OBJECTVISION™

GETTING STARTED

ObjectVision™

Getting Started

BORLAND INTERNATIONAL, INC. 1800 GREEN HILLS ROAD
P.O. BOX 660001, SCOTTS VALLEY, CA 95067-0001

PRINTED IN THE U.S.A.
10 9 8 7 6 5 4 3 2 1

R1

C O N T E N T S

T A B L E S

F I G U R E S

INTRODUCTION

This book, *Getting Started*, is the first in a set of two manuals. It's designed to introduce you to the major features of the ObjectVision application design environment. *Getting Started* focuses on the ObjectVision features you'll use in the tutorial. There are several ways to use *Getting Started*; see "How to use this book" on page 2.

The second manual, the *ObjectVision Reference Guide*, is the major source of detailed information about all of ObjectVision's features. Users of version 1.0 might want to start by reading Appendix A, "Summary of 2.0 changes," which briefly describes the changes in version 2.0.

What's in this book

The chapters in this book correspond to tasks required to create an ObjectVision application.

- **Chapter 1, "Installing ObjectVision,"** explains how to install ObjectVision and start the application.
- **Chapter 2, "Basic features,"** introduces opening, saving, and printing ObjectVision applications. Keyboard operations are also listed as an alternative to using a mouse.
- **Chapter 3, "Using an ObjectVision application,"** tells how to start the sample *Order* application provided on the ObjectVision disk. As you use the application, you'll also see how it was developed.
- **Chapter 4, "Designing forms,"** explains the process of planning the application and laying out ObjectVision forms by positioning objects. You can follow the instructions to create an exact copy of the sample application named *Order*, or you can modify the instructions to fit your own application.

Chapters 5-7 explain how to assign properties to the objects you created in Chapter 4.

- **Chapter 5**, "**Defining properties**," tells how to define properties for fields, forms, and the application. Properties let you control the display format (for example, text entry field or check boxes) and appearance (for example, color, font, and border). Two special properties, value trees and event trees, are covered in the following two chapters.
- **Chapter 6**, "**Creating value trees**," explains how to instruct the application to calculate values for fields. Calculated values can be based on constant values, values returned by @functions, or values in other fields.
- **Chapter 7**, "**Creating event trees**," explains how to make the application carry out an action based on a user action like a mouse click.
- **Chapter 8**, "**Creating links**," shows how to link the values in the application to existing external data files. The sample application, *Order*, is linked to four different Paradox tables.

Appendix A is the quick tutorial.

- **Appendix A**, "**The short course**," contains a condensed version of the tutorial. See the next section for details.
- **Appendix B**, "**Configuring the Paradox Engine**," explains how to use the Paradox Engine Configuration Utility to change settings for the Paradox Engine, which is used whenever your applications work with Paradox tables.
- **Glossary** gives definitions of key words and concepts used in this manual and in ObjectVision.

How to use this book

This tutorial is designed to be completed in any of several ways, depending on the complexity of the applications you plan to write and the time you want to spend:

- **Start-to-finish.** To become familiar with most ObjectVision features and build the *Order* application from scratch, cover all the steps starting with Chapter 3. You can pause at any point to save your work; it's most convenient to stop at the end of a chapter.
- **Selected tasks.** This method is recommended if you already have experience with ObjectVision 1.0, or if the applications you intend to write do not take advantage of all ObjectVision features.

Several files contain the sample application, *Order*, at various stages of development. If you choose to skip all or part of a particular chapter, you can complete subsequent chapters by starting with the file named on the second page of the chapter.

- **The short course.** If you have limited time and want to sample ObjectVision features quickly, follow the instructions in Appendix A. The short course directs you to finish a partially-complete sample application, trying each feature just once or twice.

  The short course presents an abbreviated version of the material covered in Chapters 4 through 8. For example, Chapter 4, "Designing forms," provides detailed instructions for each type of object you can add to the two forms in the *Order* application. In Appendix A, Chapter 4 is condensed—the explanations are summarized, and you're directed to create a sampling of objects.

Tip: There might be ObjectVision terms and concepts unfamiliar to you, especially if you are only reading selected tasks. Remember to refer to the Glossary at the end of this book.

Other resources

After completing this book, you can refer to several sources for help designing your own applications:

- *ObjectVision Reference Guide* contains detailed explanations for each of the topics covered in *Getting Started*. Scan the table of contents or index to locate information on the topic you need.
- *Quick Reference Guide* summarizes basic procedures and lists @functions. You might find it helpful to keep the *Quick Reference Guide* handy during application design.
- *Online help* can be opened by pressing *F1* or choosing any Help menu command (except Help | About). The About command displays a dialog box with memory use and other information about the active application. For a description of online help, see Chapter 3, "Using an ObjectVision application."
- *Sample applications* are provided on the ObjectVision disk and are automatically copied to your hard disk during installation. You can use the sample applications as examples of features you'd like to include in your application.

Conventions

This manual uses special fonts and icons as follows:

`Monospaced type`	This font represents text as it appears on-screen and text that you must type.
Italics	Italics are used to emphasize certain words and to introduce terms that are defined in the Glossary. Italics are also used for argument names and ObjectVision application names—for example, "The *Order* sample application."
Keycap	This font represents a particular key you should press—for example, "Press *Del* to erase the character."
ALL CAPS	All caps are used to represent DOS directories and file names, and database file names.
SMALL CAPS	@Function names are printed in small capital letters.
Menu I Command	Rather than use the phrase "Choose the Save command from the File menu," this manual uses the convention File I Save.

Each chapter in this manual includes explanations of ObjectVision features as well as step-by-step instructions that demonstrate the features. Steps are marked with a pencil icon, shown at left. Some people prefer to skip the explanations and simply perform the steps; others prefer to read the explanations only.

Related explanations of a topic found elsewhere in *Getting Started* or in *ObjectVision Reference Guide* are marked with a book icon, shown at left.

Helpful tips, suggestions, and shortcuts are indicated with the lightbulb shown at left.

Clicking the *right* mouse button has a special effect when you're designing an application—it lets you inspect an object's properties. The symbol shown at left is a reminder of the right-click action.

Terms

- **Check** means to position the pointer on a check box and press the *left* mouse button to select a dialog box or value option.
- **Click** means to press the *left* mouse button while the pointer is on an object to select an object. For example, to follow the instruction "Click OK," move the pointer to the OK button, then click the left mouse button.

Note When the instructions don't specify a right mouse button click, click the left mouse button.

- **Drag** means to hold down the *left* mouse button while you move the mouse. Depending on the type of object selected, dragging can either create, resize, or move an object.
- **Press** *Enter.* On your keyboard, the carriage-return key may be labeled with a right-angle arrow that looks like this , or with the word *Return*.

Late-breaking news

Any late changes or additions to ObjectVision are documented in the README.TXT file on your master disk. README.TXT is copied to your ObjectVision directory during installation.

To read the file, use the Windows Notepad or any word processor. You can print the README.TXT file for easy reference.

How to contact Borland

Borland offers a variety of services to answer your questions about this product. Be sure to send in the registration card; registered owners are entitled to technical support and will receive information on upgrades and supplementary products.

Borland resources

Borland Technical Support publishes technical information on a variety of topics and can answer your questions.

800-822-4269
Techfax

TechFax is a 24-hour automated service that sends free technical information to your fax machine. You can use your touch-tone phone to request up to three documents per call.

408-439-9096
File Download BBS
2400 Baud

The Borland File Download BBS has sample files, applications, and technical information you can download with your modem. No special setup is required.

Online information services

Subscribers to the CompuServe, GEnie, or BIX information services can receive technical support by modem. Use the commands in the following table to contact Borland while accessing an information service.

Service	Command
CompuServe	`GO BORLAND`
BIX	`JOIN BORLAND`
GEnie	`BORLAND`

Address electronic messages to *Sysop* or *All*. Don't include your serial number; messages are in public view unless sent by a service's private mail system. Include as much information on the question as possible; the support staff will reply to the message within one working day.

(408) 438-5300
Technical Support
6 a.m. to 5 p.m. PST

You can telephone Borland Technical Support from 6:00 a.m. to 5:00 p.m. Pacific standard time. Call from a telephone near your computer and have the program running. Keep the following information handy to help process your call:

- Product name, serial number, and version number.
- The brand and model of any hardware in your system.
- Operating system and version number. (Use the DOS command `VER` to find the version number.)
- Contents of your AUTOEXEC.BAT and CONFIG.SYS files.
- Contents of your WIN.INI and SYSTEM.INI files (located in your \WINDOWS directory).
- A daytime phone number where you can be contacted.
- The specific steps to reproduce the problem.

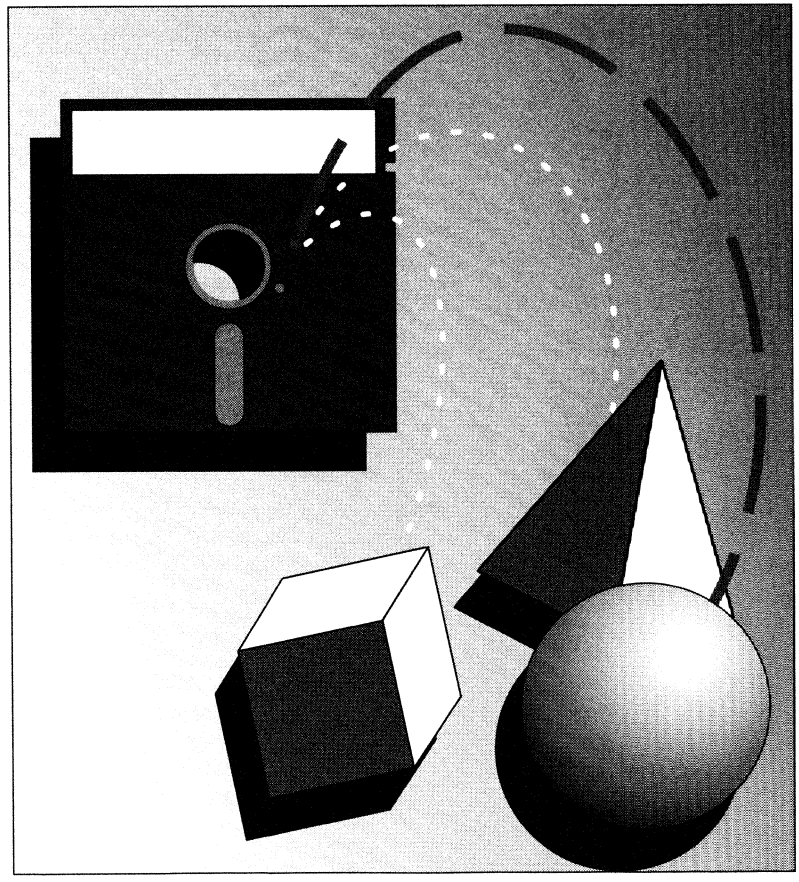
(408) 438-5300
Customer Service
7 a.m. to 5 p.m. PST

Borland Customer Service is available weekdays from 7:00 a.m. to 5:00 p.m. Pacific standard time to answer any non-technical questions you have about Borland products, including price information, upgrades, and order status.

C H A P T E R

1

Installing ObjectVision



Chapter goals:

- Running the Install program
- Installing SHARE
- Configuring the Paradox Engine
- Starting ObjectVision

This chapter explains how to begin running ObjectVision on your computer. When you finish this chapter, you'll be ready to begin using ObjectVision to create your applications.

This manual assumes you are familiar with Windows terminology. If you are not familiar with Windows, you might want to read Chapter 2, "Basic Skills," in the *Microsoft User's Guide*.

Software and hardware requirements

This section lists equipment supported by ObjectVision. In general, ObjectVision supports any Windows-certified device drivers shipped with Windows. An extensive list of Windows supported hardware, software, and devices is included in the documentation that came with your copy of Windows.

If non-standard drivers cause problems, revert to the standard Windows drivers.

Other device drivers that are *completely* Windows compatible, and are made specifically for your hardware and software configuration can also be used. If any problems arise, however, we recommend you revert to using the standard device drivers. If the problems go away, either the device drivers are improperly installed or they aren't completely Windows compatible.

For example, if you install a high-resolution screen driver and can't view all of the value tree or event tree conditions, it's probable that you are using VGA screen fonts with the high-resolution driver. In this case, high-resolution fonts need to be installed along with the high-resolution driver.

- **Computer.** ObjectVision is designed to run on IBM AT, PS/2, and fully-compatible computers using the Intel 286, 386, or higher processor.
- **Operating system.** ObjectVision runs with Windows 3.0, which requires DOS 3.1 or later. To use ObjectVision, you must run Windows in either 386 enhanced mode or standard mode. Real mode is *not* supported by ObjectVision.
- **Memory.** The requirements for running ObjectVision are different for 386 enhanced mode or standard mode:
  - To run Windows in 386 enhanced mode on your 80386 or 80486 PC, you need at least 2MB of RAM, 1024K of which is extended memory.

- To run Windows in standard mode on your 80286 PC, you need at least 1MB of RAM, 256K of which is extended memory.

You can add extended memory cards to your PC to improve Window's performance on your system.

- **Disk drive.** Your computer must have a hard disk with at least 2.5MB of free disk space.
- **Mouse.** It is *highly recommended* that you have a mouse to use this version of ObjectVision. Any mouse supported by Windows will work. If you don't have a mouse, the Object bar will not appear in the Form Tool, and you'll need to use the menu command equivalents.

Optional equipment

The following equipment is not necessary to run ObjectVision but will enhance its performance and ease of use:

- **Extended memory card.** The more memory your system has, the better.
- **Printer.** You can use any printer that is supported by Windows.

Upgrading from version 1.0

If you are upgrading to 2.0 from 1.0, notice that the default installation directory for 2.0 is also C:\VISION.

Caution! To keep your original version of ObjectVision 1.0, be sure to install 2.0 in a different directory.

Warning! If you save a 1.0 application with version 2.0, *that file can no longer be opened by ObjectVision 1.0.*

To keep your 1.0 .OVD applications and sample files, be sure that the name you specify for the sample files subdirectory during installation is one that doesn't already exist on your computer.

Caution! *ObjectVision 2.0 installation* ***will*** *overwrite any existing defaults.*

ObjectVision 2.0 has different defaults than 1.0 for the Paradox Engine. If you want to preserve your original WIN.INI settings, you can make a backup copy of your WIN.INI *before* installing ObjectVision 2.0.

For information about modifying your WIN.INI, see page 15.

Installing ObjectVision

The following instructions are for first-time users. If you are upgrading to 2.0 from 1.0, see the previous section.

What Install does

For a list of all ObjectVision files written to your hard disk during installation, see the FILELIST.DOC file on the ObjectVision disk.

The Install operations

The installation process does the following:

- creates a program directory on your hard disk.
- creates a sample files subdirectory on your hard disk.
- copies all ObjectVision files to the new directories.
- modifies your WIN.INI file.

WIN.INI is the Windows initialization file. Changes are written to this file during installation so that ObjectVision will run properly under Windows:

- **The default .OVD extension is added.**

```
[Extensions]
ovd=c:\directory\vision.exe ^.ovd
```

  where *directory* is the directory you specify for installing the ObjectVision system files.

- **Paradox Engine defaults are added.**

```
[Paradox Engine]
NetNamePath=C:\
UserName=Vision
ShareLocal=YES
NetType=5
MaxTables=16
RecBufs=128
MaxLocks=32
SwapSize=128
MaxFiles=16
```

- **The ObjectVision home directory.**

```
[ObjectVision]
HomeDir=c:\directory\
```

  where *directory* is the directory you specify for installing the ObjectVision system files.

■ **Btrieve defaults are added.**

```
[btrieve]
options=/m:38 /p:4096 /f:16 /l:20
```

The Install options

Check the installation options to:

- add an ObjectVision Windows group to the Program Manager.
- display the README.TXT notes.

If you don't check Create OV Windows Group, the ObjectVision icon will be placed in the active group on your desktop. The next section describes how to install ObjectVision on your system.

Running Install

To install ObjectVision,

1. Put the ObjectVision disk in drive A or B.
2. Start Windows, then choose File | Run from the Windows Program Manager.
3. In the Command Line text box, type `drive:install` where *drive* is the letter of the disk drive containing the ObjectVision disk as shown in Figure 1.1. Or, you can start installation from DOS by typing `win drive:install`, then pressing *Enter*.
4. Click OK.

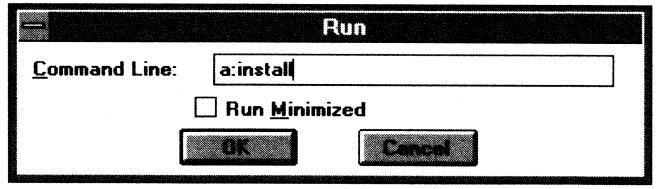
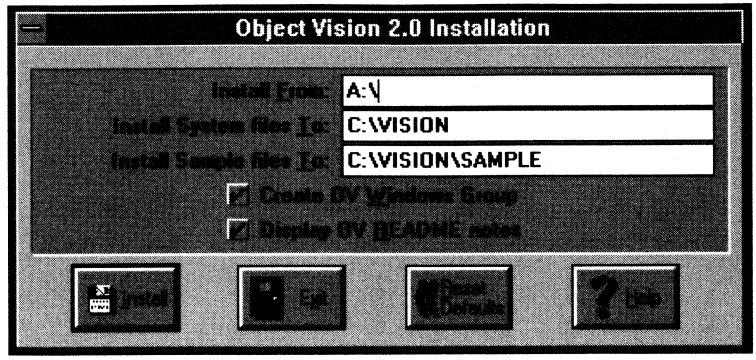


Figure 1.1
The Program Manager's File Run dialog box

5. The ObjectVision 2.0 Installation dialog box appears as shown in Figure 1.2.

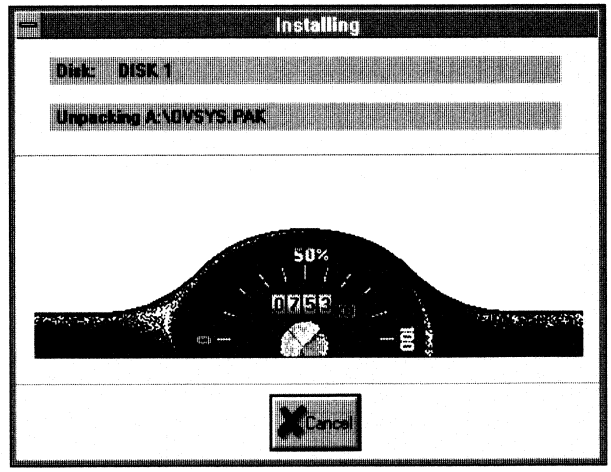
Figure 1.2
The ObjectVision 2.0 Installation dialog box



For assistance at any time before clicking the installation icon, click Help. When the Help window is onscreen, click any button or field in the installation dialog boxes to get help information.

6. When you finish selecting the installation options you want, click Install. The Installing dialog box appears and shows you what is being copied to your hard disk, as shown in Figure 1.3.

Figure 1.3
The Installing dialog box



When all files are copied to your hard disk, the installation process is nearly complete. If the DOS SHARE command is not in your AUTOEXEC.BAT file, a dialog box appears as a warning. SHARE is *not* necessary for successful installation, but it is necessary before you use Paradox tables with ObjectVision. For more information about SHARE, see the following section.

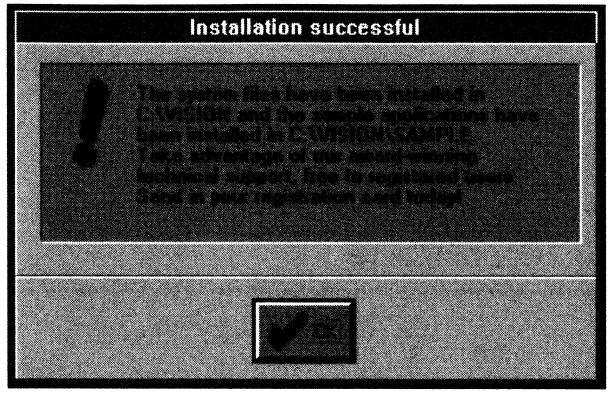
If you checked the Display OV README notes option, a dialog box appears. After you click OK, the README.TXT file

displays in a window. To close the window and continue with installation, double-click the top left corner of the window.

The Installation Successful dialog box (Figure 1.4) lets you know installation is complete.

7. Click OK to return to the Windows desktop.

Figure 1.4
The Installation Successful dialog box



If you have checked the Create OV Windows Group option, a new ObjectVision Windows group displays in front of all other windows, and the ObjectVision icon appears in it. Otherwise, the ObjectVision icon appears in the active Windows group.

Installing SHARE

If you plan to use Paradox tables with your ObjectVision applications, you must install the DOS SHARE utility. SHARE is also normally used in a network environment when files are shared.

To determine if SHARE is already loaded on your computer, exit Windows, type `share` at the DOS prompt, then press *Enter*. If SHARE is already loaded, you'll see the message `SHARE already installed`.

Tip: Running DOS *from within Windows* always reports that SHARE is installed, even when it isn't. Be sure to test for SHARE *before* you start Windows.

To install SHARE,

- If you have an AUTOEXEC.BAT file that already has C:\DOS in the PATH statement, add the following line after the PATH statement and before the WIN command that starts Windows:

  ```
  SHARE
  ```

  After you finish, restart your computer.
- If you don't have an AUTOEXEC.BAT file, type `C:\DOS\SHARE` as a line in a text file and save it using the name AUTOEXEC.BAT. Store the file in the root directory of your hard disk, then restart your computer.

The Paradox Engine

When ObjectVision links to Paradox database tables, the Paradox Engine is used to access the tables.

The Paradox Engine lets you

- create Paradox tables for your application.
- link ObjectVision applications to Paradox tables.

During installation of ObjectVision, the recommended defaults for the Paradox Engine are written to your WIN.INI.

Note that the default setting for the ShareLocal option is *Yes*. You might gain a modest performance improvement by setting ShareLocal to *No*.

Multiple instances of ObjectVision can share Paradox tables even when ShareLocal is set to *No*. You can set ShareLocal to *No* if no other process running at the same time shares the same Paradox tables with an ObjectVision application.

If you need to change the installation defaults for the Paradox Engine, see Appendix B, "Configuring the Paradox Engine."

Starting ObjectVision

You can start ObjectVision from the Windows desktop or from the DOS prompt.

To start ObjectVision from the Windows desktop, double-click the icon.

If you want an ObjectVision application to be started when an icon is double-clicked, create a new item using the Program Manager's File | New command. Type `vision filename` in the Command text box to start ObjectVision with the application named *filename*. Refer to your Windows documentation for more details.

Follow these steps to start ObjectVision from the DOS prompt:

1. Change to your ObjectVision directory by typing

   ```
   cd\vision
   ```

   and pressing *Enter*.
2. To start ObjectVision *and* Windows at the same time, type

   ```
   win vision
   ```

   and then press *Enter*.
3. To start ObjectVision and Windows, and also open an ObjectVision application, type

   ```
   win vision application-name
   ```

   and press *Enter*. For *application-name*, type the name of the ObjectVision application you want to open, omitting the .OVD extension. For example, the following command opens the *Order* sample application when you start ObjectVision from DOS:

   ```
   win vision c:\vision\sample\order
   ```

Modifying your WIN.INI file

WIN.INI is a text file that holds configuration information for Windows.

The ObjectVision installation program makes some modifications to the WIN.INI file in your Windows directory, as described on page 10.

It isn't necessary for you to make any further modifications to that file, but if you want to change the default settings, use the Windows Notepad, SysEdit, or another text editor of your choice to modify WIN.INI.

SYSEDIT.EXE is copied to the SYSTEM directory during Windows installation. You can use SysEdit to edit your AUTOEXEC.BAT, CONFIG.SYS, SYSTEM.INI, and WIN.INI files.

The load and run statements

Your WIN.INI file contains both a load statement and a run statement. By default, these two statements are blank and look like this:

```
load=

run=
```

You can load ObjectVision or specific files every time you start Windows by editing the load and run statements as follows:

- **To load ObjectVision as an icon**, enter

  ```
  load=C:\vision\vision.exe
  ```

- **To load ObjectVision as an active window**, enter

  ```
  run=C:\vision\vision.exe
  ```

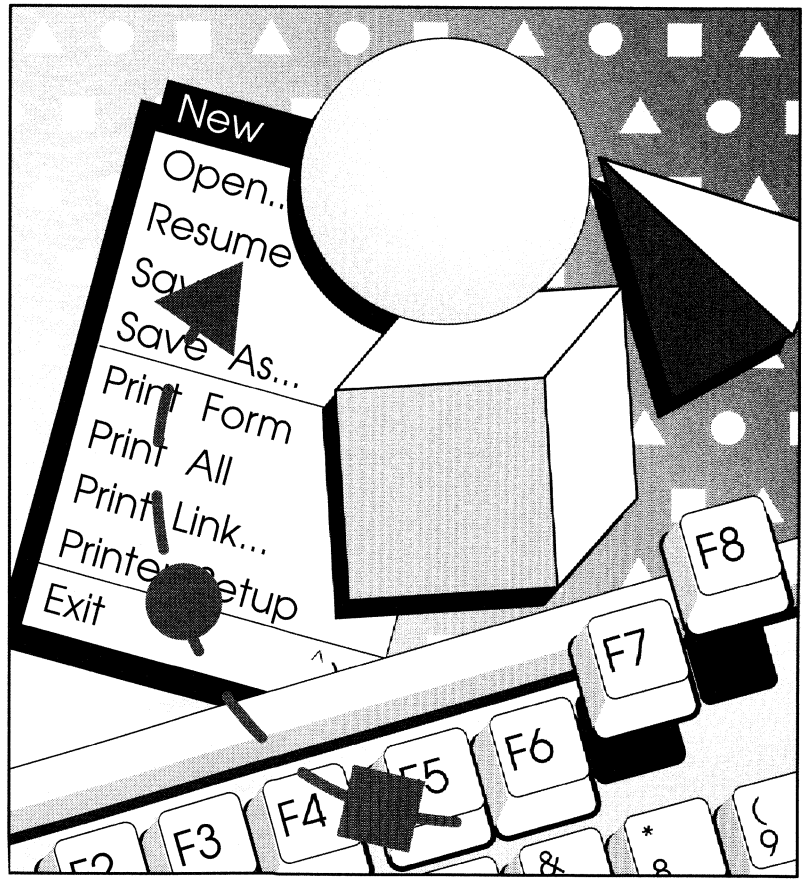
- **To load a specific ObjectVision application**, enter

  ```
  load=C:\vision\filename
  ```

C H A P T E R

2

Basic features



Chapter goals:

- Opening ObjectVision applications
- Saving ObjectVision applications
- Printing ObjectVision applications
- Learning keyboard operations

Most ObjectVision features are explained in this tutorial. This chapter covers the essential features, especially for first-time users: opening applications, keyboard operations, and saving and printing your applications.

Overview

ObjectVision lets you visually create business applications for Windows. You *do not* have to be a programmer. *No experience* with programming or scripting languages is needed.

ObjectVision applications use a forms interface because it's an easy way to collect and display information. When an application is opened, ObjectVision is in form completion mode.

You open the Form Tool to create or modify forms in an application. The Form Tool is where you visually create ObjectVision applications, by building forms, placing objects on the forms, and assigning properties to the form objects. You can also define special properties—value trees and event trees—using spreadsheet-like expressions.

Business rules and procedures can be embedded in an ObjectVision application to automatically calculate field values or carry out menu commands. The logic of the application is graphically represented and easily understood.

After fields are created, you can create links between ObjectVision fields and a variety of local and network data sources.

The following sections discuss how to open, save, and print ObjectVision applications. If you don't have a mouse, the final section in this chapter discusses the keyboard equivalents for ObjectVision operations.

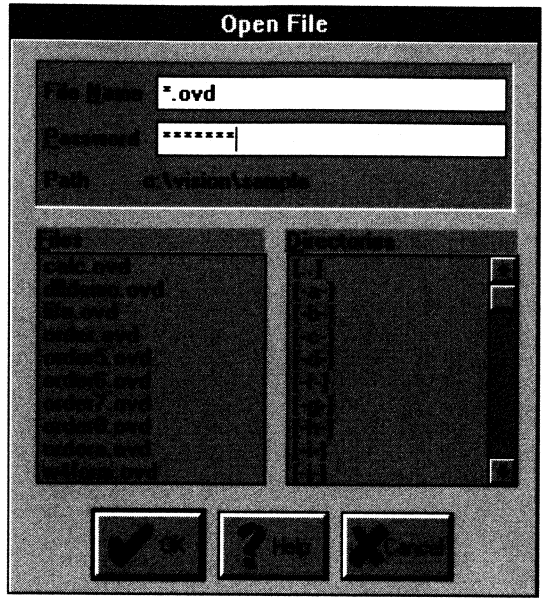
Opening ObjectVision applications

Choose File | Open to open an ObjectVision application. The Open File dialog box appears onscreen. Specify the directory and file name of the application you want to open, then click OK.

Note ObjectVision applications can be saved with or without a password. If you choose to use a password for your application, it displays as a series of asterisks in the Open File dialog box. A

password can be typed in the Open File dialog box when an application is opened, as shown in Figure 2.1.

Figure 2.1
The Open File dialog box



Warning! If a password-protected application is opened without its correct password, the Tools menu does *not* appear and the user is unable to create new forms or modify existing forms in that application.

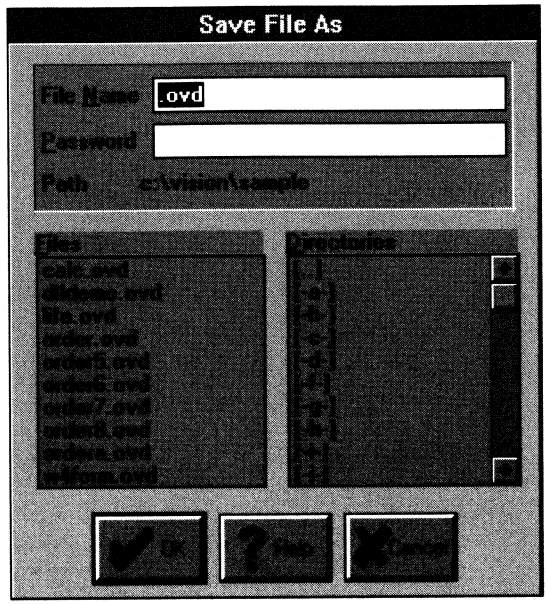
Saving ObjectVision applications

Be sure to save your work frequently as you take the tutorial.

To save changes, choose either the Save or Save As command from the File menu:

- **Save** saves the changes to the current application file name, overwriting the previous data. If you have not yet named the file by saving it previously, the Save File As dialog box appears (Figure 2.2). Specify the directory and the file name you want, then click OK.

Figure 2.2
The Save File As dialog box



- **Save As** saves the new version of the application under a different name, leaving the previous version the same as it was the last time it was saved.

  If you enter a Save As file name that already exists, ObjectVision asks if you want to replace the file. You can choose Yes to overwrite the existing file, or No to return to the File Save As dialog box and enter another file name.

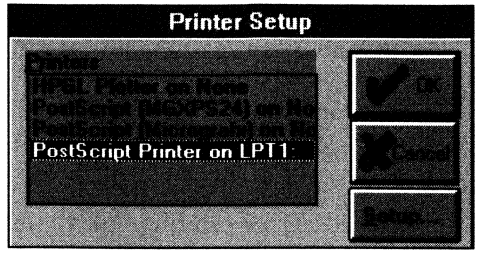
Tip: As you design ObjectVision applications, use File | Save As to save your work frequently, using different names for different versions. This strategy lets you retreat to an earlier version if you try—and then reject—a new concept.

Printing ObjectVision applications

You can print ObjectVision forms, value trees, or event trees any time during application design. You can also print link values during form completion. To set up your printer, follow these steps:

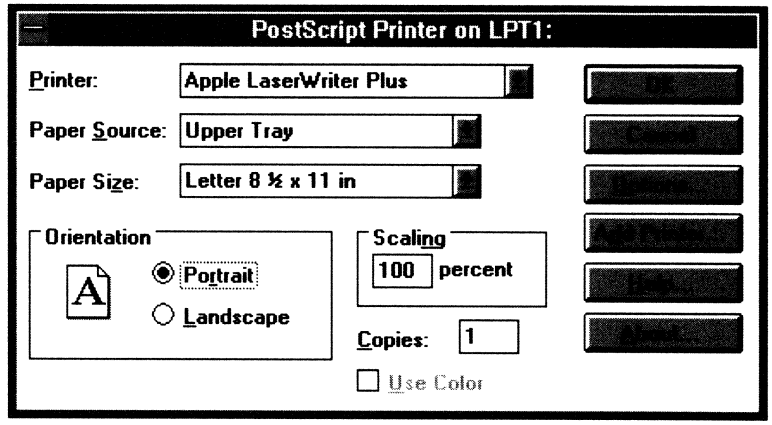
1. Choose File | Printer Setup. A list of your installed printers appears in the Printer Setup dialog box as shown in Figure 2.3.

Figure 2.3
The Printer Setup dialog box



2. Select the printer you want from the list.
3. If you want to change the printer options, click Setup. A dialog box appears for the specific printer you've chosen, as shown in Figure 2.4.

Figure 2.4
A printer-specific Setup dialog box



4. Select the settings appropriate for your printer, then click OK. These settings remain in effect until you change them. The Printer Setup dialog box is still onscreen.
5. Click OK when you finish.

After your printer is set up correctly, you can choose File | Print Form or File | Print All:

- **Print Form** prints the active form, which is the form in front of any other forms that appear onscreen.
- **Print All** prints all forms in the application.

If the open application has links to external data sources, such as the sample *Order* application, *and* form completion mode is active, you can also choose File | Print Link.

- **Print Link** prints the active form until all values in the active link are printed. If the link is connected to an ObjectVision field, the form will print once for every record in the link. If the link is connected to a table object, the total number of forms printed depends on the number of visible rows in the table object, since table objects usually contain multiple records.

Instructions for printing link values, value trees, and event trees are provided in the respective chapters of the *ObjectVision Reference Guide* manual.

Solving printing problems

If the printed form doesn't look like the onscreen form, check the following:

- Be sure that you have correctly installed *printer fonts* as well as *screen fonts*. If the appropriate screen or printer fonts are not installed, Windows substitutes another font. To view the form as it will appear on the printer, choose View | Printer.
- Be sure that you have installed the most current version of the printer driver. If you have an old printer driver, the document might not print or might compress the form objects so that they are unreadable.

If the form doesn't print, be sure that you have selected the correct printer driver from the Windows Control Panel. If the wrong printer driver is selected, either the document won't print or the quality of the printed copies will be unpredictable.

Learning keyboard operations

Although we strongly recommend that you use a mouse with ObjectVision, most operations can be performed using a keyboard. In this tutorial, most steps use the mouse because it's easier. See Table 2.1, which lists the ObjectVision keystrokes and describes what they do, if you want to use the keyboard instead.

Table 2.1: ObjectVision keystrokes

Keystroke	Product area	Description
Alt+ Letter	Menu bar	Shortcut for opening a menu.
Alt	Menu bar	Selects the menu bar (by default, the first menu name is highlighted).
Esc	Menu bar	Unselects the menu bar.
← or →	Menu bar	Highlights a menu name when the menu bar is selected.
Enter or ↓	Menu bar	Opens the selected menu.
Enter	Open menu	Carries out the selected command.
Esc	Open menu	Exits from a menu without carrying out a command.
Alt+Spacebar	ObjectVision window	Opens the Control menu for the application window.
Alt+- (hyphen)	Form window	Opens the Control menu for the form window.
Alt+ Letter	Dialog box	Shortcut for directly selecting an option.
Esc	Dialog box	Closes the dialog box and leaves the settings unchanged.
Spacebar	Dialog box	Toggles a selected check box between checked and unchecked.
Enter	Dialog box	Activates a highlighted command button such as OK or Cancel.
Tab	Dialog box	Moves the focus to the next named option or group of options.
Shift+Tab	Dialog box	Moves the focus to the previous named option or group of options.
Ctrl+Enter	Dialog box	Moves to the beginning of a new line in a text box.
↑ or ↓	Dialog box	Selects a highlighted radio button or selection list item (you can also press the first letter of an item in a list to move to that item).
Spacebar	Form completion	Activates a selected button or checks a selected box for a value option.
F1	Form completion	Displays Help for a highlighted field (if help text exists).
Ctrl+PgUp or *Ctrl+PgDn*	Form completion	Scrolls a form horizontally.
PgUp or *PgDn*	Form completion	Scrolls a form vertically.
Alt+Backspace	Form completion	Backs up to the previous field and restores the previous value (only if pressed *immediately* after pressing *Enter* or *Tab* to enter a typed value).
Enter	Form completion	Enters a value in a field and moves to the next field requiring user input.
Shift+Enter	Form completion	Enters a value in a cell, exits the table object, and then selects the next field.
Tab	Form completion	Enters a value in a field and moves to the next field, going from left to right and from top to bottom.

Table 2.1: ObjectVision keystrokes (continued)

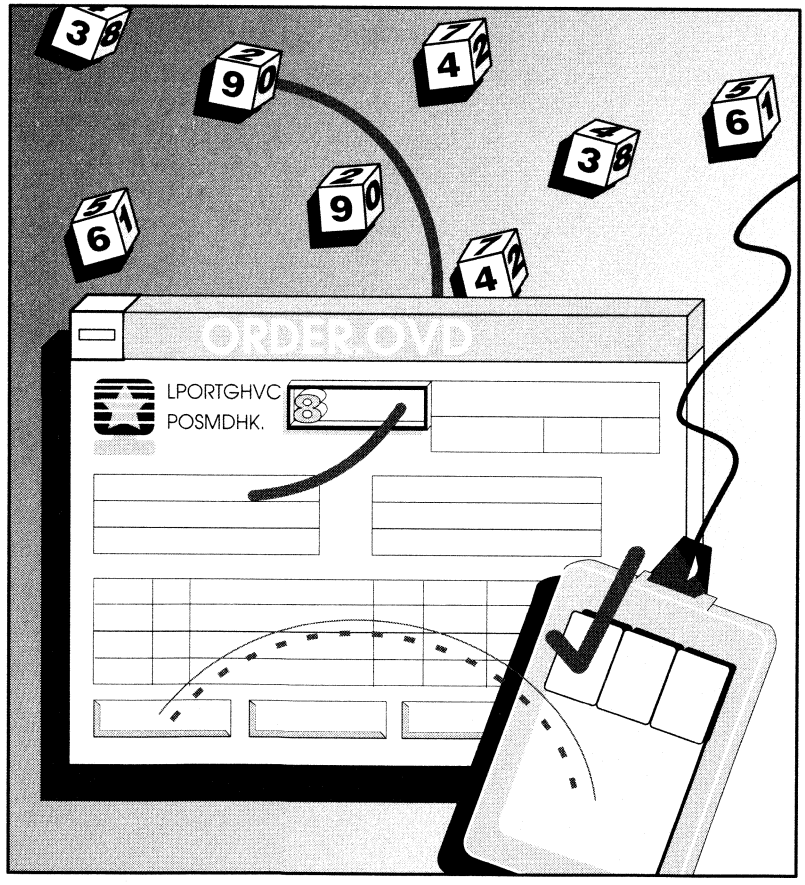
Keystroke	Product area	Description
Shift+Tab	Form completion	Moves to the previous field, going from bottom to top and from right to left.
Esc	Form completion	Removes a typed value from a field, if it hasn't been entered yet.
*Shift+*arrow keys	Form design	Resizes a selected object by moving the bottom right corner (except table objects).
Ctrl+PgUp or *Ctrl+PgDn*	Form design	Scrolls a form, a value tree, or an event tree horizontally.
PgUp or *PgDn*	Form design	Scrolls a form, a value tree, an event tree, or the Stack Tool vertically.
Tab	Form design	Selects the next object, going from left to right and from top to bottom.
Shift+Tab	Form design	Selects the next object, going from bottom to top and from right to left.
Ctrl+Tab	Form design	Selects multiple objects, going from left to right and from top to bottom.
Shift+Ctrl+Tab	Form design	Selects multiple objects, going from bottom to top and from right to left.
Home	Tree design	Moves to the root node of a value or event tree.
↑ or ↓	Tree design	Moves to the previous or next node in a value or event tree.
← or →	Tree design	Moves to a node at the preceding or subsequent nesting level in a tree.
Ctrl+Home	Tree design	Expands the display of a value tree or an event tree.
Ctrl+End	Tree design	Reduces the display of a value tree or an event tree.

Note that keyboard shortcuts for menu commands are not listed here, but they also provide quick methods you can use without a mouse.

C H A P T E R

3

Using an ObjectVision application



Chapter goals:

- Opening *Order*
- Viewing an existing record
- Adding a new record
- Editing an existing record
- Understanding form design
- Viewing online help

This chapter explains how to use the completed application and gives a glimpse of the primary processes involved in designing the application. Before beginning to design the *Order* application, try using it in its completed form, just as an end user would.

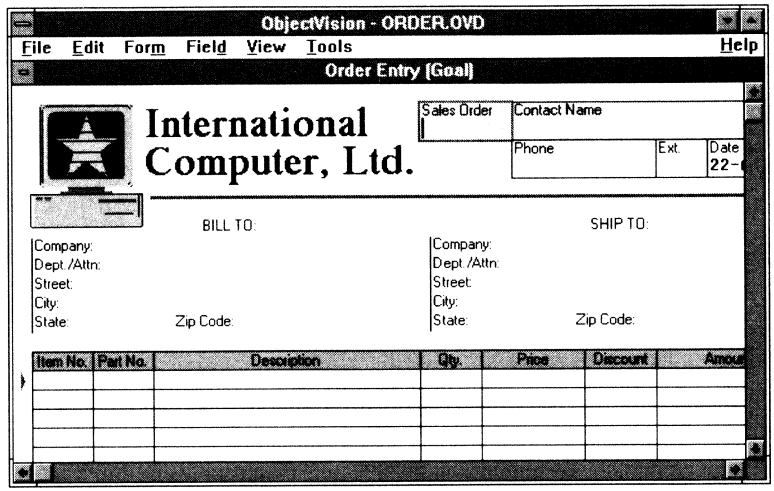
Opening the application

First, start ObjectVision and open the *Order* application:

1. Choose File | Open. The Open File dialog box appears.
2. Specify the samples directory (installation default is SAMPLE), choose ORDER.OVD, and then click OK.

   The Order Entry form appears as shown in Figure 3.1.

Figure 3.1
The Order Entry form in Order



When you open an ObjectVision application, the insertion point appears in the first field requiring a value from the user. The Sales Order field is the first field on the Order Entry form.

3. Click the Maximize button, shown at left, located in the top right corner of the window.

   You should click the Maximize button to enlarge the ObjectVision window (if it isn't already) whenever you start ObjectVision in this tutorial.

In ObjectVision applications, the user interface is called a *form* and every ObjectVision application contains one or more forms.

Currently, you're looking at the Order Entry form. All forms in an application are collectively called a *stack*.

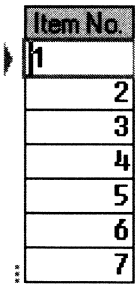
Viewing an existing record

The *Order* application is connected to four Paradox database tables: ORDERS, CUSTOMRS, ITEMS, and PARTS. If you type a value in the Sales Order field, *Order* automatically searches the linked database tables, locates that record, and fills in the other fields with the associated values.

Try typing `1` in the Sales Order field and pressing *Enter*. The *Order* application supplies the values for all the fields on the form, based on the values in the Paradox database tables ORDERS, CUSTOMRS, ITEMS, and PARTS.

There are six orders in the ORDERS table. To view another record, do the following:

1. Click the Sales Order field or press *Shift+Tab* to select it.
2. Delete the 1.
3. Type `6` (another existing sales order number), then press *Enter*. The order information associated with that order number appears.



An ObjectVision table object can display multiple values, but only one row of a table is active at a time. The *row pointer* (shown at left) displays at the left margin of the table and indicates the current or active row.

When a table object contains more values than can display at one time on the form, vertical dots appear to the left of the table object.

For example, the table object in *Order* has seven rows, but order number six contains *nine items*. The vertical dots indicate you can scroll down to see the other two items in the order.

To scroll values in the table,

- **Press a function key.** Function keys *F3* through *F8* are shortcut keys for scrolling values in a table. Pressing one of these keys is *the fastest way* to scroll values.

Refer to Table 3.1 for a description of the effect of pressing each key.

Table 3.1
Scrolling and clearing values in a table object

Function key	Description of effect
F3	View previous row (if there is one)
F4	View next row (if there is one)
F5	View moved up (the number of rows – 1)
F6	View moved down (the number of rows – 1)
F7	View first row
F8	Clear (empties the table of all values)

These function keys only work *during form completion*, when a table object is selected. The positioning of displayed records is based on the location of the row that was current before the function key was pressed.

- **Click the scroll-up arrow.** This arrow is located at the top of the scroll bar next to the table object. Clicking this arrow either displays the *previous value* in the database table (if there is one), or it makes the *previous row* in the table object the current row.
- **Click above the scroll box.** This is equivalent to moving the view of data up by the number of rows in the table, minus one.
- **Click the scroll-down arrow.** This arrow is located at the bottom of the scroll bar next to the table object. Clicking this arrow either displays the *next value* in the table (if there is one), or it makes the *next row* in the table object the current row.
- **Click below the scroll box.** This is equivalent to moving the view of data down by the number of rows in the table, minus one.
- **Click the scroll box.** This moves the view of data to the top of the database.

Adding a new record

Now use the *Order* application as an end user would to add a new order to the databases.

1. Click the New Order button. The previous information is cleared from the form and the Sales Order field is selected.

In Chapter 7, "Creating event trees," you'll see the @functions attached to this button, which enables it to remove all values from both forms.

The insertion point appears in the first field, Sales Order.

2. Type `7`, then press *Enter*. Seven is a number that doesn't exist in the database, so no information for order number seven displays.

   The insertion point appears in the next field, Contact Name.

   In ObjectVision applications, the insertion point moves through the form from left to right and top to bottom, based on the location of the lower right corner of a field that requires the user to enter a value. This is called *guided completion*.

3. Type your name in the Contact Name field. If you make a mistake typing, use ← and *Del* (or *Backspace*) to correct it. Or, since you haven't pressed *Enter* or *Tab* yet, you can press *Esc* to remove the text and start over.

4. Press *Enter* to move to the Phone field. Type your phone number in the Phone field.

   - Try typing *letters* in the Phone or Ext. fields. The application beeps because these fields, called *picture fields*, were designed to accept numeric values only.
   - Type a few numbers in the Phone field and try pressing *Enter* before the phone number is complete. ObjectVision displays the message `Field value is incomplete`.

5. Finish typing the phone number, then press *Enter*.

6. Type an extension number, then press *Enter*.

   If you don't have an extension, don't type any numbers—just press *Enter*.

   The BILL TO area Company field is selected. Note that it is not a field you can type text into; it's a *selection list* field. You can only select an existing value from the list.

   For now, assume the company you need is in the list. Adding a new company to the list is explained on page 31.

7. Press ↓ two times to highlight `Borland`, then press *Enter*. The billing and shipping information for Borland automatically fills in the form.

   Item No., the first column in the table object in the middle of the form, is now highlighted. The insertion point appears in the first cell in the Item No. column.

Note also that the row pointer appears to the left of the cell. This shows you which row of a table is the current or active row.

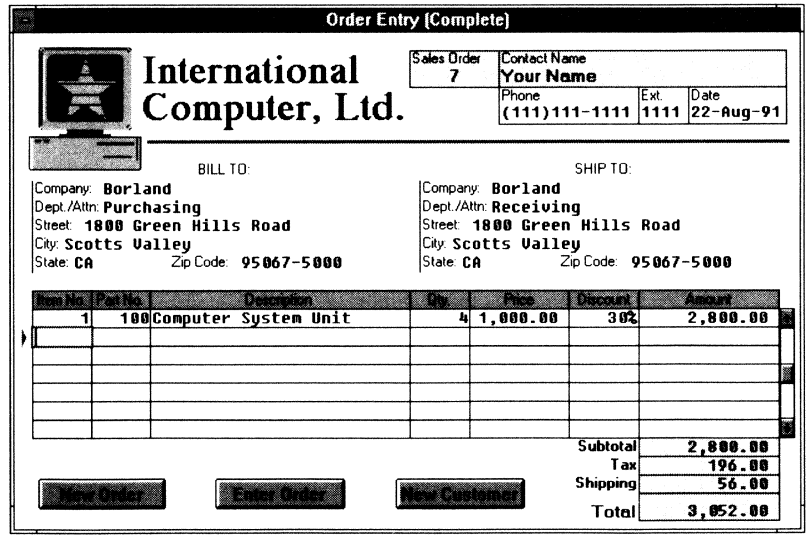
8. Type `1` in the Item No. cell, then press *Enter*.
9. Type `100` in the Part No. cell, then press *Enter*.

   In the linked Paradox table, PARTS, *Part No* is an index field—so typing a value in the Part No. column causes the application to fill in the associated values for an existing part number in the linked ObjectVision fields, Description and Price. Chapter 8, "Creating links," explains the "parts" link in detail.
10. Type a number, say `4`, in the Qty. cell and press *Enter*.

    The application automatically calculates the discount and the extended price in the Amount field, as shown in Figure 3.2. *Order* also calculates the subtotal, tax, shipping and total for this order.

    The second row of the Item No. column is now selected. The row pointer advances along with the insertion point.

Figure 3.2
The Discount, Amount, Subtotal, Shipping, Tax, and Total field values are automatically filled in



The 30% discount is associated with the customer type defined for this customer: Distributor. Since this customer is not tax exempt, the tax is calculated as 7% of the subtotal. The Shipping field value is calculated as 2% of the Subtotal.

11. Start adding another item to the order by typing `2` in the Item No. cell, and then pressing *Enter*.
12. Type `200` in the Part No. cell, then press *Enter*.
13. Type `1` in the Qty. cell, then press *Enter*.

    The application automatically calculates the extended price for this item, and updates the Subtotal, Tax, Shipping, and Total field values.

Save the new order information.

14. Press *Shift+Enter* to leave the table object.

    The New Order button at the bottom of the form is now highlighted.

    At this point, only the information you entered in the table object has been saved to a database. The links for the table object have been defined so that the information entered in the columns are automatically saved.

    To save the entire order, including the contact information and the company name, however, that information must also be saved to the database.

15. Click the Enter Order button. Or, press *Tab*, and then press *Spacebar*.

    All information you entered on the form is now saved to the ORDERS database.

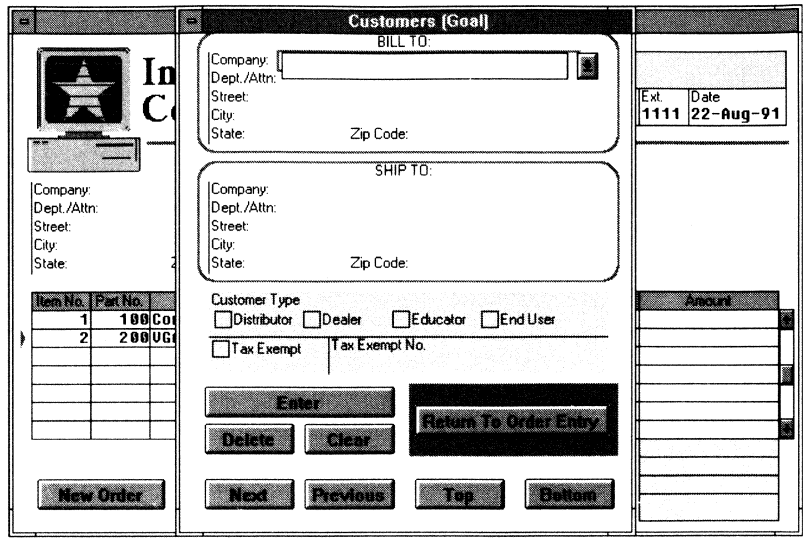
Adding a new customer

To select a new customer from the BILL TO area selection list, you first need to add that customer to the database.

To add a new customer to the database,

1. Click the New Customer button at the bottom of the Order Entry form.
2. The other form in the *Order* application, Customers, appears in front of the Order Entry form as shown in Figure 3.3.

Figure 3.3
The Customer form in Order



The Customers form is used to add new customers to the database because a user cannot enter this information into the BILL TO and SHIP TO area fields on the Order Entry form.

Switching between forms

The *Order* application includes button shortcuts for viewing the Customers and Order Entry forms. You can switch between forms by clicking the New Customer button on the Order form, or the Return to Order Entry button on the Customers form.

If the application didn't have these buttons, you could move between forms by choosing Form | Select and then choosing the form name from the list in the Form Name dialog box.

Viewing the CUSTOMRS database

Notice the combo box arrow to the right of the Company field at the top of the Customers form. Click it to see a list of customers already in the linked Paradox database. Choose any customer from the list to view its BILL TO and SHIP TO information, its Customer Type, and Tax Exempt Status.

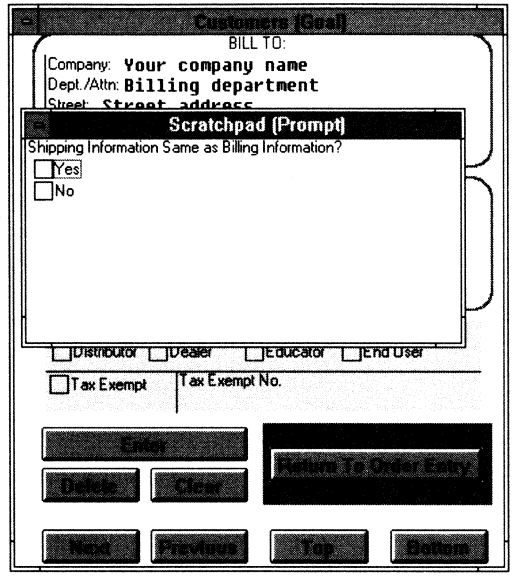
Click the Next and Previous buttons to browse through the CUSTOMRS database. When you finish browsing, click the Clear button to remove the display of values from the form.

3. Type the new customer company name into the combo-box text box, then press *Enter*.
4. Type the rest of the information into the BILL TO area fields, pressing *Enter* after each value is typed.

A window appears with the message `Shipping Information Same as Billing Information?` as shown in Figure 3.4. The window with the message is called a *Scratchpad form*. A Scratchpad form automatically appears when the application requires a value for a field that is not placed on any form.

In this example, the Scratchpad form requires a *Yes* or *No* value for whether the SHIP TO area values are all the same as the BILL TO area values.

Figure 3.4
The Scratchpad form in Order



5. Press *Spacebar* to check Yes, then press *Enter*.

   The values you enter into fields in the BILL TO area appear in the SHIP TO area fields.

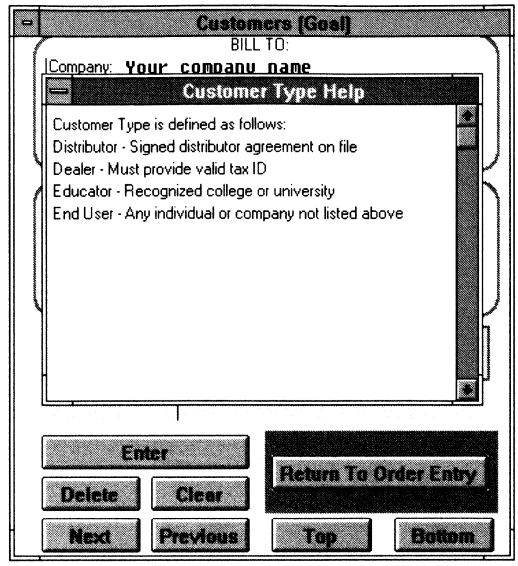
   If you check No in the Scratchpad form shown in Figure 3.4, you are prompted to determine if any of the SHIP TO area values are the same as the BILL TO area values.

   The insertion point now appears in the Customer Type field. A border appears around Distributor, indicating that it is the current focus.

Press F1 to view help text for a selected object.

6. Press *F1* to view help text associated with this field. The help text appears as shown in Figure 3.5.

Figure 3.5
Help text for Customer Type



Help text is a property that can be added to any field in an ObjectVision application. If help text has *not* been added to a field, ObjectVision Help appears when the user presses *F1* or chooses any Help menu command except the About command.

7. Press *Esc* to close the Customer Type Help window.
8. Press *Spacebar* to check the Distributor value, or click another value.
9. Press *Enter* to move to the Tax Exempt field, press *Spacebar* to check Tax Exempt, then press *Enter*.
10. Type any number in the Tax Exempt No. field, then press *Enter*.
11. Click the Enter button at the bottom of the form to add the new customer record to the CUSTOMRS database.
12. To confirm that the record has been added to the database, select the BILL TO area *Company* field and click its combo-box button. Notice that the name of the new company you entered now appears in the list.

    The *Order* application automatically displays all information you just entered for that company if you choose it from the list.
13. Now that you've saved this new customer information to the database, you can close the Customers form and return to the Order Entry form.

There are four ways to return to the Order Entry form:

- Click the Return to Order Entry button.
- Double-click the Control-menu box in the upper left corner of the Customers form window.
- Choose Form | Select, then double-click Order Entry.
- Click any visible portion of the Order Entry form.

Editing an existing record

If an order needs to be changed, you can display the order, edit the text, then save the changes to the database.

For example, you might learn that you misspelled a company name, or an item needs to be deleted from an order.

To edit any saved information, you first need to edit the text and then save the changes to the database.

For example, to update a phone number that has changed:

1. Select the Sales Order field on the Order Entry form.
2. Type the order number of the order with the old phone number.
3. Drag the pointer to highlight the old phone number.
4. Type the new phone number. The highlighted text is replaced by the new phone number and the field displays a dot pattern.
5. Press *Enter*.
6. Click the Enter Order button. The new phone number is used to update the record for the order.

Designing forms

The behavior of any ObjectVision application can be defined in terms of its objects, properties, and links. Each of these elements of application design is briefly introduced here as a preview of this tutorial.

Objects

Choose Tools | Form to open the Form Tool. When the Form Tool is open, you can create, delete, move, and resize objects, but you cannot fill in any values. This is the form edit mode. Chapter 4 gives more details on designing forms.

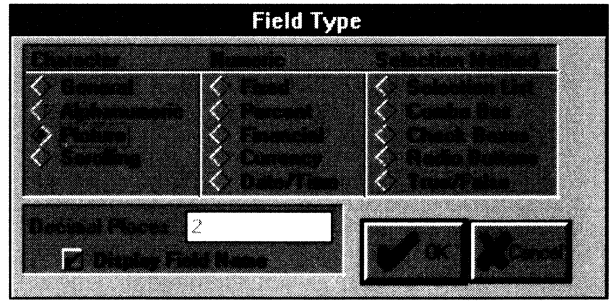
Use the mouse to select an object—such as the New Order button—and drag it to a new location. Move it back to its original location.

Notice the name STARLOGO.OVG beneath the graphic at the top left of the form. This is the name of the ObjectVision file containing the graphic.

Properties

Inspect the properties of the Phone field by clicking it with the *right* mouse button. Choose Field Type. The Field Type dialog box appears:

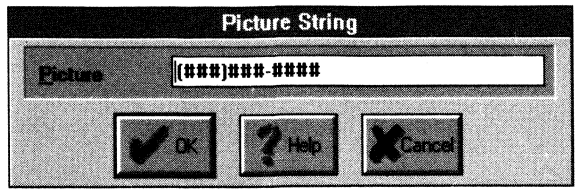
Figure 3.6
The Field Type dialog box



Notice that the Picture field type is selected. This means the phone number value must match the "template" of a phone number that was defined as the picture.

Click OK to see the picture defined as the value template for the Phone field. The Picture dialog box appears, as shown in Figure 3.7.

Figure 3.7
Picture String dialog box



The number symbol is used in a picture to restrict user entries to digits only. Click OK to close the dialog box.

Value trees and event trees

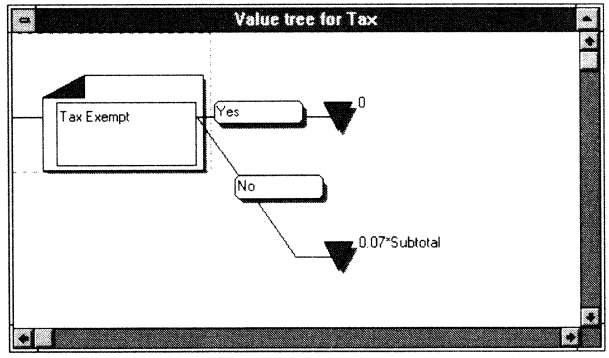
Two special types of properties are value trees and event trees. Value trees show how the application will calculate a value, and event trees show how the application will respond to specific user actions (called *events*).

The value tree property

The value tree property shows the logic used to make a calculation.

In the Order Entry form, double-click the field to the right of the word *Tax*. Its value tree appears as shown in the following:

Figure 3.8
The value tree for Tax



The value tree shows how the application calculates the value for the Tax field: If the value in the Tax Exempt field is *Yes*, it returns the value 0. If the value in the Tax Exempt field is *No*, it returns the value *0.07* multiplied by the value in the Subtotal field.

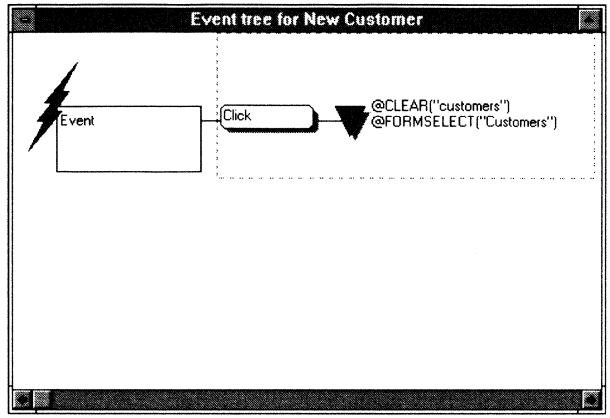
Double-click the Control-menu box to close the value tree. Building value trees is covered in Chapter 6, "Creating value trees."

The event tree property

The event tree property shows how the application responds to specific user actions.

In the Order Entry form, double-click the New Customer button. Its event tree appears as shown in Figure 3.9.

Figure 3.9
The event tree for the New Customer button



The event tree shows what action occurs when a user action triggers it. In this case, if the user clicks the New Customer button, the application removes the BILL TO and SHIP TO area values and then selects the Customers form.

- @CLEAR is an *event function* that clears the form of all values in a named link.
- @FORMSELECT is an event function that is equivalent to choosing Form | Select, then choosing a form name.

Other event functions are covered in Chapter 7, "Creating event trees."

Double-click the Control-menu box to close the event tree.

Links

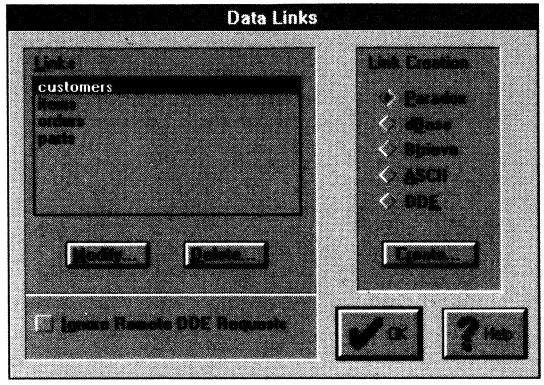
The *Order* application has four links: "customers," "items," "orders", and "parts."

You saw a demonstration of the "customers" link—when you selected a value from the BILL TO Company field list, the application automatically supplied the values for the remaining fields in the BILL TO and SHIP TO areas.

You also saw a demonstration of the "parts" link—when you entered a value in the Part No. cell, the application automatically supplied the values for the Description and Price cells.

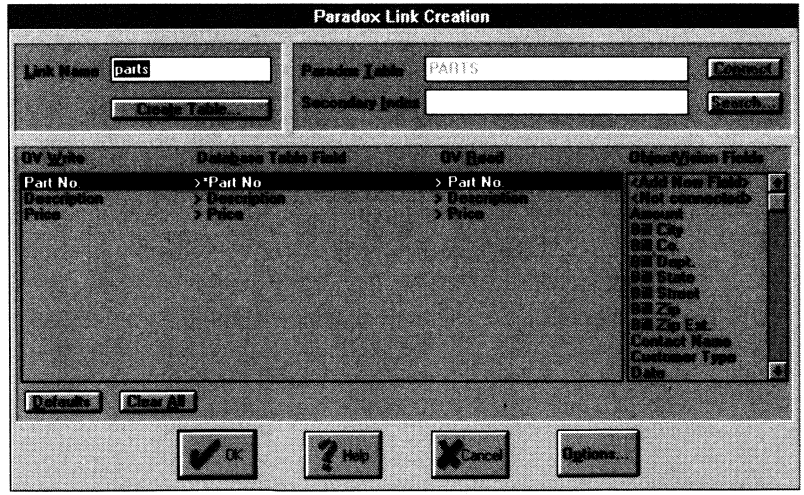
To see how these links were established, choose Tools | Links. The Data Links dialog box appears as shown in Figure 3.10.

Figure 3.10
The Data Links dialog box



Select the "parts" link from the list of link names in the dialog box, then click Modify. As a shortcut, you can double-click the link name. The Paradox Link Creation dialog box appears as shown in Figure 3.11.

Figure 3.11
The Paradox Link Creation dialog box



Notice that the Database Table Fields in the center column—Part No, Description, and Price—are connected to the corresponding ObjectVision fields on the Order Entry form. The asterisk (*) to the

left of Part No indicates that the field is an *index field*. An index field in a table is like an index in a book—it lets ObjectVision locate records quickly.

If the value for Part No. is supplied—as it was when you typed the number `100`—the "parts" link automatically retrieves the other values. Links are covered in Chapter 8, "Creating links."

Click Cancel to close the Paradox Link Creation dialog box. Then, click OK to close the Data Links dialog box.

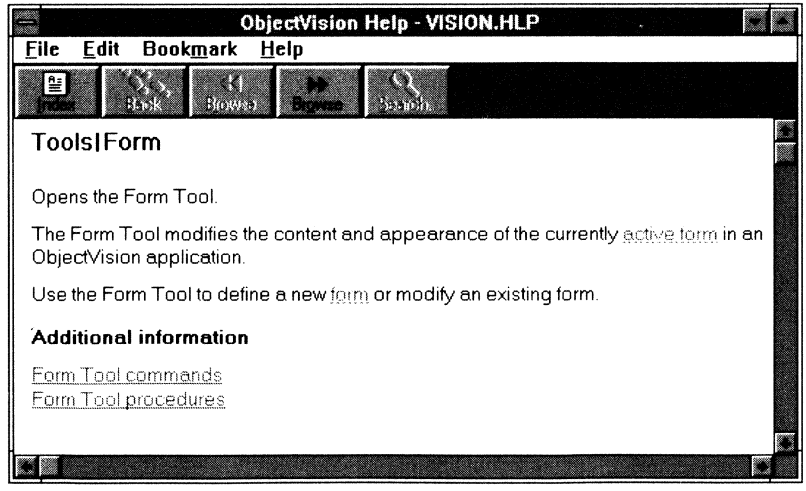
Using online help

There are three types of online help: general, context-sensitive, and object-specific.

General help

You can get general ObjectVision help by selecting Help from the menu bar and choosing a command. If you don't know which command to choose, try Index. After you select Help | Index, you can see more information on a topic that is green and has a solid underline by clicking a topic name. For example, click the Form Tool topic. The ObjectVision Help window appears, as shown in Figure 3.12.

Figure 3.12
ObjectVision Help window



While you're reading the Tools | Form topic, notice the words that are green and have a dashed underline. These are *glossary words*, which have definitions that appear when you position the pointer on them, then press and hold down the mouse button.

ObjectVision Help window

In the ObjectVision Help window, there are five buttons in the bar below the menu bar (Figure 3.12). You can press these buttons to find the information you're looking for:

- **Index** returns you to the index of topics. Additionally, if you've selected Windows Help, this returns you to ObjectVision help.
- **Back** goes back to the topic you previously viewed. A complete log of your path through the help system is kept.
- **The two Browse buttons** move backward (left-pointing triangles) and forward (right-pointing triangles) within a series of related topics. For example, you can browse from the beginning to the end of the @functions series.
- **Search** gives you a dynamic index for searching the help files for specific information you need.

If you need more information about the Windows Help system, select Help from the Help window's menu bar.

To close the ObjectVision Help window, double-click the Control-menu box.

Tip: If your computer has enough available memory, you can run both ObjectVision and ObjectVision Help at the same time. This can be helpful when you're creating applications for the first time. For example, you can arrange your desktop to display ObjectVision and the @functions help if you're creating value or event trees.

Context-sensitive help

If you need help with a ObjectVision menu command, highlight the command with the keyboard, then press *F1*.

For example, to get help for File | Print Form, press *Alt+F*, press ↓ until Print Form is highlighted, then press *F1* to see its help screen.

Pressing *F1* is a faster way of getting the help information you need, but you can always get the same information by selecting Help from the ObjectVision menu bar and navigating through the help system.

Object-specific help

Customized help messages can be created for ObjectVision applications when you design a form. These help messages are defined as a property for fields, columns, and buttons.

For example, as discussed on page 33, the Customer Type field has a help message.

For more information about creating help for your applications, see *ObjectVision Reference Guide*, Chapter 3, "Form design basics."

Tip: You can let a user of your application know that help is available for a field by putting a text object or a graphic symbol next to the field.

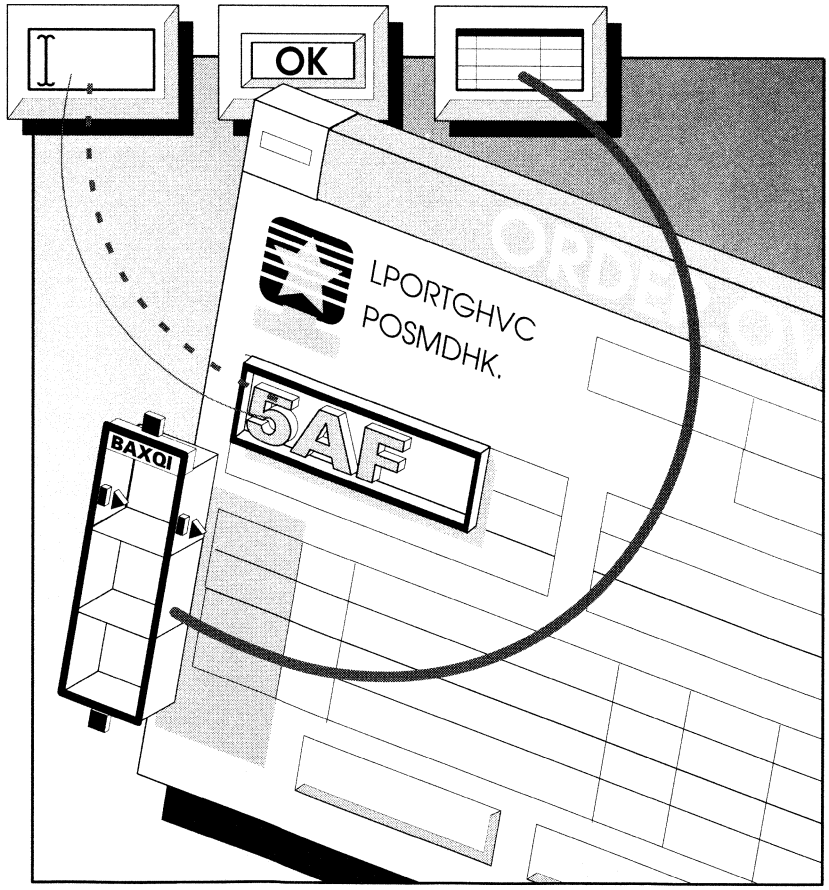
If you're filling in an application created by someone else, you might see a message unique to the selected object appear after pressing *F1*. This help message is specific to the selected object on the form.

To get help for a button, you must use the keyboard to give it the focus before pressing *F1*. To do this, press *Tab* until the button is selected. When help isn't available for the selected button, the ObjectVision Help window appears displaying the Index to online help.

C H A P T E R

4

Designing forms



Chapter goals:

- Planning the application
- Creating a new form
- Determining which objects to use
- Adding fields
- Adding buttons
- Adding table objects
- Adding text
- Adding rectangles
- Adding lines
- Adding graphics

Designing ObjectVision applications is done in three phases:

- designing the forms, which serve as the user interface
- assigning properties to the objects on the forms
- creating links between the values entered in the form and external data files, like databases

This chapter explains the first phase, designing forms.

You can complete the exercises in this chapter at any time by starting with a new file. If you started ObjectVision and didn't open a file, you are ready to create a new file. Otherwise, choose the New command from the File menu. The explanations assume you are familiar with the basic operation of the *Order* application, described in Chapter 3, "Using an ObjectVision application."

Planning the application

This section re-creates the planning process for designing the *Order* application. If you are using this tutorial to design a different application, you can follow these general planning guidelines.

There are five steps to planning the application:

1. **Develop a short, general description of the application's purpose.** For example, the description of the sample application, *Order*, is as follows: To let order entry personnel enter new orders into a company database, add new customers to the database, quickly determine the price and shipping costs, and view all items ordered by a particular company.
2. **Develop specific objectives for the application.** The objectives for the *Order* application are as follows:
   - Display all current items for an existing order.
   - If the company has never placed an order before, add the new company information to a database.
   - Based on the item ordered, supply the unit price.
   - Based on the customer type and order quantity, calculate a discount.
   - Calculate the extended price.

3. **Determine the decisions that the application will make.** Classify values into three categories: not used in any decisions, used in a decision, and calculated by the application. The categories of information used by the *Order* application are shown in the following table.

Table 4.1
Values used by the Order application

Not used in a decision	Used in a decision	Calculated by the application
Contact Name	BILL TO area fields	Date
Phone	Tax Exempt	SHIP TO area fields
Item No.	Quantity	Sales Order
Description	Customer type	Price
Tax Exempt No.	Price	Discount
Part No.	Discount	Amount
Ext.	Amount	Subtotal
	(Unseen fields)*	Tax
	Qty.	Shipping
		Total

* Represents fields which don't appear on either of *Order*'s two forms. They include Shipping Information Same as Billing Information?, Ship Co. same as Bill Co.?, Ship Dept. same as Bill Dept.?, Ship Street same as Bill Street?, Ship City same as Bill City?, Ship State same as Bill State?, and Ship Zip same as Bill Zip?

4. **Determine which actions the application will perform.** Classify the actions into two categories: directly triggered by the user and indirectly triggered by the user. The categories of events performed by *Order* are shown in Table 4.2.

Table 4.2: Categories of events in Order

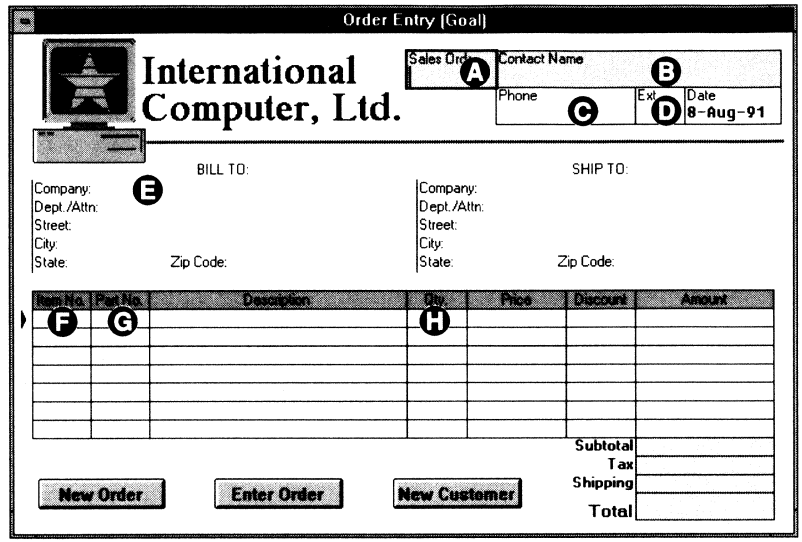
Directly triggered	Indirectly triggered	Event description
New Order		Clears both forms of all values.
Enter Order		Writes the order information to the database.
New Customer		Clears customer information and opens the Customers form.
Enter		Writes the customer information to the database.
Previous		Displays previous customer record in the database.
Clear		Clears the screen of the current record.
Next		Displays the next customer record in the database.
Delete		Erases the current record from the database.
Return to Order Entry		Opens the Order Entry form.
	Sales Order	Locates records if the order number already exists.
	Bill Co.	Locates customer information in the database.
	Item No.	Locates part information in the database.

5. **Decide the field sequence for form completion.** The sequence affects the application by guiding the user to fill out fields from

left to right and top to bottom, based on the bottom right corner of the field. Though the user can fill in the fields out of order, your application will be easier to use if you arrange the fields in the sequence the user is most likely to expect.

On the Order Entry form, for example, the user is guided to complete the fields in the order shown in Figure 4.1.

Figure 4.1
The field order for the Order application



Note that the Date field is skipped because its value is calculated automatically. Similarly, the Item No. column is selected after the BILL TO area field Bill Co., because a user cannot enter values in any other BILL TO or SHIP TO area fields.

In this tutorial, *BILL TO area* and *SHIP TO area* are used to refer to all the fields in that part of the form. For example, the BILL TO area contains the fields named Bill Co., Bill Dept., Bill City, Bill State, Bill Street, Bill Zip, and Bill Zip Ext.

After you have identified the application's goals, objectives, required fields, essential events, and field sequence, you are ready to create a form.

Creating forms

The *user interface* for your application is one or more forms in which users enter values. The *Order* application has two forms:

Order Entry and Customers. Later, you'll establish links between the fields on the forms and external data files.

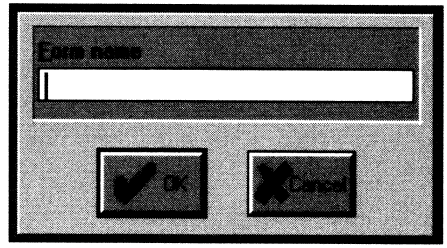
To create a form,

1. Start ObjectVision.
2. Click the Maximize button located in the top right corner of the window.

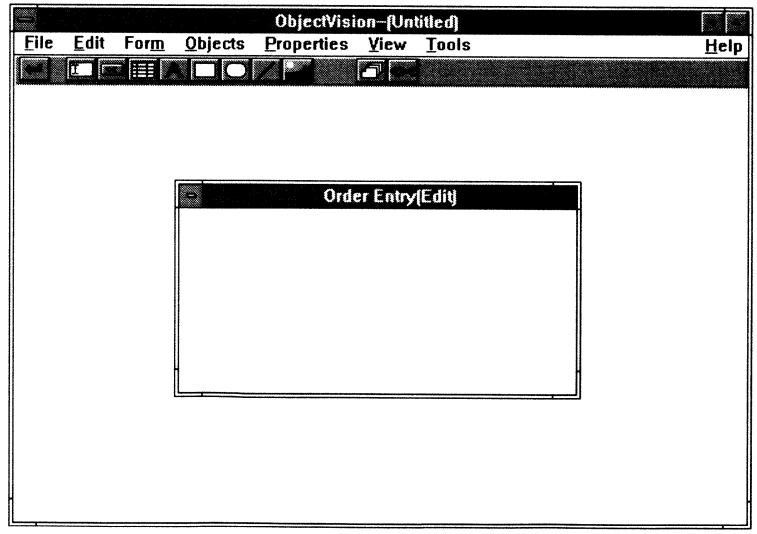
   This button (shown at left) enlarges the ObjectVision window so that it fills the entire screen. You should maximize the window whenever you start ObjectVision in this tutorial (if it isn't already maximized).
3. Choose Tools | Form. The Form Name dialog box appears, as shown in Figure 4.2.

Figure 4.2
The Form Name dialog box



4. Type `Order Entry`, then click OK or press *Enter*.
   The Order Entry form appears, as shown in Figure 4.3.

Figure 4.3
The new Order Entry form



When you edit in the Form Tool, you can use ObjectVision only to create objects and assign properties to those objects—not to enter values into fields.

You can tell when the Form Tool is active because the title of the form that appears in the application title bar is followed by the status Edit, in parentheses—for example, *Order Entry (Edit)* as shown in Figure 4.3.

At other phases of application development, the form status in parentheses can be *Goal* or *Complete*.

Changing the form size

You can reduce or enlarge the size of the form. To enlarge the Order Entry form,

1. Make the form wider by dragging its left or right borders.

   Move the pointer to the right border of the form. When the pointer changes to a double-headed arrow, drag the border to the right as far as you can, until the form border almost meets the application window border. Repeat with the left border.
2. Make the form taller by dragging its top or bottom borders.

   To make the form wider and taller at the same time, position the pointer in a corner of the form. The pointer changes to a two-headed diagonal arrow that you can use to diagonally drag the form borders.

Tip: To resize the form to exact dimensions, choose View | Ruler to display the horizontal ruler (Top), vertical ruler (Left), or both. See page 51 for more information on using the ruler.

Creating another form

The *Order* application has another form, Customers, where new customer information is entered. To create it,

1. Choose New from the Form menu.
2. Type the name for the form, `Customers`, and click OK.
3. Enlarge the form by dragging its borders so it is as tall as the maximized Form Tool window and one half of its width.

To view different forms during application design, choose Form | Select, then double-click the name of the form you want to appear

as the active form. If multiple forms are open in the Form Tool, you can also click any visible portion of a form window to make it the active window.

Determining which objects to use

A form contains any combination of the following objects, illustrated in Figure 4.4 (the Order Entry form) and in Figure 4.5 (the Customers form):

- **Fields**. Character, selection method, or numeric field types. Used for entering, selecting, or displaying values.

  Example: Sales Order, Contact Name, and Phone.

- **Buttons**. Gray beveled rectangles resembling other Windows buttons, which users click to initiate an action.

  Example: New Order, Enter Order, and New Customer.

- **Table objects**. Matrices of rows and columns designed to hold associated values.

  Example: Table1.

- **Text**. Labels on the form, such as the company name, field names, or instructions.

  Example: "International Computer, Ltd.," "BILL TO:," and "Company:."

- **Lines**. Thin or thick lines used to visually separate areas of the form.

  Example: Horizontal line directly above the BILL TO and SHIP TO areas.

- **Filled rectangles**. Shaded or colored rectangles used to fill in areas of the form that would otherwise be blank.

  Example: The gray rectangle on the Customers form which is used as a background to accent the Return to Order Entry button.

- **Rounded rectangles**. Shaded, colored rectangles with rounded corners.

  Example: White rounded rectangles surrounding BILL TO and SHIP TO areas on the Customers form.

- **Graphics**. Images created in Windows Paintbrush or other graphics applications, or images saved in the ObjectVision .OVG format.

Example: The STARLOGO.OVG image in the upper left corner of the Order Entry form.

Figure 4.4
Objects on the sample Order Entry form

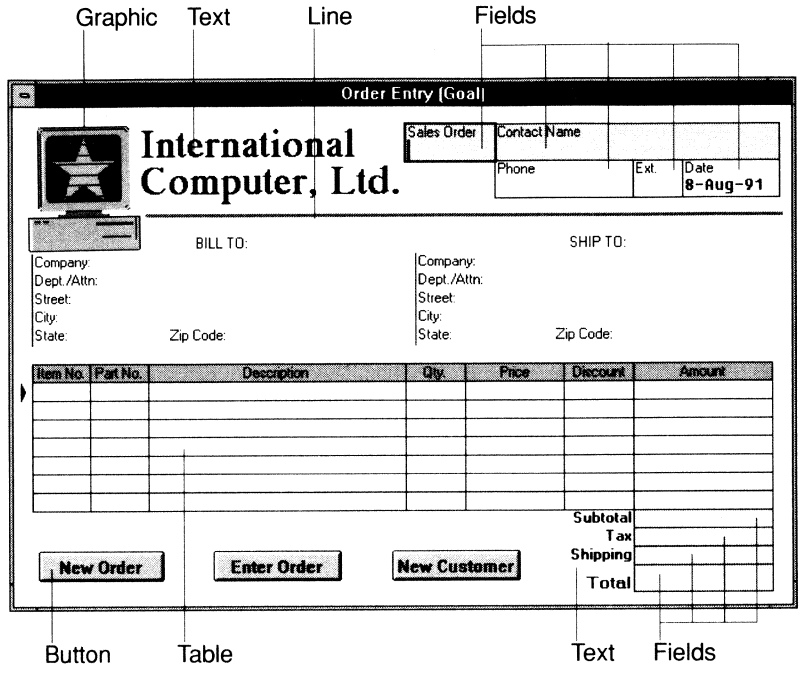
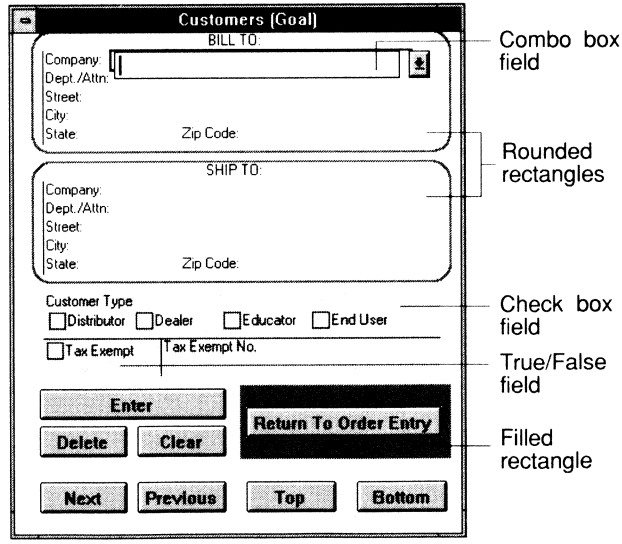


Figure 4.5
Objects on the sample Customers form



Tip: If you're creating your own application, you might find it helpful to sketch the form—including appropriate objects—before proceeding.

Selecting ruler and grid options

ObjectVision provides rulers and grids to help you position and size objects accurately. The rulers can appear at the top and left of the form, at the top only, at the left only, or not at all (the default). The ruler measurements represent inches, centimeters, or characters (the default).

To change the ruler option,

1. Choose Ruler from the View menu.
2. Check Top and Left to indicate that you want the ruler to appear at both the top and the left edges of the form.
3. Click OK.

To change the grid option,

1. Choose Grid from the View menu.
2. Check an option to set the grid to Coarse, Medium, or Fine. (Coarse is the default grid setting.)
3. Click OK.

Your grid setting doesn't appear onscreen, but it does determine how precisely you can position and size objects on the form. The options are

- **Coarse** (default) lets you size objects in increments of 1 character by 1 line.
- **Medium** lets you size objects in increments of ½ character by ½ line.
- **Fine** lets you size objects in increments of ¼ character by ¼ line.

If Coarse is selected and you attempt to create a field with dimensions of 10 ¼ characters by 1 ¾ lines, ObjectVision "snaps" the field to 10 characters by 2 lines.

Basically, the Coarse grid makes it easier to align objects; there's also less chance of accidentally moving an object. The Medium and Fine grids let you have more precise control over sizing and placing objects.

However, if you move an object created using a finer grid, you might not be able to return the object to its original location unless you first reset the grid.

Tip: It's easiest to use the default Coarse grid when you first begin creating a form and its objects. Medium and Fine grids are best for making increasingly fine size adjustments in the later stages of designing an application.

Adding fields

Fields display values from one of three sources: the user, the application, or a link to an external data file (such as a Paradox or dBASE table) or another Windows application.

To add a field to a form, you create the field and define its position on the form. To refine the appearance of your form, you define the fields' properties, explained in Chapter 5, "Defining properties."

Understanding field relationships

If you use ObjectVision to *create a new* external database table, consider the relationship between the fields you add to your forms and the database fields. The field order and field size you define in ObjectVision determines the default index field and default field sizes for the external data file. However, this database definition can be changed at the time you create the table.

- **Field order.** Determine the order in which you want users to enter values. When a user presses *Enter*, the application moves the insertion point from field to field the same way you would read a page: from left to right and from top to bottom, based on the position of the lower right corner of each field.

  In the *Order* application, for example, the application guides the user to complete the fields on the Order Entry form in the following order: Sales Order, Contact Name, Phone, Ext., Bill Co., Item No., Qty., New Order, Enter Order, and New Customer.

  The application either calculates the values for the other fields based on the logic in their value trees or delivers values by triggering links.

- **Field size.** When you use ObjectVision to create an external database file, the size of the fields on a form determines the

default field length in the external data file. Recall that the table definition can be changed before the table is created.

For example, if the field can hold four characters in the font you assigned, the default field definition for that database field will be a field length of four.

Note that two fields of the same size can hold a different number of characters if they are assigned different fonts—for example, a field with the default 12-point Courier value font can hold more characters than a same-size field assigned a larger, proportional font.

Assigning font properties to objects is covered on page 75 in Chapter 5, "Defining properties."

To add the Sales Order field to the Order Entry form,

1. Make the Order Entry form the active window. If the Order Entry form isn't displayed on your screen, choose Form | Select, select Order Entry from the list of form names, and then click OK. As a shortcut, double-click the form name Order Entry.
2. If you're not in the Form Tool, choose Tools | Form to open it.

3. Click Field (shown at left) on the Object bar. The Field Name dialog box appears with the default name *Field1*.
4. Type the name of the field as you want it to appear on the label—for the *Order* application you would type `Sales Order`—then press *Enter*. The field crosshair pointer appears.
5. Position the pointer where you want one corner of the field to appear, then drag to size the field.

   If both rulers are displayed, dark gray areas on the rulers show the height and width of the field as you drag. A dashed border appears around the field with black handles at each corner. The dotted border is replaced by a solid line when you create another field.

   When you assign properties to the field (Chapter 5), you can turn off one or more visible boundaries of the border, or change its line width or color.

In the following sections, moving, sizing, copying, and pasting of fields is discussed. After adding all other objects to the form, you will place the remaining fields on the form, as listed in Table 4.4 on page 68.

Moving fields

To move the field, position the pointer inside the field or on its border (but *not* on one of its *handles*), then drag it to the new position. Handles are small black squares that display in the Form Tool at the corners of a selected object; dragging a handle resizes the object.

Tip: If you accidentally move an object when selecting it, press *Alt+Backspace* or choose Edit | Undo Move to return it to its original position.

Resizing fields

To resize the field, position the pointer on one of the handles. After the pointer changes to a black square, drag it to reduce or enlarge the object's size.

Copying fields

You can copy a field and its properties by choosing the Copy and Paste commands from the Edit menu. It saves time to copy fields when their properties are the same.

For example, the BILL TO and SHIP TO area fields on the Order Entry form are the same size. You can use the BILL TO area fields to quickly create the SHIP TO area fields on the Order Entry form.

To copy the fields in the BILL TO area to the SHIP TO area,

1. Create the fields in the BILL TO area: Bill Co., Bill Dept., Bill Street, Bill City, Bill State, Bill Zip, and Bill Zip Ext.

   Note that the completed *Order* application reads "Company:," "Dept./Attn:," "Street:," "City:," "State:," and "Zip Code:," but these are *text objects*, not the field names. Creating these text objects is discussed on page 62.

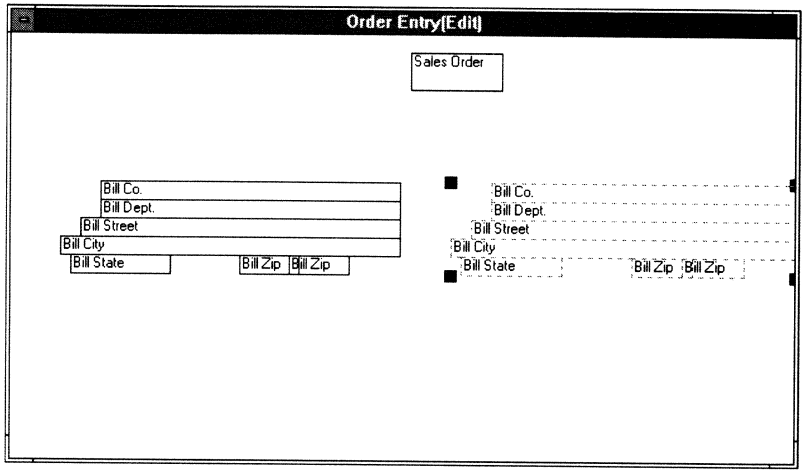
2. Select all the BILL TO area fields you just created. To select multiple fields, hold down *Ctrl* as you click each field.

   Tip: To select multiple, contiguous fields, click the Bill Co. field. While still holding down *Shift*, click the Bill Zip Ext. field. This selects all objects on the form between the first and the last selected object.

3. Choose Edit | Copy or press the shortcut *Ctrl+Ins*. The field and its properties are copied to the Clipboard, a holding area for the last object copied.
4. Choose Edit | Paste or press *Shift+Ins*. A dotted border the same size as the original fields appears on the form. Position it where you want the copies to appear, then click. The copies appear.

The Order Entry form should now look like Figure 4.6.

Figure 4.6
The Order Entry form with pasted fields



Redefining copied fields

The SHIP TO area fields are presently exact copies of the BILL TO area fields—any value the user enters in one will appear in the other. In *Order*, you need flexibility; the billing information could be different from the shipping information.

Now give the copied fields their own identities,

1. Select the field you want to redefine—in this case, the copy of the BILL TO area Bill Co. field you pasted in the SHIP TO area.

2. To inspect the SHIP TO area Bill Co. field properties, click it with the right mouse button.
3. Choose Field. The Field Name dialog box appears, containing the name Field1.
4. Replace the default name Field1 by typing the new field name for the field—`Ship Co.`— then click OK.

Caution! Be sure to choose Field from the property inspector instead of Name/Text. The Field command creates a new field. The Name/Text command simply changes the name of the original field as well as the copy.

5. Redefine the fields you pasted in the SHIP TO area for each of the remaining copies. Replace the word `Bill` with the word `Ship` at the beginning of each field name. The Order Entry form now has the complete BILL TO and SHIP TO area fields.

It also saves time to copy fields when they appear in more than one location in the application stack. For example, the BILL TO and SHIP TO areas on the Order Entry and Customers form are the same.

Use the copy and paste technique to copy the BILL TO and SHIP TO area fields from the Order Entry form to the Customers form. After copying the fields on the Order Entry form, choose Form | Select, double-click Customers, then paste the fields onto the Customers form.

If you are re-creating the Order Entry form, return to it and create the remaining fields on the Order Entry form using Field on the Object bar or by copying an existing field, resizing it, then redefining it as described in this section.

When you are done, the Order Entry form should look like Figure 4.7.

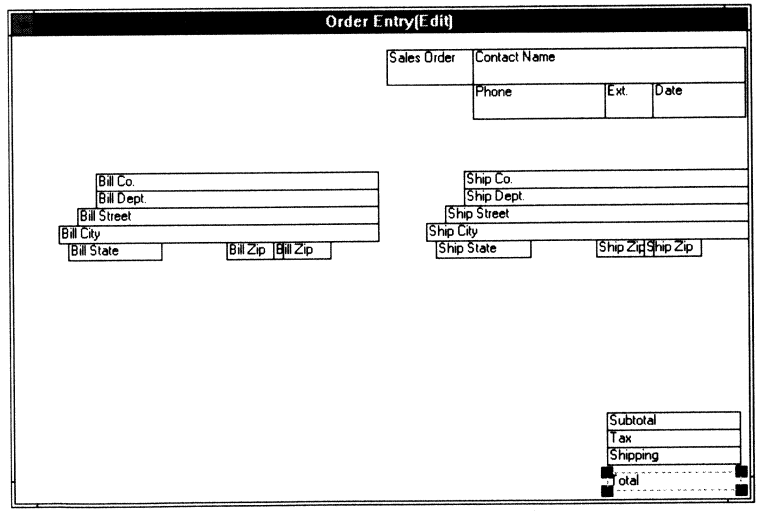
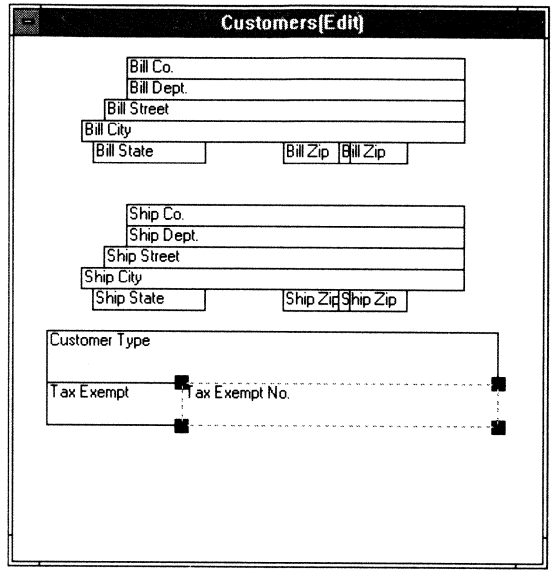


Figure 4.7
The Order Entry form with its fields

Select the Customers form and add the remaining fields to it. When you finish adding fields to the Customers form it should look like Figure 4.8.

Figure 4.8
The Customers form with its fields



Adding buttons

Buttons are objects the user clicks to initiate a single action or multiple actions.

In the *Order* application, for example, users click the New Customer button to display another form for adding new customer information. They click the Enter Order button to add the values from the new order to the external data file.

Button objects are assigned an event tree property, which defines the action the application performs when the button is clicked. The process of creating event trees is described in Chapter 7, "Creating event trees."

To add buttons to the Order Entry form in *Order*,

1. Select the Order Entry form by choosing Form | Select and double-clicking `Order Entry`.

2. Click Button (shown at left) on the Object bar. The Button Name dialog box appears displaying a default button name.

3. Replace the default button name in the Button Name text box, by typing `New Customer`, then click OK.
4. Position the button crosshair pointer where you want one corner of the button to appear, then drag to the diagonally opposite corner. The New Customer button now appears.

   Note that the label font for buttons is always 10-point System.
5. Now repeat steps 1-3 to add the Enter Order and New Order buttons to the Order Entry form and the Return to Order Entry button to the Customers form.

   The remaining buttons on the Customers form—Enter, Next, Previous, Delete, Clear, Top, and Bottom—will be added *automatically* when you create a link in Chapter 8, "Creating links."

Adding table objects

Table objects display values in a matrix of rows and columns. They are a logical way for users to enter and view associated values.

Some examples of values that can be presented in an ObjectVision table object are a part description, part number, and price; order number and date; a liquid and its freezing and boiling temperatures; or students' names, heights, weights, and birth dates.

The *Order* application has a table object for viewing and entering order information (see Figure 4.4 on page 50).

Follow these steps to create the table object:

1. Select the Order Entry form by choosing Form | Select and double-clicking Order Entry.

2. Click Table (shown at left) on the Object bar. The default name Table1 appears in the Table Name dialog box.

   Ordinarily you would type the table name you want to appear at the top of the table object. But because the table name is hidden in the *Order* application, the name doesn't matter.
3. Click OK to use the name Table1. The table crosshair pointer appears.
4. Create the first column by positioning the pointer at one corner of the table, then dragging to the diagonally opposite corner.

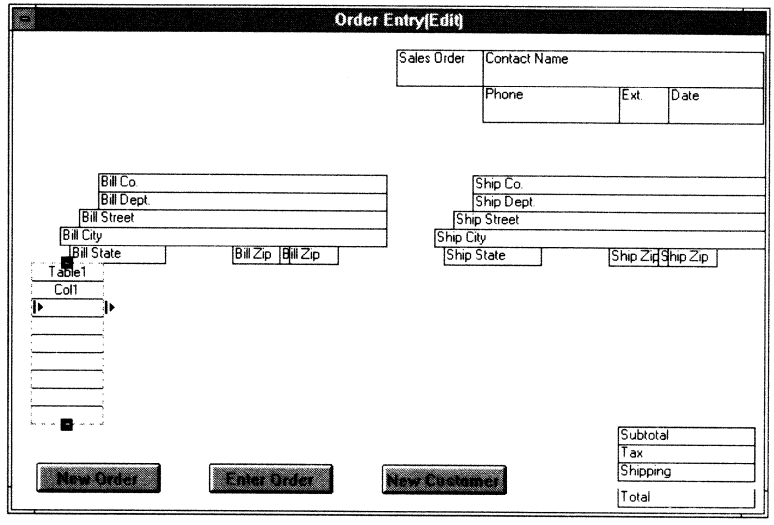
Don't drag the table out to its full size—you're only creating a *single column* now.

5. Position the pointer over the square handle at the bottom of the table object, and drag it to create the seven rows.

   Note the numbers that appear inside the upper left corner of the column. These numbers indicate the number of rows in the column that will contain values.

   The first column appears as shown in Figure 4.9.

Figure 4.9
The first table column

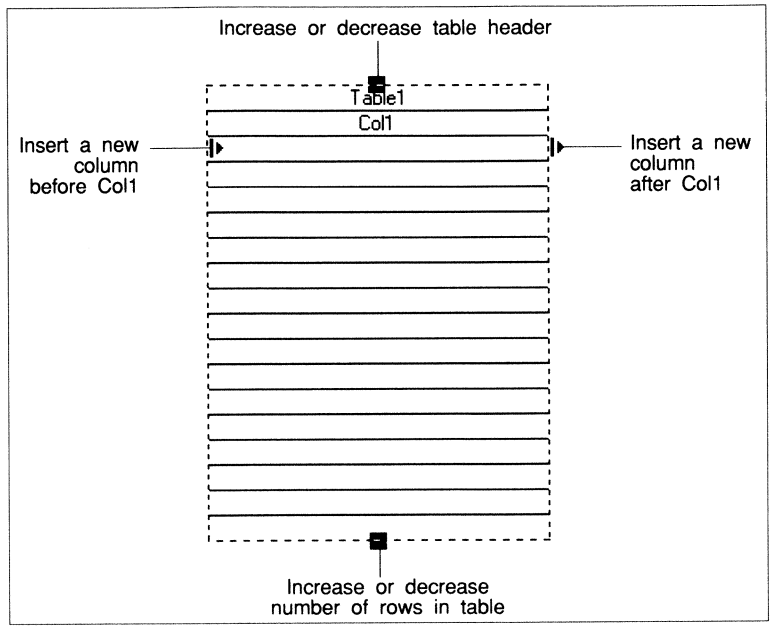


6. To create the next column, position the pointer on the right-pointing triangle located at the right edge of the table object.

   When the pointer is positioned properly it will become a right-pointing triangle, also.

   In Figure 4.10, the different table object handles are explained.

Figure 4.10
Handles for a table object



7. Drag the arrow to the right. As you drag, a green dotted line shows the width of the new column.
8. Repeat steps 6 and 7 until you've added six more columns to the table, for a total of seven.
9. To hide the table name, Table1, select the table by clicking the table header area. Then drag the top table handle down to the top of the column headers.
10. If necessary, adjust the dimensions of the table:
    - Select the table by clicking the table header.
    - **Tip:** To select a table object which has a hidden table header, *click twice* on any cell (or click once on the active cell). This is different than double-clicking—wait longer between the two clicks. When the table handles appear, you can drag the top table handle up to display the table header again.
    - To change the number of rows in the table, select the handle at the bottom of the table object and drag it up or down.
11. If necessary, adjust the dimensions of the columns:
    - Select a cell by clicking any cell below the column header.
    - To make a column wider, click any cell to select it, then drag any cell handle in that column *away* from the *horizontal center* of the cell.

- To make a column narrower, click any cell to select it, then drag any cell handle in that column *toward* the *horizontal center* of the cell.
- To make all cells in the table object taller, click any cell below the column header, then drag any cell handle *away* from the *vertical center* of the cell.
- To make all cells shorter, click any cell below the column header, then drag any cell handle *toward* the *vertical center* of the cell.
- To delete a column, click any cell within that column to select it, then resize any cell so that the cell's left and right borders meet.
- To rearrange columns, click the column header to select the entire column, then drag the column to a new position within the table object.

Labeling columns

If the column names of your table objects correspond to the field names in the external data file (to which you will be linking the application) it's easier to connect the link.

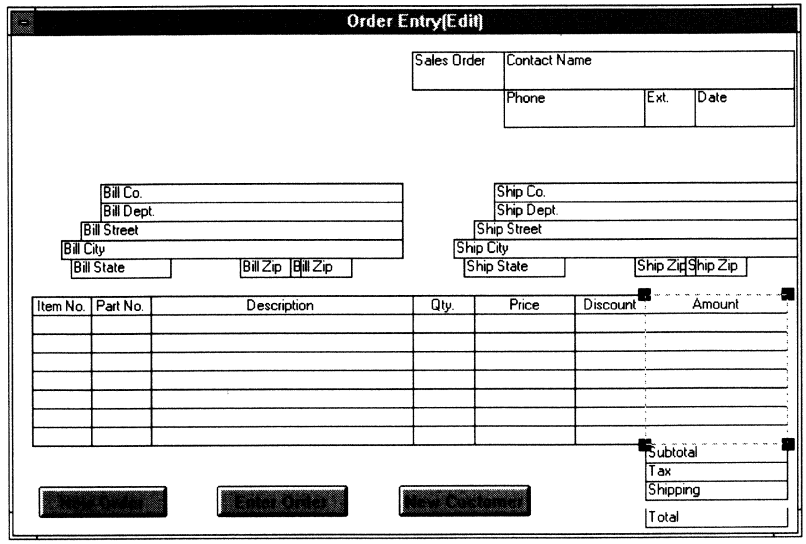
Notice that column names appear in all ObjectVision lists that contain field names. Column names must be unique.

Follow these steps to label the columns of the Table1 table object:

1. Select the column by clicking the column label, such as *Col1*.

2. Position the pointer inside the column, then click the right mouse button to inspect that column's properties. Choose Name/Text.
3. In the Field Name dialog box, type the name of the column—for example, `Item No.`—then click OK. The new column name appears on the table object.
4. Repeat these steps to label the other columns as shown in Figure 4.11.

Figure 4.11
The completed column labels



Tip: You can hide a column name by clicking the right mouse button to inspect its properties, choosing Field Type, and unchecking Display Field Name.

Adding text

Text can be used on ObjectVision forms to label forms and objects or to provide onscreen instructions that are a permanent part of the form. It's also possible to define an event tree for a text object or a graphic object, so the user can click that object to trigger an action, like displaying another form. Usually, however, buttons are used for this purpose.

Text objects are different from labels, which are the names of fields and appear within the field boundary. You can indicate a field's name with either a text object or a label, depending on which option looks better for your form.

Tip: The advantage of using labels is that they don't require an extra step. The advantage of using text objects is that you can place them wherever you want, and they are never selected during guided completion.

Table 4.3 lists the text objects for the Order Entry form and the Customers form.

Table 4.3
Text objects in the Order application

Order Entry form	Customers form
International Computer, Ltd.	
BILL TO: Company: Dept./Attn.: Street: City: State: Zip Code:	BILL TO: Company: Dept./Attn.: Street: City: State: Zip Code:
SHIP TO: Company: Dept./Attn.: Street: City: State: Zip Code:	SHIP TO: Company: Dept./Attn.: Street: City: State: Zip Code:
Subtotal Tax Shipping Total	

The *International Computer, Ltd.* company name is made of two separate text objects. This lets you position the two lines closely. (You could also create a single text object, sizing it so that the words appeared on two lines, but the large font size would result in a large vertical space between the two lines.)

To add the International text object to the Order Entry form,

1. Click Text (shown at left) on the Object bar. The Text Value dialog box appears.
2. Type the text as you want it to appear on the form: `International`, then click OK or press *Enter* when the text is correct. A text crosshair pointer appears.
3. Position the pointer where you want the text object to appear, then click to create it.

   The text value appears as you typed it, in 8-point Helv and surrounded by a border. Instructions for changing the font and border are in Chapter 5, "Defining properties."

   If the entire text string doesn't appear, the object boundary is too small. If necessary, change the size of the text object by dragging one of its handles.

4. Repeat steps 1 through 3, substituting `Computer, Ltd.` for `International`, to create the rest of the company name on the form. Place this text object below the International text object.

Tip: To create text objects with multiple lines of text, press *Ctrl+Enter* to start a new line.

Now, create the remaining text objects on the Order Entry and Customers forms. You'll see how to add a left border to the text objects in the SHIP TO and BILL TO areas in Chapter 5, "Defining properties."

Tip: You can use the Copy and Paste functions to save time. After creating the text objects in the BILL TO area, select them by holding down *Ctrl* and clicking each object. Then choose Edit | Copy. Choose Edit | Paste, move the crosshair pointer to position the fields, then click. To change the copy's name from BILL TO to SHIP TO, position the pointer on the copy of the BILL TO text object and click the right mouse button to inspect its properties. Choose Name/Text. In the Text Object dialog box, type SHIP TO, then click OK. Paste the text objects again onto the Customers form.

Adding lines

Lines can enhance the appearance of a form by separating or emphasizing one or more objects.

Use the steps that follow to create the horizontal line above the BILL TO and SHIP TO areas on the Order Entry form.

1. Choose Form | Select and open the Order Entry form.

2. Click Line (shown at left) on the Object bar. The line crosshair pointer appears.
3. Position the pointer near the top and to the left of the BILL TO text object, then drag the line to the right across the form.

   A horizontal line now appears. You'll change the width and color properties of the line in Chapter 5, "Defining properties."

Adding rectangles

A form's appearance can often be enhanced by outlining important areas or by filling unused space with square or rounded rectangles.

The Customers form contains rounded rectangles delineating the BILL TO and SHIP TO areas. It also contains a filled rectangle which accents the Return to Order Entry button.

To add rounded rectangles around the BILL TO and SHIP TO areas,

1. Choose Form | Select and open the Customers form.
2. Click Rounded Rectangle (shown at left) on the Object bar. The rounded rectangle crosshair pointer appears.
3. Position the pointer where you want one corner of the rectangle to appear, then drag to the opposite corner. A shaded, rounded rectangle appears.

   You'll want to change the fill pattern property of the rectangle from light gray to white. Properties are covered in Chapter 5, but since the gray pattern makes it difficult to see the text, it makes sense to change it now:

   - With the pointer on the rectangle, click the *right* mouse button to inspect the properties for that rectangle.
   - Choose Fill Pattern.
   - From the Fill Pattern dialog box, select the white fill pattern in the middle column, then click OK. The rectangle is now white.

   To create a rounded rectangle for the SHIP TO area, it's easier to paste a copy of the rectangle you just created.
4. So, select the rectangle, copy it, paste it, and position it around the SHIP TO area.

   (To remove the dashed selection borders, click someplace else on the form.) The form now contains two rounded rectangles.

Adding graphics

ObjectVision forms can include any of the following graphic files:

- **Windows bitmap file.** A Windows bitmap file (.BMP) is a graphic image created in Paintbrush or other Windows graphics applications.
- **Windows metafile.** When other Windows graphics applications don't save files in the .BMP format, you can copy the graphic to the Clipboard. Windows automatically converts the file to the .WMF format. The sample .OVG file, STARLOGO.OVG, was created this way.
- **ObjectVision graphic file.** Before any graphic can be placed on an ObjectVision form, you must give it a file name. The default extension for ObjectVision graphics files is .OVG.

Images can be color or monochrome—ObjectVision prints the images according to your printer's capabilities. A color image will print as a monochrome image on a monochrome printer.

To add a .WMF or .BMP graphic to an ObjectVision form, copy the graphic to the Clipboard, then choose Objects | Graphic in the Form Tool to paste it into ObjectVision. To add a file that has already been converted to .OVG file format, choose Objects | Graphic, and then specify the name of the file you want to open.

The *Order* application includes a graphic in the top left corner. To place this .BMP file as a graphic on the Order Entry form,

1. Minimize your ObjectVision application window by clicking the Minimize button.
2. Open the Paintbrush application, which is in the Program Manager Accessories group.
3. In Paintbrush, choose Open from the File menu and type the complete path name for the STARLOGO.BMP file copied to your ObjectVision directory during installation. For example, if you installed ObjectVision in C:\VISION, and the sample files in C:\VISION\SAMPLE you would type `C:\VISION\SAMPLE\STARLOGO.BMP` in the File Name text box. Then click OK.
4. From the Tool palette on the left side of the window, click the Selection Tool, which shows a scissors and a rectangle.

5. Drag the selection rubber band around the entire outlined image.
6. Choose Edit | Copy. This copies the image to the Windows Clipboard.
7. Exit Paintbrush without saving changes to the file.
8. Open ObjectVision by clicking the minimized ObjectVision icon and choosing Restore from the Control menu. Make sure you're still in the Form Tool.

9. Click Graphic (shown at left) on the Object bar. A dialog box appears with the message

   ```
   Use the Clipboard graphic (YES) or an existing ObjectVision
   graphic (NO)?
   ```

10. Click Yes to indicate you want to paste the graphic from the Clipboard. The Save Clipboard Graphic As dialog box appears where you specify the .OVG file name you want to give the Clipboard graphic.
11. Type `STAR.OVG`, then press *Enter*. Don't use the name *STARLOGO.OVG* or you'll overwrite the file by that name used in the completed *Order* application.
12. The graphic crosshair pointer appears, surrounded by a dotted border showing the size of the graphic. Move the border to the appropriate position—the upper left of the form for the *Order* application—then click. The graphic appears.
13. To size the graphic, drag its handles.

Adding the remaining fields

Add the remaining fields to the Order Entry and Customers form as shown in Table 4.4.

The Customer Type field in the completed *Order* application has been created using the medium grid and is 2½ characters high. If you do change your grid, be sure to return it to Coarse after creating Customer Type.

Some fields in *Order* don't appear on either of its two forms. These unseen fields are created automatically when you create value trees that reference them in Chapter 6, "Creating value trees." These fields automatically display on a Scratchpad form (with the status *Prompt*) when the user needs to fill in a value.

The unseen fields in *Order* include: Shipping Information Same as Billing Information?, Ship Co. same as Bill Co.?, Ship Dept. same as Bill Dept.?, Ship Street same as Bill Street?, Ship City same as Bill City?, Ship State same as Bill State?, and Ship Zip same as Bill Zip?

Table 4.4
Fields in the Order application

Order Entry form	Customers form
Sales Order	Customer Type
Contact Nam	Tax Exempte
Phone	Tax Exempt No.
Ext.	
Date	
Subtotal	
Tax	
Shipping	
Total	

You have now placed all the objects in the Order Entry form. To see how the application behaves, close the Form Tool by clicking Close Tool, shown at left, located at the left edge of the Object bar. Choose Edit | Clear All to clear any values from the form and reposition the pointer in the first field on the form.

Then try typing values in the fields.

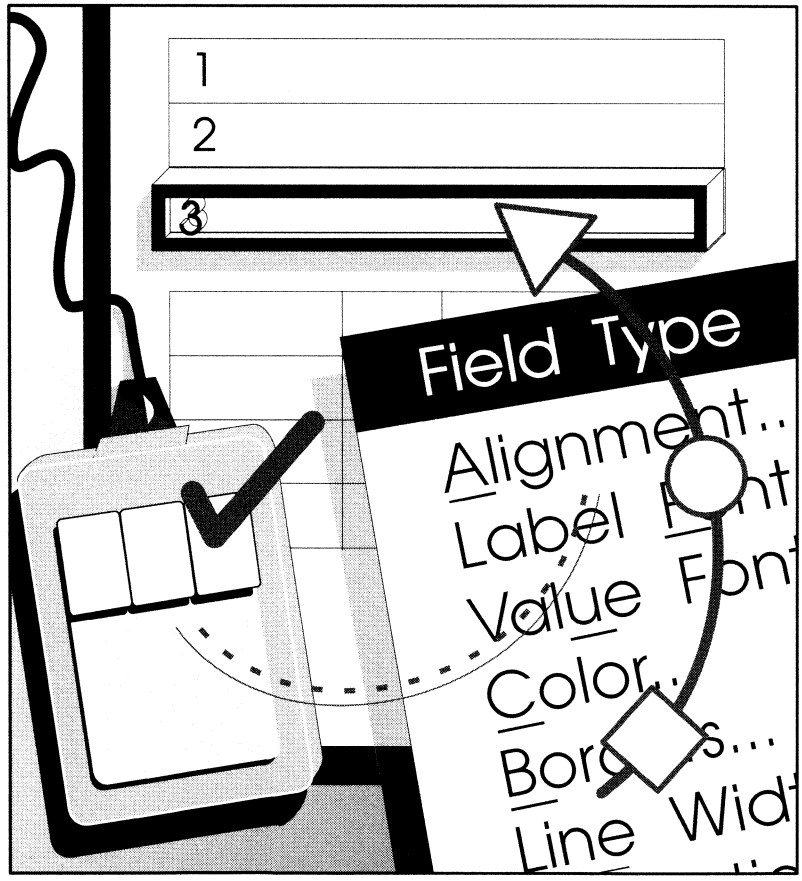
- When you press *Enter*, the application moves the insertion point to the next field that requires a value from the user—from left to right, top to bottom.
- You can still enter any value in the phone number field, even letters. You'll restrict the field to numeric entries of the appropriate length when you assign the Picture field type to this field.
- Clicking the buttons has no effect. You'll assign event trees to button objects in Chapter 7, "Creating event trees."

Chapter 5, "Defining properties," describes the next phase of application development.

C H A P T E R

5

Defining properties



Chapter goals:

- Assigning a field type
- Defining font characteristics
- Modifying borders
- Selecting line width
- Applying color
- Adding help text
- Adding scroll bars
- Defining property defaults

Every object, form, and stack of forms has a set of *properties*. Properties include object names, display attributes, data formats, font characteristics, borders, protection, help text, value trees, and event trees.

This chapter explains how to assign properties to the objects in the *Order* application. Chapters 6 and 7 explain how to assign a special class of properties: value trees and event trees.

What you need to complete this chapter:
You can complete the exercises in this chapter at any time by starting with the file ORDER5.OVD. The explanations assume you are familiar with the objects introduced in Chapter 4, "Designing forms."

Assigning field types

The field type controls which typed *character* values are accepted, how *numeric* values are displayed after the user moves to the next field, and the method of *selection* when the value must come from a predefined list. You can select only one field type for each field.

Consider the database when designing a form.

If you're using ObjectVision to create a database table, the field type you assign determines the default field type in the database table definition.

For example, if you assign the General field type to a field, the default Paradox field will be an Alphanumeric field, while the default dBASE field type will be Character.

For a list of ObjectVision field types and their associated database field definitions, see Chapter 1, "Using ObjectVision applications," in the *ObjectVision Reference Guide.*

Character fields

Select a character field type for fields that can contain both letters and numbers. Following is a list of Character field types that control the way typed data is accepted and displayed:

- **General** is the default field type used for numbers and letters.

  When numeric values are entered, they display with the minimum number of decimal places required to express the number; for example, the typed value `4.10` is displayed as 4.1. Similarly, if a comma is entered to separate thousands from

hundreds, such as 1,000, the value is converted to the numeric form, 1000.

- **Alphanumeric** is similar to the general field type, except that no value conversion takes place. Entered values display exactly as the user typed them.
- **Picture** restricts values to those that match a template showing letters, numbers, and literal characters such as a hyphen (-).

  For example, the picture string for the Phone Number field is `(###)###-####`. For the State field, it is `&&`, representing any two capital letters (if lowercase letters are entered, they are converted to uppercase).

For more on picture strings, see Chapter 1, "Using ObjectVision applications," in the *ObjectVision Reference Guide*.

- **Scrolling** is used for fields that hold more characters than can display in the field at once. As the user types in a scrolling field, the text automatically wraps and scrolls upward. A column in a table object cannot be assigned this field type.

Tip: Remember that the linked database field has limits for the number of characters it can hold. Because scrolling fields can exceed the limits of database fields, it's best to use them for displaying data, not entering it.

Numeric fields

Select a numeric field type for fields that can accept numeric values only. Following is a list of numeric field types that control the way typed values are accepted and displayed:

- **Fixed** displays the number of decimal places (0-15) you specify.
- **Percent** displays typed numeric values as percentages. For example, 1.0 is displayed as 100.00% and .415 is displayed as 41.50%.
- **Financial** is used for numeric values only. It inserts the thousands-separator character (a comma in the U.S.) where appropriate, and encloses negative values in parentheses. The thousands-separator character is determined from the International settings of your Windows Control Panel.
- **Currency** places a currency symbol before numeric values, inserts the currency separator character where appropriate, and encloses negative values in parentheses.

  The currency symbol and format are determined by the settings in your Control Panel. For example, if you open the Inter-

national icon and set Country to France, the currency symbol is set to the Franc (F), and the currency format is 1,22 F.

- **Date/Time** affects how the system displays the date and time in a field. ObjectVision offers several options for various countries' standards and also supports the Windows International settings.

To assign field types for the fields or columns on the Order Entry form, follow these steps:

1. Choose Tools | Form if the Form Tool isn't open already.
2. Select the first field to which you want to assign display format properties—say, the Sales Order field.

   To assign a field type to a column, select it by clicking the column label.

3. Click the right mouse button to display the property inspector, then choose Field Type.

   Notice that General—the default field type—is selected. On the Order Entry form, it isn't necessary to change the field type for the following fields: Sales Order, Contact Name, Company, Dept/Attn., Street, City, Part No., Description, Quantity, Discount.

   On the Customers form, it isn't necessary to change the field type of the Tax Exempt No. field.
4. Assign field types to the remaining fields, as shown in Table 5.1.

Table 5.1
Field types for the fields in Order

Field	Field type	Example
Phone	Picture	(###) ###-####
Ext.	Picture	####
Date	Time/Date	13-Aug-91
State	Picture	&&
Zip Code	Picture	#####
Zip Ext.	Picture	####
Price	Financial (2 decimal places)	
Discount	Percent	
Amount	Financial (2 decimal places)	
Subtotal	Financial (2 decimal places)	
Tax	Financial (2 decimal places)	
Shipping	Financial (2 decimal places)	
Total	Financial (2 decimal places)	

Tip: You can change the format of multiple fields at one time. First, select the fields (using *Shift*+click or *Ctrl*+click), then click the right mouse button and choose another field type. Or, after changing the format of a field, select multiple fields, then press *F4*. This repeats the previously assigned property for the currently selected fields.

Selection methods

Another set of field types controls the selection method—that is, how users select from a predefined set of value options. Note that none of the selection method field types can be assigned to columns.

The Customers form has three fields with selection methods: *Combo Box* for the Bill Co. field, *True/False* for the Tax Exempt field, and *Check Boxes* for the Customer Type field.

Selection method field types include

- **Selection List.** When a selection list field is selected during form completion, ObjectVision lists the values the user can select. Once the user selects a value, ObjectVision displays that field value in the general format. The field must be wide enough to hold all the characters in the longest option.
- **Check Boxes.** ObjectVision inserts a check box before each choice. The user selects a value by checking the box next to the desired value. Only one value can be selected.
- **Radio Buttons.** ObjectVision inserts a round radio button before each choice. The user selects a single value by clicking the button next to the desired value.

  Radio buttons are just like check boxes, except round objects are placed next to the value options instead of square check boxes.

Tip: If a field is not large enough for all its check boxes or radio buttons, the display format changes automatically to the default selection method field type, Selection List. Therefore, if check boxes or radio buttons do not appear when you think they should, try enlarging the field.

- **True/False.** ObjectVision inserts a single check box before the name of a field. The user checks the box to indicate a *Yes* value, or leaves it unchecked to indicate a *No* value.

Tip: You can place several True/False fields next to each other on a form and they will look like check boxes that let the user select multiple options.

- **Combo Box.** When a user selects a combo box list field, a combo box button (shown at left) appears to the right of the field. Clicking the combo box button displays a selection list.

  What makes a combo box list different from a selection list is that it gives users the option of typing an unlisted value into the blank text box at the top of the combo box list.

Assign selection method properties to the Customer Type and Tax Exempt fields, as follows:

1. Select the Customers form.

2. With the pointer on the Customer Type field, click the right mouse button to inspect its properties.
3. Choose Field Type.
4. In the Field Type dialog box, choose Check Boxes, then click OK. The Expected List dialog box appears.

   To let the application determine the possible value options from the value tree you will create in Chapter 6, you would leave the default Automatic option selected. For now, however, type the values so you can see them on your form.
5. In the New Value text box, type `distributor`, then click Insert.

   The value appears in the Values list. Also add `"dealer"`, `"educator"`, and `"end user"`.
6. When all values are added, click OK.
7. Repeat steps 2-4, but instead assign the True/False field type to the Tax Exempt field.

   It isn't necessary to type expected values for fields with the True/False property, because the only possible values are *True* and *False*.
8. Assign the remaining selection method field types to the other fields on the forms, as listed in Table 5.2. In the Expected List dialog box for these remaining fields, leave Automatic checked. You won't see any value options for these fields yet, because the values will be read from a database after you create the links. Creating links is covered in Chapter 8, "Creating links."

Table 5.2
Selection method field types for the remaining fields in Order

Form name	Field name	Selection method
Order Entry	Bill Co.	Selection list
Customers	Bill Co.	Combo box

Hiding field names

As discussed in the previous chapter, text objects are often used to identify fields. When they are used, you can hide the field labels.

For example, in the completed *Order* application, the BILL TO and SHIP TO area fields, Subtotal, Shipping, Tax, and Total all have hidden field names. Hide the field names for these fields now by selecting them, choosing Properties | Object | Field Type, and unchecking Display Field Name in the Field Type dialog box.

Defining value font characteristics

To control the appearance of your forms, you can assign properties like value font, font size, and field value alignment.

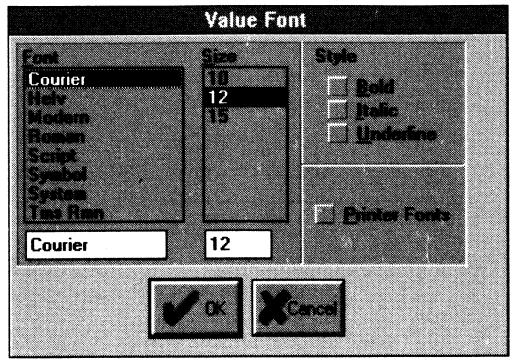
Sizing value fonts

The default value font, which is used to display values the user types, is 12-point Courier. Value options for selection method field types display in the label font. Now change the value font of the Sales Order and Contact Name fields:

1. Position the pointer on the Sales Order field and click.
2. Hold down *Ctrl* and click the Contact Name field, then release *Ctrl*. Both fields are now selected.

3. Click either of the selected objects with the right mouse button. The property inspector appears.
4. Choose Value Font. The Value Font dialog box appears as shown in Figure 5.1.

Figure 5.1
The Value Font dialog box



5. Set the value font to 8-point Helv bold, then click OK.

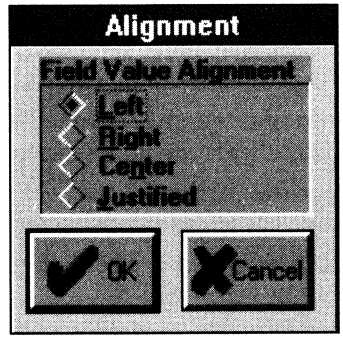
Aligning field values

All *labels* are left-aligned. Value text, however, can be left-aligned (default), right-aligned, centered, or justified.

To change the Price value so it is aligned at the right edge of the cell:

1. Position the pointer on the Price column header, then click the right mouse button to inspect its properties.
2. Choose Alignment. The Alignment dialog box appears as shown in Figure 5.2.

Figure 5.2
The Alignment dialog box



3. Check Right, then click OK. Values that display in the Price column will now be right-aligned.

Repeat steps 1-3 to right-align field values for the following fields:

- Item No.
- Part No.
- Qty.
- Price
- Discount
- Amount
- Subtotal
- Tax
- Shipping
- Total

Defining label font characteristics

To control the appearance of your forms, you can assign label font and font size properties.

Sizing label fonts

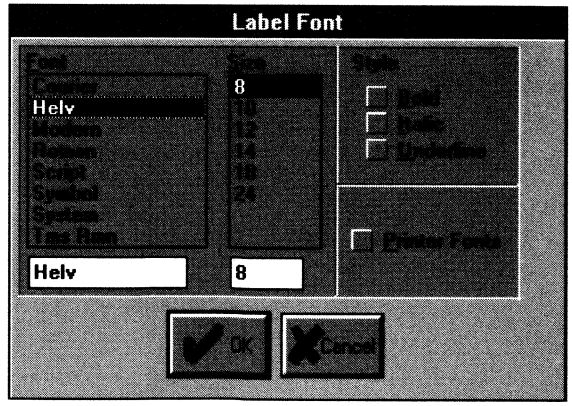
The initial default field label font is 8-point Helv. This font is also used to display value options in all selection method field types.

Now change the label font of the International and Computer, Ltd. text objects to 24-point Tms Rmn bold:

1. Position the pointer on the International text object and click.
2. Hold down *Ctrl* and click the Computer, Ltd. text object, then release *Ctrl*. Both text objects are now selected.

3. Click either of the selected objects with the right mouse button. The property inspector appears.
4. Choose Label Font. The Label Font dialog box appears as shown in Figure 5.3.

Figure 5.3
The Label Font dialog box



5. Set Font to Tms Rmn, Size to 24, Style to Bold, and then click OK.

Modifying borders

Borders improve the appearance of a form by creating a boundary between fields, adding color, and highlighting objects that you want to emphasize. When you create any object except a line or a button, a thin black border that surrounds the object is also created automatically.

You can modify or remove borders from all objects except tables and columns. A table border can't be removed, but you can modify the color and line width. Borders for columns can't be removed or modified in any way.

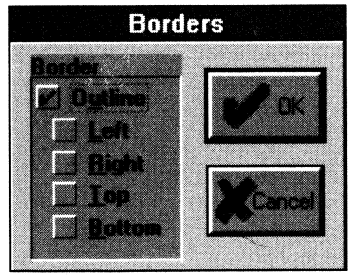
For all other objects with borders, you can control whether the border is a box or one or more lines, the line width, and the color of the border.

To remove the borders from the fields in the BILL TO and SHIP TO areas,

1. Select the fields from which you want to remove the border. Recall that you can select multiple fields by holding down either *Ctrl* or *Shift* as you click.

2. Click the right mouse button to inspect the properties for the selected objects.
3. Choose Borders. The Borders dialog box appears as shown in Figure 5.4. Outline is checked, by default.

Figure 5.4
The Borders dialog box



4. Click Outline to uncheck the box, then click OK. The borders no longer appear.

5. Now change the border for the text fields in the SHIP TO and BILL TO areas from Outline to Left border.

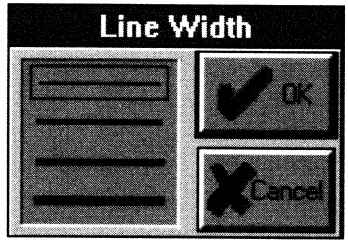
Selecting a line width

Line width is a property that applies to borders, rectangles, and lines you draw on the form.

Follow these steps to make the horizontal line near the logo thicker:

1. Click the line with the right mouse button to inspect its properties.
2. Choose Line Width. The Line Width dialog box appears as shown in Figure 5.5.
3. Select the second line from the top, then click OK. The line is now thicker.

Figure 5.5
The Line Width dialog box



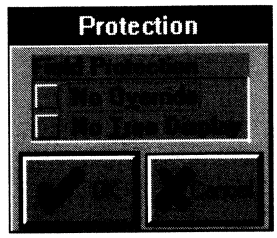
Protecting field values

You can restrict users from entering values or viewing value trees by assigning a protection property to a field. For example, in *Order*, users are unable to type values into either the BILL TO or SHIP TO area fields on the Order Entry form.

To protect these fields, do the following:

1. Select all the fields in the BILL TO and SHIP TO areas (except Bill Co.).
2. Inspect their properties by clicking the right mouse button.
3. Choose Protection. The Protection dialog box appears as shown in Figure 5.6.

Figure 5.6
The Protection dialog box



4. Check No Override, then click OK.

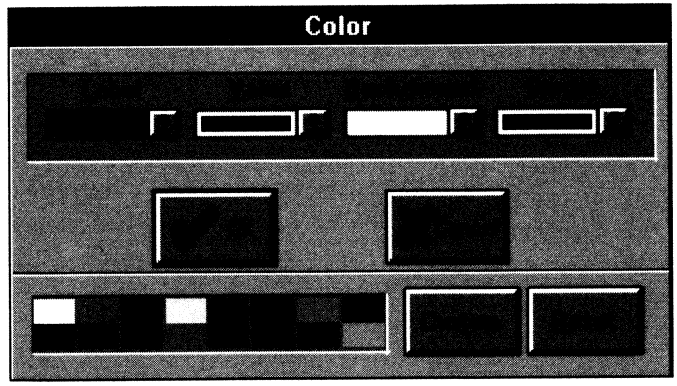
Applying color

You can color objects for style, to highlight certain information, and to show relationships. For example, in one application you might make the market research information fields yellow and the credit information fields green.

Follow these steps to make the Sales Order field pale yellow and its value font dark blue:

1. Select the Sales Order field with the right mouse button to inspect its properties.
2. Choose Color. The Color dialog box appears as shown in the following figure. Notice that there are four combo box color palettes: one each for the object's label, value text, background, and border color.

Figure 5.7
The Color dialog box



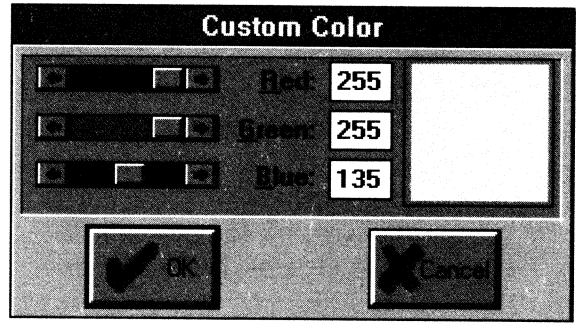
3. Click the Background combo box arrow, then click the dark yellow color.

4. Now click the Value combo box arrow, then click the dark blue color.
5. Click the dark yellow color in the palette at the bottom of the dialog box. This is the color you'll modify to create a pale yellow custom color.

6. **Tip:** When you select a color to modify, use a color you don't ordinarily use. After a color is modified, all other objects of that selected color are also changed. To return a custom color to its original settings, select the color and then click Reset.
7. Click Custom. The Custom Color dialog box appears.
8. In the Custom Color dialog box, slide the controls or type new values to change the proportion of Red, Green, and Blue in the selected color.

   To match the pale yellow color in the Sales Order field, set the sliders to Red 255, Green 255 and Blue 135.

Figure 5.8
The Custom Color dialog box



9. Click OK to close the Custom Color dialog box, then click OK to close the Color dialog box.

Define the color settings for the remaining fields on the forms as shown in Table 5.3.

Table 5.3
Color settings for the fields in Order

Form	Background	Value	Border	Label
Order Entry:				
Sales Order	Pale yellow	Dark blue	Dark blue	Black
Contact Name	Yellow	Dark blue	Dark blue	Black
Column labels	Light blue			
Customers:				
Rounded Rectangles	White		Gray	
Customer Type	Pale yellow		None	
Tax Exempt No.	Pale yellow		Black (Left,Top)	
Tax Exempt	Pale yellow		Black (Top)	
Filled Rectangle (under button)	Dark gray			

Tip: Note that background white is transparent, not an opaque color. To set a field to an opaque white background, you must first create it as a custom color.

You can apply color to different parts of a table object:

- To color the border, label, or table header of a table object, select the table header.
- To color the label or column header of a column, select the column header.
- To color the value or cell background for an entire column, select one cell in the column.

Note that you can't select multiple columns in a table; the color for each column must be assigned individually.

Adding help text

To help users complete a form, you can create online help for any field. The user can display the help text during form completion by selecting the field and pressing *F1*.

To add help information to the Customer Type field,

1. If the Customers form is not selected, choose Form | Select, then double-click Customers.
2. Position the pointer on the Customer Type field. Click the right mouse button to view the property inspector, then choose Help. The Help Text dialog box appears.

3. Type the following text:

```
Customer type is defined as follows:
Distributor - Signed distributor agreement on file
Dealer - Must provide valid tax ID
Educator - Recognized college or university
End User - Any individual or company not listed above
```

Tip: To begin typing on a new line, press *Ctrl+Enter*. If a line extends beyond the right border of the Help window, users will need to scroll the text to read it.

4. You can use standard editing commands to cut, copy, and paste text. Or, you can prepare your help text in a Windows word processor, such as Notepad, copy it to the Clipboard, and then paste it into the Help Text dialog box.
5. To attach the help text to the selected field, press *Enter* or click OK.

Adding scroll bars

To help users view all values in a table object, you can add scroll bars to any table. When users view tables that have more values than can display at one time, they can use the scroll bar to view the other values.

To add scroll bars to Table1;

1. If the Order Entry form is not selected, choose Form | Select, then double-click Order Entry.
2. Select the entire table object.

   Because Table1 doesn't have a table header, you can't click it to select the entire table object. Instead, click any cell (except the column headers) *twice*. This is different than double-clicking—let there be more time between clicks.

3. Click the right mouse button to inspect the properties of the table object, then choose ScrollBar.

   An object appears outside Table1 along the right margin. In the form edit mode, a table scroll bar isn't active and it doesn't display the way it does in form completion mode.

Defining property defaults

You can set your own defaults for object properties. After you set default properties for an object, every new object you create has the default properties. This can be handy if you create a lot of fields with the same properties.

To assign your own defaults to any object (except Button) on the Object bar, follow these steps:

1. Click a button, such as Field on the Object bar, with your right mouse button. The property inspector for that object appears.
2. Set properties such as field type, alignment, font, color, borders, line width, and protection. These properties become the new default properties for the next objects you create, and remain in effect until you change them.
3. To return to all initial default properties that shipped with ObjectVision, choose the Defaults command from the Objects menu, then choose Reset.

You can also choose Objects | Defaults to set the default properties for objects.

Looking at field properties

You have now assigned properties to all the objects on the Order Entry and Customers forms. Now close the Form Tool and experiment with the application:

- Type a valid date in the Date field. The date appears in the format you specified. In Chapter 6, you'll create a value tree that automatically calculates the current date.
- Type values in the various numeric fields, such as Price and Discount. The value appears in financial format, with two decimal places.
- Try typing numbers in the State field or letters in the Zip Code field. The application will not accept these values. You assigned these fields the Picture field type, which defines the types of values the field will accept.
- You must type values for the SHIP TO and BILL TO area fields individually on the Customers form. Because these values will

often be the same, the application should ask the user if the BILL TO values are the same as the SHIP TO values, and transfer them if the answer is yes. You will learn how to prompt the user for this additional information in Chapter 6.

- You must enter a number in the Sales Order field. For now, type 2. The application currently does not calculate values like Subtotal, Tax, and Total. You will learn how to do this in Chapter 6, "Creating value trees."
- Right now, nothing happens when you click the buttons. You will add the ability to respond to user actions in Chapter 7.
- You cannot add new values to the external data file or view old orders by entering the order number. You will learn how to do this in Chapter 8.
- Nothing happens when you click the scroll bar arrows next to Table1. The scroll bar is inactive until you are able to view orders in the external data file. You will add this property in Chapter 8.

Objects and their properties

Table 5.4 presents a summary of the properties you can add to each type of ObjectVision object. Value tree and event tree properties are not listed in this table, but are discussed in Chapters 6 and 7.

Table 5.4: Objects and their associated properties

	Field	Button	Table	Column	Cell	Text	Rectangles	Line	Graphic
Field type	X			X					
Alignment	X			X		X			
Value Font	X				X	X			
Label Font	X		X	X					
Color	X		X	X	X	X	X	X	X
Borders	X					X	X		X
Line width	X		X			X	X	X	X
Fill pattern							X		
Protection	X			X					
Name/Text	X	X	X	X		X			
Help	X	X		X					
ScrollBar			X						

Objects in multiple locations

Notice that some object properties can be different for objects that appear in multiple locations, and some must remain the same. For example, Bill Co. appears on both the Order Entry and the Customers form. It has the selection list field type on one form and the combo box field type on the other.

These properties can be different:

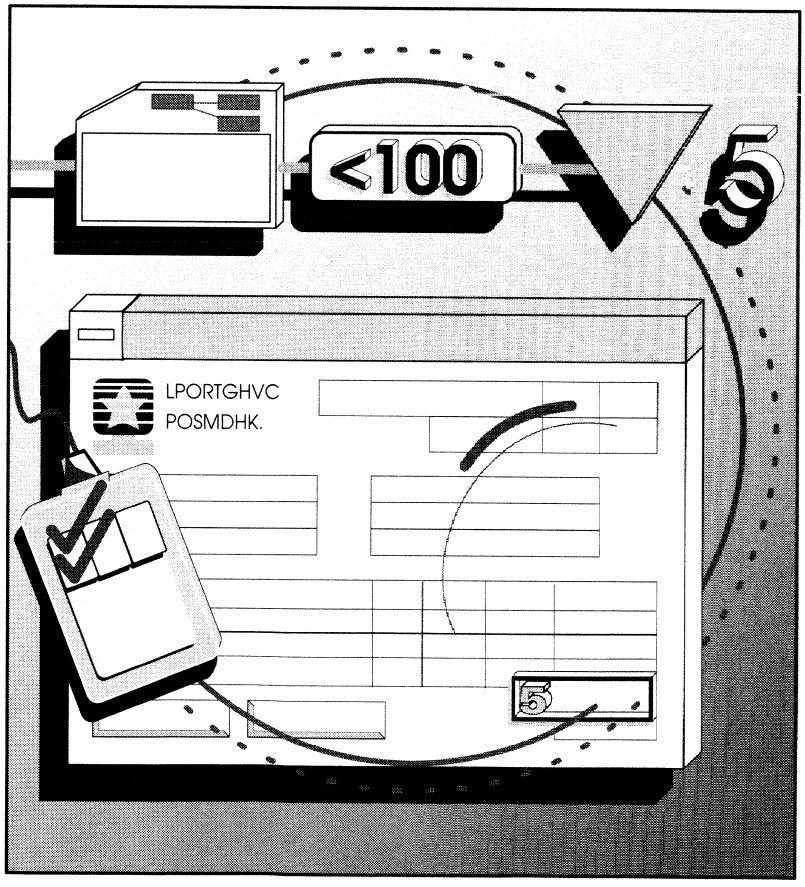
- Field type
- Alignment
- Font
- Color
- Borders
- Line width
- Fill pattern
- Protection

All other properties are the same across all instances of an object.

C H A P T E R

6

Creating value trees



Chapter goals:

- Recognizing when to use value trees
- Creating simple value trees
- Creating complex value trees

A value tree is a special type of field property. It defines the logic ObjectVision uses to calculate a field value. This chapter explains when to use value trees in your application and demonstrates how to create value trees for the *Order* application.

What you need to complete this chapter:
You can complete the exercises in this chapter at any time by starting with the file ORDER6.OVD. The explanations assume you are familiar with the objects introduced in Chapter 4, "Designing forms."

When to use value trees

Recall that fields can receive their values from three sources: user input, a calculation based on other values, or an external link. Value trees calculate values based on other values.

In the *Order* application, for example, a value tree assigned to the Total field calculates its value based on the values in the Subtotal, Shipping, and Tax fields.

In the *Order* application, the fields in Table 6.1 get their values from value trees.

Table 6.1
Fields with value trees in Order

Field	Based on
Date	@TODAY
SHIP TO area	Shipping Information Same as Billing Information?
Discount	Amount Qty. Price Customer Type
Amount	Qty. Price Discount
Subtotal	Amount
Tax	Tax Exempt Subtotal
Tax Exempt No.	Tax Exempt
Shipping	Ship State Subtotal Quantity
Total	Subtotal+Tax+Shipping

The values for the Item No., Part No., Description, and Price columns come from links to external database files. These are discussed in Chapter 8, "Creating links."

Creating simple value trees

Value trees are made of branches, conditions, and conclusions. You can think of a branch as a field, a condition as an "if" statement, and a conclusion as a "then" statement. For example, suppose the *Order* application determined the value for the Discount field based solely on the quantity shipped. The if-then statement would read:

> "If the quantity is under *10*, then don't give a discount. If the value is equal to or greater than *10*, then give a 20% discount."

The following table shows how this is calculated.

Table 6.2
Simple value tree for Discount

Branch	Condition	Conclusion
Quantity	<10	0%
	>=10	20%

If there were another condition (for example, >20), the value 21 would meet two conditions: >20 and >=10. ObjectVision value trees select a conclusion based on the *first* condition that is satisfied. Therefore, it is important that you arrange the conditions in the order that ObjectVision should evaluate them. In the preceding example, the condition >20 should be the first condition in the value tree.

Some value trees don't require the application to evaluate a field for its value because the same calculation is performed every time. For example, the Total field simply sums the values in the Subtotal, Tax, and Shipping fields:

Table 6.3
Table for the Total field

Branch	Condition	Conclusion
None	None	Subtotal + Tax + Shipping

Figures 6.1 and 6.2 show how this decision logic would appear as ObjectVision value trees: one is simple and one is more complex.

Figure 6.1
A simple value tree

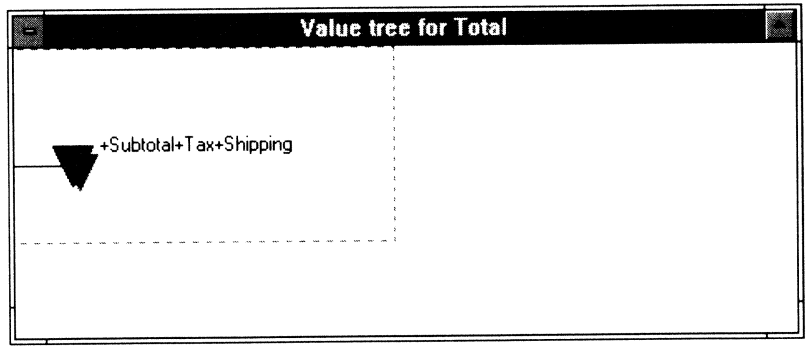
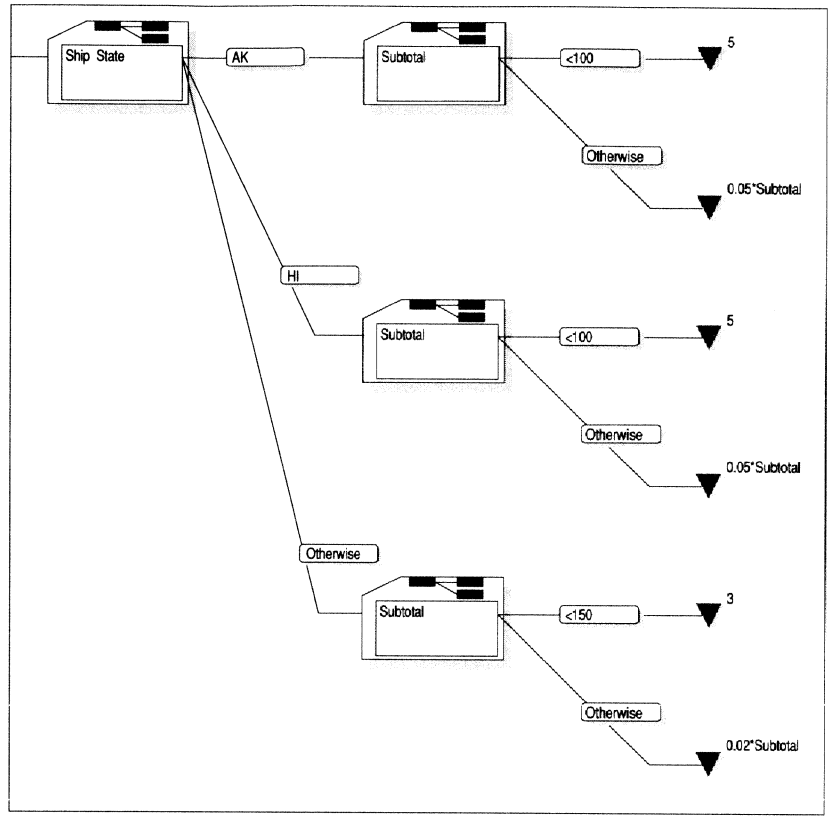


Figure 6.2
A complex value tree



Creating a value tree for Tax

A simple value tree for the Tax field includes the following if-then statements:

> "If the value for Tax Exempt is *Yes* (True), then return the value *0*. If the value for Tax Exempt is *No* (False), then calculate the value as follows: *0.07*Subtotal*." In this case, the tax rate is 7%.

The following table shows how this is calculated.

Table 6.4
Calculations for the Tax field

Branch	Condition	Conclusion
Tax Exempt	Yes (True)	0
	No (False)	.07 * Subtotal

The conclusion *.07*Subtotal* is an expression. Expressions can include combinations of constant values (such as .07), operators

(such as *, for multiplication), field names (such as Subtotal), and @functions, which are covered later in this chapter.

For a complete list of operators and @functions, see Chapter 6, "Writing Expressions," and Chapter 8, "@Function descriptions," respectively, in the *ObjectVision Reference Guide.*

Next, you'll add a value tree for the Tax field.

1. Choose Tools | Form to open the Form Tool if it is not already open.

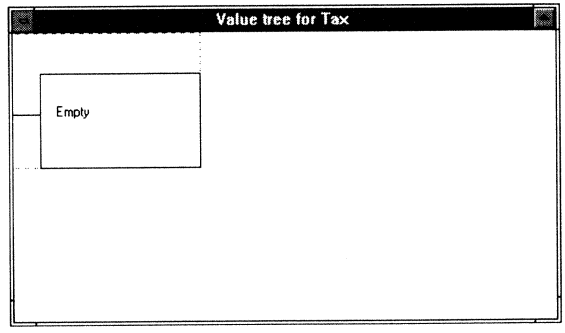
2. Click the Tax field with the right mouse button to inspect its properties.

Caution!

Be sure to click the Tax *field* and not the Tax *text object*. Because a value tree cannot be assigned to a text object, Value Tree does not appear in the text object property inspector.

3. Choose Value Tree. The Value tree for Tax dialog box appears as shown in Figure 6.3. The value tree is currently empty.

Figure 6.3
The value tree for the Tax field

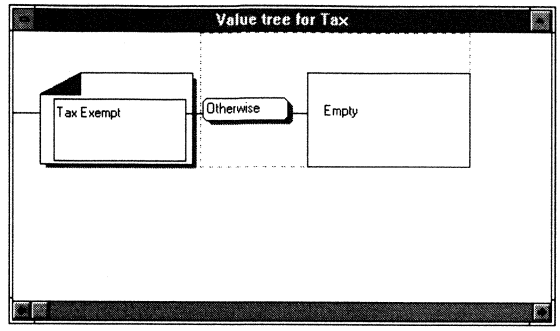


To create the value tree,

1. To add a branch, click Branch (shown at left) on the Object bar. A dialog box appears containing all Field Names in the application.
2. Scroll down the list and select Tax Exempt, then click OK.

The value tree now shows a branch, but no condition or conclusion (Figure 6.4). A dotted rectangle surrounds the conclusion node, showing that it is selected. The default condition Otherwise and default conclusion Empty exist simply as placeholders.

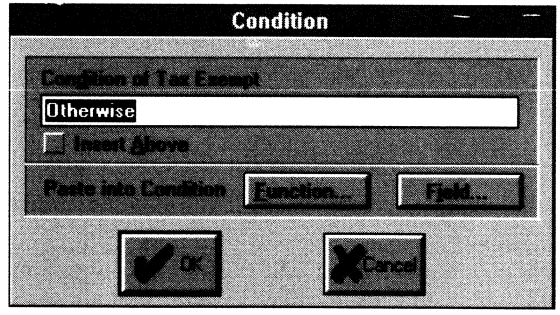
Figure 6.4
A new branch in the value tree for Tax



3. Now click Conclusion (shown at left) on the Object bar. Note that first the Condition dialog box appears so you can define the logic that will lead to this conclusion, then the Conclusion dialog box appears.

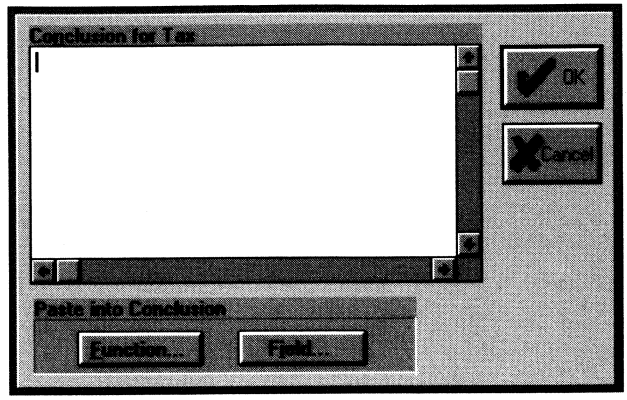
   In the Condition dialog box (Figure 6.5), enter the condition of the Tax Exempt value.

Figure 6.5
The Condition dialog box



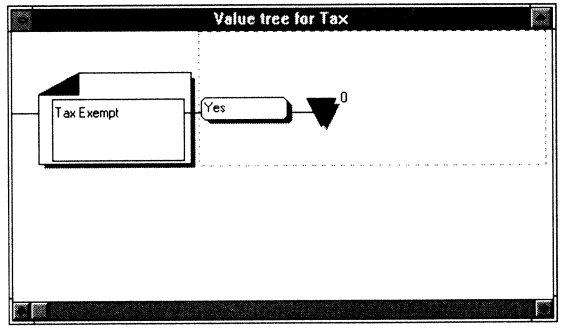
4. Type `Yes` or `True`, then click OK. The Conclusion for Tax dialog box appears as shown in Figure 6.6.

Figure 6.6
The Conclusion dialog box



5. Enter the conclusion `0` and then press *Enter*. The value tree now appears as shown in Figure 6.7.

Figure 6.7
The first branch of the Tax value tree



6. Next, add the second condition by clicking Conclusion on the Object bar. Type `No` or `False` in the Condition text box and click OK.

   When the Conclusion for Tax text box appears, type `.07*`. Instead of typing `Subtotal`, click the Paste Field button to view the names of all fields, then double-click Subtotal.

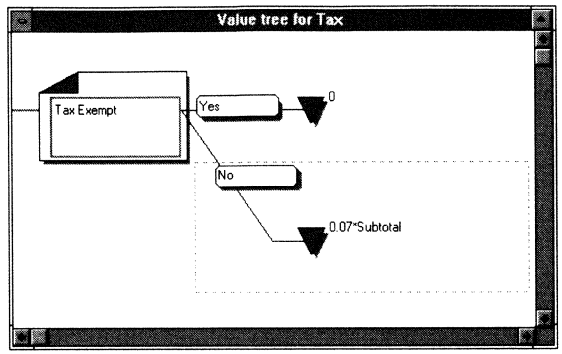
Tip: If you add @functions by clicking the Paste Function button instead of typing them, you avoid inadvertent typing mistakes, and you can see the arguments that a function requires by checking Paste Arguments.

Caution!

If you type the field name incorrectly, ObjectVision *automatically creates a new field* with that name. This new field doesn't appear on a form, but appears on a pop-up Scratchpad form if its value is required in a value tree.

7. Click OK. The value tree is now complete as shown in Figure 6.8.

Figure 6.8
The completed value tree



To enlarge the value tree display, click Expand (shown at left) on the Object bar.

To reduce the display, click Reduce (shown at left) on the Object bar.

When you create value trees with more branches, you can click Reduce several times to view more of the tree at once. You can also change the view by dragging the window's borders to enlarge it, or by scrolling.

To close the value tree, double-click the Control-menu box or click Close Tool (shown at left) on the Object bar.

If you click the right mouse button to inspect the properties for the Tax field, Value Tree is checked, indicating that a value tree exists for the field.

Testing a value tree

To test the logic in your value tree,

1. Close the Form Tool.
2. Select the Customers form.
3. Check Tax Exempt.
4. Choose Form | Select and double-click Order Entry to return to the Order Entry form. The value *0.00* displays in the Tax field because this customer is tax exempt.
5. Type `100` in the Subtotal field and press *Enter*.

6. Choose Form | Select and double-click Customers to return to the Customers form.
7. Uncheck Tax Exempt, then return to the Order Entry form. The value *7.00* now displays in the Tax field. The value tree logic is correct.

Now add the following value trees to the *Order* application:

- Amount
- Subtotal
- Total
- Tax Exempt No.

Figures 6.9 through 6.12 illustrate the completed value tree for each field.

Figure 6.9
Completed value tree for Order's Amount field

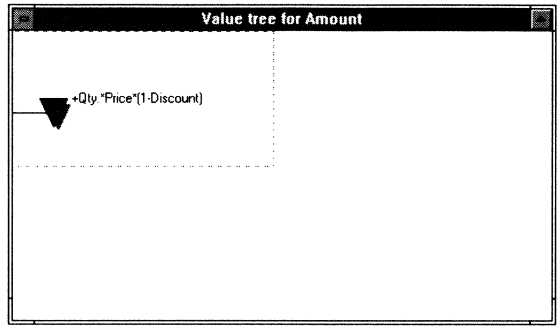


Figure 6.10
Completed value tree for Order's Subtotal field

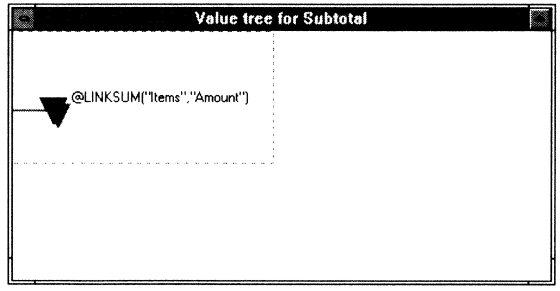


Figure 6.11
Completed value tree for Order's Total field

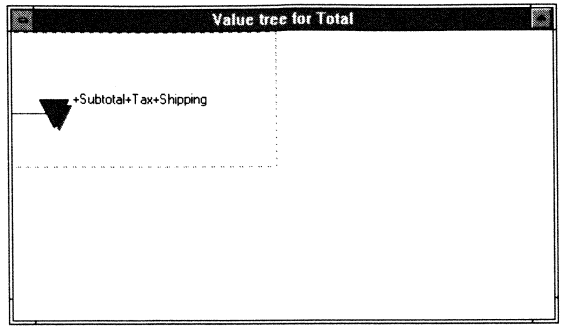
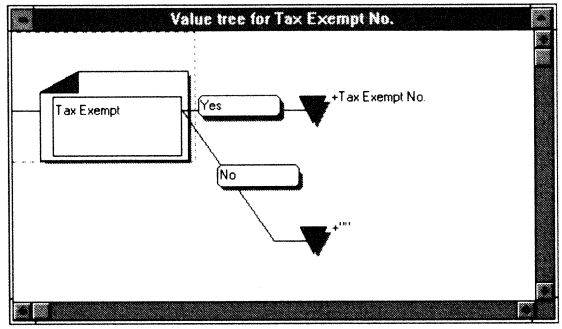


Figure 6.12
Completed value tree for Order's Tax Exempt No. field



The value trees for the Discount and Shipping fields are more complex, and are described later in this chapter.

Editing a value tree

To change a condition, first display the field's value tree in the Form Tool, then double-click the condition symbol to display the Condition dialog box. Type the new condition, then click OK.

To change a conclusion, first display the field's value tree in the Form Tool, then double-click the conclusion symbol (the inverted triangle). For example, to change the tax rate to 7.5%, you would double-click the conclusion symbol and edit the text in the Conclusion dialog box.

Tip: When you are editing a form in the Form Tool, you can double-click a field as a shortcut for displaying its value tree.

Printing a value tree

You can print a value tree whenever it appears onscreen. To print a tree,

1. Adjust the size of the tree that you want to print by clicking Reduce or Expand on the Object bar.
2. Choose File | Print Tree to print the value tree.
3. Or, choose File | Print All to print all value trees and all event trees in the application.

Creating complex value trees

The Tax value tree evaluated only one field to calculate a value. You can create more complex value trees that add powerful features to your ObjectVision applications:

- **Creating lists of value options.** When you designed the form, you specified the values for the Customer Type check boxes. In this section, you'll delete those values and let the value tree supply them automatically.
- **Evaluating two or more values to supply a value for another field.** The Discount value tree evaluates two fields: Customer Type and Quantity.
- **Including @functions in expressions.** Value functions instruct ObjectVision to calculate a value by performing mathematical, logical, financial, or string operations.
- **Writing complex logical conditions to evaluate two or more values before making a conclusion.** In some applications, this technique lets you create smaller, simpler value trees.

Creating lists of value options

The Customer Type field on the completed Customers form is defined as a Check Boxes field type with the values Distributor, Dealer, Educator, and End User. You can add a list of value options to a field in three ways:

- You type the list of value options in the Expected List dialog box when you assign one of the selection methods to the field.

This is the method you used in Chapter 4 for the Customer Type field.

- ObjectVision reads the value options from the field of a linked data file associated with the ObjectVision field. This method uses the default Automatic option and is mentioned in Chapter 8.
- ObjectVision, by default, automatically assembles the list from values you define in one or more value trees. This method uses the default Automatic option and is described in the next steps.

To create a list of values from a value tree,

1. First remove the list you typed:
   - Open the Form Tool if isn't already opened.
   - Open the Customers form.

   - Position the pointer on the Customer Type field and click the right mouse button to inspect its properties.
   - Choose Field Type. Check Boxes is already selected. Click OK to view the Expected List dialog box. You see the values that you typed earlier.
   - Check Automatic. The values you typed earlier disappear because ObjectVision now expects to use the values from the value tree, which you are about to create.
   - Click OK.
2. Now, follow the steps in the next section to create a value tree for the Discount field, which references the values for the Customer Type field. After you have completed that task, the check boxes will appear in the Customer Type field.

Evaluating two or more values

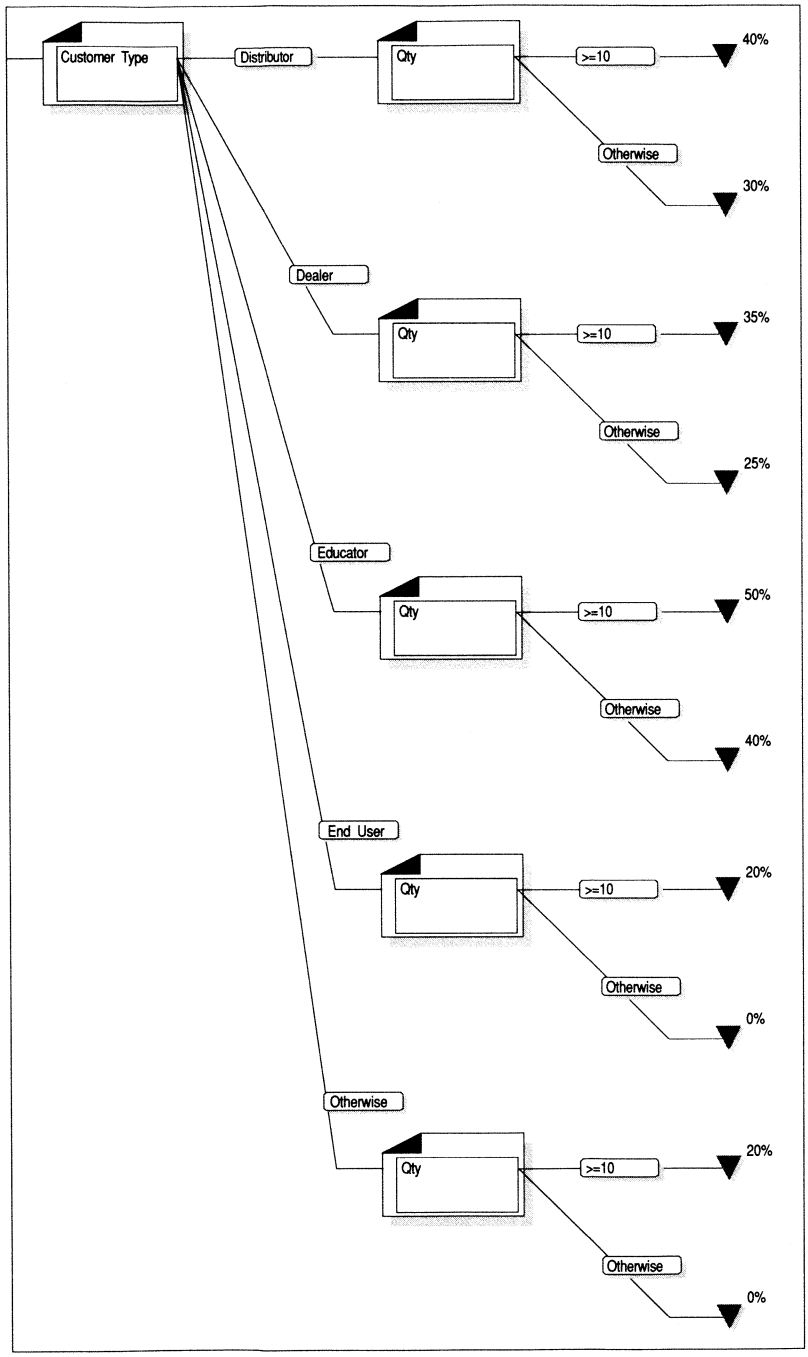
In the *Order* application, the Discount field gets its value from the values in the Customer Type and Quantity fields. Here are three of the if-then statements from the value tree:

- *If the Customer Type is distributor, then look at the Quantity.*
- *If the Quantity is equal to or greater than 10, then conclude 40%.*
- *If the Quantity is less than 10, then conclude 30%.*

Notice that the first statement tells the application to evaluate another field while the other statements return a conclusion.

Click Branch on the Object bar to add conditions in evaluating a field's value. Click Conclusion on the Object bar to add conditions that return a conclusion value. Figure 6.13 shows the completed Discount value tree.

Figure 6.13
The completed value tree for Order's Discount field



To create the Discount value tree,

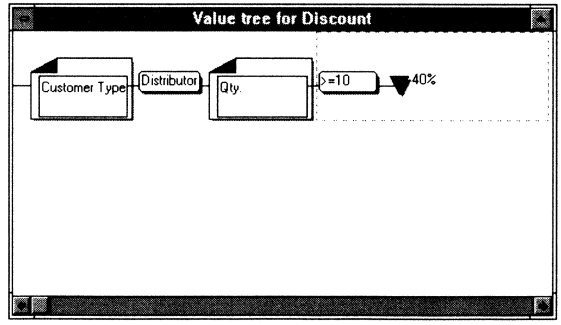
1. If the Form Tool is not already open, open it now by choosing Tools | Form.

2. Position the pointer on the column label, Discount, and click the right mouse button to inspect its properties.
3. Choose Value Tree.

4. **Create the first branch.** Click Branch (shown at left) on the Object bar, select Customer Type from the list of field names, then click OK.
5. **Create the next branch.** Click Branch again. The Condition of Customer Type dialog box appears. Type `Distributor` then press *Enter*. The Field Name dialog box appears. Select Qty. from the list, then click OK.

6. **Create the first conclusion.** Click Conclusion (shown at left) on the Object bar. The Condition of Qty. dialog box appears. Type `>=10` and then press *Enter*. The Conclusion for Discount dialog box appears. Type `40%` and then press *Enter*. The value tree now looks like Figure 6.14.

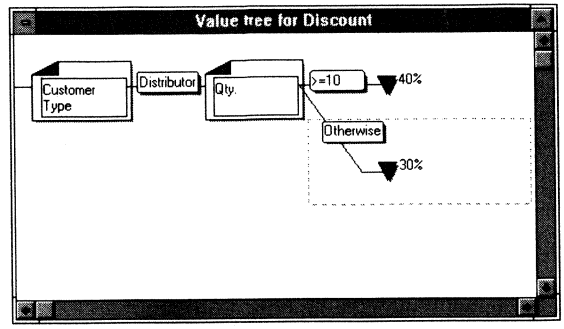
Figure 6.14
The beginning of the value tree for the Discount field



7. **Create the next conclusion.** Click Conclusion on the Object bar again. The Condition of Qty. dialog box appears again. Click OK to accept Otherwise, the default condition. The Conclusion for Discount dialog box appears again. Type `30%` and then press *Enter*.

   The value tree now looks like Figure 6.15.

Figure 6.15
The Distributor branch of the value tree for Discount



8. **Copy the existing branch.**

   You could repeat steps 4 through 7 to create the next branch, but it's easier to copy and paste. After the branch is pasted, it can be edited.

   Any branch can be copied by selecting its first node and then choosing Edit | Copy or pressing *Ctrl+Ins*. Choose Edit | Paste or press *Shift+Ins* to paste the branch at the location of the selected node.

   - Select the Distributor node, then press *Ctrl+Ins* to copy the entire branch.

9. **Paste a copy of the branch.**

   - Press *Shift+Ins* to paste the branch at the same level as the Distributor branch.

     The Condition of Customer Type dialog box appears. You enter the new condition for the branch you are about to paste.

   - Type `Dealer` and then press *Enter*. The Dealer branch appears below the Distributor branch. Note that the conclusion needs to be edited.

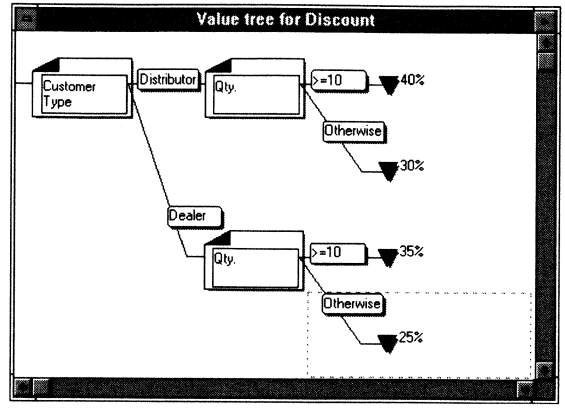
     If the entire tree doesn't display in the value tree window, you can resize it by dragging its bottom border or clicking the Maximize button.

10. **Edit the pasted branch.**

    - Double-click the first conclusion, 40%, for Dealer. The Conclusion for Discount dialog box appears again.
    - Replace 40% with 35%, then press *Enter*.
    - Double-click the second conclusion, 30% for Dealer. The Conclusion of Discount dialog box appears again.

- Replace 30% with 25%, then press *Enter*. The value tree now appears as shown Figure 6.16.

Figure 6.16
The Distributor and Dealer branches of the value tree for Discount



11. **Refer to Figure 6.13 on page 101 to complete the value tree.** Select the Dealer branch node, and paste another copy of the branch. Enter `Educator` for the condition. Paste two more branches (one for End User and one for Otherwise).
12. To finish, edit the conclusions of each branch.

As you specified values for the Customer Type field, ObjectVision added these values automatically to the list of value options for the field. If you close the value tree now by clicking Close Tool on the Object bar, you'll notice that the Customer Type field contains four check boxes: Distributor, Dealer, Educator, and End User. Because Otherwise is a special condition, it doesn't display.

Tip: Large value trees, such as the value tree for Customer Type, might not display in their entirety. To be sure that you are seeing the *entire expression* in a tree, click Expand on the Object bar to increase the size of the displayed view.

Now you'll add the value tree for Shipping, as shown in Figure 6.2 on page 91.

Adding @functions

To perform an operation on a field value, you can include @functions in value trees. @Functions return values based on calculations that can't be expressed by mathematical operators such as +, –, *, and /. For example,

- The @TODAY function inserts the current date in a field, using the format specified in the Field Type dialog box.
- @BLANK removes a value from a field.
- @AND(*LogicalList*) has one or more *arguments* enclosed in parentheses and separated by commas. Arguments are specific values required by an @function; most @functions require at least one argument. In this example, @AND returns the value Yes if all arguments are True; otherwise, it returns the value No.
- SUM(*List*) has one or more arguments separated by commas. It adds all the values in the list.
- @LINKSUM(*"LinkName"*, *"DataField"*). Returns the sum of all values in the link *"LinkName"* from the database field named *"DataField"*, even if the values are not all displayed on the form.

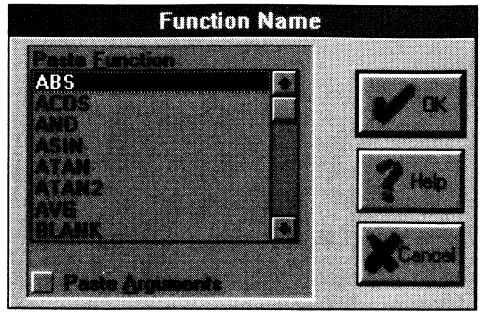
A complete list of @functions appears in Chapter 8, "@Function descriptions," in the *ObjectVision Reference Guide* and in the *ObjectVision Quick Reference Guide.*

Tip: Note that the value tree for Subtotal (Figure 6.10 on page 96) uses the @LINKSUM function to calculate the sum of *all* values in the Amount database field (even those which don't display in the table). This wouldn't be possible using the @COLUMNSUM function because it only sums the *visible values* in a column.

To add the @TODAY function to the Date field,

1. Position the pointer on the Date field and click the right mouse button to inspect its properties.
2. Choose Value Tree.
3. Click Conclusion on the Object bar. The Conclusion dialog box appears. You can either type @function names or select them from the Function Name list. Selecting the name from a list eliminates typing errors.
4. Click the Paste Function button to display the Function Name list as shown in Figure 6.17.

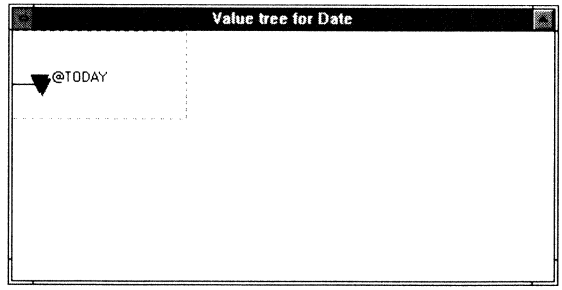
Figure 6.17
The Function Name dialog box



Tip: When Paste Arguments is checked, the argument list for a function is pasted into the dialog box in addition to the function name. This feature is helpful as you become familiar with ObjectVision @functions. It isn't necessary now because the @TODAY function has no arguments.

5. Select TODAY and click OK, then click OK to close the Conclusion for Date dialog box. The value tree for Date now appears as shown Figure 6.18.

Figure 6.18
The completed value tree for Order's Date field



6. Close the value tree by clicking Close Tool on the Object bar. Note that a value now displays in the Date field.

Tip: It isn't necessary to close the value tree window if you want to open another field's value tree. You can save time by choosing Tree | Select and double-clicking any field or column name in the list that appears. This shortcut works for both value trees and event trees.

Adding the remaining value trees

To complete the value trees for *Order*, the following fields and columns need value trees:

- Ship Co.
- Ship Dept.
- Ship Street
- Ship City
- Ship State
- Ship Zip
- Ship Zip Ext.
- Shipping

Open the completed *Order* application to refer to all of the remaining trees, print them for reference, or copy them.

Note that the value tree for the Ship Co. field evaluates a field that doesn't appear on any form—the Shipping Information Same as Billing Information? field. When the value for this field is required to complete the Ship Co. value tree, a Scratchpad form automatically appears so the user can enter a value.

The values in Description and Price aren't the result of value trees. The values in Description and Price are filled in by a link to a database. You'll learn how to link these fields to a database in Chapter 8, "Creating links.

Related topics

The following features are *not* demonstrated in *Order*.

Unrestricted branches

The unrestricted branch type is a special type of branch that can evaluate expressions as well as any number of different field's values. You can have an unrestricted branch in either value or event trees.

An unrestricted branch condition can be any valid ObjectVision expression that returns the value *Yes* or *No*. An unrestricted branch conclusion for a value tree can be any valid value tree conclusion. For example, an unrestricted branch condition can be any of the following:

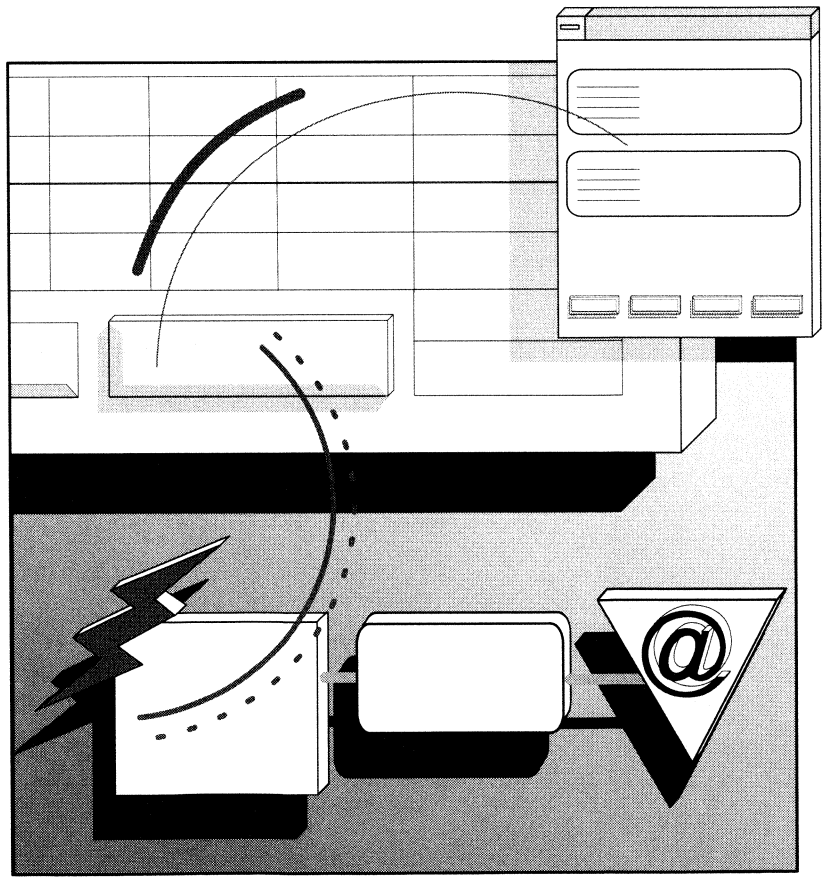
- @LENGTH(*FieldName*) > 10
- @ISBLANK(*FieldName*)
- @AND(*FieldName1*, *FieldName2*) = Yes
- (*FieldName1*) + (*FieldName2*) > 100
- @SELECTEDFORM = "*FormName*"

For more information about unrestricted branches, see Chapter 4, "Tree basics," in the *ObjectVision Reference Guide*.

C H A P T E R

7

Creating event trees



Chapter goals:

- Recognizing when to use event trees
- Assigning actions to objects

Like a value tree, an event tree is a special type of property. It defines the action ObjectVision performs in response to an event such as a click, a field or form selection, or a custom *Ctrl+* key combination.

This chapter explains when to use event trees in your application and demonstrates how to create event trees for the *Order* application.

What you need to complete this chapter:
You can complete the exercises in this chapter at any time by starting with the file ORDER7.OVD. The explanations assume you are familiar with the objects introduced in Chapter 4.

When *Order7* opens, this error message appears

```
Link is not yet created. Link name not found.
```

because the value tree for Subtotal uses the "items" link name as an argument. To continue, click OK. Creating the "items" link is discussed in Chapter 8.

When to use event trees

Event trees instruct the application to perform an action when an event occurs. Basically, event trees differ from value trees only in that they don't *usually* return values.

Event trees, however, can cause fields to get new values. For example, the @ASSIGN function can assign a value to any field. Link functions such as @NEXT deliver values to any connected fields.

Build event trees whenever you want the application to respond to one of the following events:

- **Clicks.** Buttons are usually assigned an event tree that makes them respond to a click.
- **Selecting a particular value from a list of options.** The event tree can display another form when a particular value is selected in a Selection List, Check Box, Radio Button, or ComboBox field.
- **Change.** An event tree can display a new form if a user enters a new value or overrides a default value.

- **Form events (such as Change).** An event tree can automatically close a form that has just been completely filled in, and make another form appear.
- **Stack events (such as Close, Open).** An event tree can display the message "Remember to sign out!" when the user closes an application.
- **Stack events (such as Custom shortcut keys).** An event tree can instruct the application to clear the active form when the user types a key combination, such as *Ctrl+C*.

Event trees issue instructions to the application using event functions. The event functions used in the *Order* application are

- @CLEAR(*LinkName*). Clears the linked ObjectVision fields associated with the link named *LinkName* in preparation for writing a new record.
- @CLEARALL. Clears all the values in a form except DDE link values and those values defined as constants in value trees, such as the Date field on the Order Entry form. It works identically to choosing Edit | Clear All.
- @DELETE(*LinkName*). Deletes the record at the current location in the external data file, using the *LinkName* link.
- @FORMSELECT(*FormName*). Selects the form named *FormName*. It works identically to choosing Form | Select and choosing an existing form name.
- @NEXT(*LinkName*). Displays the next record in the external data file based on the *LinkName*.
- @PREVIOUS(*LinkName*). Displays the previous record in the external data file based on the *LinkName*.
- @RESUME. Returns to guided completion.
- @STORE(*LinkName*). Updates or creates a record in the external data file with the current values.
- @TOP(*LinkName*). Displays the first record in the external data file based on the *LinkName*.

Many of these event functions affect links, which are described in Chapter 8. Don't be concerned with the details of links right now. Instead, focus on how event trees work—that is, how a user event can trigger a series of actions that affect the ObjectVision application or its values.

For a complete list of event functions, see Chapter 7, "@Function basics," in the *ObjectVision Reference Guide* and the *ObjectVision Quick Reference Guide.*

Assigning actions to objects

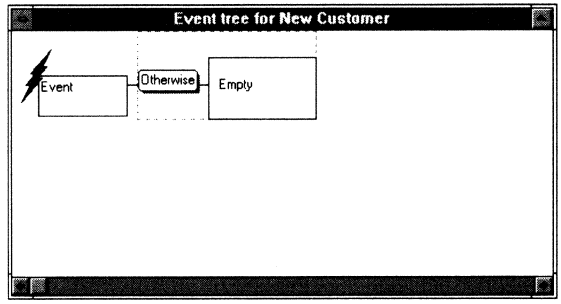
You can assign an event tree property to fields, columns, buttons, text objects, graphics, forms, and a stack. In the *Order* application, clicking the New Customer button instructs the application to display the Customers form.

To add an event tree to the New Customer button,

1. Open the Form Tool if it isn't open already.
2. Open the event tree for New Customer.

   You can open the event tree for button objects by double-clicking. (For objects other than buttons, text objects, and graphics, position the pointer on the field and click the right mouse button to inspect the properties, then choose Event Tree.) The Event tree for New Customer dialog box appears as shown in Figure 7.1.

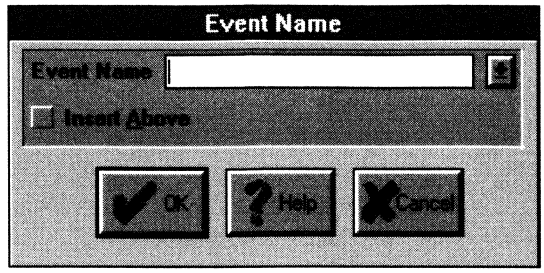
Figure 7.1
The empty event tree for Order's New Customer button



Event trees can be cut, copied, or pasted like value trees, except that the root *event node* isn't copied and must be specified before the tree can be pasted.

3. Click Conclusion (shown at left) on the Object bar. The Event Name dialog box appears as shown in Figure 7.2. The event determines which action will be initiated, just as in a value tree the condition determines which conclusion will be calculated.

Figure 7.2
The Condition dialog box for the New Customer event tree

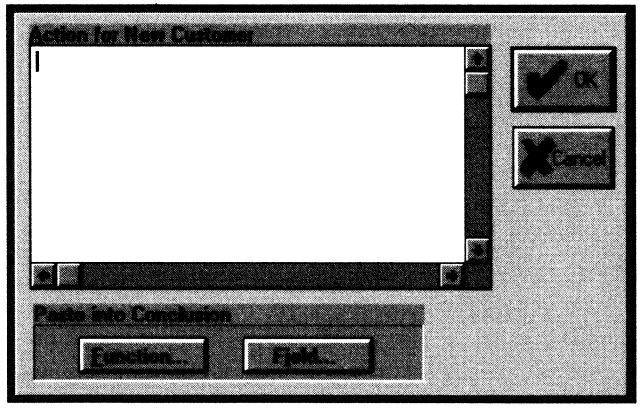


4. Type `click`. Or, click the Event Name combo box arrow and select Click.

   Click is a reserved keyword for an event in ObjectVision.

5. Click the OK button. The Action for New Customer dialog box appears as shown in Figure 7.3.

Figure 7.3
The Action for New Customer dialog box



6. Click Paste Function. The Function Name dialog box appears.
7. Scroll down the Paste Function list and select CLEAR.
8. Check Paste Arguments, then click OK.

Tip: If you add event functions by using the Paste Function button instead of typing them, you can see the arguments that a function requires by checking Paste Arguments.

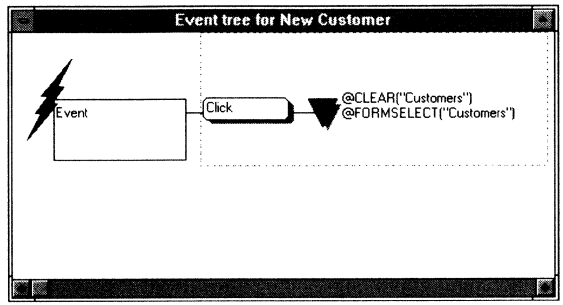
9. Replace the argument `LinkName` with `"Customers"`. Be sure to leave the double-quote marks around the argument.

   The "customers" link will be created in Chapter 8. Until then, you'll see an error message if you click this button in form completion mode. Also, clearing the current link value from the form will be ignored by ObjectVision until the link exists.

10. Press *End* to move the pointer to the right of the expression, then press *Ctrl+Enter* to add another line.
11. Click Paste Function again, scroll down the Paste Function list and select FORMSELECT.
12. Click OK.
13. When the Action for New Customer dialog box reappears, replace `FormName` with `"Customers"`. Again, be sure to include the double-quote marks.
14. Click OK.

The event tree for New Customer now looks like Figure 7.4:

Figure 7.4
The completed event tree for Order's New Customer button



Testing the event tree

You can test the event tree by following these steps:

1. Close the event tree and then the Form Tool.
2. Click the New Customer button. A message appears

   ```
   Link is not yet created. Link name not found.
   ```

   to let you know that the "customers" link hasn't been created yet. After you click OK, the Customers form appears.

   (In Chapter 8 you'll create the "customers" link and this error message won't appear.)

Now, return to the Form Tool and add the event trees for the following buttons:

- New Order (on the Order Entry form) as shown in Figure 7.5
- Enter Order (on the Order Entry form) as shown in Figure 7.6
- Return to Order Entry (on the Customers form) as shown in Figure 7.7

The other buttons on the Customers form—Enter, Next, Previous, Delete, Clear, Top, and Bottom will be created later, as described in Chapter 8.

Figure 7.5
The event tree for the New Order button

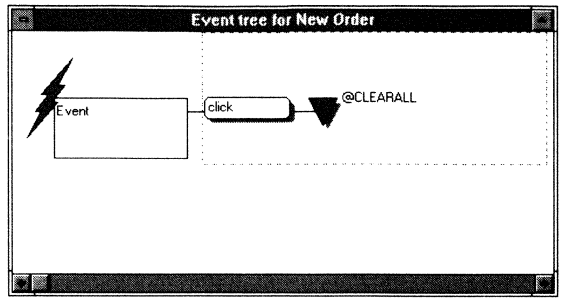


Figure 7.6
The event tree for the Enter Order button

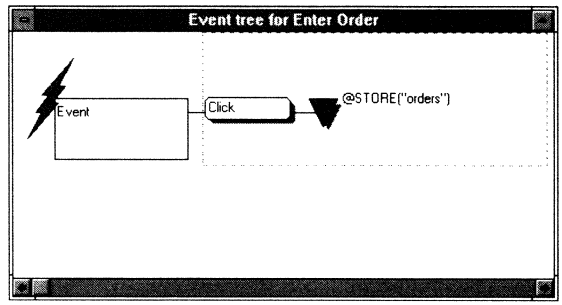
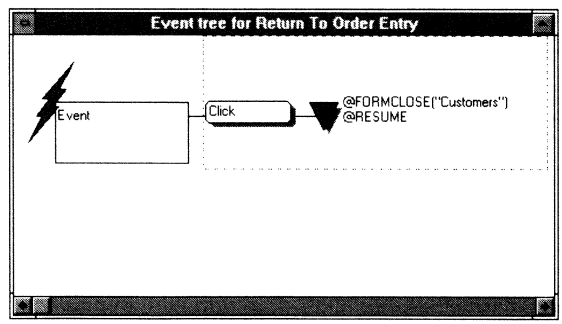


Figure 7.7
The event tree for the Return To Order Entry button



Testing your event trees

You have now assigned events to all the buttons except the link buttons (Enter, Next, Previous, Delete, Clear, Top, and Bottom) in the *Order* application. Link buttons are created automatically after

a link is successfully created. You'll learn how to create link buttons in Chapter 8.

To see how the application behaves, close the Form Tool and click some buttons.

Related topics

You can assign the event tree property to the following objects:

- Button
- Field
- Column
- Text
- Graphic
- Form
- Stack

Table 7.1 shows different objects and the different events that are appropriate for them.

Table 7.1: Objects and the events appropriate for them

Object	Click	Open	Close	Select	Unselect	Change	@EVENT
Field				X	X	X	X
Column				X	X	X	X
Button	X			X	X		X
Graphic	X						
Text	X						
Form		X	X	X	X	X	X
Stack*		X	X				X

* Recall that a stack is the entire set of forms in an application. An event tree for a stack is the only place where custom *Ctrl* key combinations and custom menu events can be defined.

Note that lines, filled rectangles, and rounded rectangles *cannot* have event trees.

Click events

A field or a column can't have a *click* event because a user might want to select a field or a column by clicking it.

Open and close events

When an application is opened in ObjectVision using File | Open, the stack receives an *open event*. Similarly, when another application is opened, an application that is already open receives the *close event* before it is replaced by the new application.

Select and unselect events

When a form is *selected*, its title bar is highlighted. A form is *unselected* when a user action changes the focus so the form title bar is unhighlighted. For example, a dialog box appears after choosing Form | New. The dialog box is the frontmost element on the screen and has the focus—the form is unselected.

Change events

Change recognizes when a value has been entered in a field or column, or when a form has been completed. After a user selects another field or column by pressing *Enter*, by pressing *Tab*, or by clicking, ObjectVision recognizes the new value in the previous field. Change also recognizes when a calculated or linked value appears in a field or column.

@EVENT function

You can use the @EVENT function in an event tree to send an event to a button or other object that has a name (lines and rectangles are unnamed objects). For example, @EVENT can send a click event to a button on a form.

For more information about @EVENT, see Chapter 8 in the *ObjectVision Reference Guide*.

Event trees for forms or stacks

The sample application, *Order*, doesn't have an example of a form or stack event. Form events let you assign actions that affect a form, such as clearing a form when it's closed. Stack events let you assign actions that affect the entire application, such as *Ctrl*+letter shortcut keys or custom menus.

Event trees for forms

An event tree for a form can initiate actions when a form is opened or closed. For example, a form event tree could have Close as a condition, and @PRINTFORM(*FormName*) as the conclusion. Every time a user closes that form, it would be printed. A form event tree could have Open as a condition, and @FIELDFIND(*FieldName*) as a conclusion. Every time a user opens that form, the specified field would be selected.

In the Form Tool, there are three ways to assign an event tree property to a form:

- Click the form title bar with the right mouse button, then choose Event Tree.

- Click the right mouse button on any part of the form that doesn't have an object on it, then choose Event Tree.
- Choose Properties | Form | Event Tree.

Build the form event tree in the same way as an event tree for an object.

For complete information about form events, see Chapter 4, "Tree basics," in the *ObjectVision Reference Guide*.

Event trees for stacks

Custom shortcut keys (*Ctrl+letter*) can only be used in a stack event tree. Defining a shortcut key in a stack event tree lets you create custom shortcuts for an application. Menus can be completely customized to limit commands the user can choose, or to add new menus and commands.

For example, a stack event tree could have *Ctrl+C* as a condition, and @FORMCLOSE(@SELECTEDFORM) as the conclusion. Every time a user presses *Ctrl+C* in that application, the selected form will be closed.

There are three ways to assign an event tree property to a stack. First, you have to open the Form Tool by choosing Tools | Form:

- In the Form Tool, choose Properties | Stack | Event Tree.

To use the other methods for assigning an event tree to a stack, click Stack Tool (shown at left) on the Object bar:

- In the Stack Tool, click the ObjectVision title bar with the right mouse button, and then choose Event Tree.

- In the Stack Tool, click the right mouse button on any part of the application window that doesn't have a form on it, then choose Event Tree.

Build the stack event tree in the same way as an event tree for an object.

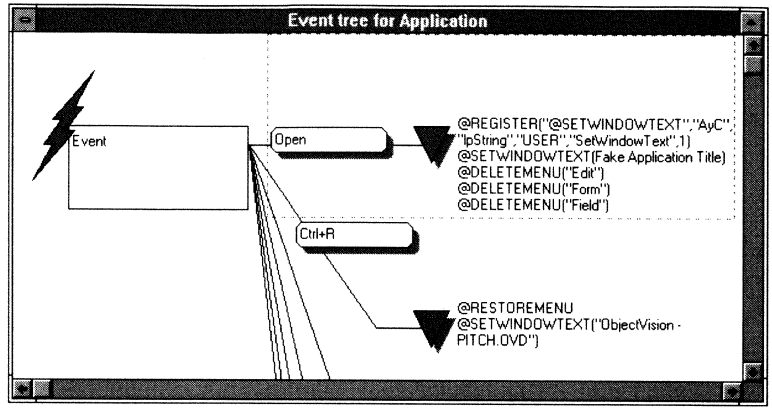
For complete information about stack events, see Chapter 4, "Tree basics," in the *ObjectVision Reference Guide*.

Registering DLL functions

You can create your own functions and register them within a ObjectVision application. Dynamic Link Libraries (DLL) functions can be new program code or code borrowed from any other existing Windows DLL files, such as USER.DLL.

A DLL is registered for use in an application by creating an event tree, which uses @REGISTER. For example, the stack event could be *Open*, and the action would use the @REGISTER function as shown in Figure 7.8.

Figure 7.8
When this application is opened, the DLL is registered



When @REGISTER successfully registers a DLL function, its name appears in the Paste Function dialog box and is available for any event tree in the application. Typically, @REGISTER is used in the stack event tree, but it can be placed in any event tree conclusion.

For a detailed explanation of DLLs and the @REGISTER function, see Chapter 20, "Using @REGISTER with DLLs," in the *ObjectVision Reference Guide*.

Unrestricted branches

Unrestricted branches for event trees are identical to unrestricted branches for value trees as described on page 107. Basically, unrestricted branches can evaluate expressions as well as any number of different field's values. An unrestricted branch condition can be any valid ObjectVision expressions that returns the value *Yes* or *No*. An unrestricted branch conclusion can be any event conclusion.

For more information about unrestricted branches, see Chapter 4, "Tree basics," in the *ObjectVision Reference Guide*.

C H A P T E R

8

Creating links

Chapter goals:

- Learning what links can do
- Linking fields
- Linking columns

Links are dynamic relationships between ObjectVision fields and external data files or other Windows applications. This chapter explains how to create links to external data files.

Links can vary from very simple links to very complex links. In this chapter, you're instructed to create one of the more complex multiple links to a table object. If you only need to use very simple links, you might want to read the linking tutorial in Appendix A, "The short course" instead of this chapter.

There are many link features and options that are not covered in this tutorial. To take full advantage of the power of ObjectVision links, see Chapters 10, "Linking basics," and Chapter 11, "Linking options," in the *ObjectVision Reference Guide*.

What you need to complete this chapter:
You can complete the exercises in this chapter at any time by starting with the file ORDER8.OVD. The explanations assume you are familiar with the basic operation of the *Order* application, described in Chapter 3, "Using an ObjectVision application."

When *Order8* opens, this error message appears

```
Link is not yet created. Link name not found.
```

because the value tree for Subtotal uses the "items" link name as an argument. To continue, click OK. Creating the "items" link is discussed in the "Linking columns" section of this chapter.

Though you don't need database experience to complete this chapter, you should be familiar with the concepts of *database tables*, *database indexes*, and *records*.

A database table is a file containing one or more sets of related values. Each set of values is called a record. For example, the *Order* application uses a database table called PARTS. Each record in the PARTS file contains two values: a part number and description.

An index on a table is like an index to a book—it lets a database program or ObjectVision locate records quickly. For example, when the user types a value into the Sales Order field in the completed *Order* application, ObjectVision searches the indexed field in the ORDERS database table to display all information about that order number.

Do not confuse a database table with an ObjectVision table object. Database tables are external data files, while ObjectVision table objects are used to enter and display values.

What links can do

Links to external data files

Links from ObjectVision applications to external data files can read values from or write values to that external file. This means that users can view data and add new data from within ObjectVision.

Links to Windows applications

When you create a DDE link between an ObjectVision application and a Windows application, you can read values from and write values to that application. You can also send commands to that application. Linking to Windows applications is *not* covered in this tutorial.

For information on this technique, see the *ObjectVision Reference Guide*, Chapter 16, "Linking through DDE."

You can create single or multiple links between an ObjectVision application and a data source. You can also create links to multiple data sources.

Once a link to a database table is created, you can connect any ObjectVision field or column to any database field within a link. The technique for connecting fields and columns varies only slightly.

Links perform these functions:

- Let your application get information ("read") from external data files. For example, in the *Order* application, the user can view the names and addresses of all companies that have placed orders.
- Let you transfer the values ("write") from your application to external data files. For example, in the *Order* application, new customer information entered on the Customers form is written to CUSTOMRS.DB, a Paradox table. Also, new order information entered on the Order Entry form is written to ORDERS.DB, another Paradox table.
- Simplify the process of viewing and updating records in external data files by letting users click buttons to perform these functions. For example, in the *Order* application, users can browse through existing orders by clicking the Next and Previous buttons. They add new order information to ORDERS.DB by clicking the Enter Order button.
- Simplify the process of adding and updating records in external data files by automatically writing changes to the file. For

example, in the *Order* application, users can enter new item information or edit existing records. Each time they move to a new row in the table object, the entered values are automatically written to the ITEMS.DB table.

When a value is entered in a field linked to a database index field, ObjectVision locates the record containing that value and delivers the other values in the record to the corresponding ObjectVision fields.

For example, as you saw in Chapter 3, if you select the value *Borland* from the BILL TO area Company field, ObjectVision delivers the associated values from the CUSTOMRS.DB table. Similarly, if you type any number from 1 to 6 in the Sales Order field, ObjectVision delivers the associated values from the Paradox ORDERS.DB table.

The value that triggers the link can be supplied in three ways: it can be entered by the user, calculated, or delivered by another link.

Types of links

ObjectVision supports links to five types of external data sources:

- Paradox
- dBASE-compatible
- Btrieve
- ASCII
- DDE

For more information on the types of links to external files, see the *ObjectVision Reference Guide*, Chapters 12 through 16.

Link fields

In this section, you'll create four links between the *Order* application and four Paradox tables.

First, you'll read how the links in the *Order* application were chosen. If you are creating your own application, this section should help you determine which links your application needs. Next, you'll create a link between your application and a Paradox database table.

Determining essential links

If you plan to link your application to existing database tables, you'll need to create links between the tables and your application.

If you plan to create new tables, however, it is important that you follow the principles of good database design. These are beyond the scope of this tutorial, but the basic principles can be expressed simply:

- **Avoid duplicate data in tables.** Each field should appear in one table only. The exception is an index field, described later.
- **Tables should contain related data.** For example, a table containing customer contact information should not also contain parts information.
- **Select an index that defines the relationships.** An index on a table is like an index to a book; it helps you locate information quickly. Generally, only unique values—such as Social Security numbers or telephone numbers—are used for indexes.

For more information on database design, see the *ObjectVision Reference Guide*, Chapter 17, "Tips for designing databases."

Following are the descriptions of the links in the *Order* application. If you are creating your own application, scan this list to get ideas for your links.

The "customers" and "orders" links are connected to ObjectVision fields.

- When a Company name is selected on the Order Entry form, the application should deliver the associated values in the BILL TO and SHIP TO areas. You'll create the "customers" link for this purpose.
- When the Sales Order number for an existing order is entered, the application should deliver the Date, Contact Name, Phone, Extension, and Bill Co. values. You'll create the "orders" link for this purpose. (Once the Bill Co. value is delivered, the "customers" link fills in the associated values from the CUSTOMRS.DB table.)

The "items" and "parts" links are connected to a table object.

- When the Sales Order number is supplied, the application should also deliver the Part No., Qty, and Extended Price values. You'll create the "items" link for this purpose. (Once the Part No. value is delivered, the "parts" link fills in the associated values from the PARTS.DB table).

- When the Part No. is supplied, the application should deliver the Description and Price values. You'll create the "parts" link for this purpose.
- When the user clicks the New Order button, the application should remove all values from both forms. This prepares the forms for receiving information about a new order.

Table 8.1 lists the links in the *Order* application, and the ObjectVision fields that hold the link values.

Table 8.1: The four links in the Order application and the database fields they connect to

Customers	Orders	Items	Parts
* Bill Co	* Order Number	* Order Number	* Part No
Bill Dept	Date	* No	Description
Bill Street	Bill Co	Part No	Price
Bill City	Contact Name	Qty	
Bill State	Phone	Amount	
Bill Zip	Ext		
Bill Zip Ext			
Ship Co			
Ship Dept			
Ship Street			
Ship City			
Ship State			
Ship Zip			
Ship Zip Ext			
Customer Type			
Tax Exempt			

* Represents an index field—that is, the field that the link uses to locate a record.

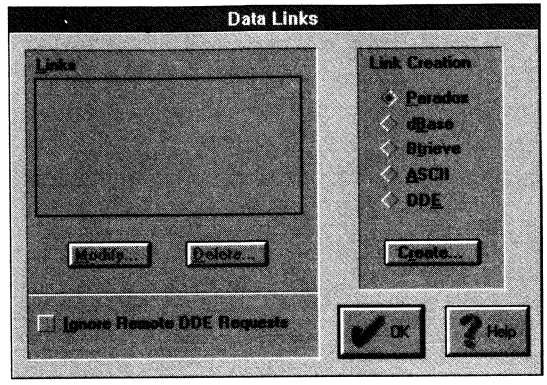
Creating the "customers" link

The "customers" link is a simple link connected to fields. To connect your application fields to CUSTOMRS.DB (a Paradox table),

1. Be sure the Customers form is the active form. The link buttons (Enter, Next, Previous, Delete, Clear, Top, and Bottom) will be automatically placed *on the active form* after the link is created.

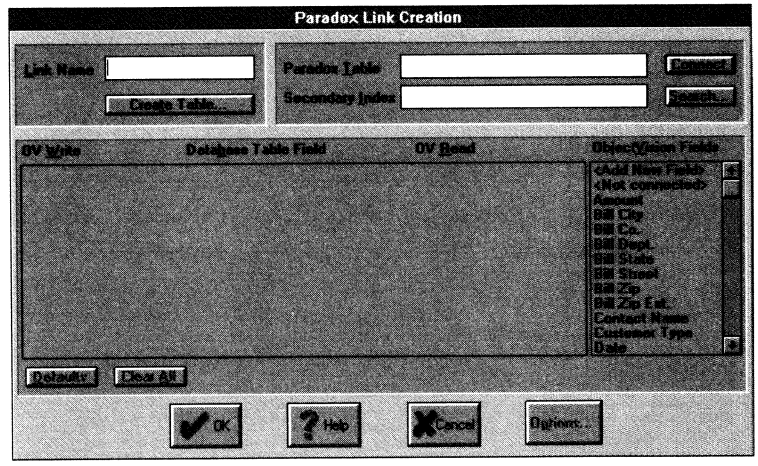
2. Choose Links (shown at left) from the Object bar. The Data Links dialog box appears as shown in Figure 8.1.

Figure 8.1
The Data Links dialog box



3. Be sure that Paradox is checked in the Link Creation list, then click Create. The Paradox Link Creation dialog box appears as shown in the following figure.

Figure 8.2
The Paradox Link Creation dialog box



4. Type `customers` in the Link Name text box, then press *Tab*.

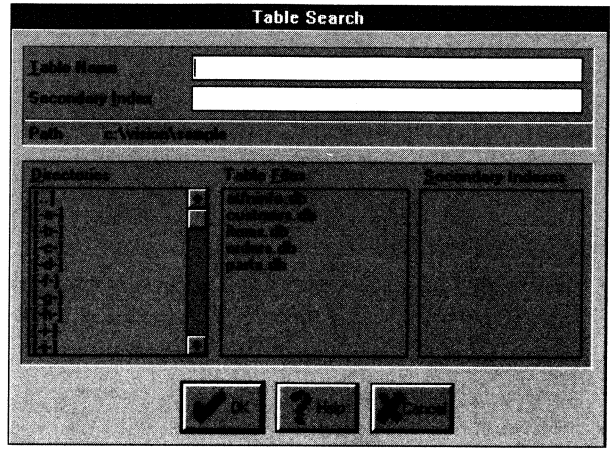
   You can type any link name that has not already been used. It's helpful to choose a name that helps you remember the purpose of the link.

5. Type `customrs` (note spelling due to the DOS 8-character limit) in the Paradox Table text box, then press *Tab*. The letters you type here are automatically converted to uppercase.

   It is *not* necessary to include the .DB extension. However, if the data file isn't in the same directory as your ObjectVision application, you *must* specify the path name.

Tip: This table has already been created. If you want to view a list of existing database tables, click Search. The Table Search dialog box appears, displaying the names of the Paradox tables in the current directory, as shown in Figure 8.3.

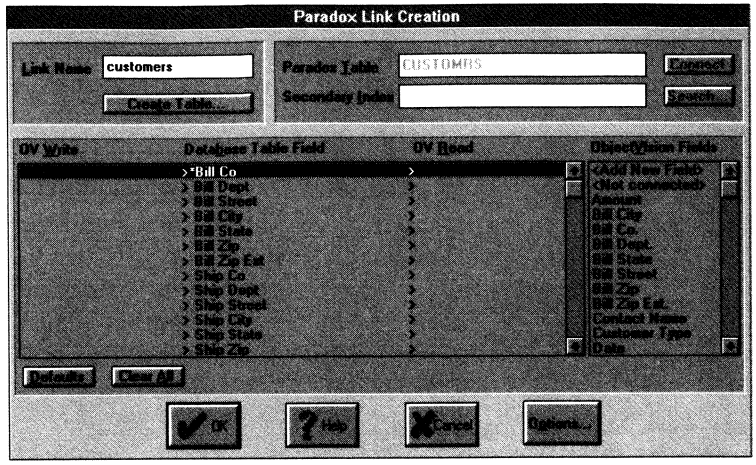
Figure 8.3
The Table Search dialog box



6. Leave the Secondary Index text box empty. This box is used to enter the name of the field which is a secondary index for a Paradox table. Similar text boxes are used for an index file for dBASE or a supplemental index number for a Btrieve table.
7. Click Connect or press *Enter*. If you typed a name other than `customrs` for the table name (note spelling), ObjectVision displays the message `Table was not found`.

   Your application links to the Paradox table name you entered, listing its fields in the Database Table Field column as shown in Figure 8.4.

Figure 8.4
The CUSTOMRS.DB fields

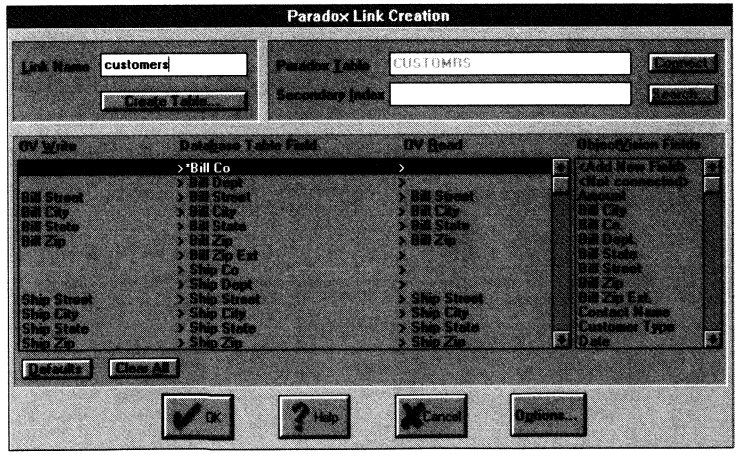


Notice that the Bill Co field is marked with an asterisk (*). The asterisk denotes it as an index field. The index field contains the unique value with which the other values in the record are associated.

8. Create connections between the listed Database Table Fields and the ObjectVision fields in your application:

   - To quickly create connections to the ObjectVision fields with the same names as the Paradox table fields, click Defaults. The OV Read and OV Write columns now contain those fields from the ObjectVision application that are spelled identically (including capitalization and special characters such as periods) as shown in Figure 8.5.

Figure 8.5
The default connections for "customers"

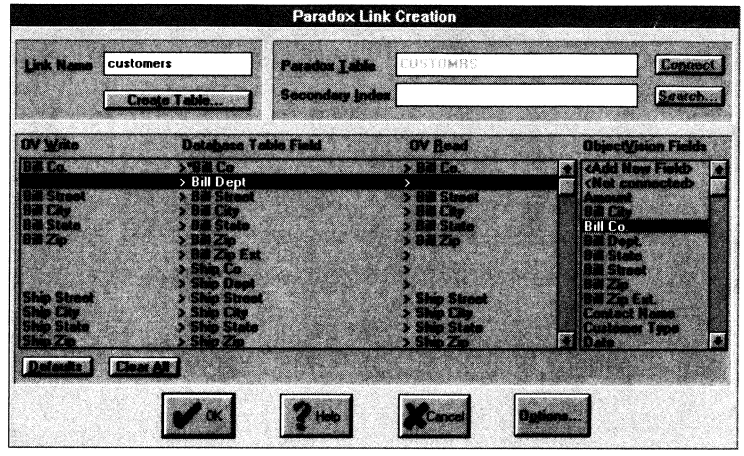


- To create connections for those database fields that are *not* spelled the same as their corresponding ObjectVision fields, select a database table field, select an ObjectVision field from the list at the right, scrolling if necessary, then double-click.

  For example, to connect the Paradox Table field Bill Co with the ObjectVision field Bill Co., select the Database Table Field Bill Co, then double-click Bill Co. in the ObjectVision Fields list. *Or*, select Bill Co. in the ObjectVision Fields list, and then double-click the Database Table Field column.

  Bill Co. now appears in the OV Read and OV Write columns, and the next row is selected, as shown in Figure 8.6.

Figure 8.6
The Bill Co. field is now connected to the database



- Connect the other Database Table Fields that are not yet connected to OV Read and OV Write fields.
- If you are creating your own application and want to add a read-only connection, select the field name in the ObjectVision fields list, then double-click the OV Read column to the right of the Database Table Fields column. The ObjectVision field name will appear in the OV Read column only.
- If you are creating your own application and want to add a write-only connection, select the ObjectVision field, then double-click the OV Write column to the left of the Database Table Fields column. The ObjectVision field will appear in the OV Write column only.

- To remove a connection, select <Not connected> from the ObjectVision Field list, then double-click the OV Read or OV Write of the Database Table Fields you want to erase.
- To remove a connection from *both* the OV Read and OV Write columns, first select the row, then double-click <Not connected>. Or, select <Not connected> first, then double-click the field name in the Database Table Field column.

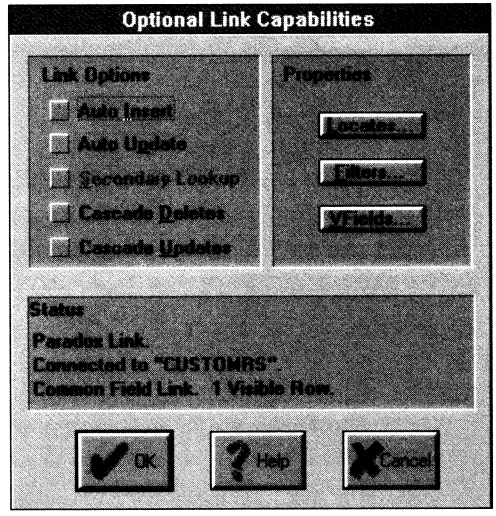
Setting link options

To specify the circumstances when the link will update the data file,

1. Click the Options button near the bottom of the Paradox Link Creation dialog box.

   The Optional Link Capabilities dialog box appears as shown in Figure 8.7.

Figure 8.7
The Optional Link Capabilities dialog box



Auto Update and Auto Insert define when changes are written to the database.

- To update the data file whenever a user modifies an existing value in a linked field and moves to another record, check Auto Update. This feature eliminates the need for special buttons that update records in the database—modifications are made automatically.

  Do *not* check Auto Update for the "customers" link.

- To insert a new record in the linked data file whenever the user creates a new record by entering new values in the linked table object, check Auto Insert. Auto Insert eliminates the need for special buttons that insert records in the database—insertions are made automatically.

  For example, the *Order* application has a button, Enter, on the Customers form that needs to be clicked to save the information to the CUSTOMRS table.

Do *not* check Auto Insert for the "customers" link.

Do *not* check Cascade Deletes or Cascade Updates.

Understanding link status

The Status box shows you information about the current link. It shows you what type of table is linked, the table name, and what type of ObjectVision object is connected.

A link can be connected to fields or columns. This link is connected to fields, as indicated by the `Common Field Link` text. Fields can only display one value at a time, so the Status box message `1 Visible Row` indicates this.

Defining Locate fields

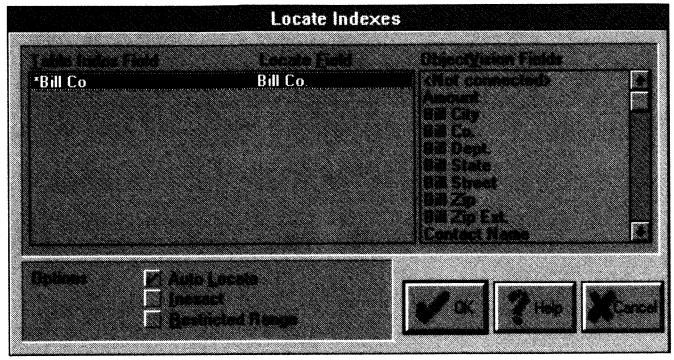
Locate fields are *the most important* ObjectVision link property. The Locate field is the ObjectVision field that triggers a link. When a new value is entered in a Locate field, the other values in the link are delivered from the database table to their respective fields.

For example, as you saw in Chapter 3, the Sales Order field triggers the link to deliver the order information. Similarly, the Bill Co. field triggers a link to deliver the customer information, and the Part No. field triggers a link to deliver the parts information.

The Locate field for the "customers" link is Bill Co. To specify the Locate field,

1. Click the Locates button in the Optional Link Capabilities dialog box. The Locate Indexes dialog box appears as shown in Figure 8.8.

Figure 8.8
The Locate Indexes dialog box



The Paradox table index field, Bill Co, appears in the left column, beneath the heading Table Index Field. The ObjectVision field, Bill Co., appears in the right column beneath the heading Locate Field.

2. Now specify Locate field options. Locate options determine two major options for delivering values:

   - Whether values are located automatically when a new value is entered in the Locate field.

     Auto Locate is checked by default. Thus, whenever a new value is entered in the Locate field, the link automatically fills in the associated values, if they exist.

   - How precisely the delivered records must match a value.

     If Inexact is checked, the application locates records based on an *incomplete* value. For example, if the user types `B`, the link returns the values for Borland.

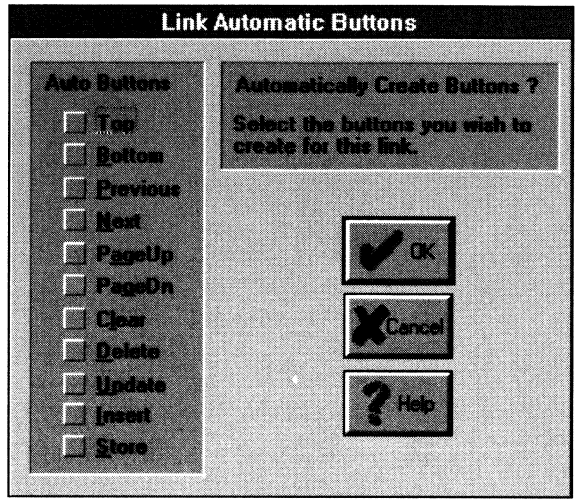
     If Restricted Range is checked, the application delivers only those records that *exactly* match the value entered in the Locate field. For example, if the user types `Borland`, the application delivers all associated values for this customer.

     Additionally, you can't move to any other record in the linked database that doesn't exactly match the locate value, because checking Restricted Range limits the delivered values to those that are exact matches.

3. For this link, check Restricted Range, and leave Auto Locate checked.
4. Click OK to close the Locate Indexes dialog box, the Optional Link Capabilities dialog box, and the Paradox Link Creation

dialog box. The Link Automatic Buttons dialog box appears, as shown in Figure 8.9.

Figure 8.9
The Link Automatic Buttons dialog box.



By checking the buttons you want and then clicking OK, ObjectVision creates link navigation buttons complete with their event trees. Each choice in the list has a corresponding @function.

5. The *Order* application should let the user view different records by clicking buttons, so check Top, Bottom, Previous, Next, Clear, Delete, and Store.
6. Click OK to create the buttons and close the Link Automatic Buttons dialog box, then click OK to close the Data Links dialog box. Automatically created buttons are placed at the bottom of the current form.

If the buttons don't appear, scroll the form to view the buttons. You can reposition the buttons by dragging them to a new location.

Rename the Store button by choosing Properties | Object | Name/Text, typing `Enter`, then clicking OK.

Creating the "orders" link

To link your application fields to the existing Paradox table, ORDERS.DB,

1. Select the Order Entry form by choosing Form | Select, then double-clicking Order Entry.
2. Choose Tools | Links to open the Links Tool.
3. Be sure that Paradox is checked as the Link Creation type, then click Create.
4. Type `orders` in the Link Name text box, then press *Tab*.
5. Type `orders` in the Paradox Table text box, then press *Enter* or click Connect.
6. Click Defaults.
7. Connect *Sales Order* as the OV Read and OV Write field to Order Number, the Database Table Field by scrolling to Sales Order in the ObjectVision Fields list and double-clicking it.

   When you finish connecting all the fields, the Paradox Link Creation dialog box should look like Figure 8.10.
8. Click OK.
9. When you see the Link Automatic Button dialog box, click Cancel. (You already created the Enter Order button in Chapter 7, "Creating event trees.") Click OK to close the Data Links dialog box.

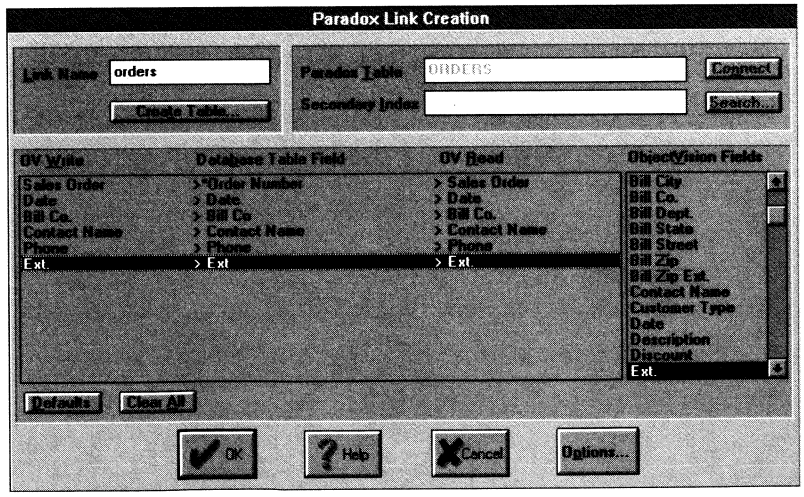


Figure 8.10
The Paradox Link Creation dialog box for "orders"

Linking columns

A column is similar to a field, but it can display values from several records instead of just one. However, only one record at a time is the current record, which can be modified. The *row pointer*, a right-pointing triangle next to the left margin of the table object, indicates the current or active row.

The order of link creation is significant for table objects.

The technique for connecting table object columns to external data files is the same as that for linking fields, with one exception: The order that you create the links to the table object *is* important.

If there are multiple links to a single table object, the *first* link created (the Primary link) *must* be the one that delivers the multiple records to the table's rows. All other links (Secondary Lookup links) to the table object can be created in any order—they simply fill in information about the values delivered by the Primary link.

In the *Order* application, for example, "items" is the Primary link because the ITEMS.DB table delivers the ordered items, which are the most important information about each order. The Secondary Lookup links simply fill in every item's description and price.

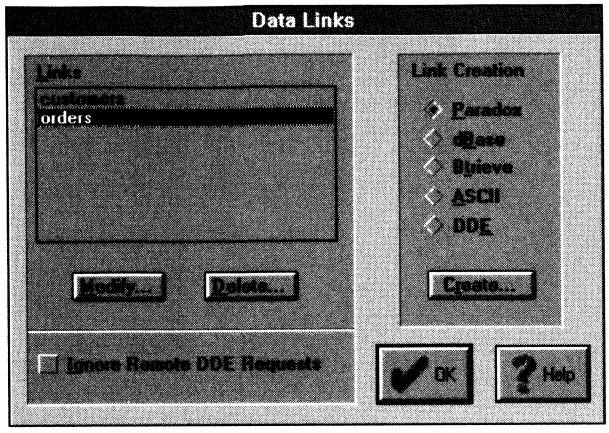
You'll specify the Primary link by creating it first.

Creating the "items" link

The "items" link is the Primary link for the table object, so you must create it first:

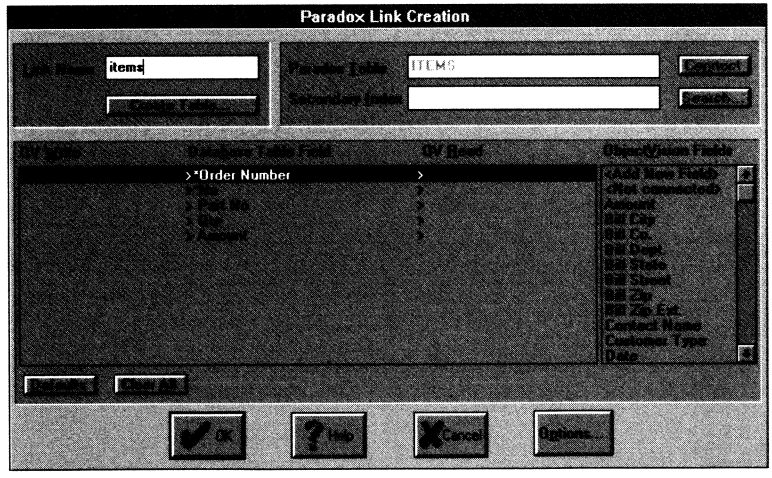
1. Click Links (shown at left) on the Object bar. The Data Links dialog box appears as shown in Figure 8.11.

Figure 8.11
The Data Links dialog box



2. Be sure that Paradox is checked as the Link Creation type, then click Create. The Paradox Link Creation dialog box appears.
3. Type `items` in the Paradox Table text box, then press *Tab*. Type `items` in the Database File Name text box, then click Connect or press *Enter*. The field names from the ITEMS table appear in the middle column, as shown in Figure 8.12.

Figure 8.12
The Paradox Link Creation dialog box after connecting "items"



Note that ITEMS has two index fields, Order Number and No, indicated by an asterisk before the field name. When a Paradox database table contains a multi-field index, the records are sorted more than once.

For example, the index on Order Number sorts the records according to the *order number*. Within each order, *item numbers* are sorted by the index on the No field.

In the *Order* application, the Sales Order number is associated with multiple Item Numbers which are kept in ascending order by the index on the No field.

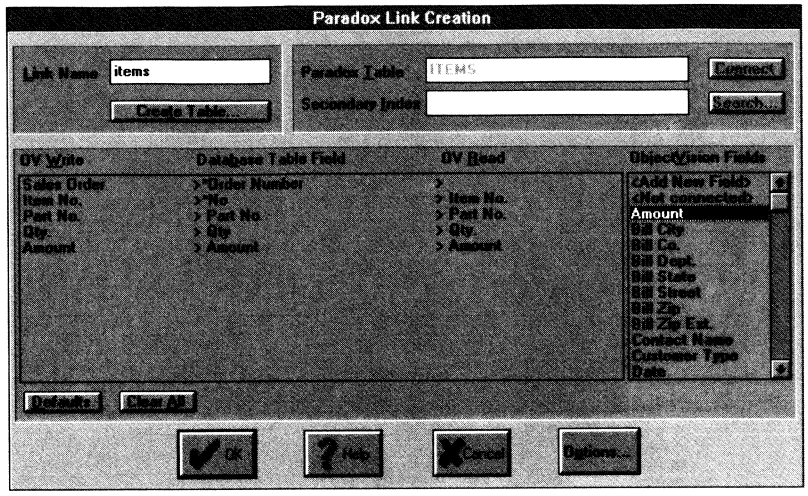
4. Define the ObjectVision field Sales Order as an OV Write field only. Select Sales Order from the ObjectVision Fields list box, then double-click in the OV Write column beside Order Number.

   The sales order number is not read back to the application field by the "items" link, so you should leave the OV Read field blank. (The Sales Order number is read into the ObjectVision field by the "orders" link.)

5. Define the ObjectVision field Item No. as an OV Read field and OV Write field connected to No (in the Database Table Field column).
6. Define the ObjectVision field Part No. as an OV Read field and OV Write field connected to Part No (in the Database Table Field column).
7. Define the ObjectVision field Qty. as an OV Read field and OV Write field connected to Qty (in the Database Table Field column).
8. Define the ObjectVision field Amount as an OV Read field and OV Write field connected to Amount (in the Database Table Field column).

The Paradox Link Creation dialog box should now appear as shown in Figure 8.13.

Figure 8.13
The completed Paradox Link Creation dialog box for "items"



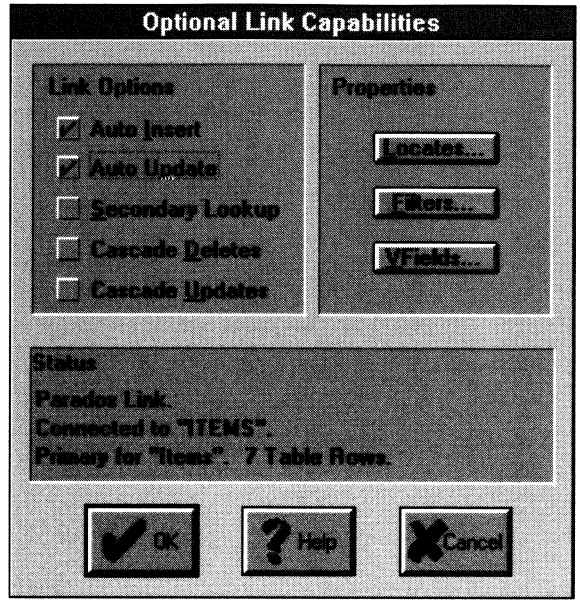
Now define link options for the "items" link.

1. Click Options.
2. Check Auto Insert and Auto Update.

   Each time the user moves to another row in the table object, the items information will be automatically written to the ITEMS.DB table.

   The Optional Link Capabilities dialog box now looks like Figure 8.14.

Figure 8.14
The Optional Link Capabilities dialog box for "items"



The Status box shows you information about the "items" link. The text `Primary for "Items"` lets you know this link is connected to columns and it is the first link to the table object. The text `7 Table Rows` indicates the number of rows in the table object.

Now define Locate fields for the "items" link.

1. Click the Locates button. The Locate Indexes dialog box appears with Order Number and No in the Table Index Field column.
2. Scroll to the Sales Order field name in the ObjectVision Fields list, and double-click it. Sales Order appears in the Locate Field column next to Order Number.

   The Locate Field isn't defined for No, the Table Index Field, because this link will locate items for an order based only on the sales order number (*not* the item number).

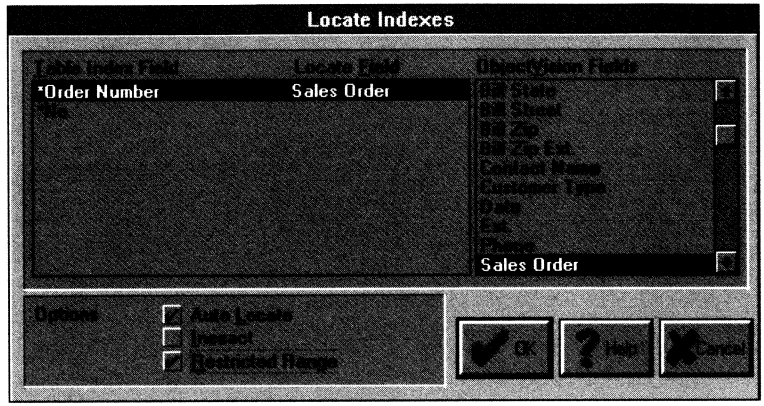
   The *Order* application will now monitor the Sales Order field. When the user types an existing order number in the Sales Order field, the link will deliver all the records in the ITEMS database table associated with that order number.
3. Select Restricted Range so that the link delivers records only for the precise Sales Order value. If you selected Inexact, the

link would deliver the closest record when the user tried to enter a new order. Inexact wouldn't let the user enter a *new* order number.

4. Leave Auto Locate checked.

   The completed Locate Indexes dialog box now appears as shown in Figure 8.15.

Figure 8.15
The Locate Indexes dialog box for "items"



5. Click OK two times to close the Locate Indexes dialog box, and the Optional Link Capability dialog box, then click OK to complete the link definition.
6. When you see the Link Automatic Buttons dialog box, click Cancel.

Adding the "parts" link

To complete the final link in *Order*, add the "parts" link. You need to create the link and define the link options as shown in the following figures.

The completed Paradox Link Creation dialog box appears as shown in Figure 8.16. The completed Locate Indexes dialog box appears as shown in Figure 8.17.

Note that Secondary Lookup is automatically checked for this link in the Optional Link Capabilities dialog box because "parts" is a Secondary Lookup link. When a part number appears in the Part No. column, "parts" automatically fills in the part description and price.

After you finish creating the "parts" link, close the dialog boxes. When the Link Automatic Buttons dialog box appears, click Cancel. Click OK to close the Data Links dialog box.

Figure 8.16
The completed Paradox Link Creation dialog box for "parts"

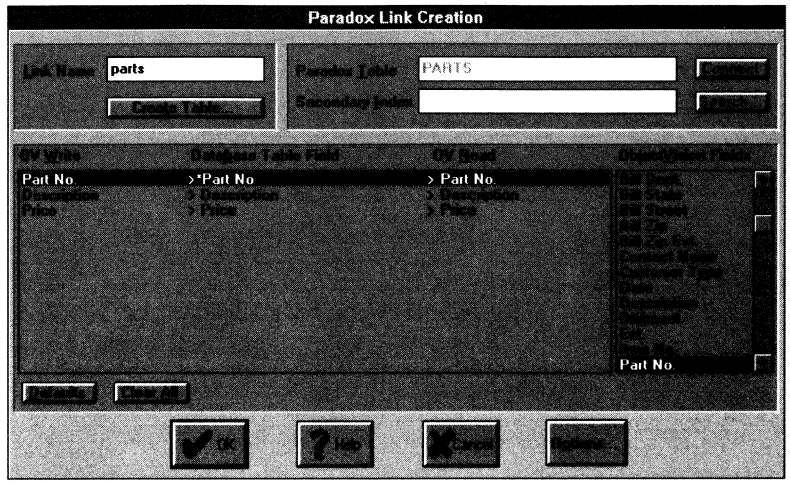
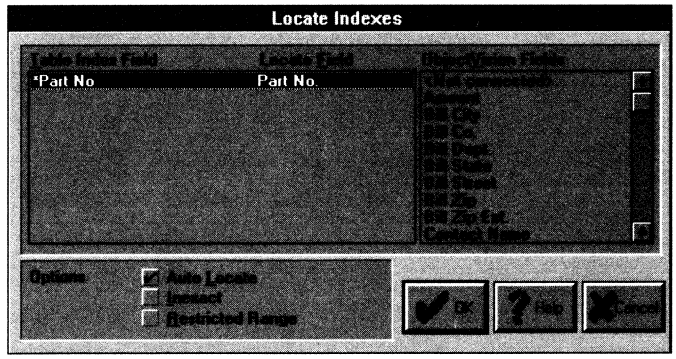


Figure 8.17
The completed Locate Indexes dialog box for "parts"



Testing your completed application

The *Order* application is now complete. Go ahead and enjoy the fruits of your labors!

1. If the Form Tool is still open, choose File | Close Tool.
2. Click the New Order button. If any values are in the fields or the table object, they are cleared from the forms (except the Date field).

3. Add yourself as a contact.
4. Click the New Customer button.
5. When the Customers form appears, add your company as the customer, and fill in the customer type and tax exempt information.
6. Click the Enter button to save the customer information to the CUSTOMRS table.
7. Click the Return to Order Entry button. When the Customers form closes, you'll see your company information in the BILL TO and SHIP TO areas.
8. Type `100`, `200`, or `501` in the Part No. column. The application fills in the Description and Price. These values are delivered by "parts" link.
9. Enter a quantity to complete the order, then click Enter Order.
10. Note the Sales Order number, then click New Order to clear the order information from the forms.
11. Type your new Sales Order number in the Sales Order field and press *Enter*.

    The Order information appears onscreen as a result of the links you created in this chapter.

    Congratulations!

Related topics for links

There are a lot of link options and table features that are not used in the *Order* sample application. The following sections touch on some of these topics.

Viewing database records

When a table object is selected during form completion, you can use function keys, the table scroll bar, or functions triggered by custom *Ctrl+letter* shortcuts to view database records, as shown in Table 8.2.

Table 8.2: Keys and buttons for positioning database records in table objects

Positioning action	Function key or button	Event tree equivalent
View previous record (if there is one)	*F3* or scroll-up arrow	@PREVIOUS
View next record (if there is one)	*F4* or scroll-down arrow	@NEXT
View the previous records (number of rows – 1)	*F5*	@PAGEUP
View the next records (number of rows –1)	*F6*	@PAGEDN
View first record in the link	*F7*	@TOP
Clear all records from the table object	*F8*	@CLEAR

Note that the function keys only work during form completion, when a table object is selected.

Note that a table object scroll bar works differently than other Windows scroll bars. The scroll box is *not* used to display values, and does *not* indicate the current row's position in the database.

Locates on a button

In the *Order* application, all Locates are defined as Auto Locates. If you want the user to trigger the locate manually, you can uncheck the Auto Locate option and add an event tree that includes the @LOCATE function.

For more about @LOCATE or any other ObjectVision function, see the *ObjectVision Reference Guide*, Chapter 8, "@Functions descriptions."

Virtual fields

When the databases for the Order application were first created, the tables were *normalized*. Normalizing is a database design strategy that arranges the data into very precise database structures in an attempt to minimize storage space and redundancy.

The ITEMS database was changed (so it *isn't* normalized) by adding the Amount field. Originally, the Amount field was a *virtual field*.

Virtual fields are a link option that extend a database table.

Virtual fields act like physical fields in a database but exist only in computer memory. VFields virtual fields calculate values based on any valid ObjectVision expression.

Virtual field expressions have the same characteristics as trees, but they can do things trees can't. For example, if a virtual field was used in *Order*, each value in the Amount column wouldn't need to be written to the database.

For more information about virtual fields, see the *ObjectVision Reference Guide.*

Creating tables with ObjectVision

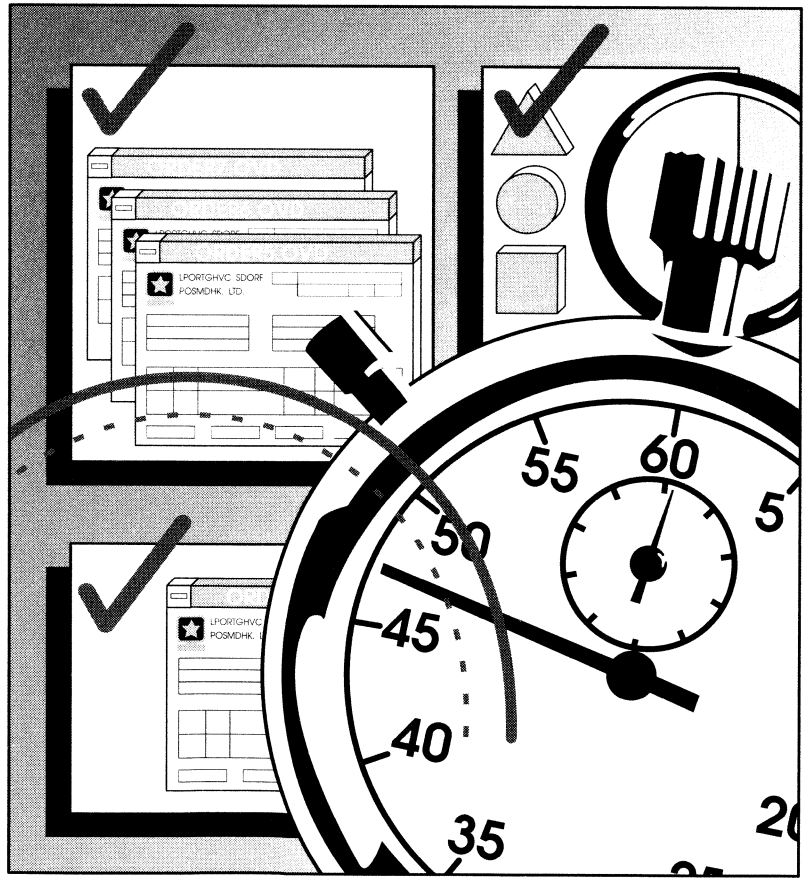
The four links you have already created—"customers," "orders," "items," and "parts"—have been to Paradox tables that already exist. You can also create Paradox, dBASE, and Btrieve tables and ASCII files from within ObjectVision.

For more information about creating tables, see Chapter 3, "Form design," in the *ObjectVision Reference Guide.*

A P P E N D I X

A

The short course



Appendix goals:

- Designing a form
- Assigning properties
- Creating a value tree
- Creating an event tree
- Creating a link

This short course is a condensed version of Chapters 4 through 8. It introduces the essential features of ObjectVision by directing you to put the finishing touches on a nearly complete *Order* application. For additional information on a particular task, look in the associated chapter of this manual.

What you need to complete this short course:
This appendix assumes you know how to use a completed ObjectVision application, as described in Chapter 3. To complete the short course, start by opening the file ORDERA.OVD.

Designing a form

This section summarizes Chapter 4, "Designing forms." If you need more information about creating or editing objects on a form, see Chapter 4.

Planning an application

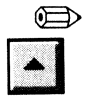
The objectives for the *Order* application are to

- Let the user display all details of an existing order.
- Let the user add a new customer to the database.
- Automatically supply an item's unit price and description.
- Automatically calculate values for particular fields.

Opening an application

An ObjectVision application contains one or more forms in which users enter values. The complete *Order* application has two forms: Order Entry and Customers.

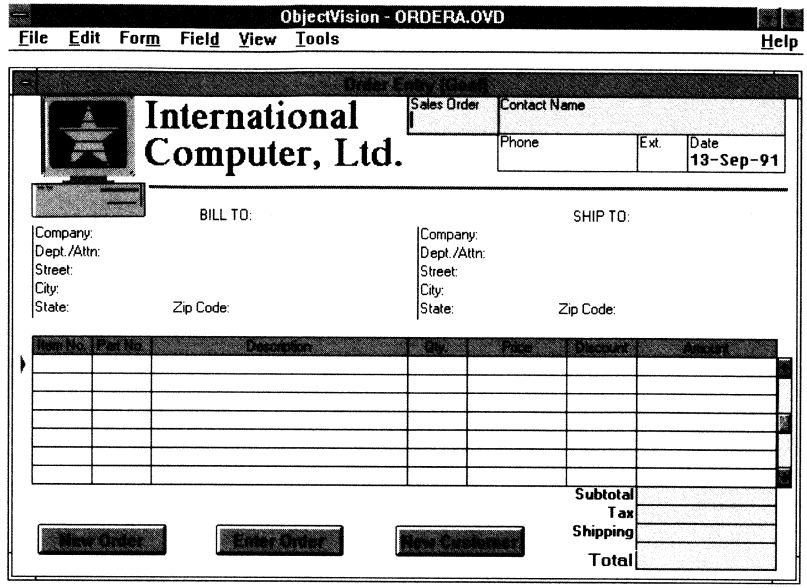
To open the *OrderA* application,



1. Start ObjectVision.
2. Click the Maximize button located in the top right corner of the ObjectVision window. This button (shown at left) enlarges the window so it fills the entire screen. You should maximize the window (if it isn't already) whenever you start ObjectVision in the short course and the window display is less than full screen.
3. Choose File | Open, select the directory where you installed the sample files, and then choose ORDERA.OVD.

The Order Entry form of the *OrderA* application appears, as shown here.

Figure A.1
The OrderA application



Creating a form

In *OrderA*, the Order Entry form is nearly complete. In this short course, you'll finish the Order Entry form and create the Customers form.

To create the Customers form,

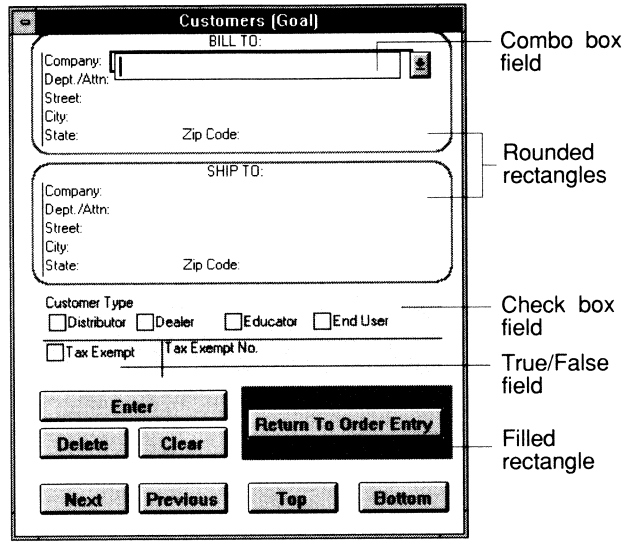
1. Choose Tools | Form. When you edit in the Form Tool, you use ObjectVision to create objects and assign properties to those objects—*not* to enter values into fields.
2. Choose Form | New.
3. Type the name for the form, `Customers`, and click OK.
4. Make the form wider and taller by dragging a corner of the form border. Size the Customers form so it's as tall as the screen and one-half of the screen width.

Creating objects on a form

ObjectVision forms can contain any combination of fields, buttons, table objects, text objects, rectangles, lines, and graphics. For more information about object types, see Chapter 4.

The completed Customers form you'll create is shown in Figure A.2.

Figure A.2
Objects on the completed Customers form



To add an object to a form you can copy and paste an existing object or create a new object.

Fields, labels, and text

The first objects you'll add to the Customer form are fields and text objects. Fields are objects that display values, and text objects are used to identify objects on the form.

In the BILL TO and SHIP TO areas, text objects are located next to the fields and identify what the fields contain.

Field names can be displayed as labels or hidden. The advantage of using labels instead of text objects is that they are created automatically when you create the field. The advantage of using text objects is that you can place the text anywhere on a form in relation to a field.

Copying and pasting objects

To add the BILL TO and SHIP TO fields and text objects to the Customers form,

1. Choose Form | Select, then double-click Order Entry in the Form Name list.
2. Select all objects in the BILL TO and SHIP TO area. Click the BILL TO: text object, press and hold down *Shift*, then click Ship Zip Ext., the last field in the SHIP TO area.

Shortcut
Ctrl Ins

3. Choose Edit | Copy. The field and its properties are copied to the Clipboard, a holding area for the last object copied.
4. Now make the Customers form the active form by choosing Form | Select and double-clicking Customers.

Shortcut
Shift Ins

5. Choose Edit | Paste to add the copied objects to the form. A dotted border the same size as the original fields appears on the form.
6. Position it where you want the copies to appear—near the top of the Customers form—then click. The copies appear and automatically enlarge the size of the form.
7. Resize the form width so it's slightly larger than the pasted objects.

Shortcut
Shift Del

8. Select all objects in the SHIP TO area of the Customer form, cut them, then paste them below the BILL TO area.

Creating a new field

To create the remaining fields on the Customer form,

1. Click Field (shown at left) on the Object bar. The Field Name dialog box appears with the default name Field1.
2. Click the combo box arrow, scroll to the field name `Customer Type`, and then press *Enter*. This field already exists because it was automatically created when the value tree for Discount was created. When a field value is evaluated in a value tree, the field is automatically created if it doesn't exist.

Caution!

Be sure to use a unique name when typing a name for a new object.

3. The field crosshair pointer appears. Position the pointer under the SHIP TO area, then drag to size the field.

4. Repeat steps 1 and 2, selecting *Tax Exempt* instead of Customer Type, to create the Tax Exempt field. The Tax Exempt field was created automatically in the value tree for Tax.
5. Repeat step 1, type `Tax Exempt No.`, and then press *Enter* to create the Tax Exempt No. field.

Adding buttons

Button objects let the user simply click a button to initiate a specific action. In the *Order* application, for example, users click the New Customer button on the Order Entry form to open the Customers form.

The Customers form needs a button to let the user return to the Order Entry form after a new customer is added to the database.

To add the Return to Order Entry button,

1. Click Button (shown at left) on the Object bar.

2. In the Button Name text box, type `Return to Order Entry`, then click OK.
3. Position the button crosshair pointer at one corner of the button you want to create, then drag to the opposite corner. The Return to Order Entry button now appears. If the entire button name isn't displayed, drag the handles to enlarge the button.

Adding rectangles

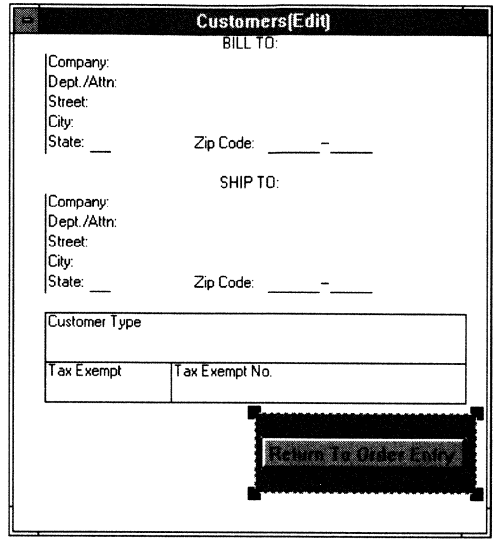
A form's appearance can be enhanced by outlining important areas with square or rounded rectangles. The Customers form needs a gray rectangle accenting the Return to Order Entry button, and rounded rectangles delineating the BILL TO and SHIP TO.

To add a rectangle to the Customers form,

1. Create a rectangle by clicking Rectangle (shown at left) on the Object bar.
2. Position the rectangle crosshair pointer where you want one corner of the rectangle to appear behind the Return to Order Entry Button, then drag to the opposite corner.

The Customers form now appears as shown in Figure A.3.

Figure A.3
Fields with default properties on the Customers form



To add rounded rectangles to the Customers form,

1. Click Rounded Rectangle (shown at left) on the Object bar. The rounded rectangle crosshair pointer appears.
2. Position the pointer where you want one corner of the rectangle to appear, then drag to the opposite corner. A shaded rectangle appears. You'll want to change the fill pattern property of the rectangle from light gray to white. Properties are covered in the section, "Defining properties."
3. To copy this rounded rectangle, choose Edit | Copy.
4. Paste the rounded rectangle by choosing Edit | Paste, using the mouse to position the copy, then clicking.

Other form design topics

The related form design topics are discussed in Chapter 4.

- field order and guided completion
- the relationship between field length and font size
- creating table objects
- default database definition
- adding graphics

Defining properties

This section summarizes topics covered in Chapter 5.

This section explains how to assign properties to the objects you create. Properties include object names, field types, display attributes, data formats, value trees, event trees, font characteristics, borders, and help text.

Every object, form, and stack of forms has a set of properties, which you can inspect by clicking it with the right mouse button.

The sections, "Creating value trees" and "Creating event trees" later in this appendix explain how to assign a special class of properties: value trees and event trees.

ObjectVision field types

There are three basic categories of ObjectVision field types:

- **Character field types** are used for holding text values. General is the default ObjectVision field type. It's used for fields that can contain both letters and numbers—for example, the fields in the SHIP TO and BILL TO areas.
- **Numeric field types** are used for displaying financial, currency, date, or time values. For example, the Price, Amount, Subtotal, Tax, Shipping, and Total fields on the Order Entry form are financial field types.
- **Selection Method field types** control how users select a value from a predefined set of options. The completed Customers form has three fields with selection methods:
  - Combo Box is the Bill Co. field type (this is the field in the BILL TO area labeled *Company*). The user can type a new company name or select an existing one from the list.
  - Check boxes is Customer Type field type. ObjectVision inserts a check box before each value option. The user selects a value by checking the box next to the desired value.
  - True/False is the Tax Exempt field type. ObjectVision inserts a single check box before the name of a field. The user checks the box to indicate a *Yes* value, or leaves it unchecked to indicate a *No* value.

Assigning field types

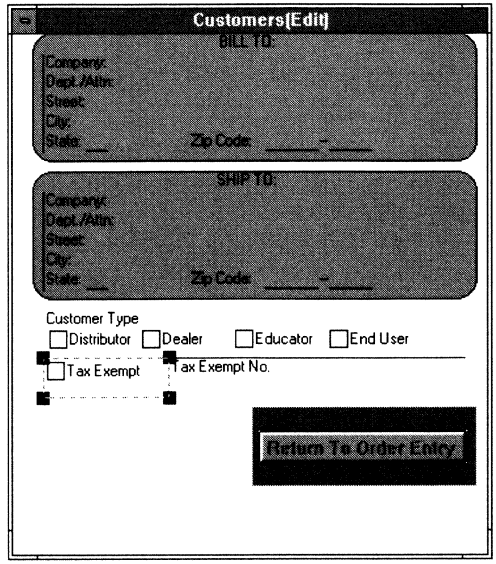
Assign selection method properties to fields on the Customers form:

1. Be sure the Customers form is still selected and the Form Tool is active.

2. With the pointer on the Customer Type field, click the right mouse button to inspect its properties.
3. Choose Field Type.
4. In the Field Type dialog box, choose Check Boxes from the Selection Method list, then click OK. The Expected List dialog box appears. A list of values is displayed because they are automatically read from the conditions for the Customer Type field specified in the value tree for Discount.
5. Click OK to close the Expected List dialog box.
6. Repeat steps 2 and 4, this time assigning the True/False field type to the Tax Exempt field. It isn't necessary to type expected values for True/False fields.
7. Now assign the Combo Box field type to the Bill Co. field (the field in the BILL TO area labeled Company).
8. When the Expected List dialog box appears, click OK to accept the default settings. The value options will appear after you create the "customers" link in the "Creating links" section.

The Customers form should now appear as it does in Figure A.4:

Figure A.4
The Customer form with assigned object properties



Protecting field values

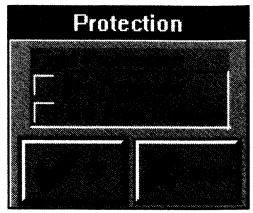
You can restrict users from entering values or viewing value trees by assigning a protection property to a field.

For example, in the *OrderA* application, users are unable to type values into either the BILL TO or SHIP TO area fields on the Order Entry form. When you copied these fields to the Customers form, their properties were also copied.

Follow these steps to *remove* protection:

1. Select all the fields in the BILL TO and SHIP TO areas except the Bill Co. field. Select the BILL TO area text object, press and hold down *Shift*, then click Ship Zip Ext., the last field in the SHIP TO area.
2. Hold down *Ctrl*, then click Bill Co., the first field in the BILL TO area. This deselects the Bill Co. field while leaving all the other fields selected.
3. Click the right mouse button to inspect the properties of the selected fields.
4. Choose Protection. The Protection dialog box appears (Figure A.5).

Figure A.5
The Protection dialog box with default settings



5. Click the No Override option to uncheck it as shown in Figure A.5, then click OK. If the Bill Co. field is still selected, the No Override option is not checked.

Applying fill patterns

The rectangles you created on the Customers form have a default fill pattern. You can remove that pattern.

To change a fill pattern,

1. Select the rectangles.
2. Click the rectangles with the right mouse button to inspect their properties.
3. Choose Fill Pattern.
4. Choose the white fill pattern.
5. Click OK.

Set the fill pattern for the rectangle under the Return to Order Entry button to dark gray.

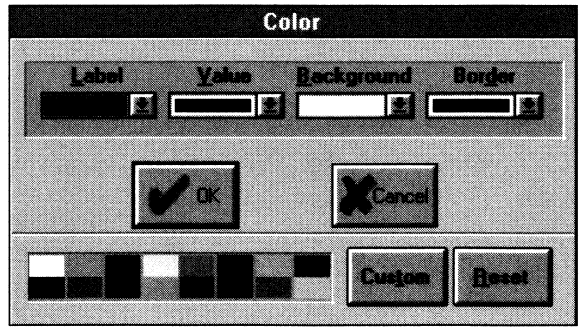
Applying color

You can color objects on your ObjectVision forms to highlight certain information or to show relationships among fields.

Follow these steps to apply color to fields:

1. Select the Customer Type, Tax Exempt, and Tax Exempt No. fields, then click the right mouse button to inspect their properties.
2. Choose Color. The Color dialog box appears as shown in Figure A.6.

Figure A.6
The Color dialog box



3. Click the Background combo box arrow, then click the dark yellow color.
4. Click OK. The new background color appears in the fields.
5. Define the color settings for the remaining fields on the form as shown in Table A.1.

Table A.1
Color settings for the fields on the Customers form.

Customers form	Background	Value	Border	Label
Rounded rectangles	White		Gray	
Customer Type	Pale yellow		None	
Tax Exempt No.	Pale yellow		Black (Left, Top)	
Tax Exempt	Pale yellow		Black (Top)	
Rectangle under button	Dark gray			

Adding help text

To help users complete a form, you can create online help for any field. The user can display the help text during form completion by selecting the field and pressing *F1*.

To add help information to the Customer Type field:

1. Position the pointer on the Customer Type field, then click the right mouse button to inspect its properties.

2. Choose Help. The Help Text dialog box appears.
3. Type the following text, pressing *Ctrl+Enter* to begin a new line:

```
Customer type is defined as follows:
Distributor - Signed distributor agreement on file
Dealer - Must provide valid tax ID
```

```
Educator - Recognized college or university
End User - Any individual or company not listed above
```

4. Press *Enter* or click OK to attach the help text to the field.

Other properties you can assign

These properties-related topics are discussed in Chapter 5.

- Line width
- Font characteristics
- Borders
- Property defaults

Creating value trees

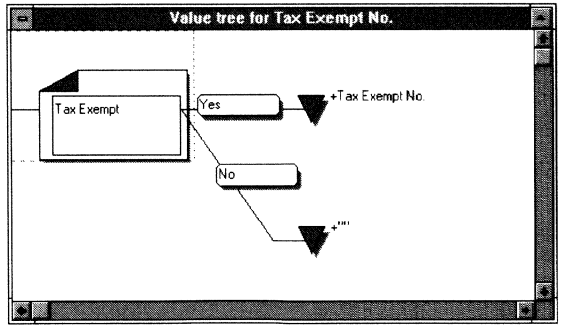
This section summarizes topics discussed in Chapter 6, "Creating value trees."

A value tree is a special type of field property.

Fields contain values that are entered by the user, calculated by the application, or connected by links to external data sources. Value trees calculate field values based on other values.

In the completed *Order* application, the Tax Exempt No. field gets its value from a value tree based on the Tax Exempt field, as shown in Figure A.7.

Figure A.7
The completed value tree for the Tax Exempt No. field



How value trees work

Value trees are made of branches, conditions, and conclusions. You can think of a branch as a value, a condition as an "if" statement, and a conclusion as a "then" statement.

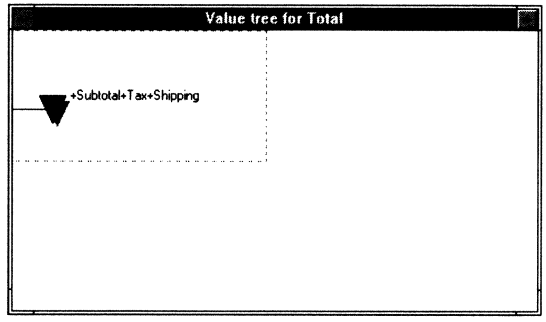
For example, the completed value tree for the Tax Exempt No. field can be thought of as an if-then statement that reads:

> "If the value for Tax Exempt is Yes (True), then have the user enter a value. If the value for Tax Exempt is No, then a tax exempt number isn't needed."

This simple tree shows one way you can help users by controlling the ObjectVision field order. When the customer isn't tax exempt, a null string (a pair of double-quote marks) is used to give the Tax Exempt No. field a value. Because this field has a value (which doesn't display), it is automatically skipped.

If the conclusion for a value tree is always the same, a value tree won't need to evaluate a field value. For example, as shown in Figure A.8, the Total field simply sums the values in the Subtotal, Tax, and Shipping fields.

Figure A.8
A value tree for Order's Total field, which has no condition

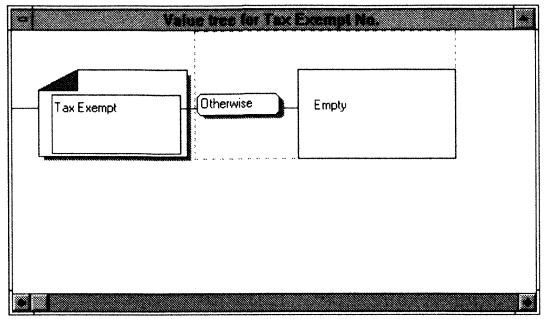


Creating the value tree for Tax Exempt No.

To create the value tree for the Tax Exempt No. field,

1. Be sure the Customers form is selected and the Form Tool is open.
2. Position the pointer on the Tax Exempt No. field and click the right mouse button to inspect its properties.
3. Choose Value Tree. The value tree for Tax Exempt No. dialog box appears—it's currently empty.
4. To add a branch, click Branch (shown at left) on the Object bar. A dialog box appears containing all Field Names in the application.
5. Scroll down the list and select Tax Exempt, then click OK. A branch now displays in the value tree, as shown in Figure A.9.

Figure A.9
Adding a new branch to Order's Tax field's value tree

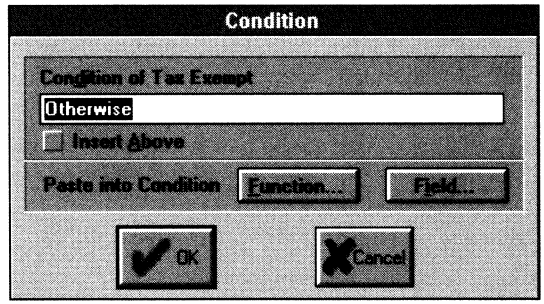


6. Click Conclusion (shown at left) on the Object bar.

   Note that first the Condition dialog box appears so you can define the logic that will lead to this conclusion, then the Conclusion dialog will appear.

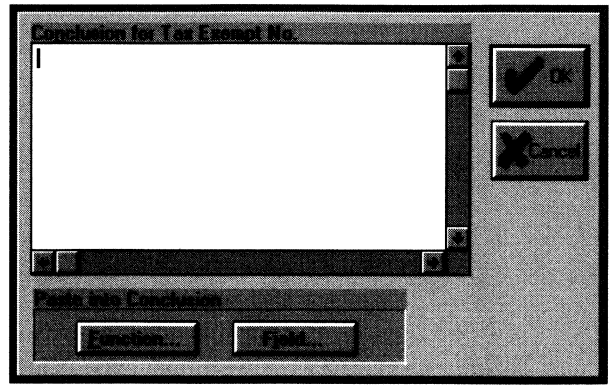
   The Condition dialog box appears as shown in Figure A.10. You'll enter the condition of the Tax Exempt value here.

Figure A.10
The Condition dialog box for the Tax Exempt branch



7. Type `Yes` or `True`, then click OK. The Conclusion for Tax Exempt No. dialog box appears as shown in Figure A.11.

Figure A.11
The Conclusion dialog box for Tax Exempt No.

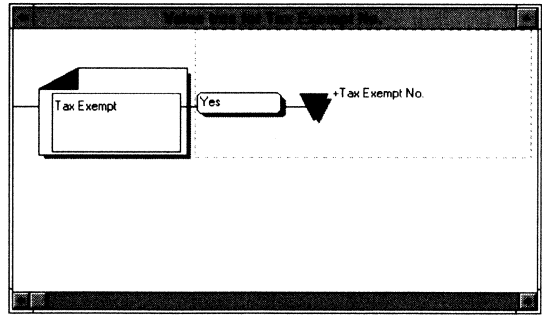


8. Enter the conclusion `+Tax Exempt No.` Or, type `+`, click the Paste Field button, double-click Tax Exempt No. to paste the field name into the conclusion, then press *Enter*.

Caution! If you type a field name incorrectly, ObjectVision *automatically creates a new field* with that name. This new field doesn't appear on a form, but appears on a pop-up Scratchpad form if its value is required in a value tree.

The value tree now appears as shown in Figure A.12.

Figure A.12
The first branch of the Tax Exempt No. value tree



Begin adding another branch

9. Next, add the second condition by clicking Conclusion.
10. Type `No` or `False` in the Condition text box and click OK. When the Conclusion for Tax text box appears, type `+""` (a plus sign followed by a pair of double-quotes).
11. Click OK. The value tree is now complete, as shown in Figure A.7 on page 159. Choose View | Expand or View | Reduce to change the display of the tree.

12. To close the value tree, double-click its Control-menu box or click Close Tool (shown at left) on the Object bar.

Testing a value tree

To test the logic in your value tree,

1. Close the Form Tool.
2. Open the Customers form, check Tax Exempt, then press *Enter*. The Tax Exempt No. field is selected.
3. Select the Tax Exempt field, uncheck the box, and then press *Enter*. The focus is on the BILL TO area Bill Co. field because the Tax Exempt No. field has been skipped automatically.

   The value tree logic is correct.

Other value tree topics

These value tree-related topics are discussed in Chapter 6.

- Adding @functions to value trees
- Editing value trees
- Printing value trees
- Creating complex value trees

Creating event trees

This section summarizes topics discussed in Chapter 7, "Creating event trees."

Like a value tree, an event tree is a special type of field property.

An event tree defines the action ObjectVision should take in response to a user event. Create event trees whenever you want an application to respond to a

- click
- custom shortcut key
- change in a field value
- stack event
- form event

What @functions do

In both value and event trees, @functions return values based on calculations that can't be expressed by the string operator & or mathematical operators such as +, –, *, and /.

In event trees, @functions can also instruct the application to carry out ObjectVision actions such as menu command equivalents.

You'll use the event functions @FORMSELECT and @RESUME when you write the event tree for the Return to Order Entry button.

Adding an event tree to a button

To add an event tree to Return to Order Entry,

1. Open the Form Tool if it isn't open already.
2. If the Customers form isn't displaying onscreen, choose Form | Select, then double-click Customers in the list of form names.
3. Double-click the Return to Order Entry button to open its event tree.

   The empty event tree for the Return to Order Entry button appears.
4. Click Conclusion (shown at left) on the Object bar. The Event Name dialog box appears as shown in Figure A.13.

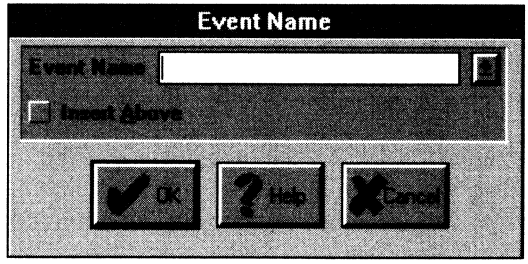
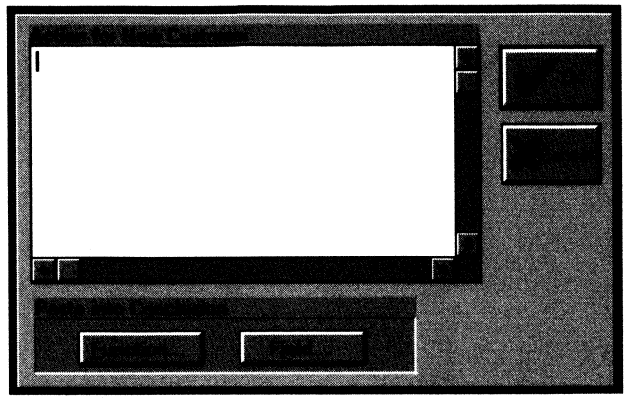


Figure A.13
The Event Name dialog box for the Return to Order Entry button

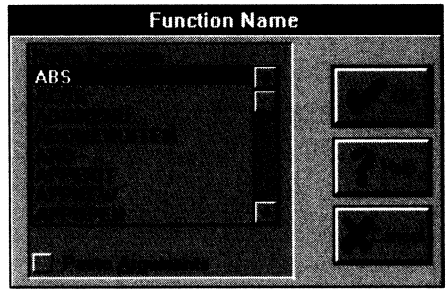
5. Type `click`. Or, click the combo box arrow and select Click as the Condition of Event. Click is a reserved keyword for events in ObjectVision.
6. Click the OK button. The Action for Return to Order Entry dialog box appears as shown in Figure A.14.

Figure A.14
The Action for Return to Order Entry dialog box



7. Click Paste Function to display the Function Name list. The Function Name dialog box appears, as shown in Figure A.15.

Figure A.15
The Function Name dialog box



Tip: When Paste Arguments is checked, the argument list for a function is pasted into the dialog box in addition to the function name.

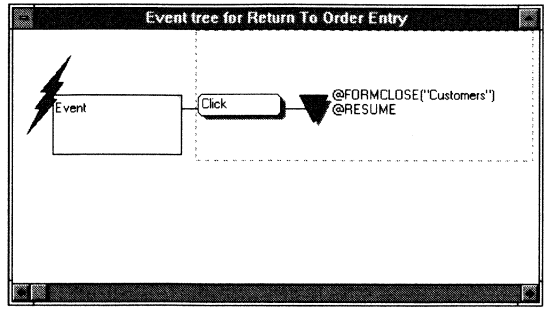
8. Scroll down the Paste Function list and select FORMCLOSE, check Paste Arguments, then click OK.

Be sure to include the double-quotes.

9. Replace the argument `"FormName"` with `"Customers"` and then press *End* to move the pointer to the right of the expression. Press *Ctrl+Enter* to begin a new line.
10. Click Paste Function again, select RESUME, and then click OK. Notice that @RESUME has no arguments.
11. Click OK to finish creating the event tree.

The event tree for the Return to Order Entry button now appears as shown in Figure A.16.

Figure A.16
The completed event tree for Order's Return to Order Entry button



Testing the event tree

To test the event tree, close the event tree by double-clicking the tree's Control-menu box. Then close the Form Tool by choosing File | Close Tool.

Click the Return to Order Entry button. The Order Entry form appears and the Sales Order field is selected.

Creating links

This section summarizes topics covered in Chapter 8, "Creating links."

ObjectVision applications can link to data files or other Windows applications.

Links are dynamic relationships between ObjectVision applications and external data sources. For example, by connecting an ObjectVision field to a linked database table, values can be read from *and* written to the database.

When you first create a link, you can let ObjectVision automatically create buttons complete with the appropriate link function and link name in the event trees. These buttons let the user browse through records in the linked database.

What a database link can do

A database link lets an application:

- **Read values** by getting the information from the database table and displaying it in the connected ObjectVision fields. ObjectVision automatically displays all associated values for a record. For example, in the *Order* application, the user can view the names and addresses of all companies that have placed orders.

- **Write values** by saving the information in the ObjectVision connected fields to the database. For example, in the *Order* application, new customer information entered on the Customers form is written to CUSTOMRS.DB, a database table.
- **Simplify viewing and updating values** by letting users click buttons to locate and edit records in the database. For example, in the *Order* application, users can browse through the records of existing customers by clicking the Top, Next, Bottom, and Previous buttons.

Linking to ObjectVision fields

In this section, you'll create "customers," a simple link between the *OrderA* application and CUSTOMRS.DB, a Paradox database table.

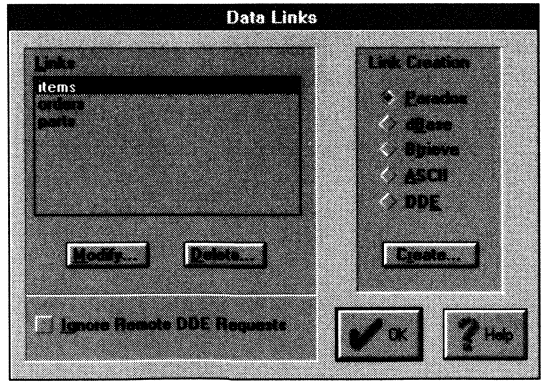
This link will let the user accomplish two functions: The user will be able to save new customer information to the database, and will also be able to view and edit existing customer information.

To connect your application fields to CUSTOMRS.DB, a Paradox database table,

1. Be sure the Customers form is the active form and the Form Tool is active. The link buttons (Enter, Next, Previous, Delete, Clear, Top, and Bottom) will be automatically placed *on the active form* after the link is created.

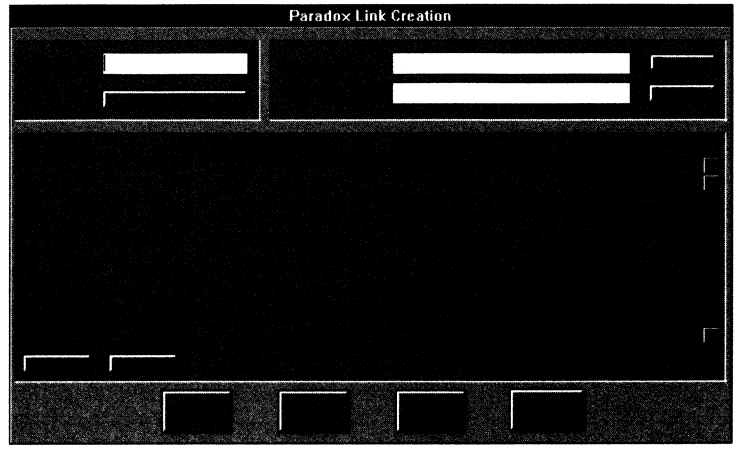
2. Choose Links (shown at left) from the Object bar. The Data Links dialog box appears as shown in Figure A.17.

Figure A.17
The Data Links dialog box



3. Be sure that *Paradox* is checked in the the Link Type Creation list, then click Create. The Paradox Link Creation dialog box appears as shown in Figure A.18.

Figure A.18
The Paradox Link Creation dialog box



4. Type `customers` in the Link Name text box, then press *Tab*.
5. Type `customrs` (note spelling due to the DOS 8-character limit) in the Paradox Table text box, then press *Tab*. If the data file isn't in the same directory as your ObjectVision application, you *must* specify the path name.

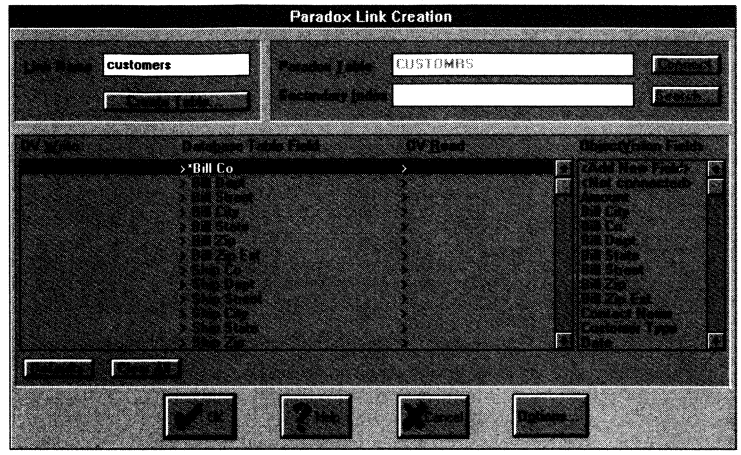
It is not necessary to include the .DB extension.

6. Leave the Secondary Index text box empty.
7. Click Connect or press *Enter*. If you typed a name other than `customrs` for the table name (note spelling), ObjectVision displays the message `Table was not found`.

   Your application connects to the Paradox table name you entered, listing its fields in the Database Table Field column as shown in Figure A.19.

   Notice that the Bill Co field is marked with an asterisk (*). The asterisk denotes it as an *index field*. The index field contains the unique value with which the other values in the record are associated.

Figure A.19
The CUSTOMRS.DB fields

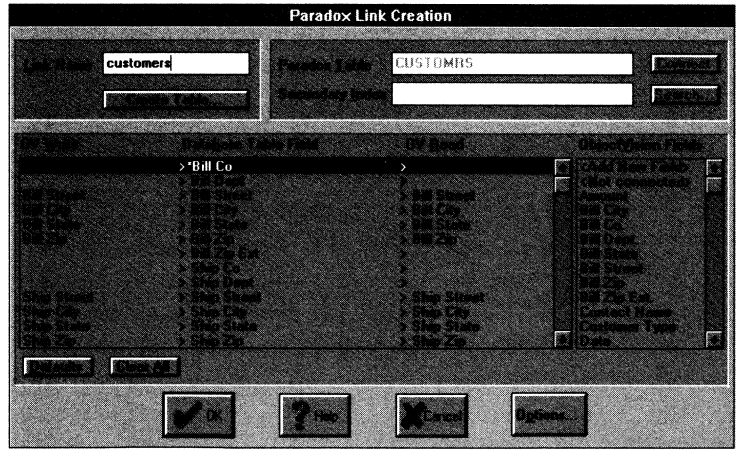


8. Create connections between the listed Database Table Fields and the ObjectVision fields in your application:

- To quickly create connections with those ObjectVision fields with the same names as the Paradox Table fields, click Defaults.

  The OV Read and OV Write columns now contain those fields from the ObjectVision application that are spelled identically to the database table fields, as shown in Figure A.20.

Figure A.20
The default connections for "customers"

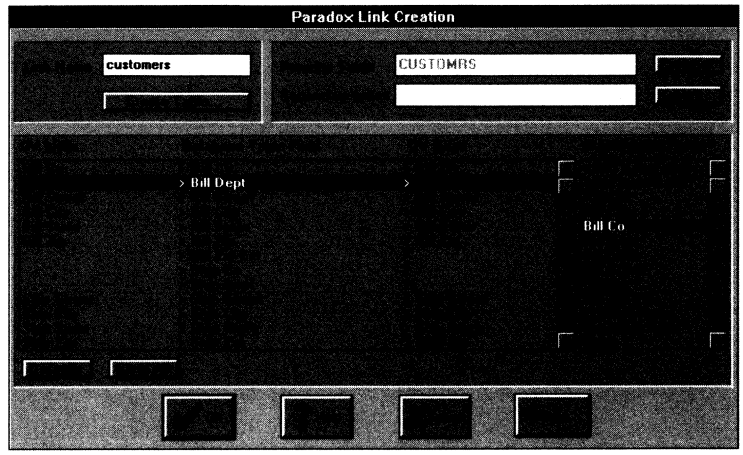


- To create links for those database fields that are not spelled the same as their corresponding ObjectVision fields, select a Database Table Field, then double-click a field name in the ObjectVision Fields list.

For example, to connect the Paradox Table field Bill Co with the ObjectVision Field Bill Co., select the Database Table Field Bill Co. Then double-click Bill Co. in the ObjectVision Fields list, *or* select Bill Co. and then double-click the Database Table Field column.

Bill Co. now appears in the OV Read and OV Write columns, and the next row is selected, as shown in Figure A.21.

Figure A.21
The Bill Co. field connected to the database



- Use this same technique to connect the other OV Read and OV Write fields to the Database Table Fields that are not yet connected.

Defining Locate fields

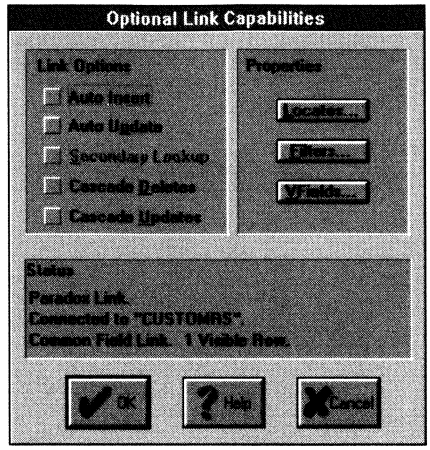
Locate fields are *the most important* ObjectVision link option. A Locate field is an ObjectVision field that triggers a link. When a new value is entered in a Locate field, the other values in that link are automatically read into the application.

For example, the Locate field for the "customers" link is the Bill Co. field. ObjectVision monitors the Bill Co. field and delivers the company information when the value in Bill Co. changes.

To specify the Locate field for the "customers" link,

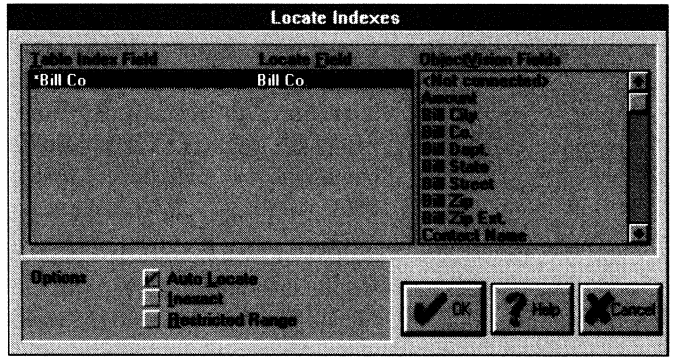
1. Click the Options button near the bottom of the Database Link dialog box. The Optional Link Capabilities dialog box appears as shown in Figure A.22.

Figure A.22
The Optional Link Capabilities dialog box



2. Click the Locates button in the Optional Link Capabilities dialog box. The Locate Indexes dialog box appears (Figure A.23).

Figure A.23
The Locate Indexes dialog box



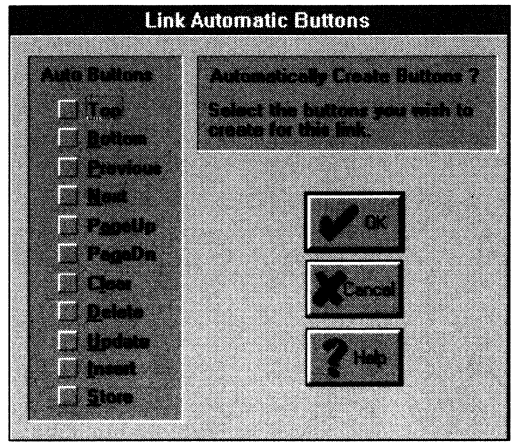
Notice the Paradox table index field, Bill Co, appears in the left column, beneath the heading Table Index Field. The ObjectVision field, Bill Co., appears in the right column beneath the heading Locate Field.

3. Leave Auto Locate checked. Auto Locate is checked by default. This option means that whenever a new value is entered in the Locate field, the link automatically fills in the associated values, if they exist.
4. Now check Restricted Range as another Locate option.

   When Restricted Range is checked, the application delivers only those records that *exactly* match the entered value.

5. Click OK three times to close the Locate Indexes dialog box, the Optional Link Capabilities dialog box, and the Paradox Link Creation dialog box. The Link Automatic Buttons dialog box appears as shown in the following figure:

Figure A.24
The Link Automatic Buttons dialog box



6. Check Top, Bottom, Previous, Next, Clear, Delete, and Store, then click OK to create the buttons and close the Link Automatic Buttons dialog box. Click OK to close the Data Links dialog box. Automatically created buttons are placed at the bottom of the current form.
7. Scroll the form to view the buttons and arrange them to fill the empty space next to the Return to Order Entry button.
8. Replace the button name Store with Enter, using Properties | Name/Text to rename the field.

Testing link buttons

The *Order* application is now complete. Now use the application by closing the Form Tool.

1. Be sure the Order Entry form is the active form, then click the New Order button. If any values are in fields or the table object, they are all cleared from the forms.
2. Type a new number (greater than 6) into the Sales Order field and press *Enter*.
3. Add yourself as a contact, add your phone number and extension, and then click the New Customer button.

4. When the Customers form appears, add your company as the customer, then click the Enter button to save the customer information to the CUSTOMRS database table.
5. Click the Return to Order Entry button. When the Customers form closes, you'll see your company information in the BILL TO and SHIP TO areas.
6. Type `100`, `200`, or `501` in the Part No. column. The application fills in the Description and Price. These values are delivered by "parts" link. Enter a quantity to complete the order, then click Enter Order.
7. Note the Sales Order number, then click New Order to clear the order information from the forms.
8. Type your new Sales Order number in the Sales Order field and press *Enter*. The Order information appears onscreen as a result of the links you created in this chapter.

Other linking topics

These related topics are discussed in Chapter 8, "Creating links":

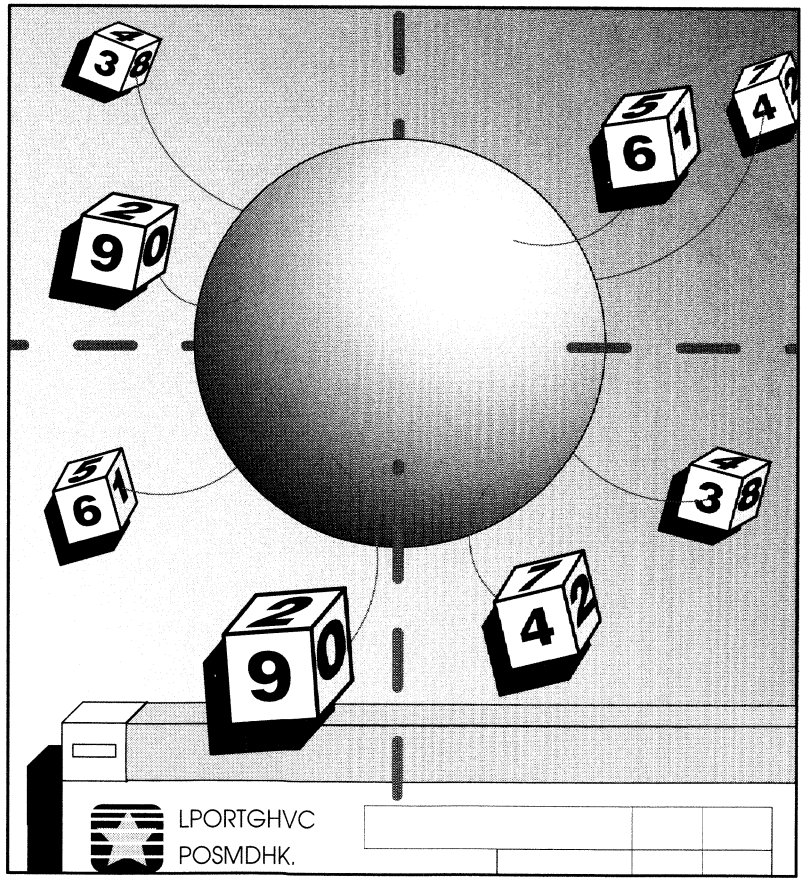
- External data source types
- Multiple links to a table object
- Linking options
- Optimizing your database design
- Creating database tables within ObjectVision

Linking is a very powerful ObjectVision feature—you might also want to refer to Chapter 10, "Linking basics," and Chapter 11, "Linking options," in the *ObjectVision Reference Guide*.

A P P E N D I X

B

Configuring the Paradox Engine



Appendix goals:

- Opening the Configuration Utility
- Understanding the configuration options
- Setting and saving your configuration settings

ObjectVision uses the Paradox Engine to provide access to Paradox tables. This open architecture lets ObjectVision share data with Paradox and any other application using the Paradox Engine.

When you use ObjectVision to access Paradox data located on a network, or share local data with Paradox running in the DOS box, ObjectVision uses the Paradox Engine. You only need to use the Paradox Configuration Utility if you aren't satisfied with the default ObjectVision settings.

Note DOS SHARE must be loaded before you start Windows. Otherwise, the Engine locking functions won't be operative. If you need details on SHARE, see your DOS manual.

Using the Configuration Utility

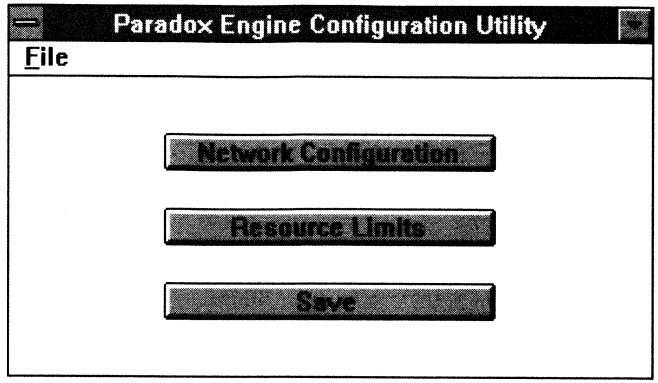
The Paradox Engine Configuration Utility is a program called PXENGCFG.EXE that runs under Windows 3.0. It lets you modify settings in WIN.INI that control how the Paradox Engine operates.

During installation, the PXENGCFG.EXE file is copied to your ObjectVision system directory. You can start this utility by choosing File | Run from the Program Manager, typing `C:\directory\PXENGCFG` in the Command Line text box, and then pressing *Enter*. *Directory* is the name of the directory you specified during installation as the ObjectVision system files directory. The default system directory is VISION.

Or, you can create a new Program Item in the Program Manager for the Paradox Engine Configuration Utility, then double click the PXENGCFG icon to start the application.

After you start the Configuration Utility, the Main menu window appears as shown in Figure B.1:

Figure B.1
Main menu window for the Paradox Engine Configuration Utility



The File menu contains the following commands: Network Configuration, Resource Limits, Save, About, and Exit. Choose these commands by opening the File menu and choosing the command you want. Alternatively, you can also choose one of the buttons displayed on the Main menu: Network Configuration, Resource Limits, or Save. You can exit the Configuration Utility by choosing Exit from the File menu, or by double-clicking the Control-menu box in the upper left corner of the window.

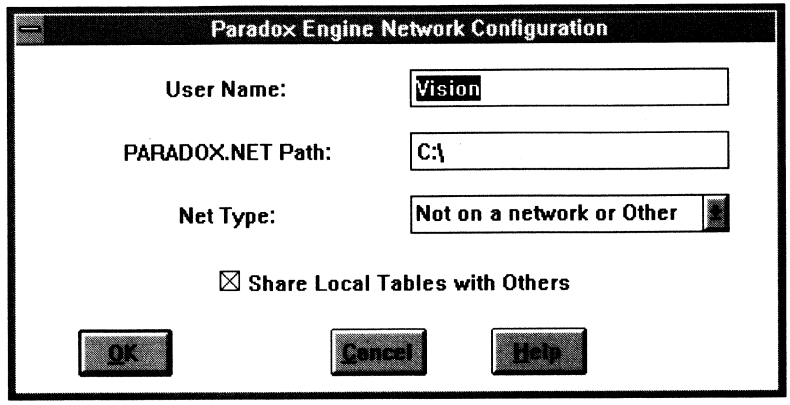
When you choose a command, you'll see a dialog box with the current values for specific variables. Edit these variables to suit your environment. Choosing Save makes your changes permanent. Closing the Configuration Utility without choosing Save leaves the original settings unchanged.

Network configuration

The Network Configuration dialog box specifies the file-sharing characteristics of the tables accessed by the Engine.

Selecting any line highlights the entire line for editing. Choosing the Help button offers an explanation of the currently highlighted line (its effects and legal values).

Figure B.2
The Network Configuration dialog box



- **User Name** lets you enter a user name. If the Paradox Engine locks a table, other Paradox or Paradox Engine applications attempting to access that table will be told, "Table *Name* is locked by user *UserName*." The current value is stored in WIN.INI; its default value is Vision.
- **PARADOX.NET Path** lets you specify the location of the PARADOX.NET file. If users are using ObjectVision (or the Paradox Engine) on tables located on a network server, they must obtain the location of PARADOX.NET from the network administrator. Users using the Paradox Engine on tables located on their own PC should use the default setting, C:\.
- **Net Type** lets you specify your network type by selecting from a list:

  Not on a network or Other
  Novell NetWare
  3Com 3+
  3Com 3+Open
  IBM PC LAN
  StarGROUP (AT&T)
  Banyan

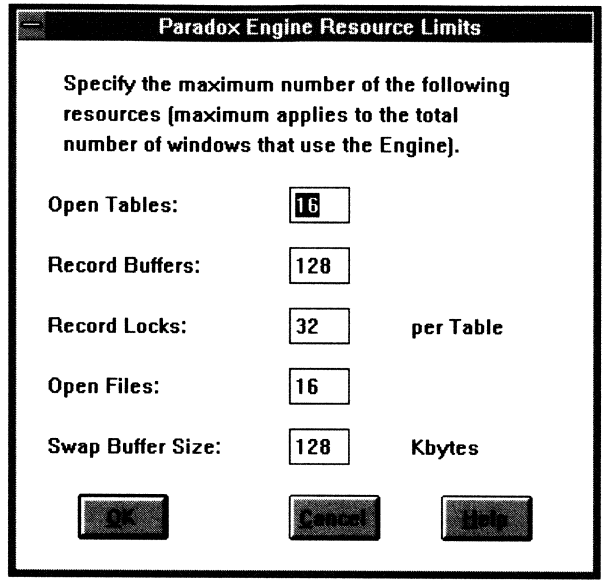
Warning! *Checking this box adversely affects performance, if you're working with large tables.*

- **Share Local Tables with Others**: Check this box if you're running either Paradox 3.5 or Engine applications concurrently from the DOS box in Windows, if you're running Engine applications in multiple windows, or if Net Type is set to Not on a network or Other.

Resource limits

The Resource Limits dialog box sets the resources for an Engine Windows application.

Figure B.3
The Resource Limits dialog box



- **Open Tables** sets how many tables can be open at once. The range is 1 to 64, the default is 16. This setting applies to the total number of open tables in all applications using the Paradox Engine.
- **Record Buffers** sets how many record transfer buffers are available at one time. The range is 1 to 512, the default is 128.

  A *record transfer buffer* is an area in memory that holds a copy of the open table's record structure to facilitate reading and writing field values.

  When your application opens a table, you have the option to save every change to disk (maximum data integrity) or to accumulate changes in a buffer and write them to disk in bigger blocks (maximum performance).

  Increasing the number of buffers speeds up reading and writing operations, but buffered data can, of course, be lost if there's a power failure.

- **Record Locks** sets the maximum number of record locks per table. The range is 1 to 128, the default is 32.

  Locks to records are created by ObjectVision applications only at the time values are written to a linked Paradox database table.

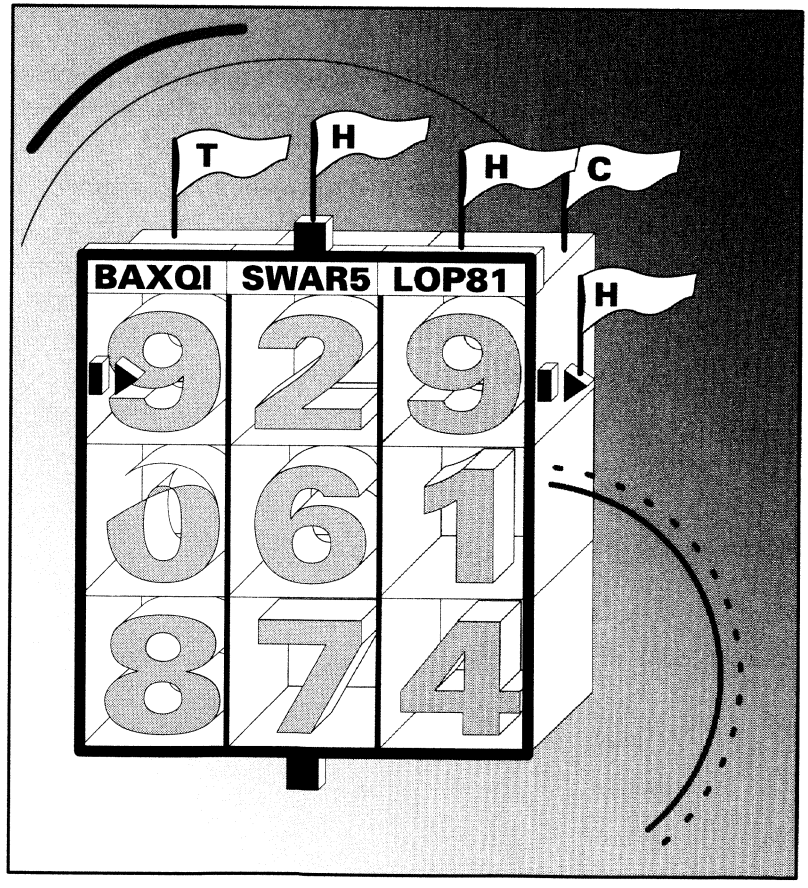
- **Open Files** sets the maximum number of file handles for all Paradox Engine applications in all windows. The range is 2 to 255, the default is 16. In your CONFIG.SYS file, FILES=*n* specifies the file handles for your computer, where *n* is the number of file handles.

- **Swap Buffer Size** determines the size (in kilobytes) of the swap buffer, which is the memory available for swapping data in and out of memory. The range is 8K to 256K, with a default of 128K. The value entered is rounded to a multiple of 4K (each table requires 4K).

  To increase performance and reduce disk access, you can allocate a larger swap buffer, so that tables aren't swapped out to disk as frequently.

  A larger swap buffer, however, can also use up memory that Windows might otherwise allocate for other purposes. Also, if there are many open tables requiring current blocks to be held in memory, too large of a swap buffer might cause the ObjectVision application to run out of memory.

G L O S S A R Y



This is an alphabetical list of key ObjectVision terms. All of these terms are described in greater detail in the sections of the manual in which they appear.

A

active form The form currently being completed or edited; it appears in front of all other forms in the application.

active window The window you are currently working in; the one that always appears in front of any others and has a title bar highlighted with a different color or intensity.

argument Specific information required by an @function. Most @functions require at least one argument.

B

block selection In the Form Tool, lets you select multiple objects in order to perform editing operations, assign or revise properties, or reposition the selected fields as a group of objects.

branch A series of related calculation-logic segments or steps that lead to a tree's conclusion.

branch node A segment of calculation logic (simple or complex) that selects a node below it by evaluating its associated field value. The name of the tree's evaluated field appears inside the branch node.

button An ObjectVision object that can be used to initiate an action. Buttons can be automatically created by ObjectVision when a link is first created.

C

calculated field A field that uses underlying decision logic to produce its value. A calculated field is indicated by a solid outline with an inner dotted line when it is selected.

calculation logic An expression, operation, or constant in a field's tree that computes its value.

cell An individual container for a value in a column of a table object.

choose To execute or carry out a command, or to activate a command button.

circular logic When a conclusion expression refers to its associated field, it re-prompts the user for the field value. Circular logic is used to prompt the user when all other conclusions in the tree are inappropriate.

Clipboard A temporary holding place for objects or forms. After you cut or copy something in ObjectVision, it is placed here. You can open another ObjectVision application and paste the Clipboard contents into it.

column An object similar to a field, except it contains multiple values. Only one value in a column is current at a time. A column is not an independent object—it is an aspect of an ObjectVision table object.

complex branch node An individual segment of calculation logic that requires evaluation of one or more trees from other fields. A complex branch node is indicated with a flow chart symbol to remind you of other, underlying logic.

concatenation The joining of two or more text strings into a single text string.

conclusion The segment or step of calculation logic that is evaluated to determine the resulting value of a field's tree. The conclusion expression is located in the end node for a logical path in a field's tree.

conclusion node The ending segment of calculation logic that provides a value to the field. A conclusion node is indicated by an inverted triangle placed to the left of the conclusion expression.

condition The segment or step of calculation logic that is evaluated to determine what node is selected next.

currency The formatting of monetary figures which includes the symbol corresponding to the country of issue. In Windows, the currency formatting is set in the Control Panel using the International application.

current row Only one row in a table object can be the active, or current, row. When a row is active, the values in the row can be edited. The current row is indicated by the row pointer, a right-pointing triangle next to the left border of the table object.

D

DateTimeNumber An argument that is a number in the range –36,522 (January 1, 1800) to 73,050 (December 31, 2099). The decimal portion of *DateTimeNumber* represents the time from 12:00 noon to 11:59:59 p.m. and is computed as a fraction of a 24-hour day.

DDE (Dynamic Data Exchange) The Windows protocol for dynamically transferring data between Windows applications. DDE links require both the Windows application *and* its document file.

decision logic A process defined as a series of small steps, or nodes. Decision logic is graphically represented for each field's tree.

decision path The segments of calculation logic used to determine a field value. This path is indicated with a bold line when a field's tree is displayed using Field | Show Tree.

default value A calculated field value that can be overridden by the user. In the *Order* application, the Date field has a default value defined by the @TODAY function.

default properties Object characteristics that define the display format and appearance of the next object created. These characteristics remain in effect until changed.

Dynamic Data Exchange See DDE.

E

Edit status The status of the active form in the Form Tool. `[Edit]` appears in the form's title bar to indicate that it receives the actions you perform.

empty node Appears when a field has no value tree or event tree, or a branch node has no nodes under it. If an empty node is evaluated, an error value is returned.

event An event is something the user does during an ObjectVision session. For example, opening an application, selecting a form, entering a value, or clicking a button. Events can happen to an application, a form, or an object.

event tree A graphical representation of the logic and instructions used to recognize an event and initiate an action.

expressions You use expressions to create complex mathematical, logical, string, or @formula operations for evaluating complex value combinations. For example, you might use expressions to multiply a series of numbers or get data from external data files.

external file An ASCII or database file used in a link. The file can be created from within ObjectVision or in another application. Although an ObjectVision form can be saved with a single set of values, most applications benefit from linking to external files that contain multiple values.

F

field A uniquely named object that contains a value either entered by a user, calculated, or delivered by a link. A field that is not put on any form automatically appears on the Scratchpad form.

field sequence The order of user movement between the fields on a form. The field sequence is determined by the physical ordering of the fields from left to right and top to bottom. The relative position of the field's bottom right corner determines whether it is before or after another field.

font A typestyle used for a field label, field value, or a text object.

form A uniquely named object that contains a collection of other objects, analogous to a paper form. A form is the primary interface between your ObjectVision applications and users.

form completion When a user fills in an ObjectVision application's field values. After a form is completed, the status indicator in the title bar reads `[Complete]`.

Form Tool The window where the application designer can create and edit forms and their objects (fields, tables, buttons, text, rounded and filled rectangles, lines, and graphics).

@function Performs calculations and operations within a field's calculation logic. Built-in ObjectVision @functions are compatible with Quattro Pro's and typically require at least one argument, or function-specific information.

G

Goal form The top form in the application's stack of forms. The Goal form appears when a user first opens the ObjectVision application. When another form is selected by the user (interrupting guided completion), it becomes the Goal form.

graphic A Windows Paintbrush (.BMP) or Windows metafile (.WMF) image. A graphic is placed on a form in ObjectVision using the Objects | Graphics command.

guided completion When a user presses *Enter*, only a field requiring user input is selected next. If a user interrupts this method of selection by selecting some other field, guided completion can be restarted with File | Resume.

H

handles In the Form Tool, small black squares in the corners of a selected object or at the ends of a selected line. You can select and reposition handles to change the object's shape or length.

I

index fields A single field or multiple fields in a database table that lets a database program or ObjectVision locate records quickly. Generally, only unique values—such as telephone numbers—are used for indexes. When multiple fields in a database table are index fields, the records are sorted more than once.

L

label An object name, such as a field name, a table name, or a column name. In the Form Tool, a label string can be changed using the Properties | Name/Text command and the label font can be changed using the Label Font command. Also, a constant string value that doesn't need to be enclosed in double quotation marks.

label prefix A single quotation mark (') used as the first character of an expression to force that expression to be evaluated as a label.

links The part of an ObjectVision application used to read from and write to ASCII, Paradox, dBASE-compatible, and Btrieve data files. Additionally, read and write links can be created for Dynamic Data Exchange (DDE) files. Links are created with the Tools | Links dialog box or using @functions.

Links Tool The ObjectVision dialog boxes where you can create, modify, or delete links to external data files.

literal characters In a picture string, any number, letter or punctuation character that isn't a match (# ? & @ !) or reserved (* [] {} ,) character. To use a match or reserved character as a literal character in a picture string, precede it with a semicolon.

load statement The area in the Windows system file WIN.INI where you can instruct Windows to load ObjectVision whenever Windows is first loaded.

Locate field The *most important* link option. The locate field is the ObjectVision field that triggers a link to deliver values from a connected data source.

logical expressions A segment or step of calculation logic that evaluates as either 1 (true) or 0 (false). Logical expressions are typically used with @functions in conditional statements.

M

match characters In a picture string, the unique characters you use to define a kind of pattern for the value users type into a field. The match characters are

- # (digit only)
- ? (letter only)
- & (letter only, convert to uppercase)
- @ (any character)
- ! (any character, convert letters to uppercase)

maximize To enlarge the active window so it occupies the entire Windows desktop by clicking the Maximize button or choosing the Control | Maximize command.

minimize To reduce the active window to an icon on the desktop by clicking the Minimize button or choosing the Control | Minimize command. When an application is minimized, it is still loaded in memory.

multiple selection In the Form Tool, highlighting several objects so subsequent actions are carried out on them all at once.

N

nesting level In a value or event tree, the number of positions away from the root node. Branches that are the same distance from the root node are at the same nesting level.

In an @function or expression, parentheses enclose operations to be performed independently, and the contents of the innermost set of parentheses are always evaluated first. For example, in the expression @INT(@MOD(@NOW,7)), the @NOW function is nested inside the @MOD function and is evaluated first to provide the single argument required by @MOD.

node An individual segment or step of calculation logic that is used to evaluate a field value. A node can be an empty node, a simple or complex branch node, a root node, or a conclusion node.

O

object Any element you can place on a form, such as a table object, a field, text, filled or rounded rectangles, a line or a graphic. Objects can be assigned different properties, or attributes.

operators Used to express a relationship (logical, mathematical, or string) between two or more values. The result of an expression depends on the order in which the operations are performed.

override Calculated fields that are unprotected let a user enter a new value. After a users overrides a calculated value, the field displays a dot pattern. The calculated value can be restored using Field | Calculate.

P

paste The Form Tool action of transferring data from the Clipboard to the active window. Or, transferring list items to an expression using the Paste Function or Paste Field buttons in a value tree or an event tree.

picture A pattern you specify to control values that users type into a field during data entry.

picture string The pattern of literal, match, and special characters you type into the Field Type | Picture | Picture String dialog box to define a template or pattern for a field value entered by a user. For example, you could type a telephone number or Social Security Number pattern users would have to follow.

points A typographers measure of font size, roughly equal to $\frac{1}{72}$ of an inch. The default Label Font is 8-point Helvetica, and the default value font is 12-point Courier.

precedence The order in which operations are evaluated in an expression. Certain operators are always evaluated after others, unless nested.

properties These are the attributes of objects that are defined in the Form Tool. For example, properties can include object names, display attributes, data formats, font characteristics, borders, protection, help text, value trees, and event trees.

property inspector In the Form Tool, you can inspect the properties for an object, a form, or a stack, by clicking it with the right mouse button. The inspector lists all properties for the selected object, and lets you modify the settings.

protection A field property you assign with Properties | Protection to keep users from changing the field value or viewing a tree.

R

reserved characters In a picture string, the unique characters you must precede with a semi-colon when you want them to appear as literals in the field value. The reserved characters are * [ ] { } ,

restricted range This link option specifies that only those records that *exactly* match an entered value are delivered to the form.

root node The first (leftmost) segment of calculation logic in a value or event tree. The root node of a value tree can be either a branch node or a conclusion node.

row pointer A right-pointing triangle that indicates the current or active row in a table object. The row pointer displays to the left of the first column in a table object

run statement The area in the Windows system file WIN.INI where you can instruct Windows to load and then run ObjectVision whenever Windows is first loaded.

S

Scratchpad form The form ObjectVision automatically creates to display any field not on a form, when the user is required to enter a value in that field.

select To position the pointer on an item and highlight that item. The highlighted item will receive the next action that ObjectVision performs.

selected field A selected field can be indicated with a solid line, a solid line with an inner dotted line, or a solid line with a dot pattern over the field. Everything a user types appears in this area on the form.

In the Form Tool, a selected field object is surrounded with a dashed line and has small black squares on its corners (for a field) or on its ends (for a line).

simple branch node An individual segment of calculation logic that is not dependent on trees from other fields.

stack The set of forms in an ObjectVision application, as displayed in the Stack Tool. The Goal form is the top form in the stack.

Stack Tool The window that displays the order of forms in your ObjectVision application and lets you add, copy, paste, delete, or rearrange forms.

status The application title bar displays the form name and the state, or condition of the form: `[Goal]`, `[Complete]`, `[Prompt]`, or `[Edit]`.

syntax The acceptable format for defining expressions or @functions in ObjectVision. For example, all expressions must begin with one of the following characters:

0 1 2 3 4 5 6 7 8 9 . + - (@

T

table object An object that contains multiple values, organized in *n* columns by *n* rows, where *n* is one or more.

Title bar The highlighted horizontal bar at the top of a window. The title bar contains the name of the active application, form, or tool. Form title bars also contain the status of the form: `[Goal]`, `[Prompt]`, `[Complete]`, or `[Edit]`.

V

values The data contained in fields. A user can type values, select them from a list, or check them; an ObjectVision application can calculate values or read them from an external data file.

value trees A graphical representation of the logic and instructions used to evaluate a value and calculate a value.

I N D E X

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

V

W

OBJECTVISION™

GETTING STARTED

CORPORATE HEADQUARTERS: 1800 GREEN HILLS ROAD, P.O. BOX 660001, SCOTTS VALLEY, CA 95067-0001,
(408) 438-5300. OFFICES IN: AUSTRALIA, DENMARK, FRANCE, GERMANY, ITALY, JAPAN, NEW ZEALAND,
SINGAPORE, SWEDEN AND THE UNITED KINGDOM. • PART # 25MN-OBV01-20 • BOR 2841