3

Session Three: The Object-Orientated Paradigm

> When I ask you for one virtue, you present me with a swarm of them, which are in your keeping. Suppose that I carry on the figure of the swarm, and ask of you, what is the nature of the bee? And you answer that there are many kinds of bees. And I reply: But do bees do differ as bees, because there are many and different kinds of them; or are they not in their to be distinguished by some other quality, as for example beauty, size, or shape? How would you answer me?
>
> ***Socrates as transcribed by Plato (Meno) 399 BC***

When Socrates proclaimed that all bees are "just the same but different" perhaps he was talking about objects but was just a little ahead of his time.

While a (now obscure) language called Simula established most of the key concepts of Object Oriented programming in 1967, object oriented programming only became mainstream with Bjarne Stroustroup's definition of the C++ programming language in 1985. Had computers been around in Socrates' time we might have been using these techniques about 2,400 years earlier!

In the last session we had a taste of running before we could crawl. In the next two sessions we'll learn enough about the Access Object model and the VBA Programming language to be able to crawl, or perhaps even walk in a slightly wobbly way.

This chapter begins by teaching you the extremely simple concepts behind objects and their three ways of interfacing with the outside world: properties, methods and events. We'll even attach a little VBA code to an event.

You will end this session with a full appreciation of what objects are, what they can do, and what an object model is.

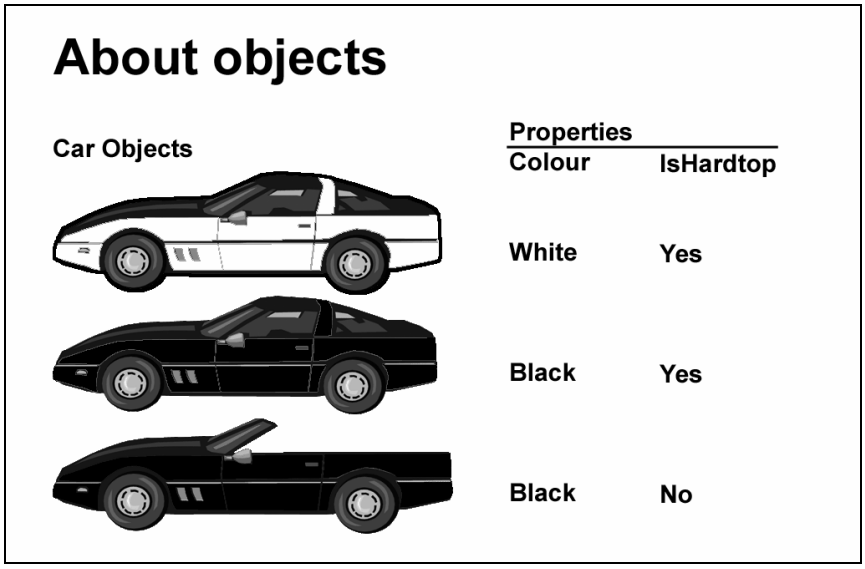
Session Objectives

By the end of this session you will be able to:

- Understand object properties
- Understand object methods
- Understand object events
- Understand some of the objects in the Access object model

Session3

Lesson 3-1: Understand properties



What are properties?

The slide above shows three cars. We can recognise them all as cars but there are some things that are different about them.

In object-speak we would say that the cars have different properties.

We can think of each as being a car object. If I were to telephone you and say "I bought a new car" you wouldn't know what color it was or whether it was a hard top—but you would still have an abstract idea of what was sitting in my driveway.

To better visualise the car you would have to ask a few questions like:

- Is it a convertible?
- What color is it?

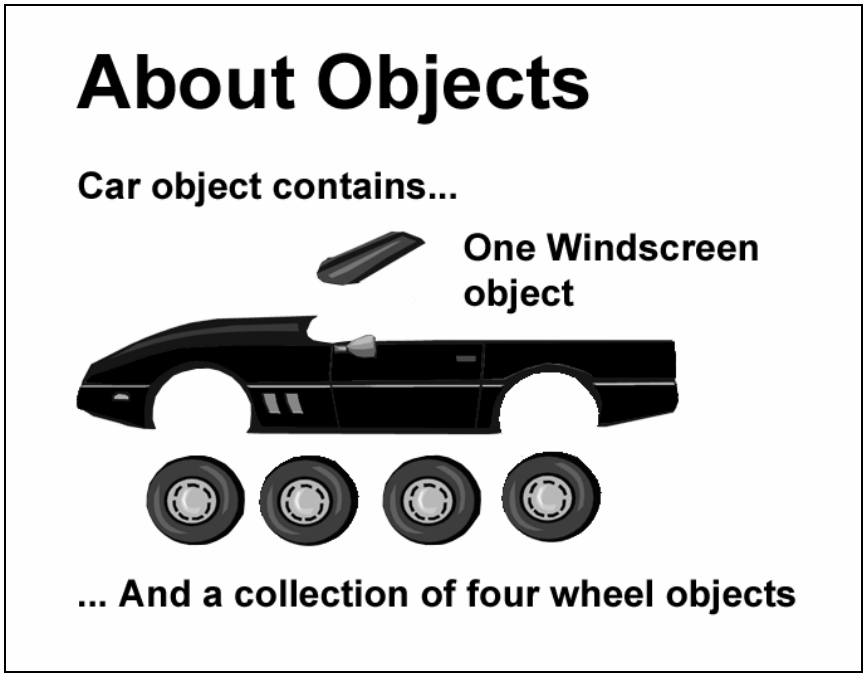
To relate this to Access consider a Form object. Form objects have a *Caption* property that puts the name of the form into the top blue bar.

If you added enough properties to the *Car* object you would eventually be able to precisely describe every car on the planet.

This would solve Socrates' bee problem too!

Session3

Objects contain objects



The car object contains a windscreen object. It also has four wheel objects that are identical (or at least very similar). In object-speak we would say that the car object contains a windscreen object and a collection of wheel objects.

The Access *Form* object contains a collection of control objects (such as command buttons and textboxes).

How properties are set for controls

1 Examine the properties of the cmdDelete command button on the frmFilm form.

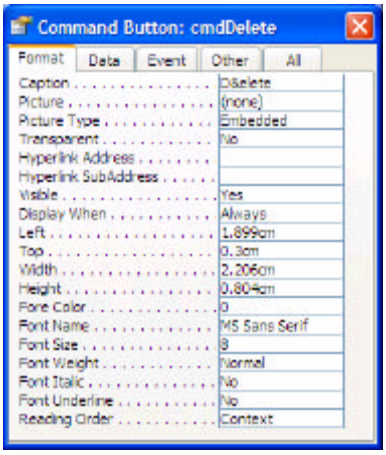
Open the form in Design View, right-click the cmdDelete button and choose Properties from the shortcut menu.

You can probably figure out what most of the properties mean.

note

Properties can be set both using the properties dialog and via VBA code.

For the moment we'll only use the properties dialog method but when we know a little more about the VBA programming language we'll be setting some properties using VBA code.

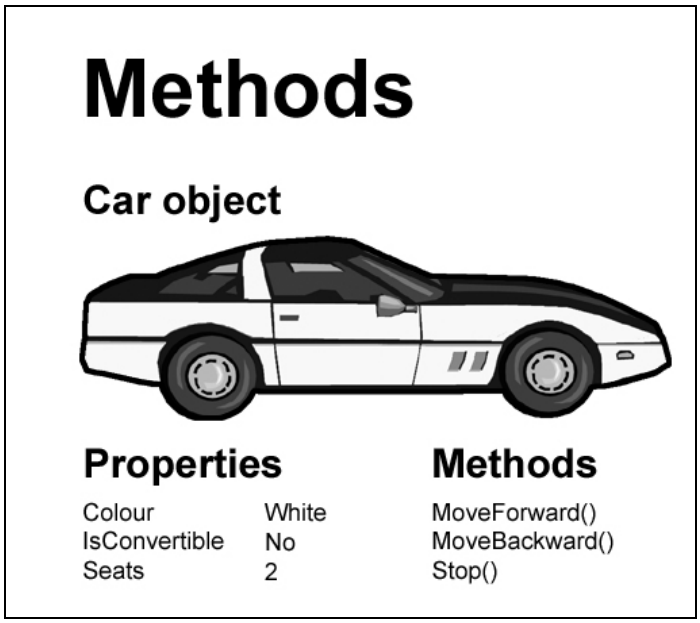


Lesson 3-2: Understand methods

What are methods?

As well as having properties to describe an object some objects have certain functionality too. Let's consider a Car object. We may have precisely described it by setting properties such as Color, NumberOfSeats and FuelType but what is the car for? Is it a film prop, a paperweight, or does it actually do something?

The things objects can do are called methods.



All cars have several methods such as StartEngine, MoveForward, Stop, SteerLeft and SteerRight. Some cars have methods that are not common to all cars such as StartAirConditioner or EnableCruiseControl.

Methods may be modified by arguments

Access objects have methods too. For example a ComboBox object has an AddItem method.

Simply invoking the MoveForward method of the Car object wouldn't really be enough. To more precisely tell the car what to do we might want to say:

Move forward three metres.

In order to do this we'd need to invoke the MoveForward method along with some instructions telling the car how to move forwards. These instructions are called parameters or arguments. The actual MoveForward() method call would more likely be something like this:

MoveForward(3)

note

Some programmers prefer the term *Parameter* instead of *Argument*. Both are correct.

Session3

Where the argument 3 represents the distance to move forwards in metres.

Examples of methods available in Access objects

A method with no parameters

Methods do not have to have parameters. A good example of a very useful method is the Form object's Requery method. Invoking the Requery method will execute the SQL query that underpins the form and will then update all of the controls on the form to reflect the current state of the data.

This could be very useful in a multi-user environment when the user wanted to be sure that the data displayed on the screen was completely up to date.

A method with parameters

The ComboBox object has a method that allows items to be added to it. This is intuitively named the *AddItem* method.

Clearly the *AddItem* method must have some parameters in order to tell the ComboBox object something about the item to be added.

The method actually has two parameters. The first is the text that will be displayed in the combobox and the second is an optional parameter to indicate the position in the list.

For Example:

```
AddItem( “London”)
AddItem( “Glasgow”)
```

Would result in a combobox with two entries, the first would be London and the second Glasgow.

```
AddItem( “London”, 1 )
AddItem( “Glasgow”, 0)
```

Would result in a combobox with two entries, the first would be Glasgow and the second London.

Lesson 3-3: Understand events

What are events?

Consider a telephone. Somebody could call the telephone at any time. The telephone doesn't know when it is going to happen, or even whether it will ever happen at all.

Suppose somebody does call, the telephone might respond by ringing its bell.

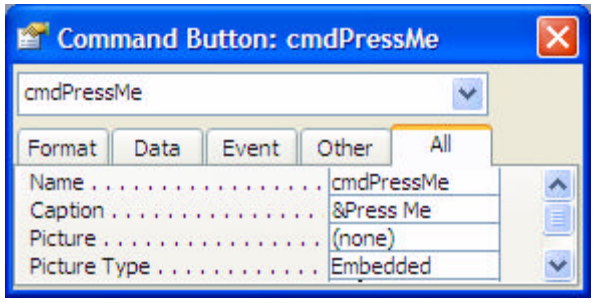
In object-speak we'd say that the telephone object had responded to its SomebodyCalling event by invoking its RingBell method.

1. Open a new blank database and save it as VBACode.mdb.
2. Create a new form in Design View and save it as frmTest.
3. Add a command button to your new form dismissing the Command Button Wizard by clicking its Cancel button.
4. Set some of the Command Button's properties.

   A Command Button (and every other control) is an object and will have its own set of properties, methods and events. It's important that you get into this mind set in order to understand how VBA and Access interact.

   To view the Command Button's property sheet right-click it and select Properties from the shortcut menu.

   Set the Command Button's Name property to *cmdPressMe* and the Caption property to *&Press Me*.



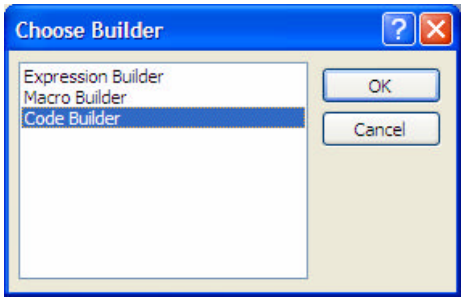
   Let's now consider an event that might happen in the life of a command button.

   The most important thing that can happen to the command button is that somebody might click it. This is referred to in object term inology as the command button's *Click* event.

5. Right-click the command button and choose Code Builder from the shortcut menu.

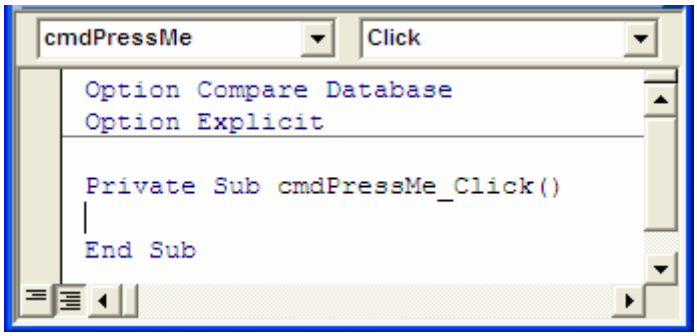
   The Choose Builder dialog appears.

Session3

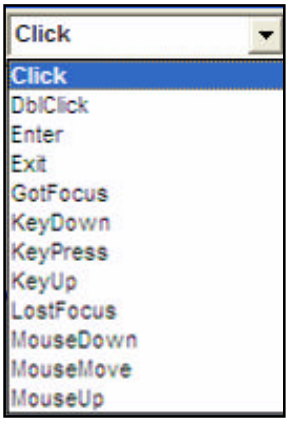


6 Choose Code Builder from the dialog.

The VBA editor appears with code for the Command Button's Click event handler displayed.



The command button does have other events but the Click event is shown by default as it is the most important event for this control.



7 Browse the Command Button's other events.

There's plenty of other less obvious things that can happen to a command button. In all there are twelve events.

Open the VBA editor and select the pull down list arrow at the top right of the code window. Browse the possible events for a Command Button. You should be able to guess what most of them mean.

8 Add the following code to the Command Button's Click event:

```
Private Sub cmdPressMe_Click()

Beep

End Sub
```

9 Test the Command Button's event handler.

Close the Visual Basic editor and display the form in Form View.

Click the button (make sure that your computer's sound is switched on). When you click the button the Click event is triggered. The code you added causes the computer to beep.

Lesson 3-4: Understand the Access object model

Great bugs have little bugs
Upon their backs to bite 'em

Little bugs have lesser bugs
And so on ad infinitum

May Louise Cooper, "Friendly Beetles", (1916)

About object hierarchies

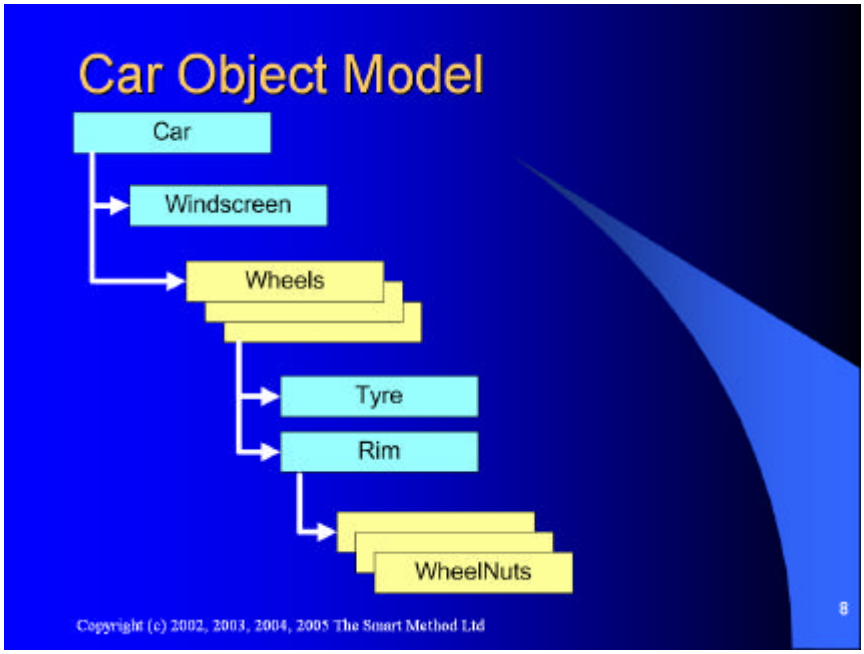
We have already seen how a Car object can contain other objects (such as a windscreen object).

Objects can also contain object collections of similar objects. In the earlier example the Car object had a collection of Wheel objects. The wheel objects did not necessarily have the same properties—for example there may have been wilder tyres fitted on the rear wheels.

Object models express the concept of a hierarchy evident in May Louise Cooper's observation above. Each of her bugs is connected to another bug higher up in the hierarchy.

The Car object model

Before considering the Access object model let's consider the Car object model that we've been discussing. Here it is:



- A Car object contains a Windscreen object and a collection of Wheel objects.

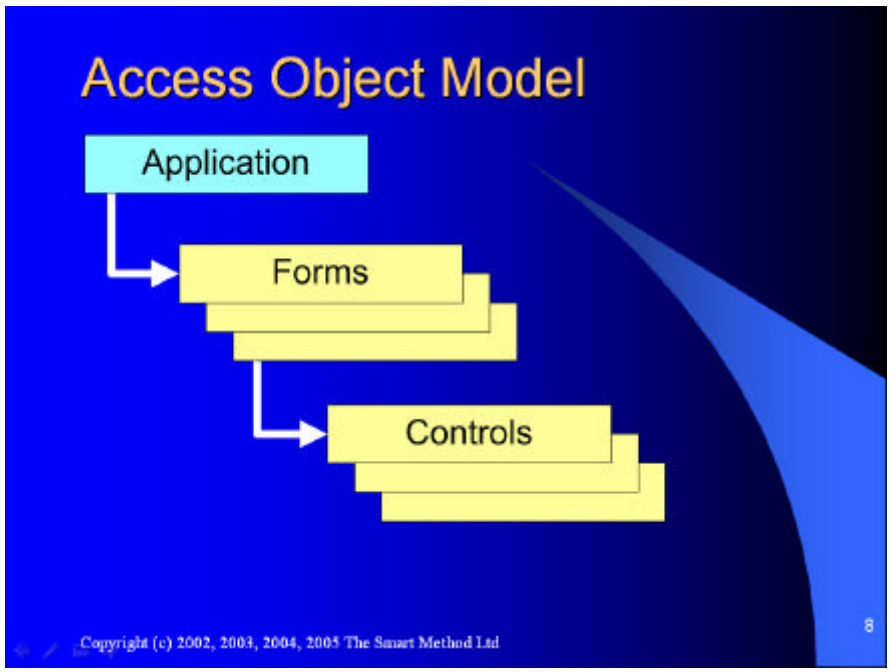
Session3

- Each wheel object contains a Rim object, and a Tyre object.
- Each Rim object contains a collection of WheelNut objects.

The Access object model

The Access object model contains a little over fifty objects. We're going to ease you gently into the model by introducing new objects as we use them during this course. We've already identified three of the most important objects.

Here's the object model as we know it so far:

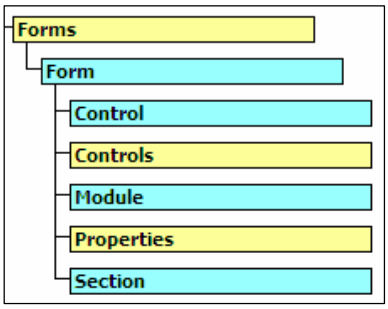


Microsoft always name collections in the plural and single objects in the singular. It is thus obvious that the Forms item in the above diagram is actually a collection of Form objects.

From the object model above it is clear that:

- One Application object may contain zero, one or more Form objects.
- Each Form object may contain zero, one or more Control objects.

Microsoft's object model diagram



If you type *AccessObjectModel* into the help dialog you'll be able to view the entire Access object model.

Microsoft use color coding to indicate collections of objects, and there's a few objects that you might not recognize, but from what you have learned in this section, and your existing Access skills, you will be able to understand the purpose of many of the objects.

Session 3: Exercise

1 **Consider a Head object. The Head object contains several single objects and several object Collections. Which of the following are single Objects and which are part of an Object Collection?**

Eye	? Single Object	? Collection
Nose	? Single Object	? Collection
Hair	? Single Object	? Collection
Ear	? Single Object	? Collection
Mouth	? Single Object	? Collection
Tongue	? Single Object	? Collection
Tooth	? Single Object	? Collection

If you are having difficulty with the concept of single objects and object collections slide the page slightly to the left for some help.

2 **Consider a Car object. Which of the following are Properties, which are Methods and which are Events?**

Color	? Property	? Method	? Event
MoveForward	? Property	? Method	? Event
Start	? Property	? Method	? Event
HitByStone	? Property	? Method	? Event
BreakDown	? Property	? Method	? Event
Make	? Property	? Method	? Event
Model	? Property	? Method	? Event
Stop	? Property	? Method	? Event

If you are having difficulty with the concept of properties, methods and events slide the page slightly to the left for some help.

3 **Name two properties, methods and events that a person object might have.**

Slide the page to the left if you need to view some sample answers or to confirm that yours are reasonable.

Session 3: Exercise answers

Q 3	Q 2	Q 1
Some examples: **Properties:** Height, Weight, Age, Nationality, BirthPlace, HairColor. **Methods:** Walk, Run, Swim, TurnLeft, TurnRight, Whistle, Shout, Talk, Sleep. **Events:** Hungry, Tired, Thirsty, Hot, Cold, Sick.	A Property often describes an object's physical appearance. For example a *Car* object has a *Color* property. A *Method* is something an object can do. For example a *Car* object has a *Stop* method. An *Event* is something that may happen unexpectedly in the life of an object. For example a *Car* object may encounter a *HitByStone* event.	An example of a single object is a *Nose* object because a *Head* can only have one *Nose*. When there are two or more very similar (though not necessarily identical) objects they are bundled into an object collection. For example: a Head object contains a collection of *Tooth* objects.