERS MANUAL FOR winREMR 1.0

3.1 Introduction

winREMR was developed to serve two different purposes. One is to be a database management system designed to store and retrieve data pertaining to the inspections of roller dam gates. The other purpose is to compute and display the Condition Index (CI) of these dam gates using rules outlined by the roller gate condition assessment report (Greimann et al., 1996). This program is intended to be a possible replacement for the existing software developed for roller gates by ISU.

This manual on winREMR uses different typefaces to represent different items contained within the program. *Italics* are used for the names of Forms displayed on the screen. *Monospace* is used to represent command buttons on the screen, pull down menus, menu commands, or keys on the keyboard. Underlining is used to indicate the start of a step-by-step instruction describing a method to accomplish a specific task.

3.2 Identifying Structure and Inspection Date for Structure

The database contained within winREMR sorts the various records by the structure and date given for the inspection.

3.2.1 Identifying a New or Different Structure

Every inspection has to uniquely identify the structure inspected. This unique structure is used by the database to aid in identifying the different records. The process of choosing a new or different structure is begun by displaying the *Identify Inspection Form Structure* dialog box (Figure 3.1). This dialog box is displayed upon starting winREMR. The option to choose a new or different structure is always available while using the program

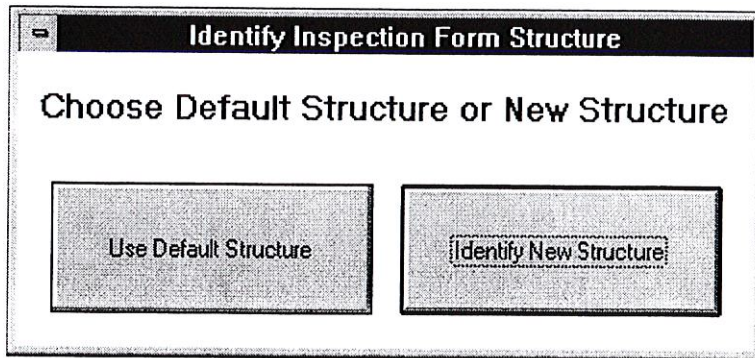


Figure 3.1 Identify Inspection Form Structure Dialog Box

by clicking the New Structure option under the File pull down menus at the top of the program. The *Identify Inspection Form Structure* dialog box contains two buttons. Each button identifies a unique option in identifying a structure. These options are:

- Use a Default Structure - Chooses the Default Structure which has been saved within the program.
- Identifying a New Structure - Identifies a structure other than the Default Structure.

3.2.1.1 Identifying a New Structure

The Identify New Structure button in the *Identify Inspection Form Structure* dialog box enables the user to identify a structure different than the Default Structure. This option also gives the user the option to save a structure as a Default Structure. The Identify New Structure button can be used without effecting or changing the Default Structure. Clicking this button will lead the user through various maps designed to simplify the process of choosing a new structure. Contained within the series of maps is a dialog box which prompts the user to either save the current structure as Default Structure or preserve the old Default Structure.

Using the Identify New Structure Button

1. If the *Identify Inspection Form Structure* dialog box is not showing, chose the New Structure option from the File pull down menu. This will display the *Identify Inspection Form Structure* dialog box.
2. Click on the Identify New Structure button. A window called *Division Map*, displaying the different divisions outlined by the Army Corps of Engineers for their civil works projects, will be displayed to the screen (Figure 3.2).

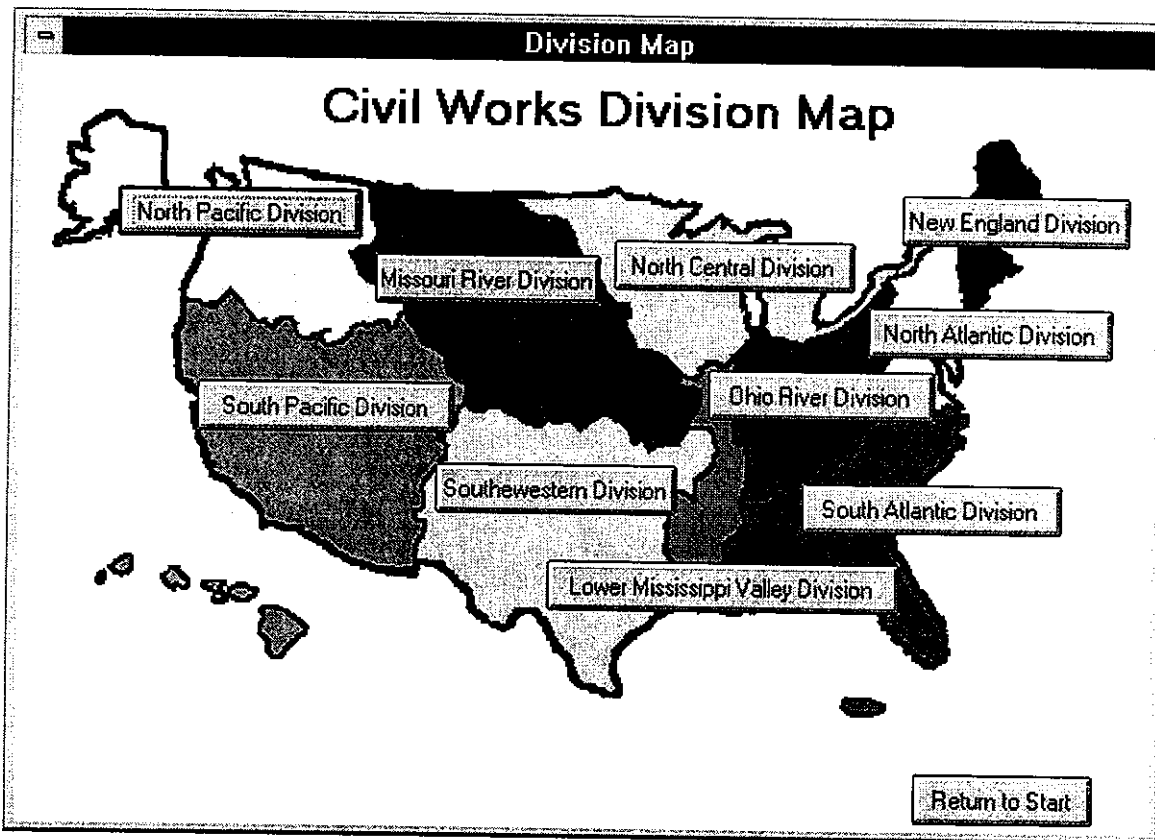


Figure 3.2 Civil Works Division Map

3. Click on the division that contains the structure that you have inspected. A window displaying a map of the districts contained within this chosen division is shown.
For example, if the North Central Division button on the *Division Map* window was clicked, the window *North Central Division* would be displayed to the screen (Figure 3.3). This window displays the districts contained within the North Central Division.
4. Click on the district that contains the structure you will investigate. A window is displayed which contains a map of the civil work navigation projects contained within

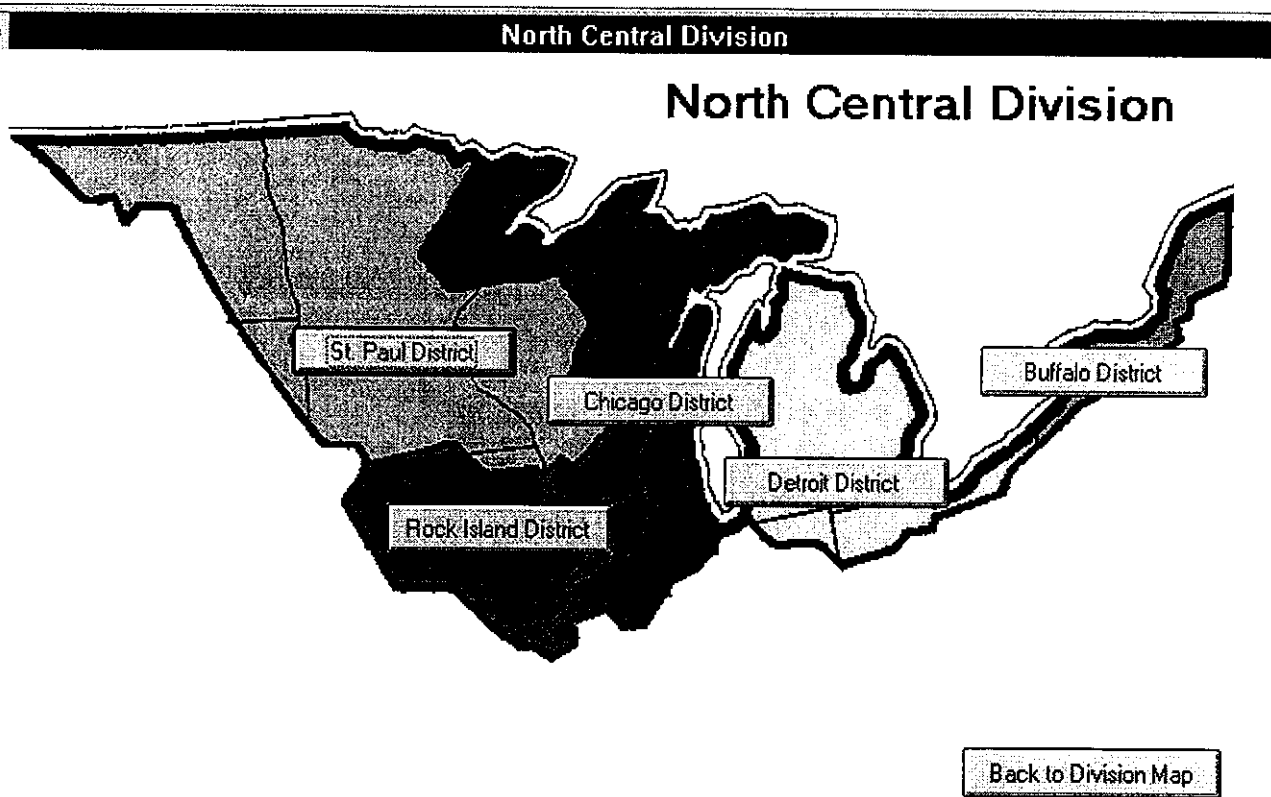


Figure 3.3 North Central Division Map

this district. Due to the number of civil work projects that are possible within one division, some of the maps are displayed in two or more windows. If this is the case, use the *Next Map* and *Last Map* buttons at the bottom of the Form to jump between the different windows.

To follow the example, clicking on the *Rock Island District* contained within the *North Central Division* window would show two maps, *Rock Island District Part 1* (Figure 3.4) and *Rock Island District Part 2* (Figure 3.5). Both maps contain lock and dams on the Mississippi River within the Rock Island District.

5. Click on the structure that you will investigate. A window and a dialog box will be displayed. The window will contain a map of the chosen structure displaying the different sub-structures available on the chosen structure. The *Save Default Structure* dialog box will appear (Figure 3.6). This dialog box gives the option to save the present structure as the Default Structure. If you chose to save the present structure, the *Use Default Structure* button mentioned in Figure 3.1 will load the Default Structure immediately and display the window containing the map shown on the screen under *Save Default Structure* dialog box. If you decide not to save the present structure as the Default Structure, the previously saved Default Structure will remain, if a Default Structure exists. Upon choosing one of the two different options, the dialog box will be removed.

Continuing the example, clicking on the *Lock and Dam 15* button on the *Rock Island District Part 1* window and then clicking on the *No* button on the *Save Default Structure* window will display the window *Lock and Dam # 15* (Figure 3.7). This

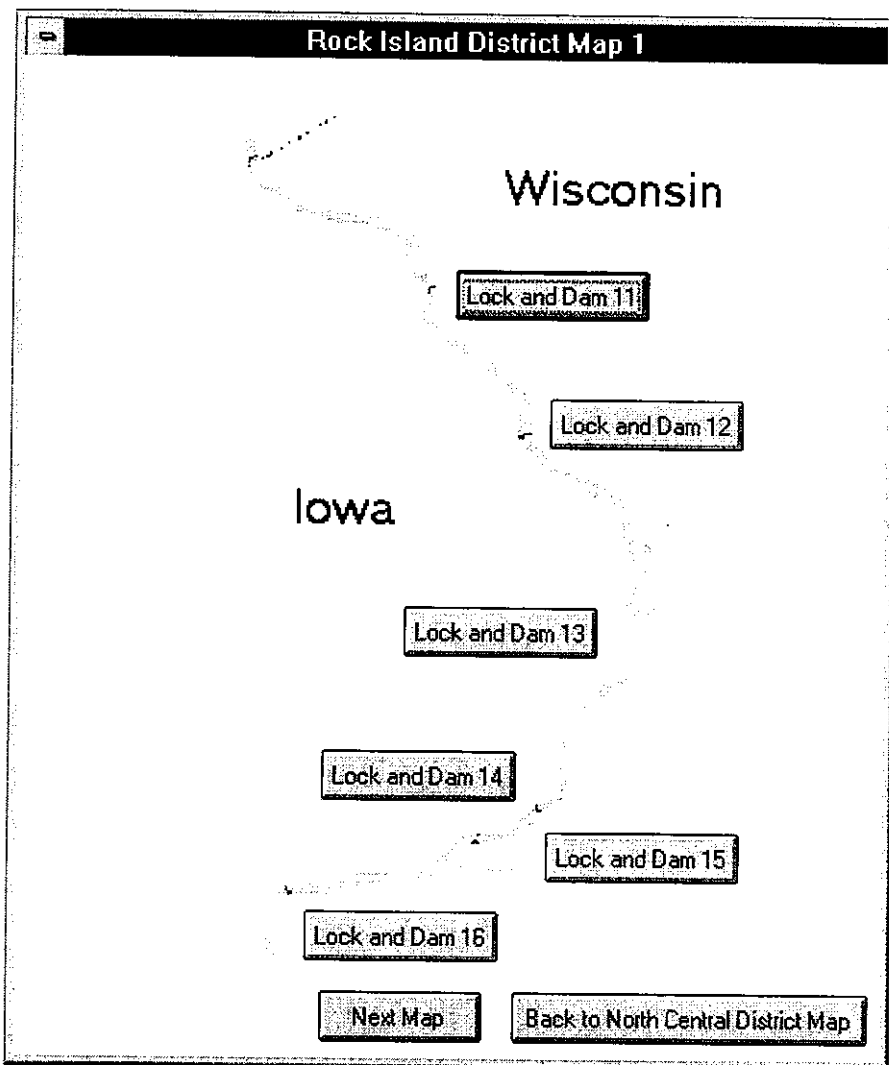


Figure 3.4 Rock Island District Map 1

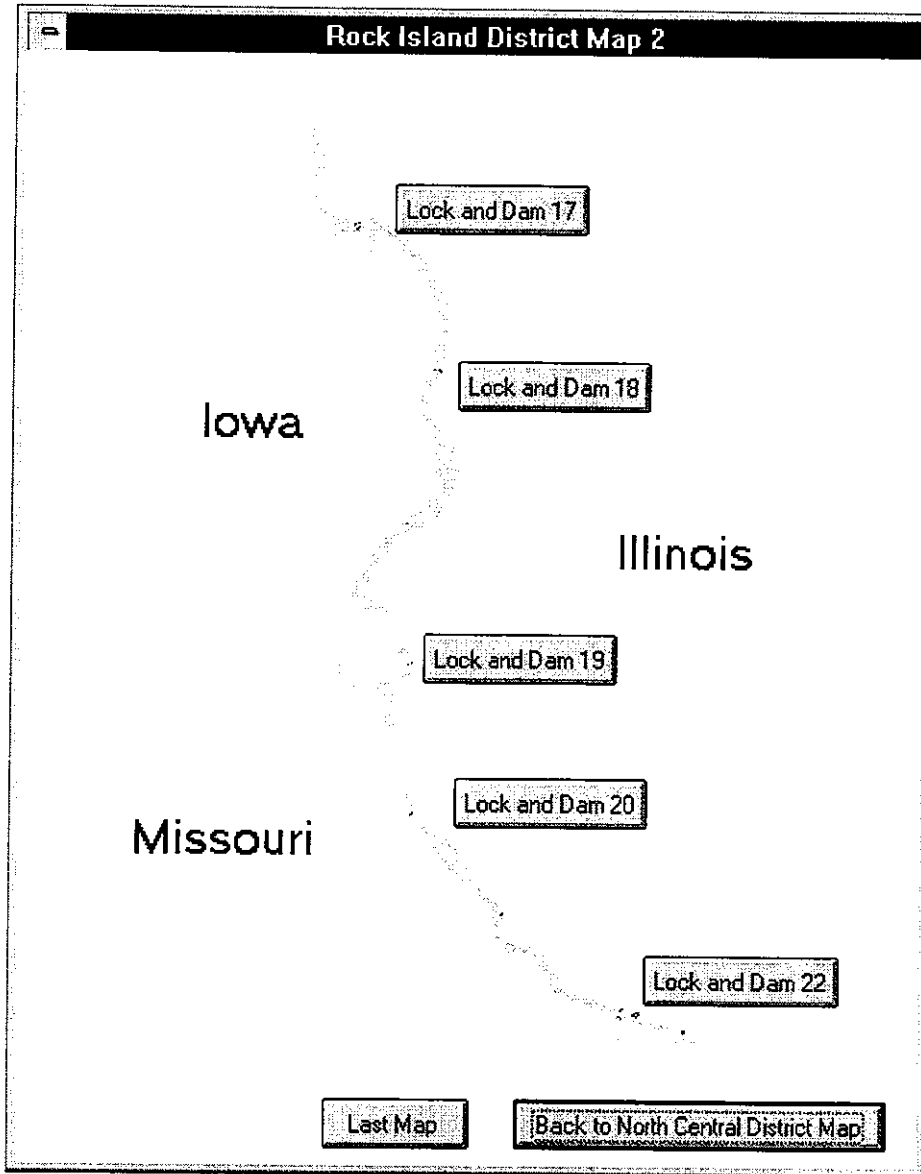


Figure 3.5 Rock Island District Map 2

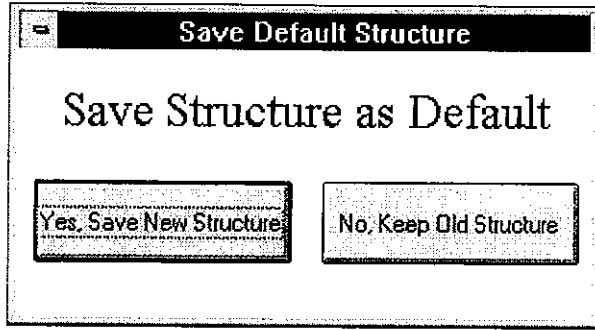


Figure 3.6 Save Default Structure Dialog Box

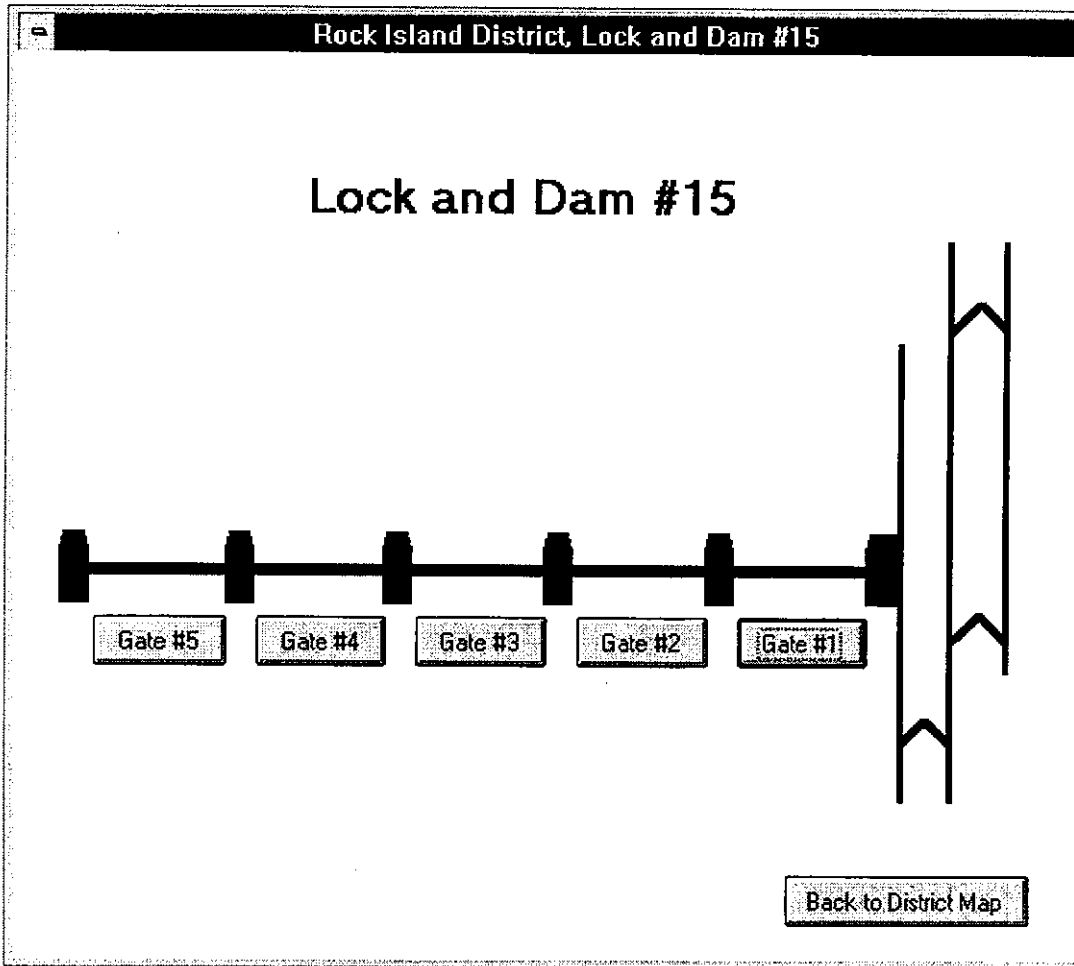


Figure 3.7 Rock Island District Lock and Dam #15 Map

window contains a map of Lock and Dam #15 displaying the different sub-structures available for evaluation.

6. Click on the sub-structure you will investigate. Choosing a sub-structure will display the *Date Identification* window. The usage of this window is explained in Section 3.2.2, Creating, Editing, and Deleting Form. Please refer to this Section of the manual for assistance with this window.

To finish the example, click Dam Gate #3 on the *Lock and Dam #15* window.

You have now specified all the variables required to identify the structure and sub-structure that you will investigate.

3.2.1.2 Using the Default Structure

The Use Default Structure button in the *Identify Inspection Form Structure* dialog box (Figure 3.1) enables the user to access a structure that has previously been saved in the *Save Default Structure* dialog box (Section 3.2.1.1). This option is helpful for users who wish to access the same structure many times. When the text on the Use Default Structure button is dark and clear, this option is available and a structure has been previously saved. In this case, the Default Structure will be identified in the *Identify Inspection Form Structure* dialog box. If the text on the Use Default Structure button seems to be “cloudy”, the Use Default Structure option is not available and the Default Structure will not be identified. This type of text indicates that a Default Structure has not been saved in the program. The Use Default Structure option will not be available the first time the program is used. Since the program automatically determines if a Default Structure exists, there is no need for the user to remember if a Default

Structure exists. To save a structure as a Default Structure, use the Identifying a New Structure (Section 3.2.1.1) option. The choice to save the structure will be available within that option.

Using the Use Default Structure button

1. If the *Identify Inspection Form Structure* dialog box is not showing, click *New Structure* under the *File* pull down menu. This will display the *Identify Inspection Form Structure* dialog box.
2. Click on the *Use Default Structure* button. This will display a window containing a map of the Default Structure showing the different sub-structure contained within the Default Structure.
3. Click on the sub-structure that you will investigate. Choosing a sub-structure will display the *Date Identification* window. The usage of this window is explained in Section 3.2.2, *Creating, Editing, and Deleting Form Dates*. Please refer to this Section of the manual for assistance with this window.

3.2.1.3 Saving a Structure as a Default Structure

Saving a Structure as a Default Structure

1. Proceed to the *Identifying a New Structure* (Section 3.2.1.1) Section of this manual. The process used to save a structure as the Default Structure is given within this Section.

3.2.2 Creating, Editing, and Deleting Form Dates

Every inspection form has a date indicating when the inspection was performed. This date, in addition to the structure, is used by the database to sort the records. The creating,

editing, and deleting of inspection form dates is done by using the *Date Identification* window contained in winREMR (Figure 3.8). This Form is displayed only after a structure and sub-structure have been identified using the methods described in Identifying a New or Different Structure (Section 3.2.1). This window displays the inspection dates saved within the database for the structure and sub-structure that has been selected by the user. Contained

The screenshot shows a window titled "Date Identification" with the heading "Choose the appropriate Date". On the left side, there is a list of dates: 1/21/97 (highlighted), 11/20/96, 11/22/96, 11/24/96, 12/16/96, and 1/8/97. On the right side, there are three buttons: "Edit Date" (positioned next to the highlighted date), "New Date", and "Delete Date".

Figure 3.8 Date Identification Form

within this Form are three options:

- Creating a New Date - Use to create a new date by which to identify data.
- Editing a Previously Existing Date - Use to edit data identified by a previously existing date.
- Delete a Previously Existing Date - Use to delete all data identified by a previously existing date.

3.2.2.1 Creating a New Date

Creating a New Date

1. Make sure that the *Date Identification* window is displayed. If the window is not displayed, chose a structure and sub-structure as described in the Identifying a New or Different Structure (Section 3.2.1) Section of the manual.
2. Enter the new date in the box above the available dates. When the *Date Identification* Form is displayed, the default date in this box is the current date set on your computer. The date should be in the form of “mm/dd/yy”. The mm refers to the numerical representation of the month, dd is the numerical representation of the day of the month, and yy is numerical representation of the year using only the last two digits. For example, June 12, 1995, would be 06/12/95.

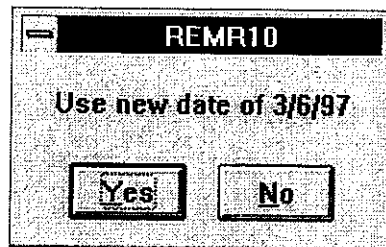


Figure 3.9 Dialog Box for Creating New Date

3. Click the New Date button. A dialog box, show in Figure 3.9, confirms that the creation of a new date will be displayed. This dialog box will contain the date that will be created. If the date shown in the dialog box is not the correct date, click on the No button within the dialog box and return to Step 2.

4. Click the **Yes** button to confirm the date. This will open a Form displaying the first page of the Data Entry Forms. To enter data into this Form, refer to the Filling out a Data Entry Form (Section 3.3.2) Section of this manual. To chose a different Form for editing, refer to the Choosing a Form (Section 3.3.1) Section of this manual.

3.2.2.2 Editing a Previously Existing Date

Editing a Previously Existing Date

1. Make sure that the *Date Identification* window is displayed. If the window is not displayed, chose a structure and sub-structure as described in the Identifying a New or Different Structure (Section 3.2.1) Section of the manual.
2. Click on the date from the list of available dates that identifies the inspection data you will edit. When you click on the date, this date will be displayed in the box above the list of available dates. If the date you want to identify is not displayed in that box, restart Step 2 until the date displayed in the box. If the date that you want to edit is not displayed in the list of available dates, you will have to create a new date. Return to the Create a New Date (Section 3.2.2.1) Section of this manual.
3. Click the **Edit Date** button. This will open a Form displaying the first page of the Data Entry Forms. Some of the data saved within the database will be displayed within the Form. To edit the data in the Form, refer to the Filling out a Data Entry Form (Section 3.3.2) Section of this manual. To choose a different Form for editing, refer to the Choosing a Form (Section 3.3.1) Section of this manual.

3.2.2.3 Deleting a Previously Existing Date

Deleting a Previously Existing Date

1. Make sure that the *Date Identification* Form is displayed. If the Form is not displayed, chose a structure and sub-structure as described in the Identifying a New or Different Structure (Section 3.2.1) Section of the manual.
2. Click on the date that you will delete from the list of available dates. When you click on the date, this date will be displayed in the box above the list of available dates. If the date you want to delete is not displayed in the box, restart Step 2.
3. Click the Delete Date button. A dialog box, shown in Figure 3.10, will be

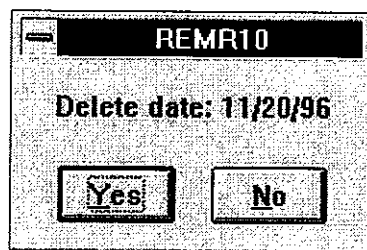


Figure 3.10 Dialog Box For Deleting a Previously Existing Date

- displayed asking if you want to delete the date shown. NOTE: Once a date is deleted, any data filed under that date will be PERMANENTLY lost. If you do not want to delete the date, click the No button in the dialog box.
4. Click the Yes button on the dialog box. The dialog box will be removed and the *Date Identification* window will be shown. The list of available dates shown in the *Date Identification* window no longer contains the deleted date.

3.3 Data Entry Forms, Data Entry Result Forms, and Result Forms

winREMR contains five different types of Forms, but only the Data Entry Forms, Data Entry Result Forms, and Result Forms are discussed in this section. Data Entry Forms are used to enter data into the database while Data Entry Result Forms and Result Forms are used to display information contained within the database. A typical Data Entry Form is shown in Figure 3.11, a typical Data Entry Result Form is shown in Figure 3.12, and a typical Result Form is shown in Figure 3.13. Data Entry Forms also help the user understand what data is needed for an inspection. The Data Entry Form contains different types of buttons, fields, and drop down menus to assist the user in entering data. The Data Entry Result Forms contain the data entered into the database in an uneditable form to assist in printing. The Result Form display the results of the calculations in a graphical method. Once a structure and date have been selected and the Data Entry Forms are displayed, there are many actions the user can take. These actions include:

- Choosing a Form - Displays a different Data Entry Form, Data Entry Result Form or Result Form then the one the user is presently using.
- Filling out a Data Entry Form - Describes how to entering new data and editing previously entered data.
- Saving the Data in Data Entry Form - Enables the data to be saved to the database.
- Opening and Closing Multiple Forms - Displays and closes many different Forms.
- Printing Result Forms - Prints a Result Form with or without data.



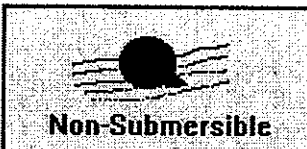
Roller Gate Inspection Form Page 1 Part 1	
Army Corps of Engineers Roller Lock Gate Structure Inspection	Page 1 Part 1
Name of the Civil Work Project	
Lock and Dam # 15	
Location of Civil Work Project (1. Body of Water, 2. Nearest Town)	
Rock Island	
First inspection data of Record	
	11/20/96
Complete List of Inspection Date(s)	
Date Page 3 was Completed (Normal Operations)	11/20/96
Inspected By:	
Date Pages 4,5,6 were completed:	11/22/96
Condition of inspection	Out of Operation 
Inspected By	
Date Page 7 was completed (Bulheaded)	12/12/96
Inspected By:	
Type of Gate:	Submersible
	
	Next Page

Figure 3.11 Typical Data Entry Form

Page 1 Result Page	
Army Corps of Engineers Roller Lock Gate Structure Inspection	Page 1
Name of the Civil Work Project <u>Lock and Dam # 15</u>	
Location of Civil Work Project (1. Body of Water, 2. Nearest Town) 1. <u>Rock Island</u> 2. _____	
Gate identification Number: _____	First inspection data of Record <u>11/20/96</u>
Complete List of Inspection Date(s) Date Page 3 was Completed (Normal Operations) <u>11/20/96</u> Inspected By: _____	
Date Pages 4,5,6 were completed: <u>11/22/96</u> Condition of inspection <u>Out of Operation</u> Inspected By _____	
Date Page 7 was completed (Bulheaded) <u>12/12/96</u> Inspected By: _____	
Type of Gate:	<u>Submersible</u>
Number of Aprons:	<u>Lower apron only</u>
Will interior of gate be accessed?	<u>Yes</u>
Length of Gate (ft.) <u>152</u>	Design Upper Pool Elevation: (ft.) <u>125.9</u>
Gate Height (ft.) <u>25</u>	Design Lower Pool Elevation: (ft.) <u>113.6</u>
Drum Radius (in.) <u>26</u>	Sill Elevation: (ft.) <u>98.4</u>
Rim Radius (in.) <u>26</u>	
Do you routinely bulkhead the Gate?	<u>No</u>
If Yes, What year was the gate last bulkheaded?	_____
Interval period: _____	Construction Date: <u>5/9/55</u>

Figure 3.12 Typical Data Entry Result Form

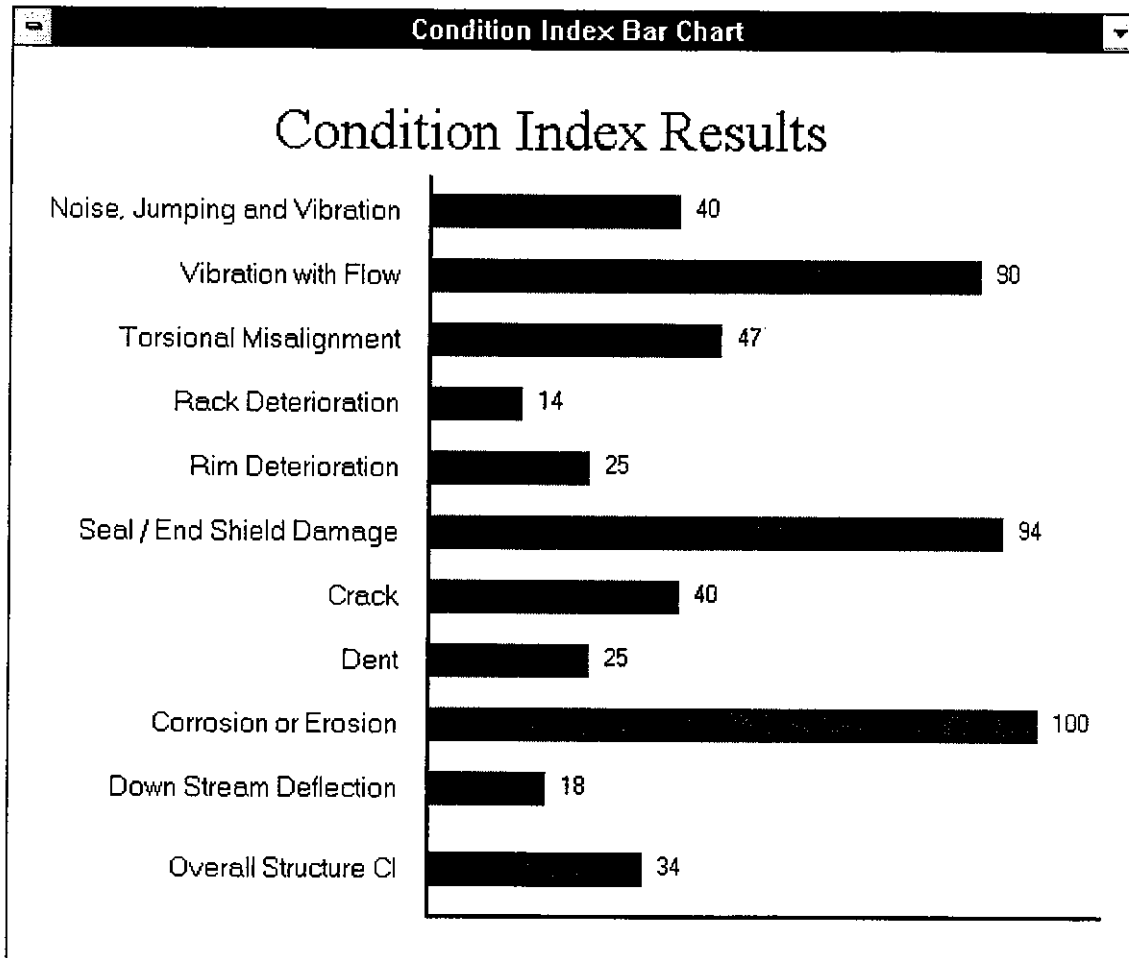


Figure 3.13 Typical Result Form

3.3.1 Choosing a Form

Choosing a Form, whether it is a Data Entry Form, Data Entry Result Form, or a Result Form, is quite important. Since many of the pages of the inspection form are too large to be displayed on one Data Entry Form, these pages of the inspection form are split up into several parts and displayed on several different Data Entry Forms. An example of this is Page One of the inspection form, which is split into three different Data Entry Forms. Each Data Entry Form is identified by a page and part number. The page and part numbers indicate the sequence that the Data Entry Forms must be viewed to be in the same sequence as the inspection forms. winREMR allows the different pages and parts to be accessed in any order. winREMR also allows the Data Entry Forms, the Data Entry Result Forms, and the Result Forms to be accessed at the same time.

Choosing a different Form

1. Make sure that either a Date Entry Form, Data Entry Result Form or a Result Form is showing. If neither type of Form is showing, proceed to the Section on Choosing Structure and Inspection Date for Structure (Section 3.2) in the manual.
2. Click on Pages from the pull down menu.
3. Click on either Data Entry Pages or Result Pages depending on what type of Form you would like to display. All the Data Entry Pages are located under Data Entry Pages while the Data Entry Result Forms and Result Forms are located under Result Pages. Figure 3.14 and Figure 3.15 show some of the menus contained under Data Entry Pages and Result Pages, respectively. Notice that both Data Entry Pages and Result Pages have a triangle to the right of

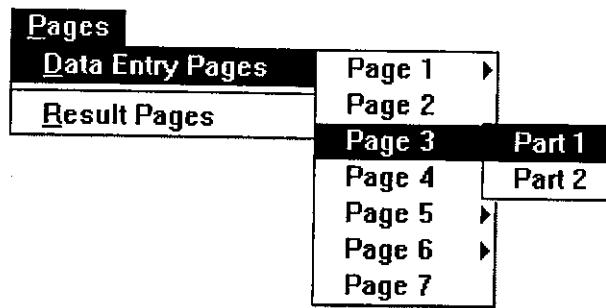


Figure 3.14 Data Entry Pages Sub Menu

the caption. This triangle is a menu indicator that informs the user another menu is contained under this option. For example, click on Data Entry Pages under the Pages pull down menu.

4. Click on the page that you would like to display. If there is a menu indicator to the right of the page that you chose, continue to Step 5. If no menu indicator is present, the Form will be displayed on the screen. To continue the example, click Page 3 on the menu.
5. Click which part of the page you would like to have displayed. This part will now be

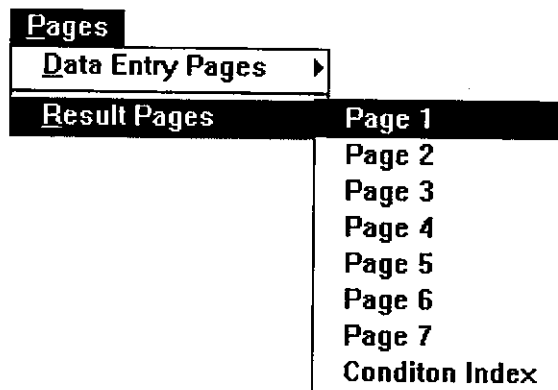


Figure 3.15 Result Pages Sub Menu

displayed on the screen. To finish the example, click on Part 2. Now the Data Entry Form for Page 3, Part 2 is visible on the screen.

3.3.2 Filling out a Data Entry Form

Filling out a Form is necessary for entering and editing the data contained within the database. The data contained within the database can be edited only in the Data Entry Forms. The Data Entry Result Forms and Result Forms do not allow data editing. The Data Entry Forms contain different methods for entering or editing data. These methods are:

- Text Entry Boxes - Use to enter data from the keyboard.
- Buttons - Use to click on an available option.
- Drop down pick lists - Use to choose from a few available options.
- Record Indicators - Used to identify one of many different records.

3.3.2.1 Entering and Editing Data in a Text Entry Box

A typical Text Entry Box is shown in Figure 3.16.

Entering and Editing Data in a Text Entry Box

1. Make sure that a Data Entry Form is showing. Data Entry Forms are the only Forms that contain Text Entry Boxes. If a Data Entry Form is not showing, proceed to the Choosing a Form (Section 3.3.1) Section of the manual.

Name of the Civil Work Project
Lock and Dam # 15

Figure 3.16 Typical Text Entry Box

2. Choose the Text Entry Box in which you would like to either enter data or edit data. This can be done either by clicking on the Text Entry Box or tabbing through the form until the cursor is within the Text Entry Box.
3. If the Text Entry Box is empty, you can now type what information you would like to enter. Press Enter when you are finished.
4. If you are editing data that already exists, the cursor will be contained within the typed text at some point. Use the keyboard to edit the pre-existing text and press Enter when you are finished.

3.3.2.2 Using Buttons on Data Entry Forms

A typical button is shown in Figure 3.17.

Using Buttons on Data Entry Forms

1. Make sure that a Data Entry Form is showing. If a Data Entry Form is not showing, proceed to the Choosing a Form (Section 3.3.1) Section of the manual.
2. Click on the Button that indicates the desired value. White backgrounds for Buttons indicate the values that are currently selected, while gray backgrounds indicate the values which are not selected.



Figure 3.17 Typical Button

3.3.2.3 Using Drop Down Pick Lists on Data Entry Forms

A typical Drop Down Pick List is shown in Figure 3.18.

Using Drop Down Pick Lists on Data Entry Forms

1. Make sure that a Data Entry Form is showing. Data Entry Forms are the only Forms that contain Drop Down Pick Lists. If a Data Entry Form is not showing, proceed to the Choosing Form (Section 3.3.1) Section of the manual.



Figure 3.18 Typical Drop Down Pick List

2. Click on the down arrow to the right of the Drop Down Pick List. A list of available options will be shown below the Drop Down Pick List box.
3. Click on the appropriate value. The Drop Down Pick List options disappears and the chosen value is displayed in the Drop Down Pick List box.

3.3.2.4 Using Record Indicators on Data Entry Forms

A typical Record Indicator is shown in Figure 3.19. This is used to choose one of many different records contained within the database. One of the sections where this is used is in the seal damage distress which could have many different entries.

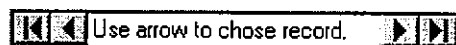


Figure 3.19 Typical Record Indicator

Using Record Indicators on Data Entry Form

1. Make sure that a Data Entry Form is showing. Data Entry Forms are the only Forms that contain Record Indicators. If a Data Entry Form is not showing, proceed to the Choosing Form (Section 3.3.1) Section of the manual.
2. Click on the arrows contained on the Record Indicator to change the record. Use the right arrow to go up a record and the left arrow to go down a record. The Text Entry Boxes on the Forms containing the Record Indicator will display the different records.

3.3.3 Saving Data on a Data Entry Forms

Saving data is necessary to retain the data after the program is terminated. Data can only be saved from within the Data Entry Forms. There are two different options for saving data:

- Saving Data on only one Data Entry Form
- Saving Data on All Data Entry Forms

3.3.3.1 Saving Data on Only One Data Entry Form

Saving Data on Only One Data Entry Form

1. Make sure that the Data Entry Form containing the data that you desire to save is showing and is active. You can make sure that the Data Entry Form is active by clicking anywhere within the Form. If a Data Entry Form is not showing, proceed to the Choosing a Form (Section 3.3.1) Section of the manual.
2. Click the File pull down menu. Figure 3.20 shows the contents of the File sub menu.

File
New Structure
Calculate Condition Index
S ave S ave All
P rint P rint All
C lose E xit

Figure 3.20 File Sub Menu

3. Click the Save option under the File menu. The data on the Form is now saved to the database.

3.3.3.2 Saving Data on All Data Entry Forms

Saving Data on All Data Entry Forms

1. Make sure that a Data Entry Form is showing. If a Data Entry Form is not showing, proceed to the Choosing a Form (Section 3.3.1) Section of the manual.
2. Click the File pull down menu.
3. Click the Save All option under the File menu. The data contained within all the Data Entry Forms is now saved to the database.

3.3.4 Opening and Closing Multiple Forms

Opening and closing multiple Form is similar to choosing different Forms. This enables the user to view several different Forms at the same time. This option applies to both Data Entry Forms and Result Forms.

3.3.4.1 Opening Multiple Forms

Opening Multiple Forms

1. Use the same procedure outlined in the Choose a Form (Section 3.3.1) Section of the manual.
2. Repeat Step 1 until all the Forms you require are open.

3.3.4.2 Closing Multiple Forms

Closing Multiple Forms

1. Click the Form that you will close. If no Forms are showing, proceed to the Choosing a Form (Section 3.3.1) Section of the manual.
2. Click the File pull down menus.
3. Click the Close option under the File menu. This will close the Form that you have chosen. If the Form was a Data Entry Form, the program will check to see if all the data on that Form was saved and will ask the user to save the data prior to closing the Form. Figure 3.21 shows a typical dialog box to prompt the user to save the Form's data.
4. Repeat Steps 1-3 until all the required Forms are closed.

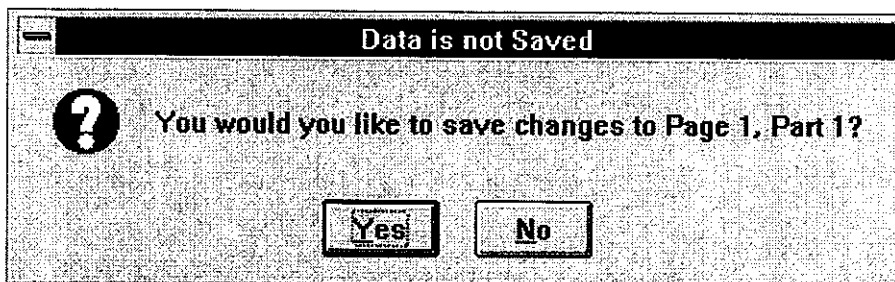


Figure 3.21 Typical Save Data Dialog Box

3.3.5 Printing Result Forms or Data Entry Result Forms

Printing a Form is available so that the user may obtain a hard copy of the data shown in a Data Entry Result Form or Result Form. Only Result Forms and Data Entry Result Forms can be printed. winREMR gives the user two different options for printing the Result Forms and Data Entry Result Forms:

- **Printing Only One Result Form or Data Entry Result Form-** Use to print only one Result Form or Data Entry Result Form to the printer.
- **Printing All the Result Forms and Data Entry Result Forms -** Use to print all the Result Forms and Data Entry Result Forms to the printer.

3.3.5.1 Printing Only One Result Form or Data Entry Result Form

Printing Only One Result Form

1. Click the Form that you will print. If no Result Forms or Data Entry Result Forms are showing, proceed to the Choosing a Form (Section 3.3.1) Section of the manual.
2. Click the File pull down menu.
3. Click the Print option in the File pull down menu. The Result Form or Data Entry Result Form is now sent to the Print Manager within Windows and will soon be printed on a specified local printer.

3.3.5.2 Printing All the Result Forms and Data Entry Result Forms

Printing All the Result Forms and Data Entry Result Forms

1. Make sure that a Result Form or Data Entry Result Form is showing. If no Result Forms or Data Entry Result Forms are showing, proceed to the Choosing a Form (Section 3.3.1) Section of the manual.

2. Click the `File` pull down menu.
3. Click the `Print All` option in the `File` pull down menu. All the `Result Forms` and `Data Entry Result Forms` are now sent to the print manager within windows and will soon be printed on a specified local printer.

3.4 Calculating and Viewing CI

One of the main features of winREMR is the ability to calculate and view the CI of the structure. The calculations of the CI's are done only within the `Data Entry Forms`, while the CI's can be viewed in both the `Result Forms` and the `Data Entry Forms`. The `Result Forms` contain graphs showing the `distress CI`, the `combined CI`, and also `sub-distress CI's`, while the `Data Entry Forms` only contain the numerical values for the `distress CI's`. The options available concerning the CI's are as follows:

- **Calculating and Viewing CI in Data Entry Forms** - The CI is calculated for only the `distresses` contained within the `Data Entry Form` and the numerical value is shown directly on the `Data Entry Form`.
- **Viewing CI in the Result Forms** - The CI's are calculated for the `distresses`, `sub-distresses`, and `structure` and are graphed on the `Result Form`.

3.4.1 Calculating and Viewing CI in Data Entry Forms

Calculating the CI is available only within the `Data Entry Forms`. This calculation is needed to achieve the CI of the `structure`, the `individual distresses` and the `sub-distresses`. The CI can be calculated two different ways:

- Calculates the CI for only the `distresses` listed on the active `Data Entry Form`.

- Calculates the CI for all distresses on all Data Entry Forms.

Regardless of the method used for calculation, the CI's are displayed at the bottom of the Form that contains the distress inspection information.

3.4.1.1 Calculating CI for Distresses Listed on Active Data Entry Form

Calculating CI for the Distress on Active Data Entry Forms

1. Make sure that the Data Entry Form that contains the distress of interest is showing on the screen and is active. To make sure that the Form is active, click on the Form. If the Data Entry Form is not showing, proceed to the Choosing a Form (Section 3.3.1) Section of this manual.
2. Make sure that all the necessary information is entered into the Data Entry Form. If not all the information is not entered, some of the distresses CI's will not be calculated correctly if at all. Any CI that does not have all the necessary information will receive a value of 0.
3. Click on the Calculate CI button on the bottom of the Form. Figure 3.22 shows a typical data entry form with the Calculate CI button. The CI is now calculated for the distresses contained on this Form.

3.4.1.2 Calculating CI for all the Distresses on all Data Entry Forms

Calculating CI for all the Distress on all Data Entry Forms

1. Make sure that at least one Data Entry Form is showing on the screen. If a Data Entry Form is not showing, proceed to the Choosing a Form (Section 3.3.1) Section of this manual.







Roller Gate Inspection Form Page 3 Part 1				
U.S. Army Corps of Engineers		Page 3 Part 1		
Roller Dam Gate Structure Inspection				
Inspection Date	<input type="text"/>	Inspected By	<input type="text" value="Me"/>	
Current Upper Pool Elevation	<input type="text" value="165.3"/>			
Current Lower Pool Elevation	<input type="text" value="120.96"/>			
<u>Open and Closing of the Gate</u>				Gate Opening
Noise  No	 Yes	Normal?	No	Yes
				<input type="text" value="2"/>
Jumping  No	 Yes	Normal?	No	Yes
				<input type="text" value="5"/>
Vibration  No	 Yes	Normal?	No	Yes
				<input type="text" value="6"/>
<u>Vibration with Flow</u>				
Vibration Level	<input type="text" value="1"/>	<input type="text" value="2"/>	<input type="text" value="3"/>	<input type="text" value="4"/>
<input type="text"/>				
Can vibrations be eliminated by gate adjustment?		No	Yes	
<u>Condition Index</u>				
Noise Jump and Vibrations	<input type="text" value="40"/>			
Vibration with Flow	<input type="text" value="90"/>			
			<input type="button" value="Calculate CI"/>	<input type="button" value="Page 3 Part 2"/>

Figure 3.22 Typical Data Entry Form with Condition Index Information

2. Make sure that all the necessary information is entered into all the Data Entry Forms. If not all the information is entered, some of the distress CI's will not be calculated correctly if at all. Any CI that does not have all the necessary information will receive a value of 0.
3. Click on the File pull down menu.
4. Click on the Calculate Condition Index option under the File menu. The CI is now calculated for all the Data Entry Forms.

3.4.1.3 Viewing CI in Data Entry Forms

Viewing CI in Data Entry Forms

1. Calculate CI as outlined in Section 3.4.1.1, Calculate CI in Data Entry Form.
2. Click on the Data Entry Form that contains the distress in which you are interested. If the Form is not available, open the Form using the method described in Section 3.3.1, Choosing a Form.
3. The CI for that distress will be shown at the bottom of the Data Entry Form. Figure 3.22 shows a typical Data Entry Form with the CI shown.

3.4.2 Viewing CI in the Result Forms

Viewing the CI in the Result Forms gives the user a graphical interpretation of the CI. It also gives all the CI's on one Form, called *Condition Index Bar Chart*, unlike the Data Entry Forms. A typical *Condition Index Bar Chart* is shown in Figure 3.23. The Result Forms also contains information on the sub-distress CI's to help the user understand what aspects of a distress affected the distress CI and how much each sub-distress affects the

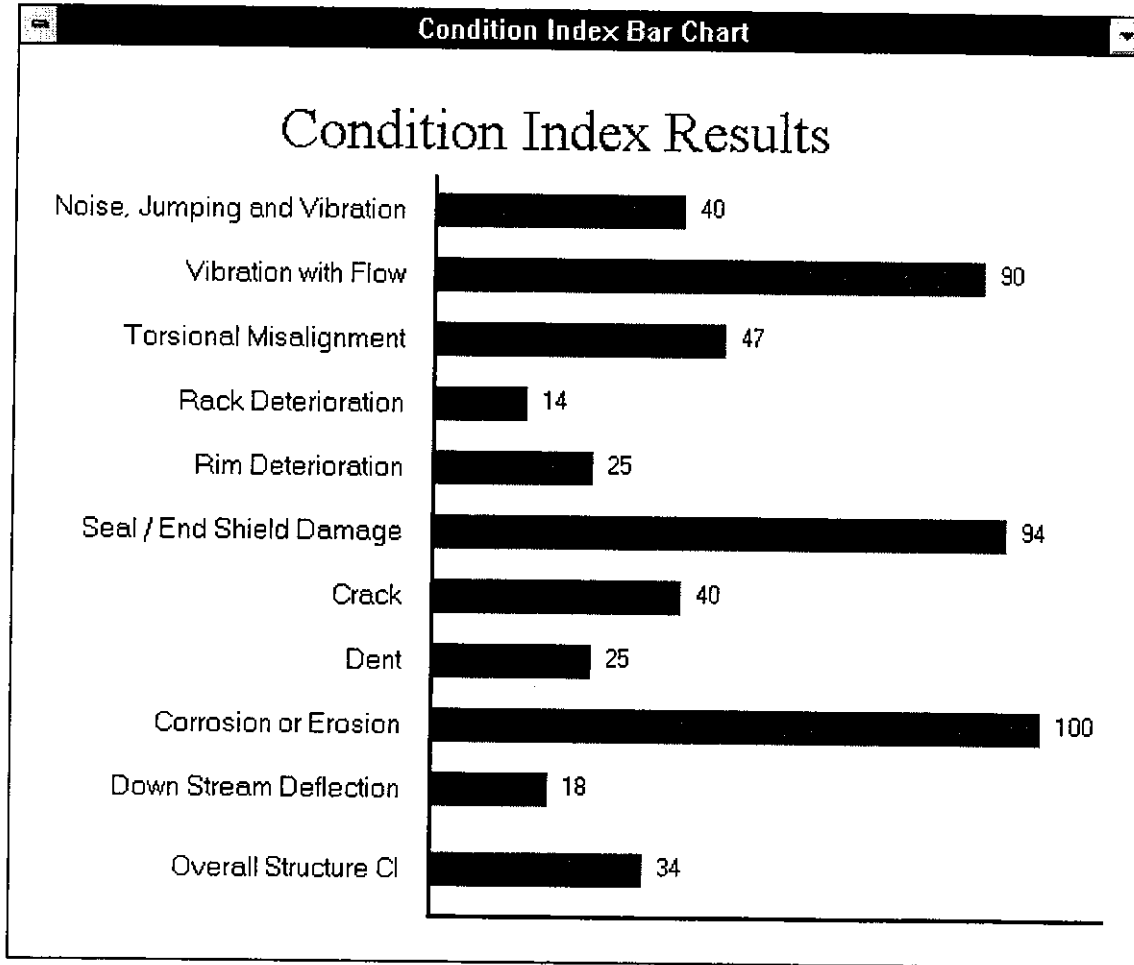


Figure 3.23 Typical Condition Index Bar Chart

distress CI. A typical sub-distress Result Form is shown in Figure 3.24. This will give the user information on how best to address repairs and maintenance of the structure.

3.4.2.1 Viewing Component CI in the Result Form

Viewing the component CI in the Result Forms is very similar to the Choosing a Form (Section 3.3.1) Section of the manual

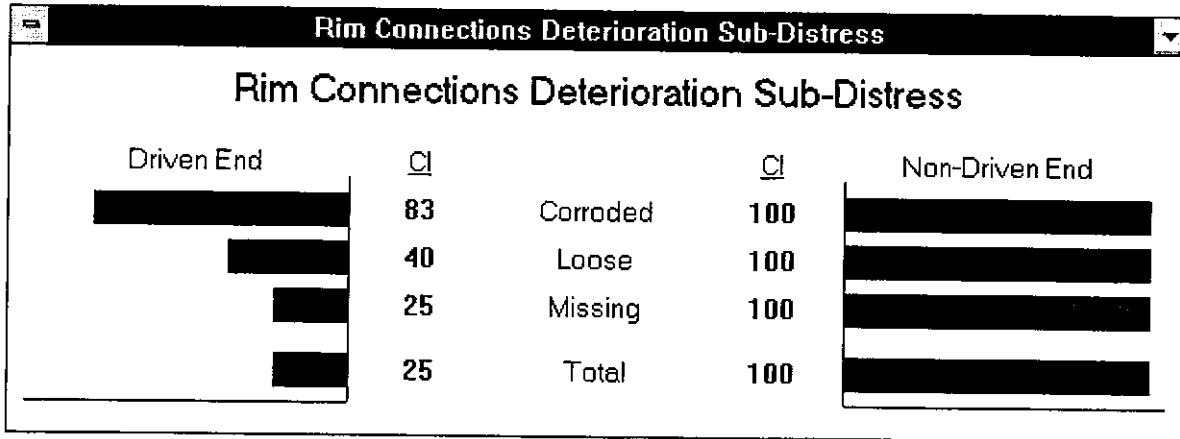


Figure 3.24 Typical Sub-Distress Condition Index Result Form

Viewing the Component CI in the Result Form

1. Make sure that either a Data Entry Form or a Result Form is showing. If neither Form is showing, proceed to the Section Choosing Structure and Inspection Date for Structure (Section 3.2).
2. Click the Pages menu from the pull down menu.
3. Click the Result Forms option in the Pages menu.
4. Click the Condition Index option under the Result Forms option. A graph showing the CI for the various distresses will be shown.

3.4.2.2 Viewing the Sub-Component CI using the Result Form

Viewing the Sub-Component CI using the Result Form

1. Make sure that the *Condition Index Bar Chart* Form is showing. If this Form is not showing, use the Viewing the Component CI in the Result Forms (Section 3.4.2.1) Section of the manual.

2. Click on the bar in the graph that represent the distress in which you are interested. A small Form showing the sub-distress CI will be shown.

CHAPTER 4 INSTALLATION GUIDE FOR winREMR 1.0

4.1 Hardware Requirements and Recommendations

The minimal hardware requirements to run winREMR is as follows:

- 386DX/25 processor or Higher
- 4 Meg Memory
- Windows 3.1 (with DOS 5.0 or Later)
- 15 Meg. Free on Hard Disk
- VGA or higher-resolution monitor
- Microsoft Mouse or compatible pointing device

The suggested hardware is as follows:

- 486DX Processor or Higher
- 8 Meg Memory
- Windows 3.1 (with DOS 5.0 or Later)
- 25 Meg Free on Hard Disk
- VGA or higher-resolution monitor
- 1024 x 768 Screen Resolution
- Microsoft Mouse or compatible pointing device

4.2 Installation Instructions for winREMR

The following gives instructions for installing winREMR with Windows 3.1 or Windows 3.11.

1. Insert Disk 1 of the installation set into your floppy drive.
2. In the Windows Program Manager, choose the **Run** menu command under the **File** drop down menu. The "Run" dialog box will be displayed.
3. In the "Command Line" field, type "A:\SETUP.EXE" where "A" is assumed to be your floppy drive (Figure 4.1). If your floppy drive is not "A", substitute the appropriate letter for your floppy drive for the "A".

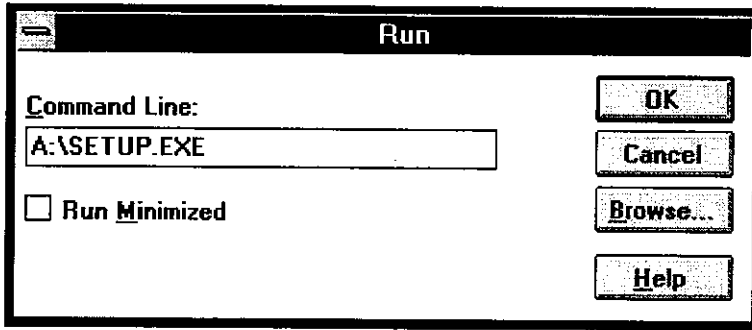


Figure 4.1 Run Dialog Box

4. Click the **OK** button or press return. This will start the installation program.
5. Insert Disk 2 of the installation set when prompted and press the **OK** button.
6. The installation program will then prompt you for a directory in which to install winREMR (Figure 4.2). For most situations, the default directory will be sufficient.
7. Insert Disk 3 of the installation set when prompted and press the **OK** button.
8. All the files needed to run winREMR will now be installed in the indicated directory. The installation program will then create a program group and icon for winREMR. Access winREMR by double-clicking on the winREMR icon.

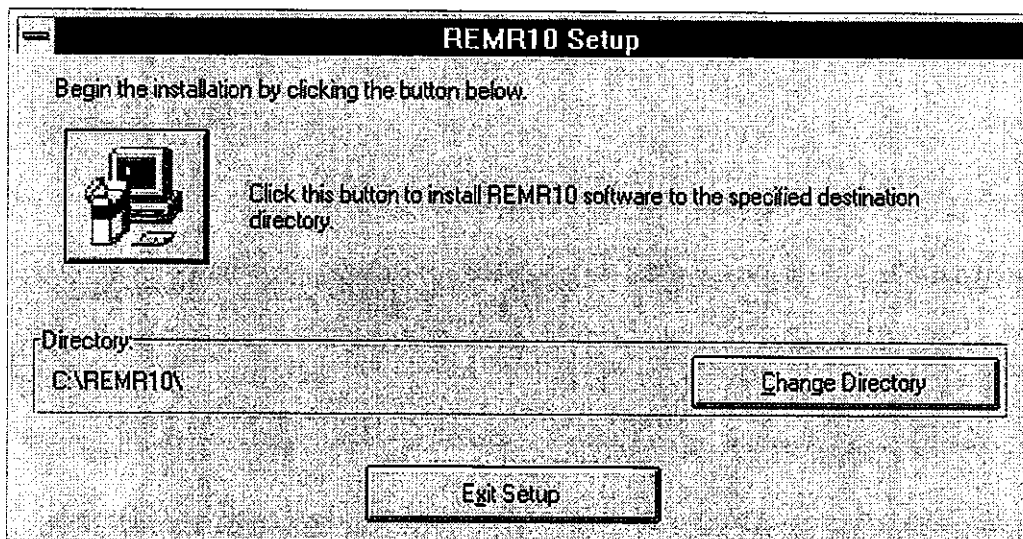


Figure 4.2 Directory Dialog Box From Installation Program

CHAPTER 5 NEW FEATURES IN winREMR

5.1 Introduction

In Chapter 2, the requirements for developing winREMR were stated. These requirements gave the developer a guideline for the structure of the program. This chapter states the new features that are contained within winREMR and how they differ from the DOS version of REMR. Although most of the features mentioned in this chapter are requirements also listed in Chapter 2, some other additional features were implemented during the development of winREMR. This chapter is included to outline what changes were implemented and where these changes were made.

5.2 Changes to the Chose the Inspection Structure Method

One of the most notable changes in winREMR is the method used for choosing the structure and sub-structure. winREMR contains a series of maps that are used to identify the structure and the sub-structure. This method differs greatly from the pick lists used in the old version of REMR. This method also gives the user a more intuitive way to chose the structure and sub-structure.

Another change utilized in winREMR is the ability to step back one or two steps in the selection of the structure without having to start over the selection process. This ability makes the program less frustrating for the user who might click on the wrong location on a map.

A third change utilized in winREMR is the ability to store a default structure within the program. This is especially useful for users that only investigate one structure with the

software. This option is used to save a structure but will not save a sub-structure contained within the structure.

5.3 Changes in the Data Entry Forms

One of the more notable changes contained within the Data Entry Forms is the graphics used to clarify the Forms. Some of these graphics are drawings that clarify a measurement while others are buttons that contain pictures to assist in clarifying the button action. The buttons also assist in simplifying the data entry process.

Another important aspect of winREMR is the ability to view more than one Form at a time. This is especially useful for users that view the inspection data at some date after the data was entered. In addition to viewing more than one Data Entry Form at a time, the user will also be able to view the Result Forms at the same time as the Data Entry Forms.

A third change that was implemented into winREMR was the ability to access the Data Entry Forms in any order. Although this is not much different from the DOS version, this version does allow the user to access Part 2 or Part 3 of a certain page without requiring that Part 1 of the same page be accessed first.

5.4 Changes to the Condition Index Calculations

One of the fundamental changes in winREMR was the location of the calculation routines. Unlike in the DOS version of REMR, the calculation routines are included in the DBMS and are not in a separate computational module. This fundamental change provides advantages to winREMR.

One of the advantages is the ability to calculate a distress CI immediately after all the data for that distress has been entered. This advantage can be utilized at the bottom of some

of the data entry pages. The Calculate Condition Index button on the bottom of the Data Entry Form calculates the CI for only the distresses listed on that page. This is useful for providing information to the user in a quick fashion. This option does not require that data for any other distresses be entered. Although the CI can be found for each distress on the Result Forms, the individual distress CI's do not have to be calculated at the level of the Result Form in order to find the overall structure CI. All the distress CI's are calculated at the time the overall structure CI is calculated.

Another advantage to having the calculation routines included in the DBMS is that the user does not have to exit the Data Entry Forms to calculate the CI for the entire structure. The combined CI can be calculated with any number of Data Entry Forms and Result Form displayed.

5.5 Changes to the Result Forms

The most noted change on the Result Forms is the graphs that display the CI's of the various components and distresses within the structure. This will enable the user to evaluate the various distresses CI's quickly and also provides information on which distress most affect the combined CI. These graphs are also used to show the CI's of the components contained within the distresses.

The second major change is the availability of the components within the distress CI's. This is especially useful to the user that desires to know what component of a distress most affects the distress CI's. This will assist the users in developing a plan to maintain or repair the structure.

A third change is the ability to view more than one Result Form at a time. This is important to fully understand what components are most affecting the CI's. In addition to viewing more than one Result Form at a time, the user can also view the Result Forms with the Data Entry Forms. This provides the user with the ability to compare the data entered in the Data Entry Forms to the results given in the Result Form.

One other change is the ability to print out a copy of the Data Entry Form filled with inspection data or without the inspection data. Printing the Forms with the data gives the user a method to obtain a hard copy of the data, while printing the Form without the data gives the user the ability to obtain a blank inspection form.

CHAPTER 6 SUMMARY AND RECOMMENDATIONS

6.1 Summary

winREMR was developed to increase the usefulness of the DBMS that was developed for roller dam gates (Greimann et al., 1996) by implementing various new features that the DOS version does not contain. Of all the new features included in winREMR, some were identified before winREMR was begun while others became apparent as winREMR was developed. One of the more important features is the inclusion of the calculational module within the DBMS. Another important feature is the ability to view the Data Entry Form, the Data Entry Result Forms, and the Result Forms all at once.

winREMR is a Windows based program which gives the user more freedom than the DOS version. The look of winREMR is similar to other Window's programs which should diminish the time needed to learn it. Since this program is a Windows program, the user can utilize Windows multi-tasking feature.

To diminish the learning curve of winREMR, a user's manual for winREMR has been included. It was written to be a stand alone manual for the user of winREMR and will lead the user through all the important aspects of the program. The user's manual provides instruction for choosing a structure and sub-structure, identifying a date, filling out the Data Entry Forms, viewing the Data Entry Result Forms and the Result Forms, saving data to the database, and printing the Data Entry Result Forms and the Result Forms. In addition to a user's manual, an installation manual has also be included.

winREMR meets and exceeds all the requirements that were prescribed and provides a useful and easy to use DBMS. In the opinion of the developer, winREMR should be considered as a tool for utilization by the U.S. Army Corps of Engineers.

6.2 Recommendations

Like with every program, winREMR does have areas that could be improved. Possible areas of improvements have been revealed after winREMR was completed. Some of the major potential improvements are listed below.

The new version of winREMR should be integrated to a Geographical Information System (GIS) to assist in the selecting of the structure and sub-structure. A GIS will also provide the user with a method for visualizing information on a number of different structures. For example, a GIS system could be used to display all the roller gates on the Mississippi River that have an overall CI of less than 50.

A GIS would only be possible if winREMR used a client/server database system. Such a system would have one database on a single server to which all copies of winREMR would connect. The present version of winREMR contains a database on each computer on which it is installed. The client/server system would provide that information to the GIS system for the entire country including information on all the structures and sub-structures. A user would not have to be at one specific computer to access information on one specific structure.

This database in the client/server system should also be a Structure Query Language (SQL) server. This SQL server is needed to interact with the GIS system correctly. The SQL server must be able to interact with VB.

Other structures investigated by ISU and others should also be included in winREMR.

This would give the user the ease of having only one program to investigate all the sub-structures contained within a specified structure.

APPENDIX 1 ROLLER GATE COMPUTATIONAL MODULE

A1.1	Introduction	66
A1.2	Software Usage.....	66
A1.2.1	Compute Conditional Index	67
A1.2.2	Display Summary Report	67
A1.2.3	Print Summary Report.....	68
A1.2.4	Return to Data Management Menu	68
A1.2.5	“What If” Scenario	68
A1.2.5.1	Change Field Inspection Data	69
A1.2.5.2	Compute New CI Values.....	69
A1.2.5.3	Display New Summary Report.....	70
A1.2.5.4	Print New Summary Report	70
A1.2.5.5	Return to Previous Menu.....	70
A1.3	Software Structure.....	70
A1.3.1	Rollmain.pas.....	71
A1.3.1.1	<i>Main_prog_Menu</i>	71
A1.3.1.2	<i>Cal_ci</i>	71
A1.3.1.3	<i>Sum_report</i>	71
A1.3.1.4	<i>Print_report</i>	72
A1.3.1.5	<i>End_main</i>	72
A1.3.1.6	<i>Whatif_main_menu</i>	72
A1.3.1.7	<i>Whatif_dis_menu</i>	72
A1.3.1.8	<i>Whatif_cal_ci</i>	72
A1.3.1.9	<i>Whatif_sum_report</i>	73
A1.3.1.10	<i>Whatif_print_report</i>	73
A1.3.2	Rollcal.pas	73
A1.3.3	Rollwif.pas & Rollwif2.pas.....	73
A1.3.4	Rollwsum.pas	74

A1.1 Introduction

The roller gate computational module (RGCM) described in the following appendix was developed to accompany a report developed for the Army Corps of Engineers on Condition rating procedure for roller gates (Greimann et al., 1996). RGCM was developed to work in conjunction with Repair, Evaluation, Maintenance, and Rehabilitation Research Program (REMR) Management Systems' Inspection and Rating Data Management Software (Greimann et al., 1994). Although RGCM can operate as a stand alone unit, it was never intended to work this way and should always be used with the REMR database management system (DBMS). RGCM is another in a line of similar programs that have been developed for the Army Corps of Engineers. The following discussion covers only the RGCM and assumes a working knowledge of the REMR DBMS. For further information on REMR DBMS, see REMR Management Systems' Inspection and Rating Data Management Software (Greimann et al., 1994).

A1.2 Software Usage

RGCM was developed to work with the REMR DBMS (Greimann et al., 1994). Choosing "Transfer Data to Evaluation Module" in the REMR DBMS edit menu will start the RGCM(Figure A1.1). The main menu of RGCM will then be displayed, thus giving the user five options to choose from(Figure A1.2). These options are:

- Compute Conditional Index
- Display Summary Report

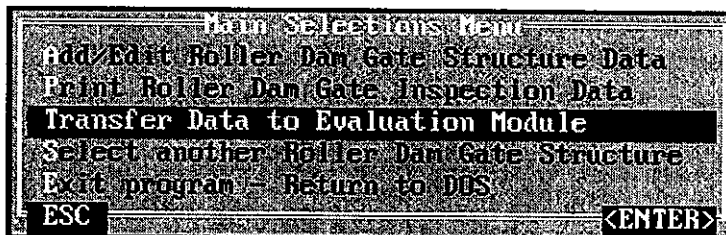


Figure A1.1 REMR DBMS Edit Menu

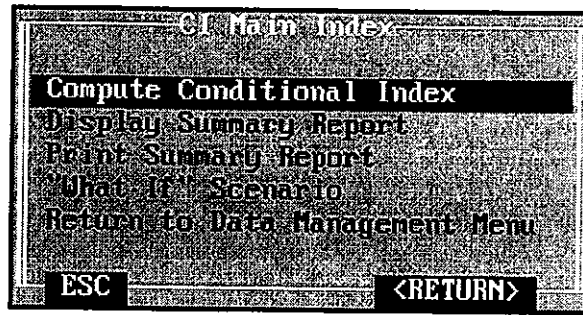


Figure A1.2 RGCM Main Menu

- Print Summary Report
- “What If” scenario
- Return to Data Management Menu

A more detailed discussion of each menu option follows.

A1.2.1 Compute Conditional Index

Field inspection data entered in the REMR DBMS is transferred to a number of data files upon starting the RGCM. The data files containing the field inspection data are read and analyzed. From these data files, a CI for each of the components and an overall CI of the entire structure is developed. The CI's are achieved by following the rules outline in the conditional rating procedure for roller gates report. These CI's are compiled and saved into a Summary Report for further reference. After the CI's are completed and the Summary Report is created, the summary report is displayed to the screen. Compute Conditional Index must be executed before any of the other options are available for use.

A1.2.2 Display Summary Report

The Summary Report developed in the computational stage is displays to the screen so the user can see the results. The Summary Report displayed is the same report that is displayed when Compute Conditional Index is chosen.

A1.2.3 Print Summary Report

The Summary Report developed in the computational stage is printed it a local printer. Print Summary Report performs a simple check to see if a printer is attached before it prints the Summary Report. The advantage of Print Summary Report is the user's ability to achieve a hard copy of the results.

A1.2.4 Return to Data Management Menu

The RGCM is terminated and the user returns to the REMR DBMS.

A1.2.5 "What If" Scenario

The "What If" Scenario was developed for the first time in RGCM. The purpose of the "What If" Scenario to give the user the option to ask what would happen to the individual component's CI and the overall structure CI if some of the data was changed. The RGCM "What If" Scenario Menu, which contains five options, is displayed upon selecting "What If" Scenario(Figure A1.3). These options are:

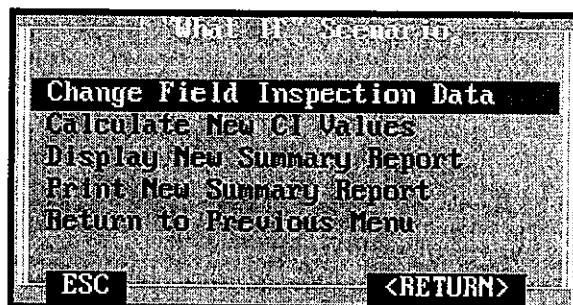


Figure A1.3 Computational Module "What If" Scenario Menu

- Change Field Inspection Data
- Calculate New CI Values
- Display New Summary Report
- Print New Summary Report
- Return to Previous Menu

A1.2.5.1 Change Field Inspection Data

A menu containing a list of distresses is displayed for the user. The menu allows the user to investigate a distress by changing the field data entered in the data management system. Change Field Inspection Data does not change the actual field data, it only makes changes to an automatically generated copy of the field data called the "What If" data. The "What If" data is produced by the RGCM when the "What If" section is entered. Any changes made to the "What If" data can not be retrieved after the "What If" section of the program is exited. The changes made to the "What If" data WILL NOT change the field data in the DBMS. Therefore the user does not have to be concerned with remembering the field data and reentering the correct data after they are done making changes.

Once a distress is chosen, a window is displayed that lists both the field inspection data and the "What If" data. The RGCM allows the user to edit only the "What If" data. In some of the windows, the RGCM permit changing the "What If" data while in other windows, the RGCM require stating if the assembly will be fixed. Each window in which changes have been made needs to be saved before the changes will effect the new "What If" calculations. The user will be prompted to save the data upon exiting the window if there are changes that are not saved. The data in this section is saved to the "What If" data and does not effect the field data in the DBMS.

A1.2.5.2 Compute New CI Values

New CI values for the "What If" data are calculated. The Compute New CI Values option will find the new CI of only the distresses that were changed and saved. All other distresses will have the same CI as found in the original CI calculation. After the new CI's are calculated, two Summary Reports are produced. The Summary Reports state the difference between the original CI's and the new CI's. One Summary Report is a short Summary Report stating only the difference in the CI's values while the other is a long Summary Report that states the difference between the field data and "What If" data in

addition to the difference between the CI's values. The longer Summary Report also states if a component of the gate will be repaired or improved. After the calculations are done, the short Summary Report is displayed to the screen.

A1.2.5.3 Display New Summary Report

The short Summary Report developed during the Calculate New CI Values is displayed to the screen. Due to the length of the longer Summary Report, the longer Summary Report is not displayed to the screen with this option. The longer Summary Report can be viewed by printing out a copy using the Print New Summary Report option..

A1.2.5.4 Print New Summary Report

The long Summary Report developed during the Calculate New CI Values is printed to a local printer. Since the long Summary Report contains all the information of the short Summary Report, the short Summary Report is not printed.

A1.2.5.5 Return to Previous Menu

Return the user to the Main Menu of the RGCM.

A1.3 Software Structure

The RGCM was developed using Borland's Turbo Pascal 7.0 for DOS. The following explanation of the working structure of the program assumes that the user has a working knowledge of Turbo Pascal. The following explanation is structured to give a program developer a general understanding of the major components of the RGCM and how these components all work together to produce the RGCM. The discussion is not intended to be a detailed explanation of the functions and procedures contained in the RGCM code. In this discussion, **Bold** indicates unit files, *Italics* indicates procedures in the units, and monotype indicates menu commands.

The RGCM code consists of one program and several units. A unit is a compiled file with procedures or functions that can be used by a program. The major difference between a unit and a program is that a unit cannot be compiled into an executable file, thus the unit can

not run as a stand alone executable program. The main program source code is called **rollmain.pas** while the unit source code names are as follows: **rolcal.pas**, **rollwif.pas**, **rollwif2.pas**, and **rollwsum.pas**. These five files are combined during the compiling process to form one executable called **rollmain.exe**. A description and explanation of the context for the program and each of the units follows.

A1.3.1 Rollmain.pas

Rollmain.pas contains all of the routines that control the program. The main procedures contained within **rollmain.pas** are: *Main_prog_menu*, *Cal_ci*, *Sum_report*, *Print_report*, *End_main*, *Whatif_main_menu*, *Whatif_dis_menu*, *whatif_cal_ci*, *whatif_sum_report*, and *Whatif_print_report*. A description and explanation of the context of the main procedures are as follows.

A1.3.1.1 Main_prog_Menu

Main_prog_menu procedure initializes and starts the program. The data files are read so the program can display the location the field data was taken from. The date is also retrieved from the internal clock in the computer and displayed. The main menu is then displayed and the user is given a choice of what option to execute. Once an option has been chosen, the proper procedure is then called.

A1.3.1.2 Cal_ci

The different calculation routines in **rolcal.pas** are called. The data is read by the calculation routines and the individual and overall CI's are found. Once the calculations are completed and the summary report is written, the summary report is displayed to the screen. *Cal_ci* is executed when the Calculate Conditional Index is chosen of the Main Menu (Figure A1.2).

A1.3.1.3 Sum_report

The summary report is displayed to the screen. *Sum_report* is executed when Display Summary Report is chosen off the Main Menu (Figure A1.2).

A1.3.1.4 *Print_report*

The summary report is printed to a local printer. A simple check to see if the printer ready to print is performed before the report is printed. *Print_report* is executed when Print Summary Report is chosen off the Main Menu (Figure A1.2).

A1.3.1.5 *End_main*

RGCM is terminated and the user is returned to DBMS. *End_main* is executed when Return to Data Management is chosen off the Main Menu (Figure A1.2).

A1.3.1.6 *Whatif_main_menu*

Whatif_main_menu starts by copying the field data files that were passed by the DBMS and producing the "What If" data files. Once the new files are produced, the "What If" main menu is displayed and the user is given a choice of what option to execute. Upon the user choosing one of the options, the proper procedure is then called. *Whatif_main_menu* is executed when "What If" Scenario is chosen from the Main Menu (Figure A1.2).

A1.3.1.7 *Whatif_dis_menu*

Whatif_dis_menu analyses what distresses can be improved by changing some of the data. The distresses that can be improved are then displayed in a list for the user to choose from. Once the user chooses what distress should be edited, *whatif_dis_menu* then calls the appropriate procedure which then displays the change window for the chosen distress. The window procedure is in either **rollwif.pas** or **rollwif2.pas**. *Whatif_dis_menu* is executed when Change Field Inspection Data is chosen from the "What If" main menu (Figure A1.3).

A1.3.1.8 *Whatif_cal_ci*

Whatif_cal_ci calculates the new CI values for the "What If" scenario. After the calculations are complete, a long and short summary stating the difference between the field inspection data and the "What If" data are written. The long summary report is written when the procedure in **rollwsum.pas** is called, while the short report is written using the a

procedure in either **rollwif.pas** or **rollwif2.pas**. With the report written, the short summary is displayed to the screen. *Whatif_cal_ci* is executed when Calculate New CI Values is chosen from the “What If” main menu (Figure A1.3).

A1.3.1.9 *Whatif_sum_report*

The short summary is displayed to the screen. *Whatif_sum_report* is executed when Display New Summary Report is chosen from the “What If” main menu (Figure A1.3).

A1.3.1.10 *Whatif_print_report*

The long summary is printed to a local printer. A simple check to see if the printer is ready to print is performed before the paper is printed. *Whatif_print_report* is executed when Print New Summary Report is chosen from the “What If” main menu (Figure A1.3).

A1.3.2 Rollcal.pas

Rollcal.pas contains all the procedures needed to calculate the individual CI’s for each distress and the overall CI for the roller gate. Since all the procedures for calculating the individual CI’s are the same in form, they will be described in general. The individual CI procedures start by reading the data from either the field data file or the “What If” data file, depending on whether *Cal_ci* or *whatif_cal_ci* called the calculation procedure. The data is then analyzed and a CI is assigned to the distress. The CI is assigned using the rules described in the condition assessment of Roller Gate report[reference]. The CI is then passed to a procedure that calculates the overall CI of the gate. Once all the CI’s are calculated, a summary report is written and saved to a file. The summary report written by this section is the small report in the “What If” section. These procedures are accessed by the *Cal_ci* and *whatif_cal_ci* procedures in the **rollmain.pas** program.

A1.3.3 Rollwif.pas & Rollwif2.pas

Rollwif.pas and **rollwif2.pas** contain procedures that display edit windows for the “What If” options. Since all the different procedures are the same in form, the procedures

will be described in general. The procedure starts by reading the data from both the field inspection data files and the new “What If” data files. With the data read, a window containing the data is displayed. Then the procedure monitors the user’s input and responds by editing the data, exiting the window, or saving the changed data. Once the user is finished, the window is erased and the procedure is exited. The procedures in **rollwif.pas** and **rollwif2.pas** are accessed by the *whatif_dis_menu* procedure in the **rollmain.pas** program.

A1.3.4 Rollwsum.pas

Rollwsum.pas contains one main procedure, *ci_cal_wifsumm* which is used to write the long summary report for the “What If” changes. *Ci_cal_wifsumm* looks at all the data in both the field inspection data files and the “What If” data files and compares them. The results of the comparison are written to a data file. This file contains information on what changes were made in the “What If” data and how this affects the new CI values. This procedure is accessed only by *whatif_cal_ci* procedure in the **rollmain.pas** program.

APPENDIX 2 OPERATING EQUIPMENT COMPUTATIONAL MODULE

A2.1	Introduction	76
A2.2	Software Usage.....	76
A2.2.1	Compute Conditional Index	77
A2.2.2	View Summary Report.....	78
A2.2.3	Print Summary Report.....	78
A2.2.4	Return to Data Management Menu	79
A2.2.5	“What If” Menu.....	79
A2.2.5.1	Chose Distress To Modify.....	80
A2.2.5.2	Compute New Conditional Index.....	82
A2.2.5.3	View New Summary Report	83
A2.2.5.4	Print New Summary Report	83
A2.2.5.5	Return to Main Menu	83
A2.3	Software Structure.....	83
A2.3.1	Oemain.pas	84
A2.3.1.1	<i>Prog_main</i>	84
A2.3.1.2	<i>Cal_ci</i>	85
A2.3.1.3	<i>Summ_report</i>	85
A2.3.1.4	<i>Print_report</i>	86
A2.3.1.5	<i>Whatif_main_menu</i>	86
A2.3.1.6	<i>Whatif_sub_menu</i>	86
A2.3.1.7	<i>Whifwin_exec</i>	86
A2.3.1.8	<i>Summall</i>	87
A2.3.1.9	<i>Menu</i>	87
A2.3.2	Oecal.pas	87
A2.3.3	Oewsumm1.pas & Oewsumm2.pas.....	88
A2.3.4	Whifwin.pas	88
A2.3.5	Oewif1.pas, Oewif2.pas, Oewif3.pas, Oewif4.pas.....	88

A2.1 Introduction

The operating equipment computational module (OECM) described in the following appendix was developed to accompany a report developed for the Army Corps of Engineers on Condition Rating Procedure for Operating Equipment (Greimann et al., 1996). The OECM was developed to work in conjunction with Repair, Evaluation, Maintenance, and Rehabilitation Research Program (REMR) Management Systems' Inspection and Rating Data Management Software (Greimann et al., 1994). Although the OECM can operate as a stand alone unit, it is not intended to work this way and should always be used with the REMR database management system (DBMS). OECM is another in a line of similar programs that have been developed for the Army Corps of Engineers. OECM most closely resembles the Roller Gate Computational Module (RGCM) described in Appendix 1. The following discussion covers only the OECM and assumes a working knowledge of the REMR DBMS. For further information on REMR DBMS, see REMR Management Systems' Inspection and Rating Data Management Software (Greimann et al., 1994).

A2.2 Software Usage

OECM was developed to work with the REMR DBMS (Greimann et al., 1994). Choosing "Transfer Data to Evaluation Module" in the REMR DBMS' edit menu will start the OECM (Figure A2.1). The main menu of the OECM is then displayed, thus giving the user five options from which to choose (Figure A2.2). These options are:

- Compute Condition Index
- View Summary Report
- Print Summary Report
- "What If" Menu
- Return to Data Management Menu

A more detailed discussion of each menu option follows.

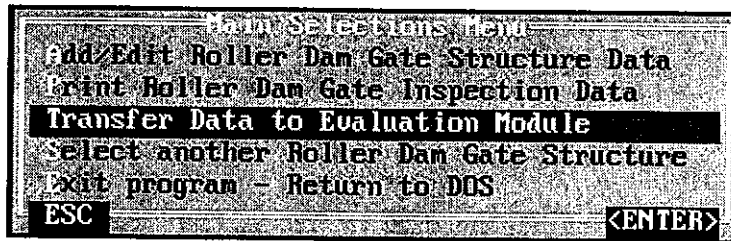


Figure A2.1 REMR DBMS Edit Menu

A2.2.1 Compute Conditional Index

Field inspection data entered in the REMR DBMS is transferred to a number of data files upon starting the OECM. These data files containing the field inspection data are read and analyzed. From the data files, many different CI values are calculated. The CI for all the

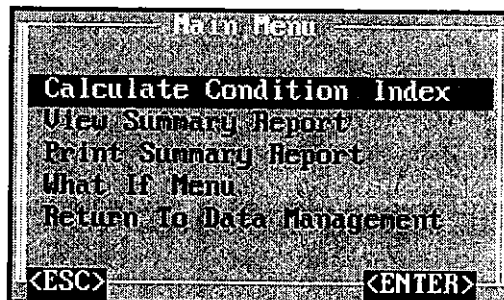


Figure A2.2 OECM Main Menu

available assemblies are computed along with the CI of the distresses that make up the assemblies. A CI value for the entire system is not computed. The CI's are calculated by following the rules outlined in the condition rating procedure for operating equipment report. These CI's are compiled and written to various Summary Reports for further reference. One of the Summary Reports, called the Sub-Structure Summary Report, contains the overall CI for each of the assemblies contained in the sub-structure. The other Summary Reports are assembly specific. These Summary Reports contain the CI's for the distresses contained in a given assembly. After the CI's are calculated and the Summary Reports are created, the Sub-

Structure Summary Report is displayed to the screen. Compute Condition Index must be executed before any other options are available for use.

A.2.2.2 View Summary Report

The View Summary Report option allows the user to view either the Sub-Structure Summary Report or any of the assembly specific Summary Reports. Choosing Display Sub-Structure Summary on the Summary Menu displays the Sub-Structure Summary Report displayed in the Compute Condition Index option (Figure A2.3). Choosing Display

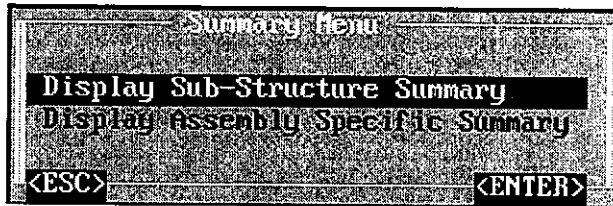


Figure A2.3 OECM Summary Report Main Menu

Assembly Specific Summary on the Summary Menu displays another menu that contains all the assemblies available in the OECM. Once an assembly has been chosen, another menu is displayed, which lists all the available Power Transfer Number (PTN) for the chosen assembly and an 'All' option (Figure A2.4). For an explanation of PTN's, please see the Condition Rating Procedure for Operating Equipment. The 'All' option displays a Summary Report that contains the individual distress' CI's values and overall CI values for all the PTN of the chosen assembly. Choosing one of the PTN will display a Summary Report that contains the individual distress' CI's values and the overall CI value for the chosen PTN only.

A2.2.3 Print Summary Report

All the Summary Reports available in Section A2.2.2, View Summary Report, are available in the Print Summary Report Section. The same menu system is used to enable the user to identify the Summary Report of interest. Print Summary Report performs a check to

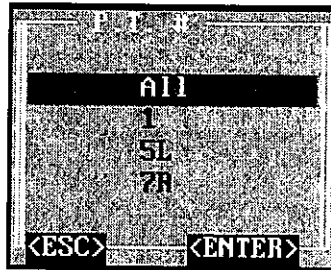


Figure A2.4 OECEM Available Power Transfer Numbers

see if a printer is attached and ready before it prints the Summary Report to the printer. The advantage of Print Summary Report is the ability to obtain a hard copy of the results for the user.

A2.2.4 Return to Data Management Menu

The OECEM is terminated and the user returns to the REMR DBMS.

A2.2.5 “What If” Menu

The “What If” Menu, developed in the OECEM, is similar to the “What If” Scenario in the RGCM. The purpose of the “What If” Menu is to give the user the ability to ask what would happen to the individual component’s CI and the overall assembly’s CI if some of the data entered in the REMR DBMS was different. The first time this section is executed, the OECEM copies the field inspection data for the “What If” section. These copies are called the “What If” data files. Any other time the “What If” Scenario is executed, a window will appear asking the user if they want to replace the old “What If” data files with new files or work with the old “What If” data files. The original field inspection data files are never changed. The OECEM’s “What If” Main Menu, which contains five options, is displayed upon selecting “What If” Menu from the Main Menu (Figure A2.5). These options are:

- Chose Distress To Modify
- Calculate New Condition Index
- View New Summary Reports

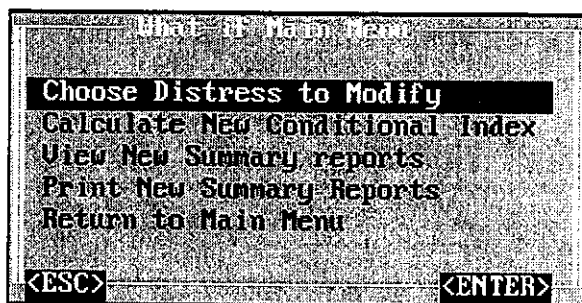


Figure A2.5 OECCM “What If” Main Menu

- Print New Summary Reports
- Return to Main Menu

A2.2.5.1 Chose Distress To Modify

The Chose Distress To Modify option gives the user a way to identify the assembly and distress they would like to examine. Improvements are made by changing the “What If” data files. Any data changed in this section will not affect the field inspection data entered in the DBMS. All changes are stored in the “What If” data files only.

The Chose Distress to Modify option displays a menu containing a list of assemblies available to the user (Figure A2.6). Once an assembly is chosen, a list of available PTN’s is displayed. Upon choosing a PTN, another list displays the distresses for the assembly. This

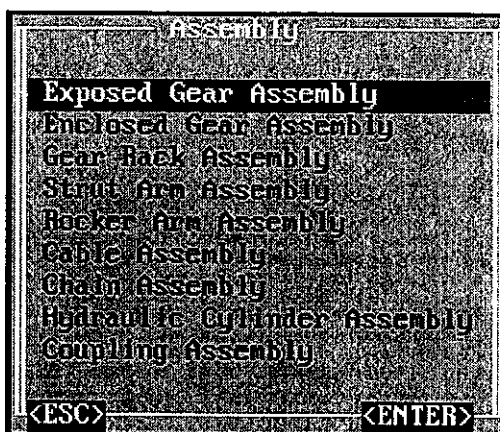


Figure A2.6 OECCM Assembly List

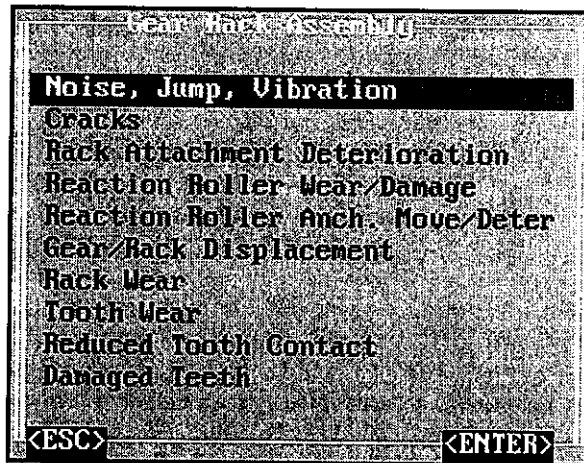


Figure A2.7 OECM Gear Rack Assembly Distress List

enables the user to choose a distress to improve. Figure A2.7 shows a list of distresses for the Gear Rack assembly. Some of the distresses also require the user to make further specifications on which part of the distress they want to evaluate. An example is the gear set number for some of the distresses in exposed gears.

Once a distress is chosen, a window is displayed listing both the field inspection data and the “What If” data. Figure A2.8 shows the Noise, Jump, and Vibration distress for the Gear Rack assembly. The OECM allows the user to edit only the “What If” data. Each window in which data has been changed must be saved before these changes will effect the new “What If” calculations. The user will be prompted to save the data upon exiting the window if the changes have not been saved (Figure A2.9).

A2.2.5.2 Compute New Conditional Index

New CI values for the “What If” data are calculated. The Compute New CI Values option will find the new CI of only the distresses that were changed and saved. All other distresses will have the same CI as found in the original CI calculation. After the new CI’s are calculated, all the Summary Reports explained in the Compute Condition Index, Section A2.2.1, are written for the new “What If” data. In addition to these Summary Reports, some

Gear Rack Noise, Jump, and Vibration			
Field Collected data			
Is there Noise?	<input type="checkbox"/>	Is this normal?	<input type="checkbox"/>
Is there Jumping?	<input type="checkbox"/>	Is this normal?	<input type="checkbox"/>
Is there Vibration?	<input type="checkbox"/>	Is this normal?	<input type="checkbox"/>
"What if" Values			
Is there Noise?	<input type="checkbox"/> No	Is this normal?	<input type="checkbox"/> Yes
Is there Jumping?	<input type="checkbox"/> No	Is this normal?	<input type="checkbox"/> Yes
Is there Vibration?	<input type="checkbox"/> No	Is this normal?	<input type="checkbox"/> Yes
<input type="button" value="ESC"/>		<input type="button" value="CTRL-ENTER"/>	

Figure A2.8 OEMC "What If" Noise, Jump, and Vibration Window For Gear Rack Assembly

new Summary Reports, called the Difference Summary Reports, are created. These Difference Summary Reports state the difference between the "What If" data files and the field inspection data files. The Difference Summary Reports will only list differences between the files that improve the CI value of the distress. Any changes that worsen the CI value are not mentioned. This limitation is implemented because the belief is held that the only changes investigated in the "What If" section will be improvements to the assemblies and distresses. Once all the Summary Reports have been written, the Sub-Structure Summary Report for the "What If" data is displayed to the screen. Compute New Condition

Data has been modified. Save Data? (Y/N/C)

Figure A2.9 OEMC Save Data Window

Index must be executed before the “What If” data Summary Reports can be viewed or printed.

A2.2.5.3 View New Summary Report

View New Summary Report is very similar to the View Summary Report Section, Section A2.2.2, found in the Main Menu of the OECM. The difference between the two is View New Summary Report has one additional option in the first menu. This new option, called View Difference Report, is used to view the Difference Summary Reports generated in the Compute New Condition Index. Once View Difference Report is chosen, a menu listing all the available assemblies is displayed. Once an assembly has been chosen, a list of available PTN's is displayed. With the PTN selected, the appropriated Difference Summary Report is given. If there is no difference between the “What If” data files and the field inspection data files, the Difference Summary Report will be displayed as a blank screen.

A2.2.5.4 Print New Summary Report

Print New Summary Report uses the same process as View New Summary Report Section, Section A2.2.5.3, to choose a desired Summary Report. This option directs the chosen summary report to a local printer instead of the screen. As in the Print Summary Report Section, Section A2.2.3, on the Main Menu, this option also does a simple check to see if the printer is ready to print.

A2.2.5.5 Return to Main Menu

Returns to Main Menu returns the user to the Main Menu of the OECM.

A2.3 Software Structure

The OECM was developed using Borland's Turbo Pascal 7.0 for DOS. The following description of the working structure of the program assumes that the reader has a working knowledge of Turbo Pascal. The following description is structured to give a program developer a general understanding of the major components of the OECM and how these components all work together to produce the OECM. This discussion is not intended to

be a detailed explanation of the functions and procedures contained in the OECCM code. In this discussion, **Bold** indicates unit files while *Italics* indicates procedures in the units.

The OECCM code consists of two programs and several units. A unit is a compiled file with procedures and/or functions that can be used by a program. The major difference between a unit and a program is that a unit cannot be compiled into an executable file, thus the unit cannot run as a stand alone executable program. The source code for the two main programs are contained in **oemain.pas** and **wifwin.pas**. The source code for the units contained in **oemain.pas** are **oecal.pas**, **oewsumm1.pas**, and **oewsumm2.pas**. The source code for the units contained in **wifwin.pas** are **oewif1.pas**, **oewif2.pas**, **oewif3.pas**, and **oewif4.pas**. During the compiling of the main program code, the units' codes are included to form an executable program. A description of the content of the main programs and the units follows.

A2.3.1 Oemain.pas

Oemain.pas contains most of the code used in the controlling of the program. The main procedures and functions in **oemain.pas** are: *prog_main*, *cal_ci*, *summ_report*, *print_report*, *whatif_main_menu*, *whatif_sub_menu*, *whifwin_exec*, *summall*, and *menu*. The following discussion contains a description of the procedure and functions and how they are used.

A2.3.1.1 Prog_main

Prog_main is the first procedure called when the program is executed. This procedure begins by reading what data files have been passed by the DBMS to the OECCM. Unlike previous programs like RGCM, the OECCM does not know in advance what files will be passed by the DBMS. Because of this, *Prog_main* starts by reading what assemblies have been passed and how many different PTN each assembly contains. This information is determined by reading the name of the data files passed by the DBMS. The method used to name the data files are listed in Table A2.1. Once this information is read and stored, the

Table A2.1 Data File Naming Table

Type	Sub-structure Type	Assembly Type	Page Number	PTN Specification
0	1	00	0	00
		01		01
1	3	02	1	02
		03	2	03
3	3	04	3	04
		05		05
4		06		
5		07		
6		08		
		09		

main menu is displayed by calling the menu function (explained later). Once an option has been chosen by the user, the proper procedure is called.

A2.3.1.2 *Cal_ci*

Cal_ci is used to call the calculation procedures used to calculate the various CI values for the different assemblies. *Cal_ci* is used to calculate the CI for both the field inspection data and the "What If" data. *Cal_ci* calls only the procedures needed to calculate the CI for the assemblies passed to the OEMCM. The procedures called by *cal_ci* is contained in *oecal.pas*. *Cal_ci* is called by *prog_main* and *whatif_main_menu*.

A2.3.1.3 *Summ_report*

Summ_report uses *menu* to display the various menus needed to equip the user with the ability to identify which summary report to view. *Summ_report* is used to display the field inspection data summary reports and also the "What If" data summary reports. The menus displayed vary slightly depending on whether the field inspection data summary reports or the "What If" data summary reports are available. Once a summary report has been chosen, the report is displayed to the screen. *Summ_report* is called by *prog_main* and *whatif_main_menu*.

A2.3.1.4 *Print_report*

Print_report displays the same menus as *summ_report*. Unlike *summ_report*, *print_report* prints the summary report to a local printer. Before the file is printed, a check to see if the printer is ready is performed. *Print_report* is called by *prog_main* and *whatif_main_menu*.

A2.3.1.5 *Whatif_main_menu*

Whatif_main_menu starts by determining if any “What If” data files exist. If no files are found, “What If” data files are created by copying the field inspection data files. If files are found, the user is asked if they want to replace existing “What If” data files with new “What If” data files, or to use the existing “What If” data files. Once the “What If” files have been attended, the “What If” main menu is displayed using *menu*. Upon choosing an option from the menu, the proper procedure is called. If the Chose Distress To Modify option is chosen off the “What If” main menu, a list of available assemblies is displayed. Once an assembly has been chosen, a list of available PTN is given from which the user can choose. *Whatif_main_menu* is called by *prog_main*.

A2.3.1.6 *Whatif_sub_menu*

Whatif_sub_menu receives the assembly and PTN the user chose from *whatif_main_menu*. With this information, a list identifying the distresses for the chosen assembly is displayed using *menu*. Some of the distresses require an additional menu to specify the distress more closely. Once a distress is chosen, *whifwin_exec* is called. *Whatif_sub_menu* is called by *whatif_main_menu*.

A2.3.1.7 *Whifwin_exec*

Whifwin_exec has the assembly, PTN, and distress passed to it by *whatif_sub_menu*. With this information, *whifwin_exec* calls the program **wifwin.exe** to display the various “What If” changes windows. *Whifwin_exec* is called by *whatif_sub_menu*.

A2.3.1.8 *Summall*

Summall is used to create the 'All' summary reports mentioned in View Summary Report. *Summall* takes all the summary reports for a specific assembly that contains the CI values for the individual PTN and combines them. *Summall* does this for all the available assemblies. If there is only one PTN for an assembly, then the 'All' summary report will look identical to the individual report. *Summall* is called by *Cal_ci*.

A2.3.1.9 *Menu*

Menu is a function written to display a menu on the screen. *Menu* is passed a list of options to be displayed and then displays this list. With the list displayed, the function monitors the user choice. Once an option is chosen, *menu* passes back to the procedure that called *menu* the selected option. *Menu* is called by *prog_main*, *summ_report*, *print_report*, *whatif_main_menu*, and *whatif_sub_menu*.

A2.3.2 *Oecal.pas*

Oecal.pas contains all the procedures needed to calculate the overall and individual CI values. Although ***oecal.pas*** contains many different procedures, there are only two main types: the assembly CI procedure and the distress CI procedure. Both procedures will be explained in general.

The distress CI procedure calculates the CI of a particular distress. This procedure begins by reading the data file to obtain the data needed in the calculation. The appropriate equations from the Operating Equipment Report are used with the data to develop a CI value for the individual distress.

The assembly CI procedure obtains the CI values from the appropriate distress CI procedures. These individual CI values are written to the assembly specific summary report for a given PTN. The individual CI values are also used to calculate an overall CI value for the assembly. This overall CI value is written in both the assembly specific summary report and the sub-structure summary report. The method used to find the overall CI is outlined in

the operating equipment report. The assembly CI procedures are the procedures called by *Cal_ci*.

A2.3.3 Oewsumm1.pas & Oewsumm2.pas

Oewsumm1.pas and **oewsumm2.pas** contain all the procedures used in writing the difference report for the “What If” section. Like **oecal.pas**, these units contain many different procedures, but all the procedures can be labeled into two categories, the distress difference procedure and the assembly difference procedure.

The distress difference procedure looks at the difference between the “What If” data files and the field inspection data files for only one specific distress. Once a difference has been found, this difference is written to the difference report.

The assembly difference procedure calls the proper distress difference procedure for a given assembly. The order that the distress procedures are called in the assembly procedure dictates the order the results are written in the difference summary report. The assembly difference procedure is called by *Cal_Ci*.

A2.3.4 Whifwin.pas

Unlike the units attached to **oemain.pas**, **whifwin.pas** is an executable program accessed by **oemain.pas**. **Whifwin.pas** contains code needed to call the “What If” windows for a given assembly and distress. The code used to display the windows are contained in the units **oewif1.pas**, **oewif2.pas**, **oewif3.pas**, and **oewif4.pas**. All needed information is passed to **whifwin.pas** using the DOS command line. This information is then used to determine which assembly and distress window is to be displayed.

A2.3.5 Oewif1.pas, Oewif2.pas, Oewif3.pas, Oewif4.pas

Oewif1.pas, **Oewif2.pas**, **Oewif3.pas**, and **Oewif4.pas** contain all the code needed to display the various windows for the “What If” section. **Oewif1.pas** contains the following objects used in displaying the windows: *ftext*, *ftextval*, *fboolean*, *fintplus*, *fcodedchoice*, and *freadit*. Objects are sections of code that can be accessed like procedures, but they allow

the programmer to define the type of object. Each object performs a specific function in the “What If” windows and decreases the amount of the code for each of the windows. Although each object is different in its use, each object is used to create a field in the “What If” window that performs a special function. These field objects are described below.

Ftext displays text without a border. The text has to be specified before compiling the program. Once the text has been displayed, it cannot be edited by the user.

Ftextval displays text in a green box. The text cannot be edited by the user. *Ftextval* is used to display the field inspection data for a specific distress. This text is read from the field inspection data files. This field object enables the user to know the field inspection data during “What If”.

Fboolean displays a blue Yes/No box. *Fboolean* allows the user to choose and edit a Yes/No value by toggling with the space bar. The initial value for the Yes/No box is read from the “What If” data files.

Fintplus displays an integer value in a blue box. *Fintplus* allows only integer values and can be edited by using the up and down arrow. The integer value displayed at the start is read from the “What If” data files.

Fcodedchoice enables the user to choose a value from a list of values. One of the available options is displayed in a blue box and is changed by toggling with the space bar. The initial value is read from the “What If” data files.

Frealedit allows the user to edit a real number. This number is read from the “What If” data file and displayed in a blue box. The user is not allowed to leave this field object until the number displayed in the object is a valid number.

The procedures used to display the different windows have the same basic structure, so they will be described in general. The procedure starts by reading the field inspection data files and the “What If” data files. Then the different field objects are defined and drawn to the screen. The user is then allowed to edit the “What If” data. Once the user is finished, the

data is saved upon request. The field objects are then erase from the screen and procedure is exited.

APPENDIX 3 OTHER CONTRIBUTIONS TO THE PROJECTS

A3.1	Introduction	92
A3.2	Roller Dam Gate and Operating Equipment Reports	92
A3.3	Site Visits	92
A3.4	Training Sessions	93
A3.5	Saint Paul District Evaluation	93

A3.1 Introduction

During my Master of Science study at ISU, I did much more than just write winREMR (thesis text), roller dam gate (Appendix 1), and operating equipment (Appendix 2) computer programs. I helped with various other U.S. Corps of Engineers projects that were being worked on including the roller dam gate and the operation equipment technical reports. I assisted with site visits while investigating roller dam gates. I also helped in the inspection and evaluation of the miter lock gates at Lock and Dam #2, #3, and #5A on the Mississippi River. Another aspect in which I helped was in a tainter dam gate training session.

A3.2 Roller Dam Gate and Operating Equipment Reports

Although I was not listed as an author for either the roller dam gate report (Greimann et al., 1996) or the operating equipment report (Greimann et al, 1996), I assisted the authors whenever possible. This assistance varied from time to time. I provided a sounding board for suggestions and ideas of improvement. I also gave feedback for some of the suggestions. One area in which I provided insight was on the legitimacy of the rules. Since I was writing the computer programs at the same time as the reports were being prepared, I would evaluate the rules and the explanations in the report to see if they made sense. This evaluation was done because, as the program developer, I had to fully understand the rules to properly write the computational modules.

A3.3 Site Visits

During the development of the roller dam gate report, the ISU team was required to go on many site visits to various lock and dams on the Mississippi river. These visits were to investigate methods that we developed for inspecting and analyzing the structure and also to

see if these methods would work in the real world. During the field trips, I helped in the evaluation of the ideas and also provided insight on the improvement of some of the ideas.

A3.4 Training Sessions

While working for the research team at ISU, I assisted in a training session for the U.S. Army Corps of Engineers. During the training session, I assisted with setting up the measurement devices on the tainter dam gate. I also helped to modify the program to include some last minute changes.

A3.5 Saint Paul District Evaluation

While at ISU, our group was hired to evaluate the miter lock gates at Lock and Dam #2, #3, and #5A on the Mississippi River. This evaluation was begun by traveling to the sites to perform the inspection. I assisted in the evaluation by provided possible methods for overcoming the difficulties that we encountered. I was also called upon to lead a second trip to the sites to record additional measurements for the report.

REFERENCES

1. Borland, Turbo Pascal: Language Guide, Borland International, Inc., Scotts Valley, California, 1992
2. Borland, Turbo Pascal: Programmer's Reference, Borland International, Inc., Scotts Valley, California, 1992
3. Borland, Turbo Pascal: User's Guide, Borland International, Inc., Scotts Valley, California, 1992
4. Borland, Turbo Vision: Programmer's Guide, Borland International, Inc., Scotts Valley, California, 1992
5. Dale, N., and C. Weems, Pascal, D.C. Heath and Company, Lexington, Massachusetts, 1991
6. Grant, Tom, Teach yourself Oracle Power Objects in 21 Days, Sams Publishing, Indianapolis, Indiana, 1996
7. Greimann, L., and J. Stecker, User's Manual, Inspection and Rating of Steel Sheet Pile Structures, Technical Report REMR-OM-3/ADA210411, US Army Corps of Engineers, Washington DC, 1989
8. Greimann, L., and J. Stecker, Maintenance and Repair of Steel Sheet Pile Structures, Technical Report REMR-OM-9/ADA2231916, US Army Corps of Engineers, Washington DC, 1990
9. Greimann, L., J. Stecker, and K. Rens, Management and Repair for Miter Lock Gates, Technical Report REMR-OM-08/ADA231469, US Army Corps of Engineers, Washington DC, 1990
10. Greimann, L., J. Stecker, and K. Rens, Condition Rating Procedure for Sector Gates, Technical Report REMR-OM-13/279299, US Army Corps of Engineers, Washington DC, 1993
11. Greimann, L., J. Stecker, and J. Veenestra, Condition and Rating Procedure for Tainter and Butterfly Valves, Technical Report REMR-OM-14/ADA279326, US Army Corps of Engineers, Washington DC, 1994
12. Greimann, L., J. Stecker, K. Rens, and M. Nop, REMR Management Systems-Navigation Structures, User's Manual for Inspection and Rating Software, Version 2.0, Technical Report REMR-OM-15, US Army Corps of Engineers, Washington, DC, 1994

13. Greimann, L., J. Stecker, and M. Nop, Condition Rating Procedure for Tainter Dam and Lock Gates, Technical Report REMR-OM-17/ADA303294, US Army Corps of Engineers, Washington DC, 1995
14. Greimann, L., J. Stecker, and T. Kraal, Condition Rating Procedure for Roller Dam Gate, Technical Report REMR-OM-118, US Army Corps of Engineers, Washington, DC, 1996
15. Greimann, L., J. Stecker, S. Mellema, and K. Rens, Condition Rating Procedures for Lock and Dam Operating Equipment, Technical Report REMR-OM-119, US Army Corps of Engineers, Washington, DC, 1996
16. Gurewich, N., and O. Gurewich, Teach yourself Visual Basic 4 in 21 Days, Third Edition, Sams Publishing, Indianapolis, Indiana, 1995
17. Jennings, R., Database Developer's Guide with Visual Basic 4, Second Edition, Sams Publishing, Indianapolis, Indiana, 1996
18. Kolste, B., and D. Petersen, Oracle: Power Objects Handbook, Osborne McGraw-Hill, Berkeley, California, 1995
19. McKay, D., and A. Koa, REMR Tools Version 2.00: An Engineering Application Development System, U.S. Army Construction Engineering Research Laboratory, Champaign, Illinois
20. Microsoft Corporation, *Knowledge Base*, <<http://www.microsoft.com/kb/>>, (Accessed Nov. 1996 - March 1997)
21. Nanney, T. R., Computing and Problem-Solving with Pascal, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1985
22. Swan, T., Type & Learn C, IDG Books, San Mateo, California, 1994
23. Trimble, J. H., and D. Chappel, A Visual Introduction to SQL, John Wiley & Sons, New York, 1989
24. Waston, B., Programming with Turbo Vision, M&T Books, New York, New York, 1994
25. Weiskamp, K., Borland Pascal: Step-By-Step, John Wiley & Sons, Inc, New York, New York, 1994

26. Wyatt, A. L. Sr., W. E. Tiley, and J. Paisley, Using MS-DOS 6.2 : Special Edition, Que Corporation, Indianapolis, Indiana, 1993