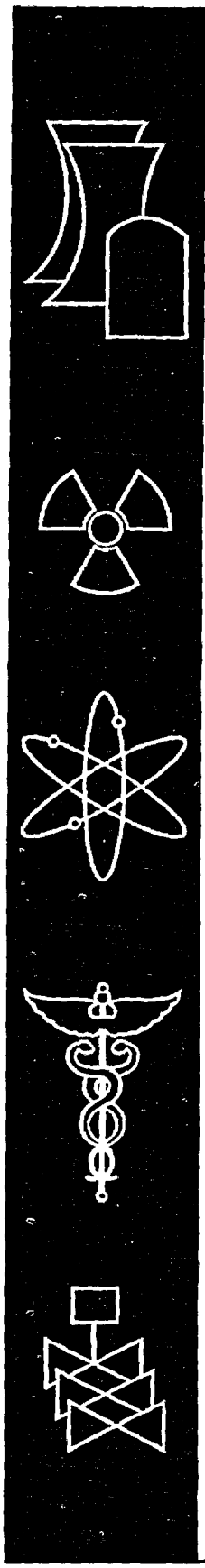
NUREG/CR-6525, Rev. 1



SECPOP2000: Sector Population, Land Fraction, and Economic Estimation Program

Sandia National Laboratories

U.S. Nuclear Regulatory Commission
Office of Nuclear Regulatory Research
Washington, DC 20555-0001

AVAILABILITY OF REFERENCE MATERIALS IN NRC PUBLICATIONS

NRC Reference Material

As of November 1999, you may electronically access NUREG-series publications and other NRC records at NRC's Public Electronic Reading Room at http://www.nrc.gov/reading-rm.html. Publicly released records include, to name a few, NUREG-series publications; *Federal Register* notices; applicant, licensee, and vendor documents and correspondence; NRC correspondence and internal memoranda; bulletins and information notices; inspection and investigative reports; licensee event reports; and Commission papers and their attachments.

NRC publications in the NUREG series, NRC regulations, and *Title 10, Energy*, in the Code of *Federal Regulations* may also be purchased from one of these two sources.

1. The Superintendent of Documents
   U.S. Government Printing Office
   Mail Stop SSOP
   Washington, DC 20402–0001
   Internet: bookstore.gpo.gov
   Telephone: 202-512-1800
   Fax: 202-512-2250
2. The National Technical Information Service
   Springfield, VA 22161–0002
   www.ntis.gov
   1–800–553–6847 or, locally, 703–605–6000

A single copy of each NRC draft report for comment is available free, to the extent of supply, upon written request as follows:

Address: Office of the Chief Information Officer,
Reproduction and Distribution
Services Section
U.S. Nuclear Regulatory Commission
Washington, DC 20555-0001
E-mail: DISTRIBUTION@nrc.gov
Facsimile: 301–415–2289

Some publications in the NUREG series that are posted at NRC's Web site address http://www.nrc.gov/reading-rm/doc-collections/nuregs are updated periodically and may differ from the last printed version. Although references to material found on a Web site bear the date the material was accessed, the material available on the date cited may subsequently be removed from the site.

Non-NRC Reference Material

Documents available from public and special technical libraries include all open literature items, such as books, journal articles, and transactions, *Federal Register* notices, Federal and State legislation, and congressional reports. Such documents as theses, dissertations, foreign reports and translations, and non-NRC conference proceedings may be purchased from their sponsoring organization.

Copies of industry codes and standards used in a substantive manner in the NRC regulatory process are maintained at—

The NRC Technical Library
Two White Flint North
11545 Rockville Pike
Rockville, MD 20852–2738

These standards are available in the library for reference use by the public. Codes and standards are usually copyrighted and may be purchased from the originating organization or, if they are American National Standards, from—

American National Standards Institute
11 West 42nd Street
New York, NY 10036–8002
www.ansi.org
212–642–4900

Legally binding regulatory requirements are stated only in laws; NRC regulations; licenses, including technical specifications; or orders, not in NUREG-series publications. The views expressed in contractor-prepared publications in this series are not necessarily those of the NRC.

The NUREG series comprises (1) technical and administrative reports and books prepared by the staff (NUREG–XXXX) or agency contractors (NUREG/CR–XXXX), (2) proceedings of conferences (NUREG/CP–XXXX), (3) reports resulting from international agreements (NUREG/IA–XXXX), (4) brochures (NUREG/BR–XXXX), and (5) compilations of legal decisions and orders of the Commission and Atomic and Safety Licensing Boards and of Directors' decisions under Section 2.206 of NRC's regulations (NUREG–0750).

NUREG/CR-6525, Rev. 1

SECPOP2000: Sector Population, Land Fraction, and Economic Estimation Program

Manuscript Completed: July 2003
Date Published: August 2003

Prepared by
N.E. Bixler, S. A. Shannon, SNL
C.W. Morrow, B.E. Meloche, SNL
J.N. Ridgely, NRC

Sandia National Laboratories
Operated by Lockheed Martin Corporation
Albuquerque, NM 87185-0748

J. N. Ridgely, NRC Project Manager

Pepared for
Division of Risk Analysis and Applications
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, DC 20555-0001
NRC Job Code Y6594

NUREG-CR-6525, Revision 1, has been reproduced from the best available copy.

Abstract

The Environmental Protection Agency's computer program, SECPOP, has been used to calculate population estimates since 1973. In 1997, SECPOP90, was created to run on a PC and to use 1990 population and economic data.

When the U.S. Census Bureau released 2000 census population data and 1997 economic data, Sandia National Laboratories was tasked to develop a new version, SECPOP2000, to include the new data, be compatible with current Windows operating systems, and also be compatible with the 1990 data.

SECPOP90 supports both site and regional analyses. Site analysis evaluates population, land use, and economic data on a polar grid centered on a prescribed site. Regional analysis screens potential sites in a geographical region against a population criterion.

SECPOP2000 currently supports only site analysis. Regional analysis can be performed by using SECPOP90 as a screening tool followed by a confirmatory calculation with SECPOP2000 to evaluate potential sites with 2000 census data.

This report provides verification of the SECPOP90 and SECPOP2000 codes by comparing them with licensee-provided population data. The verification shows that SECPOP90 and SECPOP2000 provide reasonable population estimates. SECPOP2000 also agrees well with census estimates from other sources.

Table of Contents

Abstract iii

Figures ix

Tables xi

1 Introduction 1
1.1 Background 1
1.2 Hardware and Software Requirements 1

2 Installation 3
2.1 Initialization 3
2.2 Starting SECPOP2000 3

3 User's Guide 5
3.1 Parts of the Main Screen 5
3.2 Menus 6
3.2.1 Main Screen 6
3.2.2 Toolbar Menu 6
3.3 Site Definition 7
3.3.1 New Site 7
Site Form Fields 8
Site Form Commands 9
3.3.2 Open Site 10
3.3.3 Save Site 10
3.3.4 Save As Site 10
3.4 Problem Definition 11
3.4.1 New Problem 11
Problem Form Fields 12
Problem Form Commands 14
3.4.2 Economic Regions 14
Economic Regions Form Fields 15
Economic Regions Form Commands 17
3.4.3 Open Problem 17
3.4.4 Save Problem 18
3.4.5 Save As Problem 18
3.5 Performing Site-Specific Calculations 18
Calculate Form Fields 19
Calculate Form Commands 19
3.6 Site-Specific Results 20

3.6.1 Population Rosette 20
3.6.2 Population Tables 21
Population 21
Cumulative 22
3.6.3 MACCS2 Input File 23
Display 23
Print 23
Save 25
3.6.4 Comma-Separated-Variable (CSV) File 25
Display 26
Print 26
Save 30
3.6.5 Load Previous Results 30
3.7 Performing Regional Calculations 30
3.8 Customizing SECPOP2000 30
3.8.1 Setup Form Fields 31
3.8.2 Setup Form Commands 31
3.9 Exiting SECPOP2000 32

4 Computational Methodology 33
4.1 Population Count and Density Algorithms 33
4.1.1 Boundaries Algorithm 33
4.1.2 First Element Location Algorithm 34
4.1.3 Specific Grid Element Determination 35
4.2 Land Fraction Algorithms 35
4.3 Economic Factors Algorithms 37
4.4 Database Structure 38
4.4.1 Block-Level Database 38
4.4.2 County-Level Database 39

5 Software Development 40
5.1 Block-Level Census Database 40
5.1.1 Construction of the 1990 Block-Level Census Database 40
5.1.2 Construction of the 2000 Block-Level Census Database 46
5.1.3 Verification of the 1990 Block-Level Census Database 47
5.1.4 Verification of the 2000 Block-Level Census Database 48
5.1.4.1 Verification Test 1 49
5.1.4.2 Verification Tests 2 and 3 49
5.1.4.3 Verification Test 4 51
5.2 County-Level Census Database 54
5.2.1 Construction of the 1990 County-Level Census Database 54
5.2.2 Changes in the County-Level Database with the 1997 Economic Census 57

5.2.3 Verification of the County-Level Census Database 61
5.3 Verification of the SECPOP90 Code 61
5.3.1 Internal Verification 61
5.3.2 External Verification for the SECPOP90 Code 61
5.4 Verification of the SECPOP2000 Code 65
5.4.1 Internal Verification 65
5.4.2 External Verification of the SECPOP2000 Code 70

6 References 75

Appendix A Site List ..... A - 1

Appendix B MACCS2 Site Data File Format ..... B - 1
B.1 Spatial Distances Data Block ..... B - 7
B.2 Population Data Block ..... B - 7
B.3 Land Fraction Data Block ..... B - 8
B.4 Region Index Data Block ..... B - 9
B.5 Watershed Index Data Block ..... B - 9
B.6 Crop and Season Share Data Block ..... B - 10
B.7 Watershed Definition Data Block ..... B - 10
B.8 Regional Economic Data Block ..... B - 11

Appendix C 1990-2000 County Inventory Changes ..... C - 1

Appendix D Statistical Abstracts (1990) ..... D - 1

Appendix E Statistical Abstracts (2000) ..... E - 1

Appendix F Population Tables ..... F - 1

Appendix G Census File Creation and Testing Source Code ..... G - 1
G.1 Module: "Census_File_Convert" ..... G - 1
G.2 Module: "Census_File_Test" ..... G - 5

Appendix H SECPOP2000 Source Code ..... H - 1
H.1 Visual Basic Code Module: CALCCODE.BAS ..... H - 1
H.2 Visual Basic Code Module: Cmndlg.bas ..... H - 16
H.3 Visual Basic Code Module: Const.bas ..... H - 24
H.4 Visual Basic Code Module: FormsPosition.bas ..... H - 29
H.5 Visual Basic Code Module: GridUtils.bas ..... H - 30
H.6 Visual Basic Code Module: InIt.bas ..... H - 36
H.7 Visual Basic Code Module: MainModule.bas ..... H - 44
H.8 Visual Basic Code Module: Mp_setup.bas ..... H - 45

H.9 Visual Basic Code Module: Outpcode1.bas ..... H - 46
H.10 Visual Basic Code Module: Probcode.bas ..... H - 64
H.11 Visual Basic Code Module: ReadFile.bas ..... H - 77
H.12 Visual Basic Code Module: Rptable1.bas ..... H - 87
H.13 Visual Basic Code Module: Run_mppr.bas ..... H - 90
H.14 Visual Basic Code Module: Savedata.bas ..... H - 91
H.15 Visual Basic Code Module: Sitecode.bas ..... H - 92
H.16 Visual Basic Code Module: Types.bas ..... H - 96
H.17 Visual Basic Code Module: Utility.bas ..... H - 97
H.18 Visual Basic Code Module: Utility_Rosette.bas ..... H - 102
H.19 Visual Basic Form Module: About.frm ..... H - 110
H.20 Visual Basic Form Module: Circle.frm ..... H - 113
H.21 Visual Basic Form Module: Cmndlgf.frm ..... H - 127
H.22 Visual Basic Form Module: Densform.frm ..... H - 137
H.23 Visual Basic Form Module: frmCalculate.frm ..... H - 139
H.24 Visual Basic Form Module: frmChooseDir.frm ..... H - 144

Figures

Figure 2-1 File and Directory Structure 4
Figure 3-1 Parts of the Screen 5
Figure 3-2 Toolbar Menu 6
Figure 3-3 Site Form 8
Figure 3-4 Outside of the Continental U.S. Notification 9
Figure 3-5 Previous Site Data 10
Figure 3-6 Problem Form 11
Figure 3-7 Rosette Definition 12
Figure 3-8 Radial Unit Conversion 13
Figure 3-9 Economic-Regions Form 15
Figure 3-10 Example Economic-Regions Form 16
Figure 3-11 Example Economic-Regions Rosette 17
Figure 3-12 Previous Problem Data 18
Figure 3-13 Calculation Error Message 18
Figure 3-14 Calculate Form 19
Figure 3-15 Stop Calculation Confirmation 19
Figure 3-16 Population Rosette 20
Figure 3-17 Population-Table Form 21
Figure 3-18 Cumulative-Population-Table Form 22
Figure 3-19 MACCS2 Site-Data File Form 23
Figure 3-20 CSV Site-Data File Form 26
Figure 3-21 Setup Form 31
Figure 3-22 Exit Confirmation Form 32
Figure 5-1 PL94-71 README File dBase to ASCII Conversion Program 45
Figure 5-2 Commercial Data Engine User Interface 46
Figure 5-3 Output for a Single Block 47
Figure 5-4 Verification Directory Structure 52
Figure 5-5 Developed Land Trends 58
Figure 5-6 Median Housing Values, 1991 through 1999 59
Figure 5-7 VFAus, United States Farm Business Assets Trend 59
Figure 5-8 Median Lot Size for American Homes 60
Figure 5-9 Millstone Rosette 63
Figure 5-10 Monticello Rosette 64
Figure 5-11 Diablo Canyon Rosette 64
Figure 5-12 San Onofre Circles Comparison 66
Figure 5-13 San Onofre Rosette Visualization Example 68
Figure 5-14 Peach Bottom Rosette Visualization Example 69
Figure 5-15 Peach Bottom Circles Comparison 69
Figure 5-16 Columbia Power Station Rosette 72
Figure 5-17 Limerick Rosette 73
Figure 5-18 River Bend Rosette 73

Figure 5-19 St. Lucie Rosette 74
Figure B-1 MACCS/MACCS2 sample Site Data file ..... B - 3

Tables

Table 3.1 Example MACCS2 Site-Data File 24
Table 3.2 Example CSV Site-Data File 26
Table 4.1 County Level Data File Variables and Data Sources 36
Table 4.2 Excerpt From the 1990 County-Level Database 39
Table 5.1 dBase III Census Record Format 41
Table 5.2 *CENSUS00.DAT* Verification Summary 49
Table 5.3 Test 2 Results: State-by-State Comparison 50
Table 5.4 Test 3 Results: County-by-County Comparison 51
Table 5.5 Ten Mile Comparison with Census Bureau Using SECPOP90 Program and CENSUS00.DAT file 53
Table 5.6 Census of Agriculture Data Items 55
Table 5.7 Variation in COUNTY90.DAT and COUNTY97.DAT Parameters 57
Table 5.8 Total Reported and Calculated Populations 62
Table 5.9 San Onofre Circles Comparison Cumulative Population 65
Table 5.10 San Onofre Sector by Sector Comparison 67
Table 5.11 Total Reported and Calculated Populations for a 60-Mile Radius 70
Table A.1 SECPOP2000 Site File Listing ..... A - 1
Table D.1 1993 & 1994 Statistical Abstract of the U.S. ..... D - 1
Table D.2 NUREG/CR-4551 Volume 2, Revision 1, Part 7 ..... D - 1
Table D.3 1994 County and City Data Book ..... D - 1
Table E.1 Fixed Reproducible Tangible Wealth ..... E - 1
Table E.2 Non-Federal Land Use ..... E - 2
Table E.3 Example Housing Value Table ..... E - 3
Table E.4 Farm Business Balance Sheet ..... E - 4
Table E.5 Lot Size Example ..... E - 5

1 Introduction

This report is divided into five sections. Section 1 is this introduction. Section 2 describes the installation of the SECPOP2000 software on a personal computer (PC). Section 3 is the user's guide, which explains how to use the software in a step-by-step manner. Section 4 describes the algorithms used to perform the population, land fraction, and economic estimates. Section 5 discusses the software design, development, and verification. Appendices describe input and output file formats and the conversion of U.S. Bureau of Census data (from CD-ROMs) to the block census and county economic data files that are included with SECPOP2000.

1.1 Background

In 1973 W. Athey of the U.S. Environmental Protection Agency (EPA) wrote a computer program called SECPOP that calculated population estimates. This program was intended to run on mainframe computers. It used 1970 population census data and was created to support a study on air quality. The Nuclear Regulatory Commission (NRC) adopted this program to perform siting reviews for nuclear power plant construction and to evaluate license applications.

Two things changed that suggested the need for updating the original program — more recent population censuses data became available and PC usage became widespread. The current version, SECPOP2000, can perform calculations with either 1990 or 2000 population data and runs on PCs running Windows 98™, NT™, 2000™, and XP™.

SECPOP90 (Reference 1) supports two types of analysis: site and regional. The site analysis provides population and economic data estimates for any location within the continental United States. Calculation results can be displayed, printed, or saved as a rosette, a table, a MACCS2 (MELCOR Accident Consequence Code System) site file, or a MACCS2-like comma-separated-variable file. Regional analysis assesses compliance of available sites against siting parameters (i.e., specific population density criteria). Regional analysis is not supported in SECPOP2000 because some of the software packages originally used for regional analysis are no longer available.

SECPOP2000 uses the latest (2000) census data, which provide greater resolution than the 1980 and 1990 census data. This greater resolution is achieved by employing over eight million census-block data points and performing the sector population estimates directly from these points. This is an increase of approximately two million data points over the 1990 census data.

1.2 Hardware and Software Requirements

SECPOP2000 was developed to run on personal computers that run the Windows 98™, NT™, 2000™, and XP™ operating systems. The SECPOP2000 software requires approximately 170 megabytes of hard-disk storage.

Minimum System Requirements:

Pentium™ or faster Intel™ processor;
Windows 98™, Windows NT™ 4.0 or later, Windows 2000™, or Windows XP™
64 MB of RAM;
170 MB of available hard-disk space;
CD-ROM drive; and
Minimum screen resolution of 800x600 (recommended)

2 Installation

SECPOP2000 can be installed on PCs running the Windows 98™, NT™, 2000™, and XP™ operating systems. If the user later decides to uninstall SECPOP2000, the uninstall process automatically leaves in place any files added or modified since the original installation.

To start the installation process, first exit or quit all other programs and insert the SECPOP2000 CD into the CD drive. If AutoRun is enabled, the installation program starts automatically. If it doesn't start automatically, open Windows Explorer, navigate to the CD drive, and double click on Setup. After answering a few basic questions, the program installs SECPOP2000. This process takes a few minutes to complete due to the large data files that must be copied to the hard drive. You may need to reboot your computer as part of the installation process.

When installation is complete, the directory contains the files and sub-directories shown in Figure 2-1. (The default main directory is C:\Program Files\SECPOP. The root of the directory is different if you chose another directory during installation.) For more information about these files see Subsections 3.7 and 4.4.

2.1 Initialization

Although the user can read or save input or output files to any directory, the program is set up during installation to initially access these files from the directories shown in Figure 2-1. The user can temporarily change the current directories from which information is accessed or stored by using the file dialogs that appear when saving or accessing files; the user can permanently change the default directories by using the Setup form (see Subsection 3.8).

All SECPOP2000 output files can be printed directly from within SECPOP2000 or from any word processor or other program capable of opening ASCII text files. The population tables and the population rosette (see Subsection 3.6) can also be printed from within SECPOP2000.

2.2 Starting SECPOP2000

SECPOP2000 can be started either by choosing SECPOP2000 from the *Start*, *Programs* menu or by double clicking the SECPOP2000 icon that is placed on the desktop during the installation process. (This icon can be deleted by dragging it to the Recycle Bin.)

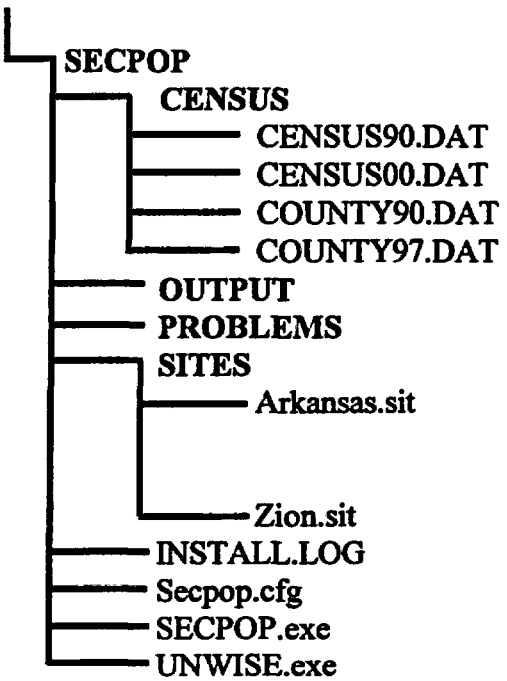


Figure 2-1 File and Directory Structure

3 User's Guide

SECPOP2000, Version 3.12, allows the user to perform population estimation calculations related to a specific site; SECPOP90 allowed the user to perform calculations related to a specific site and calculations related to a region of the continental United States. The ability to perform calculations related to a region may be added to SECPOP2000 in a future version.

The site calculation requires, among other things, a specific latitude and longitude to define the location of the site. This is typically the reactor center for a single unit site or the center point between reactors at a multi-unit site. (A small difference in location is not expected to significantly affect SECPOP2000 results or subsequent MACCS2 results.)

SECPOP2000 starts with the initial screen shown in Subsection 3.1. Site-specific calculations are described in Subsections 3.3 through 3.6. Descriptions on how to navigate through the SECPOP2000 menus and forms are provided in Subsections 3.2 and 3.3 respectively.

3.1 Parts of the Main Screen

The SECPOP2000 main screen is divided into four major sections (see Figure 3-1). Section (1), *Main Screen Title Bar*, displays the name of the SECPOP2000 program, the version number, and the Census file currently selected on the Setup form (see Subsection 3.9). Section (2), *Main Menu*, is the SECPOP2000 main menu. Section (3), *Toolbar Menu*, contains shortcuts for commonly used main menu items. Section (4), *Work Areas*, is the area where various other forms and messages appear. Most of these require the user to provide some information or make a decision. This area may be expanded to allow viewing of several forms at the same time (if the screen resolution is greater than 80OX600). The *Welcome to SECPOP2000* screen shown below closes when the user clicks on the OK button, or selects a command from the main menu.

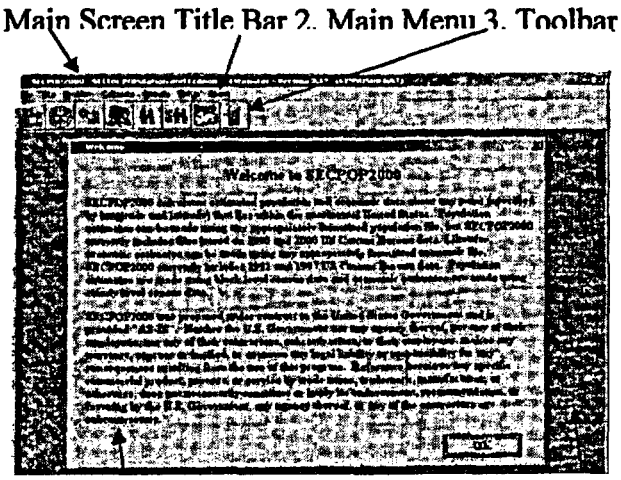


Figure 3-1 Parts of the Screen

3.2 Menus

3.2.1 Main Screen

The main screen consists of six menus or commands:

File	exit the program;
Site	site characteristics for site-specific calculations;
Problem	additional non-site-specific parameters for site-specific calculations;
Calculate	perform site-specific or regional (inoperative in SECPOP2000) calculations;
Results	results from present or previous calculation;
Setup	customize the SECPOP2000 environment; and
About	information about SECPOP2000.

The order of the selections above outlines the order of how you might solve a site-specific problem. 1) define the site parameters such as latitude and longitude using the *Site* menu. 2) define additional problem parameters such as the radius of the exclusion area using the *Problem* menu. 3) perform the calculations using the *Calculate* menu. 4) examine the results using the *Results* menu.

Additionally, the *Setup* command allows you to modify the SECPOP2000 environment. Default pathnames for site, problem, output, and database files may be set in the *Setup* form. The About menu displays brief information about the SECPOP2000 program. The *File, Exit* menu option exits the SECPOP2000 program.

3.2.2 Toolbar Menu

The toolbar menu shown in Figure 3-2 contains the commands that are most commonly used:

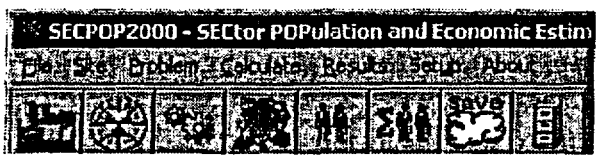
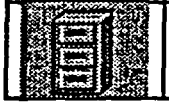


Figure 3-2 Toolbar Menu

Hovering the cursor over each icon shows a tool tip that describes function invoked when clicking on the icon.

	Open a previously saved site.
	Open a previously saved problem.
	Perform a site calculation.
	Show the population rosette.
	Show the population table.
	Show the cumulative population table.
	Save the results in a MACCS2 formatted file.
	Show the setup form.

3.3 Site Definition

Site definition consists of specifying the site name, latitude, longitude, and any desired site-specific remarks. The Site menu has four available commands. Note that some of these commands are not available at all times. For example, *Save Site* and *Save As Site* are not enabled if there is no site information to save. The four Site menu commands are:

New Site	enter the site-specific parameters for a new site;
Open Site	open a previously saved site;
Save Site	save the site information with the present filename; and
Save As Site	save the site information with a new filename.

3.3.1 New Site

Selecting New Site opens the blank form shown in Figure 3-3.

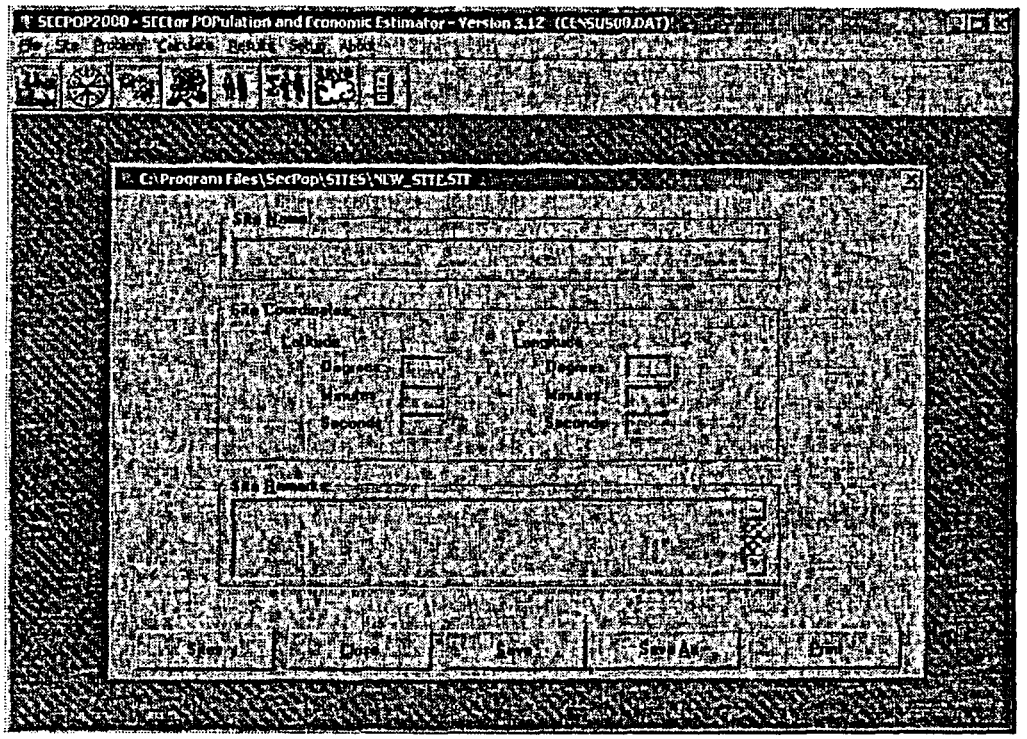


Figure 3-3 Site Form

Site Form Fields

The Site form consists of the following fields that are designed so the user may easily edit all of the site-specific information (fields are designated required or optional depending on whether they are required to perform site-specific calculations):

Title Bar	contains the filename of the site being edited. If this is a new problem, then the default filename "NEW_SITE.SIT" is used. If the Site form is edited, then the phrase "(Modified)" is displayed after the site filename. The filename is automatically updated when the user performs a Save or Save As command (see the next section).
Site Name	a descriptive name of the site. Up to eighty characters (both rows of the field in the form) may be used to describe the site. This field is displayed or printed whenever the site specific information is displayed or printed. Only the first 40 of those characters (the first row) are printed on the first line of the MACCS2 Site Data File. This field is optional.
Latitude	the latitude of the site in degrees, minutes, and seconds. Zeros (0s) must be entered if the minutes or seconds are equal to zero (0). This field is required.
Longitude	the longitude of the site in degrees, minutes, and seconds. Zeros (0s) must be entered if the minutes or seconds are equal to zero (0). This field is required.

SECPOP2000 checks to see if the coordinates of the site lie outside of the Continental U.S. by comparing the coordinates to the minimum and maximum coordinates of the continental U.S. If the coordinates are determined to be outside of the continental U.S., then SECPOP2000 displays the error message shown in Figure 3-4.

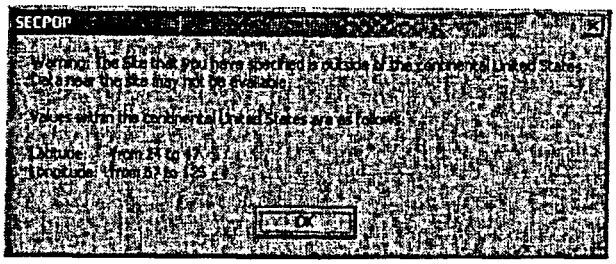


Figure 3-4 Outside of the Continental U.S. Notification

Note: this limited checking is meant to alert the user to gross typing errors. The absence of the above message does not guarantee that the site information is correct. The user should always double-check the information in the required fields with the original sources.

Site Remarks descriptive remarks about the site. Up to 80 characters (both rows of the field in the form) may be used to describe the site. This field is displayed or printed when viewing this form or printing the site-specific information (see the Print command in the next section). One use of this space could be to record plant specific information. For example:

Unit I -- GE BWR 6 with Mark III containment, licensed in 1984

This field is optional.

Site Form Commands

At the bottom of the *Site* form is a row of five command buttons. The commands support the opening, closing, saving, and printing of the *Site* form. The command buttons are as follows:

Sites opens a previously defined site file. This command is equivalent to the *Open Site* command of the *Site* menu. For a more detailed explanation see Subsection 3.3.2.

Close closes the *Site* form. All information on the form is saved in memory and can be reviewed by opening the *Site* form again. The information is lost unless the site information is stored in a file (see the *Save* and *Save As* commands below) before SECPOP2000 is exited. The user may close the *Site* form or leave it open.

Save saves the site information using the filename displayed in the *Site* form *Title Bar* as an ASCII text file. This command is equivalent to the *Save Site* command of the *Site* menu.

Save As	saves the site information using a filename specified by the user. This command is equivalent to the *Save As Site* command of the *Site* menu. For a more detailed explanation see Subsection 3.3.4.
Print	prints the site-specific information to a user-specified printer.

3.3.2 Open Site

Selecting *Open Site* allows the user to select a site that has been previously defined and saved to a file. The user may select from the list of previously defined sites (contained in the SECPOP\SITES directory) or a personal list of sites. Once the site is chosen and the *Open* button is selected, the file is loaded into memory and the *Site* form is displayed with the saved information. The user may close the *Open Site* form or leave it open.

All predefined sites have the default filename extension sit. The user is not required to follow this practice. Doing so, however, makes it easier to find previously defined site files. This is especially true when a single directory has an assortment of site, problem, and results files.

If either *New Site* or *Open Site* is selected from the *Site* menu and site data have already been entered or loaded from a file, SECPOP2000 first asks the user if the previously entered or loaded data should be reviewed before overwriting them. SECPOP2000 displays the message shown in Figure 3-5.

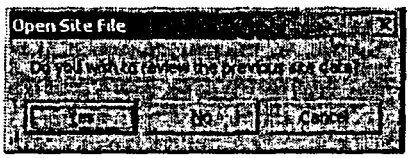


Figure 3-5 Previous Site Data

If the user selects *Yes*, then the *New Site* or *Open Site* operation is canceled and the *Site* form with the old data is displayed. If the user selects *No*, then the old data is overwritten with a blank form in the case that *New Site* was selected or with the data from the site file that was opened. Selecting *Cancel* cancels the *New Site* or *Open Site* operation.

3.3.3 Save Site

Selecting *Save Site* saves the site information using the filename displayed in the *Site* form *Title Bar*. After the file is saved, the *Title Bar* of the *Site* form is updated. This command is equivalent to the *Save* command found on the *Site* form (see Figure 3-3).

3.3.4 Save As Site

Selecting *Save As Site* allows the user to save the site information using a user-specified filename. The *Title Bar* of the *Site* form is then updated with the new filename. Selecting *Cancel* cancels the

Save As Site operation and returns the user back to the main menu. This command is equivalent to the *Save As* command found on the *Site* form (see Figure 3-3).

3.4 Problem Definition

Problem definition consists of specifying the radii of interest, the population multiplier, the name of the site file that contains the site-specific information for the problem, mapping the problem sectors to economic regions, and any desired problem-specific remarks. The *Problem* menu has four available commands. Note that some of these commands are not available at all times. For example, *Save Problem* and *Save As Problem* are not enabled if there is no problem information to save. The four *Problem* menu commands are:

New Problem opens the problem-specific parameter form for a new problem;
Open Problem opens a previously saved problem;
Save Problem saves the problem information with the present filename; and
Save As Problem saves the problem information with a new filename.

3.4.1 New Problem

Selecting *New Problem* opens the form shown in Figure 3-6. The *Problem* form allows the user to enter or edit all problem-specific information.

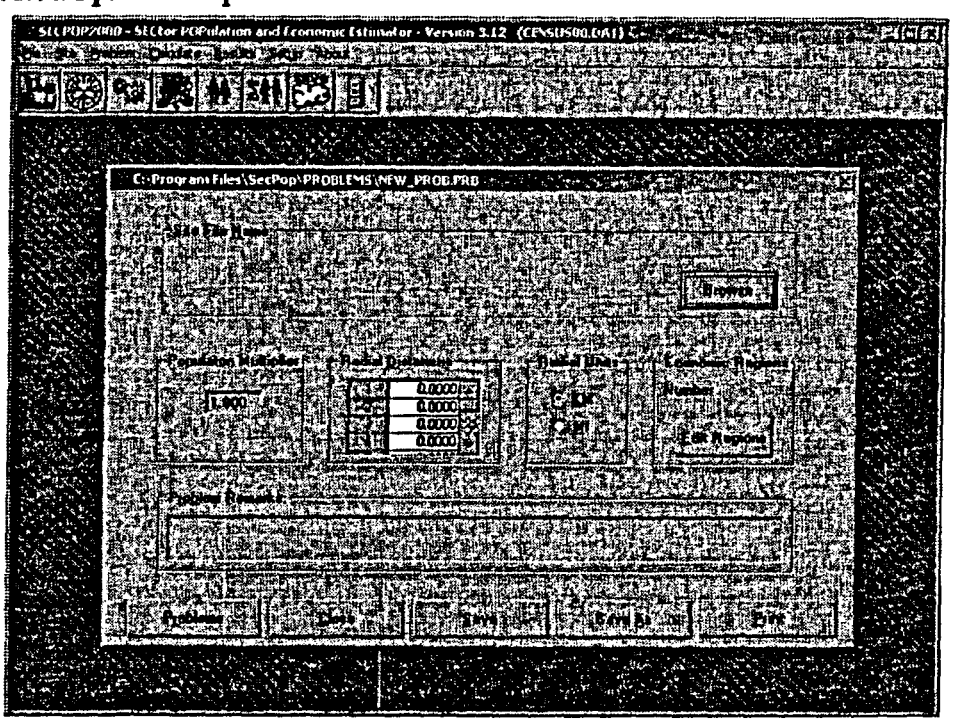


Figure 3-6 Problem Form

Problem Form Fields

The *Problem* form (see Figure 3-6) consists of the following fields that are designed so the user may easily edit all of the problem-specific information (fields are designated required or optional depending on whether they are required to perform site-specific calculations):

Title Bar	contains the filename of the problem being edited. If this is a new problem, then the default filename "NEW_PROB.PRB" is used. If the *Problem* form is edited, then the phrase "(Modified)" is displayed after the problem filename. The *Title Bar* is automatically updated when the user performs a *Save* or *Save As* command.
Site File Name	contains the name of the site file associated with this problem. This field is required.
Population Multiplier	is a factor that is applied to the census population data. This field allows manipulation of the census estimates by a uniform multiplication factor. For example, if it is expected that in the year 2005 the overall population will have increased by 10%, a population scale factor of 1.1 could be used to represent this growth. The default value is 1.0. This field is required. It must have a value between 0.0 and 100.0.
Radial Distances	are the radii of interest for the problem. The geometry of SECPOP2000 problems is based on that of MACCS2. The area around a site is divided into sixteen compass sectors that are equivalent to a standard navigational compass rosette. The rosette is then further divided into radial rings, specified according to the *Radial Distances* field. Figure 3-7 shows a rosette that has nine user-defined radial rings. The site itself is located at the very center of the rosette.

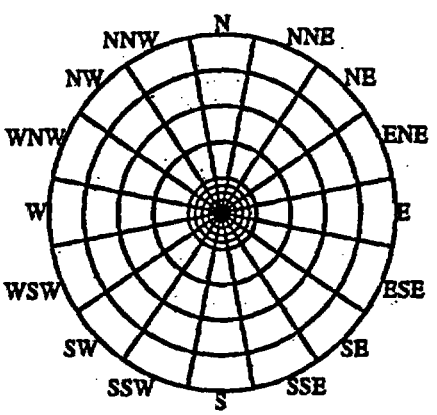


Figure 3-7 Rosette Definition

The collection of directional lines and radii naturally divide the rosette into sections that are defined by two adjacent radii and two neighboring circles. Normally the innermost circle approximates the exclusion area boundary of the site; the second radius often approximates the low population zone. These sections are mapped into economic regions by using the *Edit Regions* command described below.

The radial distances can be expressed in miles or kilometers (see the *Radial Units* field described below). The user is limited to 35 radii to maintain compatibility with the MACCS2 Site Data File format. At least two radii must be defined. All radii must be separated from each other by at least 0.05 km (0.0311 mi.). This field is required.

Radial Units, MI — indicates that the radial distances are in miles. If the radial distances are in kilometers and this option is selected, then the user is given the option of performing a conversion from kilometers to miles or only changing the unit designation. This is accomplished using the form shown in Figure 3-8. Either miles or kilometers must be selected.

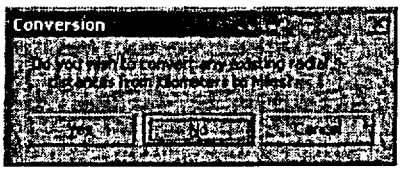


Figure 3-8 Radial Unit Conversion

Radial Units, KM — indicates that the radial distances are in kilometers. If the radial distances are in miles and this option is selected, then the user is given the option of performing a conversion from miles to kilometers or only changing the unit designation. This is accomplished using a form similar to that shown in Figure 3-8. Either kilometers or miles must be selected.

Problem Remarks — descriptive remarks about the problem. Up to eighty characters (both rows of the field in the form) may be used to describe the problem. This field is displayed or printed when viewing this form or printing the problem-specific information (see the Print command in the next section). One use of this space could be to record the source of problem-specific information. For example:

Grand Gulf Nuclear Station (GGNS) Unit: I Emergency Plan Section 2.2

This field is optional.

Problem Form Commands

The *Problem* form (shown in Figure 3-6) form has six command buttons: five at the bottom of the *Problem* form similar to the *Site* form and an additional command button for editing the economic regions. The commands are executed when the buttons are selected. The commands support editing of the economic regions and the opening, closing, saving, and printing of the *Problem* form. The command buttons are as follows:

Edit Regions	opens the *Economic Regions* form (see Figure 3-9) and allows editing of the economic regions. This form is described in detail in Subsection 3.4.2.
Problems	opens a previously defined problem file. This command is equivalent to the *Open Problem* command of the *Problem* menu. For a more detailed explanation, see Subsection 3.4.3.
Close	closes the *Problem* form. All information on the form is saved in memory and can be reviewed by opening the *Problem* form again. The information is lost unless the problem information is stored in a file (see the *Save* and *Save As* commands below) before SECPOP2000 is exited. The user may close the *Problem* form or leave it open.
Save	saves the problem information using the filename displayed in the *Problem* form *Title Bar*. This command is equivalent to the *Save Problem* command of the *Problem* menu.
Save As	saves the problem information using a filename specified by the user in an ASCII text file. This command is equivalent to the *Save As Problem* command of the *Problem* menu. For a more detailed explanation, see Subsection 3.4.5.
Print	prints the problem-specific information to a user-specified printer.

3.4.2 Economic Regions

Selecting Edit Regions from the Problem opens the form shown in Figure 3-9. The Economic Regions form allows the user to assign economic regions to the rosette sections or edit the present assignments.

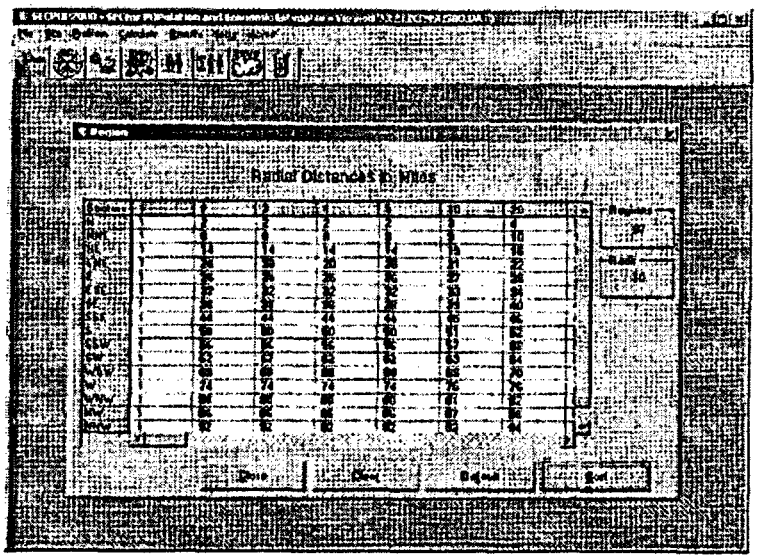


Figure 3-9 Economic-Regions Form

Economic Regions Form Fields

The *Economic Regions* form consists of the following fields that are designed so the user may assign or edit the economic regions. All of the fields on this form are required and are initially filled in by SECPOP2000. All fields except for the exclusion area (i.e., for the innermost circle) are editable by the user. These fields are designated as "User input field," in the descriptions below.

Radial Distances	the first row of the economic-regions table contains the radial distances in the units specified on the *Problem* form (see Figure 3-7).
Sectors	the first column of the economic-regions table contains the rosette direction descriptors. These descriptors correspond to the sixteen standard directions of a navigational compass.
Regions	number of economic regions. The minimum number of allowed economic regions is 1. The maximum number of allowed regions is 99.
Radii	number of radii. The minimum number of allowed radii is 2. The maximum number of allowed radii is 35. The radii are defined on the *Problem* form.
Rosette Sections	the remaining values in the economic-regions table are the economic-region designators for the displayed rosette sections. These values correspond to the outer radius shown at the top of the column and to the inner radius shown at the top of the previous column. Economic region # 1 is automatically assigned to all of the sections that lie within the innermost ring (circle). This ring is normally thought of as the exclusion area boundary and economic region # 1 is thought of as the exclusion area.

To assign economic regions to the remaining rosette sections, enter a number for that region that is greater than or equal to 2 and less than or equal to 99. That rosette section then belongs to that economic region. If, for example, the user enters a 2 in the third column, second row of the economic regions table shown in Figure 3-10, then the rosette section defined by the first and second radii in the northern direction would be part of economic region 2. An economic region number must be assigned to each rosette section. There is no need for the same economic region to occupy adjacent cells or for all economic regions to occupy the same number of cells. This allows the user to assign a single economic region to features that might occur repeatedly over the rosette. For example, a user could assign the same economic regions to lakes or desert areas.

Figures 3-10 and 3-11 illustrate the mapping of economic regions onto a rosette for a very simple situation. In this case the economic regions occupy concentric rings, as shown in Figure 3-11. The innermost ring (circle) is always designated as economic region #1 and normally corresponds to the exclusion area, as indicated in the legend of Figure 3-11.

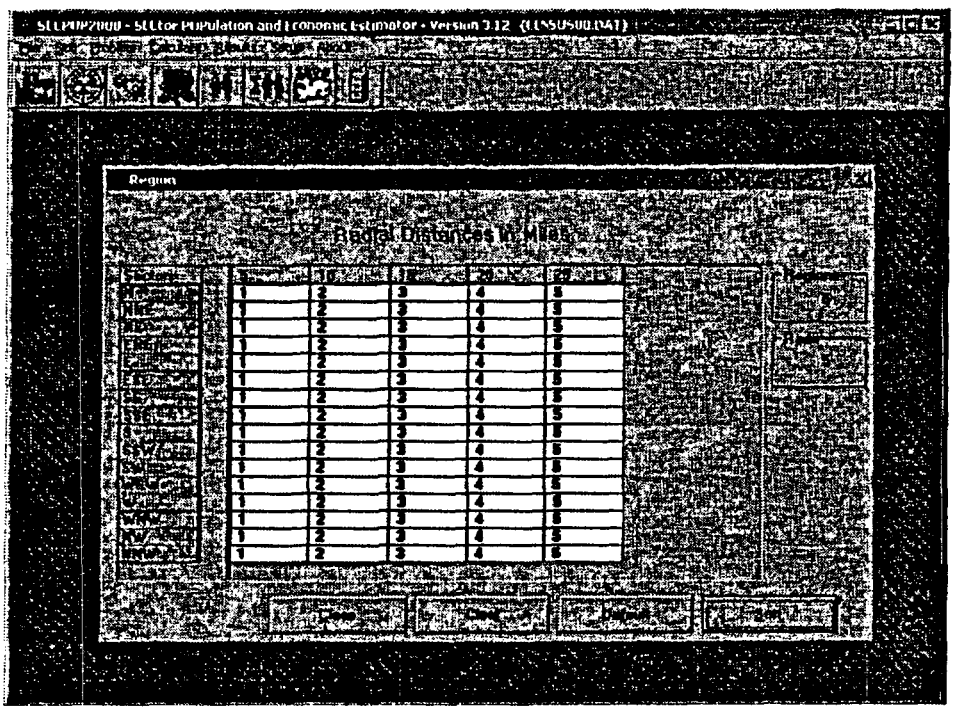


Figure 3-10 Example Economic-Regions Form

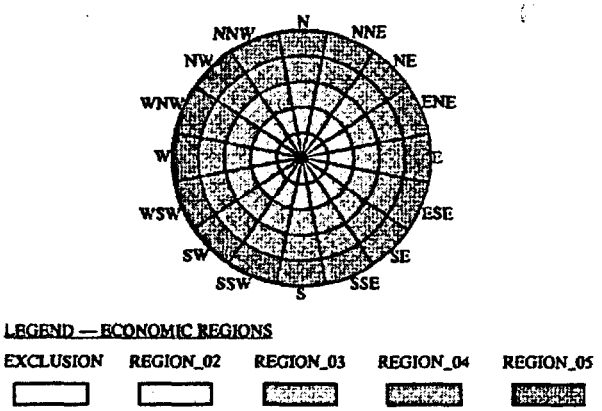


Figure 3-11 Example Economic-Regions Rosette

Economic Regions Form Commands

The *Economic Regions* form has three command buttons for performing various operations on the economic regions and a button for closing the *Economic Regions* form as follows:

Close	sorts the economic regions and closes the Economic Regions form. To save the edited economic regions, the user must select *Save* or *Save As* from the *Problem* form (see Figure 3-6) or *Save Problem* or *Save As Problem* from the *Problem* menu.
Clear	clear all values from all of the rosette sections except for economic region 1, which is always the section from the center of the rosette to the first radius and is normally considered to be the exclusion area.
Default	restores the economic regions to the default values shown in Figure 3-10.
Sort	renumbers economic regions so that there are no gaps in the numbering of the regions and the regions are sorted in row-column order. Sorting the economic regions is not required but it usually makes the economic regions mapping easier to understand.

3.4.3 Open Problem

Selecting *Open Problem* from the main menu or from the toolbar menu allows the user to select a previously defined problem file.

While users are not required to follow this practice, it is recommended that they append the filename extension .prb to all problem files. Doing so makes it easier to find previously defined problem files. This is especially true when a single directory contains an assortment of site, problem, and result files.

If either *New Problem* or *Open Problem* is selected from the *Problem* menu and problem data has already been entered or loaded from a file, SECPOP2000 first asks the user if the previously entered

or loaded data should be reviewed before overwriting it. SECPOP2000 displays the message shown in Figure 3-12.

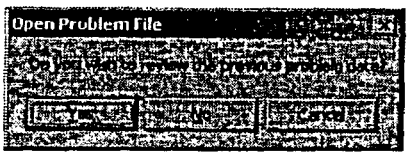


Figure 3-12 Previous Problem Data

If the user selects *Yes*, then the *New Problem* or *Open Problem* operation is canceled and the *Problem* form with the old data is displayed. If the user selects *No*, then the old data is overwritten with a blank form or the new data. Selecting *Cancel* cancels the *New Problem* or *Open Problem* operation and returns the user back to the main menu.

3.4.4 Save Problem

Selecting *Save Problem* saves the problem information using the filename displayed in the *Problem* form *Title Bar* as an ASCII text file. After the file is saved, the *Title Bar* of the *Problem* form is updated. This command is equivalent to the *Save* command found on the *Problem* form (see Figure 3-7).

3.4.5 Save As Problem

Selecting *Save As Problem* allows the user to save the problem information as a user-specified ASCII text file. The *Title Bar* of the *Problem* form is then updated with the new filename. Selecting *Cancel* cancels the *Save As Problem* operation. This command is equivalent to the *Save As* command found on the *Problem* form (see Figure 3-6).

3.5 Performing Site-Specific Calculations

Once the *Site*, *Problem*, and *Economic Regions* forms are complete, a site-specific calculation may be performed by selecting the *Calculate* menu from the main menu and then selecting *Site-Specific* from the *Calculate* menu. If there is not enough information to perform a site-specific calculation, the error message shown in Figure 3-13 is displayed, and the user must provide the necessary information before the calculation can proceed.

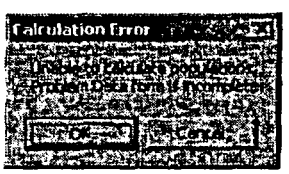


Figure 3-13 Calculation Error Message

If all of the forms have been sufficiently completed, the calculations begin and the *Calculate* form shown in Figure 3-14 is displayed.

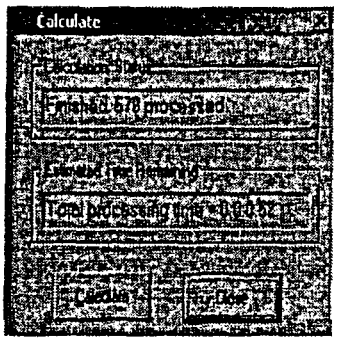


Figure 3-14 Calculate Form

Calculate Form Fields

The *Calculate* form consists of the following fields that are designed so the user may monitor the progress of the calculation. All of the fields are filled in automatically by SECPOP2000 and cannot be edited by the user.

Calculation Status	the current status of the calculation. It gives a running total of the number of records that have been processed.
Estimated Time Remaining	an estimate of how much time remains before the calculation is complete.

Calculate Form Commands

The *Calculate* form has two command buttons that are used to control the calculation and close the *Calculate* form. The commands are executed when the buttons are selected. The command buttons are as follows:

Calculate	begins the site-specific calculation; and
Close	stops the calculation if it is under way. Note: the user is asked to confirm that the calculation should be stopped, as shown in Figure 3-15. If a calculation is not under way, then this command closes the *Calculate* form.

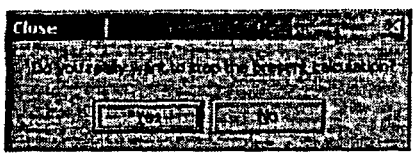


Figure 3-15 Stop Calculation Confirmation

Note: the population estimates are based on over eight million census-block records for 2000 census data (six million for 1990 census data). The economic-factor and land-fraction estimates are aggregated at the county level, which is significantly more granular than the population estimates. Subsection 5.2.1 provides notes and exceptions for the economic data.

3.6 Site-Specific Results

There are several ways to display, print, or save the site-specific calculation results. The results are accessed by selecting the *Results* menu. All of the commands and menus (except for *Regional*) apply to site-specific results and are discussed below. *Regional* calculations are not supported in SECPOP2000.

Population Rosette	displays a rosette with population per section.
Population Tables	displays rosette populations in table formats.
MACCS2-Input File	displays, prints, or saves the MACCS2 Site Data File.
CSV File	displays, prints, or saves the MACCS2 Site Data File in comma-separated-variable format. This file is not directly usable by MACCS2, but may be imported into a spreadsheet or database program.
Regional	is not functional.
Load Previous Results	loads previously saved calculated results.

3.6.1 Population Rosette

Selecting *Population Rosette* from the *Results* menu or from the toolbar menu displays a rosette like the one shown in Figure 3-16. Clicking on the *Print* button sends the rosette image to a user-specified printer. Clicking on the *Save* button saves the rosette to a file in bitmap form. This file may be imported into other applications (such as a word processor).

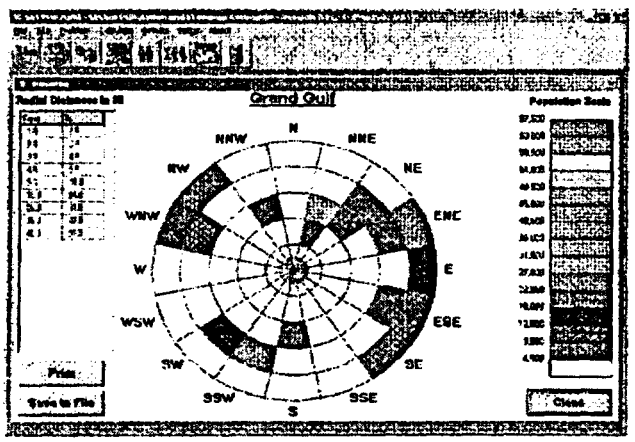


Figure 3-16 Population Rosette

3.6.2 Population Tables

Selecting *Population Tables* from the *Results* menu displays the *Population Tables* sub-menu. This sub-menu provides the user with two methods for displaying the population results: the *Population* command, which displays a tabular version of the rosette with the population values for each rosette section, and the *Cumulative* command, which displays a similar table except that the population is summed from the origin to the current radius for each sector.

Population

Selecting *Population* from the *Population Tables* sub-menu or from the toolbar menu displays a table like the one shown in Figure 3-17. The population data is shown in a manner similar to the method used to display the economic regions on the *Economic Regions* form (see Subsection 3.4.2). The table has no user input fields. There are two command buttons, *Close* and *Print*, that perform the following functions:

Close closes the *Population Table* form.
Print prints the population information to the user-specified printer.

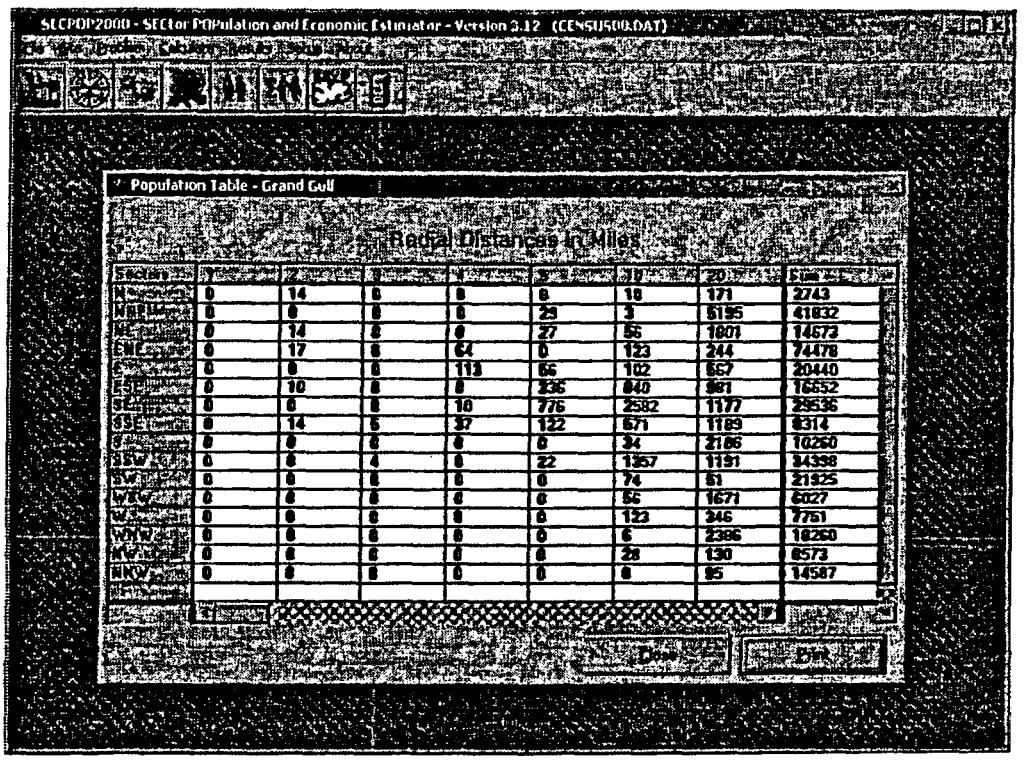


Figure 3-17 Population-Table Form

Cumulative

Selecting *Cumulative* from the *Population Tables* sub-menu or from the toolbar menu displays a table like the one shown in Figure 3-18. The cumulative population data is shown in a manner similar to the one used to display the economic regions on the *Economic Regions* form (see Subsection 3.4.2). The data displayed on this form is summed from the origin outward along each sector. This means that along each sector, the population includes all of the population up to that radius in the given sector.

For example, in Figure 3-18 the cumulative population inside the 5-mile radius along the eastern direction is 169. This value includes the population of 113 found between the 3- and 4-mile radii and the population of 56 between the 4- and 5-mile radii (see also Figure 3-17). The table has no user input fields. There are two command buttons, *Close* and *Print*, that perform the following functions:

Close	closes the Cumulative Population Table form.
Print	prints the cumulative-population information to the user-specified printer.

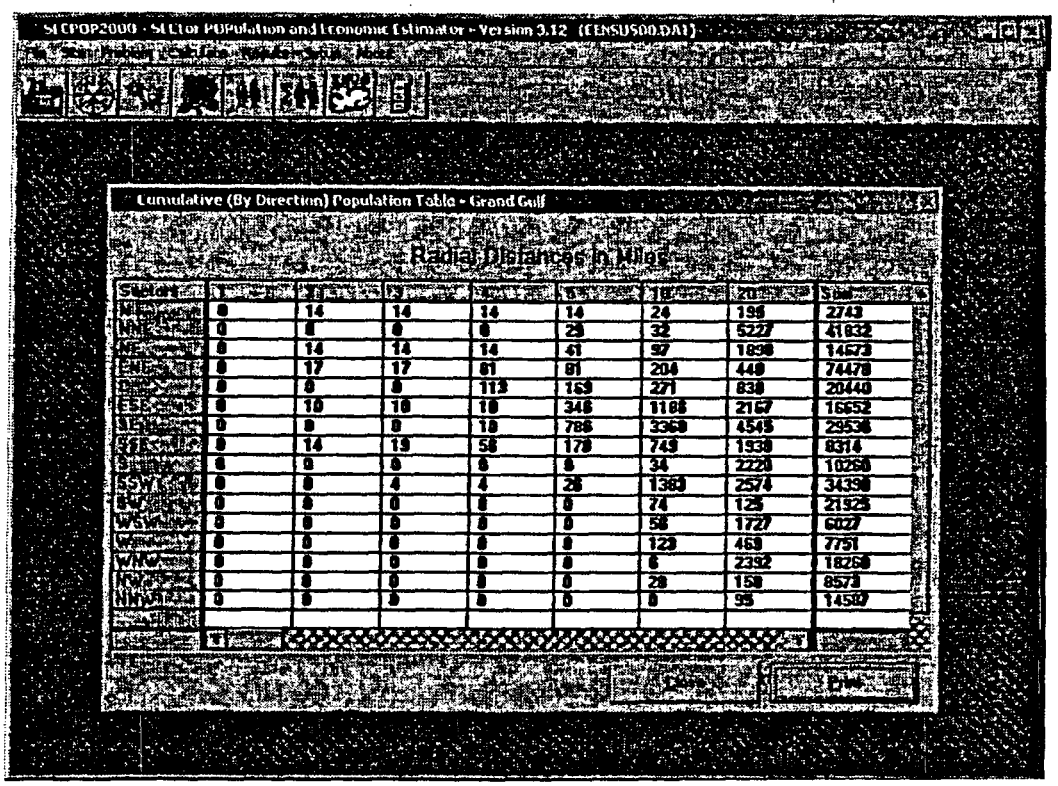


Figure 3-18 Cumulative-Population-Table Form

3.6.3 MACCS2 Input File

Selecting *MACCS2 Input File* (also referred to in MACCS2 documentation as the Site Data File) from the *Result* menu displays the *MACCS2 Input File* sub-menu. This menu allows the user to display, print, or save the results in MACCS2 input-file format.

Display

Selecting *Display* from the *MACCS2 Input File* sub-menu displays the *MACCS2 Site Data File* form shown in Figure 3-19. There are no user-input fields on this form. The user can scroll through the data using the scroll button. The command *Close* closes the form. The MACCS2 Input File is displayed as ASCII text.

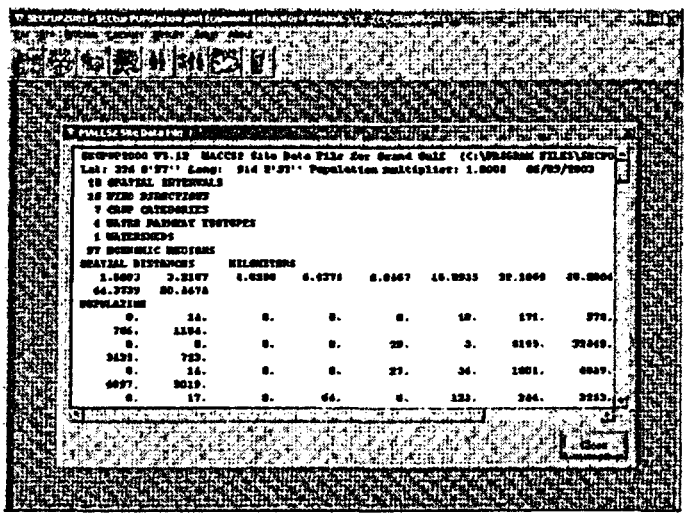


Figure 3-19 MACCS2 Site-Data File Form

Print

Selecting Print from the *MACCS2 Input File* sub-menu prints the MACCS2 Site-Data File to the user-specified printer. The MACCS2 Input File is printed as ASCH text.

Table 3.1 shows an example MACCS2 Site-Data File. SECPOP2000 does not perform any calculations for estimating crop categories, water pathway isotopes, or watersheds. The values below for those categories (watershed index, crop season and share, and watershed definition) are default values generated so that the file created is completely compatible with MACCS2. For a full explanation of all of the data fields, see Appendix B.

Table 3.1 Example MACCS2 Site-Data File

```
SECPOP2000 V3.12 MACCS2 Site Data File for Grand Gulf
Lat: 32d 0'27'' Long: 91d 2'53'' Population multiplier: 1.0000 06/19/2002
 10 SPATIAL INTERVALS
 16 WIND DIRECTIONS
  7 CROP CATEGORIES
  4 WATER PATHWAY ISOTOPES
  1 WATERSHEDS
 97 ECONOMIC REGIONS
"SPATIAL DISTANCES KILOMETERS"
 1.00 2.00 3.00 4.00 5.00 10.00 20.00 30.00
 40.00 50.00
"POPULATION"
 0. 0. 21. 0. 0. 0. 0. 42.
 551. 175.
 0. 0. 0. 0. 0. 19. 0. 2276.
 16358. 19372.
 0. 0. 15. 0. 0. 37. 264. 951.
 3426. 2219.
 0. 0. 36. 0. 0. 115. 137. 131.
 721. 3265.
 2. 0. 0. 0. 8. 197. 93. 418.
 704. 986.
 0. 0. 21. 0. 0. 556. 728. 440.
 116. 348.
 0. 0. 0. 0. 1. 2904. 806. 653.
 287. 329.
 0. 0. 4. 8. 0. 185. 1027. 373.
 192. 326.
 0. 0. 0. 0. 0. 0. 390. 1194.
 3483. 1008.
 0. 1. 0. 2. 0. 2. 1281. 307.
 650. 2487.
 0. 0. 0. 0. 0. 0. 1126. 108.
 1151. 453.
 0. 0. 0. 0. 0. 0. 273. 1516.
 296. 223.
 0. 0. 0. 0. 0. 40. 504. 65.
 156. 244.
 0. 0. 0. 0. 0. 0. 1356. 669.
 255. 174.
 0. 0. 0. 0. 0. 0. 71. 103.
 32. 39.
 0. 0. 5. 0. 0. 0. 1. 44.
 237. 9750.
"LAND FRACTION"
0.97 0.00 0.97 0.00 0.00 0.97 0.95 0.95 0.96 0.96 0.00 0.00 0.00 0.00 0.00 0.97 0.95 0.95 0.95
0.95 0.00 0.00 0.97 0.00 0.00 0.97 0.96 0.95 0.95 0.97 0.00 0.97 0.97 0.00 0.00 0.97 0.97 0.97
0.99 0.99 0.97 0.00 0.00 0.00 0.97 0.97 0.97 0.97 0.99 0.99 0.97 0.97 0.97 0.00 0.97 0.97 0.97
0.97 0.99 1.00 0.00 0.00 0.00 0.00 0.97 0.97 0.97 0.97 0.98 0.99 0.97 0.00 0.97 0.97 0.00 0.97
0.97 0.99 0.99 0.99 0.00 0.97 0.00 0.00 0.00 0.97 0.98 0.99 0.99 0.99 0.00 0.97 0.00 0.97 0.97
0.97 0.97 0.99 0.99 0.96 0.00 0.00 0.00 0.00 0.00 0.97 0.97 0.97 0.98 0.94 0.00 0.00 0.00 0.97
0.00 0.94 0.95 0.94 0.94 0.94 0.00 0.00 0.97 0.00 0.94 0.94 0.94 0.94 0.94 0.97 0.00 0.00 0.00
0.00 0.00 0.96 0.94 0.94 0.94 0.96 0.00 0.00 0.00 0.97 0.00 0.95 0.94 0.94 0.94 0.96 0.00 0.00
0.97 0.00 0.00 0.95 0.94 0.95 0.96 0.96
"REGION INDEX"
1 2 2 2 2 3 4 5 6 7
1 8 8 8 8 910111213
1141414141516171819
1202020202122232425
1262626262728293031
1323232323334353637
1383838383940414243
1444444444546474849
1505050505152535455
```

```
1565656565758596061
1626262626364656667
1686868686970717273
1747474747576777879
1808080808182838485
1868686868788899091
1929292929394959697
"WATERSHED INDEX"
1, 1, 1, 1, 1, 1, 1, 1, 1, 1
1, 1, 1, 1, 1, 1, 1, 1, 1, 1
1, 1, 1, 1, 1, 1, 1, 1, 1, 1
1, 1, 1, 1, 1, 1, 1, 1, 1, 1
1, 1, 1, 1, 1, 1, 1, 1, 1, 1
1, 1, 1, 1, 1, 1, 1, 1, 1, 1
1, 1, 1, 1, 1, 1, 1, 1, 1, 1
1, 1, 1, 1, 1, 1, 1, 1, 1, 1
1, 1, 1, 1, 1, 1, 1, 1, 1, 1
1, 1, 1, 1, 1, 1, 1, 1, 1, 1
1, 1, 1, 1, 1, 1, 1, 1, 1, 1
1, 1, 1, 1, 1, 1, 1, 1, 1, 1
1, 1, 1, 1, 1, 1, 1, 1, 1, 1
1, 1, 1, 1, 1, 1, 1, 1, 1, 1
1, 1, 1, 1, 1, 1, 1, 1, 1, 1
1, 1, 1, 1, 1, 1, 1, 1, 1, 1
"CROP SEASON AND SHARE"
 1 PASTURE                        90. 270. 0.4100
 2 STORED FORAGE                 150. 240. 0.1300
 3 GRAINS                        150. 240. 0.2100
 4 GRN LEAFY VEGETABLES          150. 240. 0.0020
 5 OTHER FOOD CROPS              150. 240. 0.0040
 6 LEGUMES AND SEEDS             150. 240. 0.1500
 7 ROOTS AND TUBERS              150. 240. 0.0030
"WATERSHED DEFINITION"
 1 Sr-89        5.00E-06        0.0
 2 Sr-90        5.00E-06        0.0
 3 Cs-134       5.00E-06        0.0
 4 Cs-137       5.00E-06        0.0
"REGIONAL ECONOMIC"
 1 EXCLUSION    0.284 0.015 225.0 1594.0 67075.0
 2 REGION_02    0.284 0.015 225.0 1594.0 67075.0
 3 REGION_03    0.284 0.015 225.0 1594.0 67075.0
 ↕      ↕       ↕     ↕     ↕     ↕      ↕
 95 REGION_95   0.617 0.000 550.0 1948.0 62374.0
 96 REGION_96   0.622 0.000 556.0 1980.0 62954.0
 97 REGION_97   0.689 0.003 643.0 2433.0 72112.0
```

Save

Selecting *Save* from the *MACCS2-Input File* sub-menu saves the MACCS2 Site Data to the user-specified file. It is recommended that the user save all result files in MACCS2 format with the filename extension inp for compatibility with MACCS2.

3.6.4 Comma-Separated-Variable (CSV) File

Selecting *CSV File* from the *Result* menu displays the *CSV File* sub-menu. This menu allows the user to display, print, or save the results in CSV file format.

Display

Selecting *Display* from the *CSV File* sub-menu displays a *CSV File* form like the one shown in Figure 3-20. The user can scroll through the data using the scroll bar. The command *Close* closes the form. The CSV File is normally displayed as ASCII text. All text strings are enclosed in double quotes and all variables (including text strings) are separated by commas. This format is compatible with the import data feature of most spreadsheet programs.

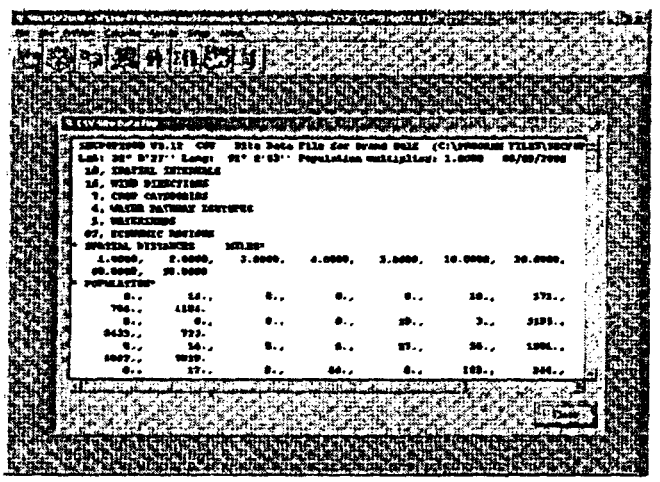


Figure 3-20 CSV Site-Data File Form

Print

Selecting *Print* from the *CSV Input File* sub-menu prints the CSV Site-Data File to the user-specified printer. The CSV Input File is printed as ASCII text.

Table 3.2 shows an example CSV Site-Data File. Note: SECPOP2000 does not perform any calculations for estimating crop categories, water pathway isotopes, or watersheds. The values below for those categories (watershed index, crop season and share, and watershed definition) are default values. For a full explanation of all of the data fields, see Appendix B.

Table 3.2 Example CSV Site-Data File

```
SECPOP2000 V3.12 CSV Site Data File for Grand Gulf (C:\PROGRAMFILES\SECPOP\SITES\GRAND GULF.SIT)
Lat: 320 0'27" Long: 910 2'53" Population multiplier: 1.0000  06/09/2003
 10, SPATIAL INTERVALS
 16, WIND DIRECTIONS
  7, CROP CATEGORIES
  4, WATER PATHWAY ISOTOPES
  1, WATERSHEDS
 97, ECONOMIC REGIONS
"SPATIAL DISTANCES MILES"
 1.0000, 2.0000, 3.0000, 4.0000, 5.0000, 10.0000, 20.0000, 30.0000
 40.0000, 50.0000
"POPULATION"
 0., 14., 0., 0., 10., 171., 578.
```

```
 786., 1184.
 0., 0., 0., 0., 29., 3., 5195., 32449.
 3433., 723.
 0., 14., 0., 0., 27., 56., 1801., 4859.
 4897. , 3019.
 0., 17., 0., 64., 0., 123., 244., 3253.
 4626., 66151.
 0., 0., 0., 113., 56., 102., 567., 1531.
 4018., 14053.
 0., 10., 0., 0., 336., 840., 981., 277.
 5726., 8482.
 0., 0., 0., 10., 776., 2582., 1177., 637.
 2720., 21634.
 0.1 14., 5., 37., 122., 571., 1189., 579.
 3261., 2536.
 0., 0., 0., 0., 0., 34., 2186., 4740.
 1970., 1330.
 0.1 0., 4., 0., 22., 1357., 1191., 1804.
 25607., 4413.
 0., 0., 0., 0., 0., 74., 51., 1223.
 16912., 3665.
 0., 0., 0., 0., 0., 56., 1671., 342.
 1863., 2095.
 0., 0., 0., 0., 0., 123., 346., 321.
 4267., 2694.
 0., 0., 0., 0., 0., 6., 2386., 97.
 4527., 11244.
 0., 0., 0., 0., 0., 28., 130., 62.
 2364., 5989.
 0., 0., 0., 0., 0., 0., 95., 11204.
 667., 2621.
"LAND FRACTION"
0.97, 0.97, 0.00, 0.00, 0.00, 0.96, 0.96, 0.96, 0.95, 0.95
0.00, 0.00, 0.00, 0.00, 0.97, 0.95, 0.95, 0.95, 0.94, 0.96
0.00, 0.97, 0.00, 0.97, 0.97, 0.96, 0.95, 0.96, 0.97, 0.98
0.00, 0.97, 0.00, 0.97, 0.00, 0.97, 0.97, 0.99, 0.99, 0.99
0.97, 0.00, 0.97, 0.97, 0.97, 0.97, 0.97, 1.00, 0.99, 0.99
0.97, 0.97, 0.00, 0.97, 0.97, 0.97, 0.97, 0.99, 1.00, 1.00
0.00, 0.00, 0.00, 0.97, 0.97, 0.97, 0.97, 0.99, 1.00, 1.00
0.97, 0.97, 0.97, 0.97, 0.97, 0.97, 0.98, 0.99, 1.00, 1.00
0.97, 0.00, 0.00, 0.00, 0.97, 0.97, 0.99, 0.99, 0.99, 0.99
0.97, 0.97, 0.97, 0.00, 0.97, 0.97, 0.98, 0.98, 0.95, 0.96
0.00, 0.00, 0.00, 0.97, 0.00, 0.97, 0.98, 0.96, 0.94, 0.93
0.00, 0.00, 0.00, 0.00, 0.94, 0.97, 0.94, 0.94, 0.95, 0.95
0.00, 0.00, 0.00, 0.94, 0.94, 0.94, 0.94, 0.95, 0.98, 0.98
0.00, 0.00, 0.00, 0.94, 0.94, 0.95, 0.94, 0.94, 0.98, 0.98
0.00, 0.00, 0.00, 0.95, 0.94, 0.94, 0.94, 0.96, 0.97, 0.99
0.00, 0.97, 0.00, 0.00, 0.96, 0.94, 0.95, 0.96, 0.96, 0.96
"REGION INDEX"
1, 2, 2, 2, 2, 3, 4, 5, 6, 7
1, 8, 8, 8, 8, 9,10,11,12,13
1,14,14,14,14,15,16,17,18,19
1,20,20,20,20,21,22,23,24,25
1,26,26,26,26,27,28,29,30,31
1,32,32,32,32,33,34,35,36,37
1,38,38,38,38,39,40,41,42,43
1,44,44,44,44,45,46,47,48,49
1,50,50,50,50,51,52,53,54,55
1,56,56,56,56,57,58,59,60,61
1,62,62,62,62,63,64,65,66,67
1,68,68,68,68,69,70,71,72,73
1,74,74,74,74,75,76,77,78,79
1,80,80,80,80,81,82,83,84,85
1,86,86,86,86,87,88,89,90,91
1,92,92,92,92,93,94,95,96,97
```

```
"WATERSHED INDEX"
1, 1, 1, 1, 1, 1, 1, 1, 1, 1
1, 1, 1, 1, 1, 1, 1, 1, 1, 1
1, 1, 1, 1, 1, 1, 1, 1, 1, 1
1, 1, 1, 1, 1, 1, 1, 1, 1, 1
1, 1, 1, 1, 1, 1, 1, 1, 1, 1
1, 1, 1, 1, 1, 1, 1, 1, 1, 1
1, 1, 1, 1, 1, 1, 1, 1, 1, 1
1, 1, 1, 1, 1, 1, 1, 1, 1, 1
1, 1, 1, 1, 1, 1, 1, 1, 1, 1
1, 1, 1, 1, 1, 1, 1, 1, 1, 1
1, 1, 1, 1, 1, 1, 1, 1, 1, 1
1, 1, 1, 1, 1, 1, 1, 1, 1, 1
1, 1, 1, 1, 1, 1, 1, 1, 1, 1
1, 1, 1, 1, 1, 1, 1, 1, 1, 1
"CROP SEASON AND SHARE"
1," PASTURE              ",  90., 270. 0.4100
2," STORED FORAGE        ", 150., 240. 0.1300
3," GRAINS               ", 150., 240. 0.2100
4," GRN LEAFY VEGETABLES", 150., 240. 0.0020
5," OTHER FOOD CROPS     ", 150., 240. 0.0040
6," LEGUMES AND SEEDS    ", 150., 240. 0.1500
7," ROOTS AND TUBERS 150., 240. 0.0030
"WATERSHED DEFINITION -- INITIAL AND ANNUAL WASHOFF AND INGESTION FACTORS"
 1," Sr-89", 5.00E-06, 0.0
 2," Sr-90", 5.00E-06, 0.0
 3," Cs-134", 5.00E-06, 0.0
 4," Cs-137", 5.00E-06, 0.0
"REGIONAL ECONOMIC DATA"
 1," EXCLUSION", 0.284, 0.015, 225.0, 1594.0, 67075.0
 2," REGION_02", 0.284, 0.015, 225.0, 1594.0, 67075.0
 3," REGION_03", 0.430, 0.006, 388.1, 1871.0, 74950.7
 4," REGION_04", 0.504, 0.000, 475.3, 2018.8, 76632.1
 5," REGION_05", 0.614, 0.000, 550.8, 1948.6, 62580.5
 6," REGION_06", 0.451, 0.000, 453.6, 2147.8, 86509.9
 7," REGION_07", 0.560, 0.000, 652.7, 2235.8, 76212.8
 8," REGION_08", 0.284, 0.015, 225.0, 1594.0, 67075.0
 9," REGION_09", 0.320, 0.002, 343.0, 2076.0, 95985.3
10," REGION_10", 0.355, 0.000, 376.8, 2111.9, 95410.7
11," REGION_11", 0.327, 0.000, 358.6, 2129.3, 98900.0
12," REGION_12", 0.351, 0.000, 440.6, 2085.9, 92563.7
13," REGION_13", 0.474, 0.002, 539.5, 2078.1, 85726.4
14., REGION_14", 0.284, 0.015, 225.0, 1594.0, 67075.0
15," REGION_15", 0.295, 0.006, 292.9, 1910.6, 87101.9
16," REGION_16", 0.301, 0.003, 323.5, 2053.2, 96125.4
17," REGION_17", 0.331, 0.003, 346.9, 2244.9, 102415.1
18," REGION_18", 0.365, 0.006, 362.2, 2411.3, 106228.3
19," REGION_19", 0.433, 0.009, 393.8, 2474.0, 104461.2
20," REGION_20", 0.284, 0.015, 225.0, 1594.0, 67075.0
21," REGION_21", 0.284, 0.015, 225.0, 1594.0, 67075.0
22," REGION_22", 0.284, 0.015, 225.0, 1594.0, 67075.0
23," REGION_23", 0.403, 0.011, 364.5, 2540.4, 106056.0
24," REGION_24", 0.415, 0.010, 378.0, 2632.0, 109827.1
25," REGION_25", 0.412, 0.011, 386.6, 2628.8, 109611.7
26," REGION_26", 0.284, 0.015, 225.0, 1594.0, 67075.0
27," REGION_27", 0.284, 0.015, 225.0, 1594.0, 67075.0
28," REGION_28", 0.284, 0.016, 227.8, 1596.5, 67107.2
29," REGION_29", 0.291, 0.068, 502.2, 2043.2, 79582.2
30," REGION_30", 0.302, 0.063, 491.1, 2095.9, 82286.1
31," REGION_31", 0.293, 0.057, 601.8, 2180.8, 83168.3
32," REGION_32", 0.284, 0.015, 225.0, 1594.0, 67075.0
33," REGION_33", 0.284, 0.015, 225.0, 1594.0, 67075.0
34," REGION_34", 0.284, 0.015, 225.0, 1594.0, 67075.0
35," REGION_35", 0.258, 0.078, 508.4, 1842.2, 70284.0
```

```
36," REGION_36", 0.255, 0.088, 537.5, 1872.1, 70715.4
37," REGION_37", 0.257, 0.177, 500.6, 1948.5, 72803.5
38," REGION_38,,, 0.284, 0.015, 225.0, 1594.0, 67075.0
39," REGION_39", 0.284, 0.015, 225.0, 1594.0, 67075.0
40," REGION_40", 0.270, 0.013, 218.0, 1558.5, 65118.1
41," REGION_41", 0.237, 0.051, 345.4, 1641.3, 64171.6
42," REGION_42", 0.237, 0.358, 354.9, 2060.0, 77175.3
43," REGION_43", 0.260, 0.461, 373.9, 2188.3, 79551.8
44," REGION_44", 0.284, 0.015, 225.0, 1594.0, 67075.0
45," REGION_45", 0.284, 0.015, 225.0, 1594.0, 67075.0
46," REGION_46", 0.209, 0.002, 186.9, 1402.0, 56495.6
47," REGION_47", 0.194, 0.017, 178.9, 1407.3, 56619.0
48," REGION_48", 0.132, 0.230, 138.9, 1794.7, 75782.9
49," REGION_49", 0.152, 0.303, 164.1, 1892.3, 75028.8
50," REGION_50", 0.284, 0.015, 225.0, 1594.0, 67075.0
51," REGION_51", 0.284, 0.015, 225.0, 1594.0, 67075.0
52," REGION_52", 0.199, 0.000, 182.0, 1377.0, 55120.0
53," REGION_53", 0.180, 0.064, 169.9, 1494.2, 60915.0
54," REGION_54", 0.134, 0.231, 140.3, 1814.7, 77232.1
55," REGION_55", 0.181, 0.316, 172.4, 1899.0, 72877.0
56," REGION_56", 0.284, 0.015, 225.0, 1594.0, 67075.0
57," REGION_57", 0.284, 0.015, 225.0, 1594.0, 67075.0
58," REGION_58", 0.201, 0.000, 182.8, 1380.8, 55328.5
59," REGION_59", 0.216, 0.000, 193.2, 1489.7, 63693.7
60," REGION_60", 0.271, 0.000, 229.9, 1859.9, 91848.2
61," REGION_61", 0.257, 0.048, 210.5, 1799.0, 85961.8
62," REGION_62", 0.284, 0.015, 225.0, 1594.0, 67075.0
63," REGION_63", 0.286, 0.015, 226.7, 1596.4, 67145.7
64," REGION_64", 0.265, 0.002, 238.6, 1489.5, 59657.0
65," REGION_65", 0.360, 0.001, 325.6, 1738.4, 73777.3
66," REGION_66", 0.434, 0.000, 405.5, 1962.4, 87058.0
67," REGION_67", 0.495, 0.000, 472.3, 2001.3, 85230.3
68," REGION_68", 0.638, 0.004, 581.0, 2085.0, 81610.0
69," REGION_69", 0.328, 0.014, 268.9, 1654.6, 68867.9
70," REGION_70", 0.638, 0.004, 581.0, 2085.0, 81610.0
71," REGION_71", 0.638, 0.004, 581.0, 2085.0, 81610.0
72," REGION_72", 0.575, 0.001, 500.1, 2009.1, 72798.2
73," REGION_73", 0.550, 0.000, 473.0, 1986.1, 71260.3
74," REGION_74", 0.638, 0.004, 581.0, 2085.0, 81610.0
75," REGION_75", 0.638, 0.004, 581.0, 2085.0, 81610.0
76," REGION_76", 0.638, 0.004, 581.0, 2085.0, 81610.0
77," REGION_77", 0.647, 0.003, 591.9, 2075.2, 76109.2
78," REGION_78", 0.656, 0.003, 603.1, 2052.2, 66095.1
79," REGION_79", 0.649, 0.002, 594.1, 2047.5, 66256.1
80," REGION_80", 0.638, 0.004, 581.0, 2085.0, 81610.0
81," REGION_81", 0.519, 0.007, 461.8, 1920.6, 76743.8
82," REGION_82", 0.638, 0.004, 581.0, 2085.0, 81610.0
83," REGION_83", 0.637, 0.003, 579.6, 2075.4, 80023.4
84," REGION_84", 0.657, 0.002, 604.6, 2044.1, 65514.3
85," REGION_85", 0.673, 0.004, 601.3, 2190.2, 72103.9
86," REGION_86", 0.562, 0.006, 505.0, 1980.2, 78508.7
87," REGION_87", 0.628, 0.004, 571.5, 2071.9, 81221.9
88," REGION_88", 0.627, 0.003, 571.6, 2071.5, 80012.3
89," REGION_89", 0.618, 0.000, 550.9, 1951.8, 62905.7
90," REGION_90", 0.637, 0.001, 572.3, 2014.3, 64862.3
91," REGION_91", 0.680, 0.007, 584.5, 2336.8, 79063.0
92," REGION_92", 0.458, 0.010, 399.5, 1834.7, 74200.2
93," REGION_93", 0.576, 0.003, 536.9, 2095.9, 85365.6
94," REGION_94", 0.496, 0.001, 473.8, 2055.8, 82308.5
95," REGION_95", 0.617, 0.000, 550.0, 1948.0, 62374.0
96," REGION_96", 0.623, 0.000, 557.0, 1984.9, 63053.0
97," REGION_97", 0.688, 0.003, 644.8, 2430.2, 71621.7
```

Save

Selecting *Save* from the *CSV Input File* sub-menu saves the CSV Site-Data File to the user-specified file. It is recommended that the user save all result files in CSV format with the filename extension csv. The user is not required to follow this practice. Doing so, however, makes it easier to find previously saved results files. This is especially true when a single directory has an assortment of site, problem, and result files.

3.6.5 Load Previous Results

Selecting *Load Previous Results* from the *Results* menu allows the user to select previously calculated results that have been saved to a file. Once a file has been chosen, the results are loaded into memory and can be examined using any of the methods described above.

If results are saved in MACCS2 format, the radial distances are saved in kilometers. (If results are saved in CSV format, the radial distances are saved in the units the user specified on the Problem Form). The radial distances can be changed back to miles by opening the *Problem* form and clicking on the "Miles" option button.

3.7 Performing Regional Calculations

Regional calculations are not supported in SECPOP2000. This functionality may be included in a future version. Please see NUREG/CR-6525, SAND93-4032, "SECPOP90: Sector Population, Land Fraction, and Economic Estimation Program" for more details on regional calculations.

3.8 Customizing SECPOP2000

Selecting *Setup* from the main menu displays the Setup form shown in Figure 3-21. This form allows the user to set default locations for the various SECPOP2000 files.

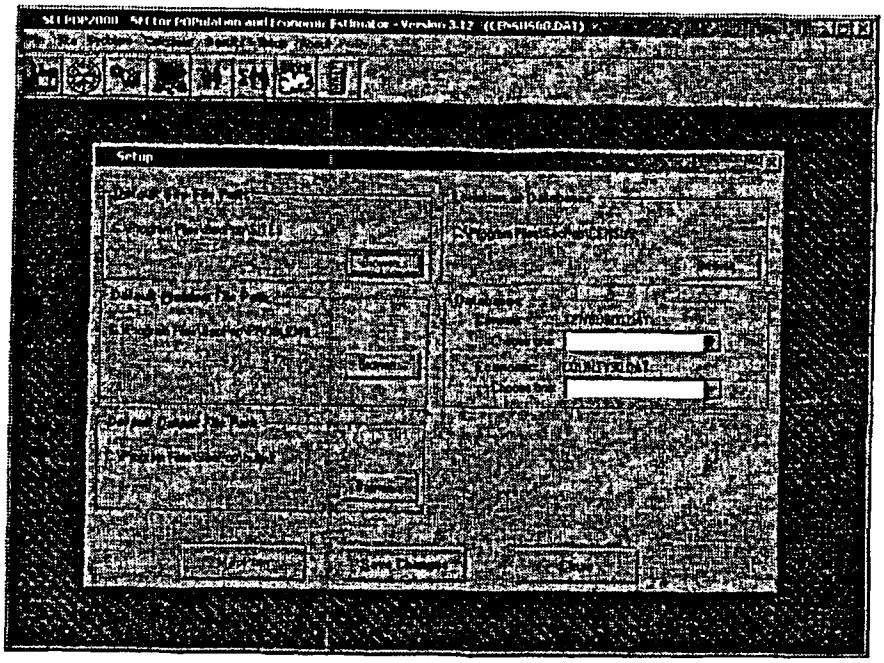


Figure 3-21 Setup Form

3.8.1 Setup Form Fields

The *Setup* form has six fields designed to allow the user to specify the default location of the input and output files. The user can override many of these values using the *File Open* and *Save As* forms discussed earlier, but these pathnames should be set to the paths that are used most often.

Default Site File Path	the default path where site files are stored.
Default Problem File Path	the default path where problem files are stored.
Default Output File Path	the default path where output files are stored.
Location of Census Databases	the default path where the block level population and county level fraction and economic census databases are stored.
Databases	Selects which census and economic files to use for the calculations. SECPOP2000 includes both the 1990 (CENSUS90.DAT) and 2000 (CENSUS00.DAT) census databases and the 1990 (COUNTY90.DAT) and 1997 (COUNTY97.DAT) economic databases are included with SECPOP2000.

3.8.2 Setup Form Commands

The *Setup* form has three command buttons: one for executing the MapPlan setup program, one for saving the user-defined defaults, and the other to close the *Setup* form.

MapPlan	is not supported at this time and is grayed out.
Save Changes	immediately saves changes to the default paths to a SECPOP2000 configuration file (secpop.cfg). This file is read the next time SECPOP2000 is started and the default values that were previously saved are restored.
Close	closes the *Setup* form. Unsaved changes are not saved to the SECPOP2000 configuration file but are used for the present SECPOP2000 session.

3.9 Exiting SECPOP2000

Selecting *File*, *Exit* from the main menu exits the SECPOP2000 program. The user is asked to confirm whether the SECPOP2000 program should be exited (see Figure 3-22). Exiting SECPOP2000 causes all unsaved information to be lost.

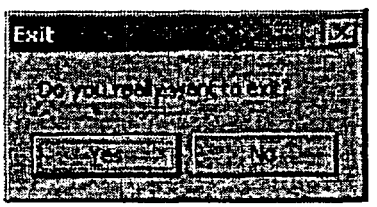


Figure 3-22 Exit Confirmation Form

4 Computational Methodology

The SECPOP2000 code uses block-level census data to calculate the population counts for each of the userdefined grid (rosette) sections, county-level census data to calculate land fraction data, and county-level census data plus other economic information published by the Federal Government to calculate economic data. The methodology involved can be broken down into (1) the algorithms used within the code to accomplish this purpose and (2) the database structures used for the block and county-level databases. Subsections 4.1 through 4.3 discuss the algorithms in detail. Subsection 4.4 discusses the database structures.

4.1 Population Count and Density Algorithms

For a user-defined site location and radial grid, SECPOP2000 establishes longitude-latitude boundaries that just encompass the grid. The census data file is a binary file sorted primarily by descending longitude (west to east) and secondarily by descending latitude (north to south). An iterative algorithm is used (within the POINTR subroutine) to find the westernmost point in the census data file that lies on or to the east of the western longitude boundary. For that data point and each subsequent data point read from the census data file, SECPOP2000 determines whether the point lies between the north and south latitudinal boundaries. When a point is found to lie between the established boundaries, the distance of that point from the site is calculated using the distance formula to determine if in fact the point lies within the outer limit of the grid. If the point meets the distance criteria, it is then processed to determine the exact grid element in which it lies based on its radial distance and direction from the site. The population associated with that data point is then added to the population in the appropriate element of the array TOTPOP. The TOTPOP array contains an array element for each grid section.

The algorithm used to determine the longitude-latitude boundaries of the grid is discussed in Subsection 4. 1. 1. The iterative algorithm used to determine the westernmost point in the census data file lying on or to the east of the western longitude boundary is discussed in Subsection 4.1.2. Finally, the algorithms used to determine if a data point lies within the grid and to determine the specific grid element within which any point lies are discussed in Subsection 4.1.3.

Regional calculations performed in SECPOP90 use the same algorithms described below except that there is no need to determine within which rosette section the population lies. The population density is calculated directly using the population and area of each census block. This option is disabled in SECPOP2000.

4.1.1 Boundaries Algorithm

The heart of the algorithm lies within the subroutine GETDIS, which finds the distance (in km) per degree latitude and per degree longitude for a specific geodetic latitude. Within the GETDIS subroutine, the geodetic latitude is first converted to radians. The corresponding latitude in the master coordinate system, which is geocentric, is then calculated using the following equation:

$$\theta_{GC}=\tan^{-1}\left[\left(\frac{R_{ep}}{R_{eq}}\right)^2\tan\theta_{GD}\right] \quad 4\text{-}1$$

where

θ_{GC} = the geocentric latitude (radians)
R_{ep} = the polar radius of the earth, 6256.8 km
R_{eq} = the equatorial radius of the earth, 6378.1 km
θ_{GD} = the geodetic latitude (radians)

The geocentric radius, RGC, is then calculated using the equation:

$$R_{GC}=\frac{R_{eq}R_{ep}}{\sqrt{R_{eq}^2\sin^2\theta_{GC}+R_{ep}^2\cos^2\theta_{GC}}} \quad 4\text{-}2$$

The distance (km) per degree latitude, DPDLAT, and the distance (km) per degree longitude, DPDLON, are derived using the following equations:

$$DPDLAT = R_{GC}\frac{\sin\theta_{GC}}{\sin\theta_{GD}}\cdot\left(\frac{\pi}{180}\right) \quad 4\text{-}3$$

$$DPDLON = R_{GC}\cos\theta_{GC}\cdot\left(\frac{\pi}{180}\right) \quad 4\text{-}4$$

It can be shown that if the west-east longitudinal boundaries are established on a latitude that intersects the site, the entire rosette lies between those same two longitudinal boundaries. The distance to those longitudinal boundaries can be found by dividing the outer radius of the grid by DPDLON, the distance per degree longitude at the geodetic latitude of the site.

Finding the north and south latitudinal boundaries is more complex. The distance per degree latitude decreases with increasing latitude. To ensure that the boundaries encompass the entire grid, distance per degree latitude, DPDLAT, is determined at the Tropic of Cancer for the half of the grid lying below the latitude of the site, and the distance per degree latitude, DPDLAT, determined at the site is used for the half of the grid north of the site. As a result, the north and south boundaries are always outside the grid, but this is acceptable since the boundaries are used only to eliminate the census data points that will not undergo further processing.

4.1.2 First Element Location Algorithm

An iterative "divide and conquer" algorithm is used to determine the first element in the census data file that lies on or to the east of the western longitudinal boundary of the grid. The set of records in the census data file is divided into two halves and it is determined in which half the western boundary lies. The procedure is then repeated concentrating on that half of the set. This is continued until one of two situations occurs. If two adjacent points are found which straddle the boundary, the easternmost record is marked. If a point is found that lies on the western boundary, the records are

searched backward sequentially until the first of the data elements is found that lies on the boundary and that point is marked.

4.1.3 Specific Grid Element Determination

When a census data element is found to lie within the longitudinal and latitudinal boundaries of the grid, a SECPOP algorithm calculates the surface distance between the census data element's reference point and the site. If the distance is less than the outer radius of the rosette, then the census element lies within the region of interest. Surface distance is calculated using the two-dimensional distance formula based on longitudinal and latitudinal distances. Longitudinal distance is calculated by multiplying the difference in longitudes between the site and the element by the value of DPDLON calculated for the site. Similarly, latitudinal distance is calculated by multiplying the difference in latitudes by an average DPDLAT. Since values for DPDLAT vary significantly with latitude, the value used is the average of values calculated for the site and for the element location. The census data element is then located within a radial ring based on this surface distance.

Once it is determined that a data element lies within the grid, it is then necessary to determine in which of the 16 sectors the data element lies. The angle (measured clockwise from true north) of a line from the site to the data element is found as follows:

$$\theta_{de} = \tan^{-1}\left(\frac{x}{y}\right) \qquad 4\text{-}5$$

where

θ_{de} = the angle made between a line from the site to the data element and true north
x = the distance from the longitude of the data element to the longitude of the site
y = the distance from the latitude of the data element to the latitude of the site

This value, θ_{de}, is then used to determine the specific directional element in which the data element lies.

4.2 Land Fraction Algorithms

In addition to location and population, every record in the block-level database also includes the area of the block and a code to indicate which county in the U.S. the block resides. This additional information is used by both the land-fraction algorithm and the economic-factors algorithms to estimate land fractions and economic factors respectively.

County-level databases contain the land-fraction data for every county in the continental U.S. The sources for the 1990 and 2000 economic factors and the sources of the data are described in Table 4.1.

Table 4.1 County Level Data File Variables and Data Sources

Variable	Description	Source for COUNTY90.DAT	Source for COUNTY97.DAT
FRMFRC	Fraction of land devoted to farming in the region	1992 Census of Agriculture, Geographic Area Series I B County-Level Data U.S. Department of Commerce Economics and Statistics Administration Bureau of the Census.	1997 Census of Agriculture Data Files Supplied by the U.S. Department of Commerce Economics and Statistics Administration Bureau of the Census.
DPF	Fraction of farm sales resulting from dairy production in the region	1992 Census of Agriculture Geographic Area Series I B County-Level Data U.S. Department of Commerce Economics and Statistics administration Bureau of the Census.	1997 Census of Agriculture Data Files Supplied by the U.S. Department of Commerce Economics and Statistics Administration Bureau of the Census.
ASFP	Annual average farm sales for the region ($/hectare)	1992 Census of Agriculture, Geographic Area Series 1B, County-Level Data, U.S. Department of Commerce, Economics and Statistics Administration, Bureau of the Census	1997 Census of Agriculture Data Files Supplied by the U.S. Department of Commerce Economics and Statistics Administration Bureau of the Census.
VFRM	Average farmland value for the region ($/hectare)	1992 Census of Agriculture, Geographic Area Series IB, County-Level Data, U.S. Department of Commerce, Economics and Statistics Administration, Bureau of the Census.	1997 Census of Agriculture Data Files Supplied by the U.S. Department of Commerce Economics and Statistics Administration Bureau of the Census.
VNFRM	Average non-farm value for the region ($/person)	1993 and 1994 Statistical Abstract of the United States, U.S. Dept. of Commerce, Economics and Statistics Admin., Bureau of the Census, 1994 County and City Data Book, U.S. Dept. of Commerce, Bureau of the Census, Data User Services Div.	1998 and 1999 Statistical Abstract of the United States, U.S. Dept. of Commerce, Economics and Statistics Admin., Bureau of the Census, 1999 County and City Data Book, U.S. Dept. of Commerce, Bureau of the Census, Data User Services Div.

The area of the blocks cannot be used to determine section land fractions directly for two reasons. First, the area given is only the landmass - no area for the water mass is given at the block level. Second, there is no simple way to aggregate the block areas to determine how much of a rosette section they "fill up" since the geometry of the blocks is unknown. Instead the area of the blocks is used to weight the county-level land fraction data.

During a site-specific calculation, a running sum for each rosette section is made of the total area of the blocks that lie within each section and a running sum is made of all of the weighted land fraction data. At the end of the calculation, the sum of the weighted land fractions is divided by the sum of the block areas. This is equivalent to the following formula for the land fraction for rosette section i, j:

$$FRCLND(i,j) = \frac{\sum_{n} A_{Block} \times FRCLND(County_{Block})}{\sum_{n} A_{Block}} \qquad 4\text{-}6$$

where

FRCLND(i, j)	= the estimated land fraction for the rosette section defined by sector i and radii j and j-1 (unless j = 1, then the section is defined only by i and j - i.e., the innermost sections of the rosette),
FRCLND($County_{Block}$)	= the land fraction of the county in which the present census block resides,
A_{Block}	= the area of the present census block, and
n	= the number of census blocks that reside in section i, j of the rosette.

Note: in areas where census blocks tend to be large–lakes, deserts, national and state parks – it is possible that no census block centroid lies within a rosette section and the resulting land fraction is estimated to be 0. While this is appropriate in a lake, it does not represent desert regions accurately. The user may need to edit the output data manually in such cases to obtain a better estimate.

4.3 Economic Factors Algorithms

SECPOP2000 estimates the economic factors that are defined in the Site Data File. The economic factors are calculated for the user-defined (or default) economic regions. The region inside the innermost radius is normally considered to be the exclusion region. While SECPOP2000 calculates values for this region, they are likely to be considerably different from the real values for the exclusion area if that area mostly contains of the nuclear reactor site.

The algorithm used to calculate the economic factors is very similar to that used in Section 4.2. The only difference is that the values are accumulated for each economic region instead of each rosette section. The countylevel database has values for each economic factor. The Economic Factor (ET) for each economic region, i, is calculated using the following equation:

$$EF(i) = \frac{\sum_{n} A_{Block} \times EF(County_{Block})}{\sum_{n} A_{Block}} \qquad 4\text{-}7$$

where

EF(i)	= the estimated economic factor for the economic region i,
EF($County_{Block}$)	= the economic factor of the county in which the present census block resides,
A_{Block}	= the area of a census block, and
n	= the number of census blocks that belong to economic region i.

4.4 Database Structure

The two main databases used by SECPOP90 are the block-level census and county-level census databases, CENSUS90.DAT and COUNTY90.DAT. The block-level census database contains over six million records - a record for every census block in the continental U.S. at the time of the 1990 census. The county-level census database contains over three thousand records, one for every county in the continental United States.

Similarly, the two main databases used by SECPOP2000 are the block-level census and county-level census databases, CENSUSOO.DAT and COUNTY97.DAT. The block-level census database contains almost eight million records, one for each block in the continental United States at the time of the 2000 census. The county-level census database contains over three thousand records, one for every county in the continental United States. The sections below describe the databases in more detail.

4.4.1 Block-Level Database

The 1990 block-level census data is stored in a binary file, CENSUS90.DAT, that contains 6,660,337 records. Each record is 12 bytes long and contains the following five pieces of information:

(1) a 2-byte integer code for the longitude of the geometric centroid of the census block,
(2) a 2-byte integer code for the latitude of the geometric centroid of the census block,
(3) a 2-byte integer for the residential population that resides within the census block,
(4) a 4-byte integer for the land area (0.001 km) of the census block, and
(5) a 2-byte integer code for the county that the census block resides in.

The integer codes for the longitude and latitude were derived to reduce the storage required for the block-level database. The longitude can be calculated using the following formula:

$$\text{longitude} = (\text{integer\_code} + 91993) / 1000.0$$

The latitude can be calculated using the following formula:

$$\text{latitude} = (\text{integer\_code} + 16610) / 1000.0$$

The integer code for the county is the index to the county-level database and is the first field in each record of that database. More information on the county-level database can be found in the next sub-section. Details on how the block-level database was constructed and verified can be found in Section 5. 1.

The 2000 block-level census data is stored in a binary file of the same format, but that file contains 7,938,746 records.

4.4.2 County-Level Database

The 1990 and 1997 county-level census data is stored in a fixed-width format ASCII text file, COUNTY90.DAT, which contains 3,111 records and one header line, and COUNTY97.DAT, which also contains 3,111 records and one header line. Each record contains the following 10 pieces of information:

CountyCode	index to the county-level database,
State	two-letter abbreviation of the state in which the county lies,
CountyName	name of the county,
FRCLND	fraction of area that is land in the county,
FRMFRC	fraction of land devoted to farming in the county,
DPF	fraction of farm sales resulting from dairy production in the county,
ASFP	annual-average farm sales for the county ($/hectare),
VFRM	average farmland value for the county ($/hectare),
VNFRM	average non-farm value for the county ($/person), and
Notes	various notes concerning exemptions to the data.

A sample of this database is shown in Table 4.2. The 1997 database follows the same format.

Table 4.2 Excerpt From the 1990 County-Level Database

CountyCode	State	CountyName	FRCLND	FRMFRC	DPF	ASFP	VFRM	VNFRM	Notes
1	ME	Androscoggin County	0.945790	0.206791	0.125332	3216	4532	109414	0
2	ME	Aroostook County	0.976975	0.078229	0.027361	908	2062	93633	0
3	ME	Cumberland County	0.686591	0.100777	0.233237	730	6779	144689	0
↕	↕	↕	↕	↕	↕	↕	↕	↕	↕
3109	CA	Ventura County	0.835891	0.271368	0.001479	5147	17519	145082	1
3110	CA	Yolo County	0.989805	0.800826	0.002811	1115	5551	133615	1
3111	CA	Yuba County	0.979641	0.581876	0.029790	1142	6226	84089	0

5 Software Development

5.1 Block-Level Census Database

This section describes how the data for the block-level database were extracted and how the results were verified to ensure the integrity of the data after each step of the process.

The block-level data and some county-level data (area of land and water) for 1990 were extracted from the CDROM set of the *Census of Population and Housing, 1990: Public Law (P.L.) 94-171, Data Technical Documentation /prepared by the Bureau of the Census. --Washington: The Bureau, 1991*[sic].

The block-level data and some county-level data (area of land and water) for 2000 were extracted from the DVD *Census 2000 Summary File 1, Census Population and Housing/prepared by the U. S. Census Bureau, September 2001.*

Both data files, *CENSUS90.DAT* and *CENSUS00.DAT*, are random access binary files. Both work with SECPOP2000. They also work with SECPOP90, except that the 2000 census database is too large to be completely read by SECPOP90. The 2000 census data contains 7,938,746 blocks in the contiguous United States versus 6,660,335 blocks in the 1990 census data. The difference in number of blocks between the 1990 and 2000 databases reflects the population increase in the United States over the ten years separating the two census activities. This issue is discussed further in 5.1.4.3.

Some geographical adjustments were made in compiling the 2000 database to be as consistent as possible with the algorithms used in SECPOP90. One adjustment involves the addition of two new blocks in Clallam County in Washington State. This area contains the western-most land in the contiguous United States. The two new blocks have centroids west of the longitude boundary that is hard coded in SECPOP90. Since neither of these blocks contains a permanent population, they were omitted in compiling the *CENSUS00.DAT* file. In fact, the new data file ignores approximately 226,000 similar blocks that contain neither land area nor population.

Other changes between 1990 and 2000 affected the compilation of the county database. The city of Miami, Florida, merged with Dade County to form the Miami-Dade County. This new entity was given the SECPOP90 county code for the old Dade County. The county area of South Boston City, Virginia, no longer exists. Population and area were subsumed into other Virginia counties. Yellowstone National Park is no longer considered a county. Population and area were subsumed into other Wyoming counties.

5.1.1 Construction of the 1990 Block-Level Census Database

The PL 94-171 files are stored on ten CD-ROMs in dBase III format. The first step in the construction of the census-level database (*CENSUS90.DAT*, located in the SECPOP\CENSUS directory) was to extract the necessary information from the dBase files and store it in binary format.

There is a dBase file for each state and the District of Columbia. The name of these files is PL9417*nn*.DBF, where *nn* is the postal code for the state or district. The format of the dBase census records is specified in another dBase file, PL94STRU.DBF, which is also included on the CD-ROMs. Table 5.1 shows the format of the state files. Note that all fields are of type character string. All of the source code referenced in this subsection can be found in the SECPOP90 Manual (Reference 1, Appendix G).

Table 5.1 dBase III Census Record Format

Name	Type	Length	Decimal Count	Description
FILEID	C	8	0	File Identification
STUSAB*	C	2	0	State/US Abreviation
SUMLEV*	C	3	0	Summary Level
GEOCOMP	C	2	0	Geographic Component
CHARITER	C	3	0	Characteristic Iteration
LOGRECNU	C	6	0	Logical Record Number
LOGRECPN	C	4	0	Logical Record Part Number
PARTREC	C	4	0	Total Number of Parts in Record
ANRC	C	2	0	Alaska Native Regional Corporation
AIANACE	C	4	0	Indian/Alaska Native Area (Census)
AIANAFP	C	5	0	American Indian/Alaska Native Area (FIPS)
AIANACC	C	2	0	American Indian/Alaska Native Area Class Code
ARTLI	C	1	0	American Indian Reservation Trust Land Indicator Code
BLCK	C	4	0	Block
BLCKGR	C	1	0	Block Group
TRACTBNA	C	6	0	Census Tract/Block Numbering Area
CONGDIS	C	2	0	Congressional District (101 st Congress)
CONCITCE	C	1	0	Consolidated City (Census)
CONCITFP	C	5	0	Consolidated City (FIPS)

Name	Type	Length	Decimal Count	Description
CONCITCC	C	2	0	Consolidated City Class Code
CONCITSC	C	2	0	Consolidated City Population Size Code
CMSA	C	0	0	Consolidated Metropolitan Statistical Area
CNTY*	C	3	0	County
CNTYSC	C	2	0	County Population Size Code
COUSUBCE	C	3	0	County Subdivision (Census)
COUSUBFP	C	5	0	County Subdivision (FIPS)
COUSUBCC	C	2	0	County Subdivision Class Code
COUSUBSC	C	2	0	County Subdivision Population Size Code
DIVIS	C	1	0	Division
EXTCITIN	C	1	0	Extended City Indicator
INTUC	C	15	0	Internal Use Code
MSACMSA	C	4	0	Metropolitan Statistical Area/Consol. Metro. Statistical Area
MSACMSAS	C	2	0	MSA/CMSA Population Size Code
PLACECE	C	4	0	Place (Census)
PLACEFP	C	5	0	Place (FIPS)
PLACECC	C	2	0	Place Class Code
PLACEDC	C	1	0	Place Description Code
PLACESC	C	2	0	Place Population Size Code
PMSA	C	4	0	Primary Metropolitan Statistical Area
REG	C	1	0	Region
STATECE*	C	2	0	State (Census)
STATEFP	C	2	0	State (FIPS)
URBANRUR	C	1	0	Urban/Rural

Name	Type	Length	Decimal Count	Description
URBAREA	C	4	0	Urbanized Area
UASC	C	2	0	Urbanized Area Population Size Code
SAC1	C	5	0	Special Area Code (1)
SAC2	C	5	0	Special Area Code (2)
SAC3	C	4	0	Special Area Code (3) Voting District Code
SAC4	C	4	0	Special Area Code (4)
SAC5	C	3	0	Special Area Code (5)
SAC6	C	3	0	Special Area Code (6)
SAC7	C	2	0	Special Area Code (7)
SAC8	C	2	0	Special Area Code (8)
SAC9	C	1	0	Special Area Code (9) Land/Water Area Code
SAC10	C	1	0	Special Area Code (10) Actual/Pseudo Voting District Code
AREALAND*	N	10	0	Area (land)
AREAWAT*	N	10	0	Area (water)
ANPSADPI*	C	66	0	Area Name/PSAD Term/Part Indicator
FUNCSTAT	C	1	0	Functional Status Code
GCUNI	C	1	0	Geographic Change User Note Indicator
HUIOO	N	9	0	Housing Unit Count (100%)
INTPTLAT*	C	9	0	Internal Point (latitude)
INTPTLNG*	C	10	0	Internal Point (longitude)
PARTFLAG	C	1	0	Part Flag
PSADC	C	2	0	Political/Statistical Area Description Code

Name	Type	Length	Decimal Count	Description
POP100*	N	9	0	Population Count (100%)
SPFLAG	C	1	0	Special Flag
IP001_0001	N	9	0	Population Table 1, Item 1
IP002_0001	N	9	0	Population Table 2, Item 1
IP002_0002	N	9	0	Population Table 2, Item 2
IP002_0003	N	9	0	Population Table 2, Item 3
IP002_0004	N	9	0	Population Table 2, Item 4
IP002_0005	N	9	0	Population Table 2, Item 5
IP003_0001	N	9	0	Population Table 3, Item 1
IP003_0002	N	9	0	Population Table 3, Item 2
IP003_0003	N	9	0	Population Table 3, Item 3
IP003_0004	N	9	0	Population Table 3, Item 4
IP003_0005	N	9	0	Population Table 3, Item 5
IP004_0001	N	9	0	Population Table 4, Item 1
IP004_0002	N	9	0	Population Table 4, Item 2
IP004_0003	N	9	0	Population Table 4, Item 3
IP004_0004	N	9	0	Population Table 4, Item 4
IP004_0005	N	9	0	Population Table 4, Item 5
IP004_0006	N	9	0	Population Table 4, Item 6
IP005_0001	N	9	0	Population Table 5, Item 1

*Indicates field used in block or county-level census database.

To extract the necessary information from the dBase state files, a highly modified version of the QuickBasic program included in the CD-ROMs' README file (see Figure 5- 1) was written, CONVERT1.BAS. This Basic program extracted both the block and county-level information and stored it in two separate files, BLOCK_1.BIN and COUNTY_1.TXT, respectively. The data stored in BLOCK_1.BIN was converted from the dBase text format to Visual Basic's native binary representation. Each record in BLOCK_1.BIN was 19 bytes long and contained the

longitude and latitude of the census block geographical centroid (4 bytes each), population (4 bytes), area (4 bytes), state census number (1 byte), and county census number (2 bytes).

```
'                              ------------------------
'                              dBASE To ASCII Conversion
'                              ------------------------
'Users without database management software who wish to convert the files to
'ASCII can follow this example. This is Microsoft QuickBASIC 4.5 code, Tests
'on a 286 PC took about 1 1/2 minutes per thousand records reading from CD-ROM
'and writing to hard disk.

TYPE Structure
     Flag AS STRING * 1
     Record AS STRING * 516
END TYPE

DIM PL94171 AS Structure

OPEN "e:\pl9417vt.dbf" FOR BINARY AS #1
OPEN "c:\pl9417vt.txt" FOR OUTPUT AS #2

SEEK #1, 2947

DO
     GET #1, , PL94171
     IF EOF(1) THEN EXIT DO
     PRINT #2, PL94171.Record
LOOP

CLOSE #1, #2
END
'     --------------------
```

Figure 5-1 PL94-71 README File dBase to ASCII Conversion Program

The second part of the block-level conversion process was to create a second binary file, BLOCK_2.BIN that contains the same information as BLOCK_1.BIN but in an encoded format so that the data records are smaller (BLOCK_1.BIN was 126 Mbytes, BLOCK_2.BIN was 80 Mbytes) and easier to use. The encoding for the latitude and longitude is discussed in Section 4.4.1. The encoding of the county code was performed using a lookup table in the CONVERT2.BAS program. This was necessary because the county code used in the census uses designators that are unique within each state but not throughout the country. The code generated for the county designator of each block is also the county index to the county-level database. The codes were ordered by the census state-level designator as the primary key and the census county designator as the secondary key. Furthermore, since each county had a unique designator in BLOCK_2.BIN, the state designator was not stored in BLOCK_2.BIN. That information can be obtained by performing a table lookup in the county-level database. SECPOP90 has no need for that information at present.

The next part of the creation of the block-level census database was to sort the records by longitude and latitude. This was done in a two-step process using two FORTRAN programs, SORTBIN1.FOR and SORTBIN2.FOR. SORTBIN1.FOR first sorted the BLOCK_2.BIN file by separating it into 30 separate files based on a census block's longitude. SORTBIN2.FOR then individually sorted the 30

files by longitude and latitude, writing the results of each sorted file to BLOCK_3.B1N. The SORTBIN2.FOR program contains an error. It incorrectly sorts files that contain records with both positive and negative longitudes. (Negative and positive longitudes were present in the intermediate BLOCK_*n*.BIN files because of the encoding of the longitude from 32 to 16 bit integers.) Fortunately, this case only existed in one of the 30, BLOCK_2.BIN files. The database was corrected during the Verification testing (see Section 5.1.2). The SORTBIN2.FOR program was not corrected and still contains the error.

The final step was the concatenation of duplicate records by the FORTRAN program RMOVEDUP.FOR. Two records were considered duplicates if they shared the same longitude, latitude, and county code. When such records were found, their areas and population counts were added together and a single record was created to represent the duplicate records. The final resulting database was written to BLOCK_4.BIN, which was then renamed to *CENSUS90.DAT*.

5.1.2 Construction of the 2000 Block-Level Census Database

The first step in the construction of the block-level database (*CENSUSOO.DAT*, located in the SECPOP\CENSUS directory) was to extract the necessary information from the DVD and store it in binary format. The DVD comes with a data engine. Figure 5-2 shows the user interface for the engine. The window on the left illustrates the selection of data at the block level. Note the exclusion of Alaska and Hawaii. The right window contains the data output for each block.

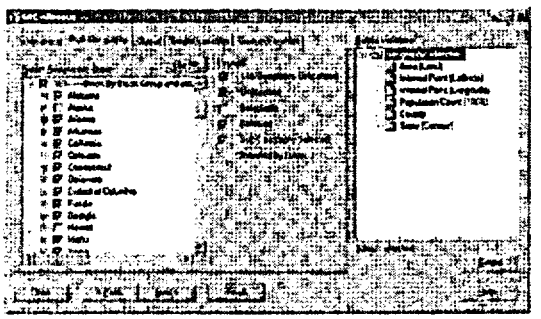


Figure 5-2 Commercial Data Engine User Interface

Figure 5-3 illustrates the output for a single block from a county in Alabama. Note the list of other blocks for that county on the left.

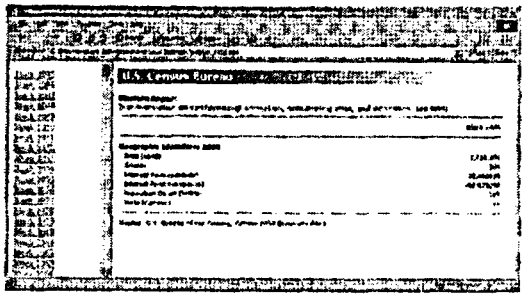


Figure 5-3 Output for a Single Block

In addition to the visual output shown in Figure 5-3, the data engine produced a comma separated variable (CSV) file DVD_OUT.CSV that contains data for all contiguous states. Appendix G contains the Visual Basic for Applications program used in an Excel spreadsheet to convert this CSV file into the SECPOP2000 file, *CENSUSOO.DAT*. The program first reads block level information from the CSV file. It converts it to a form compatible with SECPOP2000 and discards blocks that contain neither land area nor people. There were 225,972 of these, similar to the finding for SECPOP90. The program also finds and outputs blocks that are beyond the longitudinal limits of the contiguous states. There were two of these, neither of which had population nor land area. Viable blocks (7,938,746 in all) were output to an unsorted database UNSORT_BLOCKOO.DAT readable by SECPOP2000. The program used an indexed recursive Quicksort routine to create the final database.

5.1.3 Verification of the 1990 Block-Level Census Database

Upon completion of each of the steps described in 5. 1. 1, diagnostic output was produced to verify that the conversion programs were operating correctly, and additional testing was performed to verify that the block-level census data integrity was maintained. (No effort was made to determine the integrity of the initial data on the PL 94-171 CD ROMs. For a discussion of the Census Bureau's level-of-confidence in their data, see www.census.gov/main/www/cen2000.html.) All of the source code referenced in this section can be found in the SECPOP90 Manual (Reference 1, Appendix H).

After the first conversion of the dBase CD-ROM data, the TEST1.BAS basic program was executed to read in all of the census block records and print out a summary of all of the county and state populations and land areas. The summary results agreed with the state and county populations and were always less than or equal to the state and county land areas. (Aggregated block-level census land areas are usually less than county or state land areas because of the differences in tabulating the two area types. Comparisons were done using a spreadsheet. A sample of Rhode Island census block longitudes and latitudes from the CD-ROM database files was compared with those in BLOCK_1.BIN and were in agreement.

After the second conversion of the BLOCK_1.BIN data and the creation of the BLOCK_2.BIN database, the TEST2.BAS program was executed to print out a sample of 71 records from both the BLOCK_1.BIN and BLOCK_2.BIN databases. The results were examined by hand and were found

to be in agreement. The file lengths of the two databases were also examined to ensure that they contained the same number of records.

After the sorting of the BLOCK_2.BIN database was completed and the BLOCK_3.BIN file was generated, the TEST3.BAS program was executed to print out the records for Washington County, Maine. These were compared with the same records found in the CD-ROM database files and were in agreement. The file lengths of the two databases, BLOCK_2.BIN and BLOCK_3.BIN, were compared to ensure that they contained the same number of records. The file lengths were equal.

A repeat of testing done after the first conversion was performed on the final BLOCK_4.BIN database using a modified version of TEST1.BAS named TEST4.BAS. The summary results agreed with the original data files from the PL 94-171 CD-ROMs. The Rhode Island data also agreed and the number of records remaining in BLOCK_4.BIN was equal to the initial number of records minus the duplicate records. BLOCK_4.BIN's name was changed to *CENSUS90.DAT*, which did not alter the data in any way. A final test to ensure that the database records were sorted correctly, TEST5.BAS, was performed. It was discovered that a section of records were out of order. Two programs, FIX1.BAS and FIX2.BAS were run to further diagnose and correct the database. TEST5.BAS was rerun. The error was corrected and there were no other records found out of order.

5.1.4 Verification of the 2000 Block-Level Census Database

Testing on the new database file includes the following four tests against the US Census Bureau's Census 2000 data on DVD (Reference 2):

Test 1: Sequential test to ensure blocks are sorted west to east
Test 2: Comparison of state area and population against US Census Bureau Summaries
Test 3: Comparison of county area and population against Summaries
Test 4: Numerous comparisons using SECPOP90 and CENSUSOO.dat to compare to US Census Bureau estimates at known locations

Verification tests 1 through 3 were run as Visual Basic code behind the spreadsheet as the front end to facilitate comparisons. Verification test 4 was intended to show that the 2000 census data file works with SECPOP90, and that this combination can reproduce population counts based on U.S. Census Bureau software. These tests used output from SECPOP90 and results displayed in HTML format from the U.S. Census Bureau DVD. The source code used for the verification tests are listed in Appendix G.

Table 5.2 summarizes the results of the verification tests run on the CENSUS00.DAT file. The verification tests ensure the file contains the correct information in the proper format and demonstrates that the new data file works correctly with SECPOP90.

Table 5.2 *CENSUS00.DAT* Verification Summary

Verification	Description	Results
Test 1	Sequential test to ensure blocks are sorted west to east	Blocks are sorted sequentially
Test 2	Comparison of state area and population against USBC Summaries	State populations match exactly and total contiguous state areas match to within 2 square miles (5.3 km^3).
Test 3	Comparison of county area and population against USBC Summaries	County populations match exactly, and total of contiguous county areas match to within 2 square miles (5.3 km^3).
Test 4	Match of SECPOP90 runs using CENSUS00.DAT against U.S. Census Bureau results	Maximum error 1.42%

5.1.4.1 Verification Test 1

The object of this test is to sequentially read blocks of data to ensure that the blocks are correctly sorted in west-to-east order. Output included the number of data blocks read, and information regarding blocks (if any) out of sequence.

Results:

- The program read block data sequentially first to last from the data file.
- Test reports zero blocks out of sequence.

The spreadsheet *2000 CensusDB V&V.xls* provides a detailed listing of the program code used to generate Table 5.3, as well as detailed summaries of the results. This file is contained on the SECPOP2000 CD.

5.1.4.2 Verification Tests 2 and 3

The object is to verify that the CENSUS00.DAT data file contains correct data for all census blocks by comparing state and county population and area totals extracted from the CENSUS00. DA T file with summaries provided by the U.S. Census Bureau.

The code in *2000 CensusDB V&V.xls* for subroutine "Test - 4" incorporates both Tests 2 and 3. It sequentially extracts census-block information from the data file. Based on the county code contained in each block of data, the test code allocates both population and area to the appropriate state. In parallel, the same area and population data is added to the appropriate county totals. U.S.

Census Bureau summaries came directly from the database software distributed with the Census 2000 DVD.

Table 5.3 shows a summary of the results for Test 2, the state-by-state comparison. State populations agreed exactly. An acceptably small error was observed in area estimates, probably caused by rounding. This assertion is confirmed by computational experiments, which indicate that the error drops with increased numerical precision. The 2000 census reports areas to the nearest square meter. SECPOP2000 represents areas to the nearest 0.000386 mile^2 (0.001 km^2).

Table 5.3 Test 2 Results: State-by-State Comparison

Source	Population	Area (mile^2)
Census00.dat	279,583,437	2,959,062.52
Census 2000 DVD	279,583,437	2,959,064.56
Difference for total	0	-2.05
Maximum by state	0	0.09
Minimum by state	0	-0.22
Average for states	0	-0.04
Std. Deviation for states	0	0.06

Table 5.4 provides a summary of the results for the county-by-county data from Test 3. As in Test 2, populations agreed exactly for all counties. Also as with Test 2, errors exist in the county area totals due to rounding. The total difference in the integrated area agrees exactly with the value from Test 2. Statistics on differences are smaller reflecting the finer level of detail in the county-by-county comparison.

Table 5.4 Test 3 Results: County-by-County Comparison

Source	Population	Area (mile^2)
Census00.DAT	279,583,437	2,959,062.52
Census 2000 DVD	279,593,437	2,959,064.56
Difference for total	0	-2.05
Maximum by state	0	0.05
Minimum by state	0	-0.07
Average for states	0	0.00
Std Deviation for states	0	0.04

5.1.4.3 Verification Test 4

The object of this test is to use the 2000 census data file with SECPOP90 to show that the combination can reproduce population counts from the U.S. Census Bureau software. The test compares population estimates within a ten-mile radius of selected sites. The sites were chosen almost at random, with preference given to those sites that were expected to have the highest populations. Due to the file size issues discussed below, preference is given to sites in the eastern United States, as this was the more difficult task.

Combining the *CENSUS00.DAT* files with SECPOP90 required an increase in the number of census blocks, which makes the *CENSUS00.DAT* file larger than its predecessor. The original SECPOP90 contains hard-coded limits on file size. While the original program reads the new data file, it only reads the first 6,660,335 of its 7,938,746 records. Data files are sorted by longitude to read from west to east. Hence, the SECPOP90 program stops reading data from *CENSUS00.DAT* at approximately Wheeling, West Virginia.

In order to perform the verification, it was necessary to break *CENSUS00.DAT* into two files - the full file spanning the full continent, and a shorter file starting at approximately Albuquerque, New Mexico, and covering all points East from there. Figure 5-4 shows the directory structure needed to accommodate three different, but identically named, data files.

The complex directory structure is necessary because SECPOP90 is hard wired to read a single data file name (*CENSUS90.DAT*), but accepts user-specified directories. Note that the data file in directory ~\CENSUS_E\, covering all points east of Albuquerque, NM, has almost the same length as the original data file in ~\CENSUS\. The data file in ~CENSUS_W\ is larger, containing the information for the complete 2000 census.

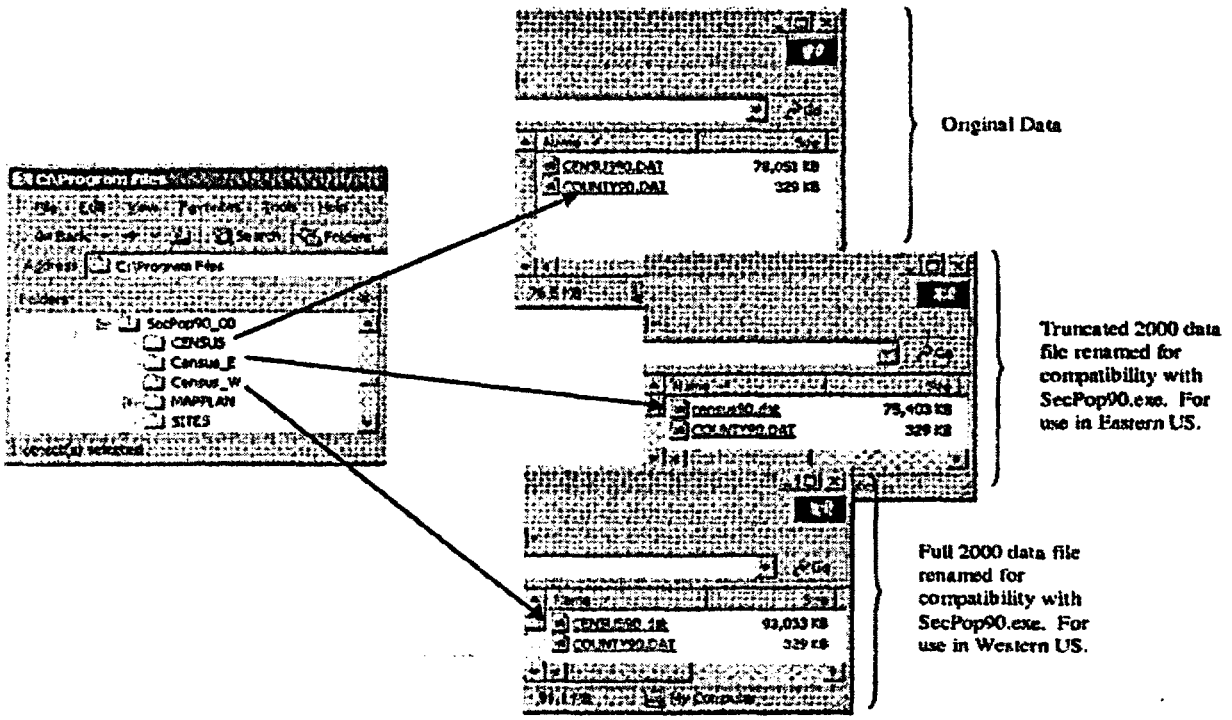


Figure 5-4 Verification Directory Structure

Table 5.5 summarizes the results. The table compares estimated population within a 10-mile radius for selected locations using SECPOP90 in combination with the census 2000 data with population estimates from the database software distributed with the U.S. Census Bureau DVD. With the exception of two of the 23 sites, agreement was within 0.6%. The Three Mile Island site had a 1.16% error, and Turkey Point had an error of 1.42%. All error calculations assume that the Census DVD provides the correct numbers.

The software on the DVD reports decimal fractions of people, which are rounded to integers in the above table. This indicates that it prorates blocks that are only partially within the 10-mile radius. SECPOP90, on the other hand, either completely includes or excludes blocks based on the location of each block's internal reference point. (The U.S. Census Bureau provides the latitude and longitude of an internal point for each census block. This point is usually the geographic centroid.) This difference in algorithms explains the small errors in the SECPOP90 results.

Table 5.5 Ten Mile Comparison with Census Bureau Using SECPOP90 Program and CENSUS00.DAT file

Site Name	SECPOP90/00	Census 2000	Percent Error	Absolute Error
Calvert Cliffs	38,918	38,868	0.13%	50
Catawba	141,378	141,231	0.10%	147
Dresden	64,513	64,508	0.01%	5
Duane Arnold	112,772	113,209	-0.39%	437
Grand Gulf	7,635	7,630	0.07%	5
Hope Creek/Salem	34,130	34,153	-0.07%	23
Kewanee	9,829	9,829	0.00%	0
LaSalle	13,678	13,678	0.00%	0
Limerick	212,218	212,423	-0.10%	205
McGuire	2,333	2,337	-0.17%	4
Millstone	116,807	117,051	-0.21%	244
Oconee	71,185	71,178	0.01%	7
Peach Bottom	41,296	41,134	0.39%	162
Sequoyah	83,011	83,057	-0.06%	46
South Texas	2,777	2,777	0.00%	0
St. Lucie	160,911	160,247	0.41%	664
Surry	117,233	117,918	-0.58%	685
Three Mile Island	189,430	187,256	1.16%	2174
Turkey Point	106,716	105,218	1.42%	1498
Waterford	80,755	80,885	-0.16%	130
Zion	262,098	261,113	0.38%	985

5.2 County-Level Census Database

The county-level census database was constructed with data from several sources. The PL 94-171 CD-ROM data files (see 5.1.1); *Census of Agriculture, 1992: Final County Files* [machine-readable datafile] conducted by the Bureau of the Census, Washington: The Bureau [producer and distributor], 1993 [sic]; *County and City Data Book 1994* on CD-ROM [machine-readable datafiles] prepared by the Bureau of the Census Washington: The Bureau [producer and distributor], 1995 [sic]; and 1993, 1994 *Statistical Abstract of the United States*, U.S. Department of Commerce, Economics and Statistics Administration, Bureau of the Census.

5.2.1 Construction of the 1990 County-Level Census Database

Land and water areas were extracted from the PL 94-171 CD-ROM data files (refer to Section 5.1.1 for details). The area data was imported into a spreadsheet and the county land fraction, FRCLND, was calculated by dividing the land area by the sum of the land and water areas. The county population data was also extracted for verification purposes but it was not used in the county-level database. There were no exceptions, omissions, or special treatment for any of the data items extracted from the PL 94-171 CD-ROM data files.

The agricultural data was extracted from the 1992 Census of Agriculture CD-ROM Geographic Area Series 113 using a program that was included on the CD-ROM called EXTRACT. The EXTRACT program allowed the data items described in Table 5.6 to be extracted for each state and saved to a file that was then imported into a spreadsheet.

It is important to note that for many of the counties one or more of the above data items were not available due to the confidentiality requirements of the census. In most cases estimates were made for the undisclosed amount by distributing the excess acreage or dollars evenly among the undisclosed farms. The excess acreage or dollars were determined by subtracting the sum of the disclosed amounts from the state totals provided. This method was chosen because the number of farms and dairy farms in a county was always provided, allowing the estimates to be made in a straightforward manner. Other more complicated methods such as apportioning the excess by acreage or estimating county values by surrounding county values did not provide any more reasonable or trusted estimates. In a small number of cases (particularly when undisclosed values were for small farms in heavily populated areas, e.g.; Queens, New York, or San Francisco, California), it was clear that average dollars per farm or acre were excessive. The District of Columbia has no agricultural products.

Table 5.6 Census of Agriculture Data Items

Item	Description
010001	Farms (number)
010002	Land in farms (acres)
010004	Estimated market value of land& buildings. Average/farm ($)
010006	Estimated market value of all machinery. Average/farm ($)
010019	Market value of agriculture products sold ($1,000)
020064	Dairy products, (farms)
020065	Dairy products, ($1,000)

From the Census of Agriculture Data, four agricultural values (these are the agricultural economic values defined for MACCS2) were calculated for each county. They are as follows:

FRMFRC	fraction of land devoted to farming in the county
DPF	fraction of farm sales resulting from dairy production in the county
ASFP	annual average farm sales for the county ($ / hectare)
VFRM	average farmland value for the county ($ / hectare). This includes the average value of farmland, buildings, and machinery

The fifth economic value calculated for each county is the average of non-farm value ($ / person), VNFRM (also a MACCS2 economic value). The methodology used to calculate this value for each county is equivalent to that described by Sprung et al. (Reference 2) First a value of VNFRM is calculated for the entire United States. Then estimates of county values are made using the following equation:

$$VNFRM_{County} = VNFRM_{US}\left(\frac{PCI_{County}}{PCI_{US}}\right) \qquad 5\text{-}1$$

where
$VNFRM_{County}$ = average non-farm value for the county ($ / person),
$VNFRM_{us}$ = average non-farm value for the United States ($ / person),
PCI_{County} = per capita income for the county ($ / person), and
PCI_{US} = per capita income for the United States ($ / person).

The values for the per capita income were extracted from the 1994 County and City Data Book CD-ROM database files. They were imported directly into Access and then modified to contain only the per capita information and the associated footnotes that indicated exceptions and omissions to the data. From Access, the data was then imported into a spreadsheet where it was incorporated into the rest of the county-level database. The exceptions to the data consisted of incidents where two or more counties or independent cities (county-like entities that have no other county affiliation) reported per capita income figures jointly. In all such cases, equal per capita income amounts were applied to all joined counties. $VNFRM_{us}$ is calculated using the equation:

$$VNFRM_{US} = \frac{RTW_{US} + VSL_{US} - VFA_{US} + VFHP_{US}}{POP_{US}} \qquad 5\text{-}2$$

where
RTW_{us} = reproducible tangible wealth for the United States ($),
VSL_{us} = value of suburban land in the United States ($),
VFA_{us} = value of farm assets in the United States ($),
$VFHP_{us}$ = value of farm household possessions in the United States($), and
POPUS = population of the United States (persons).

VSL_{us} is calculated using the equation:

$$VSL_{US} = UBL_{US} \cdot MHV_{US} \cdot LPA_{US} \cdot FLV_{US} \qquad 5\text{-}3$$

where
UBL_{us} = amount urban and built-up land in the United States (acres),
MHV_{us} = median housing value for the United States ($ / house),
LPA_{us} = average number of housing lots per acre (houses / acre), and
FLV_{us} = average fraction of house cost due to land value.

The values and the sources for the variables used in producing $VNFRM_{us}$ were all obtained from the 1993 and 1994 Statistical Abstract of the United States, except for LPA_{us} and FLV_{us} which were obtained from NUREG/CR 4551 (see Reference 2) and are equal to 5.0 houses / acre and 0.2 respectively. For details on the remaining values see Appendix D and E.

5.2.2 Changes in the County-Level Database with the 1997 Economic Census

An economic census performed in 1997 was used to create a new county-level database, COUNTY97.DAT. While this new file uses the same algorithms as the original, COUNTY90.DAT, unavoidable differences arose with the input parameters. Table 5.7 summarizes the results of calculations for both 1990 and 1997. Notes referenced on the table discuss both sources of data for 1997 and how the resulting information differs with that used in 1990. As the table shows, differences are significant enough that direct comparison of COUNTY90 and COUNTY97 data is discouraged.

Table 5.7 Variation in COUNTY90.DAT and COUNTY97.DAT Parameters

Description	Variable	1990	1997	Units	Notes
Reproducible Tangible Wealth	RTW_{us}	2.5652E+13	2.4883E+13	$ of the Day	1
Urban and Built-up Land	UBL_{us}	85,964,103	98,251,700	Acres	2
Median Housing Value	MHV_{us}	79,100	98,815	$ of the Day	3
Total Farm Assets	VFA_{us}	1.0037E+12	1.0530E+12	$ of the Day	4
Farm Household Possessions	$VFHP_{us}$	4.6400E+10	-	$ of the Day	5
Population	POP_{us}	248,719,873	281,421,906	Persons	6
Typical Lot Size		0.2	0.2	Acres	7
Houses per Acre	LPA_{us}	5	5	Houses/acre	
Land - Fraction of Housing Value	FLV_{us}	0.2	0.2	Fraction	8
U.S Per Capita Income	PCI_{us}	18,696	25,412	$/Person	9
Value of Suburban Land	VSL_{us}	6.79976E+12	9.7087E+12	$ of the Day	
Average Non-Farm Value for the United States	$VNFRM_{us}$	126,626	187,462	$/Person	

Notes:

1. Value for RTWus came from the document "Fixed Reproducible Tangible Wealth in the United States Revised Estimates for 1995-97 and Summary Estimates for 1925-97," The Survey of Current Business, September 1998. See Appendix E, Table E.1 for an example of this information.

   The 1997 county-level database uses Net Fixed Reproducible Tangible Wealth instead of Gross Fixed Reproducible Tangible Wealth as was used the 1990 version. Gross values ignore the depreciation of aging assets. As of 1996, the United States no longer estimates Gross Fixed Reproducible Tangible Wealth. Before that year both the net and the gross

values were published. In a telephone conference call with Leonard Loebach of the Bureau of Economic Analysis, Mr Loebach stated that creating gross values now would be extremely difficult, and probably misleading. In his opinion, using non-depreciated values for tangible wealth unjustifiably overestimates the consequences of losing these assets.

In 1996 the Bureau of Economic Analysis revised the way the U.S. calculates depreciation. For 1990 data, this effectively raised Net Fixed Reproducible Tangible Wealth 20% over what was previously published.

2. Values for UBL_{us} came from the Natural Resources Conservation Service's document, "Summary Report, 1997 National Resources Inventory, Revised December 2000," available at web site http://www.nrcs.usda.gov/tgchnical/NRI /1997/summary report. Figure 5-5 illustrates the trend, and its relation to previous data. See also Appendix E, Table E.2.

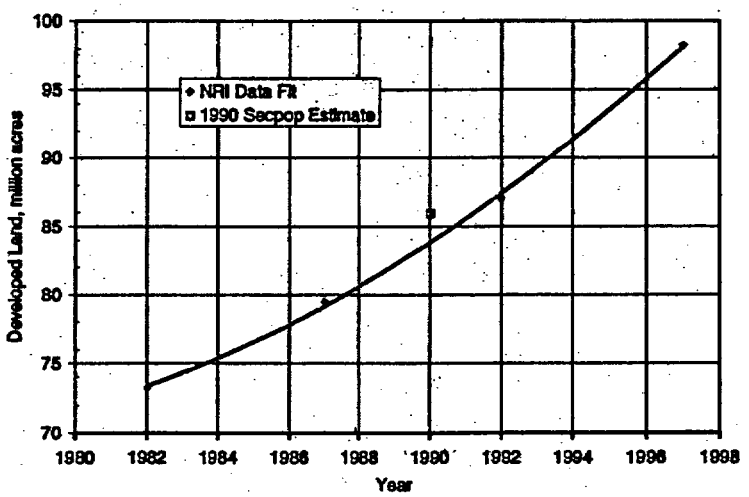


Figure 5-5 Developed Land Trends

3. Median housing costs shown in Figure 5-6 came from the 1991, 93, 95, 97 and 99 editions of the publication US Census Bureau, American Housing Survey Tables 3-14 (Appendix E, Table E.3).

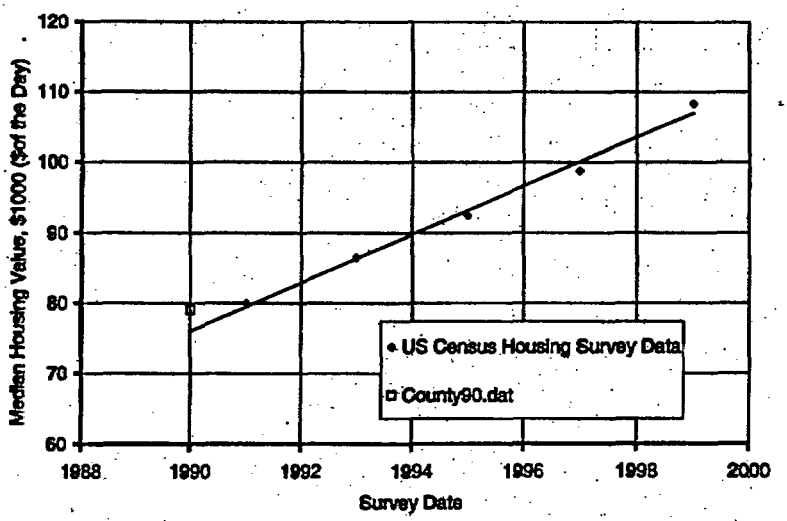


Figure 5-6 Median Housing Values, 1991 through 1999

4. Total U.S. farm assets come from the USDA/Economic Research Service. Ken Erickson stated in a telephone conversation that the current practice at the USDA is to exclude all "non-business" components from their balance sheets. This means the exclusion of household goods, personal financial assets, houses and real estate associated with houses. Figure 5-7 illustrates the impact of this current practice on the value used for VFA_{us}. The value of personal farm family possessions is no longer reported. See Appendix E, Table E.4.

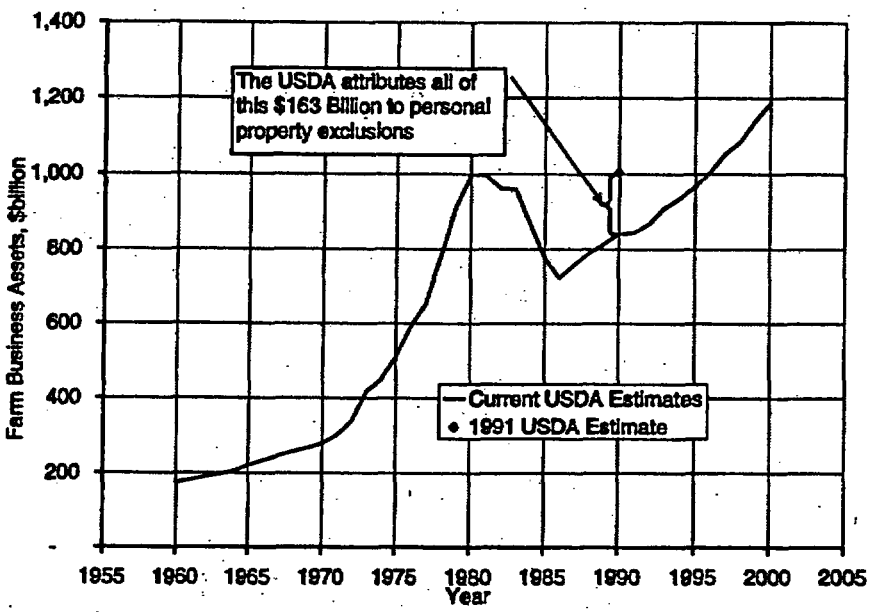


Figure 5-7 VFAus, United States Farm Business Assets Trend

5. Since the exclusions discussed above address the intent of variable $VFHP_{us}$ its value is set to zero in 1997. However, this introduces differences. As shown in Figure 5-7, personal property exclusions total $163 billion for 1990 data. The value of $VFHP_{us}$ used for the 1990 county level data was $46 billion, the value of household goods alone. The difference, if applied to 1997 data, means a change of less than 1% in the calculated value of $VNFRM_{us}$. Also, it can be argued that the earlier method double accounted for farm housing and other personal assets.

6. Population data was derived from the DVD. This DVD uses the Census 2000 Summary File 1 US Data, as discussed on the U.S. Census web site http://www.census.gov/Press-Release/www/2001/sumfilel.html.

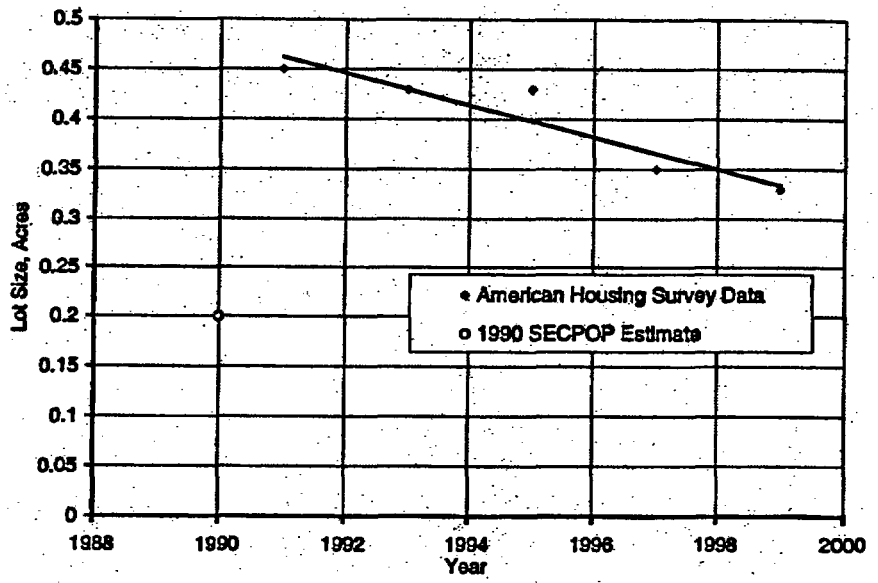


Figure 5-8 Median Lot Size for American Homes

7. The 1997 value in Table 5.7 assumes the same median lot size for U.S. homes as was used earlier (0.2 acre/house). From Figure 5-8, this value differs significantly from the information published in the American Housing Survey. This lower number was retained because it provides consistency with earlier work.

8. Unlike other items, land value as a fraction of housing value is not a statistic published by the United States. Thus, we adhered to the document referenced originally: NUREG/CR 4551, published in 1990.

9. Data on per capita income came from the U. S. Department of Commerce, Bureau of Economic Analysis CA1-3 tables, available as comma separated variable files at the web site http://www.bea.izov/bea/regionallreis/cal 3.html. These files were downloaded and imported into a spreadsheet for incorporation into the database.

5.2.3 Verification of the County-Level Census Database

Unlike the construction of the block-level database, the construction of the county-level database did not require the creation of any additional programs for conversion or sorting. The tools used were Excel , Access, and the EXTRACT program provided by the Bureau of Census. It is assumed that these programs functioned properly. However, after every operation and/or manipulation of the county-level data, checks were made to ensure that each county had the proper number of records and that all individual county amounts totaled (or other appropriate operation) to the state aggregate amounts.

5.3 Verification of the SECPOP90 Code

Verification of the SECPOP90 code concentrated on the accuracy of the results of the site-specific population calculations. There are several justifications for this approach. First, the regional calculations use the same algorithms as the site-specific calculations; therefore, the regional calculations are indirectly verified by the verification of the site-specific calculations. Second, there is no real way to verify the site-specific economic data that were derived from the county-level database. The economic values are estimates and should only be used for relative evaluations.

5.3.1 Internal Verification

The first step in the verification of the SECPOP90 code was to use the first 36 data points on the census data file to enable the code developer to readily verify that the population for each of those data points was correctly placed on a magnified grid close to the "site." The "site" was placed at each of the "corners" of the data as well at the "middle" of the data. SECPOP90 was found to work correctly in each case.

The SECPOP90 code was then exercised using the 1990 census data and the results contrasted with the results when the SECPOP code was exercised using the 1980 census data for five sites. The sites selected were distributed throughout the continental United States. It was found that the differences were, for the most part, well within differences that would have been predicted simply because of the population growth over the past 10 years.

Final internal verification efforts involved comparing the results obtained from the VAX/VMS system FORTRAN code version with the results obtained using the PC Visual Basic. The results were in good agreement.

5.3.2 External Verification for the SECPOP90 Code

A survey of U.S. commercial nuclear power plants identified those plants that have provided the NRC with updated population information based on the 1990 census data. Licensees for 12 plants have provided this information. Of these 12 plants, four only provided the total population within a specific distance from the plant. The others provided the information in tables, graphs, or both,

within 16 rings. The ring distances vary among plants. Licensee submittals may not be accurate; a deviation from the licensee's number(s) is not necessarily an error. In several plants, the licensee's population totals within a ring or sector (or both in one case) have been found to be incorrect. One licensee appears to have averaged the number of people within a ring, which is not accurate.

The following table shows the results for the total populations as reported by the licensees, as calculated by SECPOP90, and as percent of reported [((as calculated by SECPOP90) / (as reported by the licensees)) x 100%]. Superscripts in Table 5.8 correspond to reference numbers.

Table 5.8 Total Reported and Calculated Populations

Plant Name	Reported	SECPOP90	Percent of Reported
Fitzpatrick[4]	44,231	39,443	89%
Millstone 1[5]	3,363,745	2,780,995	83%
Monticello[6]	2,273,213	2,416,872	106%
*Nine Mile Point[7]	924,000	908,835	98%
*Oyster Creek[8]	96,718	101,433	105%
*Peach Bottom APS[9]	10,257,315	6,803,644	66%
DC Cook[10]	52,953	51,070	96%
Diablo Canyon[11]	374,701	376,012	100%
Fort Calhoun[12]	760,431	772,086	102%
Prairie Island[13]	25,342	24,969	99%
Turkey Point [14]	2,613,535	2,602,834	100%

* These plants only provided total population within a radius.

As shown above, on a total population basis, all of the results are in reasonable agreement. The program also provides the typical rosettes that show the populations in relation to the plant site. Examples of the rosettes for Millstone 1, Monticello, and Diablo Canyon are shown below. The results can also be provided as a MACCS site file for direct use in MACCS calculations. An example of this type of output is shown in Section 3.6.3.

Also included are population tables (see Appendix F) generated from licensee information and SECPOP90 output. These tables show the actual population by ring segment and the percent difference from the licensee's reported values. As with all percent results, caution should be used when interpreting the information. For example: at DC Cook, ring 1, south sector, the licensee

reported 5 people while SECPOP90 calculated 20 people, a 400% difference with no practical effects.

Generally, SECPOP90 calculates the total population in each ring and each sector reasonably well. However, the ring-segment population can have a large deviation from the licensee's values. Assuming that the licensee's values are accurate, this can be caused by the code's identification of the census block centroid. The entire population for a census block is counted in the rosette section in which the centroid of the block resides. This can be seen in the Diablo Canyon tables in the NW and NNW segments from 10 through 50 miles.

Another factor is the level of detail for the census data that the licensee used. If the level of detail was at the county, city, or block group level, the reported values are different. As previously mentioned, one licensee (Monticello) used average data for rings 2 through 6. The reasonableness of the results in both the ring and sector attests to the fact that the ring-segment-to-ring-segment differences are not significant. The one exception seems to be low population sites with scattered population, like Turkey Point. In these cases the percent differences are large. In such cases, engineering judgment is needed to assess the impact of the low population on the health effects of interest.

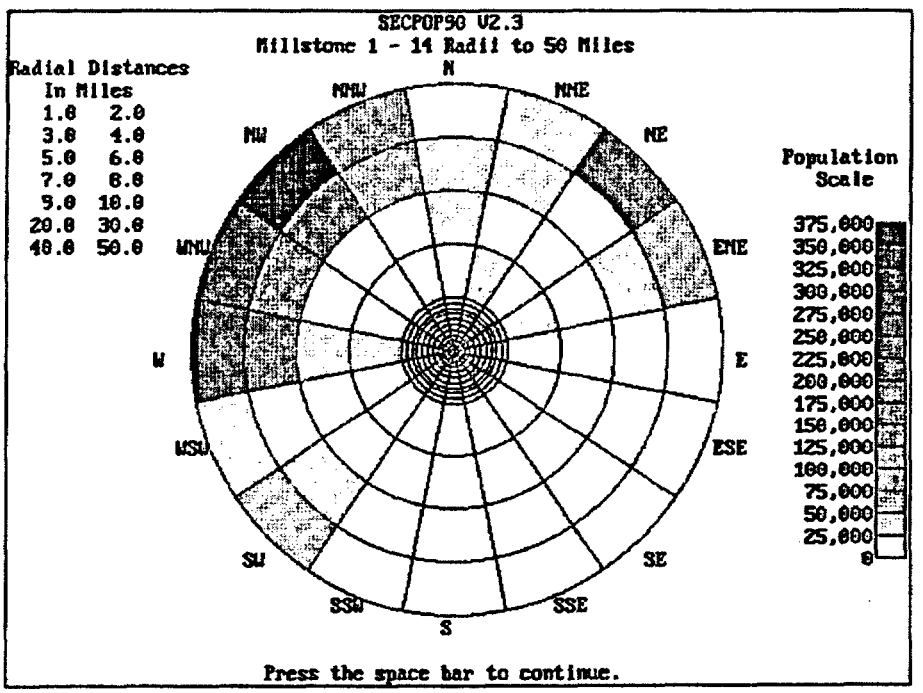


Figure 5-9 Millstone Rosette

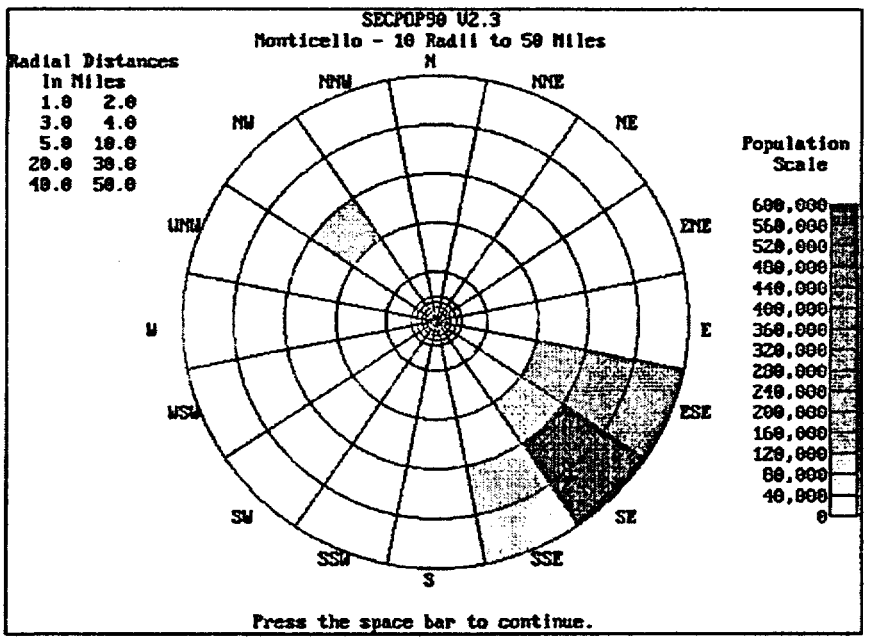


Figure 5-10 Monticello Rosette

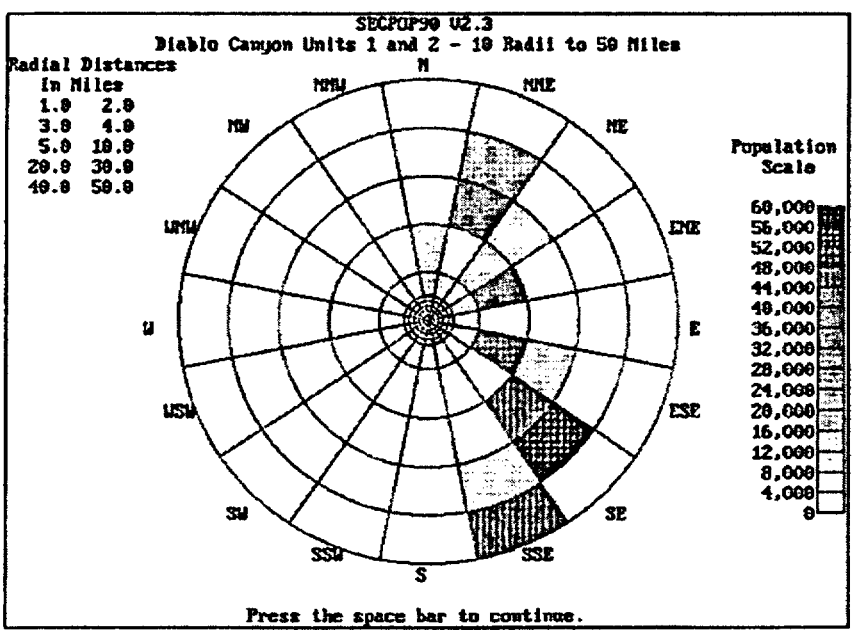


Figure 5-11 Diablo Canyon Rosette

5.4 Verification of the SECPOP2000 Code

Verification of the SECPOP2000 code concentrated on the accuracy of the results of the site-specific population calculations. There are several justifications for this approach. First, regional calculations are not supported in SECPOP2000. Second, there is no real way to verify the site-specific economic data that were derived from the county-level database. The economic values are estimates and should only be used for relative evaluations.

5.4.1 Internal Verification

The U.S. Census data engine can estimate population within a user-specified circle. This allows comparison with SECPOP2000 totals for the same circle. The data in Table 5.9 illustrate one such comparison using the San Onofre site. Percent differences are large, 8.9%, for the first circle containing population. Differences drop rapidly, tending to values less than 1%, for larger radii.

Table 5.9 San Onofre Circles Comparison Cumulative Population

Radius (miles)	US Census Database	SECPOP2000	Absolute Difference (%)
1	0	0	0.00%
2	0	0	0.00%
3	1,262	1,375	8.94%
4	6,267	6,393	2.01%
5	26,059	26,558	1.91%
6	41,638	41,773	0.32%
7	47,433	47,658	0.48%
8	54,847	55,290	0.81%
9	66,524	66,619	0.14%
10	75,588	75,796	0.27%
15	221,674	224,767	1.40%
20	700,026	700,633	0.09%
30	1,735,315	1,748,970	0.79%
40	4,064,777	4,097,536	0.81%
50	7,559,279	7,622,370	0.83%

These differences are inherent to the algorithms used by the two programs. SECPOP2000 allocates population by whole census blocks. The U. S. Census data engine, on the other hand, returns population estimates resolved to decimal fractions of a whole person, indicating that it somehow interpolates block populations. Blocks in the San Clemente area just to the north contain around 100 people, so at small radii inclusion or exclusion of one block can have a significant impact. For example, the almost 9% difference listed for the three mile radius circle results from a difference of only 113 people or about one census block. Subsequent discussion refers to this type of error as whole-block-precision error.

Figure 5-12 contains a plot of the Table 5.9 data. As expected from the above table, population estimates almost overlie each other. The figure also illustrates the downward trend of relative difference with increasing radius.

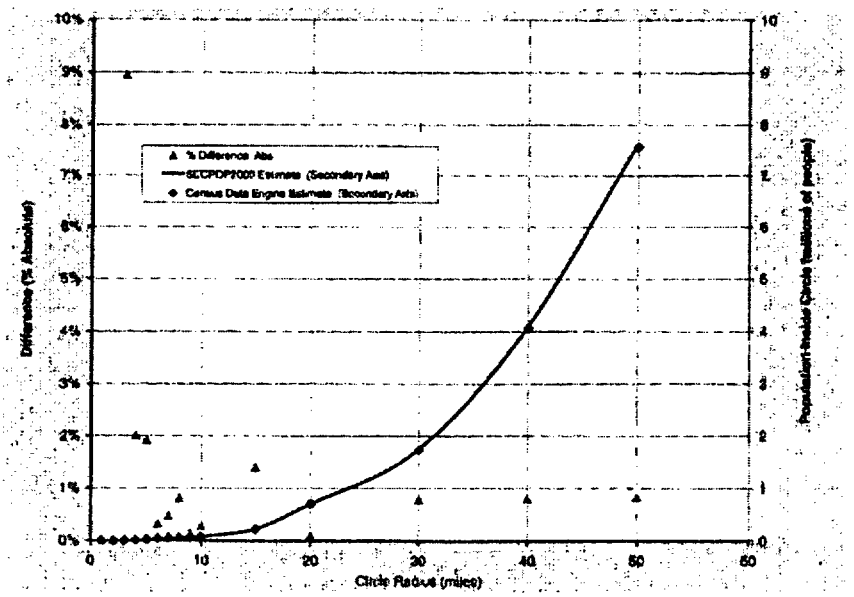


Figure 5-12 San Onofre Circles Comparison

Table 5. 10 summarizes results for a sector-by-sector comparison of the San Onofre site. The data engine supplied by the Bureau of Census can be extended, with some effort, to find populations within irregular polygons. This functionality, originally intended to support voting district restructuring, works with sectors as well. The process requires the creation of closed areas using Arc-Info or similar mapping software. These polygons can be translated into shape files interpretable by the data engine. Sectors were laid out based on radii of 5 and 50 miles using Arc-Info, and saved these shapes to a file. The Census Bureau data engine, using these shapes, generated the population estimates shown in the leftmost group of three columns in Table 5.10. These numbers compare well with the information generated by SECPOP2000 as listed in the center three columns. The last three columns show the percent difference between the two data sources. Variances were

generally within 4%, with most less than 2%. Significant errors arose in the WNW, NW, and NNW sectors, especially for small radii. In addition to the whole-block-precision error discussed above, evidence exists that errors exist with the polygon creation process, as described below.

Table 5.10 San Onofre Sector by Sector Comparison

	CENSUS/SNL			SECPOP2000			Error (abs)		
Radii(mi) / Sector	5	50	Wedge Sum	5	50	Wedge Sum	5	50	Wedge Sum
N	-	830,817	830,817	-	827,723	827,723		0.37%	0.37%
NNE	-	450,919	450,919	-	458,979	458,979		1.79%	1.79%
NE	-	252,939	252,939	-	247,910	247,910		1.99%	1.99%
ENE	-	110,752	110,752	-	111,977	111,977		1.11%	1.11%
E	11,471	50,189	61,660	11,471	50,441	61,912	0.00%	0.50%	0.41%
ESE	-	329,090	329,090	-	332,716	332,716		1.10%	1.10%
SE	-	784,194	784,194	-	779,497	779,497		0.60%	0.60%
SSE	-	500,553	500,553	-	485,503	485,503		3.01%	3.01%
S	-	-	-	-	-	-			
SSW	-	-	-	-	-	-			
SW	-	-	-	-	-	-			
WSW	-	-	-	-	-	-			
W	-	3,464	3,464	-	3,464	3,464		0.00%	0.00%
WNW	530	56,458	56,988	427	61,226	61,653	19.43%	8.45%	8.19%
NW	9,470	2,659,247	2,668,717	12,348	2,628,229	2,640,577	30.39%	1.17%	1.05%
NNW	1,520	1,548,781	1,550,301	2,312	1,608,147	16,104,559	52.11%	3.83%	3.88%
Radius Sum	22,991	7,577,403	7,600,394	26,558	7,595,812	7,622,370	15.51%	0.24%	0.29%

SECPOP2000 estimates the total population for all sixteen sectors within a 5-mile radius at 26,558 people. The table also shows the U.S. Census database estimates the people within the same 5-mile radius is 26,059, indicating a difference of less than 2%. This value is about 12% larger than the Table 5.10 radius sum of 22,991 people. Even a 12% error is acceptable for MACCS calculations, where the overall uncertainties in consequence predictions are significantly greater than 12%. The problem essentially disappears with larger radii. The SNL/Census total for a 50-mile radius is within 0.5% of the U.S. Census estimate for the same size circle.

Finally, Figure 5-17 provides a visual example of SECPOP2000's capability for the San Onofre site.

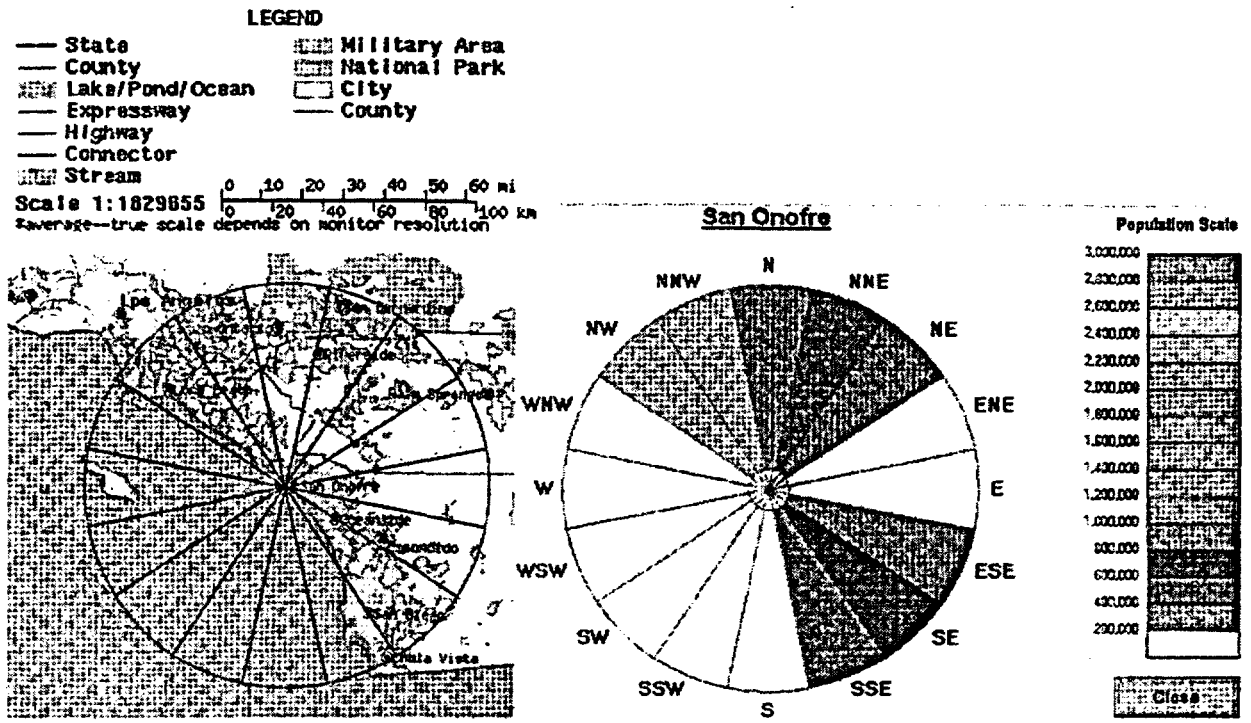


Figure 5-13 San Onofre Rosette Visualization Example

The figure shows a map on the left with a radial grid showing the sixteen sectors superimposed. A SECPOP2000 rosette appears on the right side of the figure. The rosette uses the same radii used in Table 5.10, 5 miles and 50 miles with a 1-mile dead zone. Both the rosette and the grid were sized to match the map. The rosette's population scale has increments of 200,000 people. As would be expected the NW sector, covering much of Long Beach and Los Angeles shows the highest population. Clockwise progression demonstrates a drop in population density through suburbs of Los Angeles to the San Bernardino Valley. The ENE and E sectors reflect the low (less than 200,000) populations of the desert areas outside of Palm Springs. Similar patterns emerge for the Oceanside-San Diego area. Catalina Island, which is directly west of San Onofre, exists entirely within the W sector. As Table 5.10 shows, all West sector estimates are identical at 3,464 people. Figure 5-18 contains a map/rosette graphic based on the Peach Bottom site in Eastern Pennsylvania. Radii here were chosen so that every sector contains approximately 50 square miles. The maximum radius is 50 miles. The rosette picks up the high populations in Baltimore to the SSW and SW directions. It shows higher populations to the east toward the Philadelphia suburbs and Northern Delaware. It also shows the distributed suburban and small town populations to the north.

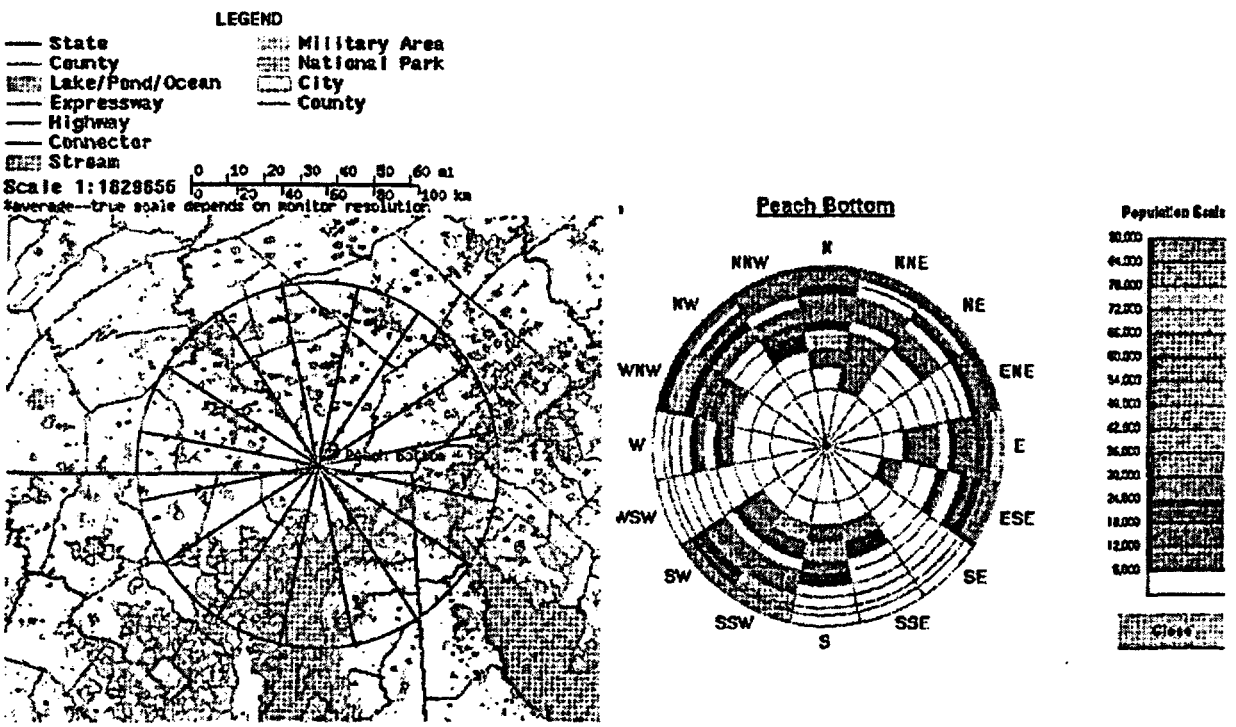


Figure 5-14 Peach Bottom Rosette Visualization Example

Figure 5-19 plots population estimates for increasing circles using SECPOP2000 and the Census data engine. As with San Onofre, the estimates match with percent differences consistently less than 1%. As before, differences trend downward with increasing radius.

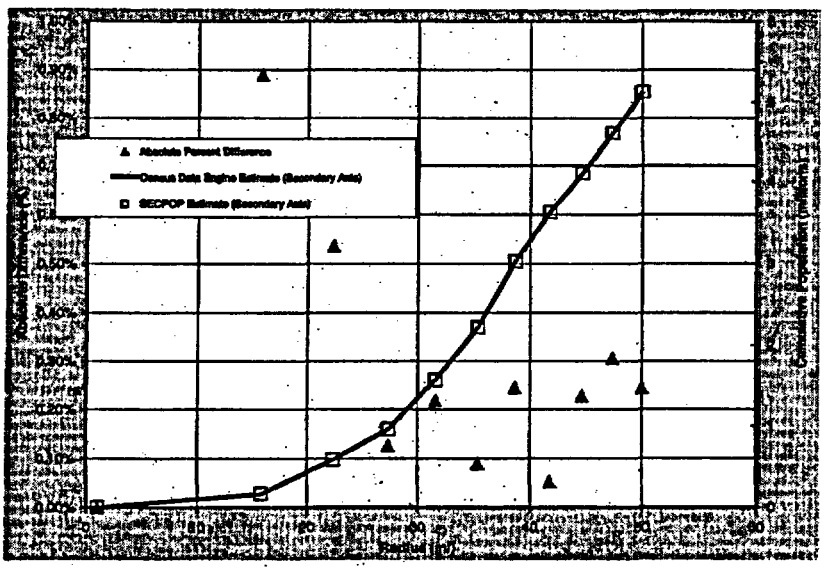


Figure 5-15 Peach Bottom Circles Comparison

5.4.2 External Verification of the SECPOP2000 Code

A survey of U.S. commercial nuclear power plants identified those plants that have provided the NRC with updated population information based on the 2000 census data. Licensees for two plants have provided this information. One of these plants only provided information out to 10 miles and there were few people within the 10 miles. An additional 20 plants estimated the population for the year 2000. Some of these estimates are very crude. Of these 20 plants, 10 provided reasonable data; that is, they did not use the same number of people in multiple adjacent sectors. The data was provided in tables, graphs or both, by rings of 16 segments. The ring distances vary among plants. Licensee submittals may not be accurate; a deviation from the licensee's number(s) is not necessarily an error. In several plants, the licensee's population totals within a ring or sector have been found to be incorrect. More than one licensee appears to have averaged the number of people within some rings.

The following table shows the results for the total populations as reported by the licensees, as calculated by SECPOP2000, and as percent of reported [((as calculated by SECPOP2000) / (as reported by the licensees)) x 100]. Superscripts in Table 5.11 correspond to reference numbers.

Table 5.11 Total Reported and Calculated Populations for a 60-Mile Radius

Plant Name	Reported	SECPOP2000	Percent of Reported	Census Data Used
Browns Ferry [15]	952,689	1,059,559	111%	2000
Columbia [16]	333,112	360,534	108%	1980
Cook [17]	4,207,374	4,252,575	101%	Not Specified
Grand Gulf [18]	321,319	330,449	102%	1970
Limerick [19]	7,109,077	7,603,283	106%	1970
Point Beach [20]	459,871	479,864	104%	1980
Robinson [21]	872,775	802,184	92%	1980
Seabrook [22]	4,426,080	4,167,639	94%	1980
South Texas [23]	352,092	255,118	73%	1980
Susquehanna [24]	1,647,043	1,686,422	102%	Not Specified
River Bend [25]	1,008,616	856,667	85%	1970

As shown above, on a total population basis, all of the results are in reasonable agreement. Only one plant, D.C. Cook, was reported in the SECPOP90 manual (September 1997) and is in this edition. The SECPOP90 manual reported the 10-mile population with a value of 96% (from Table 5.8 for

60 miles) of reported as compared with the 2000 census value of 101% (from Table 5.11). Generally, the percent of reported improved from SECPOP90 to SECPOP2000. SECPOP90 tended to underestimate the population while SECPOP2000 tends to slightly overestimate the population. SECPOP2000 also provides the typical rosettes, which show the populations graphically in relation to the plant site. Examples of the rosettes for Columbia Power Station, Limerick Generating Station, River Bend Station, and St. Lucie are shown below. The results can also be provided as a MACCS2 site file for direct use in MACCS2 calculations. An example of this type of output is shown in Section 3.6.3.

Also included are population tables (see Appendix F) generated from licensee information and SECPOP2000 output. These tables show the actual population by ring segment and the percent difference from the licensee's reported values. As with all percent results, caution should be used when interpreting the information. For example: at DC Cook, ring 1, east-northeast sector, the licensee reported 9 people while SECPOP2000 calculated 43 people, a 478% difference with little practical effect.

Generally, SECPOP2000 calculates the total population in each ring and each sector reasonably well. However, the ring-segment population can have a large deviation from the licensee's values. Assuming that the licensee's values are accurate, this can be caused by the fact that SECPOP2000 effectively places the entire block population at the block's centroid. An example of this is in Browns Ferry, ring 3, south and south-southwest sectors. The licensee reported 10 and 32 people (42 total), respectively, while SECPOP2000 reported 58 and 0 people (58 total), respectively.

Another factor is the level of detail for the census data that the licensee used. If the level of detail was at the county, city, or block group level, the reported values are different. As previously mentioned, some licensees used average data for rings. The reasonableness of the results in both the ring and sector attests to the fact that the ring-segment-to-ring-segment differences are not significant. For low population sites with scattered population, the percent differences could become large. If they do become large, it is not clear what effect this may have on MACCS2 calculations. In such cases, engineering judgment is needed to assess the impact of the low population on the health effects of interest, with consideration being given to the meteorology selected (wind rose).

There is a significant difference in the way licensees have projected population estimates. This can be seen by reviewing the population distribution patterns developed. For example, River Bend estimated the 2000 population distributions from the 1970 Census data. The River Bend population table (see Appendix F) has areas shaded for patterns of low, medium, intermediate and high populations. A review of these patterns indicates that while the absolute numbers may not match, the patterns are very similar between the licensee's projected population distribution and SECPOP2000's calculated population distribution. In contrast to River Bend is South Texas. While South Texas used the 1980 census data and the patterns are similar, the comparison with the SECPOP2000 results is not as good. For this assessment, it is the pattern that is important for all plants, but especially for those who only provided estimated population distributions.

An important issue that became obvious in this assessment is the importance of the exact physical location of the center point for the calculation. One plant, not reported here, was investigated in an attempt to match the population distributions. The coordinates initially provided were over the ocean. Other coordinates were obtained from other sources and none of them were accurate. By using commercial mapping software, the correct coordinates were obtained. This resulted in greatly improved population distribution patterns. This underscores two points to keep in mind when performing SECPOP2000 calculations: (1) the offsite consequences calculated by MACCS2 are potentially, significantly affected by inaccuracy in identification of the latitude and longitude, and (2) review of the SECPOP2000 population distribution pattern for reasonableness is essential for confidence that the latitude and longitude values are correct.

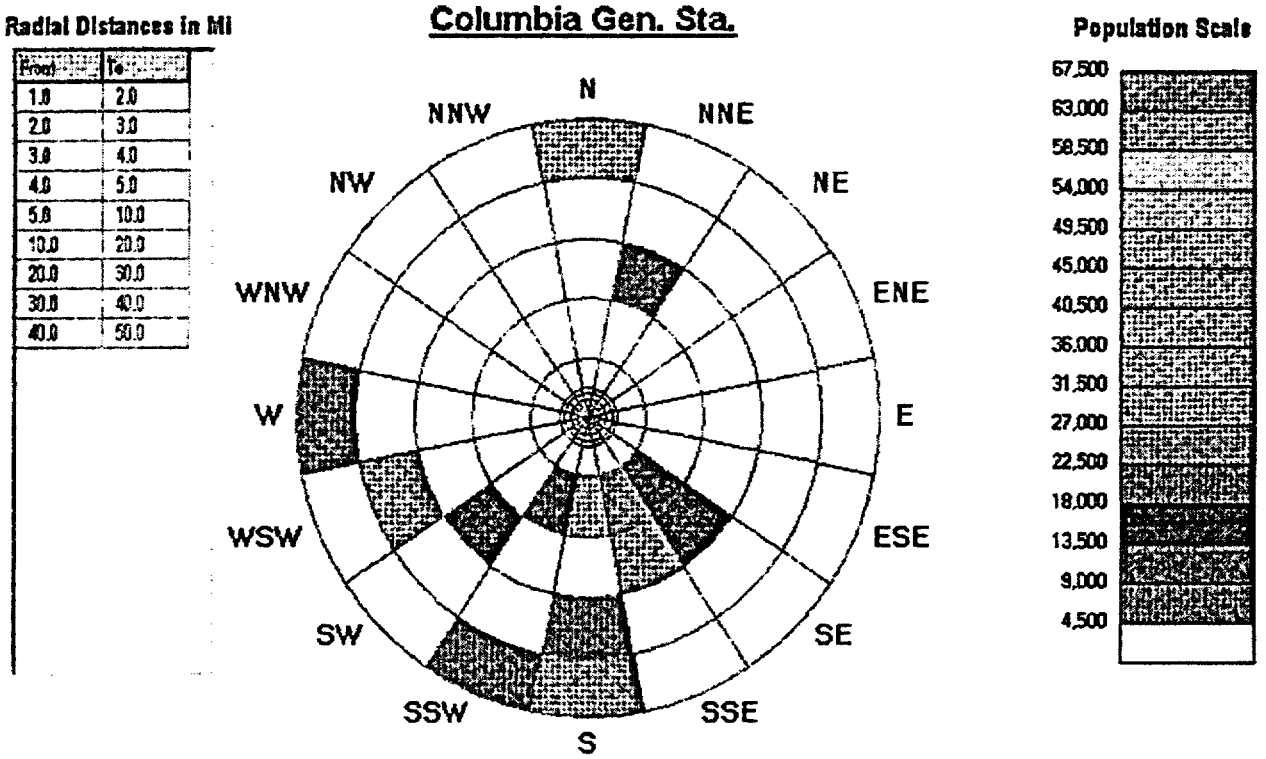


Figure 5-16 Columbia Power Station Rosette

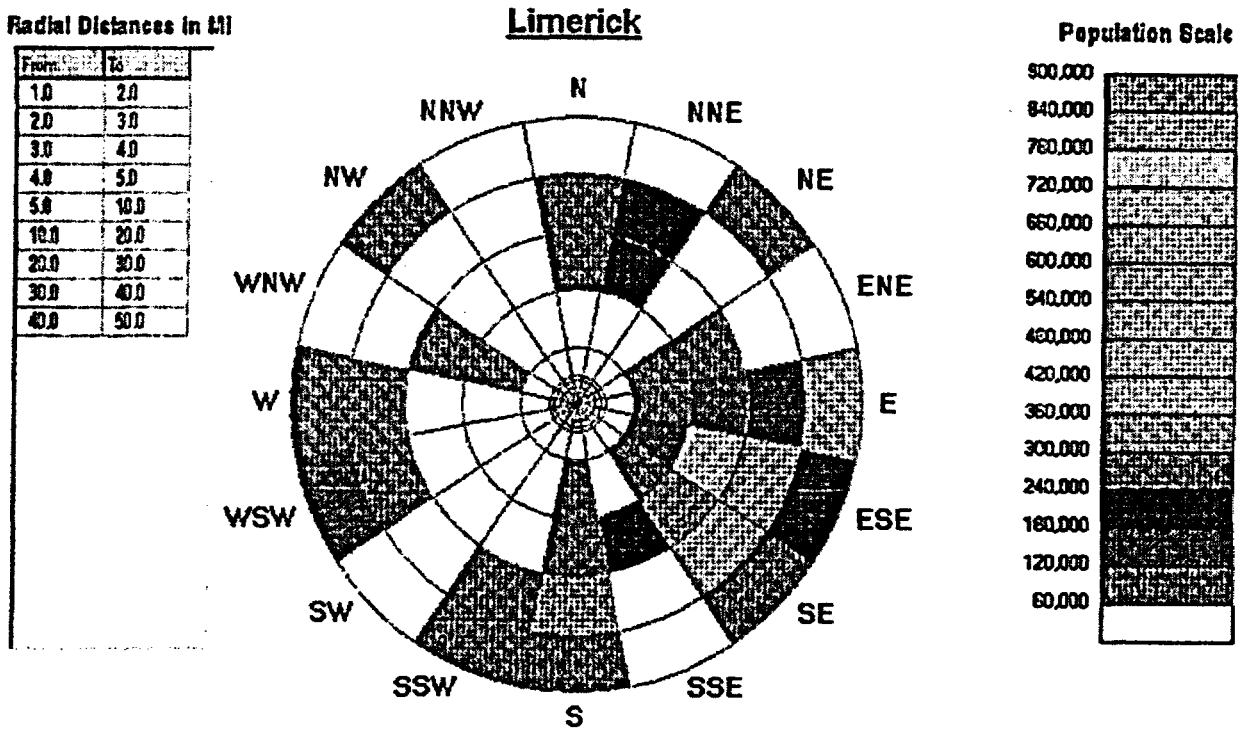


Figure 5-17 Limerick Rosette

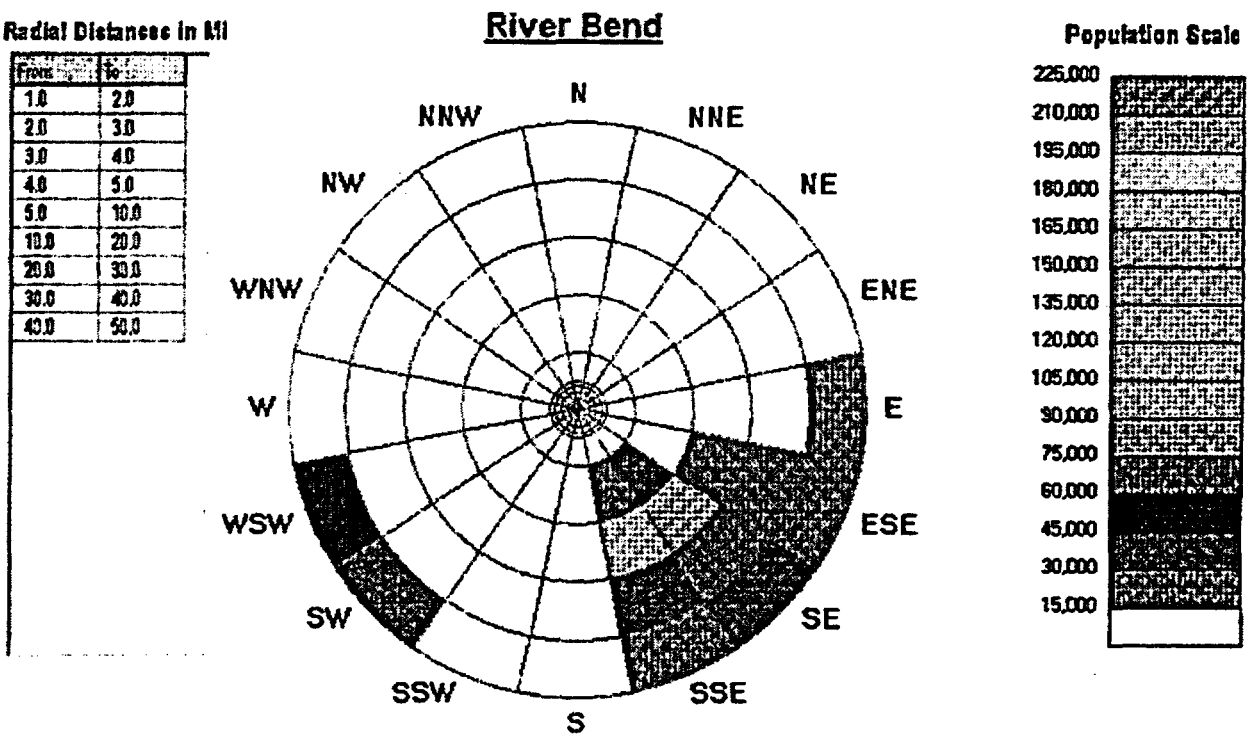


Figure 5-18 River Bend Rosette

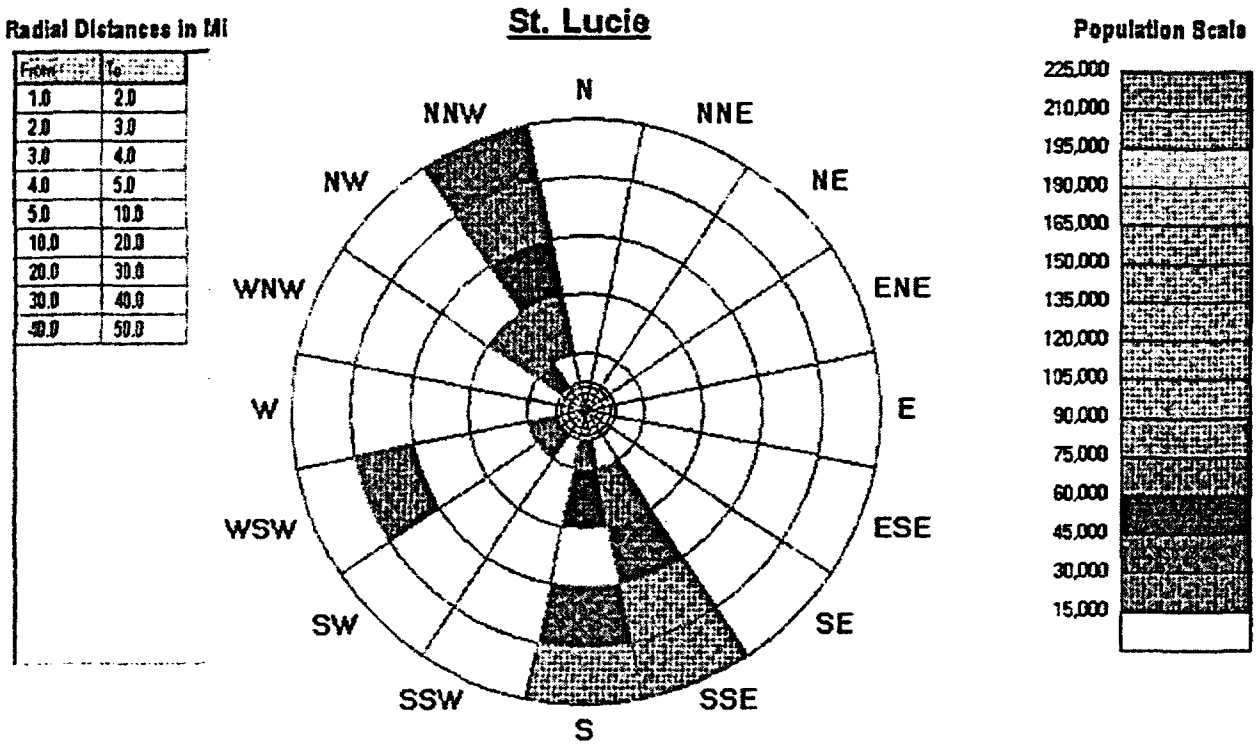


Figure 5-19 St. Lucie Rosette

6 References

1. Humphreys, S.L., Rollstin, J.A., Ridgely, J.N., SECPOP90: Sector Population, Land Fraction and Economic Estimation Program, NUREG/CR-6525 or Sand93-4032, 1993

2. Census 2000 Summary File1; Census of Population and Housing, U.S. Department of Commerce Economics and Statistics Administrations, U.S. Census Bureau; DVD issued September 2001. V1-D00-S1AS-08-US1.

3. Sprung, J.L., Rollstin, J.A., Helton, J.C., Jow, H-N, Evaluation of Severe Accident Risks: Quantification of Major Input Parameters, NUREG/CR-455 1, SAND86-1309, Vo12, Rev. 1, Part 7, U.S. Nuclear Regulatory Commission, Washington, DC, December 1990, pg. 5-4

4. James A. FitzPatrick Emergency Plan, Volume 1, Revision 2, dated September 9, 1992.

5. Millstone I Updated Final Safety Analysis Report, Chapter 2, Revision 6, dated May 1993.

6. Monticello Final Safety Analysis Report, Tables 2.2-1 and 2.2-2, Revision 11, dated December 199 1.

7. Nine Mile Point 2 Safety Evaluation Report, dated August 7, 1991, TAC Number 69095 [50 Mile radius].

8. Oyster Creek, GPU submittal dated May 19, 1992, C321-92-2118 [10 Mile radius].

9. Peach Bottom Atomic Power Station, Philadelphia Electric Company submittal dated May 21, 1992, Accession Number 9206020160 [60 Mile radius].

10. DC Cook Final Safety Analysis Report, Page 2.1-8 and Figure 2.1-5, updated July 1993.

11. Diablo Canyon Final Safety Analysis Report, Chapter 2, Revision 8, dated September 1992.

12. Fort Calhoun Station Radiological Emergency Response Plan, Section J, Figures J-6 through J-8, Pages 24 through 26.

13. Evacuation Time Estimates for the Prairie Island Nuclear Plant Plume Exposure Pathway Emergency Planning Zone, Table 3. 1, Page 3-3, dated December 1992.

14. Turkey Point Final Safety Analysis Report, Tables 2.4-1 and 2.4-3, Revision 10, dated July 1992.

15. Browns Ferry Final Safety Analysis Report, Tables 2.1-4 and 2.1-5.

16. Columbia Power Station (formerly Washington Nuclear Project (WNP-2)) Final Safety Analysis Report, Amendment 53, Table 2.1-1.

17. D.C. Cook Final Safety Analysis Report, Revision 16. 1, Table 2.2-7.

18. Grand Gulf Nuclear Station Final Safety Evaluation Report, Tables 2. 1 -1 and 2.1-3.

19. Limerick Generating Station Final Safety Analysis Report, Tables 2.1-5 and 2.1-12.

20. Point Beach Final Safety Analysis Report, Revision 1, Figures 2.3-1 and 2.3-2.

21. H.B. Robinson Final Safety Analysis Report, Tables 2.1.3-1 and 2.1.3-2.

22. Seabrook Final Safety Analysis Report, Tables 2.1-2 and 2.1-4.

23. South Texas Final Safety Analysis Report, Table 2.1-1.

24. Susquehanna Steam Electric Station Final Safety Analysis Report, Revision 54, Tables 2.1-19 and 2.1-20.

25. River Bend Final Safety Analysis Report, Tables 2.1-6 and 2.1-13, Revision dated August 1987.

Appendix A Site List

Table A-1 lists all the site files provided with the SECPOP2000 CD in the SECPOP\Sites directory. It should be noted that neither Sandia National Laboratories nor the NRC warrants that these coordinates are accurate.

Table A.1 SECPOP2000 Site File Listing

Advanced Medical Systems.sit	Hematite - Combustion Engr.sit	Prairie Island.sit
Arkansas.sit	Honeywell.sit	Quad Cities.sit
Beaver Valley.sit	Hope Creek.sit	Rancho Seco.sit
Big Rock Point.sit	Idaho NatL Engineering And Environ.sit	Richland Framatome.sit
Braidwood.sit	Indian Point.sit	River Bend.sit
Browns Ferry.sit	Indian Point 2.sit	Robinson.sit
Brunswick.sit	Joseph M Farley.sit	Safety Light Corp.sit
Bwx Technologies.sit	Kewaunee.sit	Salem.sit
Byron.sit	Lacrosse.sit	San Onofre.sit
Callaway.sit	Lasalle.sit	Seabrook.sit
Calvert Cliffs.sit	Limerick.sit	Sequoyah.sit
Catawba.sit	Maine Yankee.sit	Shearon Harris.sit
Clinton.sit	Mallinckrodt.sit	South Texas.sit
Columbia Gen Sta.sit	Mcguire.sit	Squibb.sit
Comanche Peak.sit	Millstone.sit	St Lucie.sit
Cooper.sit	Monticello.sit	Surry.sit
Crystal River.sit	Morris General Electric.sit	Susquehanna.sit
Davis-Besse.sit	Nine Mile Point.sit	Three Mile Island.sit
Diablo Canyon.sit	North Anna.sit	Trojan.sit
Donald C Cook.sit	Nuclear Fuel Services.sit	Turkey Point.sit
Dresden.sit	Oak Ridge.sit	Vermont Yankee.sit
Duane Arnold.sit	Oconee.sit	Virgil Summer.sit
Fermi.sit	Oyster Creek.sit	Vogtle.sit
Fitzpatrick.sit	Paducah.sit	Waterford.sit
Ft Calhoun.sit	Palisades.sit	Watts Bar.sit
General Atomics.sit	Palo Verde.sit	Westinghouse Elec Corp.sit
Ginna.sit	Peach Bottom.sit	Wilmington Ge.sit
Grand Gulf.sit	Perry.sit	Wolf Creek.sit
Haddem Neck.sit	Pilgrim.sit	Yankee Rowe.sit
Hanford Meteorological Tower.sit	Point Beach.sit	Zion.sit
Hatch.sit	Portsmouth.sit	

Appendix B MACCS2 Site Data File Format

The following is taken directly from and referenced to NUREG/CR 6613, *Code Manual for MACCS2: User's Guide, Volume 1*, pp. A-6 – A-16, May 1998 for the user's convenience.

In the Site Data file, the user specifies the population distribution and land use information for the region surrounding the site. Contained in the Site Data file are the geometry data used for the site (spatial intervals), population distribution, fraction of the area that is land, watershed data for the liquid pathways model, information on agricultural land use and growing seasons, and regional economic information. An example of a Site Data file is provided in NUREG/CR-4691, Volume 1, Appendix D.2.

The user specifies in the EARLY input file whether a Site Data file is to be used (see variable POPFLG in Subsection 6.3). If a Site Data file is not being used, the population density applied in the EARLY and CHRONC modules is specified in the EARLY input file. It is not possible for the user to supply differing population data for the two modules.

The Site Data file used in MACCS2 is a formatted file. The data must appear exactly as described below and in exactly the same order. In contrast to the input files for MACCS2, which are processed by a free-format input processor, the Site Data file is processed with fixed format FORTRAN READ statements.

The use of fixed format READ statements requires that the user exercise special attention to line up the data items in their proper fields. Any numeric items specified in exponential format (*e.g.*, 1.E–6) must be right-justified in the field because trailing blanks are processed as zeroes in the FORTRAN READ statements.

In contrast to the ATMOS, EARLY, and CHRONC user input files, where every value is verified by the code to ensure that it lies within a range of allowable values, the verification performed on the Site Data is only partially complete. Some of the input parameters on this file are rigorously checked to ensure that they fall within the allowed range, while other values are not checked at all.

It is recommended that the user exercise scrupulous care in constructing a Site Data file. It is very important that all items appear in their proper fields and that all numeric values lie within the range of acceptable values. Failure to conform to these requirements may lead to the generation of spurious results.

When code users edited the population counts in the Site Data file using text editors, there were several occurrences where the code diagnosed an input conversion error and aborted execution, and it has proved very difficult to identify and remedy the problem. A possible explanation for these problems is the inadvertent inclusion of nonprintable ASCII characters or control codes in the file. A likely culprit appears to be the TAB character (ASCII code decimal 9). Control codes such as the

TAB can be treated differently by different text editors, or their handling can vary, depending on user-specified editor settings.

In order to aid users in identifying the cause of these problems, and maximize the portability of the Site Data file across computer platforms, MACCS2 incorporates a test for nonprintable characters in the site data file. If any such characters are found to be present, their location on the input data line is indicated, and further execution is terminated.

The ANSI standard for FORTRAN 77 does not specify how the TAB and other control codes should be handled if they are encountered during formatted input operations. The set of processor characters handled by standard FORTRAN 77 is limited to the blank, numeric digits, upper case letters, and the following twelve symbols: '+-*/(),.:=$. However, it seems logical to extend this set to include all of the printable characters that can be generated by a IBM PC-AT-compatible keyboard; that is, the characters with decimal ASCII control codes between 32 (blank) and 126 (~). As a result, the definition of allowable characters in the site data file is that it can be allowed to contain all of the characters having decimal ASCII control codes between 32 and 126, inclusive.

A sample MACCS2 site data file is listed in Figure B-1. The first two records of the site data file contain identification information. Up to 80 characters may be used on each line. This header information is printed on the output listing.

Following the descriptive text fields, there are six data records that specify the amount of data that is being supplied on the file. The values defined on these data records must be consistent with the MACCS2 model definition data defined by the ATMOS, EARLY, and CHRONC input files. The value of the Site Data file input variable and the corresponding ATMOS, EARLY, and CHRONC variable must be identical. If any inconsistencies are detected, execution of the program is terminated.

The data are input as integers and the format is as follows:

Line	Colu	Fortran Format Descriptor	Site-File Variable	Identification	MACCS2 Variable	Allowed Range
3	1-4	I4	NSPDTS	No. of spatial intervals	NUMRAD	2-35
4	1-4	I4	NWNDIR	No. of wind directions	NUMCOR	16-16
5	1-4	I4	NCPGZN	No. of food crops	NFICRP	1-10
6	1-4	I4	NWPISO	No. of water radionuclides	NUMWPI	1-10
7	1-4	I4	NWTRSH	No. of watersheds	NUMWPA	1-4
8	1-4	I4	NECRGN	No. of economic regions	N.A.	1-99

Eight blocks of site data follow the introductory block described above. Each of these data blocks is introduced by a separator line that identifies the block of data to follow. The first line of each data block must be the separator for that block. The first character (column 1) of the separator line is ignored and the following 22 characters must match the identification field for that specific data block.

The following character separators identify the eight blocks of data:

26. SPATIAL DISTANCES
27. POPULATION
28. LAND FRACTION
29. REGION INDEX
30. WATERSHED INDEX
31. CROP SEASON AND SHARE
32. WATERSHED DEFINITION
33. REGIONAL ECONOMIC DATA

```
MACCS SITE DATA FILE
 SAMPLE SITE FILE FOR MACCS2 DOCUMENTATION
  26 SPATIAL INTERVALS
  16 WIND DIRECTIONS
   7 CROP CATEGORIES
   4 WATER PATHWAY ISOTOPES
   2 WATERSHEDS
  59 ECONOMIC REGIONS
 SPATIAL DISTANCES
      0.16       0.52       1.21       1.61       2.13       3.22       4.02       4.83
      5.63       8.05      11.27      16.09      20.92      25.75      32.19      40.23
     48.28      64.37      80.47     112.65     160.93     241.14     321.87     563.27
    804.67    1609.34
 POPULATION
        0.         0.         0.         0.         0.         0.         4.         5.
        6.        25.      3341.      7107.      2173.         0.      1305.       474.
     2252.      2945.      5403.     20169.    112004.   3431358.   1355700.   2742710.
  2487346.    104331.
        0.         0.         0.         0.         1.         2.         9.        13.
       15.        63.      1667.      3550.      1330.      1072.      3198.      2425.
      515.      9469.      5317.      7120.     13586.    198785.   1058744.  20508438.
  3290082.    830354.
        0.         0.         0.         0.         0.         0.         5.         6.
        8.        31.       822.      1752.      4543.      1713.      1597.      2296.
     6535.      1775.         0.      8555.     48596.    119411.    233382.   3003954.
  7620063.   1169436.
        0.         0.         0.         0.         0.         0.         1.         1.
        2.        11.       543.      1157.      3820.      1621.      3364.         0.
        0.       129.      6679.     11858.         0.         0.         0.         0.
        0.         0.
        0.         0.         0.         0.         0.         0.         0.         0.
        0.         0.      4798.     10202.     10348.     10480.      9570.         0.
        0.      2317.      1756.         0.         0.         0.         0.         0.
        0.         0.
        0.         0.         0.         0.         0.         0.         1.         1.
        1.         7.      8316.     17684.     16340.     30419.     39474.     74998.
    24195.     80412.     57477.         0.         0.         0.         0.         0.
        0.         0.
        0.         0.         0.         0.         0.         0.         0.         0.
        0.         0.         0.         0.      1722.      6433.     36763.     20632.
   126203.    372471.     68327.      8599.      6339.      1057.         0.         0.
        0.         0.
        0.         0.         0.         0.         0.         0.         2.         2.
        3.        13.       127.       273.      1649.      4571.      3441.      7838.
    11747.     19019.      3360.     36387.     10447.     12402.         0.         0.
```

Figure B-1 MACCS/MACCS2 sample Site Data file (continued on following page)

```
      0.       0.
      0.       0.       5.       4.       8.      23.      14.      20.
     23.      93.     301.     650.       0.       0.    1264.    4065.
   1106.   14665.    4071.   18006.   37417.   89072.   81626.       0.
      0.       0.
      0.       0.       0.       0.       0.       0.      19.      25.
     29.     117.      45.     105.       0.     510.     951.    1521.
   1223.   17636.    4926.   30765.   53265.   289674.  216165.  479431.
 280809.  8801784.
      0.       0.       0.       0.       1.       2.      14.      20.
     23.      93.     155.     338.     125.    1079.       0.    1355.
   2765.     154.    5296.   21409.   62228.   523803.  479588. 1538059.
1526840.  3099458.
      0.       0.       0.       0.       1.       2.      14.      20.
     23.      93.     110.     240.    1056.       0.      50.    1396.
    915.    3153.    4132.   16295.   35596.   239712.  709522. 2845970.
3957581. 10560254.
      3.      13.     127.     273.    1649.    4571.    3441.    7838.
  11747.   19019.    3360.   36387.   10447.   12402.       0.       0.
      0.       0.
      0.       0.       5.       4.       8.      23.      14.      20.
     23.      93.     301.     650.       0.       0.    1264.    4065.
   1106.   14665.    4071.   18006.   37417.   89072.   81626.       0.
      0.       0.
      0.       0.       0.       0.       0.       0.      19.      25.
     29.     117.      45.     105.       0.     510.     951.    1521.
   1223.   17636.    4926.   30765.   53265.   289674.  216165.  479431.
 280809.  8801784.
      0.       0.       0.       0.       1.       2.      14.      20.
     23.      93.     155.     338.     125.    1079.       0.    1355.
   2765.     154.    5296.   21409.   62228.   523803.  479588. 1538059.
1526840.  3099458.
      0.       0.       0.       0.       1.       2.      14.      20.
     23.      93.     110.     240.    1056.       0.      50.    1396.
    915.    3153.    4132.   16295.   35596.   239712.  709522. 2845970.
3957581. 10560254.
      0.       0.       0.       0.       0.       0.      25.      33.
     38.     154.      30.      70.     450.       0.     980.     517.
    155.   66531.   40902.    9557.   44818.   194801.  376828. 1492286.
2250273. 12145932.
      0.       0.       0.       0.       0.       0.       7.       9.
     12.      47.      31.      69.       0.     380.     281.     445.
   1986.   32459.  183133.  193630.   30369.   203275.   94113. 1328987.
5086913. 19537940.
      0.       0.       0.       0.       0.       0.       0.       0.
      0.       0.     223.     477.       0.    1026.     609.    2575.
   2794.    6593.   96857.  107328.   47585.   156826.  101785. 4175263.
7535605.  9667977.
      0.       0.       0.       0.       0.       0.      15.      20.
     23.      92.    2503.    5326.    3508.    1826.    1884.     275.
   3965.    2084.    6270.   10765.  103787.   970659.  472558. 1396088.
1969210.   73968.
LAND FRACTION
1.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.80 1.00 1.00 0.95 0.75 0.70 0.85
1.00 0.85 0.70 0.75 0.55 0.70 0.60 1.00 1.00 0.95
1.00 1.00 1.00 1.00 0.90 0.70 0.40 0.00 0.00 0.45 1.00 0.95 0.40 0.60 1.00 1.00
0.90 0.45 0.60 0.20 0.50 0.50 0.30 0.25 0.50 0.60
1.00 1.00 1.00 1.00 1.00 1.00 0.00 0.00 0.00 0.05 1.00 1.00 0.20 1.00 0.70 0.30
0.85 0.20 0.00 0.20 0.00 0.00 0.00 0.00 0.00 0.00
1.00 1.00 1.00 1.00 1.00 0.80 0.10 0.00 0.00 0.00 1.00 1.00 0.75 0.30 0.40 0.00
```

Figure B-1 MACCS/MACCS2 sample Site Data file (continued on following page)

```
0.15 0.00 0.45 0.30 0.00 0.00 0.00 0.00 0.00 0.00
1.00 1.00 1.00 1.00 1.00 0.60 0.00 0.00 0.00 0.00 0.95 1.00 1.00 0.70 0.40 0.10
0.00 0.50 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
1.00 1.00 1.00 1.00 1.00 0.70 0.00 0.00 0.00 0.80 0.90 0.75 0.85 1.00 1.00 0.70
0.15 0.25 0.25 0.00 0.00 0.00 0.00 0.00 0.00 0.00
1.00 1.00 1.00 1.00 1.00 1.00 0.55 0.50 0.25 0.15 0.00 0.10 0.00 0.10 0.50 0.60
0.85 1.00 1.00 0.40 0.40 0.05 0.00 0.00 0.00 0.00
1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 0.90 0.35 0.40 0.80 1.00 1.00 1.00
1.00 1.00 1.00 1.00 0.95 0.20 0.00 0.00 0.00 0.00
1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00
1.00 1.00 1.00 1.00 1.00 0.80 0.40 0.00 0.00 0.20
1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00
1.00 1.00 1.00 1.00 1.00 1.00 0.75 0.50 0.40 0.20
1.00 0.70 0.40 0.20 0.75 0.95 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00
1.00 1.00 1.00 0.95 1.00 1.00 1.00 1.00 0.80 0.20
1.00 1.00 0.00 0.00 0.00 0.00 0.05 0.70 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00
1.00 1.00 1.00 1.00 0.95 0.90 1.00 1.00 1.00 1.00
1.00 1.00 0.00 0.00 0.00 0.00 0.00 0.15 0.70 0.90 1.00 1.00 1.00 1.00 1.00 1.00
1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00
1.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.20 0.50 0.40 0.30 0.25 0.75 0.80 0.85
0.85 0.95 1.00 1.00 1.00 1.00 1.00 1.00 0.90 0.85
1.00 1.00 0.00 0.00 0.00 0.00 0.00 0.15 0.80 0.75 0.80 0.85 0.95 1.00 1.00 1.00
1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 0.50 0.45
1.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.90 1.00 1.00 1.00 1.00 1.00 1.00
1.00 1.00 1.00 0.95 0.80 0.85 1.00 1.00 0.55 0.95
REGION INDEX
44445050505050505044444444444444444444181807283054
44444444444444505044444444444444444444180728301917
44444444444450505044444444444444445044185050505050
44444444444444505050444444445044504444505050505050
44444444444450505050444444444450445050505050505050
44444444444450505044444444444444444450505050505050
44444444444444444450445044444444444444313150505050
44444444444444444444444444444444444431313150505050
44444444444444444444444444444444443131313131505008
44444444444444444444444444444444444431313131380908
44444444444444444444444444444444444444313131380901
44445050505044444444444444444444444444444444314040
44445050505050444444444444444444444444444444461511
44445050505050504444444444444444444444444444463347
44445050505050444444444444444444444444444444365320
44445050505050505044444444444444444444441818363053
WATERSHED INDEX
1 1 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 2 2 2 2 1 1 2 1 1 1 1 1 1 2 2 2 2 2 2 1
1 1 1 1 1 1 2 2 2 2 1 1 2 1 1 2 1 2 2 2 2 2 2 2 2 2
1 1 1 1 1 1 2 2 2 2 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2
1 1 1 1 1 1 2 2 2 2 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2
1 1 1 1 1 1 2 2 2 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2
1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 1 1 1 1 2 2 2 2 2 2 2
1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 2 2 2 2 2
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2
1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

Figure B-1 MACCS/MACCS2 sample Site Data file
(continued on following page)

```
CROP SEASON AND SHARE
  1 PASTURE                90. 270.   0.41
  2 STORED FORAGE         150. 240.   0.13
  3 GRAINS                150. 240.   0.21
  4 GRN LEAFY VEGETABLES  150. 240.   0.002
  5 OTHER FOOD CROPS      150. 240.   0.004
  6 LEGUMES AND SEEDS     150. 240.   0.15
  7 ROOTS AND TUBERS      150. 240.   0.003
WATERSHED DEFINITION
  1 SR-89                              5.0E-6    0.0
  2 SR-90                              5.0E-6    0.0
  3 CS-134                             5.0E-6    0.0
  4 CS-137                             5.0E-6    0.0
REGIONAL ECONOMIC DATA
  1 ALA              .354 .040      459.     1824.    62000.
  2 ARIZ             .516 .104      110.      682.    74000.
  3 ARK              .483 .041      466.     2049.    61000.
  4 CALIF            .330 .144     1022.     4394.    93000.
  5 COLO             .522 .048      211.      971.    83000.
  6 CONN             .160 .294     1605.     4980.   107000.
  7 DEL              .534 .042     1723.     3428.    82000.
  8 FLA              .375 .080      832.     3341.    80000.
  9 GA               .363 .060      613.     1885.    73000.
 10 IDAHO            .279 .144      343.     1562.    61000.
 11 ILL              .806 .044      709.     3900.    86000.
 12 IND              .713 .079      611.     3283.    72000.
 13 IOWA             .938 .060      695.     3133.    73000.
 14 KANS             .917 .035      281.     1204.    81000.
 15 KY               .571 .112      482.     1838.    61000.
 16 LA               .354 .074      459.     3284.    61000.
 17 MAINE            .079 .260      662.     1133.    70000.
 18 MD               .429 .216      956.     4489.    93000.
 19 MASS             .136 .249     1349.     2563.    97000.
 20 MICH             .313 .247      658.     2187.    81000.
 21 MINN             .597 .223      516.     2111.    82000.
 22 MISS             .470 .054      403.     2084.    53000.
 23 MO               .703 .102      322.     1647.    76000.
 24 MONT             .657 .030       61.      563.    65000.
 25 NEBR             .962 .031      318.     1148.    75000.
 26 NEV              .127 .139       63.      601.    84000.
 27 N.H.             .096 .482      518.     2018.    87000.
 28 N.J.             .203 .129     1399.     6477.   102000.
 29 N.MEX            .590 .144       53.      473.    63000.
 30 N.Y.             .310 .589      711.     1378.    94000.
 31 N.C.             .352 .065      860.     2658.    68000.
 32 N.DAK            .924 .048      164.      948.    69000.
 33 OHIO             .602 .175      581.     2686.    76000.
 34 OKLA             .751 .060      204.     1508.    67000.
 35 OREG             .292 .111      236.     1203.    73000.
 36 PA               .303 .447      855.     2534.    78000.
 37 R.I.             .108 .213     1062.     6438.    80000.
 38 S.C.             .290 .084      472.     1843.    62000.
 39 S.DAK            .915 .091      145.      587.    65000.
 40 TENN             .509 .153      360.     1850.    66000.
 41 TEX              .816 .064      164.     1492.    74000.
 42 UTAH             .225 .259      123.     1286.    60000.
 43 VT               .286 .789      628.     1472.    73000.
 44 VA               .382 .198      371.     2075.    84000.
 45 WASH             .377 .154      476.     1948.    82000.
 46 W.VA             .246 .224      150.     1728.    58000.
```

Figure B-1 MACCS/MACCS2 sample Site Data file
(continued on following page)

```
 47 WIS          .517 .591     723.     1751.    76000.
 48 WYO          .561 .028      43.      380.    70000.
 49 BRIT COL     .377 .154     476.     1948.    60000.
 50 OCEAN        .000 .000       0.        0.        0.
 51 SASKAT       .657 .030      61.      563.    60000.
 52 MANITOBA     .924 .048     164.      948.    60000.
 53 ONTARIO      .597 .223     516.     2111.    60000.
 54 QUEBEC       .310 .589     711.     1378.    60000.
 55 NOVA SCOT    .079 .260     662.     1133.    60000.
 56 BAJA CAL     .330 .144    1022.     4394.    10000.
 57 SONORA       .516 .104     110.      682.    10000.
 58 CHIHUAHUA    .590 .144      53.      473.    10000.
 59 COAHUILA     .816 .064     164.     1492.    10000.
END
```

Figure B-1 MACCS/MACCS2 sample Site Data file

B.1 Spatial Distances Data Block

The spatial distance data define the spatial grid for which the population and land use data are specified. The data define the distance in kilometers to the endpoints of the spatial intervals. The areas between the spatial interval endpoints within each of the 16 wind direction sectors are referred to as spatial elements. This grid definition must agree with the grid defined in the ATMOS input file (see the variable SPAEND in Subsection 5.3). A relative error of 10% in the endpoint distances is allowed. For larger discrepancies in the geometry data, the error flag is set and execution terminates upon completion of the Site Data file input processing.

The first line of the spatial distance data block contains the 22-character separator beginning with SPATIAL DISTANCES in column 2. Next, the endpoint distances in kilometers are specified, eight values per line, using the format described below. As many lines as are needed to define the spatial distances are used. The minimum spacing between adjacent spatial intervals is 0.1 km.

A maximum of eight interval endpoints may be input per line within ten column intervals; *i.e.*, the first interval endpoint would be input in columns 1–10, the second interval endpoint would be input in columns 11–20, etc. The data are read as a real numbers with two decimal places. The format used to process the data is as follows:

Columns	Fortran Format Descriptor	Site-File Variable	Identification	MACCS2 Variable	Allowed Range
1-80	8E10.2	SPDSTS	Spatial interval endpoints (km)	SPAEND	0.001-999.0

B.2 Population Data Block

The population data for each element in the spatial grid is defined here. The first line of the data block contains the 22-character separator beginning with POPULATION in column 2. Next, the number of people in each element is given for the first wind sector. (The first wind sector is assumed to be centered on north.) The population data are specified, eight values per line, using the format described below. As many lines as needed to cover all the spatial elements in the sector are

used. Proceeding in a clockwise rotation, the population data for the second (NNE) and subsequent sectors follow. Data for all 16-wind directions (sectors) must be provided. Data for each sector begin on a new line.

The population data are input as a real number although no fractional numbers are stored. The population data for up to eight spatial elements may be defined per line and the data are input per ten column intervals; *i.e.*, the population in the first element would be input in columns 1–10, the population in the second element would be input in columns 11–20, etc. The format used to process the data is as follows:

Columns	Fortran Format Descriptor	Site-File Variable	Identification	Allowed Range
1-80	8E10.0	POPDAT	Population data	0.0 - 1.E9

If the user specifies that fraction of the people or fraction of the time weighting is to be performed (see variable WTNAME in Subsection 6.6.8, the new code functions exactly the same as MACCS 1.5.11.1. In cases where the SUMPOP option is selected, the processing and interpretation of population data are treated in a different manner, as described below.

For each emergency response scenario (of the up to three allowed) defined in the EARLY input file, a distinct population distribution must be supplied in the Site Data file. In MACCS 1.5.11.1, the population data block is delimited with the header record POPULATION (see Figure B 3-1 in Appendix B) beginning at column 2, followed by the population data.

When the SUMPOP option is selected, the header cards for each population data block are POPULATION1, POPULATION2, and POPULATION3, corresponding to the respective emergency-response scenarios, all beginning at column 2. An error will result if the number of population data blocks does not equal the number of emergency-response scenarios defined in the EARLY input file.

B.3 Land Fraction Data Block

The fraction of each spatial element that is land (as opposed to lakes, oceans, *etc.*) must be defined. The first line of the data block contains the 22-character separator beginning with LAND FRACTION in column 2. Next, the fraction of area that is land in each radial spatial interval of the first sector is given. All values must be between 0 and 1. A value of 0 means the element has no land; a value of 1 means the element is all land. The land fraction data are specified, 16 values per line, with the format described below. As many lines as needed to define all the spatial intervals in the sector are used. The land fraction data for the second and subsequent sectors follow in a clockwise rotation. Data for all 16-wind direction sectors must be provided. The data for each sector begin on a new line.

The land fraction data are read as a real number and land fractions are input every five columns. The format used to process the data is as follows:

Columns	Fortran Format Descriptor	Variable	Identification	Allowed Range
1-80	16F5.2	FRCLND	Land Fraction	0.0-1.0

B.4 Region Index Data Block

In this data block the user assigns a user-defined economic region to each of the spatial intervals. The economic regions are defined in the Regional Economic data block. Subsection A.3.8 contains a description of the types of economic data that can be defined for each economic region.

The first line in the Region Index data block contains the 22-character separator beginning with REGION INDEX in column 2. The next line contains two-digit integers associating a region index with each of the spatial elements in the first sector. The data are specified with the format defined below. All of the region indices for one sector are input on one line. The region indices for the second and subsequent sectors are on the following lines, a new line for each sector in a clockwise rotation. A total of 16 lines are required. For example, a region index of 09 means that economic data for region number nine is used for the spatial element.

The format used to process the data is as follows:

Columns	Fortran Format Descriptor	Variable	Site-File Identification	Allowed Range
1-80	40I2		INDREG	Region Index 1 - NECRGN

B.5 Watershed Index Data Block

Each of the spatial intervals in the grid must be associated with one of the watershed classes. The definition of the watershed classes is provided in Subsection A.3.7. The watershed identification data block begins with the 22-character separator beginning with WATERSHED INDEX in column 2. The next line contains two-digit integers associating a watershed type with each of the spatial elements in the first sector. The data are specified with the format defined below.

All of the watershed type indices for one wind sector fit on a single line. The watershed indices for the second and subsequent sectors are on the following lines, a new line for each sector in a clockwise rotation. A total of 16 lines are required. For example, a watershed index of 1 means that the water ingestion factor for watershed type 1 is used for all material deposited on that spatial element. A watershed index of 2 means that the water ingestion factor for watershed type 2 is used for all material deposited on that spatial element.

The format used to process the data is as follows:

Columns	Fortran Format Descriptor	Site-File Variable	Identification	Allowed Range
1-80	40I2	INDWTR	Watershed Index	1-NWTRSH

B.6 Crop and Season Share Data Block

The length of the growing season and the average fraction of the farmland area at the site devoted to each crop type must be specified. These fractions need not sum to exactly 1, but their sum should not exceed a value of 1. If these values sum to a value less than 1, that sum indicates the fraction of farmland in production in an average year (some fraction of farmland may be fallow). Data must be given for each of the crop categories. The data block begins with the separator CROP SEASON AND SHARE in column 2.

The format used to process the data is as follows:

Columns	Fortran Format Descriptor	Site-File Variable	Identification	Allowed Range
1-4	I4	I	Crop Index	1-NCPGZN
6-25	A20	CROP	Crop Name	Not Applicable
26-30	F5.0	GBEG	Day of the Year the Growing Season Begins	1.0-GEND
31-35	F5.0	GEND	Day of the Year the Growing Season Ends	GBEG-365.0
36-45	F10.0	FRCLCP	Fraction of the Site-Averaged Farmland Devoted to this Crop	0.0-1.0

B.7 Watershed Definition Data Block

The data block begins with the 22-character separator beginning with WATERSHED DEFINITION in column 2. For each of the radionuclides considered in the liquid pathways model, an initial wash off fraction and an annual wash off rate (year^{-1}) must be specified together with an ingestion factor (Bq ingested/Bq in water) for each of the NUMWPA (NWTRSH) watershed classes.

The format used to process the data is as follows:

Columns	Fortran Format Descriptor	Site-File Variable	Identification	Allowed Range
1-4	I4	I	Radionuclide Index	1-NWPISO

6-13	A8	NMISO	Radionuclide Name	Not Applicable
16-25	E10.1	DEFAULT	Obsolete: No Longer Used	Not Applicable
26-35	E10.1	DEFAULT	Obsolete: No Longer Used	Not Applicable
36-45	E10.1	WTRINF (1)	Ingestion Factor for Watershed Class 1	0.0-1.0
46-55	E10.1	WTRINF (2)	Ingestion Factor for Watershed Class 2	0.0-1.0
56-65	E10.1	WTRINF (3)	Ingestion Factor for Watershed Class 3	0.0-1.0
66-75	E10.1	WTRINF (4)	Ingestion Factor for Watershed Class 4	0.0-1.0

B.8 Regional Economic Data Block

Economic data must be specified for each of the economic regions. The data block begins with the separator REGIONAL ECONOMIC DATA in column 2. An economic region is typically identified with an existing county, a state, or country to provide an indication of the source of the data or the type of geographical area it is intended to represent. The economic regions defined in this section are identified with spatial elements in the Region Index data block.

The format used to process this data is as follows:

Columns	Format Edit Descriptor	Site-File Variable	Identification	Allowed Range
1-4	I4	I	Region Index Number	1-NECRGN
6-15	A10	NMRGN	Name of the Region	Not Applicable
21-25	F5.3	FRMFRC	Fraction of Land Devoted to Farming in Region	0.0-1.0
26-30	F5.3	DPF	Fraction of Farm Sales Resulting from Dairy Production in Region	0.0-1.0
31-40	F10.1	ASFP	Total Annual Farm Sales for the Region (dollars/hectare)	0.0-1.E9
41-50	F10.1	VFRM	Farmland Property Value for the Region (dollars/hectare)	0.0-1.E9
51-60	F10.1	VNFRM	Non-farm Property Value for the Region (dollars/person)	0.0-1.E9

Appendix C 1990-2000 County Inventory Changes

The following documents from the U.S Census Bureau summarizes the changes in the inventory of counties between 1990 and 2000 census. The Miami-Dade County was given the same county ID code as the original Dade County.

1. Denali Borough, Alaska-a new county equivalent, effective December 7, 1990, organized from parts of Southeast Fairbanks and Yukon-Koyukuk Census Areas. The FIPS code for the new entity is: (02) 068.

2. Skagway-Hoonah-Angoon Census Area, Alaska-a new census area formed from part of the 1990 deleted Skagway-Yakutat-Angoon census area and from parts of Haines Borough, Juneau City and Borough, and Sitka City and Borough. The FIPS code for the new entity is: (02) 232.

3. Skagway-Yakutat-Angoon Census Area, Alaska-deleted after the 1990 census. Yakutat City and Borough organized from part of the territory, most of the remainder of the territory makes up the Census 2000 census area: Skagway-Hoonah-Angoon. The FIPS code for the deleted entity was: (02) 231.

4. Yakutat City and Borough, Alaska-a new county equivalent, effective September 9, 1992. The FIPS code for the new entity is: (02) 282.

5. Miami-Dade County, Florida-name changed from Dade County, effective November 13, 1997. The FIPS code for the newly named entity is: (12) 086. The FIPS code for the old entity was: (12) 025.

6. Yellowstone National Park County, Montana-deleted, effective November 7, 1997. Territory added to Gallatin and Park Counties.

7. South Boston City, Virginia-changed from an independent city to a town in Halifax County, Virginia, effective June 30, 1995. The FIPS code for the old entity was: (51) 780.

Appendix D Statistical Abstracts (1990)

This appendix contains the following values and source information for the parameters used in estimating the value of non-farm assets (VNFRM, $ / person).

Table D.1 1993 & 1994 Statistical Abstract of the U.S.

Reproducible Tangible Wealth (1990):	$ 2.5652E+13
Urban and Built-Up Land (1987):	77,187,303 acres
Urban and Built-Up Land (1992):	91,815,303 acres
Urban and Built-Up Land (1990 Straight Line approx.):	85,964,103 acres
Median Housing Value (1990):	$ 79,100
Total Farm Assets (1990):	$ 1.0036E+12
Farm Household Possessions (1990):	$ 4.6400E+10
Population (1990) :	248,709,873 persons

Table D.2 NUREG/CR-4551 Volume 2, Revision 1, Part 7

Typical Suburban Lot Size:	0.2 acres
Land – Fraction of Housing Value:	20 %
Average non-farm value for the U.S. (1990):	$126,632/person

Table D.3 1994 County and City Data Book

U.S. Per Capita Income (1990):	$18,696/person

No. 754. Gross and Net Stock of Fixed Reproducible Tangible Wealth in Current and Constant (1987) Dollars

[In billions of dollars. As of December 31]

ITEM	1970	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991
CURRENT DOLLARS													
Gross stock, total	4,428	14,306	15,670	16,527	17,277	18,259	19,330	20,503	21,774	22,966	24,361	25,652	26,699
Private	2,689	9,364	10,387	11,005	11,477	12,082	12,747	13,536	14,388	15,078	16,022	16,894	17,489
Nonresidential equipment	679	2,389	2,723	2,939	3,067	3,214	3,374	3,599	3,775	4,010	4,257	4,543	4,674
Nonresidential structures	790	2,683	3,032	3,247	3,376	3,573	3,783	3,951	4,156	4,478	4,755	4,964	5,110
Residential	1,219	4,292	4,632	4,818	5,034	5,296	5,590	5,986	6,457	6,590	7,010	7,387	7,705
Government	1,060	2,979	3,141	3,237	3,368	3,568	3,772	3,916	4,083	4,306	4,516	4,713	4,904
Equipment	274	488	543	583	621	657	695	732	764	814	873	949	1,018
Structures	785	2,491	2,599	2,654	2,748	2,911	3,077	3,184	3,319	3,492	3,643	3,764	3,886
Federal	424	885	968	1,018	1,067	1,119	1,176	1,227	1,269	1,325	1,394	1,469	1,543
Military	276	493	535	566	601	638	678	715	738	776	818	873	927
State and local	636	2,094	2,173	2,219	2,301	2,449	2,596	2,690	2,814	2,991	3,122	3,244	3,361
Consumer durable goods	680	1,963	2,142	2,285	2,433	2,608	2,811	3,051	3,303	3,582	3,823	4,045	4,306
Net stock, total	2,708	8,619	9,374	9,792	10,172	10,734	11,367	12,063	12,803	13,458	14,245	14,936	15,412
Private	1,674	5,814	6,413	6,737	6,987	7,348	7,752	8,224	8,729	9,108	9,650	10,130	10,412
Government	662	1,790	1,875	1,921	1,991	2,105	2,224	2,311	2,414	2,542	2,665	2,775	2,877
Consumer durable goods	372	1,014	1,086	1,134	1,194	1,281	1,391	1,527	1,660	1,808	1,930	2,031	2,125
CONSTANT (1987) DOLLARS													
Gross stock, total	12,476	17,469	17,963	18,370	18,845	19,441	20,091	20,753	21,400	22,062	22,717	23,339	23,849
Private	8,131	11,585	11,936	12,217	12,523	12,909	13,322	13,728	14,112	14,504	14,886	15,245	15,522
Nonresidential equipment	1,839	3,009	3,126	3,206	3,285	3,396	3,518	3,633	3,737	3,855	3,979	4,093	4,179
Nonresidential structures	2,411	3,255	3,384	3,506	3,606	3,729	3,870	3,983	4,088	4,191	4,293	4,398	4,477
Residential	3,881	5,321	5,427	5,505	5,632	5,783	5,934	6,112	6,286	6,458	6,614	6,754	6,866
Government	3,001	3,544	3,603	3,648	3,705	3,770	3,849	3,934	4,026	4,115	4,204	4,307	4,403
Equipment	662	625	630	634	647	670	698	730	768	802	837	879	917
Structures	2,339	2,918	2,972	3,014	3,058	3,100	3,152	3,204	3,259	3,313	3,367	3,428	3,487
Federal	1,141	1,122	1,138	1,142	1,156	1,177	1,204	1,233	1,266	1,293	1,319	1,351	1,379
Military	723	641	643	644	652	668	691	716	743	767	786	813	831
State and local	1,860	2,421	2,465	2,506	2,549	2,593	2,645	2,701	2,760	2,822	2,885	2,956	3,025
Consumer durable goods	1,344	2,340	2,424	2,505	2,617	2,762	2,920	3,091	3,262	3,444	3,627	3,787	3,924
Net stock, total	7,691	10,524	10,755	10,902	11,112	11,441	11,823	12,214	12,577	12,942	13,293	13,598	13,785
Private	5,074	7,198	7,384	7,499	7,642	7,866	8,112	8,346	8,558	8,774	8,980	9,157	9,258
Government	1,884	2,123	2,148	2,166	2,191	2,223	2,269	2,320	2,380	2,430	2,482	2,537	2,587
Consumer durable goods	733	1,203	1,223	1,236	1,278	1,352	1,442	1,547	1,639	1,738	1,831	1,903	1,940

Source: U.S. Bureau of Economic Analysis, Survey of Current Business, January and August 1992.

No. 353. Land Cover/Use, by State: 1987

[In thousands of acres. Excludes Alaska and District of Columbia]

REGION, DIVISION, AND STATE	Total surface area [1]	Federal land	Total	Developed [2]	Total	Rural land: Crop land	Pasture land	Range-land	Rural land: Forest land	Minor cover/uses
United States	1,937,726	404,063	1,484,156	77,805	1,406,351	422,416	129,021	401,885	393,904	59,820
Northeast	108,080	2,892	100,778	9,411	91,367	14,532	7,869	0	64,246	4,720
New England	42,679	1,332	38,609	3,005	35,604	2,311	1,245	0	30,307	1,739
Maine	21,290	161	19,517	508	19,009	943	419	0	16,933	714
New Hampshire	5,938	729	4,971	372	4,599	163	115	0	4,052	269
Vermont	6,153	335	5,656	308	5,348	655	388	0	4,184	122
Massachusetts	5,302	89	4,849	1,063	3,786	291	179	0	2,937	379
Rhode Island	776	4	661	161	500	22	37	0	404	37
Connecticut	3,212	14	3,056	693	2,362	239	110	0	1,797	218
Middle Atlantic	65,410	1,060	62,168	6,806	55,362	12,221	6,422	0	33,939	2,781
New York	31,428	234	29,782	2,485	27,297	5,774	3,686	0	16,650	1,187
New Jersey	4,984	143	4,563	1,325	3,239	673	229	0	1,690	647
Pennsylvania	28,957	677	27,823	2,796	25,027	5,774	2,507	0	15,398	1,348
Midwest	490,474	17,883	460,128	23,789	436,339	233,455	40,798	71,859	71,637	18,670
East North Central	159,056	6,269	148,602	12,368	136,233	72,743	12,982	0	42,482	8,027
Ohio	26,451	347	25,686	2,925	22,762	12,537	2,444	0	6,426	1,354
Indiana	23,159	487	22,302	1,780	20,522	13,930	2,073	0	3,698	821
Illinois	36,061	492	34,792	2,792	32,000	25,121	2,689	0	3,447	744
Michigan	37,457	3,130	33,051	2,921	30,130	9,484	2,735	0	15,483	2,429
Wisconsin	35,938	1,813	32,770	1,951	30,820	11,671	3,041	0	13,428	2,683
West North Central	331,408	11,615	311,527	11,421	300,105	160,713	27,756	71,859	29,165	10,643
Minnesota	54,017	3,390	47,077	2,136	44,941	22,890	3,425	157	13,852	4,617
Iowa	36,016	172	35,387	1,688	33,699	27,031	3,868	0	1,841	961
Missouri	44,606	2,060	41,655	2,163	39,491	15,090	12,606	56	10,959	781
North Dakota	45,250	1,882	42,255	1,242	41,013	28,064	1,206	9,933	428	1,382
South Dakota	49,354	2,873	45,467	1,064	44,403	17,819	2,354	22,162	565	1,518
Nebraska	49,507	652	48,218	1,250	46,867	20,601	1,957	22,800	728	782
Kansas	52,658	587	51,467	1,876	49,592	29,118	2,324	16,660	481	608
South	575,034	26,891	527,041	30,657	496,384	107,332	67,937	113,837	189,707	17,571
South Atlantic	170,409	12,757	156,186	13,346	142,840	26,425	16,165	3,592	88,408	8,190
Delaware	1,309	33	1,215	169	1,046	571	30	0	357	141
Maryland	6,695	159	6,048	937	5,111	1,795	514	0	2,415	383
Virginia	26,091	2,368	22,812	1,663	21,150	3,309	3,315	0	13,622	904
West Virginia	15,509	1,116	14,227	532	13,695	1,053	1,892	0	10,466	284
North Carolina	33,708	2,309	28,622	2,437	26,185	6,548	1,892	0	16,528	1,867
South Carolina	19,912	1,340	17,785	1,422	16,363	3,371	1,177	0	11,073	742
Georgia	37,702	2,062	34,664	2,375	32,289	6,307	3,040	0	21,860	1,083
Florida	37,545	5,369	30,825	3,766	27,059	3,592	4,205	3,592	12,088	3,583
East South Central	116,446	5,116	108,068	5,705	102,363	22,870	16,493	96	60,115	2,789
Kentucky	25,852	1,169	24,023	1,224	22,799	5,818	5,955	0	10,054	972
Tennessee	26,972	1,369	24,759	1,669	23,090	5,765	5,019	0	11,601	706
Alabama	33,091	904	31,230	1,640	29,591	4,210	3,595	96	21,017	673
Mississippi	30,521	1,674	28,056	1,172	26,884	7,078	3,924	0	15,443	439
West South Central	288,129	8,518	262,778	11,606	251,172	58,167	33,279	110,149	42,884	6,593
Arkansas	34,040	3,129	29,804	1,232	28,572	8,162	5,678	164	14,268	380
Louisiana	30,561	1,174	26,472	1,455	25,016	6,484	2,276	234	12,736	3,286
Oklahoma	44,772	1,176	42,431	1,716	40,715	11,557	7,930	14,546	5,805	517
Texas	170,756	3,040	163,871	7,203	156,763	31,844	17,735	95,204	8,476	2,410
West	764,128	357,403	390,209	13,247	382,962	66,896	12,697	215,889	63,514	18,825
Mountain	552,080	266,171	280,053	5,964	274,089	44,235	7,828	182,653	27,532	11,822
Montana	94,109	27,074	65,692	809	64,682	17,881	3,163	36,769	5,353	1,611
Idaho	53,481	33,190	19,628	477	19,152	6,532	1,354	6,596	4,071	600
Wyoming	62,593	29,457	32,576	501	32,075	2,362	928	26,784	984	1,017
Colorado	66,618	23,833	42,320	1,375	40,945	10,967	1,266	25,427	4,078	1,207
New Mexico	77,819	20,425	51,144	698	50,445	2,297	185	40,782	4,685	2,496
Arizona	72,960	30,647	41,994	1,116	40,878	1,306	81	31,867	4,912	2,712
Utah	54,336	35,876	16,440	465	15,975	2,002	563	8,507	3,194	1,711
Nevada	70,759	60,071	10,250	333	9,916	859	282	7,921	356	469
Pacific	211,440	91,232	116,176	7,283	108,893	22,662	4,869	33,337	40,982	7,043
Washington	43,608	12,471	29,947	1,564	28,383	7,758	1,421	5,574	12,834	857
Oregon	62,127	32,305	28,918	941	27,977	4,348	1,916	9,152	11,857	705
California	101,572	46,014	53,854	4,621	49,033	10,209	1,501	17,719	15,073	4,531
Hawaii	4,141	443	3,657	157	3,500	348	61	891	1,419	811

[1] Includes water area not shown separately.
[2] Includes urban and built-up areas in units of 10 acres or greater, and rural transportation.

Source: U.S. Dept. of Agriculture, Soil Conservation Service, and Iowa State University, Statistical Laboratory, Statistical Bulletin No. 790, Summary Report, 1987 National Resources Inventory, December 1989.

No 1280 Occupied Housing Units - Housing Value and Contract Rent, by State: 1990

In thousands of units, except as indicated. As of April. Based on the Census of Population and Housing; see Appendix III

STATE	SPECIFIED OWNER OCCUPIED UNITS										SPECIFIED RENTER OCCUPIED UNITS									
	Total housing units	VALUE CATEGORIES: Less than $50,000	$50,000 to $99,999	$100,000 to $149,999	$150,000 to $199,999	$200,000 to $299,999	$300,000 or more	Lower quartile value 1/ (dol.)	Median (dol.)	Upper quartile value 1/ (dol.)	Total housing units	Rent Categories: Less than $200	$200 to $299	$300 to $399	$400 to $499	$500 to $599	$600 to $749	$750 or more	No cash rent	Median (dol.)
United States	44,918.0	11,402.5	16,957.5	8,773.3	4,017.2	3,376.9	2,390.7	48,500	79,100	137,800	32,170.0	2,815.1	3,738.2	5,879.2	5,935.0	4,407.0	4,064.4	5,913.8	1,419.3	447
Alabama	753.8	343.9	310.7	62.5	20.1	11.3	5.4	35,500	53,700	78,800	420.0	81.6	87.3	102.1	67.2	28.1	15.7	6.8	38.1	325
Alaska	77.5	11.0	31.8	23.0	7.5	3.3	0.9	68,400	94,400	128,500	81.9	1.4	4.4	9.8	12.8	11.3	13.1	17.9	11.1	559
Arizona	648.7	90.7	372.7	118.8	40.6	23.9	14.0	58,800	80,100	109,900	486.8	26.5	46.1	111.5	113.1	71.2	53.0	38.6	25.6	458
Arkansas	427.7	235.6	158.8	23.3	6.8	3.5	1.6	31,100	46,300	68,700	250.1	38.8	57.8	64.8	30.3	17.2	8.5	4.3	27.6	328
California	4,698.3	119.0	646.6	812.1	851.5	1,151.0	1,120.0	127,100	195,500	294,800	4,563.4	136.2	198.5	379.5	621.9	756.4	1,804.1	1,361.8	114.9	620
Colorado	637.6	84.1	363.1	122.8	37.4	18.9	10.3	62,500	82,700	108,700	472.6	36.5	63.3	110.0	98.4	62.2	51.9	36.0	16.3	418
Connecticut	643.5	4.8	38.7	165.3	188.0	150.3	96.5	138,000	177,800	246,000	418.5	31.2	21.4	31.1	51.2	68.2	94.1	108.3	14.9	598
Delaware	137.5	11.8	56.8	38.5	15.6	10.5	4.3	74,800	100,100	145,900	72.2	6.0	4.4	8.7	16.8	16.0	10.9	8.7	3.4	485
District of Columbia	71.5	1.7	25.3	15.0	8.9	8.4	14.4	86,700	123,900	238,700	152.1	14.7	11.0	35.5	30.5	21.7	20.3	26.6	2.7	479
Florida	2,378.2	433.1	1,238.1	381.9	181.5	180.8	72.0	56,100	77,100	112,800	1,660.6	103.2	180.9	271.1	368.9	308.6	255.3	168.5	65.1	481
Georgia	1,188.8	314.5	531.2	183.2	68.9	39.9	25.1	47,300	71,300	102,100	808.4	97.8	98.8	134.3	158.9	128.1	98.8	68.2	41.4	433
Hawaii	144.4	3.3	15.1	16.7	21.1	38.7	50.5	186,800	245,300	358,800	162.8	9.1	7.5	12.8	15.6	17.5	26.2	52.3	22.1	850
Idaho	177.3	68.8	80.4	14.2	3.7	3.0	1.2	42,100	58,200	78,700	102.4	14.0	24.5	28.3	16.3	8.5	4.0	2.2	8.8	330
Illinois	2,084.7	525.1	786.8	406.2	163.7	114.5	71.6	60,800	80,900	120,300	1,470.4	130.7	159.0	261.6	306.8	278.8	185.0	184.3	44.5	445
Indiana	1,137.8	514.3	484.0	82.8	27.0	13.7	6.6	36,500	53,900	76,200	580.9	61.5	104.8	158.4	128.6	65.9	28.3	14.1	29.3	374
Iowa	566.6	317.8	200.7	27.7	7.0	3.3	1.1	30,200	45,900	65,900	285.7	40.5	86.1	76.7	30.7	22.1	10.8	4.6	14.3	336
Kansas	500.6	238.0	198.6	41.1	12.3	6.4	3.1	32,400	52,200	78,000	289.8	32.3	54.8	88.0	57.7	30.4	10.5	10.3	15.9	372
Kentucky	662.2	326.9	260.5	47.5	15.9	8.0	3.4	33,400	50,500	73,400	382.3	69.2	67.8	100.8	58.9	24.1	11.9	5.4	36.4	319
Louisiana	755.9	289.1	344.5	66.6	19.8	11.5	6.1	38,200	58,500	83,000	501.3	70.7	93.1	123.5	88.5	43.6	24.3	14.4	43.1	352
Maine	214.7	37.5	95.2	49.3	18.0	10.0	4.7	63,100	87,400	123,300	153.3	15.3	14.2	26.6	26.6	19.7	14.1	7.5	9.7	419
Maryland	970.9	87.3	300.9	274.2	130.6	103.3	65.6	79,000	116,500	160,300	598.3	41.2	36.0	65.4	94.0	100.0	128.9	108.2	18.8	549
Massachusetts	1,004.6	9.5	98.5	310.8	298.3	198.4	98.1	128,800	162,800	216,000	910.0	98.4	58.2	74.9	108.8	150.2	184.5	231.1	20.0	580
Michigan	1,916.1	737.2	814.5	219.2	79.3	46.8	20.0	38,500	60,600	90,600	968.2	82.1	125.4	208.0	223.3	140.7	92.0	54.6	32.1	423
Minnesota	894.3	204.9	482.5	130.2	38.6	21.1	9.2	52,200	74,000	98,200	445.9	54.8	54.7	85.1	94.3	66.0	50.2	28.4	12.8	422
Mississippi	441.8	249.3	158.1	24.2	7.0	3.6	1.6	31,400	45,600	67,800	247.5	48.8	64.0	54.0	32.7	15.7	7.9	3.4	20.9	309
Missouri	1,005.4	385.1	452.6	102.0	33.6	20.3	11.9	38,800	59,800	87,100	566.0	74.0	108.7	138.7	116.2	62.5	37.6	19.1	30.1	368
Montana	132.4	52.3	68.3	8.4	2.1	1.9	0.3	40,000	56,600	74,200	93.8	18.1	34.4	24.8	12.9	5.5	2.1	0.9	7.2	311
Nebraska	314.4	155.3	132.4	18.5	4.7	2.4	1.0	32,800	50,400	70,400	186.6	24.6	38.8	48.6	34.8	16.7	7.7	4.8	10.3	349
Nevada	183.8	8.4	93.3	51.3	16.8	8.1	5.8	75,400	95,700	129,400	200.2	8.4	11.5	20.0	48.5	44.4	38.9	22.9	5.5	509
New Hampshire	198.4	6.3	46.6	78.9	38.9	21.1	6.6	98,400	129,400	167,800	127.8	7.7	7.1	12.3	21.6	34.9	27.3	21.5	5.4	549
New Jersey	1,446.3	45.5	241.4	346.8	347.3	301.5	161.9	112,500	162,300	221,000	973.7	60.8	45.3	81.1	133.1	168.0	228.3	232.7	28.6	592
New Mexico	262.3	75.5	128.2	37.3	12.8	7.2	5.3	46,700	70,100	97,800	173.1	18.3	28.8	44.9	31.0	16.4	11.8	7.4	14.4	372
New York	2,387.6	248.8	674.4	431.6	433.0	381.0	217.8	78,200	131,600	200,400	3,150.6	242.8	281.8	480.1	606.9	488.7	464.7	582.9	78.5	428
North Carolina	1,218.0	382.8	575.7	156.7	56.3	33.1	18.0	44,700	65,800	94,900	777.9	83.3	121.5	187.4	162.2	90.8	48.8	22.3	60.6	382
North Dakota	105.7	50.6	46.3	5.1	1.1	0.5	0.1	30,800	50,800	70,000	76.5	18.6	18.7	19.8	11.4	4.0	1.8	0.7	8.3	313
Ohio	2,241.3	734.0	1,102.8	260.3	80.8	42.9	20.3	42,800	63,500	88,200	1,285.4	148.2	206.4	341.2	280.6	146.7	78.0	40.6	48.7	379
Oklahoma	618.3	325.7	235.6	36.8	10.9	6.1	3.3	30,800	48,100	71,000	370.7	44.2	85.0	98.7	60.5	30.1	17.8	9.0	25.4	340
Oregon	511.8	133.3	281.2	62.7	18.5	10.5	4.7	44,200	67,100	92,600	384.9	30.2	52.5	98.2	86.1	53.1	33.7	18.1	13.1	408
Pennsylvania	2,591.3	828.2	1,017.8	305.9	180.6	109.1	48.7	42,500	69,700	100,500	1,387.7	141.0	188.7	272.2	245.0	169.6	125.3	85.4	58.6	404
Rhode Island	178.5	3.0	31.0	78.4	36.6	19.4	9.0	106,000	133,500	171,800	152.0	17.9	12.1	18.0	28.1	29.2	25.3	15.7	4.8	488
South Carolina	615.4	225.7	280.1	63.5	23.7	14.8	7.6	41,100	61,100	88,100	308.9	45.3	58.5	83.9	77.1	38.5	21.2	10.4	33.0	376
South Dakota	118.1	64.2	42.4	4.8	1.0	0.5	0.2	27,300	48,200	64,500	81.2	18.2	18.4	18.7	11.6	4.5	2.5	0.9	7.5	306
Tennessee	938.4	371.8	416.4	91.5	32.2	17.8	8.7	38,600	58,400	86,300	568.9	86.6	101.6	184.5	110.3	55.1	27.6	13.9	38.3	357
Texas	2,948.1	1,151.2	1,201.1	314.8	111.3	68.5	44.2	38,300	59,600	88,200	2,332.9	170.0	348.2	614.2	480.6	265.7	188.8	130.7	118.6	396
Utah	303.7	61.1	188.6	35.2	10.6	5.7	2.5	53,100	68,900	92,000	168.8	12.4	31.3	53.8	33.9	15.7	9.6	6.3	6.0	369
Vermont	88.2	7.9	41.3	25.5	8.4	4.4	1.7	73,100	95,500	130,000	61.8	5.1	5.8	11.3	14.2	9.9	7.5	4.5	3.5	446
Virginia	1,192.1	208.7	466.2	203.9	132.8	116.5	65.0	60,100	91,000	155,800	746.2	50.6	65.7	100.8	124.0	103.8	114.3	128.9	40.1	495
Washington	896.4	126.0	371.3	192.7	101.0	67.8	38.6	63,700	93,400	143,400	687.0	46.0	73.5	138.2	152.7	108.0	85.1	58.2	26.3	445
West Virginia	350.1	166.3	156.4	18.9	5.4	2.8	0.8	31,200	47,900	68,400	168.3	30.7	41.7	40.4	21.4	6.1	3.8	1.4	20.6	303
Wisconsin	916.7	287.2	492.2	95.3	24.7	12.5	5.1	48,600	62,500	84,100	582.4	51.6	83.0	147.8	142.2	76.6	41.7	18.8	17.0	399
Wyoming	78.4	35.9	43.8	6.2	1.5	0.7	0.4	43,900	61,600	82,100	51.5	8.7	12.5	13.3	8.1	3.5	2.1	0.9	4.3	333

1/ This measure divides the distribution of value and rent categories into four equal parts. The lower quartile value is the value that defines the upper limit of the lowest one-quarter of the cases. The upper quartile value defines the lower limit of the upper one-quarter of the cases in the distribution.

Source: U.S. Bureau of the Census, 1990 Census of Housing, General Housing Characteristics, series CH-90-1, and Census of Population and Housing, 1990: Summary Tape File 1C on CD-ROM.

No. 1104. Balance Sheet of the Farming Sector
In Current and Constant (1987) Dollars

[In billions of dollars, except as indicated. As of December 31.
Includes farm operator households.
See Historical Statistics, Colonial Times to 1970,
series K 204-219, for data before revisions]

ITEM	1970	1975	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991
CURRENT DOLLARS														
Assets	324.3	579.5	1,089.0	1,088.9	1,056.2	1,063.6	975.7	892.8	847.7	911.3	951.5	985.8	1,003.6	1,004.1
Physical assets:														
Real estate	224.5	421.0	650.1	651.7	619.1	629.3	735.0	657.0	613.0	658.6	682.2	703.9	711.4	705.6
Nonreal estate	76.6	127.2	199.6	196.3	194.4	189.6	193.0	186.5	181.0	196.1	212.1	223.6	231.5	232.9
Livestock and poultry \1	23.7	29.4	60.6	53.5	53.0	49.5	49.5	46.3	47.8	58.0	62.2	66.2	70.9	68.4
Machinery, motor vehicles	34.4	63.1	86.9	92.5	92.6	92.1	91.1	88.3	86.1	84.5	86.1	89.2	88.6	88.0
Crops stored	8.5	20.5	32.7	29.5	25.8	23.6	28.1	22.9	16.3	17.5	23.3	23.4	22.8	23.6
Household furnishings, equipment	10.0	14.2	19.4	20.8	23.0	24.4	24.3	27.8	28.7	32.9	37.0	42.2	46.4	50.4
Purchased inputs	(NA)	(NA)	(NA)	(NA)	(NA)	(NA)	2.0	1.2	2.1	3.2	3.5	2.6	2.8	2.5
Financial assets	23.2	31.4	39.3	40.9	42.8	44.6	47.7	49.3	53.8	56.7	57.2	58.3	60.8	65.6
Investment in cooperatives	7.2	13.0	19.3	20.6	21.9	22.8	24.3	24.3	24.4	25.3	25.1	26.5	27.5	28.4
Other financial assets	16.0	18.4	20.0	20.3	20.9	21.8	23.4	25.0	29.4	31.4	32.1	32.0	33.3	37.2
Claims	324.3	579.5	1,089.0	1,088.9	1,056.2	1,063.6	975.7	892.8	847.7	911.3	951.5	985.8	1,003.6	1,004.1
Debt	52.8	91.5	178.7	195.4	203.1	208.5	204.3	187.9	166.6	153.7	148.5	146.0	145.1	147.0
Real estate debt	30.5	49.9	97.5	107.2	111.5	113.7	112.3	105.7	95.9	87.7	83.0	80.5	78.4	79.1
Nonreal estate debt	22.3	41.6	81.2	88.2	91.8	92.7	92.0	82.2	70.8	66.0	65.6	65.5	66.7	67.8
Equity	271.5	488.0	910.3	893.5	853.1	857.1	771.4	704.9	681.0	757.6	802.9	839.8	858.5	857.1
Farm debt/asset ratio (percent)	16.3	15.8	16.4	17.9	19.2	19.4	20.9	21.0	19.7	16.9	15.6	14.8	14.5	14.6
CONSTANT (1987) DOLLARS \2														
Assets	925.9	1,177.7	1,518.8	1,580.1	1,260.4	1,219.7	1,072.2	945.8	874.8	911.3	915.8	908.6	886.6	852.3
Debt	150.3	185.9	249.3	247.7	242.4	238.8	224.5	199.0	171.9	153.7	142.9	134.6	128.2	124.8
Equity	775.6	991.8	1,269.6	1,132.4	1,018.0	982.9	847.7	746.7	702.8	757.6	772.8	774.0	758.4	727.6

NA Not available.
\1 Excludes horses and mules.
\2 Constant dollar figures are based on
gross domestic product implicit price deflators for year.

Source: U.S. Dept. of Agriculture, Economic Research Service,
Economic Indicators of the Farm Sector:
National Financial Summary, 1991.

Appendix E Statistical Abstracts (2000)

Selected County Level Database Sources

Table E.1 Fixed Reproducible Tangible Wealth

Current-Cost Net Stock of Fixed Reproducible Tangible Wealth, 1925-1997
(Billions of dollars; yearend estimates)

Yearend	Total	Fixed private capital Government-owned fixed					Government-owned fixed capital			Durable goods
		Total	Nonresidential			Residential	Total	Federal	State and local	
			Total	Equipment	Structures					
1987...	15,503.2	10,796.1	5,528.4	2,062.5	3,465.8	5,267.7	3,200.2	938.8	2,261.3	1,506.9
1988...	16,501.3	11,500.3	5,897.6	2,195.5	3,702.1	5,602.7	3,359.6	993.7	2,365.9	1,641.1
1989...	17,447.4	12,149.4	6,238.3	2,322.0	3,916.3	5,911.1	3,534.6	1,041.9	2,492.7	1,763.4
1990...	18,283.7	12,706.7	6,559.4	2,452.2	4,107.3	6,147.3	3,710.7	1,089.6	2,621.1	1,866.3
1991...	18,717.5	12,955.2	6,696.7	2,519.5	4,177.2	6,258.5	3,827.2	1,126.6	2,700.6	1,935.1
1992...	19,479.8	13,484.1	6,892.7	2,590.0	4,302.7	6,591.4	3,990.7	1,168.9	2,821.8	2,005.0
1993...	20,517.9	14,198.8	7,215.6	2,686.7	4,528.9	6,983.1	4,201.2	1,227.0	2,974.3	2,107.9
1994...	21,716.6	15,064.5	7,598.7	2,823.1	4,775.6	7,465.8	4,425.8	1,263.0	3,162.8	2,226.3
1995...	22,705.5	15,736.1	7,957.1	2,980.2	4,976.9	7,779.0	4,652.5	1,294.9	3,357.6	2,317.0
1996...	23,766.4	16,496.7	8,311.2	3,116.5	5,194.7	8,185.5	4,863.7	1,326.9	3,536.8	2,406.0
1997...	24,883.3	17,316.3	8,725.3	3,257.8	5,467.5	8,590.9	5,074.7	1,334.7	3,740.1	2,492.3

Source: Department of Commerce
Bureau of Economic Analysis
Survey of Current Business, September 1998
Fixed Reproducible Tangible Wealth in the United States
Revised Estimates for 1995-1997 and Summary Estimates for 1925-1997

Table E.2 Non-Federal Land Use

Surface area of non-federal and federal land and water areas, by state and year (data per 1,000 acres)

State	Year	Federal land	Water areas	Non-federal land			Total Surface Area
				Developed	Rural	Total	
Alabama	1982	949	1,167	1,617	29,691	31,308	33,424
	1987	950	1,181	1,807	29,485	31,292	33,424
	1992	970	1,201	1,937	29,316	31,253	33,424
	1997	998	1,223	2,252	28,950	31,203	33,424
Arizona	1982	31,005	183	1,089	40,688	41,776	72,964
	1987	30,790	185	1,271	40,718	41,989	72,964
	1992	30,426	187	1,378	40,974	42,351	72,964
	1997	30,426	189	1,491	40,858	42,349	72,964
Arkansas	1982	3,042	819	1,143	29,033	30,180	34,037
	1987	3,049	853	1,180	28,955	30,135	34,037
	1992	3,103	859	1,240	28,835	30,075	34,037
	1987	3,103	887	1,409	28,638	30,048	34,037
↕	↕	↕	↕	↕	↕	↕	↕
Total	1982	399,484	48,714	73,146	1,422,686	1,495,932	1,944,130
	1987	399,886	49,894	79,505	1,414,846	1,494,350	1,944,130
	1992	401,988	49,490	87,035	1,405,617	1,492,652	1,944,130
	1997	402,138	49,980	98,252	1,393,760	1,492,011	1,944,130

Source: United States Department of Agriculture
Natural Resources Conservation Service
Summary Report, 1997 National Resources Inventory, Revised December 2000

Table E.3 Example Housing Value Table

Value, Purchase Price, and Source of Down Payment - Owner Occupied Units

(Numbers in thousands)

Characteristics	Total occupied units	Housing unit characteristics: New construction 4 years	Mobile homes	Physical problems: Severe	Physical problems: Moderate	Household characteristics: Black	Hispanic	Elderly (65 years and over)	Moved in past year	Below poverty level
Total	65 487	4 684	5 203	725	2 170	5 457	3 648	16 402	5 883	6 819
Value										
Less than $10,000	2 661	66	1 675	76	165	504	160	601	211	563
$10,000 to $19,999	1 713	124	1 004	55	189	237	115	471	156	337
$20,000 to $29,999	2 947	236	838	41	207	298	202	858	183	386
$30,000 to $39,999	2 658	212	460	65	221	471	198	826	239	466
$40,000 to $49,999	2 944	140	353	48	173	442	185	896	250	432
$50,000 to $59,999	2 897	145	198	72	172	450	237	1 155	347	458
$60,000 to $69,999	2 328	143	238	40	155	439	247	1 163	303	450
$70,000 to $79,999	4 782	165	47	67	111	482	327	1 803	332	492
$80,000 to $99,999	5 133	449	185	101	124	703	436	2 458	644	777
$100,000 to $119,999	5 600	490	82	69	112	444	397	1 898	461	672
$120,000 to $149,999	6 114	702	75	56	124	494	448	1 747	707	559
$150,000 to $199,999	5 617	679	64	67	164	457	442	1 736	605	486
$200,000 to $249,999	5 771	496	3	15	49	124	167	740	323	715
$250,000 to $299,999	2 630	255	15	10	34	65	61	419	165	173
$300,000 or more	2 641	376	6	7	30	63	62	643	268	220
Median	86 818	128 701	20 670	75 300	65 148	72 730	65 770	80 294	108 101	72 853
Ratio of Value to Current Income										
Less than 1.5	16 798	1 820	3 502	307	812	1 702	1 639	2 343	1 280	841
1.5 to 1.9	5 828	671	270	76	277	744	470	1 842	473	154
2.0 to 2.4	7 071	730	249	67	231	635	447	1 321	739	189
2.5 to 2.9	5 513	638	161	46	124	452	341	1 312	628	169
3.0 to 3.9	6 780	675	178	44	184	457	408	2 676	624	232
4.0 to 4.9	3 804	245	105	41	69	295	198	1 676	276	245
5.0 or more	11 854	663	377	143	410	950	676	6 466	671	3 817
Zero or negative income	1 207	84	133	28	48	166	87	473	100	1 150
Median	2.2	2.3	1.6	2.0	2.2	2.2	2.5	4.8	2.2	5.0+
Other Activities on Property										
Medical or commercial establishment	539	12	48	5	34	31	31	121	33	60
Neither	64 948	4 662	5 205	720	2 136	5 426	3 624	16 272	5 661	6 661
Year Unit Acquired										
2000 to 2004	–	–	–	–	–	–	–	–	–	–
1995 to 1999	13 387	[illegible]	1 634	128	417	1 166	925	980	4 929	1 084
1990 to 1994	16 943	–	1 604	141	496	1 139	1 074	1 676	80	1 309
1985 to 1989	10 311	–	852	136	328	791	805	1 708	59	844
1980 to 1984	8 444	–	449	85	187	426	370	1 223	34	539
1975 to 1979	6 203	–	845	64	179	548	345	1 708	6	615
1970 to 1974	4 714	–	208	69	128	457	165	1 832	6	642
1960 to 1969	5 602	–	77	77	221	667	175	3 415	6	737
1950 to 1959	3 219	–	7	31	144	270	123	2 059	1	677
1940 to 1949	1 513	–	–	30	48	75	35	989	1	547
1939 or earlier	280	–	–	4	28	19	6	283	–	85
Not reported	–	–	–	–	–	–	–	–	–	–
Median	1988	1995+	1992	1987	1987	1987	1991	1972	1995+	1984
First Time Owners										
First home ever owned	28 382	1 376	2 163	434	1 240	3 679	2 189	6 697	2 901	3 343
Not first home	34 971	3 180	2 844	258	831	1 549	1 338	10 166	2 829	3 048
Not reported	2 134	128	147	33	100	230	139	438	166	428
Purchase Price										
Home purchased or built	61 479	4 787	4 858	634	1 887	4 875	3 430	15 331	4 775	5 846
Less than $10,000	4 108	37	808	114	361	437	212	2 188	172	1 021
$10,000 to $19,999	5 360	71	1 602	67	291	717	218	2 140	181	865
$20,000 to $29,999	5 120	230	695	75	170	464	265	1 720	234	578
$30,000 to $39,999	4 665	353	627	63	155	439	234	1 146	217	512
$40,000 to $49,999	3 183	331	391	39	91	329	198	781	225	315
$50,000 to $59,999	3 525	188	163	30	105	324	224	626	260	275
$60,000 to $69,999	3 391	157	84	52	66	320	212	659	251	208
$70,000 to $79,999	3 105	173	72	57	66	345	184	483	263	137
$80,000 to $99,999	6 180	601	56	52	49	258	250	633	649	187
$100,000 to $119,999	3 141	412	13	14	62	126	161	434	397	120
$120,000 to $149,999	3 654	653	15	57	61	162	246	300	570	111
$150,000 to $199,999	3 611	702	10	18	70	144	233	322	488	103
$200,000 to $249,999	1 608	315	7	5	65	60	68	170	238	62
$250,000 to $299,999	685	194	–	2	28	26	59	77	120	31
$300,000 or more	1 428	254	2	–	27	37	34	113	211	80
Not reported	7 807	419	696	120	274	603	411	2 785	459	1 421
Median	67 847	117 858	21 436	33 847	28 141	38 614	60 378	35 800	94 728	26 642
Received as inheritance or gift	2 246	86	248	65	202	429	108	605	121	488
Not reported	1 762	120	150	22	61	153	108	354	185	285
Major Source of Down Payment										
Home purchased or built	61 479	4 787	4 858	634	1 887	4 825	3 430	15 334	4 775	5 846
Sale of previous home	19 427	2 675	1 367	104	308	626	603	5 670	1 481	1 447
Savings or cash on hand	29 283	1 645	2 185	129	678	3 044	2 157	5 745	2 256	2 673
Sale of other investment	452	42	32	4	10	53	19	157	69	80
Borrowing, other than mortgage on this property	2 502	85	251	47	103	217	120	446	183	379
Inheritance or gift	1 406	70	86	36	66	88	88	324	141	170
Land where building built used for financing	433	101	43	–	15	47	33	80	33	49
Other	3 582	175	825	32	101	102	144	404	219	349
No down payment	3 697	208	451	215	224	488	194	886	281	535
Not reported	2 188	118	176	25	79	281	101	613	143	272

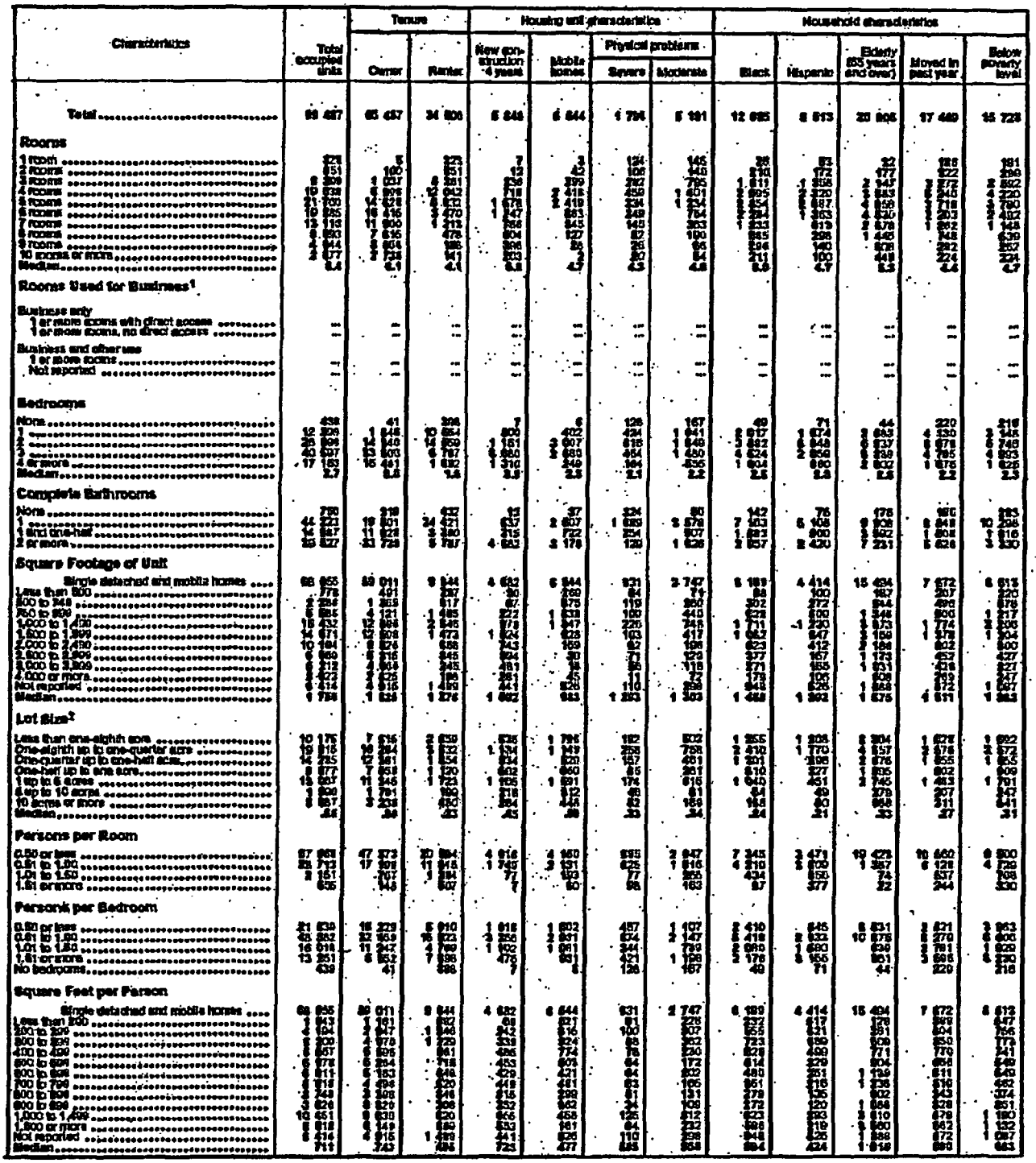
Table E.4 Farm Business Balance Sheet

($1,000)

Obs	year	ASSETSEX Farm business assets	REALASET Farm business real estate assets	DEBTEX Farm bus. debt	LIVPLTRY Livestock & poultry	MACHEX Machinery & equip.	CROPVALU Crops stored	PINPUTS Purchased inputs	COOPS Investments in cooperatives	FINANCEX Other financial assets
1	1960	174,350,621	123,280,588	22,445,668	15,607,597	19,068,447	6,359,991	.	4,205,998	5,828,000
2	1961	181,532,689	129,098,372	24,138,318	16,428,028	19,270,292	6,471,967	.	4,506,003	5,857,999
3	1962	188,869,521	134,813,549	26,665,409	17,308,979	19,202,285	6,527,710	.	4,845,999	5,869,999
4	1963	196,894,882	142,353,799	29,560,943	18,854,578	20,372,604	7,408,814	.	4,966,088	5,730,001
5	1964	204,238,278	150,485,891	32,157,241	14,488,308	21,247,171	7,012,635	.	5,177,277	5,848,999
6	1965	220,816,733	161,523,759	35,810,786	17,584,671	22,428,838	7,862,957	.	5,422,417	5,984,000
7	1966	234,023,352	171,233,796	39,223,065	18,973,530	24,067,400	8,107,008	.	5,677,818	5,964,000
8	1967	248,072,288	180,942,162	42,208,120	18,845,529	25,818,089	8,016,152	.	5,848,328	6,119,001
9	1968	257,161,488	189,389,629	43,921,577	20,249,058	27,744,429	7,393,262	.	6,101,129	6,284,000
10	1969	267,822,833	195,309,488	46,420,828	22,838,694	28,634,424	8,278,835	.	6,384,658	6,378,999
11	1970	278,823,017	202,418,084	48,752,694	23,708,781	30,383,932	8,700,574	.	7,133,666	6,501,001
12	1971	301,769,579	217,562,391	53,238,129	27,265,869	32,434,437	9,950,744	.	7,898,149	6,651,000
13	1972	335,833,243	243,002,154	58,704,291	33,879,918	34,844,363	12,949,001	.	8,718,418	6,941,998
14	1973	418,493,944	298,300,204	67,632,068	42,370,002	39,679,899	21,379,362	.	9,660,479	7,122,998
15	1974	449,181,774	338,556,231	75,852,878	24,567,478	48,454,284	22,815,224	.	11,217,558	8,871,002
16	1975	510,788,251	383,559,733	85,012,889	29,356,896	57,417,019	20,538,168	.	12,973,792	8,943,858
17	1976	590,718,844	458,539,437	96,065,408	28,047,508	63,275,288	20,584,879	.	14,322,409	8,947,129
18	1977	661,508,184	509,308,292	110,855,215	31,991,218	69,319,937	20,443,929	.	13,469,867	7,034,941
19	1978	777,198,168	601,773,359	127,400,008	50,106,788	78,292,678	23,800,887	.	16,069,753	7,136,911
20	1979	913,593,178	706,063,788	151,550,508	61,388,773	90,835,257	29,878,331	.	18,147,108	7,283,924
21	1980	999,022,607	782,820,208	166,823,858	60,633,208	96,063,978	32,831,043	.	19,316,744	7,357,433
22	1981	996,114,107	786,561,827	182,381,057	53,644,278	99,338,248	29,511,813	.	20,598,168	7,559,781
23	1982	962,488,412	750,023,252	185,806,367	52,993,051	105,853,544	25,885,600	.	21,946,968	7,788,000
24	1983	959,300,463	753,422,021	191,079,494	49,540,878	101,707,295	23,746,349	.	22,808,533	8,075,598
25	1984	987,815,227	661,750,218	193,787,208	49,525,822	95,851,857	28,128,958	1,977,709	24,259,572	8,323,997
26	1985	775,907,929	586,183,377	177,598,688	46,258,169	86,078,967	22,904,847	1,209,905	24,291,388	8,982,469
27	1986	722,017,255	542,407,974	158,970,418	47,751,409	79,036,189	18,279,821	2,067,461	24,430,692	10,049,715
28	1987	758,478,659	583,678,553	144,410,564	57,976,580	76,888,234	17,828,174	3,152,588	25,296,890	9,860,638
29	1988	788,528,981	582,311,234	139,568,639	62,211,338	80,978,413	23,651,022	3,459,087	25,564,584	10,352,308
30	1989	813,732,039	600,132,429	137,659,113	66,212,752	84,091,237	23,936,479	2,574,449	26,342,309	10,442,403
31	1990	840,606,579	618,149,243	137,961,618	70,856,067	86,287,864	23,178,199	2,807,525	27,455,238	10,864,448
32	1991	844,164,925	624,769,063	139,217,509	68,119,760	85,889,842	22,238,468	2,610,588	28,721,136	11,818,960
33	1992	868,139,358	640,790,777	139,052,269	70,962,162	88,182,099	24,208,624	3,948,492	29,428,892	13,629,313
34	1993	908,837,323	677,577,899	141,963,720	72,778,679	96,059,155	23,325,867	3,789,677	31,009,334	18,298,781
35	1994	938,765,812	704,137,647	146,799,869	67,912,189	97,804,032	23,297,799	5,036,668	32,071,823	15,505,460
36	1995	967,222,127	740,491,454	150,769,149	57,778,086	99,073,363	27,445,609	3,371,812	34,066,991	14,975,015
37	1996	1,004,540,797	769,535,894	156,073,748	60,314,851	99,601,769	31,992,878	4,361,891	34,916,938	14,118,552
38	1997	1,053,007,588	808,223,876	165,413,493	67,078,193	99,437,004	32,570,319	4,900,372	38,746,509	13,950,319
39	1998	1,085,297,900	840,421,559	172,862,459	63,391,480	91,729,424	29,939,743	5,036,407	40,515,352	14,263,937
40	1999	1,140,783,819	886,404,774	178,431,314	73,177,382	92,329,465	28,273,037	4,025,651	41,937,687	14,635,814
41	2000	1,188,260,080	928,453,518	183,977,584	76,825,534	92,038,518	27,932,429	4,895,713	42,966,068	14,159,312

Source: United States Department of Agriculture
Economic Research Service
Ken Erickson

Table E.5 Lot Size Example



Source: United States Department of Commerce, U.S. Census Bureau
American Housing Survey for the United States in 1997

Appendix F Population Tables

This appendix contains the population tables described in Subsection 5.4.2 External Verification. There are three tables for each licensee. The first table is the SECPOP2000 estimated population. The second table is the licensee's reported population. And the third table is a comparison of the SECPOP2000 estimated population as a percent of the licensee reported population.

Browns Ferry 11 Rings to 60 Miles

SECPOP2000 POPULATION

Ring:	SPATIAL DISTANCES 1	2	3	4	5	10	20	30	40	50	60	Total	
N	0	24	0	37	129	1,380	2,380	3,494	14,011	5,618	14,731	41,804	N
NNE	0	3	27	342	198	1,632	5,303	3,480	4,153	4,856	18,620	38,614	NNE
NE	0	0	0	7	189	5,015	16,521	11,230	9,254	14,323	5,790	62,329	NE
ENE	0	0	160	90	234	1,486	15,285	27,645	30,881	7,951	4,933	88,665	ENE
E	0	23	75	20	63	1,590	6,064	94,203	91,506	8,584	8,636	210,758	E
ESE	0	0	57	0	3	116	1,082	4,257	17,646	13,218	29,646	66,023	ESE
SE	0	0	0	0	0	7,578	33,950	9,532	8,513	22,700	16,116	100,389	SE
SSE	0	0	0	0	0	4,729	19,810	15,485	25,036	25,301	15,707	106,028	SSE
S	0	0	58	0	6	1,719	6,525	2,659	6,113	8,992	20,562	46,634	S
SSW	0	0	0	118	123	1,772	7,689	1,370	2,822	4,727	12,553	31,174	SSW
SW	0	0	0	81	25	1,058	4,915	2,718	3,649	14,649	5,640	32,735	SW
WSW	0	0	11	44	30	157	3,491	3,968	16,625	4,325	4,206	32,877	WSW
W	0	26	46	35	0	94	3,082	10,929	33,651	4,249	3,123	55,235	W
WNW	0	0	0	11	0	71	3,086	16,709	45,896	7,176	2,508	75,457	WNW
NW	0	0	0	0	21	572	5,704	7,459	5,777	4,483	6,163	30,179	NW
NNW	0	107	263	14	83	1,245	2,791	3,147	17,431	10,329	5,246	40,656	NNW
Totals:	0	183	697	799	1,124	30,214	139,678	218,245	332,958	161,481	174,180	1,059,559	
Density	0	15	31	33	36	103	137	138	144	113	94	People/Sq Mile	

Licensee's Reported Population

N	40	0	0	85	137	1,382	2,981	3,196	11,933	4,249	13,976	37,979	N
NNE	0	13	23	160	56	1,565	2,846	4,621	3,882	4,441	13,730	31,337	NNE
NE	0	3	23	45	127	4,708	13,791	7,681	7,861	14,539	4,393	53,171	NE
ENE	0	40	89	105	45	1,807	9,533	32,224	18,816	8,182	5,404	76,245	ENE
E	0	0	23	7	33	1,638	16,390	50,939	93,693	6,788	9,386	178,897	E
ESE	0	0	13	0	0	127	2,549	5,323	27,594	11,760	26,153	73,519	ESE
SE	0	0	0	0	0	8,777	22,981	7,642	8,618	22,346	13,553	83,917	SE
SSE	0	0	0	0	0	2,584	31,794	13,802	22,132	20,718	12,446	103,476	SSE
S	0	0	10	12	45	1,520	4,762	2,610	6,332	5,658	18,430	39,379	S
SSW	0	0	32	39	184	1,315	6,226	1,143	2,596	4,756	11,180	27,471	SSW
SW	0	0	0	15	108	828	5,590	2,414	3,454	13,338	4,578	30,325	SW
WSW	0	0	10	19	71	238	3,383	2,849	13,329	4,720	4,584	29,203	WSW
W	0	82	15	8	17	76	3,687	10,801	29,581	5,813	2,547	52,627	W
WNW	0	0	4	8	26	37	1,960	7,951	49,491	5,368	3,570	68,415	WNW
NW	0	0	78	9	62	853	4,158	7,301	6,275	4,530	5,394	28,660	NW
NNW	0	196	189	65	82	1,426	2,357	3,051	16,033	10,460	4,209	38,068	NNW
Totals:	40	334	509	577	993	28,881	134,988	163,548	321,620	147,666	153,533	932,689	
Density	13	30	31	29	31	100	132	117	130	102	84	People/Sq Mile	

SECPOP2000 as Percent of Reported

N	*	**		43.5%	94.2%	99.9%	79.8%	109.3%	117.4%	132.2%	105.4%	110.1%	N
NNE		23.1%	117.4%	213.8%	353.6%	104.3%	186.3%	75.3%	107.0%	109.3%	135.6%	123.2%	NNE
NE		*	*	15.6%	148.8%	106.5%	119.8%	146.2%	117.7%	98.5%	131.8%	117.2%	NE
ENE		*	179.8%	85.7%	520.0%	82.2%	160.3%	85.8%	164.1%	97.2%	91.3%	116.3%	ENE
E		**	326.1%	285.7%	190.9%	97.1%	37.0%	184.9%	97.7%	126.5%	92.0%	117.8%	E
ESE			438.5%		**	91.3%	42.4%	80.0%	63.9%	112.4%	113.4%	89.8%	ESE
SE						86.3%	156.4%	124.7%	98.8%	101.6%	118.9%	119.6%	SE
SSE						183.0%	62.3%	111.9%	113.1%	122.1%	126.2%	102.5%	SSE
S			580.0%	*	13.3%	113.1%	137.0%	101.9%	96.5%	158.9%	111.6%	118.4%	S
SSW			*	302.6%	66.8%	134.8%	123.5%	119.9%	108.7%	99.4%	112.3%	113.5%	SSW
SW				340.0%	23.1%	127.8%	87.9%	112.6%	105.6%	109.8%	123.2%	107.9%	SW
WSW			110.0%	231.6%	70.4%	66.0%	103.2%	139.3%	124.7%	91.6%	91.8%	112.6%	WSW
W		31.7%	306.7%	437.5%	*	123.7%	83.6%	101.2%	113.8%	73.1%	122.6%	105.9%	W
WNW			*	137.5%	*	191.9%	157.4%	210.1%	92.7%	133.7%	70.3%	110.3%	WNW
NW			*	*	33.9%	67.1%	137.2%	102.2%	92.1%	99.0%	114.3%	105.3%	NW
NNW		54.6%	139.2%	21.5%	101.2%	87.3%	118.4%	103.1%	108.7%	98.7%	124.6%	106.8%	NNW
	*	34.8%	136.9%	138.5%	113.2%	104.6%	103.3%	133.4%	103.3%	109.4%	113.4%	111.2%	

* The licensee reported people in the ring segment but SECPOP90 did not locate anyone in the ring segment.
** SECPOP90 located people in this ring segment but the licensee did not report anyone in the ring segment.
Licensee data from FSAR, Tables 2.1-4 and 2.1-5. From 2000 Census data.

Run Date: 07/10/02

Columbia 7 Rings to 50 Miles (WNP2)

SECPOP2000 POPULATION

SPATIAL DISTANCES

Ring:	3	5	10	20	30	40	50	Total	
N	0	0	103	126	848	1,044	30,189	32,310	N
NNE	0	0	44	805	10,941	3,702	679	16,171	NNE
NE	0	11	254	1,456	476	236	734	3,167	NE
ENE	4	38	360	916	3,133	370	186	5,007	ENE
E	0	0	262	541	72	134	103	1,112	E
ESE	0	0	517	511	248	1,001	206	2,483	ESE
SE	0	0	383	10,017	12,282	410	1,291	24,383	SE
SSE	0	0	174	63,802	42,898	265	222	107,361	SSE
S	0	0	203	33,479	881	6,495	25,288	66,346	S
SSW	0	0	498	6,392	149	191	6,275	13,505	SSW
SW	0	0	0	1,656	9,115	1,166	196	12,133	SW
WSW	0	0	0	24	1,893	36,252	3,646	41,815	WSW
W	0	0	0	0	92	1,204	20,963	22,259	W
WNW	0	0	0	0	190	2,757	0	2,947	WNW
NW	0	0	0	0	562	2,869	550	3,981	NW
NNW	0	0	0	0	478	3,257	1,819	5,554	NNW
Totals:	4	49	2,798	119,725	84,258	61,353	92,347	360,534	
Density	0	1	9	98	73	53	46	People/Sq Mile	

Licensee's Reported Population

N	0	0	77	398	2,055	1,127	21,572	25,229	N
NNE	0	0	152	397	7,123	3,983	1,121	12,776	NNE
NE	0	48	224	588	2,274	745	1,275	5,154	NE
ENE	0	56	177	855	1,786	475	375	3,724	ENE
E	0	56	257	544	220	141	268	1,486	E
ESE	0	56	341	576	305	182	961	2,421	ESE
SE	0	9	536	5,821	6,738	497	2,349	15,950	SE
SSE	0	0	308	70,917	36,360	955	2,072	110,612	SSE
S	0	0	483	45,434	975	6,368	17,708	70,968	S
SSW	0	0	809	1,922	1,426	529	2,972	7,658	SSW
SW	0	0	25	894	7,737	786	476	9,918	SW
WSW	0	0	0	1,108	1,908	26,890	965	30,871	WSW
W	0	0	0	0	1,429	4,273	22,176	27,878	W
WNW	0	0	0	0	297	1,579	2,043	3,919	WNW
NW	0	0	0	0	48	836	905	1,789	NW
NNW	0	0	0	0	218	1,899	642	2,759	NNW
Totals:	0	225	3,389	129,454	70,899	51,265	77,880	333,112	
Density	0	3	12	106	72	51	42	People/Sq Mile	

SECPOP2000 as Percent of Reported

N			133.8%	31.7%	41.3%	92.6%	139.9%	128.1%	N
NNE			28.9%	202.8%	153.6%	92.9%	60.6%	126.6%	NNE
NE		22.9%	113.4%	247.6%	20.9%	31.7%	57.6%	61.4%	NE
ENE	**	67.9%	203.4%	107.1%	175.4%	77.9%	49.6%	134.5%	ENE
E		*	101.9%	99.4%	32.7%	95.0%	38.4%	74.8%	E
ESE		*	151.6%	88.7%	81.3%	550.0%	21.4%	102.6%	ESE
SE		*	71.5%	172.1%	182.3%	82.5%	55.0%	152.9%	SE
SSE			56.5%	90.0%	118.0%	27.7%	10.7%	97.1%	SSE
S			42.0%	73.7%	90.4%	102.0%	142.8%	93.5%	S
SSW			61.6%	332.6%	10.4%	36.1%	211.1%	176.4%	SSW
SW				*	185.2%	117.8%	148.3%	41.2%	122.3%	SW
WSW					2.2%	99.2%	134.8%	377.8%	135.5%	WSW
W						6.4%	28.2%	94.5%	79.8%	W
WNW						64.0%	174.6%	*	75.2%	WNW
NW						1170.8%	343.2%	60.8%	222.5%	NW
NNW						219.3%	171.5%	283.3%	201.3%	NNW
	**	21.8%	82.6%	92.5%	118.8%	119.7%	118.6%	108.2%	

* The licensee reported people in the ring segment but SECPOP90 did not locate anyone in the ring segm Run Date: 07/10/02

Cook Nuclear Plant 11 Rings to 60 Miles

SPATIAL DISTANCES

Ring:	1	2	3	4	5	10	20	30	40	50	60	Total	
SECPOP2000 POPULATION													
N	0	0	0	0	0	0	0	0	0	0	0	0	N
NNE	22	0	98	178	1,226	11,720	10,316	2,553	11,851	10,268	41,227	89,459	NNE
NE	6	4	275	1,877	2,706	14,915	24,083	18,142	11,426	14,447	31,688	119,569	NE
ENE	43	122	428	616	198	3,840	4,420	6,641	19,116	34,666	188,051	258,141	ENE
E	0	32	79	170	256	1,104	12,253	14,192	7,705	20,766	19,223	75,780	E
ESE	0	65	258	413	1,095	1,508	17,363	18,872	63,053	32,929	21,305	156,861	ESE
SE	0	131	289	105	154	599	13,710	184,606	77,496	58,905	22,142	358,137	SE
SSE	11	149	1,123	450	289	820	3,457	26,247	13,646	27,013	11,491	84,696	SSE
S	14	126	595	428	579	1,533	6,979	6,786	8,719	14,735	11,215	51,709	S
SSW	0	2	337	20	0	2,311	4,531	39,499	12,102	8,486	11,823	79,111	SSW
SW	0	0	0	0	0	0	4,560	42,141	30,930	180,976	149,979	408,586	SW
WSW	0	0	0	0	0	0	0	0	0	26,366	1,120,385	1,146,751	WSW
W	0	0	0	0	0	0	0	0	0	0	1,423,775	1,423,775	W
WNW	0	0	0	0	0	0	0	0	0	0	0	0	WNW
NW	0	0	0	0	0	0	0	0	0	0	0	0	NW
NNW	0	0	0	0	0	0	0	0	0	0	0	0	NNW
Totals:	96	631	3,482	4,257	6,503	38,350	101,672	359,679	256,044	429,557	3,052,304	4,252,575	
Density	31	58	149	168	191	170	123	182	153	153	376	People/Sq Mile	
Licensee's Reported Population													
N	3	0	0	0	0	0	0	0	0	0	0	3	N
NNE	15	11	58	452	1,003	14,873	17,912	3,829	14,574	9,981	37,198	99,906	NNE
NE	12	25	340	2,017	2,518	13,770	22,769	18,275	12,351	13,580	34,091	119,748	NE
ENE	9	111	184	264	356	3,053	4,699	6,399	19,857	28,604	179,627	243,163	ENE
E	0	88	121	156	181	1,461	13,241	15,985	8,053	24,230	19,133	82,649	E
ESE	0	75	160	441	871	1,391	19,654	20,340	76,005	34,875	18,568	172,380	ESE
SE	3	107	260	160	193	703	12,280	184,094	58,265	48,499	19,032	323,596	SE
SSE	5	171	897	297	235	922	3,254	18,563	12,349	26,036	11,463	74,192	SSE
S	8	88	978	282	238	1,359	7,396	6,320	8,829	13,869	12,762	52,129	S
SSW	4	34	108	62	31	2,254	4,634	37,403	11,439	10,056	9,566	75,591	SSW
SW	0	0	0	0	0	0	4,425	43,353	26,184	188,025	156,452	418,439	SW
WSW	0	0	0	0	0	0	0	0	0	17,125	1,210,464	1,227,589	WSW
W	0	0	0	0	0	0	0	0	0	0	1,317,989	1,317,989	W
WNW	0	0	0	0	0	0	0	0	0	0	0	0	WNW
NW	0	0	0	0	0	0	0	0	0	0	0	0	NW
NNW	0	0	0	0	0	0	0	0	0	0	0	0	NNW
Totals:	59	710	3,106	4,131	5,626	39,786	110,264	354,561	247,906	414,880	3,026,345	4,207,374	
Density	19	61	137	159	174	170	130	183	152	150	372	People/Sq Mile	
SECPOP2000 as Percent of Reported													
N	*											*	N
NNE	146.7%	*	169.0%	39.4%	122.2%	78.8%	57.6%	66.7%	81.3%	102.9%	110.8%	89.5%	NNE
NE	50.0%	16.0%	80.9%	93.1%	107.5%	108.3%	105.8%	99.3%	92.5%	106.4%	93.0%	99.9%	NE
ENE	477.8%	109.9%	232.6%	233.3%	55.6%	125.8%	94.1%	103.8%	96.3%	121.2%	104.7%	106.2%	ENE
E		36.4%	65.3%	109.0%	141.4%	75.6%	92.5%	88.8%	95.7%	85.7%	100.5%	91.7%	E
ESE		86.7%	161.3%	93.7%	125.7%	108.4%	88.3%	92.8%	83.0%	94.4%	114.7%	91.0%	ESE
SE	*	122.4%	111.2%	65.6%	79.8%	85.2%	111.6%	100.3%	133.0%	121.5%	116.3%	110.7%	SE
SSE	220.0%	87.1%	125.2%	151.5%	123.0%	88.9%	106.2%	141.4%	110.5%	103.8%	100.2%	114.2%	SSE
S	175.0%	143.2%	60.8%	151.8%	243.3%	112.8%	94.4%	107.4%	98.8%	106.2%	87.9%	99.2%	S
SSW	*	5.9%	312.0%	32.3%	*	102.5%	97.8%	105.6%	105.8%	84.4%	123.6%	104.7%	SSW
SW							103.1%	97.2%	118.1%	96.3%	95.9%	97.6%	SW
WSW										154.0%	92.6%	93.4%	WSW
W											108.0%	108.0%	W
WNW													WNW
NW													NW
NNW													NNW
	162.7%	88.9%	112.1%	103.1%	115.6%	96.4%	92.2%	101.4%	103.3%	103.5%	100.9%	101.1%	

* The licensee reported people in the ring segment but SECPOP90 did not locate anyone in the ring segment.
** SECPOP90 located people in the ring segment but the licensee did not report anyone in the ring segment.
Licensee data from FSAR, Revision 16.1, Table 2.1-7. ESTIMATED.

Run Date: 07/10/02

Grand Gulf 10 Rings to 50 Miles

SPATIAL DISTANCES

SECPOP2000 POPULATION

Ring:	1	2	3	4	5	10	20	30	40	50	Total	
N	0	14	0	0	0	10	171	578	786	1,184	2,743	N
NNE	0	0	0	0	29	3	5,195	32,449	3,433	723	41,832	NNE
NE	0	14	0	0	27	56	1,801	4,859	4,897	3,019	14,673	NE
ENE	0	17	0	64	0	123	244	3,253	4,626	66,151	74,478	ENE
E	0	0	0	113	56	102	567	1,531	4,018	14,053	20,440	E
ESE	0	10	0	0	336	840	981	277	5,726	8,482	16,652	ESE
SE	0	0	0	10	776	2,582	1,177	637	2,720	21,634	29,536	SE
SSE	0	14	5	37	122	571	1,189	579	3,261	2,536	8,314	SSE
S	0	0	0	0	0	34	2,186	4,740	1,970	1,330	10,260	S
SSW	0	0	4	0	22	1,357	1,191	1,804	25,607	4,413	34,398	SSW
SW	0	0	0	0	0	74	51	1,223	16,912	3,665	21,925	SW
WSW	0	0	0	0	0	56	1,671	342	1,863	2,095	6,027	WSW
W	0	0	0	0	0	123	346	321	4,267	2,694	7,751	W
WNW	0	0	0	0	0	6	2,386	97	4,527	11,244	18,260	WNW
NW	0	0	0	0	0	28	130	62	2,364	5,989	8,573	NW
NNW	0	0	0	0	0	0	95	11,204	667	2,621	14,587	NNW
Totals:	0	69	9	224	1,368	5,965	19,381	63,956	87,644	151,833	330,449	
Density	0	5	3	6	21	24	21	32	36	42	People/Sq Mile	

Licensee's Reported Population

N	4	63	12	0	0	0	279	1,538	1,033	1,534	4,463	N
NNE	12	12	0	0	0	141	3,039	39,796	3,431	1,141	47,572	NNE
NE	4	4	4	12	71	344	2,207	6,538	5,032	4,850	19,066	NE
ENE	0	0	20	8	32	325	356	4,540	5,452	12,839	23,572	ENE
E	4	24	44	51	51	299	717	2,117	4,393	12,347	20,047	E
ESE	8	8	31	0	208	253	916	691	5,506	8,358	15,979	ESE
SE	0	4	8	44	1,334	1,827	840	807	2,029	17,967	24,860	SE
SSE	3	8	0	35	90	631	1,369	729	2,214	2,304	7,383	SSE
S	0	4	0	4	0	337	1,345	3,264	1,758	1,688	8,400	S
SSW	0	4	4	0	0	494	1,237	2,229	38,750	4,220	46,938	SSW
SW	4	0	0	0	0	317	917	2,881	21,535	5,665	31,319	SW
WSW	0	0	0	0	0	138	2,077	1,003	2,689	3,125	9,032	WSW
W	0	0	0	0	0	246	493	642	6,711	4,321	12,413	W
WNW	0	0	0	0	5	88	2,373	729	7,877	9,314	20,386	WNW
NW	8	0	0	0	0	171	478	233	3,030	8,377	12,297	NW
NNW	8	20	0	0	0	85	164	12,046	1,122	4,147	17,592	NNW
Totals:	55	151	123	154	1,791	5,696	18,807	79,783	112,562	102,197	321,319	
Density	18	16	12	10	29	25	21	38	44	41	People/Sq Mile	

SECPOP2000 as Percent of Reported

N	*	22.2%	*			**	61.3%	37.6%	76.1%	77.2%	61.5%	N
NNE	*	*			**	2.1%	170.9%	81.5%	100.1%	63.4%	87.9%	NNE
NE	*	350.0%	*	*	38.0%	16.3%	81.6%	74.3%	97.3%	62.2%	77.0%	NE
ENE		**	*	800.0%	*	37.8%	68.5%	71.7%	84.8%	515.2%	316.0%	ENE
E	*	*	*	221.6%	109.8%	34.1%	79.1%	72.3%	91.5%	113.8%	102.0%	E
ESE	*	125.0%	*		161.5%	332.0%	107.1%	40.1%	104.0%	101.5%	104.2%	ESE
SE		*	*	22.7%	58.2%	141.3%	140.1%	78.9%	134.1%	120.4%	118.8%	SE
SSE	*	175.0%	**	105.7%	135.6%	90.5%	86.9%	79.4%	147.3%	110.1%	112.6%	SSE
S		*		*		10.1%	162.5%	145.2%	112.1%	78.8%	122.1%	S
SSW		*	100.0%		**	274.7%	96.3%	80.9%	66.1%	104.6%	73.3%	SSW
SW	*					23.3%	5.6%	42.5%	78.5%	64.7%	70.0%	SW
WSW						40.6%	80.5%	34.1%	69.3%	67.0%	66.7%	WSW
W						50.0%	70.2%	50.0%	63.6%	62.3%	62.4%	W
WNW					*	6.8%	100.5%	13.3%	57.5%	120.7%	89.6%	WNW
NW	*					16.4%	27.2%	26.6%	78.0%	71.5%	69.7%	NW
NNW	*	*				*	57.9%	93.0%	59.4%	63.2%	82.9%	NNW
	*	45.7%	7.3%	143.5%	76.4%	104.7%	103.1%	80.2%	77.9%	148.6%	102.8%	

* The licensee reported people in the ring segment but SECPOP90 did not locate anyone in the ring segment.

** SECPOP90 located people in the ring segment but the licensee did not report anyone in the ring segment.

Licensee data from FSAR, Tables 2.1-1 and 2.1-3. ESTIMATED from 1970 Census data.

Run Date: 07/11/02

Limerick 10 Rings to 50 Miles

SPATIAL DISTANCES

SECPOP2000 POPULATION

Ring:	1	2	3	4	5	10	20	30	40	50	Total	
N	0	929	3,762	1,084	2,098	7,089	16,360	94,948	67,678	38,768	232,716	N
NNE	0	778	1,216	202	776	3,098	21,170	187,371	183,761	53,324	451,696	NNE
NE	0	39	93	482	356	5,292	32,235	21,947	34,805	60,340	155,589	NE
ENE	17	108	1,981	1,390	526	9,049	70,035	64,333	27,402	55,569	230,410	ENE
E	0	532	279	1,447	904	10,999	99,782	197,718	233,494	349,431	854,586	E
ESE	139	251	1,127	2,574	1,452	19,453	133,950	685,042	576,972	184,600	1,605,560	ESE
SE	25	254	512	6,644	3,791	12,577	103,458	872,952	449,859	286,513	1,736,585	SE
SSE	8	154	595	1,662	859	21,029	47,515	239,831	36,914	24,887	373,454	SSE
S	0	414	298	406	203	4,352	89,249	86,026	309,142	75,960	560,050	S
SSW	71	349	262	560	962	2,612	57,125	30,466	83,129	67,248	242,783	SSW
SW	50	342	192	433	529	2,426	27,752	41,046	26,590	26,267	125,627	SW
WSW	0	489	291	901	689	2,523	11,863	27,476	110,624	131,104	285,960	WSW
W	121	662	704	581	843	2,439	9,211	37,202	69,010	81,386	202,159	W
WNW	24	54	635	6,304	2,341	10,756	109,856	103,392	26,462	43,509	303,333	WNW
NW	0	241	4,030	9,289	2,543	3,942	14,755	33,941	18,803	61,333	148,877	NW
NNW	0	817	1,306	2,078	1,691	13,832	11,016	22,336	9,087	31,735	93,898	NNW
Totals:	455	6,412	17,283	36,037	20,363	131,468	835,332	2,700,027	2,263,732	1,371,974	7,603,283	
Density	145	546	854	1,197	1,028	676	850	1,333	1,200	968	People/Sq Mile	

Licensee's Reported Population

N	64	756	990	440	834	3,499	16,837	58,743	62,871	29,786	174,820	N
NNE	51	1,205	270	529	225	2,690	27,473	205,566	198,282	44,225	480,516	NNE
NE	51	45	223	370	306	4,134	21,141	23,177	30,320	59,686	139,453	NE
ENE	13	84	220	421	252	5,691	58,184	61,422	24,904	47,162	198,333	ENE
E	22	166	300	431	463	5,672	76,172	102,127	209,526	389,831	784,710	E
ESE	32	198	329	297	641	10,213	144,573	542,450	572,224	137,627	1,408,584	ESE
SE	6	408	156	5,365	4,491	7,566	89,099	844,309	550,741	224,521	1,726,662	SE
SSE	0	224	336	3,141	1,871	24,749	33,947	256,615	39,309	25,600	385,792	SSE
S	3	404	390	194	401	4,557	85,945	42,015	368,752	26,026	528,687	S
SSW	14	720	363	604	316	2,179	47,295	30,036	54,130	55,746	191,403	SSW
SW	81	214	241	367	353	2,102	9,915	40,738	13,566	21,947	89,524	SW
WSW	54	211	628	540	1,881	2,239	9,931	28,797	85,157	155,924	285,362	WSW
W	41	139	2,068	1,786	1,243	2,640	4,564	20,260	58,108	87,041	177,890	W
WNW	45	354	3,314	12,268	3,927	10,844	139,379	82,329	29,560	33,205	315,225	WNW
NW	22	319	2,073	7,384	1,450	4,435	5,078	19,391	19,088	70,460	129,700	NW
NNW	38	788	1,913	1,370	1,444	7,261	14,581	18,059	8,347	39,276	93,077	NNW
Totals:	537	6,213	13,814	35,507	20,098	100,471	784,114	2,376,034	2,324,883	1,448,063	7,109,738	
Density	171	537	727	1,116	970	562	765	1,180	1,126	905	People/Sq Mile	

SECPOP2000 as Percent of Reported

N	*	122.9%	380.0%	246.4%	251.6%	202.6%	97.2%	161.6%	107.6%	130.2%	133.1%	N
NNE	*	64.6%	450.4%	38.2%	344.9%	115.2%	77.1%	91.1%	92.7%	120.6%	94.0%	NNE
NE	*	86.7%	41.7%	130.3%	116.3%	128.0%	152.5%	94.7%	114.8%	101.1%	111.6%	NE
ENE	130.8%	168.8%	900.5%	330.2%	208.7%	159.0%	120.4%	104.7%	110.0%	117.8%	116.2%	ENE
E	*	320.5%	93.0%	335.7%	195.2%	193.9%	131.0%	154.4%	111.4%	89.6%	108.9%	E
ESE	434.4%	126.8%	342.6%	866.7%	226.5%	190.5%	92.7%	126.3%	100.8%	134.1%	114.0%	ESE
SE	416.7%	62.3%	328.2%	123.8%	84.3%	166.2%	116.1%	103.4%	81.7%	127.6%	100.6%	SE
SSE	**	68.8%	177.1%	52.9%	45.9%	85.0%	140.0%	93.5%	93.9%	97.2%	96.8%	SSE
S	*	102.5%	76.4%	209.3%	50.6%	95.5%	103.8%	190.5%	83.8%	291.9%	105.9%	S
SSW	507.1%	48.3%	72.2%	92.7%	304.4%	119.9%	120.8%	101.4%	153.6%	120.6%	126.8%	SSW
SW	61.7%	159.8%	79.7%	118.0%	149.9%	115.4%	279.9%	100.8%	196.0%	119.7%	140.3%	SW
WSW	*	231.8%	46.3%	166.9%	36.6%	112.7%	119.5%	95.4%	129.9%	84.1%	100.2%	WSW
W	295.1%	476.3%	34.0%	32.5%	67.8%	92.4%	201.8%	183.6%	118.8%	93.5%	113.6%	W
WNW	53.3%	15.3%	19.2%	51.4%	59.6%	99.2%	78.8%	125.6%	89.5%	131.0%	96.2%	WNW
NW	*	75.5%	194.4%	125.8%	175.4%	88.9%	290.6%	175.0%	98.5%	87.0%	114.8%	NW
NNW	*	103.7%	68.3%	151.7%	117.1%	190.5%	75.6%	123.7%	108.9%	80.8%	100.9%	NNW
	84.7%	103.2%	125.1%	101.5%	102.3%	130.9%	109.1%	113.6%	97.4%	108.6%	106.9%	

* The licensee reported people in the ring segment but SECPOP90 did not locate anyone in the ring segment.

** SECPOP90 located people in the ring segment but the licensee did not report anyone in the ring segment.

Licensee data from PSAR, Tables 2.1-3 and 2.1-12. ESTIMATED from 1970 Census data.

Run Date: 07/11/02

Point Beach 9 Rings to 40 Miles

SPATIAL DISTANCES

SECPOP2000 POPULATION

Ring:	1	2	3	4	5	10	20	30	40	Total	
N	0	0	0	2	18	391	4,382	4,612	6,304	15,709	N
NNE	0	0	0	0	0	0	0	2,554	7,948	10,502	NNE
NE	0	0	0	0	0	0	0	0	0	0	NE
ENE	0	0	0	0	0	0	0	0	0	0	ENE
E	0	0	0	0	0	0	0	0	0	0	E
ESE	0	0	0	0	0	0	0	0	0	0	ESE
SE	0	0	0	0	0	0	0	0	0	0	SE
SSE	0	24	49	14	0	14	0	0	0	101	SSE
S	0	36	49	28	212	4,178	0	0	0	4,503	S
SSW	0	0	46	19	39	8,907	33,057	3,578	61,842	107,488	SSW
SW	0	16	33	16	56	1,806	7,821	5,153	14,074	28,975	SW
WSW	0	25	0	51	74	1,624	3,867	6,559	9,902	22,102	WSW
W	0	25	0	32	176	568	2,470	3,986	48,257	55,514	W
WNW	0	44	0	53	59	422	4,299	80,277	51,711	136,865	WNW
NW	0	25	40	64	82	487	2,667	51,797	32,642	87,804	NW
NNW	2	0	71	27	18	437	4,349	4,696	701	10,301	NNW
Totals:	2	195	288	306	734	18,834	62,912	163,212	233,381	479,864	
Density	1	16	17	16	19	65	66	87	95	People/Sq Mile	

Licensee's Reported Population

N	0	0	4	19	4	379	5,195	7,651	8,586	21,838	N
NNE	0	0	0	0	0	0	0	282	6,513	6,795	NNE
NE	0	0	0	0	0	0	0	0	0	0	NE
ENE	0	0	0	0	0	0	0	0	0	0	ENE
E	0	0	0	0	0	0	0	0	0	0	E
ESE	0	0	0	0	0	0	0	0	0	0	ESE
SE	0	0	0	0	0	0	0	0	0	0	SE
SSE	0	30	111	140	2	0	0	0	0	283	SSE
S	0	12	36	40	35	8,951	0	0	0	9,074	S
SSW	4	16	20	47	48	10,622	28,767	3,625	53,433	96,582	SSW
SW	0	8	24	28	64	1,843	10,003	5,513	14,550	32,033	SW
WSW	0	32	23	29	58	1,840	3,477	7,765	11,718	24,942	WSW
W	0	24	27	44	55	728	2,701	4,651	39,072	47,302	W
WNW	4	28	23	29	51	457	4,146	58,424	39,266	102,428	WNW
NW	0	20	75	15	44	529	2,871	79,131	26,297	108,982	NW
NNW	0	20	75	27	41	513	3,817	4,332	787	9,612	NNW
Totals:	8	190	418	418	402	25,862	60,977	171,374	200,222	459,871	
Density	3	16	22	21	18	87	70	92	91	People/Sq Mile	

SECPOP2000 as Percent of Reported

N				*	10.5%	450.0%	103.2%	84.4%	60.3%	73.4%	71.9%	N
NNE									905.7%	122.0%	154.6%	NNE
NE												NE
ENE												ENE
E												E
ESE												ESE
SE												SE
SSE		80.0%	44.1%	10.0%	*	**					35.7%	SSE
S		300.0%	136.1%	70.0%	605.7%	46.7%					49.6%	S
SSW	*	*	230.0%	40.4%	81.3%	83.9%	114.9%	98.7%	115.7%		111.3%	SSW
SW		200.0%	137.5%	57.1%	87.5%	98.0%	78.2%	93.5%	96.7%		90.5%	SW
WSW		78.1%	*	175.9%	127.6%	88.3%	111.2%	84.5%	84.5%		88.6%	WSW
W		104.2%	*	72.7%	320.0%	78.0%	91.4%	85.7%	123.5%		117.4%	W
WNW	*	157.1%	*	182.8%	115.7%	92.3%	103.7%	137.4%	131.7%		133.6%	WNW
NW		125.0%	53.3%	426.7%	186.4%	92.1%	92.9%	65.5%	124.1%		80.6%	NW
NNW	**	*	94.7%	100.0%	43.9%	85.2%	113.9%	108.4%	89.1%		107.2%	NNW
	25.0%	102.6%	68.9%	73.2%	182.6%	72.8%	103.2%	95.2%	116.6%		104.3%	

* The licensee reported people in the ring segment but SECPOP90 did not locate anyone in the ring segment.

** SECPOP90 located people in the ring segment but the licensee did not report anyone in the ring segment.

Licensee data from FSAR. REvision 1. Figures 2.3-1 & 2.3-2. ESTIMATED from 1980 Census data.

Run Date: 07/15/02

River Bend 10 Rings to 50 Miles

SPATIAL DISTANCES

Ring:	1	2	3	4	5	10	20	30	40	50	Total	
SECPOP2000 POPULATION												
N	0	196	132	73	21	759	1,070	[illegible]	857	[illegible]	10,842	N
NNE	16	0	112	8	0	511	403	[illegible]	[illegible]	[illegible]	6,388	NNE
NE	0	0	0	2	22	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	13,232	NE
ENE	35	17	0	72	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	16,650	ENE
E	0	0	44	4	0	289	[illegible]	[illegible]	[illegible]	[illegible]	25,921	E
ESE	0	0	16	41	5	720	[illegible]	[illegible]	[illegible]	[illegible]	59,628	ESE
SE	0	4	0	0	9	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	287,718	SE
SSE	0	0	89	0	0	202	[illegible]	[illegible]	[illegible]	[illegible]	253,051	SSE
S	0	0	0	0	0	670	[illegible]	[illegible]	[illegible]	[illegible]	26,469	S
SSW	2	0	0	0	114	1,099	[illegible]	[illegible]	105	[illegible]	12,732	SSW
SW	0	0	0	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	43,206	SW
WSW	0	0	0	0	0	[illegible]	401	[illegible]	[illegible]	[illegible]	56,935	WSW
W	0	0	0	0	0	0	1,353	[illegible]	[illegible]	[illegible]	8,671	W
WNW	0	4	297	570	0	17	168	[illegible]	[illegible]	[illegible]	22,602	WNW
NW	0	19	250	655	825	847	438	[illegible]	221	891	9,689	NW
NNW	0	420	4	0	244	604	702	763	61	135	2,933	NNW
Totals:	53	660	944	1,425	1,240	20,311	92,219	365,017	165,460	209,338	856,667	
Density	17	57	59	61	55	78	93	170	129	109	People/Sq Mile	

Licensee's Reported Population												
N	0	109	103	37	444	622	871	[illegible]	1,328	[illegible]	10,662	N
NNE	15	291	34	48	7	280	638	[illegible]	[illegible]	[illegible]	7,193	NNE
NE	8	26	19	0	72	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	16,334	NE
ENE	0	36	19	25	76	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	16,552	ENE
E	6	14	0	26	17	306	[illegible]	[illegible]	[illegible]	[illegible]	30,701	E
ESE	3	0	23	96	23	902	[illegible]	[illegible]	[illegible]	[illegible]	87,472	ESE
SE	0	0	5	0	148	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	278,897	SE
SSE	0	0	3	0	0	492	[illegible]	[illegible]	[illegible]	[illegible]	338,039	SSE
S	0	5	0	0	0	1,199	[illegible]	[illegible]	[illegible]	[illegible]	33,156	S
SSW	0	0	0	9	333	1,207	[illegible]	[illegible]	[illegible]	[illegible]	22,279	SSW
SW	0	0	0	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	45,611	SW
WSW	0	0	0	3	3	[illegible]	1,784	[illegible]	[illegible]	[illegible]	58,740	WSW
W	0	0	2	0	6	0	1,719	[illegible]	[illegible]	[illegible]	14,495	W
WNW	0	0	214	918	27	27	777	[illegible]	[illegible]	[illegible]	23,907	WNW
NW	29	24	117	1,216	520	485	[illegible]	[illegible]	1,783	[illegible]	17,079	NW
NNW	19	164	103	40	36	549	1,000	1,040	1,690	[illegible]	7,499	NNW
Totals:	80	669	642	2,418	1,712	23,609	143,356	425,724	179,461	230,945	1,008,616	
Density	25	60	49	76	70	93	137	212	155	128	People/Sq Mile	

SECPOP2000 as Percent of Reported												
N		179.8%	128.2%	197.3%	4.7%	122.0%	122.8%	165.0%	64.5%	72.3%	101.7%	N
NNE	106.7%	*	329.4%	16.7%	*	182.5%	63.2%	108.7%	130.0%	46.3%	88.8%	NNE
NE	*	*	*	**	30.6%	78.4%	69.5%	82.0%	78.4%	97.6%	81.0%	NE
ENE	**	47.2%	*	288.0%	*	135.3%	95.8%	87.2%	73.4%	128.8%	100.6%	ENE
E	*	*	**	15.4%	*	94.4%	68.1%	40.9%	118.4%	100.0%	84.4%	E
ESE	*		69.6%	42.7%	21.7%	79.8%	40.4%	76.0%	85.5%	73.6%	68.2%	ESE
SE		**	*		6.1%	72.1%	76.9%	104.7%	112.9%	118.3%	103.2%	SE
SSE			2966.7%			41.1%	53.1%	77.1%	95.6%	61.8%	74.9%	SSE
S		*				55.9%	68.9%	79.6%	97.9%	44.0%	79.8%	S
SSW	**			*	34.2%	91.1%	116.2%	83.4%	3.1%	35.9%	57.1%	SSW
SW						61.1%	96.0%	49.4%	21.8%	127.6%	94.7%	SW
WSW				*	*	98.8%	22.5%	70.1%	55.2%	107.6%	96.9%	WSW
W			*		*		78.7%	53.7%	65.7%	55.4%	59.8%	W
WNW		**	138.8%	62.1%	*	63.0%	21.6%	103.4%	81.0%	104.8%	94.5%	WNW
NW	*	79.2%	213.7%	53.9%	158.7%	174.6%	19.1%	93.1%	12.4%	19.1%	56.7%	NW
NNW	*	256.1%	3.9%	*	677.8%	110.0%	70.2%	73.4%	3.6%	4.7%	39.1%	NNW
	66.3%	98.7%	147.0%	58.9%	72.4%	86.0%	64.3%	85.7%	92.2%	90.6%	84.9%	

* The licensee reported people in the ring segment but SECPOP90 did not locate anyone in the ring segment.

** SECPOP90 located people in the ring segment but the licensee did not report anyone in the ring segment.

Licensee data from FSAR, August 1987, Tables 2.1-6 and 2.1-13. ESTIMATED from 1970 Census data.

Run Date: 07/16/02

Robinson 10 Rings to 50 Miles

SECPOP2000 POPULATION

Ring:	SPATIAL DISTANCES 1	2	3	4	5	10	20	30	40	50	Total	
N	0	0	0	85	125	245	1,101	5,290	10,832	7,574	25,252	N
NNE	0	27	341	211	163	536	2,318	7,005	6,203	21,649	38,453	NNE
NE	0	30	195	161	66	829	1,289	10,439	7,595	23,046	43,650	NE
ENE	0	249	163	72	238	1,355	2,706	4,488	16,424	24,937	50,632	ENE
E	0	20	392	745	771	3,180	3,072	931	2,501	24,346	35,958	E
ESE	33	167	102	518	1,724	6,145	16,164	27,216	7,986	16,034	76,089	ESE
SE	0	45	133	746	1,754	3,601	4,614	49,884	14,108	8,229	83,114	SE
SSE	39	0	304	512	225	1,290	4,733	5,170	8,390	14,255	34,918	SSE
S	61	0	70	175	118	577	1,733	3,770	5,143	10,119	21,766	S
SSW	90	76	112	20	60	498	8,111	4,885	74,894	12,849	101,595	SSW
SW	121	173	171	137	116	356	2,332	5,219	8,312	9,378	26,315	SW
WSW	133	55	328	55	100	185	1,523	18,952	17,373	60,703	99,407	WSW
W	9	30	55	23	11	308	2,284	2,255	2,890	3,768	11,633	W
WNW	0	12	12	0	43	614	863	7,957	13,344	26,434	49,279	WNW
NW	114	22	5	2	86	843	1,166	3,003	12,348	21,857	39,446	NW
NNW	0	0	10	128	0	31	810	8,578	7,526	47,594	64,677	NNW
Totals:	600	906	2,393	3,590	5,600	20,593	54,819	165,042	215,869	332,772	802,184	
Density	191	120	138	149	167	107	70	90	93	102	People/Sq Mile	

Licensee's Reported Population

N	0	0	10	0	94	263	2,370	6,210	11,000	10,200	30,147	N
NNE	0	104	196	221	88	361	2,680	6,550	7,630	29,700	47,530	NNE
NE	0	56	74	56	64	527	2,770	11,900	10,300	23,900	49,647	NE
ENE	6	45	226	244	219	1,320	3,700	14,300	17,300	52,100	69,460	ENE
E	20	88	339	657	1,550	4,730	3,220	3,480	6,520	23,300	43,904	E
ESE	6	186	330	387	2,220	10,100	18,000	21,300	13,600	18,700	84,829	ESE
SE	0	35	265	860	1,640	3,370	5,120	67,600	12,900	15,600	107,390	SE
SSE	0	14	353	573	528	1,430	5,760	9,180	10,400	22,000	50,238	SSE
S	31	0	79	56	56	679	2,570	4,450	6,750	10,600	25,271	S
SSW	107	222	39	17	82	591	5,940	4,640	70,700	11,000	93,338	SSW
SW	82	247	170	44	124	351	2,380	4,950	10,500	14,000	32,848	SW
WSW	130	159	329	84	50	256	3,070	16,600	10,600	28,500	59,778	WSW
W	130	14	19	33	22	246	2,420	5,490	2,900	414	11,688	W
WNW	130	14	4	0	37	153	1,250	8,790	9,850	38,300	58,528	WNW
NW	0	39	64	42	42	1,280	1,140	5,650	12,000	22,100	42,357	NW
NNW	0	14	70	10	0	18	1,230	7,540	8,340	48,600	65,822	NNW
Totals:	642	1,237	2,567	3,284	6,816	25,675	63,620	198,630	221,290	349,014	872,775	
Density	204	150	157	154	185	128	83	107	104	111	People/Sq Mile	

SECPOP2000 as Percent of Reported

N			*	**	133.0%	93.2%	46.5%	85.2%	98.5%	74.3%	83.8%	N
NNE		26.0%	174.0%	95.5%	185.2%	148.5%	86.5%	106.9%	81.3%	72.9%	80.9%	NNE
NE		53.6%	263.5%	287.5%	103.1%	157.3%	46.5%	87.7%	73.7%	96.4%	87.9%	NE
ENE	*	553.3%	72.1%	29.5%	108.7%	102.7%	73.1%	31.4%	94.9%	77.7%	72.9%	ENE
E	*	22.7%	115.6%	113.4%	49.7%	67.2%	95.4%	26.8%	38.4%	104.5%	81.9%	E
ESE	550.0%	89.8%	30.9%	133.9%	77.7%	60.8%	89.8%	127.8%	58.7%	85.7%	89.7%	ESE
SE		128.6%	50.2%	86.7%	107.0%	106.9%	90.1%	73.8%	109.4%	52.8%	77.4%	SE
SSE	**	*	86.1%	89.4%	42.6%	90.2%	82.2%	56.3%	80.7%	64.8%	69.5%	SSE
S	196.8%		88.6%	312.5%	210.7%	85.0%	67.4%	84.7%	76.2%	95.5%	86.1%	S
SSW	84.1%	34.2%	287.2%	117.6%	73.2%	84.3%	136.5%	105.3%	105.9%	116.8%	108.8%	SSW
SW	147.6%	70.0%	100.6%	311.4%	93.5%	101.4%	98.0%	105.4%	79.2%	67.0%	80.1%	SW
WSW	102.3%	34.6%	99.7%	65.5%	200.0%	72.3%	49.6%	114.2%	163.9%	213.0%	166.3%	WSW
W	6.9%	214.3%	289.5%	69.7%	50.0%	125.2%	94.4%	41.1%	99.7%	910.1%	99.5%	W
WNW	*	85.7%	300.0%		116.2%	401.3%	69.0%	90.5%	135.5%	69.0%	84.2%	WNW
NW	**	56.4%	7.8%	4.8%	204.8%	65.9%	102.3%	53.2%	102.9%	98.9%	93.1%	NW
NNW			*	14.3%	1280.0%		172.2%	65.9%	113.8%	90.2%	97.9%	98.3%	NNW
	93.5%	73.2%	93.2%	109.3%	82.2%	80.2%	86.2%	83.1%	97.6%	95.3%	91.9%	

Run Date: 07/11/02

* The licensee reported people in the ring segment but SECPOP90 did not locate anyone in the ring segment.
** SECPOP90 located people in the ring segment but the licensee did not report anyone in the ring segment.
Licensee data from FSAR, Tables 2.1.3-1 and 2.1.3-2. ESTIMATED from 1980 Census data.

Seabrook 10 Rings to 50 Miles

SPATIAL DISTANCES

Rings:	1	2	3	4	5	10	20	30	40	50		
SECPOP2000 POPULATION											Total	
N	0	6	222	1,212	930	4,303	18,596	48,973	30,662	11,555	116,659	N
NNE	0	36	1,498	2,819	594	6,972	37,122	13,619	24,595	54,548	141,863	NNE
NE	29	47	688	2,381	1,261	1,429	1,257	414	1	158	7,665	NE
ENE	0	403	1,171	428	0	0	0	0	0	0	2,002	ENE
E	0	559	83	0	0	0	0	0	0	0	642	E
ESE	0	756	13	0	0	0	0	0	0	0	769	ESE
SE	163	140	374	8	0	0	2,645	4,177	0	0	7,499	SE
SSE	68	360	463	1,189	570	1,788	6,981	27,814	0	0	39,433	SSE
S	139	412	692	715	1,515	7,714	24,518	196,181	54,682	190,470	477,038	S
SSW	65	690	1,126	584	337	12,369	25,814	132,772	859,576	812,351	1,845,684	SSW
SW	10	1,035	385	206	3,833	8,159	84,691	243,347	162,839	123,527	628,232	SW
WSW	234	234	475	493	2,924	10,245	57,568	67,747	146,085	52,054	338,059	WSW
W	0	474	1,483	330	347	3,130	23,379	67,400	103,784	27,269	227,596	W
WNW	144	146	98	117	436	2,824	17,017	21,508	95,493	37,194	174,979	WNW
NW	0	337	277	253	192	12,018	9,182	8,059	19,164	24,562	74,044	NW
NNW	0	190	234	571	545	5,927	26,181	25,886	15,379	10,562	85,475	NNW
Totals:	852	6,023	9,282	11,298	13,484	77,083	333,006	858,097	1,512,262	1,344,250	4,167,639	
Density	271	547	372	546	521	376	361	464	562	531	People/Sq Mile	
Licensee's Reported Population												
N	20	100	760	1,150	760	9,420	27,400	32,700	22,300	6,900	101,510	N
NNE	0	0	2,780	3,250	720	12,260	34,800	11,500	18,300	43,300	126,910	NNE
NE	0	110	1,290	2,220	1,400	2,000	2,800	0	0	0	9,820	NE
ENE	0	730	1,350	180	0	0	0	0	0	0	2,260	ENE
E	0	780	0	0	0	0	0	0	0	0	780	E
ESE	0	1,530	0	0	0	0	0	0	0	0	1,530	ESE
SE	0	90	730	0	0	0	8,900	900	0	0	10,620	SE
SSE	20	150	360	360	570	4,790	14,800	24,300	0	0	45,350	SSE
S	210	410	750	620	1,100	8,400	20,500	189,000	107,500	227,900	556,390	S
SSW	410	460	510	540	440	9,920	22,300	177,600	957,100	883,600	2,052,880	SSW
SW	100	1,110	430	250	3,620	12,920	73,100	204,600	194,000	140,400	630,530	SW
WSW	0	1,110	1,020	370	3,710	9,030	28,700	125,800	151,200	41,200	362,140	WSW
W	170	1,120	410	530	1,330	4,510	24,300	63,600	118,800	30,800	245,570	W
WNW	210	90	320	90	1,080	4,730	16,300	16,200	72,700	40,600	152,320	WNW
NW	30	280	190	150	150	8,550	13,200	4,200	12,100	19,800	58,650	NW
NNW	30	340	190	280	480	7,600	16,900	20,700	15,900	6,400	68,820	NNW
Totals:	1,200	8,410	11,090	9,990	15,360	94,130	304,000	871,100	1,669,900	1,440,900	4,426,080	
Density	382	765	732	611	586	446	333	465	594	564	People/Sq Mile	
SECPOP2000 as Percent of Reported												
N	*	6.0%	29.2%	105.4%	122.4%	47.8%	67.9%	149.8%	137.5%	167.5%	114.9%	N
NNE		**	53.9%	86.7%	82.5%	56.9%	106.8%	118.4%	134.4%	126.0%	111.8%	NNE
NE	**	42.7%	53.3%	107.3%	90.1%	71.5%	44.9%	**	**	**	78.1%	NE
ENE		55.2%	86.7%	237.8%							88.6%	ENE
E		71.7%	**								82.3%	E
ESE		49.4%	**								50.3%	ESE
SE	**	135.6%	51.2%				29.7%	464.1%			70.6%	SE
SSE	340.0%	373.3%	128.6%	330.3%	100.0%	37.3%	47.2%	114.5%			87.0%	SSE
S	66.2%	100.5%	92.3%	115.3%	137.7%	91.8%	119.6%	103.8%	50.9%	83.6%	85.7%	S
SSW	15.9%	150.0%	220.8%	108.1%	76.6%	124.7%	115.8%	74.8%	89.8%	91.9%	89.9%	SSW
SW	10.0%	93.2%	89.5%	82.4%	105.9%	63.2%	115.9%	119.0%	83.9%	88.0%	99.6%	SW
WSW	**	21.1%	46.6%	133.2%	78.8%	113.5%	200.6%	53.9%	96.6%	126.3%	93.4%	WSW
W	*	42.3%	361.7%	62.3%	26.1%	69.4%	96.2%	106.0%	87.4%	88.5%	92.7%	W
WNW	68.6%	162.2%	30.6%	130.0%	40.4%	59.7%	104.4%	132.8%	131.4%	91.6%	114.9%	WNW
NW	*	120.4%	145.8%	168.7%	128.0%	140.6%	69.6%	191.9%	158.4%	124.1%	126.2%	NW
NNW	*	55.9%	123.2%	203.9%	113.5%	78.0%	154.9%	125.1%	96.7%	165.0%	124.2%	NNW
	71.0%	71.6%	83.7%	113.1%	87.8%	81.9%	110.2%	98.5%	90.6%	93.3%	94.2%	

* The licensee reported people in the ring segment but SECPOP90 did not locate anyone in the ring segment.
** SECPOP90 located people in the ring segment but the licensee did not report anyone in the ring segment.
Licensee data from FSAR, Tables 2.1-2 and 2.1-4. ESTIMATED from 1980 Census data.

Run Date: 07/11/02

South Texas 10 Rings to 50 Miles

SPATIAL DISTANCES

SECPOP2000 POPULATION

Ring:	1	2	3	4	5	10	20	30	40	50	Total	
N	0	0	15	0	0	32	1,261	462	14,305	5,315	21,390	N
NNE	0	0	0	0	0	148	21,628	1,073	2,445	10,992	36,286	NNE
NE	0	0	0	0	31	83	747	6,969	11,567	25,850	45,247	NE
ENE	0	0	0	0	0	439	235	2,606	17,612	60,494	81,386	ENE
E	0	0	0	0	3	31	83	1,117	87	9	1,330	E
ESE	0	0	0	126	47	46	2	0	0	0	221	ESE
SE	0	0	0	17	100	380	13	0	0	0	510	SE
SSE	0	0	0	0	0	167	105	0	0	0	272	SSE
S	0	0	0	0	0	0	0	0	0	0	0	S
SSW	0	0	0	0	0	0	1	0	0	0	1	SSW
SW	0	0	0	0	0	67	42	0	1,111	458	1,678	SW
WSW	0	0	1	0	6	98	5,954	1,046	14,507	3,560	25,172	WSW
W	0	0	0	0	9	131	266	682	1,559	3,558	6,205	W
WNW	0	0	0	0	0	511	1,228	416	9,746	1,275	13,176	WNW
NW	0	0	0	19	30	206	486	794	1,485	221	3,241	NW
NNW	0	0	0	0	0	34	463	3,459	12,922	2,125	19,003	NNW
Totals:	0	0	16	162	226	2,373	32,514	18,624	87,346	113,857	255,118	
Density	0	0	1	4	5	9	28	19	28	32	People/Sq Mile	

Licensee's Reported Population

N	0	0	5	0	0	36	3,661	2,584	17,385	7,094	30,765	N
NNE	0	0	0	0	0	107	26,651	2,392	7,957	9,416	46,523	NNE
NE	0	0	0	0	0	44	2,914	9,566	25,124	21,160	58,808	NE
ENE	0	0	0	0	4	576	2,201	4,198	26,855	79,928	113,762	ENE
E	0	0	0	3	0	56	1,153	1,425	0	431	3,068	E
ESE	0	0	0	134	98	77	312	0	0	0	621	ESE
SE	0	0	0	61	70	550	0	0	0	0	681	SE
SSE	0	0	0	0	0	178	60	0	0	0	238	SSE
S	0	0	0	0	0	0	0	0	0	0	0	S
SSW	0	0	0	0	3	8	201	0	0	0	212	SSW
SW	0	0	0	0	16	77	259	0	2,234	1,679	4,265	SW
WSW	0	0	7	3	25	143	6,227	1,653	14,383	11,254	33,695	WSW
W	0	0	0	0	15	151	801	974	2,426	5,140	9,507	W
WNW	0	0	0	0	38	481	944	1,840	9,146	3,125	15,574	WNW
NW	0	0	0	0	24	292	1,056	1,606	2,409	2,762	8,149	NW
NNW	0	0	0	15	13	8	169	5,749	16,111	4,159	26,224	NNW
Totals:	0	0	12	216	306	2,784	46,609	31,987	124,030	146,148	352,092	
Density	0	0	0	5	7	11	40	29	41	45	People/Sq Mile	

SECPOP2000 as Percent of Reported

N			300.0%			88.9%	34.4%	17.9%	82.3%	74.9%	69.5%	N
NNE						138.3%	81.2%	44.9%	30.7%	116.7%	78.0%	NNE
NE					**	188.6%	25.6%	72.9%	46.0%	122.2%	76.9%	NE
ENE					*	76.2%	10.7%	62.1%	65.6%	75.7%	71.5%	ENE
E				*	**	55.4%	7.2%	78.4%	**	2.1%	43.4%	E
ESE				94.0%	48.0%	59.7%	0.6%				35.6%	ESE
SE				27.9%	142.9%	69.1%	**				74.9%	SE
SSE						93.8%	175.0%				114.3%	SSE
S												S
SSW					*	*	0.5%				0.5%	SSW
SW					*	87.0%	16.2%		49.7%	27.3%	39.3%	SW
WSW			14.3%	*	24.0%	68.5%	95.6%	63.3%	100.9%	31.6%	74.7%	WSW
W					60.0%	86.8%	33.2%	70.0%	64.3%	69.2%	65.3%	W
WNW					*	106.2%	130.1%	22.6%	106.6%	40.8%	84.6%	WNW
NW				**	125.0%	70.5%	46.0%	49.4%	61.6%	8.0%	39.8%	NW
NNW				*	*	425.0%	274.0%	60.2%	80.2%	51.1%	72.5%	NNW
			133.3%	75.0%	73.9%	85.2%	69.8%	58.2%	70.4%	77.9%	72.5%	

* The licensee reported people in the ring segment but SECPOP90 did not locate anyone in the ring segment.

** SECPOP90 located people in the ring segment but the licensee did not report anyone in the ring segment.

Licensee data from FSAR, Table 2.1-1. ESTIMATED from 1980 Census data.

Run Date: 07/11/02

Susquehanna 10 Rings to 50 Miles

SPATIAL DISTANCES

SECPOP2000 POPULATION

Ring:	1	2	3	4	5	10	20	30	40	50	Total	
N	0	46	5	695	930	1,429	4,711	567	6,467	6,583	21,433	N
NNE	33	23	0	37	53	2,411	17,067	10,295	16,896	9,603	56,418	NNE
NE	0	0	114	169	139	3,406	115,889	79,316	145,553	38,992	383,578	NE
ENE	0	0	0	68	46	2,063	18,143	4,732	13,539	19,271	57,862	ENE
E	23	55	73	142	73	1,333	7,021	3,538	17,133	50,335	79,726	E
ESE	4	0	205	118	203	1,627	13,411	5,451	25,623	52,600	99,242	ESE
SE	27	88	0	216	126	3,928	37,761	28,431	38,535	229,868	338,980	SE
SSE	0	14	0	107	64	3,007	7,434	15,905	14,566	64,335	105,432	SSE
S	76	57	103	193	45	701	15,584	37,188	26,146	29,184	109,277	S
SSW	0	160	93	107	126	783	3,312	26,191	13,681	17,696	62,149	SSW
SW	0	172	130	116	1,536	800	2,582	27,489	15,978	15,232	64,035	SW
WSW	0	275	52	549	4,616	11,492	25,603	17,225	37,894	33,524	131,230	WSW
W	0	51	157	51	302	1,740	3,971	5,834	29,945	15,022	59,073	W
WNW	0	35	46	36	29	697	3,686	3,797	18,009	65,477	91,812	WNW
NW	0	134	183	0	151	1,110	2,196	1,284	1,402	1,240	7,700	NW
NNW	0	0	0	0	132	1,705	1,594	902	4,299	9,843	18,475	NNW
Totals:	163	1,110	1,161	2,604	8,371	38,232	281,963	268,143	425,666	658,805	1,686,422	
Density	52	101	86	100	173	163	266	213	204	213	People/Sq Mile	

Licensee's Reported Population

N	40	15	49	1,159	2,215	1,200	3,800	1,800	7,300	5,000	22,578	N
NNE	0	17	30	0	0	1,800	16,900	16,100	19,700	9,400	63,947	NNE
NE	0	0	166	199	379	10,000	83,300	109,100	177,400	41,900	422,444	NE
ENE	0	15	38	70	97	1,500	31,500	37,500	12,400	9,600	92,720	ENE
E	0	57	23	43	79	1,200	11,500	1,300	6,900	32,100	53,202	E
ESE	24	27	64	182	177	2,300	12,300	3,200	7,300	31,300	56,874	ESE
SE	44	249	151	149	13	2,900	41,600	27,000	34,400	198,600	305,106	SE
SSE	40	157	38	93	47	3,700	11,500	25,300	10,700	52,700	104,275	SSE
S	36	150	30	142	10	1,900	9,400	30,800	38,900	29,200	110,568	S
SSW	4	249	45	86	123	1,000	5,500	46,100	13,400	21,500	88,007	SSW
SW	4	15	344	93	1,560	2,000	1,600	19,800	25,900	14,800	66,116	SW
WSW	4	61	340	116	4,181	12,800	21,700	23,700	29,700	39,400	132,002	WSW
W	4	31	60	96	187	1,500	4,200	4,700	21,200	15,600	47,578	W
WNW	12	42	45	103	267	700	3,900	2,100	8,500	43,000	58,669	WNW
NW	16	73	94	0	97	900	1,600	1,600	2,000	2,000	8,380	NW
NNW	20	19	15	0	23	1,000	1,600	1,100	3,300	7,500	14,577	NNW
Totals:	248	1,177	1,532	2,331	9,435	46,400	261,900	331,200	419,000	533,600	1,647,043	
Density	79	113	105	109	190	193	257	239	218	210	People/Sq Mile	

SECPOP2000 as Percent of Reported

N	*	306.7%	10.2%	60.0%	42.0%	119.1%	124.0%	31.5%	88.6%	131.7%	94.9%	N
NNE	**	135.3%	*	**	**	133.9%	101.0%	63.9%	85.8%	102.2%	88.2%	NNE
NE			68.7%	84.9%	36.7%	34.1%	139.1%	72.7%	82.0%	93.1%	90.8%	NE
ENE		*	*	97.1%	47.4%	137.5%	57.6%	12.6%	109.2%	200.7%	62.4%	ENE
E	**	96.5%	317.4%	330.2%	92.4%	111.1%	61.1%	272.2%	248.3%	156.8%	149.9%	E
ESE	16.7%	*	320.3%	64.8%	114.7%	70.7%	109.0%	170.3%	351.0%	168.1%	174.5%	ESE
SE	61.4%	35.3%	*	145.0%	969.2%	135.4%	90.8%	105.3%	112.0%	115.7%	111.1%	SE
SSE	*	8.9%	*	115.1%	136.2%	81.3%	64.6%	62.9%	136.1%	122.1%	101.1%	SSE
S	211.1%	38.0%	343.3%	135.9%	450.0%	36.9%	165.8%	120.7%	67.2%	99.9%	98.8%	S
SSW	*	64.3%	206.7%	124.4%	102.4%	78.3%	60.2%	56.8%	102.1%	82.3%	70.6%	SSW
SW	*	1146.7%	37.8%	124.7%	98.5%	40.0%	161.4%	138.8%	61.7%	102.9%	96.9%	SW
WSW	*	450.8%	15.3%	473.3%	110.4%	89.8%	118.0%	72.7%	127.6%	85.1%	99.4%	WSW
W	*	164.5%	261.7%	53.1%	161.5%	116.0%	142.2%	124.1%	141.3%	96.3%	124.2%	W
WNW	*	83.3%	102.2%	35.0%	10.9%	99.6%	94.5%	188.8%	211.9%	152.3%	156.5%	WNW
NW	*	183.6%	194.7%		155.7%	123.3%	137.3%	80.3%	70.1%	62.0%	91.9%	NW
NNW	*	*	*		573.9%	170.5%	99.6%	82.0%	130.3%	431.2%	126.7%	NNW
	65.7%	94.3%	75.8%	102.9%	90.7%	82.4%	107.7%	76.4%	101.6%	119.0%	102.4%	

* The licensee reported people in the ring segment but SECPOP90 did not locate anyone in the ring segment.
Run Date: 07/15/02

** SECPOP90 located people in the ring segment but the licensee did not report anyone in the ring segment.

Licensee data from FSAR, Revision 54, Tables 2.1-19 & 2.1-20. ESTIMATED.

Appendix G Census File Creation and Testing Source Code

The VBA subroutines in Section G.1 are designed for use as macros in a spreadsheet *Census DBase Conversion.xls*. They were used to create the census file *CENSUS00.dat*. Similarly, the VBA subroutines in Section G.2 were used as macros in a spreadsheet *2000 CensusDB V&V.xls.* They were used to verify and validate the census file *CENSUS00.dat*.

G.1 Module: "Census_File_Convert"

```
Attribute VB_Name = "Census_File_Test"
Option Explicit

Type Block_Data
    longit As Integer
    latit As Integer
    pop As Integer
    area As Long
    ccode As Integer
End Type

Sub MakeStateList()

    Dim A As Range
    Set A = ActiveCell
    Dim dire As String
    Dim fname As String
    Dim n As Integer
    Dim state
    Dim ccode
    Dim county

    dire = "C:\Documents and Settings\cwmorro\My Documents\SecPop00\Convert"
    fname = dire & "\State Codes.csv"

    Close #1
    Open fname For Input As #1

    Do While Not EOF(1)
        n = n + 1
        Input #1, state, county, ccode
        A.Offset(n, 0).Value = ccode
        A.Offset(n, 1).Value = state
        A.Offset(n, 2).Value = county
    Loop
    Close #1

End Sub

Sub Test1()

    Dim A As Range
    Dim pop(3200) As Double
    Dim area(3200) As Double
    Dim directory As String
    Dim X As Block_Data
    Dim n As Long
    Dim nbyte As Long
    Dim fname As String
    Dim i As Integer
```

```
    Set A = ThisWorkbook.Worksheets("sheet3").Range("a1")

    directory = "C:\Documents and Settings\cwmorro\My Documents\SecPop00\Census Data Files"
    fname = directory & "\sort_Block00.dat"
    Close #1
    Open fname For Binary As #1

    With A
        Do While Not EOF(1)
            If n Mod 100000 = 0 Then
                A.Value = n
            End If
            n = n + 1
            nbyte = 12 * n - 11
            Get #1, nbyte, X
            pop(X.ccode) = pop(X.ccode) + X.pop
            area(X.ccode) = area(X.ccode) + X.area
        Loop
        Close #1

        For i = 1 To 3200
            A.Offset(i, 0) = i
            A.Offset(i, 1) = pop(i)
            A.Offset(i, 2) = area(i) / 1000
        Next i
    End With

End Sub

Sub Test3()

    Dim A As Range
    Dim pop(3200) As Double
    Dim area(3200) As Double
    Dim area2(3200) As Double
    Dim area3(3200) As Double
    Dim directory As String
    Dim X As Block_Data
    Dim n As Long
    Dim fname1 As String
    Dim fname2 As String
    Dim i As Integer
    Dim ib As Long
    Dim name, key, arealand, Intptlat, Intptlon, Pop100, county, StateCe

    Set A = ActiveCell

    directory = "C:\Documents and Settings\cwmorro\My Documents\SecPop00\Census Data Files"
    fname1 = directory & "\sort_Block00.dat"
    fname2 = "C:\Program Files\SecPop90\CENSUS" & "\CENSUS90.DAT"
    Close #1
    Close #2
    Open fname1 For Binary As #1
    Open fname2 For Binary As #2

    Input #2, name, key, arealand, Intptlat, Intptlon, Pop100, county, StateCe

    With A
        Do While Not EOF(1)
            If n Mod 100000 = 0 Then
                A.Value = n
            End If
            n = n + 1
            Get #1, , X
            pop(X.ccode) = pop(X.ccode) + X.pop
```

```
            area(X.ccode) = area(X.ccode) + X.area / 1000#
        Loop
        Close #1
        n = 0
        Do While Not EOF(2)
           If n Mod 100000 = 0 Then
               A.Value = n
           End If
           n = n + 1
           Let ib = n * 12 - 11
           Get #2, ib, X
           pop(X.ccode) = pop(X.ccode) + X.pop
           area2(X.ccode) = area2(X.ccode) + X.area / 1000#
        Loop

        For i = 1 To 3200
            A.Offset(i, 0) = i
            A.Offset(i, 1) = pop(i)
            A.Offset(i, 2) = area(i)
            A.Offset(i, 3) = area2(i)
        Next i
    End With

End Sub

Sub test2()

    Dim A As Range
    Dim ThisLong As Integer
    Dim LastLong As Integer
    Dim Blockfile1 As String
    Dim directory As String
    Dim i As Long
    Dim n As Long
    Dim X As Block_Data

    directory = "C:\Documents and Settings\cwmorro\My Documents\SecPop00\Census Data Files"
    Set A = ActiveCell
'
    Blockfile1 = directory & "\Sort_Block00.dat"

    Close #1
    Open Blockfile1 For Binary As #1

    LastLong = 32767

    With A
        Do While Not EOF(1)
            i = i + 1
            Get #1, 12 * (i - 1) + 1, X
            ThisLong = X.longit
            If i Mod 100000 = 0 Then A.Value = i
            If (ThisLong > LastLong) Then
                If EOF(1) Then Exit Do
                n = n + 1
                A.Offset(n, 0) = i
                A.Offset(n, 2) = ThisLong
                A.Offset(n, 1) = LastLong
            End If
            LastLong = ThisLong
        Loop
        A.Value = i
    End With
End Sub
```

```
Sub Test4()

    Dim A As Range
    Dim CountyPop(3200) As Double
    Dim CountyArea(3200) As Double
    Dim StatePop(100) As Double
    Dim StateArea(100) As Double
    Dim directory As String
    Dim StateList(3200) As Integer
    Dim X As Block_Data
    Dim n As Long
    Dim nbyte As Long
    Dim fname As String
    Dim i As Integer

    Set A = ThisWorkbook.Worksheets("State List").Range("A1")
    With A
        For i = 1 To UBound(StateList)
            StateList(i) = .Offset(i, 1).Value
        Next i
    End With

    Set A = ActiveCell

    directory = "C:\Documents and Settings\cwmorro\My Documents\SecPop00\Census Data Files"
    fname = directory & "\sort_Block00.dat"
    Close #1
    Open fname For Binary As #1

    With A
        Do While Not EOF(1)
            If n Mod 100000 = 0 Then
                A.Value = n
            End If
            n = n + 1
            nbyte = 12 * n - 11
            Get #1, nbyte, X
            CountyPop(X.ccode) = CountyPop(X.ccode) + X.pop
            CountyArea(X.ccode) = CountyArea(X.ccode) + X.area
            StatePop(StateList(X.ccode)) = StatePop(StateList(X.ccode)) + X.pop
            StateArea(StateList(X.ccode)) = StateArea(StateList(X.ccode)) + X.area
        Loop
        Close #1
        For i = 1 To 3200
            A.Offset(i, 0) = i
            A.Offset(i, 1) = CountyPop(i)
            A.Offset(i, 2) = CountyArea(i) / 1000
        Next i
        For i = 1 To 100
            A.Offset(i, 5) = i
            A.Offset(i, 6) = StatePop(i)
            A.Offset(i, 7) = StateArea(i) / 1000
        Next i
    End With
End Sub
```

G.2 Module: "Census_File_Test"

```
Attribute VB_Name = "Census_File_Test"
Option Explicit

Type Block_Data
    longit As Integer
    latit As Integer
    pop As Integer
    area As Long
    ccode As Integer
End Type

Sub MakeStateList()

    Dim A As Range
    Set A = ActiveCell
    Dim dire As String
    Dim fname As String
    Dim n As Integer
    Dim state
    Dim ccode
    Dim county

    dire = "C:\Documents and Settings\cwmorro\My Documents\SecPop00\Convert"
    fname = dire & "\State Codes.csv"

    Close #1
    Open fname For Input As #1

    Do While Not EOF(1)
        n = n + 1
        Input #1, state, county, ccode
        A.Offset(n, 0).Value = ccode
        A.Offset(n, 1).Value = state
        A.Offset(n, 2).Value = county
    Loop
    Close #1

End Sub

Sub Test1()

    Dim A As Range
    Dim pop(3200) As Double
    Dim area(3200) As Double
    Dim directory As String
    Dim X As Block_Data
    Dim n As Long
    Dim nbyte As Long
    Dim fname As String
    Dim i As Integer

    Set A = ThisWorkbook.Worksheets("sheet3").Range("a1")

    directory = "C:\Documents and Settings\cwmorro\My Documents\SecPop00\Census Data Files"
    fname = directory & "\sort_Block00.dat"
    Close #1
    Open fname For Binary As #1

    With A
        Do While Not EOF(1)
            If n Mod 100000 = 0 Then
                A.Value = n
```

```
            End If
            n = n + 1
            nbyte = 12 * n - 11
            Get #1, nbyte, X
            pop(X.ccode) = pop(X.ccode) + X.pop
            area(X.ccode) = area(X.ccode) + X.area
        Loop
        Close #1

        For i = 1 To 3200
            A.Offset(i, 0) = i
            A.Offset(i, 1) = pop(i)
            A.Offset(i, 2) = area(i) / 1000
        Next i
    End With

End Sub


Sub Test3()

    Dim A As Range
    Dim pop(3200) As Double
    Dim area(3200) As Double
    Dim area2(3200) As Double
    Dim area3(3200) As Double
    Dim directory As String
    Dim X As Block_Data
    Dim n As Long
    Dim fname1 As String
    Dim fname2 As String
    Dim i As Integer
    Dim ib As Long
    Dim name, key, arealand, Intptlat, Intptlon, Pop100, county, StateCe

    Set A = ActiveCell

    directory = "C:\Documents and Settings\cwmorro\My Documents\SecPop00\Census Data Files"
    fname1 = directory & "\sort_Block00.dat"
    fname2 = "C:\Program Files\SecPop90\CENSUS" & "\CENSUS90.DAT"
    Close #1
    Close #2
    Open fname1 For Binary As #1
    Open fname2 For Binary As #2

    Input #2, name, key, arealand, Intptlat, Intptlon, Pop100, county, StateCe

    With A
        Do While Not EOF(1)
            If n Mod 100000 = 0 Then
                A.Value = n
            End If
            n = n + 1
            Get #1, , X
            pop(X.ccode) = pop(X.ccode) + X.pop
            area(X.ccode) = area(X.ccode) + X.area / 1000#
        Loop
        Close #1
        n = 0
        Do While Not EOF(2)
           If n Mod 100000 = 0 Then
               A.Value = n
           End If
           n = n + 1
           Let ib = n * 12 - 11
```

```
            Get #2, ib, X
            pop(X.ccode) = pop(X.ccode) + X.pop
            area2(X.ccode) = area2(X.ccode) + X.area / 1000#
        Loop

        For i = 1 To 3200
            A.Offset(i, 0) = i
            A.Offset(i, 1) = pop(i)
            A.Offset(i, 2) = area(i)
            A.Offset(i, 3) = area2(i)
        Next i
    End With

End Sub

Sub test2()

    Dim A As Range
    Dim ThisLong As Integer
    Dim LastLong As Integer
    Dim Blockfile1 As String
    Dim directory As String
    Dim i As Long
    Dim n As Long
    Dim X As Block_Data

    directory = "C:\Documents and Settings\cwmorro\My Documents\SecPop00\Census Data Files"
    Set A = ActiveCell
'
    Blockfile1 = directory & "\Sort_Block00.dat"

    Close #1
    Open Blockfile1 For Binary As #1

    LastLong = 32767

    With A
        Do While Not EOF(1)
            i = i + 1
            Get #1, 12 * (i - 1) + 1, X
            ThisLong = X.longit
            If i Mod 100000 = 0 Then A.Value = i
            If (ThisLong > LastLong) Then
                If EOF(1) Then Exit Do
                n = n + 1
                A.Offset(n, 0) = i
                A.Offset(n, 2) = ThisLong
                A.Offset(n, 1) = LastLong
            End If
            LastLong = ThisLong
        Loop
        A.Value = i
    End With
End Sub

Sub Test4()

    Dim A As Range
    Dim CountyPop(3200) As Double
    Dim CountyArea(3200) As Double
    Dim StatePop(100) As Double
    Dim StateArea(100) As Double
    Dim directory As String
    Dim StateList(3200) As Integer
    Dim X As Block_Data
```

```
    Dim n As Long
    Dim nbyte As Long
    Dim fname As String
    Dim i As Integer

    Set A = ThisWorkbook.Worksheets("State List").Range("A1")
    With A
        For i = 1 To UBound(StateList)
            StateList(i) = .Offset(i, 1).Value
        Next i
    End With


    Set A = ActiveCell

    directory = "C:\Documents and Settings\cwmorro\My Documents\SecPop00\Census Data Files"
    fname = directory & "\sort_Block00.dat"
    Close #1
    Open fname For Binary As #1

    With A
        Do While Not EOF(1)
            If n Mod 100000 = 0 Then
                A.Value = n
            End If
            n = n + 1
            nbyte = 12 * n - 11
            Get #1, nbyte, X
            CountyPop(X.ccode) = CountyPop(X.ccode) + X.pop
            CountyArea(X.ccode) = CountyArea(X.ccode) + X.area
            StatePop(StateList(X.ccode)) = StatePop(StateList(X.ccode)) + X.pop
            StateArea(StateList(X.ccode)) = StateArea(StateList(X.ccode)) + X.area
        Loop
        Close #1

        For i = 1 To 3200
            A.Offset(i, 0) = i
            A.Offset(i, 1) = CountyPop(i)
            A.Offset(i, 2) = CountyArea(i) / 1000
        Next i
        For i = 1 To 100
            A.Offset(i, 5) = i
            A.Offset(i, 6) = StatePop(i)
            A.Offset(i, 7) = StateArea(i) / 1000
        Next i
    End With
End Sub
```

Appendix H SECPOP2000 Source Code

The following sections contain source code listings for the modules, forms and class structures that make up the executable file *"SECPOP2000.exe"*

H.1 Visual Basic Code Module: CALCCODE.BAS

```
Attribute VB_Name = "CalcCode"
'*******************************************************************************
'  Module:  CalcCode
'
'  Project: SECPOP
'
'  Description: This module contains the subroutines necessary to allocate CENSUS DATA
'               to a population matrix.  The subroutines take user specifications from
'               other modules to determine the census file, location and areal extent
'               to use for this analysis.  Based on this information, this module populates
'               a sector population, sector area and sector economic matrices.  This data
'               provides the basis for output as MACCS input files or other uses.
'
'  Modification History
'  Date       By   Description
'  --------   ---  ------------------------------------------------------------
'  20020110   cwm  Incorporated '90 version into '00 project.
'  20020110   cwm  Modified popcalc sub to accept any size database files.

'*******************************************************************************

' GLOSSARY OF VARIABLES:
'    AMINLA = LATITUDE OF TROPIC OF CANCER
'    BNDRYE = LONGITUDE OF EASTERNMOST POINT ON GRID
'    BNDRYN = LATITUDE OF NORTHERNMOST POINT ON GRID
'    BNDRYS = LATITUDE OF SOUTHERNMOST POINT ON GRID
'    BNDRYW = LONGITUDE OF WESTERNMOST POINT ON GRID
'    DELLAT = DIFFERENCE IN LATITUDE FROM SITE TO GRID BOUNDARIES
'    DELLON = DIFFERENCE IN LONGITUDE FROM SITE TO GRID BOUNDARIES
'    DISMAX = MAXIMUM RADIAL DISTANCE FOR GRID (KM)
'    DPDLAT = DISTANCE PER DEGREE LATITUDE (KM)
'    DPDLON = DISTANCE PER DEGREE LONGITUDE (KM)
'    NUMDIR = NUMBER OF RADIAL DIRECTIONS IN GRID
'    NUMRAD = NUMBER OF RADIAL DISTANCES DESIGNATED
'    OFFSET = RADIANS TO END OF FIRST RADIAL DIRECTION
'    RADDIS = ARRAY OF RADIAL DISTANCES IN MACCS GRID
'    SCALE  = POPULATION SCALING FACTOR (MULTIPLIER) FOR SITE
'    SDPDLA = DISTANCE PER DEGREE LATITUDE AT THE SITE
'    SDPDLO = DISTANCE PER DEGREE LONGITUDE AT THE SITE
'    SITID$ = SITE IDENTIFIER
'    SLAT   = SITE LATITUDE IN DECIMAL DEGREES
'    SLON   = SITE LONGITUDE IN DECIMAL DEGREES
'    sector_population = ARRAY OF POPULATION FOUND IN EACH GRID ELEMENT
'    XLON   = LONGITUDE OF POINT READ FROM CENSUS DATA FILE
'    YLAT   = LATITUDE OF POINT READ FROM CENSUS DATA FILE

' EXTERNALS:
'    GETDIR = DETERMINES THE RADIAL DIRECTION OF A POINT (XLON,YLAT)
'             FROM THE SITE AT (SLON,SLAT)
'    GETDIS = DETERMINES THE DISTANCE (KM) PER DEGREE FOR BOTH
'             LATITUDE AND LONGITUDE AT A GIVEN LATITUDE
'    GETRAD = DETERMINES THE RADIAL SECTOR IN WHICH POINT (XLON,
'             YLAT) LIES RELATIVE TO THE SITE (SLAT,SLON)
'    POINTR = FINDS THE FIRST RECORD IN THE SORTED DATA FILE IN
```

```
'             WHICH THE POINT LIES ON OR TO THE EAST OF THE WESTERN
'             BOUNDARY OF THE GRID'$INCLUDE: 'SECPOP90.BI'


Option Explicit

Function DIST(X, Y)

  DIST = Sqr(X ^ 2! + Y ^ 2!)

End Function

Sub GETDIR(ylat, idir As Integer)

    ' THIS SUBROUTINE RETURNS THE DIRECTIONAL ELEMENT IN WHICH POINT
    ' (XLON,YLAT) LIES

    ' GLOSSARY OF VARIABLES:
    '   ANGHI  = ANGLE OF RIGHT SIDE OF THE RADIAL DIRECTION CONSIDERED
    '   ANGLO  = ANGLE OF LEFT SIDE OF THE RADIAL DIRECTION CONSIDERED
    '   IDIR   = RADIAL DIRECTION IN WHICH POINT (XLON,YLAT) LIES
    '   OFFSET = ANGLE OF RIGHT SIDE OF RADIAL DIRECTION 1
    '   SLAT   = LATITUDE OF SITE
    '   SLON   = LONGITUDE OF SITE
    '   THETA  = CLOCKWISE ANGLE FROM NORTH AT WHICH POINT (XLON,YLAT)
    '            LIES RELATIVE TO THE SITE
    '   XLON   = LONGITUDE OF THE SITE
    '   YLAT   = LATITUDE OF THE SITE
    '

    Dim offset As Double
    Dim AVDPD As Double
    Dim DX As Double
    Dim DY As Double
    Dim THETA As Double
    Dim k As Double
    Dim j As Double
    Dim L As Double

    Dim ANGLO As Double
    Dim ANGHI As Double
    Dim i As Integer
    'Dim idir As Integer


    offset = 3.1416 / 16!

    idir = 0
    If (slat = ylat) Then
        idir = 5
        If (slon <= xlon) Then
            idir = idir + 8
        End If
    Else
        AVDPD = (sdpdla + dpdlat) / 2!
        DX = dpdlon * (slon - xlon)
        DY = AVDPD * (ylat - slat)
        THETA = Atn(DX / DY)
        If ((Abs(THETA) < offset) Or (THETA = offset)) Then
            idir = 1
            If (ylat < slat) Then
                idir = idir + 8
            End If
        Else
            j = 1
```

```
            For i = 1 To 3
                k = i * 2 + 1
                L = i * 2 - 1
                j = j + 1
                ANGLO = offset * L
                ANGHI = offset * k
                If ((ANGLO < Abs(THETA)) And (Abs(THETA) < ANGHI)) Then
                    If (THETA > 0!) Then
                        idir = j
                        If (ylat < slat) Then
                            idir = idir + 8
                        End If
                    Else
                        idir = 18 - j
                        If (ylat < slat) Then
                            idir = idir - 8
                        End If
                    End If
                End If
            Next i
        End If
    End If

    If (idir = 0) Then
        If (THETA > 0!) Then
            idir = 5
            If (xlon > slon) Then
                idir = idir + 8
            End If
        Else
            idir = 13
            If (xlon < slon) Then
                idir = idir - 8
            End If
        End If
    End If

End Sub

Sub GETDIS(Y As Single)

    ' THIS SUBROUTINE RETURNS THE DISTANCE PER DEGREE LATITUDE AND
    ' DISTANCE PER DEGREE LONGITUDE AT THE GEODETIC LATITUDE (SLAT)

    ' GLOSSARY OF VARIABLES:
    '   COSGC  = COSINE OF GEOCENTRIC ANGLE
    '   COS2GC = SQUARE OF COSGC
    '   DGTORD = CONVERSION FACTOR FOR DEGREES TO RADIANS
    '   DPDLAT = DISTANCE PER DEGREE LATITUDE AT SLAT (KM)
    '   DPDLON = DISTANCE PER DEGREE LONGITUDE AT SLON (KM)
    '   GDRLA  = GEODETIC LATITUDE IN RADIANS

    '   QLAT   = GEODETIC LATITUDE OF POINT CONSIDERED
    '   RE2    = SQUARE OF EQUATORIAL RADIUS OF EARTH
    '   RGD    = RADIUS OF GEODETIC LATITUDE CIRCLE (KM)
    '   RLONG  = RADIUS OF LONGITUDE "CIRCLE" (KM)
    '   RP2    = SQUARE OF POLAR RADIUS OF EARTH
    '   RPROD  = PRODUCT OF EQUATORIAL AND POLAR RADII
    '   SINGC  = SINE OF GEOCENTRIC ANGLE
    '   SIN2GC = SQUARE OF SINGC
    '   THGC   = GEOCENTRIC ANGLE CORRESPONDING TO THE GEODETIC
    '            LATITUDE

    Dim RE, RE2, RP, RP2, RRATIO, RPROD, GDRLA, THGC, SINGC, SIN2GC, COSGC, COS2GC
    Dim RGC, RGD, RLONG
```

```
    ' DEFINE CONSTANTS

    RE = 6378.137
    RE2 = RE ^ 2!
    RP = 6356.7523142
    RP2 = RP ^ 2!
    RRATIO = RP2 / RE2
    RPROD = RE * RP

    ' FIND THE GEODETIC LATITUDE IN RADIANS

    GDRLA = Y * DgToRd

    ' FIND THE GEODETIC LATITUDE IN THE MASTER COORDINATE SYSTEM

    THGC = Atn(RRATIO * Tan(GDRLA))
    SINGC = Sin(THGC)
    SIN2GC = SINGC ^ 2!
    COSGC = Cos(THGC)
    COS2GC = COSGC ^ 2!

    ' FIND THE GEOCENTRIC RADIUS (KM)

    RGC = RPROD / Sqr(RE2 * SIN2GC + RP2 * COS2GC)

    ' FIND THE RADIUS OF THE LONGITUDE "CIRCLE"

    RLONG = RGC * COSGC

    ' FIND THE GEODETIC RADIUS

    RGD = RLONG * (Tan(THGC) / Sin(GDRLA))

    'FIND THE DISTANCE PER DEGREE LATITUDE AND LONGITUDE (KM)

    dpdlat = RGD * DgToRd
    dpdlon = RLONG * DgToRd

End Sub

Sub GETRAD(ylat, numrad As Integer, irad As Integer)

    ' THIS SUBROUTINE GETS THE RADIAL SPATIAL ELEMENT IN WHICH THE
    ' POINT (XLON,YLAT) LIES

    ' GLOSSARY OF VARIABLES:
    '   AVDPD  = AVERAGE DISTANCE PER DEGREE LATITUDE BETWEEN THE SITE
    '            AND A POINT AT LATITUDE YLAT
    '   DELX   = DIFFERENCE IN LONGITUDE BETWEEN A POINT AT (XLON,
    '            YLAT) AND THE SITE
    '   DELY   = DIFFERENCE IN LATITUDE BETWEEN A POINT AT (XLON,YLAT)
    '            AND THE SITE
    '   DISMAX = RADIUS OF THE FULL MACCS GRID
    '   DISTST = DISTANCE FROM POINT (XLON,YLAT) TO THE SITE
    '   IRAD   = THE RADIAL SECTOR OF THE MACCS GRID IN WHICH POINT
    '            (XLON, YLAT) LIES
    '   RADDIS = ARRAY OF RADIAL DISTANCES OF THE MACCS GRID BEING USED
    '   SLAT   = LATITUDE OF THE SITE
    '   SLON   = LONGITUDE OF THE SITE

    Dim DELX, AVDPD, DELY, DISTST, JRAD

    DELX = Abs(slon - xlon) * dpdlon
    AVDPD = (sdpdla + dpdlat) / 2!
```

```
    DELY = Abs(slat - ylat) * AVDPD

    DISTST = DIST(DELX, DELY)

    irad = 0

    If (DISTST <= dismax) Then
        If (DISTST <= raddis(1)) Then
            irad = 1
        ElseIf (numrad >= 2) Then
            For JRAD = 2 To numrad
                If ((DISTST > raddis(JRAD - 1)) And (DISTST <= raddis(JRAD))) Then
                    irad = JRAD
                    Exit For
                End If
            Next JRAD
        End If
    End If

End Sub

Sub POINTR(ByVal bndryw, census_file As Long, ylat, ipop As Integer, area As Long, county_code As
Integer, ibyte As Long)

    ' THIS SUBROUTINE FINDS THE FIRST RECORD IN THE CENSUS DATA FILE
    ' WHICH IS AT THE WESTERN BOUNDARY OF THE GRID

    ' GLOSSARY OF VARIABLES:
    '   ALAT   = POINT LATITUDE IN DECIMAL NOTATION
    '   ALON   = POINT LONGITUDE IN DECIMAL NOTATION
    '   BNDRYW = LONGITUDE OF WESTERN BOUNDARY OF GRID
    '   ILAT   = CODED INTEGER REPRESENTATION OF LATITUDE OF POINT
    '            IN CENSUS DATA FILE
    '   ILOM   = CODED INTEGER REPRESENTATION OF LONGITUDE OF POINT
    '            IN CENSUS DATA FILE
    '   ILONW  = CODED INTEGER REPRESENTATION OF WESTERN BOUNDARY
    '            LONGITUDE OF GRID
    '   XLON   = LONGITUDE OF FIRST CENSUS DATA POINT LOCATED ON OR
    '            WITHIN THE MACCS GRID
    '   YLAT   = LATITUDE OF FIRST CENSUS DATA POINT LOCATED ON OR
    '            WITHIN THE MACCS GRID
    Dim Numrec As Long

    Numrec = Number_of_Records

    '*************************************************
    ' ROUTINE TO READ FROM RANDOM ACCESS BINARY FILE *
    '*************************************************

    Dim MIDREC As Long, NBYTE As Long
    Dim IBEG As Long, IEND As Long
    Dim ILONW As Long, ilon, JLON, ilat, CEND$
    Dim A As Block_Data

    ILONW = bndryw * 1000 - longitude_offset
    Get #census_file, 1, rec
    ilon = rec.longit
    If (ilon > ILONW) Then
        IBEG = 1
        IEND = Numrec
        ibyte = 1
    Else
        ibyte = 1
        GoTo process
    End If
```

```
ReadRecord:

    MIDREC = Int((IBEG + IEND) / 2)
    NBYTE = ibyte
    ibyte = ((MIDREC - 1) * record_length) + 1
    If (NBYTE <> ibyte) Then
        Get #census_file, ibyte, rec
        ilon = rec.longit
        If (ilon > ILONW) Then
            IBEG = MIDREC
            GoTo ReadRecord
        ElseIf (ilon = ILONW) Then
            GoTo FindFirst
        ElseIf (ilon < ILONW) Then
            A = rec
            ibyte = ibyte - record_length
            Get #census_file, ibyte, rec
            JLON = rec.longit
            If (JLON < ILONW) Then
                IEND = MIDREC
                GoTo ReadRecord
            ElseIf (JLON = ILONW) Then
                GoTo FindFirst
            ElseIf (JLON > ILONW) Then
                ibyte = ibyte + record_length
                rec = A
                GoTo process
            End If
        End If
'    Else
'        MsgBox "Site not within Continental US.", vbOK, "Error"
'        Exit Sub
    End If

    ' Find the first point inside the western boundary

FindFirst:

    ibyte = ibyte - record_length
    A = rec
    Get #census_file, ibyte, rec
    ilon = rec.longit
    If (ilon = ILONW) Then
        GoTo FindFirst
    Else
        If (ilon > ILONW) Then
            ibyte = ibyte + record_length
            rec = A
            GoTo process
        End If
    End If

    ' When find the first point process the information

process:

    ilon = rec.longit
    ilat = rec.latit
    ipop = rec.pop
    area = rec.area
    county_code = rec.ccode

    xlon = (ilon + longitude_offset) / 1000!
    ylat = (ilat + latitude_offset) / 1000!
```

```
End Sub

Sub PopCalc()

    Dim TempString As String
    Dim census_file As Long, county_file As Long, debug_file As Long
    Dim NUMLIN  As Integer, INUM As Integer, B$, JBEG As Integer
    Dim JEND As Integer, NLEFT As Integer, C$
    Dim L1$, L2$, L3$, L4$, L5$, L6$, L7$, LA1 As Integer, LA2 As Integer
    Dim LA3 As Integer, LO1 As Integer, LO2 As Integer, LO3 As Integer
    Dim D$, DELLON, DELLAT, ilon As Integer, ilat As Integer
    Dim NP1, NP2, NP3, NP4, NP5, NP6, NP7, NP8, CEND$
    Dim ILIN As Integer, i As Integer, j As Integer, m As Integer
    Dim events_open As Integer, average_counter As Integer
    Dim start_file_pos As Long, end_file_pos As Long
    Dim num_records As Long, num_processed As Long, census_records As Long
    Dim hours As Integer, minutes As Integer, seconds As Single
    Dim time_begin As Long, time_end As Long, total_time As Long
    Dim prev_seconds As Single
'    Dim WestMost As Variant
'    Dim EastMost As Variant

    'Open census data file, located at the position specified (qqqcensus)
    'by the user in the setup form

    TempString = frmSetup.txtData_path.text & "\" & frmSetup.txtCensus_Database
    census_file = FreeFile
    If Dir$(TempString) = "" Then
        MsgBox "The census file (" & TempString & ") cannot be found.  Please check the directory
" & _
               "specified on the Setup form."
        Exit Sub
    Else
        Open TempString For Binary Access Read As #census_file Len = 32767
    End If
    Number_of_Records = LOF(census_file) / record_length

    'Initialize the county data arrays to their startup values
    'These arrays contain various types of county economic
    'data and state abreviations.

    frmCalculate.txtStatus.text = "Reading county information."
    ipop = 0
'    TempString = App.path & "\" & frmSetup.txtData_path.text + "\COUNTY90.DAT"
'    TempString = frmSetup.txtData_path.text + "\COUNTY90.DAT"
    TempString = frmSetup.txtData_path.text & "\" & frmSetup.txtCounty_Database.text
    county_file = FreeFile
    Open TempString For Input Access Read As #county_file


    'Read in the header line. County_state(0) is used as a dummy variable.

    Line Input #county_file, county_state(0)

    'Read in the rest of the file.

    For i = 1 To Number_of_Counties
        If EOF(county_file) Then Exit For
        Input #county_file, county_code
        county_state(i) = Input$(2, #county_file)
        'county_name(i) = Input$(4, #county_file)
        county_name(i) = Input$(max_county_name_length, #county_file)
        Input #county_file, county_frclnd(i)
        Input #county_file, county_frmfrc(i)
```

```
        Input #county_file, county_dpf(i)
        Input #county_file, county_asfp(i)
        Input #county_file, county_vfrm(i)
        Input #county_file, county_vnfrm(i)
        'Skip over the notes.
        Line Input #county_file, county_state(0)
    Next i

    Close (county_file)

    aminla = 23.45

    'Initialize RADDIS array to the values in the radial distance array so
    'that the values can be changed locally without affecting other modules.

    For i = 1 To number_of_radii
        If frmProblem_Data.optKilometers.Value = True Then
            raddis(i) = radial_distance(i)
        Else
            raddis(i) = radial_distance(i) * miles_to_kilometers
        End If
    Next i

    'Initialize the population and land fraction sector arrays and economics
    'region arrays to zero.

    For i = 1 To number_of_segments Step 1
        For j = 1 To number_of_radii Step 1
            sector_population(i, j) = 0
            sector_area(i, j) = 0
            sector_frclnd(i, j) = 0
        Next j
    Next i

    For i = 0 To number_econ_regions
        econ_data(i).region_area = 0
        econ_data(i).region_frmfrc = 0
        econ_data(i).region_dpf = 0
        econ_data(i).region_asfp = 0
        econ_data(i).region_vfrm = 0
        econ_data(i).region_vnfrm = 0
    Next i

    'Set local constants for number of raddii and distance of farthest radii.

    numrad = number_of_radii
    dismax = raddis(numrad)

ProcessSite:

    'Reading interactive input.

    sitid$ = frmSite_Data.txtSite_Name.text
    L1$ = frmSite_Data.txtLatitude_Degrees.text
    L2$ = frmSite_Data.txtLatitude_Minutes.text
    L3$ = frmSite_Data.txtLatitude_Seconds.text
    L4$ = frmSite_Data.txtLongitude_Degrees.text
    L5$ = frmSite_Data.txtLongitude_Minutes.text
    L6$ = frmSite_Data.txtLongitude_Seconds.text
    L7$ = frmProblem_Data.txtPopulation_multiplier.text

    'Convert interactive input strings into numeric values.

    LA1 = Int(Val(L1$))
    LA2 = Int(Val(L2$))
```

```
    LA3 = Int(Val(L3$))
    LO1 = Int(Val(L4$))
    LO2 = Int(Val(L5$))
    LO3 = Int(Val(L6$))
    ScaleD = Val(L7$)

    'Convert Degree/Minute/Second designation to decimal degrees.

    slat = Val(L1$) + Val(L2$) / 60! + Val(L3$) / 3600!
    slon = Val(L4$) + Val(L5$) / 60! + Val(L6$) / 3600!

    'Check to assure Site within U.S.

    Get #census_file, 1, rec
'    WestMost = rec.longit
'    WestMost = (WestMost + longitude_offset) / 1000!
'    Get #census_file, ((Number_of_Records - 1) * record_length) + 1, rec
'    EastMost = rec.longit
'    EastMost = (EastMost + longitude_offset) / 1000!
'    If slon < EastMost Then
'        MsgBox "Site East of Continental US.", vbOK, "Error"
'        frmCalculate.Visible = False
'        Exit Sub
'    ElseIf slon > WestMost Then
'        MsgBox "Site West of Continental US.", vbOK, "Error"
'        frmCalculate.Visible = False
'        Exit Sub
'    End If
'

    'Find the longitude-latitude boundaries of grid.
    'Set distance per degree latitude for site.

    Call GETDIS(slat)
    sdpdla = dpdlat

    'Set eastern and western boundaries.  These boundaries determine which
    'records are searched in the census file.

    DELLON = dismax / dpdlon
    bndryw = Int((slon + DELLON) * 1000 + 0.5) / 1000
    bndrye = Int((slon - DELLON) * 1000 + 0.5) / 1000

    'Find distance change per degree latitude at some minimun latitude
    '(AMINLA).

    Call GETDIS(aminla)

    'Set northern and southern boundaries.  These boundaries do not affect
    'which records are searched, but do determine whether or not a point
    'is inside of the circle being analyzed.

    DELLAT = dismax / sdpdla
    bndryn = Int((slat + DELLAT) * 1000 + 0.5) / 1000
    DELLAT = dismax / dpdlat
    bndrys = Int((slat - DELLAT) * 1000 + 0.5) / 1000

    'Find the last record in the file with the longitude within the
    'longitudinal boundaries of the grid.  This record is used to
    'estimate the run time for the problem.

    xlon = 0!
    ylat = 0!
    area = 0&
    county_code = 0
```

```
Call POINTR(ByVal bndrye, census_file, ylat, ipop, area, county_code, ibyte)
end_file_pos = ibyte
If (xlon = 0!) Then
    end_file_pos = LOF(census_file)
End If

frmCalculate.txtStatus.text = "Searching for first census record."

ipop = 0
'Find the first record in the file with the longitude within the
'longitudinal boundaries of the grid.

ipop = 0
xlon = 0!
ylat = 0!
area = 0&
county_code = 0
Call POINTR(ByVal bndryw, census_file, ylat, ipop, area, county_code, ibyte)
If (xlon = 0!) Then
    GoTo finishup
End If
start_file_pos = ibyte

'Determine number of records to be scanned in census file (this
'number is used in determining estimated run time for a problem).

num_records = ((end_file_pos - start_file_pos) / record_length)

'Verify that the starting position is within the specified
'longitude and latitude.  If so, read in and process the data
'from the first specified record in the census database.

idir = 0
irad = 0

If ((xlon <= bndryw) And (xlon >= bndrye)) Then

    If ((ylat <= bndryn) And (ylat >= bndrys)) Then

        'Determine the distance between the specified site and the first
        'location in the census database.

        Call GETDIS(ylat)
        Call GETRAD(ylat, numrad, irad)

        'If the point is outside of the outermost radius of the circle,
        'read in and process the next record ...

        If (irad = 0) Then

            GoTo Continue1

        Else

            'Otherwise, determine in which sector the block lies.

            Call GETDIR(ylat, idir)

            'Set the population of the sector to the population of the
            'block.

            sector_population(idir, irad) = ipop * ScaleD
            sector_area(idir, irad) = area
            sector_frclnd(idir, irad) = area * county_frclnd(county_code)
```

```
                'Set the regional economic data by setting each of the
                'appropriate elements in the user defined data type.

                econ_data(region_index(irad, idir - 1)).region_area = area
                econ_data(region_index(irad, idir - 1)).region_frmfrc = area *
county_frmfrc(county_code)
                econ_data(region_index(irad, idir - 1)).region_dpf = area *
county_dpf(county_code)
                econ_data(region_index(irad, idir - 1)).region_asfp = area *
county_asfp(county_code)
                econ_data(region_index(irad, idir - 1)).region_vfrm = area *
county_vfrm(county_code)
                econ_data(region_index(irad, idir - 1)).region_vnfrm = area *
county_vnfrm(county_code)

                'Set number of census records processed thus far.

                census_records = 1

            End If

        End If

    End If

Continue1:

    'Update labels on calculate form, and begin timer to estimate
    'the run time of the calculation

    frmCalculate.txtStatus.text = "Processing census records."
    time_begin = Timer

    'Read each subsequent record in the data file until the first record
    'with a longitude outside of the longitudinal boundaries

readit:

    'If end of file is reached, finish the economic calculations and exit.

    If (EOF(census_file)) Then
        GoTo finishup
    End If

    'Change to get statement with binary file, and Go to the next record.

    ibyte = ibyte + record_length

    'Read the data from that record, split it into its constituent pieces,
    'and convert those peices from file format to program format.  This
    'portion of code is done by the pointr function for the first record.

    Get #census_file, ibyte, rec
    ilon = rec.longit
    ilat = rec.latit
    ipop = rec.pop
    area = rec.area
    county_code = rec.ccode
    xlon = (ilon + longitude_offset) / 1000!
    ylat = (ilat + latitude_offset) / 1000!

    'Check and see if the point is within the lat and long boundaries of
    'the circle.

    If ((xlon <= bndryw) And (xlon >= bndrye)) Then
```

```
        If ((ylat <= bndryn) And (ylat >= bndrys)) Then

            'if so, then determine the distance between the point and the
            'center of the circles.

            idir = 0
            irad = 0
            Call GETDIS(ylat)
            Call GETRAD(ylat, numrad, irad)

            'If the point lies outside of the farthest radii, proceed to the
            'next point.

            If (irad = 0) Then
                GoTo Continue2
            Else

                'Determine which sector the point lies in.

                Call GETDIR(ylat, idir)

                'Add the population of the block to the total population of
                'the sector in which it lies.

                sector_population(idir, irad) = sector_population(idir, irad) + (ipop * ScaleD)
                sector_area(idir, irad) = sector_area(idir, irad) + area
                sector_frclnd(idir, irad) = sector_frclnd(idir, irad) + (area *
county_frclnd(county_code))

                'Increment the regional economic data by incrementing each of
                'the appropriate elements in the user defined data type.

                econ_data(region_index(irad, idir - 1)).region_area =
econ_data(region_index(irad, idir - 1)).region_area + area
                econ_data(region_index(irad, idir - 1)).region_frmfrc =
econ_data(region_index(irad, idir - 1)).region_frmfrc + (area * county_frmfrc(county_code))
                econ_data(region_index(irad, idir - 1)).region_dpf = econ_data(region_index(irad,
idir - 1)).region_dpf + (area * county_dpf(county_code))
                econ_data(region_index(irad, idir - 1)).region_asfp =
econ_data(region_index(irad, idir - 1)).region_asfp + (area * county_asfp(county_code))
                econ_data(region_index(irad, idir - 1)).region_vfrm =
econ_data(region_index(irad, idir - 1)).region_vfrm + (area * county_vfrm(county_code))
                econ_data(region_index(irad, idir - 1)).region_vnfrm =
econ_data(region_index(irad, idir - 1)).region_vnfrm + (area * county_vnfrm(county_code))

                'Increment the number of records processed.

                census_records = census_records + 1

                If (census_records <= 1000) Then

                    'PRINT #debug_file, USING "###.### ##.### ##### ## ##"; xlon; ylat; ipop;
idir; irad
                End If

                If census_records Mod 100 = 0 Then

                    'Process any user events that may have occured while
                    'calculating. At this point in time only the close
                    'calculation button is allowed.

                    events_open = DoEvents()

                    'If the user has selected close from the calculate
                    'form, then close the census file and exit the
```

```
            'subroutine.

            If (leave_calculation) Then
                Close #census_file
                leave_calculation = False
                Exit Sub
            End If

            'Update the number of census records displayed
            'status line.

            frmCalculate.txtStatus.text = _
                LTrim$(Str$(census_records)) + _
                " census records processed."

            'Read the timer.

            time_end = Timer

            'Determine how many census records have been read
            '(including those which were not actually
            'processed).

            num_processed = _
                ((ibyte - start_file_pos) / record_length)

            'Determine the average time per record and how long
            'to process all remaining records.

            total_time = (((time_end - time_begin) * _
                (num_records - num_processed)) / num_processed)

            'Convert time from seconds to hours/minutes/seconds.

            hours = total_time \ 3600
            minutes = (total_time \ 60) - (hours * 60)
            seconds = _
                (total_time - ((hours * 60 + minutes) * 60))

            'If the time has changed by more than 5 seconds,
            'then update the display.

            If (Abs(seconds - prev_seconds) > 5) Then

                If hours = 0 Then
                    frmCalculate.txtTime.text = _
                        "Minutes:" + Str$(minutes) + _
                        "  Seconds:" + Str$(seconds)
                Else
                    frmCalculate.txtTime.text = _
                        "Hours:" + Str$(hours) + _
                        "  Minutes:" + Str$(minutes) + _
                        "  Seconds:" + Str$(seconds)
                End If

                prev_seconds = seconds

            End If

        End If

    End If

End If
```

```
        GoTo readit

    Else

        GoTo finishup

    End If

Continue2:

    GoTo readit

finishup:

    'Close census data file, and update display labels.

    Close #census_file
    'CLOSE #debug_file

    frmCalculate.txtStatus.text = "Calculating economic data."

    'Determine the land fraction values for each sector.

    For i = 1 To number_of_segments Step 1

        For j = 1 To number_of_radii Step 1

            If (sector_area(i, j) <> 0) Then

                sector_frclnd(i, j) = sector_frclnd(i, j) / sector_area(i, j)

            Else

                sector_frclnd(i, j) = 0

            End If

        Next j

    Next i

    'Determine the economic values for each region.

    For i = 1 To number_econ_regions Step 1

        If (econ_data(i).region_area <> 0) Then

            econ_data(i).region_frmfrc = econ_data(i).region_frmfrc / econ_data(i).region_area
            econ_data(i).region_dpf = econ_data(i).region_dpf / econ_data(i).region_area
            econ_data(i).region_asfp = econ_data(i).region_asfp / econ_data(i).region_area
            econ_data(i).region_vfrm = econ_data(i).region_vfrm / econ_data(i).region_area
            econ_data(i).region_vnfrm = econ_data(i).region_vnfrm / econ_data(i).region_area

        Else

            econ_data(i).region_frmfrc = 0
            econ_data(i).region_dpf = 0
            econ_data(i).region_asfp = 0
            econ_data(i).region_vfrm = 0
            econ_data(i).region_vnfrm = 0

        End If

    Next i
```

```
    'Update the user display for the final time letting the user know
    'That we are finished and the total number of census records
    'processed.

    frmCalculate.txtStatus.text = "Finished, " + _
        LTrim$(Str$(census_records)) + " processed."

    'CLOSE (debug_file)

End Sub
```

H.2 Visual Basic Code Module: Cmndlg.bas

```
Attribute VB_Name = "Cmndlg"
'''Option Explicit
'''
''''$INCLUDE: 'secpop90.bi'
'''
'''' -------------------------------------------------------------------
'''' Visual Basic for MS-DOS Common Dialog Toolkit
''''
'''' The Common Dialog Toolkit (CMNDLG.BAS and CMNDLGF.FRM)
'''' provides support for the following dialogs:
''''       FileOpen
''''       FileSave
''''       FilePrint
''''
'''' Support for each dialog is provided via procedures with
'''' these same names that create the corresponding dialog
'''' and return user input to your program.  These procedures
'''' only provide the user interface and return user input.
'''' They do not actually carry out the corresponding actions
'''' such as opening the file.  Detailed descriptions of
'''' these procedures are contained in the comment headers
'''' above each.
''''
'''' All common dialogs are created from the same form (CMNDLGF.FRM).
'''' The necessary controls for each dialog are children of
'''' a container picture box for the dialog.  Thus the
'''' form (CMNDLGF.FRM) contains a picture box with
'''' appropriate controls for common dialog listed above.
'''' When a particular common dialog is created and displayed,
'''' the container picture box for that dialog is made visible
'''' (thus all controls on that picture box become visible)
'''' and the form is centered and sized to match the
'''' container picture box.
''''
'''' To use these common dialogs in your programs, include
'''' CMNDLG.BAS and CMNDLGF.FRM in your program or use the
'''' supplied library (CMNDLG.LIB, CMNDLGA.LIB - AltMath version
'''' for Professional Edition only) and Quick library (CMNDLG.QLB)
'''' and call the appropriate procedure to invoke the dialog
'''' you need.
''''
'''' Copyright (C) 1982-1992 Microsoft Corporation
''''
'''' You have a royalty-free right to use, modify, reproduce
'''' and distribute the sample applications and toolkits provided with
'''' Visual Basic for MS-DOS (and/or any modified version)
'''' in any way you find useful, provided that you agree that
'''' Microsoft has no warranty, obligations or liability for
'''' any of the sample applications or toolkits.
'''' -------------------------------------------------------------------
'''
'''' FileOpen common dialog support routine
''''
'''' Displays Open dialog which allows users to select a
'''' file from disk.  This procedure only provides
'''' the user interface and returns user input.  It does
'''' not actually carry out the corresponding action.
''''
'''' Parameters:
''''   FileName - returns the name (without path) of the
''''            file the user wants to open.  To supply
''''            default filename in dialog, assign default
```

```
''''            to FileName then pass it to this procedure.
''''    PathName - returns the path (without filename) of
''''            the file the user wants to open.  To supply
''''            default path in dialog, assign default to
''''            PathName then pass it to this procedure.
''''            Note, only pass a valid drive and path. Do
''''            not include a filename or file pattern.
''''    DefaultExt - sets the default search pattern for the
''''            File Listbox.  Default pattern when DefaultExt
''''            is null is "*.*".  To specify a different
''''            search pattern (i.e. "*.BAS"), assign new
''''            value to DefaultExt then pass it to this
''''            procedure.
''''    DialogTitle - sets the dialog title.  Default title
''''            when DialogTitle is null is "Open".  To
''''            specify a different title (i.e. "Open My File"),
''''            assign new value to DialogTitle then pass it to
''''            this procedure.
''''    ForeColor - sets the dialog foreground color.  Does not affect
''''            SCREEN.ControlPanel color settings.
''''    BackColor - sets the dialog background color.  Does not affect
''''            SCREEN.ControlPanel color settings.
''''    Flags - unused.  Use this to customize dialog action if needed.
''''    Cancel - returns whether or not user pressed the dialog's Cancel
''''            button.  True (-1) means the user cancelled the dialog.
''''
'''Sub FileOpen(filename As String, pathname As String, _
'''     DefaultExt As String, DialogTitle As String, forecolor As Integer, _
'''     backcolor As Integer, Flags As Integer, Cancel As Integer)
'''
'''     ' Set up error handling for option validation.
'''     On Local Error GoTo FileOpenError
'''
'''     ' Set form caption.
'''     If DialogTitle = "" Then
'''         frmCmnDlg.caption = "Open"
'''     Else
'''         frmCmnDlg.caption = DialogTitle
'''     End If
'''
'''     ' Determine search pattern for file listbox.
'''     If DefaultExt <> "" Then
'''         frmCmnDlg.filOpenList.Pattern = DefaultExt
'''     Else
'''         frmCmnDlg.filOpenList.Pattern = "*.*"
'''     End If
'''
'''     ' Determine default path.
'''     If pathname <> "" Then
'''         ' Set drive and path for file-system controls.
'''         ' Set Directory listbox path.  If PathName is different
'''         ' than current path, PathChange event will be triggered
'''         ' which updates Drive listbox drive and File listbox path.
'''         frmCmnDlg.dirOpenList.Path = pathname
'''     End If
'''     ' Display current path to the user.
'''     frmCmnDlg.lblOpenPath.caption = frmCmnDlg.filOpenList.Path
'''
'''     ' Determine default filename to display in edit field.
'''     If filename <> "" Then
'''         frmCmnDlg.txtOpenFile.text = UCase$(filename)
'''     Else
'''         frmCmnDlg.txtOpenFile.text = frmCmnDlg.filOpenList.Pattern
'''     End If
'''
```

```
'''     ' Set default and cancel command buttons.
'''     frmCmnDlg.cmdOpenOK.Default = True
'''     frmCmnDlg.cmdOpenCancel.Cancel = True
'''
'''     ' Size and position Open/Save container.
'''     frmCmnDlg.pctFileOpen.BorderStyle = 0
'''     frmCmnDlg.pctFileOpen.Visible = True
'''
'''     ' Size and center dialog.
'''     frmCmnDlg.Move frmCmnDlg.Left, frmCmnDlg.Top, frmCmnDlg.pctFileOpen.Width + 2,
frmCmnDlg.pctFileOpen.Height + 2
'''     frmCmnDlg.Move (Screen.Width - frmCmnDlg.Width) \ 2, ((Screen.Height - frmCmnDlg.Height) \
2) - 2
'''
'''     ' Set dialog colors.
'''     frmCmnDlg.forecolor = forecolor
'''     frmCmnDlg.backcolor = backcolor
'''     frmCmnDlg.pctFileOpen.forecolor = forecolor
'''     frmCmnDlg.pctFileOpen.backcolor = backcolor
'''     frmCmnDlg.lblOpenFile.forecolor = forecolor
'''     frmCmnDlg.lblOpenFile.backcolor = backcolor
'''     frmCmnDlg.txtOpenFile.forecolor = forecolor
'''     frmCmnDlg.txtOpenFile.backcolor = backcolor
'''     frmCmnDlg.lblOpenPath.forecolor = forecolor
'''     frmCmnDlg.lblOpenPath.backcolor = backcolor
'''     frmCmnDlg.filOpenList.forecolor = forecolor
'''     frmCmnDlg.filOpenList.backcolor = backcolor
'''     frmCmnDlg.drvOpenList.forecolor = forecolor
'''     frmCmnDlg.drvOpenList.backcolor = backcolor
'''     frmCmnDlg.dirOpenList.forecolor = forecolor
'''     frmCmnDlg.dirOpenList.backcolor = backcolor
'''     frmCmnDlg.cmdOpenOK.backcolor = backcolor
'''     frmCmnDlg.cmdOpenCancel.backcolor = backcolor
'''
'''     ' Display dialog modally.
'''     frmCmnDlg.Show 1
'''
'''     ' Determine if user canceled dialog.
'''     If frmCmnDlg.cmdOpenCancel.Tag <> "FALSE" Then
'''         Cancel = True
'''     ' If not, return FileName and PathName.
'''     Else
'''         Cancel = False
'''         filename = frmCmnDlg.txtOpenFile.text
'''         pathname = frmCmnDlg.filOpenList.Path
'''         frmCmnDlg.cmdOpenCancel.Tag = ""
'''     End If
'''
'''     ' Hide or unload dialog and return control to user's program.
'''     ' (Hide if user chose to preload form for performance.)
'''     If Left$(frmCmnDlg.Tag, 1) = "H" Then
'''         frmCmnDlg.pctFileOpen.Visible = False
'''         frmCmnDlg.Hide
'''     Else
'''         Unload frmCmnDlg
'''     End If
'''
'''     Exit Sub
'''
'''' Option error handling routine.
'''' Ignore errors here and let dialog's controls
'''' handle the errors.
'''FileOpenError:
'''     Select Case Err
'''     Case 7:                                     ' Out of memory.
```

```
'''          MsgBox "Out of memory.  Can't load dialog.", 0, "FileOpen"
'''          Cancel = True
'''          Exit Sub
'''     Case Else
'''          Resume Next
'''     End Select
'''End Sub
'''
'''' FilePrint common dialog support routine
''''
'''' Displays Print dialog which allows users to select
'''' Print destination (PRINTER.PrintTarget) and the
'''' number of copies to print.  This procedure only provides
'''' the user interface and returns user input.  It does
'''' not actually carry out the corresponding action.
''''
'''' Parameters:
''''   Copies - returns the number of copies (1 to 99) the user wants
''''          to print.  To supply default number of copies
''''          in dialog, assign default to Copies then
''''          pass it to this procedure (default when Copies
''''          is 0 is 1).
''''   ForeColor - sets the dialog foreground color.  Does not affect
''''          SCREEN.ControlPanel color settings.
''''   BackColor - sets the dialog background color.  Does not affect
''''          SCREEN.ControlPanel color settings.
''''   Cancel - returns whether or not user pressed the dialog's Cancel
''''          button.  True (-1) means the user cancelled the dialog.
''''
'''Sub FilePrint(Copies As Integer, forecolor As Integer, backcolor As Integer, Cancel As
Integer)
'''
'''     Dim i As Integer
'''
'''     On Local Error GoTo FilePrintError
'''
'''     frmCmnDlg.caption = "Print"          ' Set form caption.
'''
'''     ' Determine default number of copies.
'''     If Copies = 0 Then
'''         frmCmnDlg.txtPrintCopies.text = "1"
'''     Else
'''         frmCmnDlg.txtPrintCopies.text = Str$(Copies)
'''     End If
'''
'''     ' Set default and cancel command buttons.
'''     frmCmnDlg.cmdPrintOK.Default = True
'''     frmCmnDlg.cmdPrintCancel.Cancel = True
'''
'''     ' Size and position Print container.
'''     frmCmnDlg.pctFilePrint.BorderStyle = 0
'''     frmCmnDlg.pctFilePrint.Visible = True
'''
'''     ' Size and center dialog.
'''     frmCmnDlg.Move frmCmnDlg.Left, frmCmnDlg.Top, frmCmnDlg.pctFilePrint.Width + 2,
frmCmnDlg.pctFilePrint.Height + 2
'''     frmCmnDlg.Move (Screen.Width - frmCmnDlg.Width) \ 2, ((Screen.Height - frmCmnDlg.Height) \
2) - 2
'''
'''     ' Set dialog colors.
'''     frmCmnDlg.forecolor = forecolor
'''     frmCmnDlg.backcolor = backcolor
'''     frmCmnDlg.pctFilePrint.forecolor = forecolor
'''     frmCmnDlg.pctFilePrint.backcolor = backcolor
'''     frmCmnDlg.lblPrintCopies.forecolor = forecolor
```

```
'''     frmCmnDlg.lblPrintCopies.backcolor = backcolor
'''     frmCmnDlg.txtPrintCopies.forecolor = forecolor
'''     frmCmnDlg.txtPrintCopies.backcolor = backcolor
'''     frmCmnDlg.txtPrintFile.forecolor = forecolor
'''     frmCmnDlg.txtPrintFile.backcolor = backcolor
'''     frmCmnDlg.fraPrintTarget.forecolor = forecolor
'''     frmCmnDlg.fraPrintTarget.backcolor = backcolor
'''     For i% = 0 To 3
'''         frmCmnDlg.optPrintTarget(i%).forecolor = forecolor
'''         frmCmnDlg.optPrintTarget(i%).backcolor = backcolor
'''     Next i%
'''     For i% = 0 To 1
'''         frmCmnDlg.optPrintAppend(i%).forecolor = forecolor
'''         frmCmnDlg.optPrintAppend(i%).backcolor = backcolor
'''     Next i%
'''     frmCmnDlg.lblPrintAppend.forecolor = forecolor
'''     frmCmnDlg.lblPrintAppend.backcolor = backcolor
'''     frmCmnDlg.cmdPrintOK.backcolor = backcolor
'''     frmCmnDlg.cmdPrintCancel.backcolor = backcolor
'''
'''     ' Display dialog modally.
'''     frmCmnDlg.Show 1
'''
'''     ' Determine if user canceled dialog.
'''     If frmCmnDlg.cmdPrintCancel.Tag <> "FALSE" Then
'''         Cancel = True
'''     ' If not, return number of copies to print.
'''     Else
'''         Cancel = False
'''         If Val(frmCmnDlg.txtPrintCopies.text) > 99 Then
'''             Copies = 99
'''         ElseIf Val(frmCmnDlg.txtPrintCopies.text) < 1 Then
'''             Copies = 1
'''         Else
'''             Copies = Val(frmCmnDlg.txtPrintCopies.text)
'''         End If
'''         frmCmnDlg.cmdPrintCancel.Tag = ""
'''     End If
'''
'''     ' Hide or unload dialog and return control to user's program.
'''     ' (Hide if user chose to preload form for performance.)
'''     If Left$(frmCmnDlg.Tag, 1) = "H" Then
'''         frmCmnDlg.pctFilePrint.Visible = False
'''         frmCmnDlg.Hide
'''     Else
'''         Unload frmCmnDlg
'''     End If
'''
'''     Exit Sub
'''
''''' Error handling routine.
'''FilePrintError:
'''     Select Case Err
'''     Case 7:                                     ' Out of memory.
'''          MsgBox "Out of memory.  Can't load dialog.", 0, "FindPrint"
'''          Cancel = True
'''          Exit Sub
'''     Case Else
'''          Resume Next
'''     End Select
'''End Sub
'''
''''' FileSave common dialog support routine
'''''
''''' Displays Save dialog which allows users to specify
```

```
'''' filename for subsequent file save operation.
'''' This procedure only provides the user interface and
'''' returns user input.  It does not actually carry out
'''' the corresponding action.
''''
'''' Parameters:
''''   FileName - returns the name (without path) of the
''''           file for the save operation.  To supply
''''           default filename in dialog, assign default
''''           to FileName then pass it to this procedure.
''''   PathName - returns the path (without filename) of
''''           the file for the save operation.  To supply
''''           default path in dialog, assign default to
''''           PathName then pass it to this procedure.
''''           Note, only pass a valid drive and path. Do
''''           not include a filename or file pattern.
''''   DefaultExt - sets the default search pattern for the
''''           File Listbox.  Default pattern when DefaultExt
''''           is null is "*.*".  To specify a different
''''           search pattern (i.e. "*.BAS"), assign new
''''           value to DefaultExt then pass it to this
''''           procedure.
''''   DialogTitle - sets the dialog title.  Default title
''''           when DialogTitle is null is "Save As".  To
''''           specify a different title (i.e. "Save My File"),
''''           assign new value to DialogTitle then pass it to
''''           this procedure.
''''   ForeColor - sets the dialog foreground color.  Does not affect
''''           SCREEN.ControlPanel color settings.
''''   BackColor - sets the dialog background color.  Does not affect
''''           SCREEN.ControlPanel color settings.
''''   Flags - unused.  Use this to customize dialog action if needed.
''''   Cancel - returns whether or not user pressed the dialog's Cancel
''''           button.  True (-1) means the user cancelled the dialog.
''''
'''Sub FileSave(filename As String, pathname As String, DefaultExt As String, DialogTitle As
String, forecolor As Integer, backcolor As Integer, Flags As Integer, Cancel As Integer)
'''    ' Set up error handling for option validation.
'''    On Local Error GoTo FileSaveError
'''
'''    ' Set form caption.
'''    If DialogTitle = "" Then
'''        frmCmnDlg.caption = "Save As"
'''    Else
'''        frmCmnDlg.caption = DialogTitle
'''    End If
'''    frmCmnDlg.Tag = frmCmnDlg.Tag + "SAVE"              ' Set form tag for common unload
procedure.
'''
'''    ' Determine search pattern for file listbox.
'''    If DefaultExt <> "" Then
'''        frmCmnDlg.filOpenList.Pattern = DefaultExt
'''    Else
'''        frmCmnDlg.filOpenList.Pattern = "*.*"
'''    End If
'''
'''    ' Determine default path.
'''    If pathname <> "" Then
'''        ' If the path ends with a backslash, remove it.
'''        If Right$(pathname, 1) = "\" Then
'''            pathname = Left$(pathname, Len(pathname) - 1)
'''        End If
'''        ' Set drive and path for file-system controls.
'''
'''        ' Set File listbox path.  If PathName is different
```

```
'''         ' than current path, PathChange event will be triggered
'''         ' which updates Drive listbox drive and Directory listbox path.
'''         frmCmnDlg.filOpenList.Path = pathname
'''     End If
'''     ' Display current path to the user.
'''     frmCmnDlg.lblOpenPath.caption = frmCmnDlg.filOpenList.Path
'''
'''     ' Determine default filename to display in edit field.
'''     If filename <> "" Then
'''         frmCmnDlg.txtOpenFile.text = UCase$(filename)
'''     Else
'''         frmCmnDlg.txtOpenFile.text = frmCmnDlg.filOpenList.Pattern
'''     End If
'''
'''     ' Set default and cancel command buttons.
'''     frmCmnDlg.cmdOpenOK.Default = True
'''     frmCmnDlg.cmdOpenCancel.Cancel = True
'''
'''     ' Size and position Open/Save container.
'''     frmCmnDlg.pctFileOpen.BorderStyle = 0
'''     frmCmnDlg.pctFileOpen.Visible = True
'''
'''     ' Size and center dialog.
'''     frmCmnDlg.Move frmCmnDlg.Left, frmCmnDlg.Top, frmCmnDlg.pctFileOpen.Width + 2,
frmCmnDlg.pctFileOpen.Height + 2
'''     frmCmnDlg.Move (Screen.Width - frmCmnDlg.Width) \ 2, ((Screen.Height - frmCmnDlg.Height) \
2) - 2
'''
'''     ' Set dialog colors.
'''     frmCmnDlg.forecolor = forecolor
'''     frmCmnDlg.backcolor = backcolor
'''     frmCmnDlg.pctFileOpen.forecolor = forecolor
'''     frmCmnDlg.pctFileOpen.backcolor = backcolor
'''     frmCmnDlg.lblOpenFile.forecolor = forecolor
'''     frmCmnDlg.lblOpenFile.backcolor = backcolor
'''     frmCmnDlg.txtOpenFile.forecolor = forecolor
'''     frmCmnDlg.txtOpenFile.backcolor = backcolor
'''     frmCmnDlg.lblOpenPath.forecolor = forecolor
'''     frmCmnDlg.lblOpenPath.backcolor = backcolor
'''     frmCmnDlg.filOpenList.forecolor = forecolor
'''     frmCmnDlg.filOpenList.backcolor = backcolor
'''     frmCmnDlg.drvOpenList.forecolor = forecolor
'''     frmCmnDlg.drvOpenList.backcolor = backcolor
'''     frmCmnDlg.dirOpenList.forecolor = forecolor
'''     frmCmnDlg.dirOpenList.backcolor = backcolor
'''     frmCmnDlg.cmdOpenOK.backcolor = backcolor
'''     frmCmnDlg.cmdOpenCancel.backcolor = backcolor
'''
'''     ' Display dialog modally.
'''     frmCmnDlg.Show 1
'''
'''     ' Determine if user canceled dialog.
'''     If frmCmnDlg.cmdOpenCancel.Tag <> "FALSE" Then
'''         Cancel = True
'''     ' If not, return FileName and PathName.
'''     Else
'''         Cancel = False
'''         filename = frmCmnDlg.txtOpenFile.text
'''         pathname = frmCmnDlg.filOpenList.Path
'''         frmCmnDlg.cmdOpenCancel.Tag = ""
'''     End If
'''
'''     ' Hide or unload dialog and return control to user's program.
'''     ' (Hide if user chose to preload form for performance.)
'''     If Left$(frmCmnDlg.Tag, 1) = "H" Then
```

```
'''         frmCmnDlg.pctFileOpen.Visible = False
'''         frmCmnDlg.Hide
'''         frmCmnDlg.Tag = "H"              ' Reset tag.
'''     Else
'''         Unload frmCmnDlg
'''     End If
'''
'''     Exit Sub
'''
'''' Option error handling routine.
'''' Ignore errors here and let dialog's controls
'''' handle the errors.
'''FileSaveError:
'''     Select Case Err
'''     Case 7:                                          ' Out of memory.
'''          MsgBox "Out of memory.  Can't load dialog.", 0, "FileSave"
'''          Cancel = True
'''          Exit Sub
'''     Case Else
'''          Resume Next
'''     End Select
'''End Sub
```

H.3 Visual Basic Code Module: Const.bas

```
Attribute VB_Name = "Const"
'Microsoft Visual Basic constants. Unused (by SECPOP90) constants are
'commented out to save memory.

' ---------------------------------------------------------------------
' Visual Basic for MS-DOS Constant Include File
'
' Include file that contains constant definitions
' for enumerated form or control property values and
' event procedure, method, and function parameter values.
'
' This file can be included and used as is in your applications.
' Each constant definition reduces the amount of memory available
' for your application, however, so for best results, include only
' those constant definitions you plan to use in your application.
' In addition, some of the constant definitions below may conflict with
' variable definitions in your existing programs.
'
' Some constants below are commented out because they
' have duplicates (for example, NONE appears in several
' places).
'
' Copyright (C) 1982-1992 Microsoft Corporation
'
' You have a royalty-free right to use, modify, reproduce
' and distribute the sample applications and toolkits provided with
' Visual Basic for MS-DOS (and/or any modified version)
' in any way you find useful, provided that you agree that
' Microsoft has no warranty, obligations or liability for
' any of the sample applications or toolkits.
' ---------------------------------------------------------------------


'''' -------
'''' General
'''' -------
'''
'''' Booleans
'''CONST TRUE  = -1
'''CONST FALSE =  0
'''
'''' ----------------
'''' Event parameters
'''' ----------------
'''
'''' Button and Shift (KeyDown, KeyUp, MouseDown, MouseMove, MouseUp)
''''CONST SHIFT_MASK = 1
''''CONST CTRL_MASK = 2
''''CONST ALT_MASK = 4
''''CONST LEFT_BUTTON = 1
''''CONST RIGHT_BUTTON = 2
'''
'''' KeyCode (KeyDown, KeyUp)
''''CONST KEY_BACK = 8
''''CONST KEY_TAB = 9
''''CONST KEY_CLEAR = 12
'''Const KEY_RETURN = 13           ' Enter key
''''CONST KEY_SHIFT = 16
''''CONST KEY_CONTROL = 17
''''CONST KEY_MENU = 18             ' Alt key
''''CONST KEY_PAUSE = 19
''''CONST KEY_CAPITAL = 20          ' Caps lock key
```

```
''''CONST KEY_ESCAPE = 27
''' Const KEY_SPACE = 32
''''CONST KEY_PRIOR = 33            ' Page up key
''''CONST KEY_NEXT = 34             ' Page down key
''''CONST KEY_END = 35
''''CONST KEY_HOME = 36
''''CONST KEY_LEFT = 37
'''Const KEY_UP = 38
''''CONST KEY_RIGHT = 39
'''Const KEY_DOWN = 40
''''CONST KEY_SELECT = 41
''''CONST KEY_PRINT = 42
''''CONST KEY_EXECUTE = 43
''''CONST KEY_SNAPSHOT = 44
''''CONST KEY_INSERT = 45
''''CONST KEY_DELETE = 127          ' Delete key returns 46 in Visual
''''                                    ' Basic for Windows.
''''CONST KEY_HELP = 47
'''
'''' KeyCode parameter in KeyDown and KeyUp event procedures
'''' returns the same value as KeyAscii in the KeyPress event
'''' procedure for keys corresponding to ASCII printable
'''' characters (A-Z, a-z, 0-9, ~,[,],{,},+,=, etc). Return
'''' values will be the ASCII value of the character (see
'''' ASCII Character Codes topic in Help). Extended ASCII
'''' characters can be returned via KeyCode and KeyAscii
'''' by holding down the ALT key, entering the ASCII number
'''' of the extended ASCII character, then releasing the
'''' ALT key. Note, the NumLock key must be on if the numeric
'''' keypad is used.
'''
''''CONST KEY_NUMPAD0 = 48
''''CONST KEY_NUMPAD1 = 49
''''CONST KEY_NUMPAD2 = 50
''''CONST KEY_NUMPAD3 = 51
''''CONST KEY_NUMPAD4 = 52
''''CONST KEY_NUMPAD5 = 53
''''CONST KEY_NUMPAD6 = 54
''''CONST KEY_NUMPAD7 = 55
''''CONST KEY_NUMPAD8 = 56
''''CONST KEY_NUMPAD9 = 57
''''CONST KEY_MULTIPLY = 42
''''CONST KEY_ADD = 43
''''CONST KEY_SUBTRACT = 45
''''CONST KEY_DECIMAL = 46
''''CONST KEY_DIVIDE = 47
''''CONST KEY_F1 = 112
''''CONST KEY_F2 = 113
''''CONST KEY_F3 = 114
''''CONST KEY_F4 = 115
''''CONST KEY_F5 = 116
''''CONST KEY_F6 = 117
''''CONST KEY_F7 = 118
''''CONST KEY_F8 = 119
''''CONST KEY_F9 = 120
''''CONST KEY_F10 = 121
''''CONST KEY_F11 = 122
''''CONST KEY_F12 = 123
''''CONST KEY_NUMLOCK = 144
''''CONST KEY_SCRLOCK = 145
'''
'''' State (DragOver)
''''CONST ENTER = 0
''''CONST LEAVE = 1
''''CONST OVER = 2
```

```
'''
'''' -------------------
'''' Function parameters
'''' -------------------
'''
'''' MSGBOX parameters
'''Const vbOK = 0                          ' OK button only
''''CONST vbOKCancel = 1                    ' OK and Cancel buttons
''''CONST vbAbortRetryIgnore = 2            ' Abort, Retry, and Ignore buttons
'''Const vbYesNoCancel = 3                 ' Yes, No, and Cancel buttons
'''Const vbYesNo = 4                       ' Yes and No buttons
''''CONST vbRetryCancel = 5                 ' Retry and Cancel buttons
'''
'''Const vbDefaultButton1 = 0                 ' First button is default
'''Const vbDefaultButton2 = 256               ' Second button is default
''''CONST vbDefaultButton3 = 512               ' Third button is default
'''
'''' MSGBOX return values
''''CONST IDOK = 1                          ' OK button pressed
'''Const IDCANCEL = 2                      ' Cancel button pressed
''''CONST IDABORT = 3                       ' Abort button pressed
''''CONST IDRETRY = 4                       ' Retry button pressed
''''CONST IDIGNORE = 5                      ' Ignore button pressed
'''Const vbYes = 6                         ' Yes button pressed
'''Const vbNo = 7                          ' No button pressed
'''
'''
'''' -----------------
'''' Method parameters
'''' -----------------
'''
'''' DRAG (controls)
''''CONST CANCEL_DRAG = 0
''''CONST BEGIN_DRAG = 1
''''CONST END_DRAG = 2
'''
'''' SHOW (form)
'''Const MODELESS = 0
'''Const  = 1
'''
'''' ---------------
'''' Property values
'''' ---------------
'''
'''' Alignment (label)
''''CONST LEFT_JUSTIFY = 0                  ' 0 - Left Justify
''''CONST RIGHT_JUSTIFY = 1                 ' 1 - Right Justify
''''CONST CENTER = 2                        ' 2 - Center
'''
'''' BackColor, ForeColor (form, controls)
'''
'''Const BLACK = 0
'''Const BLUE = 1
''''CONST GREEN = 2
'''Const CYAN = 3
''''CONST RED = 4
'''Const MAGENTA = 5
''''CONST BROWN = 6
'''Const WHITE = 7
''''CONST GRAY = 8
''''CONST BRIGHT_BLUE = 9
''''CONST BRIGHT_GREEN = 10
'''Const BRIGHT_CYAN = 11
''''CONST BRIGHT_RED = 12
''''CONST PINK = 13
```

```
''''CONST YELLOW = 14
'''Const BRIGHT_WHITE = 15
'''
'''' BorderStyle (form)
''''CONST NONE = 0                          ' 0 - None
''''CONST FIXED_SINGLE = 1                  ' 1 - Fixed Single
''''CONST SIZABLE_SINGLE = 2                ' 2 - Sizable Single
''''CONST FIXED_DOUBLE = 3                  ' 3 - Fixed Double
''''CONST SIZABLE_DOUBLE = 4                ' 4 - Sizable Double
''''CONST FIXED_SOLID = 5                   ' 5 - Fixed Solid
''''CONST SIZABLE_SOLID = 6                 ' 6 - Sizable Solid
'''
'''' BorderStyle (label, picture box, text box)
''''CONST NONE = 0                          ' 0 - None
''''CONST SINGLE_LINE = 1                   ' 1 - Single Line
''''CONST DOUBLE_LINE = 2                   ' 2 - Double Line
''''                                        ' (label and picture box only)
'''
'''' DragMode (controls)
''''CONST MANUAL = 0                        ' 0 - Manual
''''CONST AUTOMATIC = 1                     ' 1 - Automatic
'''
'''' FormType (form - Multiple Document Interface (MDI) vs Single Document
'''' Interface (SDI) applications)
''''CONST NORMAL = 0                        ' 0 - Normal
''''                                        ' (Normal form in SDI
''''                                        ' applications, child form in
''''                                        ' MDI applications)
''''CONST MDI = 1                           ' 1 - MDI (Container form in
''''                                        ' MDI application)
'''
'''' MousePointer (form, controls)
''''CONST DEFAULT = 0                       ' 0 - Default (Same
''''                                        ' MousePointer as container
''''                                        ' object's MousePointer)
''''CONST BLOCK = 1                         ' 1 - Block (ASCII 219)
''''CONST CROSSHAIR = 2                     ' 2 - Cross (ASCII 197)
''''CONST IBEAM = 3                         ' 3 - I-Beam (ASCII 73)
''''CONST ICON = 4                          ' 4 - Icon (ASCII 002)
''''CONST SIZE_POINTER = 5                  ' 5 - Size (ASCII 015)
''''CONST LEFT_ARROW = 6                    ' 6 - Left Arrow (ASCII 027)
''''CONST SIZE_N_S = 7                      ' 7 - Size North South
''''                                        ' (ASCII 018)
''''CONST RIGHT_ARROW = 8                   ' 8 - Right Arrow (ASCII 026)
''''CONST SIZE_W_E = 9                      ' 9 - Size West East
''''                                        ' (ASCII 029)
''''CONST UP_ARROW = 10                     ' 10 - Up Arrow (ASCII 024)
''''CONST HOURGLASS = 11                    ' 11 - Hourglass (ASCII 088)
''''CONST DOWN_ARROW = 12                   ' 12 - Down Arrow (ASCII 025)
'''
'''' ScrollBar (text box)
''''CONST NONE = 0                        ' 0 - None
''''CONST HORIZONTAL = 1                    ' 1 - Horizontal
''''CONST VERTICAL = 2                      ' 2 - Vertical
''''CONST BOTH = 3                          ' 3 - Both
'''
'''' Value (check box)
''''CONST UNCHECKED = 0                     ' 0 - Unchecked
''''CONST CHECKED = 1                       ' 1 - Checked
''''CONST GRAYED = 2                        ' 2 - Grayed
'''
'''' WindowState (form)
''''CONST NORMAL = 0                        ' 0 - Normal
''''CONST MINIMIZED = 1                     ' 1 - Minimized
''''CONST MAXIMIZED = 2                     ' 2 - Maximized
```

```
'''
'''' SCREEN.ControlPanel array elements
''''CONST ACCESSKEY_FORECOLOR = 0            ' Access key foreground
''''                                         ' color (0-15).
'''Const ACTIVE_BORDER_BACKCOLOR = 1        ' Active border background
''''                                         ' color (0-15).
'''Const ACTIVE_BORDER_FORECOLOR = 2        ' Active border foreground
''''                                         ' color (0-15).
'''Const ACTIVE_WINDOW_SHADOW = 3           ' Active window shadow
''''                                         ' effect (Boolean).
''''CONST COMBUTTON_FORECOLOR = 4            ' Command button foreground
''''                                         ' color (0-15).
''''CONST DESKTOP_BACKCOLOR = 5              ' Desktop background
''''                                         ' color (0-15).
''''CONST DESKTOP_FORECOLOR = 6              ' Desktop foreground
''''                                         ' color (0-15).
'''Const DESKTOP_PATTERN = 7                 ' Desktop fill pattern
''''                                         ' (ASCII 0-255).
''''CONST DISABLED_ITEM_FORECOLOR = 8        ' Disabled menu/dialog item
''''                                         ' foreground color (0-15).
'''Const MENU_BACKCOLOR = 9                 ' Menu background color (0-15).
'''Const MENU_FORECOLOR = 10                ' Menu foreground color (0-15).
''''CONST MENU_SELECTED_BACKCOLOR = 11       ' Menu selected item background
''''                                         ' color (0-15).
''''CONST MENU_SELECTED_FORECOLOR = 12       ' Menu selected item foreground
''''                                         ' color (0-15).
'''Const SCROLLBAR_BACKCOLOR = 13           ' Scrollbar background
''''                                         ' color (0-15).
'''Const SCROLLBAR_FORECOLOR = 14           ' Scrollbar foreground
''''                                         ' color (0-15).
''''CONST THREE_D = 15                       ' 3-D effect for controls with
''''                                         ' borders (Boolean).
'''Const TITLEBAR_BACKCOLOR = 16            ' Titlebar background
''''                                         ' color (0-15).
'''Const TITLEBAR_FORECOLOR = 17            ' Titlebar foreground
''''                                         ' color (0-15).
'''

'11/30/01: SECPOP90 specific constants have been moved to Secpop90.bas.

'Public Const number_of_segments = 16
'Public Const max_econ_regions = 99
'Public Const min_econ_regions = 2
'Public Const max_number_of_radii = 35
'Public Const max_county_name_length = 27
'Public Const record_length = 12
'Public Const longitude_offset = 91993
'Public Const latitude_offset = 16610&
'Public Const number_of_records = 6660337
'Public Const number_of_counties = 3111
'Public Const miles_to_kilometers = 1.60934721869
'Public Const kilometers_to_miles = 0.6213699495
'Public Const max_layers = 10
'Public Const seconds_in_a_day = 86400
'ssPublic Const pi = 3.14159265358979
```

H.4 Visual Basic Code Module: FormsPosition.bas

```
Attribute VB_Name = "FormsPosition"
Option Explicit

Public Sub SetFormPosition()

    Dim FormsTop As Integer
    Dim FormsLeft As Integer

    FormsTop = 50
    FormsLeft = 50
'    Me.Top = FormsTop
'    Me.Left = FormsLeft

End Sub
```

H.5 Visual Basic Code Module: GridUtils.bas

```
Attribute VB_Name = "GridUtils"
''''''Option Explicit
''''''
''''''Public Function StoreUserData(Bookmark As Variant, _
''''''                              Col As Integer, _
''''''                              Userval As Variant, _
''''''                              gridarray() As String, _
''''''                              maxrow As Integer, _
''''''                              maxcol As Integer) As Boolean
''''''
'''''''' StoreUserData is called from UnboundWriteData to
'''''''' write the newly changed information in the grid to
'''''''' the array.  This function is called once for each
'''''''' field that needs to be stored.  The grid cell that
'''''''' this function is called to store is specified in
'''''''' the same way as in GetUserData.
''''''
''''''    Dim Index As Long
''''''
''''''    ' Figure out which row the bookmark refers to
''''''    Index = IndexFromBookmark(Bookmark, False, maxrow)
''''''    If Index < 0 Or Index >= maxrow Or Col < 0 Or _
''''''       Col >= maxcol Then
''''''        ' Cell position is invalid, so just return null
''''''        ' to indicate failure
''''''        StoreUserData = False
''''''    Else
''''''        StoreUserData = True
''''''        gridarray(Col, Index) = Userval
''''''    End If
''''''
''''''End Function
''''''
''''''Public Sub TDBGridUnboundWriteData(ByVal RowBuf As TrueOleDBGrid70.RowBuffer, _
''''''                                   WriteLocation As Variant, _
''''''                                   gridarray() As String, _
''''''                                   maxrow As Integer, _
''''''                                   maxcol As Integer)
''''''
'''''''' RowBuf is the row buffer from which you retrieve the
'''''''' data to be stored in the data source.  Since only one
'''''''' row can be updated at a time, RowBuf.RowCount always
'''''''' equals 1 here.  If a particular column in the row has
'''''''' not been changed, its element in the RowBuf.Value
'''''''' array will be Null.
''''''
'''''''' WriteLocation is a bookmark that identifies the row
'''''''' to be updated.
''''''
'''''''' Assume that a function StoreUserData(bookmark, col,
'''''''' value) takes a row bookmark, a column index, and a
'''''''' variant with the appropriate data to be stored in an
'''''''' array or database.  The function returns True if the
'''''''' data is acceptable and can be stored, False otherwise.
''''''
'''''''' Loop over all the columns of the row, storing non-Null
'''''''' values
''''''
''''''    Dim i As Integer
''''''
''''''    For i = 0 To RowBuf.ColumnCount - 1
''''''        If Not IsNull(RowBuf.Value(0, i)) Then
```

```
''''''            If Not StoreUserData(WriteLocation, i, _
''''''                RowBuf.Value(0, i), gridarray, maxrow, maxcol) Then
''''''                RowBuf.RowCount = 0 ' Tell the grid the
''''''                                    ' update failed. so
''''''                Exit Sub            ' so exit the event.
''''''
''''''            End If
''''''        End If
''''''    Next i
''''''
''''''End Sub
''''''
''''''''Public Function GetNewBookmark(maxrow As Long) As Variant
''''''''Public Function GetNewBookmark(maxrow As Integer) As Variant
'''''''' GetNewBookmark is called when we need to create a
'''''''' bookmark for a newly added record (row).
''''''''
'''''''' Reserve space for the new row in the array
''''''''    ReDim Preserve gridarray(0 To maxcol - 1, 0 To maxrow)
''''''''
'''''''' Since the row index of the last record is (MaxRow - 1),
'''''''' the next available index for the row to be added is
'''''''' (MaxRow), so create a bookmark for the new row using
'''''''' MaxRow as an index.
''''''''    GetNewBookmark = MakeBookmark(maxrow)
''''''''
'''''''' Increment the number of rows in our dataset
''''''''    maxrow = maxrow + 1
''''''''
'''''''' Calibrate the scroll bars based on the new dataset
'''''''' size.
''''''''    TDBGrid1.ApproxCount = maxrow
''''''''End Function
''''''
''''''Public Sub TDBGridUnboundReadData(ByVal RowBuf As TrueOleDBGrid70.RowBuffer, _
''''''                                  StartLocation As Variant, _
''''''                                  ByVal ReadPriorRows As Boolean, _
''''''                                  gridarray() As String, _
''''''                                  maxrow As Integer, _
''''''                                  maxcol As Integer)
''''''
'''''' RowBuf is the row buffer where you place the data and
'''''' the bookmarks for the rows that the grid is requesting
'''''' to display.  It will also hold the number of rows that
'''''' were successfully supplied to the grid.
''''''
'''''' StartLocation is a bookmark which specifies the row
'''''' before or after the desired data, depending on the
'''''' value of ReadPriorRows.  If we are reading rows after
'''''' StartLocation (ReadPriorRows = False), then the first
'''''' row of data the grid wants is the row after
'''''' StartLocation, and if we are reading rows before
'''''' StartLocation (ReadPriorRows = True) then the first
'''''' row of data the grid wants is the row before
'''''' StartLocation.
''''''
'''''' ReadPriorRows is a boolean value indicating whether
'''''' we are reading rows before (ReadPriorRows = True) or
'''''' after (ReadPriorRows = False) StartLocation.
''''''
''''''    Dim Bookmark As Variant
''''''    Dim i As Long, RelPos As Long
''''''    Dim j As Integer, RowsFetched As Integer
''''''
'''''' Get a bookmark for the start location
```

```
''''''    Bookmark = StartLocation
''''''
''''''    If ReadPriorRows Then
''''''        RelPos = -1 ' Requesting data in rows prior to
''''''                    ' StartLocation
''''''    Else
''''''        RelPos = 1  ' Requesting data in rows after
''''''                    ' StartLocation
''''''    End If
''''''
''''''    RowsFetched = 0
''''''    For i = 0 To RowBuf.RowCount - 1
''''''        ' Get the bookmark of the next available row
''''''        Bookmark = GetRelativeBookmark(Bookmark, RelPos, maxrow)
''''''
''''''        ' If the next row is BOF or EOF, then stop
''''''        ' fetching and return any rows fetched up to this
''''''        ' point.
''''''        If IsNull(Bookmark) Then Exit For
''''''
''''''        ' Place the record data into the row buffer
''''''        For j = 0 To RowBuf.ColumnCount - 1
''''''            RowBuf.Value(i, j) = GetUserData(Bookmark, j, gridarray(), _
''''''                                             maxrow, maxcol)
''''''        Next j
''''''
''''''        ' Set the bookmark for the row
''''''        RowBuf.Bookmark(i) = Bookmark
''''''
''''''        ' Increment the count of fetched rows
''''''        RowsFetched = RowsFetched + 1
''''''    Next i
''''''
'''''' Tell the grid how many rows we fetched
''''''    RowBuf.RowCount = RowsFetched
''''''
''''''End Sub
''''''
''''''Public Function MakeBookmark(Index As Long) As Variant
'''''' This support function is used only by the remaining
'''''' support functions.  It is not used directly by the
'''''' unbound events.  It is a good idea to create a
'''''' MakeBookmark function such that all bookmarks can be
'''''' created in the same way.  Thus the method by which
'''''' bookmarks are created is consistent and easy to
'''''' modify.  This function creates a bookmark when given
'''''' an array row index.
''''''
'''''' Since we have data stored in an array, we will just
'''''' use the array index as our bookmark.  We will convert
'''''' it to a string first, using the CStr function.
''''''
''''''    MakeBookmark = CStr(Index)
''''''
''''''End Function
''''''
''''''Public Function IndexFromBookmark(Bookmark As Variant, _
''''''        ReadPriorRows As Boolean, maxrow As Integer) As Long
''''''
'''''' This support function is used only by the remaining
'''''' support functions.  It is not used directly by the
'''''' unbound events.
''''''
'''''' This function is the inverse of MakeBookmark.  Given
'''''' a bookmark, IndexFromBookmark returns the row index
```

```
''''''' that the given bookmark refers to.  If the given
''''''' bookmark is Null, then it refers to BOF or EOF.  In
''''''' such a case, we need to use ReadPriorRows to
''''''' distinguish between the two.  If ReadPriorRows = True,
''''''' the grid is requesting rows before the current
''''''' location, so we must be at EOF, because no rows exist
''''''' before BOF.  Conversely, if ReadPriorRows = False,
''''''' we must be at BOF.
'''''
'''''    Dim Index As Long
'''''
'''''    If IsNull(Bookmark) Then
'''''        If ReadPriorRows Then
'''''            ' Bookmark refers to EOF.  Since (MaxRow - 1)
'''''            ' is the index of the last record, we can use
'''''            ' an index of (MaxRow) to represent EOF.
'''''            IndexFromBookmark = maxrow
'''''        Else
'''''            ' Bookmark refers to BOF.  Since 0 is the
'''''            ' index of the first record, we can use an
'''''            ' index of -1 to represent BOF.
'''''            IndexFromBookmark = -1
'''''        End If
'''''    Else
'''''        ' Convert string to long integer
'''''        Index = Val(Bookmark)
'''''
'''''        ' Check to see if the row index is valid:
'''''        '  (0 <= Index < MaxRow).
'''''        ' If not, set it to a large negative number to
'''''        ' indicate that the bookmark is invalid.
'''''        If Index < 0 Or Index >= maxrow Then Index = -9999
'''''        IndexFromBookmark = Index
'''''    End If
'''''End Function
'''''
'''''Public Function GetRelativeBookmark(Bookmark As Variant, _
'''''                                    RelPos As Long, _
'''''                                    maxrow As Integer) As Variant
'''''
''''''' GetRelativeBookmark is used to get a bookmark for a
''''''' row that is a given number of rows away from the given
''''''' row.  This specific example will always use either -1
''''''' or +1 for a relative position, since we will always be
''''''' retrieving either the row previous to the current one,
''''''' or the row following the current one.
'''''
''''''' IndexFromBookmark expects a Bookmark and a Boolean
''''''' value: this Boolean value is True if the next row to
''''''' read is before the current one [in this case,
''''''' (RelPos < 0) is True], or False if the next row to
''''''' read is after the current one [(RelPos < 0) is False].
''''''' This is necessary to distinguish between BOF and EOF
''''''' in the IndexFromBookmark function if our bookmark is
''''''' Null.  Once we get the correct row index from
''''''' IndexFromBookmark, we simply add RelPos to it to get
''''''' the desired row index and create a bookmark using
''''''' that index.
'''''
'''''    Dim Index As Long
'''''
'''''    Index = IndexFromBookmark(Bookmark, RelPos < 0, maxrow) + RelPos
'''''    If Index < 0 Or Index >= maxrow Then
'''''        ' Index refers to a row before the first or after
'''''        ' the last, so just return Null.
```

```
''''''        GetRelativeBookmark = Null
''''''    Else
''''''        GetRelativeBookmark = MakeBookmark(Index)
''''''    End If
''''''End Function
''''''
''''''Public Function GetUserData(Bookmark As Variant, _
''''''                            Col As Integer, _
''''''                            gridarray() As String, _
''''''                            maxrow As Integer, _
''''''                            maxcol As Integer) As Variant
''''''
'''''' In this example, GetUserData is called by
'''''' UnboundReadData to ask the user what data should be
'''''' displayed in a specific cell in the grid.  The grid
'''''' row the cell is in is the one referred to by the
'''''' Bookmark parameter, and the column it is in it given
'''''' by the Col parameter.  GetUserData is called on a
'''''' cell-by-cell basis.
''''''
''''''    Dim Index As Long
''''''
'''''' Figure out which row the bookmark refers to
''''''    Index = IndexFromBookmark(Bookmark, False, maxrow)
''''''
''''''    If Index < 0 Or Index >= maxrow Or _
''''''       Col < 0 Or Col >= maxcol Then
''''''        ' Cell position is invalid, so just return null
''''''        ' to indicate failure
''''''        GetUserData = Null
''''''    Else
''''''        GetUserData = gridarray(Col, Index)
''''''         Print #22, "col = " & Col & " row = " & Index & "  " & gridarray(Col, Index)
''''''    End If
''''''End Function
''''''
''''''
''''''
''''''Public Sub TDBGridUnboundGetRelativeBookmark(StartLocation As Variant, _
''''''                                             ByVal OFFSET As Long, _
''''''                                             NewLocation As Variant, _
''''''                                             ApproximatePosition As Long, _
''''''                                             maxrow As Integer)
''''''
''''''   Dim Index As Long
''''''
'''''' If StartLocation is NULL, then we are positioning
'''''' from either BOF or EOF.  Once we determine the
'''''' correct index for StartLocation, we will simply add
'''''' the offset to get the correct destination row.
''''''    If IsNull(StartLocation) Then
''''''        If OFFSET < 0 Then
''''''            ' The row to position to is before the current
''''''            ' one, therefore we must be at EOF (There are
''''''            ' no rows before BOF).  Since (MaxRow-1) is
''''''            ' the index of the last record, we can use an
''''''            ' index of (MaxRow) to represent EOF.
''''''            Index = maxrow + OFFSET
''''''        Else
''''''            ' The row to position to is after the current
''''''            ' one, therefore we must be at BOF (There are
''''''            ' no rows after EOF).  Since 0 is the index of
''''''            ' the first record, we can use an index of -1
''''''            ' to represent BOF.
''''''            Index = -1 + OFFSET
```

```
''''''        End If
''''''    Else
''''''        ' We are not at BOF or EOF, so just get the index
''''''        ' of the current row and add the offset to it.
''''''        Index = Val(StartLocation) + OFFSET
''''''    End If
''''''
'''''' Is the new Index valid?
''''''    If Index >= 0 And Index < maxrow Then
''''''       ' Here, 0 <= Index < MaxRow, so the new location is
''''''       ' valid.
''''''       ApproximatePosition = Index
''''''       NewLocation = MakeBookmark(Index)
''''''    Else
''''''       ' The new index is out-of-bounds -- the new location
''''''       ' refers to a position before the first record or
''''''       ' after the last one.  Thus the index is invalid,
''''''       ' and we return Null.
''''''       NewLocation = Null
''''''    End If
''''''
''''''End Sub
```

H.6 Visual Basic Code Module: InIt.bas

```
Attribute VB_Name = "InIt"
'********************************************************************************
'  Module:  Init
'
'  Project: SECPOP
'
'  Description: This module contains initialization statements to be called on
'               program initialtion
'
'  Modification History
'  Date      By   Description
'  --------  ---  -----------------------------------------------------------
'  20020303 cwm  Changed inital value of number_of_radii variable.
'  20020310 ss   Added declarations needed to show the browse window
'                Added the code to find all available census files and
'                show them on the setup form
'********************************************************************************

Option Explicit

Public Type economic_data
    region_area As Long
    region_frmfrc As Single
    region_dpf As Single
    region_asfp As Single
    region_vfrm As Single
    region_vnfrm As Single
End Type

Public Type Block_Data
    longit As Integer
    latit As Integer
    pop As Integer
    area As Long
    ccode As Integer
End Type

Public Const MainTitle = "SECPOP2000 - SECtor POPulation and Economic Estimator - Version 3.11"

'SECPOP specific constants.

Public Const number_of_segments = 16
Public Const max_econ_regions = 99
Public Const min_econ_regions = 2
Public Const max_number_of_radii = 35
Public Const max_county_name_length = 27
Public Const record_length = 12
Public Const longitude_offset = 91993
Public Const latitude_offset = 166104
' cwm Public Const number_of_records = 6660337
Public Const Number_of_Counties = 3200
Public Const miles_to_kilometers = 1.60934721869
Public Const kilometers_to_miles = 0.6213699495
Public Const max_layers = 10
Public Const seconds_in_a_day = 86400
Public Const pi = 3.14159265358979
Public Const KEY_RETURN = 13

'Define all of the dynamic arrays used in SECPOP.

Public radial_distance(max_number_of_radii) As Single
Public raddis(max_number_of_radii) As Single
```

```
Public regional_radii(max_number_of_radii) As Single
Public population(max_number_of_radii) As Long
Public population_threshold(max_number_of_radii) As Long

Public region_index(max_number_of_radii, _
   number_of_segments) As Integer
Public econ_data(max_econ_regions) As economic_data

Public sector_population(number_of_segments, _
   max_number_of_radii) As Long
Public sector_area(number_of_segments, _
   max_number_of_radii) As Long
Public sector_frclnd(number_of_segments, _
   max_number_of_radii) As Single

Public county_state(Number_of_Counties) As String * 2   'defining # of characters as two since it
won't change
Public county_name(Number_of_Counties) As String * max_county_name_length
Public county_frclnd(Number_of_Counties) As Single
Public county_frmfrc(Number_of_Counties) As Single
Public county_dpf(Number_of_Counties) As Single
Public county_asfp(Number_of_Counties) As Single
Public county_vfrm(Number_of_Counties) As Single
Public county_vnfrm(Number_of_Counties) As Single

Public directions(number_of_segments) As String

Public layer_number(max_layers) As Integer
Public data_flag(max_layers) As Integer
Public display_flag(max_layers) As Integer
Public active_layer(max_layers) As Integer
Public layer_name(max_layers) As String
Public pts_color(max_layers) As Integer
Public pts_type(max_layers) As Integer
Public pts_size(max_layers) As Integer
Public pts_mode(max_layers) As Integer
Public lines_color(max_layers) As Integer
Public lines_type(max_layers) As Integer
Public lines_size(max_layers) As Integer
Public lines_mode(max_layers) As Integer
Public polyg_color(max_layers) As Integer
Public polyg_type(max_layers) As Integer
Public polyg_size(max_layers) As Integer
Public polyg_mode(max_layers) As Integer
Public text_color(max_layers) As Integer
Public text_type(max_layers) As Integer
Public text_size(max_layers) As Integer
Public text_mode(max_layers) As Integer
Public BaseDirectory As String

Public radial_area(max_number_of_radii) As Single

Public Number_of_Records As Long 'cwm
Public leave_calculation As Integer
Public rec As Block_Data    'cwm
Public xlon As Single, ylat As Single
Public ipop As Integer, area As Long, county_code As Integer
Public idir As Integer, irad As Integer, ibyte As Long

Public number_of_radii As Integer, numrad As Integer
Public Population_multiplier As Single
Public ScaleD As Single

Public slon As Single, slat As Single
Public dismax As Single, sitid$
```

```
Public dpdlat As Single, dpdlon As Single, sdpdla As Single
Public bndryw As Single, bndrye As Single
Public bndryn As Single, bndrys As Single
Public aminla As Single
Public DgToRd As Single 'qqq
Public BaseDir As String 'qqq
'Public region_index() As Integer,
Public number_econ_regions As Integer
'Public econ_data() As economic_data
Public table_type As Integer
Public x_position As Integer
Public y_position As Integer

Public InitSiteFilePath As String
Public InitProblemFilePath As String
Public InitOutputFilePath As String

Public InitSiteFilePathChanged As Boolean
Public InitProblemFilePathChanged As Boolean
Public InitOutputFilePathChanged As Boolean

'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Copyright ©1996-2002 VBnet, Randy Birch, All Rights Reserved.
' Some pages may also contain other copyrights by the author.
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' You are free to use this code within your own applications,
' but you are expressly forbidden from selling or otherwise
' distributing this source code without prior written consent.
' This includes both posting free demo projects made from this
' code as well as reproducing the code in text or html format.
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

Public Type BROWSEINFO
   hOwner          As Long
   pidlRoot        As Long
   pszDisplayName  As String
   lpszTitle       As String
   ulFlags         As Long
   lpfn            As Long
   lParam          As Long
   iImage          As Long
End Type

Public Const BIF_RETURNONLYFSDIRS = &H1
Public Const BIF_DONTGOBELOWDOMAIN = &H2
Public Const BIF_STATUSTEXT = &H4
Public Const BIF_RETURNFSANCESTORS = &H8
Public Const BIF_BROWSEFORCOMPUTER = &H1000
Public Const BIF_BROWSEFORPRINTER = &H2000
Public Const MAX_PATH = 260

Public Declare Function SHGetPathFromIDList Lib "shell32" _
   Alias "SHGetPathFromIDListA" _
  (ByVal pidl As Long, _
   ByVal pszPath As String) As Long

Public Declare Function SHBrowseForFolder Lib "shell32" _
   Alias "SHBrowseForFolderA" _
  (lpBrowseInfo As BROWSEINFO) As Long

Public Declare Sub CoTaskMemFree Lib "ole32" _
   (ByVal pv As Long)

'--end block--'
```

```
Public Sub Initializations()

    Dim i As Integer
    Dim j As Integer

    Dim strCensusFile As String
    Dim strCountyFile As String
    Dim strFname As String
    Dim strFnames(50) As String

    'Define varibles local to this event procedure.

    Dim filenum As Integer, TempString As String


    'Set up error handler for errors in the configuration
    'file, and load in the various parameters from the
    'setup file into the setup form.  These parameters
    'are set by the user in the exit menu under setup.
    'Because input will not interact directly with forms,
    'tempstring is used as a buffer for file data.

    'Load in the forms that we need to keep permanently in memory.

    Load frmProblem_Data
    frmMain.mnuSave_ProblemCom.Enabled = False
    frmMain.mnuSave_As_ProblemCom.Enabled = False


    Load frmSite_Data
    frmMain.mnuSave_Sitecom.Enabled = False
    frmMain.mnuSave_as_Sitecom.Enabled = False


    Load frmSetup
    InitSiteFilePathChanged = True
    InitProblemFilePathChanged = True
    InitOutputFilePathChanged = True

''    frmDisclaimer.Show
'    On Local Error GoTo configerr
'
'    'Check and see if config file exists.
'    tempstring = App.path & "\SECPOP.CFG"
'    If Len(Dir$(tempstring)) <> 0 Then
''    If Len(Dir$("SECPOP.CFG")) <> 0 Then
'
'        'Open file and read in the data from it.
'
'        filenum = FreeFile
''        Open "SECPOP.CFG" For Input As filenum
'        Open tempstring For Input As filenum
'        Input #filenum, tempstring
'        frmSetup.txtSite_Path.text = tempstring
'        Input #filenum, tempstring
'        frmSetup.txtProblem_path.text = tempstring
'        Input #filenum, tempstring
'        frmSetup.txtOutput_path.text = tempstring
'        Input #filenum, tempstring
'        frmSetup.txtData_path.text = tempstring
'        Input #filenum, tempstring
'        If (tempstring = "MACCS") Then
'            frmSetup.optMACCS.Value = True
'        Else
```

```
'            frmSetup.optCSV.Value = True
'        End If
'        Input #filenum, tempstring
'        frmSetup.txtCensus_Database.text = tempstring
'        Close #filenum
'
'    'else if config file doesn't exist use defaults.
'
'    Else
'
'        'Defaults set in form.
'
'    End If
'
'    'Turn off error handler.
'
'    On Local Error GoTo 0
'
    ' check if census file on setup form is legit.
    ' if not, give message and show setup form

'    Load frmSetup
    strFname = frmSetup.txtData_path.text & "\" & frmSetup.txtCensus_Database.text

    On Error Resume Next
    strCensusFile = Dir$(strFname)
    On Error GoTo 0
    If strCensusFile = "" Then
        MsgBox "SecPop cannot find the census file:  " & vbCrLf & _
                strFname & vbCrLf & vbCrLf & _
               "You will need to either place the census fie in " & _
               "the CENSUS directory, or manually edit the SECPOP.CFG " & _
               "file to point to the directory where the CENSUS file " & _
               "is located. " & vbCrLf & vbCrLf & _
               "SecPop will end."
        frmSetup.Show
        End
    End If

    'get all census files for combo box on setup form,

'    strFname = App.path & "\census\census*.dat"

'    strCensusFile = Dir$(strFname)
'    If strCensusFile = "" Then
'        MsgBox "SecPop cannot find the census file:  " & _
'                strFname & vbCrLf & _
'               "SecPop will end."
'        End
'    Else
       i = -1
       strCensusFile = Dir$(App.path & "\census\census*.dat")
       While strCensusFile <> ""
           i = i + 1
           strFnames(i) = UCase(strCensusFile)
           strCensusFile = Dir$
       Wend
'    End If

    For j = 0 To i
        frmSetup.cboCensus_Database.AddItem strFnames(j)
    Next j

    'set caption to show which census file is being used
```

```
    frmMain.caption = MainTitle & "    (" & _
                      frmSetup.txtCensus_Database.text & ")"

    'County file

    strFname = frmSetup.txtData_path.text & "\" & frmSetup.txtCounty_Database.text

    On Error Resume Next
    strCountyFile = Dir$(strFname)
    On Error GoTo 0
    If strCountyFile = "" Then
        MsgBox "SecPop cannot find the economic file:  " & vbCrLf & _
                strFname & vbCrLf & _
               "SecPop will end."
        frmSetup.Show
        End
    End If

    'get all census files for combo box on setup form,

'    strFname = App.path & "\census\census*.dat"

'    strCensusFile = Dir$(strFname)
'    If strCensusFile = "" Then
'        MsgBox "SecPop cannot find the census file:  " & _
'                strFname & vbCrLf & _
'               "SecPop will end."
'        End
'    Else
       i = -1
       strCountyFile = Dir$(App.path & "\census\county*.dat")
       While strCountyFile <> ""
           i = i + 1
           strFnames(i) = UCase(strCountyFile)
           strCountyFile = Dir$
       Wend
'    End If

    For j = 0 To i
        frmSetup.cboCounty_Database.AddItem strFnames(j)
    Next j

    'set up default dirs

    InitSiteFilePath = frmSetup.txtSite_Path
    InitProblemFilePath = frmSetup.txtProblem_path
    InitOutputFilePath = frmSetup.txtOutput_path


    With frmMain.dlgSecpop
        .CancelError = True
    End With

    'Initialize to zero the radial distance array.

'    For i = 1 To max_number_of_radii
'        radial_distance(i) = 0.1
'    Next i

    radial_distance(1) = 0.1
    For i = 2 To max_number_of_radii
        radial_distance(i) = 0
    Next i

    'Reset the number of radii to 0
```

```
    number_of_radii = 1

'Data for radii to be used for the regional calculations.

'Data 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
'Data 11, 12, 13, 14, 15, 16, 17, 18, 19, 20
'Data 30, 40, 50, 60, 70, 80, 90
'Data 100, 110, 120, 130, 140, 150, 160, 170

'Data for forms which display the compass points.

'Data "N", "NNE", "NE", "ENE", "E", "ESE", "SE", "SSE", "S", "SSW", "SW"
'Data "WSW", "W", "WNW", "NW", "NNW"

    'Initialize the radii used for regional calculations.

    'bem-array defined
    regional_radii(0) = 1
    regional_radii(1) = 2
    regional_radii(2) = 3
    regional_radii(3) = 4
    regional_radii(4) = 5
    regional_radii(5) = 6
    regional_radii(6) = 7
    regional_radii(7) = 8
    regional_radii(8) = 9
    regional_radii(9) = 10
    regional_radii(10) = 11
    regional_radii(11) = 12
    regional_radii(12) = 13
    regional_radii(13) = 14
    regional_radii(14) = 15
    regional_radii(15) = 16
    regional_radii(16) = 17
    regional_radii(17) = 18
    regional_radii(18) = 19
    regional_radii(19) = 20
    regional_radii(20) = 30
    regional_radii(21) = 40
    regional_radii(22) = 50
    regional_radii(23) = 60
    regional_radii(24) = 70
    regional_radii(25) = 80
    regional_radii(26) = 90
    regional_radii(27) = 100
    regional_radii(28) = 110
    regional_radii(29) = 120
    regional_radii(30) = 130
    regional_radii(31) = 140
    regional_radii(32) = 150
    regional_radii(33) = 160
    regional_radii(34) = 170

    'bem-array defined
    directions(0) = "N"
    directions(1) = "NNE"
    directions(2) = "NE"
    directions(3) = "ENE"
    directions(4) = "E"
    directions(5) = "ESE"
    directions(6) = "SE"
    directions(7) = "SSE"
    directions(8) = "S"
    directions(9) = "SSW"
```

```
    directions(10) = "SW"
    directions(11) = "WSW"
    directions(12) = "W"
    directions(13) = "WNW"
    directions(14) = "NW"
    directions(15) = "NNW"

    DgToRd = pi / 180#

    frmRosette.blnInputDataChanged = True

    Exit Sub

'configerr:
'
'    'If config file format is screwed up use defaults.
'
'    Close filenum
'    Exit Sub

End Sub
```

H.7 Visual Basic Code Module: MainModule.bas

```
Attribute VB_Name = "MainModule"
Option Explicit

'**************************************************************************************
'  Module:  MainModule
'
'  Project: SECPOP
'
'  Description:
'
'  Modification History
'  Date       By   Description
'  --------   ---  ----------------------------------------------------------
'
'

'**************************************************************************************

Public Sub Main()

    'Initialize variables
    Initializations

    frmMain.Show
'    frmDisclaimer.Show

End Sub
```

H.8 Visual Basic Code Module: Mp_setup.bas

```
Attribute VB_Name = "Mp_setup"
'qqq ChDir "mapplan"
'qqq Shell "setup.exe"
'qqq ChDir ".."
'qqq RUN "secpop90"
```

H.9 Visual Basic Code Module: Outpcode1.bas

```
Attribute VB_Name = "Outpcode"
'******************************************************************************
'  Module:  Outpcode
'
'  Project: SECPOP
'
'  Description: This module contains the subroutines necessary to output a
'               MACCS input file in:
'                   MACCS Compatible Format
'                   CSV format
'                   Screen
'                   Hard Copy
'
'  Modification History
'  Date      By   Description
'  --------  ---  ------------------------------------------------------------
'  20020131  cwm  Incorporated '90 version into '00 project.
'  20020213  cwm  Changed print control to dlgsecpop
'  20020213  cwm  Incorporated print to file as means of saving file.
'  20020319  cwm  upgraded print to file and fixed table print bug
'  20020610  cwm  changed print to file utility to stop trailing commas
'                 change print to printer so left characters print first for long lines
'******************************************************************************

Option Explicit


Sub print_MACCS_input_file(f, MorC)


    'This routine prints SECPOP output in MACCS compatible form or in
    'a quotation mark comma delimited form. It can be used to print to
    'a printer or a file.

    Dim quote As String, comma As String
    Dim blank As String, degree As String
    Dim i As Integer, j As Integer
    Dim Out As String


    'Set quote and comma strings to null if MACCS input file format is
    'selected or to '"' and ',' if comma separated variable (CSV) format
    'is selected.

    If (MorC = "MACCS") Then
        quote = ""
        comma = ""
        blank = " "
        degree = Chr$(100)     'Set degree character to "d" for reduced F77 ASCII set.
    ElseIf (MorC = "CSV") Then
        quote = """"
        comma = ","
        blank = " "
        degree = Chr$(176)
    Else
        quote = ""
        comma = ""
        blank = " "
        degree = Chr$(100)
    End If

    'Enable local error checking for things like printer off line, etc.
```

```
    On Local Error GoTo print_maccs_error


    'Initialize RADDIS array to Kilometers.

    For i = 1 To number_of_radii Step 1

        If (frmProblem_Data.optKilometers.Value = True) Then
            raddis(i) = radial_distance(i)
        Else
            raddis(i) = radial_distance(i) * miles_to_kilometers
        End If

    Next i


    'Print (or save) results.
    'Print the first line.

'Previous version was SECPOP90 V2.3
    Print #f, blank; "SECPOP V3.11  ";
'    If (frmSetup.optMACCS.Value) Then
    If MorC = "MACCS" Then
        Print #f, "MACCS ";
'    ElseIf (frmSetup.optCSV.Value) Then
    ElseIf MorC = "CSV" Then
        Print #f, "CSV   ";
    Else
        Print #f, "MACCS ";
    End If
    Print #f, "Site Data File for"; blank;
    Print #f, LTrim$(RTrim$(frmSite_Data.txtSite_Name.text))

    'Print the second line.
    Out = " Lat: " _
        & InsertSpace(2, Val(frmSite_Data.txtLatitude_Degrees.text)) & degree _
        & InsertSpace(2, Val(frmSite_Data.txtLatitude_Minutes.text)) & "'" _
        & InsertSpace(2, Val(frmSite_Data.txtLatitude_Seconds.text)) & "''" _
        & " Long: " _
        & InsertSpace(3, Val(frmSite_Data.txtLongitude_Degrees.text)) & degree _
        & InsertSpace(2, Val(frmSite_Data.txtLongitude_Minutes.text)) & "'" _
        & InsertSpace(2, Val(frmSite_Data.txtLongitude_Seconds.text)) & "''" _
        & " Population multiplier: " _
        & Format(Val(frmProblem_Data.txtPopulation_multiplier.text), "###0.0000") _
        & "   " & Format$(Now, "mm/dd/yyyy")
    Print #f, Out


'    Print #f, quote; blank; "Lat:"; quote; comma; blank;
'    Print #f, InsertSpace(2, Val(frmSite_Data.txtLatitude_Degrees.text)); comma; quote; degree;
quote; comma;
'    Print #f, InsertSpace(2, Val(frmSite_Data.txtLatitude_Minutes.text)); comma; quote; "'";
quote; comma;
'    Print #f, InsertSpace(2, Val(frmSite_Data.txtLatitude_Seconds.text)); comma; quote; "''";
quote; comma;
'    Print #f, quote; comma; blank; quote;
'
'    Print #f, "Long:"; quote; comma; blank;
'    Print #f, InsertSpace(3, Val(frmSite_Data.txtLongitude_Degrees.text)); comma; quote; degree;
quote; comma;
'    Print #f, InsertSpace(2, Val(frmSite_Data.txtLongitude_Minutes.text)); comma; quote; "'";
quote; comma;
'    Print #f, InsertSpace(2, Val(frmSite_Data.txtLongitude_Seconds.text)); comma; quote; "''";
quote; comma; blank;
'
'    Print #f, "Population multiplier:"; quote; comma; blank;
```

```
'    Print #f, Format(Val(frmProblem_Data.txtPopulation_multiplier.text), "###0.0000"); comma;
blank; blank; blank; quote;
'
'    Print #f, Format$(Now, "mm/dd/yyyy"); quote

    'Print the data block sizes.

    Out = Justify(4, number_of_radii, "##")
    Print #f, Out; comma; blank; "SPATIAL INTERVALS"

    Out = Justify(4, number_of_segments, "##")
    Print #f, Out; comma; blank; "WIND DIRECTIONS"
    Out = Justify(4, 7#, "##")
    Print #f, Out; comma; blank; "CROP CATEGORIES"

    Out = Justify(4, 4#, "##")
    Print #f, Out; comma; blank; "WATER PATHWAY ISOTOPES"

    Out = Justify(4, 1#, "##")
    Print #f, Out; comma; blank; "WATERSHEDS"

    Out = Justify(4, number_econ_regions, "##")
    Print #f, Out; comma; blank; "ECONOMIC REGIONS"

    'Print the geometry data block.

    Print #f, quote; blank; "SPATIAL DISTANCES     KILOMETERS"; quote
    For i = 1 To number_of_radii Step 1
        If ((i Mod 8) = 0) Then
            Print #f, Justify(10, raddis(i), "0.00")
        ElseIf i = number_of_radii Then
            Print #f, Justify(10, raddis(i), "0.00")
        Else
            Print #f, Justify(10, raddis(i), "0.00"); comma;
        End If
    Next i

'    'Go to the next line if necessary.
'    If ((number_of_radii Mod 8) <> 0) Then
'        Print #f,
'    End If

    'Print the population data block.
    'Out = blank & "POPULATION"
    Print #f, quote; blank; "POPULATION"; quote
'   Print #f, blank; "POPULATION"

    For i = 1 To number_of_segments Step 1
        For j = 1 To number_of_radii Step 1
            Out = Justify(10, sector_population(i, j), "0.")
            If ((j Mod 8) = 0) Then
                Print #f, Out
            ElseIf j = number_of_radii Then
                Print #f, Out
            Else
                Print #f, Out; comma;
            End If
        Next j
        'Go to the next line if necessary.
    Next i

    'Print the land fraction block.

    Print #f, quote; blank; "LAND FRACTION"; quote

    For i = 1 To number_of_segments Step 1
```

```
        For j = 1 To number_of_radii Step 1
            Out = Justify(5, sector_frclnd(i, j), "0.00")
            If ((j Mod 16) = 0) Then
                Print #f, Out
            ElseIf j = number_of_radii Then
                Print #f, Out
            Else
                Print #f, Out; comma;
            End If
        Next j
    Next i

    'Print the region identification block.

    Print #f, quote; blank; "REGION INDEX"; quote
    For i = 0 To (number_of_segments - 1) Step 1
        For j = 1 To number_of_radii Step 1
            Out = Justify(2, region_index(j, i), "#0")
            If ((j Mod 38) = 0) Then
                Print #f, Out
            ElseIf j = number_of_radii Then
                Print #f, Out
            Else
                Print #f, Out; comma;
            End If
        Next j
    Next i

    'Print the (dummy) watershed identification block.

    Print #f, quote; blank; "WATERSHED INDEX"; quote
    For i = 1 To number_of_segments Step 1
        For j = 1 To number_of_radii - 1 Step 1
            Out = Justify(2, 1, "#0")
            Print #f, Out; comma;
        Next j
        Print #f, Justify(2, 1, "#0")
    Next i

    'Print the (dummy) agricultural data block.
    Print #f, quote; blank; "CROP SEASON AND SHARE"; quote

'   Print #f, Justify(4, 1, "##"); comma;
'   Print #f, quote; blank; "NONE                ";  quote; comma;
'   Print #f, Justify(5, 1#, "0."); comma;
'   Print #f, Justify(5, 1#, "0.");
'   Print #f, Justify(8, 0, "0.0000")

    Print #f, Justify(4, 1, "##"); comma;
    Print #f, quote; blank; "PASTURE             ";  quote; comma;
    Print #f, Justify(5, 90#, "00."); comma;
    Print #f, Justify(5, 270#, "000.");
    Print #f, Justify(8, 0.41, "0.0000")

    Print #f, Justify(4, 2, "##"); comma;
    Print #f, quote; blank; "STORED FORAGE       ";  quote; comma;
    Print #f, Justify(5, 150#, "000."); comma;
    Print #f, Justify(5, 240#, "000.");
    Print #f, Justify(8, 0.13, "0.0000")

    Print #f, Justify(4, 3, "##"); comma;
    Print #f, quote; blank; "GRAINS              ";  quote; comma;
    Print #f, Justify(5, 150#, "000."); comma;
    Print #f, Justify(5, 240#, "000.");
    Print #f, Justify(8, 0.21, "0.0000")
```

```
    Print #f, Justify(4, 4, "##"); comma;
    Print #f, quote; blank; "GRN LEAFY VEGETABLES"; quote; comma;
    Print #f, Justify(5, 150#, "000."); comma;
    Print #f, Justify(5, 240#, "000.");
    Print #f, Justify(8, 0.002, "0.0000")

    Print #f, Justify(4, 5, "##"); comma;
    Print #f, quote; blank; "OTHER FOOD CROPS     "; quote; comma;
    Print #f, Justify(5, 150#, "000."); comma;
    Print #f, Justify(5, 240#, "000.");
    Print #f, Justify(8, 0.004, "0.0000")

    Print #f, Justify(4, 6, "##"); comma;
    Print #f, quote; blank; "LEGUMES AND SEEDS    "; quote; comma;
    Print #f, Justify(5, 150#, "000."); comma;
    Print #f, Justify(5, 240#, "000.");
    Print #f, Justify(8, 0.15, "0.0000")

    Print #f, Justify(4, 7, "##"); comma;
    Print #f, quote; blank; "ROOTS AND TUBERS     "; quote; comma;
    Print #f, Justify(5, 150#, "000."); comma;
    Print #f, Justify(5, 240#, "000.");
    Print #f, Justify(8, 0.003, "0.0000")


    'Print the (dummy) watershed data block.

    Print #f, quote; blank; "WATERSHED DEFINITION  -- INITIAL AND ANNUAL WASHOFF AND INGESTION
FACTORS"; quote

    Print #f, Justify(4, 1#, "0"); comma; quote; blank;
    Print #f, "Sr-89                          "; quote; comma;
    Print #f, Justify(10, 0.000005, "0.00E+00"); comma;
    Print #f, Justify(10, 0#, "0.0")

    Print #f, Justify(4, 2#, "0"); comma; quote; blank;
    Print #f, "Sr-90                          "; quote; comma;
    Print #f, Justify(10, 0.000005, "0.00E+00"); comma;
    Print #f, Justify(10, 0#, "0.0")

    Print #f, Justify(4, 3#, "0"); comma; quote; blank;
    Print #f, "Cs-134                         "; quote; comma;
    Print #f, Justify(10, 0.000005, "0.00E+00"); comma;
    Print #f, Justify(10, 0#, "0.0")

    Print #f, Justify(4, 4#, "0"); comma; quote; blank;
    Print #f, "Cs-137                         "; quote; comma;
    Print #f, Justify(10, 0.000005, "0.00E+00"); comma;
    Print #f, Justify(10, 0#, "0.0")


    'Print the regional economic data block.

    Print #f, quote; blank; "REGIONAL ECONOMIC DATA"; quote

    For i = 1 To number_econ_regions Step 1
        Print #f, Justify(4, i, "##"); comma;

        If i = 1 Then
            Print #f, quote; blank; "EXCLUSION"; quote; comma;
        Else
            Print #f, quote; blank; "REGION_"; Justify(2, i, "00"); quote; comma;
        End If

        Print #f, Justify(10, econ_data(i).region_frmfrc, "0.000"); comma;
        Print #f, Justify(10, econ_data(i).region_dpf, "0.000"); comma;
```

```
        Print #f, Justify(10, econ_data(i).region_asfp, "0.0"); comma;
        Print #f, Justify(10, econ_data(i).region_vfrm, "0.0"); comma;
        Print #f, Justify(10, econ_data(i).region_vnfrm, "0.0")
    Next i

    Exit Sub


'Error return.

print_maccs_error:

    'If there is a problem, tell the user and exit function.

    MsgBox Error$(Err)

    Exit Sub


End Sub

Sub Print_Problem_File_to_Printer()
    'This function creates a temporary file, and then
    'writes into that file the information for the user
    'to view on the screen.  This file is then sent to a
    'text form, where the user can view it and scroll
    'the various parts of the data.

    Dim filenum As Integer
    Dim Filename1 As String
    Dim Filename2 As String
    Dim strHeader1 As String
    Dim strHeader2 As String
    Dim strFooter1 As String
    Dim strFooter2 As String
    Dim i As Integer
    Dim iPageNO As Integer
    Dim Pathname As String
    Dim iCopy As Integer
    Dim PtF As Boolean

    Unload frmDisclaimer

    'set up page format strings
    strHeader1 = "SECPOP"
    Call parse_path_and_file(frmProblem_Data.caption, Pathname, strHeader2)
    strFooter2 = Format(Time, "m/d/yy h:mm")

    'Write the actual data to file.
    Filename1 = "$$TEMP$$.$$$"
    filenum = FreeFile
    Open Filename1 For Output As filenum
    Call Create_Temp_Problem_File(filenum)
    Close #filenum

    'Reset to printer the destination of info generated
    'by the print_MACCS_input function.

    On Error GoTo PrintErrHandler
    With frmMain.dlgSecpop
        .Flags = cdlPDHidePrintToFile
        .DialogTitle = "Print Problem File"
        .ShowPrinter
        iCopy = .Copies
'       PtF = .Flags
    End With
```

```
'    If PtF Then
'        'If print to file checked
'        With frmMain.dlgSecpop
' '           .Flags = 0
'            .Flags = cdlOFNOverwritePrompt
'            .DialogTitle = "Save Problem File"
'            .Filename = StripExt(.Filename)
'            .Filter = "All Files (*.*)|*.*|" & _
'                      "MACCS Files (*.inp)|*.inp|" & _
'                      "Comma Separated Variables (*.csv)|*.csv|" & _
'                      "Text Files(*.txt)|*.txt|" & _
'                      "Site Files (*.sit)|*.sit|" & _
'                      "Problem Files (*.prb)|*.prb|"
'            .FilterIndex = 6     'Default filter = *.prb
'            .Filename = ""
'            .ShowSave
'            Filename2 = .Filename
'        End With
'        Name Filename1 As Filename2     ' Move and rename file
'        Close filenum
'    Else
        'otherwise
        filenum = FreeFile
        Open Filename1 For Input As filenum
        For i = 1 To iCopy
            Close #filenum
            filenum = FreeFile
            Open Filename1 For Input As filenum
            Call PrinterUtility(strHeader1, strHeader2, strFooter1, strFooter2, filenum)
        Next i
        Close filenum
        Kill Filename1
'    End If

    Exit Sub

PrintErrHandler:

    'Deal with errors such as printer is off line or file not found.

    'MsgBox "Device error - check printer or other print device."
    Exit Sub

End Sub

Sub Create_Temp_Problem_File(problem_file)

    Dim i As Integer
    Dim j As Integer

    Print #problem_file, "SECPOP   V3.0"
    Print #problem_file, "Site File: "
    Print #problem_file, frmProblem_Data.txtSite_File_Name.text
    Print #problem_file, "Number of Radii:"
    Print #problem_file, Format(number_of_radii, "##")
    Print #problem_file, "Units of Measure:"
    If frmProblem_Data.optKilometers.Value = True Then
        Print #problem_file, "Kilometers"
    Else: Print #problem_file, "Miles"

    End If

    Print #problem_file, "Radial Distances:"
    For i = 1 To number_of_radii Step 1
              Print #problem_file, Format(radial_distance(i), "####.0000")
```

```
    Next i

    Print #problem_file, "Population Multiplier:"
    Print #problem_file, Format(Val(frmProblem_Data.txtPopulation_multiplier.text), "####.0000")
    Print #problem_file, "Number of Economic regions:"
    Print #problem_file, Format(number_econ_regions, "##")
    Print #problem_file, "Economic regions:"
    For i = 0 To (number_of_segments - 1)
        For j = 1 To number_of_radii
            Print #problem_file, Format(region_index(j, i), "00");
            Print #problem_file, " ";
        Next j
        Print #problem_file, ""
    Next i
    Print #problem_file, "Problem Remarks:"
    Print #problem_file, frmProblem_Data.txtProblem_Remarks.text

End Sub

Sub Print_Site_File_To_Printer()
    'This function creates a temporary file, and then
    'writes into that file the information for the user
    'to view on the screen.  This file is then sent to a
    'text form, where the user can view it and scroll
    'the various parts of the data.

    Dim filenum As Integer
    Dim Filename1 As String
    Dim Filename2 As String
    Dim strHeader1 As String
    Dim strHeader2 As String
    Dim strFooter1 As String
    Dim strFooter2 As String
    Dim i As Integer
    Dim iPageNO As Integer
    Dim Pathname As String
    Dim iCopy As Integer
    Dim PtF As Boolean

    Unload frmDisclaimer

    'set up page format strings
    strHeader1 = "SECPOP"
    Call parse_path_and_file(frmProblem_Data.caption, Pathname, strHeader2)
    strFooter2 = Format(Time, "m/d/yy h:mm")

    'Write the actual data to file.
    Filename1 = "$$TEMP$$.$$$"
    filenum = FreeFile
    Open Filename1 For Output As filenum
    Call Create_Temp_Site_File(filenum)
    Close #filenum

    'Reset to printer the destination of info generated
    'by the print_MACCS_input function.

    On Error GoTo PrintErrHandler
    With frmMain.dlgSecpop
        .Flags = cdlPDHidePrintToFile
        .DialogTitle = "Print Site File"
        .ShowPrinter
        iCopy = .Copies
'       PtF = .Flags
    End With

'   If PtF Then
```

```
'        'If print to file checked
'        With frmMain.dlgSecpop
''            .Flags = 0
'            .Flags = cdlOFNOverwritePrompt
'            .DialogTitle = "Save Site File"
'            .Filename = StripExt(.Filename)
'            .Filter = "All Files (*.*)|*.*|" & _
'                      "MACCS Files (*.inp)|*.inp|" & _
'                      "Comma Separated Variables (*.csv)|*.csv|" & _
'                      "Text Files(*.txt)|*.txt|" & _
'                      "Site Files (*.sit)|*.sit|" & _
'                      "Problem Files (*.prb)|*.prb|"
'            .FilterIndex = 5     'Default filter = *.sit
'            .Filename = ""
'            .ShowSave
'            Filename2 = .Filename
'        End With
'        Name Filename1 As Filename2     ' Move and rename file
'        Close filenum
'    Else
        'otherwise
        filenum = FreeFile
        Open Filename1 For Input As filenum
        For i = 1 To iCopy
            Close #filenum
            filenum = FreeFile
            Open Filename1 For Input As filenum
            Call PrinterUtility(strHeader1, strHeader2, strFooter1, strFooter2, filenum)
        Next i
        Close filenum
        Kill Filename1
'    End If

    Exit Sub

PrintErrHandler:

    'Deal with errors such as printer is off line or file not found.
    'MsgBox "Device error - check printer or other print device."
    Exit Sub

End Sub

Sub Create_Temp_Site_File(f)

    'This routine prints SECPOP site data to a temporary file for printing.

    Print #f, "Site Name:"
    Print #f, frmSite_Data.txtSite_Name.text
    Print #f, "Site Coordinates:"
    Print #f, "Longitude (Degrees, Minutes, Seconds)"
    Print #f, "                "; Val(frmSite_Data.txtLongitude_Degrees.text);
    Print #f, "          "; Val(frmSite_Data.txtLongitude_Minutes.text);
    Print #f, "          "; Val(frmSite_Data.txtLongitude_Seconds.text)
    Print #f, "Latitude (Degrees, Minutes, Seconds)"
    Print #f, "                "; Val(frmSite_Data.txtLatitude_Degrees.text);
    Print #f, "          "; Val(frmSite_Data.txtLatitude_Minutes.text);
    Print #f, "          "; Val(frmSite_Data.txtLatitude_Seconds.text)
    Print #f, "Site Remarks:"
    Print #f, frmSite_Data.txtSite_Remarks.text

End Sub


Private Function Justify(Columns, Value, FormatString As String)
    Dim strX As String
```

```
    If IsNumeric(Value) Then
        strX = Format(Value, FormatString)
    Else
        strX = Value
    End If

    If (Columns - Len(strX) >= 0) Then
        Justify = Space(Columns - Len(strX)) & strX
    ElseIf (Columns - Len(strX) < 0) Then
        Justify = String(Columns, "*")
    End If

End Function


Private Function InsertSpace(length, number) As String

    Dim TempString As String
    Dim FormatLength As Integer

    If number = 0 Then
        InsertSpace = " 0"
    Else
        TempString = Format(number, "#")
        FormatLength = length - Len(TempString)
        If FormatLength > 0 Then
            InsertSpace = String(FormatLength, " ") & TempString
        Else
            InsertSpace = TempString
        End If
    End If

End Function


Sub Print_MaCCS_File_To_Printer(MorC)
    'This function creates a temporary file, and then
    'writes into that file the information for the user
    'to view on the screen.  This file is then sent to a
    'text form, where the user can view it and scroll
    'the various parts of the data.

    Dim filenum As Integer
    Dim Filename1 As String
    Dim Filename2 As String
    Dim strHeader1 As String
    Dim strHeader2 As String
    Dim strFooter1 As String
    Dim strFooter2 As String
    Dim i As Integer
    Dim iPageNO As Integer
    Dim Pathname As String
    Dim iCopy As Integer
    Dim PtF As Boolean

    Unload frmDisclaimer

    'If problem is not loaded, tell user and exit subroutine
    If frmProblem_Data.txtRadial_Distance1.text = "" Then
        MsgBox "Data is not yet available - Site file has not been loaded"
        Exit Sub
    End If

    'set up page format strings
    strHeader1 = "SECPOP2000"
```

```
    Call parse_path_and_file(frmProblem_Data.caption, Pathname, strHeader2)
'    strFooter2 = Format(Time, "m/d/yy h:mm")
    strFooter2 = Date & " " & Time

    'Write the actual data to a new file.
    Filename1 = "$$TEMP$$.$$$"
    filenum = FreeFile
    Open Filename1 For Output As filenum
    If MorC = "MACCS" Then
        Call print_MACCS_input_file(filenum, "MACCS")
    Else
        Call print_MACCS_input_file(filenum, "CSV")
    End If
    Close #filenum

    'Reset to printer the destination of info generated
    'by the print_MACCS_input function.

    On Error GoTo PrintErrHandler
    With frmMain.dlgSecpop
        .Flags = cdlPDHidePrintToFile
        .DialogTitle = "Print MACCS File"
        .ShowPrinter
        iCopy = .Copies
'        PtF = .Flags
    End With

    filenum = FreeFile
    Open Filename1 For Input As filenum
    For i = 1 To iCopy
        Close #filenum
        filenum = FreeFile
        Open Filename1 For Input As filenum
        Call PrinterUtility(strHeader1, strHeader2, strFooter1, strFooter2, filenum)
    Next i
    Close filenum
    Kill Filename1

    Exit Sub

PrintErrHandler:

    'Deal with errors such as printer is off line or file not found.

    'MsgBox "Device error - check printer or other print device."
    Close filenum
    Kill Filename1
    Exit Sub

End Sub

Sub PrinterUtility(strHeader1, strHeader2, strFooter1, strFooter2, filenum)

    Dim LineString As String

    Printer.Print
    Printer.Font = "courier new"
    Printer.FontSize = 10
    Printer.ScaleMode = 5
    Printer.ScaleHeight = 9

'Print first page header
    Printer.CurrentY = 0.5
    Printer.CurrentX = (Printer.ScaleWidth - Printer.TextWidth(strHeader1)) / 2
    Printer.Print strHeader1
    Printer.CurrentY = 0.5
```

```
    Printer.CurrentX = Printer.ScaleWidth - Printer.TextWidth(strHeader2) - 1
    Printer.Print strHeader2

'Print first page footer
    strFooter1 = "Page " & CStr(Printer.Page)
    Printer.CurrentY = Printer.ScaleHeight - 0.5
    Printer.CurrentX = (Printer.ScaleWidth - Printer.TextWidth(strFooter1)) / 2
    Printer.Print strFooter1

    Printer.CurrentY = Printer.ScaleHeight - 0.5
    Printer.CurrentX = Printer.ScaleWidth - Printer.TextWidth(strFooter2) - 1
    Printer.Print strFooter2

'Print MACCS file to paper
    Printer.CurrentY = 1
    Do While Not EOF(filenum)
        Printer.CurrentX = 1
        Line Input #filenum, LineString
        Do
            If Len(LineString) > 80 Then
                Printer.Print Left(LineString, 80)
                LineString = Right(LineString, Len(LineString) - 80)
                Printer.CurrentX = 1
            Else
                Printer.Print LineString
                Exit Do
            End If
        Loop
        If Printer.CurrentY + 1 >= Printer.ScaleHeight Then
            'print new page header, footer
            Printer.NewPage
            Printer.CurrentY = 0.5
            Printer.CurrentX = (Printer.ScaleWidth - Printer.TextWidth(strHeader1)) / 2
            Printer.Print strHeader1

            Printer.CurrentY = 0.5
            Printer.CurrentX = Printer.ScaleWidth - Printer.TextWidth(strHeader2) - 1
            Printer.Print strHeader2

            strFooter1 = "Page " & CStr(Printer.Page)
            Printer.CurrentY = Printer.ScaleHeight - 0.5
            Printer.CurrentX = (Printer.ScaleWidth - Printer.TextWidth(strFooter1)) / 2
            Printer.Print strFooter1

            Printer.CurrentY = Printer.ScaleHeight - 0.5
            Printer.CurrentX = Printer.ScaleWidth - Printer.TextWidth(strFooter2) - 1
            Printer.Print strFooter2

            Printer.CurrentY = 1
        End If
    Loop
    Printer.EndDoc

End Sub

Sub SaveResultsToFile(MorC)

    'This event proceedure saves the results so that they are not lost when
    'the program is exited.  For this function to work, a problem must be
    'loaded, along with the data that corresponds to that problem

    Dim filenum As Integer
    Dim Flags As Integer
    Dim Fname As String
    Dim Fext As String
    Dim DotNo As Integer
```

```
    'If problem is not loaded, tell user and exit subroutine
    If frmProblem_Data.txtRadial_Distance1.text = "" Then
        MsgBox "Data is not yet available - Site file has not been loaded"
        Exit Sub
    End If

    Unload frmDisclaimer

    On Error GoTo ErrHandler

    frmMain.dlgSecpop.Flags = 0
    With frmMain.dlgSecpop
        .CancelError = True     ' Set CancelError is True
'        .Flags = cdlOFNHideReadOnly  ' Set flags
        .Flags = cdlOFNHideReadOnly & cdlOFNOverwritePrompt ' Set flags
        .Filter = "All Files (*.*)|*.*|" & _
                  "MACCS Files (*.inp)|*.inp|" & _
                  "Comma Separated Variables (*.csv)|*.csv|" & _
                  "Text Files(*.txt)|*.txt|" & _
                  "Site Files (*.sit)|*.sit|" & _
                  "Problem Files (*.prb)|*.prb|"

        If MorC = "CSV" Then
            .FilterIndex = 3    'Default filter = *.csv
            .DialogTitle = "Save Output as CSV file"
        Else
            .FilterIndex = 2    'Default filter = *.inp
            .DialogTitle = "Save Output as MACCS file"
        End If

        .Filename = ""
        If InitOutputFilePathChanged = True Then
            .InitDir = frmSetup.txtOutput_path.text
            InitOutputFilePathChanged = False
        Else
            .InitDir = InitOutputFilePath
        End If
        .ShowSave ' Display the Save dialog box
        Fname = .Filename
    End With

    'verify that file type has not changed
    Fext = Mid(Fname, InStrRev(Fname, "."), Len(Fname))
'    If Fext = ".inp" Then
'        frmSetup.optMACCS.Value = True
'        frmSetup.optCSV.Value = False
'    ElseIf Fext = ".csv" Then
'        frmSetup.optCSV.Value = True
'        frmSetup.optMACCS.Value = False
'    Else
'        frmSetup.optMACCS.Value = True
'    End If

    On Local Error GoTo save_results_error

    'If file already exists, delete it (to prevent appending to it)
    If Dir$(Fname) <> "" Then Kill Fname
    filenum = FreeFile
    Open Fname For Output As filenum

    'Call print function to print maccs data to the file

    Call print_MACCS_input_file(filenum, MorC)

    Close #filenum
```

```
    'Turn off local error checking

    On Local Error GoTo 0

    Exit Sub

ErrHandler:
  'User pressed the Cancel button
  Exit Sub

save_results_error:
    'If any kind of error occurs, inform the user and exit subroutine
    MsgBox "A file error has occured.  Please check all filenames and devices"
    Exit Sub

End Sub

Sub Create_Temp_Table_File(f)

    'This routine prints the displayed population table.

    Dim degree As String
    Dim pages As Integer
    Dim radius As Integer
    Dim first_radius As Integer
    Dim last_radius As Integer
    Dim i As Integer, j As Integer, k As Integer
    Dim L As Integer, m As Integer, n As Integer
    Dim sum As Long
    Dim TempString As String

    'Set quote and comma strings to null if MACCS input file format is
    'selected or to '"' and ',' if comma separated variable (CSV) format
    'is selected.

    degree = Chr$(248)

    pages = (number_of_radii + 1) \ 6
    If (((number_of_radii + 1) Mod 6) <> 0) Then
        pages = pages + 1
    End If

    If (table_type = 1) Then
        Print #f, "SECPOP   V3.0 " & "Population Data"
    Else
        Print #f, "SECPOP   V3.0 " & "Cumulative Population Data"
    End If

    Print #f, "Date:  " & Format$(Now, "mm/dd/yyyy") & "  Time:" & Format$(Now, "hh:mm:ss")
    Print #f, "Site Name: " & Trim(frmSite_Data.txtSite_Name.text)

    TempString = "Latitude:  " & Format(frmSite_Data.txtLatitude_Degrees.text, "##") & degree _
                & Format(frmSite_Data.txtLatitude_Minutes.text, "##") & "'" _
                & Format(frmSite_Data.txtLatitude_Seconds.text, "##") & "''" & "   " _
                & "Longitude:  " & Format(frmSite_Data.txtLongitude_Degrees.text, "###") & degree
_
                & Format(frmSite_Data.txtLongitude_Minutes.text, "##") & "'" _
                & Format(frmSite_Data.txtLongitude_Seconds.text, "##") & "''"
    Print #f, TempString
    Print #f, "Population Multiplier:  " & Format(frmProblem_Data.txtPopulation_multiplier.text,
"#####.####")


    radius = 1

    For j = 1 To pages
```

```
'Print out the radii header row.

If (frmProblem_Data.optKilometers.Value) Then
    TempString = "Radii(km)      "
Else
    TempString = "Radii(mi)      "
End If

'Print out the radii, setting the loop min and max to
'the radii to be printed for this page. Note: the final
'radii column is the sum column.

first_radius = radius
If ((radius + 5) <= (number_of_radii + 1)) Then
    last_radius = radius + 5
Else
    last_radius = number_of_radii + 1
End If

For k = first_radius To last_radius Step 1
    If (k <> (number_of_radii + 1)) Then
        TempString = TempString & Justify(10, radial_distance(k), "0.0000")
    Else
        TempString = TempString & "       Sum"
    End If
Next k

Print #f, TempString

'Print out the direction and  the (cumulative)
'population values. The last column is the sum of the
'population values.

For k = 1 To number_of_segments Step 1

    TempString = directions(k - 1) & String(15 - Len(directions(k - 1)), " ")

    For L = first_radius To last_radius
        If (L <> (number_of_radii + 1)) Then
            If (table_type = 1) Then
                TempString = TempString & Justify(10, sector_population(k, L), "0")
            Else
                sum = 0
                For m = 1 To L Step 1
                    sum = sum + sector_population(k, m)
                Next m
                TempString = TempString & Justify(10, sum, "0")
            End If
        Else
            sum = 0
            For m = 1 To number_of_radii Step 1
                sum = sum + sector_population(k, m)
            Next m
            TempString = TempString & Justify(10, sum, "0")
        End If
    Next L
    Print #f, TempString
Next k

TempString = "Sum" & String(12, " ")
For L = first_radius To last_radius Step 1
    If (L <> (number_of_radii + 1)) Then
        sum = 0
        If (table_type = 1) Then
            For m = 1 To number_of_segments Step 1
```

```
                    sum = sum + sector_population(m, L)
                Next m
            Else
                For m = 1 To number_of_segments Step 1
                    For n = 1 To L Step 1
                        sum = sum + sector_population(m, n)
                    Next n
                Next m
            End If
            TempString = TempString & Justify(10, sum, "#")
        Else
            sum = 0
            For m = 1 To number_of_segments Step 1
                For n = 1 To number_of_radii Step 1
                    sum = sum + sector_population(m, n)
                Next n
            Next m
            TempString = TempString & Justify(10, sum, "#")
        End If
    Next L
    Print #f, TempString

    radius = last_radius + 1

Next j

End Sub

Sub Print_Table_File_to_Printer()
    'This function creates a temporary file, and then
    'writes into that file the information for the user
    'to view on the screen.  This file is then sent to a
    'text form, where the user can view it and scroll
    'the various parts of the data.

    Dim filenum As Integer
    Dim Filename1 As String
    Dim Filename2 As String
    Dim strHeader1 As String
    Dim strHeader2 As String
    Dim strFooter1 As String
    Dim strFooter2 As String
    Dim i As Integer
    Dim iPageNO As Integer
    Dim Pathname As String
    Dim iCopy As Integer
'    Dim PtF As Boolean

    Unload frmDisclaimer

    'set up page format strings
    strHeader1 = "SECPOP2000"
    Call parse_path_and_file(frmProblem_Data.caption, Pathname, strHeader2)
'    strFooter2 = Format(Time, "m/d/yy h:mm")
    strFooter2 = Date & " " & Time

    'Write the actual data to file.
    Filename1 = "$$TEMP$$.$$$"
    filenum = FreeFile
    Open Filename1 For Output As filenum
    Call Create_Temp_Table_File(filenum)
    Close #filenum

    'Reset to printer the destination of info generated
    'by the print_MACCS_input function.
```

```
    On Error GoTo PrintErrHandler
    With frmMain.dlgSecpop
        .Flags = cdlPDHidePrintToFile
        .DialogTitle = "Print Problem File"
        .ShowPrinter
        iCopy = .Copies
'       PtF = .Flags
    End With

'    If PtF Then
'        'If print to file checked
'        With frmMain.dlgSecpop
''            .Flags = 0
'            .Flags = cdlOFNOverwritePrompt
'            .DialogTitle = "Save Table to File"
'            .Filename = StripExt(.Filename)
'            .Filter = "All Files (*.*)|*.*|" & _
'                      "MACCS Files (*.inp)|*.inp|" & _
'                      "Comma Separated Variables (*.csv)|*.csv|" & _
'                      "Text Files(*.txt)|*.txt|" & _
'                      "Site Files (*.sit)|*.sit|" & _
'                      "Problem Files (*.prb)|*.prb|"
'            .FilterIndex = 4    'Default filter = *.txt
'            .Filename = ""
'            .ShowSave
'            Filename2 = .Filename
'        End With
'        Name Filename1 As Filename2    ' Move and rename file
'        Close filenum
'    Else
        'otherwise
        filenum = FreeFile
        Open Filename1 For Input As filenum
        For i = 1 To iCopy
            Close #filenum
            filenum = FreeFile
            Open Filename1 For Input As filenum
            Call PrinterUtility(strHeader1, strHeader2, strFooter1, strFooter2, filenum)
        Next i
        Close filenum
        Kill Filename1
'    End If

    Exit Sub

PrintErrHandler:

    'Deal with errors such as printer is off line or file not found.

    'MsgBox "Device error - check printer or other print device."
    Exit Sub

End Sub

Private Function StripExt(name)

    Dim X As Integer

    X = InStr(1, name, ".")
    If X <> 0 Then
        StripExt = Mid(name, 1, X - 1)
    Else
        StripExt = name
    End If

End Function
```

H.10 Visual Basic Code Module: Probcode.bas

```
Attribute VB_Name = "Probcode"
'******************************************************************************
'  Module:  ProbCode
'
'  Project: SECPOP
'
'  Description:
'
'  Modification History
'  Date       By   Description
'  --------   ---  -----------------------------------------------------------
'
'
'
'******************************************************************************

Option Explicit


Sub browse_site_file()

    On Error GoTo DialogError

    With frmMain.dlgSecpop
        .Filter = "Site Files (*.sit)|*.sit|All Files|*.*"
        .DialogTitle = "Select a Site file to open"
        .FilterIndex = 1
        .Filename = ""
'        .InitDir = InitSiteFilePath
        If InitSiteFilePathChanged = True Then
            .InitDir = frmSetup.txtSite_Path.text
            InitSiteFilePathChanged = False
        Else
            .InitDir = InitSiteFilePath
        End If
        .ShowOpen
    End With

    'reset init dir

    InitSiteFilePath = Left(frmMain.dlgSecpop.Filename, _
                            (InStrRev(frmMain.dlgSecpop.Filename, "\") - 1))


    'this procedure allows the user to view and select previously created
    'site files when creating a new problem.

    Dim Flags As Integer, Cancel As Integer
    Dim Fname As String
    Static Pathname As String
    Fname = frmMain.dlgSecpop.Filename

    'Initialize file dialogue box.


    frmProblem_Data.txtSite_File_Name.text = Fname

    'Mark file as modified.

'    frmMain.mnuSave_ProblemCom.Enabled = True
'    frmMain.mnuSave_As_ProblemCom.Enabled = True

    If (InStr(frmProblem_Data.caption, " (Modified)") = 0) Then
```

```
        frmProblem_Data.caption = frmProblem_Data.caption + " (Modified)"
    End If

    'Call procedure to open site file and read in values.

    open_site_from_problem (Fname)

    'If cancel was selected in file dialogue box, reset focus to
    'site file name.

    frmProblem_Data.txtSite_File_Name.SetFocus

DialogError:
    'User pressed the Cancel button
    Exit Sub

End Sub

Sub default_regions()

    Dim h As Integer, i As Integer, j As Integer, k As Integer
    Dim m As Integer, n As Integer
    Dim num_sectors As Integer
    Dim dividend As Integer, remainder As Integer

    num_sectors = (((number_of_radii - 1) * number_of_segments) + 1)
    k = 2
    frmProblem_Data.lblNumber_of_regions.Tag = Str$(number_of_radii)

    For i = 0 To (number_of_segments)
        region_index(1, i) = 1
    Next i

    If (num_sectors <= max_econ_regions) Then

        number_econ_regions = num_sectors

        For i = 0 To (number_of_segments)
            For j = 2 To (number_of_radii)
                region_index(j, i) = k
                k = k + 1
            Next j
        Next i
    Else
        remainder = Fix((number_of_radii - 1) Mod 6)
        If remainder = 0 Then
            dividend = Fix((number_of_radii - 1) / 6)
        Else: remainder = Fix((number_of_radii - 1) Mod 5)
            dividend = Fix((number_of_radii - 1) / 5)
        End If

        For h = 0 To (number_of_segments)
            m = 2
            If (remainder > dividend) Then
                For m = 2 To (remainder + 1)
                    region_index(m, h) = k
                Next m
                k = k + 1
            End If

            For i = 1 To 5
                For j = m To (m + dividend - 1)
                    region_index(j, h) = k
                Next j
            m = j
            k = k + 1
```

```
                    Next i

                    If remainder <= dividend Then
                       For i = m To (number_of_radii)
                             region_index(i, h) = k
                          Next i
                          k = k + 1
                    End If
            Next h
            number_econ_regions = 97
    End If

    frmProblem_Data.lblNumber_of_regions.caption = Str$(number_econ_regions)

End Sub

Sub new_problem()

    'This procedure initializes a new problem to default settings
    'and opens the problem form.

    Dim i As Integer, j As Integer
    'Initialize settings to default.


    frmProblem_Data.lblNumber_of_regions.caption = ""
    For i = 0 To number_of_segments - 1
            For j = 2 To max_number_of_radii
                    region_index(j, i) = 0
            Next j
    Next i

'     frmMain.mnuSave_ProblemCom.Enabled = False
'     frmMain.mnuSave_As_ProblemCom.Enabled = False
'     frmProblem_Data.cmdSave.Enabled = False
'     frmProblem_Data.cmdSave_As.Enabled = False
    frmProblem_Data.caption = CurDir$ + "\" + "NEW_PROB.PRB"
'     frmProblem_Data.txtSite_File_Name.text = (frmSetup.txtSite_Path.text + "\*.SIT")
    frmProblem_Data.txtSite_File_Name.text = frmSite_Data.caption
    frmProblem_Data.vsbRadius_Number.Value = 1
    frmProblem_Data.txtRadial_Distance1.text = ""
    frmProblem_Data.txtRadial_Distance2.text = ""
    frmProblem_Data.txtRadial_Distance3.text = ""
    frmProblem_Data.txtPopulation_multiplier.text = "1.000"
    frmProblem_Data.txtProblem_Remarks.text = ""
    frmProblem_Data.optKilometers.Tag = "Previously_True"
    frmProblem_Data.optKilometers.Value = True
    frmProblem_Data.optKilometers.Tag = ""

    For i = 0 To max_number_of_radii Step 1
            radial_distance(i) = 0
    Next i

    number_of_radii = 0
    number_econ_regions = 0
'     x_position = 0
'     y_position = 0

    'Show new problem form

    frmProblem_Data.Show

End Sub

Sub open_problem()
```

```
    On Error GoTo DialogError

    With frmMain.dlgSecpop
        .Flags = 0
        .Filter = "Problem Files (*.PRB)|*.PRB|All Files|*.*"
        .DialogTitle = "Select a Problem file to open"
        .FilterIndex = 1
        .Filename = ""
'        .InitDir = InitProblemFilePath
        If InitProblemFilePathChanged = True Then
            .InitDir = frmSetup.txtProblem_path.text
            InitProblemFilePathChanged = False
        Else
            .InitDir = InitProblemFilePath
        End If
        .ShowOpen
    End With

    'reset init dir

    InitProblemFilePath = Left(frmMain.dlgSecpop.Filename, _
                               (InStrRev(frmMain.dlgSecpop.Filename, "\") - 1))


        'Open a previously created problem from file.

        Dim site_file_name As String, longitude As String, latitude As String, units_of_measure
As String
        Dim site_name As String, site_remarks As String, problem_remarks As String, dummy As
String
        Dim longitude_degrees As Integer, longitude_minutes As Integer, longitude_seconds As
Integer
        Dim latitude_degrees As Integer, latitude_minutes As Integer, latitude_seconds As Integer
        Dim i  As Integer, j As Integer
        Dim Flags As Integer, Cancel As Integer
        Dim Fname As String
        Static Pathname As String

        'Define as constants the file handle numbers for problem and site files,
        'and the file buffer size.

        Const problem_file = 10, site_file = 20, file_buffer = 1024

        'Initialize file dialog box.

'qqq         DefaultExt = "*.PRB"
'qqq         DialogTitle = "Problem Open"
'qqq         Pathname = frmSetup.txtProblem_path.text
'qqq         BackColor = WHITE
'qqq         ForeColor = BLACK

        'Enable local error checking for file errors.

'qqq         On Local Error GoTo open_prob_error

        'Open file dialog box.

'qqq         Call FileOpen(fname, pathname, DefaultExt, DialogTitle, forecolor, backcolor, Flags,
Cancel)

        'If user does not select cancel from file dialogue box.

'qqq         If Not Cancel Then

                'if fname is not in current directory, add path.
```

```
                If Pathname <> "" Then Fname = Pathname + "\" + Fname

                Fname = frmMain.dlgSecpop.Filename

                'Reset the table position variables.

'                x_position = 0
'                y_position = 0

                'Open filename and read problem information into temporary variables.

                Open Fname For Input Access Read Lock Write As #problem_file Len = file_buffer
                Line Input #problem_file, dummy
                Line Input #problem_file, dummy
                Line Input #problem_file, site_file_name
                Line Input #problem_file, dummy
                Input #problem_file, number_of_radii
                Line Input #problem_file, dummy
                Line Input #problem_file, units_of_measure
                Line Input #problem_file, dummy
                For i = 1 To number_of_radii Step 1
                        Input #problem_file, radial_distance(i)
                Next i
                For i = number_of_radii + 1 To max_number_of_radii Step 1
                    radial_distance(i) = 0#
                Next i
                Line Input #problem_file, dummy
                Input #problem_file, Population_multiplier
                Line Input #problem_file, dummy
                Input #problem_file, number_econ_regions
                Line Input #problem_file, dummy
                For i = 0 To number_of_segments - 1
                        For j = 1 To number_of_radii
                                Input #problem_file, region_index(j, i)
                        Next j
                Next i
                Line Input #problem_file, dummy
                Line Input #problem_file, problem_remarks
                Close problem_file

                'Transfer information from temporary variables to correct
                'form locations.

'                frmProblem_Data.caption = Fname
                frmProblem_Data.txtSite_File_Name.text = site_file_name
                If units_of_measure = "Kilometers" Then
                        frmProblem_Data.optKilometers.Tag = "Reading In File"
                        frmProblem_Data.optKilometers.Value = True
                        frmProblem_Data.optKilometers.Tag = ""
                Else
                        frmProblem_Data.optMiles.Tag = "Reading In File"
                        frmProblem_Data.optMiles.Value = True
                        frmProblem_Data.optMiles.Tag = ""
                End If
                frmProblem_Data.vsbRadius_Number.Value = 1
                frmProblem_Data.txtRadial_Distance1.text = Trim(Right$("    " +
Format$(radial_distance(1), "0.0000;0.0000;     "), 9))
                frmProblem_Data.txtRadial_Distance2.text = Trim(Right$("    " +
Format$(radial_distance(2), "0.0000;0.0000;     "), 9))
                frmProblem_Data.txtRadial_Distance3.text = Trim(Right$("    " +
Format$(radial_distance(3), "0.0000;0.0000;     "), 9))
                frmProblem_Data.txtPopulation_multiplier.text = Trim(Right$("    " +
Format$(Population_multiplier, "0.0000;0.0000;     "), 9))
                frmProblem_Data.txtProblem_Remarks.text = problem_remarks

                'If a site file is listed, call function to read in site data.
```

```
                If site_file_name <> "" Then
                        open_site_from_problem (site_file_name)
                End If

                'Set the number of economic regions to new value, and re-paint
                'data form.

                frmProblem_Data.lblNumber_of_regions.Tag = Str$(number_of_radii)
                frmProblem_Data.lblNumber_of_regions.caption = Str$(number_econ_regions)

                frmProblem_Data.caption = Fname

                'If problem data form is not visible, show it.

                If (frmProblem_Data.Visible = False) Then
                        frmProblem_Data.Show
                End If

'qqq         End If

''''''         Exit Sub

'qqq open_prob_error:

        'On error, inform the user, and exit subroutine.

'qqq         MsgBox Error$(Err), vbOK, "Error"
        'MSGBOX "File error - check spelling of file names."
'qqq         Exit Sub
DialogError:
     'User pressed the Cancel button

    frmProblem_Data.Show

    Exit Sub
End Sub

Sub open_site_from_problem(file_name As String)

        'Open site from problem and fill site form with data from file.

        Dim site_name As String, site_remarks As String, dummy As String
        Dim longitude_degrees As Integer, longitude_minutes As Integer, longitude_seconds As
Integer
        Dim latitude_degrees As Integer, latitude_minutes As Integer, latitude_seconds As Integer

        'Declare as constant the file handle number for site file and the
        'file buffer size.

        Const site_file = 20, file_buffer = 1024

        'Enable local error checking.

        On Local Error GoTo open_site_problem_file_error

        'Open site file named on problem form, and read input into
        'temporary variables.

        Open file_name For Input Access Read Lock Write As #site_file Len = file_buffer
        Line Input #site_file, dummy
        Line Input #site_file, site_name
        Line Input #site_file, dummy
        Line Input #site_file, dummy
        Input #site_file, longitude_degrees
        Input #site_file, longitude_minutes
```

```
        Input #site_file, longitude_seconds
        Line Input #site_file, dummy
        Input #site_file, latitude_degrees
        Input #site_file, latitude_minutes
        Input #site_file, latitude_seconds
        Line Input #site_file, dummy
        Line Input #site_file, site_remarks
        Close site_file

        'Transfer values from temp variables to site form.

        frmSite_Data.caption = file_name
        frmSite_Data.txtSite_Name.text = site_name
        frmSite_Data.txtLongitude_Degrees.text = Trim(Str$(longitude_degrees))
        frmSite_Data.txtLongitude_Minutes.text = Trim(Str$(longitude_minutes))
        frmSite_Data.txtLongitude_Seconds.text = Trim(Str$(longitude_seconds))
        frmSite_Data.txtLatitude_Degrees.text = Trim(Str$(latitude_degrees))
        frmSite_Data.txtLatitude_Minutes.text = Trim(Str$(latitude_minutes))
        frmSite_Data.txtLatitude_Seconds.text = Trim(Str$(latitude_seconds))
        frmSite_Data.txtSite_Remarks.text = site_remarks

        'verify lat and long
         verify_input (2)

        Exit Sub

open_site_problem_file_error:

        'If there is a problem, tell the user, and return to the form where
        'a correction can be made.

        If frmProblem_Data.Visible = False Then
                frmProblem_Data.Show
                MsgBox "Error: Unable to Open Site File."
                frmProblem_Data.txtSite_File_Name.SetFocus
                frmProblem_Data.Hide
        Else
                MsgBox "Unable to Open Site File.", vbOK, "File Error"
                frmProblem_Data.txtSite_File_Name.SetFocus
        End If
        Exit Sub

End Sub

'Sub paint_region(x_offset As Integer, y_offset As Integer)
'
'        Dim i As Integer, j As Integer, k As Integer, X As Integer, Y As Integer
''        Dim directions() As String
'        frmRegion.lblnum_regions.caption = Str$(number_econ_regions)
'        frmRegion.lblnum_radii.caption = Str$(number_of_radii)
'
'        If frmProblem_Data.optMiles.Value = True Then
'                frmRegion.lblUnits.caption = "Miles"
'        Else: frmRegion.lblUnits.caption = "Kilometers"
'        End If
'
'        x_position = x_position + x_offset
'        y_position = y_position + y_offset
'
'        If (number_of_radii < 8) Then
'                x_position = 0
'                If (x_offset <> 0) Then
'                        Exit Sub
'                End If
'        ElseIf (x_position < 0) Then
'                x_position = 0
```

```
'                Exit Sub
'        ElseIf ((x_position + 7) > number_of_radii) Then
'                x_position = x_position - 1
'                Exit Sub
'        End If
'
'        For i = 1 To 7
'                X = x_position + i
'                If radial_distance(X) = 0 Then
'                        frmRegion.lblRdist(i - 1).caption = ""
'                Else: frmRegion.lblRdist(i - 1).caption = Left$(Str$(radial_distance(X)), 4)
'                End If
'        Next i
'
'        k = 0
'
'        For i = 0 To 6
'                If (i = 0) And ((y_position = 16) Or (y_position = -16)) Then
'                        y_position = 0
'                End If
'                If (((i + y_position) > 15) And (y_position > 9)) Then
'                        Y = (i + y_position - 16)
'                ElseIf ((y_position < 0) And ((y_position + i) < 0)) Then
'                        Y = (i + y_position + 16)
'                Else
'                        Y = (i + y_position)
'                End If
'                frmRegion.lblSector(i).caption = directions(Y + 1)
'
'                For j = 1 To 7
'                        X = j + x_position
'                        If ((radial_distance(X) = 0) Or (region_index(X, Y) = 0)) Then
'                                frmRegion.txtsector(k).text = ""
'                                frmRegion.txtsector(k).Enabled = True
'                        ElseIf (X = 1) Then
'                                frmRegion.txtsector(k).Enabled = False
'                                frmRegion.txtsector(k).text = Right$("      " +
Str$(region_index(X, Y)), 6)
'                        Else: frmRegion.txtsector(k).text = Right$("      " +
Str$(region_index(X, Y)), 6)
'                                frmRegion.txtsector(k).Enabled = True
'                        End If
'                k = k + 1
'                Next j
'        Next i
'
'End Sub
'
```

```
Sub print_problem()
    On Error GoTo DialogError

    frmMain.dlgSecpop.Flags = 0
    frmMain.dlgSecpop.ShowPrinter

        'Print the data on a problem form.

'qqq         Dim ForeColor As Integer, BackColor As Integer
        Dim Copies As Integer, Cancel As Integer, i As Integer, j As Integer
        Dim file_name As String, latitude As String, longitude As String, site_file_name As
String, problem_remarks As String
'qqq     date AS STRING (this can probably be deleted after updating code
'        Copies = 1
        'Define as constants the margines for printing.
```

```
        Const left_margin = 10
        Const text_width = 40

                'Execute once for each copy desired by the user.

'               For i = 1 To Copies
                For i = 1 To frmMain.dlgSecpop.Copies

                        'Send data to printer.

                        Printer.Print Tab(left_margin); "File Name:";
                        file_name = frmProblem_Data.caption
                        Call print_text(file_name, text_width, left_margin + Len("Population
Multiplier:  "))
                        Printer.Print
                        Printer.Print Tab(left_margin); "Site File Name:";
                        site_file_name = frmProblem_Data.txtSite_File_Name.text
                        Call print_text(site_file_name, text_width, left_margin + Len("Population
Multiplier:  "))
                        Printer.Print
                        Printer.Print Tab(left_margin); "Number of Radii:";
                        Printer.Print "  "; Tab(left_margin + Len("Population Multiplier:  "));
number_of_radii
                        Printer.Print
                        Printer.Print Tab(left_margin); "Radial Distances:";
                        For j = 1 To number_of_radii Step 1
                                Printer.Print "         "; Tab(left_margin + Len("Population
Multiplier:  ")); radial_distance(j)
                        Next j
                        Printer.Print
                        Printer.Print Tab(left_margin); "Population Multiplier Factor:";
                        Printer.Print "         "; Tab(left_margin + Len("Population Multiplier:
")); Val(frmProblem_Data.txtPopulation_multiplier.text)
                        Printer.Print
                        Printer.Print Tab(left_margin); "Problem Remarks:";
                        problem_remarks = frmProblem_Data.txtProblem_Remarks.text
                        Call print_text(problem_remarks, text_width, left_margin +
Len("Population Multiplier:  "))
                        Printer.NewPage
                Next i

                'Tell printer that it is at the end of the document.
                Printer.EndDoc

'qqq         End If

        Exit Sub

problem_print_error:

        'If an error ocurrs, the user is notified, and the sub routine exited.

        MsgBox "Device error: Check printer or other print device."
        Exit Sub

DialogError:
    'User pressed the Cancel button
    Exit Sub

End Sub

Sub save_as_problem()
    On Error GoTo DialogError

    With frmMain.dlgSecpop
        .Filter = "Problem Files (*.prb)|*prb|All files|*.*"
```

```
        .FilterIndex = 1
        .DialogTitle = "Save File As"
        .Filename = ""
        .Flags = cdlOFNOverwritePrompt
        .DefaultExt = "prb"
'        .InitDir = InitProblemFilePath
        If InitProblemFilePathChanged = True Then
            .InitDir = frmSetup.txtProblem_path.text
            InitProblemFilePathChanged = False
        Else
            .InitDir = InitProblemFilePath
        End If
        .ShowSave
    End With

    'reset init dir

    InitProblemFilePath = Left(frmMain.dlgSecpop.Filename, _
                               (InStrRev(frmMain.dlgSecpop.Filename, "\") - 1))


        'Allow the user to specify a new name for a file that is being saved.

        Dim problem_name As String, problem_remarks As String, dummy As String, caption As String
        Dim longitude_degrees As Integer, longitude_minutes As Integer, longitude_seconds As
Integer
        Dim latitude_degrees As Integer, latitude_minutes As Integer, latitude_seconds As Integer
'qqq         Dim ForeColor As Integer, BackColor As Integer
        Dim Flags As Integer, Cancel As Integer
        Dim Fname As String, Pathname As String
'qqq         Dim DefaultExt As String, DialogTitle As String
        Dim temp As Integer
        Const problem_file = 10, file_buffer = 1024

        'Verify that the user has entered valid input into all necessary fields
        'Do not save until all input is correct.

        temp = verify_input(1)

        If temp = 0 Then
                frmProblem_Data.txtSite_File_Name.SetFocus
                Exit Sub
        ElseIf temp = -1 Then
                frmProblem_Data.txtPopulation_multiplier.SetFocus
                Exit Sub
        ElseIf temp = -2 Then
                frmProblem_Data.txtRadial_Distance1.SetFocus
                Exit Sub
        ElseIf temp = -3 Then
                frmProblem_Data.txtRadial_Distance2.SetFocus
                Exit Sub
        ElseIf temp = -4 Then
                MsgBox "Setting Economic regions to default values"
                Call default_regions
        End If

                'Break fname into two pieces - path and fname.
                caption = frmMain.dlgSecpop.Filename
'qqq                 caption = frmProblem_Data.caption
                Call parse_path_and_file(caption, Pathname, Fname)

                'Initialize file dialogue box.

'qqq                 DefaultExt = "*.PRB"
'qqq                 Pathname = frmSetup.txtProblem_path.text
'qqq                 DialogTitle = "Save As Problem"
```

```
'qqq                BackColor = WHITE
'qqq                ForeColor = BLACK

                'Open file dialogue box.

'qqq                Call FileSave(fname, Pathname, DefaultExt, DialogTitle, ForeColor, BackColor,
Flags, Cancel)

                'If the user did not select cancel in the file dialogue box
                'then proceed.

'qqq                If Not Cancel Then

                    'If the file is not in the current directory, add the path to it.

                    If Pathname <> "" Then Fname = Pathname + "\" + Fname
                    frmProblem_Data.caption = Fname

                    'Call the function to save the problem.

                    Call save_problem

'qqq                End If
DialogError:
    'User pressed the Cancel button
Exit Sub

End Sub

Sub save_problem()

        'Save problem data into previously named file.

        Dim caption As String, i As Integer, j As Integer
        Dim Fname As String, Pathname As String, TempString As String
        Dim temp As Integer, filenum As Integer
        Const problem_file = 20, file_buffer = 1024
        Dim tempstr As String

        'Remove the path from the fname.

        caption = frmProblem_Data.caption
        Call parse_path_and_file(caption, Pathname, Fname)

        'If it is a never-before-named problem, then call save as.

        If Fname = "NEW_PROB.PRB" Then
                Call save_as_problem
        Else

'''             i = InStr(frmProblem_Data.caption, " (Modified)")
'''             If i <> 0 Then
'''                 TempStr = Mid$(frmProblem_Data.caption, 1, i - 1)
'''             Else
'''                 TempStr = frmProblem_Data.caption
'''             End If
'''
'''             i = MsgBox(TempStr & " already exists.  Would you like to overwrite this file? " &
vbCrLf & _
'''                             " (If not please select Save As).", vbYesNoCancel)
'''             If i = vbYes Then

                'Verify that all of the data inputed by the user is correct.  If not,
                'go back and demand a correction.

                temp = verify_input(1)
```

```
                If temp = 0 Then
                        frmProblem_Data.txtSite_File_Name.SetFocus
                        Exit Sub
                ElseIf temp = -1 Then
                        frmProblem_Data.txtPopulation_multiplier.SetFocus
                        Exit Sub
                ElseIf temp = -2 Then
                        frmProblem_Data.txtRadial_Distance1.SetFocus
                        Exit Sub
                ElseIf temp = -3 Then
                        frmProblem_Data.txtRadial_Distance2.SetFocus
                        Exit Sub
                ElseIf temp = -4 Then
                        MsgBox "Setting Economic regions to default values"
                        Call default_regions
                End If

                If Pathname <> "" Then
                        Fname = Pathname + "\" + Fname
                End If

                'Enable local error correction.

                On Local Error GoTo prob_save_error

                'Open output file, and write information to it.

                Open Fname For Output Access Write Lock Read Write As #problem_file Len =
file_buffer
                Print #problem_file, "SECPOP Version Beta 3.0"
                Print #problem_file, "Site File: "
                Print #problem_file, frmProblem_Data.txtSite_File_Name.text
                Print #problem_file, "Number of Radii:"
                Print #problem_file, Format(number_of_radii, "##")
                Print #problem_file, "Units of Measure:"
                If frmProblem_Data.optKilometers.Value = True Then
                        Print #problem_file, "Kilometers"
                Else: Print #problem_file, "Miles"
                End If
                Print #problem_file, "Radial Distances:"
                For i = 1 To number_of_radii Step 1
                        Print #problem_file, Format(radial_distance(i), "####.0000")
                Next i
                Print #problem_file, "Population Multiplier:"
                Print #problem_file, Format(Val(frmProblem_Data.txtPopulation_multiplier.text),
"####.0000")
                Print #problem_file, "Number of Economic regions:"
                Print #problem_file, Format(number_econ_regions, "##")
                Print #problem_file, "Economic regions:"
                For i = 0 To (number_of_segments - 1)
                        For j = 1 To number_of_radii
                                Print #problem_file, Format(region_index(j, i), "00");
                                Print #problem_file, " ";
                        Next j
                        Print #problem_file, ""
                Next i
                Print #problem_file, "Problem Remarks:"
                Print #problem_file, frmProblem_Data.txtProblem_Remarks.text
                Close problem_file

                'Rename the problem window to the file name.

                frmProblem_Data.caption = Fname

'               Else
```

```
'                frmMain.mnuSave_ProblemCom.Enabled = True
'                frmMain.mnuSave_As_ProblemCom.Enabled = True
'                frmProblem_Data.cmdSave.Enabled = True
'                frmProblem_Data.cmdSave_As.Enabled = True
'            End If

        End If

        'If problem form is visible, set the focus to close.

        If frmProblem_Data.Visible = True Then frmProblem_Data.cmdClose.SetFocus

        Exit Sub

prob_save_error:

        'If an error occurs, tell the user and exit the subroutine.

        MsgBox "A FILE ERROR HAS OCCURRED.  PLEASE CHECK ALL FILE NAMES.  (File may be
readonly.)"
        Exit Sub

End Sub
```

H.11 Visual Basic Code Module: ReadFile.bas

```
Attribute VB_Name = "ReadFile"
'*******************************************************************************
'  Module:  Readfile
'
'  Project: SECPOP
'
'  Description: Subroutine to read back in MACCS files to review
'               rosettes and use other utilities
'
'  Modification History
'  Date      By   Description
'  --------  ---  ------------------------------------------------------------
'  20020213  cwm  Created module
'  20020610  cwm  Created ability to read both CSV and MACCS file formats
'*******************************************************************************

Option Explicit

Sub ReadPreviousOutput()
    'Load in previously saved output data so that the user
    'can view it again, in any of the standard forms.

    Dim tempinteger As Integer, tempsingle As Single
    Dim filenum As Integer, i As Integer, j As Integer, k As Integer
    Dim TempString As String
    Dim Flags As Integer, Cancel As Integer
    Dim Fname As String
    Static Pathname As String
    Dim DefaultExt As String, DialogTitle As String
    Dim CSVorMACCS As String

    Unload frmDisclaimer
    On Error GoTo Previous_Error

    frmMain.dlgSecpop.Flags = 0
    With frmMain.dlgSecpop
        .CancelError = True     ' Set CancelError is True
        .Flags = cdlOFNHideReadOnly  ' Set flags
        .Filter = "All Files (*.*)|*.*|" & _
                  "MACCS Files (*.inp)|*.inp|" & _
                  "Comma Separated Variables (*.csv)|*.csv|" & _
                  "Text Files(*.txt)|*.txt|All Files|*.*"

'       If frmSetup.optCSV.Value = True Then
'           .FilterIndex = 3    'Default filter = *.csv
'       Else
'           .FilterIndex = 2    'Default filter = *.inp
'       End If
        .Filename = ""
        .FilterIndex = 2
        .DialogTitle = "Load Previous Data"
'       .InitDir = InitOutputFilePath
        If InitOutputFilePathChanged = True Then
            .InitDir = frmSetup.txtOutput_path.text
            InitOutputFilePathChanged = False
        Else
            .InitDir = InitOutputFilePath
        End If
        .ShowOpen ' Display the fileopen dialog box
        Fname = .Filename
    End With

    'reset init dir
```

```
    InitOutputFilePath = Left(frmMain.dlgSecpop.Filename, _
                              (InStrRev(frmMain.dlgSecpop.Filename, "\") - 1))

    frmSite_Data.caption = " "

    filenum = FreeFile
    Open Fname For Input As filenum
    Line Input #filenum, TempString  'TempString = Input$(80, #filenum)  'Read in the header
information.
    Close filenum

    i = InStr(1, TempString, "CSV")
    j = InStr(1, TempString, "MAC")
    If i > 0 Then
        Call ReadPreviousCSVOutput(Fname)
    ElseIf j > 0 Then
        Call ReadPreviousMACCSOutput(Fname)
    Else
        MsgBox "ERROR: SECPOP cannot determine if the specified file is " & vbCrLf & _
               "CSV or MACCS format.  The program looks for the word CSV" & vbCrLf & _
               "or MACCS in the first line to decide." & vbCrLf & vbCrLf & _
               "The specified file's first line reads:  " & vbCrLf & vbCrLf & _
               TempString

    End If

Previous_Error:

    'If there is an error, identify type if possible,
    'inform the user of the problem, and leave subroutine.

    If (Err = 62) Then
        MsgBox "Error:  Attempt has been made to read past the end of file.  Data file has been
ccorrupted."
    ElseIf (Err = 53) Or (Err = 75) Or (Err = 76) Or (Err = 55) Then
        MsgBox "Invalid filename or path, or file access denied"
    ElseIf (Err = 61) Or (Err = 68) Or (Err = 71) Or (Err = 72) Then
        MsgBox "The disk drive is not ready, or a file error has occured"
    Else
    End If
    Exit Sub


End Sub

Sub ReadPreviousMACCSOutput(Fname)

    'Load in previously saved output data so that the user
    'can view it again, in any of the standard forms.

    Dim tempinteger As Integer, tempsingle As Single
    Dim NbrCropCategories As Integer, NbrWaterPathwayIsotopes As Integer
    Dim filenum As Integer, i As Integer, j As Integer, k As Integer
    Dim TempString As String
    Dim ForeColor As Integer, BackColor As Integer
    Dim Flags As Integer, Cancel As Integer
    Static Pathname As String
    Dim DefaultExt As String, DialogTitle As String

    'Unload frmDisclaimer
    On Error GoTo Previous_Error
'
'    frmMain.dlgSecpop.Flags = 0
'    With frmMain.dlgSecpop
'        .CancelError = True     ' Set CancelError is True
```

```
'        .Flags = cdlOFNHideReadOnly  ' Set flags
'        .Filter = "All Files (*.*)|*.*|" & _
'                  "MACCS Files (*.inp)|*.inp|" & _
'                  "Comma Separated Variables (*.csv)|*.csv|" & _
'                  "Text Files(*.txt)|*.txt|All Files|*.*"
'
'        If frmSetup.optCSV.Value = True Then
'            .FilterIndex = 3     'Default filter = *.csv
'        Else
'            .FilterIndex = 2     'Default filter = *.inp
'        End If
'        .Filename = ""
'        .DialogTitle = "Load Previous Data"
''        .InitDir = InitOutputFilePath
'        If InitOutputFilePathChanged = True Then
'            .InitDir = frmSetup.txtOutput_path.text
'            InitOutputFilePathChanged = False
'        Else
'            .InitDir = InitOutputFilePath
'        End If
'        .ShowOpen ' Display the fileopen dialog box
'        Fname = .Filename
'    End With
'
'    'reset init dir
'
'    InitOutputFilePath = Left(frmMain.dlgSecpop.Filename, _
'                              (InStrRev(frmMain.dlgSecpop.Filename, "\") - 1))
'
    frmSite_Data.caption = " "

    filenum = FreeFile
    Open Fname For Input As filenum

    TempString = Input$(40, #filenum)  'Read in the header information.
    Line Input #filenum, TempString
    frmSite_Data.txtSite_Name.text = TempString


    TempString = Input$(6, #filenum)
    TempString = Input$(2, #filenum)
    frmSite_Data.txtLatitude_Degrees.text = LTrim$(TempString)
    TempString = Input$(1, #filenum)
    TempString = Input$(2, #filenum)
    frmSite_Data.txtLatitude_Minutes.text = LTrim$(TempString)
    TempString = Input$(1, #filenum)
    TempString = Input$(2, #filenum)
    frmSite_Data.txtLatitude_Seconds.text = LTrim$(TempString)


    TempString = Input$(9, #filenum)
    TempString = Input$(3, #filenum)
    frmSite_Data.txtLongitude_Degrees.text = LTrim$(TempString)
    TempString = Input$(1, #filenum)
    TempString = Input$(2, #filenum)
    frmSite_Data.txtLongitude_Minutes.text = LTrim$(TempString)
    TempString = Input$(1, #filenum)
    TempString = Input$(2, #filenum)
    frmSite_Data.txtLongitude_Seconds.text = LTrim$(TempString)

    TempString = Input$(26, #filenum)
    Input #filenum, tempsingle
    Line Input #filenum, TempString
    frmProblem_Data.txtPopulation_multiplier.text = Str$(tempsingle)

    'Pull in various numbers and strings from the file,
```

```
    'discarding extraneous labels and information.

    Input #filenum, tempsingle
    number_of_radii = tempsingle

    'Discard next 5 lines from file

'    For i = 1 To 5 Step 1
'        Line Input #filenum, TempString
'    Next i

    Line Input #filenum, TempString   ' spatial intervals
    Line Input #filenum, TempString   ' wind directions
    Input #filenum, NbrCropCategories
    Line Input #filenum, TempString   ' crop categories
    Input #filenum, NbrWaterPathwayIsotopes
    Line Input #filenum, TempString   ' water pathway isotopes
    Line Input #filenum, TempString   ' watersheds

    'Determine number of economic regions.

    Input #filenum, tempsingle
    number_econ_regions = tempsingle
    Line Input #filenum, TempString

    'Fill number of radii array from file data.

    Line Input #filenum, TempString

    For i = 1 To number_of_radii
        Input #filenum, tempsingle
        radial_distance(i) = tempsingle
    Next i

    'Fill total population array for each segment and radii.

    Line Input #filenum, TempString

    For i = 1 To number_of_segments
        For j = 1 To number_of_radii
            Input #filenum, tempsingle
            sector_population(i, j) = tempsingle
        Next j
    Next i

    'Fill fraction land array for each segment and radii.

    Line Input #filenum, TempString

    For i = 1 To number_of_segments
        For j = 1 To number_of_radii
            Input #filenum, tempsingle
            sector_frclnd(i, j) = tempsingle
        Next j
    Next i

    'Fill region index array.

    Line Input #filenum, TempString

    For i = 0 To (number_of_segments - 1)
        For j = 1 To number_of_radii
            TempString = Input$(2, #filenum)
            If Val(TempString) = 0 Then
                TempString = Input$(2, #filenum)
            End If
```

```
            region_index(j, i) = Val(TempString)
        Next j
    Next i
    Line Input #filenum, TempString

    'Suck in and throw away the dummy watershed indices.

    Line Input #filenum, TempString
    For i = 1 To number_of_segments
        Line Input #filenum, TempString
    Next i

    'Suck in and throw away the dummy crop season and share info.

    For i = 1 To NbrCropCategories + 1
        Line Input #filenum, TempString
    Next i

    'Suck in and throw away the dummy watershed definition.

    For i = 1 To NbrWaterPathwayIsotopes + 1
        Line Input #filenum, TempString
    Next i

    'Fill regional data array structs.

    Line Input #filenum, TempString
    For i = 1 To number_econ_regions
        Input #filenum, tempsingle
        TempString = Input$(10, #filenum)
        Input #filenum, tempsingle
        econ_data(i).region_frmfrc = tempsingle
        Input #filenum, tempsingle
        econ_data(i).region_dpf = tempsingle
        Input #filenum, tempsingle
        econ_data(i).region_asfp = tempsingle
        Input #filenum, tempsingle
        econ_data(i).region_vfrm = tempsingle
        Input #filenum, tempsingle
        econ_data(i).region_vnfrm = tempsingle
    Next i

    Close filenum

    'Update site and problem form captions and remarks.

'   frmSite_Data.caption = Fname
    frmSite_Data.txtSite_Remarks.text = ""
'   frmProblem_Data.caption = Fname
    frmProblem_Data.txtProblem_Remarks.text = ""

    frmProblem_Data.caption = CurDir$ + "\" + "NEW_PROB.PRB"
    frmProblem_Data.txtSite_File_Name.text = (CurDir$ + "\*.SIT")
'   frmProblem_Data.txtSite_File_Name.text = (frmSetup.txtSite_Path.text + "\*.SIT")

    'Update problem form radial distances, units, and number of regions.

    frmProblem_Data.txtRadial_Distance1.text = Str$(radial_distance(1))
    frmProblem_Data.txtRadial_Distance2.text = Str$(radial_distance(2))
    frmProblem_Data.txtRadial_Distance3.text = Str$(radial_distance(3))

    frmProblem_Data.optKilometers.Tag = "ReadingIn"
    frmProblem_Data.optKilometers.Value = True
    frmProblem_Data.optKilometers.Tag = ""
    frmProblem_Data.lblNumber_of_regions.caption = _
        Str$(number_econ_regions)
```

```
    'verify lat and long on site data form
     verify_input (2)

    frmRosette.blnInputDataChanged = False

    Exit Sub

Previous_Error:

    'If there is an error, identify type if possible,
    'inform the user of the problem, and leave subroutine.

    If (Err = 62) Then
        MsgBox "Error:  Attempt has been made to read past the end of file.  Data file has been
ccorrupted."
    ElseIf (Err = 53) Or (Err = 75) Or (Err = 76) Or (Err = 55) Then
        MsgBox "Invalid filename or path, or file access denied"
    ElseIf (Err = 61) Or (Err = 68) Or (Err = 71) Or (Err = 72) Then
        MsgBox "The disk drive is not ready, or a file error has occured"
    Else
    End If
    Exit Sub

End Sub

Sub ReadPreviousCSVOutput(Fname)

    'Load in previously saved output data so that the user
    'can view it again, in any of the standard forms.

    Dim tempinteger As Integer, tempsingle As Single
    Dim NbrCropCategories As Integer, NbrWaterPathwayIsotopes As Integer
    Dim filenum As Integer, i As Integer, j As Integer, k As Integer
    Dim TempString As String
    Dim ForeColor As Integer, BackColor As Integer
    Dim Flags As Integer, Cancel As Integer
    Static Pathname As String
    Dim DefaultExt As String, DialogTitle As String

    'Unload frmDisclaimer
    On Error GoTo Previous_Error
'
'    frmMain.dlgSecpop.Flags = 0
'    With frmMain.dlgSecpop
'        .CancelError = True    ' Set CancelError is True
'        .Flags = cdlOFNHideReadOnly  ' Set flags
'        .Filter = "All Files (*.*)|*.*|" & _
'                  "MACCS Files (*.inp)|*.inp|" & _
'                  "Comma Separated Variables (*.csv)|*.csv|" & _
'                  "Text Files(*.txt)|*.txt|All Files|*.*"
'
'        If frmSetup.optCSV.Value = True Then
'            .FilterIndex = 3    'Default filter = *.csv
'        Else
'            .FilterIndex = 2    'Default filter = *.inp
'        End If
'        .Filename = ""
'        .DialogTitle = "Load Previous Data"
''        .InitDir = InitOutputFilePath
'        If InitOutputFilePathChanged = True Then
'            .InitDir = frmSetup.txtOutput_path.text
'            InitOutputFilePathChanged = False
'        Else
'            .InitDir = InitOutputFilePath
'        End If
```

```
'        .ShowOpen ' Display the fileopen dialog box
'        Fname = .Filename
'    End With
'
'    'reset init dir
'
'    InitOutputFilePath = Left(frmMain.dlgSecpop.Filename, _
'                              (InStrRev(frmMain.dlgSecpop.Filename, "\") - 1))
'
    frmSite_Data.caption = " "

    filenum = FreeFile
    Open Fname For Input As filenum

    TempString = Input$(40, #filenum)  'Read in the header information.
    Line Input #filenum, TempString
    frmSite_Data.txtSite_Name.text = TempString


    TempString = Input$(6, #filenum)
    TempString = Input$(2, #filenum)
    frmSite_Data.txtLatitude_Degrees.text = LTrim$(TempString)
    TempString = Input$(1, #filenum)
    TempString = Input$(2, #filenum)
    frmSite_Data.txtLatitude_Minutes.text = LTrim$(TempString)
    TempString = Input$(1, #filenum)
    TempString = Input$(2, #filenum)
    frmSite_Data.txtLatitude_Seconds.text = LTrim$(TempString)


    TempString = Input$(9, #filenum)
    TempString = Input$(3, #filenum)
    frmSite_Data.txtLongitude_Degrees.text = LTrim$(TempString)
    TempString = Input$(1, #filenum)
    TempString = Input$(2, #filenum)
    frmSite_Data.txtLongitude_Minutes.text = LTrim$(TempString)
    TempString = Input$(1, #filenum)
    TempString = Input$(2, #filenum)
    frmSite_Data.txtLongitude_Seconds.text = LTrim$(TempString)

    TempString = Input$(26, #filenum)
    Input #filenum, tempsingle
    Line Input #filenum, TempString
    frmProblem_Data.txtPopulation_multiplier.text = Str$(tempsingle)

    'Pull in various numbers and strings from the file,
    'discarding extraneous labels and information.

    Input #filenum, tempsingle
    number_of_radii = tempsingle

    'Discard next 5 lines from file

'    For i = 1 To 5 Step 1
'        Line Input #filenum, TempString
'    Next i

    Line Input #filenum, TempString    ' spatial intervals
    Line Input #filenum, TempString    ' wind directions
    Input #filenum, NbrCropCategories
    Line Input #filenum, TempString    ' crop categories
    Input #filenum, NbrWaterPathwayIsotopes
    Line Input #filenum, TempString    ' water pathway isotopes
    Line Input #filenum, TempString    ' watersheds

    'Determine number of economic regions.
```

```
Input #filenum, tempsingle
number_econ_regions = tempsingle
Line Input #filenum, TempString

'Fill number of radii array from file data.

Line Input #filenum, TempString

For i = 1 To number_of_radii
    Input #filenum, tempsingle
    radial_distance(i) = tempsingle
Next i

'Fill total population array for each segment and radii.

Line Input #filenum, TempString

For i = 1 To number_of_segments
    For j = 1 To number_of_radii
        Input #filenum, tempsingle
        sector_population(i, j) = tempsingle
    Next j
Next i

'Fill fraction land array for each segment and radii.

Line Input #filenum, TempString

For i = 1 To number_of_segments
    For j = 1 To number_of_radii
        Input #filenum, tempsingle
        sector_frclnd(i, j) = tempsingle
    Next j
Next i

'Fill region index array.

Line Input #filenum, TempString

For i = 0 To (number_of_segments - 1)
    For j = 1 To number_of_radii
        TempString = Input$(2, #filenum)
        If Val(TempString) = 0 Then
            TempString = Input$(2, #filenum)
        End If
        region_index(j, i) = Val(TempString)
    Next j
Next i
Line Input #filenum, TempString

'Suck in and throw away the dummy watershed indices.

Line Input #filenum, TempString
For i = 1 To number_of_segments
    Line Input #filenum, TempString
Next i

'Suck in and throw away the dummy crop season and share info.

For i = 1 To NbrCropCategories + 1
    Line Input #filenum, TempString
Next i

'Suck in and throw away the dummy watershed definition.
```

```
    For i = 1 To NbrWaterPathwayIsotopes + 1
        Line Input #filenum, TempString
    Next i

    'Fill regional data array structs.

    Line Input #filenum, TempString
    For i = 1 To number_econ_regions
        Input #filenum, tempsingle
        TempString = Input$(10, #filenum)
        Input #filenum, tempsingle
        econ_data(i).region_frmfrc = tempsingle
        Input #filenum, tempsingle
        econ_data(i).region_dpf = tempsingle
        Input #filenum, tempsingle
        econ_data(i).region_asfp = tempsingle
        Input #filenum, tempsingle
        econ_data(i).region_vfrm = tempsingle
        Input #filenum, tempsingle
        econ_data(i).region_vnfrm = tempsingle
    Next i

    Close filenum

    'Update site and problem form captions and remarks.

'    frmSite_Data.caption = Fname
    frmSite_Data.txtSite_Remarks.text = ""
'    frmProblem_Data.caption = Fname
    frmProblem_Data.txtProblem_Remarks.text = ""

    frmProblem_Data.caption = CurDir$ + "\" + "NEW_PROB.PRB"
    frmProblem_Data.txtSite_File_Name.text = (CurDir$ + "\*.SIT")
'    frmProblem_Data.txtSite_File_Name.text = (frmSetup.txtSite_Path.text + "\*.SIT")

    'Update problem form radial distances, units, and number of regions.

    frmProblem_Data.txtRadial_Distance1.text = Str$(radial_distance(1))
    frmProblem_Data.txtRadial_Distance2.text = Str$(radial_distance(2))
    frmProblem_Data.txtRadial_Distance3.text = Str$(radial_distance(3))

    frmProblem_Data.optKilometers.Tag = "ReadingIn"
    frmProblem_Data.optKilometers.Value = True
    frmProblem_Data.optKilometers.Tag = ""
    frmProblem_Data.lblNumber_of_regions.caption = _
        Str$(number_econ_regions)

    'verify lat and long on site data form
     verify_input (2)

    frmRosette.blnInputDataChanged = False

    Exit Sub

Previous_Error:

    'If there is an error, identify type if possible,
    'inform the user of the problem, and leave subroutine.

    If (Err = 62) Then
        MsgBox "Error:  Attempt has been made to read past the end of file.  Data file has been
ccorrupted."
    ElseIf (Err = 53) Or (Err = 75) Or (Err = 76) Or (Err = 55) Then
        MsgBox "Invalid filename or path, or file access denied"
    ElseIf (Err = 61) Or (Err = 68) Or (Err = 71) Or (Err = 72) Then
        MsgBox "The disk drive is not ready, or a file error has occured"
```

```
    Else
    End If
    Exit Sub

End Sub
```

H.12 Visual Basic Code Module: Rptable1.bas

```
Attribute VB_Name = "Rptable1"
Option Explicit

'This function calculates and displays all of the fields for table one.


Sub paint_table()

    Dim i As Integer, j As Integer, k As Integer

    Dim sum As Long

    'These two constants are the number of radii and sectors displayed.

    Const max_radii_displayed = 4
    Const max_sectors_displayed = 6


    'Set the table caption to reflect the table type.

    If (table_type = 1) Then

        frmTable_1.caption = "Population Table"

    Else

        frmTable_1.caption = _
            "Cumulative (By Direction) Population Table"

    End If


    'Determine the unit of measure.

    '!!!!!

    If (frmProblem_Data.optKilometers.Value) Then
        frmTable_1.lblDistance_Heading.caption = _
           "Radii (Kilometers)"
    Else
        frmTable_1.lblDistance_Heading.caption = "Radii (Miles)"
    End If


    'Verify that coordinates are within array.

    If (x_position < 1) Then
        x_position = 1
    ElseIf (number_of_radii <= max_radii_displayed) Then
        x_position = 1
    ElseIf ((x_position + max_radii_displayed - 1) > _
        number_of_radii) Then
        x_position = number_of_radii - max_radii_displayed + 1
    End If

    If (y_position < 1) Then
        y_position = 1
    ElseIf ((y_position + max_sectors_displayed + 1) > _
        number_of_segments) Then
        y_position = number_of_segments - max_sectors_displayed + 1
    End If
```

```
'Fill the radii label values.

For i = 0 To max_radii_displayed - 1 Step 1

    If ((x_position + i) <= number_of_radii) Then
        frmTable_1.lblDistance_array(i).caption = _
            Format$(radial_distance(x_position + i), "###0.0000")
    Else
        frmTable_1.lblDistance_array(i).caption = ""
    End If

Next i


'Fill the direction label values.

For i = 0 To max_sectors_displayed - 1 Step 1

    frmTable_1.lblSector_array(i).caption = _
        directions(y_position + i)

Next i


'Fill the population sector label values with either the population
'data or the cumulative population data depending on which table
'type we're displaying. Also calculate and fill the sector sum
'labels.

For i = 0 To max_sectors_displayed - 1 Step 1

    sum = 0

    frmTable_1.lblValue_array(i).caption = ""

    For j = 1 To number_of_radii Step 1

        sum = sum + sector_population(y_position + i, j)

        If ((j >= x_position) And _
            (j <= (x_position + max_radii_displayed - 1))) Then

            If (table_type = 1) Then

                frmTable_1.lblValue_array(i).caption = _
                    frmTable_1.lblValue_array(i).caption + _
                    Right$(Format$(sector_population( _
                    y_position + i, j), "         #########0"), 10)

            Else

                frmTable_1.lblValue_array(i).caption = _
                    frmTable_1.lblValue_array(i).caption + _
                    Right$(Format$(sum, "         #########0"), 10)

            End If

        End If

    Next j

    frmTable_1.lblCsum_array(i).caption = Format$(sum, "########0")

Next i
```

```
    'Calculate and fill the radial sum labels.

    For i = 0 To max_radii_displayed - 1 Step 1

        If ((x_position + i) <= number_of_radii) Then

            sum = 0

            If (table_type = 1) Then
                For j = 1 To number_of_segments Step 1
                    sum = sum + sector_population(j, x_position + i)
                Next j
            Else
                For j = 1 To number_of_segments Step 1
                    For k = 1 To (x_position + i) Step 1
                        sum = sum + sector_population(j, k)
                    Next k
                Next j
            End If

            frmTable_1.lblRsum_array(i).caption = _
                Format$(sum, "########0")

        Else

            frmTable_1.lblRsum_array(i).caption = ""

        End If

    Next i


    'Calculate and fill the total sum label.

    sum = 0

    For i = 1 To number_of_segments Step 1

        For j = 1 To number_of_radii Step 1

            sum = sum + sector_population(i, j)

        Next j

    Next i

    frmTable_1.lblTotal.caption = Format$(sum, "########0")


End Sub
```

H.13 Visual Basic Code Module: Run_mppr.bas

```
Attribute VB_Name = "Run_mppr"
'qqq ChDir "mapplan"
'qqq Shell "mppr.exe . map_out.spc"
'qqq ChDir ".."
'qqq RUN "secpop90"
```

H.14 Visual Basic Code Module: Savedata.bas

```
Attribute VB_Name = "Savedata"
'*******************************************************************************
'  Module:  Readfile
'
'  Project: SECPOP
'
'  Description: Module no longer used
'
'  Modification History
'  Date      By   Description
'  --------  ---  ----------------------------------------------
'  20020213  cwm  Made module obsolete
'
'*******************************************************************************

'Option Explicit
'
'Sub save_data()
'    Dim filenum As Integer
'    Dim ForeColor As Integer, BackColor As Integer
'    Dim Flags As Integer, Cancel As Integer
'    Dim Filename As String, Pathname As String
'    Dim DefaultExt As String, DialogTitle As String
'    DefaultExt = "*.DAT"
'    DialogTitle = "Save Data"
'
'    If frmProblem_Data.txtRadial_Distance1.text = "" Then
'        MsgBox "Data is not yet available - Site file has not been loaded"
'        Exit Sub
'    End If
'
'    Call FileSave(Filename, Pathname, DefaultExt, DialogTitle, 7, 1, Flags, Cancel)
'    If Not Cancel Then
'        If Pathname <> "" Then Filename = Pathname + "\" + Filename
'
'        On Local Error GoTo errfix
'        frmSaving.Show
'        Printer.PrintTarget = Filename
'        Call print_MACCS_input_file
'        Printer.PrintTarget = ""
'        Close filenum
'        Unload frmSaving
'    End If
'    Exit Sub
'
'errfix:
'    MsgBox "A FILE ERROR HAS OCCURRED.  PLEASE CHECK ALL FILE NAMES"
'    Exit Sub
'End Sub
```

H.15 Visual Basic Code Module: Sitecode.bas

```
Attribute VB_Name = "Sitecode"
Option Explicit


'This module allows the user to create, edit, and and view site files

Sub new_site()

    'This module initializes the new site form to its default values,
    'and shows the form.

'    frmMain.mnuSave_Sitecom.Enabled = False
'    frmMain.mnuSave_as_Sitecom.Enabled = False
    frmSite_Data.caption = CurDir$ + "\" + "NEW_SITE.SIT"
    frmProblem_Data.txtSite_File_Name.text = "NEW_SITE.SIT"
    frmSite_Data.txtSite_Name.text = ""
    frmSite_Data.txtLongitude_Degrees.text = ""
    frmSite_Data.txtLongitude_Minutes.text = ""
    frmSite_Data.txtLongitude_Seconds.text = ""
    frmSite_Data.txtLatitude_Degrees.text = ""
    frmSite_Data.txtLatitude_Minutes.text = ""
    frmSite_Data.txtLatitude_Seconds.text = ""
    frmSite_Data.txtSite_Remarks.text = ""
    frmSite_Data.Show

End Sub

Sub open_site()

    On Error GoTo DialogError

    With frmMain.dlgSecpop
        .Filter = "Site Files (*.sit)|*.sit|All Files|*.*"
        .FilterIndex = 1
        .DialogTitle = "Select a file to open"
        .Filename = ""
'        .InitDir = InitSiteFilePath
        If InitSiteFilePathChanged = True Then
            .InitDir = frmSetup.txtSite_Path.text
            InitSiteFilePathChanged = False
        Else
            .InitDir = InitSiteFilePath
        End If
        .ShowOpen
    End With

    'reset init dir

    InitSiteFilePath = Left(frmMain.dlgSecpop.Filename, _
                            (InStrRev(frmMain.dlgSecpop.Filename, "\") - 1))


    'This routine opens a previously created site file.

    Dim site_name As String, site_remarks As String, dummy As String
    Dim longitude_degrees As Integer, longitude_minutes As Integer, longitude_seconds As Integer
    Dim latitude_degrees As Integer, latitude_minutes As Integer, latitude_seconds As Integer
    Dim Flags As Integer, Cancel As Integer
    Dim Fname As String
    Static Pathname As String
    Const site_file = 20, file_buffer = 1024
```

```
    'If file is not in current directory then add path to fname.

    If Pathname <> "" Then Fname = Pathname + "\" + Fname

    'Open site file and read in the site data, placing it into temp variables.
    Fname = frmMain.dlgSecpop.Filename
    Open Fname For Input Access Read Lock Write As #site_file Len = file_buffer
    Line Input #site_file, dummy
    Line Input #site_file, site_name
    Line Input #site_file, dummy
    Line Input #site_file, dummy
    Input #site_file, longitude_degrees
    Input #site_file, longitude_minutes
    Input #site_file, longitude_seconds
    Line Input #site_file, dummy
    Input #site_file, latitude_degrees
    Input #site_file, latitude_minutes
    Input #site_file, latitude_seconds
    Line Input #site_file, dummy
    Line Input #site_file, site_remarks
    Close site_file

    'Set window caption to fname.

    frmSite_Data.caption = Fname

    'Move file values from temporary variables to the correct form locations.

    frmProblem_Data.txtSite_File_Name.text = Fname
    frmSite_Data.txtSite_Name.text = site_name
    frmSite_Data.txtLongitude_Degrees.text = Trim(Str$(longitude_degrees))
    frmSite_Data.txtLongitude_Minutes.text = Trim(Str$(longitude_minutes))
    frmSite_Data.txtLongitude_Seconds.text = Trim(Str$(longitude_seconds))
    frmSite_Data.txtLatitude_Degrees.text = Trim(Str$(latitude_degrees))
    frmSite_Data.txtLatitude_Minutes.text = Trim(Str$(latitude_minutes))
    frmSite_Data.txtLatitude_Seconds.text = Trim(Str$(latitude_seconds))
    frmSite_Data.txtSite_Remarks.text = site_remarks

    'If site form is not visible, show it.

    If (frmSite_Data.Visible = False) Then
        frmSite_Data.Show
    End If

    'verify lat and long
     verify_input (2)

DialogError:

    'User pressed the Cancel button

    frmSite_Data.Show
    Exit Sub

End Sub

Sub save_as_site()

    On Error GoTo DialogError

    With frmMain.dlgSecpop
        .Filter = "Site Files (*.sit)|*.sit|All files|*.*"
        .FilterIndex = 1
        .DialogTitle = "Save File As"
        .Filename = ""
        .Flags = cdlOFNOverwritePrompt
```

```
        .DefaultExt = "sit"
'        .InitDir = InitSiteFilePath
        If InitSiteFilePathChanged = True Then
            .InitDir = frmSetup.txtSite_Path.text
            InitSiteFilePathChanged = False
        Else
            .InitDir = InitSiteFilePath
        End If
        .ShowSave
    End With

    'reset init dir

    InitSiteFilePath = Left(frmMain.dlgSecpop.Filename, _
                            (InStrRev(frmMain.dlgSecpop.Filename, "\") - 1))


    'Save a site file using a new name.

    Const site_file = 20, file_buffer = 1024

    Dim site_name As String, site_remarks As String, dummy As String, caption As String
    Dim longitude_degrees As Integer, longitude_minutes As Integer, longitude_seconds As Integer
    Dim latitude_degrees As Integer, latitude_minutes As Integer, latitude_seconds As Integer
    Dim Flags As Integer, Cancel As Integer
    Dim Fname As String, Pathname As String
    Dim temp As Integer

    'Check validity of data entered by user, warn user if there is a problem.

    temp = verify_input(2)

    'Set default file to name of site window.

'    caption = frmSite_Data.caption
    caption = frmMain.dlgSecpop.Filename

    'Remove path from fname.

    Call parse_path_and_file(caption, Pathname, Fname)

    'If file is not in the current directory, then add the path to
    'the fname.

    If Pathname <> "" Then Fname = Pathname + "\" + Fname
    frmSite_Data.caption = caption
    frmProblem_Data.txtSite_File_Name.text = caption
'    frmSite_Data.caption = Fname
'    frmProblem_Data.txtSite_File_Name.text = Fname

    'Call function to save the data.

    Call save_site

DialogError:

    'User pressed the Cancel button
    Exit Sub

End Sub

Sub save_site()

    'Save data from a site form.
```

```
    'Declare as constant the site file handle and the file buffer size.

    Const site_file = 20, file_buffer = 1024
    Dim i As Integer
    Dim Fname As String, Pathname As String

    'Set default fname to name of site window.
    Fname = frmSite_Data.caption

    'Enable local errror checking.

    On Local Error GoTo site_save_error

    'If the site has not yet been named, call save as.

    If Fname = "" Then
        Call save_as_site
        Exit Sub
    End If

    'Open the site file and send the site info to it.

    Open Fname For Output Access Write Lock Read Write As #site_file Len = file_buffer
    Print #site_file, "Site Name:"
    Print #site_file, frmSite_Data.txtSite_Name.text
    Print #site_file, "Site Coordinates:"
    Print #site_file, "Longitude (Degrees, Minutes, Seconds)"
    Print #site_file, "                ";Val(frmSite_Data.txtLongitude_Degrees.text);
    Print #site_file, "          ";Val(frmSite_Data.txtLongitude_Minutes.text);
    Print #site_file, "          ";Val(frmSite_Data.txtLongitude_Seconds.text)
    Print #site_file, "Latitude (Degrees, Minutes, Seconds)"
    Print #site_file, "                ";Val(frmSite_Data.txtLatitude_Degrees.text);
    Print #site_file, "          ";Val(frmSite_Data.txtLatitude_Minutes.text);
    Print #site_file, "          ";Val(frmSite_Data.txtLatitude_Seconds.text)
    Print #site_file, "Site Remarks:"
    Print #site_file, frmSite_Data.txtSite_Remarks.text
    Close site_file

    'Reset window names to match site file name.

    frmSite_Data.caption = Fname
    frmProblem_Data.txtSite_File_Name.text = Fname

    Exit Sub

site_save_error:

    'If an error occurs, inform the user and exit sub.

    MsgBox "Error opening or creating site file"
    Exit Sub

End Sub
```

H.16 Visual Basic Code Module: Types.bas

```
Attribute VB_Name = "Types"
'*******************************************************************************
'  Module:  Types
'
'  Project: SECPOP
'
'  Description: This module was made obsolete by InIt module
'
'  Modification History
'  Date       By   Description
'  --------  ---  ------------------------------------------------------------
'  20020303 cwm  rendered obsolete
'

'*******************************************************************************

'Type economic_data
'
'    region_area As Long
'    region_frmfrc As Single
'    region_dpf As Single
'    region_asfp As Single
'    region_vfrm As Single
'    region_vnfrm As Single
'
'End Type
```

H.17 Visual Basic Code Module: Utility.bas

```
Attribute VB_Name = "Utility"
Option Explicit

'Various general purpose utilities:
'1. Print stuff to a non-Form screen.
'2. Print stuff to the printer.
'3. Verify the user's input in one of several forms.
'4. Copy from one file to another.

Sub parse_path_and_file(caption As String, path_name As String, file_name As String)

    'This routine removes the path from a file name.  This is necessary when
    'passing a default filename to a file dialog box.

    Dim temp As String
    Dim modified_location As Integer
    Dim slash_found As Integer, slash_location As Integer

    'Remove the modified specifier from the file name if necessary.

    modified_location = InStr(caption, " (Modified)")
    If modified_location <> 0 Then
        temp = Mid$(caption, 1, modified_location - 1)
    Else
        temp = caption
    End If

    'Search backward from the end of the string to the first slash,
    'and record the position of the first slash.

    slash_found = False
    slash_location = Len(temp)
    While ((Not slash_found) And (slash_location > 0))
        If Mid$(temp, slash_location, 1) = "\" Then
            slash_found = True
        Else
            slash_location = slash_location - 1
        End If
    Wend

    'If a slash was found, break the name into two parts - path and file.

    If slash_found = True Then
        path_name = Mid$(temp, 1, slash_location - 1)
        file_name = Mid$(temp, slash_location + 1)

    'Otherwise, return the unchanged filename.

    Else
        path_name = ""
        file_name = temp
    End If

End Sub

Sub print_text(text As String, text_width As Integer, left_margin As Integer)

    'This routine prints formatted text strings in a non-form environment.

    Dim next_line As String, character As String
    Dim character_location As Integer, character_found As Integer

    'Execute loop while the lenght of text is greater
```

```
    'than the user defined size.

    While (Len(text) > text_width)

        'Put next string of length user has requested into next line.

        next_line = Left$(text, text_width)
        character_found = False
        character_location = text_width

        'Scan input string for special characters ( <cr>, -, and " ").

        While ((Not character_found) And (character_location > 0))
            character = Mid$(next_line, character_location, 1)
            If ((character = " ") Or (character = "-") Or (character = Chr$(13))) Then
                character_found = True
            Else
                character_location = character_location - 1
            End If
        Wend

        'Process special characters found in string.

        If character_found Then
            Printer.Print Tab(left_margin); Left$(next_line, character_location - 1);
            If character = "-" Then
                Printer.Print character
            Else
                Printer.Print
            End If
            character_found = False
            text = Mid$(text, character_location + 1)

        Else

            'If no characters found, then print string.

            Printer.Print Tab(left_margin); next_line
            text = Mid$(text, text_width + 1)

        End If

    Wend

    Printer.Print Tab(left_margin); text

End Sub

Function verify_input(which_form As Integer) As Integer

    Dim filenum As Integer
    Dim TempString As String, i As Integer, j As Integer

    'Define constants representing which form is calling for input
    'verification.

    Const PROBLEM = 1, SITE = 2, REGIONAL_CALC = 3

    'Decide which form is calling, analyze the input on that form,
    'display a message if necessary, and return a value to the calling
    'function.

    verify_input = 0

    Select Case which_form
```

```
        Case PROBLEM

            'Enable local error checking for the new problem form to deal
            'with disk related errors check for the file then reset local
            'error checking.

            On Local Error GoTo problem_bad_file_name
            filenum = FreeFile
            Open frmProblem_Data.txtSite_File_Name.text For Input As #filenum
            Close filenum
            On Local Error GoTo 0

            If Val(frmProblem_Data.txtPopulation_multiplier.text) <= 0 Then

                MsgBox "Error: Population multiplier must greater than 0.  "
                verify_input = -1

            ElseIf Val(frmProblem_Data.txtPopulation_multiplier.text) > 9999.9999 Then

                MsgBox "Error: Population multiplier must be less than 10000"
                verify_input = -1

            ElseIf (radial_distance(1) <= 0) Then

                MsgBox "Error: Radial distance 1 must be greater than 0."
                 verify_input = -2

            ElseIf (radial_distance(2) <= 0) Then

                MsgBox "Error: Radial distance 2 must be greater than 0."
                verify_input = -3

            ElseIf (Val(frmProblem_Data.lblNumber_of_regions.caption) < 2 Or
Val(frmProblem_Data.lblNumber_of_regions.caption) > 99) Then

                MsgBox "Invalid Economic region settings"
                verify_input = -4

            Else

                'Everything's O.K.
                verify_input = 1

            End If

        Case SITE

            If (Val(frmSite_Data.txtLatitude_Degrees.text) > 47) Or _
               (Val(frmSite_Data.txtLatitude_Degrees.text) < 24) Or _
               (Val(frmSite_Data.txtLongitude_Degrees.text) > 125) Or _
               (Val(frmSite_Data.txtLongitude_Degrees.text) < 67) Then   ' longitude degs
origionally 172

                MsgBox "Warning: The Site that you have specified is outside of the " & _
                       "continental United States.  " & vbCrLf & _
                       "Data near the site may not be available." & vbCrLf & vbCrLf & _
                       "Values within the continental United States are as follows:  " & _
                        vbCrLf & vbCrLf & _
                       "Latitude:      from 24 to 47" & vbCrLf & _
                       "Longitude:   from 67 to 125"
                verify_input = -1

            Else

                'Everything's O.K.
                verify_input = 1
```

```
            End If

'''         Case REGIONAL_CALC
'''
'''             If ((Val(frmMake_a_Circle.txtRadii.text) < 1) Or _
'''                 (Val(frmMake_a_Circle.txtRadii.text) > 170)) Then
'''
'''                 MsgBox _
'''                     "Radii of Circles value must be greater than or" + _
'''                     Chr$(13) + Chr$(10) + _
'''                     "equal to 1 mile and less than or equal to 170" + _
'''                     Chr$(13) + Chr$(10) + _
'''                     "miles.", vbOK, "Error"
'''
'''                 verify_input = 0
'''
'''             ElseIf ((Val(frmMake_a_Circle.txtSpacing.text) < 1) Or _
'''                 (Val(frmMake_a_Circle.txtSpacing.text) > 340)) Then
'''
'''                 MsgBox _
'''                     "Longitudinal Spacing of Circles value must" + _
'''                     Chr$(13) + Chr$(10) + _
'''                     "be greater than or equal to 1 mile and less" + _
'''                     Chr$(13) + Chr$(10) + _
'''                     "than or equal to 340 miles.", vbOK, "Error"
'''
'''                 verify_input = -1
'''
'''             ElseIf ((Val(frmMake_a_Circle.txtThreshold.text) < 1) Or _
'''                 (Val(frmMake_a_Circle.txtThreshold.text) > 20000)) Then
'''
'''                 MsgBox _
'''                     "Population Density Threshold value must be" + _
'''                     Chr$(13) + Chr$(10) + _
'''                     "greater than or equal to 1 person per square" + _
'''                     Chr$(13) + Chr$(10) + _
'''                     "mile and less than or equal to 20000 persons" + _
'''                     Chr$(13) + Chr$(10) + _
'''                     "per square mile.", vbOK, "Error"
'''
'''                 verify_input = -2
'''
'''             ElseIf (frmMake_a_Circle.txtInput_Map.text = "") Then
'''
'''                 MsgBox _
'''                     "Input Map Name missing. Please enter a valid" + _
'''                     Chr$(13) + Chr$(10) + _
'''                     "filename.", vbOK, "Error"
'''
'''                 verify_input = -3
'''
'''             Else
'''
'''                 'Enable local error checking for regional calc form
'''                 On Local Error GoTo regional_file_error
'''                 filenum = FreeFile
'''                 Open frmMake_a_Circle.txtInput_Map.text _
'''                    For Input As filenum
'''                 Close filenum
'''                 On Local Error GoTo 0
'''                 'If we got here then everything's O.K.
'''                 verify_input = 1
'''
'''             End If
```

```
    End Select

    Exit Function

problem_bad_file_name:

    MsgBox "Error: Unable to open site file - check file name. "
    verify_input = -1
'    verify_input = 0
    Exit Function

regional_file_error:

    MsgBox _
        "Input Map Name invalid. Please enter a valid" + _
        Chr$(13) + Chr$(10) + _
        "filename.", vbOK, "Error"

    verify_input = -3

    Exit Function

End Function


Sub copy_file(input_filename As String, output_filename As String)

    Dim i As Long
    Dim my_byte As String * 1
    Dim input_filenumber As Integer, output_filenumber As Integer

    On Local Error GoTo error_return

    input_filenumber = FreeFile
    Open input_filename For Binary Access Read As #input_filenumber

    output_filenumber = FreeFile

    'Truncate output file to 0 length if it already existed.
    Open output_filename For Output Access Write As #output_filenumber
    Close (output_filenumber)

    Open output_filename For Binary Access Write As #output_filenumber

    For i = 1 To LOF(input_filenumber) Step 1
        Get #input_filenumber, , my_byte
        Put #output_filenumber, , my_byte
    Next i

    Close (input_filenumber)
    Close (output_filenumber)

normal_return:

    Exit Sub

error_return:

    MsgBox Error$(Err)
    Exit Sub

End Sub
```

H.18 Visual Basic Code Module: Utility_Rosette.bas

```
Attribute VB_Name = "Utility_Rosette"
Option Explicit

' This module was extracted from the Microsoft web site.
' It includes:
'    - CaptureScreen - captures the entire screen.
'    - CaptureForm   - captures the form including title bar and border.
'    - CaptureClient - captures all visible controls on the active form.
'    - CaptureActiveWindow - captures the currently active window
'    - PrintPictureToFitPage - adjusts size of picture to fit paper
'    - and all other functions and api calls to support the above.
'
' Only CatureClient is used in SecPop.  The others are included
' for completeness and may be used later.



'--------------------------------------------------------------------
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'
' Visual Basic 4.0 16/32 Capture Routines
'
' This module contains several routines for capturing windows into a
' picture.  All the routines work on both 16 and 32 bit Windows
' platforms.
' The routines also have palette support.
'
' CreateBitmapPicture - Creates a picture object from a bitmap and
' palette.
' CaptureWindow - Captures any window given a window handle.
' CaptureActiveWindow - Captures the active window on the desktop.
' CaptureForm - Captures the entire form.
' CaptureClient - Captures the client area of a form.
' CaptureScreen - Captures the entire screen.
' PrintPictureToFitPage - prints any picture as big as possible on
' the page.
'
' NOTES
'    - No error trapping is included in these routines.
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'

Option Base 0

Private Type PALETTEENTRY
   peRed As Byte
   peGreen As Byte
   peBlue As Byte
   peFlags As Byte
End Type

Private Type LOGPALETTE
   palVersion As Integer
   palNumEntries As Integer
   palPalEntry(255) As PALETTEENTRY  ' Enough for 256 colors.
End Type

Private Type GUID
   Data1 As Long
   Data2 As Integer
   Data3 As Integer
   Data4(7) As Byte
End Type
```

```
#If Win32 Then

   Private Const RASTERCAPS As Long = 38
   Private Const RC_PALETTE As Long = &H100
   Private Const SIZEPALETTE As Long = 104

   Private Type RECT
      Left As Long
      Top As Long
      Right As Long
      Bottom As Long
   End Type

   Private Declare Function CreateCompatibleDC Lib "GDI32" ( _
      ByVal hDC As Long) As Long
   Private Declare Function CreateCompatibleBitmap Lib "GDI32" ( _
      ByVal hDC As Long, ByVal nWidth As Long, _
      ByVal nHeight As Long) As Long
   Private Declare Function GetDeviceCaps Lib "GDI32" ( _
      ByVal hDC As Long, ByVal iCapabilitiy As Long) As Long
   Private Declare Function GetSystemPaletteEntries Lib "GDI32" ( _
      ByVal hDC As Long, ByVal wStartIndex As Long, _
      ByVal wNumEntries As Long, lpPaletteEntries As PALETTEENTRY) _
      As Long
   Private Declare Function CreatePalette Lib "GDI32" ( _
      lpLogPalette As LOGPALETTE) As Long
   Private Declare Function SelectObject Lib "GDI32" ( _
      ByVal hDC As Long, ByVal hObject As Long) As Long
   Private Declare Function BitBlt Lib "GDI32" ( _
      ByVal hDCDest As Long, ByVal XDest As Long, _
      ByVal YDest As Long, ByVal nWidth As Long, _
      ByVal nHeight As Long, ByVal hDCSrc As Long, _
      ByVal XSrc As Long, ByVal YSrc As Long, ByVal dwRop As Long) _
      As Long
   Private Declare Function DeleteDC Lib "GDI32" ( _
      ByVal hDC As Long) As Long
   Private Declare Function GetForegroundWindow Lib "USER32" () _
      As Long
   Private Declare Function SelectPalette Lib "GDI32" ( _
      ByVal hDC As Long, ByVal hPalette As Long, _
      ByVal bForceBackground As Long) As Long
   Private Declare Function RealizePalette Lib "GDI32" ( _
      ByVal hDC As Long) As Long
   Private Declare Function GetWindowDC Lib "USER32" ( _
      ByVal hWnd As Long) As Long
   Private Declare Function GetDC Lib "USER32" ( _
      ByVal hWnd As Long) As Long
   Private Declare Function GetWindowRect Lib "USER32" ( _
      ByVal hWnd As Long, lpRect As RECT) As Long
   Private Declare Function ReleaseDC Lib "USER32" ( _
      ByVal hWnd As Long, ByVal hDC As Long) As Long
   Private Declare Function GetDesktopWindow Lib "USER32" () As Long

   Private Type PicBmp
      Size As Long
      Type As Long
      hBmp As Long
      hPal As Long
      Reserved As Long
   End Type

   Private Declare Function OleCreatePictureIndirect _
      Lib "olepro32.dll" (PicDesc As PicBmp, RefIID As GUID, _
      ByVal fPictureOwnsHandle As Long, IPic As IPicture) As Long
```

```
#ElseIf Win16 Then

   Private Const RASTERCAPS As Integer = 38
   Private Const RC_PALETTE As Integer = &H100
   Private Const SIZEPALETTE As Integer = 104

   Private Type RECT
      Left As Integer
      Top As Integer
      Right As Integer
      Bottom As Integer
   End Type

   Private Declare Function CreateCompatibleDC Lib "GDI" ( _
      ByVal hDC As Integer) As Integer
   Private Declare Function CreateCompatibleBitmap Lib "GDI" ( _
      ByVal hDC As Integer, ByVal nWidth As Integer, _
      ByVal nHeight As Integer) As Integer
   Private Declare Function GetDeviceCaps Lib "GDI" ( _
      ByVal hDC As Integer, ByVal iCapabilitiy As Integer) As Integer
   Private Declare Function GetSystemPaletteEntries Lib "GDI" ( _
      ByVal hDC As Integer, ByVal wStartIndex As Integer, _
      ByVal wNumEntries As Integer, _
      lpPaletteEntries As PALETTEENTRY) As Integer
   Private Declare Function CreatePalette Lib "GDI" ( _
      lpLogPalette As LOGPALETTE) As Integer
   Private Declare Function SelectObject Lib "GDI" ( _
      ByVal hDC As Integer, ByVal hObject As Integer) As Integer
   Private Declare Function BitBlt Lib "GDI" ( _
      ByVal hDCDest As Integer, ByVal XDest As Integer, _
      ByVal YDest As Integer, ByVal nWidth As Integer, _
      ByVal nHeight As Integer, ByVal hDCSrc As Integer, _
      ByVal XSrc As Integer, ByVal YSrc As Integer, _
      ByVal dwRop As Long) As Integer
   Private Declare Function DeleteDC Lib "GDI" ( _
      ByVal hDC As Integer) As Integer
   Private Declare Function GetForegroundWindow Lib "USER" _
      Alias "GetActiveWindow" () As Integer
   Private Declare Function SelectPalette Lib "USER" ( _
      ByVal hDC As Integer, ByVal hPalette As Integer, ByVal _
      bForceBackground As Integer) As Integer
   Private Declare Function RealizePalette Lib "USER" ( _
      ByVal hDC As Integer) As Integer
   Private Declare Function GetWindowDC Lib "USER" ( _
      ByVal hWnd As Integer) As Integer
   Private Declare Function GetDC Lib "USER" ( _
      ByVal hWnd As Integer) As Integer
   Private Declare Function GetWindowRect Lib "USER" ( _
      ByVal hWnd As Integer, lpRect As RECT) As Integer
   Private Declare Function ReleaseDC Lib "USER" ( _
      ByVal hWnd As Integer, ByVal hDC As Integer) As Integer
   Private Declare Function GetDesktopWindow Lib "USER" () As Integer

   Private Type PicBmp
      Size As Integer
      Type As Integer
      hBmp As Integer
      hPal As Integer
      Reserved As Integer
   End Type

   Private Declare Function OleCreatePictureIndirect _
      Lib "oc25.dll" (PictDesc As PicBmp, RefIID As GUID, _
      ByVal fPictureOwnsHandle As Integer, IPic As IPicture) _
      As Integer
```

```
#End If

'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'
' CreateBitmapPicture
'    - Creates a bitmap type Picture object from a bitmap and
'      palette.
'
' hBmp
'    - Handle to a bitmap.
'
' hPal
'    - Handle to a Palette.
'    - Can be null if the bitmap doesn't use a palette.
'
' Returns
'    - Returns a Picture object containing the bitmap.
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'
#If Win32 Then
   Public Function CreateBitmapPicture(ByVal hBmp As Long, _
      ByVal hPal As Long) As Picture

      Dim r As Long
#ElseIf Win16 Then
   Public Function CreateBitmapPicture(ByVal hBmp As Integer, _
      ByVal hPal As Integer) As Picture

      Dim r As Integer
#End If
   Dim Pic As PicBmp
   ' IPicture requires a reference to "Standard OLE Types."
   Dim IPic As IPicture
   Dim IID_IDispatch As GUID

   ' Fill in with IDispatch Interface ID.
   With IID_IDispatch
      .Data1 = &H20400
      .Data4(0) = &HC0
      .Data4(7) = &H46
   End With

   ' Fill Pic with necessary parts.
   With Pic
      .Size = Len(Pic)          ' Length of structure.
      .Type = vbPicTypeBitmap   ' Type of Picture (bitmap).
      .hBmp = hBmp              ' Handle to bitmap.
      .hPal = hPal              ' Handle to palette (may be null).
   End With

   ' Create Picture object.
   r = OleCreatePictureIndirect(Pic, IID_IDispatch, 1, IPic)

   ' Return the new Picture object.
   Set CreateBitmapPicture = IPic
End Function

'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'
' CaptureWindow
'    - Captures any portion of a window.
'
' hWndSrc
'    - Handle to the window to be captured.
'
' Client
```

```
'     - If True CaptureWindow captures from the client area of the
'       window.
'     - If False CaptureWindow captures from the entire window.
'
' LeftSrc, TopSrc, WidthSrc, HeightSrc
'     - Specify the portion of the window to capture.
'     - Dimensions need to be specified in pixels.
'
' Returns
'     - Returns a Picture object containing a bitmap of the specified
'       portion of the window that was captured.
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
''''''
'
#If Win32 Then
   Public Function CaptureWindow(ByVal hWndSrc As Long, _
      ByVal Client As Boolean, ByVal LeftSrc As Long, _
      ByVal TopSrc As Long, ByVal WidthSrc As Long, _
      ByVal HeightSrc As Long) As Picture

      Dim hDCMemory As Long
      Dim hBmp As Long
      Dim hBmpPrev As Long
      Dim r As Long
      Dim hDCSrc As Long
      Dim hPal As Long
      Dim hPalPrev As Long
      Dim RasterCapsScrn As Long
      Dim HasPaletteScrn As Long
      Dim PaletteSizeScrn As Long
#ElseIf Win16 Then
   Public Function CaptureWindow(ByVal hWndSrc As Integer, _
      ByVal Client As Boolean, ByVal LeftSrc As Integer, _
      ByVal TopSrc As Integer, ByVal WidthSrc As Long, _
      ByVal HeightSrc As Long) As Picture

      Dim hDCMemory As Integer
      Dim hBmp As Integer
      Dim hBmpPrev As Integer
      Dim r As Integer
      Dim hDCSrc As Integer
      Dim hPal As Integer
      Dim hPalPrev As Integer
      Dim RasterCapsScrn As Integer
      Dim HasPaletteScrn As Integer
      Dim PaletteSizeScrn As Integer
#End If
   Dim LogPal As LOGPALETTE

   ' Depending on the value of Client get the proper device context.
   If Client Then
      hDCSrc = GetDC(hWndSrc) ' Get device context for client area.
   Else
      hDCSrc = GetWindowDC(hWndSrc) ' Get device context for entire
                                    ' window.
   End If

   ' Create a memory device context for the copy process.
   hDCMemory = CreateCompatibleDC(hDCSrc)
   ' Create a bitmap and place it in the memory DC.
   hBmp = CreateCompatibleBitmap(hDCSrc, WidthSrc, HeightSrc)
   hBmpPrev = SelectObject(hDCMemory, hBmp)

   ' Get screen properties.
   RasterCapsScrn = GetDeviceCaps(hDCSrc, RASTERCAPS) ' Raster
                                                      ' capabilities.
```

```
    HasPaletteScrn = RasterCapsScrn And RC_PALETTE        ' Palette
                                                          ' support.
    PaletteSizeScrn = GetDeviceCaps(hDCSrc, SIZEPALETTE) ' Size of
                                                          ' palette.

    ' If the screen has a palette make a copy and realize it.
    If HasPaletteScrn And (PaletteSizeScrn = 256) Then
       ' Create a copy of the system palette.
       LogPal.palVersion = &H300
       LogPal.palNumEntries = 256
       r = GetSystemPaletteEntries(hDCSrc, 0, 256, _
           LogPal.palPalEntry(0))
       hPal = CreatePalette(LogPal)
       ' Select the new palette into the memory DC and realize it.
       hPalPrev = SelectPalette(hDCMemory, hPal, 0)
       r = RealizePalette(hDCMemory)
    End If

    ' Copy the on-screen image into the memory DC.
    r = BitBlt(hDCMemory, 0, 0, WidthSrc, HeightSrc, hDCSrc, _
       LeftSrc, TopSrc, vbSrcCopy)

' Remove the new copy of the  on-screen image.
    hBmp = SelectObject(hDCMemory, hBmpPrev)

    ' If the screen has a palette get back the palette that was
    ' selected in previously.
    If HasPaletteScrn And (PaletteSizeScrn = 256) Then
       hPal = SelectPalette(hDCMemory, hPalPrev, 0)
    End If

    ' Release the device context resources back to the system.
    r = DeleteDC(hDCMemory)
    r = ReleaseDC(hWndSrc, hDCSrc)

    ' Call CreateBitmapPicture to create a picture object from the
    ' bitmap and palette handles. Then return the resulting picture
    ' object.
    Set CaptureWindow = CreateBitmapPicture(hBmp, hPal)
End Function

'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'
' CaptureScreen
'     - Captures the entire screen.
'
' Returns
'     - Returns a Picture object containing a bitmap of the screen.
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'
Public Function CaptureScreen() As Picture
    #If Win32 Then
       Dim hWndScreen As Long
    #ElseIf Win16 Then
       Dim hWndScreen As Integer
    #End If

    ' Get a handle to the desktop window.
    hWndScreen = GetDesktopWindow()

    ' Call CaptureWindow to capture the entire desktop give the handle
    ' and return the resulting Picture object.

    Set CaptureScreen = CaptureWindow(hWndScreen, False, 0, 0, _
       Screen.Width \ Screen.TwipsPerPixelX, _
       Screen.Height \ Screen.TwipsPerPixelY)
```

```
End Function

'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'
' CaptureForm
'     - Captures an entire form including title bar and border.
'
' frmSrc
'     - The Form object to capture.
'
' Returns
'     - Returns a Picture object containing a bitmap of the entire
'       form.
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'
Public Function CaptureForm(frmSrc As Form) As Picture
    ' Call CaptureWindow to capture the entire form given its window
    ' handle and then return the resulting Picture object.
    Set CaptureForm = CaptureWindow(frmSrc.hWnd, False, 0, 0, _
       frmSrc.ScaleX(frmSrc.Width, vbTwips, vbPixels), _
       frmSrc.ScaleY(frmSrc.Height, vbTwips, vbPixels))
End Function

'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'
' CaptureClient
'     - Captures the client area of a form.
'
' frmSrc
'     - The Form object to capture.
'
' Returns
'     - Returns a Picture object containing a bitmap of the form's
'       client area.
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'
Public Function CaptureClient(frmSrc As Form) As Picture
    ' Call CaptureWindow to capture the client area of the form given
    ' its window handle and return the resulting Picture object.
    Set CaptureClient = CaptureWindow(frmSrc.hWnd, True, 0, 0, _
       frmSrc.ScaleX(frmSrc.ScaleWidth, frmSrc.ScaleMode, vbPixels), _
       frmSrc.ScaleY(frmSrc.ScaleHeight, frmSrc.ScaleMode, vbPixels))
End Function

'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'
' CaptureActiveWindow
'     - Captures the currently active window on the screen.
'
' Returns
'     - Returns a Picture object containing a bitmap of the active
'       window.
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'
Public Function CaptureActiveWindow() As Picture
    #If Win32 Then
        Dim hWndActive As Long
        Dim r As Long
    #ElseIf Win16 Then
        Dim hWndActive As Integer
        Dim r As Integer
    #End If
    Dim RectActive As RECT

    ' Get a handle to the active/foreground window.
    hWndActive = GetForegroundWindow()
```

```
    ' Get the dimensions of the window.
    r = GetWindowRect(hWndActive, RectActive)

    ' Call CaptureWindow to capture the active window given its
    ' handle and return the Resulting Picture object.
 Set CaptureActiveWindow = CaptureWindow(hWndActive, False, 0, 0, _
       RectActive.Right - RectActive.Left, _
       RectActive.Bottom - RectActive.Top)
 End Function

'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'
' PrintPictureToFitPage
'    - Prints a Picture object as big as possible.
'
' Prn
'    - Destination Printer object.
'
' Pic
'    - Source Picture object.
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'
Public Sub PrintPictureToFitPage(Prn As Printer, Pic As Picture)
    Const vbHiMetric As Integer = 8
    Dim PicRatio As Double
    Dim PrnWidth As Double
    Dim PrnHeight As Double
    Dim PrnRatio As Double
    Dim PrnPicWidth As Double
    Dim PrnPicHeight As Double

    ' Determine if picture should be printed in landscape or portrait
    ' and set the orientation.
    If Pic.Height >= Pic.Width Then
       Prn.Orientation = vbPRORPortrait    ' Taller than wide.
    Else
       Prn.Orientation = vbPRORLandscape   ' Wider than tall.
    End If

    ' Calculate device independent Width-to-Height ratio for picture.
    PicRatio = Pic.Width / Pic.Height

    ' Calculate the dimentions of the printable area in HiMetric.
    PrnWidth = Prn.ScaleX(Prn.ScaleWidth, Prn.ScaleMode, vbHiMetric)
    PrnHeight = Prn.ScaleY(Prn.ScaleHeight, Prn.ScaleMode, vbHiMetric)
    ' Calculate device independent Width to Height ratio for printer.
    PrnRatio = PrnWidth / PrnHeight

    ' Scale the output to the printable area.
    If PicRatio >= PrnRatio Then
       ' Scale picture to fit full width of printable area.
       PrnPicWidth = Prn.ScaleX(PrnWidth, vbHiMetric, Prn.ScaleMode)
       PrnPicHeight = Prn.ScaleY(PrnWidth / PicRatio, vbHiMetric, _
          Prn.ScaleMode)
    Else
       ' Scale picture to fit full height of printable area.
       PrnPicHeight = Prn.ScaleY(PrnHeight, vbHiMetric, Prn.ScaleMode)
       PrnPicWidth = Prn.ScaleX(PrnHeight * PicRatio, vbHiMetric, _
          Prn.ScaleMode)
    End If

    ' Print the picture using the PaintPicture method.
    Prn.PaintPicture Pic, 0, 0, PrnPicWidth, PrnPicHeight
End Sub
'---------------------------------------------------------------------
```

H.19 Visual Basic Form Module: About.frm

```
VERSION 5.00
Begin VB.Form frmAbout
   Caption         =   "SecPop2000"
   ClientHeight    =   6450
   ClientLeft      =   60
   ClientTop       =   345
   ClientWidth     =   8490
   Icon            =   "About.frx":0000
   LinkTopic       =   "Form1"
   MDIChild        =   -1  'True
   ScaleHeight     =   6450
   ScaleWidth      =   8490
   Begin VB.Label Label11
      Caption         =   "SECPOP2000 - SECtor POPulation and Economic Estimator"
      BeginProperty Font
         Name            =   "MS Sans Serif"
         Size            =   12
         Charset         =   0
         Weight          =   700
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      Height          =   705
      Left            =   900
      TabIndex        =   8
      Top             =   930
      Width           =   5235
   End
   Begin VB.Label Label10
      Caption         =   "JNR@NRC.gov"
      BeginProperty Font
         Name            =   "MS Sans Serif"
         Size            =   9.75
         Charset         =   0
         Weight          =   700
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      Height          =   235
      Left            =   1050
      TabIndex        =   7
      Top             =   4635
      Width           =   2370
   End
   Begin VB.Label Label7
      Caption         =   "Nuclear Regulatory Commission"
      BeginProperty Font
         Name            =   "MS Sans Serif"
         Size            =   9.75
         Charset         =   0
         Weight          =   700
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      Height          =   235
      Left            =   1050
      TabIndex        =   6
      Top             =   4020
      Width           =   3525
   End
```

```
Begin VB.Label Label6
   Caption         =   "John N. Ridgely"
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   9.75
      Charset         =   0
      Weight          =   700
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   Height          =   235
   Left            =   1050
   TabIndex        =   5
   Top             =   3735
   Width           =   2370
End
Begin VB.Label Label5
   Caption         =   "(301)415-6555"
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   9.75
      Charset         =   0
      Weight          =   700
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   Height          =   235
   Left            =   1050
   TabIndex        =   4
   Top             =   4305
   Width           =   2370
End
Begin VB.Label Label4
   Caption         =   "Version:  3.11"
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   12
      Charset         =   0
      Weight          =   700
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   Height          =   375
   Left            =   900
   TabIndex        =   3
   Top             =   1995
   Width           =   3930
End
Begin VB.Label Label2
   Caption         =   "Developed for the NRC by Sandia National Laboratories, June 2002"
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   9.75
      Charset         =   0
      Weight          =   700
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   Height          =   495
   Left            =   900
   TabIndex        =   2
   Top             =   2670
```

```
      Width           =   6195
   End
   Begin VB.Label Label1
      Caption         =   "Contact: "
      BeginProperty Font
         Name            =   "MS Sans Serif"
         Size            =   9.75
         Charset         =   0
         Weight          =   700
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      Height          =   225
      Left            =   900
      TabIndex        =   1
      Top             =   3420
      Width           =   2370
   End
   Begin VB.Label mnuTitle
      Caption         =   "Secpop2000"
      BeginProperty Font
         Name            =   "MS Sans Serif"
         Size            =   24
         Charset         =   0
         Weight          =   700
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      Height          =   600
      Left            =   900
      TabIndex        =   0
      Top             =   225
      Width           =   2850
   End
End
Attribute VB_Name = "frmAbout"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit
```

H.20 Visual Basic Form Module: Circle.frm

```
VERSION 5.00
Begin VB.Form frmMake_a_Circle
   Appearance      =   0  'Flat
   BackColor       =   &H80000004&
   BorderStyle     =   1  'Fixed Single
   Caption         =   "Regional Population Densities Above The Threshold"
   ClientHeight    =   4680
   ClientLeft      =   1005
   ClientTop       =   570
   ClientWidth     =   6300
   ControlBox      =   0   'False
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   8.25
      Charset         =   0
      Weight          =   700
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H00000000&
   MaxButton       =   0   'False
   MinButton       =   0   'False
   PaletteMode     =   1  'UseZOrder
   ScaleHeight     =   4680
   ScaleWidth      =   6300
   StartUpPosition =   1  'CenterOwner
   Begin VB.TextBox txtSpacing
      Appearance      =   0  'Flat
      BackColor       =   &H80000004&
      ForeColor       =   &H00000000&
      Height          =   720
      Left            =   2808
      TabIndex        =   3
      Top             =   720
      Width           =   2268
   End
   Begin VB.TextBox txtThreshold
      Appearance      =   0  'Flat
      BackColor       =   &H80000004&
      ForeColor       =   &H00000000&
      Height          =   720
      Left            =   216
      TabIndex        =   5
      Top             =   2400
      Width           =   2268
   End
   Begin VB.TextBox txtInput_Map
      Appearance      =   0  'Flat
      BackColor       =   &H80000004&
      ForeColor       =   &H00000000&
      Height          =   720
      Left            =   2808
      TabIndex        =   7
      Top             =   2400
      Width           =   2268
   End
   Begin VB.CommandButton cmdInput_files
      Appearance      =   0  'Flat
      BackColor       =   &H00808000&
      Caption         =   "&Files"
      Height          =   720
      Left            =   5184
```

```
    TabIndex        =   8
    Top             =   2400
    Width           =   756
End
Begin VB.OptionButton optPoints
    Appearance      =   0  'Flat
    BackColor       =   &H80000004&
    Caption         =   "&Points"
    ForeColor       =   &H00000000&
    Height          =   240
    Left            =   3888
    TabIndex        =   10
    TabStop         =   0   'False
    Top             =   3360
    Width           =   1080
End
Begin VB.OptionButton optCircles
    Appearance      =   0  'Flat
    BackColor       =   &H80000004&
    Caption         =   "&Circles"
    ForeColor       =   &H00000000&
    Height          =   240
    Left            =   2484
    TabIndex        =   9
    Top             =   3360
    Value           =   -1  'True
    Width           =   1296
End
Begin VB.CommandButton cmdCalculate_Regional
    Appearance      =   0  'Flat
    BackColor       =   &H00808000&
    Caption         =   "C&alculate"
    Height          =   720
    Left            =   1080
    TabIndex        =   11
    Top             =   3840
    Width           =   1836
End
Begin VB.CommandButton cmdClose
    Appearance      =   0  'Flat
    BackColor       =   &H00808000&
    Cancel          =   -1  'True
    Caption         =   "C&lose"
    Height          =   720
    Left            =   3240
    TabIndex        =   12
    Top             =   3840
    Width           =   1836
End
Begin VB.TextBox txtRadii
    Appearance      =   0  'Flat
    BackColor       =   &H80000004&
    ForeColor       =   &H00000000&
    Height          =   720
    Left            =   216
    TabIndex        =   1
    Top             =   720
    Width           =   2268
End
Begin VB.Label Label7
    Appearance      =   0  'Flat
    AutoSize        =   -1  'True
    BackColor       =   &H80000004&
    Caption         =   "Display results as:"
    ForeColor       =   &H00000000&
    Height          =   240
```

```
      Left            =   216
      TabIndex        =   13
      Top             =   3360
      Width           =   2052
   End
   Begin VB.Label Label2
      Appearance      =   0  'Flat
      BackColor       =   &H80000004&
      Caption         =   "&Input Map Name:"
      ForeColor       =   &H00000000&
      Height          =   240
      Left            =   3132
      TabIndex        =   6
      Top             =   2160
      Width           =   1512
   End
   Begin VB.Label Label1
      Appearance      =   0  'Flat
      BackColor       =   &H80000004&
      Caption         =   "&Radii of Circles (mi):"
      ForeColor       =   &H00000000&
      Height          =   240
      Left            =   216
      TabIndex        =   0
      Top             =   480
      Width           =   2268
   End
   Begin VB.Label Label3
      Alignment       =   2  'Center
      Appearance      =   0  'Flat
      BackColor       =   &H80000004&
      Caption         =   "Longitudinal &Spacing of Circles (mi):"
      ForeColor       =   &H00000000&
      Height          =   480
      Left            =   2808
      TabIndex        =   2
      Top             =   240
      Width           =   2160
   End
   Begin VB.Label Label4
      Alignment       =   2  'Center
      Appearance      =   0  'Flat
      BackColor       =   &H80000004&
      Caption         =   "Population &Density Threshold (People/Square mi):"
      ForeColor       =   &H00000000&
      Height          =   720
      Left            =   324
      TabIndex        =   4
      Top             =   1680
      Width           =   1944
   End
End
Attribute VB_Name = "frmMake_a_Circle"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
'''Option Explicit
'''
''''$INCLUDE: 'secpop90.bi'
'''
''''This form deals with much of the overhead associated with regional
''''calculations, including the user interface parsing the data into a
''''usable format, and reporting the status of the calculation.
'''
'''Private Sub cmdCalculate_Regional_Click()
```

```
'''
'''     'Declare a few local variables. Most of these variables are required
'''     'to put the data in a format acceptable to mapplan.
'''
'''     Dim TempString As String, tempchar As String
'''     Dim orig_input As String
'''     Dim number_of_circle_segments As Integer, i As Integer
'''     Dim x_int As Integer, y_int As Integer, spacing As Double
'''     Dim rotation_increment As Double, radii As Double, X As Double
'''     Dim y_radius As Double, y_space As Double
'''     Dim y0 As Double, y2 As Double
'''     Dim x_radius As Double, x_space As Double
'''     Dim x0 As Double, x1 As Double, x2 As Double
'''     Dim xmin As Integer, ymin As Integer
'''     Dim xmax As Integer, ymax As Integer
'''     Dim two As Integer, max_points As Integer
'''     Dim last_x As Integer, last_y As Integer
'''     Dim map_name As String, link_field As String, OFFSET As Integer
'''     Dim input_map As String, output_map As String
'''     Dim output_database As String
'''     Dim min_longitude As Double, min_latitude As Double
'''     Dim max_longitude As Double, max_latitude As Double
'''     Dim longitude_factor As Double, latitude_factor As Double
'''     Dim number_of_layers As Integer
'''     Dim number_of_map_records As Long, number_of_data_records As Long
'''     Dim bytes_in_map_header As Integer, bytes_in_data_header As Integer
'''     Dim bytes_in_map_record As Integer, bytes_in_data_record As Integer
'''     Dim mid1 As String * 18, layerno As String * 3
'''     Dim object_type As String * 1
'''     Dim stp_x As String * 5, stp_y As String * 5, no_elm As String * 3
'''     Dim pos_off As String * 8, nop As String * 5
'''     Dim object_xmin As String * 5
'''     Dim object_ymin As String * 5, object_xmax As String * 5
'''     Dim object_ymax As String * 5, deleted As String * 1
'''     Dim mid2 As String * 8, density As String * 1
'''     Dim vector_offset As Long
'''     Dim threshold As Single
'''     Dim exceeded As Integer
'''     Dim total_circles As Long, number_of_circles As Long
'''     Dim rows As Long, row0 As Long, row1 As Long
'''     Dim m As Double, previous As Double
'''     Dim time_begin As Single, time_end As Single, total_time As Single
'''     Dim hours As Integer, minutes As Integer, seconds As Single
'''     Dim reply As Integer
'''
'''     'Check for valid input from the user.
'''
'''     Select Case verify_input(3)
'''         Case 0
'''             frmMake_a_Circle.txtRadii.SetFocus
'''             Exit Sub
'''         Case -1
'''             frmMake_a_Circle.txtSpacing.SetFocus
'''             Exit Sub
'''         Case -2
'''             frmMake_a_Circle.txtThreshold.SetFocus
'''             Exit Sub
'''         Case -3
'''             frmMake_a_Circle.txtInput_Map.SetFocus
'''             Exit Sub
'''     End Select
'''
'''     'Start timer.
'''
'''     time_begin = Timer
'''
```

```
'''     'Put filenames into an acceptable format.
'''     'Remove extension from input map name, if necessary. This section
'''     'allows the user to enter file names on frmMake_a_circle with or
'''     'without extension.  If an extension is present, it is temporarily
'''     'removed from the file name, so that secpop can append the necessary
'''     'extensions for the various map files.  The filename and extension
'''     'offered by the user are preserved.
'''
'''     TempString = ""
'''     For i = 1 To (Len(frmMake_a_Circle.txtInput_Map.text))
'''         tempchar = Mid$(frmMake_a_Circle.txtInput_Map.text, i, 1)
'''         If tempchar = "." Then
'''             Exit For
'''         Else
'''             TempString = TempString + tempchar
'''         End If
'''     Next i
'''     orig_input = frmMake_a_Circle.txtInput_Map.text
'''     frmMake_a_Circle.txtInput_Map.text = TempString
'''
'''     'Set up final error checking to catch file names that are too long
'''     'and anything else that may have slipped by.
'''
'''     On Local Error GoTo regional_file_error
'''
'''     'Read in the user input. Convert miles to kilometers.
'''
'''     spacing = Val(frmMake_a_Circle.txtSpacing.text) * _
'''         miles_to_kilometers
'''     threshold = Val(frmMake_a_Circle.txtThreshold.text) / _
'''         (miles_to_kilometers ^ 2)
'''
'''     'Set the radii to be equal to 1 mile increments out to 20 miles
'''     'and 10 mile increments after that. 170 miles is the largest
'''     'possible radii. Convert the miles to kilometers.
'''
'''     radii = Val(txtRadii.text)
'''     For i = 1 To max_number_of_radii Step 1
'''         If (regional_radii(i) < radii) Then
'''             raddis(i) = regional_radii(i) * miles_to_kilometers
'''         Else
'''             raddis(i) = radii * miles_to_kilometers
'''             numrad = i
'''             Exit For
'''         End If
'''     Next i
'''     radii = radii * miles_to_kilometers
'''
'''     'Intialize popultion threshold array.
'''
'''     For i = 1 To numrad Step 1
'''         population_threshold(i) = Fix(threshold * pi * _
'''             (raddis(i) ^ 2)) + 1
'''     Next i
'''
'''     input_map = frmMake_a_Circle.txtInput_Map.text
'''     output_map = "mapplan\map_out"
'''     output_database = "mapplan\db_out"
'''
'''     'Copy the inputmap .dbf and .vtr files to the output map files,
'''     'and supress the dos messages by dumping them into a temp file.
'''
'''     Call copy_file(input_map + ".dbf ", output_map + ".dbf")
'''     Call copy_file(input_map + ".spc ", output_map + ".spc")
'''     Call copy_file(input_map + ".vtr ", output_map + ".vtr")
'''     Call copy_file("mapplan\database.hdr", output_database + ".dbf")
```

```
'''
'''     'Open the input map .cfg file and read in the data from it.
'''
'''     Open input_map + ".cfg" _
'''         For Input Access Read Lock Read Write As #30 Len = 1024
'''     Input #30, xmin, ymin, xmax, ymax, two, max_points
'''     Line Input #30, map_name
'''     Line Input #30, link_field
'''     Input #30, min_longitude, min_latitude, max_longitude, max_latitude
'''     Input #30, longitude_factor, latitude_factor
'''     Close #30
'''
'''     'Open the input map .lyr file, and read in the data from it.
'''
'''     Open input_map + ".lyr" _
'''         For Input Access Read Lock Read Write As #30 Len = 1024
'''     i = 1
'''
'''     'Read in information on each layer of the map until end of file is
'''     'reached.
'''
'''     While (Not EOF(30))
'''         i = i + 1
'''         Input #30, layer_number(i), data_flag(i), display_flag(i)
'''         Input #30, active_layer(i), layer_name(i)
'''         Input #30, pts_color(i), pts_type(i), pts_size(i), pts_mode(i)
'''         Input #30, lines_color(i), lines_type(i), lines_size(i), _
'''             lines_mode(i)
'''         Input #30, polyg_color(i), polyg_type(i), polyg_size(i), _
'''             polyg_mode(i)
'''         Input #30, text_color(i), text_type(i), text_size(i), _
'''             text_mode(i)
'''         layer_number(i) = layer_number(i) + 1
'''         data_flag(i) = 0
'''         active_layer(i) = 0
'''     Wend
'''     number_of_layers = i
'''     Close #30
'''
'''     'Open .lyr output map, put values to put in this file into
'''     'a set of variables, and write these variables to the file.
'''
'''     Open output_map + ".lyr" _
'''         For Output Access Write Lock Read Write As #30 Len = 1024
'''
'''     layer_number(1) = 1
'''     data_flag(1) = 2
'''     display_flag(1) = 1
'''     active_layer(1) = 1
'''     layer_name(1) = "CIRCLES"
'''     pts_color(1) = 6
'''     pts_type(1) = 0
'''     pts_size(1) = 1
'''     pts_mode(1) = 0
'''     lines_color(1) = 6
'''     lines_type(1) = 1
'''     lines_size(1) = 1
'''     lines_mode(1) = 0
'''     polyg_color(1) = 6
'''     polyg_type(1) = 1
'''     polyg_size(1) = 1
'''     polyg_mode(1) = 0
'''     text_color(1) = 0
'''     text_type(1) = 0
'''     text_size(1) = 1
'''     text_mode(1) = 0
```

```
'''
'''     'Print first layer data to file.
'''
'''     Print #30, Mid$(Str$(layer_number(1)), 2); ",";
'''     Print #30, Mid$(Str$(data_flag(1)), 2); ",";
'''     Print #30, Mid$(Str$(display_flag(1)), 2); ",";
'''     Print #30, Mid$(Str$(active_layer(1)), 2); ",";
'''     Print #30, layer_name(1); ",";
'''     Print #30, Mid$(Str$(pts_color(1)), 2); ",";
'''     Print #30, Mid$(Str$(pts_type(1)), 2); ",";
'''     Print #30, Mid$(Str$(pts_size(1)), 2); ",";
'''     Print #30, Mid$(Str$(pts_mode(1)), 2); ",";
'''     Print #30, Mid$(Str$(lines_color(1)), 2); ",";
'''     Print #30, Mid$(Str$(lines_type(1)), 2); ",";
'''     Print #30, Mid$(Str$(lines_size(1)), 2); ",";
'''     Print #30, Mid$(Str$(lines_mode(1)), 2); ",";
'''     Print #30, Mid$(Str$(polyg_color(1)), 2); ",";
'''     Print #30, Mid$(Str$(polyg_type(1)), 2); ",";
'''     Print #30, Mid$(Str$(polyg_size(1)), 2); ",";
'''     Print #30, Mid$(Str$(polyg_mode(1)), 2); ",";
'''     Print #30, Mid$(Str$(text_color(1)), 2); ",";
'''     Print #30, Mid$(Str$(text_type(1)), 2); ",";
'''     Print #30, Mid$(Str$(text_size(1)), 2); ",";
'''     Print #30, Mid$(Str$(text_mode(1)), 2)
'''
'''     'Print all other layers to file.
'''
'''     For i = 2 To number_of_layers Step 1
'''         Print #30, Mid$(Str$(layer_number(i)), 2); ",";
'''         Print #30, Mid$(Str$(data_flag(i)), 2); ",";
'''         Print #30, Mid$(Str$(display_flag(i)), 2); ",";
'''         Print #30, Mid$(Str$(active_layer(i)), 2); ",";
'''         Print #30, layer_name(i); ",";
'''         Print #30, Mid$(Str$(pts_color(i)), 2); ",";
'''         Print #30, Mid$(Str$(pts_type(i)), 2); ",";
'''         Print #30, Mid$(Str$(pts_size(i)), 2); ",";
'''         Print #30, Mid$(Str$(pts_mode(i)), 2); ",";
'''         Print #30, Mid$(Str$(lines_color(i)), 2); ",";
'''         Print #30, Mid$(Str$(lines_type(i)), 2); ",";
'''         Print #30, Mid$(Str$(lines_size(i)), 2); ",";
'''         Print #30, Mid$(Str$(lines_mode(i)), 2); ",";
'''         Print #30, Mid$(Str$(polyg_color(i)), 2); ",";
'''         Print #30, Mid$(Str$(polyg_type(i)), 2); ",";
'''         Print #30, Mid$(Str$(polyg_size(i)), 2); ",";
'''         Print #30, Mid$(Str$(polyg_mode(i)), 2); ",";
'''         Print #30, Mid$(Str$(text_color(i)), 2); ",";
'''         Print #30, Mid$(Str$(text_type(i)), 2); ",";
'''         Print #30, Mid$(Str$(text_size(i)), 2); ",";
'''         Print #30, Mid$(Str$(text_mode(i)), 2)
'''     Next i
'''     Close #30
'''
'''     'Place degrees to radians conversion factor into DGTORD (DeGrees TO
'''     'RaDians).
'''
'''     'dgtord = pi / 180!
'''
'''     'Set Distance per degree longitude/latitude at average latitude.
'''
'''     Call GETDIS(CSng((min_latitude + max_latitude) / 2#))
'''
'''     'Set up variables used to determine how far apart vertically (rows,
'''     'row*) the circles are, and how far apart horizontally (*_radius,
'''     '*_space) these same circles are.
'''
'''     x_radius = ((radii / dpdlon) / longitude_factor) * 1000000#
```

```
'''     y_radius = ((radii / dpdlat) / latitude_factor) * 1000000#
'''     x_space = ((spacing / dpdlon) / longitude_factor) * 1000000#
'''     y_space = ((spacing / dpdlat) / latitude_factor) * 1000000# / 2#
'''
'''     rows = -Int(-(ymax + y_radius - ymin + 1!) / y_space)
'''     row0 = -Int(-(xmax + x_radius - xmin + 1!) / x_space)
'''     row1 = -Int(-(xmax + x_radius - xmin - (x_space / 2!) + 1!) / _
'''         x_space)
'''
'''     'Determine total number of circles.
'''
'''     total_circles = _
'''         Int((rows / 2!) + 0.75) * row0 + Int((rows / 2!)) * row1
'''
'''     'Open all output map files.
'''
'''     Open output_map + ".dbf" _
'''         For Binary Access Read Write Lock Read Write As #30
'''     Open output_map + ".vtr" _
'''         For Binary Access Read Write Lock Read Write As #31
'''     Open output_database + ".dbf" _
'''         For Binary Access Read Write Lock Read Write As #32
'''     Open output_map + ".cfg" _
'''         For Output Access Write Lock Read Write As #33 Len = 1024
'''
'''     'Read in and process various bits of data from files.
'''
'''     Get #30, 5&, number_of_map_records
'''     Get #30, , bytes_in_map_header
'''     Get #30, , bytes_in_map_record
'''
'''     Get #32, 5&, number_of_data_records
'''     Get #32, , bytes_in_data_header
'''     Get #32, , bytes_in_data_record
'''
'''
'''     'Initialize file related variables.
'''
'''     vector_offset = LOF(31)
'''     deleted = " "
'''     layerno = "  1"
'''     no_elm = "  1"
'''     object_type = "3"
'''
'''     If (frmMake_a_Circle.optCircles.Value) Then
'''
'''         'Determine the number of "sides" on each circle.
'''
'''         X = 1# / (4# * radii)
'''         number_of_circle_segments = _
'''             Int(pi / Atn(X / Sqr(1# - X * X)) + 1#)
'''         If (number_of_circle_segments < 10) Then
'''             number_of_circle_segments = 10
'''         End If
'''
'''         'Determine the number of vertices for each circle.
'''
'''         nop = Right$("      " + Str$(number_of_circle_segments + 1#), 5)
'''
'''         'Determine the rotational increment for each circle.
'''
'''         rotation_increment = 2# * pi / number_of_circle_segments
'''
'''         'Update the maximum number of vertices for the map if necessary.
'''
'''         If (max_points < number_of_circle_segments) Then
```

```
'''             max_points = number_of_circle_segments + 1
'''         End If
'''
'''     Else
'''
'''         'If the user has selected "points" from the circle form, set
'''         'number of vertices to 3, and ignore all of the circle
'''         'variables. This will give the user a triangle, which is the
'''         'smallest item that mapplan can process.
'''
'''         nop = "3"
'''
'''     End If
'''
'''     'Print data to the .cfg map file.
'''
'''     Print #33, Mid$(Str$(xmin), 2); ymin; xmax; ymax; two; max_points
'''     Print #33, map_name
'''     Print #33, link_field
'''     Print #33, min_longitude; min_latitude; max_longitude; max_latitude;
'''     Print #33, longitude_factor; latitude_factor
'''     Close #33
'''
'''     'Print data to all of the other map files.
'''
'''     density = "1"
'''
'''     Seek #30, bytes_in_map_header + 1#
'''     Seek #31, LOF(31) + 1#
'''     Seek #32, LOF(32)
'''
'''     'Initialize variables for calculation, hide user input form, and
'''     'show density form.
'''
'''     y0 = ymin
'''     last_y = False
'''     OFFSET = 0
'''     number_of_circles = 0
'''     frmMake_a_Circle.Hide
''''     frmDensity.Show MODELESS
'''     frmDensity.Show
'''     frmDensity.lblPercent_Complete.caption = "0%"
'''
'''     'While latitude and longitude are within bounds, process census data
'''     'files.
'''
'''     While (y0 - y_radius < ymax)
'''         If (OFFSET = 0) Then
'''             x0 = xmin
'''             OFFSET = 1
'''         Else
'''             x0 = xmin + (0.5 * x_space)
'''             OFFSET = 0
'''         End If
'''         x1 = x0 + x_radius
'''         While (x0 - x_radius < xmax)
'''
'''             'Check to see if density form is still visible.  If not,
'''             'reset all variables, close all open files, and exit
'''             'subroutine.
'''
'''             reply = DoEvents()
'''             If frmDensity.Visible = 0 Then
'''                 Close
'''                 Unload frmMake_a_Circle
'''                 Exit Sub
```

```
'''             End If
'''
'''             'Set starting latitude and longitude.
'''
'''             slat = (y0 * 0.000001 * latitude_factor) + min_latitude
'''             slon = -((x0 * 0.000001 * longitude_factor) + min_longitude)
'''
'''             'Reset variable which tells if the calculations on a circle
'''             'are done or not.
'''
'''             exceeded = False
'''
'''             'Call function to perform actual calculations.
'''
'''             Call popdensity(exceeded)
'''
'''             'If the number of people in the circle are greater than the
'''             'threshold, then plot the circle in the map file, in a
'''             'manner understood by mapplan, and increment counters.
'''
'''             If (exceeded) Then
'''                 number_of_map_records = number_of_map_records + 1
'''                 number_of_data_records = number_of_data_records + 1
'''
'''                 mid1 = Left$("POP" + _
'''                     Mid$(Str$(number_of_data_records), 2) + _
'''                     "                    ", 18)
'''                 stp_x = Right$("      " + Str$(Int(x0 + 0.5)), 5)
'''                 stp_y = Right$("      " + Str$(Int(y0 + 0.5)), 5)
'''                 pos_off = Right$("         " + Str$(vector_offset), 8)
'''
'''                 If (Int(x0 - x_radius + 0.5) >= xmin) Then
'''                     object_xmin = Right$("      " + _
'''                         Str$(Int(x0 - x_radius + 0.5)), 5)
'''                 Else
'''                     object_xmin = Right$("      " + Str$(xmin), 5)
'''                 End If
'''                 If (Int(y0 - y_radius + 0.5) >= ymin) Then
'''                     object_ymin = Right$("      " + _
'''                         Str$(Int(y0 - y_radius + 0.5)), 5)
'''                 Else
'''                     object_ymin = Right$("      " + Str$(ymin), 5)
'''                 End If
'''                 If (Int(x0 + x_radius + 0.5) <= xmax) Then
'''                     object_xmax = Right$("      " + _
'''                         Str$(Int(x0 + x_radius + 0.5)), 5)
'''                 Else
'''                     object_xmax = Right$("      " + Str$(xmax), 5)
'''                 End If
'''                 If (Int(y0 + y_radius + 0.5) <= ymax) Then
'''                     object_ymax = Right$("      " + _
'''                         Str$(Int(y0 + y_radius + 0.5)), 5)
'''                 Else
'''                     object_ymax = Right$("      " + Str$(ymax), 5)
'''                 End If
'''
'''                 'Write data to output file.
'''
'''                 Put #30, , deleted
'''                 Put #30, , mid1
'''                 Put #30, , layerno
'''                 Put #30, , object_type
'''                 Put #30, , stp_x
'''                 Put #30, , stp_y
'''                 Put #30, , no_elm
'''                 Put #30, , pos_off
```

```
'''             Put #30, , nop
'''             Put #30, , object_xmin
'''             Put #30, , object_ymin
'''             Put #30, , object_xmax
'''             Put #30, , object_ymax
'''
'''             mid2 = Left$(mid1, 8)
'''
'''             Put #32, , deleted
'''             Put #32, , mid2
'''             Put #32, , density
'''
'''             'Plot the actual points for the circle, and place them
'''             'into the vector file.
'''
'''             If (frmMake_a_Circle.optCircles.Value) Then
'''                 x_int = Int(x1 + 0.5)
'''                 y_int = Int(y0 + 0.5)
'''                 Put #31, , x_int
'''                 Put #31, , y_int
'''                 vector_offset = vector_offset + 4#
'''
'''                 For i = 1 To number_of_circle_segments Step 1
'''                     x2 = x_radius * Cos(i * rotation_increment) + x0
'''                     y2 = y_radius * Sin(i * rotation_increment) + y0
'''                     If (x2 <= xmin) Then x_int = Int(xmin + 0.1)
'''                     If (x2 >= xmax) Then y_int = Int(ymin + 0.1)
'''                     If (y2 <= ymin) Then x_int = Int(xmin - 0.1)
'''                     If (y2 >= ymax) Then y_int = Int(ymin - 0.1)
'''                     x_int = Int(x2 + 0.5)
'''                     y_int = Int(y2 + 0.5)
'''                     vector_offset = vector_offset + 4#
'''                     Put #31, , x_int
'''                     Put #31, , y_int
'''                 Next i
'''             Else
'''
'''                 'Draw very small triangles instead of circles if
'''                 'that is what the user has requested. This method
'''                 'is much faster than plotting circles.
'''
'''                 x2 = (x0 - 5#)
'''                 y2 = (y0 - 4.33)
'''                 If (x2 <= xmin) Then x_int = Int(xmin + 0.1)
'''                 If (x2 >= xmax) Then y_int = Int(ymin + 0.1)
'''                 If (y2 <= ymin) Then x_int = Int(xmin - 0.1)
'''                 If (y2 >= ymax) Then y_int = Int(ymin - 0.1)
'''                 x_int = Int(x2 + 0.5)
'''                 y_int = Int(y2 + 0.5)
'''                 Put #31, , x_int
'''                 Put #31, , y_int
'''
'''                 y2 = (y0 + 4.33)
'''                 If (x2 >= xmax) Then y_int = Int(ymin + 0.1)
'''                 If (y2 >= ymax) Then y_int = Int(ymin - 0.1)
'''                 y_int = Int(y2 + 0.5)
'''                 x_int = Int(x0 + 0.5)
'''                 Put #31, , x_int
'''                 Put #31, , y_int
'''
'''                 x2 = (x0 + 5#)
'''                 y2 = (y0 - 4.33)
'''                 If (x2 <= xmin) Then x_int = Int(xmin + 0.1)
'''                 If (x2 >= xmax) Then y_int = Int(ymin + 0.1)
'''                 If (y2 <= ymin) Then x_int = Int(xmin - 0.1)
'''                 If (y2 >= ymax) Then y_int = Int(ymin - 0.1)
```

```
'''                     x_int = Int(x2 + 0.5)
'''                     y_int = Int(y2 + 0.5)
'''                     Put #31, , x_int
'''                     Put #31, , y_int
'''
'''                     vector_offset = vector_offset + 12#
'''                 End If
'''             End If
'''
'''             'Increment the number of circles, and tell the user what %
'''             'of calc is complete.
'''
'''             number_of_circles = number_of_circles + 1
'''             If ((number_of_circles / total_circles * 100) < 0.1) Then
'''                 frmDensity.lblPercent_Complete.caption = "0 %"
'''             Else
'''                 frmDensity.lblPercent_Complete.caption = _
'''                     Left$(Str$(number_of_circles / _
'''                     total_circles * 100), 5) + "%"
'''             x0 = x0 + x_space
'''             x1 = x0 + x_radius
'''             End If
'''         Wend
'''         y0 = y0 + y_space
'''     Wend
'''
'''     'When this point is reached, the calculation is finished. This part
'''     'finishes up writing to the files, and closes all files.
'''
'''     Put #32, 5&, number_of_data_records
'''     Close 31
'''     Close 32
'''
'''     Open input_map + ".dbf" _
'''         For Binary Access Read Write Lock Read Write As #31
'''     Get #31, 9&, bytes_in_map_header
'''     Seek #31, bytes_in_map_header + 1#
'''
'''     While (Not EOF(31))
'''         Get #31, , deleted
'''         Get #31, , mid1
'''         Get #31, , layerno
'''         layerno = Right$("   " + Str$(Val(layerno) + 1), 3)
'''         Get #31, , object_type
'''         Get #31, , stp_x
'''         Get #31, , stp_y
'''         Get #31, , no_elm
'''         Get #31, , pos_off
'''         Get #31, , nop
'''         Get #31, , object_xmin
'''         Get #31, , object_ymin
'''         Get #31, , object_xmax
'''         Get #31, , object_ymax
'''
'''         Put #30, , deleted
'''         Put #30, , mid1
'''         Put #30, , layerno
'''         Put #30, , object_type
'''         Put #30, , stp_x
'''         Put #30, , stp_y
'''         Put #30, , no_elm
'''         Put #30, , pos_off
'''         Put #30, , nop
'''         Put #30, , object_xmin
'''         Put #30, , object_ymin
'''         Put #30, , object_xmax
```

```
'''         Put #30, , object_ymax
'''     Wend
'''
'''     Put #30, 5&, number_of_map_records
'''     Close 30
'''     Close 31
'''
'''     'Stop timer.
'''
'''     time_end = Timer
'''
'''     'Convert the time into standard format.
'''
'''     If time_end <= time_begin Then
'''         total_time = time_end - time_begin + seconds_in_a_day
'''     Else
'''         total_time = time_end - time_begin
'''     End If
'''     hours = total_time \ 3600
'''     minutes = (total_time \ 60) - (hours * 60)
'''     seconds = (total_time - ((hours * 60 + minutes) * 60))
'''
'''     'Display time to user.
'''
'''     frmDensity.cmdClose.caption = "Continue"
'''     frmDensity.lblCaption.caption = "Total Time"
'''     frmDensity.lblPercent_Complete.caption = Str$(hours) + ":" + _
'''         LTrim$(Str$(minutes)) + ":" + Format$(seconds, "0.00")
'''     While (frmDensity.Visible <> 0)
'''        reply = DoEvents()
'''     Wend
'''
'''     'Run mapplan.
'''
''''     RUN "run_mppr.exe"
'''
'''Exit Sub
'''
'''
'''regional_file_error:
'''
'''     If (Err = 6) Then
'''         MsgBox Error$(Err) + Chr$(13) + Chr$(10) + _
'''             "In Regional Calulation Routine." + Chr$(13) + _
'''             Chr$(10) + "Use smaller circles, spacing, or map."
'''     Else
'''         MsgBox Error$(Err) + Chr$(13) + Chr$(10) + _
'''             "In Regional Calulation Routine."
'''     End If
'''
'''     'msgbox "Line number =" + str$(erl)
'''
'''     frmMake_a_Circle.txtInput_Map.text = orig_input
'''     If frmDensity.Visible = True Then
'''         Unload frmDensity
'''     End If
'''     Exit Sub
'''End Sub
'''
'''
'''Private Sub cmdClose_Click()
'''
''''Close regional calculation before the calculation
''''has been run.
'''
'''     Unload frmMake_a_Circle
```

```
'''
'''End Sub
'''
'''
'''Private Sub cmdInput_files_Click()
'''
''''This routine allows the user to select an input map file for regional
''''calculations.
'''
'''     Dim filenum As Integer, ForeColor As Integer, BackColor As Integer
'''     Dim Flags As Integer, Cancel As Integer
'''     Dim Filename As String
'''     Dim Pathname As String
'''     Dim DefaultExt As String, DialogTitle As String
'''
'''     'Initialize file dialogue box.
'''
'''     DefaultExt = "*.CFG"
'''     DialogTitle = "Input Map File"
'''     BackColor = vbWhite
'''     ForeColor = vbBlack
'''     Pathname = "mapplan\maps"
'''
'''     'Prompt user for file name (from dialogue box).
'''
''''ss     Call FileOpen(Filename, Pathname, DefaultExt, DialogTitle, _
''''ss         ForeColor, BackColor, Flags, Cancel)
'''
'''     'If user did not choose cancel in dialogue box...
'''
'''     If Not Cancel Then
'''
'''         'If filename is not in current directory, append path name
'''         'to it.
'''
'''         If Pathname <> "" Then Pathname = Pathname + "\"
'''         Filename = Pathname + Filename
'''         frmMake_a_Circle.txtInput_Map.text = Filename
'''
'''         'Set focus on next field in regional form.
'''
'''         frmMake_a_Circle.cmdCalculate_Regional.SetFocus
'''
'''     Else
'''
'''         frmMake_a_Circle.txtInput_Map.SetFocus
'''
'''     End If
'''
'''End Sub
'''
'''
'''Private Sub optCircles_Click()
'''
'''     frmMake_a_Circle.optCircles.Value = True
'''     frmMake_a_Circle.cmdCalculate_Regional.SetFocus
'''
'''End Sub
'''
'''
'''Private Sub optPoints_Click()
'''
'''     frmMake_a_Circle.optPoints.Value = True
'''     frmMake_a_Circle.cmdCalculate_Regional.SetFocus
'''
'''End Sub
```

H.21 Visual Basic Form Module: Cmndlgf.frm

```
VERSION 5.00
Begin VB.Form frmCmnDlg
   Appearance      =   0  'Flat
   BackColor       =   &H00C0C0C0&
   BorderStyle     =   1  'Fixed Single
   Caption         =   "Common Dialog"
   ClientHeight    =   6195
   ClientLeft      =   1335
   ClientTop       =   1050
   ClientWidth     =   6840
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   8.25
      Charset         =   0
      Weight          =   700
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H00000000&
   MaxButton       =   0   'False
   MinButton       =   0   'False
   PaletteMode     =   1  'UseZOrder
   ScaleHeight     =   6195
   ScaleWidth      =   6840
   StartUpPosition =   1  'CenterOwner
   Begin VB.PictureBox pctFilePrint
      Appearance      =   0  'Flat
      BackColor       =   &H00C0C0C0&
      ForeColor       =   &H00000000&
      Height          =   2400
      Left            =   0
      ScaleHeight     =   2370
      ScaleWidth      =   6450
      TabIndex        =   9
      TabStop         =   0   'False
      Top             =   -60
      Visible         =   0   'False
      Width           =   6480
      Begin VB.Frame fraPrintTarget
         Appearance      =   0  'Flat
         BackColor       =   &H00C0C0C0&
         Caption         =   "Print Target"
         ForeColor       =   &H00000000&
         Height          =   1920
         Left            =   108
         TabIndex        =   10
         Top             =   240
         Width           =   4644
         Begin VB.OptionButton optPrintTarget
            Appearance      =   0  'Flat
            BackColor       =   &H00C0C0C0&
            Caption         =   "LPT&1"
            ForeColor       =   &H00000000&
            Height          =   240
            Index           =   0
            Left            =   108
            TabIndex        =   11
            Top             =   0
            Value           =   -1  'True
            Width           =   972
         End
         Begin VB.OptionButton optPrintTarget
```

```
    Appearance      =   0  'Flat
    BackColor       =   &H00C0C0C0&
    Caption         =   "LPT&2"
    ForeColor       =   &H00000000&
    Height          =   240
    Index           =   1
    Left            =   108
    TabIndex        =   12
    TabStop         =   0   'False
    Top             =   240
    Width           =   972
End
Begin VB.OptionButton optPrintTarget
    Appearance      =   0  'Flat
    BackColor       =   &H00C0C0C0&
    Caption         =   "LPT&3"
    ForeColor       =   &H00000000&
    Height          =   240
    Index           =   2
    Left            =   108
    TabIndex        =   13
    TabStop         =   0   'False
    Top             =   480
    Width           =   972
End
Begin VB.OptionButton optPrintTarget
    Appearance      =   0  'Flat
    BackColor       =   &H00C0C0C0&
    Caption         =   "&File:"
    ForeColor       =   &H00000000&
    Height          =   240
    Index           =   3
    Left            =   108
    TabIndex        =   14
    TabStop         =   0   'False
    Top             =   720
    Width           =   972
End
Begin VB.TextBox txtPrintFile
    Appearance      =   0  'Flat
    BackColor       =   &H00C0C0C0&
    Enabled         =   0   'False
    ForeColor       =   &H00000000&
    Height          =   720
    Left            =   1188
    TabIndex        =   15
    Top             =   480
    Width           =   3132
End
Begin VB.PictureBox pctPrintAppend
    Appearance      =   0  'Flat
    BackColor       =   &H00C0C0C0&
    BorderStyle     =   0  'None
    Enabled         =   0   'False
    ForeColor       =   &H00000000&
    Height          =   240
    Left            =   0
    ScaleHeight     =   240
    ScaleWidth      =   4425
    TabIndex        =   16
    TabStop         =   0   'False
    Top             =   1200
    Width           =   4428
    Begin VB.OptionButton optPrintAppend
        Appearance      =   0  'Flat
        BackColor       =   &H00C0C0C0&
```

```
            Caption         =   "A&ppend"
            ForeColor       =   &H00000000&
            Height          =   240
            Index           =   0
            Left            =   1836
            TabIndex        =   18
            TabStop         =   0   'False
            Top             =   0
            Width           =   1188
         End
         Begin VB.OptionButton optPrintAppend
            Appearance      =   0   'Flat
            BackColor       =   &H00C0C0C0&
            Caption         =   "&Replace"
            ForeColor       =   &H00000000&
            Height          =   240
            Index           =   1
            Left            =   3132
            TabIndex        =   19
            Top             =   0
            Value           =   -1  'True
            Width           =   1188
         End
         Begin VB.Label lblPrintAppend
            Appearance      =   0   'Flat
            BackColor       =   &H00C0C0C0&
            Caption         =   "If file exists:"
            ForeColor       =   &H00000000&
            Height          =   240
            Left            =   108
            TabIndex        =   17
            Top             =   0
            Width           =   1728
         End
      End
   End
   Begin VB.CommandButton cmdPrintOK
      Appearance      =   0   'Flat
      BackColor       =   &H00C0C0C0&
      Caption         =   "OK"
      Height          =   720
      Left            =   4968
      TabIndex        =   20
      Top             =   240
      Width           =   1404
   End
   Begin VB.CommandButton cmdPrintCancel
      Appearance      =   0   'Flat
      BackColor       =   &H00C0C0C0&
      Caption         =   "Cancel"
      Height          =   720
      Left            =   4968
      TabIndex        =   21
      Top             =   960
      Width           =   1404
   End
   Begin VB.TextBox txtPrintCopies
      Appearance      =   0   'Flat
      BackColor       =   &H00C0C0C0&
      ForeColor       =   &H00000000&
      Height          =   240
      Left            =   5832
      TabIndex        =   23
      Top             =   1920
      Width           =   540
   End
```

```
   Begin VB.Label lblPrintCopies
      Appearance      =   0  'Flat
      BackColor       =   &H00C0C0C0&
      Caption         =   "&Copies:"
      ForeColor       =   &H00000000&
      Height          =   240
      Left            =   4968
      TabIndex        =   22
      Top             =   1920
      Width           =   756
   End
End
Begin VB.PictureBox pctFileOpen
   Appearance      =   0  'Flat
   BackColor       =   &H00C0C0C0&
   ForeColor       =   &H00000000&
   Height          =   3864
   Left            =   120
   ScaleHeight     =   3840
   ScaleWidth      =   5370
   TabIndex        =   0
   TabStop         =   0   'False
   Top             =   2340
   Visible         =   0   'False
   Width           =   5400
   Begin VB.TextBox txtOpenFile
      Appearance      =   0  'Flat
      BackColor       =   &H00C0C0C0&
      ForeColor       =   &H00000000&
      Height          =   720
      Left            =   1296
      TabIndex        =   2
      Top             =   240
      Width           =   2484
   End
   Begin VB.CommandButton cmdOpenOK
      Appearance      =   0  'Flat
      BackColor       =   &H00C0C0C0&
      Caption         =   "OK"
      Height          =   720
      Left            =   3996
      TabIndex        =   7
      Top             =   240
      Width           =   1296
   End
   Begin VB.CommandButton cmdOpenCancel
      Appearance      =   0  'Flat
      BackColor       =   &H00C0C0C0&
      Caption         =   "Cancel"
      Height          =   720
      Left            =   3996
      TabIndex        =   8
      Top             =   960
      Width           =   1296
   End
   Begin VB.DriveListBox drvOpenList
      Appearance      =   0  'Flat
      BackColor       =   &H00C0C0C0&
      ForeColor       =   &H00000000&
      Height          =   240
      Left            =   2052
      TabIndex        =   5
      Top             =   1440
      Width           =   1728
   End
   Begin VB.FileListBox filOpenList
```

```
      Appearance      =   0  'Flat
      BackColor       =   &H00C0C0C0&
      ForeColor       =   &H00000000&
      Height          =   2370
      Left            =   108
      TabIndex        =   4
      Top             =   1188
      Width           =   1728
   End
   Begin VB.DirListBox dirOpenList
      Appearance      =   0  'Flat
      BackColor       =   &H00C0C0C0&
      ForeColor       =   &H00000000&
      Height          =   2160
      Left            =   2052
      TabIndex        =   6
      Top             =   1680
      Width           =   1728
   End
   Begin VB.Label lblOpenFile
      Appearance      =   0  'Flat
      BackColor       =   &H00C0C0C0&
      Caption         =   "File &Name:"
      ForeColor       =   &H00000000&
      Height          =   240
      Left            =   108
      TabIndex        =   1
      Top             =   480
      Width           =   1188
   End
   Begin VB.Label lblOpenPath
      Appearance      =   0  'Flat
      BackColor       =   &H00C0C0C0&
      Caption         =   "C:\"
      ForeColor       =   &H00000000&
      Height          =   240
      Left            =   216
      TabIndex        =   3
      Top             =   960
      Width           =   3564
   End
End
End
Attribute VB_Name = "frmCmnDlg"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
'''
'''Option Explicit
'''
''''$INCLUDE: 'secpop90.bi'
'''
'''' -------------------------------------------------------------------
'''' Visual Basic for MS-DOS Common Dialog Toolkit
''''
'''' The Common Dialog Toolkit (CMNDLG.BAS and CMNDLGF.FRM)
'''' provides support for the following dialogs:
''''       FileOpen
''''       FileSave
''''       FilePrint
''''
'''' Support for each dialog is provided via procedures with
'''' these same names that create the corresponding dialog
'''' and return user input to your program.  These procedures
'''' only provide the user interface and return user input.
```

```
'''' They do not actually carry out the corresponding actions
'''' such as opening the file.  Detailed descriptions of
'''' these procedures are contained in the comment headers
'''' above each.
''''
'''' All common dialogs are created from the same form (CMNDLGF.FRM).
'''' The necessary controls for each dialog are children of
'''' a container picture box for the dialog.  Thus the
'''' form (CMNDLGF.FRM) contains a picture box with
'''' appropriate controls for common dialog listed above.
'''' When a particular common dialog is created and displayed,
'''' the container picture box for that dialog is made visible
'''' (thus all controls on that picture box become visible)
'''' and the form is centered and sized to match the
'''' container picture box.
''''
'''' To use these common dialogs in your programs, include
'''' CMNDLG.BAS and CMNDLGF.FRM in your program or use the
'''' supplied library (CMNDLG.LIB, CMNDLGA.LIB - AltMath version
'''' for Professional Edition only) and Quick library (CMNDLG.QLB)
'''' and call the appropriate procedure to invoke the dialog
'''' you need.
''''
'''' Copyright (C) 1982-1992 Microsoft Corporation
''''
'''' You have a royalty-free right to use, modify, reproduce
'''' and distribute the sample applications and toolkits provided with
'''' Visual Basic for MS-DOS (and/or any modified version)
'''' in any way you find useful, provided that you agree that
'''' Microsoft has no warranty, obligations or liability for
'''' any of the sample applications or toolkits.
'''' ------------------------------------------------------------------------
'''
'''' Click event procedure for Open/Save dialog Cancel button.
'''' Makes dialog invisible to return control to FileOpen or FileSave
'''' procedure (dialog was shown modally).
''''
'''Private Sub cmdOpenCancel_Click()
'''     txtOpenFile.SetFocus
'''     Visible = False
'''End Sub
'''
'''' Click event procedure for Open/Save dialog OK button.
'''' Determines whether user has selected a file or whether
'''' path and pattern need to be changed.
''''
'''Private Sub cmdOpenOK_CLick()
'''
'''     Dim OldPattern As String
'''
'''     ' Set up error handling for directory/drive change errors.
'''     On Local Error GoTo OKError
'''
'''     cmdOpenOK.SetFocus                      ' Set focus to button, so focus can be reset to
edit field if needed.
'''
'''     ' Update Directory listbox path if user single
'''     ' clicked or used arrow keys in Directory listbox
'''     ' (only double click automatically changes path).
'''     dirOpenList.Path = dirOpenList.List(dirOpenList.ListIndex)
'''
'''     ' If edit field filename does not match File listbox filename
'''     ' then assign edit field value to File listbox filename
'''     ' and let it determine if path or pattern need to be
'''     ' changed.
'''     If filOpenList.filename <> txtOpenFile.text Then
```

```
'''         OldPattern$ = filOpenList.Pattern    ' Save old pattern.
'''
'''         ' Let File listbox control determine if path
'''         ' or pattern or filename needs to be updated.
'''         ' PathChange event will be triggered if path needs
'''         ' updating, PatternChange event will be triggered if
'''         ' pattern needs updating, and DblClick event will
'''         ' be triggered if a valid filename has been given.
'''         filOpenList.filename = txtOpenFile.text
'''
'''         ' If a valid filename was not given (dialog is
'''         ' still visible to user after DblClick event),
'''         ' then update the edit field appropriately.
'''         If Visible = True Then
'''             ' If no pattern change was indicated then either
'''             ' a new filename was specified for Save dialog
'''             ' or file was not found for Open dialog.
'''             If (InStr(txtOpenFile.text, "*") + InStr(txtOpenFile.text, "?") < 1) Then
'''                 If InStr(Tag, "SAVE") Then
'''                     Call filOpenList_DblClick
'''                 Else
'''                     MsgBox "File not found", 0, caption
'''                     filOpenList.Pattern = OldPattern$    ' Restore old File listbox search
pattern.
'''                     txtOpenFile.SetFocus
'''                 End If
'''             ' Pattern change was indicated so just update
'''             ' textbox with pattern.
'''             Else
'''                 txtOpenFile.text = filOpenList.Pattern
'''                 txtOpenFile.SetFocus
'''             End If
'''         End If
'''     ' File has been selected by user so close dialog
'''     ' and return control to FileOpen or FileSave routine.
'''     Else
'''         Call filOpenList_DblClick
'''     End If
'''
'''OKErrorReturn:
'''     Exit Sub
'''
'''' Drive/Path error handling routine.
'''OKError:
'''     MsgBox Error$, 0, caption           ' Display error message.
'''     txtOpenFile.SetFocus                ' Set focus to edit field so error can be fixed.
'''     Resume OKErrorReturn                ' Exit procedure.
'''End Sub
'''
'''' Click event procedure for Print dialog Cancel button.
'''' Makes dialog invisible to return control to FilePrint
'''' procedure (dialog was shown modally).
''''
'''Private Sub cmdPrintCancel_Click()
'''     Visible = False
'''End Sub
'''
'''' Click event procedure for Print dialog OK button.
'''' Sets print destination (PRINTER.PrintTarget) and
'''' makes dialog invisible to return control to FilePrint
'''' procedure (dialog was shown modally).
''''
'''Private Sub cmdPrintOK_Click()
'''     ' Set up error handling for print to file errors.
'''     On Local Error GoTo PrintError
'''
```

```
'''     ' Set print target
'''     If optPrintTarget(0).Value Then
'''         Printer.PrintTarget = "LPT1:"       ' Use Basic LPT1 device (colon specifies this).
'''     ElseIf optPrintTarget(1).Value Then
'''         Printer.PrintTarget = "LPT2:"       ' Use Basic LPT2 device (colon specifies this).
'''     ElseIf optPrintTarget(2).Value Then
'''         Printer.PrintTarget = "LPT3"        ' No Basic LPT3 device, treat as a normal file
open.
'''     Else
'''         ' Print target is a file.
'''         Printer.PrintTarget = txtPrintFile.text
'''         ' If user specified "Replace" instead of "Append"
'''         ' option, delete existing file.
'''         If optPrintAppend(1).Value Then Kill txtPrintFile.text
'''     End If
'''     Visible = False
'''     cmdPrintCancel.Tag = "FALSE"
'''
'''     Exit Sub
'''
'''' Print to file error handling routine.
'''' Ignores File Not Found error that occurs when
'''' deleting a file that does not exist (when user
'''' chooses "Replace" option).
'''PrintError:
'''     Resume Next
'''End Sub
'''
'''' Change event procedure for Open/Save dialog Directory listbox.
'''' Updates the File listbox path to reflect
'''' the directory change.
''''
'''Private Sub dirOpenList_Change()
'''     ' Set up error handling for path errors.
'''     On Local Error GoTo DirChangeError
'''
'''     ' Update file listbox path.
'''     filOpenList.Path = dirOpenList.Path
'''
'''     ' Display new path to the user.
'''     lblOpenPath.caption = dirOpenList.Path
'''
'''     ' Update text box with search pattern.
'''     txtOpenFile.text = filOpenList.Pattern
'''
'''DirChangeErrorReturn:
'''     Exit Sub
'''
'''' Path change error handling routine.
'''DirChangeError:
'''     MsgBox Error$, 0, caption           ' Display error message.
'''     txtOpenFile.SetFocus                ' Set focus to edit field so error can be fixed.
'''     Resume DirChangeErrorReturn         ' Exit procedure.
'''End Sub
'''
'''' Change event procedure for Open/Save dialog Drive listbox.
'''' Updates the Directory listbox path to reflect
'''' the drive change.
''''
'''Private Sub drvOpenList_Change()
'''     ' Set up error handling for path errors.
'''     On Local Error GoTo DriveChangeError
'''
'''     ' Update Dir listbox path.
'''     dirOpenList.Path = drvOpenList.Drive
'''
```

```
'''DriveChangeErrorReturn:
'''     Exit Sub
'''
'''' Path change error handling routine.
'''DriveChangeError:
'''     MsgBox Error$, 0, caption               ' Display error message.
'''     drvOpenList.Drive = dirOpenList.Path    ' Reset drive.
'''     Resume DriveChangeErrorReturn           ' Exit procedure.
'''End Sub
'''
'''' Click event procedure for Open/Save dialog File listbox.
'''' Selects the file and updates the edit field.
''''
'''Private Sub filOpenList_Click()
'''     txtOpenFile.text = filOpenList.filename
'''End Sub
'''
'''' Double Click event procedure for Open/Save dialog File listbox.
'''' File has been selected by the user so make dialog
'''' invisible to return control to FileOpen or FileSave
'''' procedure (dialog was shown modally).
''''
'''Private Sub filOpenList_DblClick()
'''     txtOpenFile.SetFocus
'''     Visible = False
'''     cmdOpenCancel.Tag = "FALSE"
'''End Sub
'''
'''' PathChange event procedure for Open/Save dialog File listbox.
'''' Updates the Drive listbox drive and Directory
'''' listbox path to reflect the change.
''''
'''Private Sub filOpenList_PathChange()
'''     ' Set up error handling for path errors.
'''     On Local Error GoTo FileChangeError
'''
'''     ' Update drive and path.
'''     drvOpenList.Drive = filOpenList.Path
'''     dirOpenList.Path = filOpenList.Path
'''
'''FileChangeErrorReturn:
'''     Exit Sub
'''
'''' Path change error handling routine.
'''FileChangeError:
'''     MsgBox Error$, 0, caption               ' Display error message.
'''     drvOpenList.Drive = dirOpenList.Path    ' Reset drive.
'''     filOpenList.Path = dirOpenList.Path     ' Reset path.
'''     Resume FileChangeErrorReturn            ' Exit procedure.
'''End Sub
'''
'''' PatternChange event procedure for Open/Save dialog File listbox.
'''' Uppercases search pattern for subsequent display
'''' in edit field.
''''
'''Private Sub filOpenList_PatternChange()
'''     filOpenList.Pattern = UCase$(filOpenList.Pattern)
'''End Sub
'''
'''' Click event procedure for Print dialog PrintTarget option buttons (control array)
'''' Handles print target selection.
''''
'''Private Sub optPrintTarget_Click(index As Integer)
'''     ' If file is chosen as print target, enable
'''     ' filename edit field and append/replace options.
'''     If index = 3 Then
```

```
'''         txtPrintFile.Enabled = True
'''         pctPrintAppend.Enabled = True
'''     ' If LPT1, LPT2, LPT3 is chosen as print target,
'''     ' disable filename edit field and append/replace options.
'''     Else
'''         txtPrintFile.Enabled = False
'''         pctPrintAppend.Enabled = False
'''     End If
'''End Sub
```

H.22 Visual Basic Form Module: Densform.frm

```
VERSION 5.00
Begin VB.Form frmDensity
   Appearance      =   0  'Flat
   BackColor       =   &H80000004&
   BorderStyle     =   1  'Fixed Single
   Caption         =   "Calculation Status"
   ClientHeight    =   2520
   ClientLeft      =   2955
   ClientTop       =   1290
   ClientWidth     =   2520
   ControlBox      =   0   'False
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   8.25
      Charset         =   0
      Weight          =   700
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H00000000&
   MaxButton       =   0   'False
   MinButton       =   0   'False
   PaletteMode     =   1  'UseZOrder
   ScaleHeight     =   2520
   ScaleWidth      =   2520
   StartUpPosition =   1  'CenterOwner
   Begin VB.Frame Frame2
      Appearance      =   0  'Flat
      BackColor       =   &H80000004&
      ForeColor       =   &H00000000&
      Height          =   720
      Left            =   216
      TabIndex        =   0
      Top             =   480
      Width           =   1944
      Begin VB.Label lblPercent_Complete
         Appearance      =   0  'Flat
         BackColor       =   &H80000004&
         ForeColor       =   &H00000000&
         Height          =   240
         Left            =   0
         TabIndex        =   1
         Top             =   0
         Width           =   1728
      End
   End
   Begin VB.CommandButton cmdClose
      Appearance      =   0  'Flat
      BackColor       =   &H00808000&
      Caption         =   "&Close"
      Height          =   720
      Left            =   216
      TabIndex        =   2
      Top             =   1440
      Width           =   1944
   End
   Begin VB.Label lblCaption
      Appearance      =   0  'Flat
      BackColor       =   &H80000004&
      Caption         =   "Percent Complete"
      ForeColor       =   &H00000000&
      Height          =   240
```

```
      Left            =   324
      TabIndex        =   3
      Top             =   240
      Width           =   1728
   End
End
Attribute VB_Name = "frmDensity"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

'This form allows the user to monitor the status
'of a calculation, and allows the user to terminate
'the calculation before it has been completed.

Private Sub cmdClose_Click()

'This proceedure allows the user to end a regional
'calculation.

    Dim reply As Integer

    'If a calculation is not in progress then unload the calculate form.

    If frmDensity.cmdClose.caption = "Continue" Then

        Unload frmDensity

    Else

        'If the calculation is in progress ask the user if they really
        'want to leave.

        reply = MsgBox("Do you really want to stop the present " + _
            "calculation?", vbYesNo + vbDefaultButton1, "Close")

        If (reply = vbYes) Then

            Unload frmDensity

        End If

    End If

End Sub
```

H.23 Visual Basic Form Module: frmCalculate.frm

```
VERSION 5.00
Begin VB.Form frmCalculate
   BorderStyle     =   1  'Fixed Single
   Caption         =   "Calculate"
   ClientHeight    =   2985
   ClientLeft      =   45
   ClientTop       =   330
   ClientWidth     =   3885
   Icon            =   "frmCalculate.frx":0000
   LinkTopic       =   "Form1"
   MaxButton       =   0   'False
   MDIChild        =   -1  'True
   MinButton       =   0   'False
   ScaleHeight     =   2985
   ScaleWidth      =   3885
   Begin VB.Frame Frame2
      Caption         =   "Estimated Time Remaining"
      Height          =   840
      Left            =   248
      TabIndex        =   4
      Top             =   1215
      Width           =   3495
      Begin VB.TextBox txtTime
         BackColor       =   &H8000000F&
         BeginProperty Font
            Name            =   "MS Sans Serif"
            Size            =   9.75
            Charset         =   0
            Weight          =   400
            Underline       =   0   'False
            Italic          =   0   'False
            Strikethrough   =   0   'False
         EndProperty
         Height          =   375
         Left            =   120
         TabIndex        =   5
         Top             =   315
         Width           =   3255
      End
   End
   Begin VB.Frame Frame1
      Caption         =   "Calculation  Status"
      Height          =   840
      Left            =   240
      TabIndex        =   2
      Top             =   195
      Width           =   3495
      Begin VB.TextBox txtStatus
         BackColor       =   &H8000000F&
         BeginProperty Font
            Name            =   "MS Sans Serif"
            Size            =   9.75
            Charset         =   0
            Weight          =   400
            Underline       =   0   'False
            Italic          =   0   'False
            Strikethrough   =   0   'False
         EndProperty
         Height          =   375
         Left            =   120
         TabIndex        =   3
         Top             =   315
         Width           =   3255
```

```
      End
   End
   Begin VB.CommandButton cmdClose
      BackColor       =   &H00C0C0C0&
      Caption         =   "Close"
      Height          =   480
      Left            =   2115
      TabIndex        =   1
      Top             =   2325
      Width           =   1188
   End
   Begin VB.CommandButton cmdCalculate
      BackColor       =   &H00C0C0C0&
      Caption         =   "&Calculate"
      Height          =   480
      Left            =   540
      TabIndex        =   0
      Top             =   2310
      Width           =   1188
   End
End
Attribute VB_Name = "frmCalculate"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

'This form handles much of the overhead associated
'with regional calculations, including the user
'interface, reporting, pausing, and canceling

Public Sub cmdCalculate_Click()

    'This subroutine drives the function that performs site calculations
    'It calls the popcalc function and keeps track of the amount of
    'time spent on a site calculation.

    Const seconds_in_a_day = 86400

    Dim time_begin As Single
    Dim time_end As Single
    Dim total_time As Single
    Dim hours As Integer
    Dim minutes As Integer
    Dim seconds As Single

    'Disable calculate button on form and reset the leave_calculations
    'flag.

    cmdCalculate.Enabled = False
    leave_calculation = False

    'Read the timer.

    time_begin = Timer

    'Call function to perform the actual calculation.

    Call PopCalc

    'If the form is still visible, ie it wasn't closed before the
    'calculation was complete, then perform the following actions.

    If (frmCalculate.Visible) Then
```

```
        'Read the timer.

        time_end = Timer

        'Convert the time into standard format.

        If time_end < time_begin Then
            total_time = time_end - time_begin + seconds_in_a_day
        Else
            total_time = time_end - time_begin
        End If

        hours = total_time \ 3600
        minutes = (total_time \ 60) - (hours * 60)
        seconds = (total_time - ((hours * 60 + minutes) * 60))

        'Display time to user.

        frmCalculate.txtTime.text = "Total processing time =" + Str$(hours) + ":" +
LTrim$(Str$(minutes)) + ":" + Format$(seconds, "0.00")

        'Enable calculate button.

        cmdCalculate.Enabled = True

    End If

    cmdCalculate.Enabled = True
    frmRosette.blnInputDataChanged = False

End Sub

Private Sub cmdClose_Click()

    'This procedure allows the user to exit a calculation
    'before it has been completed, and it allows the
    'user to close the calculate window when a
    'calculation is finished.

    Dim reply As Integer

    'If a calculation is not in progress then unload the calculate form.

    If frmCalculate.cmdCalculate.Enabled = True Then

        Unload frmCalculate

    Else

        'If the calculation is in progress ask the user if they really
        'want to leave.
'TODO
'        reply = MsgBox("Do you really want to stop the present calculation?", _
'                       "MB_YESNO + MB_DEFBUTTON1", "Close")
        reply = MsgBox("Do you really want to stop the present calculation?", _
                       vbYesNo + vbDefaultButton1, "Close")

        If (reply = vbYes) Then

            Unload frmCalculate
            leave_calculation = True

        End If

    End If
```

```
    frmRosette.blnInputDataChanged = False

End Sub

Private Sub Form_Load()

    'Initialize status and time lables.

    frmCalculate.txtStatus.text = "Press Calculate to start."
    frmCalculate.txtTime.text = ""

    'Enable the Calculate button and reset the leave calculation flag.

    frmCalculate.cmdCalculate.Enabled = True
    leave_calculation = False

End Sub


VERSION 5.00
Begin VB.Form frmChooseDir
   Caption         =   "Choose Directory"
   ClientHeight    =   6900
   ClientLeft      =   60
   ClientTop       =   345
   ClientWidth     =   9615
   ControlBox      =   0   'False
   Icon            =   "frmChooseDir.frx":0000
   LinkTopic       =   "Form1"
   ScaleHeight     =   6900
   ScaleWidth      =   9615
   StartUpPosition =   2  'CenterScreen
   Begin VB.CommandButton cmdCancel
      Appearance      =   0  'Flat
      BackColor       =   &H00808000&
      Caption         =   "Cancel"
      Height          =   480
      Left            =   5805
      MaskColor       =   &H8000000F&
      TabIndex        =   7
      Top             =   6120
      Width           =   1728
   End
   Begin VB.CommandButton cmdOK
      Appearance      =   0  'Flat
      BackColor       =   &H00808000&
      Caption         =   "OK"
      Height          =   480
      Left            =   7740
      MaskColor       =   &H8000000F&
      TabIndex        =   6
      Top             =   6120
      Width           =   1728
   End
   Begin VB.FileListBox File1
      Height          =   3210
      Left            =   4965
      TabIndex        =   5
      Top             =   2175
      Width           =   4500
   End
   Begin VB.DriveListBox Drive1
      Height          =   315
      Left            =   300
      TabIndex        =   1
      Top             =   1215
```

```
      Width           =   4485
   End
   Begin VB.DirListBox Dir1
      Height          =   3240
      Left            =   285
      TabIndex        =   0
      Top             =   2130
      Width           =   4500
   End
   Begin VB.Label lblTitle
      Caption         =   "Label1"
      Height          =   465
      Left            =   165
      TabIndex        =   4
      Top             =   150
      Width           =   5625
   End
   Begin VB.Label lblChooseFolder
      Caption         =   "Double click to choose folder:"
      Height          =   255
      Left            =   330
      TabIndex        =   3
      Top             =   1815
      Width           =   3135
   End
   Begin VB.Label lblChooseDrive
      Caption         =   "Choose drive:"
      Height          =   225
      Left            =   300
      TabIndex        =   2
      Top             =   855
      Width           =   3135
   End
End
```

H.24 Visual Basic Form Module: frmChooseDir.frm

```
Attribute VB_Name = "frmChooseDir"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
'*******************************************************************************
'  Form:    frmChooseDir
'
'  Project: SECPOP2000
'
'  Description: This module is called from various places on the Setup form
'               and allows the user to choose a directory.  The chosen
'               directory is stored in the tag property of the Setup form to
'               pass the directory name back to the Setup form.
'
'  Created:  6/2003, ss
'
'*******************************************************************************

Option Explicit

Private Sub cmdCancel_Click()

    Unload Me

End Sub

Private Sub cmdOK_Click()

    frmSetup.Tag = Dir1.Path
    Unload Me

End Sub

Private Sub Dir1_Change()

    On Error Resume Next
    File1.Path = Dir1.Path

End Sub

Private Sub Drive1_Change()

    On Error Resume Next
    Dir1.Path = Drive1.Drive
    Drive1.Refresh
    Dir1.Refresh

End Sub


Private Sub Form_Load()

    lblTitle = frmSetup.FormLabel

    If frmSetup.Tag <> "" Then
        Drive1.Drive = frmSetup.Tag
        Dir1.Path = frmSetup.Tag
    End If

End Sub
```

H-25 Visual Basic Form Module: Main.frm

```
VERSION 5.00
Object = "{F9043C88-F6F2-101A-A3C9-08002B2F49FB}#1.2#0"; "COMDLG32.OCX"
Object = "{831FDD16-0C5C-11D2-A9FC-0000F8754DA1}#2.0#0"; "MSCOMCTL.OCX"
Begin VB.MDIForm frmMain
   Appearance      =   0  'Flat
   AutoShowChildren=   0   'False
   BackColor       =   &H80000003&
   Caption         =   "SECPOP2000 - SECtor POPulation and Economic Estimator - Version 3.11"
   ClientHeight    =   8310
   ClientLeft      =   -135
   ClientTop       =   60
   ClientWidth     =   11880
   Icon            =   "Main.frx":0000
   StartUpPosition =   2  'CenterScreen
   Begin MSComctlLib.ImageList ImageList1
      Left            =   1005
      Top             =   1845
      _ExtentX        =   1005
      _ExtentY        =   1005
      BackColor       =   -2147483643
      ImageWidth      =   32
      ImageHeight     =   32
      MaskColor       =   12632256
      _Version        =   393216
      BeginProperty Images {2C247F25-8591-11D1-B16A-00C0F0283628}
         NumListImages   =   8
         BeginProperty ListImage1 {2C247F27-8591-11D1-B16A-00C0F0283628}
            Picture         =   "Main.frx":0CCA
            Key             =   ""
         EndProperty
         BeginProperty ListImage2 {2C247F27-8591-11D1-B16A-00C0F0283628}
            Picture         =   "Main.frx":1BA4
            Key             =   ""
         EndProperty
         BeginProperty ListImage3 {2C247F27-8591-11D1-B16A-00C0F0283628}
            Picture         =   "Main.frx":2A7E
            Key             =   ""
         EndProperty
         BeginProperty ListImage4 {2C247F27-8591-11D1-B16A-00C0F0283628}
            Picture         =   "Main.frx":3758
            Key             =   ""
         EndProperty
         BeginProperty ListImage5 {2C247F27-8591-11D1-B16A-00C0F0283628}
            Picture         =   "Main.frx":4632
            Key             =   ""
         EndProperty
         BeginProperty ListImage6 {2C247F27-8591-11D1-B16A-00C0F0283628}
            Picture         =   "Main.frx":550C
            Key             =   ""
         EndProperty
         BeginProperty ListImage7 {2C247F27-8591-11D1-B16A-00C0F0283628}
            Picture         =   "Main.frx":63E6
            Key             =   ""
         EndProperty
         BeginProperty ListImage8 {2C247F27-8591-11D1-B16A-00C0F0283628}
            Picture         =   "Main.frx":6700
            Key             =   ""
         EndProperty
      EndProperty
   End
   Begin MSComctlLib.Toolbar Toolbar1
      Align           =   1  'Align Top
```

```
    Height          =   660
    Left            =   0
    TabIndex        =   0
    Top             =   0
    Width           =   11880
    _ExtentX        =   20955
    _ExtentY        =   1164
    ButtonWidth     =   1032
    ButtonHeight    =   1005
    Appearance      =   1
    ImageList       =   "ImageList1"
    _Version        =   393216
    BeginProperty Buttons {66833FE8-8583-11D1-B16A-00C0F0283628}
       NumButtons       =   8
       BeginProperty Button1 {66833FEA-8583-11D1-B16A-00C0F0283628}
          Key              =   "Site"
          Object.ToolTipText     =   "Open a site file"
          ImageIndex       =   1
       EndProperty
       BeginProperty Button2 {66833FEA-8583-11D1-B16A-00C0F0283628}
          Key              =   "Problem"
          Object.ToolTipText     =   "Open a problem file"
          ImageIndex       =   2
       EndProperty
       BeginProperty Button3 {66833FEA-8583-11D1-B16A-00C0F0283628}
          Key              =   "CalculateSite"
          Object.ToolTipText     =   "Run the site calculation"
          ImageIndex       =   3
       EndProperty
       BeginProperty Button4 {66833FEA-8583-11D1-B16A-00C0F0283628}
          Key              =   "PopulationRosette"
          Object.ToolTipText     =   "Show the population rosette"
          ImageIndex       =   4
       EndProperty
       BeginProperty Button5 {66833FEA-8583-11D1-B16A-00C0F0283628}
          Key              =   "PopulationTable"
          Object.ToolTipText     =   "Show the population table"
          ImageIndex       =   5
       EndProperty
       BeginProperty Button6 {66833FEA-8583-11D1-B16A-00C0F0283628}
          Key              =   "CumulativeTable"
          Object.ToolTipText     =   "Show the cumulative population table"
          ImageIndex       =   6
       EndProperty
       BeginProperty Button7 {66833FEA-8583-11D1-B16A-00C0F0283628}
          Key              =   "Save"
          Object.ToolTipText     =   "Save to a MACCS file"
          ImageIndex       =   7
       EndProperty
       BeginProperty Button8 {66833FEA-8583-11D1-B16A-00C0F0283628}
          Key              =   "Setup"
          Object.ToolTipText     =   "Modify Setup"
          ImageIndex       =   8
       EndProperty
    EndProperty
End
Begin MSComDlg.CommonDialog dlgSecpop
    Left            =   645
    Top             =   1200
    _ExtentX        =   688
    _ExtentY        =   688
    _Version        =   393216
    CancelError     =   -1  'True
    DialogTitle     =   "Open Site"
    FilterIndex     =   1
End
```

```
Begin VB.Menu mnuFile
   Caption         =   "&File"
   Begin VB.Menu mnuFileExit
      Caption         =   "E&xit"
   End
End
Begin VB.Menu mnuSiteName
   Caption         =   "&Site"
   Begin VB.Menu mnuNew_Sitecom
      Caption         =   "&New Site"
   End
   Begin VB.Menu mnuOpen_Sitecom
      Caption         =   "&Open Site"
   End
   Begin VB.Menu mnuSave_Sitecom
      Caption         =   "&Save Site"
   End
   Begin VB.Menu mnuSave_as_Sitecom
      Caption         =   "Save Site &As"
   End
End
Begin VB.Menu mnuProblemName
   Caption         =   "&Problem"
   Begin VB.Menu mnuNew_ProblemCom
      Caption         =   "&New Problem"
   End
   Begin VB.Menu mnuOpen_ProblemCom
      Caption         =   "&Open Problem"
   End
   Begin VB.Menu mnuSave_ProblemCom
      Caption         =   "&Save Problem"
   End
   Begin VB.Menu mnuSave_As_ProblemCom
      Caption         =   "Save Problem &As"
   End
End
Begin VB.Menu mnuCalculateName
   Caption         =   "&Calculate"
   Begin VB.Menu mnuSite_SpecificCom
      Caption         =   "&Site Specific"
   End
   Begin VB.Menu mnuRegionalCom
      Caption         =   "&Regional"
      Enabled         =   0   'False
   End
End
Begin VB.Menu mnuOutputName
   Caption         =   "&Results"
   Begin VB.Menu mnuRosetteCom
      Caption         =   "&Population Rosette"
   End
   Begin VB.Menu mnuTableCom
      Caption         =   "Population &Tables"
      Begin VB.Menu mnuTable_1com
         Caption         =   "&Population"
      End
      Begin VB.Menu mnuTable_2com
         Caption         =   "&Cumulative"
      End
   End
   Begin VB.Menu mnuMACCScom
      Caption         =   "&MACCS-Input File"
      Begin VB.Menu mnuMACCS_Display_Com
         Caption         =   "&Display"
      End
      Begin VB.Menu mnuMACCS_Print_Com
```

```
            Caption         =   "&Print"
         End
         Begin VB.Menu mnuSave_ResultsMACCScom
            Caption         =   "&Save"
         End
      End
      Begin VB.Menu mnuCSVcom
         Caption         =   "CS&V File"
         Begin VB.Menu mnuCSV_Display_Com
            Caption         =   "&Display"
         End
         Begin VB.Menu mnuCSV_Print_Com
            Caption         =   "&Print"
         End
         Begin VB.Menu mnuSave_ResultsCSVcom
            Caption         =   "&Save"
         End
      End
      Begin VB.Menu mnuRegional_Displaycom
         Caption         =   "&Regional"
         Enabled         =   0   'False
      End
      Begin VB.Menu mnuPrevioustom
         Caption         =   "&Load Previous Results"
      End
   End
   Begin VB.Menu mnuSetupcom
      Caption         =   "Set&up"
   End
   Begin VB.Menu mnuExitname
      Caption         =   "E&xit"
      Visible         =   0   'False
      Begin VB.Menu mnuDos_Shellcom
         Caption         =   "&DOS Shell"
         Enabled         =   0   'False
      End
      Begin VB.Menu mnuExitcom
         Caption         =   "E&xit"
         Visible         =   0   'False
      End
   End
   Begin VB.Menu mnuAbout
      Caption         =   "About"
   End
End
Attribute VB_Name = "frmMain"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
'*****************************************************************************
'  Module:  CalcCode
'
'  Project: SECPOP
'
'  Description: Main Window
'
'  Modification History
'  Date      By   Description
'  --------  ---  ------------------------------------------------------------
'  20020131  cwm  Updated MACCS output routines
'  20020204  cwm  Removed DOS Shell Exit call
'  20020213  cwm  Updated MACCS filesave routine
'
'*****************************************************************************
```

```
Option Explicit

Private Sub MDIForm_Load()

'     'Define varibles local to this event procedure.
'
'     Dim i As Integer, filenum As Integer, TempString As String
'
'
'     'Set up error handler for errors in the configuration
'     'file, and load in the various parameters from the
'     'setup file into the setup form.  These parameters
'     'are set by the user in the exit menu under setup.
'     'Because input will not interact directly with forms,
'     'tempstring is used as a buffer for file data.
'
'     'Load in the forms that we need to keep permanently in memory.
'
'     Load frmProblem_Data
'     frmMain.mnuSave_ProblemCom.Enabled = False
'     frmMain.mnuSave_As_ProblemCom.Enabled = False
'     frmProblem_Data.cmdSave.Enabled = False
'     frmProblem_Data.cmdSave_As.Enabled = False
'
'
'     Load frmSite_Data
'     frmMain.mnuSave_Sitecom.Enabled = False
'     frmMain.mnuSave_as_Sitecom.Enabled = False
'     frmSite_Data.cmdSave.Enabled = False
'     frmSite_Data.cmdSave_As.Enabled = False
'
'
''    Load frmSetup
'
    mnuSave_As_ProblemCom.Enabled = False
    mnuSave_ProblemCom.Enabled = False
    mnuSave_as_Sitecom.Enabled = False
    mnuSave_Sitecom.Enabled = False

    frmDisclaimer.Show

'     On Local Error GoTo configerr
'
'     'Check and see if config file exists.
'     TempString = App.path & "\SECPOP.CFG"
'     If Len(Dir$(TempString)) <> 0 Then
''     If Len(Dir$("SECPOP.CFG")) <> 0 Then
'
'         'Open file and read in the data from it.
'
'         filenum = FreeFile
''         Open "SECPOP.CFG" For Input As filenum
'         Open TempString For Input As filenum
'         Input #filenum, TempString
'         frmSetup.txtSite_Path.text = TempString
'         Input #filenum, TempString
'         frmSetup.txtProblem_path.text = TempString
'         Input #filenum, TempString
'         frmSetup.txtOutput_path.text = TempString
'         Input #filenum, TempString
'         frmSetup.txtData_path.text = TempString
'         Input #filenum, TempString
'         If (TempString = "MACCS") Then
'             frmSetup.optMACCS.Value = True
```

```
'         Else
'             frmSetup.optCSV.Value = True
'         End If
'         Input #filenum, TempString
'         frmSetup.txtCensus_Database.text = TempString
'         Close #filenum
'
'     'else if config file doesn't exist use defaults.
'
'     Else
'
'         'Defaults set in form.
'
'     End If
'
'     'Turn off error handler.
'
'     On Local Error GoTo 0
'
'     Exit Sub
'
'configerr:
'
'     'If config file format is screwed up use defaults.
'
'     Close filenum
'     Exit Sub
'
End Sub


Private Sub MDIForm_QueryUnload(Cancel As Integer, UnloadMode As Integer)

    Dim reply

    'Ask the user if they really want to exit.

    reply = MsgBox("Do you really want to exit?", vbYesNo + vbDefaultButton1, "Exit")

    If (reply = vbNo) Then
        Cancel = True
    Else
        Cancel = False
    End If

'    If (reply = vbYes) Then
'        End
'    End If

End Sub

Private Sub MDIForm_Unload(Cancel As Integer)


    Dim i As Integer
    Dim j As Integer

    ' Loop through the forms collection and unload
    ' each form.

''    j = Forms.Count - 1
''    For i = j To 0 Step -1
''       Unload Forms(i)
''    Next
''
End Sub
```

```
Private Sub mnuAbout_Click()
    frmAbout.Show
End Sub

'Private Sub mnuDOS_Shellcom_Click()

'CWM:  This functionality was disabled in the 2002 revision.
'      As a Windows code, calling a DOS Shell provides no benefit
'
'    'Remove all forms from the screen.
'
'    Unload frmDisclaimer
'
'    Screen.Hide
'    Cls
'    LOCATE 1, 1
'
'    'Place shell message on screen.
'
'    Shell "ECHO Type EXIT to return to SECPOP90"
'    LOCATE 1, 1
'
'    'Run the DOS shell.
'
'    Shell
'
'    'Redisplay screen.
'
'    Screen.Show
'
'End Sub

Private Sub mnuExitCom_Click()

'    Dim reply
'
'    'Ask the user if they really want to exit.
'
'    reply = MsgBox("Do you really want to exit?", vbYesNo + vbDefaultButton1, "Exit")
'
'    If (reply = vbYes) Then
'        End
'    End If

End Sub

Private Sub mnuFileExit_Click()

    Unload frmMain

''    mnuExitCom_Click
'    Dim Cancel As Integer
'    Dim UnloadMode As Integer
'    Call MDIForm_QueryUnload(Cancel, UnloadMode)

End Sub

Private Sub mnuMACCS_Display_Com_Click()

    'This function creates a temporary file, and then
    'writes into that file the information for the user
    'to view on the screen.  This file is then sent to a
    'text form, where the user can view it and scroll
    'the various parts of the data.
```

```
    Dim filenum As Integer
    Dim Filename As String
    Dim tempstr As String


    Unload frmDisclaimer

    'check temp file, if it exists, turn readonly off just in case it's on

    Filename = App.path & "\$$TEMP$$.$$$"
    tempstr = Dir$(Filename)
    If tempstr <> "" Then
        SetAttr Filename, vbNormal
    End If

    'Create temporary file
    filenum = FreeFile
    Open Filename For Output As filenum  'Printer.PrintTarget = "$$TEMP$$.$$$"
    Call print_MACCS_input_file(filenum, "MACCS")     'Write the actual data to file.
    Close #filenum
    filenum = FreeFile
    Open Filename For Input As filenum

    'Display information in text form.
    frmOutput.txtOutput.text = ""
    frmOutput.txtOutput.text = Input$(LOF(filenum), filenum)
    frmOutput.caption = "MACCS Site Data File"
    frmOutput.Show
    frmOutput.ZOrder

    Close filenum  'Close and delete temporary file.
    Kill Filename

End Sub

Private Sub mnuCSV_Display_Com_Click()

    'This function creates a temporary file, and then
    'writes into that file the information for the user
    'to view on the screen.  This file is then sent to a
    'text form, where the user can view it and scroll
    'the various parts of the data.

    Dim filenum As Integer
    Dim Filename As String
    Dim tempstr As String

    Unload frmDisclaimer

    'check temp file, if it exists, turn readonly off just in case it's on

    Filename = App.path & "\$$TEMP$$.$$$"
    tempstr = Dir$(Filename)
    If tempstr <> "" Then
        SetAttr Filename, vbNormal
    End If

    'Create temporary file
    filenum = FreeFile
    Open Filename For Output As filenum  'Printer.PrintTarget = "$$TEMP$$.$$$"
    Call print_MACCS_input_file(filenum, "CSV")     'Write the actual data to file.
    Close #filenum
    filenum = FreeFile
    Open Filename For Input As filenum

    'Display information in text form.
```

```
    frmOutput.txtOutput.text = ""
    frmOutput.txtOutput.text = Input$(LOF(filenum), filenum)
    frmOutput.caption = "CSV Site Data File"
    frmOutput.Show
    frmOutput.ZOrder

    Close filenum  'Close and delete temporary file.
    Kill Filename

End Sub


Private Sub mnuMACCS_Print_Com_Click()

'   This event calls a subroutine in module Outpcode that produces
'   a hard copy of the MACCS file

    Call Print_MaCCS_File_To_Printer("MACCS")

End Sub

Private Sub mnuCSV_Print_Com_Click()

'   This event calls a subroutine in module Outpcode that produces
'   a hard copy of the MACCS file

    Call Print_MaCCS_File_To_Printer("CSV")

End Sub

Private Sub mnuSaveCom_Click()
    Dim MaccsOrCSV As String

'   This event calls a subroutine in module Outpcode that produces
'   a hard copy of the MACCS file

    MaccsOrCSV = "MACCS"
    Call SaveResultsToFile(MaccsOrCSV)

End Sub

Private Sub mnuNew_ProblemCom_Click()

    'This function asks the user if they want to review
    'previously entered data or create a new problem.

    Dim i As Integer, reply As Integer

    Unload frmDisclaimer

    'If no problem has already been loaded, drop user
    'directly into the new problem form.

    If frmProblem_Data.caption = "" Then
        Call new_problem

        'Otherwise ask user if they want to review the old
        'data.  If they do, then show it, otherwise call
        'the new problem form.
    Else

        reply = MsgBox("Do you wish to review the previous problem data?", vbYesNoCancel, "New
Problem")
        If reply = vbYes Then
            frmProblem_Data.Show
```

```
        ElseIf reply = vbNo Then
            Call new_problem
        End If
        frmProblem_Data.ZOrder

    End If

'     frmMain.mnuSave_ProblemCom.Enabled = False
'     frmMain.mnuSave_As_ProblemCom.Enabled = False

    frmMain.mnuSave_ProblemCom.Enabled = True
    frmMain.mnuSave_As_ProblemCom.Enabled = True

End Sub

Private Sub mnuNew_SiteCom_Click()

    'This function asks the user if they want to review
    'previous data or create a new site.

    Dim reply As Integer

    Unload frmDisclaimer

    'If no data has been entered, go directly to new site form.

    If frmSite_Data.caption = "" Then

        Call new_site

    Else

        'If data has been entered, ask the user if they want to see it.
        'If yes, then show old data, otherwise go to new site form.

        reply = MsgBox("Do you wish to review the previous site data?", vbYesNoCancel, "New
Site")
        If reply = vbYes Then
            frmSite_Data.Show
        ElseIf reply = vbNo Then
            Call new_site
        End If
        frmSite_Data.ZOrder

    End If

'     frmMain.mnuSave_Sitecom.Enabled = False
'     frmMain.mnuSave_as_Sitecom.Enabled = False

    frmMain.mnuSave_Sitecom.Enabled = True
    frmMain.mnuSave_as_Sitecom.Enabled = True

End Sub

Private Sub mnuOpen_ProblemCom_Click()

    'This subroutine allows the user to look at a previously
    'created problem data or review problem data that has already
    'been loaded.

    Dim reply As Integer
    Dim tempstr As Integer

    Unload frmDisclaimer

    'If data has not yet been entered, go directly to the
```

```
    'open problem function.

    If frmProblem_Data.caption = "" Then

        Call open_problem

    'Otherwise, ask the user if they want to review the
    'previous problem data.  If so, show it to them, otherwise
    'call open problem.

    Else

        reply = MsgBox("Do you wish to review the previous problem data?", vbYesNoCancel, "Open
Problem File")
        If reply = vbYes Then
            frmProblem_Data.Show
            frmProblem_Data.ZOrder
        ElseIf reply = vbNo Then
            Call open_problem
        End If

    End If

'     frmMain.mnuSave_ProblemCom.Enabled = False
'     frmMain.mnuSave_As_ProblemCom.Enabled = False

    frmMain.mnuSave_ProblemCom.Enabled = True
    frmMain.mnuSave_As_ProblemCom.Enabled = True

End Sub

Private Sub mnuOpen_SiteCom_Click()

    'This subroutine allows the user to look at a previously
    'created site data or review site data that has already
    'been loaded.

    Dim reply As Integer

    Unload frmDisclaimer

    'If data has not yet been entered, go directly to the
    'open site function.

    If frmSite_Data.caption = "" Then

        Call open_site

    'Otherwise, ask the user if they want to review the
    'previous site data.  If so, show it to them, otherwise
    'call open site.

    Else

        reply = MsgBox("Do you wish to review the previous site data?", vbYesNoCancel, "Open Site
File")
        If reply = vbYes Then
            frmSite_Data.Show
        ElseIf reply = vbNo Then
            Call open_site
        End If
        frmSite_Data.ZOrder

    End If

'     frmMain.mnuSave_Sitecom.Enabled = False
```

```
'     frmMain.mnuSave_as_Sitecom.Enabled = False

     frmMain.mnuSave_Sitecom.Enabled = True
     frmMain.mnuSave_as_Sitecom.Enabled = True

End Sub

Private Sub mnuPreviouscom_Click()

    'Load in previously saved output data so that the user
     'can view it again, in any of the standard forms.

     Unload frmDisclaimer
     Call ReadPreviousOutput  'Readfile Module
     Unload frmRosette

End Sub

'Private Sub mnuRegionalCom_Click()
'
'     'Call the forms and functions which handle the
'     'regional calculations
'
'     Unload frmDisclaimer
'
'     'load form which prompts user for certain input
'     'necessary for regional calculations
'
'     Load frmMake_a_Circle
'     frmMake_a_Circle.Show
'
'End Sub

'Private Sub mnuRegional_Displaycom_Click()
'
'     Unload frmDisclaimer
'
'     'Display regional calculation results.
'
'     RUN "run_mppr.exe"
'
'End Sub


Private Sub mnuRosetteCom_Click()

    Unload frmDisclaimer

    'Draw the Rosette graph.

    frmRosette.Show
    frmRosette.ZOrder

End Sub

Private Sub mnuSave_As_ProblemCom_Click()

    'This event procedure calls the save as problem subroutine
    'which saves the problem data with a new file name.

    Unload frmDisclaimer

'     frmMain.mnuSave_ProblemCom.Enabled = False
'     frmMain.mnuSave_As_ProblemCom.Enabled = False

    Call save_as_problem
```

```
End Sub

Private Sub mnuSave_As_SiteCom_Click()

    'This subroutine calls the save as site subroutine which saves
    'the site data with a new file name.

    Unload frmDisclaimer

    Call save_as_site

End Sub

Private Sub mnuSave_ProblemCom_Click()

    'This subroutine calls the save problem subroutine
    'which saves the problem data.

    Unload frmDisclaimer

'    frmMain.mnuSave_ProblemCom.Enabled = False
'    frmMain.mnuSave_As_ProblemCom.Enabled = False

'    Call save_problem

    frmProblem_Data.cmdSave_Click

End Sub

'Private Sub mnuSave_Resultscom_Click()
'
'    'This event procedure calls the sub SaveResultsToFile
'    '(Outpcode module).  Depending on the default site
'    'specific output file format, the file is either saved as a MACCS
'    'file (*.inp) or a comma separated variables file (*.csv).  User
'    'also has the option of changing format during the save process.
'
'    Call SaveResultsToFile
'
'End Sub

Private Sub mnuSave_ResultsMACCScom_Click()

    'This event procedure calls the sub SaveResultsToFile
    '(Outpcode module).  Depending on the default site
    'specific output file format, the file is either saved as a MACCS
    'file (*.inp) or a comma separated variables file (*.csv).  User
    'also has the option of changing format during the save process.

    Call SaveResultsToFile("MACCS")

End Sub

Private Sub mnuSave_ResultsCSVcom_Click()

    'This event procedure calls the sub SaveResultsToFile
    '(Outpcode module).  Depending on the default site
    'specific output file format, the file is either saved as a MACCS
    'file (*.inp) or a comma separated variables file (*.csv).  User
    'also has the option of changing format during the save process.

    Call SaveResultsToFile("CSV")

End Sub
```

```
Private Sub mnuSave_SiteCom_Click()

     'This subroutine calls the save site subroutine
     'which saves the site data.

'     Dim i As Integer
'
'     i = MsgBox(frmMain.dlgSecpop.Filename & " already exists.  Would you like to overwrite this
file? " & _
'               " (If not please select Save As).", vbYesNoCancel)
'     If i = vbYes Then
'
''         frmMain.mnuSave_Sitecom.Enabled = False
''         frmMain.mnuSave_as_Sitecom.Enabled = False
'
'         Call save_site
'
'     End If

     frmSite_Data.cmdSave_Click

End Sub

Private Sub mnusetupcom_Click()

     'This subroutine shows the setup form.  This
     'form allows the user to set up paths and file
     'saving formats.

     Unload frmDisclaimer

     frmSetup.Show
     frmSetup.ZOrder

End Sub

Private Sub mnuSite_specificCom_Click()

     Unload frmDisclaimer

     'This subroutine checks to see if there is enough data to peform
     'site calculations, and if so, calls the form to do the actual math.
     'For further documentation, see frmCalculate.

     'Verify that there are radial distances and coordinates.

     If ((number_of_radii = 0) Or _
          (Val(frmProblem_Data.txtPopulation_multiplier.text) = 0)) Or _
          frmProblem_Data.txtSite_File_Name.text = "" Then
          MsgBox "Unable to calculate populations." + Chr$(KEY_RETURN) + "Problem Data form is
incomplete.", vbOK, "Calculation Error"
     ElseIf ((Val(frmSite_Data.txtLongitude_Degrees.text) = 0) Or
(Val(frmSite_Data.txtLatitude_Degrees.text) = 0)) Then
          MsgBox "Unable to calculate populations." + Chr$(KEY_RETURN) + "  Site Data form is
incomplete.", vbOK, "Calculation Error"
     Else
          Load frmCalculate
          frmCalculate.Show
          frmCalculate.ZOrder  ' bring form to top
          frmCalculate.cmdCalculate_Click
     End If

End Sub

Private Sub mnuTable_1com_Click()
```

```
    'This event proceedure allows the user to view data
    'in tabular format by showing the first table
    'form.  This form must be loaded and unloaded each
    'time that it is displayed so that the values
    'displayed will be updated. For further documentation
    'see frmTable_1.

    Unload frmDisclaimer

    table_type = 1
'    Load frmTable_1
'    frmTable_1.Show
    Unload frmPopulationTables
'    frmPopulationTables.caption = "Population Table - " & frmSite_Data.txtSite_Name.text

    frmPopulationTables.Show
    frmPopulationTables.ZOrder

End Sub

Private Sub mnuTable_2com_Click()

    'This event proceedure allows the user to view data
    'in tabular format by showing the second table
    'form.  This form must be loaded and unloaded each
    'time that it is displayed so that the values
    'displayed will be updated.  For further documentation
    'see frmTable_1.

    Unload frmDisclaimer

    table_type = 2
'    Load frmTable_1
'    frmTable_1.Show
    Unload frmPopulationTables
'    frmPopulationTables.caption = "Cumulative (By Direction) Population Table - " &
frmSite_Data.txtSite_Name.text
    frmPopulationTables.Show
    frmPopulationTables.ZOrder

End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)

    Select Case Button.Key

        Case "Site"

            mnuOpen_SiteCom_Click

        Case "Problem"

            mnuOpen_ProblemCom_Click

        Case "CalculateSite"

            mnuSite_specificCom_Click

         Case "PopulationRosette"

            mnuRosetteCom_Click

        Case "PopulationTable"

            mnuTable_1com_Click
```

```
        Case "CumulativeTable"
            mnuTable_2com_Click
        Case "Save"
            mnuSaveCom_Click
        Case "Setup"
            mnusetupcom_Click
    End Select
End Sub
```

H.26 Visual Basic Form Module: Outpform.frm

```
VERSION 5.00
Begin VB.Form frmOutput
   Appearance      =   0  'Flat
   BackColor       =   &H80000004&
   BorderStyle     =   1  'Fixed Single
   Caption         =   "MACCS Site Data File"
   ClientHeight    =   5820
   ClientLeft      =   150
   ClientTop       =   570
   ClientWidth     =   10200
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   8.25
      Charset         =   0
      Weight          =   700
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H00000000&
   Icon            =   "Outpform.frx":0000
   MaxButton       =   0   'False
   MDIChild        =   -1  'True
   MinButton       =   0   'False
   PaletteMode     =   1  'UseZOrder
   ScaleHeight     =   5820
   ScaleWidth      =   10200
   Begin VB.TextBox txtOutput
      Appearance      =   0  'Flat
      BackColor       =   &H00FFFFFF&
      BeginProperty Font
         Name            =   "Courier New"
         Size            =   9.75
         Charset         =   0
         Weight          =   700
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H00000000&
      Height          =   4905
      Left            =   180
      MultiLine       =   -1  'True
      ScrollBars      =   3  'Both
      TabIndex        =   1
      Top             =   120
      Width           =   9900
   End
   Begin VB.CommandButton cmdClose
      Appearance      =   0  'Flat
      BackColor       =   &H00808000&
      Cancel          =   -1  'True
      Caption         =   "Close"
      Default         =   -1  'True
      Height          =   480
      Left            =   8895
      TabIndex        =   0
      Top             =   5205
      Width           =   1188
   End
End
Attribute VB_Name = "frmOutput"
```

```
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False

Option Explicit

'Output is displayed on this form.
'it is basically just a text viewer with a close box
'attatched to allow the user to leave.

Private Sub cmdClose_Click()

    'This command allows the user to close the output
    'text box.

    frmOutput.Hide
End Sub
```

H.27 Visual Basic Form Module: PopulationTable.frm

```
VERSION 5.00
Object = "{0D623638-DBA2-11D1-B5DF-0060976089D0}#7.0#0"; "tdbg7.ocx"
Begin VB.Form frmPopulationTables
   Appearance      =   0  'Flat
   AutoRedraw      =   -1  'True
   BackColor       =   &H80000004&
   BorderStyle     =   1  'Fixed Single
   Caption         =   "Population Table"
   ClientHeight    =   5820
   ClientLeft      =   30
   ClientTop       =   570
   ClientWidth     =   9600
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   8.25
      Charset         =   0
      Weight          =   700
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H00000000&
   Icon            =   "PopulationTable.frx":0000
   MaxButton       =   0   'False
   MDIChild        =   -1  'True
   MinButton       =   0   'False
   PaletteMode     =   1  'UseZOrder
   ScaleHeight     =   5820
   ScaleWidth      =   9600
   Tag             =   "5055"
   Begin TrueDBGrid70.TDBGrid TDBGrid1
      Height          =   4110
      Left            =   45
      TabIndex        =   3
      Top             =   930
      Width           =   9495
      _ExtentX        =   16748
      _ExtentY        =   7250
      _LayoutType     =   0
      _RowHeight      =   -2147483647
      _WasPersistedAsPixels=   0
      Columns(0)._VlistStyle=   0
      Columns(0)._MaxComboItems=   5
      Columns(0).DataField=   ""
```

```
        Columns(0)._PropDict=    "_MaxComboItems,516,2;_VlistStyle,514,3"
        Columns(1)._VlistStyle=   0
        Columns(1)._MaxComboItems=   5
        Columns(1).DataField=    ""
        Columns(1)._PropDict=    "_MaxComboItems,516,2;_VlistStyle,514,3"
        Columns.Count    =   2
        Splits(0)._UserFlags=    0
        Splits(0).RecordSelectorWidth=   503
        Splits(0).DividerColor=   13160660
        Splits(0).SpringMode=    0   'False
        Splits(0)._PropDict=    "_ColumnProps,515,0;_UserFlags,518,3"
        Splits(0)._ColumnProps(0)=    "Columns.Count=2"
        Splits(0)._ColumnProps(1)=    "Column(0).Width=3281"
        Splits(0)._ColumnProps(2)=    "Column(0).DividerColor=0"
        Splits(0)._ColumnProps(3)=    "Column(0)._WidthInPix=3175"
        Splits(0)._ColumnProps(4)=    "Column(0).Order=1"
        Splits(0)._ColumnProps(5)=    "Column(1).Width=3281"
        Splits(0)._ColumnProps(6)=    "Column(1).DividerColor=0"
        Splits(0)._ColumnProps(7)=    "Column(1)._WidthInPix=3175"
        Splits(0)._ColumnProps(8)=    "Column(1).Order=2"
        Splits.Count     =   1
        PrintInfos(0)._StateFlags=   3
        PrintInfos(0).Name=    "piInternal 0"
        PrintInfos(0).PageHeaderFont=
"Size=8.25,Charset=0,Weight=700,Underline=0,Italic=0,Strikethrough=0,Name=MS Sans Serif"
        PrintInfos(0).PageFooterFont=
"Size=8.25,Charset=0,Weight=700,Underline=0,Italic=0,Strikethrough=0,Name=MS Sans Serif"
        PrintInfos(0).PageHeaderHeight=   0
        PrintInfos(0).PageFooterHeight=   0
        PrintInfos.Count=   1
        DataMode         =   2
        DefColWidth      =   0
        HeadLines        =   1
        FootLines        =   1
        MultipleLines    =   0
        CellTipsWidth    =   0
        DeadAreaBackColor=   13160660
        RowDividerColor =   13160660
        RowSubDividerColor=   13160660
        DirectionAfterEnter=   1
        MaxRows          =   250000
        _PropDict        =
"_ExtentX,2003,3;_ExtentY,2004,3;_LayoutType,512,2;_RowHeight,16,3;_StyleDefs,513,0;_WasPersisted
AsPixels,516,2"
        _StyleDefs(0)    =   "_StyleRoot:id=0,.parent=-
1,.alignment=3,.valignment=0,.bgcolor=&H80000005&"
        _StyleDefs(1)    =   ":id=0,.fgcolor=&H80000008&,.wraptext=0,.locked=0,.transparentBmp=0"
        _StyleDefs(2)    =
":id=0,.fgpicPosition=0,.bgpicMode=0,.appearance=0,.borderSize=0,.ellipsis=0"
        _StyleDefs(3)    =   ":id=0,.borderColor=&H80000005&,.borderType=0,.bold=-
1,.fontsize=825,.italic=0"
        _StyleDefs(4)    =   ":id=0,.underline=0,.strikethrough=0,.charset=0"
        _StyleDefs(5)    =   ":id=0,.fontname=MS Sans Serif"
        _StyleDefs(6)    =   "Style:id=1,.parent=0,.namedParent=33,.bold=-1,.fontsize=825,.italic=0"
        _StyleDefs(7)    =   ":id=1,.underline=0,.strikethrough=0,.charset=0"
        _StyleDefs(8)    =   ":id=1,.fontname=MS Sans Serif"
        _StyleDefs(9)    =   "CaptionStyle:id=4,.parent=2,.namedParent=37"
        _StyleDefs(10)   =   "HeadingStyle:id=2,.parent=1,.namedParent=34,.bold=-
1,.fontsize=825,.italic=0"
        _StyleDefs(11)   =   ":id=2,.underline=0,.strikethrough=0,.charset=0"
        _StyleDefs(12)   =   ":id=2,.fontname=MS Sans Serif"
        _StyleDefs(13)   =   "FooterStyle:id=3,.parent=1,.namedParent=35,.bold=-
1,.fontsize=825,.italic=0"
        _StyleDefs(14)   =   ":id=3,.underline=0,.strikethrough=0,.charset=0"
        _StyleDefs(15)   =   ":id=3,.fontname=MS Sans Serif"
```

```
      _StyleDefs(16)  =
"InactiveStyle:id=5,.parent=2,.bgcolor=&H8000000F&,.fgcolor=&H80000012&"
      _StyleDefs(17)  =   "SelectedStyle:id=6,.parent=1,.namedParent=36"
      _StyleDefs(18)  =   "EditorStyle:id=7,.parent=1"
      _StyleDefs(19)  =   "HighlightRowStyle:id=8,.parent=1,.namedParent=38"
      _StyleDefs(20)  =   "EvenRowStyle:id=9,.parent=1,.namedParent=39"
      _StyleDefs(21)  =   "OddRowStyle:id=10,.parent=1,.namedParent=40"
      _StyleDefs(22)  =   "RecordSelectorStyle:id=11,.parent=2"
      _StyleDefs(23)  =   "FilterBarStyle:id=12,.parent=1,.namedParent=42"
      _StyleDefs(24)  =   "Splits(0).Style:id=13,.parent=1"
      _StyleDefs(25)  =   "Splits(0).CaptionStyle:id=22,.parent=4"
      _StyleDefs(26)  =   "Splits(0).HeadingStyle:id=14,.parent=2"
      _StyleDefs(27)  =   "Splits(0).FooterStyle:id=15,.parent=3"
      _StyleDefs(28)  =   "Splits(0).InactiveStyle:id=16,.parent=5"
      _StyleDefs(29)  =   "Splits(0).SelectedStyle:id=18,.parent=6"
      _StyleDefs(30)  =   "Splits(0).EditorStyle:id=17,.parent=7"
      _StyleDefs(31)  =   "Splits(0).HighlightRowStyle:id=19,.parent=8"
      _StyleDefs(32)  =   "Splits(0).EvenRowStyle:id=20,.parent=9"
      _StyleDefs(33)  =   "Splits(0).OddRowStyle:id=21,.parent=10"
      _StyleDefs(34)  =   "Splits(0).RecordSelectorStyle:id=23,.parent=11"
      _StyleDefs(35)  =   "Splits(0).FilterBarStyle:id=24,.parent=12"
      _StyleDefs(36)  =   "Splits(0).Columns(0).Style:id=28,.parent=13"
      _StyleDefs(37)  =   "Splits(0).Columns(0).HeadingStyle:id=25,.parent=14"
      _StyleDefs(38)  =   "Splits(0).Columns(0).FooterStyle:id=26,.parent=15"
      _StyleDefs(39)  =   "Splits(0).Columns(0).EditorStyle:id=27,.parent=17"
      _StyleDefs(40)  =   "Splits(0).Columns(1).Style:id=32,.parent=13"
      _StyleDefs(41)  =   "Splits(0).Columns(1).HeadingStyle:id=29,.parent=14"
      _StyleDefs(42)  =   "Splits(0).Columns(1).FooterStyle:id=30,.parent=15"
      _StyleDefs(43)  =   "Splits(0).Columns(1).EditorStyle:id=31,.parent=17"
      _StyleDefs(44)  =   "Named:id=33:Normal"
      _StyleDefs(45)  =   ":id=33,.parent=0"
      _StyleDefs(46)  =   "Named:id=34:Heading"
      _StyleDefs(47)  =
":id=34,.parent=33,.valignment=2,.bgcolor=&H8000000F&,.fgcolor=&H80000012&"
      _StyleDefs(48)  =   ":id=34,.wraptext=-1"
      _StyleDefs(49)  =   "Named:id=35:Footing"
      _StyleDefs(50)  =
":id=35,.parent=33,.valignment=2,.bgcolor=&H8000000F&,.fgcolor=&H80000012&"
      _StyleDefs(51)  =   "Named:id=36:Selected"
      _StyleDefs(52)  =   ":id=36,.parent=33,.bgcolor=&H8000000D&,.fgcolor=&H8000000E&"
      _StyleDefs(53)  =   "Named:id=37:Caption"
      _StyleDefs(54)  =   ":id=37,.parent=34,.alignment=2"
      _StyleDefs(55)  =   "Named:id=38:HighlightRow"
      _StyleDefs(56)  =   ":id=38,.parent=33,.bgcolor=&H8000000D&,.fgcolor=&H8000000E&"
      _StyleDefs(57)  =   "Named:id=39:EvenRow"
      _StyleDefs(58)  =   ":id=39,.parent=33,.bgcolor=&HFFFF00&"
      _StyleDefs(59)  =   "Named:id=40:OddRow"
      _StyleDefs(60)  =   ":id=40,.parent=33"
      _StyleDefs(61)  =   "Named:id=41:RecordSelector"
      _StyleDefs(62)  =   ":id=41,.parent=34"
      _StyleDefs(63)  =   "Named:id=42:FilterBar"
      _StyleDefs(64)  =   ":id=42,.parent=33"
   End
   Begin VB.CommandButton cmdprint
      Appearance      =   0  'Flat
      BackColor       =   &H00808000&
      Caption         =   "&Print"
      Height          =   480
      Left            =   7665
      TabIndex        =   0
      Top             =   5250
      Width           =   1728
   End
   Begin VB.CommandButton cmdexit
      Appearance      =   0  'Flat
      BackColor       =   &H00808000&
```

```
      Caption         =   "&Close"
      Height          =   480
      Left            =   5820
      TabIndex        =   1
      Top             =   5250
      Width           =   1728
   End
   Begin VB.Label lblDistance_Heading
      Alignment       =   2  'Center
      Appearance      =   0  'Flat
      BackColor       =   &H80000004&
      Caption         =   "Radii (Kilometers)"
      BeginProperty Font
         Name            =   "MS Sans Serif"
         Size            =   12
         Charset         =   0
         Weight          =   700
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H00000000&
      Height          =   360
      Left            =   3555
      TabIndex        =   2
      Top             =   435
      Width           =   2295
   End
End
Attribute VB_Name = "frmPopulationTables"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Dim MaxCol As Integer
Dim MaxRow As Integer
Dim GridArray() As String
Dim Sink As New GridClass


'This is the actual table form for table 1.  See rptable_1 for more
'info.

Private Sub cmdExit_Click()

    'Remove table 1 from memory.

'    Unload frmTable_1
    Unload Me

End Sub

Private Sub cmdPrint_Click()

    Call Print_Table_File_to_Printer

End Sub

Private Sub Form_Load()

    Me.Top = 50
    Me.Left = 200

End Sub
```

```
Private Sub Form_Activate()

    Unload frmDisclaimer

    Dim sum As Long
    Dim i As Integer
    Dim j As Integer
    Dim k As Integer

    Dim row As Long, col As Integer
    Dim C As TrueDBGrid70.Column
    Dim Cols As TrueDBGrid70.Columns

    Dim blnFirstTime As Boolean

    blnFirstTime = True


'   check to see if anything (problem or site info) has changed

    If frmRosette.blnInputDataChanged = True Then
        For i = 0 To number_of_segments
            For j = 0 To max_number_of_radii
'            sector_population(number_of_segments, max_number_of_radii)
               sector_population(i, j) = 0
            Next j
        Next i
    End If



    If table_type = 1 Then
        frmPopulationTables.caption = "Population Table - " & frmSite_Data.txtSite_Name.text
    Else
        frmPopulationTables.caption = "Cumulative (By Direction) Population Table - " &
frmSite_Data.txtSite_Name.text
    End If


    If (frmProblem_Data.optKilometers.Value) Then
        lblDistance_Heading.caption = "Radii (Kilometers)"
    Else
        lblDistance_Heading.caption = "Radii (Miles)"
    End If

'    row = CLng(txtIn(0).text)
'    col = CInt(txtIn(1).text)
    col = number_of_radii + 2      ' 35  ' one column
    row = number_of_segments + 2   ' 16

    MaxCol = number_of_radii + 2   ' 35  ' one column
    MaxRow = number_of_segments + 2 ' 16

    Screen.MousePointer = vbHourglass

    ' Initialize storage
    Sink.SetDims row, col

   'fill the grid (gridarray) with directions
    For i = 0 To row - 3
        Sink.Value(i, 0) = directions(i)
    Next i
'    GridArray(0, row - 1) = "Sum"
    Sink.Value(row - 1, 0) = "Sum"
```

```
    'fill the grid (gridarray) with data

    For i = 0 To MaxRow - 3
        sum = 0
'        For j = 1 To MaxCol - 1
        For j = 1 To MaxCol - 2
            sum = sum + sector_population(i + 1, j)
            If (table_type = 1) Then
'                GridArray(j, i) = Str(sector_population(i + 1, j))
                Sink.Value(i, j) = Str(sector_population(i + 1, j))
            Else
'                GridArray(j, i) = Str(sum)
                Sink.Value(i, j) = Str(sum)
            End If
        Next j
'        GridArray(MaxCol - 1, i) = sum
        Sink.Value(i, MaxCol - 1) = Str(sum)
    Next i

    'calculate and fill the radial sum labels

'    For i = 1 To MaxCol - 1
    For i = 1 To MaxCol - 2
        sum = 0
        If (table_type = 1) Then
            For j = 1 To number_of_segments
                sum = sum + sector_population(j, i)
            Next j
        Else
            For j = 1 To number_of_segments
                For k = 1 To i
                    sum = sum + sector_population(j, k)
                Next k
            Next j
        End If

'        GridArray(i, MaxRow - 1) = sum
        Sink.Value(MaxRow - 1, i) = Str(sum)
    Next i


    'calculate and filll the total sum label

    sum = 0

'    For i = 1 To MaxCol - 1
    For i = 1 To MaxCol - 2
        For j = 1 To number_of_segments
            sum = sum + sector_population(j, i)
        Next j
    Next i
'    GridArray(MaxCol - 1, MaxRow - 1) = sum
    Sink.Value(MaxRow - 1, MaxCol - 1) = Str(sum)


    ' Make sure no events are fired for the grid yet
    Sink.Attach Nothing

    Set Cols = TDBGrid1.Columns

    ' Remove all columns
    While Cols.Count
        Cols.Remove 0
    Wend
    TDBGrid1.ReBind
```

```
    ' Add columns to the grid
    For col = 0 To Sink.ColCount - 1
        Set C = Cols.Add(col)
        C.Visible = True
        C.Width = 1000
        C.Locked = True
        If col = 0 Then
            C.caption = "Sectors"
        ElseIf col = Sink.ColCount - 1 Then
            C.caption = "Sum"
            C.Width = 1200
        Else
            C.caption = Str(radial_distance(col))
        End If
    Next col

    If blnFirstTime Then

        blnFirstTime = False

    ' set misc parameters

    TDBGrid1.Columns(0).BackColor = &HC0C0C0
    TDBGrid1.Columns(0).DividerStyle = 8  ' double line dividing col 0 and col 1
    TDBGrid1.RecordSelectors = False

'    ' remove existing splits
'
'    For i = 1 To TDBGrid1.Splits.Count - 1
'        TDBGrid1.Splits.Remove 0
'    Next i
'
    ' set splits

    Dim s As TrueDBGrid70.Split
    Set s = TDBGrid1.Splits.Add(0)
    Set s = TDBGrid1.Splits.Add(0)

    Set Cols = TDBGrid1.Splits(0).Columns  ' sector column
    For i = 1 To MaxCol - 1
        Cols(i).Visible = False
    Next i

    Set Cols = TDBGrid1.Splits(1).Columns  ' data columns
    Cols(0).Visible = False
    Cols(MaxCol - 1).Visible = False

    Set Cols = TDBGrid1.Splits(2).Columns  ' sum column
    For i = 0 To MaxCol - 2
        Cols(i).Visible = False
    Next i

    TDBGrid1.Splits(0).AllowSizing = False
    TDBGrid1.Splits(1).AllowSizing = True
    TDBGrid1.Splits(2).AllowSizing = True


    TDBGrid1.Splits(0).SizeMode = dbgNumberOfColumns
    TDBGrid1.Splits(1).SizeMode = dbgNumberOfColumns
    TDBGrid1.Splits(2).SizeMode = dbgNumberOfColumns

    TDBGrid1.Splits(0).Size = 1
'    TDBGrid1.Splits(1).Size = MaxCol - 2
    TDBGrid1.Splits(1).Size = 7
    TDBGrid1.Splits(2).Size = 1
```

```
    ' Calibrate the VScroll bar
    TDBGrid1.ApproxCount = Sink.RowCount

    End If


    ' Initialise the class
    Sink.Attach TDBGrid1

    ' Reinitialize the grid
    TDBGrid1.Bookmark = Null
    TDBGrid1.ReBind


    Screen.MousePointer = vbDefault


    ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'    MaxCol = number_of_radii + 2      ' 35  ' one column
'    MaxRow = number_of_segments + 2   ' 16
'
''    region table:       maxcol = number_of_radii + 1  (10)
''                        maxrow = 16
''    gridarray(maxcol10,maxrow16)
'
''   make gridarray with 1 dummy column to overcome apparent bug
''    ReDim gridarray(0 To maxcol + 1, 0 To maxrow) ' need 1 more col and 2 more rows for sums
'    ReDim GridArray(0 To MaxCol, 0 To MaxRow) ' need 1 more col and 2 more rows for sums
'
'    'fill the grid (gridarray) with directions
'    For i = 0 To MaxRow - 3
'        GridArray(0, i) = directions(i)
'    Next i
'    GridArray(0, MaxRow - 1) = "Sum"
'
'    For i = 0 To 18
'        sum = 0
'        For j = 1 To 12
'                GridArray(j, i) = "hi"
'        Next j
'    Next i
'
'
'    'fill the grid (gridarray) with data
'
'    For i = 0 To MaxRow - 3
'        sum = 0
'        For j = 1 To MaxCol - 1
'            sum = sum + sector_population(i + 1, j)
'            If (table_type = 1) Then
'                GridArray(j, i) = Str(sector_population(i + 1, j))
'            Else
'                GridArray(j, i) = Str(sum)
'            End If
'        Next j
'        GridArray(MaxCol - 1, i) = sum
'    Next i
'
'    'calculate and fill the radial sum labels
'
'    For i = 1 To MaxCol - 1
'        sum = 0
'        If (table_type = 1) Then
'            For j = 1 To number_of_segments
```

```
'                sum = sum + sector_population(j, i)
'            Next j
'        Else
'            For j = 1 To number_of_segments
'                For k = 1 To i
'                    sum = sum + sector_population(j, k)
'                Next k
'            Next j
'        End If
'
'        GridArray(i, MaxRow - 1) = sum
'    Next i
'
'
'    'calculate and filll the total sum label
'
'    sum = 0
'
'    For i = 1 To MaxCol - 1
'        For j = 1 To number_of_segments
'            sum = sum + sector_population(j, i)
'        Next j
'    Next i
'    GridArray(MaxCol - 1, MaxRow - 1) = sum
'
'    'set up grid
'
'    'remove existing columns
'
'    While TDBGrid1.Columns.Count <> 0
'        TDBGrid1.Columns.Remove 0
'    Wend
'
'    'add new columns
'
'    Dim c As TrueOleDBGrid70.Column
'
'    For i = 0 To MaxCol - 2
'         Set c = TDBGrid1.Columns.Add(i)
'         c.caption = Str(radial_distance(i))
'         c.Width = 1000
'         c.Visible = True
'         c.Locked = True
''         TDBGrid1.Columns(i).Locked = True
'    Next i
'    Set c = TDBGrid1.Columns.Add(MaxCol - 1)
'    c.caption = "Sum"
'    c.Visible = True
'    c.Width = 1000
'    c.Locked = True
''    TDBGrid1.Columns(maxcol - 1).Locked = True
'
'    Set c = TDBGrid1.Columns.Add(MaxCol)
'    c.caption = "dummy"
'    c.Width = 1000
'    c.Visible = True
'    c.Locked = True
''    TDBGrid1.Columns(maxcol - 1).Locked = True
'
'    'set various parameters of the grid
'
'    TDBGrid1.ApproxCount = 18
'
'    TDBGrid1.Columns(0).caption = "Sectors"
'    TDBGrid1.Columns(0).DividerStyle = 8  ' double line dividing col 0 and col 1
'
```

```
'     TDBGrid1.Columns(0).Locked = True
'     TDBGrid1.Columns(1).Locked = True
'
''     TDBGrid1.ApproxCount = maxrow
'
'     TDBGrid1.Columns(0).BackColor = &HC0C0C0
''     TDBGrid1.Columns(1).ForeColor = &H808080
'
'     'Initialize current cell position to upper left corner:
'
'     TDBGrid1.row = 0
'     TDBGrid1.col = 0
'     TDBGrid1.Refresh
'
'
'     Open "gridtest.txt" For Output As 22
End Sub

'Private Sub hsbRadii_Change()
'
'     x_position = hsbRadii.Value
''     Call paint_table
'
'End Sub
'
'Private Sub vsbSectors_Change()
'
'     y_position = vsbSectors.Value
''     Call paint_table
'
'End Sub

'''Private Sub TDBGrid1_UnboundReadData(ByVal RowBuf As TrueOleDBGrid70.RowBuffer, _
'''                                    StartLocation As Variant, _
'''                                    ByVal ReadPriorRows As Boolean)
'''
'''' UnboundReadData is fired by an unbound grid whenever
'''' it requires data for display.  This event will fire
'''' when the grid is first shown, when Refresh or ReBind
'''' is used, when the grid is scrolled, and after a
'''' record in the grid is modified and the user commits
'''' the change by moving off of the current row.  The
'''' grid fetches data in "chunks", and the number of rows
'''' the grid is asking for is given by RowBuf.RowCount.
'''
'''     Call TDBGridUnboundReadData(RowBuf, _
'''                                 StartLocation, _
'''                                 ReadPriorRows, _
'''                                 GridArray(), _
'''                                 MaxRow, _
'''                                 MaxCol)
'''End Sub
'''
'''Private Sub TDBGrid1_UnboundGetRelativeBookmark(StartLocation As Variant, _
'''                                               ByVal offset As Long, _
'''                                               NewLocation As Variant, _
'''                                               ApproximatePosition As Long)
'''
'''' TDBGrid1 calls this routine each time it needs to
'''' reposition itself.  StartLocation is a bookmark
'''' supplied by the grid to indicate the "current"
'''' position -- the row we are moving from.  Offset is
'''' the number of rows we must move from StartLocation
'''' in order to arrive at the desired destination row.
'''' A positive offset means the desired record is after
'''' the StartLocation, and a negative offset means the
```

```
'''' desired record is before StartLocation.
'''
'''     Call TDBGridUnboundGetRelativeBookmark(StartLocation, _
'''                                            offset, _
'''                                            NewLocation, _
'''                                            ApproximatePosition, _
'''                                            MaxRow)

Private Sub Form_Unload(Cancel As Integer)
        Set Sink = Nothing
End Sub

'''
'''End Sub
'''
'''Private Function GetNewBookmark(MaxRow As Long) As Variant
''''Private Function GetNewBookmark(maxrow As Integer) As Variant
'''' GetNewBookmark is called when we need to create a
'''' bookmark for a newly added record (row).
'''
'''' Reserve space for the new row in the array
''''    ReDim Preserve gridarray(0 To maxcol - 1, 0 To maxrow)
'''
'''' Since the row index of the last record is (MaxRow - 1),
'''' the next available index for the row to be added is
'''' (MaxRow), so create a bookmark for the new row using
'''' MaxRow as an index.
'''    GetNewBookmark = MakeBookmark(MaxRow)
'''
'''' Increment the number of rows in our dataset
'''    MaxRow = MaxRow + 1
'''
'''' Calibrate the scroll bars based on the new dataset
'''' size.
'''    TDBGrid1.ApproxCount = MaxRow
'''End Function
'''
'''
Private Sub TDBGrid1_Click()

End Sub
```

H.28 Visual Basic Form Module: Probform.frm

```
VERSION 5.00
Begin VB.Form frmProblem_Data
   Appearance      =   0  'Flat
   BackColor       =   &H80000004&
   BorderStyle     =   1  'Fixed Single
   ClientHeight    =   5820
   ClientLeft      =   2040
   ClientTop       =   2610
   ClientWidth     =   9585
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   8.25
      Charset         =   0
      Weight          =   700
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H00000000&
   Icon            =   "Probform.frx":0000
   MaxButton       =   0   'False
   MDIChild        =   -1  'True
   MinButton       =   0   'False
   PaletteMode     =   1  'UseZOrder
   ScaleHeight     =   5820
   ScaleWidth      =   9585
   Tag             =   "Problem_Data"
   Visible         =   0   'False
   Begin VB.Frame Frame2
      Caption         =   "Pro&blem Remarks:"
      Height          =   1065
      Left            =   660
      TabIndex        =   26
      Top             =   3840
      Width           =   8175
      Begin VB.TextBox txtProblem_Remarks
         BackColor       =   &H80000004&
         ForeColor       =   &H00000000&
         Height          =   600
         Left            =   120
         MultiLine       =   -1  'True
         TabIndex        =   6
         Top             =   330
         Width           =   7920
      End
   End
   Begin VB.Frame Frame5
      BackColor       =   &H80000004&
      Caption         =   "Populaton Multiplier"
      Height          =   1440
      Left            =   630
      TabIndex        =   25
      Top             =   2070
      Width           =   1905
      Begin VB.TextBox txtPopulation_multiplier
         BackColor       =   &H80000004&
         ForeColor       =   &H00000000&
         Height          =   285
         Left            =   630
         TabIndex        =   1
         Top             =   495
         Width           =   735
```

```
   End
End
Begin VB.CommandButton cmdClose
   Appearance      =   0  'Flat
   BackColor       =   &H00808000&
   Cancel          =   -1  'True
   Caption         =   "&Close"
   Height          =   480
   Left            =   2130
   MaskColor       =   &H8000000F&
   TabIndex        =   8
   Tag             =   "Close"
   Top             =   5250
   Width           =   1728
End
Begin VB.CommandButton cmdPrint
   Appearance      =   0  'Flat
   BackColor       =   &H00808000&
   Caption         =   "&Print"
   Height          =   480
   Left            =   7665
   MaskColor       =   &H8000000F&
   TabIndex        =   11
   Top             =   5235
   Width           =   1728
End
Begin VB.CommandButton cmdSave_As
   Appearance      =   0  'Flat
   BackColor       =   &H00FFFFFF&
   Caption         =   "Save &As"
   Height          =   480
   Left            =   5820
   MaskColor       =   &H8000000F&
   TabIndex        =   10
   Top             =   5250
   Width           =   1728
End
Begin VB.CommandButton cmdSave
   Appearance      =   0  'Flat
   BackColor       =   &H00808000&
   Caption         =   "&Save"
   Height          =   480
   Left            =   3975
   MaskColor       =   &H8000000F&
   TabIndex        =   9
   Top             =   5250
   Width           =   1728
End
Begin VB.CommandButton cmdOpen
   Appearance      =   0  'Flat
   BackColor       =   &H00808000&
   Caption         =   "P&roblems"
   Height          =   480
   Left            =   285
   MaskColor       =   &H8000000F&
   TabIndex        =   7
   Tag             =   "Open"
   Top             =   5250
   Width           =   1728
End
Begin VB.Frame fraRadial_Distances
   BackColor       =   &H80000004&
   Caption         =   "Radial &Distances"
   Height          =   1440
   Left            =   2850
   TabIndex        =   15
```

```
   Top             =   2070
   Width           =   2220
   Begin VB.VScrollBar vsbRadius_Number
      Height          =   840
      Left            =   1560
      Max             =   35
      Min             =   1
      TabIndex        =   16
      TabStop         =   0   'False
      Top             =   360
      Value           =   1
      Width           =   225
   End
   Begin VB.TextBox txtRadial_Distance3
      Alignment       =   1  'Right Justify
      Appearance      =   0  'Flat
      BackColor       =   &H80000004&
      BorderStyle     =   0  'None
      ForeColor       =   &H00000000&
      Height          =   285
      Left            =   465
      MultiLine       =   -1  'True
      TabIndex        =   13
      TabStop         =   0   'False
      Top             =   930
      Width           =   990
   End
   Begin VB.TextBox txtRadial_Distance2
      Alignment       =   1  'Right Justify
      Appearance      =   0  'Flat
      BackColor       =   &H80000004&
      BorderStyle     =   0  'None
      ForeColor       =   &H00000000&
      Height          =   285
      Left            =   465
      MultiLine       =   -1  'True
      TabIndex        =   12
      TabStop         =   0   'False
      Top             =   645
      Width           =   990
   End
   Begin VB.TextBox txtRadial_Distance1
      Alignment       =   1  'Right Justify
      Appearance      =   0  'Flat
      BackColor       =   &H80000004&
      BorderStyle     =   0  'None
      ForeColor       =   &H00000000&
      Height          =   285
      Left            =   480
      MultiLine       =   -1  'True
      TabIndex        =   2
      Top             =   360
      Width           =   990
   End
   Begin VB.Label lblRadius3
      Appearance      =   0  'Flat
      BackColor       =   &H80000004&
      Caption         =   " 3"
      ForeColor       =   &H00000000&
      Height          =   240
      Left            =   240
      TabIndex        =   19
      Top             =   952
      Width           =   210
   End
   Begin VB.Label lblRadius2
```

```
      Appearance      =   0  'Flat
      BackColor       =   &H80000004&
      Caption         =   " 2"
      ForeColor       =   &H00000000&
      Height          =   240
      Left            =   240
      TabIndex        =   18
      Top             =   667
      Width           =   210
   End
   Begin VB.Label lblRadius1
      Appearance      =   0  'Flat
      AutoSize        =   -1  'True
      BackColor       =   &H80000004&
      Caption         =   " 1"
      ForeColor       =   &H00000000&
      Height          =   195
      Left            =   240
      TabIndex        =   17
      Top             =   360
      Width           =   180
   End
End
Begin VB.Frame Frame1
   BackColor       =   &H80000004&
   Caption         =   "Radial Units"
   Height          =   1440
   Left            =   5385
   TabIndex        =   14
   Top             =   2070
   Width           =   1320
   Begin VB.OptionButton optKilometers
      Caption         =   "&KM"
      Height          =   240
      Left            =   300
      TabIndex        =   3
      TabStop         =   0   'False
      Top             =   465
      Value           =   -1  'True
      Width           =   756
   End
   Begin VB.OptionButton optMiles
      Caption         =   "&MI"
      Height          =   240
      Left            =   300
      TabIndex        =   4
      Top             =   825
      Width           =   756
   End
End
Begin VB.Frame Frame3
   BackColor       =   &H80000004&
   Caption         =   "Site File &Name:"
   Height          =   1215
   Left            =   675
   TabIndex        =   20
   Top             =   405
   Width           =   8175
   Begin VB.CommandButton cmdBrowse
      Appearance      =   0  'Flat
      BackColor       =   &H00808000&
      Caption         =   "Browse"
      Height          =   480
      Left            =   6675
      TabIndex        =   0
      Tag             =   "Browse"
```

```
      Top               =   600
      Width             =   1245
   End
   Begin VB.TextBox txtSite_File_Name
      Appearance        =   0  'Flat
      BackColor         =   &H80000004&
      BorderStyle       =   0  'None
      ForeColor         =   &H00000000&
      Height            =   480
      Left              =   105
      Locked            =   -1  'True
      MultiLine         =   -1  'True
      TabIndex          =   21
      TabStop           =   0   'False
      Top               =   420
      Width             =   6345
   End
End
Begin VB.Frame Frame4
   BackColor         =   &H80000004&
   Caption           =   "Economic Regions"
   Height            =   1440
   Left              =   7020
   TabIndex          =   22
   Top               =   2070
   Width             =   1815
   Begin VB.CommandButton cmdEdit_regions
      Appearance        =   0  'Flat
      BackColor         =   &H00808000&
      Caption           =   "&Edit Regions"
      Height            =   480
      Left              =   285
      TabIndex          =   5
      Top               =   840
      Width             =   1245
   End
   Begin VB.Label Label1
      Alignment         =   2  'Center
      AutoSize          =   -1  'True
      Caption           =   "Number"
      Height            =   195
      Left              =   120
      TabIndex          =   24
      Top               =   360
      Width             =   675
      WordWrap          =   -1  'True
   End
   Begin VB.Label lblNumber_of_regions
      Alignment         =   2  'Center
      Appearance        =   0  'Flat
      AutoSize          =   -1  'True
      BackColor         =   &H80000000&
      Caption           =   " "
      ForeColor         =   &H00000000&
      Height            =   195
      Left              =   1095
      TabIndex          =   23
      Top               =   360
      Width             =   585
   End
End
End
Attribute VB_Name = "frmProblem_Data"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
```

```
Attribute VB_Exposed = False
'*******************************************************************************
'  Form:  Problem Data
'
'  Project: SECPOP
'
'  Description: Create Problem
'
'
'  Modification History
'  Date       By   Description
'  --------   ---  ------------------------------------------------------------
'  20020213   cwm  added header and updated kilometer/mile conversion
'
'*******************************************************************************

Option Explicit

'This form allows the user to create, edit, and save different problems

Private Sub cmdBrowse_Click()

    'This routine marks the problem form as modified, and if the
    'enter key is pressed, it attempts to open the site file and read in
    'the appropriate site data.

    Dim site_file_name As String

'    frmMain.mnuSave_ProblemCom.Enabled = True
'    frmMain.mnuSave_As_ProblemCom.Enabled = True

    If (InStr(frmProblem_Data.caption, " (Modified)") = 0) Then
        frmProblem_Data.caption = frmProblem_Data.caption + " (Modified)"
    End If

'    If keyascii = KEY_RETURN Then
'        keyascii = 0
        site_file_name = txtSite_File_Name.text
        'open_site_from_problem (site_file_name)
'    End If

    'Call the function which allows the user to see which
    'site files are available.  For more info, see browse_site_file.

    Call browse_site_file

    frmRosette.blnInputDataChanged = True

End Sub

Private Sub cmdClose_Click()

    'This routine allows the user to close the problem
    'window without saving.

'    Dim temp As Integer, filenum As Integer
'    Dim tempstring As String
'
'    'Verify that all of the data entered by the user
'    'is valid.
'    temp = verify_input(1)
'
'    'Remove the problem data form from the screen.
'
'     frmProblem_Data.Hide
```

```
    Unload Me

End Sub


Private Sub cmdEdit_regions_Click()

    If (number_of_radii < 2) Then
        MsgBox "Error: At least two radii must be specified before regions are edited."
        Exit Sub
    ElseIf (InStr(frmProblem_Data.caption, " (Modified)") = 0) Then
        frmProblem_Data.caption = frmProblem_Data.caption + " (Modified)"
'        frmMain.mnuSave_ProblemCom.Enabled = True
'        frmMain.mnuSave_As_ProblemCom.Enabled = True
    End If

    If (number_econ_regions = 0) Or (number_econ_regions > max_econ_regions) Or
(number_econ_regions < min_econ_regions) Then
        Call default_regions
    End If

    frmRegion.Show

End Sub

Private Sub cmdOpen_Click()

    'This routine allows the user to open a previously
    'defined problem from the problem form.  For more
    'info see the open_problem procedure

    Call open_problem

End Sub

Private Sub cmdPrint_Click()

    'This routine allows the user to print the data in
    'a problem file.  For more information
    'see the Print_Problem_File_to_Printer procedure in the Outpcode module.


'    Call print_problem
    Call Print_Problem_File_to_Printer

End Sub

Private Sub cmdSave_As_Click()

    'This routine allows the user to save a file by a
    'new name.  For more information, see the
    'save_as_problem procedure.

'    frmMain.mnuSave_ProblemCom.Enabled = False
'    frmMain.mnuSave_As_ProblemCom.Enabled = False

    Call save_as_problem

End Sub

'Private Sub cmdSave_Click()
Public Sub cmdSave_Click()

    'This routine allows the user to save data entered
    'in a problem form.  For more information,
    'see the save_problem procedure.
```

```
    Dim i As Integer
    Dim tempstr As String

    i = InStr(frmSite_Data.caption, "NEW_PROB")
    If i <> 0 Then
        Call cmdSave_As_Click
    Else

        'Remove the modified specifier from the file name if necessary.

        i = InStr(frmProblem_Data.caption, " (Modified)")
        If i <> 0 Then
            tempstr = Mid$(frmProblem_Data.caption, 1, i - 1)
        Else
            tempstr = frmProblem_Data.caption
        End If
        frmProblem_Data.caption = tempstr

        i = MsgBox(tempstr & " already exists.  Would you like to overwrite this file? " & vbCrLf
& _
                   " (If not please select Save As).", vbYesNoCancel)
        If i = vbYes Then

'        frmMain.mnuSave_ProblemCom.Enabled = False
'        frmMain.mnuSave_As_ProblemCom.Enabled = False

        Call save_problem

        End If
    End If
End Sub

Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)

    'This routine allows the user to close the problem
    'window without saving.

    Dim temp As Integer, filenum As Integer
    Dim TempString As String

    If UnloadMode <> vbFormMDIForm Then  'vbformmdiform:  the mdi parent is closing
        'Verify that all of the data entered by the user
        'is valid.
        temp = verify_input(1)
'        If temp <> 1 Then
            frmProblem_Data.Hide
            Cancel = True
'        End If
    End If

    frmMain.mnuSave_ProblemCom.Enabled = False
    frmMain.mnuSave_As_ProblemCom.Enabled = False

End Sub

Private Sub Form_Unload(Cancel As Integer)

'    'This routine allows the user to close the problem
'    'window without saving.
'
'    Dim temp As Integer, filenum As Integer
'    Dim TempString As String
'
'    'Verify that all of the data entered by the user
'    'is valid.
```

```
'     temp = verify_input(1)
'
'     'Remove the problem data form from the screen.

'      frmProblem_Data.Hide
'      Load frmProblem_Data
'      frmProblem_Data.Show

End Sub

Private Sub optKilometers_Click()

    'This event procedure allows the user to convert miles to kilometers if
    'necessary, and it allows the user to specify which unit measures
    'should be in.

    Dim i As Integer, j As Integer, reply As Integer, TempString As String, tempchar As String

    If (InStr(frmProblem_Data.caption, " (Modified)") = 0) Then
        frmProblem_Data.caption = frmProblem_Data.caption + " (Modified)"
    End If

'     frmMain.mnuSave_ProblemCom.Enabled = True
'     frmMain.mnuSave_As_ProblemCom.Enabled = True


    'First check to see if there are values for Radial Distances

    If txtRadial_Distance1.text = "" Then Exit Sub

    'Ask if a conversion is necessary.

    If (optKilometers.Tag = "") Then       ' tag is not blank when a file is being read in
        reply = MsgBox("Do you wish to convert any existing radial" + Chr$(13) + "    distances
from Miles to Kilometers?", vbYesNoCancel + vbDefaultButton2, "Conversion")
        If reply = vbYes Then

            For i = 1 To number_of_radii
                If (radial_distance(i) < 0.125) And (radial_distance(i) <> 0) Then
                    MsgBox "Error: unable to convert to kilometers distances of less than .125
miles."
                    optMiles.Tag = "DoNotCheck"
                    optMiles.Value = True
                    optMiles.Tag = ""
'                    optMiles.SetFocus
                    Exit Sub
                ElseIf radial_distance(i) > 6213 Then
                    MsgBox "Error: unable to convert to kilometers distances of more than 6213
miles."
                    optMiles.Tag = "DoNotCheck"
                    optMiles.Value = True
                    optMiles.Tag = ""
'                    optMiles.SetFocus
                    Exit Sub
                End If
            Next i

            'If conversion is requested, perform the math.

            For i = 1 To number_of_radii Step 1


                radial_distance(i) = Round(radial_distance(i) _
                                    * miles_to_kilometers, 5)

'''                tempstring = Str$(radial_distance(i) * miles_to_kilometers)
```

```
'''                 tempchar = "0"
'''
'''
'''                 'Examine the converted number, and round it to three digits
'''                 'after the decimal point.
'''
'''                 For j = 1 To (Len(tempstring) - 3)
'''                     If (Mid$(tempstring, j, 1) = ".") Then
'''                         tempchar = Mid$(tempstring, j + 4, 1)
'''                         tempstring = Left$(tempstring, j + 3)
'''                     End If
'''                 Next j
'''
'''             radial_distance(i) = Val(tempstring)

                If (radial_distance(i) >= 9999) Then
                    radial_distance(i) = 9999
                ElseIf (radial_distance(i) <= 0.1) Then
                    radial_distance(i) = 0.1
                End If

'''                 If (Val(tempchar) <= 5) Then radial_distance(i) = radial_distance(i) + 0.001

            Next i

            'Change the radial distances already listed to new unit of
            'measure.

            frmProblem_Data.txtRadial_Distance1.text = Right$("    " +
Format$(radial_distance(vsbRadius_Number.Value), "0.0000;0.0000;      "), 9)
            frmProblem_Data.txtRadial_Distance2.text = Right$("    " +
Format$(radial_distance(vsbRadius_Number.Value + 1), "0.0000;0.0000;      "), 9)
            frmProblem_Data.txtRadial_Distance3.text = Right$("    " +
Format$(radial_distance(vsbRadius_Number.Value + 2), "0.0000;0.0000;      "), 9)

        ElseIf reply = vbNo Then

            For i = 1 To number_of_radii
                If (radial_distance(i) < 0.1) Or (radial_distance(i) > 9999) Then
                    MsgBox "Error: Radial distance out of bounds"
                    optMiles.Tag = "DoNotCheck"
                    optMiles.Value = True
                    optMiles.Tag = ""
'                    optMiles.Tag = "Previously_True"
'                    optMiles.SetFocus
                    Exit Sub
                End If
            Next i

        End If

'''         'If necessary, mark which unit is recognized as being the
'''         'previous unit.
'''
        If reply <> vbCancel Then

'            optKilometers.Tag = "Previously_True"
'            optMiles.Tag = ""

            'Mark form as modified.

'            frmMain.mnuSave_ProblemCom.Enabled = True
'            frmMain.mnuSave_As_ProblemCom.Enabled = True
            If (InStr(frmProblem_Data.caption, " (Modified)") = 0) Then
                frmProblem_Data.caption = frmProblem_Data.caption + " (Modified)"
```

```
            End If

        Else

            'if cancel, then don't change anything on form.

            optMiles.Tag = "Previously_True"
            optMiles.Value = True
            optMiles.Tag = ""
'            optMiles.SetFocus

        End If

    End If


'''    For i = 1 To number_of_radii
'''        If (radial_distance(i) < 0.125) And (radial_distance(i) <> 0) Then
'''            MsgBox "Error: unable to convert to kilometers distances of less than .125 miles."
'''            Exit Sub
'''        ElseIf radial_distance(i) > 6213 Then
'''            MsgBox "Error: unable to convert to kilometers distances of more than 6213 miles."
'''            Exit Sub
'''        End If
'''    Next i
'''
'''    If frmProblem_Data.optKilometers.Tag <> "Previously_True" Then
'''        For i = 1 To number_of_radii Step 1
'''            radial_distance(i) = Round(radial_distance(i) _
'''                                * miles_to_kilometers, 5)
'''
'''            If (radial_distance(i) >= 9999) Then
'''                radial_distance(i) = 9999
'''            ElseIf (radial_distance(i) <= 0.1) Then
'''                radial_distance(i) = 0.1
'''            End If
'''        Next i
'''
'''        'Change the radial distances already listed to new unit of
'''        'measure.
'''
'''        frmProblem_Data.txtRadial_Distance1.text = Right$("    " +
Format$(radial_distance(vsbRadius_Number.Value), "0.0000;0.0000;      "), 9)
'''        frmProblem_Data.txtRadial_Distance2.text = Right$("    " +
Format$(radial_distance(vsbRadius_Number.Value + 1), "0.0000;0.0000;      "), 9)
'''        frmProblem_Data.txtRadial_Distance3.text = Right$("    " +
Format$(radial_distance(vsbRadius_Number.Value + 2), "0.0000;0.0000;      "), 9)
'''
'''    End If
'''
'''    frmProblem_Data.optKilometers.Tag = "Previously_True"
'''    frmProblem_Data.optMiles.Tag = ""
'''
    frmRosette.blnInputDataChanged = True

End Sub

Private Sub optMiles_Click()

    'This event procedure allows the user to convert miles to kilometers if
    'necessary, and it allows the user to specify which unit measures
    'should be in.

'    Dim i As Integer
'    Dim j As Integer
```

```
    Dim i As Integer, j As Integer, reply As Integer, TempString As String, tempchar As String

    If (InStr(frmProblem_Data.caption, " (Modified)") = 0) Then
        frmProblem_Data.caption = frmProblem_Data.caption + " (Modified)"
    End If

'     frmMain.mnuSave_ProblemCom.Enabled = True
'     frmMain.mnuSave_As_ProblemCom.Enabled = True

    'First check to see if there are values for Radial Distances

    If txtRadial_Distance1.text = "" Then Exit Sub



    'Allow the user to convert kilometers to miles or enter data in miles.


    'If user has changed units then proceed.

    If (optMiles.Tag = "") Then    ' tag is not blank when a file is being read in
        reply = MsgBox("Do you wish to convert any existing radial" + Chr$(13) + "    distances
from Kilometers to Miles?", vbYesNoCancel + vbDefaultButton2, "Conversion")

        'If user wishes to convert numbers to new units, then proceed.

        If reply = vbYes Then

            For i = 1 To number_of_radii
                If (radial_distance(i) <= 0.2) And (radial_distance(i) <> 0) Then
                    MsgBox "Error: unable to convert to miles distances of less than .2
kilometers."
                    optKilometers.Tag = "DoNotCheck"
                    optKilometers.Value = True
                    optKilometers.Tag = ""
'                    optKilometers.SetFocus
                    Exit Sub
                ElseIf radial_distance(i) > 9999 Then
                    MsgBox "Error: unable to convert to miles distances of more than 9999
kilometers."
                    optKilometers.Tag = "DoNotCheck"
                    optKilometers.Value = True
                    optKilometers.Tag = ""
'                    optKilometers.SetFocus
                    Exit Sub
                End If
            Next i

            For i = 1 To number_of_radii Step 1

                radial_distance(i) = Round(radial_distance(i) _
                                / miles_to_kilometers, 5)

'''                tempchar = "0"
'''                tempstring = Str$(radial_distance(i) * kilometers_to_miles)
'''
'''                'Analyze converted number and round it to three decimal
'''                'places.
'''
'''                For j = 1 To (Len(tempstring) - 3)
'''                    If (Mid$(tempstring, j, 1) = ".") Then
'''                        tempchar = Mid$(tempstring, j + 4, 1)
'''                        tempstring = Left$(tempstring, j + 3)
'''                    End If
'''                Next j
```

```
'''
'''                radial_distance(i) = Val(tempstring)

                If (radial_distance(i) >= 6212.999) Then
                    radial_distance(i) = 6213
                ElseIf (radial_distance(i) <= 0.07) Then
                    radial_distance(i) = 0.07
                End If

'''                If Val(tempchar) > 4 Then radial_distance(i) = radial_distance(i) + 0.001

            Next i

            'Write new radial distances to problem form.

            frmProblem_Data.txtRadial_Distance1.text = Right$("    " +
Format$(radial_distance(vsbRadius_Number.Value), "0.0000;0.0000;      "), 9)
            frmProblem_Data.txtRadial_Distance2.text = Right$("    " +
Format$(radial_distance(vsbRadius_Number.Value + 1), "0.0000;0.0000;      "), 9)
            frmProblem_Data.txtRadial_Distance3.text = Right$("    " +
Format$(radial_distance(vsbRadius_Number.Value + 2), "0.0000;0.0000;      "), 9)

        ElseIf reply = vbNo Then

            For i = 1 To number_of_radii
                If (radial_distance(i) < 0.07) Or (radial_distance(i) > 6213) Then
                    MsgBox "Error: Radial distance out of bounds"
                    optKilometers.Tag = "DoNotCheck"
                    optKilometers.Value = True
                    optKilometers.Tag = ""
'                    optKilometers.Tag = "Previously_True"
'                    optKilometers.SetFocus
                    Exit Sub
                End If
            Next i

        End If

        'If user has not cancled, update which unit was previously used.

        If reply <> vbCancel Then
'            optMiles.Tag = "Previously_True"
'            optKilometers.Tag = ""

            'Mark form as modified.

'            frmMain.mnuSave_ProblemCom.Enabled = True
'            frmMain.mnuSave_As_ProblemCom.Enabled = True
            If (InStr(frmProblem_Data.caption, " (Modified)") = 0) Then
                frmProblem_Data.caption = frmProblem_Data.caption + " (Modified)"
            End If
        Else

            'If cancel is selected, leave values as they were.

            optKilometers.Tag = "DoNotCheck"
            optKilometers.Value = True
            optKilometers.Tag = ""
'            optKilometers.SetFocus

        End If

    End If

'''    'Current_Units is a public string variable
'''
```

```
'''     For i = 1 To number_of_radii
'''         If (radial_distance(i) < 0.1) And (radial_distance(i) <> 0) Then
'''             MsgBox "Error: unable to convert to miles distances of less than .1 kilometers."
'''             Exit Sub
'''         ElseIf radial_distance(i) > 9999 Then
'''             MsgBox "Error: unable to convert to miles distances of more than 9999 kilometers."
'''             Exit Sub
'''         End If
'''     Next i
'''
'''
'''     If frmProblem_Data.optMiles.Tag <> "Previously_True" Then
'''         For i = 1 To number_of_radii Step 1
'''             radial_distance(i) = Round(radial_distance(i) _
'''                                 / miles_to_kilometers, 5)
'''
'''             If (radial_distance(i) > 6213) Then
'''                 radial_distance(i) = 6213
'''             ElseIf (radial_distance(i) < 0.125) Then
'''                 radial_distance(i) = 0.125
'''             End If
'''         Next i
'''
'''         'Change the radial distances already listed to new unit of
'''         'measure.
'''
'''         frmProblem_Data.txtRadial_Distance1.text = Right$("    " +
Format$(radial_distance(vsbRadius_Number.Value), "0.0000;0.0000;      "), 9)
'''         frmProblem_Data.txtRadial_Distance2.text = Right$("    " +
Format$(radial_distance(vsbRadius_Number.Value + 1), "0.0000;0.0000;      "), 9)
'''         frmProblem_Data.txtRadial_Distance3.text = Right$("    " +
Format$(radial_distance(vsbRadius_Number.Value + 2), "0.0000;0.0000;      "), 9)
'''     End If
'''
'''     frmProblem_Data.optMiles.Tag = "Previously_True"
'''     frmProblem_Data.optKilometers.Tag = ""

    frmRosette.blnInputDataChanged = True


End Sub

'Private Sub optMiles_GotFocus()
'
'     'Allow the user to convert kilometers to miles or enter data in miles.
'
'     Dim i As Integer
'     Dim j As Integer
'
'     'If user has changed units then proceed.
'
'     If (optMiles.Tag <> "Previously_True") Then
'         reply = MsgBox("Do you wish to convert any existing radial" + Chr$(13) + "    distances
from Kilometers to Miles?", vbYesNoCancel + vbDefaultButton2, "Conversion")
'
'         'If user wishes to convert numbers to new units, then proceed.
'
'         If reply = vbYes Then
'
'             For i = 1 To number_of_radii
'                 If (radial_distance(i) <= 0.2) And (radial_distance(i) <> 0) Then
'                     MsgBox "Error: unable to convert to miles distances of less than .2
kilometers."
'                     optKilometers.Value = True
'                     optKilometers.SetFocus
'                     Exit Sub
```

```
'                ElseIf radial_distance(i) > 9999 Then
'                    MsgBox "Error: unable to convert to miles distances of more than 9999
kilometers."
'                    optKilometers.Value = True
'                    optKilometers.SetFocus
'                    Exit Sub
'                End If
'            Next i
'
'            For i = 1 To number_of_radii Step 1
'                tempchar = "0"
'                TempString = Str$(radial_distance(i) * kilometers_to_miles)
'
'                'Analyze converted number and round it to three decimal
'                'places.
'
'                For j = 1 To (Len(TempString) - 3)
'                    If (Mid$(TempString, j, 1) = ".") Then
'                        tempchar = Mid$(TempString, j + 4, 1)
'                        TempString = Left$(TempString, j + 3)
'                    End If
'                Next j
'
'                radial_distance(i) = Val(TempString)
'
'                If (radial_distance(i) >= 6212.999) Then
'                    radial_distance(i) = 6213
'                ElseIf (radial_distance(i) <= 0.07) Then
'                    radial_distance(i) = 0.07
'                End If
'
'                If Val(tempchar) > 4 Then radial_distance(i) = radial_distance(i) + 0.001
'            Next i
'
'            'Write new radial distances to problem form.
'
'            frmProblem_Data.txtRadial_Distance1.text = Right$("    " +
Format$(radial_distance(vsbRadius_Number.Value), "0.0000;0.0000;      "), 9)
'            frmProblem_Data.txtRadial_Distance2.text = Right$("    " +
Format$(radial_distance(vsbRadius_Number.Value + 1), "0.0000;0.0000;      "), 9)
'            frmProblem_Data.txtRadial_Distance3.text = Right$("    " +
Format$(radial_distance(vsbRadius_Number.Value + 2), "0.0000;0.0000;      "), 9)
'
'        ElseIf reply = vbNo Then
'            For i = 1 To number_of_radii
'                If (radial_distance(i) < 0.07) Or (radial_distance(i) > 6213) Then
'                    MsgBox "Error: Radial distance out of bounds"
'                    optKilometers.Value = True
'                    optKilometers.Tag = "Previously_True"
'                    optKilometers.SetFocus
'                    Exit Sub
'                End If
'            Next i
'        End If
'
'        'If user has not cancled, update which unit was previously used.
'
'        If reply <> vbCancel Then
'            optMiles.Tag = "Previously_True"
'            optKilometers.Tag = ""
'
'            'Mark form as modified.
'
'            frmMain.mnuSave_ProblemCom.Enabled = True
'            frmMain.mnuSave_As_ProblemCom.Enabled = True
'            If (InStr(frmProblem_Data.caption, " (Modified)") = 0) Then
```

```
'                frmProblem_Data.caption = frmProblem_Data.caption + " (Modified)"
'            End If
'        Else
'
'            'If cancel is selected, leav values as they were.
'
'            optKilometers.Value = True
'            optKilometers.SetFocus
'
'        End If
'
'    End If
'
'End Sub

Private Sub txtPopulation_multiplier_KeyPress(keyascii As Integer)

    'This routine marks a problem form as modifed if keys are pressed in
    'the population multiplier field.

'    frmMain.mnuSave_ProblemCom.Enabled = True
'    frmMain.mnuSave_As_ProblemCom.Enabled = True

    If (InStr(frmProblem_Data.caption, " (Modified)") = 0) Then
        frmProblem_Data.caption = frmProblem_Data.caption + " (Modified)"
    End If

    frmRosette.blnInputDataChanged = True

End Sub

Private Sub txtPopulation_multiplier_Validate(Cancel As Boolean)

    If Val(txtPopulation_multiplier.text) > 100 Then
        MsgBox "Please enter a value for Population Multiplier that is less " & _
               "than or equal to 100"
        txtPopulation_multiplier.text = ""
        Cancel = True
    End If


End Sub

Private Sub txtProblem_Remarks_KeyPress(keyascii As Integer)

    'This routine marks a problem as modified if keys are pressed, and
    'deals with the enter key being pressed.

    'If enter key is pressed, display message.

    If keyascii = KEY_RETURN Then

        'Erase keystroke.

        keyascii = 0
        MsgBox "Sorry, no hard returns allowed." + Chr$(13) + "Words will wrap automatically.",
vbOK, "Error"

    'Mark problem form as modified.

    Else

'        frmMain.mnuSave_ProblemCom.Enabled = True
'        frmMain.mnuSave_As_ProblemCom.Enabled = True

        If (InStr(frmProblem_Data.caption, " (Modified)") = 0) Then
```

```
            frmProblem_Data.caption = frmProblem_Data.caption + " (Modified)"
        End If

    End If

End Sub

Private Sub txtProblem_Title_KeyPress(keyascii As Integer)

    'This routine marks the problem form as modified if a key is pressed
    'within the title field.  It also deals with the user pressing enter.

    'If the user presses the enter key, display warning message.

    If keyascii = KEY_RETURN Then

    'Erase keystroke.

        keyascii = 0
        MsgBox "Sorry, no hard returns allowed." + Chr$(13) + "Words will wrap automatically.",
vbOK, "Error"

    Else

    'Mark problem form as modified.

'         frmMain.mnuSave_ProblemCom.Enabled = True
'         frmMain.mnuSave_As_ProblemCom.Enabled = True

        If (InStr(frmProblem_Data.caption, " (Modified)") = 0) Then
            frmProblem_Data.caption = frmProblem_Data.caption + " (Modified)"
        End If

    End If

End Sub

Private Sub txtRadial_Distance1_KeyDown(KeyCode As Integer, Shift As Integer)

    'This routine allows the user to scroll up and down on the list of
    'radial distances.

    'If up arrow on keyboard has been pressed then scroll up one entry.

    If KeyCode = vbKeyUp Then

        If vsbRadius_Number.Value > 1 Then
            Call txtRadial_Distance1_LostFocus
            vsbRadius_Number.Value = vsbRadius_Number.Value - 1
        Else
            Call txtRadial_Distance1_LostFocus
        End If

    'If down arrow has been pressed, scroll down one position.

    ElseIf KeyCode = vbKeyDown Then

        txtRadial_Distance2.SetFocus

    End If

    If (InStr(frmProblem_Data.caption, " (Modified)") = 0) Then
        frmProblem_Data.caption = frmProblem_Data.caption + " (Modified)"
    End If

'     frmMain.mnuSave_ProblemCom.Enabled = True
```

```
'     frmMain.mnuSave_As_ProblemCom.Enabled = True

    frmRosette.blnInputDataChanged = True


End Sub

Private Sub txtRadial_Distance1_KeyPress(keyascii As Integer)

    'This routine marks the problem form as modified if a key is pressed
    'in the distance1 field.

'     frmMain.mnuSave_ProblemCom.Enabled = True
'     frmMain.mnuSave_As_ProblemCom.Enabled = True

    If (InStr(frmProblem_Data.caption, " (Modified)") = 0) Then
        frmProblem_Data.caption = frmProblem_Data.caption + " (Modified)"
    End If

    If keyascii = KEY_RETURN Then
        txtRadial_Distance2.SetFocus
    End If

End Sub

Private Sub txtRadial_Distance1_LostFocus()

    'This routine sorts the entries in the radial distance list so that they
    'appear in ascending order, and updates the list

    Dim temp As Double, units As Integer
    Dim zero_found As Integer, radius_zero As Integer
    Dim radius_edited As Integer, i As Integer, j As Integer
    Dim tempchar As String, TempString As String

    'Record the value of the newly edited radius into the radial
    'distance array.

    radius_edited = Val(lblRadius1.caption)
    radial_distance(radius_edited) = Val(txtRadial_Distance1.text)

    'Examine the Raddius, and round it to three digits after the decimal.

    TempString = Str$(radial_distance(radius_edited))
    For j = 1 To (Len(TempString) - 3)
        If (Mid$(TempString, j, 1) = ".") Then
            tempchar = Mid$(TempString, j + 4, 1)
            TempString = Left$(TempString, j + 3)
        End If
    Next j
    radial_distance(radius_edited) = Val(TempString)

    'Check to see if the radius is within acceptable limits for maccs
    'compatability.

    If (radial_distance(radius_edited) = 0) Then
        radius_zero = True
    ElseIf TypeOf Screen.ActiveControl Is OptionButton Then
    ElseIf (optMiles.Value = True) Then
        If (radial_distance(radius_edited) > 6213) Then
            MsgBox "Error: this program can't process radii greater than 6213 Miles."
            radial_distance(radius_edited) = 0
            txtRadial_Distance1.text = ""
            radius_zero = True
        ElseIf (radial_distance(radius_edited) < 0.0699) Then
            MsgBox "Error: this program can't process radii smaller than .07 Miles."
```

```
        radial_distance(radius_edited) = 0
        txtRadial_Distance1.text = ""
        radius_zero = True
    End If
Else
    If (radial_distance(radius_edited) > 9999) Then
        MsgBox "Error: this program can't process radii greater than 9999 Kilometers."
        radial_distance(radius_edited) = 0
        txtRadial_Distance1.text = ""
        radius_zero = True
    ElseIf (radial_distance(radius_edited) < 0.099) Then
        MsgBox "Error: this program can't process radii smaller than .1 Kilometers."
        radial_distance(radius_edited) = 0
        txtRadial_Distance1.text = ""
        radius_zero = True
    End If
End If

'If radius is not equal to zero, check to see if it is equal to
'any previous radial distance. If it is, set it equal to zero.

If Not radius_zero Then
    For i = 1 To radius_edited - 1 Step 1
        If radial_distance(radius_edited) = radial_distance(i) Then
            radial_distance(radius_edited) = 0
            radius_zero = True
            Exit For
        End If
    Next i
End If

'If distance is still not equal to zero, check to see if it is equal
'to any following radial distance.  If it is, set it equal to zero.

If Not radius_zero Then
    For i = radius_edited + 1 To max_number_of_radii Step 1
        If radial_distance(radius_edited) = radial_distance(i) Then
            radial_distance(i) = 0
            radius_zero = True
            Exit For
        End If
    Next i
End If

'Use standard sorting algorithm to put radii in ascending order.

For i = 1 To max_number_of_radii - 1 Step 1
    For j = i + 1 To max_number_of_radii Step 1
        If (radial_distance(j) <> 0) Then
            If ((radial_distance(i) > radial_distance(j)) Or (radial_distance(i) = 0)) Then
                temp = radial_distance(i)
                radial_distance(i) = radial_distance(j)
                radial_distance(j) = temp
                If (Not radius_zero) Then
                    If i = radius_edited Then
                        radius_edited = j
                    ElseIf j = radius_edited Then
                        radius_edited = i
                    End If
                End If
            End If
        End If
    Next j
Next i

'Determine where list of radii ends.
```

```
    For i = 1 To max_number_of_radii Step 1
        If radial_distance(i) = 0 Then
            Exit For
        End If
    Next i

    'Determine number of radii.

    number_of_radii = i - 1

    If radius_edited > max_number_of_radii - 2 Then
        radius_edited = max_number_of_radii - 2
    End If

    'Update user display (problem form).

    lblRadius1.caption = Right$(Str$(radius_edited), 2)
    txtRadial_Distance1.text = Right$("    " + Format$(radial_distance(radius_edited),
"0.0000;0.0000;      "), 9)
    lblRadius2.caption = Right$(Str$(radius_edited + 1), 2)
    txtRadial_Distance2.text = Right$("    " + Format$(radial_distance(radius_edited + 1),
"0.0000;0.0000;      "), 9)
    lblRadius3.caption = Right$(Str$(radius_edited + 2), 2)
    txtRadial_Distance3.text = Right$("    " + Format$(radial_distance(radius_edited + 2),
"0.0000;0.0000;      "), 9)

    vsbRadius_Number.Value = radius_edited

    If (Val(frmProblem_Data.lblNumber_of_regions.Tag) <> number_of_radii) Then
        Call default_regions
    End If

End Sub

Private Sub txtRadial_Distance2_KeyDown(KeyCode As Integer, Shift As Integer)

    'If an arrow key is pressed, then call the distance 1 function
    'to scroll radial distances up or down.

    If KeyCode = vbKeyUp Then
        txtRadial_Distance1.SetFocus
    ElseIf KeyCode = vbKeyDown Then
        txtRadial_Distance3.SetFocus
    End If

    If (InStr(frmProblem_Data.caption, " (Modified)") = 0) Then
        frmProblem_Data.caption = frmProblem_Data.caption + " (Modified)"
    End If

'    frmMain.mnuSave_ProblemCom.Enabled = True
'    frmMain.mnuSave_As_ProblemCom.Enabled = True

    frmRosette.blnInputDataChanged = True

End Sub

Private Sub txtRadial_Distance2_KeyPress(keyascii As Integer)

    'This routine marks the problem form as modified, and goes to the next
    'radii if enter is pressed.

'    frmMain.mnuSave_ProblemCom.Enabled = True
'    frmMain.mnuSave_As_ProblemCom.Enabled = True

    If (InStr(frmProblem_Data.caption, " (Modified)") = 0) Then
```

```
        frmProblem_Data.caption = frmProblem_Data.caption + " (Modified)"
    End If

    If keyascii = KEY_RETURN Then
        keyascii = 0
        txtRadial_Distance3.SetFocus
    End If

End Sub

Private Sub txtRadial_Distance2_LostFocus()

    'See radial_Distance1_lostfocus for documentation of this routine.

    Dim temp As Double
    Dim zero_found As Integer, radius_zero As Integer
    Dim radius_edited As Integer, i As Integer, j As Integer
    Dim TempString As String, tempchar As String

    radius_edited = Val(lblRadius2.caption)
    radial_distance(radius_edited) = Val(txtRadial_Distance2.text)

    TempString = Str$(radial_distance(radius_edited))
    For j = 1 To (Len(TempString) - 3)
        If (Mid$(TempString, j, 1) = ".") Then
            tempchar = Mid$(TempString, j + 4, 1)
            TempString = Left$(TempString, j + 3)
        End If
    Next j
    radial_distance(radius_edited) = Val(TempString)

    If (radial_distance(radius_edited) = 0) Then
        radius_zero = True
    ElseIf TypeOf Screen.ActiveControl Is OptionButton Then
    ElseIf (optMiles.Value = True) Then
        If (radial_distance(radius_edited) > 6213) Then
            MsgBox "Error: this program can't process radii greater than 6213 Miles."
            radial_distance(radius_edited) = 0
            txtRadial_Distance2.text = ""
            radius_zero = True
        ElseIf (radial_distance(radius_edited) < 0.0695) Then
            MsgBox "Error: this program can't process radii smaller than .07 Miles."
            radial_distance(radius_edited) = 0
            radius_zero = True
        End If
    Else
        If (radial_distance(radius_edited) > 9999) Then
            MsgBox "Error: this program can't process radii greater than 9999 Kilometers."
            radial_distance(radius_edited) = 0
            txtRadial_Distance2.text = ""
            radius_zero = True
        ElseIf (radial_distance(radius_edited) < 0.099) Then
            MsgBox "Error: this program can't process radii smaller than .1 Kilometers."
            radial_distance(radius_edited) = 0
            txtRadial_Distance2.text = ""
            radius_zero = True
        End If
    End If

    If Not radius_zero Then
        For i = 1 To radius_edited - 1 Step 1
            If radial_distance(radius_edited) = radial_distance(i) Then
                radial_distance(radius_edited) = 0
                radius_zero = True
                Exit For
            End If
```

```
        Next i
    End If

    If Not radius_zero Then
        For i = radius_edited + 1 To max_number_of_radii Step 1
            If radial_distance(radius_edited) = radial_distance(i) Then
                radial_distance(radius_edited) = 0
                radius_zero = True
                Exit For
            End If
        Next i
    End If

    For i = 1 To max_number_of_radii - 1 Step 1
        For j = i + 1 To max_number_of_radii Step 1
            If (radial_distance(j) <> 0) Then
                If ((radial_distance(i) > radial_distance(j)) Or (radial_distance(i) = 0)) Then
                    temp = radial_distance(i)
                    radial_distance(i) = radial_distance(j)
                    radial_distance(j) = temp
                    If (Not radius_zero) Then
                        If i = radius_edited Then
                            radius_edited = j
                        ElseIf j = radius_edited Then
                            radius_edited = i
                        End If
                    End If
                End If
            End If
        Next j
    Next i

    i = 1
    zero_found = False

    While ((Not zero_found) And (i <= max_number_of_radii))
        If radial_distance(i) = 0 Then
            zero_found = True
        Else
            i = i + 1
        End If
    Wend

    number_of_radii = i - 1

    If radius_edited > max_number_of_radii - 1 Then
        radius_edited = max_number_of_radii - 1
    ElseIf radius_edited < 2 Then
        radius_edited = 2
    End If

    lblRadius1.caption = Right$(Str$(radius_edited - 1), 2)
    txtRadial_Distance1.text = Right$("    " + Format$(radial_distance(radius_edited - 1),
"0.0000;0.0000;     "), 9)
    lblRadius2.caption = Right$(Str$(radius_edited), 2)
    txtRadial_Distance2.text = Right$("    " + Format$(radial_distance(radius_edited),
"0.0000;0.0000;     "), 9)
    lblRadius3.caption = Right$(Str$(radius_edited + 1), 2)
    txtRadial_Distance3.text = Right$("    " + Format$(radial_distance(radius_edited + 1),
"0.0000;0.0000;     "), 9)

    vsbRadius_Number.Value = radius_edited - 1

    If (Val(frmProblem_Data.lblNumber_of_regions.Tag) <> number_of_radii) Then
        Call default_regions
'sss        Call paint_region(0, 0)
```

```
    End If

End Sub

Private Sub txtRadial_Distance3_KeyDown(KeyCode As Integer, Shift As Integer)

    'See txtRadial_distance1_keydown  for documentation of this routine.

    If KeyCode = vbKeyUp Then
        txtRadial_Distance2.SetFocus
    ElseIf KeyCode = vbKeyDown Then
        If vsbRadius_Number.Value < max_number_of_radii - 2 Then
            Call txtRadial_Distance3_LostFocus
            vsbRadius_Number.Value = vsbRadius_Number.Value + 1
        Else
            Call txtRadial_Distance3_LostFocus
        End If
    End If

    If (InStr(frmProblem_Data.caption, " (Modified)") = 0) Then
        frmProblem_Data.caption = frmProblem_Data.caption + " (Modified)"
    End If

'    frmMain.mnuSave_ProblemCom.Enabled = True
'    frmMain.mnuSave_As_ProblemCom.Enabled = True

    frmRosette.blnInputDataChanged = True


End Sub

Private Sub txtRadial_Distance3_KeyPress(keyascii As Integer)

    'See frmtxtRadial_distance1_keypress for documentation of this function.

'    frmMain.mnuSave_ProblemCom.Enabled = True
'    frmMain.mnuSave_As_ProblemCom.Enabled = True

    If (InStr(frmProblem_Data.caption, " (Modified)") = 0) Then
        frmProblem_Data.caption = frmProblem_Data.caption + " (Modified)"
    End If

    If keyascii = KEY_RETURN Then
        Call txtRadial_Distance3_LostFocus
        If vsbRadius_Number.Value < max_number_of_radii - 2 Then
            vsbRadius_Number.Value = vsbRadius_Number.Value + 1
        End If
    End If

End Sub

Private Sub txtRadial_Distance3_LostFocus()

    'See txtRadial_distance1_lostfocus for documentation of this function.

    Dim temp As Double
    Dim zero_found As Integer, radius_zero As Integer
    Dim radius_edited As Integer, i As Integer, j As Integer
    Dim TempString As String, tempchar As String

    radius_edited = Val(lblRadius3.caption)
    radial_distance(radius_edited) = Val(txtRadial_Distance3.text)

    TempString = Str$(radial_distance(radius_edited))
    For j = 1 To (Len(TempString) - 3)
        If (Mid$(TempString, j, 1) = ".") Then
```

```
            tempchar = Mid$(TempString, j + 4, 1)
            TempString = Left$(TempString, j + 3)
        End If
    Next j
    radial_distance(radius_edited) = Val(TempString)

    If (radial_distance(radius_edited) = 0) Then
        radius_zero = True
    ElseIf TypeOf Screen.ActiveControl Is OptionButton Then
    ElseIf (optMiles.Value = True) Then
        If (radial_distance(radius_edited) > 6213) Then
            MsgBox "Error: this program can't process radii greater than 6213 Miles."
            radial_distance(radius_edited) = 0
            radius_zero = True
        ElseIf (radial_distance(radius_edited) < 0.0695) Then
            MsgBox "Error: this program can't process radii smaller than .07 Miles."
            radial_distance(radius_edited) = 0
            radius_zero = True
        End If
    Else
        If (radial_distance(radius_edited) > 9999) Then
            MsgBox "Error: this program can't process radii greater than 9999 Kilometers."
            radial_distance(radius_edited) = 0
            radius_zero = True
        ElseIf (radial_distance(radius_edited) < 0.099) Then
            MsgBox "Error: this program can't process radii smaller than .1 Kilometers."
            radial_distance(radius_edited) = 0
            radius_zero = True
        End If
    End If

    If Not radius_zero Then
        For i = 1 To radius_edited - 1 Step 1
            If radial_distance(radius_edited) = radial_distance(i) Then
                radial_distance(radius_edited) = 0
                radius_zero = True
                Exit For
            End If
        Next i
    End If

    If Not radius_zero Then
        For i = radius_edited + 1 To max_number_of_radii Step 1
            If radial_distance(radius_edited) = radial_distance(i) Then
                radial_distance(radius_edited) = 0
                radius_zero = True
                Exit For
            End If
        Next i
    End If

    For i = 1 To max_number_of_radii - 1 Step 1
        For j = i + 1 To max_number_of_radii Step 1
            If (radial_distance(j) <> 0) Then
                If ((radial_distance(i) > radial_distance(j)) Or (radial_distance(i) = 0)) Then
                    temp = radial_distance(i)
                    radial_distance(i) = radial_distance(j)
                    radial_distance(j) = temp
                    If (Not radius_zero) Then
                        If i = radius_edited Then
                            radius_edited = j
                        ElseIf j = radius_edited Then
                            radius_edited = i
                        End If
                    End If
                End If
```

```
            End If
        Next j
    Next i

    i = 1
    zero_found = False

    While ((Not zero_found) And (i <= max_number_of_radii))
        If radial_distance(i) = 0 Then
            zero_found = True
        Else
            i = i + 1
        End If
    Wend

    number_of_radii = i - 1

    If radius_edited < 3 Then
        radius_edited = 3
    End If

    If number_of_radii <= max_number_of_radii Then
        lblRadius1.caption = Right$(Str$(radius_edited - 2), 2)
        txtRadial_Distance1.text = Right$("    " + Format$(radial_distance(radius_edited - 2),
"0.0000;0.0000;     "), 9)
        lblRadius2.caption = Right$(Str$(radius_edited - 1), 2)
        txtRadial_Distance2.text = Right$("    " + Format$(radial_distance(radius_edited - 1),
"0.0000;0.0000;     "), 9)
        lblRadius3.caption = Right$(Str$(radius_edited), 2)
        txtRadial_Distance3.text = Right$("    " + Format$(radial_distance(radius_edited),
"0.0000;0.0000;     "), 9)

        vsbRadius_Number.Value = radius_edited - 2
    Else
        lblRadius1.caption = Right$(Str$(radius_edited - 3), 2)
        txtRadial_Distance1.text = Right$("    " + Format$(radial_distance(radius_edited - 3),
"0.0000;0.0000;     "), 9)
        lblRadius2.caption = Right$(Str$(radius_edited - 2), 2)
        txtRadial_Distance2.text = Right$("    " + Format$(radial_distance(radius_edited - 2),
"0.0000;0.0000;     "), 9)
        lblRadius3.caption = Right$(Str$(radius_edited - 1), 2)
        txtRadial_Distance3.text = Right$("    " + Format$(radial_distance(radius_edited - 1),
"0.0000;0.0000;     "), 9)

        vsbRadius_Number.Value = radius_edited - 3
    End If

    If (Val(frmProblem_Data.lblNumber_of_regions.Tag) <> number_of_radii) Then
        Call default_regions
'sss         Call paint_region(0, 0)
    End If

End Sub

'Private Sub txtSite_File_Name_Change()
'
'    'This routine marks the problem form as modified, and if the
'    'enter key is pressed, it attempts to open the site file and read in
'    'the appropriate site data.
'
'    Dim site_file_name As String
'
'    frmMain.mnuSave_ProblemCom.Enabled = True
'    frmMain.mnuSave_As_ProblemCom.Enabled = True
'    cmdSave.Enabled = True
'    cmdSave_As.Enabled = True
```

```
'
'    If (InStr(frmProblem_Data.caption, " (Modified)") = 0) Then
'        frmProblem_Data.caption = frmProblem_Data.caption + " (Modified)"
'    End If
'
''    If keyascii = KEY_RETURN Then
''        keyascii = 0
'        site_file_name = txtSite_File_Name.text
'        open_site_from_problem (site_file_name)
''    End If
'
'End Sub

'Private Sub txtSite_File_Name_KeyPress(keyascii As Integer)
'
'    'This routine marks the problem form as modified, and if the
'    'enter key is pressed, it attempts to open the site file and read in
'    'the appropriate site data.
'
'    Dim site_file_name As String
'
'    frmMain.mnuSave_ProblemCom.Enabled = True
'    frmMain.mnuSave_As_ProblemCom.Enabled = True
'    cmdSave.Enabled = True
'    cmdSave_As.Enabled = True
'
'    If (InStr(frmProblem_Data.caption, " (Modified)") = 0) Then
'        frmProblem_Data.caption = frmProblem_Data.caption + " (Modified)"
'    End If
'
'    If keyascii = KEY_RETURN Then
'        keyascii = 0
'        site_file_name = txtSite_File_Name.text
'        'open_site_from_problem (site_file_name)
'    End If
'
'End Sub

Private Sub txtSite_File_Name_LostFocus()

    'Check and see if the site file name is correct, and if it is read
    'in the site data.

    Dim site_file_name As String

    If frmProblem_Data.Visible Then
        If (Screen.ActiveControl.Tag <> "Browse") And _
           (Screen.ActiveControl.Tag <> "Hiding Form") And _
           (Screen.ActiveControl.Tag <> "Open") And _
           (Screen.ActiveControl.Tag <> "Close") Then
            site_file_name = txtSite_File_Name.text
            'open_site_from_problem (site_file_name)
        End If
        If Screen.ActiveControl.Tag = "Hiding Form" Then
            Screen.ActiveControl.Tag = ""
        End If

    End If

End Sub

Private Sub vsbRadius_Number_Change()

    'This function updates the problem form when a user has changed it.
    'It simply puts the correct values for radii back in the correct places.
    'It is called whenever the scroll bar on the radii box is moved.
```

```
    Dim radius_number(3) As String
    Dim i As Integer

    If (vsbRadius_Number.Value <= max_number_of_radii - 2) Then
        radius_number(1) = Right$(Str$(vsbRadius_Number.Value), 2)
        radius_number(2) = Right$(Str$(vsbRadius_Number.Value + 1), 2)
        radius_number(3) = Right$(Str$(vsbRadius_Number.Value + 2), 2)

        lblRadius1.caption = radius_number(1)
        txtRadial_Distance1.text = Right$("    " +
Format$(radial_distance(vsbRadius_Number.Value), "0.0000;0.0000;     "), 9)

        lblRadius2.caption = radius_number(2)
        txtRadial_Distance2.text = Right$("    " + Format$(radial_distance(vsbRadius_Number.Value
+ 1), "0.0000;0.0000;     "), 9)

        lblRadius3.caption = radius_number(3)
        txtRadial_Distance3.text = Right$("    " + Format$(radial_distance(vsbRadius_Number.Value
+ 2), "0.0000;0.0000;     "), 9)
    End If

End Sub
```

H.29 Visual Basic Form Module: Region.frm

```
VERSION 5.00
Object = "{0D623638-DBA2-11D1-B5DF-0060976089D0}#7.0#0"; "tdbg7.ocx"
Begin VB.Form frmRegion
   Appearance      =   0  'Flat
   AutoRedraw      =   -1  'True
   BackColor       =   &H80000004&
   BorderStyle     =   1  'Fixed Single
   Caption         =   "Region"
   ClientHeight    =   5820
   ClientLeft      =   45
   ClientTop       =   330
   ClientWidth     =   9810
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   8.25
      Charset         =   0
      Weight          =   700
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H00000000&
   Icon            =   "Region.frx":0000
   MaxButton       =   0   'False
   MDIChild        =   -1  'True
   MinButton       =   0   'False
   PaletteMode     =   1  'UseZOrder
   ScaleHeight     =   5820
   ScaleWidth      =   9810
   Begin TrueDBGrid70.TDBGrid TDBGrid1
      Height          =   4110
      Left            =   165
      TabIndex        =   10
      Top             =   930
      Width           =   8280
      _ExtentX        =   14605
      _ExtentY        =   7250
      _LayoutType     =   0
      _RowHeight      =   -2147483647
      _WasPersistedAsPixels=   0
      Columns(0)._VlistStyle=   0
      Columns(0)._MaxComboItems=   5
      Columns(0).DataField=   ""
      Columns(0)._PropDict=   "_MaxComboItems,516,2;_VlistStyle,514,3"
      Columns(1)._VlistStyle=   0
      Columns(1)._MaxComboItems=   5
      Columns(1).DataField=   ""
      Columns(1)._PropDict=   "_MaxComboItems,516,2;_VlistStyle,514,3"
      Columns.Count   =   2
      Splits(0)._UserFlags=   0
      Splits(0).RecordSelectorWidth=   503
      Splits(0).DividerColor=   13160660
      Splits(0).SpringMode=   0   'False
      Splits(0)._PropDict=   "_ColumnProps,515,0;_UserFlags,518,3"
      Splits(0)._ColumnProps(0)=   "Columns.Count=2"
      Splits(0)._ColumnProps(1)=   "Column(0).Width=3281"
      Splits(0)._ColumnProps(2)=   "Column(0).DividerColor=0"
      Splits(0)._ColumnProps(3)=   "Column(0)._WidthInPix=3175"
      Splits(0)._ColumnProps(4)=   "Column(0).Order=1"
      Splits(0)._ColumnProps(5)=   "Column(1).Width=3281"
      Splits(0)._ColumnProps(6)=   "Column(1).DividerColor=0"
      Splits(0)._ColumnProps(7)=   "Column(1)._WidthInPix=3175"
```

```
      Splits(0)._ColumnProps(8)=   "Column(1).Order=2"
      Splits.Count     =   1
      PrintInfos(0)._StateFlags=   3
      PrintInfos(0).Name=   "piInternal 0"
      PrintInfos(0).PageHeaderFont=
"Size=8.25,Charset=0,Weight=700,Underline=0,Italic=0,Strikethrough=0,Name=MS Sans Serif"
      PrintInfos(0).PageFooterFont=
"Size=8.25,Charset=0,Weight=700,Underline=0,Italic=0,Strikethrough=0,Name=MS Sans Serif"
      PrintInfos(0).PageHeaderHeight=   0
      PrintInfos(0).PageFooterHeight=   0
      PrintInfos.Count=   1
      DataMode         =   2
      DefColWidth      =   0
      HeadLines        =   1
      FootLines        =   1
      MultipleLines    =   0
      CellTipsWidth    =   0
      DeadAreaBackColor=   13160660
      RowDividerColor =   13160660
      RowSubDividerColor=   13160660
      DirectionAfterEnter=   1
      MaxRows          =   250000
      _PropDict        =
"_ExtentX,2003,3;_ExtentY,2004,3;_LayoutType,512,2;_RowHeight,16,3;_StyleDefs,513,0;_WasPersisted
AsPixels,516,2"
      _StyleDefs(0)    =   "_StyleRoot:id=0,.parent=-
1,.alignment=3,.valignment=0,.bgcolor=&H80000005&"
      _StyleDefs(1)    =   ":id=0,.fgcolor=&H80000008&,.wraptext=0,.locked=0,.transparentBmp=0"
      _StyleDefs(2)    =
":id=0,.fgpicPosition=0,.bgpicMode=0,.appearance=0,.borderSize=0,.ellipsis=0"
      _StyleDefs(3)    =   ":id=0,.borderColor=&H80000005&,.borderType=0,.bold=-
1,.fontsize=825,.italic=0"
      _StyleDefs(4)    =   ":id=0,.underline=0,.strikethrough=0,.charset=0"
      _StyleDefs(5)    =   ":id=0,.fontname=MS Sans Serif"
      _StyleDefs(6)    =   "Style:id=1,.parent=0,.namedParent=33,.bold=-1,.fontsize=825,.italic=0"
      _StyleDefs(7)    =   ":id=1,.underline=0,.strikethrough=0,.charset=0"
      _StyleDefs(8)    =   ":id=1,.fontname=MS Sans Serif"
      _StyleDefs(9)    =   "CaptionStyle:id=4,.parent=2,.namedParent=37"
      _StyleDefs(10)   =   "HeadingStyle:id=2,.parent=1,.namedParent=34,.bold=-
1,.fontsize=825,.italic=0"
      _StyleDefs(11)   =   ":id=2,.underline=0,.strikethrough=0,.charset=0"
      _StyleDefs(12)   =   ":id=2,.fontname=MS Sans Serif"
      _StyleDefs(13)   =   "FooterStyle:id=3,.parent=1,.namedParent=35,.bold=-
1,.fontsize=825,.italic=0"
      _StyleDefs(14)   =   ":id=3,.underline=0,.strikethrough=0,.charset=0"
      _StyleDefs(15)   =   ":id=3,.fontname=MS Sans Serif"
      _StyleDefs(16)   =
"InactiveStyle:id=5,.parent=2,.bgcolor=&H8000000F&,.fgcolor=&H80000012&"
      _StyleDefs(17)   =   "SelectedStyle:id=6,.parent=1,.namedParent=36"
      _StyleDefs(18)   =   "EditorStyle:id=7,.parent=1"
      _StyleDefs(19)   =   "HighlightRowStyle:id=8,.parent=1,.namedParent=38"
      _StyleDefs(20)   =   "EvenRowStyle:id=9,.parent=1,.namedParent=39"
      _StyleDefs(21)   =   "OddRowStyle:id=10,.parent=1,.namedParent=40"
      _StyleDefs(22)   =   "RecordSelectorStyle:id=11,.parent=2,.namedParent=41"
      _StyleDefs(23)   =   "FilterBarStyle:id=12,.parent=1,.namedParent=42"
      _StyleDefs(24)   =   "Splits(0).Style:id=13,.parent=1"
      _StyleDefs(25)   =   "Splits(0).CaptionStyle:id=22,.parent=4"
      _StyleDefs(26)   =   "Splits(0).HeadingStyle:id=14,.parent=2"
      _StyleDefs(27)   =   "Splits(0).FooterStyle:id=15,.parent=3"
      _StyleDefs(28)   =   "Splits(0).InactiveStyle:id=16,.parent=5"
      _StyleDefs(29)   =   "Splits(0).SelectedStyle:id=18,.parent=6"
      _StyleDefs(30)   =   "Splits(0).EditorStyle:id=17,.parent=7"
      _StyleDefs(31)   =   "Splits(0).HighlightRowStyle:id=19,.parent=8"
      _StyleDefs(32)   =   "Splits(0).EvenRowStyle:id=20,.parent=9"
      _StyleDefs(33)   =   "Splits(0).OddRowStyle:id=21,.parent=10"
      _StyleDefs(34)   =   "Splits(0).RecordSelectorStyle:id=23,.parent=11"
```

```
      _StyleDefs(35)  =   "Splits(0).FilterBarStyle:id=24,.parent=12"
      _StyleDefs(36)  =   "Splits(0).Columns(0).Style:id=28,.parent=13"
      _StyleDefs(37)  =   "Splits(0).Columns(0).HeadingStyle:id=25,.parent=14"
      _StyleDefs(38)  =   "Splits(0).Columns(0).FooterStyle:id=26,.parent=15"
      _StyleDefs(39)  =   "Splits(0).Columns(0).EditorStyle:id=27,.parent=17"
      _StyleDefs(40)  =   "Splits(0).Columns(1).Style:id=32,.parent=13"
      _StyleDefs(41)  =   "Splits(0).Columns(1).HeadingStyle:id=29,.parent=14"
      _StyleDefs(42)  =   "Splits(0).Columns(1).FooterStyle:id=30,.parent=15"
      _StyleDefs(43)  =   "Splits(0).Columns(1).EditorStyle:id=31,.parent=17"
      _StyleDefs(44)  =   "Named:id=33:Normal"
      _StyleDefs(45)  =   ":id=33,.parent=0"
      _StyleDefs(46)  =   "Named:id=34:Heading"
      _StyleDefs(47)  =
":id=34,.parent=33,.valignment=2,.bgcolor=&H8000000F&,.fgcolor=&H80000012&"
      _StyleDefs(48)  =   ":id=34,.wraptext=-1"
      _StyleDefs(49)  =   "Named:id=35:Footing"
      _StyleDefs(50)  =
":id=35,.parent=33,.valignment=2,.bgcolor=&H8000000F&,.fgcolor=&H80000012&"
      _StyleDefs(51)  =   "Named:id=36:Selected"
      _StyleDefs(52)  =   ":id=36,.parent=33,.bgcolor=&H8000000D&,.fgcolor=&H8000000E&"
      _StyleDefs(53)  =   "Named:id=37:Caption"
      _StyleDefs(54)  =   ":id=37,.parent=34,.alignment=2"
      _StyleDefs(55)  =   "Named:id=38:HighlightRow"
      _StyleDefs(56)  =   ":id=38,.parent=33,.bgcolor=&H8000000D&,.fgcolor=&H8000000E&"
      _StyleDefs(57)  =   "Named:id=39:EvenRow"
      _StyleDefs(58)  =   ":id=39,.parent=33,.bgcolor=&HFFFF00&"
      _StyleDefs(59)  =   "Named:id=40:OddRow"
      _StyleDefs(60)  =   ":id=40,.parent=33"
      _StyleDefs(61)  =   "Named:id=41:RecordSelector"
      _StyleDefs(62)  =   ":id=41,.parent=34"
      _StyleDefs(63)  =   "Named:id=42:FilterBar"
      _StyleDefs(64)  =   ":id=42,.parent=33"
   End
   Begin VB.Frame Frame1
      Appearance      =   0  'Flat
      BackColor       =   &H80000004&
      Caption         =   "Regions"
      ForeColor       =   &H00000000&
      Height          =   720
      Left            =   8550
      TabIndex        =   6
      Top             =   990
      Width           =   1188
      Begin VB.Label lblnum_regions
         Alignment       =   1  'Right Justify
         Appearance      =   0  'Flat
         BackColor       =   &H80000004&
         Caption         =   "99"
         ForeColor       =   &H00000000&
         Height          =   240
         Left            =   405
         TabIndex        =   7
         Top             =   315
         Width           =   330
      End
   End
   Begin VB.CommandButton cmdSort
      Appearance      =   0  'Flat
      BackColor       =   &H00808000&
      Caption         =   "&Sort"
      Height          =   480
      Left            =   7665
      TabIndex        =   0
      Top             =   5250
      Width           =   1728
   End
```

```
Begin VB.CommandButton cmdDefault
   Appearance      =   0  'Flat
   BackColor       =   &H00808000&
   Caption         =   "De&fault"
   Height          =   480
   Left            =   5820
   TabIndex        =   1
   Top             =   5250
   Width           =   1728
End
Begin VB.CommandButton cmdClear
   Appearance      =   0  'Flat
   BackColor       =   &H00808000&
   Caption         =   "Clea&r"
   Height          =   480
   Left            =   3975
   TabIndex        =   2
   Top             =   5250
   Width           =   1728
End
Begin VB.CommandButton cmdClose
   Appearance      =   0  'Flat
   BackColor       =   &H00808000&
   Caption         =   "&Close"
   Height          =   480
   Left            =   2130
   TabIndex        =   3
   Top             =   5250
   Width           =   1728
End
Begin VB.Frame Frame2
   Appearance      =   0  'Flat
   BackColor       =   &H80000004&
   Caption         =   "Radii"
   ForeColor       =   &H00000000&
   Height          =   720
   Left            =   8550
   TabIndex        =   8
   Top             =   1815
   Width           =   1188
   Begin VB.Label lblnum_radii
      Alignment       =   1  'Right Justify
      Appearance      =   0  'Flat
      BackColor       =   &H80000004&
      Caption         =   "50"
      ForeColor       =   &H00000000&
      Height          =   240
      Left            =   405
      TabIndex        =   9
      Top             =   285
      Width           =   330
   End
End
Begin VB.Label lblRadial_Distances
   Alignment       =   1  'Right Justify
   Appearance      =   0  'Flat
   AutoSize        =   -1  'True
   BackColor       =   &H80000004&
   Caption         =   "Radial Distances in "
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   12
      Charset         =   0
      Weight          =   700
      Underline       =   0   'False
      Italic          =   0   'False
```

```
         Strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H00000000&
      Height          =   300
      Left            =   2895
      TabIndex        =   4
      Top             =   405
      Width           =   2415
   End
   Begin VB.Label lblUnits
      Appearance      =   0  'Flat
      AutoSize        =   -1  'True
      BackColor       =   &H80000004&
      Caption         =   "Miles"
      BeginProperty Font
         Name            =   "MS Sans Serif"
         Size            =   12
         Charset         =   0
         Weight          =   700
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H00000000&
      Height          =   300
      Left            =   5310
      TabIndex        =   5
      Top             =   405
      Width           =   750
   End
End
Attribute VB_Name = "frmRegion"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
'*****************************************************************************
'  Form:  frmRegion
'
'  Project: SECPOP
'
'  Description: This form allows the user to display and edit region data.
'
'
'  Modification History
'  Date      By   Description
'  --------  ---  ------------------------------------------------------------
'  20020118  ss   Changed the display of the region data from being drawn on
'                 the form using textboxes, to being loaded into a data grid.
'
'*****************************************************************************

Option Explicit

Dim MaxRow As Integer
Dim MaxCol As Integer
Dim i As Integer
Dim j As Integer
Dim k As Integer
Dim GridArray() As String
Dim Sink As New GridClass

Private Sub cmdClear_Click()

    Dim i As Integer, j As Integer
```

```
    number_econ_regions = 0
    frmProblem_Data.lblNumber_of_regions.caption = 0
    lblnum_regions = ""

    For i = 2 To MaxCol - 1
       For j = 0 To MaxRow - 1
'          GridArray(i, j) = " "
          Sink.Value(j, i) = "0"
          region_index(i, j) = 0
       Next j
    Next i
    TDBGrid1.Refresh

End Sub

Private Sub cmdClose_Click()

    Unload Me

'    For i = 1 To MaxCol - 1
'       For j = 0 To MaxRow - 1
''          If GridArray(i, j) <> " " Then
''            region_index(i, j) = CInt(GridArray(i, j))
'           region_index(i, j) = CInt(Sink.Value(j, i))
''          End If
'       Next j
'    Next i
'
'    Call cmdSort_Click
'    If frmRegion.cmdSort.Tag = "F" Then
'        Exit Sub
'    Else
'        Unload frmRegion
'    End If
'
'
End Sub

Private Sub cmdDefault_Click()

    Call default_regions

    For i = 1 To MaxCol - 1
       For j = 0 To MaxRow - 1
'          GridArray(i, j) = Str(region_index(i, j))
          Sink.Value(j, i) = Str(region_index(i, j))
       Next j
    Next i
    TDBGrid1.Refresh
    lblnum_regions = number_econ_regions

End Sub


Private Sub cmdSort_Click()

    'Renumber the economic regions so that all numbers are contiguous

    Dim i As Integer, j As Integer, m As Integer, n As Integer
    Dim current_region As Integer, start_number_of_sectors As Integer
    Dim previous_value As Integer

    For i = 0 To number_of_segments - 1
        For j = 1 To number_of_radii
            If (region_index(j, i) <= 0) Then
                MsgBox "Error: Every sector must be part of a region"
```

```
                frmRegion.cmdSort.Tag = "F"
                Exit Sub
            End If
        Next j
    Next i
    frmRegion.cmdSort.Tag = "Y"

    current_region = 2

    For i = 0 To number_of_segments - 1
        For j = 2 To number_of_radii
            If (region_index(j, i) >= current_region) Then
                previous_value = region_index(j, i)
                For m = i To number_of_segments - 1
                    For n = 2 To number_of_radii
                        If (region_index(n, m) = previous_value) Then
                            region_index(n, m) = current_region
                        ElseIf (region_index(n, m) = current_region) Then
                            region_index(n, m) = 100 + current_region
                        End If
                    Next n
                Next m
                current_region = current_region + 1
            End If
        Next j
    Next i

    number_econ_regions = current_region - 1
    frmProblem_Data.lblNumber_of_regions.caption = Str$(number_econ_regions)
    lblnum_regions = number_econ_regions

    For i = 1 To MaxCol - 1
       For j = 0 To MaxRow - 1
'          GridArray(i, j) = Str(region_index(i, j))
           Sink.Value(j, i) = Str(region_index(i, j))
       Next j
    Next i

'    TDBGrid1.Update
'    TDBGrid1.DataChanged
'    TDBGrid1.EmptyRows
    TDBGrid1.Refresh

End Sub

Private Sub Form_Load()


    Dim row As Long, col As Integer
    Dim C As TrueDBGrid70.Column
    Dim Cols As TrueDBGrid70.Columns

'   set form's position

    Dim FormsTop As Integer
    Dim FormsLeft As Integer
    FormsTop = 50
    FormsLeft = 50
    Me.Top = FormsTop
    Me.Left = FormsLeft


    lblnum_regions = number_econ_regions
    lblnum_radii = number_of_radii

    'maxcol = UBound(region_index, 1)   ' 35
```

```
    MaxCol = number_of_radii + 1
    MaxRow = UBound(region_index, 2)   ' 16
    col = number_of_radii + 1
    row = UBound(region_index, 2)   ' 16

    Screen.MousePointer = vbHourglass

    ' Initialize storage
    Sink.SetDims row, col

'   fill one grid (gridarray) with both directions and data
'
'    ReDim GridArray(0 To MaxCol - 1, 0 To MaxRow)
'
'    For i = 0 To MaxRow - 1
'        GridArray(0, i) = directions(i)
'    Next i
    For i = 0 To MaxRow - 1
        Sink.Value(i, 0) = directions(i)
    Next i

    For i = 1 To MaxCol - 1
       For j = 0 To MaxRow - 1
          Sink.Value(j, i) = Str(region_index(i, j))
'          GridArray(i, j) = Str(region_index(i, j))
       Next j
    Next i

    ' Make sure no events are fired for the grid yet
    Sink.Attach Nothing

    Set Cols = TDBGrid1.Columns

    ' Remove all columns
    While Cols.Count
        Cols.Remove 0
    Wend
    TDBGrid1.ReBind

    ' Add columns to the grid
    For col = 0 To Sink.ColCount - 1
        Set C = Cols.Add(col)
        C.Visible = True
        C.Width = 1000
        If col = 0 Then
            C.caption = "Sectors"
        Else
            C.caption = Str(radial_distance(col))
        End If
    Next col

    'set various parameters of the grid

    TDBGrid1.Columns(0).DividerStyle = 8   ' double line dividing col 0 and col 1

    TDBGrid1.Columns(0).Locked = True
    TDBGrid1.Columns(1).Locked = True

'    TDBGrid1.ApproxCount = MaxRow

    TDBGrid1.Columns(0).BackColor = &HC0C0C0
    TDBGrid1.Columns(1).ForeColor = &H808080

    TDBGrid1.RecordSelectors = False

    'Initialize current cell position to upper left corner:
```

```
    TDBGrid1.row = 0
    TDBGrid1.col = 0

    'set splits

    Dim s As TrueDBGrid70.Split
    Set s = TDBGrid1.Splits.Add(0)

    Set Cols = TDBGrid1.Splits(0).Columns  ' sector column
    For i = 1 To MaxCol - 1
        Cols(i).Visible = False
    Next i

    Set Cols = TDBGrid1.Splits(1).Columns  ' data columns
    Cols(0).Visible = False
'    Cols(MaxCol - 1).Visible = False

    TDBGrid1.Splits(0).Size = 1
    TDBGrid1.Splits(1).Size = MaxCol - 1

    ' Initialise the class
    Sink.Attach TDBGrid1

    ' Reinitialize the grid
    TDBGrid1.Bookmark = Null
    TDBGrid1.ReBind

    ' Calibrate the VScroll bar
    TDBGrid1.ApproxCount = Sink.RowCount
'    TDBGrid1.SetFocus
    Screen.MousePointer = vbDefault

End Sub

Private Sub Form_Unload(Cancel As Integer)


    For i = 1 To MaxCol - 1
       For j = 0 To MaxRow - 1
'          If GridArray(i, j) <> " " Then
'            region_index(i, j) = CInt(GridArray(i, j))
            region_index(i, j) = CInt(Sink.Value(j, i))
'          End If
       Next j
    Next i

    Call cmdSort_Click
    If frmRegion.cmdSort.Tag = "F" Then
        Exit Sub
    Else
        Unload frmRegion
    End If

    Set Sink = Nothing

'    frmMain.mnuSave_ProblemCom.Enabled = True
'    frmMain.mnuSave_As_ProblemCom.Enabled = True
'    frmProblem_Data.cmdSave.Enabled = True
'    frmProblem_Data.cmdSave_As.Enabled = True
'
End Sub

Private Sub TDBGrid1_BeforeColUpdate(ByVal ColIndex As Integer, _
                                     OldValue As Variant, _
                                     Cancel As Integer)
```

```
'    check for valid input

     If TDBGrid1.text <> "" Then
         If CInt(TDBGrid1.text) < 2 Or _
            CInt(TDBGrid1.text) > 99 Then
     '        schedule the PostEvent event
              TDBGrid1.PostMsg 1
              Cancel = True
         End If
     Else
'         schedule the PostEvent event
          TDBGrid1.PostMsg 1
          Cancel = True
     End If

End Sub

Private Sub TDBGrid1_KeyPress(keyascii As Integer)

'     If (KeyAscii >= 48 And KeyAscii <= 57) Or KeyAscii = 8 Then ' 0-9 or backspace
     If (keyascii >= 48 And keyascii <= 57) Or _
         keyascii = vbKeyBack Then ' 0-9 or backspace
         'continue
     Else
         MsgBox "Values must be between 2 and 99.  "
         keyascii = 0
         Beep
     End If

End Sub

Private Sub TDBGrid1_LostFocus()

     TDBGrid1.Update   '(forces update of underlying database)
     For i = 1 To MaxCol - 1
        For j = 0 To MaxRow - 1
               region_index(i, j) = CInt(Sink.Value(j, i))
'               region_index(i, j) = CInt(GridArray(i, j))
        Next j
     Next i

End Sub

Private Sub TDBGrid1_PostEvent(ByVal MsgId As Integer)

    If MsgId = 1 Then MsgBox "Values must be between 2 and 99.  " & _
                             "This cell will revert back to it's origional value."

End Sub
```

H.30 Visual Basic Form Module: Rosette_frm1.frm

```
VERSION 5.00
Object = "{0D623638-DBA2-11D1-B5DF-0060976089D0}#7.0#0"; "tdbg7.ocx"
Begin VB.Form frmRosette
   Appearance      =   0  'Flat
   AutoRedraw      =   -1  'True
   BackColor       =   &H80000005&
   BorderStyle     =   1  'Fixed Single
   Caption         =   " Rosette"
   ClientHeight    =   6960
   ClientLeft      =   7020
   ClientTop       =   2430
   ClientWidth     =   11670
   BeginProperty Font
      Name            =   "Arial Narrow"
      Size            =   9
      Charset         =   0
      Weight          =   400
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   FontTransparent =   0   'False
   Icon            =   "Rosette_frm1.frx":0000
   LinkTopic       =   "Form1"
   MaxButton       =   0   'False
   MDIChild        =   -1  'True
   MinButton       =   0   'False
   ScaleHeight     =   348
   ScaleMode       =   2  'Point
   ScaleWidth      =   583.5
   Begin VB.PictureBox pctCapture
      AutoSize        =   -1  'True
      Height          =   510
      Left            =   1845
      ScaleHeight     =   450
      ScaleWidth      =   435
      TabIndex        =   45
      TabStop         =   0   'False
      Top             =   5820
      Visible         =   0   'False
      Width           =   495
   End
   Begin VB.CommandButton cmdSave
      Cancel          =   -1  'True
      Caption         =   "Save to File"
      Default         =   -1  'True
      BeginProperty Font
         Name            =   "Arial"
         Size            =   11.25
         Charset         =   0
         Weight          =   700
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      Height          =   540
      Left            =   90
      Style           =   1  'Graphical
      TabIndex        =   2
      ToolTipText     =   "Saves this form in bitmap form."
      Top             =   6300
      Width           =   1650
```

```
   End
   Begin VB.CommandButton cmdPrint
      Caption         =   "Print"
      BeginProperty Font
         Name            =   "Arial"
         Size            =   11.25
         Charset         =   0
         Weight          =   700
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      Height          =   540
      Left            =   90
      Style           =   1  'Graphical
      TabIndex        =   1
      ToolTipText     =   "Sends this form to a printer."
      Top             =   5670
      Width           =   1650
   End
   Begin TrueDBGrid70.TDBGrid TDBGrid1
      Height          =   5130
      Left            =   90
      TabIndex        =   44
      TabStop         =   0   'False
      Top             =   405
      Width           =   1680
      _ExtentX        =   2963
      _ExtentY        =   9049
      _LayoutType     =   0
      _RowHeight      =   -2147483647
      _WasPersistedAsPixels=   0
      Columns(0)._VlistStyle=   0
      Columns(0)._MaxComboItems=   5
      Columns(0).DataField=   ""
      Columns(0)._PropDict=   "_MaxComboItems,516,2;_VlistStyle,514,3"
      Columns(1)._VlistStyle=   0
      Columns(1)._MaxComboItems=   5
      Columns(1).DataField=   ""
      Columns(1)._PropDict=   "_MaxComboItems,516,2;_VlistStyle,514,3"
      Columns.Count   =   2
      Splits(0)._UserFlags=   0
      Splits(0).RecordSelectorWidth=   503
      Splits(0).DividerColor=   12632256
      Splits(0).SpringMode=   0   'False
      Splits(0)._PropDict=   "_ColumnProps,515,0;_UserFlags,518,3"
      Splits(0)._ColumnProps(0)=   "Columns.Count=2"
      Splits(0)._ColumnProps(1)=   "Column(0).Width=2196"
      Splits(0)._ColumnProps(2)=   "Column(0).DividerColor=0"
      Splits(0)._ColumnProps(3)=   "Column(0)._WidthInPix=2117"
      Splits(0)._ColumnProps(4)=   "Column(0).Order=1"
      Splits(0)._ColumnProps(5)=   "Column(1).Width=2196"
      Splits(0)._ColumnProps(6)=   "Column(1).DividerColor=0"
      Splits(0)._ColumnProps(7)=   "Column(1)._WidthInPix=2117"
      Splits(0)._ColumnProps(8)=   "Column(1).Order=2"
      Splits.Count    =   1
      PrintInfos(0)._StateFlags=   3
      PrintInfos(0).Name=   "piInternal 0"
      PrintInfos(0).PageHeaderFont=
"Size=9,Charset=0,Weight=400,Underline=0,Italic=0,Strikethrough=0,Name=Arial Narrow"
      PrintInfos(0).PageFooterFont=
"Size=9,Charset=0,Weight=400,Underline=0,Italic=0,Strikethrough=0,Name=Arial Narrow"
      PrintInfos(0).PageHeaderHeight=   0
      PrintInfos(0).PageFooterHeight=   0
      PrintInfos.Count=   1
      DataMode        =   2
```

```
        DefColWidth     =   0
        HeadLines       =   1
        FootLines       =   1
        MultipleLines   =   0
        CellTipsWidth   =   0
        DeadAreaBackColor=   16777215
        RowDividerColor =   12632256
        RowSubDividerColor=   12632256
        DirectionAfterEnter=   1
        MaxRows         =   250000
        _PropDict       =
"_ExtentX,2003,3;_ExtentY,2004,3;_LayoutType,512,2;_RowHeight,16,3;_StyleDefs,513,0;_WasPersisted
AsPixels,516,2"
        _StyleDefs(0)   =   "_StyleRoot:id=0,.parent=-
1,.alignment=3,.valignment=0,.bgcolor=&H80000005&"
        _StyleDefs(1)   =   ":id=0,.fgcolor=&H80000008&,.wraptext=0,.locked=0,.transparentBmp=0"
        _StyleDefs(2)   =
":id=0,.fgpicPosition=0,.bgpicMode=0,.appearance=0,.borderSize=0,.ellipsis=0"
        _StyleDefs(3)   =
":id=0,.borderColor=&H80000005&,.borderType=0,.bold=0,.fontsize=900,.italic=0"
        _StyleDefs(4)   =   ":id=0,.underline=0,.strikethrough=0,.charset=0"
        _StyleDefs(5)   =   ":id=0,.fontname=Arial Narrow"
        _StyleDefs(6)   =   "Style:id=1,.parent=0,.namedParent=33,.bold=0,.fontsize=900,.italic=0"
        _StyleDefs(7)   =   ":id=1,.underline=0,.strikethrough=0,.charset=0"
        _StyleDefs(8)   =   ":id=1,.fontname=Arial Narrow"
        _StyleDefs(9)   =   "CaptionStyle:id=4,.parent=2,.namedParent=37"
        _StyleDefs(10)  =
"HeadingStyle:id=2,.parent=1,.namedParent=34,.bold=0,.fontsize=900,.italic=0"
        _StyleDefs(11)  =   ":id=2,.underline=0,.strikethrough=0,.charset=0"
        _StyleDefs(12)  =   ":id=2,.fontname=Arial Narrow"
        _StyleDefs(13)  =
"FooterStyle:id=3,.parent=1,.namedParent=35,.bold=0,.fontsize=900,.italic=0"
        _StyleDefs(14)  =   ":id=3,.underline=0,.strikethrough=0,.charset=0"
        _StyleDefs(15)  =   ":id=3,.fontname=Arial Narrow"
        _StyleDefs(16)  =
"InactiveStyle:id=5,.parent=2,.bgcolor=&H8000000F&,.fgcolor=&H80000012&"
        _StyleDefs(17)  =   "SelectedStyle:id=6,.parent=1,.namedParent=36"
        _StyleDefs(18)  =   "EditorStyle:id=7,.parent=1"
        _StyleDefs(19)  =   "HighlightRowStyle:id=8,.parent=1,.namedParent=38"
        _StyleDefs(20)  =   "EvenRowStyle:id=9,.parent=1,.namedParent=39"
        _StyleDefs(21)  =   "OddRowStyle:id=10,.parent=1,.namedParent=40"
        _StyleDefs(22)  =   "RecordSelectorStyle:id=11,.parent=2,.namedParent=41"
        _StyleDefs(23)  =   "FilterBarStyle:id=12,.parent=1,.namedParent=42"
        _StyleDefs(24)  =   "Splits(0).Style:id=13,.parent=1"
        _StyleDefs(25)  =   "Splits(0).CaptionStyle:id=22,.parent=4"
        _StyleDefs(26)  =   "Splits(0).HeadingStyle:id=14,.parent=2"
        _StyleDefs(27)  =   "Splits(0).FooterStyle:id=15,.parent=3"
        _StyleDefs(28)  =   "Splits(0).InactiveStyle:id=16,.parent=5"
        _StyleDefs(29)  =   "Splits(0).SelectedStyle:id=18,.parent=6"
        _StyleDefs(30)  =   "Splits(0).EditorStyle:id=17,.parent=7"
        _StyleDefs(31)  =   "Splits(0).HighlightRowStyle:id=19,.parent=8"
        _StyleDefs(32)  =   "Splits(0).EvenRowStyle:id=20,.parent=9"
        _StyleDefs(33)  =   "Splits(0).OddRowStyle:id=21,.parent=10"
        _StyleDefs(34)  =   "Splits(0).RecordSelectorStyle:id=23,.parent=11"
        _StyleDefs(35)  =   "Splits(0).FilterBarStyle:id=24,.parent=12"
        _StyleDefs(36)  =   "Splits(0).Columns(0).Style:id=28,.parent=13"
        _StyleDefs(37)  =   "Splits(0).Columns(0).HeadingStyle:id=25,.parent=14"
        _StyleDefs(38)  =   "Splits(0).Columns(0).FooterStyle:id=26,.parent=15"
        _StyleDefs(39)  =   "Splits(0).Columns(0).EditorStyle:id=27,.parent=17"
        _StyleDefs(40)  =   "Splits(0).Columns(1).Style:id=32,.parent=13"
        _StyleDefs(41)  =   "Splits(0).Columns(1).HeadingStyle:id=29,.parent=14"
        _StyleDefs(42)  =   "Splits(0).Columns(1).FooterStyle:id=30,.parent=15"
        _StyleDefs(43)  =   "Splits(0).Columns(1).EditorStyle:id=31,.parent=17"
        _StyleDefs(44)  =   "Named:id=33:Normal"
        _StyleDefs(45)  =   ":id=33,.parent=0"
        _StyleDefs(46)  =   "Named:id=34:Heading"
```

```
      _StyleDefs(47)  =
":id=34,.parent=33,.valignment=2,.bgcolor=&H8000000F&,.fgcolor=&H80000012&"
      _StyleDefs(48)  =   ":id=34,.wraptext=-1"
      _StyleDefs(49)  =   "Named:id=35:Footing"
      _StyleDefs(50)  =
":id=35,.parent=33,.valignment=2,.bgcolor=&H8000000F&,.fgcolor=&H80000012&"
      _StyleDefs(51)  =   "Named:id=36:Selected"
      _StyleDefs(52)  =   ":id=36,.parent=33,.bgcolor=&H8000000D&,.fgcolor=&H8000000E&"
      _StyleDefs(53)  =   "Named:id=37:Caption"
      _StyleDefs(54)  =   ":id=37,.parent=34,.alignment=2"
      _StyleDefs(55)  =   "Named:id=38:HighlightRow"
      _StyleDefs(56)  =   ":id=38,.parent=33,.bgcolor=&H8000000D&,.fgcolor=&H8000000E&"
      _StyleDefs(57)  =   "Named:id=39:EvenRow"
      _StyleDefs(58)  =   ":id=39,.parent=33,.bgcolor=&HFFFF00&"
      _StyleDefs(59)  =   "Named:id=40:OddRow"
      _StyleDefs(60)  =   ":id=40,.parent=33"
      _StyleDefs(61)  =   "Named:id=41:RecordSelector"
      _StyleDefs(62)  =   ":id=41,.parent=34"
      _StyleDefs(63)  =   "Named:id=42:FilterBar"
      _StyleDefs(64)  =   ":id=42,.parent=33"
   End
   Begin VB.ListBox List1
      Appearance      =   0  'Flat
      BeginProperty DataFormat
         Type            =   1
         Format          =   "#,##0.00"
         HaveTrueFalseNull=   0
         FirstDayOfWeek  =   0
         FirstWeekOfYear =   0
         LCID            =   1033
         SubFormatType   =   1
      EndProperty
      Height          =   270
      Index           =   1
      IntegralHeight  =   0   'False
      Left            =   840
      TabIndex        =   34
      TabStop         =   0   'False
      Top             =   585
      Width           =   735
   End
   Begin VB.ListBox List1
      Appearance      =   0  'Flat
      CausesValidation=   0   'False
      Height          =   270
      Index           =   0
      IntegralHeight  =   0   'False
      ItemData        =   "Rosette_frm1.frx":0CCA
      Left            =   120
      List            =   "Rosette_frm1.frx":0CCC
      TabIndex        =   33
      TabStop         =   0   'False
      Top             =   585
      Width           =   735
   End
   Begin VB.CommandButton CancelButton
      Caption         =   "Close"
      BeginProperty Font
         Name            =   "Arial"
         Size            =   11.25
         Charset         =   0
         Weight          =   700
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
```

```
   Height          =   540
   Left            =   9900
   Style           =   1  'Graphical
   TabIndex        =   0
   Top             =   6300
   Width           =   1650
End
Begin VB.PictureBox Picture1
   AutoRedraw      =   -1  'True
   AutoSize        =   -1  'True
   BackColor       =   &H8000000E&
   BorderStyle     =   0  'None
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   8.25
      Charset         =   0
      Weight          =   400
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H80000007&
   Height          =   6525
   Left            =   2025
   ScaleHeight     =   435
   ScaleMode       =   3  'Pixel
   ScaleWidth      =   457
   TabIndex        =   3
   TabStop         =   0   'False
   Top             =   405
   Width           =   6855
   Begin VB.Label Direction
      Alignment       =   2  'Center
      AutoSize        =   -1  'True
      BackColor       =   &H8000000E&
      BackStyle       =   0  'Transparent
      Caption         =   "NE"
      BeginProperty Font
         Name            =   "MS Sans Serif"
         Size            =   12
         Charset         =   0
         Weight          =   700
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      Height          =   300
      Index           =   2
      Left            =   5445
      TabIndex        =   41
      Top             =   1080
      Width           =   405
   End
   Begin VB.Label Direction
      Alignment       =   2  'Center
      AutoSize        =   -1  'True
      BackColor       =   &H8000000E&
      BackStyle       =   0  'Transparent
      Caption         =   "SE"
      BeginProperty Font
         Name            =   "MS Sans Serif"
         Size            =   12
         Charset         =   0
         Weight          =   700
         Underline       =   0   'False
         Italic          =   0   'False
```

```
      Strikethrough   =   0   'False
    EndProperty
    Height          =   300
    Index           =   6
    Left            =   5490
    TabIndex        =   40
    Top             =   5205
    Width           =   405
End
Begin VB.Label Direction
    Alignment       =   2   'Center
    AutoSize        =   -1  'True
    BackColor       =   &H8000000E&
    BackStyle       =   0   'Transparent
    Caption         =   "SSE"
    BeginProperty Font
       Name             =   "MS Sans Serif"
       Size             =   12
       Charset          =   0
       Weight           =   700
       Underline        =   0   'False
       Italic           =   0   'False
       Strikethrough    =   0   'False
    EndProperty
    Height          =   300
    Index           =   7
    Left            =   4380
    TabIndex        =   39
    Top             =   5910
    Width           =   720
End
Begin VB.Label Direction
    Alignment       =   2   'Center
    AutoSize        =   -1  'True
    BackColor       =   &H8000000E&
    BackStyle       =   0   'Transparent
    Caption         =   "SSW"
    BeginProperty Font
       Name             =   "MS Sans Serif"
       Size             =   12
       Charset          =   0
       Weight           =   700
       Underline        =   0   'False
       Italic           =   0   'False
       Strikethrough    =   0   'False
    EndProperty
    Height          =   300
    Index           =   15
    Left            =   1635
    TabIndex        =   38
    Top             =   5910
    Width           =   645
End
Begin VB.Label Direction
    Alignment       =   2   'Center
    AutoSize        =   -1  'True
    BackColor       =   &H8000000E&
    BackStyle       =   0   'Transparent
    Caption         =   "SW"
    BeginProperty Font
       Name             =   "MS Sans Serif"
       Size             =   12
       Charset          =   0
       Weight           =   700
       Underline        =   0   'False
       Italic           =   0   'False
```

```
      Strikethrough   =   0   'False
   EndProperty
   Height          =   300
   Index           =   14
   Left            =   855
   TabIndex        =   30
   Top             =   5205
   Width           =   405
End
Begin VB.Label Direction
   Alignment       =   2  'Center
   AutoSize        =   -1  'True
   BackColor       =   &H8000000E&
   BackStyle       =   0  'Transparent
   Caption         =   "WSW"
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   12
      Charset         =   0
      Weight          =   700
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   Height          =   300
   Index           =   13
   Left            =   30
   TabIndex        =   29
   Top             =   4320
   Width           =   705
End
Begin VB.Label Direction
   Alignment       =   2  'Center
   AutoSize        =   -1  'True
   BackColor       =   &H8000000E&
   BackStyle       =   0  'Transparent
   Caption         =   "W"
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   12
      Charset         =   0
      Weight          =   700
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   Height          =   300
   Index           =   12
   Left            =   270
   TabIndex        =   28
   Top             =   3210
   Width           =   285
End
Begin VB.Label Direction
   Alignment       =   2  'Center
   AutoSize        =   -1  'True
   BackColor       =   &H8000000E&
   BackStyle       =   0  'Transparent
   Caption         =   "WNW"
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   12
      Charset         =   0
      Weight          =   700
      Underline       =   0   'False
      Italic          =   0   'False
```

```
      Strikethrough   =   0   'False
   EndProperty
   Height          =   300
   Index           =   11
   Left            =   90
   TabIndex        =   27
   Top             =   2085
   Width           =   705
End
Begin VB.Label Direction
   Alignment       =   2  'Center
   AutoSize        =   -1  'True
   BackColor       =   &H8000000E&
   BackStyle       =   0  'Transparent
   Caption         =   "NW"
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   12
      Charset         =   0
      Weight          =   700
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   Height          =   300
   Index           =   10
   Left            =   915
   TabIndex        =   26
   Top             =   1080
   Width           =   465
End
Begin VB.Label Direction
   Alignment       =   2  'Center
   AutoSize        =   -1  'True
   BackColor       =   &H8000000E&
   BackStyle       =   0  'Transparent
   Caption         =   "NNW"
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   12
      Charset         =   0
      Weight          =   700
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   Height          =   300
   Index           =   9
   Left            =   1830
   TabIndex        =   25
   Top             =   450
   Width           =   645
End
Begin VB.Label Direction
   Alignment       =   2  'Center
   AutoSize        =   -1  'True
   BackColor       =   &H8000000E&
   BackStyle       =   0  'Transparent
   Caption         =   "S"
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   12
      Charset         =   0
      Weight          =   700
      Underline       =   0   'False
      Italic          =   0   'False
```

```
      Strikethrough   =   0   'False
   EndProperty
   Height          =   300
   Index           =   8
   Left            =   3270
   TabIndex        =   24
   Top             =   6195
   Width           =   225
End
Begin VB.Label Direction
   Alignment       =   2  'Center
   AutoSize        =   -1  'True
   BackColor       =   &H8000000E&
   BackStyle       =   0  'Transparent
   Caption         =   "ESE"
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   12
      Charset         =   0
      Weight          =   700
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   Height          =   300
   Index           =   5
   Left            =   6030
   TabIndex        =   23
   Top             =   4320
   Width           =   585
End
Begin VB.Label Direction
   Alignment       =   2  'Center
   AutoSize        =   -1  'True
   BackColor       =   &H8000000E&
   BackStyle       =   0  'Transparent
   Caption         =   "E"
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   12
      Charset         =   0
      Weight          =   700
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   Height          =   300
   Index           =   4
   Left            =   6240
   TabIndex        =   22
   Top             =   3210
   Width           =   225
End
Begin VB.Label Direction
   Alignment       =   2  'Center
   AutoSize        =   -1  'True
   BackColor       =   &H8000000E&
   BackStyle       =   0  'Transparent
   Caption         =   "ENE"
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   12
      Charset         =   0
      Weight          =   700
      Underline       =   0   'False
      Italic          =   0   'False
```

```
         Strikethrough   =   0   'False
      EndProperty
      Height          =   300
      Index           =   3
      Left            =   6015
      TabIndex        =   21
      Top             =   2085
      Width           =   585
   End
   Begin VB.Label Direction
      Alignment       =   2  'Center
      AutoSize        =   -1  'True
      BackColor       =   &H8000000E&
      BackStyle       =   0  'Transparent
      Caption         =   "NNE"
      BeginProperty Font
         Name            =   "MS Sans Serif"
         Size            =   12
         Charset         =   0
         Weight          =   700
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      Height          =   300
      Index           =   1
      Left            =   4380
      TabIndex        =   20
      Top             =   450
      Width           =   585
   End
   Begin VB.Label Direction
      Alignment       =   2  'Center
      AutoSize        =   -1  'True
      BackColor       =   &H8000000E&
      BackStyle       =   0  'Transparent
      Caption         =   "N"
      BeginProperty Font
         Name            =   "MS Sans Serif"
         Size            =   12
         Charset         =   0
         Weight          =   700
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      Height          =   300
      Index           =   0
      Left            =   3270
      TabIndex        =   19
      Top             =   225
      Width           =   195
   End
End
Begin VB.PictureBox Picture2
   Appearance      =   0  'Flat
   BackColor       =   &H80000005&
   BorderStyle     =   0  'None
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   8.25
      Charset         =   0
      Weight          =   400
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
```

```
EndProperty
ForeColor       =   &H80000008&
Height          =   6255
Left            =   8955
ScaleHeight     =   6255
ScaleWidth      =   2640
TabIndex        =   4
TabStop         =   0   'False
Top             =   15
Width           =   2640
Begin VB.Label Label1
   Alignment       =   1  'Right Justify
   Appearance      =   0  'Flat
   BackColor       =   &H80000005&
   Caption         =   "0"
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   8.25
      Charset         =   0
      Weight          =   400
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H80000008&
   Height          =   255
   Index           =   0
   Left            =   60
   TabIndex        =   37
   Top             =   5880
   Width           =   1200
End
Begin VB.Label Label1
   Alignment       =   1  'Right Justify
   AutoSize        =   -1  'True
   BackColor       =   &H80000009&
   Caption         =   "Label1"
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   8.25
      Charset         =   0
      Weight          =   400
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   Height          =   255
   Index           =   1
   Left            =   60
   TabIndex        =   36
   Top             =   5520
   Width           =   1200
End
Begin VB.Label Label2
   BackColor       =   &H8000000E&
   Caption         =   "Population Scale"
   BeginProperty Font
      Name            =   "Arial Narrow"
      Size            =   12
      Charset         =   0
      Weight          =   700
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   Height          =   285
```

```
    Left            =   960
    TabIndex        =   32
    Top             =   45
    Width           =   1725
 End
 Begin VB.Label Label1
    Alignment       =   1  'Right Justify
    BackColor       =   &H80000009&
    Caption         =   "Label1"
    BeginProperty Font
       Name             =   "MS Sans Serif"
       Size             =   8.25
       Charset          =   0
       Weight           =   400
       Underline        =   0   'False
       Italic           =   0   'False
       Strikethrough    =   0   'False
    EndProperty
    ForeColor       =   &H80000008&
    Height          =   255
    Index           =   15
    Left            =   60
    TabIndex        =   31
    Top             =   480
    Width           =   1200
 End
 Begin VB.Label Label1
    Alignment       =   1  'Right Justify
    BackColor       =   &H80000009&
    Caption         =   "Label1"
    BeginProperty Font
       Name             =   "MS Sans Serif"
       Size             =   8.25
       Charset          =   0
       Weight           =   400
       Underline        =   0   'False
       Italic           =   0   'False
       Strikethrough    =   0   'False
    EndProperty
    ForeColor       =   &H80000008&
    Height          =   255
    Index           =   14
    Left            =   60
    TabIndex        =   17
    Top             =   840
    Width           =   1200
 End
 Begin VB.Shape Shape1
    FillColor       =   &H000000FF&
    FillStyle       =   0  'Solid
    Height          =   375
    Index           =   14
    Left            =   1380
    Top             =   600
    Width           =   1215
 End
 Begin VB.Label Label1
    Alignment       =   1  'Right Justify
    BackColor       =   &H80000009&
    Caption         =   "Label1"
    BeginProperty Font
       Name             =   "MS Sans Serif"
       Size             =   8.25
       Charset          =   0
       Weight           =   400
       Underline        =   0   'False
```

```
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   ForeColor          =   &H80000008&
   Height             =   255
   Index              =   13
   Left               =   60
   TabIndex           =   16
   Top                =   1200
   Width              =   1200
End
Begin VB.Label Label1
   Alignment          =   1  'Right Justify
   BackColor          =   &H80000009&
   Caption            =   "Label1"
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   8.25
      Charset         =   0
      Weight          =   400
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   ForeColor          =   &H80000008&
   Height             =   255
   Index              =   12
   Left               =   60
   TabIndex           =   15
   Top                =   1560
   Width              =   1200
End
Begin VB.Label Label1
   Alignment          =   1  'Right Justify
   BackColor          =   &H80000009&
   Caption            =   "Label1"
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   8.25
      Charset         =   0
      Weight          =   400
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   ForeColor          =   &H80000008&
   Height             =   255
   Index              =   11
   Left               =   60
   TabIndex           =   14
   Top                =   1920
   Width              =   1200
End
Begin VB.Label Label1
   Alignment          =   1  'Right Justify
   BackColor          =   &H80000009&
   Caption            =   "Label1"
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   8.25
      Charset         =   0
      Weight          =   400
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
```

```
      ForeColor       =   &H80000008&
      Height          =   255
      Index           =   10
      Left            =   60
      TabIndex        =   13
      Top             =   2280
      Width           =   1200
   End
   Begin VB.Label Label1
      Alignment       =   1  'Right Justify
      BackColor       =   &H80000009&
      Caption         =   "Label1"
      BeginProperty Font
         Name            =   "MS Sans Serif"
         Size            =   8.25
         Charset         =   0
         Weight          =   400
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H80000008&
      Height          =   255
      Index           =   9
      Left            =   60
      TabIndex        =   12
      Top             =   2640
      Width           =   1200
   End
   Begin VB.Label Label1
      Alignment       =   1  'Right Justify
      BackColor       =   &H80000009&
      Caption         =   "Label1"
      BeginProperty Font
         Name            =   "MS Sans Serif"
         Size            =   8.25
         Charset         =   0
         Weight          =   400
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H80000008&
      Height          =   255
      Index           =   8
      Left            =   60
      TabIndex        =   11
      Top             =   3000
      Width           =   1200
   End
   Begin VB.Label Label1
      Alignment       =   1  'Right Justify
      BackColor       =   &H80000009&
      Caption         =   "Label1"
      BeginProperty Font
         Name            =   "MS Sans Serif"
         Size            =   8.25
         Charset         =   0
         Weight          =   400
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H80000008&
      Height          =   255
      Index           =   7
```

```
   Left            =   60
   TabIndex        =   10
   Top             =   3360
   Width           =   1200
End
Begin VB.Label Label1
   Alignment       =   1  'Right Justify
   BackColor       =   &H80000009&
   Caption         =   "Label1"
   BeginProperty Font
      Name             =   "MS Sans Serif"
      Size             =   8.25
      Charset          =   0
      Weight           =   400
      Underline        =   0   'False
      Italic           =   0   'False
      Strikethrough    =   0   'False
   EndProperty
   ForeColor       =   &H80000008&
   Height          =   255
   Index           =   6
   Left            =   60
   TabIndex        =   9
   Top             =   3720
   Width           =   1200
End
Begin VB.Label Label1
   Alignment       =   1  'Right Justify
   BackColor       =   &H80000009&
   Caption         =   "Label1"
   BeginProperty Font
      Name             =   "MS Sans Serif"
      Size             =   8.25
      Charset          =   0
      Weight           =   400
      Underline        =   0   'False
      Italic           =   0   'False
      Strikethrough    =   0   'False
   EndProperty
   ForeColor       =   &H80000008&
   Height          =   255
   Index           =   5
   Left            =   60
   TabIndex        =   8
   Top             =   4080
   Width           =   1200
End
Begin VB.Label Label1
   Alignment       =   1  'Right Justify
   BackColor       =   &H80000009&
   Caption         =   "Label1"
   BeginProperty Font
      Name             =   "MS Sans Serif"
      Size             =   8.25
      Charset          =   0
      Weight           =   400
      Underline        =   0   'False
      Italic           =   0   'False
      Strikethrough    =   0   'False
   EndProperty
   ForeColor       =   &H80000008&
   Height          =   255
   Index           =   4
   Left            =   60
   TabIndex        =   7
   Top             =   4440
```

```
    Width           =   1200
 End
 Begin VB.Label Label1
    Alignment       =   1  'Right Justify
    BackColor       =   &H80000009&
    Caption         =   "Label1"
    BeginProperty Font
       Name             =   "MS Sans Serif"
       Size             =   8.25
       Charset          =   0
       Weight           =   400
       Underline        =   0   'False
       Italic           =   0   'False
       Strikethrough    =   0   'False
    EndProperty
    ForeColor       =   &H80000008&
    Height          =   255
    Index           =   3
    Left            =   60
    TabIndex        =   6
    Top             =   4800
    Width           =   1200
 End
 Begin VB.Label Label1
    Alignment       =   1  'Right Justify
    BackColor       =   &H80000009&
    Caption         =   "Label1"
    BeginProperty Font
       Name             =   "MS Sans Serif"
       Size             =   8.25
       Charset          =   0
       Weight           =   400
       Underline        =   0   'False
       Italic           =   0   'False
       Strikethrough    =   0   'False
    EndProperty
    ForeColor       =   &H80000008&
    Height          =   255
    Index           =   2
    Left            =   60
    TabIndex        =   5
    Top             =   5160
    Width           =   1200
 End
 Begin VB.Shape Shape1
    FillColor       =   &H008080FF&
    FillStyle       =   0  'Solid
    Height          =   375
    Index           =   13
    Left            =   1380
    Top             =   960
    Width           =   1215
 End
 Begin VB.Shape Shape1
    FillColor       =   &H000080FF&
    FillStyle       =   0  'Solid
    Height          =   375
    Index           =   12
    Left            =   1380
    Top             =   1320
    Width           =   1215
 End
 Begin VB.Shape Shape1
    FillColor       =   &H00004080&
    FillStyle       =   0  'Solid
    Height          =   375
```

```
   Index           =   11
   Left            =   1380
   Top             =   1680
   Width           =   1215
End
Begin VB.Shape Shape1
   FillColor       =   &H00FFFFFF&
   FillStyle       =   0  'Solid
   Height          =   375
   Index           =   10
   Left            =   1380
   Top             =   2040
   Width           =   1215
End
Begin VB.Shape Shape1
   FillColor       =   &H0000C0C0&
   FillStyle       =   0  'Solid
   Height          =   375
   Index           =   9
   Left            =   1380
   Top             =   2400
   Width           =   1215
End
Begin VB.Shape Shape1
   FillColor       =   &H0000FF00&
   FillStyle       =   0  'Solid
   Height          =   375
   Index           =   8
   Left            =   1380
   Top             =   2760
   Width           =   1215
End
Begin VB.Shape Shape1
   FillColor       =   &H0000C000&
   FillStyle       =   0  'Solid
   Height          =   375
   Index           =   7
   Left            =   1380
   Top             =   3120
   Width           =   1215
End
Begin VB.Shape Shape1
   FillColor       =   &H00FFFF00&
   FillStyle       =   0  'Solid
   Height          =   375
   Index           =   6
   Left            =   1380
   Top             =   3480
   Width           =   1215
End
Begin VB.Shape Shape1
   FillColor       =   &H00C0C000&
   FillStyle       =   0  'Solid
   Height          =   375
   Index           =   5
   Left            =   1380
   Top             =   3840
   Width           =   1215
End
Begin VB.Shape Shape1
   FillColor       =   &H00FF0000&
   FillStyle       =   0  'Solid
   Height          =   375
   Index           =   4
   Left            =   1380
   Top             =   4200
```

```
      Width           =   1215
   End
   Begin VB.Shape Shape1
      FillColor       =   &H00800000&
      FillStyle       =   0  'Solid
      Height          =   375
      Index           =   3
      Left            =   1380
      Top             =   4545
      Width           =   1215
   End
   Begin VB.Shape Shape1
      FillColor       =   &H00C000C0&
      FillStyle       =   0  'Solid
      Height          =   375
      Index           =   2
      Left            =   1380
      Top             =   4920
      Width           =   1215
   End
   Begin VB.Shape Shape1
      FillColor       =   &H00800080&
      FillStyle       =   0  'Solid
      Height          =   375
      Index           =   1
      Left            =   1380
      Top             =   5280
      Width           =   1215
   End
   Begin VB.Shape Shape1
      BackColor       =   &H00FFFFFF&
      BorderColor     =   &H00000000&
      FillColor       =   &H00FFFFFF&
      FillStyle       =   0  'Solid
      Height          =   375
      Index           =   0
      Left            =   1380
      Top             =   5640
      Width           =   1215
   End
End
Begin VB.Label lblTo
   BackColor       =   &H80000005&
   Caption         =   "To"
   Height          =   210
   Left            =   855
   TabIndex        =   43
   Top             =   345
   Visible         =   0   'False
   Width           =   690
End
Begin VB.Label lblFrom
   BackColor       =   &H80000005&
   Caption         =   "From"
   Height          =   210
   Left            =   120
   TabIndex        =   42
   Top             =   345
   Visible         =   0   'False
   Width           =   690
End
Begin VB.Label DistanceLabel
   BackColor       =   &H80000009&
   Caption         =   "DistanceLabel"
   BeginProperty Font
      Name            =   "Arial Narrow"
```

```
         Size            =   12
         Charset         =   0
         Weight          =   700
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      Height          =   285
      Left            =   15
      TabIndex        =   35
      Top             =   45
      Width           =   2160
   End
   Begin VB.Label TitleBlock
      Alignment       =   2  'Center
      AutoSize        =   -1  'True
      BackColor       =   &H80000009&
      Caption         =   "Blank Rosette"
      BeginProperty Font
         Name            =   "Arial"
         Size            =   15.75
         Charset         =   0
         Weight          =   700
         Underline       =   -1  'True
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H80000008&
      Height          =   360
      Left            =   2430
      TabIndex        =   18
      Top             =   -30
      Width           =   5835
   End
End
Attribute VB_Name = "frmRosette"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
'*****************************************************************************
'  Module:  frmRosette
'
'  Project: SECPOP
'
'  Description: This module contains the subroutines necessary to output a
'               MACCS input file in:
'                   MACCS Compatible Format
'                   CSV format
'                   Screen
'                   Hard Copy
'
'  Modification History
'  Date      By   Description
'  --------  ---  ------------------------------------------------------------
'  20020131  cwm  Updated Rosette, fixed some format problems
'  20020227  cwm  Fixed Rosette empty start error
'
'*****************************************************************************

Option Explicit

Dim MaxRow As Integer
Dim MaxCol As Integer
Dim i As Integer
```

```
Dim j As Integer
Dim k As Integer
Dim GridArray() As String
Dim Sink As New GridClass
Public blnInputDataChanged As Boolean ' indicates that the user has changed
                                      ' something with the problem or site
                                      ' data since the last display of
                                      ' the rosette

Private Sub CancelButton_click()
    Unload Me
End Sub

Private Sub cmdPrint_Click()

    Dim iCopy As Integer

    CancelButton.Visible = False
    cmdprint.Visible = False
    cmdSave.Visible = False
    TDBGrid1.Height = 286.5


    frmRosette.Refresh
    Set pctCapture.Picture = Nothing
    Set pctCapture.Picture = CaptureClient(Me)

    frmMain.dlgSecpop.Flags = cdlPDHidePrintToFile
    frmMain.dlgSecpop.CancelError = True

    On Error Resume Next

    frmMain.dlgSecpop.ShowPrinter
    iCopy = frmMain.dlgSecpop.Copies

    If Err.number <> cdlCancel Then    ' user did not click on cancel
        For i = 1 To iCopy
            PrintPictureToFitPage Printer, pctCapture.Picture
            Printer.EndDoc
        Next i
    End If

    On Error GoTo 0

    CancelButton.Visible = True
    cmdprint.Visible = True
    cmdSave.Visible = True
    TDBGrid1.Height = 256.5


End Sub

Private Sub cmdSave_Click()

    CancelButton.Visible = False
    cmdprint.Visible = False
    cmdSave.Visible = False
    TDBGrid1.Height = 286.5


    On Error Resume Next

    With frmMain.dlgSecpop
        .Filter = "Bitmap Files (*.bmp)|*bmp|All files|*.*"
        .FilterIndex = 1
        .DialogTitle = "Save File As"
```

```
        .Filename = ""
        .Flags = cdlOFNOverwritePrompt
        .DefaultExt = "bmp"
        .CancelError = True

        If InitOutputFilePathChanged = True Then
            .InitDir = frmSetup.txtOutput_path.text
            InitOutputFilePathChanged = False
        Else
            .InitDir = InitOutputFilePath
        End If
        .ShowSave
    End With

    If Err.number <> cdlCancel Then    ' user did not click on cancel

        'reset init dir

        InitOutputFilePath = Left(frmMain.dlgSecpop.Filename, _
                                  (InStrRev(frmMain.dlgSecpop.Filename, "\") - 1))

        frmRosette.Refresh
        Set pctCapture.Picture = Nothing
        Set pctCapture.Picture = CaptureClient(Me)
        SavePicture pctCapture.Image, frmMain.dlgSecpop.Filename

    End If

    On Error GoTo 0

    CancelButton.Visible = True
    cmdprint.Visible = True
    cmdSave.Visible = True
    TDBGrid1.Height = 256.5

End Sub

Private Sub Form_Activate()

    Dim row As Long, col As Integer
    Dim C As TrueDBGrid70.Column
    Dim Cols As TrueDBGrid70.Columns

'   build rosette, but only if a calculation has been done subsequent
'   to the user changing any data on the forms.

    If frmRosette.blnInputDataChanged = True Then
        For i = 0 To number_of_segments
            For j = 0 To max_number_of_radii
'            sector_population(number_of_segments, max_number_of_radii)
                sector_population(i, j) = 0
            Next j
        Next i
    End If

    Call Rosette

'   build grid

    MaxRow = number_of_radii
    MaxCol = 2
    col = 2
    row = number_of_radii

    ' Initialize storage
    Sink.SetDims row - 1, col
```

```
    For i = 1 To number_of_radii - 1
          Sink.Value(i - 1, 0) = Format(radial_distance(i), "  ###.0 ")
          Sink.Value(i - 1, 1) = Format(radial_distance(i + 1), "  ###.0 ")
'          Sink.Value(1, i) = Str(region_index(i, j))
'          GridArray(i, j) = Str(region_index(i, j))
    Next i

'    For i = 1 To number_of_radii - 1
'        List1(0).List(i - 1) = Format(radial_distance(i), "  ###.0 ")
'        List1(1).List(i - 1) = Format(radial_distance(i + 1), "  ###.0 ")
'    Next i

    ' Make sure no events are fired for the grid yet
    Sink.Attach Nothing

    Set Cols = TDBGrid1.Columns

    ' Remove all columns
    While Cols.Count
        Cols.Remove 0
    Wend
    TDBGrid1.ReBind

    ' Add columns to the grid
    For col = 0 To 1
        Set C = Cols.Add(col)
        C.Visible = True
        C.Width = 40
        If col = 0 Then
            C.caption = "From"
        Else
            C.caption = "To"
        End If
    Next col

    TDBGrid1.Columns(0).Locked = True
    TDBGrid1.Columns(1).Locked = True

    TDBGrid1.RecordSelectors = False
    TDBGrid1.ScrollBars = dbgAutomatic

    'Initialize current cell position to upper left corner:

'    TDBGrid1.row = 0
'    TDBGrid1.col = 0

    ' Initialise the class
    Sink.Attach TDBGrid1

    ' Reinitialize the grid
    TDBGrid1.Bookmark = Null
    TDBGrid1.ReBind

    ' Calibrate the VScroll bar
    TDBGrid1.ApproxCount = Sink.RowCount

    TDBGrid1.Width = 92

End Sub


Private Sub Rosette()
    'This sub is the heart of the SecPop Rosette drawing routine.

    Dim Rad As Double            'radial distance normalized to pixels
```

```
    Dim i As Integer            'loop counter
    Dim j As Integer            'loop counter
    Dim k As Integer            'index used to invert lookup of populations.
    Dim Pie_start As Double     'Start location of segment in radians
    Dim Pie_end As Double       'End location of segment in radians
    Dim PopColors(14) As Long   'Vector containing the scale of colors
    Dim Paint_Index As Integer     'Index working with PopColors vector to output color
    Dim Max_Pop As Long         'Maximum population
    Dim Max_Distance As Double  'Equivalent to the distance of the largest radius
    Dim Scale_Interval As Long  'Interval between population increments

    Const side As Double = 225  'Number of pixels in 1/2 the side of the rosette picture box
'    Const side As Double = 270  'Number of pixels in 1/2 the side of the rosette picture box
    Const pi As Double = 3.14159265359
    Const CircleEnd As Double = -2 * pi

    Call SetScale(Scale_Interval, Max_Pop)      'Returns parameters used to set fill color
    Call DistancesList                          'Outputs list of radius distances on right
    Call Legend(PopColors, Scale_Interval)      'Outputs Population Scale Legend
    TitleBlock.caption = frmSite_Data.txtSite_Name.text

    'Begin drawing Rosette Segments.  Program works from the outside in,
    'drawing pie charts at progressively smaller radii.
    Picture1.ScaleMode = vbPixels
    For i = number_of_radii To 1 Step -1
        Pie_start = (0.25 + 1 / 32) * 2 * pi
        Rad = i / number_of_radii
        For j = 1 To number_of_segments
            'Set Segment Fill Color
            k = 17 - j
            Paint_Index = Int(15 * sector_population(k, i) / Max_Pop)
            If Paint_Index = 15 Then Paint_Index = 14
            If i = 1 Then Paint_Index = 0
            Picture1.FillColor = PopColors(Paint_Index)
            Picture1.FillStyle = 0

            'Calculate Radius normalized to pixels
            Max_Distance = radial_distance(number_of_radii)
            Rad = (0.9 * side * radial_distance(i) / Max_Distance) * 0.9
'            Rad = 0.9 * side * radial_distance(i) / Max_Distance

            'Set begin/end of pie segment and draw.
            Pie_end = (1 / number_of_segments)
            Let Pie_end = Pie_start - Pie_end * CircleEnd
            If Pie_end > 2 * pi Then Pie_end = Pie_end - 2 * pi
            Picture1.Circle (side, side), Rad, 255, -Pie_start, -Pie_end
            Pie_start = Pie_end
        Next j
    Next i

End Sub

Sub DistancesList()
    Dim i As Integer
    Dim ir As Integer

    If frmProblem_Data.optMiles.Value = True Then
        DistanceLabel.caption = "Radial Distances in " & "Mi"
    Else
        DistanceLabel.caption = "Radial Distances in " & "Km"
    End If

    If number_of_radii <= 5 Then
        ir = 5
    Else
        ir = number_of_radii
```

```
    End If

'    List1(0).Height = 13.5 * (ir - 1)
'    List1(1).Height = 13.5 * (ir - 1)
'
'    For i = 1 To number_of_radii - 1
'        List1(0).List(i - 1) = Format(radial_distance(i), "  ###.0 ")
'        List1(1).List(i - 1) = Format(radial_distance(i + 1), "  ###.0 ")
'    Next i

End Sub


Sub SetScale(Scale_Interval, Max_Pop)

    Dim i As Integer
    Dim j As Integer

    For i = 1 To number_of_segments
        For j = 1 To number_of_radii
            If Max_Pop < sector_population(i, j) Then
                Max_Pop = sector_population(i, j)
            End If
        Next j
    Next i

    If Max_Pop <= 15 Then
        Scale_Interval = 1
        Max_Pop = 15
    ElseIf Max_Pop <= 150 Then
        Scale_Interval = 10
        Max_Pop = 150
    ElseIf Max_Pop <= 1500 Then
        Scale_Interval = 100
        Max_Pop = 1500
    Else
        Scale_Interval = 7.5 * (10 ^ (Int(Log(Max_Pop) / Log(10)) - 1))
        Max_Pop = Scale_Interval * (Int(Max_Pop / Scale_Interval) + 1)
        Scale_Interval = Max_Pop / 15
    End If

End Sub

Sub Legend(PopColors, Scale_Interval)

    Dim A As Long
    Dim i As Integer

    PopColors(0) = RGB(255, 255, 255)
    PopColors(1) = RGB(170, 0, 255)
    PopColors(2) = RGB(85, 0, 255)
    PopColors(3) = RGB(0, 0, 255)
    PopColors(4) = RGB(0, 85, 255)
    PopColors(5) = RGB(0, 170, 255)
    PopColors(6) = RGB(0, 255, 255)
    PopColors(7) = RGB(0, 255, 170)
    PopColors(8) = RGB(0, 255, 85)
    PopColors(9) = RGB(0, 255, 0)
    PopColors(10) = RGB(85, 255, 0)
    PopColors(11) = RGB(170, 255, 0)
    PopColors(12) = RGB(255, 255, 0)
    PopColors(13) = RGB(255, 170, 0)
    PopColors(14) = RGB(255, 85, 0)

    For i = 0 To 14
        Shape1(i).FillColor = PopColors(i)
```

```
    Next i

    A = 0
    For i = 0 To 15
'         Label1(i).caption = A
        Label1(i).caption = Format(A, "###,###,###,###,###")
        A = A + Scale_Interval
    Next i

End Sub

Private Sub Form_Load()

    Me.Top = 10
    Me.Left = 10

End Sub
```

H.31 Visual Basic Form Module: Setup.frm

```
VERSION 5.00
Begin VB.Form frmSetup
   Appearance      =   0  'Flat
   BackColor       =   &H80000004&
   BorderStyle     =   1  'Fixed Single
   Caption         =   "Setup"
   ClientHeight    =   5820
   ClientLeft      =   210
   ClientTop       =   330
   ClientWidth     =   9585
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   8.25
      Charset         =   0
      Weight          =   700
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H00000000&
   Icon            =   "Setup.frx":0000
   MaxButton       =   0   'False
   MDIChild        =   -1  'True
   MinButton       =   0   'False
   PaletteMode     =   1  'UseZOrder
   ScaleHeight     =   5820
   ScaleWidth      =   9585
   Begin VB.Frame fraCensusFile
      Caption         =   "Databases:"
      Height          =   1620
      Left            =   4890
      TabIndex        =   21
      Top             =   1695
      Width           =   4560
      Begin VB.TextBox Text4
         Appearance      =   0  'Flat
         BackColor       =   &H80000004&
         BorderStyle     =   0  'None
         ForeColor       =   &H00000000&
         Height          =   200
         Left            =   435
         Locked          =   -1  'True
         TabIndex        =   27
         Text            =   "Economic:"
         Top             =   930
         Width           =   900
      End
      Begin VB.TextBox Text3
         Appearance      =   0  'Flat
         BackColor       =   &H80000004&
         BorderStyle     =   0  'None
         BeginProperty Font
            Name            =   "MS Sans Serif"
            Size            =   8.25
            Charset         =   0
            Weight          =   400
            Underline       =   0   'False
            Italic          =   0   'False
            Strikethrough   =   0   'False
         EndProperty
         ForeColor       =   &H00000000&
         Height          =   270
```

```
   Left            =   660
   Locked          =   -1  'True
   TabIndex        =   26
   Text            =   "Choose one:"
   Top             =   1215
   Width           =   900
End
Begin VB.ComboBox cboCounty_Database
   Height          =   315
   Left            =   1665
   Style           =   2  'Dropdown List
   TabIndex        =   6
   Top             =   1170
   Width           =   2250
End
Begin VB.TextBox txtCounty_Database
   Appearance      =   0  'Flat
   BackColor       =   &H80000004&
   BorderStyle     =   0  'None
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   8.25
      Charset         =   0
      Weight          =   400
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H00000000&
   Height          =   200
   Left            =   1635
   Locked          =   -1  'True
   TabIndex        =   25
   Text            =   "Database"
   Top             =   930
   Width           =   2340
End
Begin VB.TextBox Text1
   Appearance      =   0  'Flat
   BackColor       =   &H80000004&
   BorderStyle     =   0  'None
   ForeColor       =   &H00000000&
   Height          =   200
   Left            =   420
   Locked          =   -1  'True
   TabIndex        =   24
   Text            =   "Census:"
   Top             =   270
   Width           =   900
End
Begin VB.TextBox txtCensus_Database
   Appearance      =   0  'Flat
   BackColor       =   &H80000004&
   BorderStyle     =   0  'None
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   8.25
      Charset         =   0
      Weight          =   400
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H00000000&
   Height          =   200
   Left            =   1650
```

```
      Locked          =   -1  'True
      TabIndex        =   23
      Text            =   "Database"
      Top             =   270
      Width           =   2340
   End
   Begin VB.ComboBox cboCensus_Database
      Height          =   315
      Left            =   1650
      Style           =   2  'Dropdown List
      TabIndex        =   5
      Top             =   510
      Width           =   2250
   End
   Begin VB.TextBox txtChoose_Database
      Appearance      =   0  'Flat
      BackColor       =   &H80000004&
      BorderStyle     =   0  'None
      BeginProperty Font
         Name            =   "MS Sans Serif"
         Size            =   8.25
         Charset         =   0
         Weight          =   400
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H00000000&
      Height          =   270
      Left            =   645
      Locked          =   -1  'True
      TabIndex        =   22
      Text            =   "Choose one:"
      Top             =   570
      Width           =   900
   End
End
Begin VB.Frame fraData_Path
   BackColor       =   &H80000004&
   Caption         =   "&Location of Databases:"
   Height          =   1335
   Left            =   4830
   TabIndex        =   18
   Top             =   255
   Width           =   4575
   Begin VB.ComboBox cboCountyDatabase
      Height          =   315
      Left            =   870
      Style           =   2  'Dropdown List
      TabIndex        =   3
      Top             =   885
      Visible         =   0   'False
      Width           =   2250
   End
   Begin VB.CommandButton cmdData_Path
      Caption         =   "Browse..."
      BeginProperty Font
         Name            =   "MS Sans Serif"
         Size            =   8.25
         Charset         =   0
         Weight          =   400
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      Height          =   375
```

```
      Left            =   3360
      TabIndex        =   4
      Top             =   855
      Width           =   1095
   End
   Begin VB.TextBox txtData_path
      Appearance      =   0  'Flat
      BackColor       =   &H80000004&
      BorderStyle     =   0  'None
      BeginProperty Font
         Name            =   "MS Sans Serif"
         Size            =   8.25
         Charset         =   0
         Weight          =   400
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H00000000&
      Height          =   270
      Left            =   150
      Locked          =   -1  'True
      TabIndex        =   20
      Text            =   "CENSUS"
      Top             =   495
      Width           =   3780
   End
End
Begin VB.Frame fraOutput_Path
   BackColor       =   &H80000004&
   Caption         =   "Default &Output File Path:"
   Height          =   1335
   Left            =   90
   TabIndex        =   17
   Top             =   3450
   Width           =   4575
   Begin VB.CommandButton cmdOutput_Path
      Caption         =   "Browse..."
      BeginProperty Font
         Name            =   "MS Sans Serif"
         Size            =   8.25
         Charset         =   0
         Weight          =   400
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      Height          =   375
      Left            =   3375
      TabIndex        =   2
      Top             =   795
      Width           =   1095
   End
   Begin VB.TextBox txtOutput_path
      Appearance      =   0  'Flat
      BackColor       =   &H80000004&
      BorderStyle     =   0  'None
      BeginProperty Font
         Name            =   "MS Sans Serif"
         Size            =   8.25
         Charset         =   0
         Weight          =   400
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
```

```
      ForeColor       =   &H00000000&
      Height          =   360
      Left            =   120
      Locked          =   -1  'True
      TabIndex        =   19
      Text            =   "SITES"
      Top             =   480
      Width           =   3780
   End
End
Begin VB.Frame fraProb_path
   BackColor       =   &H80000004&
   Caption         =   "Default &Problem File Path:"
   Height          =   1620
   Left            =   105
   TabIndex        =   15
   Top             =   1695
   Width           =   4575
   Begin VB.CommandButton cmdProb_Path
      Caption         =   "Browse..."
      BeginProperty Font
         Name            =   "MS Sans Serif"
         Size            =   8.25
         Charset         =   0
         Weight          =   400
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      Height          =   375
      Left            =   3360
      TabIndex        =   1
      Top             =   840
      Width           =   1095
   End
   Begin VB.TextBox txtProblem_path
      Appearance      =   0  'Flat
      BackColor       =   &H80000004&
      BorderStyle     =   0  'None
      BeginProperty Font
         Name            =   "MS Sans Serif"
         Size            =   8.25
         Charset         =   0
         Weight          =   400
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H00000000&
      Height          =   270
      Left            =   135
      Locked          =   -1  'True
      TabIndex        =   16
      Text            =   "SITES"
      Top             =   480
      Width           =   3780
   End
End
Begin VB.Frame fraSite_Path
   BackColor       =   &H80000004&
   Caption         =   "&Default Site File Path:"
   Height          =   1335
   Left            =   120
   TabIndex        =   14
   Top             =   255
   Width           =   4575
```

```
    Begin VB.TextBox txtSite_Path
       Appearance      =   0  'Flat
       BackColor       =   &H80000004&
       BorderStyle     =   0  'None
       BeginProperty Font
          Name             =   "MS Sans Serif"
          Size             =   8.25
          Charset          =   0
          Weight           =   400
          Underline        =   0   'False
          Italic           =   0   'False
          Strikethrough    =   0   'False
       EndProperty
       ForeColor       =   &H00000000&
       Height          =   195
       Left            =   120
       Locked          =   -1  'True
       TabIndex        =   11
       Text            =   "SITES"
       Top             =   495
       Width           =   3780
    End
    Begin VB.CommandButton cmdSite_Path
       Caption         =   "Browse..."
       BeginProperty Font
          Name             =   "MS Sans Serif"
          Size             =   8.25
          Charset          =   0
          Weight           =   400
          Underline        =   0   'False
          Italic           =   0   'False
          Strikethrough    =   0   'False
       EndProperty
       Height          =   375
       Left            =   3360
       TabIndex        =   0
       Top             =   855
       Width           =   1095
    End
End
Begin VB.CommandButton cmdMapplan_Setup
   Appearance      =   0  'Flat
   BackColor       =   &H00E0E0E0&
   Caption         =   "M&apPlan"
   Enabled         =   0   'False
   Height          =   480
   Left            =   1275
   TabIndex        =   12
   TabStop         =   0   'False
   Top             =   5250
   Width           =   1728
End
Begin VB.CommandButton cmdSave_Settings
   Appearance      =   0  'Flat
   BackColor       =   &H00E0E0E0&
   Caption         =   "&Save Changes"
   Height          =   480
   Left            =   3525
   TabIndex        =   9
   Top             =   5250
   Width           =   1728
End
Begin VB.CommandButton cmdCancel
   Appearance      =   0  'Flat
   BackColor       =   &H00E0E0E0&
   Cancel          =   -1  'True
```

```
      Caption         =   "&Close"
      Default         =   -1  'True
      Height          =   480
      Left            =   5925
      TabIndex        =   10
      Top             =   5250
      Width           =   1728
   End
   Begin VB.Frame fraOutput_Format
      BackColor       =   &H80000004&
      Caption         =   "Site-Specific Output File Format:"
      Enabled         =   0   'False
      Height          =   1335
      Left            =   4920
      TabIndex        =   13
      Top             =   3450
      Visible         =   0   'False
      Width           =   4620
      Begin VB.OptionButton optCSV
         Caption         =   "CS&V"
         Enabled         =   0   'False
         Height          =   240
         Left            =   2400
         TabIndex        =   8
         TabStop         =   0   'False
         Top             =   600
         Width           =   756
      End
      Begin VB.OptionButton optMACCS
         Caption         =   "&MACCS"
         Enabled         =   0   'False
         Height          =   240
         Left            =   990
         TabIndex        =   7
         Top             =   600
         Value           =   -1  'True
         Width           =   972
      End
   End
End
Attribute VB_Name = "frmSetup"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Private Sub cboCensus_Database_Click()

    txtCensus_Database.text = cboCensus_Database.text
    frmMain.caption = MainTitle & "  (" & cboCensus_Database.text & ")"

End Sub

Private Sub cboCounty_Database_Click()
    txtCounty_Database.text = cboCounty_Database.text
End Sub

'This form allows the user to set certain paths which will be remembered
'by the program until they are changed.  These paths include input and
'output directories, among other things.  Setup also allows the user to
'specify which file format they would like to use with their output
'data files - MACCS input file or comma seperated variable formats.

Private Sub cmdCancel_Click()
```

```
    'If the user selects the cancel button from setup, the setup form is
    'hidden, but the changes are retained until the program is exited.

    frmSetup.Hide

End Sub

Private Sub cmdData_Path_Click()

    Dim strFolderName As String
    strFolderName = GetFolderName(strFolderName)
    If strFolderName <> "" Then
        txtData_path.text = strFolderName
    End If

End Sub

Private Sub cmdMapplan_Setup_Click()

'    RUN "mp_setup.exe"

End Sub

Private Sub cmdOutput_Path_Click()

    Dim strFolderName As String
'    txtOutput_path.text = GetFolderName(strFolderName)

    strFolderName = GetFolderName(strFolderName)
    If strFolderName <> "" Then
        txtOutput_path.text = strFolderName
    End If

End Sub

Private Sub cmdProb_Path_Click()

'    Dim strFolderName As String
'    txtProblem_path.text = GetFolderName(strFolderName)

    Dim strFolderName As String

    strFolderName = GetFolderName(strFolderName)
    If strFolderName <> "" Then
        txtProblem_path.text = strFolderName
    End If

End Sub

Private Sub cmdSave_Settings_Click()

    'This routine saves setup data to a file so that the program has
    'access to it every time the program is run.

    Dim filenum As Integer
    Dim strFileName As String

    'Locate first free file handle.

    filenum = FreeFile

    'Open file (called secpop.cfg) and output all of the data to it.
    strFileName = App.path & "\SECPOP.CFG"
'    Open "SECPOP00.CFG" For Output As #filenum
    Open strFileName For Output As #filenum
    Print #filenum, frmSetup.txtSite_Path.text
```

```
    Print #filenum, frmSetup.txtProblem_path.text
    Print #filenum, frmSetup.txtOutput_path.text
    Print #filenum, frmSetup.txtData_path.text
'    If frmSetup.optMACCS.Value = True Then
        Print #filenum, "MACCS"
'    Else
'        Print #filenum, "CSV"
'    End If
    Print #filenum, frmSetup.txtCensus_Database.text
    Print #filenum, frmSetup.txtCounty_Database.text
    Close #filenum

End Sub

Private Sub cmdSite_Path_Click()

    Dim strFolderName As String

    strFolderName = GetFolderName(strFolderName)
    If strFolderName <> "" Then
        txtSite_Path.text = strFolderName
    End If

'    txtSite_Path.text = GetFolderName
'    txtSite_Path.text = GetFolderName(strFolderName)

End Sub

Private Sub Frame1_DragDrop(Source As Control, X As Single, Y As Single)

End Sub

Private Sub Combo1_Change()

End Sub

Private Sub Label1_Click()

End Sub

Private Sub Text2_Change()

End Sub

Private Sub Form_Load()

    Dim TempString As String
    Dim filenum As Integer

'    frmDisclaimer.Show
    On Local Error GoTo configerr

    'Check and see if config file exists.
    TempString = App.path & "\SECPOP.CFG"
    If Len(Dir$(TempString)) <> 0 Then
'    If Len(Dir$("SECPOP.CFG")) <> 0 Then

        'Open file and read in the data from it.

        filenum = FreeFile
'        Open "SECPOP.CFG" For Input As filenum
        Open TempString For Input As filenum
        Input #filenum, TempString
        frmSetup.txtSite_Path.text = TempString
        Input #filenum, TempString
        frmSetup.txtProblem_path.text = TempString
```

```
        Input #filenum, TempString
        frmSetup.txtOutput_path.text = TempString
        Input #filenum, TempString
        frmSetup.txtData_path.text = TempString
        Input #filenum, TempString
'        If (TempString = "MACCS") Then
'            frmSetup.optMACCS.Value = True
'        Else
'            frmSetup.optCSV.Value = True
'        End If
        Input #filenum, TempString
        frmSetup.txtCensus_Database.text = TempString
        Input #filenum, TempString
        frmSetup.txtCounty_Database.text = TempString
        Close #filenum

    'else if config file doesn't exist use defaults.

    Else

        'Defaults set in form.

    End If

    'Turn off error handler.

    On Local Error GoTo 0

configerr:

    'If config file format is screwed up use defaults.

    Close filenum
    Exit Sub

End Sub

Private Sub txtData_path_KeyPress(keyascii As Integer)

    'Convert all input to upper case.

    Dim C As String

    C = Chr$(keyascii)

    keyascii = Asc(UCase$(C$))

End Sub

Private Sub txtOutput_path_Change()

    InitOutputFilePath = frmSetup.txtOutput_path
    InitOutputFilePathChanged = True

End Sub

Private Sub txtOutput_path_KeyPress(keyascii As Integer)

    'Convert all input to upper case.

    Dim C As String

    C = Chr$(keyascii)

    keyascii = Asc(UCase$(C$))
```

```
    'set up default dirs

    InitOutputFilePath = frmSetup.txtOutput_path

End Sub

Private Sub txtProblem_path_Change()

    InitProblemFilePath = txtProblem_path.text
    InitProblemFilePathChanged = True

End Sub

Private Sub txtProblem_path_KeyPress(keyascii As Integer)

    'Convert all input to upper case.

    Dim C As String

    C = Chr$(keyascii)

    keyascii = Asc(UCase$(C$))

    InitProblemFilePath = frmSetup.txtProblem_path

End Sub

Private Sub lblSite_Path_KeyPress(keyascii As Integer)

    'Convert all input to upper case.

    Dim C As String

    C = Chr$(keyascii)

    keyascii = Asc(UCase$(C$))

End Sub

'Public Function GetFolderName() As String
Public Function GetFolderName(strFileName As String) As String

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Copyright ©1996-2002 VBnet, Randy Birch, All Rights Reserved.
' Some pages may also contain other copyrights by the author.
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' You are free to use this code within your own applications,
' but you are expressly forbidden from selling or otherwise
' distributing this source code without prior written consent.
' This includes both posting free demo projects made from this
' code as well as reproducing the code in text or html format.
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

    Dim bi As BROWSEINFO
    Dim pidl As Long
    Dim path As String
    Dim pos As Integer

   'Fill the BROWSEINFO structure with the
   'needed data. To accomodate comments, the
   'With/End With sytax has not been used, though
   'it should be your 'final' version.

   'hwnd of the window that receives messages
   'from the call. Can be your application
   'or the handle from GetDesktopWindow().
```

```
    bi.hOwner = Me.hWnd

    'Pointer to the item identifier list specifying
    'the location of the "root" folder to browse from.
    'If NULL, the desktop folder is used.
    bi.pidlRoot = 0&

    'message to be displayed in the Browse dialog
    bi.lpszTitle = "Select your Windows\System\ directory"

    'the type of folder to return.
    bi.ulFlags = BIF_RETURNONLYFSDIRS
'    bi.ulFlags = BIF_RETURNONLYFSDIRS + BIF_USENEWUI

    'show the browse for folders dialog
    pidl = SHBrowseForFolder(bi)

    'the dialog has closed, so parse & display the
    'user's returned folder selection contained in pidl
    path = Space$(MAX_PATH)

    If SHGetPathFromIDList(ByVal pidl, ByVal path) Then
       pos = InStr(path, Chr$(0))
       GetFolderName = Left(path, pos - 1)
    End If

    Call CoTaskMemFree(pidl)

End Function

Private Sub txtSite_Path_Change()

    'set up default dirs

    InitSiteFilePath = frmSetup.txtSite_Path
    InitSiteFilePathChanged = True

End Sub
```

H.32 Visual Basic Form Module: Siteform.frm

```
VERSION 5.00
Begin VB.Form frmSite_Data
   Appearance      =   0  'Flat
   BackColor       =   &H80000004&
   BorderStyle     =   1  'Fixed Single
   ClientHeight    =   5820
   ClientLeft      =   2115
   ClientTop       =   2625
   ClientWidth     =   9645
   BeginProperty Font
      Name            =   "MS Sans Serif"
      Size            =   8.25
      Charset         =   0
      Weight          =   700
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H00000000&
   Icon            =   "Siteform.frx":0000
   LockControls    =   -1  'True
   MaxButton       =   0   'False
   MDIChild        =   -1  'True
   MinButton       =   0   'False
```

```
PaletteMode     =   1  'UseZOrder
ScaleHeight     =   5820
ScaleWidth      =   9645
Begin VB.Frame Frame5
   Caption          =   "Site &Name:"
   Height           =   855
   Left             =   1260
   TabIndex         =   15
   Top              =   225
   Width            =   6735
   Begin VB.TextBox txtSite_Name
      BackColor        =   &H80000004&
      ForeColor        =   &H00000000&
      Height           =   360
      Left             =   150
      MultiLine        =   -1  'True
      TabIndex         =   0
      Top              =   330
      Width            =   6450
   End
End
Begin VB.Frame Frame4
   BackColor        =   &H80000004&
   Caption          =   "Site Coordinates:"
   Height           =   1935
   Left             =   1245
   TabIndex         =   14
   Top              =   1305
   Width            =   6735
   Begin VB.TextBox txtLongitude_Seconds
      BackColor        =   &H80000004&
      ForeColor        =   &H00000000&
      Height           =   285
      Left             =   4875
      TabIndex         =   6
      Top              =   1335
      Width            =   540
   End
   Begin VB.TextBox txtLongitude_Minutes
      BackColor        =   &H80000004&
      ForeColor        =   &H00000000&
      Height           =   285
      Left             =   4875
      TabIndex         =   5
      Top              =   1005
      Width            =   540
   End
   Begin VB.TextBox txtLongitude_Degrees
      BackColor        =   &H80000004&
      ForeColor        =   &H00000000&
      Height           =   285
      Left             =   4875
      TabIndex         =   4
      Top              =   645
      Width            =   555
   End
   Begin VB.TextBox txtLatitude_Seconds
      BackColor        =   &H80000004&
      ForeColor        =   &H00000000&
      Height           =   285
      Left             =   2160
      TabIndex         =   3
      Top              =   1335
      Width            =   540
   End
   Begin VB.TextBox txtLatitude_Minutes
```

```
    BackColor       =   &H80000004&
    ForeColor       =   &H00000000&
    Height          =   285
    Left            =   2160
    TabIndex        =   2
    Top             =   1005
    Width           =   540
End
Begin VB.TextBox txtLatitude_Degrees
    BackColor       =   &H80000004&
    ForeColor       =   &H00000000&
    Height          =   285
    Left            =   2160
    TabIndex        =   1
    Top             =   645
    Width           =   540
End
Begin VB.Label Label2
    Caption         =   "Lon&gitude"
    Height          =   225
    Left            =   3510
    TabIndex        =   23
    Top             =   375
    Width           =   1290
End
Begin VB.Label Label7
    Appearance      =   0  'Flat
    BackColor       =   &H80000004&
    Caption         =   "Seconds"
    ForeColor       =   &H00000000&
    Height          =   240
    Left            =   3900
    TabIndex        =   22
    Top             =   1350
    Width           =   750
End
Begin VB.Label Label6
    Appearance      =   0  'Flat
    BackColor       =   &H80000004&
    Caption         =   "Minutes"
    ForeColor       =   &H00000000&
    Height          =   240
    Left            =   3900
    TabIndex        =   21
    Top             =   1020
    Width           =   750
End
Begin VB.Label Label5
    Appearance      =   0  'Flat
    BackColor       =   &H80000004&
    Caption         =   "Degrees"
    ForeColor       =   &H00000000&
    Height          =   240
    Left            =   3900
    TabIndex        =   20
    Top             =   675
    Width           =   750
End
Begin VB.Label Label1
    Caption         =   "&Latitude"
    Height          =   225
    Left            =   735
    TabIndex        =   19
    Top             =   375
    Width           =   1290
End
```

```
    Begin VB.Label Label10
       Appearance      =   0  'Flat
       BackColor       =   &H80000004&
       Caption         =   "Seconds"
       ForeColor       =   &H00000000&
       Height          =   240
       Left            =   1200
       TabIndex        =   18
       Top             =   1350
       Width           =   750
    End
    Begin VB.Label Label9
       Appearance      =   0  'Flat
       BackColor       =   &H80000004&
       Caption         =   "Minutes"
       ForeColor       =   &H00000000&
       Height          =   240
       Left            =   1200
       TabIndex        =   17
       Top             =   1020
       Width           =   750
    End
    Begin VB.Label Label8
       Appearance      =   0  'Flat
       BackColor       =   &H80000004&
       Caption         =   "Degrees"
       ForeColor       =   &H00000000&
       Height          =   240
       Left            =   1200
       TabIndex        =   16
       Top             =   675
       Width           =   750
    End
End
Begin VB.Frame Frame1
   Caption         =   "Site &Remarks:"
   Height          =   1335
   Left            =   1260
   TabIndex        =   13
   Top             =   3420
   Width           =   6735
   Begin VB.TextBox txtSite_Remarks
      BackColor       =   &H80000004&
      ForeColor       =   &H00000000&
      Height          =   960
      Left            =   150
      MultiLine       =   -1  'True
      ScrollBars      =   2  'Vertical
      TabIndex        =   7
      Top             =   255
      Width           =   6435
   End
End
Begin VB.CommandButton cmdOpen
   Appearance      =   0  'Flat
   BackColor       =   &H00E0E0E0&
   Caption         =   "S&ites"
   Height          =   480
   Left            =   285
   TabIndex        =   8
   Top             =   5250
   Width           =   1635
End
Begin VB.CommandButton cmdSave
   Appearance      =   0  'Flat
   BackColor       =   &H00E0E0E0&
```

```
      Caption         =   "&Save"
      Height          =   480
      Left            =   3928
      TabIndex        =   10
      Top             =   5250
      Width           =   1728
   End
   Begin VB.CommandButton cmdSave_As
      Appearance      =   0  'Flat
      BackColor       =   &H00E0E0E0&
      Caption         =   "Save &As"
      Height          =   480
      Left            =   5796
      TabIndex        =   11
      Top             =   5250
      Width           =   1728
   End
   Begin VB.CommandButton cmdPrint
      Appearance      =   0  'Flat
      BackColor       =   &H00E0E0E0&
      Caption         =   "&Print"
      Height          =   480
      Left            =   7665
      TabIndex        =   12
      Top             =   5250
      Width           =   1728
   End
   Begin VB.CommandButton cmdClose
      Appearance      =   0  'Flat
      BackColor       =   &H00E0E0E0&
      Caption         =   "&Close"
      Height          =   480
      Left            =   2070
      TabIndex        =   9
      Top             =   5250
      Width           =   1728
   End
End
Attribute VB_Name = "frmSite_Data"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

'This form allows the user to view and edit site files.

Private Sub cmdClose_Click()

    Dim Fname As String

    'This routine checks to see if the user has entered
    'valid input and then unloads the site form.  If the
    'user data is incorrect, the program generates
    'a warning.

    Dim temp As Integer

'    (txtLatitude_Degrees.text = "" And _
'        txtLatitude_Minutes.text = "" And _
'        txtLatitude_Seconds.text = "" And _
'        txtLongitude_Degrees.text = "" And _
'        txtLongitude_Minutes.text = "" And _
'        txtLongitude_Seconds.text = "" Then
'        frmSite_Data.caption = ""
'
```

```
'
'    If Fname <> "" And _
'      (txtLatitude_Degrees.text = "" And _
'        txtLatitude_Minutes.text = "" And _
'        txtLatitude_Seconds.text = "" And _
'        txtLongitude_Degrees.text = "" And _
'        txtLongitude_Minutes.text = "" And _
'        txtLongitude_Seconds.text = "") Then
'
'    If Fname <> "" Then
'        temp = verify_input(2)
'    Else
'        frmSite_Data.Hide
'    End If

'    temp = verify_input(2)

'    frmSite_Data.Hide

    Unload Me

End Sub

Private Sub cmdOpen_Click()

    'This routine calls the function which opens a
    'previously created site file.

    Call open_site

End Sub

Private Sub cmdPrint_Click()

    'This routine calls a procedure in module "OutPcode" which prints a hardcopy of
    'the information in a site file

    'Call print_site
    Print_Site_File_To_Printer

End Sub

Private Sub cmdSave_As_Click()

    'This routine calls a procedure which saves a file
    'by a new name.

'    frmMain.mnuSave_Sitecom.Enabled = False
'    frmMain.mnuSave_as_Sitecom.Enabled = False

    Call save_as_site

End Sub

'Private Sub cmdSave_Click()
Public Sub cmdSave_Click()

    'This routine calls a procedure which saves a
    'site file.

    Dim i As Integer
    Dim tempstr As String

    'check if a new file

    i = InStr(frmSite_Data.caption, "NEW_SITE")
```

```
    If i <> 0 Or frmSite_Data.caption = "  (Modified)" Or frmSite_Data.caption = " " Then
        Call cmdSave_As_Click
    Else

        'Remove the modified specifier from the file name if necessary.

        i = InStr(frmSite_Data.caption, " (Modified)")
        If i <> 0 Then
            tempstr = Mid$(frmSite_Data.caption, 1, i - 1)
        Else
            tempstr = frmSite_Data.caption
        End If
        frmSite_Data.caption = tempstr

        i = MsgBox(tempstr & " already exists.  Would you like to overwrite this file? " & vbCrLf
& _
                " (If not please select Save As).", vbYesNoCancel)
        If i = vbYes Then

'            frmMain.mnuSave_Sitecom.Enabled = False
'            frmMain.mnuSave_as_Sitecom.Enabled = False

            Call save_site

        End If
    End If

'    'check if a new file
'
'    i = InStr(frmSite_Data.caption, "NEW_SITE")
'    If i <> 0 Then
'        Call cmdSave_As_Click
'    End If
'
'    'Remove the modified specifier from the file name if necessary.
'
'    i = InStr(frmSite_Data.caption, " (Modified)")
'    If i <> 0 Then
'        TempStr = Mid$(frmSite_Data.caption, 1, i - 1)
'    Else
'        TempStr = frmSite_Data.caption
'    End If
'
'    i = MsgBox(TempStr & " already exists.  Would you like to overwrite this file? " & vbCrLf &
_
'            " (If not please select Save As).", vbYesNoCancel)
'    If i = vbYes Then
'
'        frmMain.mnuSave_Sitecom.Enabled = False
'        frmMain.mnuSave_as_Sitecom.Enabled = False
'        cmdSave.Enabled = False
'        cmdSave_As.Enabled = False
'
'        Call save_site
'
'    End If

End Sub

Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)

    frmMain.dlgSecpop.FilterIndex = 1


    'This routine checks to see if the user has entered
    'valid input and then unloads the site form.  If the
```

```
    'user data is incorrect, the program generates
    'a warning.

    Dim temp As Integer

'   if all fields are blank, don't verify

    If txtLatitude_Degrees.text = "" And _
        txtLatitude_Minutes.text = "" And _
        txtLatitude_Seconds.text = "" And _
        txtLongitude_Degrees.text = "" And _
        txtLongitude_Minutes.text = "" And _
        txtLongitude_Seconds.text = "" Then
        frmSite_Data.caption = ""
    Else
        If UnloadMode <> vbFormMDIForm Then 'vbformmdiform:  the mdi parent is closing
            temp = verify_input(2)
            frmSite_Data.Hide
            Cancel = True
        End If
    End If

    frmMain.mnuSave_Sitecom.Enabled = False
    frmMain.mnuSave_as_Sitecom.Enabled = False

'    frmSite_Data.Hide
'    Cancel = True

End Sub

Private Sub txtLatitude_Degrees_Change()

        frmRosette.blnInputDataChanged = True

End Sub

Private Sub txtLatitude_Degrees_KeyPress(keyascii As Integer)

    'Verify that the latitude degrees are a number,
    'that the final number is two digits or shorter,
    'and mark the Site file as modified.

'    If ((keyascii < 48) Or (keyascii > 57) Or (Len(frmSite_Data.txtLatitude_Degrees.text) >= 2))
And (keyascii <> 8) And keyascii <> 127 Then
'    If ((keyascii < 48) Or (keyascii > 57) Or (Len(frmSite_Data.txtLatitude_Degrees.text) > 1))
And (keyascii <> 8) And keyascii <> 127 Then
'    If ((keyascii < 48) Or (keyascii > 57) Or (Len(frmSite_Data.txtLatitude_Degrees.text) > 1))
Then
'    If ((keyascii < 48) Or (keyascii > 57) Or  Then


    If (keyascii >= 48 And keyascii <= 57) Or keyascii = 8 Then ' 0-9 or backspace

'        frmMain.mnuSave_Sitecom.Enabled = True
'        frmMain.mnuSave_as_Sitecom.Enabled = True

        If (InStr(frmSite_Data.caption, " (Modified)") = 0) Then
            frmSite_Data.caption = frmSite_Data.caption + " (Modified)"
        End If
        If (InStr(frmProblem_Data.txtSite_File_Name.text, " (Modified)") = 0) Then
            frmProblem_Data.txtSite_File_Name.text = frmSite_Data.caption
        End If

'        If txtLatitude_Degrees <> "" And txtLongitude_Degrees <> "" Then
'            verify_input (2)
'        End If
```

```
    Else
        keyascii = 0
        Beep
    End If

    frmRosette.blnInputDataChanged = True

End Sub

Private Sub txtLatitude_Degrees_LostFocus()

'    If txtLatitude_Degrees <> "" And txtLongitude_Degrees <> "" Then
'        verify_input (2)
'    End If

End Sub

Private Sub txtLatitude_Degrees_Validate(Cancel As Boolean)

    If txtLatitude_Degrees <> "" And txtLongitude_Degrees <> "" Then
        verify_input (2)
    End If

End Sub

Private Sub txtLatitude_Minutes_Change()

    frmRosette.blnInputDataChanged = True

End Sub

Private Sub txtLatitude_Minutes_KeyPress(keyascii As Integer)

    'For documentation, see latitude degrees.

    If (keyascii >= 48 And keyascii <= 57) Or keyascii = 8 Then ' 0-9 or backspace

'        frmMain.mnuSave_Sitecom.Enabled = True
'        frmMain.mnuSave_as_Sitecom.Enabled = True

        If (InStr(frmSite_Data.caption, " (Modified)") = 0) Then
            frmSite_Data.caption = frmSite_Data.caption + " (Modified)"
        End If
        If (InStr(frmProblem_Data.txtSite_File_Name.text, " (Modified)") = 0) Then
            frmProblem_Data.txtSite_File_Name.text = frmSite_Data.caption
        End If

    Else

        keyascii = 0
        Beep

    End If

    frmRosette.blnInputDataChanged = True

End Sub

Private Sub txtLatitude_Minutes_Validate(Cancel As Boolean)

    If frmSite_Data.txtLatitude_Minutes <> "" Then
        If frmSite_Data.txtLatitude_Minutes.text > 60 Then
            MsgBox "Please enter a value for latitude minutes that is less or equal to 60"
            frmSite_Data.txtLatitude_Minutes.text = ""
            Cancel = True
```

```
        End If
    End If

End Sub

Private Sub txtLatitude_Seconds_Change()

    frmRosette.blnInputDataChanged = True

End Sub

Private Sub txtLatitude_Seconds_KeyPress(keyascii As Integer)

    'For documentation, see latitude degrees.

    If (keyascii >= 48 And keyascii <= 57) Or keyascii = 8 Then ' 0-9 or backspace

'        frmMain.mnuSave_Sitecom.Enabled = True
'        frmMain.mnuSave_as_Sitecom.Enabled = True

        If (InStr(frmSite_Data.caption, " (Modified)") = 0) Then
            frmSite_Data.caption = frmSite_Data.caption + " (Modified)"
        End If
        If (InStr(frmProblem_Data.txtSite_File_Name.text, " (Modified)") = 0) Then
            frmProblem_Data.txtSite_File_Name.text = frmSite_Data.caption
        End If

    Else

        keyascii = 0
        Beep

    End If

    frmRosette.blnInputDataChanged = True

End Sub

Private Sub txtLatitude_Seconds_Validate(Cancel As Boolean)

    If frmSite_Data.txtLatitude_Seconds <> "" Then
        If frmSite_Data.txtLatitude_Seconds.text > 60 Then
            MsgBox "Please enter a value for latitude seconds that is less or equal to 60"
            frmSite_Data.txtLatitude_Seconds.text = ""
            Cancel = True
        End If
    End If

End Sub

Private Sub txtLongitude_Degrees_Change()

    frmRosette.blnInputDataChanged = True

End Sub

Private Sub txtLongitude_Degrees_KeyPress(keyascii As Integer)

    If (keyascii >= 48 And keyascii <= 57) Or keyascii = 8 Then ' 0-9 or backspace

'        frmMain.mnuSave_Sitecom.Enabled = True
'        frmMain.mnuSave_as_Sitecom.Enabled = True

        If (InStr(frmSite_Data.caption, " (Modified)") = 0) Then
            frmSite_Data.caption = frmSite_Data.caption + " (Modified)"
        End If
```

```
        If (InStr(frmProblem_Data.txtSite_File_Name.text, " (Modified)") = 0) Then
            frmProblem_Data.txtSite_File_Name.text = frmSite_Data.caption
        End If

'        If txtLatitude_Degrees <> "" And txtLongitude_Degrees <> "" Then
'            verify_input (2)
'        End If

    Else

        keyascii = 0
        Beep

    End If

    frmRosette.blnInputDataChanged = True

End Sub

Private Sub txtLongitude_Degrees_LostFocus()

'    If txtLatitude_Degrees <> "" And txtLongitude_Degrees <> "" Then
'        verify_input (2)
'    End If

End Sub

Private Sub txtLongitude_Degrees_Validate(Cancel As Boolean)

    If txtLatitude_Degrees <> "" And txtLongitude_Degrees <> "" Then
        verify_input (2)
    End If

End Sub

Private Sub txtLongitude_Minutes_Change()

    frmRosette.blnInputDataChanged = True

End Sub

Private Sub txtLongitude_Minutes_KeyPress(keyascii As Integer)

    'For documentation, see longitude degrees.

    If (keyascii >= 48 And keyascii <= 57) Or keyascii = 8 Then ' 0-9 or backspace

'        frmMain.mnuSave_Sitecom.Enabled = True
'        frmMain.mnuSave_as_Sitecom.Enabled = True

        If (InStr(frmSite_Data.caption, " (Modified)") = 0) Then
            frmSite_Data.caption = frmSite_Data.caption + " (Modified)"
        End If
        If (InStr(frmProblem_Data.txtSite_File_Name.text, " (Modified)") = 0) Then
            frmProblem_Data.txtSite_File_Name.text = frmSite_Data.caption
        End If

    Else

        keyascii = 0
        Beep

    End If

    frmRosette.blnInputDataChanged = True
```

```
End Sub

Private Sub txtLongitude_Minutes_Validate(Cancel As Boolean)


    If frmSite_Data.txtLongitude_Minutes <> "" Then
        If frmSite_Data.txtLongitude_Minutes.text > 60 Then
            MsgBox "Please enter a value for longitude minutes that is less or equal to 60"
            frmSite_Data.txtLongitude_Minutes.text = ""
            Cancel = True
        End If
    End If

End Sub

Private Sub txtLongitude_Seconds_Change()

    frmRosette.blnInputDataChanged = True

End Sub

Private Sub txtLongitude_Seconds_KeyPress(keyascii As Integer)

    'For documentation, see longitude degrees.

    ' 48 = 0
    ' 57 = 9
    ' 8 = backspace

    If (keyascii >= 48 And keyascii <= 57) Or keyascii = 8 Then ' 0-9 or backspace

'         frmMain.mnuSave_Sitecom.Enabled = True
'         frmMain.mnuSave_as_Sitecom.Enabled = True

        If (InStr(frmSite_Data.caption, " (Modified)") = 0) Then
            frmSite_Data.caption = frmSite_Data.caption + " (Modified)"
        End If
        If (InStr(frmProblem_Data.txtSite_File_Name.text, " (Modified)") = 0) Then
            frmProblem_Data.txtSite_File_Name.text = frmSite_Data.caption
        End If

    Else

        keyascii = 0
        Beep

    End If

    frmRosette.blnInputDataChanged = True

End Sub

Private Sub txtLongitude_Seconds_Validate(Cancel As Boolean)

    If frmSite_Data.txtLongitude_Seconds <> "" Then
        If frmSite_Data.txtLongitude_Seconds.text > 60 Then
            MsgBox "Please enter a value for longitude seconds that is less or equal to 60"
            frmSite_Data.txtLongitude_Seconds.text = ""
            Cancel = True
        End If
    End If

End Sub

Private Sub txtSite_Name_Change()
```

```
    frmRosette.blnInputDataChanged = True

End Sub

Private Sub txtSite_Name_KeyPress(keyascii As Integer)


    'Disable the return key in the site field, and mark
    'the site form as modified.

    If keyascii = KEY_RETURN Then

        keyascii = 0
        MsgBox "Sorry, no hard returns allowed." + Chr$(13) + "Words will wrap automatically.",
vbOK, "Error"

    Else

'         frmMain.mnuSave_Sitecom.Enabled = True
'         frmMain.mnuSave_as_Sitecom.Enabled = True

        If (InStr(frmSite_Data.caption, " (Modified)") = 0) Then
            frmSite_Data.caption = frmSite_Data.caption + " (Modified)"
        End If
        If (InStr(frmProblem_Data.txtSite_File_Name.text, " (Modified)") = 0) Then
            frmProblem_Data.txtSite_File_Name.text = frmSite_Data.caption
        End If

    End If

    frmRosette.blnInputDataChanged = True

End Sub

Private Sub txtSite_Remarks_KeyPress(keyascii As Integer)

    'Disable the enter key in the remarks field, and
    'Mark the site form as modified.
    If keyascii = KEY_RETURN Then

        keyascii = 0
        MsgBox "Sorry, no hard returns allowed." + Chr$(13) + "Words will wrap automatically.",
vbOK, "Error"

    Else

'         frmMain.mnuSave_Sitecom.Enabled = True
'         frmMain.mnuSave_as_Sitecom.Enabled = True

        If (InStr(frmSite_Data.caption, " (Modified)") = 0) Then
            frmSite_Data.caption = frmSite_Data.caption + " (Modified)"
        End If
        If (InStr(frmProblem_Data.txtSite_File_Name.text, " (Modified)") = 0) Then
            frmProblem_Data.txtSite_File_Name.text = frmSite_Data.caption
        End If

    End If

End Sub
```

H-33 Visual Basic Form Module: Table1.frm

```
VERSION 5.00
Begin VB.Form frmTable_1
   Appearance      =   0  'Flat
   AutoRedraw      =   -1  'True
   BackColor       =   &H80000004&
   BorderStyle     =   1  'Fixed Single
   Caption         =   "Population Table"
   ClientHeight    =   4680
   ClientLeft      =   30
   ClientTop       =   570
   ClientWidth     =   8355
   BeginProperty Font
      Name             =   "MS Sans Serif"
      Size             =   8.25
      Charset          =   0
      Weight           =   700
      Underline        =   0   'False
      Italic           =   0   'False
      Strikethrough    =   0   'False
   EndProperty
   ForeColor       =   &H00000000&
   MaxButton       =   0   'False
   MinButton       =   0   'False
   PaletteMode     =   1  'UseZOrder
   ScaleHeight     =   4680
   ScaleWidth      =   8355
   StartUpPosition =   1  'CenterOwner
   Begin VB.Frame Frame1
      Appearance       =   0  'Flat
      BackColor        =   &H80000004&
      ForeColor        =   &H00000000&
      Height           =   4320
      Left             =   108
      TabIndex         =   31
      Top              =   0
      Width            =   6696
   End
   Begin VB.HScrollBar hsbRadii
      Height           =   240
      Left             =   216
      Max              =   50
      Min              =   1
      TabIndex         =   29
      Top              =   4080
      Value            =   1
      Width            =   6480
   End
   Begin VB.VScrollBar vsbSectors
      Height           =   3840
      Left             =   6696
      Max              =   16
      Min              =   1
      TabIndex         =   30
      Top              =   240
      Value            =   1
      Width            =   108
   End
   Begin VB.CommandButton cmdexit
      Appearance       =   0  'Flat
      BackColor        =   &H00808000&
      Caption          =   "&Close"
      Default          =   -1  'True
```

```
   Height          =   720
   Left            =   7020
   TabIndex        =   1
   Top             =   1440
   Width           =   972
End
Begin VB.CommandButton cmdprint
   Appearance      =   0  'Flat
   BackColor       =   &H00808000&
   Caption         =   "&Print"
   Height          =   720
   Left            =   7020
   TabIndex        =   0
   Top             =   2160
   Width           =   972
End
Begin VB.Label lblCsum_array
   Alignment       =   1  'Right Justify
   Appearance      =   0  'Flat
   BackColor       =   &H00808000&
   Caption         =   "123Sum789"
   ForeColor       =   &H00000000&
   Height          =   240
   Index           =   4
   Left            =   5508
   TabIndex        =   32
   Top             =   2880
   Width           =   972
End
Begin VB.Label lblCsum_array
   Alignment       =   1  'Right Justify
   Appearance      =   0  'Flat
   BackColor       =   &H00808000&
   Caption         =   "123Sum789"
   ForeColor       =   &H00000000&
   Height          =   240
   Index           =   5
   Left            =   5508
   TabIndex        =   33
   Top             =   3360
   Width           =   972
End
Begin VB.Label lblCsum_array
   Alignment       =   1  'Right Justify
   Appearance      =   0  'Flat
   BackColor       =   &H00808000&
   Caption         =   "123Sum789"
   ForeColor       =   &H00000000&
   Height          =   240
   Index           =   3
   Left            =   5508
   TabIndex        =   24
   Top             =   2400
   Width           =   972
End
Begin VB.Label lblCsum_array
   Alignment       =   1  'Right Justify
   Appearance      =   0  'Flat
   BackColor       =   &H00808000&
   Caption         =   "123Sum789"
   ForeColor       =   &H00000000&
   Height          =   240
   Index           =   2
   Left            =   5508
   TabIndex        =   25
   Top             =   1920
```

```
    Width           =   972
 End
 Begin VB.Label lblCsum_array
    Alignment       =   1  'Right Justify
    Appearance      =   0  'Flat
    BackColor       =   &H00808000&
    Caption         =   "123Sum789"
    ForeColor       =   &H00000000&
    Height          =   240
    Index           =   1
    Left            =   5508
    TabIndex        =   26
    Top             =   1440
    Width           =   972
 End
 Begin VB.Label lblCsum_array
    Alignment       =   1  'Right Justify
    Appearance      =   0  'Flat
    BackColor       =   &H00808000&
    Caption         =   "123Sum789"
    ForeColor       =   &H00000000&
    Height          =   240
    Index           =   0
    Left            =   5508
    TabIndex        =   27
    Top             =   960
    Width           =   972
 End
 Begin VB.Label lblTotal
    Alignment       =   1  'Right Justify
    Appearance      =   0  'Flat
    BackColor       =   &H00808000&
    Caption         =   "SumSum789"
    ForeColor       =   &H00000000&
    Height          =   240
    Left            =   5508
    TabIndex        =   34
    Top             =   3840
    Width           =   972
 End
 Begin VB.Label Label21
    Appearance      =   0  'Flat
    AutoSize        =   -1  'True
    BackColor       =   &H00808000&
    Caption         =   "Sum"
    ForeColor       =   &H00000000&
    Height          =   240
    Left            =   5832
    TabIndex        =   9
    Top             =   480
    Width           =   324
 End
 Begin VB.Label lblCumulative_Header
    Appearance      =   0  'Flat
    AutoSize        =   -1  'True
    BackColor       =   &H00808000&
    Caption         =   "Sum"
    ForeColor       =   &H00000000&
    Height          =   240
    Left            =   324
    TabIndex        =   14
    Top             =   3840
    Width           =   324
 End
 Begin VB.Label lblSector_array
    Appearance      =   0  'Flat
```

```
      BackColor       =   &H00808000&
      Caption         =   "A"
      ForeColor       =   &H00000000&
      Height          =   240
      Index           =   0
      Left            =   324
      TabIndex        =   15
      Top             =   960
      Width           =   324
   End
   Begin VB.Label lblSector_array
      Appearance      =   0  'Flat
      BackColor       =   &H00808000&
      Caption         =   "B"
      ForeColor       =   &H00000000&
      Height          =   240
      Index           =   1
      Left            =   324
      TabIndex        =   16
      Top             =   1440
      Width           =   324
   End
   Begin VB.Label lblSector_array
      Appearance      =   0  'Flat
      BackColor       =   &H00808000&
      Caption         =   "C"
      ForeColor       =   &H00000000&
      Height          =   240
      Index           =   2
      Left            =   324
      TabIndex        =   17
      Top             =   1920
      Width           =   324
   End
   Begin VB.Label lblSector_array
      Appearance      =   0  'Flat
      BackColor       =   &H00808000&
      Caption         =   "D"
      ForeColor       =   &H00000000&
      Height          =   240
      Index           =   3
      Left            =   324
      TabIndex        =   18
      Top             =   2400
      Width           =   324
   End
   Begin VB.Label lblSector_array
      Appearance      =   0  'Flat
      BackColor       =   &H00808000&
      Caption         =   "E"
      ForeColor       =   &H00000000&
      Height          =   240
      Index           =   4
      Left            =   324
      TabIndex        =   19
      Top             =   2880
      Width           =   324
   End
   Begin VB.Label lblSector_array
      Appearance      =   0  'Flat
      BackColor       =   &H00808000&
      Caption         =   "F"
      ForeColor       =   &H00000000&
      Height          =   240
      Index           =   5
      Left            =   324
```

```
    TabIndex        =   28
    Top             =   3360
    Width           =   324
 End
 Begin VB.Label lblRsum_array
    Alignment       =   1  'Right Justify
    Appearance      =   0  'Flat
    BackColor       =   &H00808000&
    Caption         =   "Sum456789"
    ForeColor       =   &H00000000&
    Height          =   240
    Index           =   0
    Left            =   972
    TabIndex        =   20
    Top             =   3840
    Width           =   972
 End
 Begin VB.Label lblRsum_array
    Alignment       =   1  'Right Justify
    Appearance      =   0  'Flat
    BackColor       =   &H00808000&
    Caption         =   "Sum456789"
    ForeColor       =   &H00000000&
    Height          =   240
    Index           =   1
    Left            =   2052
    TabIndex        =   21
    Top             =   3840
    Width           =   972
 End
 Begin VB.Label lblRsum_array
    Alignment       =   1  'Right Justify
    Appearance      =   0  'Flat
    BackColor       =   &H00808000&
    Caption         =   "Sum456789"
    ForeColor       =   &H00000000&
    Height          =   240
    Index           =   2
    Left            =   3132
    TabIndex        =   22
    Top             =   3840
    Width           =   972
 End
 Begin VB.Label lblRsum_array
    Alignment       =   1  'Right Justify
    Appearance      =   0  'Flat
    BackColor       =   &H00808000&
    Caption         =   "Sum456789"
    ForeColor       =   &H00000000&
    Height          =   240
    Index           =   3
    Left            =   4212
    TabIndex        =   23
    Top             =   3840
    Width           =   972
 End
 Begin VB.Label lblDistance_array
    Alignment       =   1  'Right Justify
    Appearance      =   0  'Flat
    BackColor       =   &H00808000&
    Caption         =   "1000.0000"
    ForeColor       =   &H00000000&
    Height          =   240
    Index           =   0
    Left            =   972
    TabIndex        =   10
```

```
    Top             =   480
    Width           =   972
End
Begin VB.Label lblDistance_array
    Alignment       =   1  'Right Justify
    Appearance      =   0  'Flat
    BackColor       =   &H00808000&
    Caption         =   "2000.0000"
    ForeColor       =   &H00000000&
    Height          =   240
    Index           =   1
    Left            =   2052
    TabIndex        =   11
    Top             =   480
    Width           =   972
End
Begin VB.Label lblDistance_array
    Alignment       =   1  'Right Justify
    Appearance      =   0  'Flat
    BackColor       =   &H00808000&
    Caption         =   "3000.0000"
    ForeColor       =   &H00000000&
    Height          =   240
    Index           =   2
    Left            =   3132
    TabIndex        =   12
    Top             =   480
    Width           =   972
End
Begin VB.Label lblDistance_array
    Alignment       =   1  'Right Justify
    Appearance      =   0  'Flat
    BackColor       =   &H00808000&
    Caption         =   "4000.0000"
    ForeColor       =   &H00000000&
    Height          =   240
    Index           =   3
    Left            =   4212
    TabIndex        =   13
    Top             =   480
    Width           =   972
End
Begin VB.Label lblValue_array
    Appearance      =   0  'Flat
    BackColor       =   &H00808000&
    BorderStyle     =   1  'Fixed Single
    Caption         =   " 123456789 123456789 123456789 123456789 "
    ForeColor       =   &H00000000&
    Height          =   720
    Index           =   0
    Left            =   756
    TabIndex        =   2
    Top             =   720
    Width           =   4644
End
Begin VB.Label lblValue_array
    Appearance      =   0  'Flat
    BackColor       =   &H00808000&
    BorderStyle     =   1  'Fixed Single
    Caption         =   " 123456789 123456789 123456789 123456789 "
    ForeColor       =   &H00000000&
    Height          =   720
    Index           =   1
    Left            =   756
    TabIndex        =   3
    Top             =   1200
```

```
    Width           =   4644
 End
 Begin VB.Label lblValue_array
    Appearance      =   0  'Flat
    BackColor       =   &H00808000&
    BorderStyle     =   1  'Fixed Single
    Caption         =   " 123456789 123456789 123456789 123456789 "
    ForeColor       =   &H00000000&
    Height          =   720
    Index           =   2
    Left            =   756
    TabIndex        =   4
    Top             =   1680
    Width           =   4644
 End
 Begin VB.Label lblValue_array
    Appearance      =   0  'Flat
    BackColor       =   &H00808000&
    BorderStyle     =   1  'Fixed Single
    Caption         =   " 123456789 123456789 123456789 123456789 "
    ForeColor       =   &H00000000&
    Height          =   720
    Index           =   3
    Left            =   756
    TabIndex        =   5
    Top             =   2160
    Width           =   4644
 End
 Begin VB.Label lblValue_array
    Appearance      =   0  'Flat
    BackColor       =   &H00808000&
    BorderStyle     =   1  'Fixed Single
    Caption         =   " 123456789 123456789 123456789 123456789 "
    ForeColor       =   &H00000000&
    Height          =   720
    Index           =   4
    Left            =   756
    TabIndex        =   6
    Top             =   2640
    Width           =   4644
 End
 Begin VB.Label lblValue_array
    Appearance      =   0  'Flat
    BackColor       =   &H00808000&
    BorderStyle     =   1  'Fixed Single
    Caption         =   " 123456789 123456789 123456789 123456789 "
    ForeColor       =   &H00000000&
    Height          =   720
    Index           =   5
    Left            =   756
    TabIndex        =   7
    Top             =   3120
    Width           =   4644
 End
 Begin VB.Label lblDistance_Heading
    Alignment       =   2  'Center
    Appearance      =   0  'Flat
    BackColor       =   &H00808000&
    Caption         =   "Radii (Kilometers)"
    ForeColor       =   &H00000000&
    Height          =   240
    Left            =   2160
    TabIndex        =   8
    Top             =   240
    Width           =   1944
 End
```

```
End
Attribute VB_Name = "frmTable_1"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
'''Option Explicit
'''
''''$INCLUDE: 'secpop90.bi'
'''
''''This is the actual table form for table 1.  See rptable_1 for more
''''info.
'''
'''Private Sub cmdExit_Click()
'''
'''     'Remove table 1 from memory.
'''
'''     Unload frmTable_1
'''
'''End Sub
'''
'''Private Sub cmdPrint_Click()
'''
'''     'This routine prints the displayed population table.
'''
'''     Dim ForeColor As Integer, BackColor As Integer
'''     Dim Copies As Integer, Cancel As Integer
'''     Dim quote As String, comma As String
'''     Dim blank As String, degree As String
'''     Dim pages As Integer, radius As Integer
'''     Dim first_radius As Integer, last_radius As Integer
'''     Dim i As Integer, j As Integer, k As Integer
'''     Dim L As Integer, m As Integer, n As Integer
'''     Dim sum As Long
'''
'''
'''     'Set quote and comma strings to null if MACCS input file format is
'''     'selected or to '"' and ',' if comma separated variable (CSV) format
'''     'is selected.
'''
'''     If (frmSetup.optMACCS.Value) Then
'''         quote = ""
'''         comma = ""
'''         blank = " "
'''     ElseIf (frmSetup.optCSV.Value) Then
'''         quote = """"
'''         comma = ","
'''         blank = ""
'''     Else
'''         quote = ""
'''         comma = ""
'''         blank = " "
'''     End If
'''
'''     'Set degree character.
'''
'''     degree = Chr$(248)
'''
'''     'Set colors for Dialog box
'''
'''     BackColor = vbWhite
'''     ForeColor = vbBlack
'''
'''     'Call the standard print dialog.
'''
'''     Call FilePrint(Copies, ForeColor, BackColor, Cancel)
```

```
'''
'''     'If cancel is not selected by user in the printer dialogue box,
'''     'then proceed.
'''
'''     If (Not Cancel) Then
'''
'''         'Enable local error checking.
'''
'''         On Local Error GoTo table1_printer_error
'''
'''         'If the user is printing to a printer adjust the margins so that
'''         'there are no extra line feeds after the lines that have 80
'''         'columns otherwise prepend the output path to the printer
'''         'target.
'''
'''         If ((Printer.PrintTarget = "LPT1:") Or _
'''             (Printer.PrintTarget = "LPT2:") Or _
'''             (Printer.PrintTarget = "LPT3:")) Then
'''
'''             width printer.PrintTarget, 81
'''
'''         Else
'''
'''             Printer.PrintTarget = frmSetup.txtOutput_path.text + _
'''                 "\" + Printer.PrintTarget
'''
'''         End If
'''
'''         'Determine the number of "pages" that need to be printed per
'''         'copy. A single page is considered to be 16 directions and a
'''         'sum row long, by at most 6 radii or 5 radii and a sum column.
'''
'''         pages = (number_of_radii + 1) \ 6
'''         If (((number_of_radii + 1) Mod 6) <> 0) Then
'''             pages = pages + 1
'''         End If
'''
'''         'Execute loop once for each copy requested by the user.
'''
'''         For i = 1 To Copies Step 1
'''
'''             'Print the first line.
'''
'''             Printer.Print quote;
'''             Printer.Print "SECPOP90 V2.3 ";
'''             If (table_type = 1) Then
'''                 Printer.Print "Population Data";
'''             Else
'''                 Printer.Print "Cumulative Population Data";
'''             End If
'''             Printer.Print quote;
'''             Printer.Print
'''
'''             'Print the second line.
'''
'''             Printer.Print quote;
'''             Printer.Print "Date:";
'''             Printer.Print quote; comma; blank;
'''             Printer.Print quote;
'''             Printer.Print Format$(Now, "mm/dd/yyyy");
'''             Printer.Print quote; comma; blank; blank; blank;
'''             Printer.Print quote;
'''             Printer.Print "Time:";
'''             Printer.Print quote; comma; blank;
'''             Printer.Print quote;
'''             Printer.Print Format$(Now, "hh:mm:ss");
```

```
'''             Printer.Print quote;
'''             Printer.Print
'''
'''             'Print the third line.
'''
'''             Printer.Print quote;
'''             Printer.Print "Site Name:";
'''             Printer.Print quote; comma; blank;
'''             Printer.Print quote;
'''             Printer.Print _
'''                 LTrim$(RTrim$(frmSite_Data.txtSite_Name.text));
'''             Printer.Print quote;
'''             Printer.Print
'''
'''             'Print the fourth line.
'''
'''             Printer.Print quote;
'''             Printer.Print "Latitude:";
'''             Printer.Print quote; comma; blank;
'''             Printer.Print USING; "##"; _
'''                 Val(frmSite_Data.txtLatitude_Degrees.text);
'''             Printer.Print comma;
'''             Printer.Print quote;
'''             Printer.Print degree;
'''             Printer.Print quote; comma;
'''             Printer.Print USING; "##"; _
'''                 Val(frmSite_Data.txtLatitude_Minutes.text);
'''             Printer.Print comma;
'''             Printer.Print quote;
'''             Printer.Print "'";
'''             Printer.Print quote; comma;
'''             Printer.Print USING; "##"; _
'''                 Val(frmSite_Data.txtLatitude_Seconds.text);
'''             Printer.Print comma;
'''             Printer.Print quote;
'''             Printer.Print "''";
'''             Printer.Print quote; comma; blank; blank; blank;
'''             Printer.Print quote;
'''             Printer.Print "Longitude:";
'''             Printer.Print quote; comma; blank;
'''             Printer.Print USING; "###"; _
'''                 Val(frmSite_Data.txtLongitude_Degrees.text);
'''             Printer.Print comma;
'''             Printer.Print quote;
'''             Printer.Print degree;
'''             Printer.Print quote; comma;
'''             Printer.Print USING; "##"; _
'''                 Val(frmSite_Data.txtLongitude_Minutes.text);
'''             Printer.Print comma;
'''             Printer.Print quote;
'''             Printer.Print "'";
'''             Printer.Print quote; comma;
'''             Printer.Print USING; "##"; _
'''                 Val(frmSite_Data.txtLongitude_Seconds.text);
'''             Printer.Print comma;
'''             Printer.Print quote;
'''             Printer.Print "''";
'''             Printer.Print quote;
'''             Printer.Print
'''
'''             'Print the fifth line.
'''
'''             Printer.Print quote;
'''             Printer.Print "Population Multiplier:";
'''             Printer.Print quote; comma; blank;
'''             Printer.Print USING; "####.####"; _
```

```
'''             Val(frmProblem_Data.txtPopulation_multiplier.text);
'''         Printer.Print
'''
'''         'Print out the "pages" of population data.
'''
'''         'Initialize the radius index to point to the first radius.
'''
'''         radius = 1
'''
'''         For j = 1 To pages Step 1
'''
'''             'Print out the radii header row.
'''             Printer.Print
'''             Printer.Print
'''             Printer.Print quote;
'''             Printer.Print "Radii(";
'''             If (frmProblem_Data.optKilometers.Value) Then
'''                 Printer.Print "km)";
'''             Else
'''                 Printer.Print "mi)";
'''             End If
'''             Printer.Print quote; comma;
'''
'''             'Print out the radii, setting the loop min and max to
'''             'the radii to be printed for this page. Note: the final
'''             'radii column is the sum column.
'''
'''             first_radius = radius
'''             If ((radius + 5) <= (number_of_radii + 1)) Then
'''                 last_radius = radius + 5
'''             Else
'''                 last_radius = number_of_radii + 1
'''             End If
'''
'''             For k = first_radius To last_radius Step 1
'''                 If (k <> (number_of_radii + 1)) Then
'''                     Printer.Print USING; "######.###"; _
'''                         radial_distance(k);
'''                     If (k <> last_radius) Then
'''                         Printer.Print comma;
'''                     End If
'''                 Else
'''                     Printer.Print "      "; blank;
'''                     Printer.Print quote;
'''                     Printer.Print "Sum";
'''                     Printer.Print quote;
'''                 End If
'''             Next k
'''
'''             Printer.Print
'''             Printer.Print
'''
'''             'Print out the direction and  the (cumulative)
'''             'population values. The last column is the sum of the
'''             'population values.
'''
'''             For k = 1 To number_of_segments Step 1
'''                 Printer.Print quote;
'''                 Printer.Print USING; "\       \"; directions(k);
'''                 Printer.Print quote; comma;
'''                 For L = first_radius To last_radius Step 1
'''                     If (L <> (number_of_radii + 1)) Then
'''                         If (table_type = 1) Then
'''                             Printer.Print USING; "##########"; _
'''                                 sector_population(k, L);
'''                         Else
```

```
'''                             sum = 0
'''                             For m = 1 To L Step 1
'''                                 sum = sum + sector_population(k, m)
'''                             Next m
'''                             Printer.Print USING; "##########"; sum;
'''                         End If
'''                         If (L <> last_radius) Then
'''                             Printer.Print comma;
'''                         End If
'''                     Else
'''                         sum = 0
'''                         For m = 1 To number_of_radii Step 1
'''                             sum = sum + sector_population(k, m)
'''                         Next m
'''                         Printer.Print USING; "##########"; sum;
'''                     End If
'''                 Next L
'''                 Printer.Print
'''             Next k
'''
'''             Printer.Print quote;
'''             Printer.Print "Sum      ";
'''             Printer.Print quote; comma;
'''             For L = first_radius To last_radius Step 1
'''                 If (L <> (number_of_radii + 1)) Then
'''                     sum = 0
'''                     If (table_type = 1) Then
'''                         For m = 1 To number_of_segments Step 1
'''                             sum = sum + sector_population(m, L)
'''                         Next m
'''                     Else
'''                         For m = 1 To number_of_segments Step 1
'''                             For n = 1 To L Step 1
'''                                 sum = sum + sector_population(m, n)
'''                             Next n
'''                         Next m
'''                     End If
'''                     Printer.Print USING; "##########"; sum;
'''                     If (L <> last_radius) Then
'''                         Printer.Print comma;
'''                     End If
'''                 Else
'''                     sum = 0
'''                     For m = 1 To number_of_segments Step 1
'''                         For n = 1 To number_of_radii Step 1
'''                             sum = sum + sector_population(m, n)
'''                         Next n
'''                     Next m
'''                     Printer.Print USING; "##########"; sum;
'''                 End If
'''             Next L
'''             Printer.Print
'''
'''             radius = last_radius + 1
'''
'''         Next j
'''
'''         Printer.NewPage
'''
'''     Next i
'''
'''         Printer.EndDoc
'''
'''  End If
'''
'''  Exit Sub
```

```
'''
'''table1_printer_error:
'''
'''     'If there is a problem, tell the user and exit the subroutine.
'''
'''     MsgBox Error$(Err)
'''
'''     Exit Sub
'''
'''End Sub
'''
'''Private Sub Form_Load()
'''
'''     'Perform first call to paint table to initialize the table.
'''
'''     Unload frmDisclaimer
'''
'''     x_position = 1
'''     y_position = 1
'''     Call paint_table
'''
'''End Sub
'''
'''Private Sub hsbRadii_Change()
'''
'''     x_position = hsbRadii.Value
'''     Call paint_table
'''
'''End Sub
'''
'''Private Sub vsbSectors_Change()
'''
'''     y_position = vsbSectors.Value
'''     Call paint_table
'''
'''End Sub


Private Sub cmdPrint_Click()

    Call Print_Table_File_to_Printer

End Sub
```

H.34 Visual Basic Form Module: Welcome.frm

```
VERSION 5.00
Begin VB.Form frmDisclaimer
   Appearance      =   0  'Flat
   AutoRedraw      =   -1  'True
   BorderStyle     =   1  'Fixed Single
   Caption         =   "Welcome"
   ClientHeight    =   6915
   ClientLeft      =   1110
   ClientTop       =   810
   ClientWidth     =   9570
   BeginProperty Font
      Name            =   "Times New Roman"
      Size            =   12
      Charset         =   0
      Weight          =   700
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
   EndProperty
   ForeColor       =   &H00000000&
   Icon            =   "Welcome.frx":0000
   MaxButton       =   0   'False
   MDIChild        =   -1  'True
   MinButton       =   0   'False
   PaletteMode     =   1  'UseZOrder
   ScaleHeight     =   6915
   ScaleWidth      =   9570
   Begin VB.CommandButton Command1
      Caption         =   "OK"
      Height          =   465
      Left            =   7560
      TabIndex        =   0
      Top             =   6285
      Width           =   1575
   End
   Begin VB.Label Label4
      BackColor       =   &H80000004&
      BackStyle       =   0  'Transparent
      Caption         =   $"Welcome.frx":0CCA
      Height          =   2505
      Left            =   300
      TabIndex        =   3
      Top             =   930
      Width           =   9075
   End
   Begin VB.Label Label2
      BackColor       =   &H80000004&
      BackStyle       =   0  'Transparent
      Caption         =   $"Welcome.frx":0F5B
      Height          =   2670
      Left            =   315
      TabIndex        =   2
      Top             =   3465
      Width           =   9030
   End
   Begin VB.Label Label1
      BackStyle       =   0  'Transparent
      Caption         =   "Welcome to SECPOP2000"
      BeginProperty Font
         Name            =   "Times New Roman"
         Size            =   15.75
         Charset         =   0
```

```
         Weight          =   700
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      Height          =   345
      Left            =   3075
      TabIndex        =   1
      Top             =   300
      Width           =   4035
   End
   Begin VB.Image Image1
      Height          =   4485
      Left            =   270
      Picture         =   "Welcome.frx":1216
      Top             =   120
      Visible         =   0   'False
      Width           =   4500
   End
End
Attribute VB_Name = "frmDisclaimer"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
'This is just your general purpose information and disclaimer message
'which is displayed at the beginning of the program.

Private Sub Command1_Click()
    Unload Me
End Sub

Private Sub Form_KeyPress(keyascii As Integer)

    Unload frmDisclaimer

End Sub

Private Sub Form_Load()

    Me.Top = 50
    Me.Left = 1000

End Sub

Private Sub Form_LostFocus()

    Unload frmDisclaimer

End Sub

Private Sub lblNotice_Click()

    Unload frmDisclaimer

End Sub
```

H.36 Visual Basic Class Module: GridClass.cls

```
VERSION 1.0 CLASS
BEGIN
  MultiUse = -1  'True
  Persistable = 0  'NotPersistable
  DataBindingBehavior = 0  'vbNone
  DataSourceBehavior  = 0  'vbNone
  MTSTransactionMode  = 0  'NotAnMTSObject
END
Attribute VB_Name = "GridClass"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = False
Attribute VB_Exposed = False
Option Explicit

Private WithEvents TDBG As TrueDBGrid70.TDBGrid
Attribute TDBG.VB_VarHelpID = -1
Private GridArray() As Variant
Private MaxCol As Integer
Private MaxRow As Long


Public Sub Attach(g As TrueDBGrid70.TDBGrid)
    Set TDBG = g
End Sub

' Returns number of rows
Public Property Get RowCount()
    RowCount = MaxRow
End Property

' Returns number of columns
Public Property Get ColCount()
    ColCount = MaxCol
End Property

' Retrieves value from our array
Public Property Get Value(ByVal Rows As Long, ByVal Cols As Integer)
    Value = GridArray(Cols, Rows)
End Property

' Sets value from array
Public Property Let Value(ByVal Rows As Long, ByVal Cols As Integer, ByVal Val As Variant)
    GridArray(Cols, Rows) = Val
End Property

' Initialises array
Public Sub SetDims(ByVal Rows As Long, ByVal Cols As Integer)
    If Rows <= 0 And Cols > 0 Then
        ReDim GridArray(0 To Cols - 1, 0)
    ElseIf Rows <= 0 And Cols <= 0 Then
        ReDim GridArray(0, 0)
    ElseIf Rows > 0 And Cols <= 0 Then
        ReDim GridArray(0, 0 To Rows - 1)
    Else
        ReDim GridArray(0 To Cols - 1, 0 To Rows - 1)
    End If

    MaxRow = Rows
    MaxCol = Cols
End Sub
```

```
' Fired when a new record is added

Private Sub TDBG_UnboundAddData(ByVal RowBuf As TrueDBGrid70.RowBuffer, NewRowBookmark As
Variant)
    Dim col As Integer

    On Error GoTo NoAdd

    ' Add one row
    MaxRow = MaxRow + 1
    ReDim Preserve GridArray(0 To MaxCol - 1, 0 To MaxRow - 1)

    ' Cells that have been modified are non Null
    For col = 0 To RowBuf.ColumnCount - 1
        If Not IsNull(RowBuf.Value(0, col)) Then
            GridArray(col, MaxRow - 1) = RowBuf.Value(0, col)
        Else
            ' Fill in a default value
            GridArray(col, MaxRow - 1) = TDBG.Columns(col).DefaultValue
        End If
    Next col

    ' Return the bookmark for this row
    NewRowBookmark = MaxRow - 1

    ' Calibrate the VScroll bar
    TDBG.ApproxCount = MaxRow

    Exit Sub
NoAdd:
    RowBuf.RowCount = 0
End Sub

'' Fired upon deleting a row
'Private Sub TDBG_UnboundDeleteRow(Bookmark As Variant)
'    Dim row As Long, col As Integer
'
'    On Error GoTo NoDelete
'    MaxRow = MaxRow - 1
'
'    ' Shift all rows up by one
'    For row = Bookmark To MaxRow - 1
'        For col = 0 To MaxCol - 1
'            GridArray(col, row) = GridArray(col, row + 1)
'        Next col
'    Next row
'
'    ' Resize the array
'    If MaxRow > 0 Then
'        ReDim Preserve GridArray(0 To MaxCol - 1, 0 To MaxRow - 1)
'    Else
'        ReDim Preserve GridArray(0 To MaxCol - 1, 0)
'    End If
'
'    ' Calibrate the VSCroll bar
'    TDBG.ApproxCount = MaxRow
'    Exit Sub
'
'NoDelete:
'    Bookmark = Null
'End Sub

' Fired when the grid needs a bookmark or data

Private Sub TDBG_UnboundReadDataEx(ByVal RowBuf As TrueDBGrid70.RowBuffer, StartLocation As
Variant, ByVal offset As Long, ApproximatePosition As Long)
```

```
    Dim ColIndex As Integer, j As Integer
    Dim RowsFetched As Integer, i As Long
    Dim NewPosition As Long, Bookmark As Long
    Dim StartRow As Long

    Dim Cols As Long, Rows As Long
    Cols = RowBuf.ColumnCount - 1
    Rows = RowBuf.RowCount - 1

    RowsFetched = 0

    If IsNull(StartLocation) Then
        ' StartLocation reffers to either BOF (-1) or EOF (MaxRow)
        StartRow = IIf(offset < 0, MaxRow + offset, -1 + offset)
    Else
        ' StartLocation is an actual bookmark
        StartRow = StartLocation + offset
    End If

    For i = 0 To Rows
        Bookmark = StartRow + i
        ' If we are out of bounds quit this loop
        If Bookmark < 0 Or Bookmark >= MaxRow Then Exit For

        ' Fill the RowBuffer with data
        For j = 0 To Cols
            ColIndex = RowBuf.ColumnIndex(i, j)
            RowBuf.Value(i, j) = GridArray(ColIndex, Bookmark)
        Next j

        ' Assign a bookmark for this row
        RowBuf.Bookmark(i) = Bookmark
        ' Increment number of rows fetched
        RowsFetched = RowsFetched + 1
    Next i

    RowBuf.RowCount = RowsFetched

    ' Callibrate the VScroll bar
    If StartRow >= 0 Then ApproximatePosition = StartRow
End Sub

' Fired when the user modifies an existing row

Private Sub TDBG_UnboundWriteData(ByVal RowBuf As TrueDBGrid70.RowBuffer, WriteLocation As
Variant)
    Dim col As Integer

    On Error GoTo NoWrite

    ' Write modified cells
    For col = 0 To RowBuf.ColumnCount - 1
        If Not IsNull(RowBuf.Value(0, col)) Then
            GridArray(col, WriteLocation) = RowBuf.Value(0, col)
        End If
    Next col

    Exit Sub
NoWrite:
    RowBuf.RowCount = 0
End Sub
```

NRC FORM 335 (2-89) NRCM 1102, 3201, 3202

U.S. NUCLEAR REGULATORY COMMISSION

BIBLIOGRAPHIC DATA SHEET

(See instructions on the reverse)

1. REPORT NUMBER (Assigned by NRC, Add Vol., Supp., Rev., and Addendum Numbers, If any.)

NUREG/CR-6525, Rev.1

2. TITLE AND SUBTITLE

SECPOP2000: Sector Population, Land Fraction, and Economic Estimation Program

3. DATE REPORT PUBLISHED

MONTH	YEAR
8	2003

4. FIN OR GRANT NUMBER

Y6594

5. AUTHOR(S)

N.E. Bixler, SNL
S.A. Shannon, SNL
C.W. Morrow, SNL
B.E. Meloche, SNL
J.N.Ridgely, NRC

6. TYPE OF REPORT

Technical

7. PERIOD COVERED *(Inclusive Dates)*

8. PERFORMING ORGANIZATION - NAME AND ADDRESS *(If NRC, provide Division, Office or Region, U.S. Nuclear Regulatory Commission, and mailing address; if contractor, provide name and mailing address.)*

Sandia National Laboratories
PO Box 5800
Albuquerque, New Mexico 87185

9. SPONSORING ORGANIZATION - NAME AND ADDRESS *(If NRC, type "Same as above"; if contractor, provide NRC Division, Office or Region, U.S. Nuclear Regulatory Commission, and mailing address.)*

Division of Risk Analysis and Applications
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, D.C. 20555 -0001

10. SUPPLEMENTARY NOTES

J. N. Ridgely, NRC Project Manager

11. ABSTRACT *(200 words or less)*

SECPOP2000 uses the U.S. Census Bureau released 2000 census population and 1997 economic data to estimate the population, land use, and economic data on a polar grid centered on at prescrib ed grid coordinate that can be located at any point in the continental United States. SECPOP2000 can produce a site input fi le for the MELCOR Accident Consequence Code System (MACCS2) computer code.

12. KEY WORDS/DESCRIPTORS *(List words or phrases that will assist researchers in locating the report.)*

SECPOP
MACCS2
population distribution
site population

13. AVAILABILITY STATEMENT

unlimited

14. SECURITY CLASSIFICATION

(This Page)

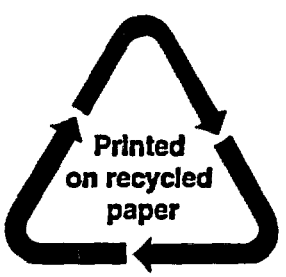
unclassified

(This Report)

unclassified

15. NUMBER OF PAGES

16. PRICE



Federal Recycling Program

UNITED STATES
NUCLEAR REGULATORY COMMISSION
WASHINGTON, DC 20555-0001

OFFICIAL BUSINESS