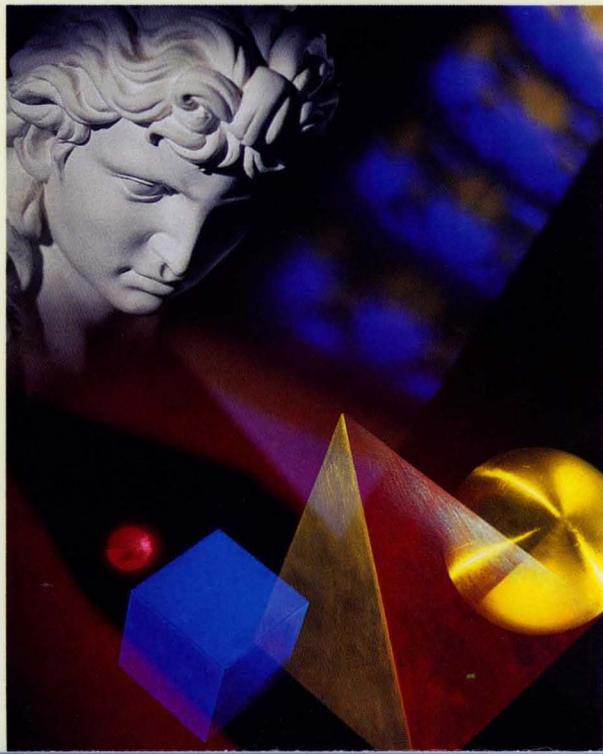


OBJECTVISION™

FOR WINDOWS

TUTORIAL



B O R L A N D

ObjectVision™ Tutorial

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

Copyright © 1991 by Borland International, Inc. All Rights Reserved. Borland and ObjectVision are trademarks of Borland International. Microsoft and MS are trademarks of Microsoft Corporation. Windows, as used in the manual, refers to Microsoft's implementation of a windows system.

C O N T E N T S

Introduction	1	Restoring guided completion	20
What's in this manual	1	Minimal recalculation	20
Typography and naming conventions	2	Active assistance	21
Late-breaking news	3	Chapter 2 Creating a form	23
How to contact Borland	3	Creating your form	24
Chapter 1 Getting started	5	Resizing your form	25
Before you begin	5	Naming and saving your application	26
Creating your application	5	Chapter 3 Creating form objects	27
Sequence	6	Form objects	27
Conventions and terms	7	Fields	28
The ObjectVision tools	7	Ordering your fields	28
Incremental development	8	Text	28
ObjectVision concepts	8	Rectangles	28
Users	9	Lines	29
Forms	9	Graphics	29
Form Tool	10	Inserting fields	29
Distribution	10	Dialog boxes	30
Using forms	10	Sizing the field	32
Fields	11	Testing the size	33
Values	11	Moving fields	33
Scratchpad forms	11	Copying fields	34
Decision trees	12	Renaming copied fields	35
Tree Tool	14	Adding the remaining fields	36
Fields	14	Adding text objects	37
Highlights for field types	14	Changing the font	38
Overriding values	14	Adding the address	39
Expressions	15	Removing the border	40
Condition and Conclusions	15	Adding filled rectangles	41
External links	16	Inserting a graphic	43
ASCII files	16	Chapter 4 Defining field properties	47
Paradox	17	Field properties	47
dBASE	17	Changing display formats	48
Btrieve	18	Selecting the date/time format	49
DDE	19	Selecting other display formats	50
Guided completion	19		

Changing alignment	50	Adding the tree	89
Adding selection lists and check boxes ..	51	Chapter 8 Creating conditions and conclusions	91
Creating value option lists	51	Expressions	91
Adding help to a field	54	Syntax	92
Protecting calculated fields	55	Operators	93
Creating a picture field	57	Expression dialog boxes	93
Chapter 5 Creating a decision tree	59	Adding a decision tree for the Amount field	94
Tree Tool	59	Logic	94
Elements of a decision tree	60	Pasting field names	95
Decision tree	60	Adding the tree	95
Decision logic	60	Testing the tree	97
Node	60	Adding more decision trees	97
Branch node	60	Total Price and Extended Price	97
Condition	61	Shipping Cost tree	99
Conclusion node	61	Using @functions in expressions	101
Root node	61	Syntax	102
Empty node	61	@NOW, @INT, and @ROUND	102
Nesting level	61	Decision tree for Order Date	103
Summary	61	Decision tree for Less Discount	104
Decision tree for Shipping Method	63	Decision tree for Sales Tax	106
Creating the decision tree	64	Input validation	107
Conclusion order	68	Evaluating the Quantity tree	109
Testing your decision tree	69	Chapter 9 Linking to databases	111
Chapter 6 Editing a decision tree	71	External links	111
The Discount decision tree	71	Creating a Paradox table	112
Adding the first nodes	73	Using links with button fields and decision trees	115
Adding more conclusion nodes	75	Chapter 10 Entering data with your application	119
Scrolling a decision tree	76	Clearing data	119
Adding a form for a new field	77	Form navigation	120
Deciding design criteria	77	Completing the form	120
Creating a new form	77	Taking an order	121
Displaying the order of forms	79	Correcting text	121
Copying and editing parts of a tree	80	Finishing the order	123
Testing the tree	84	Printing the form	126
Viewing and printing the tree	84	Entering a new order	126
Sizing a tree	84	Making "what if" changes	127
Printing a tree	85	Finding data	128
Creating lists of values automatically	85	Summary	129
Chapter 7 Handling default and error values	87		
Providing default values	87		
Using circular logic in a decision tree	88		

Appendix A Keyboard and mouse operations

Choosing menu commands	131
Running applications	132
ObjectVision Control menu commands	133
Getting help online	134
Using dialog boxes	136
Completing forms	137
Fields	137
Viewing decision trees	138
Form Tool	139
Stack Tool	141

Glossary

active form	143
active window	143
argument	143
block selection	143
branch	143
branch node	143
button fields	143
calculated field	144
calculation logic	144
choose	144
circular logic	144
complex branch node	144
concatenation	144
conclusion	144
conclusion node	144
condition	144
DateTimeNumber	144
DDE (Dynamic Data Exchange)	144
decision logic	144
decision path	145
decision tree	145
default	145
Dynamic Data Exchange (DDE)	145
Edit form	145
empty node	145
expressions	145
external links	145
field	145
field sequence	145
font	145

form	146
Form Clipboard	146
form completion	146
Form Tool	146
@function	146
Goal form	146
graphic	146
guided completion	146
handles	146
label	147
label prefix	147
links	147
Links Tool	147
literal characters	147
load statement	147
logical expressions	147
match characters	147
maximize	147
minimize	147
multiple selection	148
nesting level	148
node	148
object	148
operators	148
override	148
paste	148
picture	148
picture string	149
points	149
precedence	149
protection	149
reserved characters	149
root node	149
run statement	149
Scratchpad form	149
select	149
selected field	149
simple branch node	149
stack	150
Stack Clipboard	150
stack order	150
Stack Tool	150
status	150
syntax	150

Title bar	150
Tree Clipboard	150
Tree Tool	150

values	150
--------------	-----

Index	151
--------------	------------

T A B L E S

8.1: Expression operators 93

F I G U R E S

1.1: Sales Order form from the <i>Order</i> sample application	6
1.2: <i>Order's</i> stack of forms	11
1.3: Scratchpad form in the <i>Expense</i> sample application	12
1.4: Decision tree for one of <i>Credit's</i> fields .	13
2.1: The Sales Order form	23
2.2: Newly created Sales Order form	24
2.3: Resized Sales Order form	26
3.1: Sales Order form	29
3.2: Objects Field Field Name dialog box	30
3.3: Placement of the Name field	32
3.4: Pasted copies of Name field	35
3.5: Properties Field Name dialog box . . .	36
3.6: Properties Label Font dialog box . . .	38
3.7: Your Sales Order form with text objects	41
3.8: Properties Fill Pattern dialog box . . .	42
3.9: A graphic in the Sales Order form . . .	45
4.1: Date/Time dialog box	49
4.2: Values of dialog box	53
4.3: Help dialog box	55
4.4: Properties Protection dialog box . . .	56
4.5: Field Type Picture dialog box	57
5.1: Decision tree for Approval Required .	62
5.2: The completed decision tree for Shipping Method	63
5.3: Branch node	65
5.4: Condition dialog box	66
5.5: The Conclusion dialog box	67
5.6: Shipping Method's list of value options	69
6.1: Decision Tree for Discount	72
6.2: Your Discount decision tree with one conclusion node	75
6.3: Continued decision tree for Discount .	76
6.4: Completed Distributor Information form	78
6.5: Decision Tree for Discount	81
7.1: Item decision tree	88
7.2: Completed decision tree for Unit Price	88
8.1: Completed Decision tree for Amount .	94
8.2: Paste Field dialog box	96
8.3: Completed decision tree for Total Price	98
8.4: Decision tree for Extended Price	98
8.5: Completed decision tree for Shipping Cost	99
8.6: First nodes in Shipping Cost decision tree	100
8.7: Decision tree for Sales Tax	106
8.8: The error message for illegal values .	108
9.1: Tools Links dialog box	112
9.2: The Link Type dialog box	113
9.3: The Paradox Links dialog box	113
9.4: The Paradox Links dialog box with the linked field names	114
9.5: Decision tree for Save to database . .	116
10.1: Completing the Quantity field	122
10.2: Distributor Information form	123
10.3: Scratchpad form	124
10.4: Completed Sales Order form	125

This book is the first in a set of two manuals:

- This book, *ObjectVision Tutorial*, tells you how to get your application up and running, and includes step-by-step instructions for recreating *Order*, one of the sample applications included with ObjectVision.
- *ObjectVision Reference* explains how you design ObjectVision applications and how your applications are used. It goes into thorough detail about each area of ObjectVision.

What's in this manual

This manual is divided into ten chapters, one appendix, and a glossary:

Part 1: ObjectVision features

- **Chapter 1, "Getting started,"** describes the sample application you will create using the tutorial and explains the conventions the tutorial follows. This chapter also explains how users fill in the forms you create, and is designed to give you an overview of ObjectVision concepts and applications.
- **Chapter 2, "Creating a form,"** is the beginning of the tutorial. It shows you how to open the Form Tool, how to create and resize a form, and how to save your application.
- **Chapter 3, "Creating form objects,"** shows you how to insert objects into forms, copy and insert fields, rename fields, change label and text fonts, remove text-object borders, and paste graphics from the Clipboard.
- **Chapter 4, "Defining field properties,"** shows you how to change field formats, how to add possible-value lists to fields, how to add help information to a field, and how to protect a field from user input.

- **Chapter 5, “Creating a decision tree,”** shows you how to start the Tree Tool; how to insert branch nodes, conditions, and conclusion nodes into a decision tree; and how to test a decision tree.
- **Chapter 6, “Editing a decision tree,”** shows you how to add a new field with the Tree Tool, how to scroll a decision tree, how to copy and paste parts of a decision tree, how to write conditions and conclusions, how to resize a decision tree, and how to print a decision tree.
- **Chapter 7, “Handling default and error values,”** shows you how to use decision trees to provide default field values and how to use circular references to prompt users for valid input.
- **Chapter 8, “Creating conditions and conclusions,”** gives you more practice creating decision trees and using the Condition and Conclusion dialog boxes; shows you how to use the Paste Field, Paste Function, and Paste Arguments commands; and shows you how to use the @MESSAGE and @BLANK functions for input validation.
- **Chapter 9, “Linking to databases,”** shows you how to link fields in your application to external data files and how to add decision trees to button fields.
- **Chapter 10, “Entering data with your application,”** shows how your application works from a user’s perspective, and explains various elements of form design.
- **Appendix A, “Keyboard and mouse operations,”** lists the keys, key combinations, and mouse techniques you can use to perform ObjectVision operations.
- The **Glossary** provides definitions for those words and phrases used in the manual that might be unfamiliar to you. The first time each of these words and phrases is used in the text, it is set in italic type.

Typography and naming conventions

These manuals use special fonts to help you distinguish among material of various types:

Monospaced type

This font represents text as it appears onscreen, as well as anything you must type.

<i>Italics</i>	Italics are used to emphasize certain words and to introduce new terms, which are defined in the Glossary. We also use this font for the names of books and the names of the sample applications included with ObjectVision.
<i>Keycap</i>	This special font represents a keyboard key that you press; for example, "Press <i>Enter</i> or choose OK."
ALL CAPS	All caps are used to indicate a DOS file name, such as the sample ORDER.DB database file.
SMALL CAPS	@Function names appear in small capital letters.
Menu Command	Rather than use the phrase "Choose the Save command from the File menu," the manuals use the convention File Save to indicate a command path.

In addition to special fonts, two icons are used in this manual:



The pencil icon indicates the next steps to perform in the tutorial.



The mouse icon indicates mouse techniques and shortcuts you can use in ObjectVision.

Late-breaking news

If there have been any changes or additions to ObjectVision since this manual was printed, you can read about them in the README.TXT file provided on your ObjectVision disk.

Be sure to correct your manual as needed.

How to contact Borland

The best way to contact Borland is to log on to Borland's Forum on CompuServe: Type GO BORAPP from the CompuServe main menu. To locate the ObjectVision Forum follow the onscreen

instructions. Leave your questions or comments there for the support staff to process. If you need assistance with CompuServe, type the command `help`.

Be sure to provide a daytime phone number in case the support engineer needs to contact you for more information.

If you prefer, you can write a letter with your comments. Include a daytime phone number where you can be reached in case the support engineer responding to your letter has questions and needs to contact you for additional information. Send your letter to

Borland International
Technical Support Department
1800 Green Hills Road
P.O. Box 660001
Scotts Valley, CA 95067-0001, USA

(408) 438-5300

You can also telephone our Technical Support department between 6:00 a.m. to 4:45 p.m. (Pacific Standard Time).

Note that you must be a registered user of Borland software to receive technical support from Borland.

Whichever way you choose, you must provide the following information:

- Product name and serial number on your original distribution disk. Please have your serial number ready, or we won't be able to process your call.
- Product version number. The version number for ObjectVision is displayed when you first load the program, before you press any keys.
- Computer brand and model, and the brands and model numbers of any additional hardware.
- Windows and DOS version numbers. The version number for Windows displays in the Windows title screen after you type `win` and press *Enter*. The version number for DOS can be determined by typing `VER` at the DOS prompt.
- Contents of your AUTOEXEC.BAT and CONFIG.SYS files.
- The specific steps that reproduce the problem you're having.

Getting started

ObjectVision is a tool for creating applications that gather and manipulate information. Applications you create with ObjectVision present a familiar forms interface to users.

This chapter briefly discusses *Order*, the sample application you will recreate in the tutorial, the tutorial conventions used, and ObjectVision features and tools you'll use in this tutorial. Additionally, this chapter discusses incremental development, a method you can use to create your own ObjectVision applications.

Before you begin

Before you begin, ObjectVision must be installed. Chapter 2 of the *ObjectVision Reference* explains how to install ObjectVision and get up and running. If you aren't familiar with Windows applications, you might want to read the "Using Windows" section in Chapter 2 of that manual.

Creating your application

In this tutorial, you recreate the order-entry sample application, *Order*. You link the application to an order database so that your application can both read in values from the database and update the database.

The database files for your application—the ORDER.DB database and the database’s index file, ORDER.PX—are supplied with your ObjectVision software.

Order is a simple application designed to show you how forms, fields, and decision trees work together. It is easy to create, and uses many important ObjectVision features, including links to external data and calculated fields. *Order’s* Sales Order form is shown in the following figure:

Figure 1.1
Sales Order form from the
Order sample application

Sales Order (Goal)			
		4030 Braker Lane West Suite 2001 Austin, TX 78759-5332	
Name		Order Date 10/12/90	
Company		Customer Type <input type="checkbox"/> distributor	
Address		<input type="checkbox"/> dealer	
City, State and ZIP		<input type="checkbox"/> educator	
		<input type="checkbox"/> other	
Item Widget	Quantity	Unit Price \$295.00	Amount
Shipping Method	Discount		Less Discount
<div style="background-color: #cccccc; padding: 10px; text-align: center;"> Next Previous Save to database </div>			Total Price
			Sales Tax
			Shipping Cost
			Extended Price

Sequence

The tutorial is divided into several chapters, and each chapter may contain several procedures. You can go through the whole tutorial at once, or start and stop as you wish. However, you must do the tutorial in order, because each part builds on what came before. You should save your work each time you quit.

Conventions and terms

The tutorial mixes explanatory information and procedures. A procedure, with steps that you are to carry out at your computer, has a pencil icon in the sidebar.

 *Your next steps:*

Look for this icon if you want to skip the explanations and concentrate on actual procedures (or if you prefer, read the explanations first and skip the procedures).

The key called *Enter* in the tutorial may be labeled *Return* on your keyboard or identified with a bent arrow.

Two terms refer to the process of performing an ObjectVision operation:

- **Select** means to position the pointer on an item and highlight it. The highlighted item will receive the next action that ObjectVision performs. For example, when you highlight a field on a form by clicking it or pressing *Enter*, the next action you perform applies to that field. When you select text, it displays in reverse video and is replaced by the next characters you type.
- **Choose** means to execute or carry out a command. If you have already selected an item, the action of the command is applied to the selected item. For example, you choose a command on a menu by clicking it or pressing *Enter*. Similarly, you choose a dialog box command button by clicking it or selecting it and pressing *Enter*. In some cases, double-clicking an item both selects and chooses it.

Keyboard and mouse operations are listed in Appendix A.

The ObjectVision tools

ObjectVision includes four tools for developing applications:

- The Form Tool, with which you create and modify forms and their elements
- The Tree Tool, with which you add and modify decision trees
- The Links Tool, with which you create links to external sources
- The Stack Tool, with which you display or change the order of forms in an application

Incremental development

To help you learn how to use ObjectVision, this tutorial has you recreate a sample application, *Order*. As you work with the tutorial, keep in mind that ObjectVision differs from other application tools in one important aspect: Instead of knowing in advance the exact details of the forms you want to create, you will be starting with a general concept of how you want your application to work, and adding the details as you go.

When you work on your own application, you shouldn't try to get it perfect the first time. Instead, create a rough prototype. You can keep changing your prototype, each time getting a clearer idea of how it should work, until it does what you want. At that point, you can go back and add the finishing touches, like resizing fields, adding help information, inserting patterns, and so on.

ObjectVision concepts

Your applications can include standardized procedures and decision processes users need to follow when filling in the data required to complete the application. You can design applications to gather and present information for a great variety of user needs.

An application can contain either a single form or multiple forms. You can specify which portions of your applications are completed automatically and in what sequence users complete your applications. You can also link your application to data files produced by other applications.

Your applications can automate any task that requires users to follow complex procedures and accurately capture data, thus making the task almost effortless for the user. Some typical areas for ObjectVision applications are

- Loan/credit approval
- Insurance risk/rate determination
- Insurance claims processing
- Personnel benefit administration
- Customer support
- Order configuration/quote generation

- Telemarketing support
- Equipment diagnosis and repair
- Factory quality/yield control
- Factory scheduling decisions
- Process startup/shutdown procedures
- Purchasing decision support
- Tax reporting/planning decisions

Users

ObjectVision is easy to use for both designing applications and using them.

- **ObjectVision** can be used to create an application for the specific requirements of an individual or a group. Programming skills are *not* necessary to create an ObjectVision application. Application designers can be experienced personal computer users or systems analysts familiar with the application task. Application users can easily modify applications for specific requirements.
- **ObjectVision Runtime** lets end users complete ObjectVision applications without letting them modify an application.

People who are familiar with an application's requirements can be involved in its design and maintenance. Application designers can easily incorporate user suggestions resulting in custom applications that meet the needs of the particular organization.

Forms

ObjectVision presents a forms interface to users because it is a familiar way to collect and display information. You can specify how your application's information displays in the forms' fields.

Your blank form contains fields the user both fills in and uses as a guide for making decisions about a specific topic area. After users fill in all the fields, the completed form documents the gathered data and makes it easy to share the information with others.

Because most organizations already use paper forms, your applications can use existing forms as models. Your ObjectVision forms

can be exact replicas or enhancements of existing paper forms. When you create applications for which no paper forms already exist, you can quickly design forms to contain the relevant information.

Form Tool

As you create an ObjectVision application, you use the ObjectVision Form Tool to define your forms. With the Form Tool, a high-level graphical interface, you specify what fields you want on a form and how those fields appear. You can adjust field attributes such as numeric formats, fonts, patterns, and borders.

Distribution

After you create and save an application, you can make copies of your disk file and distribute them to users. Users can then run your application using ObjectVision or ObjectVision Runtime, because no conversion of your file is necessary.

For more information about licensing ObjectVision Runtime, see the ObjectVision Runtime brochure included with your ObjectVision materials.

Using forms

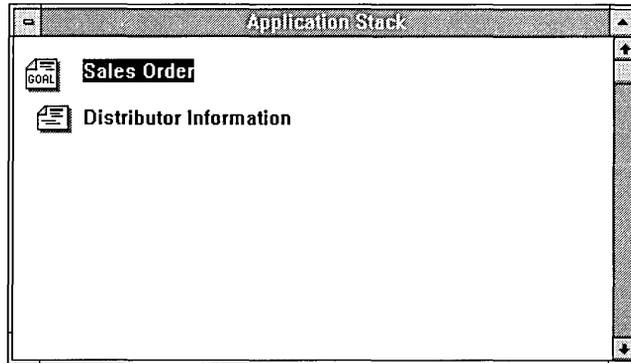
ObjectVision applications automatically determine *guided completion*, the field sequence users follow while filling in forms. The user can interrupt guided completion at any time to open, close, or rearrange forms on the display.

A stack is equivalent to a stack of paper forms.

The set of forms in an application is ordered in a *stack*. Figure 1.2 shows the stack of forms for the *Order* sample application. Each form is represented as an icon. Its unique title appears to the right of the icon.

The form at the top of the stack is the Goal form and it is the initial form that appears when your application is first opened. After a form is selected by either the user or guided completion, that form moves in front of the initial Goal form. The user can enter information only into the active form, even when other forms are visible under it.

Figure 1.2
Orders stack of forms



Fields

Fields are the blanks on your forms that contain user-entered or automatically calculated values. You give a unique name to each ObjectVision field in your application.

You can put the same field on two or more of your forms. You might want a field on multiple forms to save users the bother of entering the same information repeatedly. For example, a client's name or identification number that must appear on all tax forms needs to be entered only once. When a field is used on multiple forms, it has the same value on all forms, regardless of which copy of the field the user actually enters the information into.

Values

4,096 is the maximum number of characters in a field value.

Field *values* are undefined until the user enters a value, ObjectVision calculates a value, or a link to external data provides a value. Any field can contain any type of value: numeric, text, or error.

A field can have only one value at a time. When the user enters a new value, the new value replaces the old one.

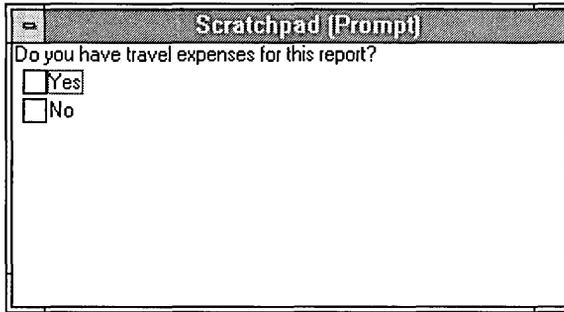
Scratchpad forms

Scratchpad forms are temporary forms created automatically.

It is not necessary for you to place all your fields on a form. ObjectVision displays a small *Scratchpad form* when prompting users for information necessary to complete a field that is not present on any form.

Suppose, for example, that you are modeling your application on existing forms that do not contain all of the background information needed to calculate a value. Rather than add a field to one of your forms or create a new form, you can have ObjectVision use a Scratchpad form to prompt the user for the field's value.

Figure 1.3
Scratchpad form in the
Expense sample application



Scratchpad (Prompt)

Do you have travel expenses for this report?

Yes

No

Decision trees

You use the Tree Tool to create *decision trees* that define procedures for calculating field values. Some procedures may be simple math operations such as adding a column of numbers. Other procedures may be more complex and involve examining a variety of conditions before calculating a value.

For example, the amount of temporary insurance granted in an insurance application may be determined by a procedure that considers the applicant's age and health and type and amount of insurance requested.

You can use the Quattro Pro-compatible @functions (described in Chapter 8 of *ObjectVision Reference*) in decision tree expressions. You can also create decision trees to represent conditional logic.

Figure 1.4
Decision tree for one of
Credit's fields

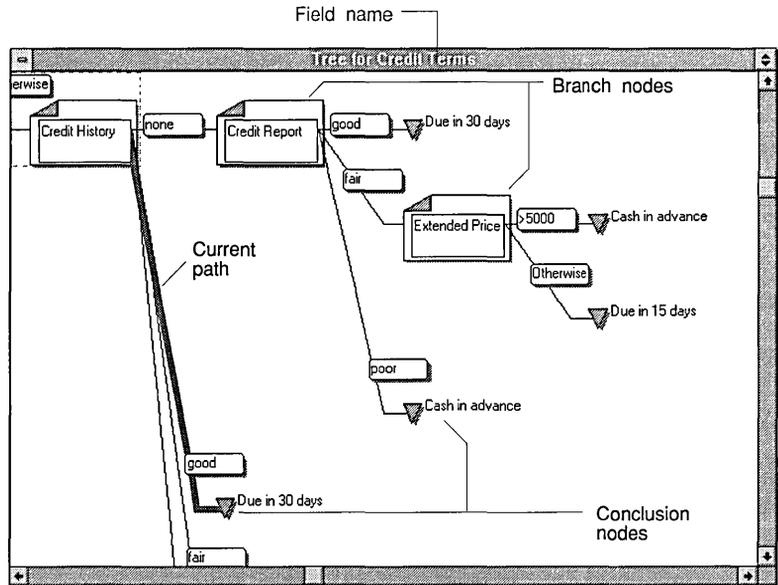


Figure 1.4 shows part of the Credit Terms decision tree from the *Credit* sample application. This field calculates credit terms for a customer based on the customer's credit history and order size.

Decision trees can define a complex decision-making process by breaking it into a series of smaller steps. At each step, a single element of the tree, called a *node*, is evaluated.

Your decision trees return the value defined by the conclusion node.

When you create a decision tree, you define all anticipated field values as nodes in your tree. You can place one or more nodes in a tree *branch*, which details the logic for each anticipated field value the decision tree calculation is based on.

ObjectVision follows the branch appropriate to the evaluated condition of a field value. When ObjectVision reaches the last branch node, the decision tree defines a conclusion that results in a calculated field value.

Tree nodes contain field names for other fields in the application. When a node is on the current *decision path* in a decision tree, that node is used to calculate the selected field's value.

Tree Tool

You use the Tree Tool to create and edit decision trees. The Tree Tool's cut, copy, and paste operations let you manipulate your decision trees graphically. Whenever you create a new decision tree for a field or modify an existing tree, you also change the decision-making logic associated with that field.

Decision trees calculate fields automatically.

As the user fills in your form's required field values, ObjectVision calculates any decision tree values dependent on the user-entered values. You can create a decision tree for any field in your application.

Fields When ObjectVision determines which of your tree branches to follow, the actual field values are evaluated. This process is very similar to using formulas in spreadsheets that refer to other spreadsheet cells.

Your users can perform "what if" analyses simply by changing the value of a field. ObjectVision recalculates all of your application's decision trees that contain references to the changed field.

Highlights for field types ObjectVision distinguishes among fields of different types by displaying different highlights around the field border:

- A dashed line within a heavy line appears when a user selects one of your form's calculated fields.
- A heavy line appears when either ObjectVision or a user selects a field requiring user input.
- A field you have protected from user input displays the heavy line highlight without a text pointer.

Overriding values

Users can *override* unprotected, calculated fields in your application's forms by entering new values. When a user overrides the calculated value, the user-entered value replaces the calculated value without removing the decision tree logic you assigned to that field.

After a user overrides a field, a gray background appears in that field, along with the value, to help remind the user that the calculated field value has been overridden. If users want to

restore the calculated value, they can use the Field | Calculate command to recalculate the value based on the decision tree logic.

Expressions

Within decision trees, you can use *expressions* to compute values based on a combination of other values. For example, you might need to multiply a series of numbers. You can use ObjectVision expressions that are identical in concept, use, and syntax to Quattro Pro's spreadsheet formulas.

Expressions used in decision tree conclusion nodes can contain constant values, such as "Yes" or "No." They can also contain expressions to calculate values based on other field values.

Your expressions can specify the condition that determines which branch in the tree is used. An expression can be either a constant value or an expression that computes a value. A constant value, such as "Excellent," "Good," or "Fair" can be used as a branch condition. An expression, such as " \geq Total," can also be used as a branch condition.

A decision tree without branches has only one conclusion node and always returns the same value.

The syntax and use of ObjectVision expressions is compatible with Quattro Pro formulas. An expression can contain combinations of constant values, references to field values, mathematical and comparison operators (such as + and \geq), and @functions (such as @INT). Like field values, expressions can contain a maximum of 4,096 characters.

Condition and Conclusions

When you use the Tree Tool to create a decision tree, you can use the Condition and Conclusion dialog boxes to create and modify expressions. You can type or paste @function names and field names into an expression. When you finish creating or editing an expression, the dialog box checks the validity of the expression.

External links

Your applications can use *external links* to data as an alternative to requiring user input or computing a value in a decision tree. You can use ObjectVision's Links Tool to create, modify, or delete external links that read from and write to data files created by other applications.

You can use ObjectVision's external links in your applications to link with these data file types:

- ASCII
- Paradox
- dBASE-compatible
- Btrieve
- Dynamic Data Exchange (DDE) to other Microsoft Windows applications

You can create links of all five types by choosing the Tools | Link dialog box. When the Tools | Link dialog box appears, you specify the type of link you want, where the external data is, and which ObjectVision fields to connect.

ObjectVision also provides a set of @functions for links you can use in your decision trees. For more information, see Chapter 9.

When your applications link to external data, the operations your links perform observe the conventions of the application that created the data file. For example, when a user searches for an existing field value using wildcard characters, and a match for that pattern is unavailable, Paradox returns the closest matching record, and dBASE returns nothing.

ASCII files

Your ObjectVision applications can exchange data with ASCII files. Many popular spreadsheet, database, and word-processing applications read and write ASCII files. You can also use ASCII files for transferring data via a modem to a remote location.

Your ObjectVision applications can read data from ASCII files and connect that data to fields in your forms. You can also write any of your form's field values to a linked ASCII file.

Paradox

Your ObjectVision applications make it easy for users to get existing data from Paradox files. The links you create can also either append your application's linked field values to the database or update the database's records.

After you choose the Tools | Links command, you can create, modify, or delete a Paradox link by choosing the corresponding button in the Tools | Links dialog box.

After you choose Paradox as the link type, the Link Type | Paradox dialog box appears. This is where you specify the link name and the path and file name of the Paradox file. ObjectVision reads the header of the Paradox file to locate the primary index file and the Paradox data field names.

The Paradox field names display in the Field Name list, and you can select which fields you want to connect with your ObjectVision fields. The number of links to your application is limited only by the amount of memory your system has.

You can also use the @PXOPEN, @CLOSE, and @STORE functions in your decision tree expressions to create links to Paradox. The @TOP, @BOTTOM, @PREVIOUS, @NEXT, @CLEAR, and @DELETE functions enable you to locate a particular record or position precisely in the Paradox file.

Your applications can be used in a multiple-user environment, where data integrity is essential. All users can use your ObjectVision application to read linked Paradox records. ObjectVision notifies users trying to save revised values if other users have already updated the same record during the current session.

dBASE

Your ObjectVision applications make it easy for users to get existing data from dBASE-compatible files. The links you create can also either append your application's linked field values to the database or update the database's records.

After you choose the Tools | Links command, you can create, modify, or delete a dBASE link by choosing the corresponding button in the Tools | Links dialog box.

After you choose dBASE as the link type, the Link Type | dBASE dialog box appears. This is where you specify the link name, the path and file name of the dBASE file, and the (optional) index file. ObjectVision reads the dBASE file header to locate the dBASE data field names, and reads the index file header to determine the indexing scheme.

The dBASE field names display in the Field Name list, and you can select the fields you want to connect with your ObjectVision fields. The number of links to your application is limited only by the amount of memory in your system.

In your decision tree expressions, you can use the same @functions you use for Paradox when you create links to dBASE files (with the exception of @DBOPEN, which you must use to open a link to a dBASE-compatible file).

Your applications can be used in a multiple-user environment, where data integrity is essential. All users can use your ObjectVision application to read linked dBASE records. ObjectVision notifies users trying to save revised values if other users have already updated the same record during the current session.

Btrieve

Your ObjectVision applications make it easy for users to get existing data from Btrieve database tables. The links you create can also either append your application's linked field values to the database or update the database's records.

After you choose the Tools | Links command, you can create, modify, or delete a Btrieve link by choosing the corresponding button in the Tools | Links dialog box.

After you choose Btrieve as the link type, the Link Type | Btrieve dialog box appears. This is where you specify the link name and the path and file name of the Btrieve file. You also need to include the path name of the directory that contains the XQL data dictionary files if the Btrieve table is an existing table.

Once the link is established, the Btrieve field names display in the Field Name list, and you can select the fields you want to connect to your ObjectVision fields. The number of links your application opens is limited only by the amount of memory in your system.

In your decision tree expressions, you can use the same @functions you use for Paradox when you create links to Btrieve

files (with the exception of @BTRVOPEN, which you must use to open a Btrieve link).

Your applications can be used in a multiple-user environment, where data integrity is essential. All users can use your ObjectVision application to read linked Btrieve records. ObjectVision notifies users trying to save revised values if other users have already updated the same record during the current session.

DDE

Windows provides the Dynamic Data Exchange (DDE) link mechanism to allow Windows applications to transfer information directly, without using ASCII files.

You can create DDE links from your application to other Windows applications such as Excel and Word for Windows.

DDE links have the advantage of transferring data automatically whenever the information source file receives a new value. With DDE links, you can seamlessly integrate your application's data with files from a variety of Windows products. You can also link your application to multiple ObjectVision applications when you need to create an application larger than the ObjectVision system limits allow.

Guided completion

When users select a field, they interrupt guided completion as defined by your Goal form.

By default, ObjectVision determines guided completion based on your application's Goal form. This feature selects only those fields requiring a user to supply a value. Users who are unfamiliar with your application need only to supply information in fields selected by guided completion.

When a field gets its value from either a decision tree or an external link, it is omitted from guided completion. Whenever guided completion selects a field, the form containing that field appears so the user can supply a value.

Form status [Complete] displays when the user has completed filling in your application's forms.

A user can let ObjectVision select the next required field and form using guided completion, or they can interrupt guided completion by manually selecting any form or field.

Any incomplete form selected by the user becomes the Goal form. The form's status appears in the application title bar after the

application name. The initial Goal form is the form occupying the top position in the stack.

Restoring guided completion

During form completion, the user can interrupt guided completion by manually selecting another field or another of your application's forms. After a user selects a new, incomplete form, ObjectVision determines guided completion for the newly selected form.

File | Resume quickly restores guided completion.

This user-selected form continues to be the Goal form until the user closes it, selects a new Goal form, or restores guided completion. When the user closes a Goal form, ObjectVision selects the previous Goal form as the new Goal form. To resume guided completion for the initial Goal form, the user can choose the File | Resume command.

Minimal recalculation

Whenever the user enters a new value or changes an existing value in your application, only those values that are based on that value recalculate. Recalculation occurs whether the user is following guided completion or manually selecting fields.

Minimal recalculation enables users to perform "what if" analyses within your application. For example, the user can use recalculation to determine the effect of changing a field's value.

When field values in your application need recalculation, ObjectVision first removes the current value of any field that is based on the changed value. Next, the new value displays in all the dependent fields.

After the user changes a value, ObjectVision may be unable to recalculate a value for all dependent fields. When a changed value requires ObjectVision to follow other branches in your decision tree, new values may be required.

The Field | Calculate command determines guided completion for calculated values.

The user can choose the Field | Calculate command to recalculate any field that lacks its calculated value. After a user chooses this command, ObjectVision determines guided completion for the required information and selects the first field.

Active assistance

You can provide support for users of your applications by creating online help for fields in your applications. When users are unable to understand what value is required in a field, they press *F1* to display help for that field. If you omitted help for that field, the general online help for ObjectVision appears.

Creating a form

In this chapter, you start to recreate *Order*, the sample application. The following figure shows *Order's* Sales Order form, which you can use for reference. You might also want to print this form to refer to as you recreate it. If you do print it, notice that buttons don't appear on the printed copy.

Figure 2.1
The Sales Order form

Sales Order (Goal)			
		4030 Braker Lane West Suite 2001 Austin, TX 78759-5332	
Name		Order Date 12/10/90	
Company		Customer Type	
Address		<input type="checkbox"/> distributor <input type="checkbox"/> dealer <input type="checkbox"/> educator <input type="checkbox"/> other	
City, State and ZIP			
Item	Quantity	Unit Price	Amount
Widget		\$295.00	
Shipping Method	Discount	Less Discount	
<div style="background-color: #cccccc; height: 100px; width: 100%;"></div>			Total Price
			Sales Tax
			Shipping Cost
			Extended Price
<input type="button" value="Next"/>		<input type="button" value="Previous"/>	
<input type="button" value="Save to database"/>			

In this chapter, you will perform these procedures:

- Open the Form Tool.

- Create and resize the Sales Order form.
- Save *Myorder*, the application you are creating.

Creating your form

When you're ready to create your own form, follow these four steps:

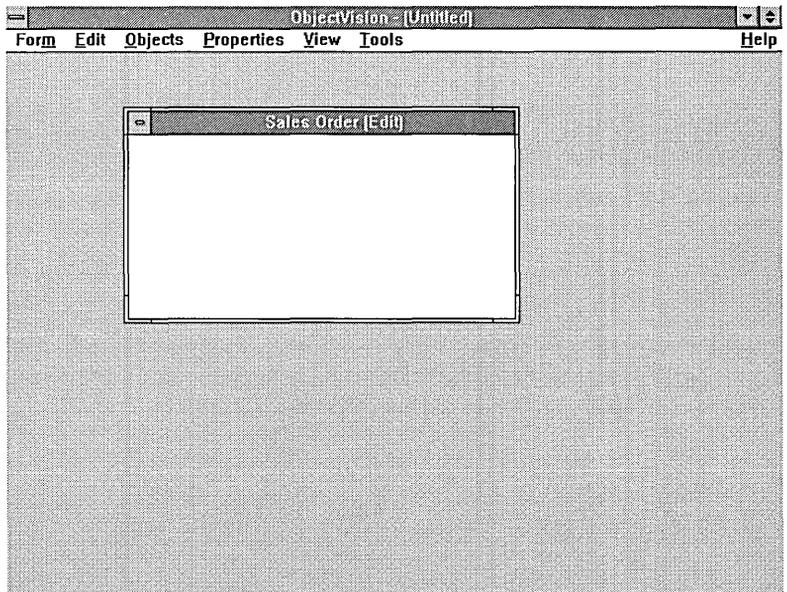
➡ *Your next steps:*



1. Start ObjectVision, but don't select an application.
2. Choose the Maximize button located in the upper right corner of the window. This button (shown at left) enlarges the window so that it fills the screen. You should maximize the window whenever you start ObjectVision in this tutorial.
3. Choose the Form command from the Tools menu. The Form | Form Name dialog box appears.
4. Type Sales Order and then choose OK or press *Enter*.

The newly created Sales Order form appears, as shown in the next figure:

Figure 2.2
Newly created Sales Order
form



Notice the form's *title bar*. It contains the name of the form and the form's status: [Edit]. The Form Tool is still open, and you can continue to edit the new form.

Also notice the size of the form, which is the default size of newly created forms. You can resize the form with the Form Tool. In the next section, you'll enlarge the form.

Resizing your form

When you position an object on a form, the size of the form increases, when necessary, to make room on the form for the new object. You can also change the size of the form manually. Now you will change the size of the Sales Order form to match the *Order* sample application.



Using the mouse:

 Your next steps:

1. 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 until it is close to the application window's right border. Move the left border to the same distance from the application window's left border.
2. Use the mouse to enlarge the form vertically by moving the top and bottom borders until they almost touch the application window's borders. If you position the pointer in a corner of the window, it changes to a four-headed arrow that lets you move two borders at the same time.

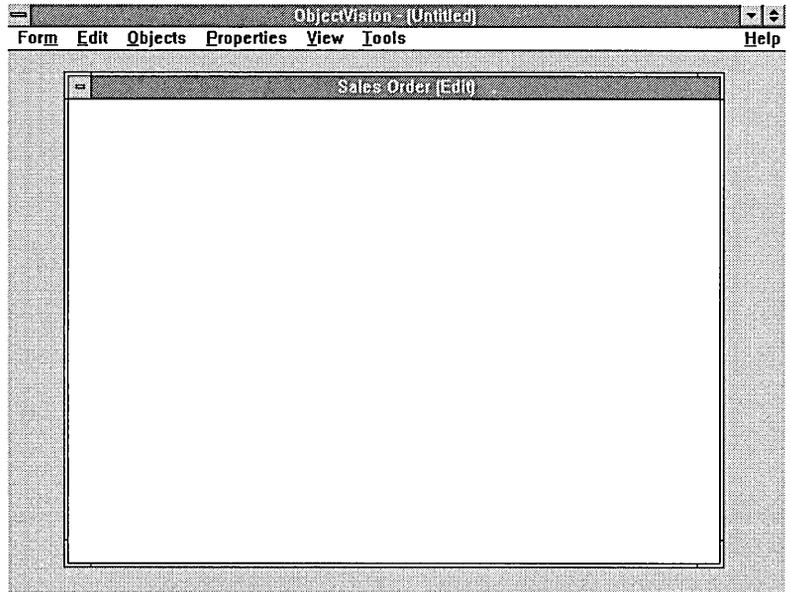
Now the Sales Order form should look like the one shown in Figure 2.3 on page 26.

Or, using the keyboard:

1. Open the Sales Order form Control-menu box by pressing *Alt+-* (hyphen).
2. Select the Size command, then press *Enter*.
3. Press *←*, *↑*, *→*, or *↓* to select a border to change, and then press *←*, *↑*, *→*, or *↓* to move the window border. After you move a border, and before you press *Enter*, you can move an adjacent border by pressing another arrow key. Then press *Enter*. Now move two more borders and then press *Enter*.

Now the Sales Order form should look like this:

Figure 2.3
Resized Sales Order form



Naming and saving your application

In the next steps, you name and save the application you are designing. The original order application included with your ObjectVision software is named *Order*. Be sure to name your application *Myorder* so you don't overwrite the sample application.

 *Your next steps:*

1. Choose **Form | Close Tool** to return to form completion mode.
2. Choose **File | Save** or **File | Save As**.
3. After the **File | Save** dialog box appears type *Myorder* (or *myorder*—the case doesn't matter). ObjectVision automatically adds the extension *.OVD*.
4. Choose **OK**, or press *Enter*.

You have finished this part of the tutorial. You can quit ObjectVision and return to the tutorial later, or you can continue.

Creating form objects

In this chapter, you insert objects into the Sales Order form. After inserting the objects, you will define some of their properties. You will learn these ObjectVision procedures:

- Inserting fields, text elements, and a filled rectangle into the Sales Order form with the Objects commands.
- Copying and inserting fields with Edit | Copy and Edit | Paste.
- Renaming copied fields with the Properties | Name/Text command.
- Changing the label font of a text object.
- Removing the border of a text object.
- Applying the same border change to another text object with the Properties | Repeat command.
- Pasting a graphic from the Clipboard.

Form objects

You can insert six kinds of objects into a form: fields, text, filled rectangles, rounded rectangles, lines and graphics.

Fields

Fields are the areas of a form that display information. Depending on how you define fields, they get values in one of three ways: The user types or selects a value, ObjectVision calculates a value from a decision tree, or an external link provides a value.

After you insert a field into your form, you can define the field's display properties (text/numeric, selection list, button, and so on), label text, border, and the alignment of its value (left, right, or centered).

You can also protect the field from user modification, hide its decision tree from the user, attach help information to the field, or keep the field name from displaying.

Ordering your fields

When you insert fields into a form, remember that the order of the fields influences guided completion of an application during form completion and should be appropriate to your application's logic. ObjectVision evaluates fields from left to right and from top to bottom, ordering the fields by the position of their bottom left corner.

Text

Text objects are constant text strings that are always displayed on the form. You can use a text object to identify a field or to explain its use. After you insert a text object into a form, you can define its alignment, label font, and border.

Rectangles

Filled rectangles can improve the appearance of forms by filling in unused areas, and can improve readability by separating fields. When you insert a filled rectangle, you select one of several patterns. After you insert the filled rectangle, you can define its border. You should avoid a fill pattern that is similar to the fill pattern used to indicate a field value override.

Rounded rectangles are just like filled rectangles, except their corners are rounded instead of straight.

Lines

Lines can be drawn on a form to emphasize certain areas. You can select one of four line thicknesses using Properties | Line Width. These line thicknesses can also be applied to the borders of a selected field, text object, rectangle, or graphic.

Graphics

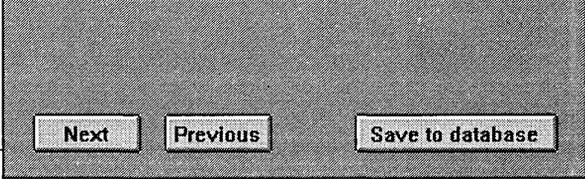
Graphics are visual elements you can use to help identify your forms and enhance their aesthetic appeal. Before you can insert a graphic, it must be placed onto the Windows Clipboard. Otherwise, you will see the error message,

Clipboard does not contain graphic

Inserting fields

Now you'll insert fields into your Sales Order form. Figure 3.1 shows the fully defined form again for you to refer to.

Figure 3.1
Sales Order form

Sales Order (Goal)			
 ACME Widgets		4030 Braker Lane West Suite 2001 Austin, TX 78759-5332	
Name		Order Date 10/12/90	
Company		Customer Type <input type="checkbox"/> distributor <input type="checkbox"/> dealer <input type="checkbox"/> educator <input type="checkbox"/> other	
Address			
City, State and ZIP			
Item Widget	Quantity	Unit Price \$295.00	Amount
Shipping Method	Discount		Less Discount
			Total Price
			Sales Tax
			Shipping Cost
			Extended Price

In this section, you'll use the following commands:

- Tools | Form, to open the Form Tool
- Objects | Field, to add fields
- Edit | Copy and Edit | Paste, to add more fields
- Properties | Field, to change the names of copied fields

Dialog boxes

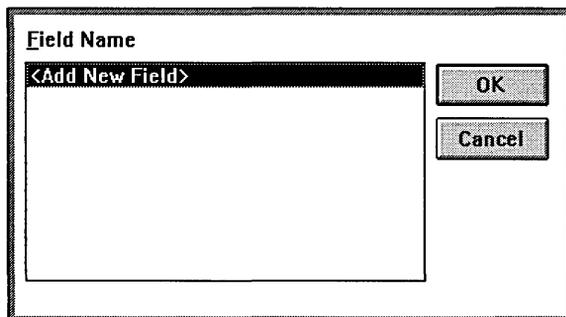
During the next procedure (and throughout the rest of the tutorial), you use dialog boxes to set options. Setting dialog box options involves these steps:

- Highlight the item you want (if it's not already highlighted).
- Choose the OK button to carry out your settings. With the keyboard you press *Tab* to highlight OK and then press *Enter* to carry out your settings. You can use the mouse to click OK.

 *Your next steps:*

1. If *Myorder* is no longer loaded, choose File | Open, and then select MYORDER.OVD from the list of application files displayed. Choose Open, or press *Alt+O* to open the *Myorder* application. The *Myorder* application opens in form completion mode.
2. Choose the Form command from the Tools menu. The Form Tool opens, with the Sales Order form displayed in Edit mode. Notice the Form menu in the menu bar. You use this menu to insert and define the form objects.
3. Choose Objects | Field. The Objects | Field | Field Name dialog box appears, as shown in the following figure:

Figure 3.2
Objects | Field | Field Name
dialog box



The dialog box contains only one item, <Add New Field>. After you add fields, the field names appear in this list.

4. Select <Add New Field>. (Because <Add New Field> is already highlighted, you can just press *Enter* or choose OK.)
5. In the Objects | Field | Field Name dialog box that then appears, type *Name*, and then press *Enter* or click OK.

Now a cross-hair pointer appears on the otherwise blank Sales Order form. This pointer lets you create the Name field.



Using the mouse:

 Your next steps:

1. Hold down the left mouse button while you drag the cross-hair pointer down and to the right. As you do so, a dashed rectangle appears showing the size and position of the field.
2. Release the mouse button when the rectangle is the correct size.

Note that the line surrounding the field remains dashed and that a small black square appears at each corner of the field. This indicates that the field is still selected.

You can change its size by putting the mouse pointer on one of the corners and dragging the corner to a new position. You can also change the field's position by putting the mouse pointer anywhere inside the field and dragging the entire field to a new position.

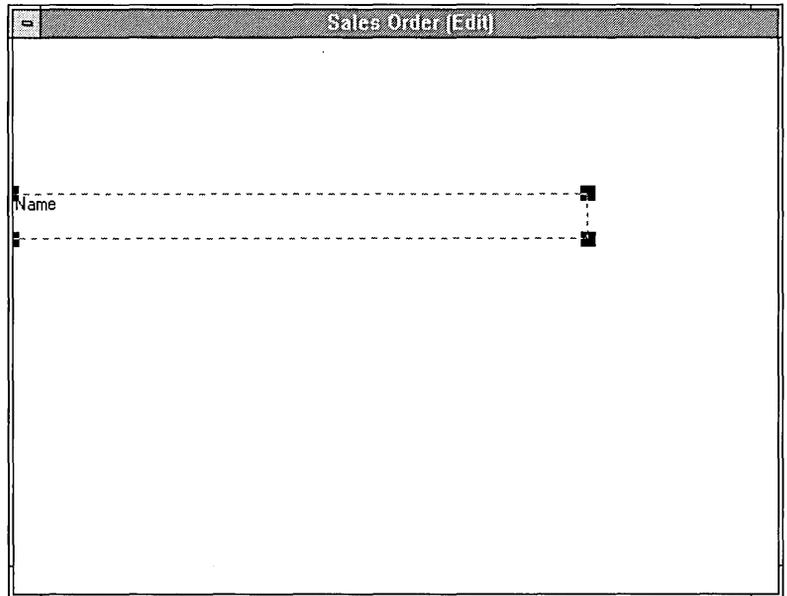
You can move the entire field by using the \uparrow , \downarrow , \rightarrow , and \leftarrow keys. You can move the lower right corner of the field by pressing *Shift* and any one of the arrow keys.

Using the keyboard:

1. Move the pointer to the left border of the form, to the appropriate height for the top of the Name field, then press *Enter* to anchor the position.
2. Press \rightarrow and \downarrow until the Name field is the correct size, then press *Enter* again when you finish.

Figure 3.3 shows how your Sales Order form should look when you have correctly placed the Name field.

Figure 3.3
Placement of the Name field



Sizing the field

It's important to make a field high enough and wide enough to display its name and the largest possible value that the field can have.

If the field is linked to a database field, you must make the Object-Vision field the same width as the database field. For example, the Name field in the Sales Order form is linked to a database field that can contain up to 48 characters, so you must make the Name field big enough to contain 48 characters. If an ObjectVision field isn't large enough to contain its linked value, users won't be able to display or enter new data longer than the field size.

When you create a field, you can make it whatever size you want. If you later find that the field is the wrong size, you can easily change it:

 *Your next steps:*

1. To make the Name field wider, begin by moving the pointer to the lower right corner of the field.



2. When the pointer changes to a black square, hold down the mouse button and drag the corner to the right until the field is the correct width.
3. To make a field wider using the keyboard, press *Shift+ →* to move the lower right corner of the field (and thus the right border of the field) to the right.

Testing the size

After you have created a field, you should test it to make sure that it's the right size for the largest possible value it can contain. You can return to form completion mode and test the field immediately after you have created and sized it, or you can wait until you have finished this section. To test the field, type the largest anticipated field value and count the number of characters.

To return to form completion mode and test the Name field now, follow these steps:

Your next steps:

1. Choose Form | Close Tool.

ObjectVision returns you to form completion mode. The form status, [Goal], appears after the form name in the title bar. The Name field is selected and contains a blinking text pointer.

2. Type 48 characters into one line of the field and then delete them. If you run out of room, you need to make the field wider.

Whether or not you need to change the width of the Name field, you must now return to the Form Tool to continue the tutorial.

3. Choose Tools | Form.

Moving fields

As you continue creating the Sales Order form, you will insert many fields into it. You may need to move a field after you insert or paste it. Here's how to move a field:



Using the mouse:

Your next steps:

1. Move the pointer to any point inside the Name field and hold down the mouse button.

2. Move the field to a new position and then release the mouse button. Notice that while you are moving the field, the mouse pointer changes to a cross-hair pointer.
3. Repeat steps 1 and 2 to move the field back to its proper position.

Using the keyboard:

1. Press ←, →, ↑, or ↓ to move the selected field.
2. Press *Tab* when you finish and you want to select another field. (In this case, of course, there aren't any other fields to select.)

Copying fields

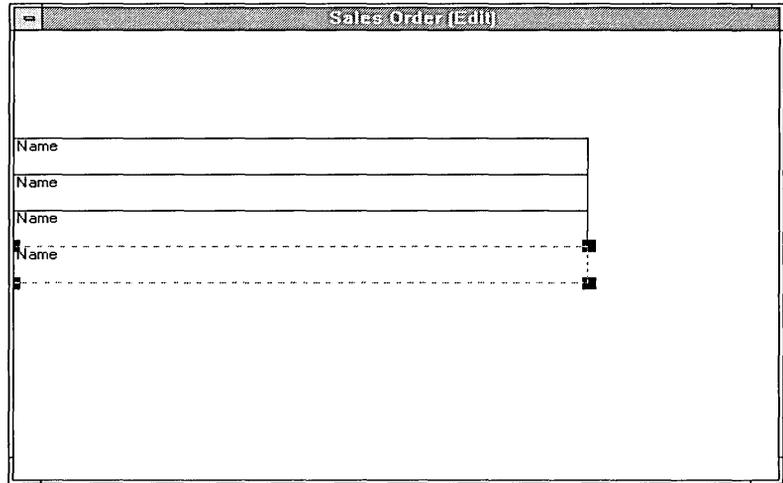
The Company; Address; and City, State, and ZIP fields are the same size as the Name field that you just inserted. You can insert each of these fields individually, as you did the Name field, or you can create them by copying the Name field and then changing their names. Here you will learn how to copy the Name field.

The copy and paste commands use the *Form Clipboard*, a temporary holding place (not visible on your screen) for cut and copied form objects.

 *Your next steps:*

1. The Name field is already highlighted with dashes, so choose Edit | Copy to copy the field to the Form Clipboard.
2. Choose Edit | Paste to paste the copy onto the form. You can also use the shortcut key for the Edit | Paste command, *Shift+Ins*.
3. Use the mouse or the arrow keys to position the copy directly under the Name field. Then click or press *Enter* to finalize the placement of the copy.
4. Choose Edit | Paste or press *Shift+Ins* twice more and paste the copies below the other two fields. (You don't have to copy the Name field again because the copy stays on the Form Clipboard when you paste it.) Figure 3.4 shows how your screen should look now.

Figure 3.4
Pasted copies of Name field



Renaming copied fields

Two commands change the name of a copied field:

- **Properties|Field** changes the name of the copy, while the original field—the one you copied—keeps its own name. Thus the Properties | Field command creates a new field.
- **Properties|Name/Text** changes the name of both the original field and the copy (thus giving a new name to an existing field). The Properties | Name/Text command changes the name of a field on every form in which it occurs, in decision trees, and in external links. (If you used the field name in help text, though, you have to change it yourself.)

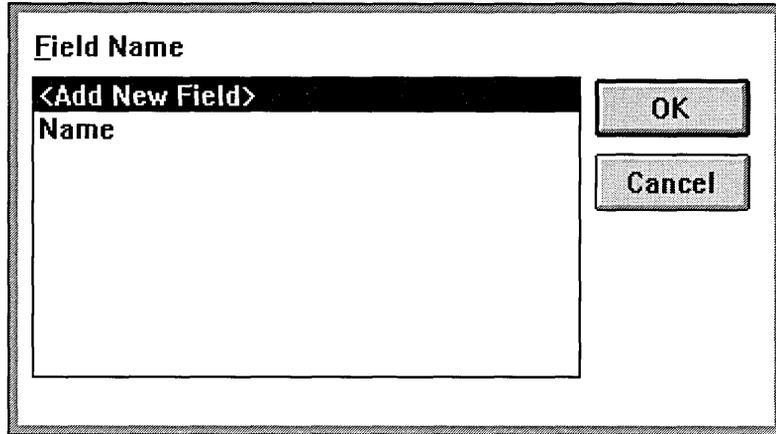
Here you'll use the Properties | Field command because you want the copied field to be a new field, distinct from the original field.

 *Your next steps:*

1. Select the first field you pasted—the one below the original Name field—and choose Properties | Field.

The Properties | Field Name dialog box appears and you can select <Add New Field> or a field name that is already defined.

Figure 3.5
Properties | Field Name
dialog box



2. Choose <Add New Field>.
3. Type *Company* in the Field Name dialog box that appears.
4. Repeat steps 1 through 3 to rename the other two fields Address, and City, State, and ZIP.

Adding the remaining fields

Now you will add the rest of the fields on the Sales Order form.

 *Your next steps:*

1. Choose Objects | Field to add the Customer Type field, positioning and sizing it as shown on your printed copy of the Sales Order form or in Figure 3.1. You will later format Customer Type as a check box field and add its list of possible values, but for now the field is empty.
2. Choose Objects | Field to add the Save to database field, positioning and sizing it as shown in Figure 3.1. On the finished form, the Save to database field has a rounded shape, but the field is rectangular on your form. In Chapter 5, you will change Save to database's field type to Button, which gives the field the shaded-button look.
3. Choose Objects | Field to add the remaining fields for the Sales Order form. Refer to Figure 3.1 (or your printed form) for their names, sizes, and positions. Note that the form border might resize automatically as you add objects.
4. Choose Form | Close Tool, then File | Save to save your work.

5. Choose Tools | Form to return to form edit mode.

In Chapter 5, you'll define the properties of the fields.

Adding text objects

The *Myorder* application uses text objects to display the company name and address on the Sales Order form.

You will use these commands:

- Objects | Text, to add the text objects
- Properties | Label Font, to change the font of the text
- Properties | Borders, to remove the border from one text object
- Properties | Repeat, to remove the border from the other text object. Properties | Repeat applies the last format change you made to a selected object.

The company name and its address are two separate text objects. The first text object you add is the company name, *ACME Widgets*.

 Your next steps:

1. Choose Objects | Text.
The Objects | Text | Text Value dialog box appears, and you can enter the text you want to display.
2. Type *ACME*.
3. Press *Ctrl+Enter* to move to a new line.
4. When you press *Enter*, the dialog box closes, the same as choosing OK. If you inadvertently press *Enter*, you can return to edit your text in the dialog box by choosing Properties | Name/Text.
5. Type *Widgets* and press *Enter*.
6. Using the cross-hair pointer, position and size the *ACME Widgets* text object where it goes at the top of the form, referring to Figure 3.1 or your printed copy of the Sales Order form.
When you finish, *ACME Widgets* appears in the upper left corner of the text field.

Changing the font

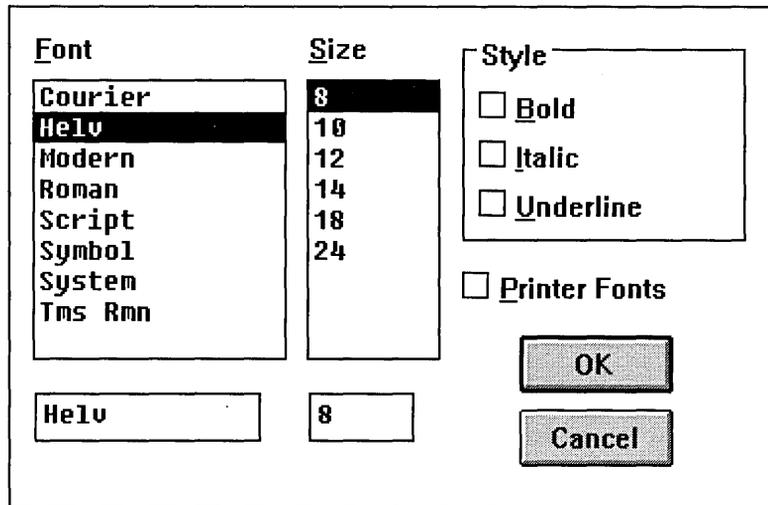
In the next steps, you change the displayed font for the company-name text object.

 Your next steps:

1. With the ACME Widgets text object still selected, choose Properties | Label Font.

The Properties | Label Font dialog box appears, as shown in Figure 3.6. Your list of fonts and sizes might be different. See your Windows documentation for details about font installation.

Figure 3.6
Properties | Label Font dialog box



With this dialog box, you can define the font style and size you want for the text. You can also check the Bold, Italic, or Underlined options. Note that the default font size of the ACME Widgets object is 8. This means that the text displays in 8-point type.

2. Select font size 24.
3. Check Bold Style.
4. Choose OK or press *Enter*.

The ACME Widgets characters are now larger and darker. You might need to reposition and resize the object to fit them on the form.

Using the mouse:

- You drag a selected field to reposition it. To resize a selected field, place the pointer on one of the black rectangles, called handles, at the corners of the object. Hold down the mouse button and drag the handle until the object is the size you want, then release the mouse button.
- You can also select multiple fields and then reposition or assign new properties for the fields as a group. For example, to move all existing fields below ACME Widgets, you can select the Name field, then hold down *Shift* and click the Extended Price field with the mouse. Note that all the fields you created between Name and Extended Price are now selected.
- You can also hold down *Ctrl* and click a field to select multiple fields. This method, unlike *Shift*, adds one field at a time to the multiple selection.

Using the keyboard:

- You reposition an object by selecting it and then pressing an arrow key in the direction you want to move it. To resize an object, hold down *Shift* while you press an arrow key.
- To select multiple fields from left to right and top to bottom, press *Ctrl+Tab*. To select multiple fields in the opposite direction, press *Shift+Ctrl+Tab*.

Adding the address

In the next steps, you add the company address to the Sales Order form.

 *Your next steps:*

1. Choose Objects | Text.
2. Type this value, pressing *Ctrl+Enter* for each new line. Press *Enter* when you finish:
4030 Braker Lane West
Suite 2001
Austin, TX 78759-5332
3. Position and size the field, referring to Figure 3.1 or your printed copy of the Sales Order form. You'll remove the border later in the tutorial.

4. Choose Properties | Label Font, select the Times Roman (Tms Rmn) font, and set the font size to 10.
5. Choose OK.

Removing the border

In the next steps, you remove the border from the text objects you added.

 *Your next steps:*

1. With the company address text object selected, choose Properties | Borders. Notice that the default value, Outline, is checked.
2. Uncheck the Outline box.
3. Choose OK.
The border has been removed from the object, but it still displays a dashed selection border. When you select another field, your text object displays without a border.
4. Use Properties | Repeat to remove the borders from the company-name text object. First, select the company-name text object. Then you can choose the Repeat command from the Properties menu or press *F4*, the shortcut key.
5. Choose Form | Close Tool, then File | Save to save your work.
6. Choose Tools | Form to return to form edit mode.

You have almost finished adding objects to the Sales Order form. Next, you'll add filled rectangles. The following figure shows how your screen should look now:

Figure 3.7
Your Sales Order form with
text objects

Sales Order (Edit)			
ACME Widgets		4030 Braker Lane West	
		Suite 2001	
		Austin, TX 78759-5332	
Name			Order Date
Company			Customer Type
Address			
City, State, and ZIP			
Item	Quantity	Unit Price	Amount
Shipping Method	Discount		Less Discount
			Total Price
			Sales Tax
Next			Shipping Cost
Previous			Extended Price
Save to database			

Adding filled rectangles

Filled rectangles can add to the clarity and attractiveness of your applications. You can use a filled rectangle to occupy a blank spot in a form or to set off fields. You can also place a filled rectangle and a field or text object in the same position to emphasize the field or text object (unless the pattern is so dark that it covers the characters). The *Myorder* application uses filled rectangles to fill the areas below the Shipping Method and Discount fields and between the Discount and Less Discount fields.

To add filled rectangles to the Sales Order form, you will use the Objects | Filled Rectangle command.

To remove the borders from the filled rectangles, you will use the Properties | Borders command.

 Your next steps:

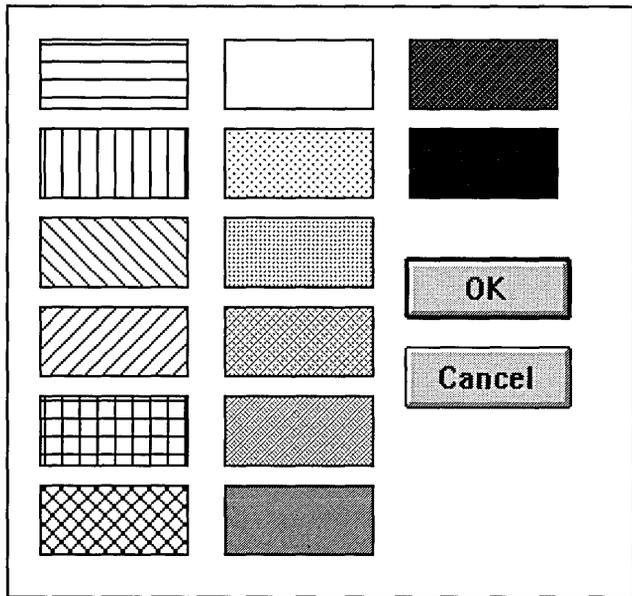
1. Choose Objects | Filled Rectangle. The cross-hair pointer appears.
2. Position the pointer at the upper right corner of the Discount field, and then drag the pointer down and to the right until the

rectangle exactly fills the space between the Discount and Less Discount fields.

The resulting rectangle displays a default fill pattern. The default fill pattern doesn't match the fill pattern shown in Figure 3.1, so you can change the fill pattern by choosing Properties | Fill Pattern.

The Fill Pattern dialog box then appears, as shown in the following figure. You can choose any of the fill patterns shown.

Figure 3.8
Properties | Fill Pattern dialog
box



3. Choose Objects | Filled Rectangle again.

This time, position the rectangle so that it occupies all the space from the lower left corner of the Shipping Method field to the lower left corner of the Extended Price field.

The Next, Previous, and Save to database fields show through the fill pattern (unless your monitor is unable to display the fill patterns as shown). Next you'll remove the line separating the first pattern element from the second. This line is the object's border.

4. Select the second filled rectangle, if it isn't still selected, and choose Properties | Borders. The Properties | Borders dialog box displays.

5. Uncheck the Outline box.
6. Choose OK.
7. Select the first filled rectangle and choose Properties | Borders.
8. Uncheck the Outline box in the Properties | Borders dialog box. Now the border is gone and the two filled rectangles appear to be one undivided pattern.
9. Choose Form | Close Tool, then File | Save to save your work.
10. Choose Tools | Form to return to form edit mode.

Inserting a graphic

ObjectVision can display and print any monochrome bitmap you paste into your form from the Clipboard. Color bitmaps display properly in ObjectVision, but can't be printed.

You'll use the Windows Paintbrush graphics application to place the ALOGO.BMP on the Clipboard so you can paste it onto your Sales Order form.

 *Your next steps:*

1. Minimize your ObjectVision application window by clicking the Minimize button, or by choosing Control | Minimize.
2. Open the Paintbrush application, which is in the Program Manager Accessories group.
3. In Paintbrush, choose File | Open and type the complete path name for the ALOGO.BMP file copied to your ObjectVision directory during installation. For example, if you installed ObjectVision in C:\VISION, you would type C:\VISION\ALOGO.BMP in the File Name field. Then choose OK.
4. Select the Selection Tool from the Tool palette on the left side of the window. It's the icon in the top row on the right side; showing a scissors and a rectangle.
5. Drag the selection rubber band around the entire outlined image.
6. Choose Edit | Copy. That copies the image to the Windows Clipboard.
7. Exit the Paintbrush application by choosing File | Exit from the Paintbrush menu bar, without saving the ALOGO.BMP file.

You just copied the graphic to the Windows Clipboard. Next, paste a copy of that graphic into your Sales Order form:

 Your next steps:

1. Open the ObjectVision icon by choosing Restore from the Control menu or double-clicking the icon.
2. Choose Objects | Graphic.
3. The Objects | Graphic | Save File Name As dialog box appears. Use this dialog box to give the graphic a name. Type *MyaLogo* and press *Enter*.
4. The graphic appears as a small, unplaced field. Move it and resize it to match the *Order* application's Sales Order form.
5. Choose Form | Close, then File | Save to save your work.
6. Choose Tools | Form to return to form edit mode.

Note that a file named *MYALOGO* has been created in the current directory and given the default file extension *.BMP*. This is an ObjectVision file format and can't be opened in Windows Paintbrush.

The graphic object now on your form has a default border just like any other object. Remove the border using the Properties | Borders command.

In the Form Tool, the name you specified for the graphic (or as much of it as will fit) appears in the field. That name only helps identify the graphic file—it is not displayed when users fill out the form, and it doesn't print. Your form now looks like this:

Figure 3.9
A graphic in the Sales Order form

Sales Order (Edit)			
		ACME Widgets	
		4030 Braker Lane West Suite 2001 Austin, TX 78759-5332	
Name		Order Date	
Company		Customer Type	
Address			
City, State, and ZIP			
Item	Quantity	Unit Price	Amount
Shipping Method	Discount		Less Discount
			Total Price
			Sales Tax
<input type="button" value="Next"/> <input type="button" value="Previous"/> <input type="button" value="Save to database"/>			Shipping Cost
			Extended Price

You have now inserted a graphic and all the necessary fields, text objects, and filled rectangles into the Sales Order form.

At this point, the fields in the Sales Order form have many default properties. For example, they all have the general display format type. In the next chapter, you will define different properties for many of the fields.

If you were to have a user complete the form now, ObjectVision would determine guided completion that included all the fields, because you haven't yet defined any decision trees or external links from which ObjectVision can calculate field values.

You have finished this part of the tutorial. You can quit ObjectVision now and return to the tutorial later, or you can continue now. If you intend to continue, you should save your form first. To do so, choose the Form | Close Tool command. Then choose the File | Save command.

Defining field properties

In this chapter, you define the characteristics of the fields you added to the Sales Order form. You will perform these Object-Vision procedures:

- Change the formats of several fields.
- Define possible values for the Shipping Method and Customer Type fields.
- Add help information to a field.
- Protect some fields from user input.

Field properties

When you insert a field into a form, the Form Tool gives the field a set of standard properties:

- **Display type:** General format (a format that is appropriate for displaying both text and numbers).
- **Values alignment:** Left alignment (values entered into the field are aligned at the left side of the field).
- **Label font:** The default character font for field names (8-point Helvetica unless you have changed it).
- **Border:** An outline around the field.
- **Protection:** No protection from modification or from decision-tree display during form completion.

- **Line width:** The thinnest line is used to draw field borders.

Once you have inserted the field, you can change any or all of these properties. In this chapter, you will change the properties of several fields in the Sales Order form.

Changing display formats

In this section, you'll change the display format type of several fields. The Form Tool provides a variety of display format types for fields so that you can choose the format most appropriate for the type of information the field will contain. For example, you can choose from among several formats suitable for numbers:

- **General** displays numeric values with the minimum number of decimal places to completely specify the number.
- **Fixed** displays numeric values with the number of decimal places you specify.
- **Percent** displays numeric values as percentages.
- **Financial** separates thousands with commas and displays negative values within parentheses.
- **Currency** places a dollar sign before numeric values, separates thousands with commas, and displays negative values within parentheses.
- **Picture** controls the format of the values users enter.

The Form Tool also offers a variety of formats for dates and times. For example, you can format a field to display the date that the user enters as numbers only (like *10/03/90*) or as words and numbers (like *03-Oct-90*).

The Form Tool also provides formats for selection lists, check boxes, true or false values, and button fields.

In this section, you'll also change the alignment of some fields. The value that a user enters in a field can be aligned left or right or centered. Values are aligned left unless you change the alignment. Numeric values are often better aligned right (so decimal points will line up, for example).

Selecting the date/time format

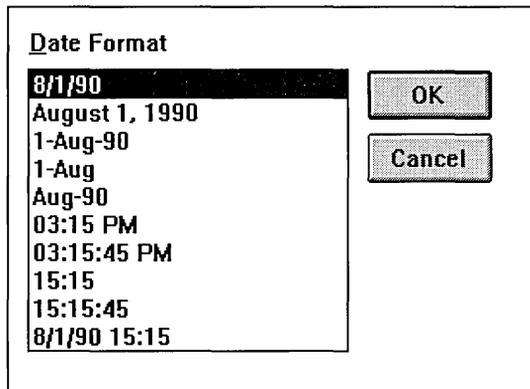
If you had quit ObjectVision before beginning this part of the tutorial, start ObjectVision and select the *Myorder* application. Then choose Tools | Form to open the Form Tool.

Next, you'll use Properties | Field Type to change the format of the Order Date field to Date/Time:

 Your next steps:

1. Select the Order Date field.
2. Choose Properties | Field Type.
The Field Type dialog box appears. This dialog box lists the format types you can select.
3. Select Date/Time, and click OK or press *Enter*. Now the Date Format dialog box appears, as shown in the following figure:

Figure 4.1
Date/Time dialog box



4. Select the first entry, 8/1/90, and click OK or press *Enter*. Now if a user enters May 13, 1991, in the Order Date field, the date will be displayed as 05/13/91.

If you would like to experiment with this format, you can select different date formats that you can test in form completion mode. You can close the Form Tool, type a value in the Order Date field, and press *Enter* to see how the value appears.

Selecting other display formats

In the next steps, you'll change the display format type of several numeric fields. You'll also format the Next, Previous, and Save to database fields as buttons. For some of the changes, you'll use Properties | Repeat.

 Your next steps:

1. Choose Tools | Form.
2. Select the Discount field, choose Properties | Field Type, and select the Percent format.
3. Click the mouse or press *Tab* to move the pointer to the Decimal Places box.
4. Type 0 for number of decimal places, and choose OK.
5. Select the Unit Price field, choose Properties | Field Type, and choose the Currency format, retaining the default value, 2, for Decimal Places.
6. Select the Amount field and press *F4* (the Properties | Repeat shortcut key). The Form Tool applies the last format you selected (Currency) to the Amount field.
7. Using Properties | Repeat, change the format type of the Less Discount, Total Price, Sales Tax, Shipping Cost, and Extended Price fields to Currency.
8. Select the Save to Database field, choose Properties | Field Type, select Button, and then choose OK. Note that the button field is now shaded just like other buttons in Windows.
9. Select both the Next and Previous fields. Using the keyboard, select Next, then press *Ctrl+Tab*. With the mouse, click Next, hold down *Ctrl* and click Previous. Choose Properties | Repeat to assign them the button field type.

Select multiple objects by holding down *Ctrl* and pressing *Tab*

Changing alignment

Next, you change the alignment of the Quantity and other numeric fields from left alignment (the default) to right alignment. With right alignment, values in these fields align at the right edge of the field.

 Your next steps:

1. Select the Quantity field, choose Properties | Alignment, and choose Right alignment.
2. Use the Properties | Repeat command to apply right alignment to the Unit Price, Amount, Discount, Less Discount, Total Price, Sales Tax, Shipping Cost, and Extended Price fields.
Or, select all the fields using *Ctrl+Tab* or *Ctrl+Click* before choosing Properties | Alignment.
3. Choose Form | Close Tool then File | Save to save your work.

Adding selection lists and check boxes

In this part of the tutorial, you change the format display type of the Shipping Method and Customer Type fields. Each of these fields display a list for the user to select from (although the Shipping Method list doesn't usually appear because Object-Vision calculates that field's value from a decision tree).

You will format Shipping Method as a selection-list field; the user selects one of the items listed. After the user selects an item, only that item is displayed in the field.

You will format Customer Type as a check-box field; each item listed is preceded by a check box, and the user selects an item by clicking its box. After the user selects an item, all items continue to be displayed, and the selected item's check box contains an X.

You can select either of these format types for a field that will have a set of possible values. The check-box format lets the user make "what if" changes easily. The selection-list field is especially appropriate when the list of value options is too large to be continuously displayed.

Creating value option lists

A field that you format as a check-box or selection-list field presents the user with a list of items from which to choose. When you format a field as check box or selection list, you can use either of two methods to provide the list of possible values for the field:

- Enter the list of values when you format the field. With this method, the field gets its list of possible values right away.

- Let ObjectVision automatically determine the list from values you define in one or more decision trees, or from a linked database. (This is the default method.) If you choose this method, the field will get a list of possible values when you add the relevant decision tree or establish links. (Later in this tutorial you will see how ObjectVision creates the value options list, examining the application's decision trees or linked database fields.)

When you are creating a form, you might want to add the list of possible values (or some of the values) for a selection-list or check-box field yourself, so that you can have users test your application. Later, when you have added decision trees (and possibly defined new values for the field in the trees), you can easily modify the field type so that ObjectVision determines the values automatically.

Adding the list of possible values yourself also gives you an idea of how large the field needs to be.

In this part of the tutorial, you will add the possible values yourself. Later you will change the Customer Type field so that ObjectVision determines its values automatically.

Now you will format the Shipping Method and Customer Type fields with the Properties | Field Type, adding a list of possible values for each field:

 *Your next steps:*

1. Choose Tools | Form and select the Shipping Method field.
2. Choose Properties | Field Type, and choose the Selection List entry.

The Values of dialog box appears, as shown in the following:

Figure 4.2
Values of dialog box

The dialog box is titled "Values of: Shipping Method". It features a checked checkbox labeled "Automatic". Below this is a section titled "New Value" with an empty text input field. To the right of this field are two buttons: "Insert" and "Delete". Below the "New Value" field is a larger empty list box titled "Values". To the right of this list box are three buttons: "OK" and "Cancel".

Notice that the Automatic check box is checked (Automatic is the default value) and the Values field is empty.

3. Move the pointer to the New Value field and type *Commercial Carrier*.
4. Press *Enter*.

Commercial Carrier appears in the Values list. Notice that the Automatic box is now unchecked, showing that ObjectVision won't determine the possible values for the field. The pointer remains in the New Value field.

5. Type *Regular UPS* and press *Enter*.
6. Type *2-day Express* and press *Enter*.

Now the Values box contains three possible values for Shipping Method.

7. Choose OK. The Values of dialog box closes.

Notice that the selection list is not yet displayed in the Shipping Method field. A selection list is not displayed until the user reaches the selection-list field while in form completion mode. If you want, you can check the selection list in the Shipping Method field by closing the Form Tool and then selecting the Shipping Method field in form completion mode.

Now you will format the Customer Type field and add a list of possible values.

 Your next steps:

1. Select the Customer Type field, and then choose Properties | Field Type and Check Boxes.
2. When the Values of dialog box appears, add the following values, using the same procedure you used for the Shipping Method field:
 - distributor
 - dealer
 - educator
 - other
3. The four values you typed for the Customer Type field appear with check boxes. If the boxes do not appear, your Customer Type field is not large enough to display the values.

Adding help to a field

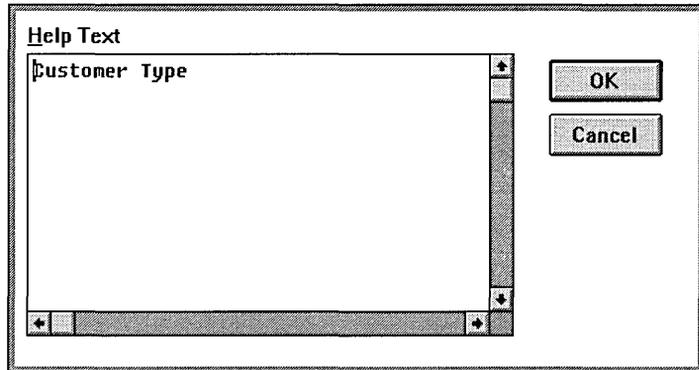
ObjectVision provides help for ObjectVision commands, tools, and procedures. You can also add help for individual fields. If there is any possibility that the user might not understand what is required in a field, you should add help information to the field. Then, after selecting that field, the user can see its help information by pressing *F1*.

Now add help information to the Customer Type field:

 Your next steps:

1. From within the Form Tool, select the Customer Type field, and then choose Properties | Help.
The Form Tool prompts you for the help information, as shown in Figure 4.3. Notice that the prompt window already contains the name of the field, Customer Type.

Figure 4.3
Help dialog box



2. Press *End* to move the insertion point to the end of the field name, press *Spacebar*, and then type this text: is defined as follows:
3. Press *Ctrl+Enter* twice to move down two lines. (If you press *Enter* alone, the Form Tool thinks you have finished and closes the window.)
4. Add the rest of the help information shown next, using *Ctrl+Enter* to get new lines. The Form Tool automatically scrolls your text horizontally as you enter long lines, so you don't have to press *Ctrl+Enter* as you approach the right edge of the window.

Type

Distributor - Signed distribution agreement on file

Dealer - Must provide valid tax ID

Educator - Recognized college or university

Other - any individual or company not listed above

5. Choose OK or press *Enter* to close the Help dialog box. To test your new field-specific help, you can choose Form | Close Tool, and press *F1*.

Protecting calculated fields

You can use the Form Tool to add two kinds of protection to ObjectVision fields:

- You can prevent the user from overriding a calculated field.
- You can prevent the user from viewing a field's decision tree.

Not all calculated fields require protection. The user might find a calculated field's decision tree useful for understanding form results; in many cases, the user should be able to change the value of a calculated field.

For example, in the *Order* application, ObjectVision calculates the value of the Order Date field, but in some cases a user may need to enter a different date. ObjectVision also calculates a value for the Item field, to provide a default value (because the great majority of the company's orders are for widgets). A customer may occasionally order something else, though, so the user should be able to override the Item field's calculated value.

In many cases, however, protection for calculated fields is necessary. Some calculated fields in an application should not be overridden or bad values can result. For example, ObjectVision might calculate an inconsistent value for the Extended Price field if a user overrode the calculated value for the Total Price field.

Some applications (those in sensitive areas like personnel management, for example) may also require that certain decision trees be hidden from users.

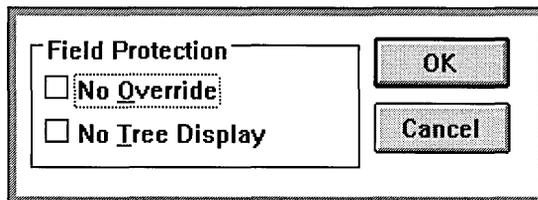
In the next steps, you use the Properties | Protection command to protect several fields in the Sales Order form from being overridden.

 Your next steps:

1. From within the Form Tool, select the Amount field, and then choose Properties | Protection.

The Form Tool prompts you to select the type of protection you want to add to the field, as shown in the following figure:

Figure 4.4
Properties | Protection dialog
box



2. Check No Override and then click OK or press *Enter*.
3. Use Properties | Repeat to add no override protection to the Less Discount, Total Price, Shipping Cost, and Extended Price fields.

Note that you didn't add override protection to the Discount and Sales Tax fields. The user may need to change those values in particular circumstances, and changing them won't cause Object-Vision to calculate inconsistent results.

If you close the Form Tool now and try to modify one of the protected fields, you will find that you can't. However, you can still select the fields (so that you can later add or view decision trees if you want).

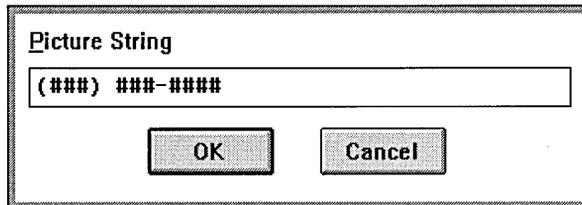
Creating a picture field

Your *Myorder* application doesn't have a picture field, which is a powerful feature that controls user input. In this lesson, you'll create a Phone field and change it from a general field to a picture field. Then you'll define the *picture string* so the user will be restricted to typing the phone number in the format you specify.

 Your next steps:

1. Choose Objects | Field, select <Add New Field>, and then choose OK. The Add New Field | Field Name dialog box appears.
2. Type the field name *Phone* and press *Enter*.
3. Position the new field somewhere near the bottom of your form (you'll be removing it later, so its exact placement doesn't matter).
4. Choose Properties | Field Type.
5. In the Properties | Field Type dialog box, choose Picture.
6. In the Properties | Field Type | Picture String dialog box, type *(###) ###-####*. Your dialog box should look like this:

Figure 4.5
Field Type | Picture dialog
box



7. Choose OK or press *Enter*.

You might want to close the Form Tool and try typing values into the Phone field now. Pictures can help you easily maintain data entry consistency for unusual field values, such as Social Security numbers, phone numbers, serial numbers, and part numbers.

8. Choose Tools | Form to return to form edit mode.

9. Select the new Phone field, and then choose Edit | Cut to remove the field from your form.

For more information about picture fields, see Chapter 3 in *Object-Vision Reference*. You have finished this part of the tutorial. If you intend to continue, be sure to save your form first.

Creating a decision tree

In this chapter, you add a decision tree to the Shipping Method field. You will perform these ObjectVision procedures:

- Start the Tree Tool.
- Use the Objects menu commands to insert branch nodes, *conditions*, and conclusion nodes into the Shipping Method decision tree.
- Test your decision tree.

Tree Tool

A decision tree graphically expresses the logic that ObjectVision uses to calculate a field value.

Not every field has a decision tree. You define a decision tree for a field only when you want ObjectVision to calculate the field's value. If the field will get its value from user input or from an external link, it doesn't need a decision tree.

To create a decision tree, use the Tree Tool. Like the Form Tool, the Tree Tool is a high-level, easy-to-use graphical utility. You create a tree one step at a time by adding nodes, the individual pieces of the tree. The Tree Tool helps you by prompting for the appropriate information at each step and by displaying the nodes that you create.

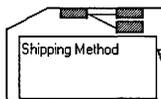
Elements of a decision tree

To understand how to use and read decision trees, you need to understand the parts of decision trees. This section defines terms that are used to describe different parts of decision trees. Glance through the list before you continue with the tutorial. Then refer to the list as necessary as you proceed.

- Decision tree** A *decision tree* is a graphical expression of decision logic. It is made up of one or more nodes, and is used to calculate a field value. Each field whose value must be calculated, rather than entered or retrieved through a link, must have a decision tree. Each decision tree must have at least one *conclusion node*.
- Decision logic** *Decision logic* is a process of small steps. At each step, a single field value is evaluated to determine which subsequent branch to take.
- Node** A *node* is an individual segment of a decision tree. A node can be either a branch node or a conclusion node.
- Branch node** A *branch node* is a node that evaluates a single field value to determine which subsequent path to follow. Each branch node refers to a single ObjectVision field. The name of that field displays within the branch node. Branch nodes can be either simple or complex. Simple and complex branch nodes function the same way.

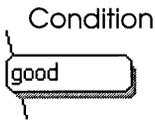


A *simple branch node* is a node that ObjectVision can evaluate without referring to the decision tree of any other field. The illustration on the left shows the symbol used for simple branch nodes. The entire decision logic for the node is displayed in the Tree Tool when the decision tree for a simple branch node is shown.



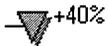
A *complex branch node* is a mode that ObjectVision can evaluate only by referring to the decision tree of at least one other field. The decision trees of any other fields do not display in the Tree Tool while the decision tree for the currently selected field displays. The illustration on the left shows the symbol used for complex branch nodes. If a complex branch node is double-

clicked with the mouse, the associated decision tree is shown in the Tree Tool.



All nodes that follow a branch node, regardless of their type, have a *condition* associated with them. That condition is compared to the current value of the branch node's field. The condition is evaluated to determine which subsequent path to take.

Conclusion node



A *conclusion node* (marked with an inverted triangle placed to the left of an expression in a decision tree) is an end node that supplies a result for the decision process. A conclusion node can follow a branch node, or it can be the only node in a decision tree. A button field, for example, usually has only a single conclusion node in its decision tree. A conclusion node provides a possible value for the calculated field. If the conditions leading to a conclusion node are all true when ObjectVision evaluates the tree, the conclusion node's value becomes the field's value.

Root node

The *root node* is the first, or leftmost node in the decision tree. A root node can be either a branch node or a conclusion node. A root node that is a conclusion node is the only node in the decision tree.

Empty node

An *empty node* acts as a place holder for other nodes while a decision tree is edited. An empty node indicates the absence of a decision tree, or that there are no nodes beneath a branch node. If an empty node is evaluated, ObjectVision returns an error value.

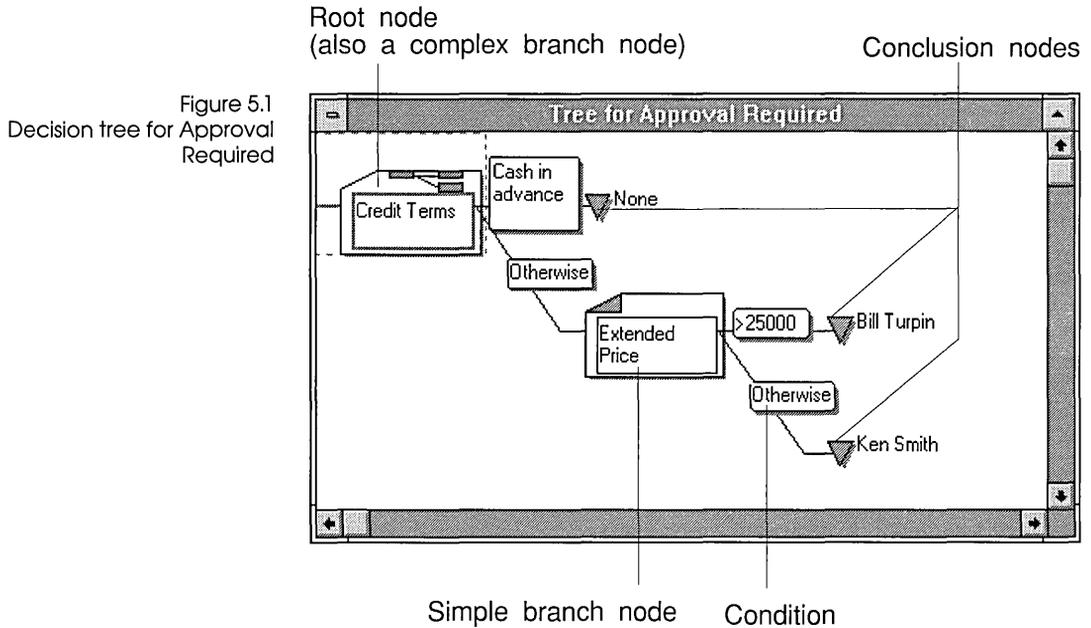
Nesting level

A node's *nesting level* is the distance from that node to the root node from left to right.

Summary

You can create many different decision trees, depending on the logic you need to express. Decision trees can be very simple, with only a conclusion node, or very complex, with many branch nodes and conclusion nodes.

Figure 5.1 shows the decision tree for the Approval Required field from the *Credit* application. This example is more complex than any decision tree you will create for the *MyOrder* application.



In the decision tree for Approval Required, there are three conclusion nodes, with results of *None*, *Bill Turpin*, and *Ken Smith*. The conclusion node with the result *None*, has a condition of Cash in Advance.

The root node is Credit Terms, which is also a complex branch node with no condition. If Credit Terms is double-clicked with the mouse, another decision tree is displayed.

Extended Price is a simple branch node, with a condition of Otherwise.

Read from left to right and—from top to bottom, the decision tree is a graphic expression of these steps:

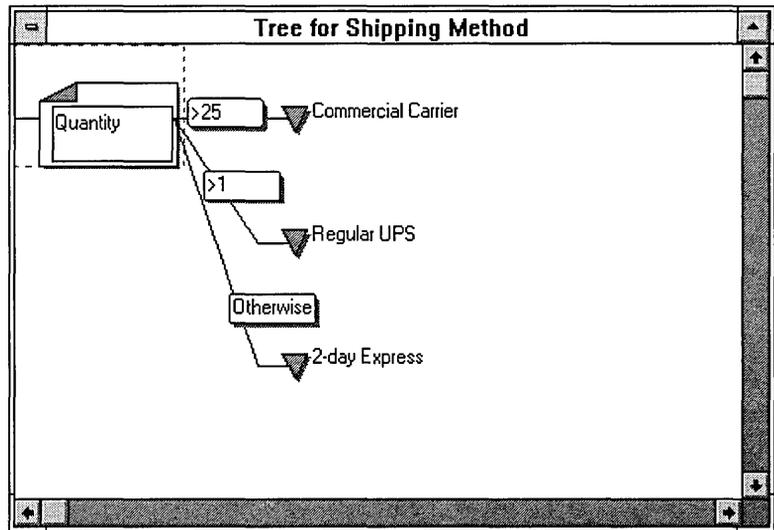
1. If the value of Credit Terms is Cash in advance, then the value of Approval Required is None.
2. If the value of Credit Terms is anything other than Cash in advance, then evaluate Extended Price.
3. If the value of Extended Price is greater than 25,000, then the value of Approval Required is Bill Turpin.

4. If the value of Extended Price is anything other than greater than 25,000, (that is, the extended price is not greater than 25,000), then the value of Approval Required is Ken Smith.

Decision tree for Shipping Method

The completed Shipping Method decision tree from the *Order* sample application is shown in the following figure:

Figure 5.2
The completed decision tree for Shipping Method



The logic expressed in the Shipping Method decision tree is explained as follows:

1. If the value of the Quantity field is greater than 25, then the value of Shipping Method is Commercial Carrier.
2. Or else, if the value of the Quantity field is greater than 1, then the value of Shipping Method is Regular UPS.

Note that a value of, say, 26 is both greater than 25 and greater than 1; it satisfies both of the first two conditions. When a value satisfies the first condition, however, subsequent conditions are never evaluated. In effect then, the second condition really applies only to values greater than 1 and less than or equal to 25.

3. Otherwise (if the value of the Quantity field satisfies neither of the preceding conditions), the value of Shipping Method is 2-day Express.

Note that a Quantity of 1, 0, or a negative number would satisfy the Otherwise condition and return a Shipping Method of 2-day Express. Since a Quantity of 0 requires no Shipping Method, and a negative Quantity is clearly impossible, this logic may seem flawed. In Chapter 8, however, you will learn how to validate user input—you won't allow the user to enter 0 or a negative number in the Quantity field.

Creating the decision tree

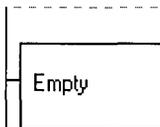
You will add the Shipping Method decision tree to your form in the next steps. That decision tree, as shown in Figure 5.2, will contain the following items:

- a simple branch node (the root)
- conditions
- conclusion nodes

If you had quit ObjectVision at the end of the last part of the tutorial, start ObjectVision now, and open the *Myorder* application. Then follow these steps:

 *Your next steps:*

1. While in form completion mode, select the Shipping Method field.
2. Choose Tools | Tree. After you open the Tree Tool, it displays the tree for the selected field. Once the Tree Tool is open, you can also select other trees to work with.



Note the title at the top of the tree: *Tree for Shipping Method*. Below the title is an empty branch node. The node is highlighted with a dotted line (so any edit or insert command you choose applies to that node).

Before continuing, take a moment to look at the commands available in the Tree Tool. Notice that from the Tree menu you can select another decision tree, find a tree that contains a specific field, or print the current tree (or all trees). From the other menus, you can modify nodes, cut and paste nodes, and perform other actions.

Now you will define the Shipping Method tree. First you add a branch node.

 Your next steps:

1. Choose Objects | Branch.

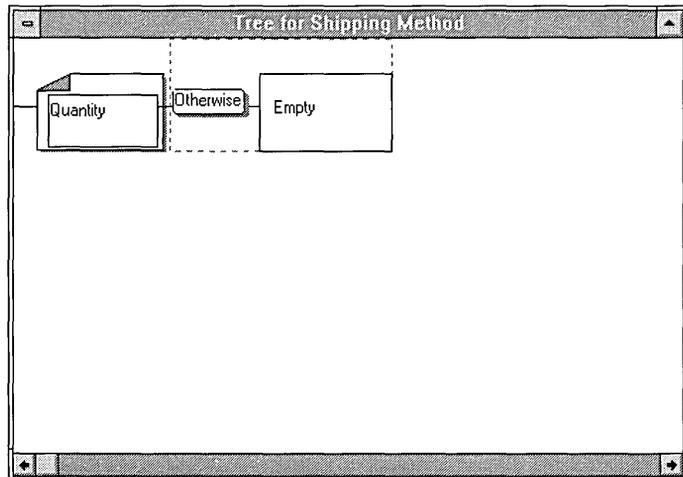
The Tree Tool displays a list of all the fields you have defined in the *Myorder* application. You can select a field from the list or select the Add New Field entry.

If you add a new field instead of selecting an already-defined field, and if you do not later add that field to the form, Object-Vision will use the Scratchpad form when it needs that field's value during form completion.

2. Select Quantity (to reach Quantity, you can scroll or press Q), then press *Enter* or choose OK.

Figure 5.3 shows how your screen looks now. Notice that the first node of the tree (the root node) is now labeled "Quantity," rather than "Empty." The tree still contains an empty node though, because you haven't yet added a conclusion node. Also note that the empty node has a condition associated with it: "Otherwise". Otherwise is a special ObjectVision condition to cover all cases not included in preceding conditions.

Figure 5.3
Branch node

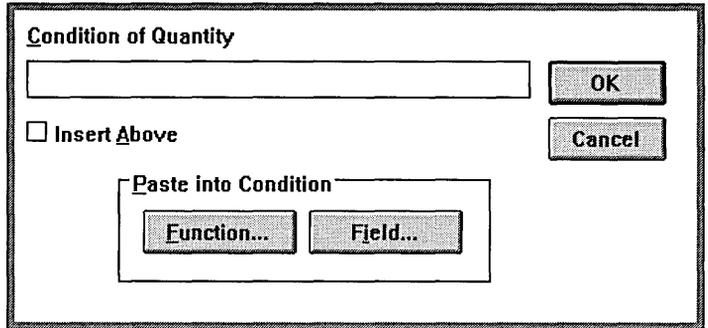


Next, you'll add a conclusion node.

 Your next steps:

1. Choose the Objects | Conclusion command. The dialog box shown in the following figure appears:

Figure 5.4
Condition dialog box

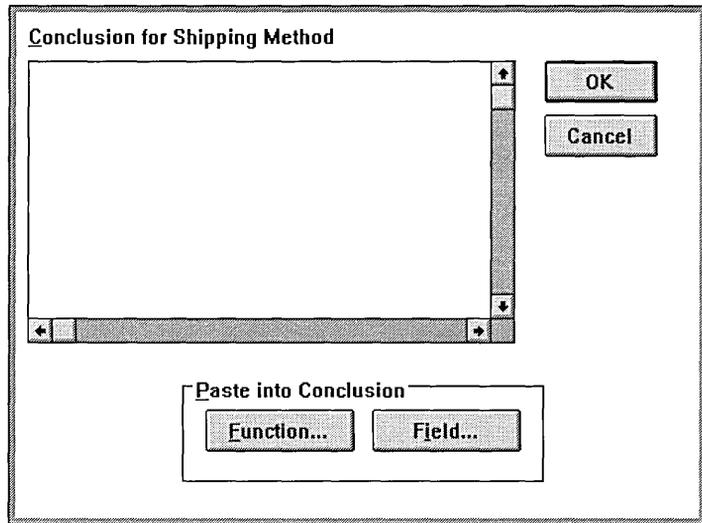


2. Type the condition for the conclusion you are adding. Notice that you can choose to have the conclusion node inserted above the currently highlighted node (if there is one); otherwise, the Tree Tool inserts the new node below the highlighted node.

Conditions can include field names and ObjectVision functions. Instead of typing in the names of the fields and functions, you can select them from a list (a good idea because it prevents typing mistakes). You'll paste field and function names later, when you add other decision trees. For now though, just type the value >25 , and click OK or press *Enter*.

Now the Conclusion dialog box appears. This is where you enter the conclusion expression. Here's what the Conclusion dialog box looks like:

Figure 5.5
The Conclusion dialog box



You can paste a field or function name into your conclusion expression, by choosing Paste Field or Paste Function. For now though, just type the conclusion.

3. Type `Commercial Carrier` and press *Enter*.
The Tree Tool displays the decision tree. Notice that the conclusion node you added, marked by an inverted triangle, has replaced the empty node. Next, you'll add the other two conclusions.
4. Choose `Objects | Conclusion`. Type `>1` for the condition and `Regular UPS` for the conclusion.
5. Choose `Objects | Conclusion`, and type `Otherwise` for the condition and `'2-day Express` for the conclusion. Notice that the apostrophe *must* precede the conclusion `2-day Express` because without it ObjectVision interprets it as an expression. The expression would create a new field named `day Express`, and ObjectVision would try to subtract the value of `day Express` from the number 2.

Now you have completely defined the Shipping Method decision tree. Figure 5.2 shows how your screen should look now (except that, in Figure 5.2, the selection box surrounds the root node instead of the Otherwise conclusion node).

Conclusion order

An important point about the way you created this decision tree is the order in which you entered conclusions. This section discusses conclusion order.

When you add more than one conclusion to a decision tree, the order in which you put them tells ObjectVision how to evaluate the corresponding field. ObjectVision evaluates the conditions that follow a branch node from top to bottom. The tree you just added for Shipping Method tells ObjectVision to follow this sequence to derive a value for the field:

1. Get a value for Quantity.
2. Look at the top condition: Is the value of Quantity greater than 25? If so, return the value Commercial Carrier for the Shipping Method field. If not, evaluate the next condition.
3. Is the value of Quantity greater than 1? If so, return the value Regular UPS for the Shipping Method field. If not, evaluate the next condition.
4. Return the value 2-day Express for the Shipping Method field.

If you changed the order in which you put these conclusions, ObjectVision might get the wrong value for the Shipping Method field. For example, suppose you put the second conclusion *Regular UPS* at the top of the set of conclusion nodes. Then ObjectVision would return Regular UPS for all cases in which the value of Quantity was greater than 1 (and wouldn't return Commercial Carrier for values greater than 25).

If you want to change the evaluation order of conclusion nodes you have already added, you can use Edit | Cut and Edit | Paste to rearrange them.

Testing your decision tree

Before you add more decision trees, you should test the Shipping Method tree to make sure its logic works the way you want it to.

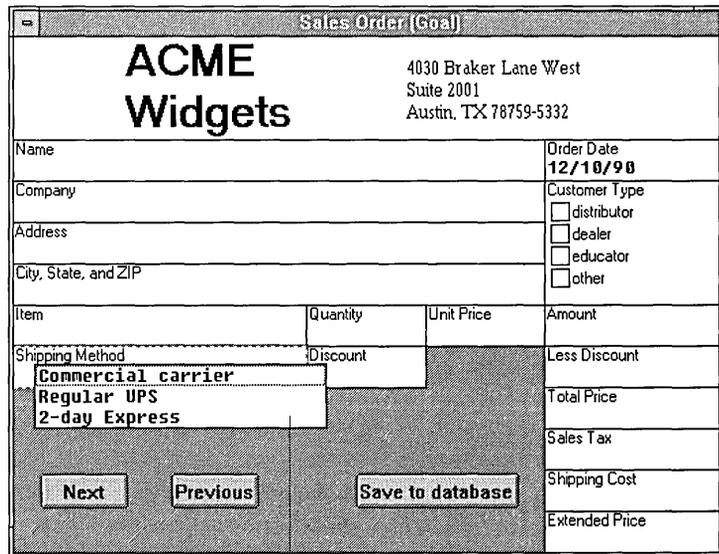
When a user overrides a calculated value, that user-entered value replaces the calculated value without removing the decision tree logic assigned to the field. To restore a field's calculated value, select the field and then choose Field | Calculate.

 Your next steps:

1. Choose Tree | Close Tool to close the Tree Tool and return to form completion mode.
2. Select the Shipping Method field.

Notice that the Shipping Method field is displaying a selection list, as shown in the following figure:

Figure 5.6
Shipping Method's list of value options



Sales Order (Goal)			
ACME Widgets		4030 Braker Lane West Suite 2001 Austin, TX 78759-5332	
Name		Order Date 12/10/98	
Company		Customer Type <input type="checkbox"/> distributor <input type="checkbox"/> dealer <input type="checkbox"/> educator <input type="checkbox"/> other	
Address			
City, State, and ZIP			
Item	Quantity	Unit Price	Amount
Shipping Method	Discount		Less Discount
Commercial carrier Regular UPS 2-day Express			Total Price
			Sales Tax
			Shipping Cost
			Extended Price

Next Previous Save to database

List of value options

The Shipping Method field is displaying its selection list because it is still highlighted, and ObjectVision is trying to get a value for it. Because the Quantity field doesn't have a value, ObjectVision can't calculate Shipping Method and needs to prompt the user for input.

3. Highlight the Quantity field, type the value 13, and then press *Enter*.
4. Verify that ObjectVision displays the value Regular UPS in the Shipping Method field.
5. Keep changing the value of the Quantity field and verifying that ObjectVision calculates the right value for Shipping Method in each case. (If you didn't precede the conclusion *2-day Express* with an apostrophe, entering a quantity of 1 will not return a value in the Shipping Method field.)

You have finished this part of the tutorial. You can quit ObjectVision and return to the tutorial later, or you can continue now. If you intend to continue, you should save your form first.

Editing a decision tree

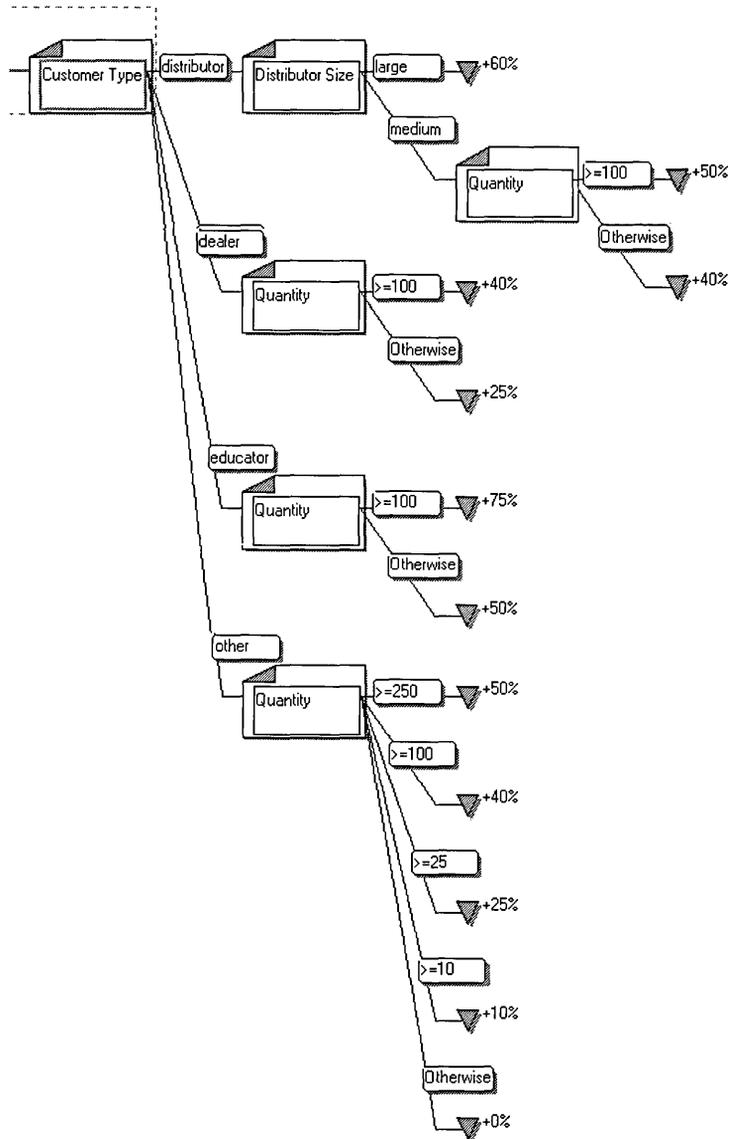
In this chapter, you create and modify a decision tree for the Discount field. This decision tree contains many branch and conclusion nodes. You will perform these ObjectVision procedures:

- Add a new field with the Tree Tool and create a new form for it.
- Scroll a decision tree.
- Copy and paste parts of a tree with Edit | Copy and Edit | Paste.
- Write conditions and conclusions with the Objects | Condition and Objects | Conclusion commands.
- Resize the tree display with View | Expand and View | Reduce.
- Print a tree.

The Discount decision tree

The following figure shows the completed decision tree for the Discount field. Notice that it contains a node for a new field, Distributor Size, which is not included on the Sales Order form. (Later you will create another form, Distributor Information, for the Distributor Size field.)

Figure 6.1
Decision Tree for Discount



The decision tree shows that the value of the Discount field depends on the values of the Customer Type, Distributor Size, and Quantity fields. The following text describes in part the logic shown graphically in the decision tree. Keep in mind that each step in the sequence (except for the first one) occurs only if the condition expressed in the previous step is not true.

If the value of the Customer Type field is *distributor*, then the value of Discount depends on the value of Distributor Size, as follows:

1. If the value of the Distributor Size field is *large*, then the value of Discount is 60 percent.
2. Or, if the value of the Distributor Size field is *medium*, and the value of the Quantity field is greater than or equal to 100, then the value of Discount is 50 percent.
3. Or, if the value of the Distributor Size field is *medium*, and the Quantity field has a value that is not greater than or equal to 100, then the value of Discount is 40 percent.

If the value of the Customer Type field is *dealer*, then the value of Discount depends on the value of Quantity:

- If the value of the Quantity field is greater than or equal to 100, then the value of Discount is 40 percent.
- Or, if the value of the Quantity field is not greater than or equal to 100, then the value of Discount is 25 percent.

Similar evaluations occur if the value of the Customer Type field is *educator* or *other*.

The logic of the decision tree continues along the same lines, from left to right and from top to bottom. At each step, the value of Customer Type is considered first. If the Customer Type is *distributor*, Distributor Size is considered next. Finally, Quantity is considered. Each step evaluates to true or false. If a condition is true, ObjectVision follows the path from that condition. If a condition is false, ObjectVision evaluates the next condition below. This continues until ObjectVision reaches a conclusion node. ObjectVision then assigns a value to Discount.

Adding the first nodes

You begin creating the Discount decision tree by adding the nodes for the first step in the decision logic:

 *Your next steps:*

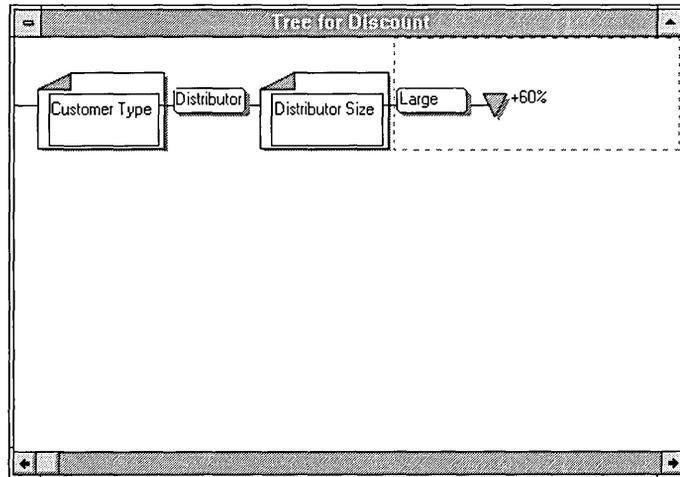
1. Open the *Myorder* application and select the Discount field.
2. Choose Tools | Tree.
3. Choose Objects | Branch, select Customer Type from the list of field names, and then choose OK.

4. Choose Objects | Branch again. The Condition dialog box appears.
5. For the condition of Customer Type, type `distributor` and press *Enter*. The Field Name dialog box appears.
Because you are creating a new branch node (and not a conclusion node), ObjectVision is expecting the name of a field. The field you need has not yet been created though. You want to create a new `Distributor Size` field, to categorize the types of distributors you do business with. In the next steps, you'll create a new field and assign it to the new branch node.
6. Select <Add New Field>, and choose OK. The Add New Field | Field Name dialog box appears.
7. For the new field name, type `Distributor Size` and press *Enter*.
8. Choose Objects | Conclusion to create a conclusion node. The Condition dialog box appears. This is because you must provide a condition for the conclusion node you are adding.
The conclusion node must provide a value for `Discount`, to be returned if the preceding conditions are true. In this case, you want to assign a particular value to `Discount` if the `Customer Type` is `distributor` and the `Distributor Size` is `large`. Because you are defining `Distributor Size` now, you must enter a value in the condition dialog box for `Distributor Size`.
9. For the condition of `Distributor Size`, type `large` and press *Enter*.
10. When the Tree Tool prompts you for the conclusion for `Discount`, type `+60%` and press *Enter*.

Note that the plus sign (+) is optional. Since the conclusion starts with a number, it is considered an expression, not a constant. You must use a single-quotation mark (') before any constants that start with a number (just like in Quattro Pro).

Now your decision tree should look like the one shown in the following figure:

Figure 6.2
Your Discount decision tree
with one conclusion node



Adding more conclusion nodes

In the next steps, you add the second and third conclusions to the Discount decision tree. These conclusions require another branch node.

 *Your next steps:*

1. Select the conclusion node +60% and then choose Objects | Branch. The Condition dialog box appears.

You selected the conclusion node because ObjectVision will insert the new branch node below the conclusion, at the same nesting level as the +60% conclusion node. You want the new branch node at that nesting level because you want to consider only Distributor Size here. If you wanted to add a branch node that evaluated other Customer Types, you would have selected the Distributor Size branch node, and entered a different Customer Type (as you will do later in this chapter).

2. When the Tree Tool prompts you for the condition of Distributor Size, type `medium` and press *Enter*.
3. For the field name, select Quantity from the list and choose OK or press *Enter*.
4. Choose Objects | Conclusion.
5. When the Tree Tool prompts you for the condition of Quantity, type `>=100` and press *Enter*.
6. For the conclusion of Discount, type `+50%` and choose OK.

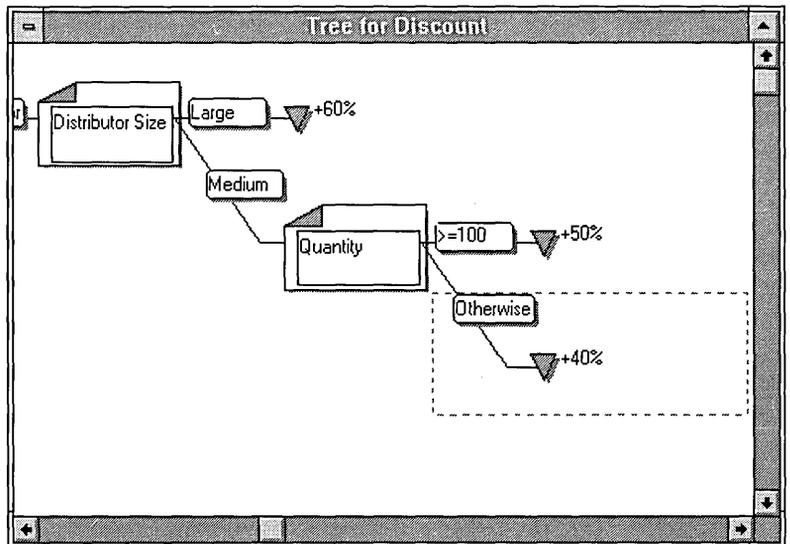
This means that, for all customers who are medium-size distributors and order at least 100 items, the value of Discount will be 50%.

Now add the third conclusion for this branch:

7. With the +50% conclusion node still selected, choose the Objects | Conclusion command.
8. When the Tree Tool prompts you for the condition of Quantity, type *Otherwise* and press *Enter*.
9. For the conclusion of Discount, type +40% and press *Enter*.

Now your decision tree should look like the one shown in the following figure. You'll finish the tree later in the tutorial.

Figure 6.3
Continued decision tree for
Discount



Scrolling a decision tree

As you add parts to a decision tree, the tree may become too big to see all at once (unless you shrink it with the View | Reduce command, explained later). You can easily display all parts of a tree by using the scroll bars at the sides and bottom of the Tree Tool screen. Press *Home* to return to the root node of the tree.

Before you go on to the next section, take a moment to scroll around the screen.

Adding a form for a new field

When you added the first nodes of the Discount decision tree, you added a new field, Distributor Size. When you add a field while creating a decision tree, you can put the field on a form later, using the Form Tool. If you don't put the field on a form, Object-Vision will use the Scratchpad form to prompt for the field's value when it is needed.

In this section, you will create a new form for the Distributor Size field and you will use the Stack Tool to see the order of the forms in your application.

Deciding design criteria

Whether you add a field to a form or not depends on the characteristics of your application. The main factors you should consider are

- Whether you have a place to add other fields if you are modeling your form on existing paper forms.
- Whether you have to change the field's properties. Fields on the Scratchpad form receive a set of default properties. If you need to change a field's properties, you *must* put it on a form.
- Whether the user will want to change the field's value (to make "what if" changes, for example). A user can see a field on a form but has to use the Field | Find command to get to a field not defined on a form.

In the case of the Distributor Size field, it doesn't make much difference whether you let ObjectVision use the Scratchpad form for Distributor Size or define a separate form for it (as in the original *Order* application). However, it's better not to have the field on the main form, Sales Order, because its value is not always required. (It's needed only when the value of Customer Type is *distributor*.)

For practice, you'll put the Distributor Size field on a new form.

Creating a new form

In addition to the Distributor Size field, your new form will also include the Name and Company fields already defined in the Sales Order form.

The completed form will look like this:

Figure 6.4
Completed Distributor
Information form

Distributor Information (Goal)	
Name	
Company	
Distributor Size	
<input type="checkbox"/> Large	
<input type="checkbox"/> Medium	

You will use these commands to create a new form:

- Form | New, to define the new form
- Objects | Field, to add fields to the new form
- Form | Select, to open the Sales Order form
- Edit | Copy and Edit | Paste, to copy a field from one form to another
- Properties | Field Type, to change the type of a field

 Your next steps:

1. Close the Tree Tool, if it's still open. Open the Form Tool, and choose Form | New. The Form Name dialog box appears.
2. For the form name, type *Distributor Information* and press *Enter*. The new form appears in the default form size.
3. Choose Objects | Field, select the Name field from the list, and then choose OK.
4. Position the Name field as shown in Figure 6.4. (You will resize it later.) Notice that if a field appears on multiple forms, as Name does, you can format it differently on each form if you want.

In some cases, it may be more convenient to copy a field from one form to another. Now you'll copy the Company field on the Sales Order form to the Distributor Information form. (You could easily have copied the Name field too, but it's good practice to try both methods.)

5. Choose Form | Select, select the Sales Order form, and choose OK. The Sales Order form displays in front of Distributor Information. When your application window is maximized, you can position the forms within it to make it easy to copy and paste between them.
6. Copy the Company field with the Edit | Copy command (or press *Ctrl+Ins*, the Edit | Copy shortcut key).

7. Select the Distributor Information form using Form | Select.
8. Paste the Company field into the form with the Edit | Paste command (or press *Shift+Ins*, the Edit | Paste shortcut key).
Notice that the Form Tool automatically expands the Distributor Information form so that the Company field fits.
9. Resize the Name field to match the Company field.
The next field you add, Distributor Size, doesn't exist yet on a form. You created it when you defined the Discount decision tree.
10. Choose Objects | Field, select the Distributor Size field, and then choose OK. Position and size the field as shown in Figure 6.4.
Now you need to format Distributor Size as a check box field.
11. Choose Properties | Field Type, select Check Boxes, and then choose OK.
12. The Check Boxes | Values of dialog box appears. Note that Automatic is checked, and the values *large* and *medium* appear in the Values list. When Automatic is checked, ObjectVision automatically determines value options from the conditions you define in the decision trees.
13. Choose OK to close the Check Boxes | Values of dialog box.
14. Close the Form Tool.
15. Close the Distributor Information form.
16. Choose File | Save to save your work.

Displaying the order of forms

Now you have two forms in your application, Sales Order and Distributor Information. ObjectVision maintains an application's forms in a specific order. This order determines which form is the Goal form when the user opens the application.

You can use the Stack Tool to display and rearrange the order of the forms in your application. By making a different form the top form, you can change which form is the Goal form when an application is loaded.

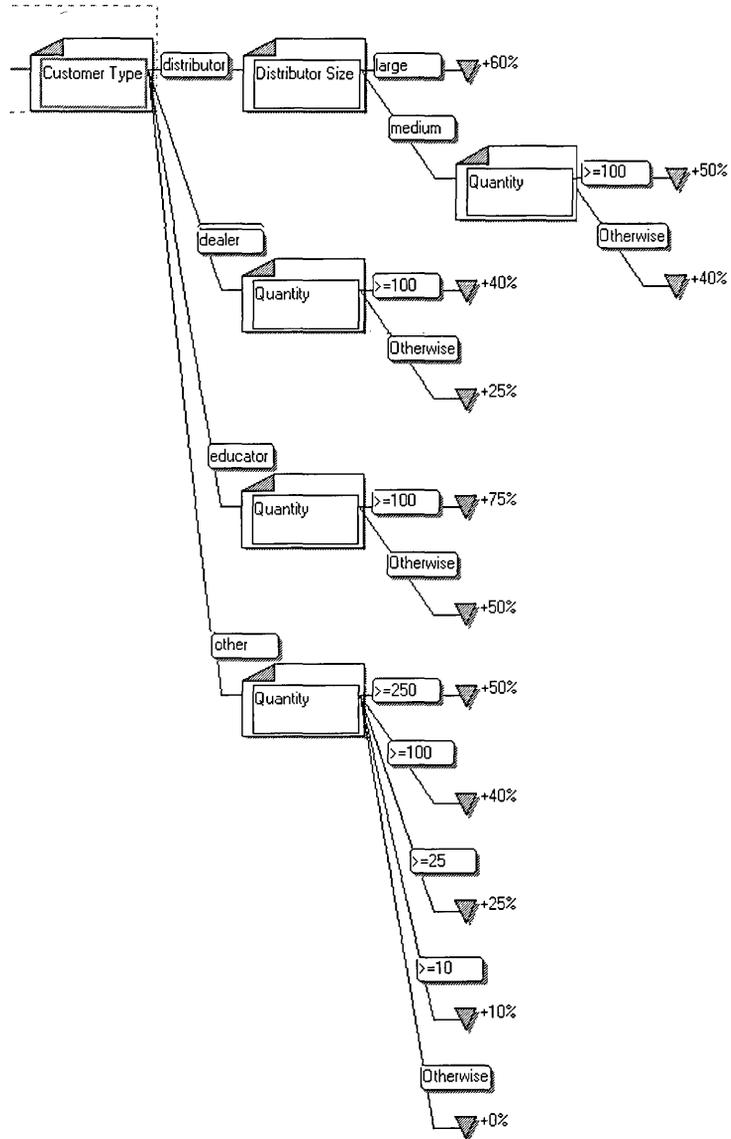
Copying and editing parts of a tree

In the next steps, you add more branches and conclusions to the Discount decision tree with the Edit | Copy and Edit | Paste commands. You can copy and paste parts of a decision tree and then modify them as necessary with the Properties | Conclusion, Objects | Branch, and Objects | Conclusion commands.

Figure 6.5 repeats the completed Discount decision tree first shown in Figure 6.1. Looking at Figure 6.5, you'll see that there are three tree sections below the Distributor Size node and each section begins with the Quantity branch node. Two of the tree sections end with two conclusions, and the bottom branch ends with five conclusions.

The structure of the sections is like that of the Quantity branch node and its conclusion nodes, which you have already created. As a shortcut, you will copy and paste sections of the tree. Then you will edit the conclusions and conditions in the copied sections to make them match the tree shown in the next figure.

Figure 6.5
Decision Tree for Discount



 Your next steps:

1. Choose Tools | Tree to open the Tree Tool.
2. Choose Tree | Select, select the Discount field, and then choose OK.
3. Maximize the window, and scroll it to the right. This lets you see the Quantity node and its conclusions.

4. Select the Quantity node (which will include the condition *medium*), and then choose Edit | Copy (or press *Ctrl+Ins*).
5. Select the Distributor Size node, and then choose Edit | Paste (or press *Shift+Ins*).
6. When the Tree Tool prompts you for the condition of Customer Type, type *dealer* and choose OK. This specifies the condition of Customer Type that applies to the new tree path you are pasting.

Now you will use the Properties | Conclusion command to change the conclusions you copied.

1. Select the first conclusion in the node you just pasted (+50%), and choose Properties | Conclusion. The Conclusion dialog box displays the expression.
2. Change the conclusion to +40% (remember that you can use editing keys, like *Backspace* to delete), and then press *Enter* or choose OK. The Tree Tool displays the updated expression.



In the next step, you use the mouse shortcut to open the Conclusion dialog box. You can double-click a conclusion node as an alternative to choosing Properties | Conclusion. Similarly, double-clicking a condition node opens the Condition dialog box.

3. Double-click the second conclusion in the node you just pasted (under the Otherwise condition).

Be sure to double-click directly on the conclusion node (the inverted triangle) or the conclusion expression. If you double-click the condition node (Otherwise in this case), the Condition dialog box appears instead of the Conclusion dialog box. The Conclusion dialog box appears.

4. Change the conclusion to +25% and choose OK or press *Enter*.
5. Select the branch you just edited (Quantity, with the condition *dealer*).
6. Choose Edit | Paste. Notice that you don't have to copy the node again; it's still on the Clipboard.
7. For the condition of Customer Type, type *educator* and press *Enter*.
8. Change the first conclusion of the node you just pasted to +75% and the second to +50%.

Next you will use the **Objects | Branch** and **Objects | Conclusion** commands to add the last branch node to the Discount tree. You could also copy and paste an existing node again and then edit it, but this node is quite a bit different from the ones you just added; its conditions are different and it contains more conclusions.

 *Your next steps:*

1. Select the branch you just edited (**Quantity**, with the condition *educator*).
2. Choose **Objects | Branch**.
3. For the condition of **Customer Type**, type *other* and press *Enter*.
4. For the field name, select **Quantity** from the list.
5. Choose **Objects | Conclusion**.
6. When the Tree Tool prompts you for the condition of **Quantity**, type ≥ 250 and press *Enter*.
7. For the conclusion for **Discount**, type $+50\%$ and press *Enter*.
8. Choose **Objects | Conclusion**.
9. When the Tree Tool prompts you for the condition of **Quantity**, type ≥ 100 and press *Enter*.
10. For the conclusion for **Discount**, type $+40\%$ and press *Enter*.

Now add the third conclusion:

1. Choose **Objects | Conclusion**.
2. When the Tree Tool prompts you for the condition of **Quantity**, type ≥ 25 and press *Enter*.
3. For the conclusion for **Discount**, type $+25\%$ and press *Enter*.

Now add the fourth conclusion:

1. Choose **Objects | Conclusion**.
2. When the Tree Tool prompts you for the condition of **Quantity**, type ≥ 10 and press *Enter*.
3. For the conclusion for **Discount**, type $+10\%$ and press *Enter*.

Now add the fifth and final conclusion:

1. Choose **Objects | Conclusion**.
2. When the Tree Tool prompts you for the condition of **Quantity**, type *otherwise* and press *Enter*.
3. For the conclusion for **Discount**, type $+0\%$ and press *Enter*.

Your decision tree should now match the one shown in Figure 6.5.

Testing the tree

To make sure your decision tree is constructed properly, you should test it:

- Close the Tree Tool.
- In form completion mode, supply different values in all three fields relating to the decision tree for Discount (Customer Type, Quantity, and Distributor Size) to test the possible conclusions of your new decision tree.
- Be sure to press *Enter*, not *Tab*, after you type values. The Distributor Information form won't appear if you press *Tab*.

Viewing and printing the tree

You can view decision trees at different resolutions. You can reduce a big tree to see its structure more clearly, or you can enlarge part of a tree to see it in detail. You can print a single decision tree or all the decision trees in an application.

Sizing a tree

The Discount decision tree is too big to fit entirely on the screen in normal display size. To see the whole structure of the tree, you can reduce it. You can also expand part of a tree.

 *Your next steps:*

1. To see the entire Discount tree onscreen, select the Discount field, if it isn't already selected, and choose Tools | Tree.
2. Choose the View | Reduce command three times.
Notice that no branch node names or conditions are displayed.
3. Now select the last Quantity node and choose the View | Expand command four times. The Tree Tool enlarges the tree, keeping the Quantity node at the center of the display.

Printing a tree

You can print a decision tree at any time while you are using the Tree Tool. The Tree Tool prints the tree at its current size. For example, if you have reduced the tree size with View | Reduce, the small version of the tree prints.

 *Your next steps:*

1. Before printing the Discount tree, choose View | Reduce to shrink the tree to its original size.
2. Choose Tree | Print to print the Discount tree.

Creating lists of values automatically

Earlier in the tutorial you learned how to manually supply a list of values for a check-box or selection-list field. You can also let ObjectVision determine a list of possible values automatically from your application's decision trees.

As you design applications, you might need to add, delete, or change values to refine your forms. Instead of adding a new value to the list of options manually, you can easily change a field's properties to have ObjectVision automatically determine its possible values.

When you check Automatic in the Values Of dialog box, ObjectVision finds all the conditions you defined for a field in all decision trees and generates a list of values. This automatically generated list of values is added to all forms that contain the field.

For example, if you add another value to the decision tree for Customer Type, you can have ObjectVision add it to the list of check box items. In the Form Tool you can select the Customer Type field, choose Properties | Field Type, select Check Boxes, and then choose OK. When the Values Of dialog box appears, check the Automatic box, then choose OK.

ObjectVision replaces the previous list of values for the field with the values it determines from the field's decision tree.

You can quit ObjectVision now, or you can continue. Be sure to save your work in either case.

Handling default and error values

This chapter discusses two decision tree features that can add flexibility to your application's logic:

- Using decision trees to provide default values for fields
- Using circular references in decision trees to let ObjectVision prompt for another value if a user enters an invalid value.

Providing default values

The decision tree you added for the Discount field contains many branch and conclusion nodes, providing a large number of possible values for the field. You can also add a decision tree to provide a default value for a field.

A default value is a convenience for the user. If a field typically has a particular value, you can have ObjectVision calculate that value for the field. In the few cases when the field requires a different value, the user can override the calculated value. A user can choose Field | Calculate to restore a field's calculated value.

In *Order's* Sales Order form, the decision tree for the Item field provides the default value *Widget*. In this chapter, you'll recreate the Item decision tree in *Myorder*.

 Your next steps:

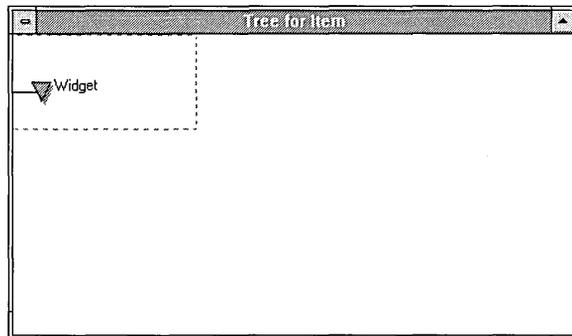
1. In form completion mode, select the Item field.
2. Choose Tools | Tree.

3. Choose Objects | Conclusion.

The Objects | Conclusion dialog box appears. Notice that you weren't prompted for a condition. Because you didn't add a branch node, no condition applies to the conclusion.

4. Type *Widget* and press *Enter*. The following figure shows how your screen should look now:

Figure 7.1
Item decision tree

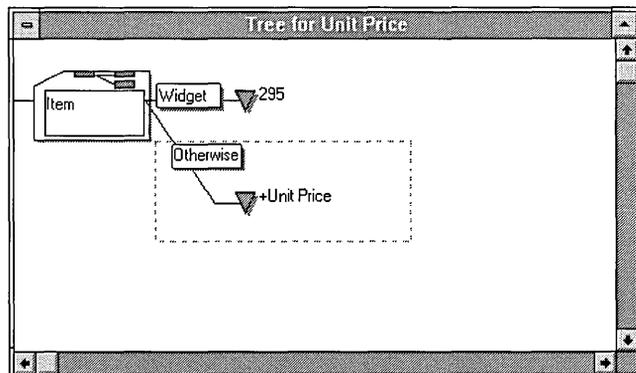


5. Choose Tree | Close Tool to return to form completion mode. Notice that the value *Widget* appears in the Item field.

Using circular logic in a decision tree

The decision tree for the Unit Price field uses circular logic—it refers to itself in one of its conclusion nodes. The following figure shows the completed decision tree for Unit Price:

Figure 7.2
Completed decision tree for
Unit Price



Circular logic can be very useful in a decision tree. It lets ObjectVision prompt the user for a field's value if other conclusions in the tree don't apply. The Unit Price tree, for example, lets ObjectVision calculate the value 295 if the value of the Item field is the default, Widget. If the value of Item isn't Widget, the Unit Price tree lets ObjectVision prompt the user for the Unit Price field.

Without circular logic, the Unit Price field couldn't have a decision tree, and the user would always have to enter the field's value—an inconvenience when the value is almost always the same.

Consider what would happen if Unit Price had a decision tree that didn't include the circular reference. As long as the Item field kept its default value, Widget, ObjectVision would be able to calculate a value for Unit Price. If the user overrode the Item value, however, ObjectVision would be unable to conclude a value for Unit Price; the error result NA (for Not Available) would appear in the field.

Adding the tree

Now you add the Unit Price decision tree.

 *Your next steps:*

1. In form completion mode, select the Unit Price field.
2. Choose Tools | Tree.
3. Choose Objects | Branch. The Field Name dialog box appears.
4. For the field name, select *Item* from the list and then choose OK.
5. Choose Objects | Conclusion.
6. In the Condition dialog box, type *Widget* and choose OK.
7. In the Conclusion dialog box, type 295 for Unit Price and choose OK.
8. Choose Objects | Conclusion.
9. In the Condition dialog box for Item, type *Otherwise* and choose OK.
10. In the Conclusion dialog box for Unit Price, type +Unit Price and choose OK. You could have used the Paste Field button instead to insert Unit Price after the plus sign.

Notice that you entered a plus sign as the first character in this conclusion. The other conclusion in this tree, 295, didn't start with a plus sign. The first conclusion value didn't require an

opening plus sign because 295 is already an expression (since it starts with a number). The plus sign must precede a value that ObjectVision is to evaluate, as it must evaluate the value of Unit Price. See Chapter 7 in *ObjectVision Reference* for more information.

11. Choose the Tree | Close Tool command to return to form completion mode. Notice that the value 295 is displayed in the Unit Price field. After you press *Enter*, the value \$295 is displayed in the Unit Price field (ObjectVision is displaying the value in the Currency format you assigned to the field.)

Now see what happens when you override the default value *Widget* in the Item field:

 *Your next steps:*

1. Select the Item field and type the value *Gadget*, then press *Enter*. Note that a dot pattern appears in the Item field to show that its calculated value is overridden. Now look at the Unit Price field. The value \$295 has disappeared.
2. Select the Unit Price field, type the value 750 and press *Enter*. The Unit Price field is not overridden because you haven't changed a calculated value.

You have finished this part of the tutorial. You can quit ObjectVision or continue the tutorial. Whichever you decide to do, be sure to save your work first.

Creating conditions and conclusions

In the decision trees you have added so far, you defined several conclusion nodes and specified conditions for those nodes. Each conclusion and each condition you defined was an expression.

In this chapter, you will add additional trees to become more familiar with expressions and the Condition and Conclusion dialog boxes.

You will perform these procedures:

- Create decision trees for several fields in the Sales Order form.
- Use the Condition and Conclusion dialog boxes.
- Use the Paste Field, Paste Function, and Paste Arguments options.
- Use the @MESSAGE and @BLANK functions in a decision tree for input validation.

Expressions

An expression defines a method for calculating a value based on the current situation. You have defined expressions in conditions and conclusions: ≥ 100 , 40%, and so on. To determine the value of an expression in the current situation, ObjectVision must evaluate it.

Expressions can contain elements of several different kinds. The decision trees you have added so far have included these elements:

- field names, like *Distributor Size* in the Discount decision tree
- constant values, like 100 and 25
- operators, like the comparison operator \geq and the percent operator %

Expressions can also contain @functions. ObjectVision provides a set of Quattro Pro-compatible @functions. @Functions usually perform an operation on a value or set of values. For instance, the @INT function truncates its argument.

Syntax The syntax for ObjectVision expressions is also Quattro Pro compatible. An expression can contain any combination of constant values, field names, functions, and legal operators. (The legal operators are listed in the next section.)

These syntax rules apply:

- If the first character is one of the following: ' , " , or ^ , the remaining characters are taken literally as a string value.
- If the first character is a digit or one of the following characters:
. + - (@ #
all of the conclusion or condition is evaluated as an expression.
- If a condition doesn't begin with a conditional operator, the equal operation (=) is assumed.
- A conclusion cannot begin with a conditional operator.
- Numeric constants can contain any combination of decimal digits without internal spaces.
- Text constants must be contained within two matching double quotation marks ("text") so that they can be distinguished from field names.
- You must use two double quotation marks to enclose a double quotation mark that is part of the text value. ("The word ""Text"" is embedded in this text.")
- Blanks are significant within both text constants and field names.
- You must enclose a field name within single quotation marks if the name meets any of these conditions:
 - It starts with any character other than a letter.

- Its last character is a blank.
- It contains any character that is not a letter, number, space, period, or underscore. For example, 'Make, Model' and 'Vacation remaining?'.
- It is one of the logical keywords Yes, No, True, or False.
- You must use two single-quotation marks to specify a single quote as part of the field name (you must also enclose the whole field in single-quotation marks because it contains a character that is not a letter, number, space, period, or underscore). For example, for a field named *Last Year's Tax*, you would have to type 'Last Year' 's Tax'. Note the single-quotation marks surrounding the field name and the double single-quotation marks between the *r* and the *s*.

Operators The following table lists the ObjectVision expression operators and the precedence assigned to each. Operators with the highest precedence (7) are performed first.

Table 8.1
Expression operators

Operator	Operation	Precedence
&	String concatenation	1
=	Equal	2
<>	Not equal	2
<	Less than	2
>	Greater than	2
<=	Less than or equal	2
>=	Greater than or equal	2
+	Addition	3
-	Subtraction	3
*	Multiplication	4
/	Division	4
-	Negative	5
+	Positive	5
^	To the power of (exponential)	6
%	Percent	7

Expression dialog boxes

When you defined decision trees earlier, you used ObjectVision's Condition and Conclusion dialog boxes.

You can enter up to 4,096 characters in each dialog box, using cut and paste and the other standard editing functions. You can also paste field names and function names into expressions.

When you enter an expression in the Conclusion dialog box, ObjectVision checks the expression to make sure it's valid. If it contains an error, ObjectVision tells you the type of the first error.

Adding a decision tree for the Amount field

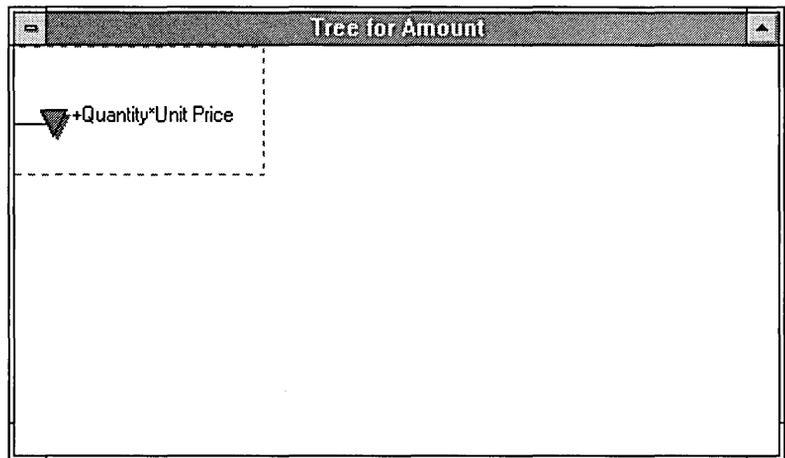
In the next procedure, you add a decision tree for the Amount field. The next sections define the logic of the tree and discuss a powerful dialog box feature, pasting field names.

Logic

The logic for calculating the Amount field is simple: The amount (base cost) of an order is the number of widgets ordered times the unit price for a widget.

The completed decision tree for the Amount field is shown here.

Figure 8.1
Completed Decision tree for
Amount



The tree has one conclusion node and no branch nodes (and therefore, no conditions). The conclusion for Amount is an expression. It starts with a plus sign (otherwise it would be interpreted as a string constant since it begins with *Q*), and it contains two field names, Quantity and Unit Price, and the multiplication operator, *.

ObjectVision evaluates the conclusion node for Amount by multiplying the current value of the Quantity field by the current value of the Unit Price field.

Pasting field names

The Condition and Conclusion dialog boxes include options for pasting field names and function names as you enter an expression. Pasting field names prevents you from accidentally creating a new field with a typographical error. If you mistype a field name, ObjectVision assumes that you want to create a new field.

Adding the tree

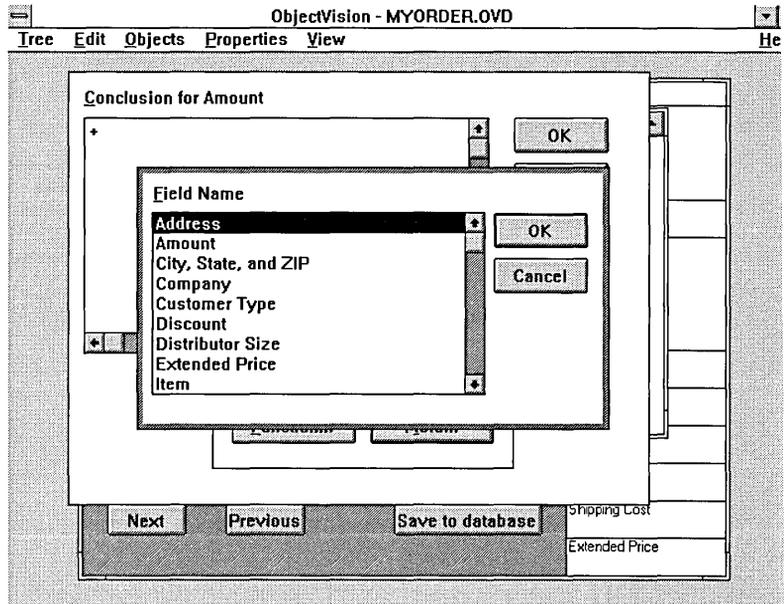
The structure of the sections is like that of the Quantity branch node and its conclusion nodes, which you have already created.

Now you'll create the Amount decision tree.

 Your next steps:

1. Choose Tools | Tree to open the Tree Tool.
2. Choose Tree | Select, select the Amount field, and then choose OK.
3. Choose Objects | Conclusion. The Objects | Conclusion dialog box appears.
4. Type + (the plus sign) but don't press *Enter* or choose OK.
5. Choose the Paste Field button. The Tree Tool lists the fields you have defined. The following figure shows how your screen should look now:

Figure 8.2
Paste Field dialog box



6. Select Quantity by scrolling, or typing Q to get to the field. Once you have selected Quantity, choose OK or press *Enter*. The word *Quantity* is pasted into the expression you are defining.
7. Type * (an asterisk, the multiplication operator).
You can type the operator right after the field name or you can put a space before or after the operator (but remember that you can't put a space before the plus sign that starts the expression).
8. Choose the Paste Field button again, select Unit Price, and then choose OK. The words *Unit Price* are pasted into the expression.
9. Press *Enter* or choose OK to close the Objects | Conclusion dialog box.
Now the Amount decision tree should look like the one shown in Figure 8.1 on page 94. If you typed any blank spaces before or after the operator, notice that they are now stripped out.

Testing the tree

Now test the Amount decision tree:

- Close the Tree Tool.
- In form completion mode, test the tree by entering a range of values in the Quantity field and verifying that the Amount field displays the correct value in each case.

Adding more decision trees

In the rest of this chapter, you will add decision trees for the Total Price, Extended Price, and Shipping Cost fields.

Total Price and Extended Price

The decision trees for the Total Price and Extended Price fields are very similar to the Amount decision tree. Each contains one conclusion node that contains field names and one or more operators.

The Total Price decision tree expresses this logic: The total price of the order equals the amount minus the discount. The tree's conclusion node contains this expression:

+Amount-Less Discount

The Extended Price decision tree expresses this logic: The extended (overall) price of the order equals the total price plus the shipping cost and the sales tax. The tree's conclusion node contains this expression:

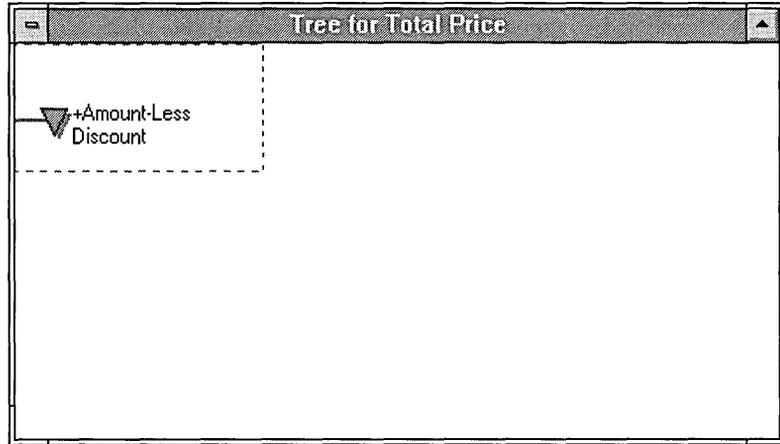
+Total Price+Shipping Cost+Sales Tax

Now you will add decision trees for the Total Price and Extended Price fields.

 Your next steps:

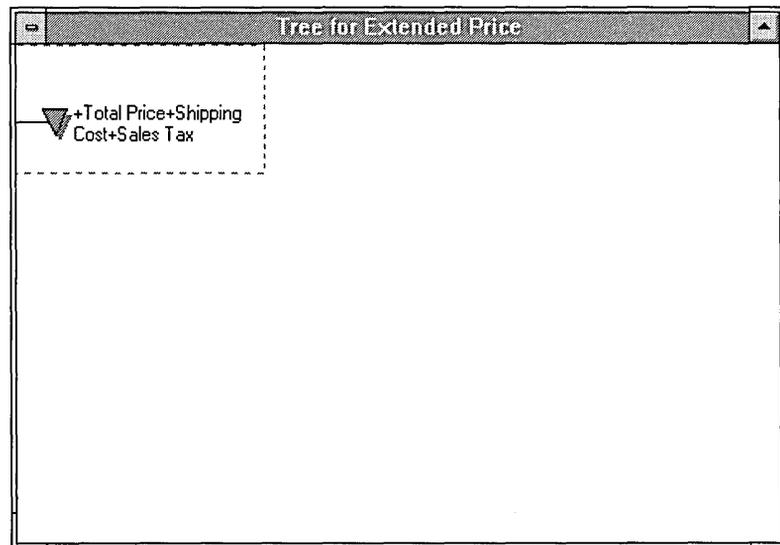
1. Follow the same procedure you used for the Amount decision tree to add the Total Price decision tree, using the Paste Field function in the Objects | Conclusion dialog box for the field names. The following figure shows the completed Total Price decision tree:

Figure 8.3
Completed decision tree for
Total Price



2. Next add the Extended Price decision tree, using the same method. The following figure shows the completed Extended Price decision tree:

Figure 8.4
Decision tree for Extended
Price

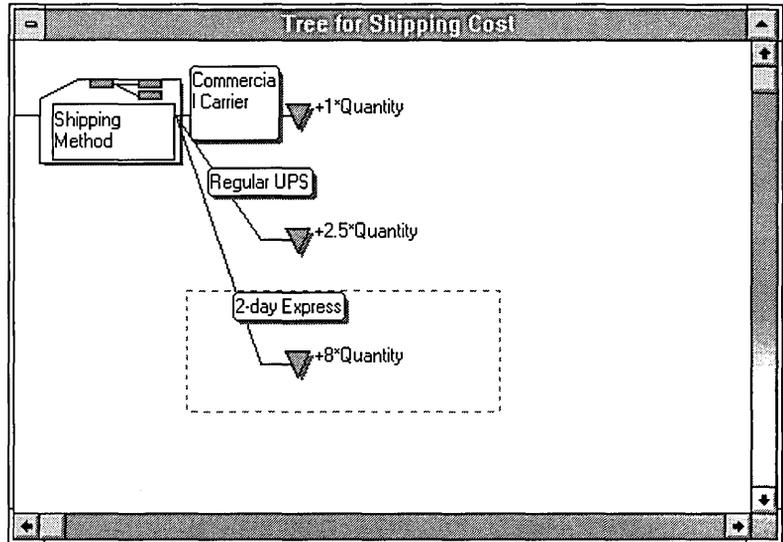


Note that you can't yet test the Total Price and Extended Price trees. Both trees require the value of the Less Discount field, but you haven't added a decision tree for Less Discount yet. You can't type in a value for the Less Discount field, either, because it is protected from being overridden.

Shipping Cost tree

The decision tree for the Shipping Cost field is more complex than the Amount decision tree. It includes a branch node and three conditions, each with corresponding conclusion nodes. This figure shows the completed tree:

Figure 8.5
Completed decision tree for
Shipping Cost



The Shipping Cost decision tree expresses this logic:

- If Shipping Method is Commercial Carrier, then Shipping Cost is Quantity times \$1.00.
- If Shipping Method is Regular UPS, then Shipping Cost is Quantity times \$2.50.
- If Shipping Method is 2-day Express, then Shipping Cost is Quantity times \$8.00.

Notice that the condition expressions are *constants*. They consist of single values that don't change: Commercial Carrier, Regular UPS, and 2-day Express. The conclusions are all expressions that include a field name, a numeric value, and an operator.

Because the Shipping Cost decision tree has both conditions and conclusions, you will use the Object | Condition dialog box and the Object | Conclusion dialog box to add the tree.

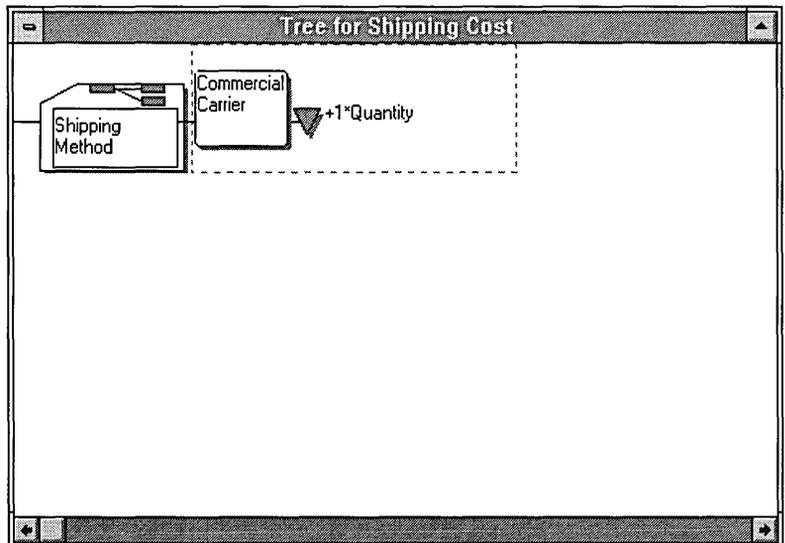
Now you will add the Shipping Cost decision tree. You begin by defining the branch node and the first conclusion node.

 Your next steps:

1. If it is not already open, open the Tree Tool with the Tools | Tree command.
2. With the Tree | Select command, select the Shipping Cost field, then choose OK.
3. Choose the Objects | Branch command and paste Shipping Method from the list of field names, then choose OK.
4. Choose the Objects | Conclusion command. The Object | Conclusion dialog box appears.
5. For the condition, type Commercial Carrier and choose OK. The Object | Conclusion dialog box appears.
6. Type +1* and then, using the Paste Field function, paste the field name *Quantity* into the expression. Note that the initial plus sign is optional because with 1 as the first character, ObjectVision evaluates the expression as a math operation.
7. Press *Enter* or choose OK.

You have added the branch node and the first conclusion node. This is how your screen should look now:

Figure 8.6
First nodes in Shipping Cost
decision tree



You'll add the other two conclusion nodes by copying, pasting, and editing the first one.

 Your next steps:

1. With the conclusion node still selected, choose **Edit | Copy**.
2. Choose **Edit | Paste**.
3. For the condition of **Shipping Method**, type `Regular UPS` and press *Enter*. The new condition and the copy of the first conclusion node are pasted into the tree.
4. Choose **Properties | Conclusion**.
5. Delete the `1` in the expression (leaving the `+`) and type `2.5` in its place. Then press *Enter* or choose **OK**.
6. Choose **Edit | Paste** again.
7. For the condition of **Shipping Method**, type `'2-day Express` and press *Enter*. Be sure to precede this condition with an apostrophe, or a new field is created and the decision tree will not work properly.

The new condition and another copy of the first conclusion node are pasted into the tree, below the second conclusion node.

8. Choose **Properties | Conclusion**, change the `1` in the expression to `8`, and then choose **OK**.

The **Shipping Cost** decision tree is complete. The tree you added should match the one shown in Figure 8.5.

Close the **Tree Tool**, and save your work. Test your tree by varying the value of **Quantity** and checking what value is assigned to the **Shipping Method** field.

Using @functions in expressions

You can use @functions in expressions to perform operations on one or more values. For example, @functions can determine the length of field values, find their type, or update their corresponding database fields. The @functions fall into these categories:

- Link functions
- Mathematical functions
- String functions
- Logical functions

- Date and time functions
- Miscellaneous functions

Chapter 8 of *ObjectVision Reference* discusses using these functions in detail.

In this part of the tutorial, you will add decision trees that use the @NOW, @INT, and @ROUND functions.

Syntax

All functions have the same basic syntax:

function-name (argument1, argument2, ...)

Each function has a name that helps identify the operation that it performs. You can specify the function name in uppercase, lowercase, or mixed case. You can either type a function name into an expression or paste it with the Paste Function option.

Each function has a specific number of *arguments*. An argument is information required by the function. Depending on the function, the number of arguments varies from 0 to 14.

You include the arguments for the function inside parentheses and separate them with the list separator character named in the International portion of the Control Panel (usually a comma).

You can put spaces before and after any argument. If no arguments are required, the parentheses are optional.

Each argument to a function can be a constant value, a field name, an expression, or another function. If any of the function arguments are expressions or functions, ObjectVision evaluates those arguments first and passes their values on to the main (outermost) function.

When necessary, ObjectVision tries to convert arguments to the data type that the function requires.

@NOW, @INT, and @ROUND

Expressions in the Order Date, Less Discount, and Sales Tax decision trees use the @NOW, @INT, and @ROUND functions.

The @NOW function returns the serial number of the date and time currently in the clock of the computer on which ObjectVision is running. The serial number has an integer part and a fractional part:

- The integer represents dates from January 1, 1800, up to and including December 31, 2099. Its value can range from -36,522 to 73,050 and is computed as the number of days before or after December 30, 1899 (which is represented as 0).
- The fractional part of the serial number represents the time from 12:00 midnight to 11:59:59 p.m. and is computed as a fraction of a 24-hour day.

The @NOW function changes values only when ObjectVision evaluates it (when the application is opened or when ObjectVision calculates the value of a field that refers to @NOW in its decision tree, for example).

@INT truncates its argument to an integer by removing any fractional part of the argument. @ROUND rounds its first argument to the number of digits specified by its second argument.

Decision tree for Order Date

The decision tree for the Order Date field consists of one conclusion node that contains this expression:

`@INT(@NOW)`

This expression begins with a function name, @INT with a left parenthesis immediately following. @NOW is another function, and it's used here as an argument to @INT. @NOW is immediately followed by a right parenthesis because @NOW has no arguments. The final right parenthesis ends @INT's argument list.

This expression tells ObjectVision to get the current date and time from the computer clock in the form of a serial number and then truncate the serial number, making it an integer. Because the fractional portion of the serial number, representing the time, is discarded, the formula evaluates to the current date, which ObjectVision displays in the Order Date field.

The Condition and Conclusion dialog boxes have a Paste Function button similar to the Paste Field button that you have already used. When you choose Paste Function, you select the function you want from a list. You can also select a Paste Arguments option to have a template of the function's argument list pasted into the expression with it.

 *Your next steps:*

1. Select Order Date, then choose Tools | Tree.
2. Choose Objects | Conclusion.

3. Choose the Paste Function button and choose INT from the list that appears in the Paste Function dialog box. The expression in the Objects | Conclusion dialog box looks like this:

`@INT()`

Notice that the pointer is positioned so that the next thing you paste or type will go between the parentheses.

4. Choose the Paste Function button again and choose NOW. Now the expression is complete:

`@INT(@NOW)`

5. Press *Enter* or choose OK to close the dialog box.
6. Close the Tree Tool. Notice that today's date is displayed in the Order Date field.

The value of the Order Date field won't change until ObjectVision evaluates it again. If you were to work on the application past midnight without causing ObjectVision to evaluate @NOW (by reloading the application or by using Field | Calculate on the Order Date field, for example), you would see that the Order Date field still contains the value it has now.

Decision tree for Less Discount

The decision tree for the Less Discount field consists of one conclusion node that contains this expression:

`@ROUND(Amount*Discount,2)`

The @ROUND function uses this syntax:

`@ROUND(X,Num)`

Where *X* is the numeric expression to be rounded, and *Num* tells how *X* should be rounded. If *Num* is greater than 0, *X* is rounded to *Num* digits to the right of the decimal point. (Chapter 8 in *ObjectVision Reference* explains @ROUND in detail.)

In the Less Discount expression, *X* is the value that results from multiplying the value of the Amount field by the value of the Discount field. *Num* is 2.

This expression tells ObjectVision to calculate the value of the Less Discount field by multiplying the Amount value by the Discount value and then rounding the result to two decimal places.

You will use the Paste Function, Paste Arguments, and Paste Field options when you add the Less Discount decision tree.

 Your next steps:

1. Open the Tree Tool and select the Less Discount field.
2. Choose Objects | Conclusion.
3. Choose the Paste Function button and highlight ROUND which is listed in the Paste Function dialog box that appears. (But don't press *Enter* or choose OK yet).
4. Check the Paste Arguments box.
5. Choose OK. The @ROUND function and its argument names are pasted into the expression. The expression in the dialog box should now look like this:

@ROUND(*X,Num*)

6. Choose Paste Field, select Amount, and choose OK. The field name *Amount* replaces the highlighted *X*.
7. Type the multiplication operator (*).
8. Choose Paste Field again, select Discount, and choose OK.
9. Delete the word *Num* and replace it with 2. Now the expression is complete:

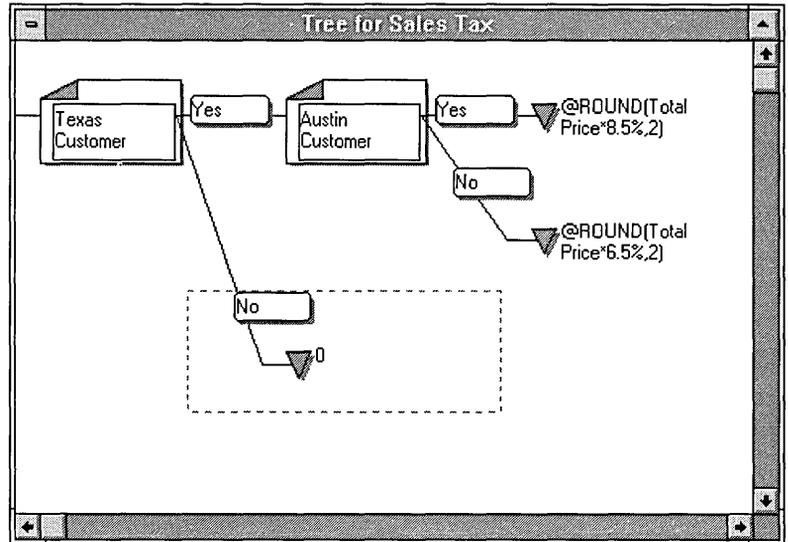
@ROUND(*Amount*Discount,2*)

10. Press *Enter* or choose OK to close the Objects | Conclusion dialog box.
11. Close the Tree Tool by choosing Tree | Close Tool.
12. Test the tree by varying the values of the Quantity and Discount fields. At this point, you can also test the Total Price tree. You will be able to test the Extended Price tree after you've added the tree for Sales Tax.

Decision tree for Sales Tax

The following figure shows the Sales Tax decision tree, which is more complicated than the Order Date or Less Discount tree.

Figure 8.7
Decision tree for Sales Tax



Notice that this tree has two branch nodes and three conclusion nodes, and it also contains two new fields, Texas Customer and Austin Customer. You add these fields when you define the branch nodes. ObjectVision will use the Scratchpad form to prompt for those fields, because they don't appear on any form.

 Your next steps:

1. Choose Tools | Tree, select the Sales Tax field, then choose OK.
2. Choose Objects | Branch.
3. Select <Add New Field> from the list that appears, then choose OK.
4. For the name of the field, type Texas Customer and choose OK.
5. Choose Objects | Branch again.
6. For the condition of Texas Customer, type Yes and choose OK.
7. For the branch name, add another new field, Austin Customer, then choose OK.
8. Choose Objects | Conclusion.

9. For the condition of Austin Customer, type *Yes* and press *Enter*.
10. For the conclusion of Sales Tax, type this expression:
`@ROUND (Total Price*8.5%,2)`
11. Choose OK.

Now add a second conclusion:

 *Your next steps:*

1. Choose Objects | Conclusion.
2. For the condition of Austin Customer, type *No* and press *Enter*.
3. For the conclusion of Sales Tax, type this expression:
`@ROUND (Total Price*6.5%,2)`
4. Choose OK.

You could also have added the second conclusion by copying the first one, pasting it, and then editing it.

Now you'll add one more conclusion to the tree:

 *Your next steps:*

1. Select the Austin Customer branch.
2. Choose Objects | Conclusion.
3. For the condition of Texas Customer, type *No* and choose OK.
4. For the conclusion for Sales Tax, type 0 (zero) and choose OK.
5. Close the Tree Tool and test the Sales Tax tree by varying the values of the Texas Customer, Austin Customer, and Total Price fields. You use the Field | Find command to select the Texas Customer and Austin Customer fields.

Input validation

You can use the @MESSAGE and @BLANK functions together for validation on fields that require user input. You construct a decision tree so that the @MESSAGE function displays a message when the user enters an invalid value (which the decision tree tests for), and the @BLANK function removes the value the user entered so the user can enter a new one.

In this part of the tutorial, you create a decision tree for the Quantity field. The tree logic tells ObjectVision to display an explanatory message and remove the value entered in the field if the value is 0 or a negative number.

 Your next steps:

1. In the *Myorder* application, open the Tree Tool, select the Quantity field, then choose OK.
2. Choose Objects | Branch, select Quantity as the branch node, then choose OK.
3. Choose Objects | Conclusion.
4. For the condition of Quantity, type ≤ 0 and press *Enter*.
5. For the conclusion of Quantity, type the value shown below. Note that you type a space before the word *is*. Note also that the text for @MESSAGE *must be on a single line*, or you get an error message (don't press *Ctrl+Enter* to start a new line until you're ready to type @BLANK).

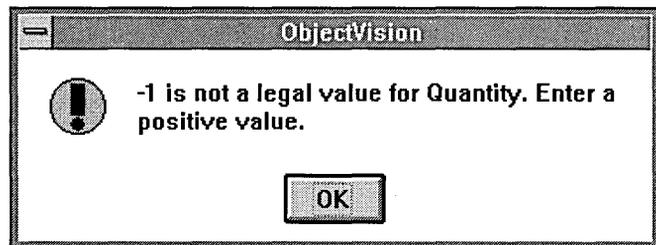
```
@MESSAGE (Quantity&" is not a legal value for Quantity.  
Enter a positive value.")
```

```
@BLANK
```

6. Choose OK.

The & operator in the first conclusion performs text concatenation. During form completion, ObjectVision displays the current value of the Quantity field (the invalid value that the user just entered) and then the rest of the message as shown in the next figure.

Figure 8.8
The error message for illegal
values



7. Insert another conclusion, with the condition *Otherwise* and the conclusion $+Quantity$.
8. Return to form completion mode and test the decision tree by entering 0 or a negative value in the Quantity field.
9. To continue, choose the OK button in the message dialog box. The @BLANK function retracts the negative value from the Quantity field, and ObjectVision highlights it for input. You can now specify a new value in the field.

Evaluating the Quantity tree

This sequence occurs when ObjectVision needs a value for the Quantity field:

1. The field has a decision tree associated with it, so ObjectVision attempts to evaluate the tree. Because the first node is a branch node with the value Quantity, ObjectVision prompts the user to specify a value for Quantity.
2. When the user types a value for Quantity, ObjectVision checks the first condition in the tree.
3. If the user typed a value greater than zero, the first condition is not true, and ObjectVision evaluates the second condition, Otherwise. Otherwise is always true, so ObjectVision evaluates the conclusion, +Quantity, setting the value of the field to the value the user entered.

If the user typed a value less than or equal to zero, the first condition is true, and ObjectVision evaluates the first two conclusions. First it evaluates @MESSAGE and displays the text value you specified, and then it evaluates @BLANK and retracts the value that the user entered in the Quantity field. Now the field has no value, and ObjectVision goes through the tree again, starting at step 1 (and thus prompting the user for the value again).

You have finished this part of the tutorial. If you decide to quit now, be sure to save your form first.

Linking to databases

Your ObjectVision applications can use external links to read information from data files created by other programs. They can also update the data from another program with information entered in a form.

In this chapter, you complete the application you've been working on. You'll link the application to a Paradox database file and assign the correct functions to the button fields at the bottom of the Order form.

You will complete the following procedures:

- Create a new Paradox table and link to it using the Tools | Links dialog box. At the same time, you'll automatically create read and write links between the new Paradox table and the ObjectVision application fields.
- Add decision trees to the button fields.

External links

ObjectVision uses external links to store and retrieve information that you enter in forms. ObjectVision reads and writes data to and from ASCII files, Paradox tables, dBASE-compatible files, and Btrieve files. Additionally, ObjectVision supports the Windows Dynamic Data Exchange (DDE) protocol. Using external links you can easily use another program to process information that you entered with your ObjectVision application.

You can create several links in a single application to transfer information between a variety of sources and destinations. The value from a single field can update several different data files.

You can use an external link to give a field a value without using a decision tree or requiring the user to enter the data. When existing data is linked to your application, you free the user from retyping information.

There are two ways to create links to your ObjectVision applications: You can use @functions or the Tools | Links dialog box. Both ways of creating links are equivalent, but sometimes one is more appropriate than the other. The Tools | Links dialog box is usually easier because there is no need to remember the necessary syntax for @function statements. @Functions are more advanced and are necessary only in more complex links. In this chapter you will use the Tools | Links dialog box only.

See *ObjectVision Reference* for complete explanations of the five types of external links.

Creating a Paradox table

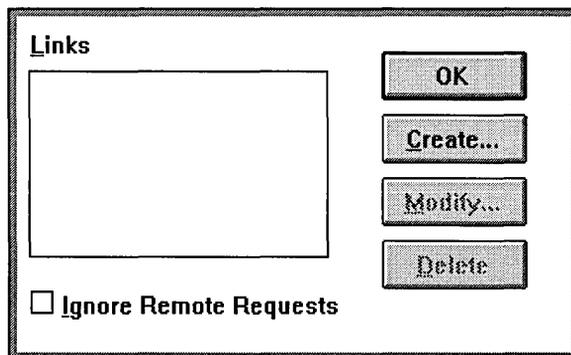
With the Links dialog box, you can create a new Paradox table from within ObjectVision; there's no need to have Paradox.

In the following steps, you'll link your application to a new Paradox table. If you have quit ObjectVision, restart the program and open the *Myorder* application.

 *Your next steps:*

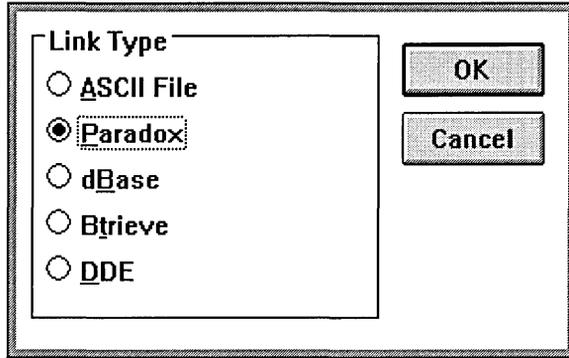
1. Choose Tools | Links. The dialog box shown here appears:

Figure 9.1
Tools | Links dialog box



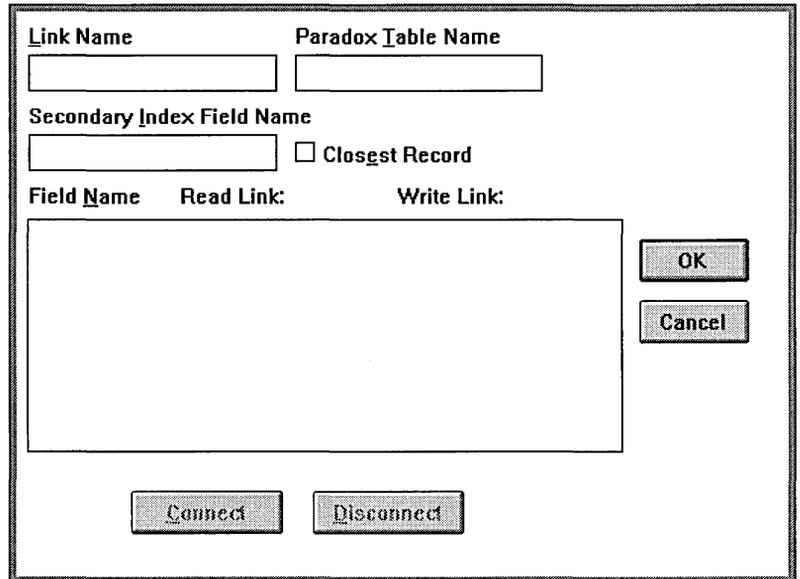
2. In the Links dialog box, choose Create, and then select Paradox. Your screen now looks like this:

Figure 9.2
The Link Type dialog box



3. Choose OK. The Link Type | Paradox dialog box appears:

Figure 9.3
The Paradox Links dialog box



You are going to create a link to a new Paradox table to store the values entered into the Order form. Since each application can contain several links, each link must be named.

4. Type Paradox Link in the Link Name field, then press *Tab*.
5. Type Myorder in the Paradox Table Name field, then press *Tab*.

It isn't necessary to enter the Paradox .DB extension, since ObjectVision already knows that it is looking for a Paradox table.

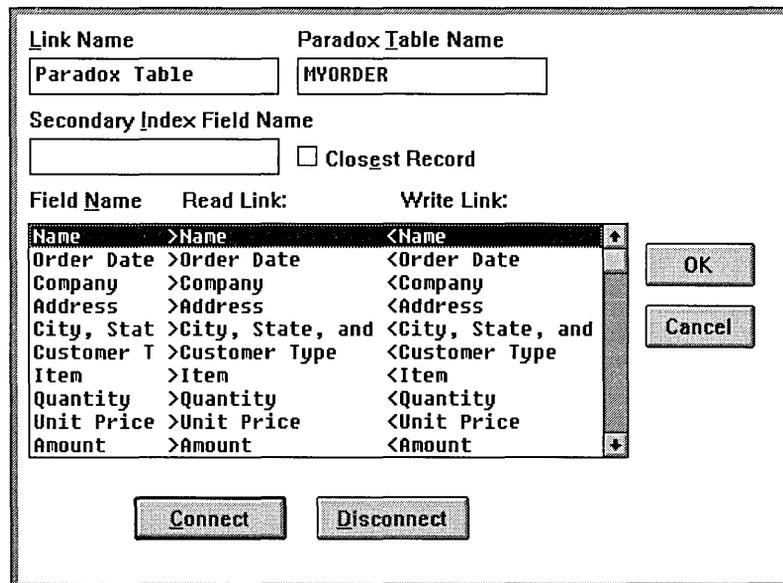
ObjectVision tries to find the MYORDER.DB table in the current directory. Since there isn't one already created, ObjectVision asks you if you want to create the table or cancel the operation.

Note: If someone else has already done this lesson, ObjectVision may find a MYORDER.DB file in the current directory and fill in the Field Name column with the values from the existing file. If this occurs, enter a different Paradox table name (*ORDER1.DB* or *ORDER2.DB*, for example) to continue the lesson.

6. Choose OK to create the new MYORDER.DB table.

ObjectVision creates a new Paradox table using the active form (the Order form) as the template. The fields appear in the Field Name column in the same order as they are arranged on the Order form. The Paradox Links dialog box should look like this:

Figure 9.4
The Paradox Links dialog box
with the linked field names



If any of the ObjectVision field names were too long, or contained characters that Paradox doesn't accept in a valid field name, then the name would be truncated.

ObjectVision has created a Paradox table with the same field names as the form. The type and size of the Paradox fields are taken from the properties of the fields in ObjectVision. ObjectVision also created a primary index for the new Paradox table, using the first field in the form, Name.

In addition to the Field Name column, the ObjectVision field names appear in the Read and Write link columns. This means that every field in the ObjectVision form is linked for both reading and writing to the field with the same name in the newly created Paradox table. ObjectVision automatically created the links when it created the Paradox table. You can use the Connect and Disconnect buttons to disable reading or writing of individual fields, but in this example you leave them all the way they are.

7. Choose OK to save the link.

Because this is a new link, a dialog box appears that asks you if you want to add the appropriate buttons for the new Paradox link. This function can add Next, Previous, Clear, Delete, and Enter button fields to your form. In this lesson, however, you will create these buttons manually so that you can learn how they work.

8. Choose No, and then OK to leave the Tools | Links dialog box.

Your application is now linked to the Paradox table. In the next section, you will create decision trees for the button fields on the Sales Order form so that you can store and retrieve data using the link you've created.

Using links with button fields and decision trees

In this section, you add decision trees to the Save to Database, Next, and Previous button fields. These decision trees will contain simple @functions that control external links. They do not change the values of any of the fields on the form.

 *Your next steps:*

1. Choose Tools | Tree.

2. Choose Tree | Select, select the Save to Database field, then press *Enter*.

The decision tree for Save to Database appears, displaying an empty branch node.

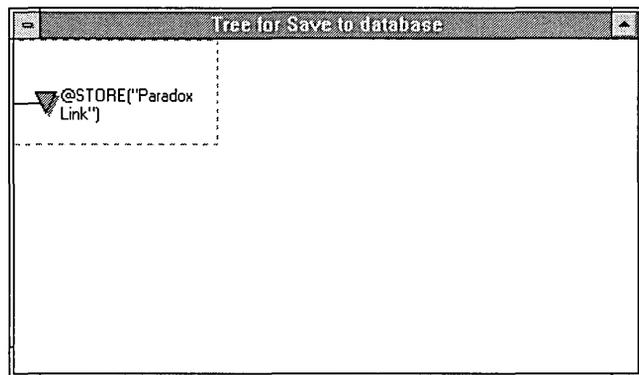
3. Choose Objects | Conclusion.

4. When the Conclusion dialog box appears, choose Paste Functions. Choose STORE from the list of functions that appears.
5. Type "Paradox Link" inside the parentheses. The expression should look like this:

```
@STORE("Paradox Link")
```
6. Press *Enter* to save the conclusion.

When you press this button field in form completion mode, the @STORE function stores the values in the current form as specified by the link name that you entered as the argument.

Figure 9.5
Decision tree for Save to database



Now you will repeat this procedure and use @NEXT and @PREVIOUS to create the decision trees for the Next and Previous button fields.

7. Choose Tree | Select to open the decision tree for Next, and add the following conclusion node:

```
@NEXT("Paradox Link")
```

The @NEXT function causes ObjectVision to clear the data from all the forms in the application and read in the information from the next record in the database. It imports the information according to the link that you enter as the argument.

8. Choose Tree | Select to open the decision tree for the Previous field, and add the following conclusion node:

```
@PREVIOUS("Paradox Link")
```

The @PREVIOUS function causes ObjectVision to clear the data from all the forms in the application and read in the data from the previous record in the database.

Now that you have added the link functions, the application is complete. The next chapter, which is optional, leads you through the application you have created from a user's perspective. It may help you understand the importance of what you have learned so far.

If you exit the program now, be sure to save your work.

Entering data with your application

This final chapter of the tutorial shows how your application works from a user's perspective. It describes the basic data-entry features and leads you through the completion of the forms in *Myorder*. In the process, it explains various elements of the form design.

This chapter describes the following features:

- Navigating through forms.
- Basic data entry.
- Making “what if” comparisons.
- Finding existing data.

Clearing data

To understand how your application works from the very beginning, you should clear out any data that you might have entered already.

 *Your next steps:*

1. Make sure that you are in form completion mode (the form status is [Goal]).
2. Choose Edit | Clear All to clear values from all the forms in the application. Note that calculated fields such as Order Date, Item, and Unit Price still display a value which is determined by the corresponding decision tree.

The Sales Order form now appears as a user that just opened the application would see it. Since the Sales Order form is the form at the top of the form stack, it appears as the Goal form. ObjectVision determines which fields on the Goal form the user needs to complete.

Form navigation

ObjectVision uses guided completion during form completion. This means that ObjectVision determines the field sequence needed to complete the application, prompting only for fields requiring values. It skips over fields if their values are calculated by decision trees. If necessary, ObjectVision will temporarily display a new form to get information it needs to calculate a value on the Goal form.

Guided completion ensures that users fill in every required field on the form. If you don't want to follow the usual field sequence, you can fill fields in any order. File | Resume returns a user to the last empty field on the original Goal form so that the rest of the form can be completed in the usual order.

Here is a list of important keystrokes and commands to use for form navigation:

- Press *Enter* to use guided completion and move to the next field that requires a value.
- Press *Tab* to move to next field on the form, whether it requires a value or not. You can also highlight any field by clicking it with the mouse. Both of these methods interrupt guided completion.
- If you interrupt guided completion of fields, use File | Resume to return to the field sequence determined by ObjectVision.
- Choose Form | Select to select a different form in the application.
- Choose Field | Find to find any field in the application.

Completing the form

In this part of the tutorial, you begin filling out the Sales Order form by taking a telephone order from a customer.

Taking an order

Now you're ready to start completing the Sales Order form. Pat Myers of Stevens, Inc., a large distributor of Widgets in Austin, Texas, is placing their first order with your company. Assume that the customer gives you the information for each field. Notice that the Name field, in the upper left portion of the form, is highlighted, showing that it's ready for you to enter a value.

Correcting text Before you start, a few tips about correcting text:

- If you make a mistake before going to a new field, you can backspace over it. You can also use the arrow keys or mouse to position the pointer and then insert or delete text. Pressing *Home* moves the pointer to the beginning of the text you are entering, and *End* moves the pointer to the end.
- If you want to completely remove what you typed, restoring the field to the state it was in when you started, and you haven't yet moved to another field, press *Esc*.
- You can also use Edit | Undo to undo the last edit that you made.
- If you notice a mistake in a field after moving to another field, you can click the field to select it and then correct your entry.
- If you want to correct an error in the previous field you just left and haven't started typing yet, pressing *Alt+Backspace* selects the previous field and clears its value.

 Your next steps:

1. Type Pat Myers in the Name field and press *Enter*.
ObjectVision highlights the Company field. Notice that ObjectVision's guided completion automatically skips the Order Date field and selects the next field that requires a value from you.
2. Type Stevens, Inc. and press *Enter*.
3. In the next highlighted field, Address, type 1214 19th St. and press *Enter*.
4. In the next highlighted field, City, State and ZIP, type Austin, TX 78751 and press *Enter*.

The next highlighted field is Customer Type. Remember that in Chapter 4 you defined help text for the Customer Type field. If you want to see it, press *F1* while the field is still highlighted.

5. Press *Enter* to check Distributor and move to the next field. The X in the box shows that you selected that value.
6. The next highlighted field is Quantity, another text/numeric field. Type 50. Your screen should now look like this:

Figure 10.1
Completing the Quantity
field

Sales Order (Goal)			
ACME Widgets		4030 Braker Lane West Suite 2001 Austin, TX 78759-5332	
Name Pat Myers		Order Date 12/10/90	
Company Stevens, Inc.		Customer Type <input checked="" type="checkbox"/> distributor <input type="checkbox"/> dealer <input type="checkbox"/> educator <input type="checkbox"/> other	
Address 1214 19th St.			
City, State, and ZIP Austin, TX 78751			
Item Widget	Quantity 50	Unit Price \$295.00	Amount
Shipping Method	Discount		Less Discount
<div style="text-align: center;"> <input type="button" value="Next"/> <input type="button" value="Previous"/> <input type="button" value="Save to database"/> </div>			Total Price
			Sales Tax
			Shipping Cost
			Extended Price

7. Press *Enter*.

Figure 10.2
Distributor Information form

Sales Order (Goal)			
		4030 Braker Lane West Suite 2001	
Distributor Information (Prompt)			
Name	Pat Myers		
Company	Stevens, Inc.		
Distributor Size	<input type="checkbox"/> Large <input type="checkbox"/> Medium		
Address	1214 19		
City, State, and Zip	Austin, Texas 78721		
Item	Quantity	Unit Price	Amount
Widget	50	\$295.00	\$14,750.00
Shipping Method	Discount	Less Discount	
Commercial Carrier			
			Total Price
			Sales Tax
			Shipping Cost
			\$50.00
			Extended Price

Several things happen when you enter the value for Quantity, as shown in the preceding figure.

First, notice that ObjectVision has calculated values for the Amount, Shipping Method, and Shipping Cost fields. Because you entered values for all fields required by the decision trees for Amount, Shipping Method, and Shipping Cost, ObjectVision can calculate their values.

Second, the Distributor Information form has appeared.

In the next steps, you'll complete the Distributor Information form and two Scratchpad forms. You will also write the application's values to the ORDER.DB database.

Finishing the order

When you created the decision tree for the Discount field, one of the conditions was the value of Distributor Size. Since you have not completed the Distributor Size field, ObjectVision opens the Distributor Information form and prompts you to supply a value. Notice that the form status is [Prompt]. Figure 10.2 shows how your screen looks now (except that the date on your form will be different).

The Distributor Information form contains three fields: Name, Company, and Distributor Size. Notice that the Name and Company fields already have values. You already completed these fields on the Sales Order form.

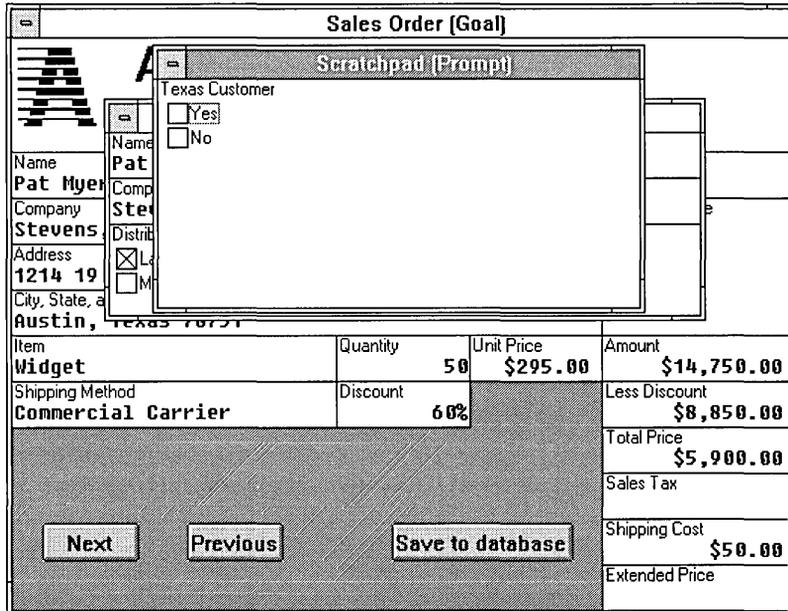
 Your next steps:

1. Press *Enter* to choose the large option, or click it with the mouse.

ObjectVision can now calculate values for the Discount, Less Discount, and Total Price fields.

Now a Scratchpad form appears as shown in the following figure. Remember that in Chapter 8 you created two fields that do not appear on any form, Texas Customer, and Austin Customer. These fields only appear in the Sales Tax decision tree. When ObjectVision tries to calculate a value for Sales Tax, it uses Scratchpad forms to prompt you for the values of these fields.

Figure 10.3
Scratchpad form



Item	Quantity	Unit Price	Amount
Widget	50	\$295.00	\$14,750.00
Shipping Method	Discount		Less Discount
Commercial Carrier	60%		\$8,850.00
			Total Price
			\$5,900.00
			Sales Tax
			Shipping Cost
			\$50.00
			Extended Price

The Texas Customer field is a check box field with two values, Yes and No.

2. Press *Enter* to choose Yes for Texas Customer.

The decision tree for Sales Tax specifies that if the value of Texas Customer is Yes, prompt for the value of Austin

Customer. A second Scratchpad form appears, prompting you to choose Yes or No for Austin Customer.

3. Press *Enter* to choose Yes for Austin Customer.

The Scratchpad form disappears. ObjectVision now has all the information it needs to calculate the Sales Tax and Extended Price fields on the Sales Order form.

4. Close the Distributor Information form. The following figure shows how your form should look now:

Figure 10.4
Completed Sales Order form

Sales Order [Goal]			
ACME Widgets		4030 Braker Lane West Suite 2001 Austin, TX 78759-5332	
Name Pat Myers		Order Date 12/12/90	
Company Stevens, Inc.		Customer Type <input checked="" type="checkbox"/> distributor <input type="checkbox"/> dealer <input type="checkbox"/> educator <input type="checkbox"/> other	
Address 1214 19th St.			
City, State, and ZIP Austin, TX 78751			
Item Widget	Quantity 50	Unit Price \$295.00	Amount \$14,750.00
Shipping Method Commercial Carrier	Discount 60%		Less Discount \$8,850.00
			Total Price \$5,900.00
			Sales Tax \$501.50
			Shipping Cost \$50.00
			Extended Price \$6,451.50
<input type="button" value="Next"/> <input type="button" value="Previous"/> <input type="button" value="Save to database"/>			

Notice that the form title bar now says [Complete]. You have filled in all the information required for the form. The next step is to enter the completed order into the order database.

When you choose the Save to Database button, ObjectVision stores the data on the current form according to the external database link that you created in Chapter 9.

5. Press *Tab* until the Save to Database button is highlighted and press *Spacebar* to activate it. You can also click it with the mouse.

ObjectVision enters the order information into the MYORDER database. Now other applications can use that information to perform credit approval, accounts receivable, shipping dock, status report, and other functions. The information will also be

available when you want to change the Pat Myers order (so you won't have to retype the company, address, customer type, and so on).

Printing the form

Now you will print the form that you've been working on. Two commands print forms:

- The File | Print Form command prints the current form.
- The File | Print All command prints all forms in the application.

Before printing forms, you can use the Windows Control Panel to select a printer or to change the attributes of the printer.

Because of differences among printers, the size and exact appearance of printed forms depend on the printer you are using.

 *Your next steps:*

1. Choose File | Print Form.
2. ObjectVision displays a message that identifies the form and printer, and starts printing the Sales Order form. Notice that you can stop the printing by choosing the Cancel button.

If you had used the File | Print All command, ObjectVision would have printed all the forms in the Order application (the Sales Order form, and the Distributor Information form) instead of just the Sales Order form.

Entering a new order

In this section, you will enter a new order and see how it differs from the order you just entered. You will also try some "what if" comparisons to see how changing parts of the order would affect its costs.

If you followed the steps in the previous section correctly, the Sales Order form is still displaying the values of the Pat Myers order. Before you can enter a new order, you must clear the forms.

The Clear All command, on the Edit menu, removes values from all the forms in an application. The Clear command, on the Form menu, removes values from only the currently selected form, leaving the values in the other forms.

 Your next steps:

1. Choose the Edit | Clear All command to remove the data from all the forms in the application. Don't worry about losing data, you have already saved it to the Order database.

If you had selected Form | Clear, ObjectVision wouldn't clear the Distributor Information form and you might get incorrect values in the new order.

2. Complete the Name, Company, Address, and City, State, Zip fields however you please.
3. Choose *dealer* as the Customer Type.
4. Type 15 as the Quantity and press *Enter*.

Notice that after you enter the Quantity, ObjectVision does not prompt you for Distributor size. The decision tree for Discount does not require a value for Distributor Size if the value of Customer Type is *dealer*.

The Texas Customer field now appears on the Scratchpad form. ObjectVision needs the value to calculate a value for Sales Tax.

5. Choose No for Texas customer.

Since you chose No, the Austin Customer field does not appear. Because your customer is not from Texas, there is no sales tax.

The second form is now complete. Compare the costs in this form to the printed copy of the Pat Myers form you completed. Notice that the discount is different (because the new customer is a dealer rather than a distributor). Note also that the shipping method is regular UPS rather than commercial carrier (because of the difference in quantity).

Before you enter the completed order into the order database, you can see what happens when you change values in some fields.

Making "what if" changes

In this section, you will change values of fields in a completed form to let the customer see how making changes, like increasing the size of the order, would affect the final price. Suppose that your customer wants to find out what the discount would be if the order was increased to 30 units.

 Your next steps:

1. Press *Tab* until the Quantity field is highlighted, or click the Quantity field with the mouse.
2. Delete the number *15* (you can use *Backspace* or *Del* or highlight the number with the mouse and then press *Del*).
3. Type *30* and press *Enter*.

ObjectVision recalculates all the fields affected by the new value. Amount and Shipping Method change, but Discount stays at 25 percent.

Notice that fields that depend on the Amount and Shipping Method fields have also changed value: Less Discount, Total Price, Shipping Cost, and Extended Price.

Your customer now wants to know how many Widgets must be bought to increase the discount. If you can't remember, you can easily find out by viewing the Discount decision tree.

4. Highlight the Discount field and choose Tools | Tree.
5. The Discount decision tree appears. Scroll down until you can see the branch node for the Dealer value of Customer Type. Notice that the conclusion for Discount is 40% if the Quantity is greater than or equal to 100, otherwise it is 25%.
6. Since there is no additional discount, the customer tells you to change the quantity back to 15 widgets.
When you change the value of the Quantity field back to 15, ObjectVision recalculates the affected fields.
7. Use the Save to Database button to enter the order into the database.

You can try changing the values for other fields in the form to see their effect.

Notice that you can't change the values for the Amount, Less Discount, Total Price, and Extended Price fields. In Chapter 4, you protected these fields so users could not enter incorrect values. Their values can only be calculated by their decision trees.

Finding data

You established a database link in Chapter 9 to a Paradox table so that you could store and retrieve information. You have already used the Save to Database button to store two records.

The Next and Previous buttons let you move sequentially through the records in the database provided you are already on a record.

- Enter several orders and then use the Next and Previous buttons to move between the completed orders. When you reach the beginning or end of the file a message will tell you that you can't go farther.

You can search for specific records using the Name field. Recall that when you created the Paradox table, the first field automatically became the index field. This means that you can search for records according to the value of the Name field.

- To find a specific record, type the name you are searching for in the Name field and press *Enter*. For example, type Pat Myers. If ObjectVision finds a match in the database, it fills in all the fields in the form with the values from the database. The name you type must exactly match the name in the database.

If you make any changes to the data in a field from an existing record, a dot pattern appears in the field. When you save a modified form with the Save to Database button, the new values update the old.

Summary

You have now completed the ObjectVision tutorial, and you're ready to create your own applications. *ObjectVision Reference* contains complete reference material for all the topics discussed in this tutorial.

Keyboard and mouse operations

This appendix lists the keys, key combinations, and mouse techniques you can use to perform ObjectVision operations. Keys and mouse techniques are listed in two main categories:

- for running applications
- for using ObjectVision tools



Mouse techniques you can use in ObjectVision immediately follow the keys listed in each category. This mouse symbol indicates that the text next to it begins the list of mouse techniques.

Choosing menu commands

When a menu command has a shortcut key or key combination, you can press the specified key or keys to directly carry out the action. The shortcut key or key combination, if it exists, follows the command name on the menu. For example, on the ObjectVision menu, *Alt+F4* follows the Close command.

You must choose other menu commands by selecting the menu you want, opening the menu, and choosing the command you want.



Selects the menu bar. By default, the first menu name is highlighted.

 After you have selected the menu bar, you can press ← or → to highlight the menu name you want.
or


 Opens the selected menu. (If a command is highlighted in an open menu, pressing *Enter* chooses that command.)
or

 Notice that each menu name has an underlined letter. As a shortcut to opening a menu, press *Alt* plus the underlined letter. For example,

  Opens the File menu.

  Opens the Form menu.

  Opens the Tools menu.

After you have opened the menu, highlight the menu command you want:

 Highlights a command in an open menu. Or press the underlined letter in the command name to choose that command. For example, when the File menu is open, you can press *O* to open a file, *S* to save a file, or *X* to leave ObjectVision and unload it from memory.
or


 Carries out a selected command or command button (except buttons on forms).

 Activates a selected button on an application's form or checks a box when a value option is highlighted.

 Exits from a menu without choosing a command.

 Click a menu name to open it. Then click a command name to choose it.

As a shortcut, position the pointer on the menu name you want and drag to highlight the command. When you release the mouse button, the command is carried out.

Running applications

When the ObjectVision window isn't occupying the entire Windows desktop you can move the window by choosing Move.

Press the arrow keys to reposition the window, then *Enter* when you finish.



Move to another application on the Windows desktop, activating it if it is loaded in memory.



Restore a minimized application to its previous size. Press *Alt+Tab* until the icon you want is highlighted, then release *Alt* only. After the application window opens, release *Tab*.



Opens the Task List window so you can switch to another application loaded in memory.

ObjectVision Control menu commands

ObjectVision application window Control menu:



Opens the Control menu. Use the arrow keys to select a command, then press *Enter* to choose it.



Closes the application window. Windows will prompt you for confirmation if your current work is unsaved.

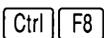
ObjectVision form window Control menu:



Opens the Control menu. Use the arrow keys to select a command, then press *Enter* to choose it.



Lets you move the form window with the arrow keys.



Lets you resize the form window with the arrow keys.



Closes the form window.



Double click the Control-menu box to close the application or form window. Windows prompts you for confirmation if your current work is unsaved.

Click any visible portion of another window on the desktop to switch to that other window. If no other windows are visible, resize the front-most window first.

Click the Maximize button to maximize the window.

Click the Minimize button to minimize the window.

If the window is open on the desktop, click the Restore button to return the window to its previous size.

If the icon is on the desktop, click the icon and choose Restore to return the window to its previous size. Or, double-click an icon to restore the application window to its previous size.

Getting help online

 Opens the Help menu.

Help for a field (during form completion):

 Displays Help for a highlighted field *if* the form designer created help for that field.

 Closes the Help window.

or





Scrolls down one line at a time.

or



Scrolls up one line at a time.

or





Scroll the help window one screenful at a time.

and





Opens the Help Control menu.



Closes the Help window.



Opens the Task List window so you can switch to another application loaded in memory.



Click outside the Help window to close it.

Drag the Help window border to resize it. (Move the pointer to the window's border until it changes to a double arrow.)

ObjectVision Help:

Alt H Opens the Help menu.

F1 Opens the ObjectVision Help window (unless you're in form completion mode *and* the form designer created help for the selected field). If a command or a tool is highlighted, help for that topic appears; otherwise, the Help Index appears.

Alt Spacebar Opens the Help Control menu.

Alt F4 Closes the Help window.

Ctrl Esc Opens the Task List window so you can switch to another application loaded in memory.

Alt F4 Closes the Help window.

or

Alt F X

Alt I Displays the Help Index.

Alt B Moves back to the previous help topic. Repeatedly pressing this shortcut moves back through all the topics viewed in the current session until Index was chosen. Choosing Index erases your Help history.

Alt R Moves back to the previous help topic if the topic is a part of a browse sequence.

Alt O Moves forward to the next help topic if the topic is part of a browse sequence.

Alt S Displays the Search For dialog box which lets you locate help topics from a list of subjects.

↑ Scrolls the help text one line at a time in the direction of the arrow key.

or
↓

PgUp Scrolls the help text up or down one screenful at a time.

or

PgDn

 Scroll the Help window one screenful at a time.

and



 Selects the next underlined help topic.

 Selects the previous underlined help topic.

 Chooses the selected help topic. If the topic has a solid underline, help for that topic appears. If the topic has a dashed underline, definition for that term appears onscreen as long as you hold down the key.

 Drag the Help window border to resize it, or click the Minimize, Maximize, or Restore buttons.

Click a word that has a solid underline to jump to help for that topic.

Position the pointer on a word that has a dotted underline and hold down the mouse button to display a definition for that word

Using dialog boxes

 Closes the dialog box and leaves the settings unchanged.

 Toggles a selected check box between checked and unchecked.

 Activates a highlighted command button.

 Moves to the next named option or group of options. Or, press *Alt* plus an underlined letter in the option name to select that option directly.

 Moves to the previous named option or group of options.

 Moves to a new line in a text box.

 Selects a highlighted radio button.

or



 Highlights a selection list item. Or, press the first letter of the item name to highlight the first item in the list beginning with that letter.

or



Completing forms

Open the current form's Control menu.

Closes a form. Or, choose Close from the Control menu.

Lets you move a form with the arrow keys. Or, choose Move from the Control menu; then press the arrow keys to reposition the window and press *Enter* when you finish.

Lets you resize a Scratchpad form. Or, choose Size from the Control menu; then press the arrow keys to resize the form's border and press *Enter* when you finish.

Scrolls a form horizontally.

or

Scrolls a form vertically.

or

Fields

Moving between fields:

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).

Enters a typed value in the field and moves to the next field requiring user input.

Enters a typed value in the field and moves to the next field, going from left to right and from top to bottom.

Moves to the previous field, going from bottom to top and from right to left.

Selecting field value options:

Activates a highlighted button or checks a highlighted check box.

Enter Checks a highlighted check box. Highlight a check box by pressing ↓ or ↑ when the field is selected, or press the first letter of the response you want, repeating if necessary until you move to the response you want.

Enter Selects a highlighted list item. Highlight the item you want by pressing the first letter of the item you want, or press ↑ and ↓.

or

Tab

Spacebar Toggles a true/false field between checked and unchecked.

Editing text in fields:

Shift  Highlights text. Highlighted text is replaced by the next characters you type.

or

Shift 

Esc Restores the previous field value (*only* if you are still in the field and haven't pressed *Enter* to enter the new value).

Home Moves to the left of the first character in the field.

End Moves to the right of the last character in the field.

Del Erases the character immediately to the right of the pointer. Or, erases any highlighted text.

Backspace Erases the character immediately to the left of the pointer. Or, erases any highlighted text.

 Double-click any letter in a word to highlight the entire word.

Drag to highlight any amount of text in a field. Highlighted text is replaced by the next characters you type.

Viewing decision trees

Moving between nodes:

Home Moves to the root node of the decision tree.



Moves to the previous node, if any, at the same nesting level in the decision tree.



Moves to the next node, if any, at the same nesting level in the decision tree.

-  Moves to the node, if any, at the preceding nesting level in the decision tree.
-  Moves to the first node, if any, at the subsequent nesting level in the decision tree.

Scrolling:

  Scrolls a tree horizontally.

or

 Scrolls a tree vertically.

or



Changing the display:

  Enlarges the size of the decision tree.

  Reduces the size of the decision tree.

Form Tool

Control menu commands:

  Opens the form Control menu.

  Lets you move a form with the arrow keys. Or, choose Move from the Control menu; then press the arrow keys to reposition the form and press *Enter* when you finish.

 Moves a selected form object in the direction of the key.





or



Shift ↑

Lets you position and size a new form object. First, position the pointer using the arrow keys, then press *Enter*. Press the arrow keys again to define the size and shape of the new object, then press *Enter*.

Shift ←

Shift →

or

Shift ↓

Ctrl F8

Lets you resize a form using the arrow keys. Or, choose Size from the Control menu; then press the arrow keys to resize the form's border and press *Enter* when you finish.

Shift

Arrow keys

Lets you resize a select form object. *Shift+arrow* keys move the object's lower right corner.

Scrolling:

Ctrl PgUp

Scrolls a form horizontally.

or

Ctrl PgDn

PgUp

Scrolls a form vertically.

or

PgDn

Selecting a form object:

Tab

Moves to the next object, going from left to right and from top to bottom.

Shift Tab

Moves to the next object, going from bottom to top and from right to left.

Ctrl Tab

Lets you select multiple objects on a form, going from left to right and from top to bottom.

Shift Ctrl Tab

Lets you select multiple objects on a form, going from bottom to top and from right to left.



Click to select a form object.

Shift

Click

Begins or ends a contiguous block selection of form objects. If the begin block selection is above the end block selection, the objects are selected going from left to right and from top to bottom. If the begin block selection is below the end block selection, the objects are selected going from right to left and from bottom to top.

Ctrl Selects multiple, non-contiguous form objects.
Click

Stack Tool

↑ Moves toward the top form in the stack.

or



↓ Moves toward the bottom form in the stack.

or



PgUp Scrolls the stack display vertically.

or



This chapter 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; 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 fields 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 decision 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 fields** An ObjectVision field type that is evaluated as unactivated (false) or activated (true). Button fields can be automatically created by ObjectVision when a link is first created. Buttons fields can perform complex tasks. For example, button fields can be assigned a single conclusion node to activate an external link.

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 decision tree that computes its value.
- 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 decision tree are inappropriate.
- complex branch node** An individual segment of calculation logic that requires evaluation of one or more decision 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 decision tree. The conclusion expression is located in the end node for a logical path in a field's decision 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.

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 are read-only links that require the Windows application and its document file to work.
- decision logic** A process defined as a series of small steps, or nodes. Decision logic is graphically represented for each field's decision tree in the Tree Tool.

- 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.
- decision tree** A graphical representation of ObjectVision's field-value calculation process that uses one or more segments of calculation logic, or nodes.
- default** A calculated field value that can be overridden by the user. In the *Myorder* application, you assigned the Item field a default value of Widget.
- Dynamic Data Exchange (DDE)** The Windows protocol for dynamically transferring data between Windows applications. DDE links are read-only links that require the Windows application and its document file to work.

E

- Edit form** The status of the active form in the Form Tool. The [Edit] status appears in the form's title bar to indicate that it receives the actions you perform.
- empty node** Appears when a field has no decision tree, or a branch node has no nodes under it. If an empty node is evaluated, an error value is returned.
- 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 links** The part of an ObjectVision application used to read from and write to ASCII, Paradox, dBASE-compatible, and Btrieve data files. Additionally, read-only links can be created for Dynamic Data Exchange (DDE) files. Links are created with the Tools | Links dialog box or using @functions.

F

- field** A uniquely named, blank object that contains values entered by a user or calculated or linked values. 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 or text object.

form

- form** A uniquely named object that contains a collection of fields, analogous to a paper form. A form is the primary interface between your ObjectVision applications and users.
- Form Clipboard** An ObjectVision clipboard for cutting, copying, and pasting forms in the Form Tool. Unlike the Windows Clipboard, the Form Clipboard can't be used to transfer data between different applications.
- 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 (text, fields, rounded and filled rectangles, lines, and graphics).
- @function** Performs advanced 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 or Windows metafile image copied to the Clipboard. A graphic is pasted into ObjectVision from the Windows Clipboard 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 Edit | 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.

L

- label** 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-only 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.
- 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

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 decision 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 text, a field, filled or rounded rectangles, a line or a graphic. Objects can be assigned different properties, or attributes. For example, there are fourteen options for fill pattern and four options for line weight.

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 action of transferring data from a Clipboard to the active window. Or, transferring list items to an expression using the Paste Function or Paste Field buttons.

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 value font is 10-point Courier.
- precedence** The order in which operations are evaluated in an expression. Certain operators are always evaluated after others, unless nested.
- protection** A field property you assign with Properties | Protection to keep users from changing the field value or viewing a decision 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 * [] { } ,
- root node** The first (leftmost) segment of calculation logic in the decision tree. The root node can be either a branch node or a conclusion node.
- 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.
- 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 decision trees from other fields.

stack

- stack** The form order in an ObjectVision application, as displayed in the Stack Tool. The Goal form is the top form in the stack.
- Stack Clipboard** A temporary holding place for forms you cut or copy in the Stack Tool. The Stack Clipboard is different from the Windows Clipboard, and can't be used to transfer data between two different ObjectVision applications.
- stack order** The arrangement of an application's forms, as displayed in the Stack Tool, with the Goal form at the top.
- 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

- 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].
- Tree Clipboard** A temporary holding place for tree objects you cut or copy in the Stack Tool. The Tree Clipboard is different from the Windows Clipboard, and can't be used to transfer data to any other application.
- Tree Tool** The ObjectVision window where you can create or modify decision trees.

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.

& operator 108

A

Add New Field option 31

alignment

changing 50

field values 47

Alignment command 51

ALOGO.BMP 43

applications 8, 10, *See also* files

compared to spreadsheets 9-21

developing 8, 52, 77

calculated fields and 55

online line help 21

prototypes 9

distributing 10

end users 9

linking 111-117

compatible file types 16

opening 30

sample 5, 23, 26

decision trees 63, 71

saving 26

shared 17, 18, 19

validity checking 15

arguments 102

template 103

truncating 103

ASCII files 16

ASCII links 16

attributes *See* properties

Automatic option (values) 79

B

bitmap graphics *See* graphics

@BLANK function 107

Bold option 38

borders

field 14

line patterns 29

removing 40, 41

Borders command 40, 41

Borland, contacting 3

Branch command 65

branches *See* decision trees, nodes

Btrieve dialog box 18

Btrieve files, accessing 18

Btrieve links 18

button fields 48

creating 50

linking to decision trees 115

buttons *See also* specific button

adding to linked fields 115

print cancel 126

window resizing 24, 43

C

Calculate command 15, 20

calculated fields, protecting 55

Cancel button 126

caption text *See* text objects

carriage-return character 39

case sensitivity

@functions 102

file names 26

check boxes 48

adding to forms 51-54, 85

Check Boxes option 85

circular logic 88

Clear All command 119, 126

Clear command 126

Clipboard 34

clock 102

Close Tool command 26, 33, 69

- commands 120, *See also* specific command
 - choosing 7
 - repeating last used 40
- communications, remote 16
- Complete status indicator 125
- concatenation operator 108
- Conclusion command 66, 80, 82
- Conclusion dialog box 15, 66, 82
 - expressions and 95
- conclusion nodes *See* decision trees
- Condition command 80
- Condition dialog box 15, 82
 - expressions and 95
- conditional logic 12
- conditions *See* decision trees
- Connect button 115
- constants 15, 91
 - numeric 92
 - syntax 91
 - text 92
- Copy command 34, 78, 80
- currency, formatting 50
- currency display format 48
- customer assistance 3
- Cut command 68

D

- data
 - analyzing 14, 127
 - clearing 116, 119, 126
 - conversion 102
 - editing 121
 - entering 120, 126
 - picture fields and 58
 - problems with 32
 - validating 107
 - formatting 48-51
 - remote 16, 111, 116
 - removing 121
 - restoring previous entry 121
- databases 116
 - data-translating conventions 114
 - finding specific record in 129
 - moving sequentially through records in 128
- Date Format dialog box 49
- dates, formatting 49, 102
- .DB files 113

- dBASE-compatible files *See* dBASE files
- dBASE dialog box 18
- dBASE files, accessing 17
- dBASE links 17
- DDE links 19
- decimal places 48
- Decimal Places option 50
- decision trees 12-15, 59-70
 - branching 15, 60
 - building 14, 59, 64, 74
 - default values 87
 - circular logic 88
 - conditions 61, 66
 - entering 68
 - Otherwise 65
 - testing 84
 - copying 81
 - editing 14, 80-85
 - linking to button fields 115
 - nesting levels 61
 - nodes 13, 59
 - branch 60, 82
 - building 65, 73, 82
 - conclusion 13, 15, 61, 66
 - empty 61
 - root 61
 - truncated 84
 - printing 84, 85
 - protecting 55
 - resizing 84
 - scrolling 76
 - testing 69
 - viewing 84
- dialog boxes *See also* specific dialog box
 - entering text in 93
 - selecting items in 30
- Disconnect button 115
- display formats 48-50
- dots in fields 90
- Dynamic Data Exchange *See* DDE links

E

- empty nodes 61
- error messages 107
- Expand command 84
- expressions 15, 91-94
 - creating 15

- editing 15
- elements 92
- entering @functions in 101-103
- numeric 92
- operators 93
- pasting in 95
- syntax 92
 - constants 92
- external links *See* links

F

- Field command 30, 35
- Field Name dialog box 30, 35
- field names 95
 - decision trees and 66, 67, 74
 - entering in expressions 15, 92, 95
 - truncated 114
- field properties 10, 47, 77
- field sequence
 - guided completion and 120
- Field Type command 49
- Field Type dialog box 49
- field values 11, 19, 107, 127
 - alignment 47
 - calculating 12, 13, 14, 15
 - decision trees and 59
 - external links and 16, 112
 - problems with 53
 - changing 20, 128
 - clearing 119, 126
 - constant 15, 91
 - entering 12, 20, 33
 - automatically 79
 - decision trees and 13
 - picture fields and 57
 - selection lists and 51, 85
 - overriding 14, 90
 - prompting for 88
 - recalculating 14, 20
 - restoring 15
 - setting default 87
 - storing 116
 - writing 16
- fields 9, 11-12, 28
 - adding
 - buttons to 50
 - help 54
 - selection lists to 51-54, 85
 - attributes *See* field properties
 - borders 14
 - button 48, 50, 115
 - calculated 55
 - entering data in 120
 - finding specific 77
 - formatting 47, 51
 - grayed 129
 - inserting in forms 11, 30, 33, 77
 - copying already existing 34
 - linking 115
 - moving 33
 - naming 31
 - on multiple forms 78
 - ordering 28
 - picture 57
 - protecting 55
 - renaming 35
 - resizing 32
 - selecting 39
 - types 14
 - validating user input 107
- files *See also* specific file name
 - ASCII *See* ASCII files
 - Btrieve 18
 - dBASE-compatible *See* dBASE files
 - Paradox 17
 - reading from 16, 111
 - disabling 115
 - README 3
 - remote 16
 - saving 26
 - text *See* ASCII files
 - writing to 16, 111
 - disabling 115
- Fill Pattern command 42
- Fill Pattern dialog box 42
- fill patterns 28, 42
- Filled Rectangle command 41
- financial display format 48
- Find command 77, 120
- fixed display format 48
- fonts 47
 - changing 38
 - default size 38
- Form command 24, 30

- form completion mode *26, 119*
 - returning to *33*
- form edit mode *30*
- Form Name dialog box *24*
- form objects *27-29*
 - multiple *50*
- Form Tool *7, 10, 47*
 - activating *24*
 - closing *26*
- format options *48*
- forms *8, 9-11*
 - adding check boxes to *51-54, 85*
 - choosing *78*
 - clearing data from *116, 119, 126*
 - creating *24, 52, 77*
 - entering error messages in *107*
 - inserting graphics in *29, 43*
 - multiple *78*
 - ordering *10, 79*
 - printing *126*
 - problems with *32*
 - resizing *25*
 - startup *10*
 - status indicators *19*
 - temporary *11*
- forms interface *9*
- formulas *See expressions*
- @functions
 - @BLANK *107*
 - @INT *103*
 - @MESSAGE *107, 108*
 - @NEXT *116*
 - @NOW *102, 103, 104*
 - @PREVIOUS *116*
 - @ROUND *103, 104*
 - @STORE *116*
- arguments *102*
 - template *103*
 - truncating *103*
- categories *101*
- decision trees and *66, 67*
- entering in expressions *15, 92, 95, 101-103*
- link *17, 18, 112*
- syntax *102*

G

- general display format *48*

- Goal form *10, 19*
 - setting *79*
- Graphic command *44*
- graphics *27*
 - inserting in forms *29, 43*
- grayed fields *129*
- guided completion *10, 19, 20, 120*
 - interrupting *20*

H

- help *21*
 - adding to fields *54*
- Help command *54*

I

- icons, form *10*
- incremental development *8*
- indicators *See status indicators*
- input, validating *107*
- @INT function *103*
- Italic option *38*

K

- key shortcuts
 - Copy command *78*
 - Paste command *34*
 - Repeat command *40*
- keys *131-141*
 - editing *121*
 - form completion *120*

L

- Label Font command *38*
- Label Font dialog box *38*
- leading spaces in expressions *92*
- line patterns *29*
- Line Width command *29*
- lines *29*
- Link dialog box *16*
- link functions *17, 18, 112*
- links *16-19, 32, 111-117*
 - creating *16, 112*
 - data-handling conventions *114*
- Links dialog box *112, 113*

Links Tool 7, 16, 111-117
 activating 112
 logic 13, 60, 61
 circular 88
 conditional 12
 Otherwise condition 65
 logical values 48, 92
 lookup *See* links

M

Maximize button 24
memory, links and 17
menus *See also* specific menu
@MESSAGE function 107, 108
Microsoft Windows
 Control Panel application 126
 Dynamic Data Exchange 19
 files, accessing 19
 Paintbrush application 43
Minimize button 43
Minimize command 43
mouse 131-141
multi-users *See* shared applications
MYORDER.DB 114

N

Name/Text command 35
nesting levels, decision trees 61
newline character 39
Next button 128
@NEXT function 116
nodes *See* decision trees
@NOW function 102, 103, 104
numbers
 formatting 48
 negative 48
 rounding 104
 serial 102, 103
 thousands indicator 48
numeric constants 92
numeric expressions 92
numeric values 48

O

objects *See* form objects; text objects

Objects command 83
ObjectVision Runtime 10
 form completion and 19, 120
online help 21
 adding to fields 54
operators 93
 text concatenation 108
Order application 5, 23
 decision trees 63, 71
 saving 26
ORDER.DB 6
ORDER.PX 6
Otherwise condition 65
Outline option 40

P

Paradox
 files, accessing 17
 tables, creating 112
Paradox dialog box 17
Paradox links 17
 creating 113
 saving 115
parentheses in @functions 102
Paste Arguments option 103
Paste command 34, 68, 80
Paste Field button 95
Paste Function button 103
percent display format 48
Percent option 50
picture display format 48
picture fields, creating 57
Picture String dialog box 57
plus signs in field values 90
Previous button 128
@PREVIOUS function 116
Print All command 126
print cancel button 126
Print Form command 126
printers 126
properties 10, 47, 77
 default 77
Protection command 56

Q

Quattro Pro compatibility 15, 92

quotation marks
 in decision trees 74
 in expressions 92, 93

R

README 3
recalculation 14, 20
rectangles 28, 41
Reduce command 84
remote files 16
Repeat command 40, 50
repeating last used command 40
Restore command 20
Resume command 20, 120
root nodes *See* decision trees
@ROUND function 103, 104
rounded rectangles 28
Runtime *See* ObjectVision Runtime

S

Save As command 26
Save dialog box 26
Save File Name As dialog box 44
Save to Database button 128
Scratchpad form 11, 65, 77
 default properties 77
scroll bars, decision trees 76
Select command 78, 120
select vs. choose 7
selection, multiple fields 39
Selection List option 52
selection lists 48
 adding to fields 51-54, 85
 decision trees and 66
 viewing 53
serial numbers, returning 102, 103
shared applications 17, 18, 19
single quotes *See* quotation marks
Size command 25
spreadsheets *See also* Quattro Pro
 applications vs. 9-21
stack 10
Stack Tool 7, 79
startup form 10
status indicators 19, 125
@STORE function 116

string values 92
syntax
 @functions 102
 expressions 92
 system clock 102

T

tasks, automating 8
technical support 3
text *See also* text objects
 editing 121
 entering in
 dialog boxes 93
 help screens 54
 removing 121
Text command 37
text concatenation operator 108
text constants 92
text files *See* ASCII files
text objects 28
 adding to forms 37
 properties, fonts 38
 removing borders 40
text strings 28
Text Value dialog box 37
time, formatting 49, 102
title bar 24
trailing spaces in expressions 92
transfer *See* graphics
Tree command 64
Tree menu 64
Tree Tool 7, 12, 14, 59
 activating 64
 closing 69
truncated arguments 103
truncated branch nodes 84
truncated field names 114

U

Underlined option 38
Undo command 121

V

validity checking 15
value checking 109
value types 11

values 28, *See also* field values; numeric values
 default 87
 formatting 48
 invalid 107
 logical 48, 92
 string 92
 text 92
Values of dialog box 52

W

what if analysis 14, 127
Windows *See* Microsoft Windows
windows
 maximizing 24
 minimizing 43

TUTORIAL

OBJECTVISION™

FOR WINDOWS

B O R L A N D

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-OBV02-10 ■ BOR 2099