The Story Behind Apex/XPL0 and the 6502 Group

by Larry Fish

It all started with the release of a brand new microprocessor. In 1974, a group of former Motorola engineers formed a new company called MOS Technology. Later that year, they released a microprocessor called the 6501. It was pin-compatible with the 6800, and this feature soon had MOS Technology in court facing a lawsuit from Motorola. With a few pin changes, all was forgiven, and in the spring of 1975 MOS Technology released the 6502. In spite of its inauspicious beginnings, the processor had several advantages. It had a fast, pipelined architecture; and it was one of the few processors with indirect and indexed addressing modes. But most of all, it was cheap. Where an Intel or Motorola chip might set you back as much $500, the sweet little 6502 was only $25.

In 1975, Denver was a hotbed of computer activities. There was a new computer club called the Denver Amateur Computer Society (DACS), and every meeting was packed with radio amateurs and hobbyists anxious to learn about this exciting new technology. DACS was dominated by Intel 8080 based computers, but there were a few people interested in the 6502. Wayne Wall was one of the first to appreciate the power and speed of the 6502, and he decided to form a subgroup of DACS for 6502 users.

Wayne was a genius assembly language and systems programmer who worked at the Colorado School of Mines. He began holding meetings every Tuesday night at the School of Mines Computing Center for anyone interested in the 6502. At first it was called the 6502 sub-group of DACS, but as time went on, it became the 6502 Group.

All told, there were probably about 100 members, but on a typical night, 30 to 40 people would crowd into the meeting room, anxious to hear Wayne's latest ideas and see his demonstrations. He was famous for his magic shows that invariably consisted of doing something hitherto unimaginable like simulating human speech with 13 bytes of assembly code. Wayne quickly gained the well deserved reputation as a computer wizard.

Most members of the group owned Digital Group systems. The Digital Group was a local company that had a short but spectacular career manufacturing computer kits. They were powerful, easy to assemble and way ahead of their time, with memory-addressable video cards, system monitors and relatively large memory boards. In addition to the Digital Group systems, the group had a scattering of Kim's, Tim's, Jolt's and even a few odd homebrew machines.

At the time, there was virtually no software available for the 6502. All programs had to be tediously hand-assembled and typed into the machines. Wayne decided that his first task was to find a way to deal with this problem.

The School of Mines had a huge black mainframe called a PDP-10, and Wayne wanted to use the power and capabilities of this computer to write 6502 code. The first possibility was to use a FORTAN cross-assembler that was available from MOS Technology, but it was slow and buggy. Instead, Wayne chose to write 6502 macros for the PDP-10's "PAL" assembler. After a few days work, Wayne had a fast new assembler that he called "PAL-65." Suddenly we had the ability to write relatively large assembly language programs for the 6502.

Initially, Wayne cranked out a series of assembly language games. The first was a simple text based game where you shot arrows at a mythical beast called a Wumpus. Most of our computers had between 512 and two kilobytes of memory, but as Wayne's games grew in complexity, we struggled to keep up, spending our nights soldering memory chips onto circuit boards.

Wayne now turned his attention to a high level language. The Intel based microprocessors had a version of BASIC written by Microsoft. Wayne decided that we should have a similar language, and he chose to write a version of the DEC language called FOCAL. Even though it was a relatively obscure language, FOCAL had some subtle advantages over BASIC, the biggest was its ability to evaluate strings as expressions.

Wayne wanted FOCAL to be a group project, and one evening he showed up with a stack of photocopies of the DEC flowcharts for the language. His plan was to assign each section of the chart to one of the 6502 Group members. We were supposed to go home, write 6502 assembly code for the section, and in a few weeks, we'd have an operating version of FOCAL. Unfortunately, Wayne had overestimated our skills. Most of us had been assembling code by hand and had no idea how to work from a flow chart. In the end, Wayne and Bob Ulshafer did most of the coding.

When it was finally finished, FOCAL was a big breakthrough. Suddenly we had a true programming language. Complex new games began to appear on our computers including a version of the famous mainframe game Star Trek, written by Larry Fish. Over the next few weeks Wayne wrote a five-byte floating point package for FOCAL based on a Dr. Dobbs article. With the new floating point power, simulations and mathematical programs sprouted up everywhere.

The next major task was an assembler. Wayne didn't really need an assembler for his own use, since he had the PDP-10, but the group whined and pleaded and he finally gave in. Wayne wrote the core of the assembler and Larry Fish added a tree sorted symbol table for speed.

The 6502 now lacked one crucial piece of software: an editor. At this point, Peter Boyle stepped in to fill the gap. Peter Boyle was another of the computer geniuses that the 6502 Group seemed to attract. He was urbane with a commanding voice and a strong South African accent. A brilliant mathematician, he held a doctorate degree in statistics. While he was working at the University of Utah, he and some of his colleagues experimented with many new computer ideas. One of their ideas was the framework for a new kind of editor. Up until that time, most editors were written for teletype-like devices that could only scroll single lines of text from the bottom of a screen. Peter realized that with the Digital Group's random access screen memory, he could write the kind of editor he had always wanted. He called his new editor VED and it encompassed several radical new concepts. First, it had a scrolling window that showed a whole page of text instead of one line. Second, it incorporated a two-pointer buffer scheme that allowed for easy insertion of text without transferring large blocks of characters. Finally, it incorporated an intuitive keyboard command layout. Not all the home brewed machines could run Peter's editor, so Larry Fish wrote a version of DEC's powerful macro editor, TECO.

The proliferation of programs put the Group at the center of the small universe of 6502 software. People around the country were clamoring for this software and the Group began sporadic attempts to market the programs.

It was at this time that we began working with the 6502 Program Exchange. This small company in Reno Nevada was run by a physics professor and his son. They were delighted with the Group software, and they began marketing the software nation-wide. A steady stream of income began flowing into the Group treasury. At times it averaged $500 a month. Other attempts to market the software were less successful and at one point two east coast companies were caught pirating our programs. In spite of the problems, the 6502 Group had more money in its treasury than the parent club DACS.

About a year after Wayne developed FOCAL, Peter began giving a series of seminars on high-level, block-structured, objected-oriented languages. Peter was a big fan of Smalltalk and he spent many evenings extolling the virtues of local variables, parameter passing, extensibility, encapsulation and object orientation.

At the end of one of the seminars, Peter announced that he was working on a new block structured language called XPL0. He also said that he was going to market the program and asked us all to pay one penny to become licensed users.

At first, everyone was a bit confused by XPL. Although everybody was impressed by the sophisticated features of the language, most people were writing simple programs in FOCAL that didn't need these features. It was hard to see the advantages of XPL when you were writing elementary programs like Tic Tac Toe. It wasn't until the arrival of Apex that the real power of XPL became apparent.

Apex came about because of the need for a reliable mass storage system for the 6502. At the time, most computers had some kind of cassette interface; but they were slow, unreliable, and could not be controlled by the computer. Peter, Wayne and a few other people had access to mainframe and minicomputers, which allowed them to develop software. The situation came to a head when Peter considered taking a new job and had to face the possibility that he would lose access to the minicomputer at work.

In retrospect, it seems obvious that floppy disk drives were the answer to the problem, but at the time there were big questions about their reliability. There were several DEC employees in the 6502 Group, and they used floppy disks to boot some of their big mainframes. Their experience with these drives was not good. They found that the disk wore out after just a few uses, and they felt they were too unreliable to use with microcomputers. Wayne thought that the solution was to use some kind of inexpensive tape drive, and he began experimenting with a simple tape drive called a Phi-Deck. He also began writing a simple assembly language operating system he called EDMOPS.

As floppy disk drives started appearing on the 8080 machines, it became obvious that the drives actually worked very well. It turned out that the new floppy disk drives had glass heads that made them much more reliable than the old metal heads that were used on the DEC machines. Peter and Larry quickly bought 8-inch floppy disk drives and built interfaces for their computers. Within a week, Peter had written a fully functional operating system modeled after OS/12, one of the DEC operating systems. Again his experiences and the University of Utah served him well. He had worked out the concept of an operating system that would swap itself in and out of memory to conserve resources. In addition, Peter knew that because of the slow rotation speed of floppy disk drives, the operating system had to use files organized with sequential sectors. His new operating system embodied these principles; and as a result, it was faster and more responsive than many mainframe systems. Peter called the new operating system the Floppy File System, or FFS for short.

This was the first demonstration of the power of XPL0. Over the next few months Peter expanded the capabilities of FFS rapidly, writing utilities and adding new features. Everyone could see how quickly Peter was able to write powerful, sophisticated programs and soon everyone was climbing on the FFS/XPL bandwagon.

The development of the new operating system once again thrust the group to the center of the 6502 universe. Companies began contacting Peter with propositions to buy or market the software. Documentation was needed before the software could be used, so Larry wrote manuals for XPL, the assembler and the VED editor.

Up until that time, the 8080 and the Z80 had been the dominate microprocessors; but in 1979, the Apple-II took the computing world by storm. Suddenly the 6502 was at the heart of the most popular microcomputer in the world. For the first year, the Apple didn't have a floppy disk drive so it wasn't very useful as a development system. Nevertheless, Peter began working on an integrated compiler/editor for the Apple that could save files and programs to the Apple cassette interface. Larry wrote an article for Kilobaud Magazine extolling virtues of XPL and block structured languages. The article concluded by offering the integrated editor/compiler and manuals for $40. In this way, XPL became the first block structured high-level language for microcomputers to reach the market. It was months ahead of the much ballyhooed UCSD Pascal.

When Apple finally released a disk drive for the Apple-II, it had an awkward operating system centered around the BASIC interpreter built into the computer. Peter began working to port FFS to the Apple. He also decided that FFS wasn't a very catchy name for the operating system. Larry suggested the name Apex (after an old mining camp west of Denver) and it stuck.

Jim Lauletta was one of the early 6502 Group members. In the intervening years he turned his hobby into a mail order business called Apparat. Later he owned a small chain of retail computer stores in Denver. Jim had been very successful marketing an alternative operating system for the TRS80, Radio Shack's microcomputer. Jim felt the new Apex operating system would be a good addition to his line of products, and Aparat became the first company to market the integrated development system of Apex and XPL. Over the years, several other companies marketed Apex and XPL including our old friends at the 6502 Program Exchange and a California company called Computer Sight.

At this point XPL, had one glaring deficiency: the lack of floating point capability. Loren Blaney took it upon himself to correct the problem. He took Wayne's FOCAL floating point package and incorporated it into XPL. The implementation extended XPL's simple but powerful pointer structure to include multidimensional floating point arrays.

During the early eighties, XPL and Apex were put to use in a variety of applications including medical equipment, accounting systems, telescope controllers and home automation systems. Ultimately, XPL was ported to an amazing variety of machines including the PDP10 mainframe, IBM360, 8080, 6800, 68000 and 80X86.

Up until this time, XPL had always been an interpreted "P-Code" language. Although P-Coded languages are much faster than pure interpreters, a native language compiler could be even faster. Peter had examined the possibilities of writing a native language compiler for the 6502, but he concluded that the 8-bit data path was just too small.

In 1984, Loren Blaney decided that the elegant architecture of the 68000 would be ideally suited for a native language compiler. By keeping all the important language pointers in CPU registers, Loren was able to create one of the fastest 68000 compilers ever developed. It was 20 times faster than the Apple version and could generate Mandelbrot sets 200 times faster than anything we had seen before.

With the new 68000 version, XPL and Apex began to spread outside the 6502 world. Over the next three years, versions of Apex appeared on the Stride, Amiga, and Atari systems. Eventually even the Macintosh had its own version of Apex and XPL.

As the Apple-II faded into history the obvious successor was the Macintosh. Unfortunately, Apple was determined to maintain control over the market, and they were reluctant to provide technical information for the new Macintosh models. This made it very difficult to maintain an operating system like Apex on the Macintosh.

IBM's market strategy was completely different. They made all the PC's technical specifications public and soon the market was flooded with inexpensive IBM clones. At this time, Peter and Larry began to explore the PC world. The availability of a quality operating system like DOS, relegated Apex to the history books, but there was still a need for a powerful high-level language.

At the time, Microsoft had been marketing several generations of C compilers for the PC. Unfortunately, these compilers had weak text screen support and almost no graphics capability.

In 1987 Larry began the job of porting XPL to the IBM-PC. Over a two year period, Larry created both interpreted and native language compilers, IEEE floating point support, and extensions to support for the DOS environment. Later, he developed an optimizing compiler, color text windowing support, graphics support, coprocessor support and the capability of addressing up to a megabyte of memory. As the 386 became available, Larry worked out a scheme for accessing 32-bit memory in real mode. Loren Blaney later used this scheme to implement a special 32-bit version of the compiler.

Through the early nineties, XPL still had a few dedicated users. Today, DFM Engineering uses it to control over 50 large telescopes around the world. Also, The Certex Company uses it to control thousands of check writing machines. In addition to various types of controllers and embedded processors, XPL is still being used for application programming. One of the most widely distributed XPL applications is COMPASS, a cave mapping program written by Larry Fish. Currently, there are more than 1500 COMPASS users world wide including 15 National Parks in the United States, Australia, South Africa and Europe. At the same time, with over 48,000 lines of code, it is probably the largest XPL program ever written. Even strains of the original VED editor still exist in the form of XED, an editor written in XPL by Rich Ottosen.

With the availability of powerful operating systems and languages, the glory days of XPL, Apex and the 6502 Group are pretty much over. Still it is instructive to look at the accomplishments of this small group as a model. The 6502 Group created operating systems, languages and applications many years ahead of its time. This came about because of the group's unique ability to share ideas and build on each other's knowledge.

Return to 6502 Group Home Page