CAMA 2000

The Manual

by MicroSolve

CAMA 2000

CONTENTS

1 INTRODUCTION 1 – 1

2 GETTING STARTED 2 – 1
INSTALLING CAMA, INSTALLING APEX, STARTING CAMA, SECURITY, RENAMING A DATABASE

3 DISPLAYING AND MODIFYING DATA 3 – 1
DATABASE FILES, THE FIND OPTION, RELATIONAL STRUCTURE, BROWSE MODES THE FILTER, DISPLAY/INPUT, SKETCHES, PHOTOS, MEMOS

4 COMPARABLE PROPERTIES 4 – 1
BACKGROUND, SELECTION ALGORITHM, BUILDING A MODEL, INCLUSIONS AND EXCLUSIONS, SEPARATE SALES FILE

5 THE COST APPROACH 5 – 1
COST TABLES, EXCEL LINK, REPORT TEMPLATE, USIT PROGRAM, RUNNING THE COST APPROACH, ERROR REPORTING

6 USER'S SIMPLIFIED INSTRUCTION TABLE (USIT) 6 – 1
FORMAT, REPORT VIEW, MODIFYING THE TABLE, PROGRAM FLOW SPECIAL CHARACTERS, FUNCTIONS, FORMAT OPTIONS

7 BASIC STATISTICS 7 – 1
SALES RATIO ANALYSIS, BASIC STATISTICS, FREQUENCY DISTRIBUTIONS, HISTOGRAMS

8 SPSS STATISTICAL FUNCTIONS 8 – 1
CONFIGURING, RUNNING REGRESSION (SETUP, ANALYSIS), IMPORTING PREDICTED VALUES, APPLYING COEFFICIENTS

9 DATA DICTIONARY 9 – 1
LAYOUT, FACTOR TYPES AND PARAMETERS, FACTOR BEHAVIOR AND PLACEMENT, REPOSITIONING, FACTOR EQUIVALENCE TABLE, ADDING/MODIFYING/DELETING

10 REPORT DESIGNER 10 – 1
WINDOW OPTIONS, REPORT CONTROLS TOOLBAR, REPORT MENU DATA ENVIRONMENT, DIALOG BOXES, PROPERTY RECORD CARDS

11 IMPORT/EXPORT AND OTHER UTILITIES 11 – 1
IMPORT WIZARD, USER MAINTENANCE, ERROR LOG, DBC MAINTENANCE, SALES HISTORY

12 INCOME APPROACH 12 – 1

CAMA 2000

1 ■ INTRODUCTION

Welcome to CAMA 2000, MicroSolve's premier Computer Assisted Mass Appraisal program. This manual is designed to help you get started, to keep you going, and along the way to answer questions about the great range of features packed into this multi-faceted program.

What is CAMA 2000?

CAMA 2000 is a versatile, interactive database application used by assessors and appraisers to maintain property databases and to determine property values. Developed by MicroSolve Corporation, CAMA 2000 integrates industry-standard practical programs – for valuation, sketching, statistical analysis, photo imaging, and report writing – with its own interactive database structure. CAMA 2000 has a standard graphical user interface, using drop-down menus, on-line help, and dialog boxes. It will work under any Windows operating system starting with Windows 95, up to the systems that are current at the time of delivery.

What can CAMA 2000 do?

CAMA 2000 offers a full spectrum of practical tools for assessing professionals. It will allow you to maintain databases and determine property values, using any one of several widely accepted valuation techniques. It is designed to simplify complex operations, enhancing your productivity. Some of the skills that CAMA 2000 supports include:

- storing and analyzing data
- adding, deleting, and modifying records
- valuing properties
- sketching and photo imaging
- performing statistical analysis
- graphing
- report writing

The system offers many new improvements and features designed to simplify assessment procedures and maximize your effectiveness. Based on state-of-the-art advances in database techniques and graphic presentation, CAMA 2000 includes innovations such as:

- user-modifiable data forms
- multi-level modifiable relational database
- complete report writer
- user programming capabilities

These and other tools make the system extremely configurable and responsive to the needs of each jurisdiction in which it is installed.

How is this manual organized?

As you read this manual, you will discover many features of the program that will offer practical solutions to specific assessing problems. It is a good idea to practice the skills and lessons as you proceed through this manual. To reinforce these skills, you should test yourself with the "how-do-I" lessons and "what-if hypothetical" problems included in each workbook chapter. The chapters are organized to explain the tasks sequentially, using associated graphics and guiding you through each step of the process.

- Chapters 1 and 2 provide introductory information about the program to help get you started.
- Chapter 3 explains how to display and modify data. It explores the DATA menu option, encompassing the Display/Input and Browse views. It provides a practical context for the features on the display form. You will learn how to add/delete records or parts of records, sketches, and photos. Moreover, you will navigate among the records, setting the index and filter.
- Chapters 4 and 5 explain the major valuation methods: comparable properties and the cost approach. You will learn how to complete a comparables routine, run a cost approach, and set parameters that affect the values produced. You will also learn how tables and user-modifiable instructions interact to produce property values.
- Chapter 6 presents the features of Usit (the User's Simplified Instruction Table) as a reference for users who may need to create or modify a routine.
- Chapter 7 is a guide to the basic statistics that are built into CAMA 2000. Here you will learn how to conduct a sales ratio study, to produce useful statistics such as the coefficient of dispersion and the price-related differential, and how to display histograms and scatter diagrams that provide quick visual snapshots of your data.

- Chapter 8 explains how the SPSS statistical package is integrated into CAMA 2000 and can be used to perform multiple regression and other statistical analysis techniques..

- Chapter 9 discusses the Data Dictionary, a key tool in the program's ability to adapt to user needs and specifications. Using this tool you will be able to control the appearance of your data display and input screens and the order in which factors are presented.

- Chapter 10 presents the report designer, the Visual FoxPro tool that makes it possible for users to develop custom reports involving data, text, and illustrations.

- Chapter 11 discusses the import and export functions of the system, as well as other important utilities for maintaining peak database performance.

- Chapter 12 presents the income approach and shows how to customize that process using simple spreadsheet tools.

Although this manual is a practical guide, providing task-oriented steps, some users will surely want to know more about the conceptual framework of the database system. These users will find, printed in smaller type, a limited amount of more advanced information on the structure of the CAMA 2000 database environment. Users whose concerns are more practical can skip this supplementary material, since it is not essential for operation of the program.

CAMA 2000

Installing CAMA 2000

MicroSolve CAMA 2000 operates under Windows – whichever version you have on your computer or file server. To install CAMA 2000, follow these steps:

- Insert the installation CD into the d: drive (or other designated CD drive).
- From the Start menu, select Run.
- In the Open field, key in **d:setup** or use the Browse button to find the setup on the installation disk.
- Click OK to start the CAMA 2000 installation. Follow screen directions and prompts. If you are installing on a network server, see the special instructions supplied with the installation disk.

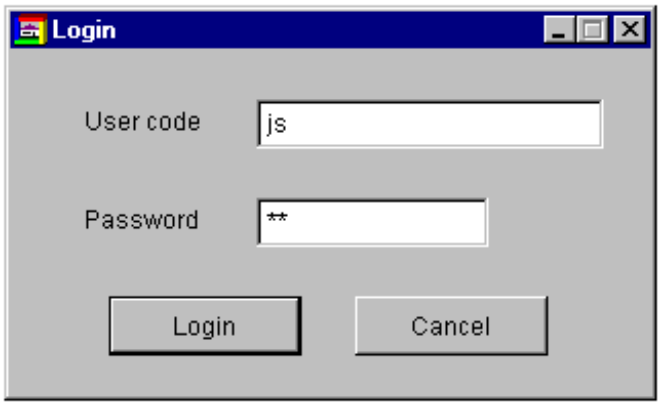
Installing Apex IV

An integral component of CAMA 2000 is Apex, the assessor's sketch program. Apex is sold as a stand-alone system, but it is also integrated with MicroSolve's CAMA 2000 and will run from within that program. The Apex installation files can be found in the **apex_iv** folder created in the **msol** folder during CAMA 2000 installation. To install Apex, follow this procedure:

- Select Run from the Start menu.
- In the Open field, key in **c:\msol\apex_iv\setup** or use the Browse button to locate the setup program in the **apex_iv** folder.
- Click OK to start the installation process. Follow screen directions and prompts, noting licensing information.

Starting CAMA 2000

Create an icon-shortcut to CAMA 2000 and place it on your desktop. Double click it to launch the program. A log-in screen will prompt you to key in your user code and password. Then click on **Login** or press <Enter>.

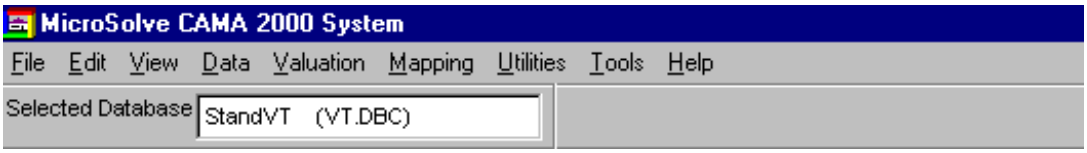


Password security

Users can access the various CAMA 2000 features consistent with their assigned security level.. Security levels are defined according to purpose and policy. However, only users with A-level security (highest level) have authority to add, modify, or delete passwords. For more information on this option, refer to **CAMA Utilities** in chapter 11.

Main screen

Once you have logged on, you will see the main MicroSolve CAMA 2000 screen. Various menu options are displayed at the top.



The system automatically signs on to the database most recently accessed.

How to rename a database

Sometimes it is necessary to create a new database by starting with an old one and renaming it so it can be used in a new context. This can only be done if you have a copy of Visual FoxPro. It is a step that should not be undertaken by people without basic computer knowledge and skills.

The following paragraphs describe how to rename a database in FoxPro and how to update CAMA 2000 so that it recognizes the new database name. In this example we will change a database named VT.DBC with a descriptive name of “StandVT” to RESI.DBC with a descriptive name of “StMary.” This explanation assumes the reader has some knowledge of FoxPro, the Windows operating system, and MicroSolve’s CAMA 2000. It

is also assumed that CAMA 2000 is installed on the C: drive in the MSOL folder. IMPORTANT: *Read the entire section before starting this process.*

Step 1 – Rename the database container and folder

Go into the Microsoft file explorer and rename the following files in the C:\MSOL\DATA\VT folder:

Rename VT.DBC to RESI.DBC
Rename VT.DCT to RESI.DCT
Rename VT.DCX to RESI.DCX

You can also rename the folder containing this database. If you call it STMARY you will get the following path to the data: C:\MSOL\DATA\STMARY.

(If you have created a new folder or directory called STMARY and have copied the files from C:\MSOL\DATA\VT into it, then you will go into this folder in the first place to rename the files, and you will not need to rename the folder.)

Step 2 – Repair the table back links to the database

Launch Microsoft Visual FoxPro. In the command window type the following commands to repair the database:

```
SET EXCLUSIVE ON
OPEN DATABASE C:\MSOL\DATA\STMARY\RESI
VALIDATE DATABASE RECOVER
```

The system will prompt you to confirm the creation of the back links to the new database name. Answer yes to this question each time it comes up (be patient – there are quite a few tables in the database container).

Rerun the VALIDATE DATABASE RECOVER command and make sure the message "Database container is valid" appears in the main FoxPro screen.

Step 3 – Replace the descriptive name of the database

Once again in FoxPro, open and replace the database's descriptive name with "STMARY" in the main definition file by typing the following commands in the command window:

```
USE C:\MSOL\DATA\STMARY\MAINDEF.DBF
REPLACE ALL CDESIGN WITH "StMary"
```

You have successfully renamed the database. You can now exit FoxPro and launch CAMA 2000. Find the new database with the **Find** button on the Change DBC screen form and it will appear in your database list.

Making sure the system recognizes your database

CAMA 2000 maintains a master list of the databases under the DATA subfolder. This list is recreated each time you sign on to the system. It should always show all the available databases under \MSOL\DATA. The databases are identified by DBC name, folder name, and alias (or the "nickname" assigned in the MAINDEF file). Often the easiest way to locate a database is through the name of the folder containing it, and in fact the list is alphabetized by folder name.

If a database that should be present does not show up in the list, the most common reason is that the DBC (database container) is missing. If that is the case, the database cannot be opened and the program will not recognize it when compiling the list of databases.

REVIEW QUESTIONS

1. How do you install CAMA 2000?

2. What is Apex? How do you install it?

3. How do you start CAMA 2000?

4. What level of security is required in order to change a password?

CAMA 2000

3 ■ DISPLAYING AND MODIFYING DATA

This chapter covers the following topics:

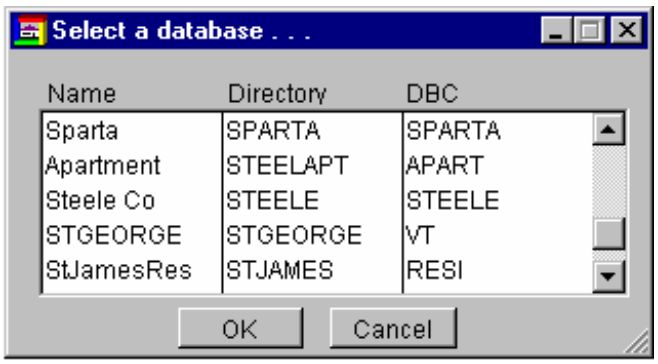
- Identifying and changing database files
- Using the Find option
- The CAMA 2000 relational structure
- Two primary modes of data display
- The data filter
- Entering and modifying data

Changing database files

When you first sign on, the system automatically opens the last database accessed. To open a different database, follow these steps:

1. Click on **File** in the menu bar, then click **Change DBC**.

2. In the Select DBC window there will be a list of available databases. If the database you want appears there, highlight it and click **OK**. Remember that all databases used by CAMA 2000 must be in folders under the Data folder, and all must have the extension dbc (database container).



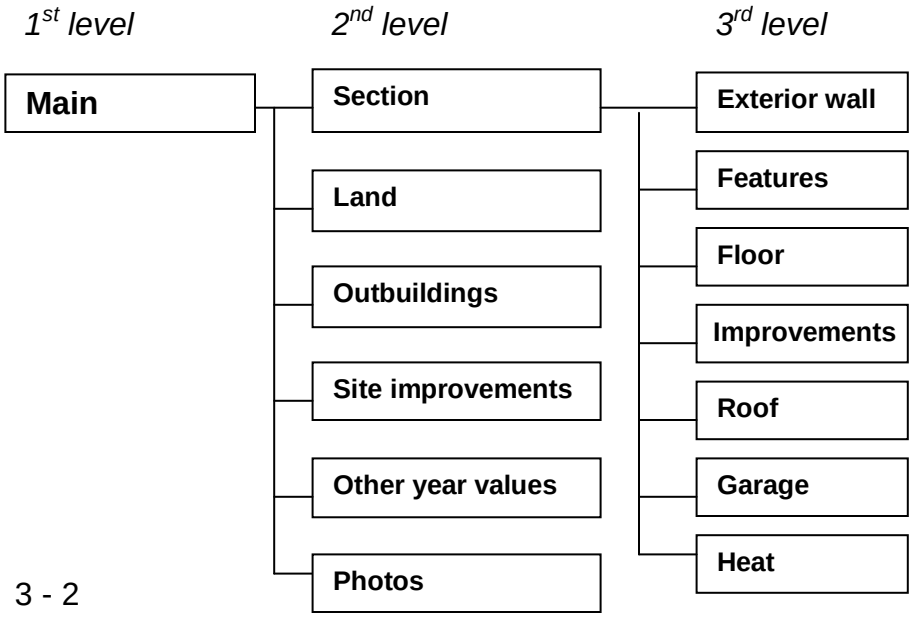
Understanding the relational structure of CAMA 2000

A database is a repository for various pieces of information describing an entity, such as a residential parcel. From a conceptual point of view, all the pieces of information, called fields or factors, pertaining to a particular entity are part of a single record. However, on a technical level, the elements of this record are made up of individual records in separate tables that are associated through the database container.

MicroSolve's CAMA 2000 uses a relational database structure, which offers significant advantages over non-relational or "flat" files. A relational database can best be thought of as a set of connected tables, related by a common feature. In a property database, for example, there is a table containing a record (a row) for each property, with columns containing identifying information such as parcel ID, owner name, address, etc. Sale price and total value may also be in this table. Since there can be several buildings or building sections on a property, these are recorded in a subtable, two of whose fields are parcel ID and section (or building) number. Using the parcel ID, all the sections pertaining to a particular parcel can be located. This subtable will also contain building characteristics such as area, quality, year built, plumbing fixtures, etc.

Each building in turn may have several porches whose size and other characteristics (floor, wall, roof, ceiling) are contained in a sub-subtable. In this table, porches will be located by a combination of parcel ID and building identifier. In a relational database you can have as many instances of a given feature as necessary, as long as there is a subtable pertaining to that feature.

Because the identifiers are indexed, they can be located very rapidly in any table. That means that all the characteristics of a property, regardless of their number or their distribution in tables, subtables, or sub-subtables, can be quickly assembled and displayed or used in analysis. Furthermore, when cost calculations are run, certain values can be immediately totaled and placed in relevant fields. For example, if there are several buildings on a parcel which is put through a cost calculation, the total value of all the buildings can be placed automatically in a field in the main table.

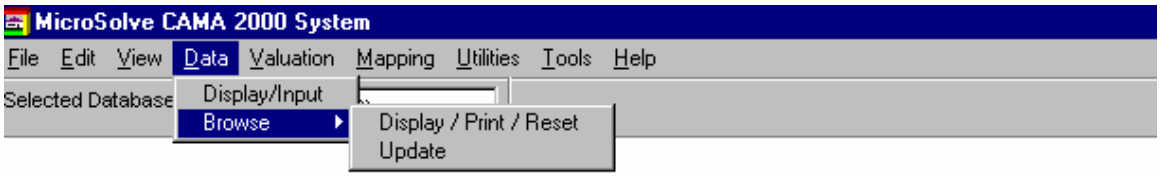


The relational database structure facilitates adding and updating records, viewing record data, performing analyses, and printing special reports. All records in the database are updated when changes are made to the entries in the data dictionary (see chapter 9). Changes are distributed to the related tables and subtables as part of the relational database structure. Thus, when factors are modified or fields added to a page in the data dictionary, the modifications will be mirrored in the record display form. Likewise, analyses, calculations, etc. will be based on the revised fields.

Two modes of data display

Sometimes it is desirable to view a particular record in its entirety, in order to examine or change all the factors that compose it. Sometimes it is preferable to view only selected features of many records in a tabular form, where each record is a row and each feature or factor is a column. These two options are embodied in the two modes of data display: *display/input* and *browse/reset*. We will describe the browse mode first.

Using browse



To begin a browse session, click on **Data**, then **Browse > Display/Print/Reset**. (The "reset" in the name refers to the capability, described below, of setting a field to a single value chosen by the user.) The form that appears has two tabs: **Browse** and **Field selection**. These functions work together. Clicking on **Field selection** enables you to determine which factors will be displayed in the browse window. Clicking on **Browse** displays all the data for the selected factors in a single table. To browse the records and view the chosen fields, follow these steps:

- Click the **Field selection** tab. To select individual factors, hold down **Ctrl** while clicking on any fields you want to select from the left-hand panel; then when all are highlighted click the single arrow to move them to the right-hand (selected) panel. Alternatively, you can double-click on one factor at a time and it will immediately move to the right-hand panel. Do not place more factors in the right-hand panel than you can see at one time.

- Arrange the factors in your preferred order by dragging the blocks at the left of the window up or down.

- If you wish to save these factors and this order for later display, click **Save Profile**. Give the profile a name. You can have any number of profiles. The one most recently used becomes the browse default.

- Click the **Browse** tab to view the records and fields you have chosen. Note that column widths can be adjusted while you are viewing the factors.

- If you wish to print a report consisting of the selected factors, click the **Report** button. Column widths in the printed report respond to changes you make in the browse grid.
- If you wish to set a particular factor to a constant value for all the records you are displaying, or to copy a value from another factor in the same record, highlight the factor to be changed, then click the **Reset** button. You can specify a new value and apply it to all records or a filtered subgroup (see below). Alternatively, you can specify another factor whose values should be copied into the current one.

Editing columns from the keyboard

The printable browse grid does not allow any of the data it displays to be modified. Because some users want to edit data in columns (moving up or down from record to record), an older form of the browse function is also available as a second choice (the **Update** option). It also lets users select factors, but it only permits the current setup to be saved for use in the next session. Data in this browse screen cannot be printed or globally reset, but they can be modified from the keyboard. Filter functions used with this browse option may not always be completely accurate.

Filtering data with Browse (and other functions)

Filtering allows you to determine the criteria for selecting the records that will be displayed. Essentially the process "filters in" the chosen records (or "filters out" unnecessary ones), so that you can narrow your focus. To filter your data, follow these steps:

- While in Browse mode, click the filter (crosshatched) button on the database toolbar.

- On the Filter panel, click the **Add** button.

- Choose the table and field on which you want the expression to be based.

- Choose the operator (equals, is greater than, is less than, etc.) from the available options. Place a number in the **Value** field. Then click the **OK** button.

- If other expressions must be combined with the first one, click **Add** on the filter panel, choose the appropriate conjunction (*and* or *or*), and repeat the above process.

When you are finished, you can click **Count** to get the total number of records that match your criteria, or click **OK** to return to the browse screen, which will now display only the filtered records. Even after you close the browse window, the filter will remain active until you change it or end the program.

Any actions that process a number of records will observe the filter. For example, if you now run the cost approach in batch (*run all*) mode, only filtered records will be processed. If you run descriptive statistics, the results will be based only on the selected records. HOWEVER:

- All records can still be accessed for individual display and data entry.

- All records will show in the pick list for comparable properties, even though only filtered records will be considered in choosing the comparables. This means that any record in the database can be a subject, but the population searched for comparables will be that defined in the filter setup.

Example: You are in the browse mode. Suppose you wish to display only records of parcels that sold for a price between $50,000 and $100,000.

- Click on the filter icon near the top of the screen, then click **Add**.

- In the **Field** box, locate and click on Main [Sale Price].

- In the **Operator** box, find and click on the expression "is greater than or equal." In the **Value** box, type 50000. Click **OK** at the bottom of the panel.

- Click on **Add** at the bottom of the Filter window.

- Again locate and click on Main [Sale Price] in the **Field** box. Choose "is less than or equal" in the **Operator** box.

- Key in 100000 in the **Value** box to complete the expression. Click the **OK** button at the bottom of the panel.

The complete expression in the window should now read:

MAIN [Sale Price] is greater than or equal 50000
and MAIN [Sale Price] is less than or equal 100000

You can click on **Count** to determine how many records meet your criteria. Then you can click on **OK** to return to the Browse screen. There you will see (if you have included sale price as one of the browse factors) that all the records now displayed have a sale price within the specified range.

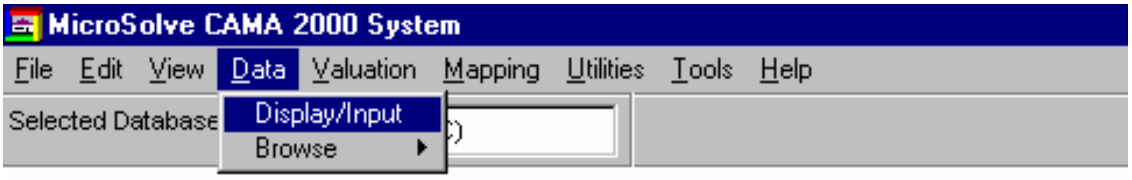
You can add further elements to the filter expression if you wish. If you need to select not only records in the specified price range but also those that sold after September 30, 1998, you can add another line to the filter expression.

Display/input

The same form is used both for displaying an entire record and for entering data in that record. Any user with the appropriate security level can enter data in a record at any time. When you wish to access the display/input form, all other windows, including Browse, must be closed.

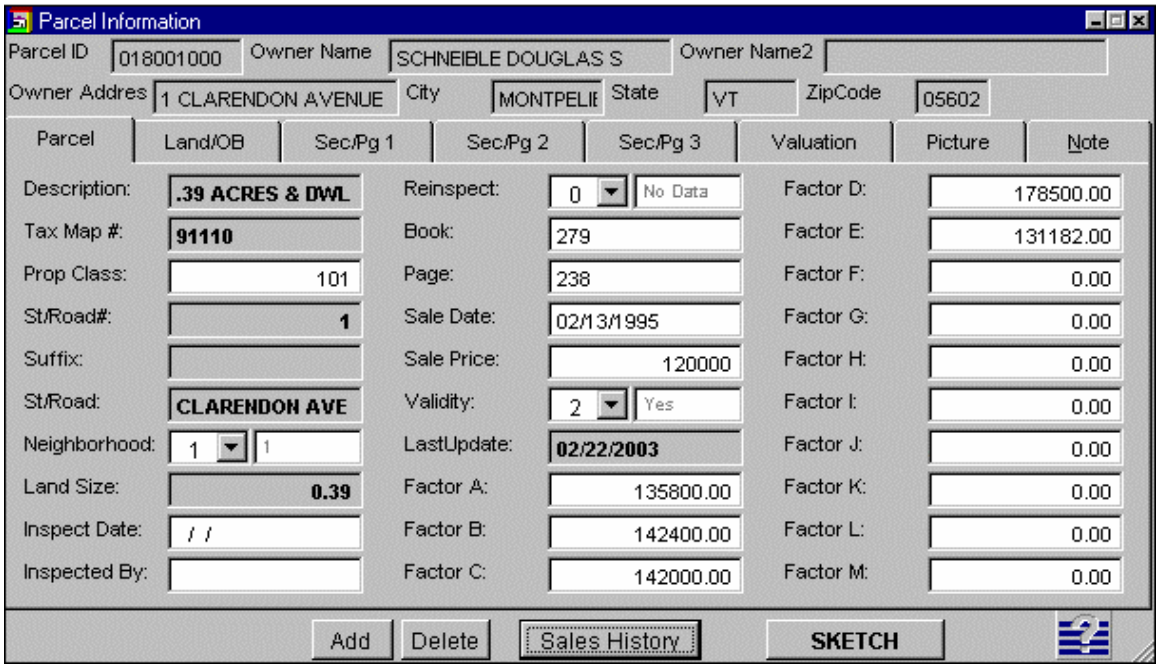
To display a data record:

- Click on **Data** in the toolbar.
- Click on **Display/input**.



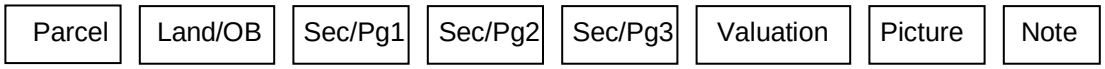
The next screen displays the features of a parcel record from the database you are working in. The record shown is the last one selected using the navigation arrows or the "binoculars," but you can readily select a new one using these same tools.

This compact form, with its numerous tabs, contains all the information in a given record, including character and numeric data, notes, sketch, and photo.



Structure of the display form

Pages and tabs. A property potentially has hundreds of characteristics. This information is stored and displayed in a stacked series of pages, each with a labeled tab at the top. In the display form above, the tabs are titled, from left to right:



Placement of the information fields on these pages is controlled by the Data Dictionary, which is accessible to users with suitable security (**Utilities/Data Dictionary**). The Data Dictionary can be modified to display or not display individual factors on any page, in any order. For a comprehensive discussion of the Data Dictionary, see chapter 9.

Factors that identify the record. Factors such as "Parcel ID," "Owner Name," "Owner Address," etc. are identifying factors and are generally shown in the first two lines at the top of the display form. These lines are always visible, regardless of the tab the user selects. The parcel identifier is central to the relational database and cannot be changed once it is created.* Users can, however, use the Data Dictionary to indicate which factors are to be displayed in this section and how much space is allocated for each, provided not more than two lines are used altogether.

Pages, rows, and columns. As explained above, each page is headed by a tab bearing a unique name. Factors are arranged in ten rows and three columns per page. Data are intended to be both stored and viewed from "north to south," moving down the rows in the first column, then up to the top of the next column, and so on. Thus, when factors are added to a page by means of the Data Dictionary, each column is filled before the next is started.

Factor types. Factors can be character fields, numeric fields, or date fields. Users can determine the minimum and maximum value allowed for each. The parcel-identifying factors ("Owner Name," "Address," etc.) are typically character fields. Zip codes should also be created as character fields, for two reasons: (1) a numeric field cannot begin with a zero, and (2) character fields allow for international, alphanumeric postal codes.

A distinct type of numeric field is the categorical factor, which allows the user to select from a series of pre-established levels, each with its own name and assigned value, known as the "linear value." Linear values are essentially quality ratings using 100 as an average index reference. A categorical factor for "Exterior Wall" might have the following levels:

---

* If you need to change a parcel identifier, use the following procedure: Click the + button near the top of the screen, in the same series where you find the binoculars. This allows you to add a parcel to the database. You can specify whether you want to add a new parcel or copy an existing one. In this case specify "copy." Then type in the new identifier and click OK. A new parcel will be created with all the characteristics of the present parcel. After that happens you may delete the existing "old" parcel, using the X button at the top of the screen.

Level no.	*Name*	*Linear value*
1	Plywood	80
2	Conc. blk	90
3	Vinyl	100
4	Wood	110
5	Shingle	110
6	Stone	120

While not used in the standard Marshall & Swift cost approach, linear values are used in many other places, including the Comparable Properties routine and Multiple Regression Analysis. It is therefore desirable to set appropriate linear values for those categorical factors that may affect property estimates. Choosing linear values is a matter requiring the professional judgment of an experienced appraiser.

On the display form, a categorical factor has a "spinner" or arrow next to its associated value window. This arrow brings up a pull-down chart listing the available lines for that factor. Level 0 of a categorical factor is always "No Data." Categorical level names and the associated linear values can be modified using the **Categorical factors** option under **Utilities**. (See chapter 11.)

Calculated fields. Some fields, called calculated fields, are set up for display and not for user input. You can easily identify these fields by their darkened windows which do not permit direct data entry. The values in calculated fields are usually produced by cost approach calculations, based on property characteristics. To change the calculated value, you must change the characteristic(s) on which it is based, then rerun the cost approach.

For a complete discussion of the cost approach, see chapter 5.

Adding, copying, or deleting a record

Basic utilities for manipulating records are controlled from the toolbar at the top of the screen. To add a parcel, click the + button. You are given an opportunity to enter a new parcel ID. (It must be different from any other parcel ID in the database.) You can indicate whether this record is a completely new one or intended as a copy of the record you are currently on. A new record will have no information other than the new parcel ID; a copy will have the new parcel ID but all other information will be copied from the current record.

If you mistakenly click the + to add a new record and then want to back out, you should leave the ID field blank (or remove anything presently in there), then click **OK**. The system will say, "A record with a blank parcel ID cannot be added." Click **OK** and you're back at the original record.

To delete a record, click on the X button in the toolbar. The program will ask if you're sure. All associated subtable records pertaining to this parcel ID will be marked for dele-

tion. (*Note: When a record is marked for deletion it is no longer accessible from within CAMA 2000. However, it has not been wiped out of the database. Users who have Fox-Pro can restore a deleted record, though it is best to seek guidance from MicroSolve's technical staff before doing so. Running DBC Maintenance in the Utilities option will "pack" the files – meaning that deleted records are eliminated for good.*) Global deletes are not possible within CAMA 2000.

Subtable data

As noted earlier, the records in a relational database are contained in subtables organized around particular property features, such as land, outbuildings, porches, attached garages, and the like. It is a good idea to know, when adding or modifying data, which subtable is being changed. At some points it may be necessary to add a further record to a subtable, in order to accommodate another porch, for example. And sometimes it is necessary to delete a record from a subtable.

Adding instances of a characteristic. Consider a property for which you need to record several different types of land. By default, the land table starts with a blank level in which you can record the first land type. This level is known to the system as **Land ID 1**. Let us enter land type 1 (building lot) for this ID, indicate (in Calc Method) that it will be priced as a site, and give it an area of .5 acres and a grade of 110 (i.e., 10 percent better than average).

The property also includes a wood lot, so we need to add a second level to the land table, using the following steps:

- Click on the **Add** button at the bottom of the form.
- Click on **Land** in the window that appears, then click **OK**.
- When asked if you want to save the first set of values, click **yes**.
- Enter the appropriate values for the second land type.

Repeat this process for each additional land type you wish to add.

If you are visualizing the mechanics of a relational database, you will understand that you are adding lines to a subtable for land each time you go through the above routine. All the lines will be associated with this parcel because all contain the parcel ID in addition to the land ID.

Deleting an instance. It may be necessary to delete information that has been recorded in a subtable. Suppose you wish to remove Land ID 2 and its associated values. First display Land ID 2 on the Land/OB page, then click the **Delete** button at the bottom of the parcel form page. This will remove Land ID 2 and its associated values. Technically, this step

deletes a line from the Land subtable. If there are additional land types above ID 2 (such as type 3 and 4), they are renumbered after a deletion. What was ID 3 becomes ID 2; what was ID 4 becomes ID 3. The information associated with them is preserved.

NOTE: Every table must have at least one level for each parcel ID, even if that level contains no data. If you try to delete level 1 from a subtable, the program will only remove non-zero values from its factors, but the level will remain.

Adjunct data: sketches, photos, memos

Adding or deleting a sketch. Sketching in CAMA 2000 is done by means of the Apex program, which is integrated into the package. Apex is installed at the time you install CAMA 2000; thereafter it can be called up whenever you are in the data entry mode. In the Apex Manual (which supplements this manual) you will find detailed documentation of the Apex program and procedures. Here we simply describe how to create a sketch for a particular parcel and how to delete a sketch that already exists.

To create a sketch

- With the data entry form displayed for the parcel you want to sketch, click on the Sketch button in the lower right-hand corner of the form. (If it is not in boldface, there is no pre-existing sketch.) A standard database browse window will appear. You can use it to search for an existing sketch (perhaps made previously) that you now want to associated with this parcel, or you can click on the New button at the bottom of this window.

- If an existing sketch is now associated with the parcel, you can modify it as needed. If not, you can create one, using the techniques described in the Apex Manual.

- When you finish, it is not necessary to save the sketch in JPG format, even if you intend to print it later on a property record card. Apex has the ability to create a JPG sketch “on the fly” when you go to print a report.

- Click on the top x in the upper right corner to save the sketch and exit Apex. The sketch will automatically be stored in the parcel record. However, if you have set the program to transfer calculated areas automatically from the sketch to designated factors in the CAMA 2000 database, this will not happen until you either move to another record or click on the diskette icon in the toolbar at the top of the screen. After that the letters on the Sketch button will be in boldface, indicating a sketch exists for that record.

To delete a sketch:

- If the Sketch button in the lower right-hand corner of the data entry form is in boldface, you can delete the sketch by right-clicking on the button.

- You will be asked to confirm the deletion. If you click Yes, the sketch will be removed

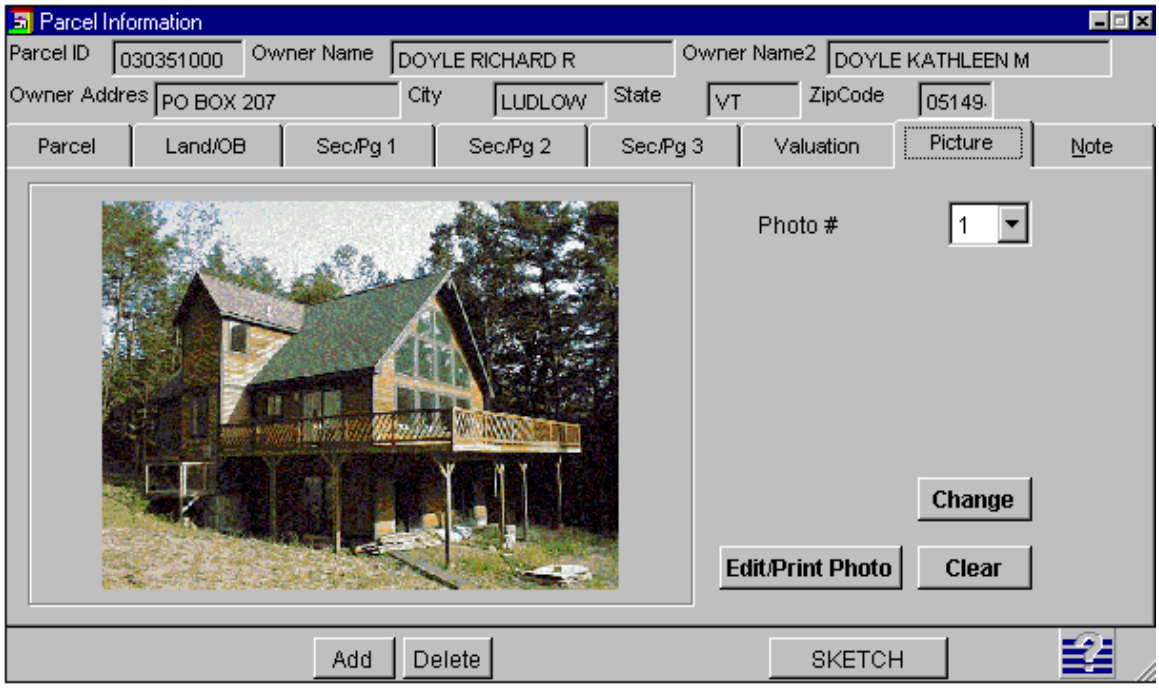
Photos. You can view and replace photographs for a property from the photos page on the parcel form. More than one photo of a property can be part of the record. To add a photo to a parcel, follow these steps:

- Click on the photo tab. If the photo area is blank, you can use the **Change** button to locate a picture. If it is not blank and you want to keep the picture that is already there but add another picture to it, use the spinner to advance to the next available photo identifier (or add another photo record using the **Add** button on the form), then click on **Change**. If you want to replace the picture you see with a new one, just click on **Change**. If you want to get rid of a picture that is currently displayed, click on **Clear**.

- Use the standard Microsoft "Open" window to search among the folders on your disk for the photo file you want. The default file type is .JPG but you can set the program to display all files, so you can access those in .GIF or .BMP or other photo storage formats. Once you have found the file you want, highlight it and click OK. It will now be associated with this parcel record.

Clicking on the Edit/Print button brings up the Microsoft Internet Explorer with the picture displayed. Depending on your system resources, you can open a picture editing program that will allow cropping, resizing, etc. You can also print the picture.

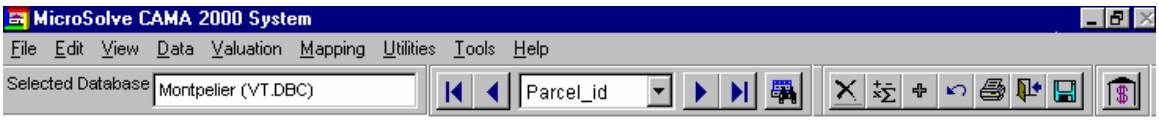
Notes or memos. The last tab on the parcel record form is for Notes. These can consist of assessor's comments, legal descriptions, memoranda, etc. You may input any amount of text directly from the keyboard. You can also cut, copy, and paste material from other sources into a notes field using standard Windows edit procedures.



Navigating among records with the toolbar

When you first opened the data entry form, a new toolbar was displayed directly below the top menu (see below). From this toolbar you can:

- change the index governing record order
- move forward or backward through the records
- search the records
- add or delete records
- filter the records



Setting the index. When a browse or data display/input form is on the screen, the current order of records is displayed in a small window beneath the top menu, to the right of the window showing the database you're signed on to. As you move from record to record

using the navigation arrows, you are proceeding through an indexed list. To change the order in which the records are shown, you must change the active index. Using the pull-down arrow at the right of the small display window, you can display and select any existing index. As soon as you do this, the order of that index will govern the presentation of data.

Indexes are rebuilt automatically as records are updated. If, however, you receive an error message saying an index does not match the data table, you will need to rebuild the index "manually."

- Click Utilities in the menu bar.
- Choose DBC Maintenance.
- Be sure all other users are logged off before proceeding The system needs exclusive use of the file to perform this task.
- The system validates the database and rebuilds the indexes. If any tables are missing from the database, you will be notified here. Otherwise the message announces, "Database container is valid."
- Close the temporary window. At this point the individual tables are packed and reindexed. This procedure ensures relational integrity.

To display records in the order in which they were originally entered, select "No order" from the display window.

Moving forward or backward through records. Clicking on the horizontal arrows to the right and left of the display window will advance or "rewind" one record in the indexed order. You can advance all the way to the end or revert to the beginning using the arrows with the vertical bars.

Searching the records. To locate a particular record quickly, click on the **binoculars** icon in the toolbar. A popup window showing a table of identifiers and a blank line appears. If the identifier you want (such as Parcel ID or Owner Name) is not highlighted, click on it. Once it is highlighted, the records below it are ordered according to that identifier. You can then scroll down to find the desired record, or you can type the first few characters of the identifier (such as owner name) to locate the record immediately. Click on Select once you have found the record you want.

Costing from Display/Input

The cost approach can be applied to any record that is being examined or updated via the display/input form. Simply click on the cost button  in the toolbar at the top of the screen. The cost report form will be displayed when the record is costed, and after you close the form, you will find that the appropriate fields in the display/input form have been updated with current cost values. If you have chosen "no print" in the report selection option under "Valuation > Cost Approach," no cost sheet will be displayed, and the fields in the display/input form will simply be updated

Be sure to save the new values (by clicking on the diskette icon in the toolbar) before moving on to the next record, unless you are just testing out "what-if" possibilities and do not wish to update your record permanently.

REVIEW QUESTIONS

1. How do you display selected factors for all records in your database?

2. How do you tell the system to display only records selected according to certain criteria?

3. How do you display an entire record with an option to modify its fields or factors?

4. How do you select a different record by ID or owner name?

5. How can you navigate through your database according to various indices?

6. How do you create a new record?

7. How can you cost a record while you're viewing or updating it on the screen?

8. What steps do you take to add a sketch or a photo to a record?

4 ■ COMPARABLE PROPERTIES

This chapter deals with one of the oldest and most basic appraiser's tools, used both in generating values for properties and in defending values established by other means. The computerized version of the comparables approach greatly simplifies the selection of comparable properties and makes the process more accurate and reliable. In this chapter you will learn:

- To establish or modify parameters, using the comparables setup form
- To complete a comparable properties routine resulting in a report

Background on comparable properties

The comparable properties routine is a specialized search technique. When you specify a subject parcel and indicate to the computer that you want to find the five parcels that are most like it in terms of certain particular characteristics (examples might include size, age, exterior finish, number of rooms, building quality, etc.), the computer searches through the database and displays the parcels that come closest to the subject parcel on those characteristics.

You can control the role each characteristic plays in the selection. To a computer a difference of one room between two parcels matters no more than a difference of one square foot. So you need to specify how much weight to give to each factor; therefore the *weighted* difference becomes the basis for selection.

In assigning a weight, you have a means of fine-tuning the selection. Since it is unlikely that five parcels will be found, all of which are identical to the subject parcel, it is often important to let the computer know which characteristics or qualities are most important. For example, you might feel that a difference in size or quality of a dwelling should be more important in the selection process than a difference in age. Therefore, you may wish to weight those factors more heavily than age, saying to the computer, in effect, "If you have to choose between a property that's nearly the same age as the subject but smaller and one that's nearly the same size but older, pick the latter."

The comparables selection process is often used as a way of valuing properties or of verifying a value arrived at by other means. When it is used this way, you must restrict the universe of possible comparables to those properties that have recently sold. Then you specify an "adjustment" value that must be applied to each factor difference. This value represents the worth, in dollars, of the factor difference. For example, suppose a subject property (A) has 8 rooms and was built in 1975. The most comparable property (B) has 7

rooms and was built in 1980. We know that property B sold for $200,000. What can we say about A?

That's where appraisal judgment comes in. Let's say you set the value of a room at $10,000, and the value of a year of difference in age at $1,000. Then because property B has only 7 rooms, we would want the computer to add $10,000 to its selling price to estimate what an 8-room house would be worth. However, property B is newer by 5 years. If each year of difference is worth $1,000, we would subtract $5,000 to determine what a similar house would be worth had it been built 5 years earlier. So the net estimated value of property A would be $205,000 ($200,000 + 10,000 – 5,000).

The comparables selection algorithm

In sum, the comparables routine allows users to establish weights and adjustments for each of the factors specified as a basis for selection. Mathematically, what the computer is doing is calculating a score for each possible (candidate) parcel, based on the sum of the weighted differences between its factors and those of the subject parcel:

$$Q = \Sigma\left(w_i \times (sf_i - cf_i)^2\right)$$

– where w_i represents the weight of the i^{th} factor in the list of factors used as a basis for comparison, sf_i represents the value of the i^{th} subject parcel factor, and cf_i represents the value of the i^{th} candidate parcel factor. (Because in this algorithm the factor difference is squared, thus eliminating the distinction between positive and negative differences, it turns out that the weight should be proportional to the square of the dollar value of a difference between the factors. This issue will be discussed in greater detail later.)

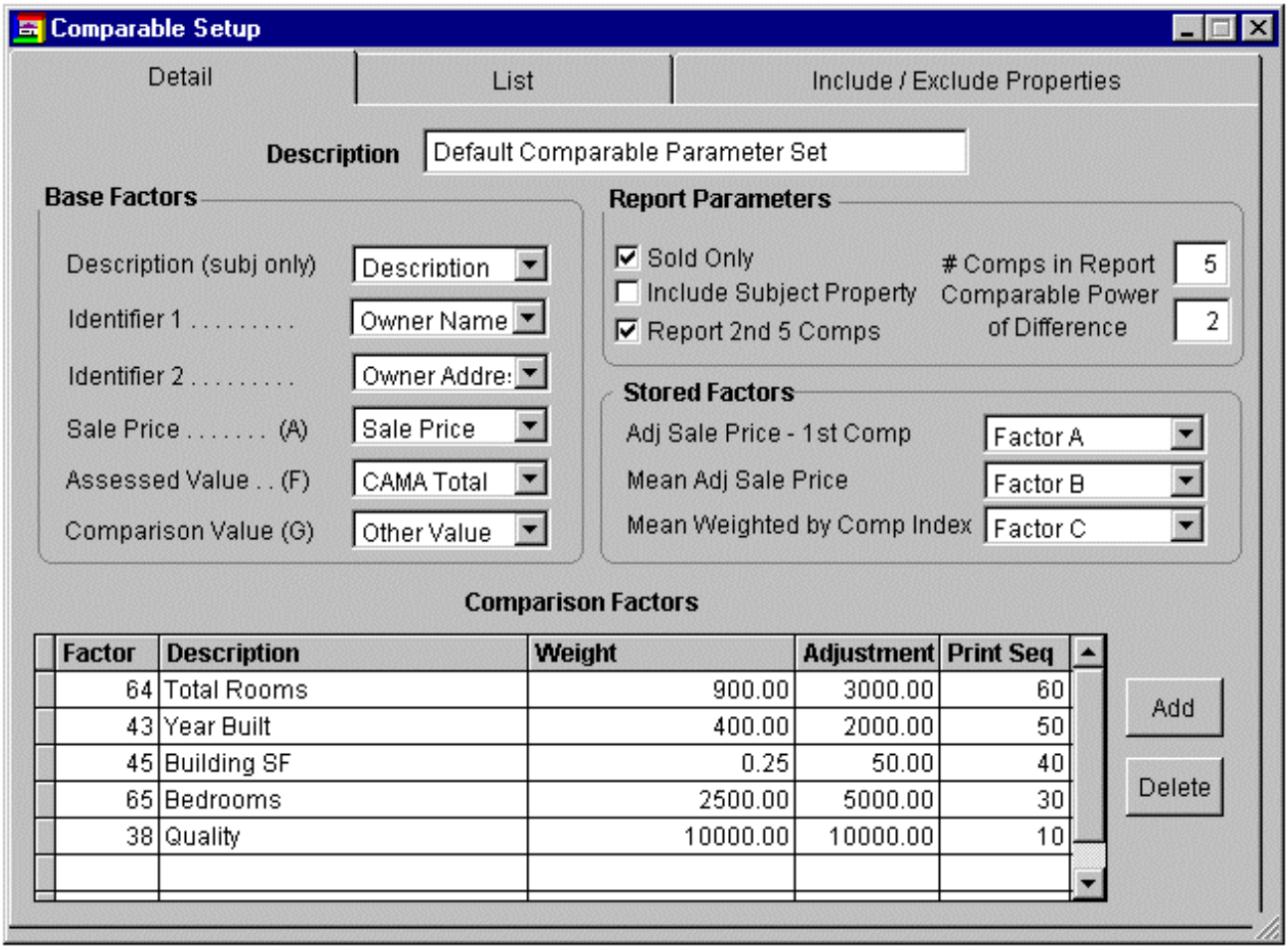
Using this algorithm, a value of Q is computed for each candidate parcel, and the five parcels with the lowest Q scores are considered the most comparable. As we will see, the system can be adjusted to produce fewer than five "most comparable" properties. This again is up to the judgment of the appraiser.

Building a model

Click on **Valuation**, then **Comparable** to run the comparables program. The Comparable Valuation window displays a list of parcels with three buttons at the bottom: **Setup**, **Choose Report**, and **Run Comparable**. To establish the parameters discussed in the previous section (which may be saved for later reuse), click on **Setup**.

There are three tabs in the comparables setup window: **Detail**, **List**, and **Include/Exclude Properties**. The window opens on the **List** tab, which lists the setups that have already been created and saved. If you highlight one before proceeding to **Detail**, you will be able to modify it. If you do not highlight a setup (or highlight a blank line), you will be able to create a new one.

If you do not see a list of setup files, click on the **Detail** tab anyway, then click **Add** to specify the factors the system should use in finding comparables. Highlight the desired factor from the list at left and click the right-pointing arrow to select it for your routine. Repeat for each factor you want to include. Then click **Close** and proceed to input weights and adjustment values as described below.



The name of this parameter set is shown in the bar near the top of the Comparable Setup screen. In the grid at the bottom of the form you will see the factors currently usable in the comparables selection routine. To add factors, click on **Add** and select the factors you wish to include, then click **Close** to return to this form. To delete a factor, highlight it, then click **Delete** and confirm you wish to delete it.

Enter weights and adjustments for each factor. A good way to proceed is to start with the adjustment, and put in a number approximating the dollar value of the characteristic. For "Square feet of living area" you might put in 55, for example, or for "Number of rooms" you might put in 10000. Remember that if a categorical factor (e.g. "Condition") is used, the computer will read the associated linear value. There must be a linear value in each of the factor levels (other than level 0 – "No Data") in order for the selection algorithm to

operate properly with a categorical factor. If the subject property is in good condition (linear value 110) and the candidate property in average condition (linear value 100), the difference will be read as 10 and multiplied by the specified adjustment. An adjustment of 2000 in that case would result in a $20,000 adjustment in selling price.

Once all the adjustments are specified, you can enter the weights, making them proportional to the *square* of the adjustment. Here is a list of factors, along with weights and adjustments:

Factor	Description	Weight	Adjustment	Print seq
64	Total Rooms	900.00	3000.00	50
43	Year Built	400.00	2000.00	30
45	Building SF	0.25	50.00	20
65	Bedrooms	2500.00	5000.00	40
41	Condition	100.00	1000.00	10

Note that in each case the weight is calculated by dividing the adjustment (a dollar approximation) by 100, then squaring the result. It is not necessary to divide by 100, but it is often convenient to divide by some power of 10 to keep the weight at a manageable size. The absolute value of the weight is not important, only the value in relation to other factor weights.

If you wanted to make one of the selection factors (e.g. Building SF) play a greater role than the others in choosing the comparables, you could increase the weight significantly. The result would be to magnify the effect of any differences, based on that factor, between the subject property and any candidate, thus ensuring that those properties in which such differences were minimal would be given priority.

The *Print seq* column allows you to determine the order in which the factors will be presented in the report. It is useful to enter numbers in increments of 10 in case you later decide to add a factor and want it to be displayed between two other factors.

Here are some guidelines for filling in the other parts of the setup screen:

- Base factors are used by the system in the comparables report. *Location* is usually the physical address of the property. *Sale Price* is the factor used to determine whether a parcel has sold and, if so, for how much (thus providing a basis for adjustment). *Assessed Value* and *Estimated Market Value* are used to provide ratios indicating how well the adjusted sale price produced in this report compares with those values.

- If you check the “Sold Only” box, only sold properties (those with a non-zero value in the sale price field) will be considered.

- If you check “Include Subject Property,” the subject will appear as one of the comparables.

- If you check "Report 2nd 5 comps," the records numbers of the second 5 comparable properties will be displayed at the bottom of the report.

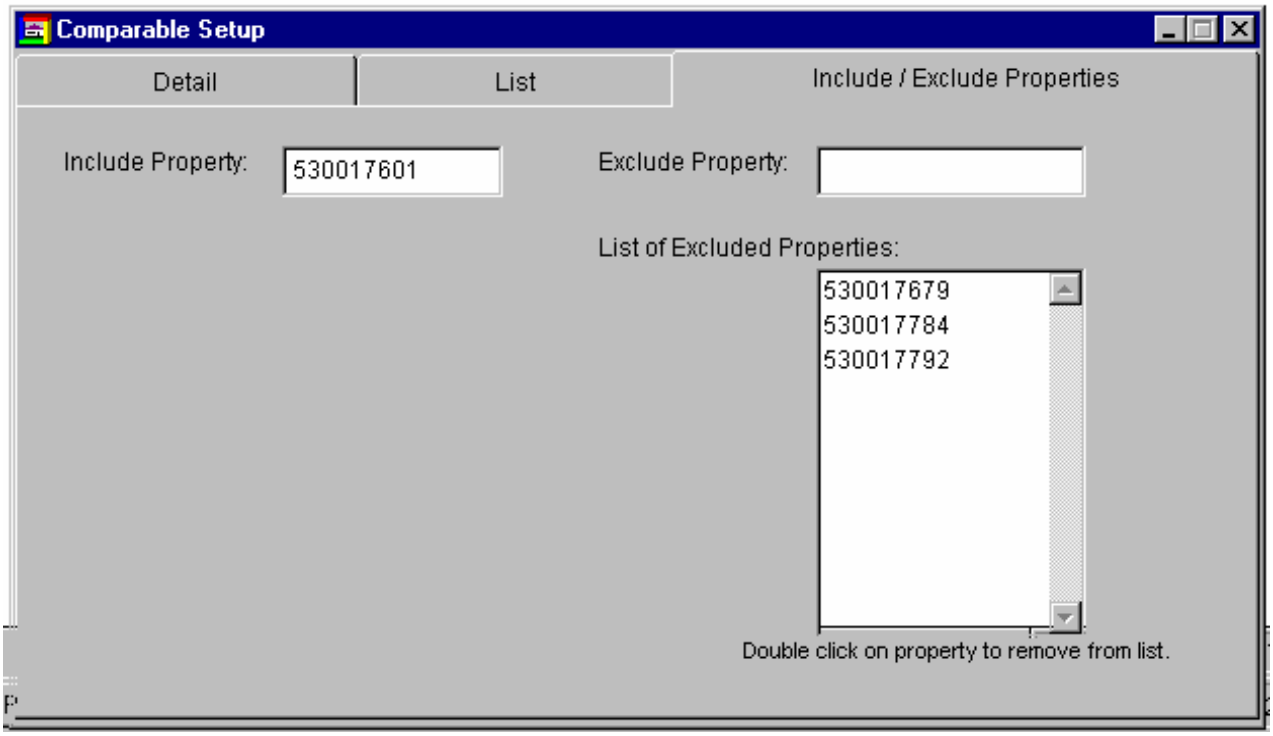
- You can have fewer than 5 properties in your report. In the box next to "# Comps in Report" place the number you wish.

- The difference between the factor values can be raised to a power other than 2, though this is not common. You can specify the power in the box next to "Comparable Power of the Difference."

On the right side of the screen the Stored Factors are listed. The comparables routine produces three values that may be stored in specified factors. These are: (1) the adjusted sale price of the first comparable, (2) the mean adjusted sale price of all the comparables selected, and (3) the mean adjusted sale price weighted by the comparability index. (The comparability index is an indicator of the degree of comparability. The index of the most comparable property is set at 100, and the others are given indices of lesser value depending on the extent of their difference from the most comparable property.) Each of these three parameters can be saved to a factor. Use the "spinners" to select from among the available factors the ones to which these values should be written.

Inclusions and exclusions

If you wish to force a parcel into the report or to exclude certain parcels from consideration, click on the right-hand tab labeled **Include/Exclude Properties**. You will see a screen similar to the following:



In the left-hand data box you can enter the parcel ID of any parcel you wish to force into the comparables report. The system will check to ensure that the parcel actually exists. In the right-hand data box you can enter the parcel IDs of any ten parcels you wish to exclude. They will be recorded in the box below. To remove any parcel from that list, simply double-click on the ID. The included and excluded parcels are treated as part of the overall setup.

When you close the comparable setup window, the parameter setup will be saved under the name you have given it, to be recalled by the system whenever it is designated.

Remaining steps

Click on **Choose Report** to select the report you want to print. Ordinarily you will have only one or two choices – either a detailed comparables report or a set of photos of the subject parcel and the selected comparable parcels. You can choose to print the report or simply to display it on the screen. If you choose the latter option, you will still be able to print the report upon exit from the screen.

Once the report is selected, you can click on **Run Comparable**. The computer will sift through the records in the current database to determine those that are most comparable according to the parameters you have specified, and will produce a report (see page 4-7) displaying them.

If you wish to further limit the universe of candidate parcels by means of a filter, you can make sure a filter is active before running the comparables program. See chapter 3 for instructions on setting up the filter expression.

Running against a separate sales file

If you have a separate file of properties that sold over a certain period or in a particular neighborhood, you can select a property from the current file and search the separate file for comparables. Both files must have identical structures – the same factors in each table, same data types, sizes, etc.

To engage a second data file, click on the bar labeled “Use Comparables from file ...” and then choose the file you want. The program will check to make sure the structure of this file matches that of the file you’re currently signed on to. If not, you will be told you must choose another file. If the structures match, the name of the second file will appear in gray letters on the form. Now, when you click the "Run Comparable” button, the subject parcel will be compared with candidate parcels in the other database, and the report will show comparables from that database.

REVIEW QUESTIONS

1. What does the computer need to know in order to select properties that are comparable to a subject property?

2. How is an adjustment value determined? A weight?

3. How do you tell the system how many comparables to select?

4. Is it possible to force a parcel into the comparables display even if it is not technically selected?

5. What are the values produced by the comparables routine and how are they stored?

Property File: F:\MSOL\DATA\MONTPEL\MAIN.DBF **Parcel ID** 018007000 .31 ACRES & DWL/APT

632 Sold parcels were considered under Comparison Set # 1. (The subject record was excluded from consideration.)

5 weighted factors were used in the comparables selection. (The difference was raised to a power of 2.)

	Subj. Record	Most Comparable		Next Most Comp.		3rd Most Comp.		4th Most Comp.		5th Most Comp.			
Parcel I.D.:	018007000	005083000		155108000		155039000		018009000		018006000			
Owner Name	RITZ SUSAN Z	PEARL ROBERT E		GOLDMAN ALAN B		MAGUIRE GREGORY M		TOMASI ERNEST P		JACOBSON DAVID S			
Owner Addres	7 CLARENDON	410 HUFF AVENUE		PO BOX 158		39 TERRACE ST		9 CLARENDON AVE		6 CLARENDON AVE			
# Factor	Fac. Val.	Fac. Val.	Adjust	Fac. Val.	Adjust	Fac. Val.	Adjust	Fac. Val.	Adjust	Fac. Val.	Adjust	Comp.Wt.	Adj. Rate
38 Quality	3.00	3.00	0	3.50	-5000	3.00	0	3.50	-5000	5.00	-20000	10000	10000
65 Bedrooms	5	6	-5000	4	5000	5	0	3	10000	4	5000	2500	5000
64 Total Rooms	13	14	-3000	12	3000	11	6000	8	15000	9	12000	900	3000
43 Year Built	1907	1900	700	1900	700	1900	700	1920	-1300	1931	-2400	400	100
45 Building SF	3420	3800	-19000	2812	30400	2524	44800	2632	39400	3720	-15000	0	50
A. Sale Price:	$224,250	$150,000		$522,000		$120,000		$180,000		$-20,400			
B. Total Adjustments Above:		$-26,300		$34,100		$51,500		$58,100		$202,500		**Mean**	**Std.Dev.**
C. Adjusted Sale Price:		$123,700		$556,100		$171,500		$238,100		$182,100		$254,300	173,544
D. Comparability Index:		100		88		66		59		49			
E. Mean Weighted by Comparability Index.....:			$263,971					**COV for (C.) =**		68 %			
F. CAMA Total:	$173,300	$141,800		$222,700		$117,100		$120,800		$193,900			
G. CAMA Total:	$173,300	$141,800		$222,700		$117,100		$120,800		$193,900			
H. Ratio of F/C:		1.1463		0.4005		0.6828		0.5073		1.0648		0.7603	0.3321
J. Std. Deviation of (H.) About 1.0.....:			0.4267										

The 6th thru 10th comps. are respectively Parcel IDs 063010000, 155037000, 043166000, 155045000, 005101000

The relative comparability of 1-5 to 6-10 is 0.49

CAMA 2000

5 ■ THE COST APPROACH

There are many different kinds of cost approach. Some have become widespread among appraisers and assessors; others are used in particular localities and are custom designed. MicroSolve's CAMA 2000 accommodates any kind of cost approach you might wish to employ. In this chapter you will learn:

- To examine and modify fields and values in the cost tables
- To run the cost approach as a separate procedure or as part of data modification
- To analyze the Usit procedures that govern the cost approach and the report template that determines its format

The cost approach included with your CAMA 2000 program will depend on the configuration you specified when you ordered the program. By default, MicroSolve installs the Marshall & Swift residential and commercial cost routines, since they are the most widely used of the cost approach programs. MicroSolve also supplies the companion manuals, which you can consult for detailed information on the relation between specific factors or property characteristics and the resulting property value.

However, cost approaches based on completely different principles and different rate tables are possible. This radical flexibility is built into the design of CAMA 2000. What makes it possible are four major elements:

- A user-configurable database, based on a data dictionary you can adjust.
- A set of cost tables that can be designed for particular purposes, with as many pages, rows, and columns as desired, and with each labeled to correspond with certain factor values.
- A report template, created by the FoxPro report writer, which lays out the format of the cost report worksheet.
- The User's Simplified Instruction Table (Usit), a programming environment designed to work with the tables and the factors in the CAMA 2000 database. The cost routine developed in the Usit environment is what produces the values and displays the cost worksheet based on the report template.

The Marshall & Swift commercial cost routine is an exception, in that it is not based on a Usit routine or on embedded cost tables that you can examine and change. Instead it

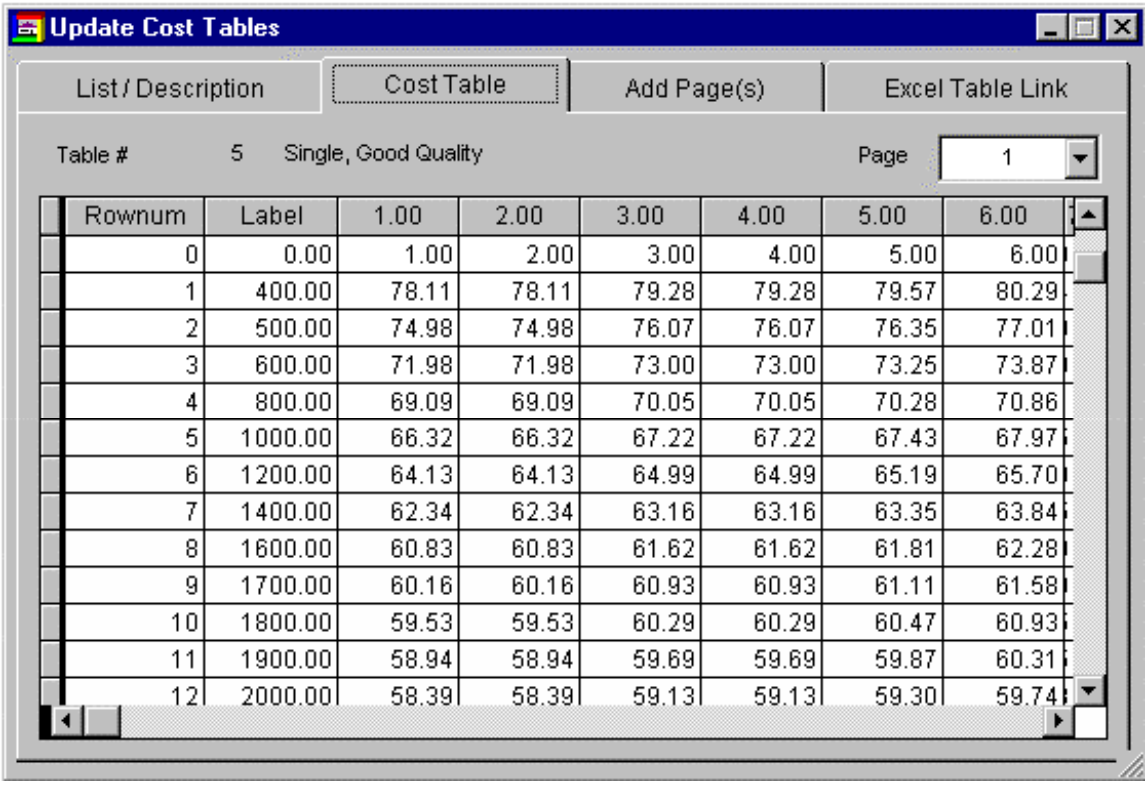
makes use of the Marshall and Swift Commercial Estimator software, which is installed at the time CAMA 2000 is installed and treated as a "black box" by the MicroSolve program. When you operate the commercial cost routine against a record in the database, the system sends information about that record to the Commercial Estimator and receives back the estimated value, which is then presented, along with supporting data, in a cost report. To change the base year of the commercial cost values, it is necessary to install a new version of the Commercial Estimator, based on a different year. These elements are discussed in more detail below.

The user-configurable database

A cost approach will work only if the data it needs have been collected and stored in a database. For that reason, every CAMA 2000 system design must start with a consideration of the cost approach that will be used with it. The relational structure of the system gives it the flexibility to work with virtually any cost approach, and users frequently add factors not needed by the cost approach but useful for administrative or other purposes. If you are using the Marshall & Swift cost approach, consult the specialized cost manual supplement to determine which factors in the database are needed to produce values.

The cost tables

To view or modify the cost tables, click on **Valuation** in the menu bar, then on **Cost Approach** and **View/Modify Cost Tables**. Tables are listed by both name and number. When you bring up a particular table you will see the definitions of its pages, rows, and columns in the **List/Description** tab. (You can add or modify these definitions yourself.) Click on **Cost Table** to see the actual data. Select a page from the spinner and then view the rows and columns comprising that page.



Rownum	Label	1.00	2.00	3.00	4.00	5.00	6.00
0	0.00	1.00	2.00	3.00	4.00	5.00	6.00
1	400.00	78.11	78.11	79.28	79.28	79.57	80.29
2	500.00	74.98	74.98	76.07	76.07	76.35	77.01
3	600.00	71.98	71.98	73.00	73.00	73.25	73.87
4	800.00	69.09	69.09	70.05	70.05	70.28	70.86
5	1000.00	66.32	66.32	67.22	67.22	67.43	67.97
6	1200.00	64.13	64.13	64.99	64.99	65.19	65.70
7	1400.00	62.34	62.34	63.16	63.16	63.35	63.84
8	1600.00	60.83	60.83	61.62	61.62	61.81	62.28
9	1700.00	60.16	60.16	60.93	60.93	61.11	61.58
10	1800.00	59.53	59.53	60.29	60.29	60.47	60.93
11	1900.00	58.94	58.94	59.69	59.69	59.87	60.31
12	2000.00	58.39	58.39	59.13	59.13	59.30	59.74

Note that each row has a numeric label, as does each column. When you run the cost approach, the computer matches the page number, the row label, and the column label with factor values from the record being costed to locate the cell whose value will be applied in the cost routine. With an appropriate security level you can change values in the cells of the table to suit particular needs. Remember that in any table, *each page must have the same number of rows and columns*, or the Usit program will not work properly. Further, all row labels and all column labels must be in ascending order, and the labels must be identical from page to page. You can verify that this is the case by running "Verify Table Structure" from the List/Description tab of the tables form.

Tables ordinarily have a maximum of 27 columns, although it is possible to configure your system to accept more. Individual tables may have only one column, or just a few, but to add more you need only specify a label in row 0 and then place the values below it. To add a row to a table, click the + button in the row of buttons near the top of the screen. This creates a blank line at the bottom of the table page, which you can then fill in with a row label and values. *If you add a row or column to one page, you must add it to all pages in the table.* To add one or more pages, choose the third tab of the tables form and specify the number of pages to add. Ordinarily the new page(s) will be blank (except for column and row labels), but if you wish you can specify that the information in the last active page is to be copied into the new page.

Excel link

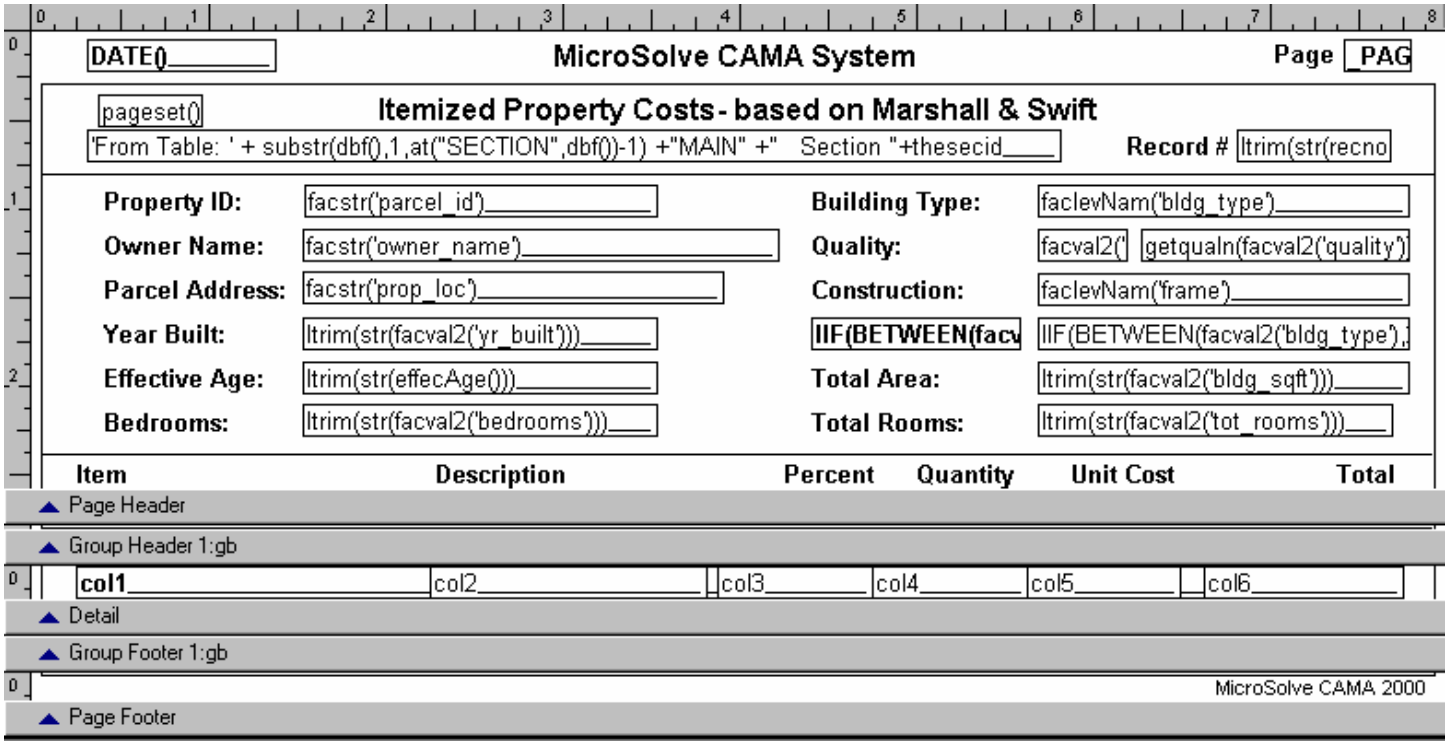
More complex table manipulation, such as copying portions of a table, value interpolation, or extension of series, is more easily done in a spreadsheet environment. MicroSolve provides a link between CAMA 2000 and Excel, allowing you to export any table (pages, rows, and columns) into Excel, manipulate it there, and then re-import it back to CAMA 2000. To do this, bring up the table you wish to process, then go to the fourth tab and click "Start Excel." Doing so will force an automatic table verification, and you will be notified of any discrepancies among row or column lengths or labels. Once the table is in Excel, you can make any changes required. You can even substitute a completely different table, as long as it is in the same general format and obeys of the "rules" of MicroSolve tables. When your table is ready to be imported back into CAMA 2000, click the CAMA 2000 icon on the status bar at the bottom of the screen, then click the button labeled "Import Table Changes." Again, the table structure will be checked, and if it is acceptable the table will be re-imported.

Cost tables are stored on the disk in the same folder that contains your database. The files are COSTABLS.DBF and COSTABLS.CDX. While they can be copied to other folders, they may not work with a database of a different type, and even if the new database is identical in structure to the one they were created for, a new link must be established.

The report template

The FoxPro report writer allows you to produce reports based on the database fields. You can view it by clicking on **Tools** in the main menu, then on **Create Report**. Look for a

file called MSCOST.FRX and open it. (This table may be in the local directory or the Msol directory. The cost program looks in the local directory first.) It consists of several configurable sections: a page header, a group header, a detail section, a group footer, and a page footer. The group header and footer are used when subgroups of data are presented together, but they are not used in the standard cost report. The page header is used to present information that is common on all reports: the main heading, the jurisdiction name, and the parcel and owner identifying information. It also presents the column labels, under which parcel-specific data will be listed. The detail section indicates the position and width of each column in which data will be displayed. It and the header section also contain a rule (or box outline) that defines the edge of the report. The footer section contains only the bottom edge of this rule. This template can be changed for other types of cost report, but it should rarely need to be modified once it has been established.



The User's Simplified Instruction Table (Usit)

Usit (reached through the Usit Editor under **Tools** in the main menu) is a programming environment allowing access to a specialized subset of the FoxPro programming language, augmented by certain additional commands that perform such functions as display of categorical levels, table lookup, and data storage. Chapter 6 provides detailed information about the Usit table and the associated commands and functions. Except for the commercial Marshall & Swift cost approach, all cost approach routines are written in the Usit environment. Only people who have received special training in Usit should attempt to change or add to the table. Since it governs all value lookups, cost calculations, and

report displays, any change in a Usit table can alter the way it calculates or displays values.

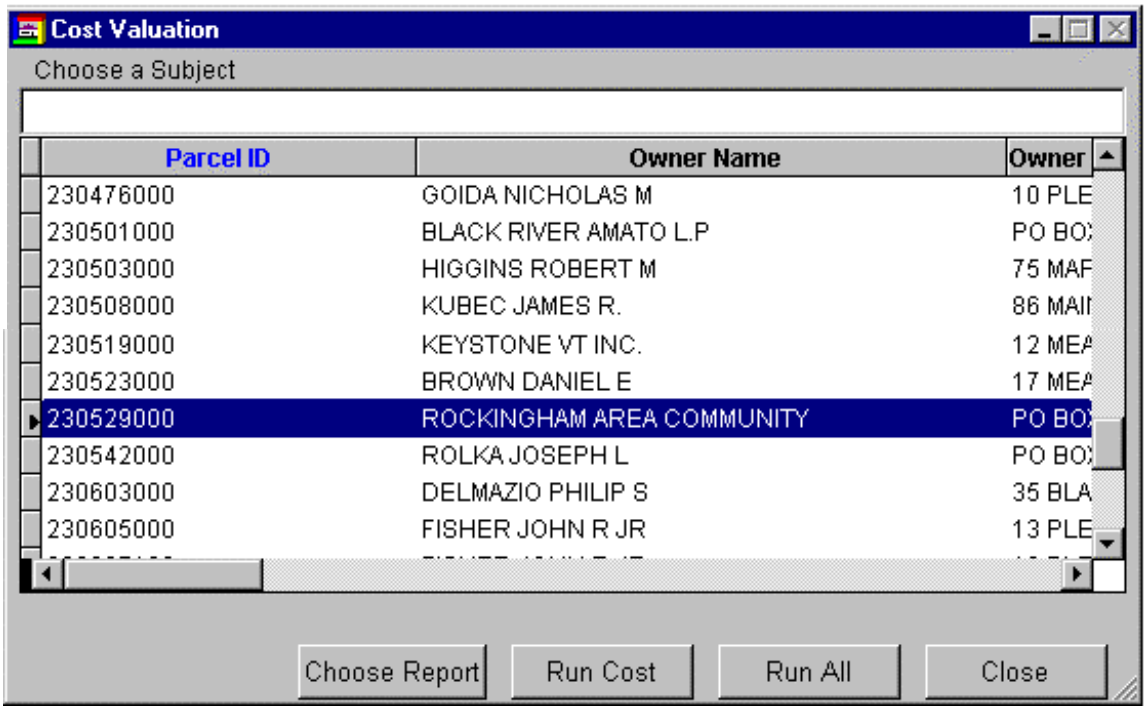
Three files comprise the Usit table: COSTEQU.DBF, COSTEQU.CDX, and COSTEQU.FTP. As with the cost tables, they can be copied to other folders, but they will not work with a database of a different type, and even if the new database is identical in structure to the one they were created for, a new link must be established.

Running the cost approach

There are two ways to run the cost approach:

1. You can click on the **Cost** button in the lower left-hand corner of the data entry form. If a report format has been previously chosen, the report will be displayed, and then the data input form will return to the screen, with any new totals now incorporated.

2. You can click on Valuation in the main menu, then choose Cost Approach and Run Cost Approach. The form that appears on the screen lets you choose a report; choose whether to print or just display (preview) the report or merely to calculate without printing; run the cost on an individual record (selectable from the browse table built into the form); or run costs on all the records in the database. If the latter option is desired – for mass appraisal in a production environment – you may wish to invoke a filter before clicking on **Run All**. The filter button near the top of the screen gives you the same options you saw in connection with the Browse data display mode. In this way it would be possible to run costs on all the properties in Neighborhood 8, or all mobile homes, or all parcels not reappraised since 1995.



Cost error reporting

If errors were produced during the cost routine, you will be given a chance to examine a list of them after the cost report has been displayed and closed. Errors can arise for a variety of reasons. Here are the principal ones:

- A needed value was not specified in the database. For example, a neighborhood number may not have been given, and this would make it impossible for the program to calculate a land value.

- A property feature may have a value that is out of the range allowed for in a table. For example, a porch may have an area of 3000 square feet entered, although the maximum value allowed for in the cost table is 300 square feet. These errors should occur infrequently because Usit routines usually include instructions to the computer saying, in effect, if the value of the porch area is greater than 300 square feet, use 300 as the lookup value for purposes of finding the cost rate. If such an error does occur and the data in the record proves to have been correct, bring the error to the attention of your in-house Usit expert or a member of the MicroSolve support staff so the program can be adjusted to prevent future occurrences.

- There is a flaw or bug in the Usit routine. Sometimes this results from a failure in the program logic, sometimes from missing or corrupted lines, occasionally from missing files that the program needs to work properly. In all such cases, you should consult qualified support staff who will have the resources to fix the problem.

REVIEW QUESTIONS

1. What components are involved in producing a cost report?

2. What options are available for costing a property?

3. Where are the cost tables located and how do you view and/or modify them?

4. What is Usit and how is it used in the costing process?

5. If a property feature does not appear to be costing properly, what are possible explanations and how would you check them out?

Itemized Property Costs
Town of Haagen Daz

Property ID:	020111000	**Building Type:**	Single
Owner Name:	SALERNO RICHARD	**Quality:**	3.25 AVG/GOOD
Prop. Location:	7 NORMAN DRIVE	**Construction:**	Studded
Year Built:	1970	**Style:**	1.5 Fin
Effective Age:	15	**Total Area:**	2241
Bedrooms:	5	**Total Rooms:**	12

Item	Description	Percent	Quantity	Unit Cost	Total
BASE COST					
Exterior Wall #1:	WdSidng	100.00		42.92	
ADJUSTMENTS					
Roof #1:	Metal-Chn	100.00		-0.83	
Subfloor	Wood				
Floor cover #1:	Allowance	100.00		2.31	
Heat/cooling #1:	HW BB	100.00		1.27	
Energy Adjustment	Good			1.10	
ADJUSTED BASE COST			2,241.00	46.77	104,811
ADDITIONAL FEATURES					
Fixtures (beyond allowance of 8)			3.00	850.00	2,550
Roughins (beyond allowance of 1)				296.25	
Porch #1:	WoodDck/None/Cov		74.00	23.46	1,736
Porch #2:	WoodDck/WdRail		814.00	7.19	5,853
Basement	Conc 8"		1,264.00	11.69	14,776
Finished Basement	Partition		1,632.00	16.22	26,471
Subtotal					156,196
Local multiplier		1.09			
Current multiplier		1.01			
REPLACEMENT COST NEW					171,956
Condition	Avg/Good	Percent			
Physical depreciation		5.00			-8,598
Functional depreciation					
Economic depreciation					
REPLACEMENT COST NEW LESS DEPRECIATION					163,358
LAND PRICES	Size	Nbhd Mult	Grade	Depth	
FR Bldg Lot	90.00	1.00	0.90	130.00	127,800
SITE IMPROVEMENTS	Quantity	Quality			
Water	Typical	Average			4,000
TOTAL PROPERTY VALUE					295,200

NOTES

Possible new heating plant. Still needs interior inspection.

CAMA 2000

6 ■ USER'S SIMPLIFIED INSTRUCTION TABLE

CAMA 2000 puts at your disposal a simple table of commands and functions in which a routine can be created and modified. This table or programming environment is known as the User's Simplified Instruction Table, or **Usit**. A Usit routine can have a variety of functions, but the most important one is to perform a cost approach according to your specifications. The cost approach can involve value table lookups, calculations, and production of a customized report.

Not every user will, or should, have a need to modify or write a Usit routine. For those who do, this chapter is intended to supplement special training and provide a reference. Others can skip the chapter altogether.

Information placed in the Usit table is stored in a file called COSTEQU.DBF (with associated tables COSTEQU.CDX and COSTEQU.FPT). The Usit table is a complete guide to the computer for performing your instructions. The program interprets this table and creates another file called COSTDETL which is the detail table for the report. The results are entered into the COSTDETL table and the report writer is invoked to simply print from the detail table. The report writer must therefore have a template called MSCOST.FRX containing a header and up to six field positions that will define where data can be displayed and/or printed.

To run properly, the system also requires a factor equivalence table called FACEQUIV (extensions CDX and DBF), which uses names to identify each factor. All references in the Usit table must be to names in FACEQUIV. This ensures that all references are to unique factors (since each name in this table is unique) and that a cost program or other routine will function the same way even with different databases, provided the data elements referred to by the names in FACEQUIV perform equivalent functions.

The Usit table is accessed by clicking on **Tools** in the main toolbar, then on **Usit Editor**. The table is saved automatically when you close it.

Format of the table

Below is an example of a portion of a Usit table

Order	Name	Equation	Result	Print
510	cl	facval("class")	3.00	0
520	area	facval("lan_area")	8.40	0
530	row	1150	1150.00	0
540	row	iif(cl=3,1160,row)	1160.00	0
550	row	iif(cl=4,1170,row)	1160.00	0
560	row	iif(cl=5,1180,row)	1160.00	0

The “Order” field is maintained automatically by the system. Each time a line is added (via the + sign in the toolbar at the top of the screen), a new line number is created. If lines are inserted, they are numbered automatically. If space is needed between lines, click on the “Renumber” button and all lines (and line references) are automatically re-numbered at intervals of 10.

In the “Name” column one ordinarily places variables which are calculated or produced by the expression in the “Equation” column. However, four special symbols may also be placed here: #, &, “, and @. These are described under **Special characters** below.

The “Equation” column contains expressions the program can evaluate to produce a value that is stored in the variable listed in the “Name” column. An expression may be any FoxPro function such as *iif()* or it may be a specific function developed for Usit, such as a table lookup function. More detail on these functions is given below.

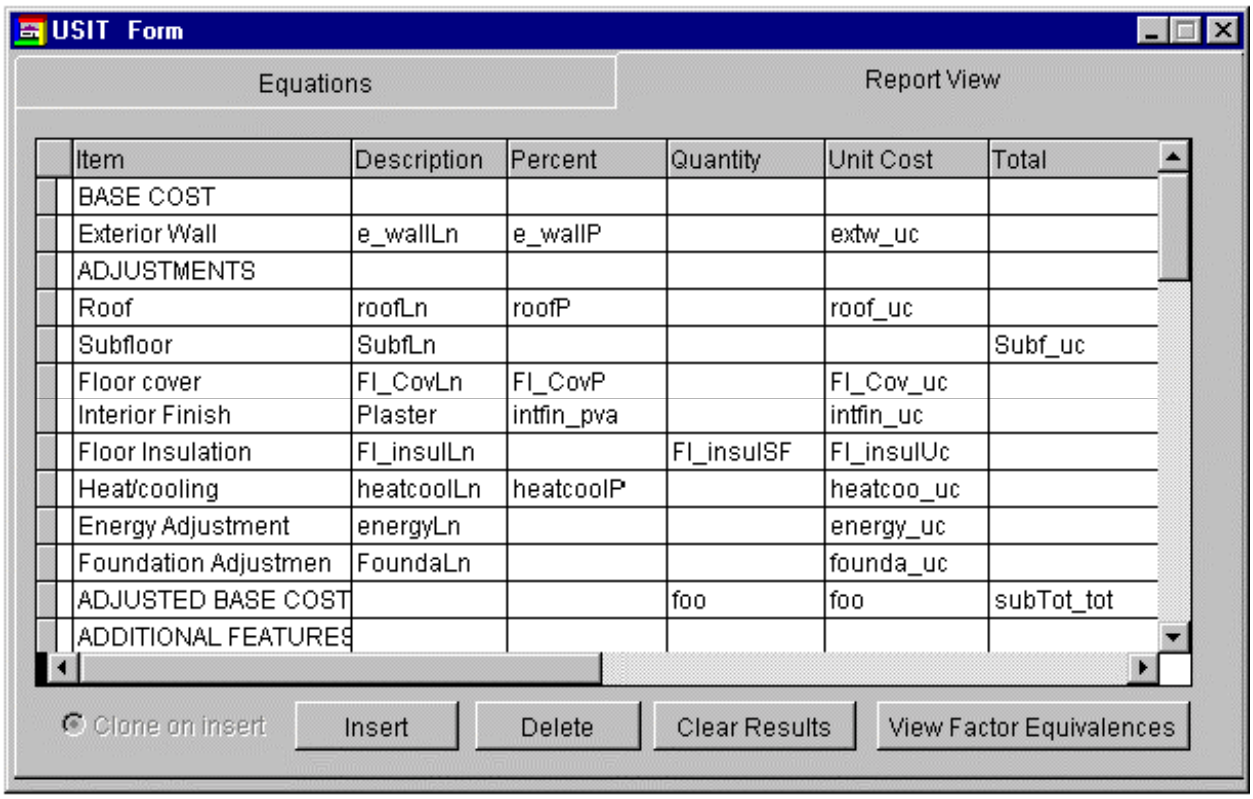
You ordinarily do not enter anything in the “Result” column. This column is used by the system to store results from the operation specified in the “Equation” column. That is, this column often contains the value of the variable identified in the “Name” field. As such, it is very useful for debugging or simply tracing the operations of the program one is writing. The one occasion when you may enter a value in “Result” is when a quotation mark has been placed in “Name”; then the value in “Result” is the text line that is to be printed at the position indicated in “Print.”

The “Print” column may contain a value from 1 to 6. These numbers may be thought of as tab positions indicating columns in which text or values will be displayed. The actual tab positions on the page can be altered by editing the report MSCOST.FRX using the **Create Report** option under **Tools** on the main toolbar. Tabs are usually designed so that material placed in them can extend to the right-hand margin of the page if necessary. A value (other than 0) in the “Print” column should be specified only if something is to be actually displayed by the program at this point. In general, information is displayed as it is generated by calculations or table lookups. Reports are laid out accordingly.

The report view tab

The right-hand tab gives access to a “report view” which is a schematic table of report positions and entries for those Usit routines that print or display their results. This report view is a handy first view of the report that will emerge when Usit is run. It is also a quick way of finding your way around in the Usit code. If you click on any part of it, then click the **Equations** tab, you will be located in the portion of the code that produces the information displayed at that point in the report.

Near the bottom of this page there is a bar labeled **View Factor Equivalences**. Clicking on it will display the factor equivalence table, which is discussed in chapter 9 on the data dictionary. This allows you to check the factor equivalence name of particular factors you might want to use in the Usit routine.



The adjacent bar, **Clear Results**, can be used to delete from the Results column all entries placed there by the previous Usit run. Since results are not automatically deleted at the start of a Usit run, using this bar is the only way to ensure that all the entries in the column after the next run are the result of that run alone and not residues of earlier runs. The feature will not delete any entry in the Results column that accompanies a quotation mark in the Name column and therefore represents a phrase to be printed on the report.

It is inadvisable to use the **Insert** or **Delete** bars on this page. More precise control is achieved by modifying the Usit program from the Equations table.

Modifying the table

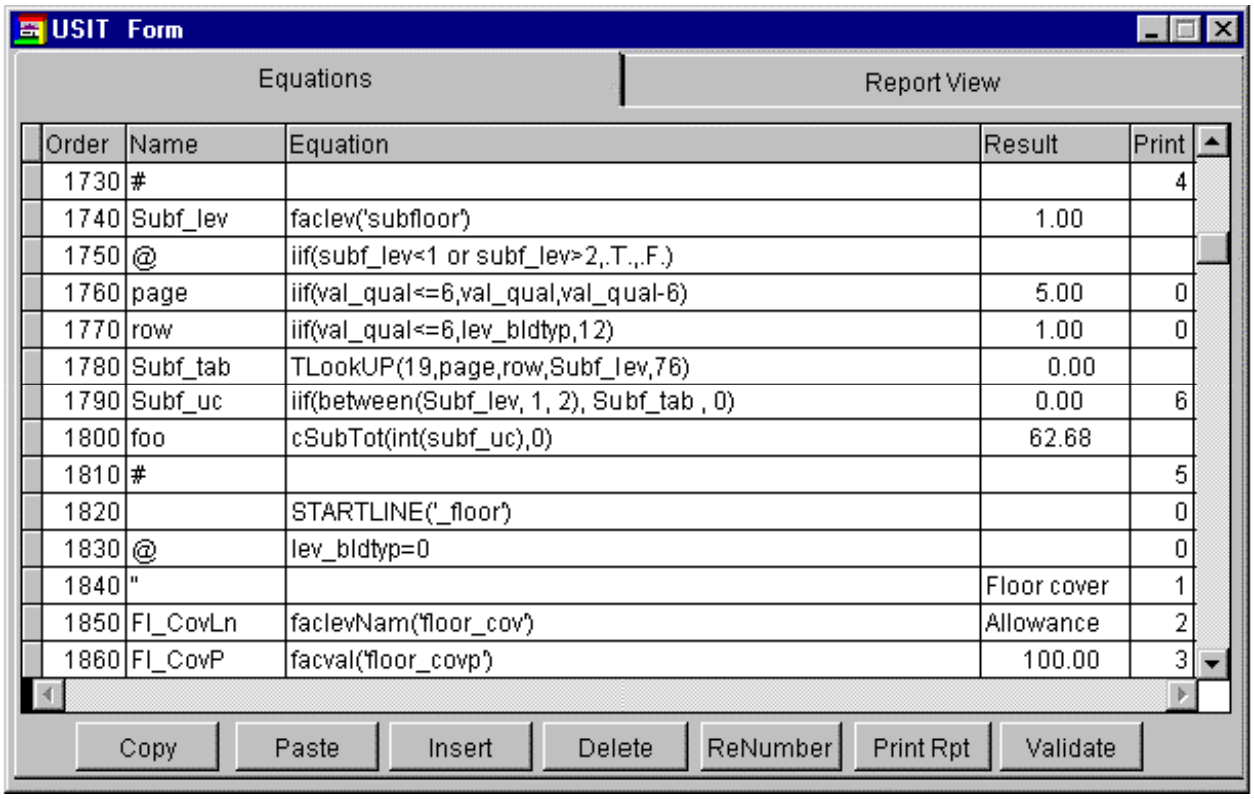
The remainder of this chapter pertains to the Equations table (i.e. the left-hand tab page of the Usit form).

To add a line to the table, click the + symbol in the toolbar. This places a line at the end, and you may then enter data in its cells.

To insert a line, click on the line *above which* you want the new line to go, then click the **Insert** button at the bottom of the form. This opens up a line. If the line below the insertion was line 50, the inserted line will be numbered 49. You can insert up to 9 lines at once at the same point before having to renumber.

To delete a line, click on it and then click on the **Delete** button at the bottom of the form. You will be asked to confirm your decision.

The material in a line may be copied to another line and then modified, thus speeding the process of writing code. Highlight the line you wish to copy by clicking anywhere on it, then click on **Copy** at the bottom of the form. Now move to the line you wish to copy the material *to*, and click on **Paste**. A copy of the material on the first line is written to the second.



Program flow

The Usit routine runs once each time a new record is processed. Variables are automatically initialized between records. This means that any variables intended to accumulate values from previous parcels or sections of parcels should be stored in a field at the end of a run and then read from that field at the start of the next.

The Startline() command in the “Equation” column (with nothing in the “Name” column) informs the system that a new print or display line is starting. It will cause the program to display any following printed material one line below the previous printed line. Startline() also indicates a new group of information to be processed. Any **&** function placed in the “Name” column must direct program control to a line *before* the next Startline().

If the routine is to process a series of entries in a subtable of the relational database (porches or attached garages, for example), you should precede the lines that do this processing with the function Startline(_porch) [for example], where *porch* is the name of the subtable or child table as identified by "Table ID" in *facequiv*. Be sure to use the underscore as shown. The lines should be followed by a new Startline(). In processing, the lines of code thus enclosed will repeat for every entry in the subtable before the program proceeds to the next section.

An example may help. Suppose we have a need to sum all the area values in a land subtable and use that information in the calculation of rates for each line (land type). To do this, we would use the Startline(_land) function before reading the value of a land area, then another Startline() to indicate that the system should loop back and read the next area before proceeding. We would keep a running total of land area in a variable. When we are ready to calculate individual rates and print them, we again insert Startline(_land). The result is that we execute one loop to read each land record and then execute a second loop to calculate and print each one. We did not print the first set of land records, but did print the second. (We could have printed the first set as well.) The sequence is:

	Startline(_land)
Total_area	Total area+Land_area
	Startline()
[Now Total_area contains the sum of all land area records for this parcel]	
	Startline(_land)
Land_Child_Value	calculated from land type and total area
	Startline()
[If there were three land types, three lines would be calculated and printed]	

If we have several pieces of information about each child record, they need to be all displayed in the same line. Moving to a second line is possible only if we also move to a second child record.

It is not necessary to place any filter or index information in the Usit table. The program will proceed according to any index that has been specified at runtime, and will select records according to any filter currently in effect.

Special characters in the name column

Variables in the "Name" column can be deternined by you, but they must begin with an alpha character. A variable stores the result of the equation or table lookup value calculated in the "Equation" column. There are, however, some special characters that can be used to achieve specific purposes.

" If a double quote is used in the first position of the "Name" column, then the program will display or print a text value specified by the programmer. The text value to

be displayed should be placed in the “Result” column and the tab position in the “Print” column.

[#] If a pound sign is used in the first position, then this line is merely a placeholder. In other words, nothing happens. This symbol is mainly used to introduce a programmer comment, with the comment placed in the “Equation” column and the “Print” column set to 0.

[@] The “at” sign instructs the system to skip to the next startline() instruction if the function in “Equation” is true. Even if the program has already instructed the printer to print information at one or more tab positions in a line, when this symbol is encountered (and a “true” condition obtains), that instruction will be ignored and the program will proceed to the next startline(). For example, if there is no garage on a property we would want to instruct the program to skip that portion of the routine dealing with garage costs.

[&] The ampersand instructs the computer to skip to a different section of code. It causes an effective GOTO in the program, but only within the portion of code preceding the next Startline() instruction. The format is &99999 where ‘&’ is the GOTO indicator and ‘99999’ is the line number to which control is transferred. If the result of the expression in “Equation” is true, then control is transferred to the specified line number.

[&&] A double ampersand in the midst of a line indicates that what follows is comment. This feature allows you to comment on a code line within the line itself.

[blank] The word ‘Blank’ in the name field instructs the program to skip a line (i.e. leave one line blank).

Built-in functions

In addition to the standard FoxPro expressions, certain special functions are understood by the program. Note that the case is not significant; upper case is used here and there simply for clarity.

cSubTot() This function subtotals values passed to it. The first parameter adds the amount to the unit cost subtotal; the second parameter adds the amount to the total subtotal column. The variable ‘SubTot_uc’ contains the running total of unit cost, and the variable ‘SubTot_Tot’ contains the running total.

Callerror() This routine invokes an error message in the log file and aborts the processing of the entire parcel. It is used when certain basic information is not available and the parcel therefore cannot be costed. Example: callerror('Illegal bldg type or quality')

effecage() Determines the effective age of the current property. It does not require parameters. It does require that the names *yr_built* and *eff_age* be defined in the factor equivalence table.

facLevNam() Returns the level name for the specified categorical factor.

facLinVal() Returns the appropriate linear value for the specified categorical factor.

facval() Gets the value of the specified factor. The parameter used in this function should be one of the fields in the factor equivalence table. For example, facval("quality") yields the value of the factor for the current record whose entry in the equivalence table is *quality*. Facval() will return the correct type based on the field type; that is, it will return a value from a numeric field or a text string from an alpha field. From a categorical factor it returns a level number. A date field can be converted to a text string using the *transform()* function, and portions of it can then be converted to numeric form using the *val()* function.

round() Allows you to set the rounding for a variable. The format is round(*variable,n*), where *n* is the number of decimal places. If *n* = 2, the variable will be stored to an accuracy of two decimal places; if *n* = 0 it will be stored as an integer (rounded to the nearest 1); if *n* = -1 it will be rounded to the nearest 10, and so on.

SecIsOne() This routine returns true if the record in the secondary level table (*Section* in the standard database) is 1. It is used for determining if the program should print or process material not associated with subsequent sections (e.g. land/outbuildings in the standard database).

When a cost is run, each section will print a page in the report. Typically, the level 2 tables (such as land etc.) would be printed on the first page of the report. For the majority of cases where there is only one section, this provides a nice one-page report. Subsequent sections would then be printed on subsequent pages, but the level 2 tables would not appear on that page. The function SecIsOne() is used to determine if the program is on the first section or subsequent sections. The level 2 table information is printed only on the first section.

SecIsLast() This is a parallel function to SecIsOne(). It determines whether the current section is the last one. If so, certain summary information may be printed or displayed.

SetFormat() This function formats numeric values using the mask supplied by the programmer. The function should be placed just before a value to be printed. For example: SetFormat('999,999.99') If this function is not

used, the default formats will take precedence. Refer to the section below on format options.

Startline() This function is required to start a new line in the report. It also indicates that a new set of variables is being processed.

Startline('_roof') This form of the above function indicates that the current report line deals with child records. The parameter is the cursor name indicated in facequiv for that child. The Usit program will repeat a set of lines enclosed by startline() for every record in the child table, printing information on each. For instance, if there are three child roof records, there will be three lines pertaining to roofs in the report.

StartPg() This function causes a horizontal line to be printed on the cost report, indicating that the information following relates to a different set of variables. It should be placed directly after a Startline() instruction.

StrMain() Stores a value in the Main table of the relational database. Format: StrMain("[factor name]",*variable*).

StrTbl() Stores a value in any subtable of the relational database below Main. Format: StrTbl("[factor name]",*variable*,"[table name]").

TLookUp() This function does a lookup in the cost table and returns the contents of the specified cell. The function requires five parameters: *table number, page number, row number, column number, mode*. The *mode* parameter indicates whether the system will interpolate if one or more of the cell-definition parameters does not exactly match a page, row, or column label, or will settle for the lower label (known as truncating). At the moment, parameter 76 in this position indicates interpolation, whereas 77 indicates truncation. These numbers will probably be changed to 0 and 1 respectively. Each of the five parameters can of course be passed as a variable.

If you study existing Usit programs, you will notice that the global variables used throughout the cost equations are defined at the beginning of the program and set to 0. This is a recommended procedure. Fields that are printed in the header of the cost program must be modified not in the Usit table but in the report writer.

Format options

Each column of the report has a default format value for numeric fields. The standard format can, however, be overridden by use of the SetFormat() function. If this function is not used, the following formats are the default:

Columns 1 and 2 are formatted for text (character) fields.

Columns 3, 4 and 5 are formatted for numeric fields and default to 999,999.99 (i.e., they display two decimal places).

Column 6 is formatted as a numeric field and defaults to 999,999,999 (i.e. it is a larger field with no decimal places). You can control rounding in the Usit program, and ordinarily numbers in the last column of the report, where totals are printed, are rounded to the nearest 10 or 100 dollars.

Renumber

As the programmer develops a new routine, the Order numbers may become too close to permit insertion of a line. The **Renumber** feature will renumber all lines in increments of 10.

Print Report

The **Print Rpt** button can be clicked at any time to produce a printed copy of the Usit routine.

Validate

The **Validate** button performs several useful functions: it checks for valid line referencing, it ensures that print instructions are in the proper order, and it ensures that portions of the code dealing with child tables are properly identified. It is important to run Validate after making any substantial revisions in the Usit table, especially after inserting or removing lines between two startline() instructions per4taining to a child table. When you do, it may tell you that some print flags are out of place. It is safe to ignore this warning if you did not intend to print anything in a particular column position. It may tell you that a GOTO instruction (beginning with the & symbol) is incorrect because a referenced line does not exist or is outside the group of lines contained within startline() instructions. This is a problem you will have to rectify by hand. And it may tell you that a child table is improperly referenced; this problem it will fix automatically if you click **OK**. Sometimes, when running a Usit program involving a child table, you may find that it appears to process the last line in the table but not the others. This problem can usually be rectified by running Validate.

REVIEW QUESTIONS

1. How does CAMA 2000 use a Usit routine to value a property?

2. What is the role of the factor equivalence table in the Usit routine?

3. What is the role of the report writer in the Usit routine?
4. How would you get the program to print a variable in the fourth column of the report?
5. How can you navigate in Usit by means of the Report View screen?
6. How do you add, delete, and modify lines in Usit?
7. How do you ensure that Usit will process all porches entered for a record?
8. What are the functions of the Validate key?

CAMA 2000

7 ■ BASIC STATISTICS

Certain functions commonly used by assessors and appraisers have been incorporated into CAMA 2000. They are available under the **Valuation** option in the menu bar, These functions include:

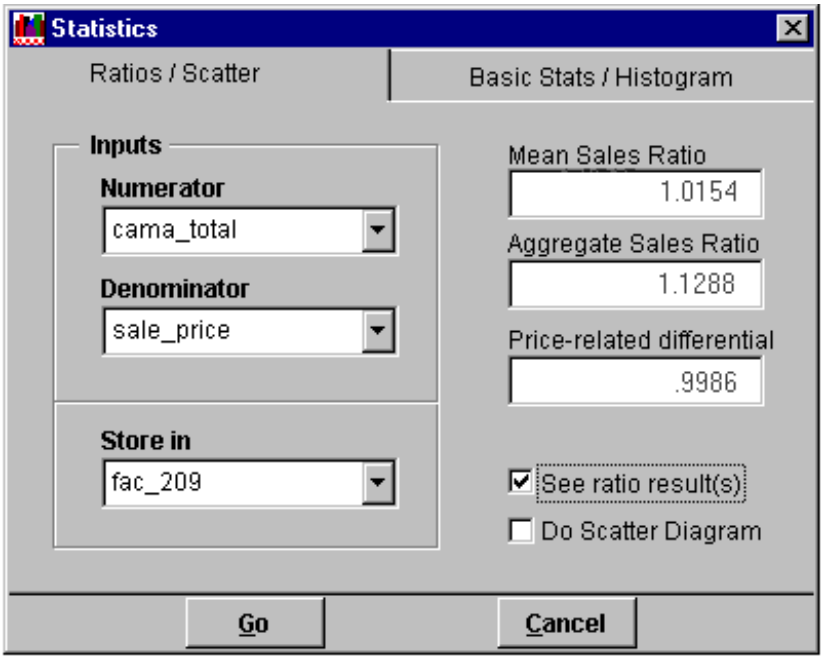
- Sales ratio analysis
- Descriptive statistics
- Scatter diagrams
- Frequency distributions and histograms

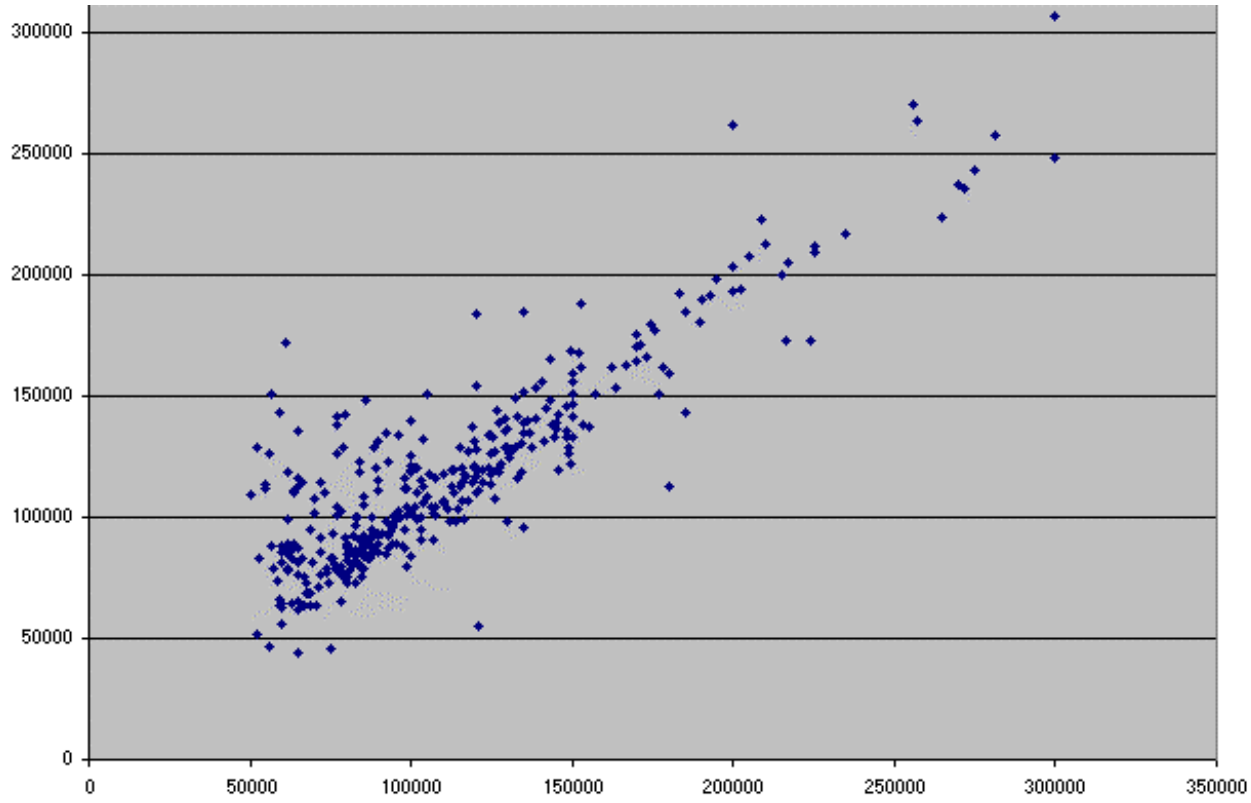
Sales ratio analysis

A sales ratio is produced by dividing an estimated value (such as the CAMA total which sums land and building estimates) by the sale price. On the main menu bar, click on **Valuation**, then on **Statistics**. With the Statistics window on the screen, click on the **Ratios/Scatter** tab if it is not already foremost. You may wish to set a filter so that only certain parcels (such as those that recently sold) are processed. Do this by clicking on the filter icon near the top of the screen. When the filter is in place, follow these steps:

- In the **Numerator** box, use the spinner to locate the factor you wish to place there. (This should be a factor representing the total estimated property value.)
- In the **Denominator** box, use the spinner to locate the factor representing the property selling price.
- In the box labeled **Store In**, use the spinner to locate a factor in which the result can be stored. It must be a numeric factor allowing decimal values.
- If you wish to see the results in tabular form, click to check the box next to **See ratio results** at the right of the form.
- If you wish to see a scatter diagram in which the estimated value is graphed against the sale price for all parcels, click to check the box next to **Do Scatter Diagram**.
- Then click **Go**.

In addition to the results that the system stores in the designated factor, the screen will show you the mean sales ratio, the aggregate sales ratio, and the price-related differential. If you have checked **Do Scatter Diagram**, the system will call up an Excel graphic screen to show the distribution of points representing CAMA total (y-axis) and sale price (x-axis). This graph may be printed. Click the x to return to the statistics screen.

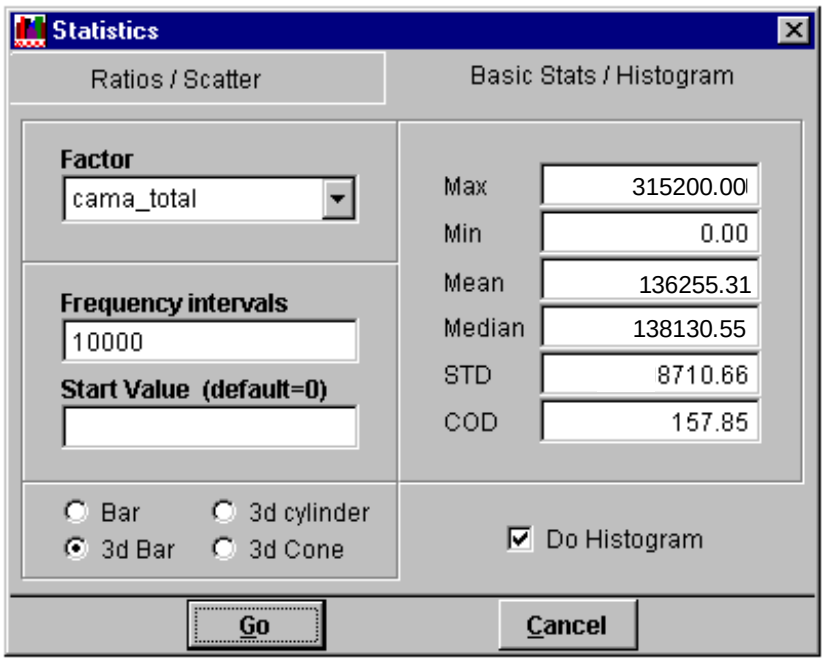


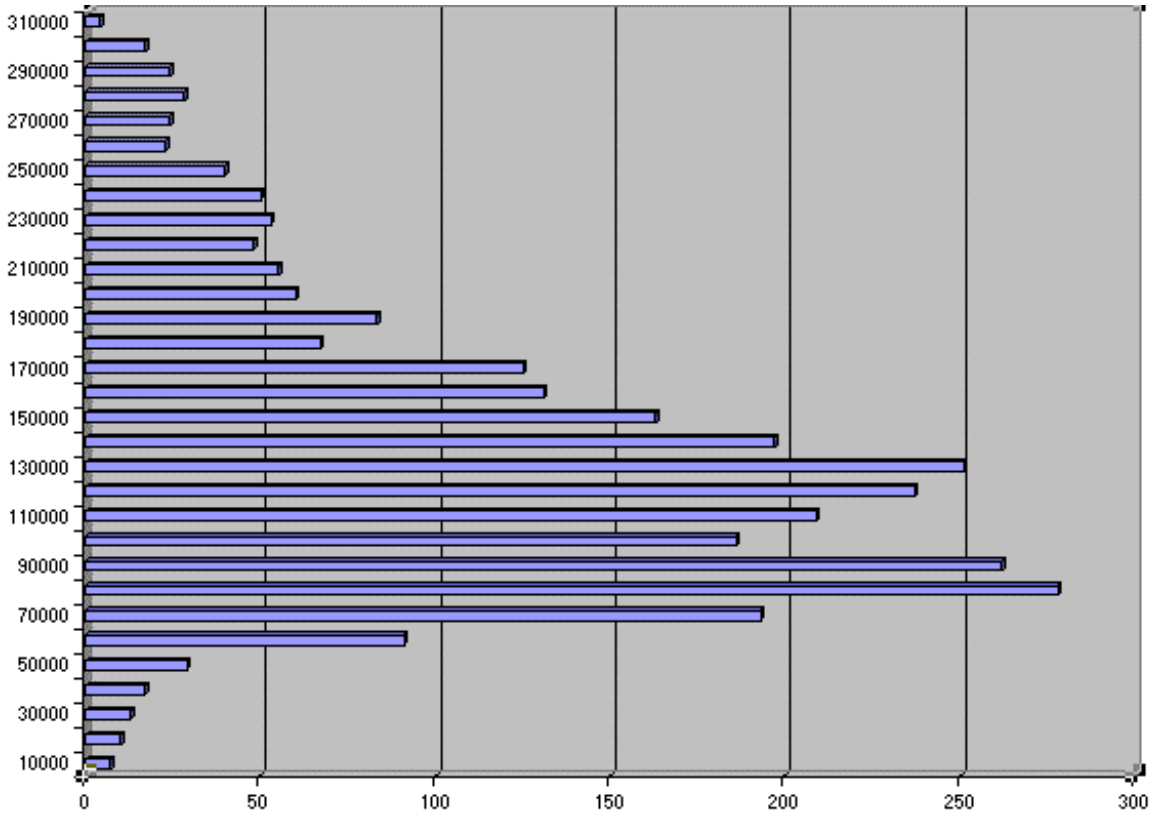


Basic statistics, frequency distributions, and histogram

Any numeric factor can be analyzed to determine certain basic statistical features about it: the minimum and maximum values, the mean (average), the median, the standard deviation, and the coefficient of dispersion about the median. To perform these functions, click on the **Basic Stats/Histogram** tab in the statistics window, then use the spinner to locate the factor on which you want the analysis performed. Click on **Go** to start the calculation. Results are presented in the boxes at the right of the window.

If you want to have frequency distributions displayed in graphic form (a histogram or bar graph), click to check the box next to **Do Histogram**. In that case you will also have to indicate the frequency interval within which records should be counted. For example, if you are analyzing sale price and you specify 20000 as the frequency interval, then all the parcels that sold for $0–19,999 will be counted in one group, all that sold for $20,000 to 39,999 in the next group, and so on. If you wish to start counting at a value other than 0, specify it in the "Start value" box. The program also allows you to specify the style of bar that Excel will use in displaying your bar graph. Click on **Go** to start processing.





REVIEW QUESTIONS

1. What is a sales ratio? How is it constructed in CAMA 2000?

2. How can you see a graphic representation of the sales ratio?

3. How do you create a histogram or frequency distribution report on a factor?

4. Where would you go to view the price-related differential associated with a sales ratio?

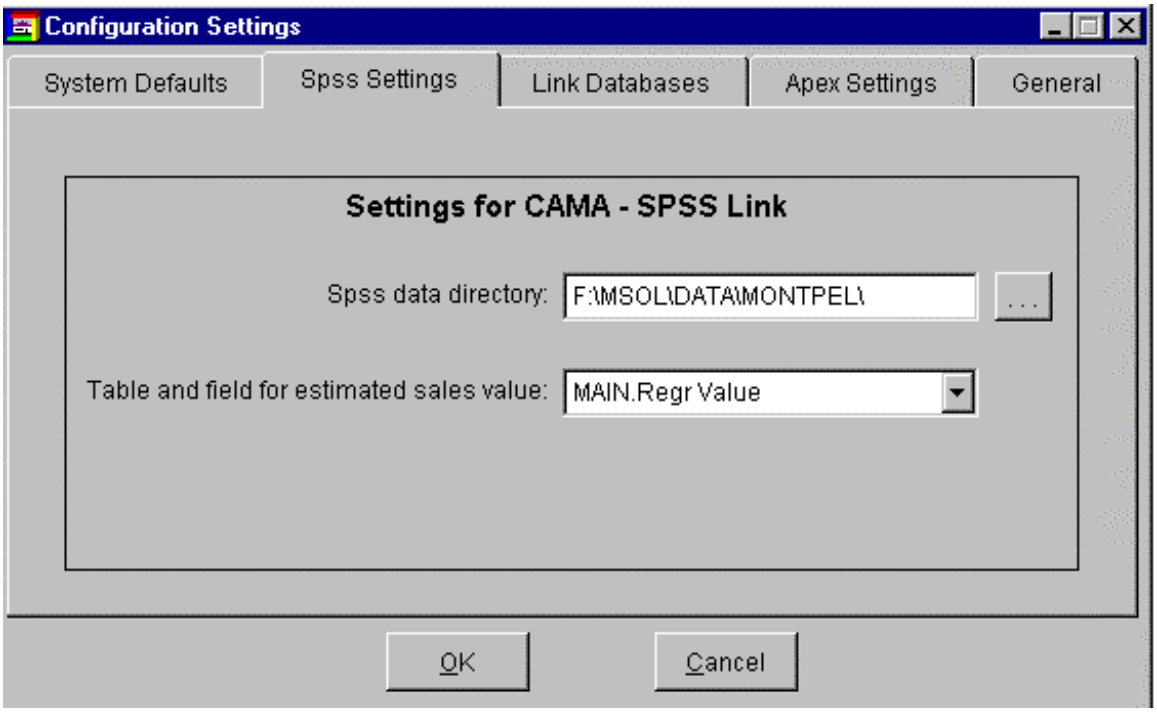
8 ■ SPSS STATISTICAL FUNCTIONS

Running multiple regression and most other statistical processes (other than basic sales ratio statistics) in CAMA 2000 requires that the program be integrated with SPSS. If you have not yet purchased the SPSS component, you will have to do so before you can proceed. Once you have purchased SPSS you should install it according to instructions. Then follow the directions below.

Configuring

At the Windows desktop, click on My Computer, then on Control Panel, then on ODBC Data Sources. (Newer versions of Windows may have other routes to the ODBC controls.) You are going to specify a CAMA 2000 database that SPSS will be able to read. Click the Add button, then from the list of drivers choose Microsoft Visual FoxPro Driver. On the screen for database information, enter the name of the database (e.g., York Residential), a description (optional), and the database type: Visual FoxPro Database (DBC). In the line for the path, you can use the browse button to locate the database. That will automatically fill in the correct path. Then click OK.

Sign on to CAMA 2000 as “sysadmin” (or have your administrator do so) and switch (if necessary) to the database that will be used in the regression process (the one named in the previous steps). Click Utilities, then Configuration. Click the SPSS configuration tab.



Browse to identify the data directory where the SPSS files should be stored. These are files that will be created from the subset of data you select for analysis. Also select the field in your database where the estimated sales value is to be stored.

Running Regression - Setup

Click on Valuation, then Regression/SPSS, then Run SPSS. SPSS will load while CAMA 2000 remains in the background. When it comes up, cancel the tutorial, if it appears. Click on File, then Open Database, then New Query. (Later, if you've already saved a query, you can click on Edit or Run a Query.) This brings up a "wizard" which guides you through the query-building process.

Your database (identified when you first configured the system) should appear on a list. Highlight it, then click Next. The database tables appear in a tree format. At the left of each table name is a + sign. If you click on this sign, a list of fields in the table will appear. You can now use the "hand" to grab any field and drag it across to the open window at the right. These are the fields that will be used in your regression run. Be sure to include the parcel ID field (so a record can be identified and later re-imported into CAMA) and a field for sale price, which will be your dependent variable.

When your list of factors is complete, click Next. If you have picked fields from Main and Section, you should see those tables with a connection line between them. If other tables are involved, they should appear as well. Click Next.

You will probably need to establish limits for the import process. For example, you may want to import only records in which Building SF > 0 and Sale Price => 50000. The screen labeled "Limit Retrieved Cases" allows you to build an expression that instructs the computer accordingly. You can use the spinners in the columns to select a field, or you can drag a field into the Expression 1 column, then use the spinner in the Relation column to select the relationship (=, >, etc.), and finally you can type the appropriate numeric value into the Expression column at the right. If there is more than one criterion, place AND or OR in the Connector column and continue as above.

You can proceed past the Define Variables screen, unless it is necessary to give other names to some variables so they can be more easily identified. Specify where you want to save your query, if you do want to save it. Then click Finish.

The program should create columns in the screen matching the fields you selected, and should stock them with the data from your database. If the columns do not contain values, check that your selection criteria were correctly formed.

If you included categorical factors among your independent variables, you will have to recode them to assign linearized values to the categorical levels brought over by SPSS. This must be done before you run the regression analysis. Use the "Transform" option in SPSS, choose "Recode ... into Same Variables," and after identifying the factor to be

transformed, pick “old and new values.” Each “old” value should be a level number, and each “new” value should be a linearized quality rating with the average value pegged at 100, lesser values ranging down to perhaps 80, and higher values ranging up to perhaps 120.

Running Regression – Analysis

Click Analyze, then Regression, then Linear. You must stock the Dependent [variable] window with the appropriate value – usually sale price. Highlight that variable in the list at left and click the arrow button next to the window. Then stock the Independent(s) window by highlighting the variables you want and clicking the appropriate arrow button. Click the spinner next to Method and choose Stepwise. Click the Save button and check the options under Predicted Values: Unstandardized, Standardized, and/or Adjusted. Then click Continue.

Under Options, indicate that you wish to use F values as the stepping method criteria. Typically a model builder will use an F value of 1 for entry and .5 for removal, but knowledgeable analysts may choose other values.

Back at the Linear Regression window, click OK. The program will run and you will next see an output window in which the regression model is specified, with the coefficients listed that ended up passing the significance tests for stepwise entry into the regression equation. You can save this information in an output file: click File, then Save As.

If you specified which predicted values should be saved, they will appear as additional columns at the end of your variable matrix. You should save this matrix (again, click File, then Save As) so the results column(s) can be imported into CAMA 2000. When you have done this you can either modify your model and run it again or exit and return to CAMA 2000.

Importing Predicted Values

Back in CAMA 2000, click on Utilities, then Import. You will get an import wizard which outlines four steps.

First, specify the source file – i.e. the SPSS file you just saved. (The SPSS Data import type is chosen by default.) Then specify the target database (the list is available by clicking the spinner). Ordinarily you will select the database you are currently signed on to, but this is not required. When the database is chosen, click Next.

Indicate which field(s) in the SPSS table you want to import, then indicate the factor(s) in CAMA 2000 into which they will be imported. Note that in order for the import to work, the SPSS file must contain the parcel ID, since that will be used to provide a match into this database.

When you have chosen both the source file and the target database, click the OK button. The program reads both files and the Next button becomes dark.

Next you indicate the field in the SPSS file that contains the predicted value you wish to import into CAMA 2000. Below that, indicate the factor in the CAMA database into which you want to copy the values. It must be a numeric field capable of storing a value of the appropriate size. Click OK when you have specified the fields, then click Next.

In the next step there is only one choice: Replace data based on target file primary keys. Click Next.

The final step is to finish the import. Click Finish. The system warns you that you are about to overwrite the destination factor with new data. Unless you have had second thoughts, click OK.

You have now placed MRA-derived values in each record of your database.

Applying the regression coefficients to unsold parcels

Multiple regression is a valuable tool for assessors because it yields coefficients allowing calculation of property values that are measurably close to actual sale prices. If a sufficient number of arm's-length sales were used as the basis for producing those coefficients, assessors can have confidence that even when applied to unsold properties, those coefficients will produce values close to what the property would bring in the current market.

The statistical process for determining how many recently sold properties are required to form an adequate basis for generalization is beyond the scope of this manual. However, as a rule of thumb, if you have 100 or more recent arm's-length sales, and the variations in quality or neighborhood type are not too extreme, you can use the regression coefficients based on these sales with some confidence to estimate values of similar properties.

The easiest way to apply regression coefficients is to use CAMA 2000's internal programming feature. Under Tools in the menu bar, select Modify Program, then go to the directory contaiing the program you are signed on to. There may already be a generic program there (with extension prg) which you can build into a regression tool. If there is not, ask MicroSolve's technical staff to provide you with a copy.

The generic program contains some lines of code near the top that create an array containing parcel IDs of all the records selected by the current filter. As a result, when the program is applied, it will affect only filtered records. Below the line of asterisks separating overhead from the changeable part of the program, you can delete old material there and enter the lines needed to apply regression coefficients, then save the resulting program under a new name. Here is an example of the code you might use, with comments explaining what the lines do.

```
use main in 0 order parcel_id                 && These are the tables that will be examined
use section in 0 order parcel_id              && in your database, and whose field values
use land in 0 order parcel_id                 && will be used to calculate an MRA value

FOR z=1 TO ALEN(laParcels,1)                  && Start a loop through the array laParcels
                                              && which is created automatically when the
                                              && program starts
  pid=laParcels(z)                            && Set variable pid to current member of array
  select section                              && Select section table
  mra=[constant]                              && Start the variable mra at value of the
                                              && constant produced by SPSS regression run
  if seek(pid)                                && What follows will occur only if record is found
    mra = mra + bldg_sf  *  [coeff 1]         && Building square feet are multiplied by the
                                              && relevant coefficient and added to mra total
    mra = mra + yr_built * [coeff2]           && Same with year built
    mra = mra + quality * [coeff3]            && Same with quality
   . . .                                      && Continue for all section table fields
  endif
  sele land                                   && Now choose the land table
  if seek(pid)                                && If correct record is found, do the following
    mra = mra + acres * [coeff4]              && Multiply acres by coefficient and add to mra
    . . .                                     && Continue for any other land variables
  endif
                                              && If other tables are involved, process them
  sele main                                   && Now select the main table
  if seek(pid)
    replace fac_210 with mra                  && fac_210 is called Regr Value
  endif                                       && This code stores the calculated value there
endfor                                        && Closes the "for" loop, allowing the program to
                                              && cycle through all the filtered records
```

Once this program is set up and tested, it can be modified to accommodate the results of future regression runs and saved under various names. Thus each version of the program can represent a particular regression model.

REVIEW QUESTIONS

1. How to you tell SPSS what database you want it to access from MicroSolve's CAMA 2000?

2. How do you tell CAMA 2000 what fields should be used to store incoming values from SPSS?

3. How do you set up a query in SPSS to import from CAMA 2000 the fields you intend to analyze?

4. What are independent and dependent values? How are they specified in an SPSS regression run?

5. What measures of “goodness of fit” are available in SPSS?

6. When you have produced a regression run that appears satisfactory, how do you import the predicted (estimated) values back into CAMA 2000?

CAMA 2000

9 ■ DATA DICTIONARY

CAMA 2000 allows a limitless variety of database structures and data elements. Users in different jurisdictions can configure their systems for local conditions and valuation purposes. This flexibility is made possible by the data dictionary, a tool that allows you to develop, within broad guidelines, any desired factors, factor types, and factor placement on the data entry form.

The following topics are covered in this chapter:

- Layout of the data dictionary
- Factor types and parameters
- Categorical factors
- Controlling factor behavior and placement
- Factor names and numbers
- The factor equivalence table
- Adding, modifying, and deleting factors
- Positioning factors on the display/input form

Layout of the data dictionary

To get to the data dictionary, click on **Utilities** in the main menu bar, then click **Data Dictionary**. A three-part form comprises the data dictionary: the left-hand tab is labeled **Detail**, the middle tab **List**, and the right-hand tab **Page name and field order**. The **Detail** panel contains most of the critical information pertaining to factor characteristics and behavior. The **List** panel is essentially a table allowing you to browse through all the factors. A factor highlighted in the **List** panel will be selected when the **Detail** panel is brought up. The third panel relates to the data display form and shows the tab labels associated with each factor, the order in which the factors appear, and the table in the relational database with which each is associated. If it is necessary to reorder factors or to associate one or more of them with a different page in the data display form, one must go to this panel to accomplish that.

Factor types and parameters

Factors used in the CAMA 2000 databases are typically any of the following types:

Character Alpha characters (text) in which numbers may be included but which are used for display and identification rather than calculation.

Numeric Any number, with or without decimal places. You can specify the size of the number and the number of decimal places it may have. If the number is used for a categorical factor, it must be three digits, with no decimal places.

Integer A number in which decimal places cannot appear. It may be positive or negative.

Date An eight-digit field which displays as 00/00/0000. The filter, the report writer, and Usit routines can perform date comparisons and date arithmetic based on these fields.

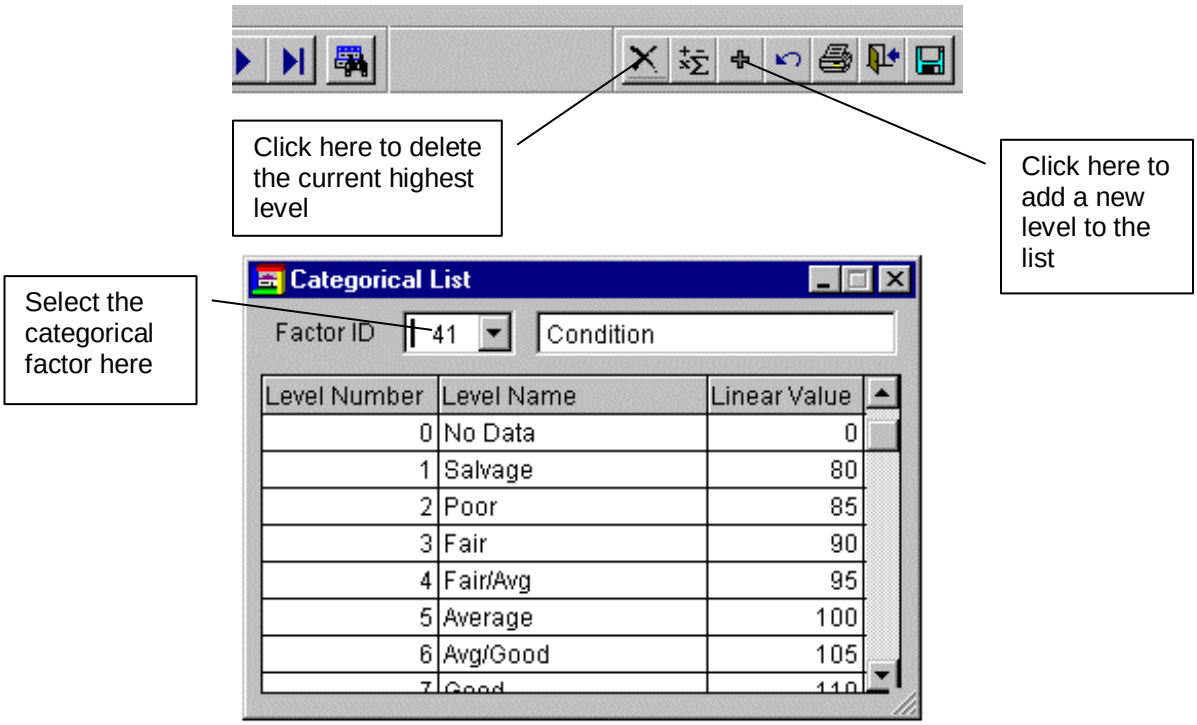
Memo A text field which is not limited in size, though it may be displayed in a window of restricted size.

General A field allocated for storage of a sketch or other object requiring a specialized format.

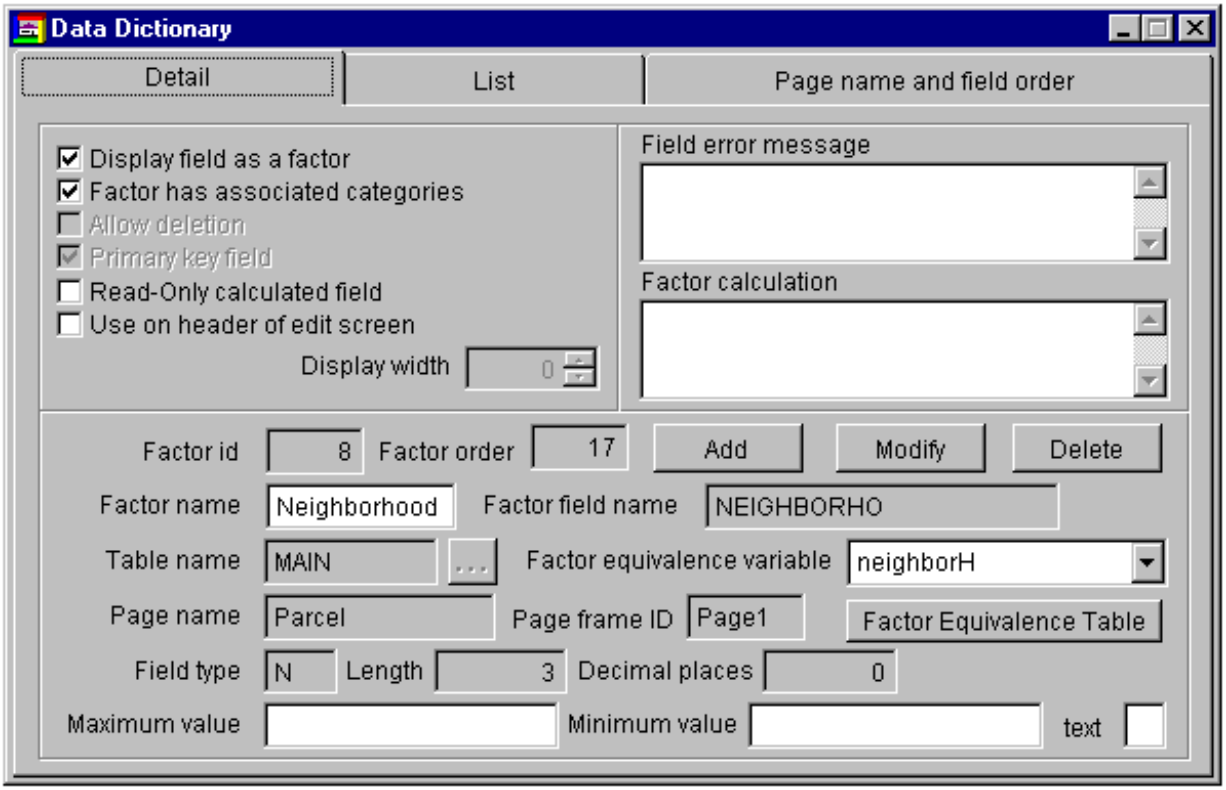
Factors can be changed from one type to another, even when the database contains data, but if this is done the database will no longer be compatible with others that initially resembled it. That would make it impossible, for example, to run comparables against an unaltered database when the subject was in a database that had been changed.

Categorical factors

Described in chapter 3, categorical factors are made from three-digit numeric fields. In the **Detail** page of the data dictionary, when a factor is selected that you intend to make categorical, you must check the box labeled “factor has associated categories.” Doing so creates an entry in the table of categorical levels. If you close the data dictionary and click on **Utilities**, then on **Categoricals**, you will see a spinner containing all the available categorical factors. Locate the one in which you wish to add or modify levels. If the categorical factor was newly created, it will have only level 0, labeled No Data. Add further levels by clicking the + in the toolbar near the top of the screen. In each added line, enter a level name and (if desired) a “linear value” indicating the relative quality of this level. Close the categorical window when you are done; your changes are saved automatically. (You can also delete levels; however, only the currently highest level can be deleted. Level names and linear values can be changed.)



Controlling factor behavior and placement



In the top left corner of the **Detail** page of the data dictionary are several boxes which can be checked or left unchecked. These determine the nature of the factor and how (or whether) it will be displayed.

To start at the bottom of the list, one may check "Use on header of edit screen." If that option is checked, the factor will always be displayed in one of the first two lines at the top of the display/input form. This box is usually checked for identifying factors such as Owner Name.

If "Use on header of edit screen" is checked, the option at the top of the list will read "Enable header field for updates." Place a check there if you want to be able to update the field from the keyboard. Leave it blank if they should not be allowed to alter the field.

If "Use on header of edit screen" is not checked, the option at the top of the list will read "Display field as a factor." Check this box if the field should show up on the data display/input form Only rarely is it desirable to have an active factor that is not displayed.

If the option "Factor has associated categories" is checked, the factor is treated by the system as a categorical. That means a list of categorical levels is created for it (you must add levels to the list if they are not already there) and the factor on the data display/input form will be accompanied by a "spinner" allowing you to select the categorical level for that factor.

The next two options – "Allow deletion" and "Primary key field" – are systems parameters that can be set only by MicroSolve support staff.

The option "Read-Only calculated field" can be checked if you want the field to be updated not from the keyboard but only as a result of a calculation. The field will appear grayed out on the screen, but its value will change if it is a result field for a calculation and if the calculation is rerun.

Beneath these options is an indented field called "Display width." If the last option – "Use on header of edit screen" – is checked, this field will be open for entry. It refers to the size of the box in which data can be displayed or input. That box is customizable if it appears in the first two lines of the form, where identifying information is displayed. It is measured in pixels. At a screen resolution of 800 x 600, 80 pixels make approximately 1 inch.

To the right of these option lines are two windows: for "Field error message" and "Factor calculation." Ordinarily these windows are used only by system support providers. The first allows them to indicate the error message the system will display if a field is incorrectly entered. The second allows specification, in the FoxPro programming language, of any calculation that might be done automatically to populate a given factor. For example, in the edit window of the cama_total factor on the main table, a user might place the following command:

```
replace main.cama_total with main.cama_dwell + main.cama_land
+ main.cama_outb + main.cama_siteimp
```

Whenever a person entering data clicks in the cama_total field, the field will be updated with the sum of the other specified fields.

There is a useful function available for the edit window: sometimes it is desirable to place the total value of all members of a subtable in a separate field of the parent table, and to have this field updated whenever a subtable value is changed. The **calc_sum** function accomplishes this. Its four parameters are the field to be replaced, the table containing that field, the field to be summed, and the table containing it. As an example, consider how a user might add up the areas of all porches attached to a house and put the total in a factor of the section table. In the edit window of the data dictionary field representing porch area he would place the following instruction:

```
thisform.calc_sum('porc_total', 'section', 'porch_area', 'porch')
```

Then, whenever a user changes a porch area, the total area (in the section table) will be changed as well.

In the lower portion of the **Detail** tab page are grayed out fields for Factor ID and Factor order. The ID field is important as a permanent numerical identifier of the factor. It correlates with the ID shown in the table associated with the **List** tab. The buttons in the same row – Add, Modify, Delete – refer to changes in the factor table and will be discussed later.

The next line shows two further ways of identifying the factor: “Factor name” and “Factor field name”. The latter is the name given when the database is created. Because of systems requirements, it cannot consist of separate words; all must be joined by an underscore. For that reason, the name listed in the “Factor name” box is the one that will appear on the data display/input form. It can consist of separate words in upper or lower case or a combination. You can customize this name as you choose.

Below that line, the table name is shown. Although you cannot change this name, it is important to know what it is: it represents the table in the relational database structure (see p. 3-2) with which this factor is associated. Next to it is an indication of the “Factor equivalence variable,” yet another way of identifying the current factor. For use in Usit routines, each factor must be uniquely identified, regardless of its original name, its position, or which relational subtable it belongs to. The factor equivalence table (discussed further below) is a list of such unique identifiers, tied to the factor ID and the table in the relational structure to which each belongs.

The line below shows the “Page name” (i.e. the tab label for the data display/input form page on which the factor appears), the “Page frame ID” (an internal system marker), and a bar on which you can click to bring up the “Factor Equivalence Table.” The use of the factor equivalence table will be demonstrated later in this chapter.

The next line shows “Field type” (the types are listed on p. 9-2), “Length” (number of characters or digits left of the decimal), and “Decimal places.” These fields are for infor-

mational purposes only. They can be altered only by pressing the "Modify" button, a process described later.

On the bottom line are fields in which you can indicate the maximum and minimum allowable values for a field. If these fields are not filled in, any value is allowed. The rightmost box, "Text," allows input of a single character such as **%**. This character will appear immediately to the right of the field in the display/input form. It can indicate that the value represents a percentage, or if it is a different character (<, for example), it may be used to indicate that this field is essential to achieving a property value via the Usit cost routine.

Navigating through the factors. Only one factor at a time is displayed on the **Detail** page. You can move from one factor to the next, displaying each one in turn, by clicking on the arrow keys at the top of the page. The default order, called *Cfacorder*, is that of the current factor arrangement as shown in the **Page name and field order** tab page. You can change this order by selecting a different index, but you will probably find the default order the most useful.

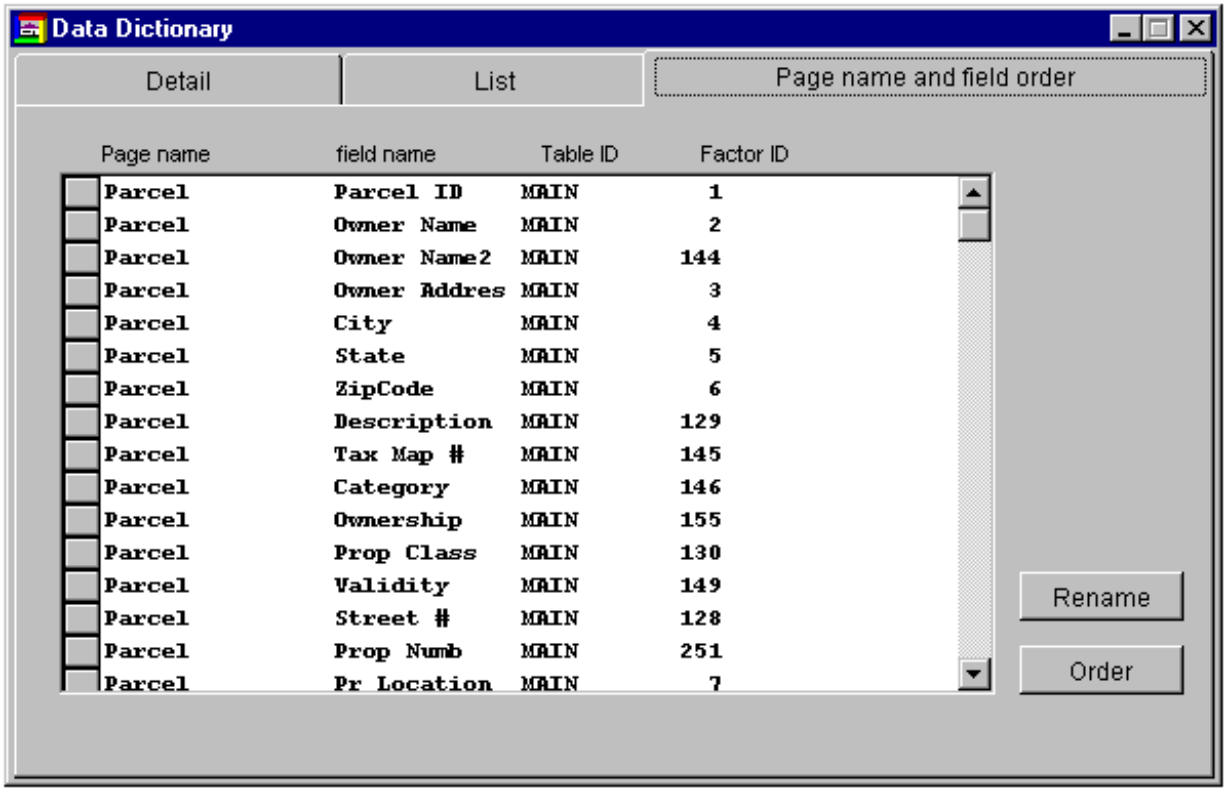
The other way to locate a particular factor is to click on the **Page name and field order** tab, then find the factor you want in the list. This screen has an interesting and useful arrangement. It is organized by the pages of the data display/input form. All the factors that appear on a particular page are grouped together, in the order of their appearance (reading from the top of the first column down, then from the top of the second column, and so on). Each factor name is shown as it appears on the display form, as is the identifier of the table with which that factor is associated. (Each factor in the database is associated with one and only one table, although identifying factors such as Parcel ID, Section ID, and the like will appear as well in subtables that they control.) Finally, each factor on this page displays its factor ID, which is the unique number associated with the factor and indicates the order in which it was originally created. By highlighting a factor in the list on this page, then clicking on the **Detail** tab, you will immediately bring up all the details on that factor for examination or change.

Repositioning a factor

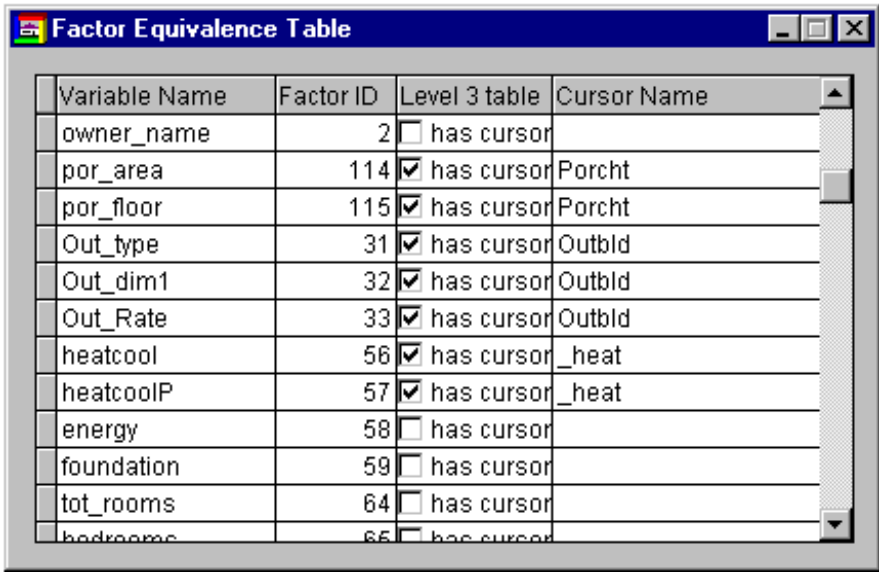
Factors can be arranged to appear on the data display/input form in a location of your choice. The form has numerous pages, whose names you can also choose and modify, though they are broadly associated with the functions and subtables in the relational database (e.g., land, outbuildings, section, etc.). When positioning factors, bear in mind that each tab page contains a maximum of three columns, and each column contains up to ten factors. Factors designated for a given page will be arranged down a column until there are ten of them, then they will start a new column.

You can use the **Page name and field order** tab to reorder one or more factors. Click on that tab and find the factor you wish to reposition. At the left of the page name in the factor line is a box. Click on this box and drag it up or down to the location you want. Understand that this will move subsequent factors on that page down one notch. Now click

the **Order** button to establish the new factor order. If you have moved the factor from one page to another, you must rename the page associated with the factor, otherwise the system will treat your inserted factor as belonging to a single tab page inserted between two other pages. Rename the page by clicking on the factor to highlight it, then on **Rename;** then type in the proper page name exactly as it appears for the surrounding factors. Press <Enter>. The factor is now repositioned.



The factor equivalence table



As stated in the previous section, the factor equivalence table is a list of unique identifiers associated with each factor that may be called in the Usit routine. If a factor does not

have an entry in the factor equivalence table, the box on the **Detail** page next to the label "Factor equivalence variable" will show "No Variable." Should it become necessary to add an entry in the factor equivalence table for a particular factor, here are the steps to follow:

- Bring up the **Detail** page for the factor you wish to enter.
- Note the factor ID number; this is the unique identifier for that factor.
- Click on the **Factor Equivalence Table** bar to bring up the table.
- Because you wish to add an entry to the table, click on the + button in the toolbar and then scroll to the bottom of the table, where a new line is available
- In the first column enter the unique name by which this variable should be called from the Usit routine. In the next column, enter the factor ID number you had noted previously.
- If the factor belongs to the Main or the Section table, you do not need to make any further entries. Click the x in the upper right corner of the window to close it.
- If the factor belongs to any other table (including land), check the box next to "has cursor" and type the name of the table exactly as it appears elsewhere in the factor equivalence table. Note that some table names begin with an underscore, and others do not. Close the window when you are finished.

Adding, modifying, and deleting factors

To add a factor to your database, you must know which table in the relational structure you want your factor to belong to. Then you must bring up the **Detail** page for a factor associated with that table. That is because the system will treat an "add" instruction as a request to add a line to a table currently active in the data dictionary. Once you have made these preparations, follow these steps:

- Click on **Add** in the **Detail** page, to bring up a window called "Add a field to the database." In this window specify the field name (if you use the wrong form the system will automatically "edit" your name), the field type (chosen from the spinner list), the length, and if appropriate the decimal places. Click **OK** to return to the **Detail** page. If your database already has data, there will be a pause while the program sets a default value (blank or zero) for all current records.
- Your new factor now appears on the **Detail** page. Place a check in "Display field as a factor." If the factor is to be categorical, you must also check "Factor has associated categories." In the lower part of the form, next to Factor name, type the name of the factor as you wish it to appear on the data display/input form.

- Now click on the **Page name and field order** tab so you can position your factor as you want it. Scroll to the bottom of the list and locate your factor there. It is probably on "NoPage." Click on the box to the left of the page name and drag it up to the position where you want it, for example to the Outbuilding page. Place it exactly where it should go in relation to the other factors. Click the **Order** button to reset the order of the list so your factor will stay in this position.

- Click again on the factor to highlight it so it turns a solid color. Then click on **Rename**. Type in the name of the tab page exactly as it appears in entries for the neighboring factors. Press <Enter>. Your factor is now created and positioned so it will appear on the form and can receive data from the keyboard.

Modifying an existing factor is much like adding a factor. Bring up the factor in question and click on **Modify** on the **Detail** page. On the ensuing window the field name is not changeable, but the field type, length, and decimal places are. Again, if your database contains data, there will be a pause while the records are adjusted to accommodate the new factor type.

To delete a factor, proceed in the same way as above, but click on **Delete**. You will be given an opportunity to confirm the deletion by clicking **OK**. Click **Cancel** if you wish to rescind the order.

REVIEW QUESTIONS

1. What are the functions of the data dictionary?

2. How would you move a factor so it displays on a different page, or before another factor?

3. How do you modify a factor?

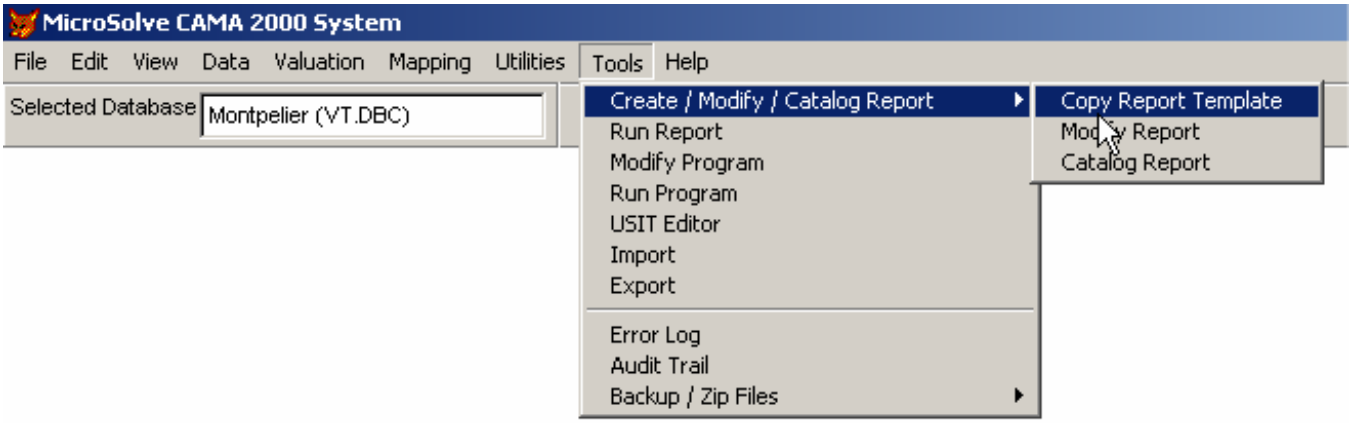
4. How do you add a factor?

5. What are the steps in creating a categorical factor?

10 ■ REPORT DESIGNER

The Report Designer allows you to create custom reports based on data contained in the parcel records. One example is the template for the cost report, but many other reports are possible. Using this tool, for example, you can create a custom property record card or an assessment notice.

Under the Tools menu choice there is a line reading Create/Modify/Catalog Report. This line leads to three separate menu items: Copy Report Template, Modify Report, and Catalog Report. The following sections describe each of these functions.



Copy Report Template

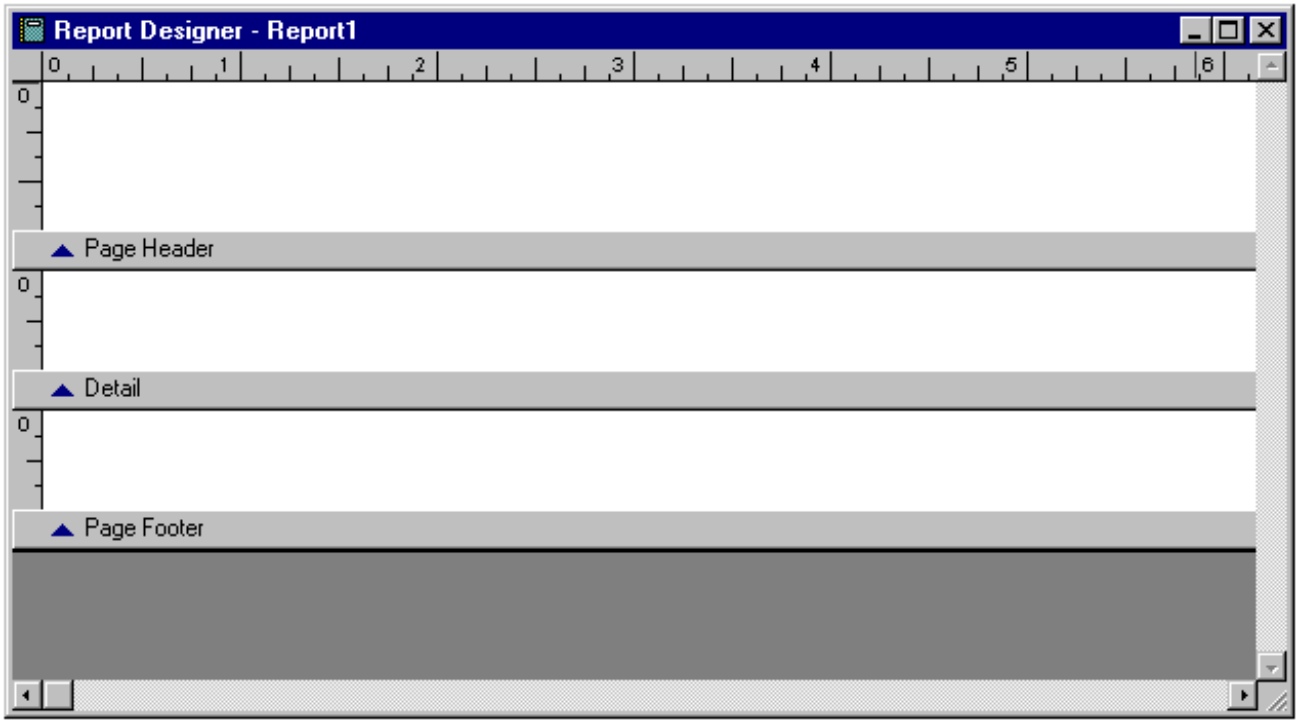
Choosing this function takes you to a list of the standard report templates in the Repts folder. **Mscost**, **commcost**, and **costrpt** are templates for, respectively, the standard residential Marshall & Swift cost report, the commercial Marshall & Swift cost report, and the condominium cost report designed for use with MicroSolve's condo database. **Comp01** reports work with the comparables module. The suffixes **s** and **vs** indicate small and very small fonts respectively, allowing reports to fit on a single page when numerous factors are listed. **Comp02** is the comparables report that shows photos of the subject and its comparable properties. **Prcnew** is the template used for the residential property record card, and **commcard** is used for the commercial property record card.

Modify Report

Window options

Bands. By default, the Report Designer displays three bands: Page Header, Detail, and Page Footer. A separator bar is located at the bottom of each band. The name of the band

type appears on the bar next to a blue arrow, which indicates that the band is above, not below, the bar.



You can add the following bands to your report.

Band	Printed	Typical Contents
Column Header	Once per column	Column title
Column Footer	Once per column	Summary, totals
Group Header	Once per group	Preface to following data
Group Footer	Once per group	Calculated values for group data
Title	Once per report	Title Date or page number Company logo Box around the title
Summary	Once per report	Grand totals Text such as Grand Totals

Ruler. The Report Designer has a vertical and horizontal ruler you use to more precisely position objects in the bands. Use the ruler in conjunction with the View menu's Show Position command to help you position objects.

The scale for the ruler is determined by your system measurement settings. You can change from the system default scale (inches or centimeters) to pixels from within Visual FoxPro. If you want to change to the system default, change the measurement setting for your operating system.

- To change the ruler scale to pixels

1 From the Format menu, choose Set Grid Scale. The Set Grid Scale dialog box is displayed.

2 In the Set Grid Scale dialog box, select Pixels and choose OK.

The ruler scale is set to pixels, and the position indicator in the status bar (if Show Position is checked on the View menu) also displays positions in pixels.

Report controls toolbar

You can create controls on your report or label with the report controls toolbar. Click the button for the control you want, position the mouse pointer on the report, and click the report to place the control or drag it to size.

On a report, you can double-click any control to display a dialog box for setting options.

If this toolbar is not displayed when you open the Report Designer, click on **View** in the menu bar, then click on **Report Controls Toolbar** to activate it.

This toolbar contains the following buttons.

Button	Description
Select Objects 	Resizes and moves controls. After you create a control, the Select Objects button is automatically reselected, unless the Button Lock button is pressed down.
Label **A**	Creates a label control, used for text that you don't want to change, such as a caption above check boxes or under a graphic.
Field 	Creates a field control, used to display the contents of a table's field, memory variable, or other expression.
Line 	Use to draw a variety of line styles on your report at design time.

Rectangle 	Use to draw rectangles on your report.
Rounded Rectangle 	Use to draw rectangles with rounded corners and ellipse shapes on your report.
Picture/OLE Bound Control 	Use to display a picture or contents of a general data field on your report.
Button Lock	Allows you to add multiple controls of the same type without having to click the control button on the toolbar more than once.

Report menu

The Report menu contains options that allow you to create and modify reports:

Title/Summary	Displays the Title/Summary dialog box, in which you specify whether a Title and/or Summary band is included in the report.
Data Grouping	Displays the Data Grouping dialog box so you can create data groups and specify their properties.
Variables	Displays the Report Variables dialog box, in which you create variables within a report.
Default Font	Displays the Font dialog box, in which you specify the persistent fonts, font styles, and font sizes for label and field controls in reports and labels. The settings are saved with the report, so each time the report is modified, the default is the same.
Private Data Session	Opens the tables used in the report in a private set of work areas so that they are not affected by other reports, forms, or programs. Click this command to turn it on or off.
Quick Report	Automatically places selected fields in an empty Report Designer window. The command prompts you to select a table and then displays the Quick Report dialog box, in which you select fields and a field layout.
Run Report	Displays the Print dialog box so you can send the report to a printer. Running a report doesn't change the data in the table, index, or memo files.

The following items are included for completeness but are seldom needed in MicroSolve reports.

Data environment designer

The Data Environment Designer can be used to visually create and modify the data environment of forms, form sets, and reports. When the Data Environment Designer window is active, Visual FoxPro displays the Data Environment menu, which allows you to manipulate data environment objects. To display the Properties window and Code window, right-click to display the Data Environment Designer shortcut menu, and then select Properties and Code.

Note: In most reports designed for use with CAMA 2000, there should be nothing in the data environment. That is because all the necessary tables are already open and automatically available to the report when it is run. Adding tables to the data environment will tie the report to only the database for which it was created. It cannot then be transferred to any other database.

When the data environment of a form or report is defined, Visual FoxPro automatically opens the tables or views when you open or run the file and closes them when the file is closed or released. However, in CAMA 2000 all the files already have to be open in order to support the SET FILTER option. So the Data Environment should *not* be opened when constructing reports to be used in CAMA 2000. By not specifying the environment, you also make it possible to pass a report from one database to another. If the report was not created with reference to a particular database, it won't balk when used with another, provided the requisite fields can be found in the new database.

Report variables dialog box

The dialog box (Report > Variables) lets you create variables within a report. Visual FoxPro uses variables to store the results of calculations performed while the report is being printed. Use this command to add new variables, change or delete existing variables, or change the order in which the variables are evaluated.

Dialog box options

Variables	Displays the variables in the current report, and provides space for new variables.
Value to Store	Displays expressions stored in the current variable, or allows you to type an expression in the text box. To create an expression that will be stored in the variable, choose the dialog button to display the Expression Builder dialog box.
Initial Value	Displays the selected variable's value before any calculations are performed, and the value to which the variable is reset. You can enter a value in the text box, or, if you want to create an expression for the initial value, choose the dialog button to display the Expression Builder dialog box.

Release after Report	Releases the report variable from memory after the report is printed. If this option is not selected, the variable remains available in memory until you exit Visual FoxPro or release the variable with the CLEAR ALL or CLEAR MEMORY command.
Reset	Allows you to specify the point at which the variable is reset to its initial value. By default, End of Report is displayed in the list box. You can also select End of Page or End of Column. If you have specified groups in a report in the Data Grouping dialog box, the Reset box displays an item for each group in the report.
Insert	Inserts a blank text box in the Variables box so you can type a new variable name.
Delete	Deletes the selected variable name from the Variables box.

Calculate options

These options allow you to specify a calculation that the variable performs. The variable begins calculating with its initial value, and continues until it is reset to this initial value.

Nothing	Specifies that no calculations will be made on this variable.
Count	Counts the number of times a variable is printed per group, page, column, or report (depending on your selection in the Reset box). The calculation is based on the number of times the variable occurs, not on the variable's value.
Sum	Computes the additive sum of the variable's values. This option keeps a running total of the variable values for a group, page, column, or report (depending on your selection in the Reset box).
Average	Computes the arithmetic mean (average) of the variable values within a group, page, column, or report (depending on your selection in the Reset box).
Lowest	Displays the lowest value that occurred in that variable for a group, page, column, or report (depending on your selection in the Reset box). Places in the variable the value of the first record in the group. When a lower value is encountered, the variable's value changes accordingly.
Highest	Displays the highest value that occurred in that variable for a group, page, column, or report (depending on your selection in the Reset box). Places in the variable the value of the first record in the group. When a

higher value is encountered, the variable's value changes accordingly.

Std. Deviation	Returns the square root of the variance for the variable values within a group, page, column, or report (depending on your selection in the Reset box).
Variance	Measures the degree to which individual field values vary from the average of all the values in the group, page, column, or report (depending on your selection in the Reset box).

Data grouping dialog box

Lets you create group bands for separating groups of records visually, and displaying introductory and summary data for each group. These properties include printing text in headers and footers to identify specific groups, printing each group on a new page, and resetting page numbers when groups are printed on a new page.

Dialog box options

Group Expressions	Displays group break expressions, such as a field name, for the current report and allows you to type new ones. If you want to build an expression, choose the dialog button to display the Expression Builder dialog box.

Group Properties

These options allow you to specify options for page breaks.

Start Group On New Column	Starts a new column every time the group changes.
Start Each Group On a New Page	Starts a new page every time the group changes.
Reset Page Number to 1 for Each Group	Starts a new page and restarts page numbering every time the group changes.
Reprint Group Header On Each Page	Specifies that the group header follow the page header on all pages for the group when the group spans more than one page.
Start Group On New Page When Less Than	Sets the minimum distance from the page bottom that a group header is printed.
Insert	Inserts a blank text box in the Group Expressions box so you can define a new group expression.

Delete	Deletes the selected group expression from the Group Expressions box.

Expression builder dialog box

Allows you to create and edit expressions. An expression can be as simple as a field name or as complex as a calculation involving immediate IF functions, concatenations, and data type conversions. The main purpose of the Expression Builder is to facilitate the creation of expressions by providing you with lists of appropriate options each step of the way. This dialog box can be accessed from designers, windows, builders, and wizards.

To create an expression, you can type it directly into the expression box, or you can select items from the functions drop-down lists in the dialog box to have Visual FoxPro paste them into the expression box for you.

You might find some of the following functions useful when manipulating strings in expressions.

If you want to	**Use this function**
Remove leading and trailing blanks from character expressions	ALLTRIM( )
Remove leading blanks	LTRIM( )
Remove trailing blanks	RTRIM( )
Add specified characters to the left, right, or both sides of a string	PADL( ), PADR( ), PADC( )
Work with parts of a character string for comparisons	SUBSTR( )
Use a specified number of characters starting with the left of a string	LEFT( )
Use a specified number of characters starting from the right of a string	RIGHT( )
Change uppercase to lowercase, or lowercase to uppercase	UPPER( ), LOWER( )
Convert a string to initial capitals	PROPER( )
Have a numeric field interpreted as a character string	STR( )

Dialog box options

Functions Contains list boxes of four function types. When you select a function from one of the four types, Visual FoxPro automatically pastes it into the expression box. When you are building expressions for remote views, Visual FoxPro lists only the functions specific to the target back-end data.

String	Lists available character string functions.
Logical	Lists available logical functions.
Math	Lists available math functions.
Date	Lists available date and time functions.
Expression	Displays the expression that you are creating or editing.
Fields	Lists the fields in the current table or view. To paste a field into the Expression box, either double-click the field or select the field and press <Enter>. To display fields from a different table, select a different table in the From Table box.
From Table	Lists tables and views that are open. Select a table or view to update the Fields box.
Variables	Lists system memory variables, arrays, and memory variables that you have created. To paste a variable into the Expression box, either double-click the variable, or select the variable and press <Enter>.
Verify	Validates the syntax of the expression in the expression box if the corresponding table is open. If the expression is valid, “Expression is valid” is displayed in the status bar. If it is not valid or if the corresponding table is not open, Visual FoxPro displays an error message. This option is not enabled for remote views. Note If you include a user-defined function call in the expression, Verify will indicate an error, but there will not necessarily be an error when the expression is evaluated at run time.
Options	Displays the Expression Builder Options dialog box, in which you can set preferences for the Expression Builder.

Property record cards

The CAMA 2000 property record card is a specialized kind of report, making use of the Visual FoxPro Report Writer, which is available as part of the CAMA 2000 package.

Certain features have been added to the report writer to enable it to handle properties in an assessor's database:

- A function to display level names (not just numeric values) for any categorical factor
- An ability to produce a two-page report for each property
- An ability to produce additional pages in the case of multi-section properties
- Routines to cycle through all the members of second-level tables (e.g. land) and third-level tables (e.g. porches) to display relevant data before proceeding to the next property characteristics

Organization of the property record card

As with other FoxPro report forms, the form used in the property record card is divided into sections by bands. The bands that will be found in the residential and commercial property record cards, together with the material associated with each, are as follows:

PAGE HEADER	[Not used]
GROUP HEADER 1	Parcel information Valuation information Arrays for second-level tables: Land, Site Improvements, Outbuildings
GROUP HEADER 2	Main building information (from section table) Notes Arrays for third-level tables
DETAIL	[Not used]
GROUP FOOTER 2	[Not used]
GROUP FOOTER 1	[Not used]
PAGE FOOTER	[Not used]

Thus, all the information for the first page (overall parcel data plus land, site improvements, and outbuildings) is contained in group header 1, and all information for the second page (section data, which will be repeated if there are more sections) is contained in group header 2. The page header can be used for information that will be repeated on each page; in this case the title "Residential Property Record Card" must be accompanied on the second page with the section number, so it is placed at the top of each group header rather than in the page header. Each header is automatically paired with a footer, but the footers are not used in this report.

Broadly speaking, reports including the property record card contain data consisting of *labels* and *fields*. To add or modify either, you should select the View option in the toolbar, then choose Data Environment (minimize the display after selecting) and Report Controls Toolbar. Labels can be typed in anywhere, once you have clicked on the A icon.

Fields are selected by clicking on the abl icon, then picking the appropriate field name from the list. Note that the "raw" names are used, not the more user-friendly names by which the fields are identified on the data display form. These "raw" names may need to be looked up in the data dictionary.

The "arrays" are seen on the report form as packed fields eleven columns across and 15 (first page) or 30 (second page) rows deep. These fields do not have to be identified with particular factors. Rather they are available to be populated with data from the record subtables once the relevant fields are identified in the data dictionary. The program automatically places the table name in the first column, the field names in the first row, the field values in the second row, and any values from subsequent records of the subtable (e.g., additional porches) in the following row(s). A further row in the array is left blank (producing a space on the page) before information from the next subtable enters the array.

Therefore *only main and section factors need to be specified in the property record card report form*. Factors from other second-level tables and all third-level tables will be automatically read into the arrays if they are identified in the data dictionary as "available for report." Bear in mind when doing so that a maximum of ten factors can be so specified for any one second- or third-level table.

Creating and maintaining property record cards

It is much easier to start with a property record card designed for a different database and adapt it, rather than create one from scratch. Either the standard residential or commercial property record cards (two pages each) or the standard condo record card (one page) can be used as a basis. If the property record card you choose was developed for a database with the same name and structure as your present one, you should follow procedure A. If not, use procedure B.

Procedure A (a property record card developed for a database with the same name and structure): Copy the property record card files (e.g., RESCARD.FRT and RESCARD.FRX) into the local directory where your database files reside. Sign onto CAMA 2000 (this database), and bring up the data dictionary under Utilities. Working your way through the second-level tables other than Section and all the third-level tables (the children of Section tables), find the factors currently in use that you wish to display on the PRC. On the detail tab of each such factor, check the box called "available for report." When you are finished, you can run the property record card report to check the setup.

Procedure B (adapting a property record card developed for a different database): Starting in the directory of the database for which the card was developed, copy the files to the new database's directory and rename them, keeping the same extensions. Then copy them back to the "home" directory, sign onto the originating database, and use Tools, Create Report to open the frx table. From the View menu in the toolbar, call up the Environment window. Delete the existing tables in the environment, then save the file. Using Explorer, cut the two property record card files (-.frx and -.frt) from the directory and paste them into the new database's directory. Sign onto the new database and access the file again. Click on Environment and Add in the toolbar. Add Main and Section tables from the new database. *Make sure you get the right tables from the right folder*; this will determine where all the data displayed in the property record card comes from. Close FoxPro and sign onto CAMA 2000 and the new database. Go to Create Report under Tools and locate the correct frx file under the local directory. Make whatever changes are needed to bring the property record card into line with the new database. Remember that only records in the Main and Section table should be used in populating the card. Fields in other tables will be flagged through the data dictionary as described in Procedure A.

Where a categorical level name from the Main or Section table is to be displayed instead of a level number, you must indicate this by a special function. (In the arrays populated from other second-level tables and all third-level tables, the factor level names will display automatically if the factor is categorical.) Double-click on the field box. In the Expression space of the Report Expression window you will see the name of the factor that will appear in this position on the report. Edit it so that it reads: getcateg('*factorname'*). Then click OK.

After the information from the Main and Section tables is displayed, you can control only which tables and which factors in those tables will contribute information to the record card, but you cannot control their order of appearance. After you have identified them by clicking "available for report" in the data dictionary, the tables will be presented in alphabetical order going down the page, and the fields in each table will appear in alphabetical order going across the page.

Photos and sketches can be placed on the property record card, using the "picture" button on the report controls toolbar. Photos are identified by placing the host table and field name (e.g. photos.photoloc) in the blank next to "File" on the control form. Sketches are identified by placing the following expression (including parentheses) in the same place: (tmpsktchs.tmpname)

If a sketch is to be printed in the property record card or any other non-standard report, the report name must include the embedded letters 'skt.' For example, a property record card name could be rsktcard.frx. More details on combining sketches with other parts of the CAMA system will be found in chapter 13.

Using the property record card

The Tools option on the menu bar allows you to choose Run Report. Any report you have created via the Create Report option can be run from this interface. In all cases you are able to choose a record via an index, to indicate whether you want to run in “print” or “preview” mode, and to say whether you want to display a single record or all records. (If you choose “all” records keep in mind that the program observes any filter that may be in place, so you will get reports on just those records that are admitted by the filter.)

Once you have made your choices, click on the Select Report button. This will show you the available report templates (frx files) in your local data folder. Double-click on the one that represents the property record card, and the card will be displayed on the screen or sent to the printer. If it is displayed on the screen, you still have the option of clicking on the printer icon to print a copy. You can also use the navigation buttons to move back and forth between pages. Close the preview by clicking on the door icon or the printer icon.

Certain printers may allow printing of double landscape pages of card stock, joined along one edge. These pages can then be folded to produce a single "card" that can be placed in a taxpayer file. However, most often users will print in landscape mode on ordinary single-sheet paper and may staple two sheets together to form a “card.”

REVIEW QUESTIONS

1. How do you bring up the Report Designer in CAMA 2000?

2. What are the functions of the report controls toolbar?

3. How do you change the font or size of a label or data element on the report?

4. How is the Data Environment Designer used in the creation or modification of reports?

5. What are report variables and how are they used?

6. How can you cause the report to perform calculations?

7. What is an expression and how do you construct one?

8. How do you modify details of a property record card template?

11 ■ IMPORT/EXPORT AND OTHER UTILITIES

This chapter discusses the MicroSolve standalone import tool known as Itool. Other utilities covered include:

- exort functions
- user maintenance
- the error log
- DBC maintenance
- the *verify structures* feature
- creating and using a sales history file

Using Itool, the standalone data transfer protocol

Itool is a standalone utility designed to work with CAMA 2000. It can be used to import data from ASCII files into CAMA 2000 databases, or to transfer records from one CAMA 2000 database (or any other dbf-format database) to another. The following instructions are for people who wish to import data from an ASCII source file into the corresponding fields of a database.

Installation

To install the program, create a folder called ITOOL and copy the program files into it. All folders under the ITOOL folder on your source disk should also be copied. If your source files are zipped up, you should simply unzip them into an empty folder, where they will be arranged in the proper hierarchical order.

Importing from an ASCII *source file to a database*

Let us assume you are working with a destination CAMA 2000 database called VT.DBC and that you have a source flat file to import. The source file may be either delimited or fixed-field, and it is in ASCII format. The following instructions do not have to be followed rigidly; they are meant to ensure orderly and accessible locations for the source and destination files.

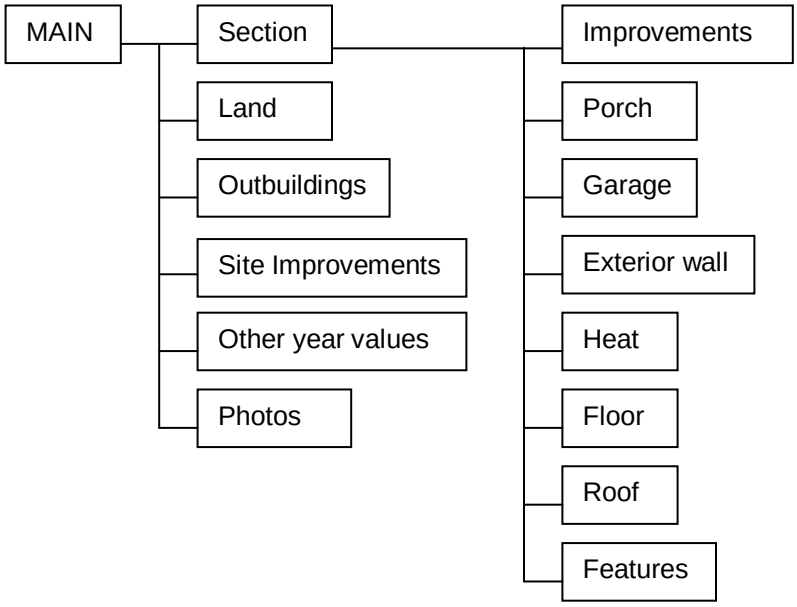
Under the ITOOL folder create a subfolder called VT, and under that create two subfolders: VTDEST and VTSOURCE. In the VTDEST folder you should place a *copy* of the destination database into which you wish to import data. That means you will copy the contents of an entire folder with all the files that comprise the database (MAIN, SECTION, HEAT, etc.) into the VTDEST folder. (If you do not want to retain data that may presently be in the destination files, they can be purged (zapped) by the program before new data is imported. However, it is also possible to add data from your source files to data that already exists in the destination files.) Into the VTSOURCE subfolder copy the ASCII file or files you wish to import *from*.

Configuring the import setup

We will assume you do not already have a setup (including column definition and mapping) that will work with the type of database file you are using. These instructions will help you create such a setup. The Itool program distinguishes between a source *data file* and a destination *database*.

Data file: a "flat" file in ASCII format representing a data structure that can be laid out in a single table, where each row is a record and each column is a factor.

Database: a set of interrelated tables connected by common identifiers. Each table has its own factors, and all are keyed to an ID factor. This allows you to add features (such as land types or outbuildings) to a record as often as necessary. See the diagram below.



Operation

We will discuss each tab on the Itool screen in turn, since that is the order in which you will operate the program to effect a data transfer.

1. Setup tab

Assuming you have not already created a setup for this source file and destination database, click on the "Add New Setup" button at the bottom of the screen. This blanks out any values that may have been present. Now give the setup a name, specify the destination database, and indicate the directory in which the program will look for the source file. (Note that Itool is designed to transfer data from one database to another, but if your source file is a flat ASCII file, there should be a dot next to No at the line "Source Is A Database." In that case, nothing can be entered in the "Source Database" line, but the extension of the file (such as ASC) should be typed into the "Source File Extension" window.

As you proceed through the program, you should save your work frequently by clicking the "Save" button at the bottom of the screen. This will ensure that important information, such as mapping instructions, is not lost.

When the destination database has been specified, click on Refresh Destination. This will read the destination table structures into the computer's memory.

If you have specified a directory in which source files of the proper type reside, and if you have indicated the extension(s) of such files in the "Source File Extension" window, then clicking on "Refresh Source" will load into memory the source files available in that folder or directory.

Recalling a previous set of specifications

If you have previously specified a data mapping and want to use it again here, follow the instructions in this section. Otherwise, skip to "*2. Destination Tables*."

Click on "Load Source Specs" to bring into memory information on the particular source file you wish to use in this data transfer. If this file exists, you should see a window headed "Split File – Copy Source Column Specs." Below that, bright yellow letters will say, "For current setup: [setup name] ... with specs from this setup: [window]."

For each source file in this setup, you may recall the mappings used by another source file from another setup. Note that column definitions will not be recalled; only mappings of columns to data fields.

To do this:

1. Select the other setup from "with specs from this setup" (there is a drop-down window where you can find other setups).
2. Highlight the current setup file's source file in the grid below (column "Load this source file with")
3. In the "Specs from this file" column, select the source file from the other setup that you want to use the mappings of.

You can only select "Specs from this file." for those source files in the setup selected in "with specs from this setup." So if you want to load mapping from files in two different setups, do the following: Select one setup, select the specs, click OK on this screen. When you return to the main screen, call this screen up again. Then select the next file.

Any file in the "Load this source file with" column that has an empty "with specs from this setup" list will be left alone. Only when an item in "specs from this file" is selected will the mappings be loaded for your current source file of the current setup.

If the mappings of the source file from the other setup change or you change the mappings for a source file in this setup; the mappings will not be made in both setup's source files. The change only occurs in the source file of the current setup.

2. Destination Table(s)

If you have loaded your destination database structure into memory, this screen will show the tables that make up the database, along with the interrelations among them. You can highlight any table and click on “View Table” to see the current contents of the database. To erase all records in the database, you can click on the “Zap Database” button. However, you do not need to do this now. Just before you proceed to import new data into the database you will have an opportunity to “zap” the records.

3. Source File(s)

On this tab the source file or files are listed. Highlight the one you want to work on. If you click on “view file” the rows and columns making up this file will be displayed.

Under “Supported File Type” make sure the correct type is selected. Most commonly this will be “Fixed,” meaning fixed field length and fixed record length, but it can also be “Delimited,” “Excel,” or “DBF.” Here is a brief set of definitions:

Fixed Files in which each record is the same number of bytes in length, and in which each field begins at the same point as the corresponding field in the next record.

Delimited Files of variable length in which the fields are separated by a delimiter such as a comma or space. Character strings are often enclosed in quotation marks to show that they belong to a single field.

Excel Files saved under Microsoft Excel.

DBF Database files such as those produced by FoxPro (including CAMA 2000) or dBase IV.

Once the file type has been specified (and the delimiter if it is a delimited file), you should indicate the number of columns. Regardless of file type, all records must have the same number of columns. In this case a "column" is not a character or a byte but a field in the flat source file. Specifying the number of columns is the first step toward defining the fields that will be imported into CAMA 2000.

On this page you can also indicate whether the entire source file is to be processed. Unless you are doing a provisional test run, you will probably check "yes" next to the instruction "Go to end to file."

Ignore the "Preprocess" and "Postprocess" buttons, which are not activated in this release of the Itool software.

Click the "Generate Columns" button. If you have performed the previous steps correctly, the system will now create an internal table headed by column designators for each of the columns you indicated. You will fill in starting and ending information in this table to guide the program in extracting data for placement in your new database.

4. Define Source Columns

When you move to this tab you will see the source columns listed in the window at the left. Destination fields are listed in the corresponding window at the right. It is possible to name each column at the left with a field name from the database at the right, but this exercise would be for information only; it would not tell the computer how to import the data.

What the computer needs to know is the position of each column in the source database: where it starts and where it ends. Once it has that information it will be able to extract the data and place each item in the destination field you will indicate. Highlight each column and type the starting and ending positions in the boxes labeled "Beg. Col#" and "End Col#."

Note that there might be confusion here because "Col#" does not mean column in the sense of a data field. It means the byte position in the record. Thus column 15, which might represent square feet, may begin at byte position 126 and end at position 130. These numbers should be entered in "Beg. Col#" and "End Col#" respectively.

You do not need to place any values in the "Type" box, or in the boxes for "Width" and "Decimal." Although there is a provision for repeating columns (i.e., columns in which the same type of information is repeated in a series of column groups), this function is not implemented in the present release and should not be used.

When you have finished defining all the columns, you can proceed to the next tab.

5. Map Data

On this screen the fields in the destination database appear in blue in the leftmost window, while the column names appear in red at the right. Your task is to highlight each destination field that you want to populate and then click on the name of the column in the right-hand window that contains the appropriate data. This is the most critical step in the process.

Note that the factor name in the destination field list at the left is followed by a period and (in caps) the name of the table to which that factor belongs. Be sure that you get the factor belonging to the right table. For example, there may be a field called *Area* in the SECTION table, another in the GARAGE table, another in the PORCH table, and still another in the OUTBUILDING table.

Sometimes a source database will have more than one instance of a group of fields. For example, a single flat file may have fields for porch type 1, porch type 2, porch type 3, porch area 1, porch area 2, porch area 3, and so forth. If these columns repeat in a regular pattern, you can map only the first instance of porch type 1, then check the "Repeat" option, then indicate how many columns must be skipped to get to the next porch type, and how long it should continue. (If the first porch type is column 12 and the next is column 15, indicate that the program should skip 3 columns. If there are three more instances of porch type after the first, indicate that the program should continue for 3 times.)

In the case of repeated values of this kind, you should check the box next to "Don't load this field if value in Source file is empty." Otherwise child table records will be created whether or not there is data to put in them. However, only one field per group should be checked. In the porch example, there may be fields for porch type, porch area, floor, wall, roof, and ceiling. But if porch area is zero, there will not (or should not) be information in the other fields. Therefore, only check the "Don't load" option in connection with the area field. This will ensure that none of the other fields in that data grouping will be imported.

These comments apply to situations where you are populating an empty database from scratch. They also apply to cases where you may be updating an existing database. But in the latter case there is a further consideration. You do not need to reimport the parcel ID into either the MAIN table or the child tables beneath it. Yet you do need to tell the program where to find the parcel ID in the source table, so it can determine where the information in a given record should go. In such cases, set things up as if you were importing the parcel ID but place a check next to "Unique value ID." That will give the program the information it needs to find the correct record, but prevent it from reimporting the field.

If you are importing dates, you must specify the format of the date in the source file. Options are M-D-Y, M/D/Y, Y-M-D, Y/M/D, YYYYMMDD, and YYMMDD. The last is

most commonly used in exports from the MicroSolve DOS program. However, in two instances adjustments may have to be made before the import process will work correctly.

1) Some MicroSolve DOS dates (those based on integer factors) are in only YYMM format. If these are to be imported into CAMA 2000 they must be modified to add a day to each one. Typically 9805 is transformed into 980501 by manipulating the ASCII file.

2) Dates after 991231 are stored in MicroSolve's DOS program as 1000101, etc. That is, the leftmost three digits, when added to 1900, will give the correct year. These dates need to be changed to YYYYMMDD format by manipulating the ASCII file so that 1900 is added to all the "year" values in each field.

In order for the date import function to work, all dates in a given field must begin at exactly the same point and the Itool mapping must indicate the precise starting position of that field. (In the case of numerals it is possible to indicate a starting point to the left of the actual number, and the number will be imported as long as it is between the start and end column indicators; but that will not work with dates.)

A similar restriction applies when you are importing a section number which will indicate the section to which third-level records (roof, garage, heat, etc.) belong. The flat file may have indicated that a particular record was "2 of 3," and the "2" must be imported into the new database to identify the section. The exact position of the "2" must be specified; there cannot be spaces or zeroes before it, or anything after it.

Once you have mapped all the destination factors you wish to populate, you can click the button labeled "View all Dest. fields that are currently mapped." This will eliminate any extraneous fields from the list at the left and let you review each field to confirm that it is mapped to the right column. If any field is incorrectly mapped, click the button labeled "Clear Dest. Map to Source" and remap it.

If you want to see a sample conversion (in memory only), first go back to the Source File(s) tab and indicate how many records you want to process (i.e. place "No" next to "Go to end of file" and indicate a row number to end at), then return to this tab and click "View Sample Conversion." Then (before doing the final conversion), go back to the Source File(s) tab and change the parameters again so that the process *will* go to the end of the file.

Save your setup.

Proceed to the Finish tab.

Finish

If you are populating a database from scratch (even if you have to zap data from a copy of an existing database), you should check (or leave checked) the top option: "Replace highest current record." If you are importing into an existing database but want to overwrite fields in its tables, you should also check this option.

If you are adding data to an existing database but do not want to overwrite what is already there, check the second option: "Create new child record."

As the program populates each table in succession, you may wish to review that table to make sure the import has been performed correctly. If so, check the box "Browse each file as it is completed." If you want to speed up the process or intend to leave the computer unattended while the import is taking place, remove the check mark.

Now click the "Finish" button. You will be asked if you want to proceed with the import, and if you want to zap the existing database. DO NOT ZAP THE DATABASE IF YOU WISH TO PRESERVE ANY DATA.

After the import is completed, you can move the entire database to the appropriate directory and sign onto it through CAMA 2000. Check to make sure the fields you expect to have values really do have them. It is often useful to call up a record in the DOS MicroSolve program and compare it with the same record in CAMA 2000. If there are problems that require changing the mapping and redoing the conversion, you can recall the setup in the Itool program, make the changes, zap the database, and repeat the process.

If there are values that need adjustment, such as a quality value that is 100 times what it should be, these changes can be made via FoxPro if you have that program. MicroSolve's technical staff can give advice on procedures for making these minor corrections.

Notes on the import process

A number of problems can complicate the import process, and it is good to review a checklist before proceeding.

Categorical matches. In a source file categorical factor for Exterior Wall, a level 3 may denote vinyl siding. In the destination file, a 3 may denote stucco. If such discrepancies exist, it is important to discover them before starting the import process. The source file will have to be processed to replace all such values with the new values that will work in the new database.

Dates. Different databases store dates in different ways. It is necessary to determine how dates appear in the source ASCII file and how they will be interpreted by the file you're importing to. Once again, it may be necessary to transform date values before proceeding to the import stage.

File length. When you are importing from a fixed-field, fixed-record-length file, it is essential to know the precise length of the incoming records. If the length is not stated correctly, all records after the first one will be misread and incorrect values placed in the target database. In particular, it is easy to forget the carriage return / line feed combination (two bytes) at the end of ASCII file records. Many standard editing tools do not tell you the exact record length. If you have any doubts about record length, consult MicroSolve technical specialists.

Child tables. Frequently you will be importing data from a "flat file" (a non-relational database) into a set of CAMA 2000 relational tables. This is usually a straightforward process as long as you bear in mind that some fields will go into subtables (land or outbuildings or porches). However, sometimes extra fields have been created in a flat file to give it some of the advantages of a relational database. There may have been more than one set of fields for porch characteristics, for example. In such cases, the second set of characteristics will normally go into a second row of the target porch table, keyed by parcel ID, section ID, and porch ID. Sometimes "phantom" levels get created in these subtables: the first level may have no data, but a second level has data that should have gone to the first. It may then be necessary to delete the first level and renumber any levels above it MicroSolve's technical specialists can assist you when necessary.

Data export

Functional description

MicroSolve's data export feature provides the capability to export data from CAMA 2000 to flat files, Excel spreadsheets, or FoxPro tables.

You may select records to be exported using the existing SET FILTER function, and review those records in a browse screen similar to that of the current Browse/Reset screen. In addition to being able to select columns (in a format similar to the Reset/ Browse) for review and export, you may save the fields chosen for export to a profile list that can be used again, or further modified and saved to facilitate future exports.

The Export function utilizes SQL SELECT statements that are dynamically created, based on the selection criteria. This provides for significantly faster exports than would otherwise be possible.

Exporting records from a CAMA database

Exporting records from a database involves three steps:

1. Determine the population or selection of records that you want (use the SET FILTER (cross-hatch) button located in the toolbar at the top of the screen).

2. Select the fields you wish to export from the tables in the database. Use the folder tabs: the second tab lets you select the fields, the first tab shows you the data you will be exporting. You can save profiles and/or share them with the Browse function.

3. On the left-hand tab, press the Extract button, then indicate the format of output you wish to have the system produce, using the radio buttons.

Since similar data exports are often repeated periodically, the system provides the optional capabilities to store your SET FILTER (population selection criteria) as well as the extract profiles. The saving of the filter is done in the standard Set Filter dialogue.

User maintenance

This feature lets you maintain a list of authorized users with special privileges in CAMA 2000. Click on **Utilities**, then on **User maintenance** to access a screen with two tabs. The Detail tab allows you to enter a user ID, a user name, and a password for that user. If a user is inactive but should still be kept in the file, you can place a check in the "inactive" box.

The List tab provides a summary of information about the users entered on the Detail screen. This screen is read-only. However, any line highlighted here will have its details displayed (and able to be changed) when you click on the Detail tab.

To delete an entire user-information record, highlight the line on the List screen and click on the X in the toolbar.

As more security features are added to CAMA 2000, the user maintenance feature will be expanded and enhanced.

The error log

This feature is used by system technicians to analyze problems that may have occurred in prior operation of CAMA 2000. If errors occur that appear to need technical assistance, copy the error log (\msol\data\error.log.*) and send it to the technician, along with an explanation of just what you were doing in the system when the error occurred.

DBC maintenance

A DBC in FoxPro parlance is a database container: a master file that contains references to all the other files used by the system to constitute a complete database. Running the maintenance procedure on the DBC rebuilds all indexes used by the program, permanently removes records that have been marked for deletion (when you delete a record you do not remove it from the system; you only add a flag instructing the system to ignore it), and identifies any errors or problems that may impede system functioning. Ordinarily you do not need to operate this procedure, but if you have experienced errors, or if you have

received messages indicating that some indexes are defective or missing, it is good to try this step before seeking other technical assistance.

Before running the procedure, make sure you are the only user signed onto the system. The DBC maintenance procedure requires exclusive use of the files.

Click on **Utilities** in the menu bar, then on **DBC maintenance**. As a first step, the program will validate the database, which means it will check the database container against all the tables actually on the disk. If there is a proper match, a window appears saying the database container is valid. Close this window. Then the routine rebuilds all indexes in sequence, removes deleted records, and packs the files (i.e., places all active records together with no gaps between them). A box near the right of the screen tells what indexes are being processed.

If no errors are displayed, you can proceed with your regular work. If you do see an error announcement, you should record it and pass it on to your MicroSolve technical representative.

Verify structures

This utility checks the data dictionary against each of the data tables in the database container. It examines the field names, types, and sizes, and identifies any discrepancies. It may report the following types of errors:

- fields in a table that do not appear in the data dictionary
- fields in the data dictionary that do not appear in the tables
- discrepancies in field lengths between a table and the data dictionary
- discrepancies in field types between a table and the data dictionary

The utility will automatically correct some of these errors. It can add entries to the data dictionary, and it can alter field lengths in the data dictionary. However, it will not add fields to tables or change a field type or type designation. You must make these more radical changes directly, using FoxPro. MicroSolve's technical specialists can make the changes for you, if necessary.

Using the sales history function

MicroSolve's optional Sales History feature allows users to save the parcel currently being viewed with the data display screen to a previously specified sales history database. Saving the parcel will copy the main table record and all supporting records in subsidiary tables for that parcel into the Sales History database, which is accessible to CAMA using the standard SELECT DBC function. The first time a parcel is saved, it retains the original parcel ID. Subsequent saves of the same parcel are given suffixes of A, B, C, etc.

To set up and use the system you must first create, name, and activate the sales history database by running a function called Make New DBC located in the Utilities menu. The computer will ask you to select a source database (the database from which parcels will be sent) and to name the

sales history database. The new database will then be created in a directory of the same name under the DATA folder.

Once the empty database has been created, a checkbox becomes available on the configuration screen, which will allow the SALES HISTORY button to appear on the Data Display screen. The button can be removed and the Sales History capability thereby suspended by returning to the options screen and unchecking this checkbox. The SALES HISTORY button is a good way of storing individual properties to the Sales History file.

Batch load / update activation (deactivation)

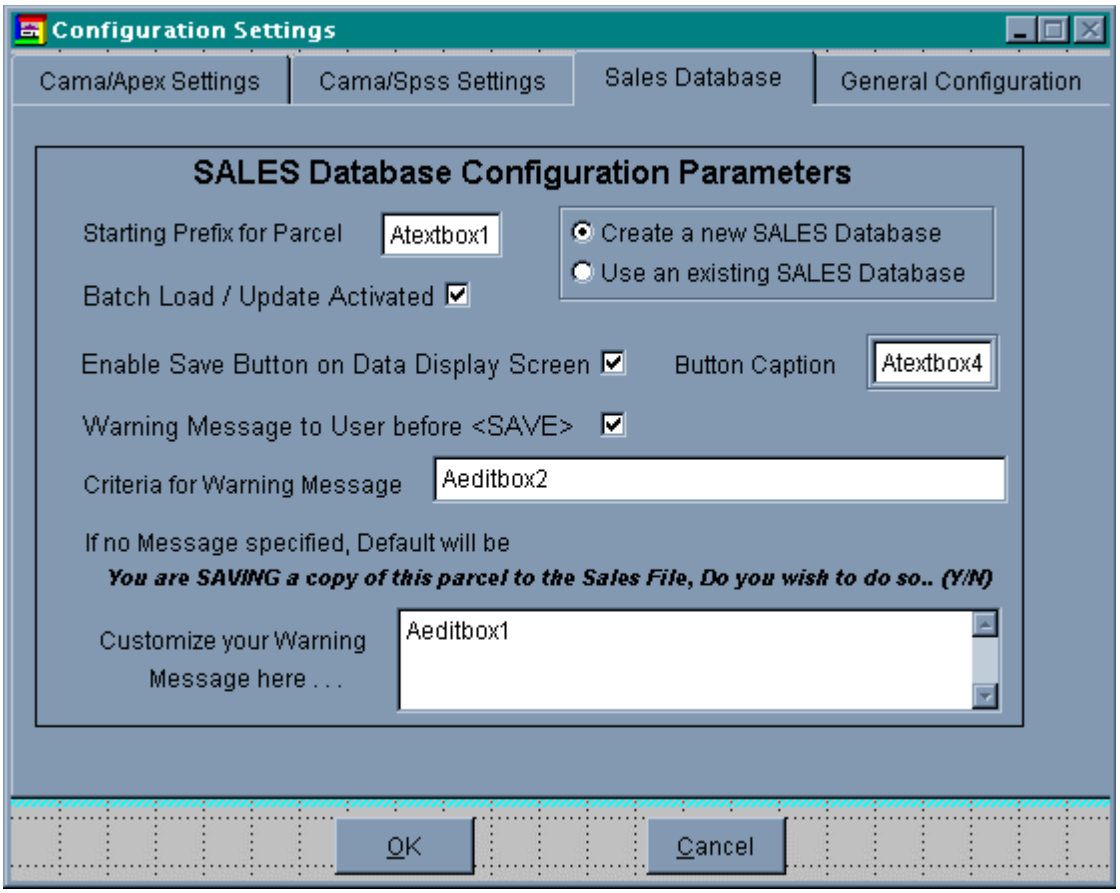
To be able to save a number of properties to the Sales History file, you must check the "Batch Load / Update Activation" checkbox. Activation and deactivation are implemented in the same way as the Save Button (immediately above).

When saving multiple properties in a batch mode to the sales history file, you will first want to set a filter to define the properties to be saved. Select the Export function (Utilities → Export), then set the filter (available in the tool bar at the top of the screen), verify in the browse window that the properties are the ones you wish to transfer, choose the Output Format of "To Sales History," and press Extract.

If any properties are being transferred to sales history that have been previously added to that file, the new copies are given a suffix that is one alpha character greater than the one that has already been saved. This prevents overwriting of previously saved properties. All “saves” of a single property will appear together in Browse/Reset or next to each other as you navigate [◄ ►] through the CAMA sales history database, provided that the index is set to Parcel_ID.

Adding and deleting properties in the sales history database

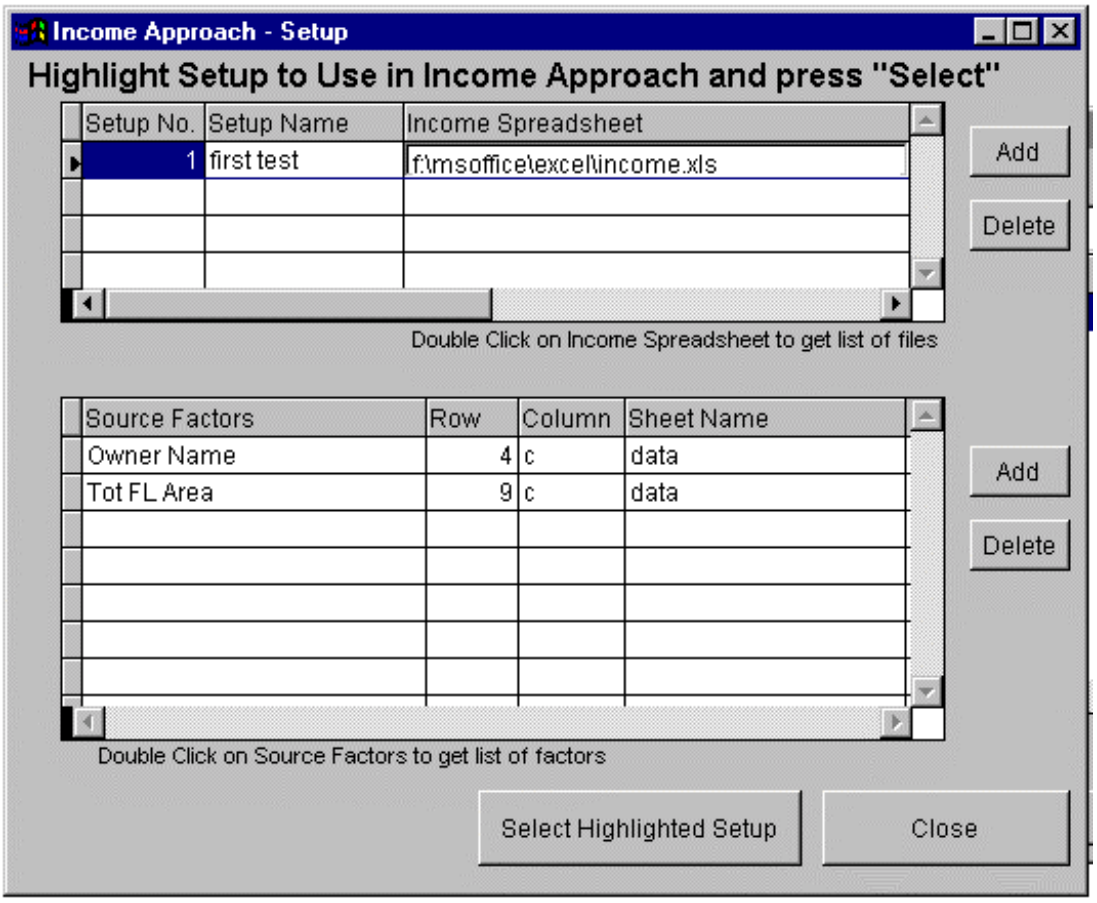
When you have finished copying properties to the sales history file, you may wish to refer to the properties, verify that they have transferred, and possibly cost them for comparison with the property that has been changed in the original database. You can do these things by treating the Sales History database in exactly the same way as you would the original. You can move to the database using Select DBC… and then use the standard capabilities of CAMA 2000.


Configuration Settings
Cama/Apex Settings
Cama/Spss Settings
Sales Database
General Configuration
SALES Database Configuration Parameters
Starting Prefix for Parcel
Atextbox1
Create a new SALES Database
Use an existing SALES Database
Batch Load / Update Activated
Enable Save Button on Data Display Screen
Button Caption
Atextbox4
Warning Message to User before <SAVE>
Criteria for Warning Message
Aeditbox2
If no Message specified, Default will be
You are SAVING a copy of this parcel to the Sales File, Do you wish to do so.. (Y/N)
Customize your Warning Message here . . .
Aeditbox1
OK
Cancel

12 ■ INCOME APPROACH

The income approach in CAMA 2000 can be as simple or as complicated as you want because it is based on a spreadsheet that you develop yourself to connect income data, expense data, tax data, occupancy rates, capitalization rates, and other relevant information, producing an estimated value by means of either a direct capitalization or a discounted cash flow procedure. Each record in your CAMA 2000 database is presumed to contain raw information about the commercial property, including the data from an income and expense questionnaire. The spreadsheet, on the other hand, contains the formulas that permit calculation of a final value from these data elements.

To operate the CAMA 2000 income approach, choose Valuation from the menu bar, then click on Income Approach in the dropdown menu. You will see a window with a pick list of properties, very similar to the one that appears when you operate the cost approach. At the bottom near the left is a button labeled “Setup.” Click on this button and the following screen appears.



The top part of the window is used to identify spreadsheets you may want to use in the income application. Each one is given a number and a name. You must specify the path in which the spreadsheet file (an Excel file) resides. Click the “Add” button to create a line in which these details can be placed.

To indicate where in a particular spreadsheet your data elements should be sent, go to the lower window. In the first column specify the name of the source factor (you can pick this from a list brought up by double clicking on “source factors”), then indicate the row, column, and sheet name to which this factor will be copied. (It is helpful to study the spreadsheet prior to performing this step so that you can note the address of each data element.) Again, you can use the “Add” and “Delete” buttons to create the setup you want.

Closing the window saves your setup automatically.

To run a setup that has been saved, click on the “Setup” button and highlight a setup in the upper window. Then click the button that says “Select Highlighted Setup.” Then click “Close.” Now pick a record from the available list of properties and then click on “Run Income Approach.” This action opens the spreadsheet and stocks it with the relevant data from the record you selected. You can then move around in the spreadsheet, examining the results and making any further adjustments you wish. You can also print any or all pages. Finally, when you quit the Excel program, you will return to the CAMA 2000 income approach, where you may run another record or move on to another operation.

If you wish to keep a spreadsheet file related to a particular record, you should save it under another name, since the file named in the income setup window will be overwritten with new data each time it is used to process an individual record.

13 ■ INTEGRATING SKETCHES

Property sketches in CAMA 2000 are produced by the Apex software, which is incorporated into the program suite. Detailed instructions for drawing sketches are contained in a separate manual published by Apex. This chapter describes the procedures for ensuring that Apex is properly invoked from the CAMA data entry screen and that the calculated areas of each portion of a sketch will be recorded in the relevant factor of the CAMA 2000 database.

Taking advantage of Windows 32-bit architecture, Apex calculates areas for defined areas within sketches, displays that information, and stores it as part of the sketch. CAMA 2000 accepts this area information and automatically stores it in indicated fields. If records already exist for the areas being created in the sketch (Building Square Feet, Porch Area, Garage Area, etc.), the area information will, at your discretion, be saved into the appropriate records in the CAMA database when the sketch is saved. In the event that you choose to draw areas for new structures (e.g. a new porch) not previously defined in the CAMA database, the appropriate record(s) will be dynamically created within CAMA when the sketch is saved.

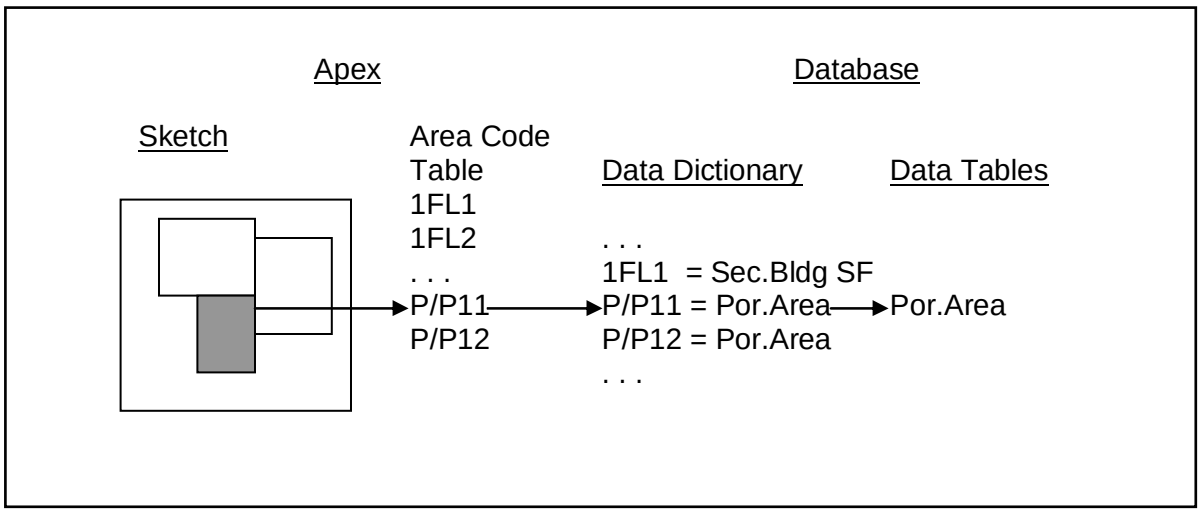
Note: While defining of areas within a sketch can force creation of CAMA records in which to store that information, it does not follow that CAMA records representing areas or structures not defined in the sketch will be updated or deleted. For example, adding a porch to a sketch will add a porch record to the database; however, deleting a porch from the sketch will not remove a porch record from the database. You must make and execute these decisions by conventional methods within CAMA.

You can continue to make changes to CAMA records, including area information, directly from the Data Display screen. Changes made in this way will not affect area sizes represented on the sketch.

How Your Database Shares Information with the Sketch Routine

CAMA 2000 “knows” which fields are to store area information because of the set-up information you provide when you use the Data Dictionary to define the fields and tables manually, or because the list is provided when you run the conversion utility called CONVSKET. Sketches drawn using the AXWDDA1.DAT area code table, which MicroSolve installs as part of the file conversion process, complete the link between the two types of data.

A simple scheme shows how the areas you select in drawing the sketch are linked with the data dictionary and thus with the factors in which the calculated square-foot values will be stored.



Connecting Apex Areas with CAMA 2000 Factors

First, some nomenclature. In this context, "area" is ambiguous, since it can mean either a defined region of a building or land tract, or the calculated square footage associated with that region. In this section, the word "area" refers to a defined region or space, and the term "SF" refers to the calculated square footage (length X width) of such a space.

If more than one area is designated with the same label, SF will always accumulate. For example, if two distinct parts of a sketch are labelled 1FL1 (see below), the total SF from both areas will be recorded in the appropriate factor. (Ditto perimeters, if they are used.) If you don't want to accumulate SF values, you must assign a different label to each area.

The database has both second- and third-level tables, so the following conventions are used. Labels applying to second-level tables (land, section, outbuilding) are up to four characters long. The first characters are descriptive. The last are numeric, indicating the record in the relevant table. Thus the gross building area label for a one-story house (section 1) is 1FL1. A second-floor area in section 2 would be 2FL2, but more likely you would sketch both areas together and label them 2FL5, which denotes first- and second-floor areas in section (building) 2. (Apex automatically applies a multiplier of 2 when the area applies to first-and-second-floor SF.)

Labels applying to third-level tables (garage, porch) are up to five characters long. Again, the first characters will be descriptive (GAR or P/P) and, of the last two, the first character will indicate the section ID, and the second will specify the ID in the third-level table. Thus the first porch attached to building section 2 would get the label P/P21; whereas the third porch attached to building section 1 would be labelled P/P13.

You are responsible for understanding this scheme and selecting the appropriate label. The program will parse the label correctly and send the SF value to the correct record.

Multipliers

Apex allows multipliers to be associated with defined area types. These are factors that are automatically multiplied by the calculated SF before it is recorded in the CAMA 2000 database. For example, if you draw the footprint of a two-story house and label it with an area code of 1FL5 (section 1, first and second floor), the calculated SF will be multiplied by 2, and this result will be placed in the Bldg SF factor of CAMA 2000.

Remember that Marshall & Swift distinguishes two kinds of 1½ story houses: finished and unfinished. Unfinished 1½ story houses are valued on the basis of the ground-floor SF only, but the rates used are somewhat higher than for the equivalent SF in a one- or two-story house. Finished houses are priced on the basis of the first-floor SF plus the usable SF of the second floor. We included an automatic multiplier of 1.6 for such area types. If you draw a first-floor area and label it 1FL4 (1½ floor finished), a SF value 1.6 times the calculated SF will be placed in the Bldg SF factor of CAMA 2000. Likewise, finished 2½ story houses are given a multiplier of 2.6.

Multipliers for all the area types are shown in the following chart.

Area	Subarea	Description	Mult	Used in M&S SF method
GBA		Gross Building Area		
	1FL1	S1 – 1st Floor	1	✔
	1FL2	S1 – 2nd Floor	1	
	1FL3	S1 – 3rd Floor	1	
	1FL4	S1 – 1½ Fl Fin	1.6	✔
	1FL5	S1 – 1 + 2 Floor	2	✔
	1FL6	S1 – 2 ½ Fl Fin	2.6	✔
	1FL7	S1 – 1 + 2 + 3 Floor	3	✔
	2FL1	S2 – 1st Floor	1	✔
	2FL2	S2 – 2nd Floor	1	
	2FL3	S2 – 3rd Floor	1	
	2FL4	S2 – 1½ Fl Fin	1.6	✔
	2FL5	S2 – 1 + 2 Floor	2	✔
	2FL6	S2 – 2 ½ Fl Fin	2.6	✔
	2FL7	S2 – 1 + 2 + 3 Floor	3	✔
BSMT		Basement		
	1BS	S1 – Basement	1	✔
	2BS	S2 – Basement	1	✔
	1BF	S1 – Finished Bsmt	1	✔
	2BF	S2 – Finished Bsmt	1	✔

Area	Subarea	Description	Mult	Used in M&S SF method
P/P		Porch/Patio		
	P/P11	S1 – Porch 1	1	✔
	P/P12	S1 – Porch 2	1	✔
	P/P13	S1 – Porch 3	1	✔
	P/P21	S1 – Porch 1	1	✔
	P/P22	S2 – Porch 2	1	✔
	P/P23	S3 – Porch 3	1	✔
GAR		Garage/Carport		
	GAR11	S1 – Garage 1	1	✔
	GAR12	S1 – Garage 2	1	✔
	GAR13	S1 – Garage 3	1	✔
	GAR21	S2 – Garage 1	1	✔
	GAR22	S2 – Garage 2	1	✔
	GAR23	S2 – Garage 3	1	✔
LAND		Land	1	
SITE		Site Plan	1	
OTH		Other	1	
NCA		Non-Calculated Area	1	
DTG		Detached Garage		
	1DG	1st Det Garage	1	✔
	2DG	2nd Det Garage	1	✔
	3DG	3rd Det Garage	1	✔

Creating, Displaying, and Changing Sketches from CAMA

A single button labelled <SKETCH> on the data entry form allows you to create, update, and display sketches from within CAMA 2000. You can create a new sketch (by pressing the <New> button on the following screen) or import a previously defined sketch (by locating and clicking on it). The existence of a sketch is denoted by bold type on the sketch button.

Once Apex opens and you have a drawing screen in front of you, you must define the area you intend to draw. Go to DEFINE AREA, pick an area, and click on it. You will see how the areas are defined. For Porches, P/P11, P/P12, etc. will appear. These new codes were added to Apex during the installation process (above). Go ahead and add an area, then close the sketch, using the [x] in the upper right corner. Apex will ask if you wish to save. You should answer Yes without changing the specified location. The sketch will automatically save into the record from which you invoked Apex, and you are returned to that record. The factors linked to the areas of your sketch are updated only when you move to the next record or press the <SAVE> button (the diskette icon in the toolbar near the top of the screen).

Printing Sketches

Within Apex a sketch can be printed directly from the display. Anywhere else, including the rest of CAMA 2000, a picture file in JPG format is required to display or print a sketch. The system creates the JPG files “on the fly,” while a report is being printed, eliminating the need to store JPGs permanently on disk. However, should you wish to store JPGs anyway, so that they can be used independently of CAMA 2000, you can do so by entering the desired location on the configuration screen and, in Apex, checking the Auto-save Image box on the “Save As ...” screen. In that case you should also set the Apex options to auto-save to the directory of your choice.

Deleting Sketches

Sketches are deleted from within Apex. While viewing a sketch, draw a square encircling all visible components of the sketch (Microsoft calls this “lassoing”). Press the <Delete> key on your keyboard and close the Apex screen. Now either <Save> from the toolbar or simply move to the next record.