

# *Amateur Radio Software: It Keeps Getting Better*

---

*The tools to develop Amateur Radio  
applications steadily improve.*

---

By Stephen J. Gradijan, WB5KIA

**M**y first effort in programming software for Amateur Radio was a program written 20 years ago in *BASIC*. It provided accurate calculations but it was not very pretty, as its graphic capabilities were limited. Today, amateurs are creating commercial and near-commercial-quality Amateur Radio software to do all kinds of things and are making it free to the amateur community. Not all hams programming are professional programmers. High-level programming is easier to do than ever, and personal computers are permitting execution of mathematical routines that would have brought the PCs of 10 years ago to their knees.

Sophisticated graphics are within the capabilities of the novice programmer. Many ham-developed computer programs are available at no cost to the amateur—they track satellites, log contacts, predict radio propagation and support every digital mode available. Until recently, ham-oriented freeware and shareware could be found on the Internet, but without its

source code (the code required to modify the program). Most of the programs came with only the executable code. State-of-the-art programming-code examples of ham-radio processes, with a few exceptions, were difficult to find. Today, some amateurs are freely sharing source code with others and even sharing the program-development experience.

The purposes of this article are to show that it is relatively easy to write code with modern programming packages and point out some options for obtaining and using software for your projects.

Internet sites like N1MM's logging project, WA0TTN and AE4YJ's pages for development of PSK, SV2AGW's AGWPacket Engine site for development of packet communications and JE3HHT's MMTTY RTTY pages feature ham programming at its best. Such sites let radio amateurs, worldwide, get involved with the program development, have access to the source code or to a control or DLL to use with your program. Most of what is available is free of charge and usually subject to very generous fair-use terms. Information on the sites describes how the material may be used. The source

material available at these sites can shorten the program-development time of your project or help a beginning programmer discover how to do something that otherwise might seem impossible.

## **Software on the Web**

The *N1MM Logger* (Fig 1) Web site is at **n1mm.com**. On the N1MM Yahoo Discussion Group, one can request new features to the current version of the versatile N1MM logging program and discuss problems (called bugs) with the current logging program. Several hams help Tom Walker, the program's principal developer, develop new code and documentation. Tom and his crew are a cooperative development venture, programming in *Visual Basic 6*. *N1MM Logger* uses Microsoft Access files for the logging database. However, most of the code can be read using a simple text editor such as *Notepad*, which comes with the various versions of Windows. Routines can be used in various projects that you might program in another version of *Visual Basic* or that could be translated into another language.

JE3HHT has made the dynamic-link library he programmed for

---

1902 Middle Glen Dr  
Carrollton, TX 75007  
**wb5kia@arrl.net**

MMTTY available for use by radio amateurs. The MMTTY page is at [www.qsl.net/mmhamsoft/mmtty/](http://www.qsl.net/mmhamsoft/mmtty/). It has a copy of the RTTY engine and examples for Visual Basic 6 and Borland C++ developers. Dynamic link libraries (DLLs) are program modules that contain code, data or resources that can be shared among many Windows applications. They can be thought of as compact programs that can be accessed by various Windows programs to provide specific functions. Makato's DLL allows others to incorporate his RTTY engine in their own programs.

Moe Wheatley, AE4YJ, has made his PSK Core DLL available for amateur use. Dave Cook, WA0TTN, took Moe's idea farther and created a WinPSK ActiveX control (Fig 2) that can be used with a variety of programming platforms. Moe has also made available the original code on which the PSK Core DLL is based. His links to a Visual Basic 6 demonstration by Eric Sundstrup, VK7AAB, have disappeared but a Delphi demonstration by Julian Moss, G4ILO, is still linked at his site [www.qsl.net/ae4jy/pskcoredll.htm](http://www.qsl.net/ae4jy/pskcoredll.htm). G4ILO has his own site, [www.qsl.net/g4ilo/main.html](http://www.qsl.net/g4ilo/main.html) with additional code and programs. WA0TTN's site is [www.netdave.com/wa0ttt](http://www.netdave.com/wa0ttt). The effort that went into programming the DLL and ActiveX control was awesome. All these tools are free for amateur use.

You can write your own programs too. It takes time to become familiar with the programming languages, but once you get started, programming can be fun. Hamming and computer programming do go together!

## History

For decades, radio amateurs have made use of computers to enhance their operating with programs to calculate engineering values, log contacts and keep track of awards and QSLs, operate packet, RTTY or other digital modes, predict radio propagation conditions and track Amateur Radio satellites.

Efforts in the 1960s involved main-frame computers, the FORTRAN language and punch cards, but few hams were able to get access to the institutional computers or do the necessary work. Beginning in the 1970s and with the advent of personal computers and the spread of the BASIC language in its various forms, hams began to program in earnest.

It was practical in the early days to publish BASIC code listings in magazines. In 1981, Tom Clark, W3IWI, wrote a satellite tracking program described in an article called

"BASIC Orbits" (Orbit, March/April 1981, pp 10-11, 19-20, 29). It was very popular and became the basis for many later Windows programs. QST contained the MINIMUF propagation program and the Super Duper logging program in 1982 and 1985, respectively. Keyboarding of long programs

was a chore and error-prone. After 1986, program listings disappeared from most journals, as shareware and freeware programs became available on telephone and packet bulletin-board systems (BBSs), disks and CD ROMs. This was a huge improvement in the ability to share software code.

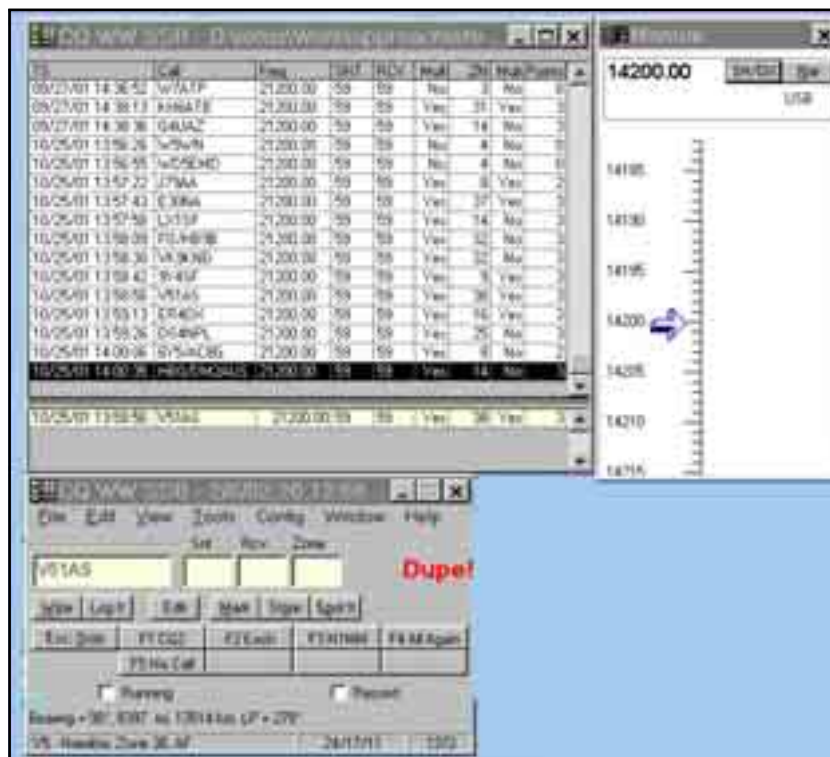


Fig 1—The N1MM Logger is being developed by ham users through exchange of comments and ideas at an Internet discussion Web site. Visual Basic 6 is the main tool for development of the project.



Fig 2—WA0TTN's documentation for the WinPSK ActiveX control he wrote based on AE4JY's PSK Core DLL is very thorough. It works with Visual Basic, Delphi, C++ and such.

**BASIC** was the primary tool for nonprofessional programmers. Professional programmers used other high-level languages like Pascal and FORTRAN (in its various forms) on PCs, along with assembly language.

Early Windows-environment programmers used C and C++ followed by *Visual Basic* in about 1991. It was very difficult for the amateur to write a program for Windows. *Visual Basic* changed that as it became relatively easy to program, so Windows-based ham programs began to appear. It is difficult to provide Windows program listings because it is impractical to print a listing of the graphical properties that are attached to specific controls. Consequently it became difficult to print the source code listings in magazines.

Since 1995 and the creation of *Windows 95*, tools like *Visual Basic* and *Delphi* have created 32-bit programs that can execute at lightning speed on today's fast PCs.

### My Experience with Programming Projects

I have developed numerous programs through the years for my personal enjoyment using interpreted **BASIC**, Computer Associates *Realizer*, *Visual Basic*, C++, *Delphi 1* and *Delphi 5*. Today, I program almost exclusively with *Delphi 5 Professional*.

Many of my DOS and Windows programs were near commercial quality at the time they were written, but many were only partially fleshed out or documented. I knew how to run them and did not require a polished finished product, but only the functionality. Similar programs were available commercially but my own programs cost me only my time and resulted in a better understanding of the basis for the programs (their underlying mathematics and principles that allowed execution).

I programmed antenna pointing, satellite tracking and logging programs, a program to control my PK-232 Multi-mode controller, Technician and General Class License study guides, and so forth. My most recent effort is a PSK program with an attached logger (see Fig 3). Although I may not be typical, I believe it is possible for you to write programs for yourself too!

*KIApsk Logger* is a PSK program I developed recently because the transmit and receive frequencies of my aging ICOM IC-740 are off by 16 Hz (or there may be a problem with input versus output frequencies of my clone sound card on my six-year-old computer). Using any of the excellent available PSK software programs, I

constantly received complaints of slightly off-frequency operation or "you must have left your RIT control on." I do not get such reports any more.

*KIApsk Logger* uses the *WinPSK* ActiveX control (copyrighted in 2001 by Dave Cook, WA0TTN and Moe Wheatley, AE4JY) based on the *PSKCore* DLL developed by Moe Wheatley. The ActiveX control incorpo-

rates Moe Wheatley's DLL code. It provides a convenient COM object for developing PSK31 applications in Microsoft *Visual Basic*, *Delphi* and C++. The ActiveX control provides a visual display of the common data viewing modes of spectrum, waterfall, input and data sync. The program is similar to most of the available PSK programs today; the difference is that I coded it



Fig 3—This is the main screen of the *KIApsk Logger* that was programmed with *Delphi 5* and using the ActiveX control by WA0TTN described in the text.

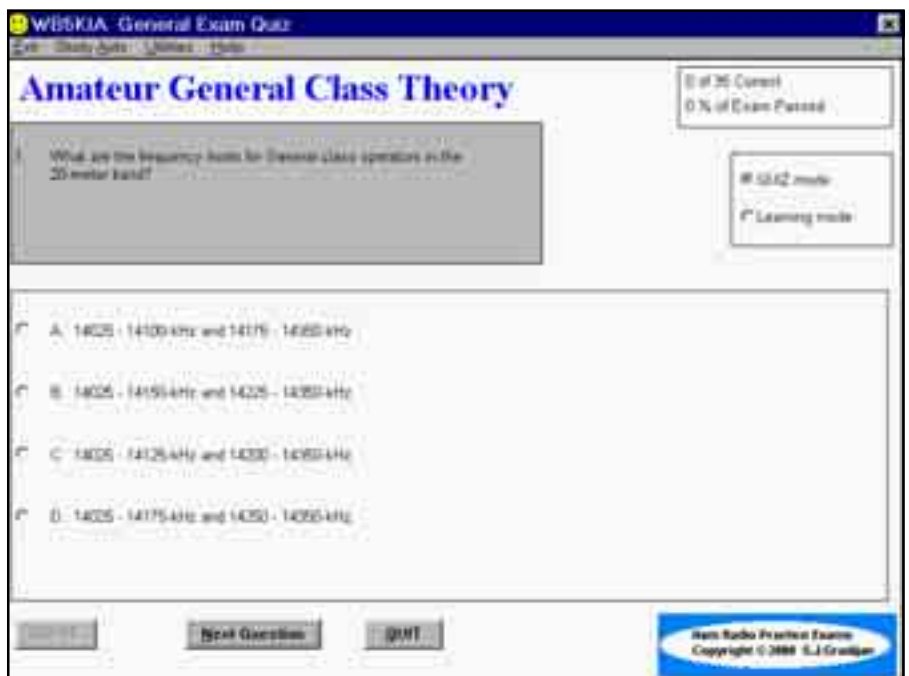


Fig 4—A General class license quiz generator and study guide was easily programmed with *Realizer*.



and it does what I want it to.

My son Francis, KD5HTB, found it relatively easy to pass his Technician Plus and General exams using software learning programs I developed using the ARRL/VEC question pool. I created a trial exam generating program and study guide using *Realizer* (see Fig 4). This is one of the simplest types of program. Plenty of material was available on the Internet with source files showing various ways of creating generic test generating programs for educators. Potential program developers of educational ham programs have a wealth of example information and code available.

Putting together various pieces of code can result in new and useful programs. You might need to modify the code for your version of the language or even for a different programming environment. For several years, I used a propagation forecast/beam-heading program based on a modification of the *BASIC* code in the *MINIMUF QST* article, code modified from various other magazines and independently developed code (see Fig 5). The initial code came from the article; the rest came through curiosity and a desire to have additional features.

## Programming Tools

What is needed to get a start in programming for Amateur Radio may be as simple as acquiring a copy of *QBasic*, *QuickBasic*, *PowerBasic*, *Visual Basic* or *Turbo Pascal* for DOS programming. *BASIC* interpreters come in various forms such as *GWBasic*, *BasicA* and so on. Microsoft *Qbasic* is an interpreted version of its compiled *QuickBasic*. The main difference is that *Qbasic* requires the presence of the *Qbasic* interpreter to run the program code. *QuickBasic* can create an executable file that is self-contained. It also has more capabilities. *PowerBasic* by Powerbasic is offering similar to *QuickBasic*. The original *Visual Basic* was a DOS programming tool with expanded graphics capabilities. *Turbo Pascal* is a Borland product using the Pascal programming language. Pascal and *BASIC* have both similarities and differences. Noncommercial versions of both the original *BASIC* and Pascal languages have been developed and can be found for downloading on the Internet. While interpreters can be used to code complex problems, they probably should be avoided today because everyone using the program has to have a copy of the interpreter that was used to develop it. On the other hand, the actual *BASIC* program is readable by anyone who has a copy of *BASIC* or a text editor to read the code. Compiled languages are pre-

ferred because they create an executable file and process code faster.

Although some versions of the DOS programming languages have enhanced graphics abilities, DOS programming for ham radio purposes is limited to text output and very simple graphic displays (Fig 6). Most programs that you would code using DOS will run under Windows in a DOS Window, so do not exclude the DOS compiled languages just because they are old technology. If you do not require lightning speed or fancy graphics, you

should be quite happy with DOS.

However, Windows programming may be easier to learn. This was not always true. Early Windows programs were coded in *C* and *C++*; development was difficult and slow. The visual programming tools developed for *Windows 3.1* that made programming simpler include *Visual Basic 3*, *Delphi 1* and *Realizer*. These languages are based on the concept of visual objects. Components that provide basic functions can be dragged onto a form in the programming environment. One can modify

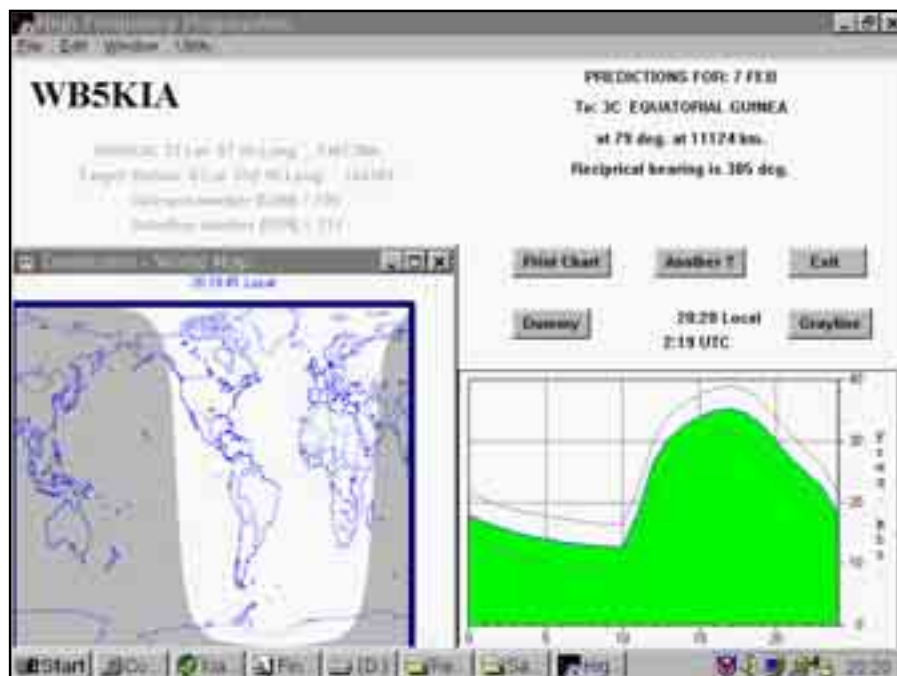


Fig 5—This highly updated and improved *MINIMUF* HF propagation program was coded with *Realizer*.

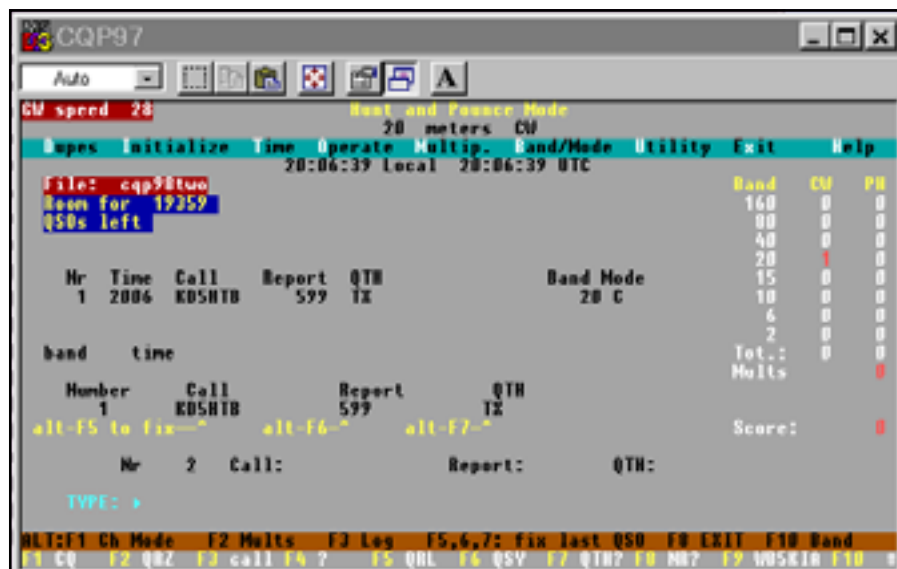


Fig 6—A DOS contest logger running in a Windows DOS window was programmed with *QuickBasic 4.5*.

their characteristics from a table and avoid detailed coding for many usual tasks. *Visual Basic 3* and *Realizer* are based on the *BASIC* language. *Delphi 1* is based on Pascal.

Tools developed for *Windows 95* and later operating systems running on 32-bit processors include *Visual Basic 4-6*, *Delphi 2-6* and *Visual C++*. The 16-bit tools can run on 32-bit processors but cannot take advantage of all the increases in processing speed and features of the modern processors and operating systems. Both 16-bit and 32-bit programming systems have enhanced graphics capabilities; they can be used to produce very good-looking programs (see Fig 7).

Tools for Apple users are more difficult to come by and are beyond the scope of this article. A *BASIC* language was available at one time for Apple users. *Linux* users might consider *Kylix* by Borland. It is a *Delphi*-like clone developed by Borland for the *Linux* operating system.

Table 1 describes some of the tools that you might use for your programming activities and a few can be seen in Figs 8-10.

These programming tools may be available free of charge or used at reasonable prices, especially if the software is designed for 16-bit Windows or DOS programming. Check the terms of the individual software license. Licenses may be transferred in most instances if the media is ex-

changed and the programs have been removed from the original owner's computers. Copies of *Delphi 1* were made available at no charge in the United Kingdom a few years ago but with the stipulation that programs generated were not for distribution, that is, only for personal use. Copies of *Turbo Pascal* and *Turbo C++* are available free for personal use from the Borland Museum ([www.community.Borland.com](http://www.community.Borland.com)) if you register. Whatever programming language you choose or find available, make sure the software license terms allow you to either sell or give away the resulting program before you start sharing your programs.

Used bookstores frequently have copies of software available in its original packaging at discounted prices. Various E-vendors have last year's version of some programming software at discounted or reduced prices. Recently, I have found "Professional" editions of *Visual Basic 4* at under \$50 and of *Visual Basic 3* and *Realizer* under \$20 in a national used-book chain. They were sealed in their original packaging. New "starter" versions of the latest editions of *Visual Basic* and *Delphi* are available for around \$100. These are very usable versions of the larger professional or enterprise programs but without all of their parent-program features. The license terms vary with the various versions but most do not permit commercial distribution of the devel-

oped programs. The good news is that it is possible to upgrade to the full version of many of these products after you have developed confidence in your programming skills. Also, free or shareware tools are available on the Internet that augment the functionality of these cut-down versions of the professional software packages.

Table 2 is a compilation of Internet sites that have information and resources that can help you with your programming experience. The Internet site addresses frequently change, so be resourceful if they do. If you cannot find what is described at the indicated sites, use a search engine to find what you need.

What is the best language for you? Most of the DOS-based languages are very easy to learn and may be practical if you are comfortable with DOS or have an ancient machine. DOS programs, with a few exceptions, do run on computers having a Windows operating system. Although it is possible to use graphics, it is not as easy as with the Windows programming tools. DOS is fine for "number crunching" applications but consider the arithmetic precision possible with the language and/or program you select.

Windows programming tools come in several flavors. Some are more user friendly to beginners than others. Some tools provide executable files that have a significant speed advantage. Programs designed with 16-bit pro-

**Table 1—Some Programming Tools**

**For DOS**

Language	Description	Source	Type	Ease of Use
<i>GWBasic*</i>	Simple <i>BASIC</i>	Microsoft	Interpreter	Easy
<i>Qbasic*</i>	Advanced <i>BASIC</i>	Windows disks	Interpreter	Easy
<i>QuickBasic*</i>	Structured <i>BASIC</i>	Microsoft	Compiler	Relatively Easy
<i>PowerBasic</i>	Structured <i>BASIC</i>	PowerBASIC Inc	Compiler	Relatively Easy
<i>Turbo Pascal*</i>	Pascal language	Borland	Compiler	Intermediate difficulty

**For Microsoft Windows**

<i>Visual Basic</i>	Object-oriented structured <i>BASIC</i>	Microsoft	Compiler with runtime DLL	Relatively Easy
<i>Realizer*</i>	Object-oriented structured <i>BASIC</i>	Computer Associates	Compiler with runtime DLL	Easy
<i>Delphi 1</i>	16-bit Visual Pascal	Borland	Compiler	Relatively Easy
<i>Delphi 2-6</i>	32-bit Visual Pascal	Borland	Compiler	Relatively Easy
<i>C++</i>	various flavors	Borland or Microsoft	Compiler	Difficult
<i>Visual C++</i>	various flavors	Borland or Microsoft	Compiler	Relatively Difficult

**Linux**

<i>Kylix</i>	32-bit Visual Pascal	Borland	Compiler	Relatively Easy
<i>Visual C++</i>		Various	Compiler	Difficult

\*These languages are no longer supported by the companies that originated them, but there is some community support on the Web. They are no longer sold at retail.

**Table 2—Internet Sites with Programming Information and/or Source Code**

**General Amateur Radio Resources**

N1MM Logger

**www.n1mm.com**

Development of contest logging program. Specific source code is available by request. A great site/discussion group

WA0TTN Web page

**www.netdave.com/wa0ttt**

PSK ActiveX control for *Visual Basic*, *Delphi* and *C++*, other controls to make developing PSK programs easier. A remarkable free tool for working with a homemade PSK program

PSK Core DLL

**www.qsl.net/ae4jy/pskcoredll.htm**

PSK Core DLL and example code for *Delphi*. Free DLL works fine.

MMTTY

**www.qsl.net/mmhamsoft**

RTTY DLL by Makoto Mori, JE3HHT—amazing RTTY

MMTTY Programmer's Page

**www.qsl.net/mmhamsoft/programmer/p-download.htm**

Examples of how to use the DLL

SV2AGW Sound-Card Packet

**www.elcom.gr/sv2agw/agwsc.htm**

Packet engine (Packet without TNC hardware)—difficult to use.

HB9JNX/AE4WA Multplatform Sound-Card Packet

**www.baycom.org/~tom/ham/soundmodem**

Another sound-card packet implementation—untested.

Official PSK Web site

**www.aintel.bi.ehu.es/psk31.html**

Code snippets and partial program showing how to use AE4JY's PSK Core DLL with *Visual Basic* and *Delphi*. Many examples of quality programs coded by hams

DX Atlas (shareware)

**www.dxatlas.com**

Advice on how to interface your project to the DX Atlas via COM or OLE automation is in the program's help file. Two *Delphi 5* example programs with code. Example of techniques to interface your project with another project. One of the numerous shareware/ commercial sites that provide information for developers to interface with their products.

Ten-Tec Programers' Page

**tentec.com/rfsquared**

Follow the *update* links to the *Ten-Tec Programmer's Reference Guide* for programming Pegasus and Jupiter radios. General public license source code (Microsoft *Visual C++*, 16-bit) for control of Pegasus/Jupiter and reference guide.

ICOM America

**icomamerica.com**

Kenwood\*

**Kenwood.net/ama\_page.cfm**

In downloads section, "Software" has several free rig-memory programming applications. Amateur/RCPSoftware section has a free TS-570 control program that also works with many TS-2000 features.

Yaesu\*

**Yaesu.com/amateur/amateur.html** or **soft.html**

Only commercial software listed.

The Plicht Brothers

**www.plicht.de/ekki/**

*Delphi* code snippets for ICOM CT-17 level control by DF4OR. Lots of information about controlling radios with PCs in the software and ICOM CI-V sections.

AA6YQ / Ambersoft

**Ambersoft.com/Amateur\_Radio/Index.htm**

*Visual Basic* code fragments for programmers controlling radios with PCs

Commander

**www.qsl.net/civ\_commander**

Free radio frequency-control program for ICOM, Kenwood and Yaesu. Those developing this program solicit your input

\*There is no online developer support from any of these manufacturers.

**Programming Resources**

Mapping site

**www.versamap.com/webdoc03.htm**

CIA public domain map of the world in digital form

Voice of America/Department of Commerce

**elbert.its.bldrdoc.gov/**

*VOCAP*—predicts performance of HF broadcast systems. May no longer be available, contains both *Visual Basic* and *BASIC* code.

Intel

**www.intel.com/software/products/perflib/index.htm**

Intel signal-processing library—free signal processing DLLs, controls and so on. Examples with *Delphi* and *Visual Basic*

CIA—The World Fact Book

**www.cia.gov/cia/publications/factbook/**

Amazing public domain data regarding countries of the world including maps.

CIA World DataBank II

**www.evl.uic.edu/pape/data/Earth/**

Public domain information about countries of the world, up-to-date map information. Dave Pape has digital map data at different resolutions and bitmaps for download.

QRZ.com  
[www.qrz.com](http://www.qrz.com)

Specifications for format of Keplerian elements in file `kep_fmt.txt`. Explanation of satellite orbital elements available from NASA, AMSAT and so on.

### Delphi Resources

Yahoo Discussion Forum

**Yahoo.com**

Delphi programming forum, general code examples and answers to programming questions

Borland

**Borland.com**

General code examples and downloads for *Delphi* and *C++ Builder*. The main *Delphi* site. A free download of *Delphi 6 Standard* might still be available on the site. It requires registration. There are restrictions on the use of programs developed with the free download.

Delphi Super Page

**delphi.icm.edu.pl/**

Freeware, shareware controls and code examples.

Torry's *Delphi* Pages

**www.torry.net/**

Freeware, shareware controls and code examples.

Efg's Reference Library *Delphi*

**www.efg2.com/lab/library/index.html**

Extensive source for programming code for graphics, math routines and such. Numerous links and code for various programming platforms.

### General Visual Basic Resources

Visual Basic World

**www.vb-world.net**

Source code shareware, freeware, tutorials

### General QBasic, Quick Basic PowerBasic Resources

ABC Basic

**www.allbasiccode.com**

Over 2000 pieces of free source code—lots of example code and complete nonham projects.

Qbasic.com

**www.qbasic.com**

300 program examples (not associated with Microsoft)

## Table 3—Suggested Reading

### Beginners

G. Perry, *The Complete Idiot's Guide to Qbasic*, (Indianapolis, Indiana: Alpha Books, 1994).

N. Rubenking, *Delphi Programming for Dummies*, (New York: IDG Books Worldwide, 1995).

D. Stivison, *Introduction to Turbo Pascal*, (Alameda, California: Sybex, 1987).

B. Watson, *Delphi By Example*, (New York: Que, 1995).

K. Reisdorph, *Sams' Teach Yourself Borland C++ Builder 3 in 21 Days*, (Indianapolis, Indiana: Sams Publishing, 1998).

### Advanced

M. Cantu', *Mastering Delphi 4*, (Alameda, California: Sybex, 1998).

S. Teixeira and X. Pacheco, *Delphi 5 Developer's Guide*, (Indianapolis, Indiana: Sams Publishing, 2000).

M. Waite and others, *Microsoft QuickBasic Bible*, (Redmond, Washington: Microsoft Press, 1990).

C. Calvert, *Charlie Calvert's Borland C++ Builder*, (Indianapolis, Indiana: Sams Publishing, 1997).

gramming tools (*Visual Basic 3*, *Delphi 1*, *Realizer*) execute more slowly than those designed with 32-bit tools (*Visual Basic 4-6*, *Delphi 2-6* and such).

I have learned to program C++ and *Visual C++*, but I find that it takes a special kind of person to become proficient with those languages. Generally, development time is longer with these tools than with *Visual Basic* or *Delphi* but project execution, if you followed best programming practices, is faster, especially execution of mathematical routines. *Delphi 5* and *Delphi 6*, in many programming situations are almost as fast as C++ implementations versus the much slower *Visual Basic*.

My personal preference is the *Delphi 5 Professional* that I acquired

new about a year ago. My learning curve was very steep. Programming in DOS is still practical and enjoyable, but for the time employed and for a few more dollars to buy the programming software, you should be programming Windows. If you do not like *Delphi*, try *Visual Basic*, and if you are a perfectionist try *Visual C++*, but be prepared to work hard.

### Start to Program

"Just do it!" exhorts a sign in the office of one of my friends. Depending on what tool one is using, this may work for most of you. For the rest, learning to program is not necessarily any more difficult than getting your amateur license provided you arm yourself with

some of the programmers' study guides. If you can find them, several beginning books might be useful. "Suggested Reading" contains some you might wish to consider.

Some of these books are quite expensive, but it is possible to find earlier, used or new editions in secondhand bookstores. Many of these include a CD-ROM with coding examples. Your local library may be an excellent resource. If the book covers a version not too far removed from the language version you are using, a book purchase might be worthwhile.

While programming today is less difficult than it once was, most programs contain hundreds of lines of code that someone else spent considerable time to develop. While it does take some skill to program, modifying existing programs can be considerably easier than developing a new program. That is when the material available at Websites like those made available by N1MM, WA0TTN, AE4JY and others become important to beginning and accomplished software programmers.

Other sites provide full source code (see Table 2). If you are lucky, the code will be usable directly with your programming software. If it is not compatible with the language you are using, aptitude in several programming languages may permit you to translate what you can find into the form you require. Sites on the Internet may

help you with translations.

If you find a program that interests you, look at the source code and compile it using your programming software. If the program runs, you are in luck. If not, the source code might have been programmed in an earlier or later version of the software you have. Do not give up hope. Most probably, only small changes are needed to get the source code to compile with your compiler. Some compilers, like *Delphi*, give you hints as to what the difficulty might be.

I am a great believer in learning by example. There are lots of ham radio source code examples out there—and even more general examples. Hams do a lot to help one another. If you decide to write your own programs, you will find that the programming fraternity is similar; however, many programmers rightfully consider their code proprietary. When you use someone else's code, give credit where it is due. If it is copyrighted, follow the terms of the license agreement. If you have questions about how to do something program-

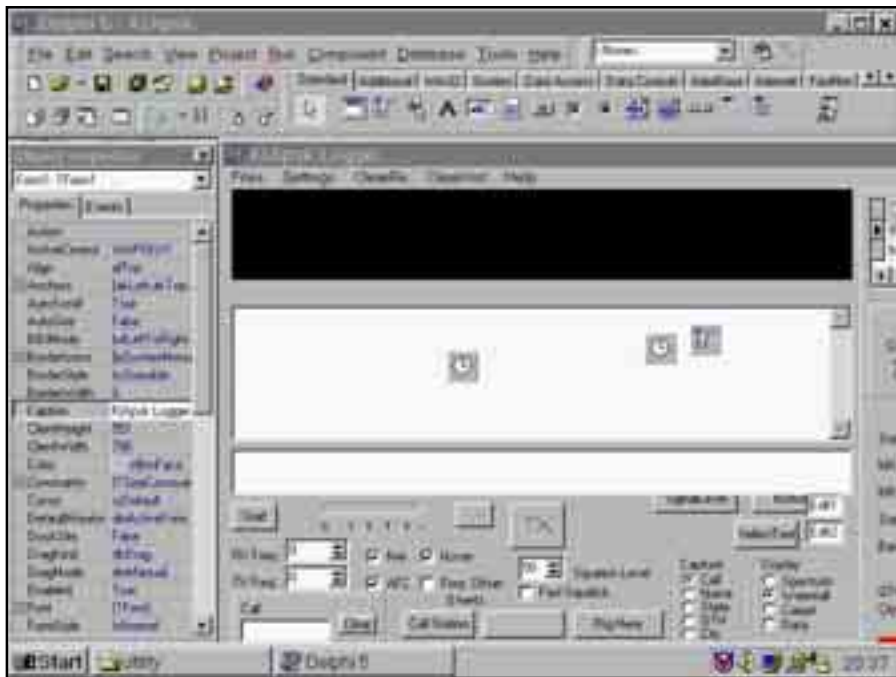


Fig 7—Delphi 5 design screen showing *K1Apsk Logger* development in progress. *Delphi* uses visual objects and controls to simplify programming tasks.

#### Table 4—Code Sources

##### Radio Propagation

- R. Rose, K6GKU, "MINIMUF: A Simplified MUF-Prediction Program for Microcomputers," *QST*, Dec 1982, pp 36-38, 43. This is a very simple HF propagation prediction program.
  - J. Priedigkeit, W6ZGN, "A Simple Computer Model for VHF/UHF Propagation," *QST*, July 1983, pp 32-33.
  - T. Frenaye, K1KI, "The KI Edge," *QST*, Jun 1984, pp 54-56. Discusses the gray-line and provides mathematical information for calculation.
  - K. Arneberg, LA9YF, "Beregning av soloppgang og solnedgang," *Amatorradio*, Mars 1985, pp 75-76, *BASIC* code for sunrise/sunset times (Norwegian Radio Relay League Journal).
- Source code for *VOACAP*, the propagation program developed by NTIA/ITS for the Voice of America.

##### Satellites

- T. Clark, W3IWI, "BASIC Orbits," *Orbit*, Mar/Apr 1981, pp 10-11, 19-20, 29. Article contains a *BASIC* listing of a satellite-tracking program.
- I. Jefferson, G4IXT, "Tracking Satellites with a Microcomputer," *Wireless World*, Apr 1983. Describes the mathematics of satellite tracking. A *BASIC* listing was available from the magazine.

##### Finding Directions

- Svein, LA6PV, "Avstandsberegning," *Amatorradio*, Mars 1985, pp 93-94. The article describes distances, bearings and grid-square calculation program in *BASIC* (Norwegian Radio Relay League Journal).

##### Logging Programs

- J. Hess, W9KTP, "The Would-be Contest Killer," *QST*, Oct 1983, pp 20-22. Article contains a *BASIC* program listing.
- S. Horzempa, WA1LOU, "BASIC Duping," *QST*, Dec 1982, p 74. Article contains a *BASIC* "hash table" listing.
- R. Cheek, W3VT, "LOGPROG—A DXer's Log in *BASIC*," *QST*, Sep 1984, pp 24-29. Article contains a *BASIC* logging program listing.
- G. Allison, K5IJ, "The Super Duper," *QST*, Pt 1, Sep 1985, pp 27-30; Nov 1985, pp 44-50. Learn *BASIC* language programming techniques while designing a contest duping and logging program.

- R. Keller, K3PCS, "Super-Double Bubble," (Technical Correspondence) *QST*, Dec 1985, pp 52-53. Article contains another bubble sort for the Super Duper.
  - P. Wisiolek, K1TKL, "Super-Double Bubble," (Technical Correspondence) *QST*, Dec 1985, p 53. Still another bubble sort for the Super Duper.
  - G. Schulz, WB9NDM, "Super Duper POOP," (Technical Correspondence) *QST*, Mar 1986, p 46. Article contains suggested changes to the Super Duper listing.
  - J. Scott, KA8FSM, "Super Duper Printer," (Technical Correspondence) *QST*, Apr 1986, pp 41-42. Article contains a *BASIC* listing to print log results.
  - T. Karnauskas, N9BWY, "Better Sort," (Technical Correspondence) *QST*, Aug 1986, p 40. Article contains a better sort routine.
- "The Cabrillo File Format," *QST*, Nov 1999, p 102. Article shows the file format ARRL requires for submission of digital contest logs.

##### CW Sending

- D. Whipkey, N3DN, "A Keyboard Keyer and Code-Practice System," *QST*, Jan 1984, pp 13-16. Article contains a *BASIC* program listing and machine-language routine for a Commodore VIC 20.
- R. Schetgen, KU7G, "C 64 Keyboard," *QST*, May 1984, p 45. Article adapts "A Keyboard Keyer and Code-Practice System" to work on a Commodore C64 computer.

##### Technical

- C. MacKeand, WA3ZKZ, "The Smith Chart in *BASIC*," *QST*, Nov 1984, pp 28-31. Article contains a listing in *BASIC*.

##### Radio Control and Information Display

- B. Wood, W0DZ, "The Return of the Slide Rule Dial," *QST*, Feb 2002, pp 33-35. Article contains a code snippet from the *Visual Basic* project. Source code and the executable are free by e-mail ([w0dz@arrl.net](mailto:w0dz@arrl.net)).
- AA6YQ Web site, *Visual Basic* code fragments for ICOM radio control ([www.ambersoft.com/Amateur\\_Radio/index.htm](http://www.ambersoft.com/Amateur_Radio/index.htm)).  
The Plickt Brothers Website, Ekkis, DF4OR has *Delphi* code fragments for ICOM and others. ([www.plickt.de/ekki/](http://www.plickt.de/ekki/)).

##### Amateur Radio Education

The ARRL/VEC exam question pool is available from the ARRL ([www.arrl.org](http://www.arrl.org)) as a text or PDF file and is useful although not a program listing.



matically, someone might have an answer on the Internet forum devoted to your particular programming language.

**BASIC** code listings for various Amateur Radio activities are described in the literature and sources (oldest to youngest) described in "Code Sources." These are useful if you are programming in *BASIC* or *Visual Basic* because they can be used in a more modern program with some rewriting. The algorithms can also be adapted for other languages.

Program listings are available in books by the *ARRL*, *CQ* magazine and so forth or from various sites on the Internet. Sources of material suitable for designing computer programs include *QST*, *The ARRL Handbook* and various books and magazines.

Looking at some of the material discussed above will give you a feel for what a fully coded project will look like. If you have the right language software, you probably will want to load some of the source code you find (after reading instructions about how to do this with your particular coding software). If this has not frightened you away, and it should not, try the program out.

What language should you start to program in? It depends on your budget. The latest professional editions of Windows software start at about \$400 but that is because they contain tools to program Internet sites, servers, networks and all kinds of things in addition to desktop applications. What you do need is something that will let you program desktop applications. Both Borland and Microsoft have beginners' versions of their programming software available for less than \$100 at present. I would recommend investing in one of these rather than working with the older software. You will get the latest controls and they can be augmented from various freeware and shareware components available on the Internet. Older software is fine, I still use it, depending on what I want to do.

### DLLs and ActiveX Controls

The programming languages developed for the Windows operating system can operate with various mini-programs or program libraries. These tools can work with most of the individual programming languages although they themselves might have been developed with a different language. Such tools may be DLLs or ActiveX controls. A DLL is a function library that works in conjunction with programs that link to it. Several free



Fig 8—Programming-language media for *Delphi 5*, *C++ Builder*, *QuickBasic 4.5* and *Realizer*.

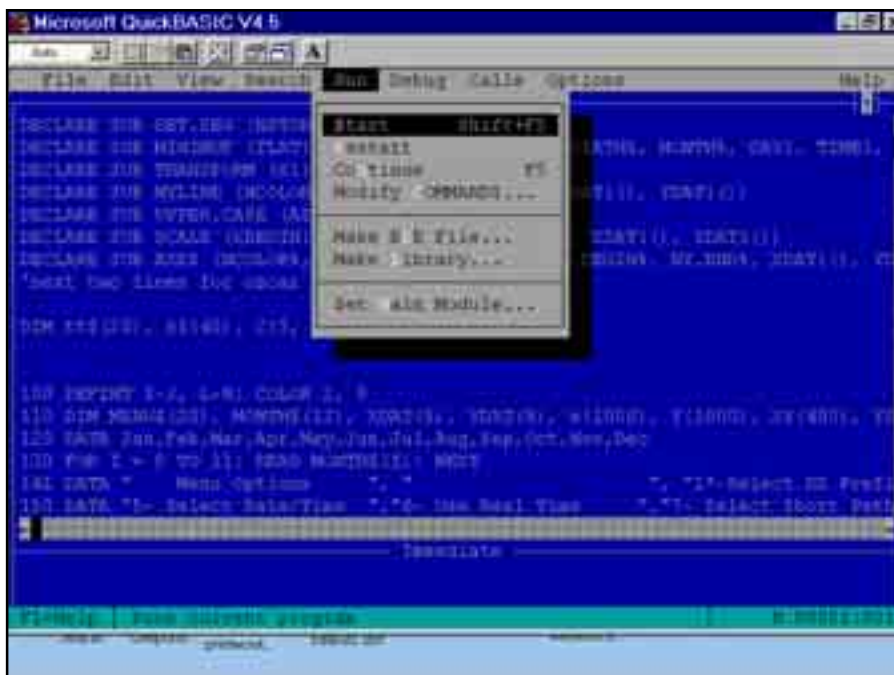


Fig 9—(right) The *QuickBasic 4.5* development screen showing one of its possible programming menus.

DLLs related to ham radio topics are available to link to your project including AE4JY's *PSK Core* DLL and JE3HHT's *RTTY* DLL. The only free ActiveX control I know of for amateur purposes is WA0TTN's *WinPSK*. DLLs can be generated by *Delphi*, and most versions of *C++*. *Visual Basic 3* does not create DLLs but can make use of those generated by other programs. *Visual Basic 4* and above can be used to make DLLs.

There are two types of Windows executable files, programs and DLLs. When you write a Windows application, you typically generate a program file that is an independent program. The executable programs (those with the familiar form *PROGRAM.EXE*) may use calls to functions stored in DLLs. Dynamic link libraries are program modules that contain code, data or resources that can be shared. They allow programs to be modular and simplify updating applications. They are language-independent, so a DLL can be used by *C++*, *Visual Basic*, *Delphi* or any other language that supports DLLs.

ActiveX is a Microsoft technology that is an extension of the older ODCX technology. It provides controls that can be used in a Windows visual-programming environment to provide certain functionality. ActiveX controls are add-ons to your programming environment and should be cross platform compatible with most 32-bit systems.

### What Should You Program?

If you have gotten this far in the article, you may have in mind a particular project or perhaps not. It is possible to write a software program describing just about any conceivable process or mathematical relationship. Many very good ham-related programs are available already. You may want to duplicate an expensive piece

of commercial software to save costs or just for the fun and satisfaction of doing it. I hope that you have something new in mind or a new way to present familiar material.

### Giving Your Software the "Commercial" Look

On-board help and "about" boxes or splash screens containing information about the programmer/copyright holder can make your program easier for the user to understand and help protect your rights as a developer by providing a place for a copyright notice or licensing terms, even if you choose to give your software away.

Tools necessary to develop Microsoft compatible help files are usually in-

cluded with the Windows programming language. The Microsoft Windows Help File Compiler was provided with the copies of the development programs I have as part of the development product. The compiler, in its various versions, is a script-like language similar to HTML, the language used to program many Web sites.

"About" boxes are drop-down or pop-up information boxes. Splash screens display a picture or other data on your screen when your program starts (Fig 12). Both of these methods can provide information about your program or its author in a pleasing way.

### Conclusion

Programming for DOS or for Win-

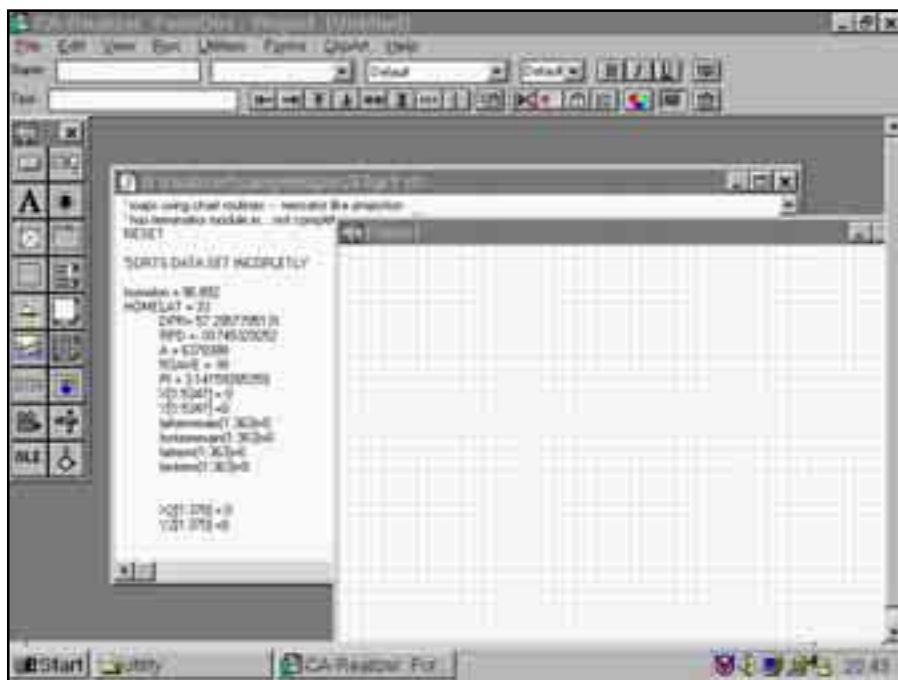


Fig 10—The *Realizer* for Windows development screen was one of the first to use visual objects.

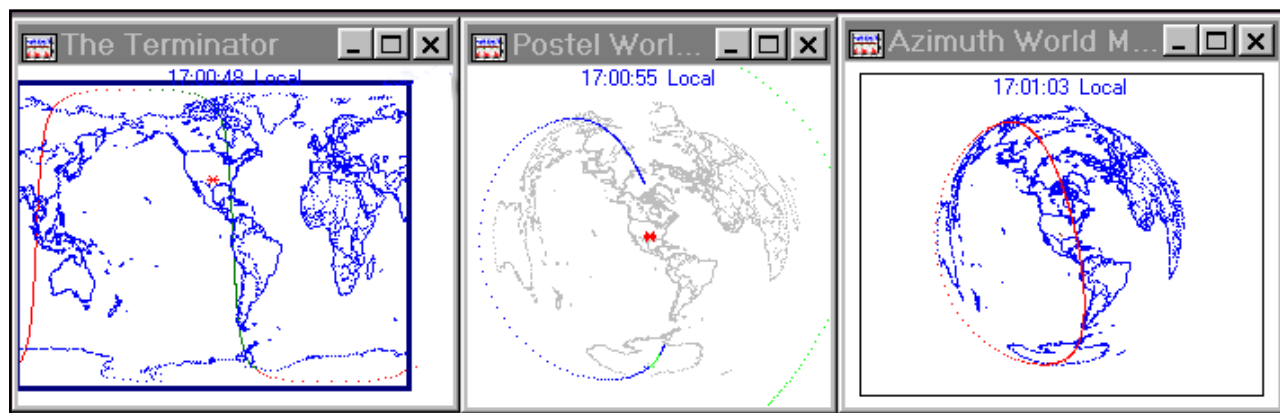


Fig 11—Pseudo-Mercator, Postel and azimuth-equal area map projections of the world using the CIA World Map public domain database coordinates described in Table 2. A gray-line track of the boundary between areas of sunrise and sunset is also shown.

## Additional Programming Resources

Readers may want to visit the sites of the Free Software Foundation ([www.fsf.org](http://www.fsf.org)) and Sun microsystems ([www.sun.com](http://www.sun.com)). Sun gives away *Java* tools. FSF does the same with many development tools. The only tool I use of these is the *C* compiler from FSF (*Gnu C*); it is a high-quality piece of code. I have downloaded *Java* from Sun, but never got a chance to use it (the book has been gathering dust for three years). It is similar to *Delphi* and *Visual Basic* in its abilities to create Windows programs. There is a lot more than just *C* and *C++* available from FSF. They have an editor (*Emacs*), a free version of just about every *Unix* tool ever written (available for *Windows*, *Linux*, *Unix* etc) and numerous things that are unique to FSF. I just cruised to their Web site and found projects for *C*, *Pascal*, Software Defined Radios, *Java* and a ham-radio section under "Hobbies."

The Free Software Foundation has what they call a "copyleft." They basically detest copyrights, but in order to protect their work, assert an actual copyright that let's you use the software for any purpose (including commercial purposes), but if you improve it, you *must* give your improvements away to the whole world. This is a good model for getting things moving in the ham community. *Linux* is now a viable product because of this model.—  
Ray Mack, WD5IFS, QEX Contributing Editor; [wd5ifs@arrl.org](mailto:wd5ifs@arrl.org)

dows certainly is not for everyone, but with the right tools, it can be an enjoyable and satisfying experience. If all you

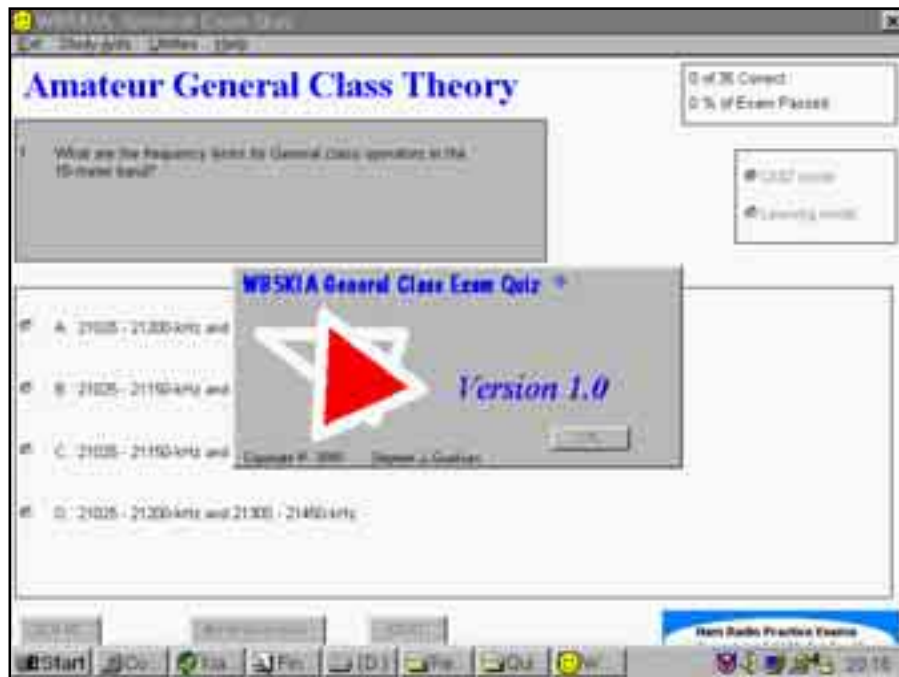


Fig 12—An about screen from the General license quiz generator and study guide programmed with *Realizer*.

have available are DOS tools, remember it is possible to run most DOS programs in a window in the various versions of Windows. If you do not program, but have some good ideas for a program or for its improvement, share them. If you do program for fun, consider making your project and code available for others to use in their projects.

*A ham since 1963, Steve Gradjan, WB5KIA, is a geological consultant in the Dallas, Texas area. He holds an Extra class license. Computer programming has been his second hobby since the late 70s. He has previously been licensed as WA8KBK and LA0DY. His wife Chris is WD5EML (ex LA0DZ) and 15-year-old son Francis is KD5HTB.* □□