

## Pascal on the Macintosh Part 1 - LightSpeed Pascal Review

by Bob Thong

LightSpeed Pascal Version 1.0

Produced by Think Technologies (0101-617-863-1099)

Distributed in the UK by MacSerious Software in the UK (041-332-5622)

Price £110 + VAT.

**THIS** is the first of a series of articles looking at programming in Pascal on the Macintosh. Back in the "early days" circa 1984, the only Pascal language system available was Lisa Pascal which didn't even run on the Macintosh. This was soon followed by Macintosh Pascal however, which did run on the Macintosh and certainly set new standards for interpreted languages. Nice though it was, Macintosh Pascal was an interpreter and programs ran rather slowly. In addition, it was not possible to create stand-alone applications and most features of the Macintosh ROM Toolbox except for Quickdraw were not accessible directly except via the "inline" procedure; something not recommended unless you know what you're doing. It was only until TML Pascal became generally available in the last year or so that we had a full scale Pascal development system running on the Macintosh which produced standalone compiled applications with full access to the ROM Toolbox. As all of the "official" technical documentation like 'Inside Macintosh' use Pascal notation and even the ROM routines in the Macintosh assume you are using Pascal, this situation was rather perplexing, in direct contrast to the plethora of C compilers which began appearing for the Macintosh within months of its launch. Pascal has now returned with a vengeance with two new compilers just becoming available, namely LightSpeed Pascal and Turbo Pascal. The latter is Borland International's port of its highly successful development system on the IBM PC to the Macintosh and I hope to be able to give you a report on this in a future issue.

### **LightSpeed Pascal**

Hot on the heels of the recently launched LightSpeed C comes this, the latest product from Think Technologies. LightSpeed Pascal is a complete development system for producing standalone "double-clickable" applications. In fact any general CODE resource including device drivers, desk accessories etc can be created. It comes on three single-sided disks including examples and various utilities. The bare minimum development environment is a 512K with external drive although it is possible to use just the internal drive with disk swapping. The review system was a Mac Plus with external drive although most of the time, I used only a RamDisk and one floppy. Documentation was in the form of a hefty softbound A4 sized volume which looked like it started life on a Laserwriter. The manual is split into a user guide, reference section and appendices, is very readable as well as informative and includes a three-page "quick start" guide for people (like me) who don't read Macintosh product manuals before launching the application from the Finder.

## Features

Users of Macintosh Pascal will recognize the development environment immediately - the same program animation and trace facilities, "Instant" and "Observe" windows, "Stop Signs" for debugging break-points and standard "Text" and "Drawing" windows; these also work in exactly the same way. This is not surprising since Think Technologies also wrote Macintosh Pascal for Apple. However unlike Macintosh Pascal, LightSpeed programs compile to machine code and run as fast as you would expect a compiled program to. You have the option to run your program within the LightSpeed environment whence all the above debugging facilities are available or to create a standalone application. In the development stage, you would almost certainly do the former most of the time as a whole range of debugging facilities is then made available. Besides those facilities mentioned already, there is also "LightsBug", a machine code debugger and monitor facility. In fact it is possible to have all your source code windows on screen at the same time as your program is running. Your program can have its own menus and windows open simultaneously and to switch between your program's environment and LightSpeed's, you click on a little bug-spray can icon in the far right of the menu bar. This is reminiscent of Switcher and it appears that LightSpeed runs the program in a memory partition much like Switcher does. It is a cinch to click on the bug spray icon while your program is running, inspect or even change a few variables and then continue on in the program. Alternatively you could have an animated trace on-screen in a source window as your program executes in another window. One of the example programs, a cut-down version of MacPaint, even has deliberate errors which the manual invites you to fix in real time as the program executes.

Running within LightSpeed, a lot of program run-time errors are trapped by the system rather than causing the ubiquitous "time-bomb" and you are dropped back into an editor window with a little hand icon pointing to where in the source code the error occurs. The editor is context sensitive (i.e. like the Macintosh Pascal one it indents and boldfaces key words) and is an integral part of the system. The whole setup is so integrated that LightSpeed's windows and your own program's windows coexist happily on the same desktop and you have to look at the menu bar to see whether your program is executing!

Separate compilation is not only possible, it is an integral part of the whole program development process. To start developing an application, you first have to set up a "project". This is a document which keeps track of all your separate source files including when each was last modified. A change in one source file and a recompilation only causes that file to be recompiled. The project document also keeps track of dependencies so that a change in a low-level program module also causes any high-level modules which depend on it to be recompiled, reminiscent of a Unix-style "make" facility. You can specify a single resource file (e.g. output from RMaker) from which your program gets its resources and also the segmentation of your final application (eg so that initialisation modules can be unloaded). External machine language routines can also be linked in. A utility is supplied which converts ".Rel" files produced by the MDS assembler to LightSpeed format. This should allow routines written in say Consulair Mac C which uses the MDS format to be linked in although I didn't test this. Compiler options to enable/disable range-checking, arithmetic overflow checking, debugging code, and name generation for use by Apple's Macsbug machine-code debugger (also supplied) are available.

## **Examples and Other Utilities Included**

Besides those already mentioned, there is a desk accessory "shell" which allows you to test a desk accessory in source form without installing in the system file, a code profiler which measures the time spent in each routine of your program, RMaker, ResEdit, a full scale multi-window text editor with changeable fonts etc in source code, a generic desk accessory and even source code for an FKEY resource. Libraries include support for all managers and packages in "Inside Macintosh" as well as AppleTalk, MacinTalk, the new 128K ROMs, SCSI manager etc.

## **Compatabilities with other Pascals**

All Macintosh Pascal programs should compile unchanged (mine did) as all its standard routines are implemented e.g. NewFileName, GetDrawingRect, inLineP, etc, etc. An extremely nice feature of LightSpeed is that all standard types and routines as defined in "Inside Macintosh" are pre-defined i.e. you do not need any "include" files. This saves a lot of compilation time. Most Lisa Pascal and TML Pascal programs should compile unchanged - there is a section in the manual about porting programs from other compilers. Separate compilation is handled by the "unit" concept with interface and implementation parts set up for each program module. I managed to convert the TMLskel program (from one of the MUG PD disks) in only a few minutes. One point to watch out for is that LightSpeed is quite strict on comments being one-line only so that you have to put end braces on each comment line.

## **Performance**

I hope to do a more comprehensive set of benchmarks for the next issue in comparison to Turbo Pascal but as a rough guideline, LightSpeed compiled and linked the TMLskel skeleton application, including debugging code etc in 20 seconds flat compared to 2 minutes 7 seconds on TML Pascal. The latter's time is heavily influenced by its extensive use of "include" files. In normal development use, LightSpeed's project managed "smart" compilation will cut down the compile/link/test cycle by an even greater factor not to mention the debugging facilities available.

## **Gripes**

Each program unit in the project is only limited to about 3000 lines though you can have an unlimited number of program units. This can cause problems in porting from other Pascal compilers as you will have to break the program up into smaller chunks. There is a lack of small simple examples, especially for someone new to writing Macintosh applications as opposed to generic Pascal programs. The numerous small examples supplied with TML PAscal for example are very useful. The text editor example program is however an excellent guide on how to build a full-scale application with support for desk accessories, clipboard cut and paste, etc. This example text editor is so good you can use it to write RMaker scripts - LightSpeed's integral text editor will attempt to format and indent any file as a Pascal program although you can save in standard 'TEXT' format. A minor irritant is that you need a project file for even the smallest program. This is a pain when you just want to knock something up quickly to see how some feature works.

## Summary

LightSpeed Pascal is a very powerful and innovative development tool - the compilation speed and debugging features have to be seen to be believed. I see this as an first class upgrade for anyone already using Macintosh Pascal or TML Pascal but I have my doubts on its suitability for new Pascal programmers. The latter are probably better off starting with Macintosh Pascal as it is a better educational tool; LightSpeed offers too many confusing features for the novice. Experienced programmers from other non-Mac environments, armed with their copies of "Inside Macintosh" or "Macintosh Revealed", may also find LightSpeed a good way to get productive quickly on a Macintosh. As to whether this is the best Pascal development environment for the Macintosh, I will hedge my bets until I see what Turbo Pascal for the Macintosh looks like. It's the best I've actually **seen** to date on any machine Macintosh or non-Mac, by a long way though. Recommended.

## Acknowledgements

Thanks to Jim Mangles of MacSerious Software for the loan of the review copy.

©1986 The Macintosh User Group UK, 55 Linkside Avenue, Oxford OX2 8JE, England

This article first appeared in **The Mac Times UK** Vol 1 Issue 4 (Dec/Jan 86/87) and may be reproduced free of charge by other User Groups provided that this notice remains intact.