

InternetBasic

Introduction

The best way to learn a programming language is to look at source code. You can quickly see how programs are built by studying the structure of the code, the syntax of the statements, and the relationships among the components of the language. That's the philosophy of this book and the accompanying compact disc.

This book contains source listings for many InternetBasic programs. Even more sample programs are included on the compact disc. You can study the source code, compile the programs (we've included a free InternetBasic compiler on the CD) and run them. In the process, you will see how easy it is to become an InternetBasic expert. Soon you'll be designing, compiling, and running your own Internet applications with this powerful, straightforward language. We think you'll enjoy using InternetBasic.

This book is divided into five sections:

- Part one covers BASIC applications and includes some very simple examples that demonstrate the rudiments of InternetBasic (IB). The following chapters explore many of the language's more powerful features, and are generally organized by the type of problem to be solved (e.g., printing, working with files). As a result, you will learn IB in the most meaningful context possible.
- Part two covers Internet applications, including Winsock programming and CGI programming. The samples in this section are built on the basic concepts and statements that are demonstrated in the first part of the book.
- Part three describes other practical IB applications, such as interfacing with ODBC data sources and communicating with specialized hardware (i.e., serial devices and internal fax devices).
- Part four covers the graphical user interface (GUI) programming tools that are available with InternetBasic.
- Part five describes the utility programs that are essential in managing an InternetBasic system/network, including configuration tools, file maintenance programs, status programs, program development tools, remote file transfer tools, and more. All of these utility programs are contained on the compact disc that is included with this book.

This book is meant as a tutorial. As you proceed through each chapter, you'll notice that each IB statement is explained in some depth, including information on how each statement works with other parts of the program. The sample programs build on each

other, and get more involved with each chapter. In the latter chapters, the sample programs form integrated systems that we think will be very useful to you.

Beyond this book, we invite you to visit the InternetBasic web site (www.internetbasic.com) for a complete technical reference guide.

But, before we begin describing the detailed features, we need to answer a very important question:

What is InternetBasic?

There are really two answers to this question – the simple answer and the not-so-simple one.

Let's start with the simple answer first. **InternetBasic is an application programming language meant for Internet and/or intranet applications.**

As you've no doubt guessed from looking at the name, InternetBasic has its roots in BASIC. That means you'll be seeing familiar statements such as PRINT, READ, WRITE, FOR, NEXT, IF, THEN, DO, LOOP, SELECT, CASE, etc.

Things don't stop there, however. InternetBasic offers advanced concepts and constructs such as cryptography functions (ENCRYPT, DECRYPT, BASE64, BASE256), security functions (SECURE, PASSWORD), Internet controls (MERGE, SEND, REDIR, COOKIE), gateway driver controls (i.e., ODBC driver, serial driver, fax driver), powerful file system statements (LOCK, UNLOCK, EXTRACT, INQUIRE, INSERT, REWRITE, UPDATE, DELETE), subprogram control statements (ENTER, EXIT), inter-process communications features (ACTIVATE, INTERRUPT, TERMINATE), system-level controls (LAUNCH, SHELL EXECUTE), and more.

InternetBasic offers a platform for rapid development of professional applications. In addition, IB provides a fast and efficient runtime environment for those applications. The combination of these two characteristics makes for a potent programming language.

Here's the not-so-simple answer. **InternetBasic is meant for variety of applications running in a variety of topologies**, including:

- text-based applications for both local and remote users
- graphical applications (with Windows controls and dialogs) for both local and remote users
- web-based applications for intranet and Internet users, based on the patent-pending eXtended Active Pages (XAP) technology, an advanced implementation of the Common Gateway Interface (CGI) model
- Internet protocol applications, such as SMTP e-mail
- applications that interface with ODBC data sources, such as Microsoft Access, SQL Server, Oracle, and more

- applications that interface with specialized hardware, such as a serial communications device or internal fax device

All of these applications rely on the **InternetBasic runtime system**. This software executes all of the instructions in a compiled IB program and supports up to 999 concurrent tasks. The runtime system can communicate with several user interfaces, including the **InternetBasic client** software as well as **web browsers**. The runtime system can also communicate with several data sources, including files stored on the local machine and/or files stored on a file server

To help clarify these options, look at the following illustrations. The first one shows both the IB runtime and IB client running on the same computer. This represents a development system, such as your computer. There's another way to think about this type of system. It could be described as a "single user" IB installation, where the programs, data, and operator activity all reside within one computer. This description will prove useful when we talk about scalability of IB systems (below).

(illustration)

Note: The IB client provides both text and GUI user interfaces, both of which are demonstrated in this book.

The IB client can run on a remote computer, as shown in the following illustration. In this configuration, the IC client communicates with the IB runtime via TCP/IP.

(illustration)

Since IB runtime supports up to 999 tasks, multiple remote users can run IB programs from a single "host" system. In this sense, the IB host computer is an **Internet application server**.

The IB runtime can also communicate with a web browser. The following illustration shows the simple case, where the browser is running on the same computer as IB runtime. As with above, this might represent a development environment or a single user installation.

(illustration)

The logical extension of this is where IB runtime executes on a host system, while the web browser runs on a remote system, with communications being done via TCP/IP. Since the IB runtime supports multiple tasks, the host system functions as a **web application server**.

Beyond the IB client and web browsers, IB includes instructions for general-purpose Internet communications using virtually any standard protocol (if you know the protocol,

you can write an IB program that uses it). For example, the following illustration shows how an IB host computer can send e-mail via an SMTP server.

(illustration)

As mentioned above, the IB runtime can communicate with several data sources. When you start to explore the sample IB programs, you will be using the language's local data base system, where all data files are stored on the local hard drive.

However, IB is capable of accessing data that is stored on a file server. There are two ways to do this, one where no server-based software is running and one where some is. The first method provides access to data files on a server, but is limited in terms of efficiency and lacks some important features (such as the ability to lock an individual record in a data base). The second method involves the **InternetBasic Data Base Engine**, extremely fast data access software that runs on a file server. This engine handles all of the runtime system's requests for IB data on the server, which provides true client-server architecture for IB systems. This is shown in the following illustration.

(illustration)

Note: No programming changes are required to switch from IB's local data base system to the IB Data Base Engine. The only change is a configuration one, which is made by editing the IB configuration file (see Chapter ??? for more information).

In addition to the above, IB contains programming statements designed for the express purpose of communicating with data sources based on the Open DataBase Connectivity (ODBC) standard. These include data sources such as Microsoft Access, SQL Server, Oracle, Microsoft Excel, and more.

(illustration)

IB also includes instructions for communicating with specialty hardware. As mentioned above, the IB runtime can interface with an internal fax device. This means you can write IB applications to send faxes and monitor the fax status (with such applications being linked to your IB data base or ODBC data source). Also, IB can interface with any serial device connected to the host computer. Potential applications include electronic postage scales, point-of-sale systems, electronic control systems, and much more.

(illustration)

Scalability

It is very easy to scale an InternetBasic system. Right now, you own a single user IB system (client, runtime and data are all within one computer). Soon, you may want to deploy IB programs in a local area network and store IB data on a file server. Your

system might have multiple IB client/host computers connected to a server that's running the ID Data Base Engine.

(illustration)

Perhaps you might deploy some remote IB clients. Whether these remote users are employees of your own organization (intranet users) or are customers/vendors with whom you do business on a regular basis (extranet users), they can easily sign on to your IB host computer (application server) and gain access to IB programs and data. The programs look and run the same, whether they are run locally or remotely. Multiple security levels and data encryption are provided in the IB runtime system and via IB utility programs (see Chapter ??? for more information). This gives you a very easy way to build a business-to-business e-commerce system.

(illustration)

Next you might establish one of your IB host computers as a web application server using XAP technology. Thus, Internet web browsers can run web-based IB programs on your IB host, while accessing data from your IB-compatible data sources. The IB host receives data from a web browser (an HTML form), process that data, and send a customized web page back to the browser. Your imagination is the limit when it comes to building web-based applications using IB.

(illustration)

Regardless of the topology you choose, the IB code you write for one system is the same code you'll use for a large network. Likewise for the data source(s) you choose. This makes the best use of the time you will spend developing applications in InternetBasic.

Scalability Isn't Free, However

The compact disc that accompanies this book includes a free copy of the InternetBasic compiler. It also includes a free version of the IB runtime system, a free IB client, and a collection of very helpful (and free) IB utility programs. That's a lot of free software, and certainly everything you will need to start building IB applications.

There's only one catch. This free software may only be used within one computer. That means that the IB runtime, IB client, and data files must all be local (in the previous terminology, you have a "single user" system). The reason is simple: the developer of InternetBasic, Signature Systems, Inc., sells software licenses for users who want to scale their IB systems beyond one computer.

For example, if you want to use the InternetBasic Data Base Engine in order to access data files on a file server, you can purchase a software license for that product. If you want to deploy an intranet/extranet with remote IB clients access your IB host system,

you can purchase a license for that capability. Likewise for remote XAP users and non-local ODBC data sources.

For information on how to purchase these licenses, please visit the InternetBasic web site (www.internetbasic.com). Meanwhile, enjoy reading this book and learning a new programming language.