

## Chapter 1

# Access Application Development

### IN THIS CHAPTER

- ◆ The various versions of Access and how they differ
- ◆ Developing database applications with Access
- ◆ The future of Access

MICROSOFT ACCESS WAS INTRODUCED as a desktop database management system (DBMS) in 1992, and in the years that followed, became a leader in its category. As a part of the popular Microsoft Office suite of applications, Access is used daily by millions of people. Since its inception, Access has grown progressively more capable and easier to use. The most recent versions, while retaining their position as the most popular desktop DBMS in the world, enable developers to build applications that access information from large networked client/server databases based on Microsoft's enterprise scale SQL Server client/server DBMS.

Although many people use Access to perform relatively simple data management tasks that don't require any kind of programming, the Visual Basic for Applications (VBA) language gives Access the power to host systems that are customized to meet the exact needs of a wide variety of organizations, regardless of their size or the kinds of tasks that they need to perform.

The purpose of this book, after laying a little groundwork, is to take you from a non-programming Access power user to a developer who is comfortable using the full power of VBA to build sophisticated applications that execute complex tasks while making things easy for the user.

## Access Versions

Access 1.0 appeared in 1992, shortly after the appearance of Microsoft Windows. Access 1.1 came along in 1993, and in 1994, Access 2.0 arrived on the scene. Each new version was an incremental improvement that built on the strengths of the prior versions. Access 95 followed in 1995, Access 97 in 1997, Access 2000 in 1999, and Access 2002 in 2001. With each new version, Access has grown easier to use and has expanded its data-handling capability.

## Access 1.0, 1.1, and 2.0

When Access 1.0 first saw the light of day, it introduced features that were new to personal database products, or for that matter, any kind of database products:

- ◆ *OLE technology* in Table Designer enabled users to create databases that contained graphical images and audio and video files, in addition to text and numbers. OLE stands for *object linking and embedding*, which is rather descriptive of what OLE does. Users were even able to incorporate Word documents and Excel spreadsheets into an Access database.
- ◆ The *Forms Package* enabled users to create custom forms by dragging form elements onto a blank page. This was a major step forward in ease of use.
- ◆ The Report Wizard made report creation equally easy.
- ◆ The *drag-and-drop Query Designer* used *query by example* (QBE) technology to enable users to create queries without writing program code.
- ◆ The *Graph Wizard* turned chart production into an intuitive step-by-step process.
- ◆ In addition to its native files, Access 1.0 could also operate on dBASE, Paradox, and Btrieve files with native drivers, as well as others, such as Sybase SQL Server, DEC Rdb, and FoxPro) via an ODBC interface.

ODBC stands for Object DataBase Connectivity, and provides a common application programmer's interface (API) to a large number of different and mutually incompatible databases.

Access 1.1 added the following improvements:

- ◆ Enhanced ODBC and Btrieve support.
- ◆ Improved connectivity to FoxPro.
- ◆ Easy data export to Word for mail-merge applications.
- ◆ Database sizes up to 1GB (about eight times larger than 1.0 could handle).
- ◆ An Access *runtime* became available. Developers could create applications that could operate without Access installed on the system.

Access 2.0 added more capabilities:

- ◆ Wizards that automated tasks that had previously required coding on the part of the application developer.
- ◆ More sophisticated event handling.
- ◆ A debugger in the development tools, plus an automatic documenter and an Add-in Manager.
- ◆ Jet database engine improvements, allowing more complex databases to be built and making queries run over twice as fast.

## Access 95

Access 95 was the first version to officially be a part of Microsoft Office. Its controls were brought into conformance with the other members of the Office application suite.

- ◆ VBA was one of several development tools that Access shared with other Office programs, such as Word and Excel.
- ◆ The Import Wizard and the PivotTable Wizard were added.
- ◆ The Form and Report Wizards were improved.

## Access 97

Access 97 became more Web-aware:

- ◆ It stored hyperlinks as a native data type and allowed saving to HTML. Users could publish database objects to the Web, and extract data from an HTML document that contained a table, and then put the data into a database table.
- ◆ Class modules could contain the definition of a new object.
- ◆ Procedures in a module could hold the properties and methods of that object.
- ◆ Conditional compilation enabled developers to create both debug and production compilations of their applications by setting flags in the code.
- ◆ Developers could remove source code from an application, thus speeding execution and preventing people from viewing and copying it.
- ◆ New ActiveX controls were introduced, and compatibility with source code control systems was added to the Developer Edition of Office 97.

## Access 2000

Many aspects of Access were enhanced for this version, improving performance and expanding capability. This Access version introduced these features:

- ◆ ActiveX Data Objects (ADO) replaced practically all the data access functions that were previously performed with Data Access Objects (DAO).
- ◆ Access Project (.adp) files work with SQL Server and Microsoft Data Engine files as smoothly as .mdb files work with the Jet engine.
- ◆ Data access pages (DAPs) were introduced, enabling you to put things that look like Access forms and reports on the Web.

## Access 2002

The previous Access release added these features:

- ◆ The Data Access Page Designer was improved with these features:
  - Multi-level undo and redo
  - Better grouping with multi-table data entry
- ◆ Enhanced support for the SQL Server Desktop Engine was included, plus cascading updates and deletes.
- ◆ The Upsizing Wizard converted applications written to drive the Jet engine so they work with the SQL Server Desktop Engine.
- ◆ XML support was added.

## Access 2003

New features of Access 2003 include:

- ◆ Improved ease of use
- ◆ Expanded ability to import, export, and work with XML data files
- ◆ Flagging of common errors
- ◆ Identification of object dependencies

# Access as an Application Development Environment

Access provides a particularly congenial application development environment for several different classes of developers. It is congenial for the following reasons:

- ◆ **Low barrier to entry.** You can start developing useful applications with a minimum of study or training.
- ◆ **Intuitive.** As you use Access more and increase your proficiency, you can tackle more complex assignments, thus using more of Access's power.
- ◆ **Part of the Office suite.** Access can draw on the capabilities of the other members of Microsoft Office, such as Word and Excel.

- ◆ **Help from wizards.** The many wizards that automate a wide variety of tasks can dramatically reduce development time.
- ◆ **Tight integration with Microsoft SQL Server.** Because of this, Access is applicable to a broad range of development tasks, from the smallest personal database application to enterprise-wide applications for large multinational firms. Other DBMS products typically aim for one end of the size spectrum or the other.

## The difference between databases and database applications

Sometimes people fail to make the distinction between databases and database applications. This is particularly true of people who “learn by doing” with Access rather than of those who take a formal course in database theory. Access lumps *database development* and *database application development* together, with no apparent separation of the two:

- ◆ A *database* is a self-describing collection of integrated records. It is *self-describing* because the information that describes the structure of the database is included in the database. The records are *integrated* because relationships between data items are explicitly maintained. These two characteristics differentiate a database from a mere collection of data items.

A database is a structured collection of data. By itself, it doesn't solve any problem or serve any purpose. The database application uses the data in the database to solve problems or to accomplish a purpose.

- ◆ A *database application* is a program that works with a database to maintain and deliver desired information. A database application developer may write the database application or it may be generated semi-automatically by a DBMS, such as Access.

In a relational database, the data is stored in tables, which give the data its structure. A database application may include forms and reports that enable users to interact with the data in the database, with the objective of accomplishing some purpose.

## The Database window

When you enter Access and specify the name of the database that you want to work on, the Database window shown in Figure 1-1 is displayed.

The column shown on the left lists the kinds of objects that it deals with: *tables*, *queries*, *forms*, *reports*, *pages*, *macros*, and *modules*.

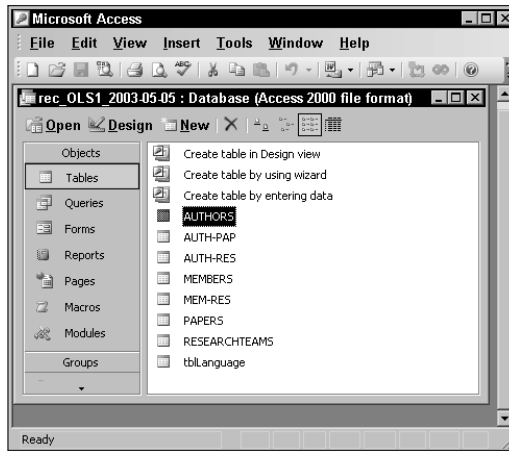


Figure 1-1: Database window.

For example, if Tables is selected, the pane on the right displays the names of all the tables in the current database plus three options for creating tables, either in Design View, with a wizard, or just by entering data. Similarly, selecting Forms displays the names of any forms that have been created, and gives you the option of creating a new form either in Design View or by using a wizard. Access makes it supremely easy to create all the different database objects that you may need for your application.

## Creating tables

Because this is an advanced book on programming with Access, we assume that you can already create a table using Design View. Figure 1-2 shows an example of Design View after entering the specification for the first field of a table. You enter the field name, data type, and description of the field, and then specify some of its properties, such as field length and whether the field is indexed. After you have done this for all the fields in a table, name and save the table. That is all that's involved in creating a database table. Of course, we assume that you have already carefully thought about what fields should be included in the table, what their data types should be, what properties the fields should have, and how this table relates to other tables in the database.

After you have created the tables, you have created the database skeleton or structure. All you need to do now to complete the database is to fill it with data. You can do this in several ways, including through a data-entry form, by typing directly into the table, or by filling the table automatically from an existing file.

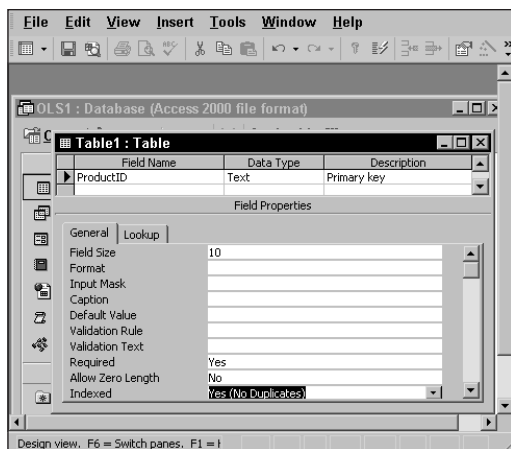


Figure 1-2: Design View of table creation.

The following sections are about creating parts of a database application (forms, reports, data access pages, and so forth), and are not about creating the database itself. Unlike tables, forms, reports, and data access pages are all involved in performing specific tasks or answering specific questions.

## Creating forms

For most applications, users see and interact with forms. Access forms are flexible tools. Forms can be different sizes, and they can contain such objects as controls, text, and graphics. For example, you can

- ◆ Create switchboard forms that control navigation around an application from one screen to another.
- ◆ Use a form to
  - Enter data into one or more database tables
  - Modify existing table data
  - Delete data from tables
  - View the data in one or more tables

Figure 1-3 shows a blank form.

## 16 Part I: Access Application Development

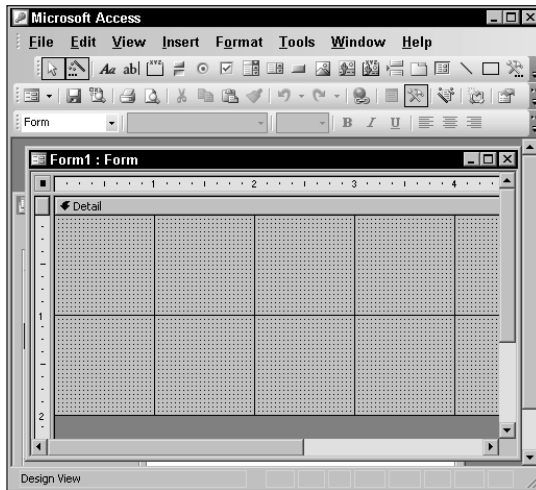


Figure 1-3: Design View of form creation.

## Creating reports

The report creation process is similar to the form creation process. The main difference is that reports are more likely to be multi-page documents, and as a result, are more likely to be divided into sections, such as report header and footer, page header and footer, detail section, and group sections. Forms can also have form and page headers and footers, but often do not. Figure 1-4 shows the Design View of a blank report. You can add controls and other objects from the toolbox to appropriate sections of the report to achieve the report that you want.

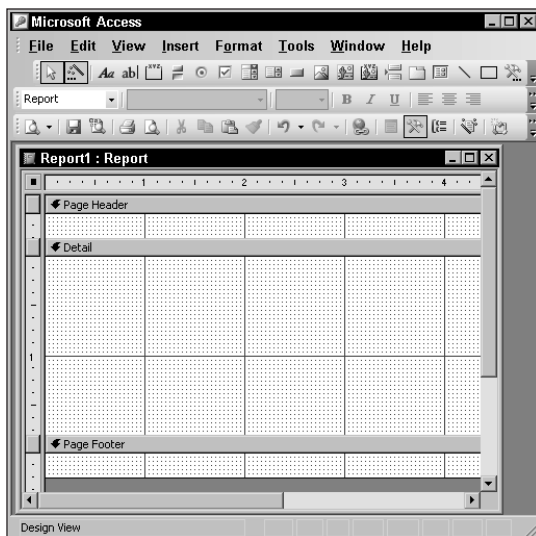


Figure 1-4: Design View of report creation.

## Creating data access pages

Data access pages, introduced in Access 2000, provide an easy way to display information on the Web that on a local machine would be contained in forms and reports. Data access pages are HTML files that are bound to data in a data source. With data access pages you can

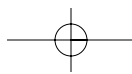
- ◆ Browse through the records in a data source
- ◆ Add, modify, and delete data
- ◆ Display the data either *sorted* or *grouped*

You can display database data on the Web in other ways, but data access pages, which are specifically designed for the task, provide an easy way to put your data online.

## Using macros

*Macros* are simple constructs for executing a sequence of operations. Before VBA became available to Access developers, macros provided the only way to automate execution. Now, there is little reason to use macros for any serious development. The many advantages of VBA make it the preferred alternative.

- ◆ VBA has these capabilities that are missing from macros:
  - Case structure
  - Loops
  - Constants
  - Variables
  - Functions
  - Transactions
  - Database objects creation
  - Graceful error handling
  - User-defined functions
  - Access Windows API functions
  - Automated operations with OLE and DDE
- ◆ Macros may be appropriate for these uses:
  - Quick prototyping of an application
  - A simple application that only runs on single-user systems
- ◆ Both VBA and macros support an *if-then-else* structure.



## Using modules

When you write code in VBA to perform operations on database data, the code is stored in *modules*. VBA is migrating toward becoming an object-oriented language, and is becoming more object-oriented with each release, although it still lacks some essential features of an object-oriented language.

In Chapter 6, we discuss modules in-depth. For now, all you need to know is that modules are the containers that hold the VBA code that you write to make your application perform the tasks that you want it to perform.

## Creating classes

In addition to the predefined objects that Access provides, you can create your own objects that apply specifically to the task at hand. You can also create classes of objects. Each class has its own particular attributes. A class can have multiple instances; each is different from another in some way, but all share the common attributes that define the class. A class's attributes are called *properties* and *methods*, which are mentioned in the next section and described in detail in Chapter 5.

## Properties and methods

In Access, data and program components are *encapsulated* into objects. In this architecture, class attributes are data items that are called *properties*, and program components are processes that operate on the object and are called *methods*.

All the objects in an object class share the same properties and methods.

## The Visual Basic Editor (VBE)

The Visual Basic Editor, known to its friends as VBE, is the primary tool Access provides you for developing VBA applications. Chapter 7 describes VBE and tells you how to use it to code the procedures that comprise your application. As you follow the examples in this book, you will become familiar with VBE and all the things it can do for you.



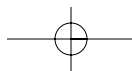
---

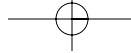
VBE is an Access application developer's most important tool.

---

## Toolbars

Access comes with over two dozen *standard* toolbars. You can also create your own *custom* toolbars, which we describe in Chapter 4.





Using the Database toolbar, you can quickly access database functions by clicking on an icon. Figure 1-5 shows the Database toolbar.



Figure 1-5: Database toolbar.

All of the Database toolbar functions deal directly with the database or with the application that you are building, based on that database. The functions on the toolbar are, from left to right:

- ◆ New
- ◆ Open
- ◆ Save
- ◆ Search
- ◆ Print
- ◆ Print Preview
- ◆ Spelling
- ◆ Cut
- ◆ Copy
- ◆ Paste
- ◆ Undo
- ◆ Office Links
- ◆ Analyze
- ◆ Code
- ◆ Microsoft Script Editor
- ◆ Relationships
- ◆ New Object Autoform
- ◆ Microsoft Access Help

The second standard toolbar type is the Task Pane toolbar. Shown in Figure 1-6, the Task Pane, usually anchored to the right edge of the Database window, presents you with the option of opening one of several recently used databases or the option of creating a new database. You may also connect to Microsoft Office Online for information or support.

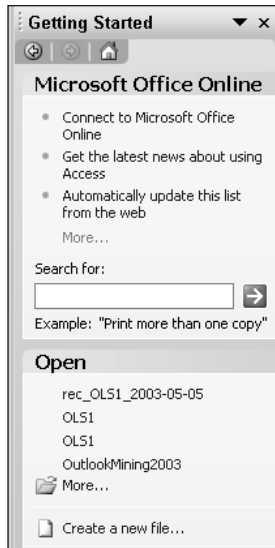


Figure 1-6: Task Pane.

The functions of the options on the Task Pane are:

- ◆ Open a file
- ◆ New
- ◆ New from existing file
- ◆ New from template

After you select a database file to work on, you can dismiss the Task Pane. It has done its job and you won't need it any more.

The Web toolbar is the third and final standard toolbar in Access. It gives you access to the most basic Web browser functions, as shown in Figure 1-7.



Figure 1-7: Web toolbar.

The functions on the Web toolbar are:

- ◆ Back
- ◆ Forward
- ◆ Stop
- ◆ Refresh

- ◆ Start Page
- ◆ Search the Web
- ◆ Favorites
- ◆ Go
- ◆ Show Only Web Toolbar
- ◆ Address

Clicking on any of the icons in the Web toolbar launches your browser and performs the indicated function. These functions are handy when you are building a database application for access over the Web.

## Access's Role in Microsoft's Strategy

Microsoft Office is a key part of Microsoft's overall strategy for providing business tools that apply to the vast majority of businesses. By continually enhancing and expanding the capabilities of Office, the folks at Microsoft have shown that they consider Access to be a major element of their future product plans. Access is one of the core components of Office. It is the tool of choice for developing business applications that involve the storage, manipulation, and retrieval of important data.

More than 600,000 Access developers worldwide are building applications based on the Access DBMS. They are operating in different environments. Some are developing applications of limited scope that are used by a small number of people, or perhaps only one person at a time. Other developers, working in large enterprises, build applications that share data among hundreds or even thousands of users. Recently, demand has grown for database applications that are available to potentially millions of users over the Web.

As the market for database products has evolved, Microsoft's strategy for Access has evolved with it. The vision Microsoft has for Access is much larger than what it was when Access 1.0 first hit the market in 1992.

### Access as a personal database management system

Access was originally designed as a personal database management system to run on personal computers, but in 1992, PCs were not nearly as powerful as they are now. Furthermore, the connection of PCs in an organization using a local area network (LAN) was not common. Access was easy to learn and use and gradually took market share from competing products, such as dBASE, R:BASE, and Paradox.

True to its heritage, the latest versions of Access are still suited for small personal database applications. However, Access has gained additional capabilities that make it a valuable tool for much larger, more complex problems.

## Access in the enterprise

With each version of Access, support for larger, more complex applications has grown. Key to this evolution is the ability of Access to handle data generated by other data sources, first through the ODBC connectivity standard, then by OLE DB, and most recently (in Access 2002), by XML and XSL.

These advances allow Access to seamlessly integrate with enterprise data sources. SQL Server, an enterprise class DBMS from Microsoft, integrates even more tightly with Access because Access can handle native-mode SQL Server data directly through the Access Data Projects feature. Applications written with an Access front end and a SQL Server back end suffer no performance degradation due to format translations.

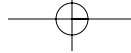
## Access and the Web

The browser-based technology pioneered on the World Wide Web is now widely used on corporate intranets. This has driven the expansion of Access's capabilities to ever-greater support of Web-based database operation:

- ◆ Access 2000 introduced data access pages, which provide a method for creating HTML pages that are bound to data and that can be displayed either over an intranet or on the Web.
- ◆ Access 2002 brought enhancements to data access pages that improve reliability and give stronger drill-down capability.
- ◆ Access 2002 supports Web-based operation in other ways:
  - You can save any Access report as a Web page by using the Save As Data Access Page feature.
  - You can save a report as an XML document. Doing so reduces demand on the database when people later want to view the report.
  - You can use the enhanced PivotTable and PivotChart capability to improve analysis and presentation capabilities. Because you can save PivotTables and PivotCharts as data access pages, the power of these analysis tools can be made available to Web-based users.
- ◆ Access 2003 does everything that Access 2002 does for Web-based operation, plus it has expanded ability to deal with XML.

## Access and .NET

.NET is Microsoft's much ballyhooed software technology for connecting dissimilar, incompatible systems, devices, data sources, and people. It uses XML Web services to connect discrete, building-block application components and complete applications via the Internet. Microsoft Windows .NET Server 2003 builds XML natively



into the operating system, giving an unprecedented level of interoperability to far-flung systems.

With its support of XML, Access 2002 enables the development of applications that can fully share data at the enterprise level and beyond. Access 2003's expanded XML support makes that data sharing even easier. It is evident that Access is a key component of Microsoft's vision of the future of enterprise computing.

## Summary

Access has been in a constant state of evolution in two directions for more than a decade. First, the continual addition of wizards and other new tools makes databases and database applications easier to create without programming. Second, the continual addition of powerful features has expanded the scope of the problems to which an Access solution can be applied. The folks at Microsoft have shown by their actions and words that they consider Access to be a key part of their overall solution to the problems of businesses, and to the problems of organizations of all kinds. Learning to program and operate with Access is an investment of time and energy that is well spent. The skills gained are relevant and marketable for the foreseeable future.

