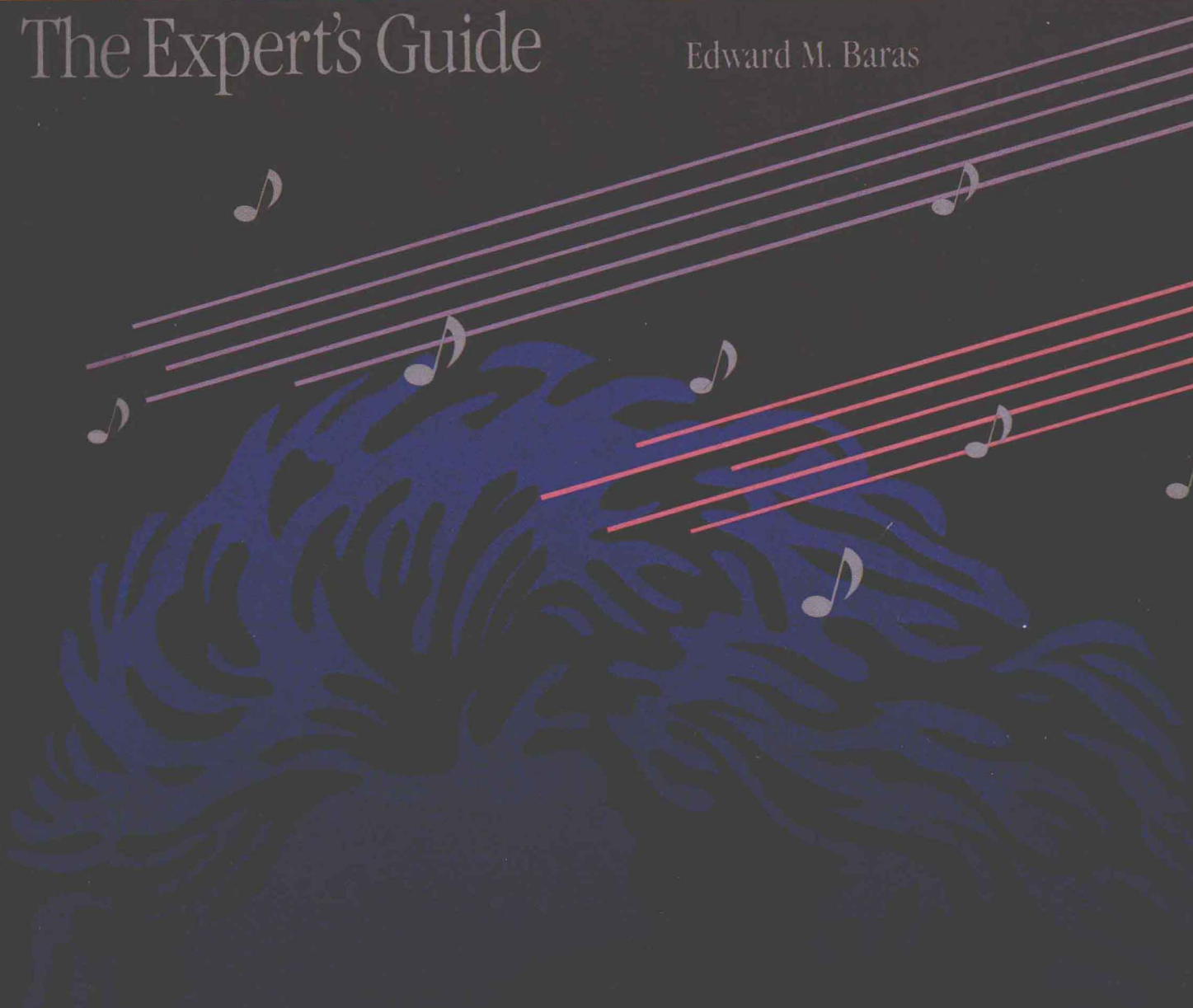
Osborne/McGraw-Hill

Symphony™ Master

The Expert's Guide

Edward M. Baras



Symphony™ Master:
The Expert's Guide

Edward M. Baras

Osborne **McGraw-Hill**
Berkeley, California

Published by
Osborne **McGraw-Hill**
2600 Tenth Street
Berkeley, California 94710
U.S.A.

For information on translations and book distributors outside of the U.S.A., please write to Osborne **McGraw-Hill** at the above address.

SYMPHONY™ MASTER

1234567890 SLSL 898765

ISBN 0-07-881170-8

Cindy Hudson, Acquisitions Editor
Jean Stein, Technical Editor
Steven E. Miller, Technical Reviewer
Fran Haselsteiner, Copy Editor
Melinda Breitmeyer, Composition
Yashi Okita, Cover Design

Symphony™ Master:
The Expert's Guide

For my parents

Acknowledgments

I acknowledge gratefully the assistance and advice of the following individuals:

- Steve Miller of Lotus Development Corporation for reviewing the manuscript and contributing greatly to its improvement and accuracy;
- Jean Stein and Cindy Hudson of Osborne/McGraw-Hill for the diligence, patience, and encouragement that are the trademarks of great editors;
- Sondra Oster Baras, whose talents—editorial, intellectual, and culinary—provided food for thought, thought, and food during the preparation of the manuscript;
- Ron Baras for his assistance on the communications chapters.

Preface

This book will transform you from a Symphony user into a Symphony master.

Symphony's five integrated environments—spreadsheet, database, word processing, graphics, and communications—offer an impressive array of capabilities. Knowing the commands and features of each environment allows you to put Symphony to work for you. But learning these essential elements is only the beginning.

Beyond this familiarity with Symphony's five environments lies another, higher level of knowledge. This is the level of the Symphony master. *Symphony Master: The Expert's Guide* develops and expands your expertise so that you can take full advantage of Symphony's capabilities. This book teaches the advanced methods and techniques which make Symphony one of the most powerful and versatile application development tools available.

The best way to learn these techniques is through practical experimentation. That is why this book presents this material within the context of practical applications that you can implement. Each application encompasses and illustrates a particular group of commands and concepts. While you can follow the material strictly by reading, you will get the most out of the book by working through the examples step-by-step on your own computer.

This book begins by revealing the backbone of Symphony mastery—macro programming—and by teaching the Symphony command language. Macros allow you to build automated systems that can be customized for specific tasks. A simple macro might automatically execute a sequence of keystrokes that you would ordinarily enter keystroke-by-keystroke. Chapters 1 through 3 discuss such keystroke macros and introduce learn mode, a method of recording keystrokes into a macro as you press them. You will use learn mode in Chapter 2 to create a word processing utility that erases paragraphs, and in Chapter 3 to automate a price-quote system.

The Symphony command language is the programming language of macros. The command language macros give you capabilities unavailable in the commands of the five environments. Chapters 4 and 5 cover the major elements of the command language. These chapters lay the foundation for the financial forecasting applications presented in Chapter 6. Macro subroutines, the building blocks from which module systems are created, are illustrated through this application.

You can use macros to create customized systems, or *templates*, that make Symphony look as if it were created to suit a specific application. You can even design your own menus, which will function just like Symphony's command menus. To demonstrate user-defined menus, Chapter 7 guides you through the development of two menu-driven systems. One is a word processing utility that lets you incorporate special print attributes—such as boldface and italics—into printed text. The second menu-driven system is a utility that synthesizes the selection of special printer features (like condensed print) for Symphony printouts.

Chapters 8 through 17 demonstrate several advanced techniques that are important to the expert user. Windows and windowing techniques are the subjects of Chapter 8. File combination is the focal point of Chapters 9 and 10, which use a business consolidation and a stock portfolio tracking system to illustrate the commands discussed. A marketing survey demonstrates the use of database selection

criteria, database statistics, and "what-if" tables in Chapters 11 and 12. Then, in Chapters 13 and 14 you will learn how to alter and customize the appearance of database entry forms in the FORM environment, using an invoicing system to print information onto preprinted forms.

The last three chapters of the book are dedicated to communications. Chapter 15 covers the fundamentals of telecommunications using Symphony. Chapters 15 and 16 exemplify Symphony's communications features by using macros to create an automated file exchange system between two microcomputers operating under Symphony. The concepts covered in these two chapters also relate to communications between the Symphony user and a large computer, another microcomputer, or an information service. Chapter 17 discusses methods for converting text files to Symphony's worksheet format. The information presented in this chapter will also assist you in exchanging data between Symphony and other software packages.

Experience builds expertise. Working through these 17 chapters will familiarize you with the commands, capabilities, and methodologies of advanced Symphony use. Equally important is the experience you will gain in creating functionally efficient Symphony systems. This knowledge and experience will transform you into a *Symphony Master*.

Contents

Chapter 1

Using Macros

As an experienced Symphony user, you are already acquainted with Symphony's spreadsheet, word processing, database management, graphics, and telecommunications capabilities. Mastering the commands and menus of each of these environments represents one level of proficiency with the program, but there is much more.

Symphony has capabilities that not only give you greater control over the program, but allow you to customize Symphony applications that are so easy to operate that even inexperienced users can work them. Such is the power of macros.

A *macro* is a set of stored keystrokes and commands that Symphony executes automatically when you invoke it. Macros are as easy to use as they are powerful. This chapter, Chapter 2, and Chapter 3 introduce macros: how they are created, invoked, used, and altered. These chapters also provide some examples of macros you can use in your own Symphony applications, and they provide a foundation for using the Symphony *command language*, which combines decision-making abilities with the keystrokes and commands of macros to allow you to create your own Symphony programs.

What Macros Can Do

What is the advantage of storing commands and keystrokes in macros? The benefit lies in the ability to execute and re-execute a procedure whenever you like with a single keystroke.

Many applications involve repetitive procedures. Suppose, for example, you are word processing and a particular phrase, "John Doe, Inc.", occurs many times in your document. Rather than typing the words each time, you could create a macro. Then whenever you reach a point in the document where "John Doe, Inc." should appear, you could press a single key to cause the macro to type the words for you. In addition to saving you a great many keystrokes, the macro never makes a typing error. And unless you type with amazing speed, the macro will save you time, because macros are extremely fast.

You could use a keystroke macro in a similar way to make data entries in a database. For instance, your personnel records may contain a field that stores the name of the division in which each employee works. If one of the divisions were John Doe, Inc., then a macro could be used to enter it for every employee who worked in that division.

Macros do more than type data, however. You can use a macro to enter virtually any combination of keystrokes that you would routinely use in a Symphony session, including the keystrokes that you use to issue Symphony commands. Thus, you could use a macro to input frequently used sequences of commands or a mixture of commands and data entries.

This feature has many possibilities. If you were to use a spreadsheet named INVENTORY to keep track of the inventory of your firm, you would probably retrieve and update the file on a daily basis. After each use of the file, you would resave it with the *File Save command.* Instead of having to invoke the Services menu, issue the command, indicate the file name, and affirm that you want to update the existing file, you could simply invoke a macro designed to perform the routine for you. You could also develop an automatic procedure to create a format line in a document, type a particular phone number in the communications package, bring a named graph window to the screen, or print a series of reports.

In a sense, you can use macros to customize Symphony commands to your needs

by combining existing Symphony commands into a procedure you can invoke with a single keystroke. Symphony includes a wealth of commands, but there is always a task or two that requires multiple commands. For example, you might have created a spreadsheet that must be revised by erasing every fourth cell in a given column. Not surprisingly, Symphony does not include a single command to erase every fourth cell in a column. However, by using the macro capability, you can store a combination of commands and keystrokes, thereby creating your own erase-every-fourth-cell command.

The macros that have been discussed so far are called *keystroke macros* because they are nothing more than combinations of valid Symphony keystrokes that you could have pressed yourself without storing them in a macro.

Symphony macros can perform feats even greater than the duplication of keystrokes. The Symphony command language contains special commands that let you program Symphony and, in effect, put you in control of the way a macro functions: you can create your own menus, instruct a macro to execute only under certain conditions, execute a macro repeatedly, and perform many other tasks that would not be possible with keystroke macros. Chapter 4 will describe the Symphony command language in greater depth.

How Macros Work

After considering the sophisticated tasks that macros can perform, you might assume they are hard to develop. They aren't.

The first step in creating a macro is to decide what you want the macro to do and what keystrokes and commands you want it to contain. The next step is to record the macro's contents as a label cell in the spreadsheet. If the macro is more than a few keystrokes, it must be stored in multiple label cells stacked one after another in a vertical range.

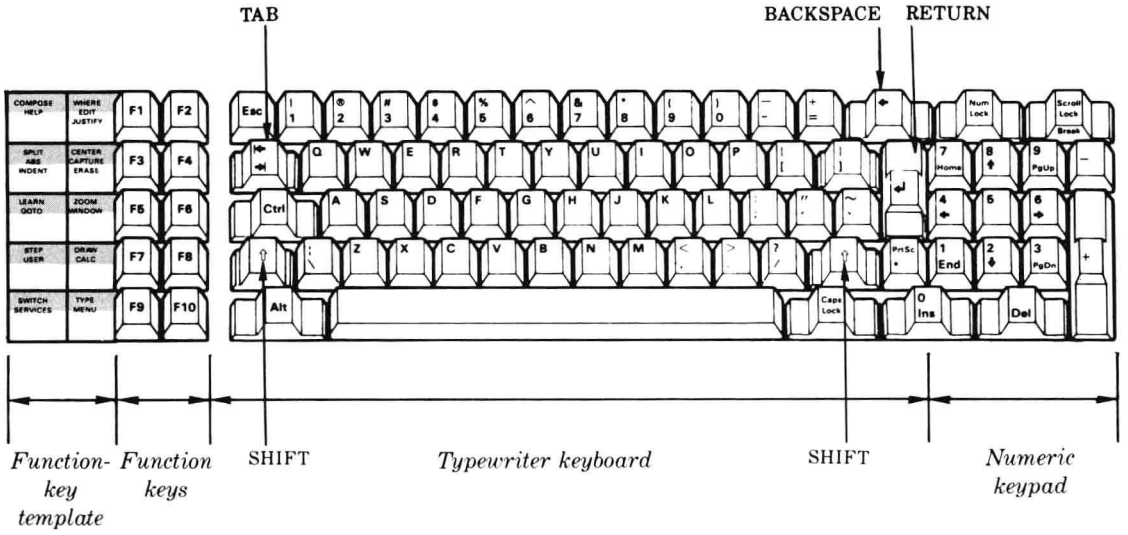
Merely entering these label cells in the spreadsheet does not conclude the task of creating a macro. You must give the range containing the macro a name. Later, when you want Symphony to execute the macro, you will invoke it by using the range name you assigned to it; Symphony will know which macro you want. (Remember, there may be several macros in your worksheet.)

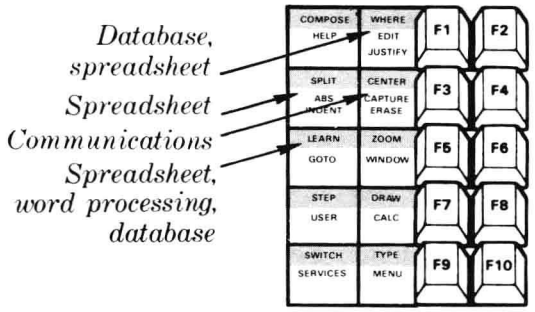
In simple terms, developing a macro requires storing its contents as a vertical range of label cells and then naming the range so you can invoke it whenever you want.

When you invoke the macro by name, Symphony reads the contents of the range of label cells and begins executing the keystrokes stored in that cell. If there are label cells below the first cell of the macro, Symphony continues reading and executing them. This process continues until Symphony finds no more label cells in the vertical range, or until Symphony has been instructed to stop reading the contents of the macro.

You already have enough information at this point to create a simple macro. Before you do, however, you should take a moment now to read "Keyboard Notations Used in This Book" (see box).

Keyboard Notations Used in This Book





Shaded areas indicate that you hold down ALT and press the key.

The illustration shows the IBM standard keyboard, labeled to show several special keys used by Symphony.

This book will use small capital letters whenever it refers to these special keys. For example, the "enter" or "return" key (the key marked ↵ on the right side of the keyboard) is represented in this book as the RETURN key. The four directional arrow keys (←, →, ↑, and ↓ on the far right of the keyboard) are described in this book as the left, right, up, and down arrow keys. The key marked ↹ on the keyboard is referred to as the TAB key, and ← is the BACKSPACE key. Other special key names will also be printed in small capital letters. For example, HOME is the key marked "Home" on the IBM PC keyboard, and PG UP is the "Pg Up" (page up) key.

Creating Your First Keystroke Macro

The first macro you will create in this chapter is a simple procedure to type the words "John Doe, Inc." The keystrokes that must be stored in the macro are only those of the phrase itself. You will store the macro in cell A1. This means that you cannot use this cell for any other purposes.

To begin,

1. Load Symphony and make sure the cell pointer is in cell A1 of the spreadsheet.

(Throughout this book, procedures that you should follow on your own keyboard will be presented in numbered steps, and keystrokes that you should type will be highlighted in **boldface** print.)

Entering a label cell that contains the macro's keystrokes is simply a matter of typing the keystrokes of the phrase as a cell entry.

2. Type **John Doe, Inc.** and press RETURN.

The keystrokes are now stored in cell A1, ready for Symphony to read and execute. In order to invoke the macro, though, you must first assign it a name.

Naming and Invoking Macros

To assign a name to the macro, you use the spreadsheet menu's *Range Name Create command.* An explanation of naming ranges is contained in "About Range Names" (see box).

Special rules apply to the range names permitted for macros. The names you are allowed to use depend on the way in which you will invoke the macro.

There are three methods of invoking macros. The first method begins with pressing the USER (F7) key. Symphony displays a "User" indicator in the bottom right corner of the screen and waits for you to type a range name that has been assigned to the first cell of the macro. (Actually, you can assign the range to all or some of the macro's cells, as long as the top cell address of the range name's assigned range is the first cell of the macro.) In this case, the range name can be any valid range name of 15 characters or less. After pressing the USER (F7) key, you type the range name and press RETURN. Symphony then begins executing the macro. Note that invoking the macro by name not only tells Symphony which macro to execute; it also tells Symphony where to look for the macro, because every range name has an address assigned to it.

A second method requires that you use special names for your macros. With this method, your macro's name must contain exactly two characters. The first character must be a backward slash (\). The second character must be a letter of the alphabet. Such range names as \A, \J, and \Z are valid macro names.

What makes this method special is that you invoke the macro by pressing the MACRO (ALT) key and then pressing the letter of the macro name. To invoke a macro named \A, you would press the MACRO (ALT) key and press the letter A. On an IBM PC, you hold down the ALT key and simultaneously press the letter key; some other computers require you to press and release the MACRO key before typ-

About Range Names

Symphony allows you to assign a name to a range of cells. When it becomes necessary to refer to a range in a command like Copy or Erase, you can simply substitute the range's name for the range's beginning and ending coordinates.

A range name may contain from 1 to 15 characters or numbers. Avoid names that may be confused with cell references, such as A15 or IC256. Also refrain from using arithmetic operator symbols such as the hyphen (which Symphony translates as a minus sign). Names serve as cues, so choose a name that will help you remember the contents of the range it describes.

To assign a range name, you must be in a SHEET window. First move the pointer to the first cell of the range you are about to name (this is not mandatory, but it makes it easier later). Then press the MENU (F10) key and select Range Name Create. The display panel will display a prompt for a range name as well as a list of any existing range names that you may have created previously. To create a new range name, type the name and press RETURN.

Next, Symphony needs to know what range to assign the name to. It assumes that the current position of the pointer is the beginning of the range. (That is why you should move to that position before invoking the Range command.) To name a single cell as a range, you would press RETURN at this point, concluding the name assignment. If you want to assign a range of more than one cell (such as a rectangular range), you may press the . key to anchor the range and use the pointer movement keys to expand the pointer over a range, as you do with other commands (such as the Copy command) that involve ranges. Alternatively, you may type the range coordinates explicitly, in which case you must type the beginning address (even if it is displayed in the control panel already), press the . key, and type the end address.

ing the letter. The instructions used in this book will refer to the IBM PC's ALT key rather than the MACRO key.

This method of invoking a macro is far simpler than the first method. Instead of pressing USER (F7), then typing the full range name (which can be up to 15 characters), and then pressing RETURN, this second method requires only one keystroke sequence: the ALT key and the letter name of the macro.

In most of this book, you will be using the second method because it requires fewer keystrokes. The first method is useful when a descriptive macro name is more important to you than the convenience of reducing keystrokes.

As mentioned earlier, there is a third method of invoking macros. With this method, your macro names must begin with a backward slash (\) followed by a number from 1 to 10. Names such as \1 and \10 are valid. To invoke the macro, you press USER (F7) and then press the function key corresponding to the number contained in the macro's name. To invoke a macro named \10, you would press and release the USER (F7) key and then press function key 10 (F10).

Although this method requires only two keystrokes, it can add confusion because it makes use of the function keys, which are already frequently used in Symphony to perform other tasks. Therefore, this third method will not be used in this book.

Let's assign the name \A to the macro in cell A1. With the pointer still on this cell,

1. Press MENU (F10), select Range Name Create, type **\A**, and press RETURN twice. (It makes no difference whether you use an upper- or lowercase A in the range name.)

The macro is now ready for use.

Invoking Macro \A

One use for macro \A is in word processing. To demonstrate, let's change to the DOC environment and invoke the macro while typing a sentence.

1. Change the window type by pressing the TYPE key sequence (hold down ALT and press F10). Select DOC from the menu.

You will be typing the sentence "There is no company with more personality than John Doe, Inc."

2. Press the down arrow key twice to move to line 3 so as not to interfere with the macro cell.
3. Type the following phrase:

```
There is no company with more personality than
```

4. Press the space bar.

Instead of typing "John Doe, Inc." at this point, invoke the macro that types this phrase for you. When you invoke macro \A, notice how quickly the macro performs its function.

5. Hold down the ALT key and type **A**.

Symphony has typed the phrase exactly as you would have, but much faster. (If the phrase is misspelled, there is a typing error in your macro. You can fix it by retyping or editing the macro label stored in cell A1.) After typing the last letter of the phrase, Symphony leaves the cursor on the next character space (character 62).

You can use this type of macro in other contexts as well. For example, you could use it to enter a label cell in the spreadsheet. You could also use it in the middle of a command.

Invoking a Macro in the Middle Of a Command

Let's switch gears for a moment. Suppose you wanted to create a range whose name just happened to be John Doe, Inc. "John Doe, Inc." is a valid range name, after all. To assign this name to a cell, you would use the Range Name Create command. You can use your \A macro to assist you in entering the command.

First, return to the SHEET window, because range names can only be assigned in a spreadsheet.

1. Press the TYPE (ALT+F10) keystroke sequence. Select SHEET from the menu.

To assign the range name "John Doe, Inc." to cell A3,

2. Move the pointer to cell A3.
3. Press MENU (F10); select Range Name Create.

Symphony is waiting for you to type the range name. Instead of typing it yourself,

4. Invoke macro \A (hold down ALT and type **A**).
5. Press RETURN to conclude the range name entry.
6. Press RETURN again to conclude the Range command.

Now you have a range named JOHN DOE, INC. To check, you can use the GOTO (F5) key to move the pointer to the cell named JOHN DOE, INC. If the range name is valid, GOTO will work without error.

7. Move to cell C1, so that you can observe the effect of pressing GOTO.
8. Press the GOTO (F5) key.

At this point, Symphony asks you to supply the address to go to. You could type the range name, but instead, let macro \A enter the range name for you.

9. Hold down ALT and type **A**.
10. Press RETURN.

The pointer moves to cell A3, the cell named JOHN DOE, INC. Not only have you verified that the range name is in effect, but you also have demonstrated that a macro can be invoked in the middle of a command.

Another Use for the \A Macro

It was mentioned earlier that a keystroke macro such as \A can be used in the FORM environment to enter data into a record. In the example of a personnel database with a division field, where many of the employees came from the John Doe, Inc. division, you could use your macro to reduce typing time.

If you have used Symphony for managing a database—or if you have read *The Symphony Book* (Edward M. Baras, Osborne/McGraw-Hill, 1985)—you may already know that you can place a default value in a database's definition range. Symphony automatically displays the default value in the entry form it uses for inserting new records into a database. You can override this default if you want; otherwise, Symphony enters it for you. If most employees are from the John Doe, Inc. division, you would be correct to question why you should use a macro like \A when you could simply create a default field within the form itself.

You don't have to change the scenario very much to find an important use for your "automatic-typing" macro. Suppose the corporation was made up of four divisions, each one with hundreds of employees. In this situation, you might save a lot of time by creating four automatic-typing macros, one for each division. To fill