

**Chapter
1**

**GETTING
STARTED**

*Get on the
Fast Track!*



TM

**SYS-ED/
COMPUTER
EDUCATION
TECHNIQUES, INC.**

Objectives

You will learn:

- C Objects.
- C Properties.
- C Events.
- C Role of forms.
- C Referring to objects and their values.
- C Event-driven applications.
- C Data access objects.
- C Comments.

1 Introduction

Access Basic is an event-driven programming language that supports the structured programming features found in Microsoft Visual Basic and other programming languages.

Microsoft Access applications are made up of the following objects:

C Tables	C Forms	C Reports
C Macros	C Modules	

These are the same objects which are used in a Microsoft Access database and stored in MDB files.

Programming in Microsoft Access application involves working with:

C Objects	C Properties	C Events
-----------	--------------	----------

With Microsoft Access, the objects manage themselves; there is no main program.

If there is a requirement for something to happen when the user clicks a button, it is not necessary to have code which checks to see whether the user clicks the button. Instead the code is attached to the button in order that it will be executed when the Click event occurs.

When the event occurs, Microsoft Access runs the code automatically.

2 Objects

Access Basic applications are made up of objects.

There are two types of objects:

Direct Objects	These are objects that the users see and use directly such as forms and reports.
Supporting Objects	These are objects which control how the Direct Objects work.

- C Forms and reports (i.e. Direct Objects) are controlled by tables, queries, macros, and modules (i.e. Supporting Objects).
- C Forms and other objects are designed in their respective Design views.
- C Unlike old style conventional programming, do not start by writing code, and then observing how it works.
 - C Code in Microsoft Access application runs only in the context of an object and an associated event.
- C Until objects have been built, there is no use for the code.

3 Properties

Objects have properties. An objects' properties are set to make them look and behave the way you want.

All forms have a DefaultEditing property that specifies whether users can edit data on the form or only view it.

Once a property is set, the form opens automatically in the correct mode.

4 Events

Actions taken by a user in a form such as changing data in a field, clicking a command button, and moving the mouse are recognized by Microsoft Access as events.

Microsoft Access responds to these events automatically.

- C When a user changes the data in a text box, Microsoft Access checks to make sure that the data is the right data type.
- C When a user clicks a command button, Microsoft Access displays the button as sunken.

Event Procedure and Macros

A custom response can be added to an event. This is done by either a macro or an event procedure.

An event procedure is an Access Basic procedure that's attached to a form, report, or control; it's executed when a specified event occurs. You specify in the event procedure or macro what you want to take place when the event occurs.

It is possible to change an object properties, open or close objects, or manipulate data.

Event properties of the form are used for determining whether in response to an event, Microsoft Access runs a macro or an event procedure.

In order to have a macro run in response to a command button's Click event, the button's OnClick event property is set to the name of the macro.

5 Role of Forms

Forms are more than screens for entering and editing data; they make up most of the application's interface. From the perspective of a user, forms are the application.

By building an application around forms, it is possible to control the flow of the application through the events that occur on the form.

In addition to using forms as the application's interface, fields on hidden forms can be used for storing and passing values in macros from form to form or from operation to operation.

If users need the ability to enter a range of dates in a dialog box and then print a series of reports based on that range of dates, then the dialog box is the form which needs to be used. When a user chooses OK in the dialog box, the form is hidden rather than closed. The dates that the user entered are now available to the macro that prints each of the reports.

6 Referring to Objects and Their Values

Microsoft Access requires a specific syntax structure for identifying the control that contains the required value.

In order to refer to an object or a value, start with an object and identify each element in turn, separated by the following operators:

Operator	Use	Example
! Exclamation point.	Before an element of a collection. Typically this is an object that has been named by the programmer.	In order to refer Phone control to a form that has been named Customer, use this expression: Forms![Customer]![Phone]
. Dot	Before properties, methods, and collections. Usually the objects that Microsoft Access names.	In order to refer to the Phone control's Width property on the Customer form, use this expression: Forms![Customer]![Phone].Width

7 Event-Driven Applications

An event is an action recognized by a form, report, or control. Objects in Access Basic automatically recognize a predefined set of events and respond to a particular event.

When the requirement is to have a control respond to an event in a particular way, an Access Basic event procedure for that event can be written.

Series of Occurrences in a Typical Event-Driven Application

1. A user starts the application and automatically opens a startup form specified in the application's AutoExec macro.
2. The startup form or a control on the form receives an event.

The event can be caused by the user (for example, a keystroke) or by your code (for example, an Open event when your code opens a form).
3. If there is an event procedure corresponding to that event, it executes.
4. The application waits for the next event.

Some events automatically trigger other events.

For example, when the DbClick event occurs, the MouseDown, MouseUp, and Click events also occur.

8 Data Access Objects

In addition to the objects defined by Microsoft Access, the Microsoft Jet database engine defines objects that handle the data management tasks of an application.

Objects created by the Microsoft Jet database engine include:

C	Tables	C	Queries	C	Relations	C	Indexes
---	--------	---	---------	---	-----------	---	---------

Objects defined by the Jet database engine are called data access objects.

Data access objects in Access Basic procedures can be used for sharing this code with other database applications that use the Jet database engine.

Microsoft Visual Basic also uses the Jet database engine.

9 Adding Comments

Whenever new procedures are created or existing code is modified, it is a good practice to add comments.

Comments begin with an apostrophe ('). This symbol tells Access Basic to ignore any words that follow on that line. Comments can also begin with the Rem statement, which was used in earlier versions of Basic and is equivalent to the apostrophe.

Comments can be entered on a line by themselves or at the end of a line of code.

Example:

```
Function FirstOfNextMonth (AnyDate)
    ' This function calculates and returns the date of
    ' the first day of the month following the date passed by
    ' the argument.
    ' Note that this works even if Month(AnyDate) = 12.
    FirstOfNextMonth = DateSerial(Year(AnyDate), _
        Month(AnyDate) + 1, 1)
End Function
```

Although comments increase the size of a module, they do make code easier to read and maintain.