

A Brief History of As-Easy-As

An interview with co-creator Paris Karahalios

Copyright © 2025 Technically We Write

Paris Karahalios, Jim Hall (ed.). (2025) *A Brief
History of As-Easy-As*. Technically We Write.

Published under the Creative Commons
Attribution ShareAlike license.

Contents

Introduction.....	7
Why I loved As-Easy-As.....	7
About Paris.....	13
Let's start with an introduction. Who are you, and what do you do?.....	13
You were a co-founder of TRIUS. How did the company start?.....	14
How did TRIUS change over time?.....	17
TRIUS Inc shared the registration code for As-Easy- As, that was a great gesture to say "Thanks" to the community of original users. Were you behind this decision?.....	24
About As-Easy-As.....	27
You co-created the As-Easy-As spreadsheet for DOS. How did that come about? Why write a spreadsheet?.....	27

How did you create As-Easy-As? What was the process to design and create a spreadsheet application?.....	29
What were some standout features in As-Easy-As?	31
As-Easy-As was shareware. What was “shareware” and how did that work out as a business model?..	34
As-Easy-As was originally a DOS program. What things do you have to “balance” when you write DOS programs?.....	41
What’s an example of something you can do in DOS programming that’s hard to do in Windows or Linux?.....	43
What things have gotten easier in programming? ..	44
What are some “tricks” or methods you used to write a DOS application like this?.....	45
As-Easy-As was originally written in Turbo Pascal, then Delphi. Why change to Delphi?.....	46
How did you “discover” computers?.....	47
What was your first personal computer? What did you find exciting about it?.....	49
What was your first programming language?.....	51
Did you use computers and programming to help you in your Nuclear Engineering program?.....	51
Do you still do programming today?.....	52
Do you have any examples of interesting ways that people used As-Easy-As?.....	53

How did you decide what features to add in the next version?.....	55
Was As-Easy-As used mostly by engineers, or was it used by office workers too?.....	57
Are there any features that you wish you had added to As-Easy-As for DOS?.....	64
Programmers are the best at spotting their own bugs after-the-fact. Are there any bugs in As-Easy-As that you look back on and think “I wish we’d fixed that”?.....	67
About the manual.....	71
Registered users got a printed manual. How was this written?.....	71
What system or word processor did you use to produce the manual?.....	72
What was the process to write the first manual?...73	
What fonts did you use for the manual? What’s your favorite font for print?.....	74
What was the editing and publishing process like?	75
What was the printing process like? How did you print the manual?.....	76
What did the manual look like?.....	77
Thank you!.....	93
How I used As-Easy-As.....	94

Introduction

Why I loved As-Easy-As

Jim Hall (editor)

In 1990, entered the undergraduate physics program at the University of Wisconsin-River Falls. Our physics department had a very strong lab program; every semester, we physics students performed experiments that exercised concepts we learned in the classroom.

The professor who designed the lab program made sure we understood the error inherent in taking measurements. Every experiment required us to do an analysis of uncertainties of our measurements, a grinding process we called “error analysis.” In theory,

you can calculate the mean and standard deviation of a measurement by hand, but it's hard work. It really calls for a computer to do it.

And that's where As-Easy-As really stood out.

As-Easy-As was an example of shareware, a new way to sell software. Instead of purchasing an expensive "boxed" application from a computer store, you could download a copy of a shareware program from a dial-up bulletin board system. I also subscribed to a catalog that listed the best shareware applications and games; for a few dollars, they would mail you a floppy disk with the shareware programs you wanted to try. That's probably where I found As-Easy-As.

With shareware, you could try out the program before deciding if you wanted to buy it. And they encouraged you to share the program with others so they might buy a copy too.

For only \$69, I had a powerful spreadsheet application that let me do all of my lab analysis from the DOS computer in my dorm room. With As-Easy-As, I could quickly calculate the values (with uncertainties) for each of my labs, including linear regression. When I was done with my analysis, I

could print out tables and graphs to my Epson FX-80 dot matrix printer.

I think I bought every new version of As-Easy-As from then until version 5.7, the last release of the of the DOS version, released in 1997. I used it for everything, even after I started a career in IT. Instead of calculating gradient values and standard deviations, I used As-Easy-As for typical home office work such as managing a home budget and figuring out how much my wife and I could afford in monthly payments for our first home loan.

I still use DOS today; in fact, I started the FreeDOS Project in 1994 to create an open source version of DOS. So it shouldn't surprise you to know that I boot FreeDOS to play my favorite DOS games, and to use my favorite DOS applications. Even in 2025, I still count As-Easy-As 5.7 for DOS as my most-favorite spreadsheet application, on any platform.

I found Paris Karahalios on LinkedIn, and I reached out to ask if I could interview him for an article I wanted to write. I find that most people will write about 800 words in an interview, no matter how many questions you ask. If you ask four questions, they'll write long answers for a total of about 800

words. If you ask ten questions, they'll write shorter answers for about 800 words.

Paris wrote a lot more than that. And he provided a ton of technical detail about how he and Dave Schulz created the prototype for As-Easy-As, based on a sample "spreadsheet"-like program in Borland TurboPascal. Paris also described how they continued to add features to the prototype until it had parity with other spreadsheets like Lotus 1-2-3.

And I was surprised that Paris volunteered to write more. I sent several follow-up questions, and Paris shared more details about how they grew TRIUS Inc as the company behind As-Easy-As, his technical background, and how they wrote the manual. He also shared a collection of photos, articles, and letters about As-Easy-As.

In the end, Paris wrote almost 8,000 words. By comparison, a short novel (a "novella") might have 10,000 to 30,000 words. Paris wrote so much that I wasn't able to include it all in one article; I split up the interview into several series:

1. An interview on Coaching Buttons about how they grew the company

2. Another interview on Technically We Write about writing the manual
3. A series of four interviews on All Things Open with a “deep dive” on the technology behind As-Easy-As, and programming the spreadsheet

I wanted to share Paris’s story, so others could read the whole thing. I collected everything Paris sent me —every interview answer, every photo, every story— into this book. This is a brief history of As-Easy-As, but it’s also a demonstration of why As-Easy-As was my favorite spreadsheet, and still is.

Dave and Paris created a rare thing: a program that I love to use. If I didn’t need to share spreadsheets with others, I think As-Easy-As would fit almost every spreadsheet need I have today. As-Easy-As can do all of the modeling and analysis that I need. It can even do conditional formatting, although in a different way than you might be used to with more “modern” spreadsheets like Microsoft Excel, LibreOffice Calc, or Google Sheets.

I hope this history shows why I loved As-Easy-As, and maybe you might like to use it too.

About Paris

Let's start with an introduction. Who are you, and what do you do?

My name is Paris Karahalios. I have a BS and MS degrees in Nuclear Engineering with an option in Fusion and emphasis on Radiation accident analysis and health consequence modeling. I worked in the nuclear industry for about 10 years before I got into software development and IT around 1988. I started my third career about 10 years ago, when TRIUS stopped operating, as a senior technical project manager. I am currently the VP of Technical Project Management at Spire. I thoroughly enjoy what I do and the group of people I work with. I think of it as a

continuous learning game, where I am called to solve new, challenging problems every day.

While in college, I became infatuated with computers and by the time I finished in 1980, I had become fairly proficient in using the CDC Cyber 6400 mainframe at the school. By that time, I had also written a number of programs for the TI-59 programmable calculator with the magnetic cards, which were published in the PPX TI Library and I had written a payroll program for the TI-99/4a computer. (I had to save and load the program each time I used it from a handheld cassette recorder using a regular cassette - no DAT tapes yet.)

This is just a long-winded way of saying that I had become a computer geek early in life, and continue to be one to this date.

You were a co-founder of TRIUS. How did the company start?

In 1984-85, I was working in the Nuclear Division of a large AE firm in Boston (Stone & Webster Engineering Co), where I was spending most of my time on mathematical and computer modeling of nuclear accident consequences, using IBM-360 mainframes.

Personal Computers were becoming popular and the company bought a true blue IBM PC-XT for each of the groups. Real powerful at the time, IBM PC-XT with two double-sided 5.25" floppy drives, 256 KB RAM and a single application - Lotus 1-2-3. We had to sign-up early in the day, or the day before, to reserve time on the shared PC, it was a hot item!

Some of us Stone & Webster employees formed a PC-Users group, where we met once a week, during lunch, and discussed the latest updates to PCs, new software, ideas on how to use PCs in new ways, etc. I met Dave Schulz in one of those meetings and we started hanging around. Dave was a Civil/Piping Engineer working in a different division of the same company, but his fundamental knowledge of computers, computing and software was unmatched! It was as if his brain worked in the decimal system for day-to-day life, and in bits and bytes at a native CPU level when it came to computers. I was impressed with his depth of knowledge and ability to translate theory into computer code then, and for the remaining 28+ years we worked together!

While I was fascinated with the capabilities of Lotus 1-2-3, he had just purchased a copy of Turbo Pascal

and was dissecting the rudimentary grid (spreadsheet) sample app that came with it. Pretty soon, he started expanding its capabilities and we started to meet at his desk, discuss and review progress every day. It was still a basic program, but it was continuously improved.

At the time, 9-pin dot matrix printers were becoming inexpensive, but the print quality was lousy. 24-pin printers could generate “letter quality” print, but were very expensive. Dave and I spent time understanding how the printers were driven through the interrupts and wrote a .COM program (written at the command line with the debugger included on the DOS floppy disk). When it ran, it would terminate and stay resident, it would interrupt BIOS calls to the printer and would then take over and manipulate the print head, so that a cheap inexpensive 9-pin printer could print “letter quality” text. We thought we could make some extra (side) money selling the program.

Something not many people know is that Dave and I formed TRIUS so that we could continue developing and marketing the Printer Enhancement program (if I recall correctly, it was less than 80 bytes long). In the meantime, work on the “spreadsheet” program

continued. One of Dave's co-workers had cutely nicknamed it "As-Easy-AS 1-2-3" (a play on "Lotus 1-2-3"). We had reached the point where the program could now read and write Lotus 1-2-3 (.WKS) files - a breakthrough! All while Dave and I were still working for Stone & Webster.

How did TRIUS change over time?

As-Easy-As was a very successful program. It was translated into German, French, Portuguese, Spanish, Chinese, Italian. It was privately labeled for a number of Publishers/Distributors. At one time, it was locally published in 11 countries. In the early 1990s, Dave and I ended up quitting our jobs and working full time at TRIUS, eventually hiring a dozen employees, etc.

Meanwhile, TRIUS had developed and sold other software, in addition to As-Easy-As. A number of successful ones were in the area of Computer Aided Design (CAD).

In the late 1990's - early 2000's, a few companies were interested in licensing our CAD technology to use it

as “mapping” engines and we started shifting away from spreadsheets and CAD and more into GIS/Mapping. We did not have enough resources to give As-Easy-As the attention it needed, so we eventually decided to sunset it.

TRIUS paired up with a new company, UnderTow Software, which was primarily the main GIS/Mapping entity. Both companies eventually phased out in the early 2010s. I keep the triusinc.com domain still registered alive, because it feels that it’s part of me, we ran TRIUS for 29 years, with Dave for 26 of those years.

By the way, for reference, some of the software products TRIUS developed and published were: As-Easy-As, Alite, DraftChoice, ProtoCAD 3D, StarFlic, Pivot, DraftChoice Pro,...

As-Easy-AS and DraftChoice have received multiple awards like the Shareware Industry Best Business Application and Best Graphics Software award, the PC Magazine Best Application readers award and so on.

Unfortunately, we did not keep a lot of the original articles, etc. of the successful era of As-Easy-As and

TRIUS, but over the last few years people have forwarded me some clippings in electronic form.

CHEMPUTING

Could AS-EASY-AS Spreadsheet Be Easier?

M. Randerson, MCIC

Has sliced bread been superseded by the AS-EASY-AS spreadsheet? Not quite, but there is no doubt that this spreadsheet is one of the greatest things since SB to become available to the home computer community. Version 4.0 of AS-EASY-AS has been in circulation on every computer bulletin board (BBS) in North America for some time and this summer Version 5.0 has been released. The latter has made most of the commands compatible with those used by LOTUS 1-2-3 and allows import of text and data files from most programs such as WordPerfect, Microsoft Word, Lotus 1-2-3, and dBASE in ASCII format.

For those who use spreadsheet programs at the office or lab, there is little or no adaptation to sitting down at home to input and manipulate data on your own machine. For those who have never before used a spreadsheet it is simplicity itself to learn how to access and make the most of it. The 249 page instruction book gives very straight forward information on how to expand one's usage, as well as indexing all the features and macros which make it a truly versatile tool.

This spreadsheet forms a grid of 256 columns (A through IV) and rows 1 through 8,192 making 2,097,152 cells. A mix of descriptive text and numbers can be placed in each cell with calculations dynamically linked to other cell contents. "Dynamically" means that if you change a number the whole array recalculates immediately, but you can also suppress that if you wish input to progress as fast as you can type and then, when you are ready, telling it to recalculate. The width of a column or row height can be changed easily, but most PC monitors are limited to displaying 25 lines by 80 characters. To expand its access without losing the header and row titles of as many left hand columns or rows as is desired, the latter can be frozen in display; thus, as the cursor is moved to the right, the unfrozen columns will scroll across to the left but the chosen frozen columns will continue to display their contents, similarly for the frozen rows as you scroll down or up throughout your array.

Michael Randerson, MCIC, is with the Department of Regional Industrial Expansion. Chemputing is edited by Marvin Silbert, FCC.

Confidential data can be entered but hidden by setting the column width to zero. The information is not lost and can be involved in calculations, but it is not displayed for view by any nosy interloper strolling past the PC operator.

Even though it is such a powerful little package, it takes up fewer than 600K of RAM. It can run on any IBM PC, XT, AT, PS/2 or compatible computer operating on DOS version 2.11 or later, works with Windows 3.0 or later and responds to a mouse installed with its own mouse driver. For those without a hard drive, it can run directly off a single floppy (720K minimum) or dual floppy disc.

The graphics packages are very clear on Hercules and CGA, EGA and VGA screens. They give very impressive copy in so-called 3-D on all other compatible printers including dot matrix and my bubblejet. The latter cannot print landscape from other programs without purchase of an extended font package, but AS-EASY-AS carries its own landscape instructions for your selection; all print instructions are sent to the printer in ASCII form and so a separate printer driver is not required.

Thirteen different graph formats can be selected including X-Y, pie, line, bar, stacked bar, polar, strip and cumulative. Selection can be made of fill patterns, colors, line styles and tick-mark styles. Semi-log or log-log form of X-Y graphs can be specified. Labels can be custom positioned and titles inserted. Up to six curves can be displayed, with an additional six if two graphs are merged.

Ten steps on a bar chart are the maximum before a two-tiered or staggered labelling system is forced; the width of the bars automatically adjust according to the number of bars selected. Not only are pie charts differentially shaded but a slice or slices can be exploded out of the circle for emphasis in a presentation. Beware of being covered with pie filling when exploding your pie!

The print capability is not only self-contained by sending to the printer in ASCII but you can view your masterpiece at full size or 200% before sending to the printer. Page breaks can be inserted, automatic page numbering, font sizes for the main titles, legends, values and labels (with their positions on the

graph) can all be customized. The graphics can then be sent to a separate file, saved for the data file or printed in hard copy form. The graphics file type selected can be .PIC, the format for LOTUS 1-2-3, or .DBF, the extension for dBASE III or IV.

The Help information accessed on pressing the F1 key is very full, over 50 pages, but not always intelligible to the beginner who needs help the most. However, the instruction book is much clearer, partly because examples are plentiful. The Help Function can be particularly useful for picking one of over 70 macros or an @ Function out of 80 already available or which has been created by the user. It can also be used to select a printer set up code, or choose a range previously named in the spreadsheet (up to 250 labels or named ranges can be stored, each up to 11 characters long).

Access to the spreadsheet is effected through menus activated by the forward-slash "/" command. The menu will appear top left of screen and can be toggled to top right by pressing the period key. Selection is by pressing the highlighted letter of the desired option, or the cursor-down key, followed by "ENTER". For each option, the alternative sub-options of that item are shown in a horizontal list above the menu.

Multiple calculations can be invoked for a cell or group of cells using standard or custom @ Functions, results may be inverted, or regression formulae of straight-line data calculated by AS-EASY-AS in order to predict a future value. Values or formulae may be copied, moved, added or subtracted.

A search with find or replace function exists; data may be justified, sorted or manipulated in all manner of ways.

Temporary exit to DOS may be effected by using a SHELL command while AS-EASY-AS stays resident in RAM taking up only whatever memory is occupied by the worksheet.

AS-EASY-AS is a copyrighted software product developed and owned by TRIUS Inc., P.O. Box 249, North Andover, MA, ZIP 01845-0249. You may obtain a copy for US\$65 by writing or by Fax 508-688-6512. Bulletin Board 508-794-0762; telephone 508-794-0377. Technical support is provided on 508-794-0140.

IO L'ACTUALITÉ CHIMIQUE CANADIENNE, JANVIER 1993

As-Easy-As in L'Actualite Canadienne, 1993

Has sliced bread been superseded by the AS-EASY-AS spreadsheet? Not quite, but there is no doubt that this spreadsheet is one of the greatest things since SB to become available to the home community. Version 4.0 of AS-EASY-AS has been in circulation on every computer bulletin board (BBS) in North America for some time and this summer Version 5.0 has been released. The latter has made most of the commands compatible with those used by LOTUS 1-2-3 and allows import of text and data files from most programs such as WordPerfect, Microsoft Word, Lotus 1-2-3, and dBASE in ASCII format ...

Shareware Shop

Shareware graphics programs rival the power of major commercial releases.

No longer can Mac and Amiga fans argue that the PC is an inferior graphics machine.

Although slow processors and low-resolution graphics boards hampered PCs at one time, that is hardly true today—and, just as IBM-compatible machines have entered the graphics world, shareware has come along for the ride. You can find graphics shareware of all types, for everything from creating great drawings to laying out newsletters or creating on-screen presentations.

This month, I'll look at great graphics shareware. The half-dozen programs I've chosen include a desktop publisher, Windows and DOS paint programs and graphics conversion utilities, a CAD program, and a chart creator/presentation program. Library 11 (Shareware) in the *Computer Shopper* Forum on ZiffNet (**GO ZIFF-COMPSHOP**) contains all the files. I look at the address information to find filenames.

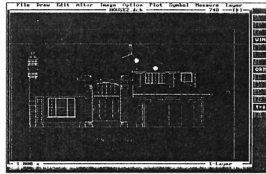
DRAFT CHOICE

If your drawing needs are technical—architectural drawings or any others that would normally require drafting tools—then a traditional drawing or paint program won't do the job for you. General graphics software lacks the special features that computer-aided design (CAD) programs pack, such as the array command (to copy and rotate a group of objects), the ability to animate objects, the creation of sophisticated 3-D effects or Bezier (flexible) curves, and a number of even more complicated features.

That's where DRAFT Choice comes in. Like Envision Publisher and NeoPaint, it does a remarkable job of combining ease of use with sophistication. With a pull-down menu across the top and a series of buttons down the right-hand side, you'll find everything within easy reach. Despite this simplicity, DRAFT Choice doesn't sacrifice flexibility. For example, it lets you work with up to 256 layers. And

the program has support for a wide variety of printers and plotters (even PostScript), so you should be able to print your work easily.

No, it isn't AutoCAD, but then again, that professional package has more power than you'd ever know what to do with. Anyone who does technical drawings will find DRAFT Choice a boon.



DRAFT Choice is a technical-drawing application with heavyweight CAD power minus the unwieldy price.

Computer Shopper, April 1993; Copyright © 1993 Coastal Associates Publishing, L.P.; A.R.R.

DraftChoice in Computer Shopper, 1993

If your drawing needs are technical—architectural drawing or any others that would normally require drafting tools—then a traditional drawing or paint program won't do the job for you ... That's where DRAFT Choice comes in. Like Envision Publisher and

NeoPant, it does a remarkable job of combining ease of use with sophistication. With a pull-down menu across the top and a series of buttons down the right-hand side, you'll find everything within easy reach ...

TRIUS Inc.

TRIUS Inc. of North Andover, Massachusetts, is a booming international company. It is only seven years old and has just six employees, but being small energizes rather than limits co-founders Paris Karahalios and David A. Schulz.

The publisher of the spreadsheets As-Easy-As and Alite, as well as Draft Choice, a two-dimensional CAD program, Pivot!, a sideways printing program, and Odyssey, a communications program, TRIUS views the

UTING ICES

world as its marketplace. The flagship product, the award-winning As-Easy-As, has been translated and published in English, Spanish, German, French, Portuguese and Chinese, and is marketed not by a large staff in various countries, but rather through local publishers worldwide. "This gives us the luxury of small overhead, while offering maximum exposure and allows the local publishers to handle the market as they know best—a win-win situation," says Karahalios.

Two years after As-Easy-As made its debut, TRIUS was approached by two small publishing companies in France and Germany. Translating the user's manual was a long and tedious but straightforward job. More difficult was translating the source code. To protect his interests, Karahalios would not allow the code to be licensed. Sending programmers abroad was too expensive for any of the three small companies. But all were on CompuServe.

A simple procedure was formulated to get the complex task accomplished with the participants an ocean apart. TRIUS sent program text extracted from the source code via CompuServe Mail to the other companies where they translated it and sent it back electronically. TRIUS then implemented the translated text into the program, and sent draft executable versions to the publishers via CompuServe Mail. "Nothing could be simpler," Karahalios says.

The French and German programs were initially ready in about two weeks with final versions prepared in just over a month. "Being able to complete a project in record time at minimal cost leaves more time available for other activities," he says, noting that one key to success is to not think small. "If you are going to do something, give it 100 percent or don't do it at all."

TRIUS supports its products online in the PC Vendor Forum (GO PCVENF). ◀

TRIUS in CompuServe, 1993

TRIUS Inc of North Andover, Massachusetts, is a booming international company. It is only seven years and has just six employees, but being small energizes rather than limits co-founders Paris Karahalios and David A. Schulz ...

TRIUS Inc shared the registration code for As-Easy-As, that was a great gesture to say "Thanks" to the community of original users. Were you behind this decision?

Yes, it was *my* decision to make these programs available to everyone for free, at that time. There was some internal pushback, because a couple of companies had expressed interest in purchasing the source code, but I felt that people in general had been good to us and since we were not interested in investing any more resources in updating these programs (we had shifted focus by then), we should let people have them!

Some have asked me "why didn't you just remove the registration requirement altogether?" Over the years we had built a sophisticated registration detection system interweaved within a number of program modules, installation module, validations, etc. The

effort to “untangle” that, so we can release a version that would not require registration would be significant and not really needed.

The forums aren’t there anymore, but Internet Archive has the announcements, here’s the one for DOS:

AS-EASY-AS for DOS - Free!

 by **Moderator** » Sun Sep 11, 2011 12:47 am

As of 03/11/2004, As-Easy-As for DOS was discontinued. Copies of the program can still be downloaded and FREE full licenses can be installed using the information below. TRIUS no longer supports or answers questions regarding this program, but other users visiting these forums might. Here are the links for the program and the user's manual,

Old As-Easy-As v5.7 for DOS program
http://www.triusinc.com/old_files/asa57.exe
Ser Number: 570-110-0424-3 *

* You are asked for this sernum during the installation of the program. The installation prompted you to find it on the inside cover of the manual. There no manual or inside cover - THIS is the number to enter.

Users Manual
http://www.triusinc.com/old_files/asa57_manual.exe
Manual Password: ASA57Licensed3974

Archived announcement from TRIUS Inc forums

As of 03/11/2004, As-Easy-As for DOS was discontinued. Copies of the program can still be downloaded and FREE full licenses can be installed using the information below. TRIUS no longer supports or answers questions regarding this program, but other users visiting these forums might. Here are the links for the program and the user’s manual,

Old As-Easy-As v5.7 for DOS program

http://www.triusinc.com/old_files/asa57.exe

Ser Number: 570-110-0424-3 *

** You are asked for this sernum during the installation of the program. The installation prompted you to fnd it on the inside cover of the manual. There no manual or inside cover - THIS is the number to enter.*

Users Manual

http://www.triusinc.com/old_files/asa57_manual.exe

Manual Password: ASA57Licensed3974

About As-Easy-As

You co-created the As-Easy-As spreadsheet for DOS. How did that come about? Why write a spreadsheet?

On April 26, 1986 the Chernobyl reactor accident took place. Because I had done a lot of research and had published a number of technical papers on Reactor Accident Consequence Analyses, a month or so later the state department selected me to be the US delegate to the first meeting of Group of Experts in Consequence Analyses (GRECA) meeting in Paris France (Russia would not allow us to go to Chernobyl, yet). The meeting was attended by one delegate from each of 62 countries, to discuss the accident, validate mathematical prediction models and advise governments on next actions.

At the end of the meeting, it was decided that all radiation measurements reports for every country (the radioactive plume was travelling around the world) would be sent to me (in the US) where we would validate it, analyze it and use it to possibly determine (a) the level of damage to the Chernobyl reactor core, and (b) movement and depletion of the radioactive plume over the world.

When I came back to the US and the data started arriving on a daily basis, I reserved the group's PC exclusively each day of the week and realized that we needed additional storage. We purchased two external Bernoulli drives (the original model, using the 8.5"x11", thick, Bernoulli disks, each storing a whopping 5 MB worth of data). We solved the data storage problem. We were using Lotus 1-2-3 to do all the data manipulation, which served us well, but we had a new issue. We needed to generate plots of the tabulated data and the Lotus 1-2-3 graphing capabilities (which were primarily meant for business) could not adequately handle the log/exponential curves we needed to plot needed for radioactivity decay.

Dave and I decided that the way to solve the problem was to spend more time on the development of As-Easy-As and focus on scientific plotting. Once these capabilities were in place (in a matter of a month or so), we started using As-Easy-As for analyzing the Chernobyl data and preparing the reports.

No more printer enhancement software, we now started concentrating on the further development of As-Easy-As.

How did you create As-Easy-As? What was the process to design and create a spreadsheet application?

The process was very simple. Dave and I would meet at his desk during lunch, or after work. We'd review the current version of the program, with the changes we decided to make the day before, make any adjustments, we'd discuss any new features that needed to be added, I would compile a list of tests that I needed to run to validate (a) the user interface, and (b) more importantly the built-in functions.

You have to realize that all built-in functions in the program were programmed by us using first

principles. Whether it was building amortization functions, or linear optimization functions, or trigonometric functions,... we had to come up with the base analytical formulas, write the code representing those formulas, implement them and then independently verify them. And, this was done for every function in the program!

Sometimes we'd have differing opinions with Dave and that would result in lengthy discussions later at night, usually at my house.

The program was written in Turbo Pascal and eventually in Delphi. Portions of the code were embedded assembly code and some even optimized in-line hex code. We'd prototype routines that needed speed optimization in Assembler, compile it, then extract the hex code and insert it in Pascal routines. We re-wrote many of the graphing functions to bypass the operating system and write directly to display memory for increased efficiency. Our motto was "produce the smaller footprint and most efficient code you can." Given the limitations on RAM, storage and distribution media, it made sense.

Compatibility with .WKS files was an early goal, because every spreadsheet program at the time was using .WKS files.

Focus on Physics and Engineering was because of our background in science/engineering and because of the need to use the program, as I mentioned earlier, for some of the Chernobyl analyses.

What were some standout features in As-Easy-As?

Standout features? It's like asking a parent "which one of your children do you love more?" I'll just list a couple that come to mind... (a) a very powerful macro programming language, which was only used by maybe 2% of As-Easy-As users, but for those who used it, was irreplaceable. Most users didn't even know that the built-in macro language could be used to model powerful apps with a UI. My comment about being irreplaceable for some, was based on the feedback we were getting from users (a benefit of shareware - the direct communications with the developers). Users let us know that they had used macros to model loading cargo ships, real time monitoring of hundreds of stocks, optimizing floor

layouts of new homes, thermodynamic analyses of new design engine blocks, leasing scenarios for dealerships across the USA, etc. (b) the ability to define your own functions. This feature was used by many users to write functions that were specific to their graduate studies fields, their businesses, etc. And, since they were stored in an external file, once built, they could be imported and used in any worksheet... There are many more, but if I don't stop here, I'll end up listing every feature of the program.

By the way, this is not known (how could it be), but before the decision was made to stop development on As-Easy-As, we were a long way towards developing a spreadsheet SDK that would allow transparent use of spreadsheet capabilities integrated within other applications and had also started work on a built-in programming language (much more powerful than macros).

As-Easy-AS received a number of awards from magazines of that era (PC Magazine, Computer Shopper, etc.), but for us the most prestigious awards were:

- Shareware Industry Awards - Best Application (1992) (DOS Version)
- Shareware Industry Awards - Best Application (1999) (Windows Version)

Shareware Industry Awards

Best Utility—*PKZIP*, by PKWare
 Best Application Software—*Envision Publisher*, by Software Vision
 Best Graphics Application—*Draft Choice*, by Trus Software
 Best Business & Finance—*AM-Tax*, by AM Software
 Best Educational Software—*Math Rescue*, by Karen Crowther/Electric Paintbrush
 Best Entertainment Software—*Wolfenstein 3-D*, by Id/Apogee
 Best Home or Hobby Software—*Family Tree Journal*, by Richard Cherry
 Best Technical, Engineering, Math & Programming Software—*Technojock's Turbo Toolbox*, by Robert Ainsbury
 Best New Utility/Technical Software—*Drag and View*, by Dan Baumbach
 Best New Business & Applications Software—*Envision Publisher*, by Software Vision
 Best New Entertainment, Education, Home or Hobby Software: (TIE)—*Cross Stitch Designer*, by Scott Horton and *Wolfenstein 3-D*, by Id/Apogee
 Best New MS-DOS Program—*Envision Publisher*, by Software Vision
 The People's Choice Award—*Wolfenstein 3-D*, by Id/Apogee
 There was also a "Special Recognition" Award which was awarded to Michael Callahan (aka Dr. File Finder), for his years of service to the shareware industry.



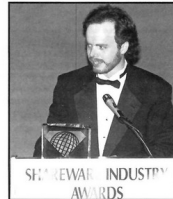
Dan Baumbach—Best Utility



Julian Achin—Best Application; Best New DOS Program; Best Business Application Program



Paris Karahalios—Best Graphics Application



John Carmack accepts for ID



Jay Johnson with Bob—Jaaaaaahaaay



Scott Miller accepts for Karen Crowther



Len Gray—Best New Windows Program

I don't think that As-Easy-As would compare favorably with today's Excel, in terms of features and capabilities that exploded with the new development environments that were not available to us. However, up until a couple of years ago, we used to get messages from users telling us that they were able to run As-Easy-As for Windows, using the wine emulator on Linux, and they liked the small footprint, meager memory requirements and simplicity of use.

As-Easy-As was shareware. What was "shareware" and how did that work out as a business model?

There are a number of varying stories as to who first coined the term "shareware." Andrew Fluegelman is credited with first releasing his program PC-Talk as "freeware," and Jim Button (Knopf) releasing his PC-File program as "shareware." Bob Wallace, the developer of PC-Write, always mentioned those two as the ones who started the revolution. There are many other software developers and small start-ups that went on to build successful companies using the "shareware" business model like Bob Wallace (PC-

Write), Marshall Magee (Automenu), Tom Rawson (4DOS), Phil Katz (PKZIP), Scott Miller (Kingdom of Kroz), John McAfee (McAfee Antivirus), Apogee Software, id Software, etc.) who could be called “the grandfathers of shareware,” in a way. (I’m sure I’m forgetting a few dozen names...)

In any case, it was a fairly straightforward business model. You develop a software program for the PC. You give free copies of it to people so they can try it, encouraging that they give free copies to others. In the early days, if they liked it and they continued using it, you asked for a donation. Later on, programmers asked for payment that entitled users to discounted upgrades, a printed manual, technical support, etc. All this in an environment where mainstream software publishers were taking full page ads in computer magazines announcing that copying their software, even for solely back-up purposes, was illegal. Shareware became the de-facto business model for individual developers and small companies.

Some shareware companies would turn off certain functions after the evaluation period (eventually called crippleware), while others, like TRIUS would never disable any features, but would display a

reminder to register when you started the program. Yet other companies would display the registration notice every XX keystrokes, or every YY screen displays, which eventually made them known as “nagware.”

The shareware concept was ideal for those early days of PCs. It worked for users, because they could try hundreds of programs, before deciding if they were willing to pay for some of them and it worked for the developers because they could release a program into the “sharing” community with zero marketing budget, generate a captive audience, get useful feedback, etc. and make some money.

Having said that, a lot of shareware companies, including TRIUS, eventually ended up spending capital on marketing and advertising in computer magazines of the era, as well as placing their products in retail channels, private labelling them for larger well known publishers, etc.

meet the author

HAVE you ever wondered about the people behind your favourite shareware program? What led them to become shareware authors; how they got started; whether people register; what it is really like to make a living from shareware?

This month, in the first of a series of reports Paul visits two very different companies in the Boston area: Trius Inc, authors of the As Easy As spreadsheet and Draft Choice CAD package, and JP Software whose 4Dos program is the leading (and indeed the only) replacement for MS-DOS's limited COMMAND.COM program.

SHAREWARE FOR AUTHORS

Whenever you buy software an important factor is the kind of company behind it. Will they provide support? Will any bugs be fixed promptly? Will there be future updates?

When we buy a program from Microsoft we all know who they are. And of course over the years many of us have learned just how much notice a very large software company takes of individual users! Despite Microsoft's recent efforts to put right this situation the impression still exists that if you want to get your complaints listened to, you had better talk to a small company for whom your order is still important.

But what are the people who write shareware like? Are they, as some people imagine, a bunch of part-time amateurs, producing badly written unfinished programs with non-existent documentation and little support? Hopefully readers of this magazine know better: With the large number of extremely good shareware programs around there is no longer any room (if there ever was) for badly written or sub-standard software. Nor is there

Paul Mullen, PC Shareware Magazine's roving reporter, has been talking to some leading Shareware authors to see just what it is like

room for companies who don't deal with their customers professionally, particularly in the US where customers are much less willing to accept bad service than they are in Europe.

It is often said, but bears repeating, that shareware is not a kind of software or a kind of company. It is a marketing method by which software companies may choose to sell their product. It follows that shareware companies will vary, in their size, their way of working, and in their support policies. Undeniably shareware companies tend to be small. Well, much smaller than Microsoft anyway, since the shareware approach provides a low-cost market entry opportunity that is ideal for start-up businesses. But that does not stop them being professional! Throughout this series of interviews the authors I met have emphasised that they are there to run a business and to make that business successful they have to adopt a professional attitude.

The sheer variety of shareware companies makes them interesting. Yes, there are part-time, one man businesses, but there are also a surprising number of large and growing businesses. There are some who entered the shareware market with the aim of founding a large software business and

others who chose shareware because they wanted a business that would not grow larger than they could handle themselves. Some authors simply wanted to "work from home, while others had the ideal of being their own boss and making a fortune. Still others never really intended to make shareware a full-time business but found themselves thrust into it by chance.

Like any business success story, the most interesting thing is how the authors got started in the business: What made them get into shareware in the first place; how they overcame the initial problems; the learning process that they went through; how long it took to become established and finally, the secret every potential author wants to know, what it takes to become successful.

Over the next few months we will tell you about some of the authors behind your favourite programs. I found their stories fascinating and I hope you will too.

TRIUS Inc., author of As Easy As and Draft Choice

We go first to the little New England town of North Andover, some 40 miles north of Boston. It is a pretty little town, with its large colonial-style timber-boarded homes set in well-maintained tree-lined gardens. It is the sort of town that wealthy company executives from Boston like to bring up their children in – indeed US President George Bush grew up nearby.

The area, with its proximity to such famous universities as Harvard and MIT, is also an attractive location for a growing number of high-tech start-up companies, despite notoriously high local taxation levels and other costs by US standards. Office costs though, even here, are considerably below the costs faced by UK firms and spacious

Meet The Author, PC Shareware Magazine, 1992

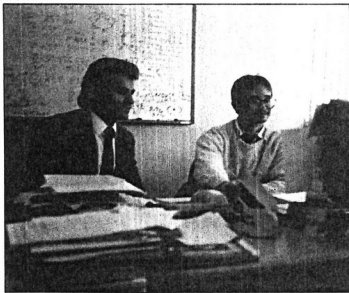
office suites, sharing a large building with other similar companies, are freely available at reasonable rents. In such an office suite I found the attractive modern offices of Trius Inc, best known as the author of *As Easy As*, the top-selling shareware spreadsheet.

Trius was founded by Paris Karahalios and Dave Schulz, by profession nuclear and civil engineers, who met while they were both working for a big engineering company. Both were interested in using PCs in engineering applications and they talked a lot about PC applications. Since Dave was always interested in programming and Paris always wanted to run his own business, perhaps it was inevitable that they would eventually get into shareware, but the idea for *As-Easy-As* came about when Dave decided to learn to program in Pascal.

Like many others, Dave had bought a copy of Borland's Turbo Pascal and one of the sample programs provided was a very simple spreadsheet. This got Dave interested in how a spreadsheet was written. Very soon he could see ways of improving on the sample program and, just to improve his programming skills, started writing his own version.

About this time Paris commented to Dave that he had been unable to find a spreadsheet that offered features such as log-log scale graphs, needed by many engineers. Very soon Paris and Dave had gathered together a list of other features which people needed but hadn't been able to find in an existing spreadsheet and, mainly for their own use, Paris and Dave began developing the first version of what was to become *As-Easy-As*.

Long interested in running his own company, Paris had by this time left his old employer to become a partner in a small firm of Consultant Engineers. However, he was still interested in getting into the computer software industry: This was where he



Paris Karahalios and Dave Schulz of Trius Inc

saw real growth occurring. So, as Dave got more interested in programming, the two partners decided to form a company to develop software that appealed to specialists such as engineers without being a specialised vertical market type of application.

The spreadsheet was ideal, being a mass-market product but one where there were sufficient specialised requirements to distinguish their product from the market leaders. Throwing aside version 1, which had only been used by themselves and a few friends, both started working hard on the development of *As Easy As V2*. After many late-night sessions, this appeared early in 1986.

Any company needs a name of course, preferably one which no one else has thought of. Perhaps this is the most difficult aspect of setting up a company. The name Trius came from the origins of the partners - Paris is Greek, Dave Canadian, both working in the USA. Hence TRI (from the Greek for three) representing three countries, and US which, apart from being where they were based, gives the made-up name a kind of classical feel to it.

Dave and Paris's early experience bears out that quality is vitally important in shareware: People will only pay for a professional quality product.

Although the early version, written by Dave for his own use, was released on a "give a \$15 donation if you like it" basis, hardly anyone responded. It was only after the release of V2, redesigned specifically for resale, that orders started to come in regularly. Almost at once they received a \$500 order from a university for a site licence. They had to hurry to get the manual finished, printing it on a dot matrix printer. That was when they knew for sure that there was a demand for the product!

That first year receipts totalled \$5,000. The next, 1987, they were ten times that - not bad for a part-time home business! Very soon, with sales still growing, they needed to take on a full-time employee to answer the phone, take orders and handle all the administration.

Needing someone who could work without supervision and develop the business for them, Dave Leonard, a friend from their old employer, was hired as office manager. A small office suite was rented near his home. Dave is still office manager and they are still in that same building, taking over more offices as their needs expand.

Dave did not have to work alone for long - very quickly two more employees were needed. From there the business grew quickly and by the end of 1988 their business was attracting attention of serious software companies and Paris was confident enough in the future to turn down a bid for the rights to *As-Easy-As* worth three quarters of a million dollars. Not bad for two years work! But he knew they could do better.

Paris explained that right from the start they decided to finance the business from within their own means - never having to worry about owing people money. It was a lot of hard work but they like to be able to sleep at night!

Shareware gave them the opportunity to get started without massive

Meet The Author, PC Shareware Magazine, 1992

marketing costs. In fact the company was started with just \$500 capital, most of which went to the lawyer for setting up a corporation. Caution prevailed throughout the company's development: Even as late as 1989 both Paris and Dave had kept on their full-time jobs, working on Trius in the evenings. Paris started working full-time in 1990, while Dave only gave up his job in 1991 by which time the company had seven other employees.

Paris explains that his philosophy is that if you undertake any project you must go into it 100 per cent. Right from the start he insisted that if they wanted the company to grow to be a big business it had to be run like a big business.

The offices themselves are modern and neat, very professional, the only slightly untidy area being programmer Dave's room, which clients don't normally see. Office rules insist that all male employees should wear suits and ties, with female staff dressing with similar formality.

Rigid standards are set for the time taken to answer the phone, respond to mail and so on. Even Paris and Dave have employment contracts, just as in any other business, and annual leave has to be applied for in advance and recorded.

There is no room for a "I'll do the work when I feel like it" attitude even when you are the owner of the business. I did however notice a bit of dissension from the dress rules from his co-owner - Dave's tie being hardly visible under a baggy sweater, while his suit jacket hung over the back of a chair.

Both have complementary skills: Paris, being the more experienced businessman, handles the commercial side of the business, while Dave has always been the programming guru and now works full-time on developing new products. Both though, plan product development jointly and Paris still handles a share of the programming himself.

While Trius is reluctant to give sales figures, As Easy As has been very

	X range	Y range	Z range	Time or Age
1	3.27196	11.10222		
2	6.43592	11.10222		
3	9.65188	11.10222		
4	12.86784	11.10222		
5	16.08380	11.10222		
6	19.29976	11.10222		
7	22.51572	11.10222		
8	25.73168	11.10222		

As Easy As

well-accepted and is by far the leading shareware spreadsheet. In fact many have suggested that it is the only spreadsheet worth considering.

The UK has always been a prominent market for the product and at one time, thanks to the efforts of Shareware Marketing, accounted for over half total world-wide sales. Currently As-Easy holds an eight per cent share of the total UK spreadsheet market.

This reputation should be enhanced by the recent release of the latest version 5. Many users are Fortune 500 and Fortune 100 companies. Trius has distributors in 29 different countries and versions are now available in many different languages - German, French, Spanish and even Chinese!

Following on from the success of As Easy As, Paris and Dave next turned their attention to the rapidly growing market for Computer Aided Design software. Drawing on their experience of engineering world, they once again found a list of features which they required but few packages, except the expensive market leader Autocad, supported.

The result was Draft Choice - a CAD package that has many features not available in software costing ten times the price. However its crucial feature was that it was easy to use - vital for the growing number of people who only want to make occasional use of a CAD system.

As new and more powerful versions of As Easy As appeared so it, like other spreadsheets on the market,

grew in size and in the amount of memory required. Soon users were complaining that few spreadsheets would run on old IBM PCs with just 256k of memory and no hard disk, or on older laptop computers with limited disk space. To meet the needs of these users Trius brought out A-Lite, a version which fitted the most popular features of As-Easy-As, omitting many of the less used additions to fit into a much smaller size program.

PROGRAMMING DIFFICULTIES

Obviously programming two such sophisticated applications as a spreadsheet and a CAD program, both of which are very performance sensitive, was not easy. I asked Dave what were the most difficult bits were. It turned out to be non-standard hardware, such as old Olivetti M24 PCs where the low-level screen output code didn't always work as expected.

These problems are even more critical on Draft Choice where for speed the program uses unused parts of the VGA memory to store the image, so it can be quickly restored after being overwritten by drop-down menus and such. Again variations in VGA board architecture are to blame.

Technical support has always been very important to Trius. It is constantly looking for user feedback to improve its product and regards each technical support call as an opportunity to see how it can improve the product. All technical support queries are logged and analysed - if three or more users suggest the same improvement it is put on the list for possible inclusion in the next version.

While the majority of users find they need little or no use of the technical support service, 90 days free support is included in the USA, with a nominal charge of \$25 per year if further support is needed after that. But in any case, if a user finds something does not work, support is free. Pre-sales support is also provided for

unregistered users.

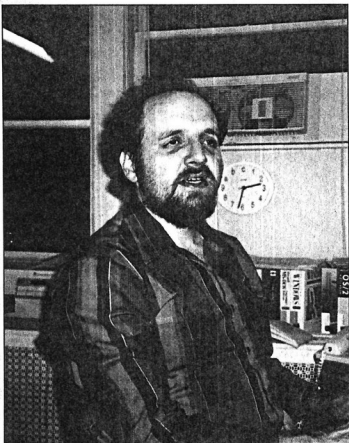
HOW TO SUCCEED AS SHAREWARE

Paris first of all emphasises the need for quality, quality and quality. Shareware after all, is the only kind of software where users can legally get to see all its features before buying. If it does not work well they simply won't buy it.

Quality has always been important for Trius – even now it employs a full-time technical writer to ensure the documentation is as clear as possible. It makes sure that the quality of service matches the quality of the product. Again Paris emphasises that a successful shareware company is no different from any other business – you have got to have a professional approach to give your customers exactly what they want.

An aggressive approach to distribution is also important: You have got to get the software into as many users hands as possible. Trius makes a point of sending out new versions to hundreds of shareware distributors and bulletin boards. It also has a policy of trying at every opportunity to get the product noticed by the press – press coverage is very important in getting a new product's name known, though you can never tell when the press coverage will come and many favourable press comments resulted from copies picked up from bulletin boards or from CompuServe.

Such publicity is vital – no matter how good the quality of your program, people won't evaluate it unless they have heard of its name. Then, however much publicity you get, quality becomes vital – they won't pay unless it is of a quality comparable to or better than the leading commercial equivalents. If you want to make a commercial suc-



Tom Rawson of J P Software

cess of shareware then commercial quality is vital.

THE FUTURE FOR SHAREWARE

With a growth in earnings and profits that are the envy of many other software companies, Trius has proved that the shareware marketing concept works. It is not about to give up the vehicle of that success easily.

Trius doesn't rule out offering its software through retail channels. Many corporate buyers still shy away from the idea of shareware – but are quite happy to pay the retail price of the product in order to evaluate it. In some overseas markets, depending on the local distributor's discretion, As-Easy-As is only sold as a retail product. Because of this ambivalence in some parts of the market, Paris feels that per-

haps the best marketing strategy at present is to market the product through both shareware and retail channels.

Yet, no matter how large the company becomes, Trius insists that there will always be a shareware version and shareware users will continue to get new versions. Paris and Dave look forward to the day when shareware will be so widely accepted that they can just go back to having one version. Citing as example the Microsoft "demo" disks that are almost full versions of the software, Paris believes shareware is such an inevitable software marketing development that one day all software will be sold this way.

Now, haven't I heard that phrase somewhere before?

THE HISTORY OF SHAREWARE FROM THE 1980S TO THE 1990S

A FEW miles away from Trius lies the Boston suburb of Arlington, an older community situated midway between the university town of Cambridge and the historic and stylish town of Lexington. Cambridge, whose elegant college buildings, hosts of bookshops and student throngs look just like its English name-sake, is the home of Harvard University and the Massachusetts Institute of Technology (often known just as MIT) which like to think that they are the two leading US universities.

Arlington, a couple of miles up the road, is a long-established, slightly seedy looking community, favoured by many students from the nearby universities and very reminiscent of how the less popular areas of our own English city of Cambridge were before they were redeveloped. This student

Meet The Author, PC Shareware Magazine, 1992

As-Easy-As was originally a DOS program. What things do you have to “balance” when you write DOS programs?

Well, it was an evolving process. When we first got started, extended memory was not a thing and 640K was a luxury for most PC users before the mid 1980s. Since files were compiled into .COM files that needed to be loaded in a single 64KB memory segment, because they lacked allocation information, that presented another limitation. I remember multiple iterations of optimization to try and keep the code small and fast. As systems were getting better and available memory larger, we kept enhancing our memory models to use more RAM, e.g., the expanded/extended memory to 1 MB, above the directly addressable 640K, and later to use up to 500 of 16k-pages or extended RAM (up to 8 MB!). Eventually, we even implemented a method to simulate up to 2 MB of virtual RAM on disk, for users that needed the additional memory, but their system did not have it.

Dave was primarily the person doing the optimizing and he was *very* good at it. I think I’ve mentioned before how impressive his understanding of computers and software was! Early on, there were no

math co-processors either, so, for example, if you were writing code that would be using a lot of math operations, you had to think of possibly implementing direct bit manipulations by shifting register bits left or right. Fortunately, Turbo Pascal allowed you to include in-line machine code (Assembly Language), which we used a lot to manipulate hardware directly and get optimization that was not available in standard Pascal code. We'd use assembler to optimize specific chunks of code, then take the assembled hex code and include it in Pascal functions and procedures.

One of the drawbacks, of course, was that inline assembly code is highly dependent on the target platform (CPU architecture and memory model). For the most part, this was not a problem, though we did encounter inconsistencies moving from 80286 to 80386 and the 80387 math co-processor. Early on, we had conditional branching for running the program on systems with a co-processor and those without one. The user had to know what they were running and start the program with an appropriate command line option, `/N486` to bypass CO-processor error trapping.

The other area where we used in-line machine code a lot, was to take over from the operating system and write directly to video RAM, bypassing the BIOS. Again, this resulted in better, faster screen writes and updates, but opened the door to problems with displays or display adapters that were not 100% conforming to published specs. More command lines to start the program in a “special” mode for such systems, if I remember correctly, `/ATT` if you were using an AT&T system with a monochrome monitor, `/ATT2` if you were using a Toshiba laptop, `/E` for enhanced graphics adapter, `/EM` for monochrome EGA monitors, and so on... I can't remember them all by heart, but we ended up with over 30 command line switches and hundreds of combinations. However, most PC users at the time were hackers and they could figure out the right combinations to get the most out of their PCs.

What's an example of something you can do in DOS programming that's hard to do in Windows or Linux?

The ability to generate the smallest possible files was always a goal for us and DOS was good at it. Not sure

how realistic this example is, but in the early days I would demonstrate to people how simple it was to write a program that would display “Hello World,” using the built in DOS Debugger and the generated .COM file was only 22 bytes. I think someone had reduced that to even 20 bytes! What’s the smallest Windows .EXE that can do that? I believe because DOS allowed access at the lowest level, it made it possible to write very tight and fast code. I’m sure there are many more things one could do in DOS that cannot, or at least not easily, in Windows. Linux may be a bit more accessible.

What things have gotten easier in programming?

What I think was the advantage also made it harder. No built-in function libraries or external ones that could be linked from your code, you had to write pretty much everything you needed (well, with the exception of basic functions). Writing code in Windows makes thousands of built-in functions available to you. And, that’s my second pet peeve! Load everything, whether you need/use it, or not! So what if your code needs 16 GB of RAM? RAM is

cheap, right? No need to optimize, no need to look at how this affects execution speed, or how it impacts other apps. Applications now have assumed the behavior of gasses, “they expand to fill the volume they are released in.” I have vivid memories of Dave and I spending hours on what to do to make the program 128 or 256 bytes smaller, yes bytes. Because of the limited resources available on the hardware systems, anything you could do to save even 128 bytes was a plus!

What are some “tricks” or methods you used to write a DOS application like this?

Not sure I can remember all the “specific coding tricks,” but as I have mentioned in many other places, taking over certain operations from the operating system and bypassing the BIOS was a main goal. Processing keyboard interrupts and video RAM interrupts were almost all taken over. Also, coming up with our own, optimized Run-Length-Encoding for representing data was a big help. Manipulating the registers directly to perform routine integer operations was also used to speed up execution. We

made abundant use of overlay files to allow us to run the program on systems with limited resources, having a resident portion of the code that was always loaded and swapping different overlay segments in and out of memory, as needed.

As-Easy-As was originally written in Turbo Pascal, then Delphi. Why change to Delphi?

We moved to Delphi when we started porting the program to Windows. Delphi is essentially an enhanced, object-oriented version of Turbo Pascal, so the code conversion was not as bad as it sounds. We did, however, have to re-architect the program to take advantage of the OOP model. Delphi's visual development environment and support for GUI applications made development a bit easier. Delphi's introduction of a RAD (Rapid Application Development) approach with visual tools for building applications was also a great tool.

How did you “discover” computers?

As part of the 3rd year curriculum, I had to take a course in FORTRAN. I found it interesting, but for some reason it didn't click! It may have been that we didn't have immediate access to the computer. The university was using an older Digital computer, with punch cards. We'd write the code long hand in a notebook, then go to the punch card terminals room and type it, to generate the punch cards, then put together the punch cards with a rubber band and drop them off in the computer operator inbox (the first card had our name, course, etc.). We'd then have to wait until the next day, go to the computer operator room and get our printout (on the wide green lined computer paper). If the program had worked and gave us the correct (expected) results, we'd turn it in for grading. If it failed to execute, or gave the incorrect results because of errors in the coding, then we'd go back to the punch card terminals, make the corrections, generate new punch cards, drop them in the inbox of the computer operator and wait till the following day to get our results. And, if the programming was complex and/or

you kept making errors in your coding, this iterative process would take days... Not a pleasant experience.

Then, in my senior year, I took a course in “numerical analysis with computers” and the university had just acquired a CDC Cyber 6400 main computer.

Although we still used punch cards, turnaround time was now 1-2 hours. What progress - you could now afford to make coding mistakes and still get your code corrected and resubmitted a few times all in the same day! I started getting real interested, and getting “the bug,” but senior year was almost over, so...

I started on my master’s degree the following year, same university, same CDC computer, but now we didn’t have to use punch cards! We used DECwriters, connected to the computer! You type your code, submit it real time, it compiles, it executes and you get the results printed on the DECwriter, right away! At the same time, a course in Monte Carlo simulations with computers (required for grad school), taught by a brilliant professor and a Christmas present of a TI-59 programmable calculator - now I was hooked! On top of it I get unlimited access to the PDP-9 computer of the nuclear center at the university. How many ways can you spell Happy! Still coding mostly

in FORTRAN, using TI OpCodes to code on the TI-59, taught myself Basic and then moved on to Pascal. PCs were not available yet, Amiga and Commodore came out, but I looked at them more like game machines and I never got into computer games, so I never worked with them. I did, however, purchase a TI-99/4A color console, which I used to write some short programs for operating a family restaurant. Became very familiar with CP/M and worked on a number of small side projects writing code for insurance companies, payroll companies. I had also started working and I now had access to an IBM System/360 and the job to write code to solve equations. It didn't feel like a job, it felt like someone was paying me to have fun! Then, when the IBM PC came out it was all of a sudden, access to computers at a different, much lower level! Unix, C, Assembler,...

What was your first personal computer? What did you find exciting about it?

I mentioned that the first computer I had fun with was a CDC Cyber 6400 mainframe. It gave me the

opportunity to write code to numerically solve mathematical models that up until then, to me, seemed unsolvable! But the real joy came when the PC and DOS arrived, because I could now access the computer at its lowest level! I could not afford a true blue PC, so I bought a compatible, SANYO MBC-550-2 with 256K RAM, two floppy drives and a monochrome monitor. I reasoned that I needed to buy it because I was doing a side job for a small company that was using Lotus 1-2-3 for their business and I needed to have something at home to work on the project, so I wouldn't have to go to their location every day. It was an excuse to buy the PC, but...

I was so happy when I brought it home, unpacked it and fired it up. My own computer! The client provided me with a license for Lotus 1-2-3 the next day, and to my dismay, Lotus 1-2-3 would not work on my SANYO. I came to find out that in order to make it work, I needed a piggyback board to make the video compatible with the IBM PC. It came with DOS 2.11 and CP/M 86. I bought the board and was able to run the program, but found quite a number of incompatibilities along the way.

What was your first programming language?

I started coding in FORTRAN and continued coding in it for many years. When I started using PCs, I would also code in Basic, Pascal (and still some FORTRAN).

- Although not a high level language, I wrote lots of programs for the TI Professional Program Exchange (PPX) on the TI-59, mostly on nuclear-related subjects.
- A Payroll system for a restaurant with about 30 employees (TI Basic)
- A Quality control system for safety related engineering drawings (MS Basic)
- An Inventory Control System (Pascal)

Did you use computers and programming to help you in your Nuclear Engineering program?

Yes, I spent a lot of time during my graduate studies and after I started working, in front of a punch card machine, a DECwriter, or a monitor working on nuclear-related programs. Some of them were utilities to prep data, others were programs to calculate complex multi-member radioactive decay and some

were custom, specialized code to inject into large code systems we'd acquire from one of the National Labs.

Do you still do programming today?

I don't do much programming these days, and haven't done for many years. I manage a group of very good developers, so I leave the coding part to them. On the rare occasion that I code, it's usually C# and SQL. Once in a while, I get personally interested in a topic and will spend some time at home with SciLAB.

Have done very little coding on the Mac, many years ago. We are a Microsoft house using Windows. However, on my system, I am using VMWare running Linux Ubuntu, ArchBang and I just installed FreeDOS (not sure how much time I will have to spend with it).

I don't code anything worthwhile or of interest to other developers. Those days are gone. Developers now have access to great tools, no need to rely on old relic coders like me. I have to say though, that some nowadays may not have the instinctive knowledge of base principles that we needed to have back in the

day. Sure, you can get ChatGPT, or some other on-line system to instantly convert a number from Base-10 to Base-16, or do binary arithmetic for you, but I think it's important to understand what is the process that produces those results, that it's not just magic. It's important to understand how registers work, how the CPU is just a very fast processor of straight forward operations, it's not magic! Did you get that this is one of my pet peeves?

Do you have any examples of interesting ways that people used As-Easy-As?

There were many customers that had used As-Easy-As for niche applications, it's hard to remember them all. Here are some that come to mind.

A user wrote a spreadsheet that would get real time stock quotes through a paid service (using a dial-up modem) and then dump the raw data to As-Easy-As for DOS through the serial port. He had come up with algorithms in As-Easy-As to analyze stock prices, identify trends and try to determine good/bad investments. We never followed up to see if they made any money from those investments.

A major West coast fast food chain's (2000+ locations) corporate Dev team integrated As-Easy-As for DOS into their management system at every location, with a connection to headquarters, to manage daily sales, cost analyses, goals, etc. I was directly involved in that integration and learned a lot about "corporate" software deployment workflows from them.

The Chinese version of As-Easy-As for DOS was incorporated as a mandatory course in the Taiwanese Computer Technical Schools curriculum. Multiple instructional books were written in Chinese (I have copies of some of them). As an aside... As-Easy-As was also translated into Portuguese, Italian, Spanish, German,... We did all the implementations at TRIUS. We'd extract all strings from the code (menus, sub-menus, error messages, function names, etc.), we'd send them to our partners/licensees for the foreign versions, they would translate them and then send them back to us. We would then go back and replace all strings in the source code with the foreign translations. I was primarily responsible for doing all these foreign implementations. The Chinese version presented a problem because Chinese characters need 2 bytes for their representation, so the BIOS, built-for 1-byte ASCII character display, could not

accommodate them. Our Chinese partner had to come up with a TSR, partial BIOS replacement that would intercept interrupt calls from the program to remap and display 2-byte characters on the screen. A fun project.

Mazda US headquarters developed an integrated leasing system using As-Easy-As for DOS that was used by every US Mazda dealer.

A special custom version of As-Easy-As, for the HP Palmtop, was used by a hospital group to schedule and manage daily rounds.

A user interfaced As-Easy-As to a CNC table, through the serial port, and used it to drive a cutter to create all sorts of designs based on complex trigonometric equations.

How did you decide what features to add in the next version?

Anytime a user communicated a request for a new feature, it was added to a long, running list of future changes. When one of us thought of a new feature, it was also added to the list. If a bug was reported or

discovered, it was also added to the list. On a daily basis, we'd sit down and review the list and assign priorities, using a multi-factor grading system.

Reported bugs and discovered problems were pushed to the top of the list and were addressed immediately, or not, depending on factors like...

- Is this an issue that affects some core programming and capabilities of the program?
- Does it involve a commonly used (by many users) function or feature?
- Does the user reporting have a dire need to use it in some immediate work?

Then new features were reviewed based on...

- The number of users requesting it
- Whether it would have a wide impact, used by most, many, a few potential users
- Effort level to implement the new feature
- How close are we to a new release

Although I can't say it never happened, usually, the decision for implementing a new feature was not based on whether we added it to the list, or it was suggested by the user. We tried to make the decision on merit and factors like those listed above.

Was As-Easy-As used mostly by engineers, or was it used by office workers too?

Although we don't really have any real statistics, I'd say it was the other way around. As-Easy-As was mostly used by individuals and small businesses to make managing their businesses easier, without having to pay \$400 for Lotus 1-2-3.

The engineering portion was primarily driven by our needs. Having said that, since many engineers in those days were PC hackers and they liked to build things, they started adopting As-Easy-As, once they discovered they had access to additional engineering functions and formulas.

We also found that the program was used by a large number of small investors that built their own formulas to predict future stock performance, since they thought "they" had the secret sauce.

79
TRIUS, Inc.,
231 Sutton Street, Suite 2D-3,
PO Box 249,
North Andover, MA 01845-1639
United States of America

U.S. Patent 4,444,444
U.S. Patent 4,444,444
U.S. Patent 4,444,444
U.S. Patent 4,444,444

4th July 1993

Product Number/Registration: 20-0117-75324

Technical question resolution Ref.# 93061803
Line Styles - Paper Sizes

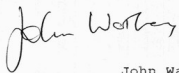
Dear TRIUS,

Thank you for your reply to my recent enquiry about a difficulty I was having with Draft Choice. Your explanation was exactly right; I had failed to appreciate the importance of the File, New dialog box in this context.

Can I express again my great satisfaction with this excellent product - I also have T*rb*c*d, and Draft Choice knocks it into a cocked hat! The new features in Version 2.0a, especially the enhanced dot matrix printing facilities, are much appreciated.

My thanks again,

Yours sincerely,



John Warbey

Customer letter to TRIUS

Thank you for your reply to my recent enquiry about a difficulty I was having with Draft Choice. Your explanation was exactly right; I had failed to appreciate the importance of the File, New dialog box in this context.

*Can I express again my great satisfaction with this excellent product – I also have T*rb*c*d, and Draft Choice knocks it into a cocked hat! The new features in Version 2.0a, especially the enhanced dot matrix printing facilities, are much appreciated.*

dr Wincenty Raczkowski

PL 00000 00000 0000
000000

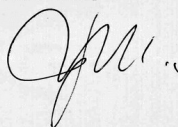
ASP Ombudsman
PO Box 5786
Bellevue, WA 98006
U.S.A.

Tymien, September, 28, 1993

Dear Sir,

I'm very glad to tell You, that the program named
DRAFT CHOICE
is for me an economist a very simple and perfect program to
draw figures and graphics what I need.
I'm working by a PC/XT 640 KB machine, with 2 FDD 360
KB, without HDD, with HERCULES card, b/w monitor mono and 9 pint
printer (ROBOTRON compatible with IBM Graphics).
Now I please to be a registered user of this program.
Because I'm a invalid man, and my disablement benefit behave
only \$ 150 per month, so I cannot send You any donation.
I thnak You for understanding my problems .

Yours truly



Customer letter to TRIUS

I'm very glad to tell you that the program named DRAFT CHOICE is for me an economist a very simple and perfect program to draw figures and graphics what I need.

I'm working by a PC/XT 640 kB machine, with 2 FDD 360 kB, without HDD, with HERCULES card, b/w monitor mono and 9 pint [sic] printer (ROBOTRON compatible with IBM Graphics).

Now I please to be a registered user of this program. Because I'm a invalid man, and my disablement benefit behave only \$150 per month, so I cannot send You any donation.

Date: 11-8-91

E. M. Da Silva
11 Montpelier Ave.
North Andover, MA 01845-0249
781-442-1171

Mr. Paris Karahalios - Trius Inc.
231 Sutton St, Suite 2d-3
P.O. Box 249
North Andover, MA 01845-0249

Dear Mr. Karahalios:

A few weeks ago I ordered a copy of PIVOT. Naturally I had to delay installing it so you can imagine my disappointment when I finally found a few minutes to get started and it didn't seem to work. Apparently there was nothing on the disk.

The next day I called your sales office, assuming I would ultimately get some satisfaction but not before the usual complications and delays. Lots of account number verification, evidence of suitable documentation, appropriate authorizations and a trail of correspondence, of course.

You can imagine how pleased I was to be transferred to a young man who immediately projected a readiness to assist me as a Customer. With a few brief questions he determined that the solution required a replacement disk and proceeded to make the necessary arrangements. Three days later the Disk arrived. What a surprise!

Unfortunately I don't remember the young man's name but I hope you will extend my appreciation for the responsiveness and courtesy he extended.

Congratulations to you and your Staff.



Edward M. Da Silva

Customer letter to TRIUS

A few weeks ago I ordered a copy of PIVOT. Naturally I had to delay installing it so you can imagine my disappointment when I finally found a few minutes to get started and it didn't seem to work. Apparently there was nothing on the disk.

The next day I called your sales office, assuming I would ultimately get some satisfaction but not before the usual complications and delays. Lots of account number verification, evidence of suitable documentation, appropriate authorizations and a trail of correspondence, of course.

You can imagine how pleased I was to be transferred to a young man who immediately projected a readiness to assist me as a Customer. With a few brief questions he determined that the solution required a replacement disk and proceeded to make the necessary arrangements. Three days later the Disk arrived. What a surprise!

CHARLES HODGSON

11 October 1991

The President
TRIUS Inc
231 Sutton Street
Suite 2D-3
North Andover, MA 01845
U S A

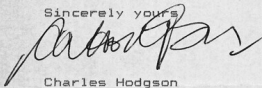
Dear Sir

AS-EASY-AS
ALITE
PIVOT!

I am the registered user of your above-named products which I to date have found to be excellent programs, notwithstanding one or two defects which I understand are being looked at by your software development division.

However, my main purpose of writing to you, is to extend my most sincere thanks to your Technical Support staff who have been outstanding in providing me with solutions to the problems that I have experienced so far. What has really impressed me has been the promptness of their replies, never later than 24 hours after my FAX enquiry. I take my hat off to them. I think a few other shareware developers should take note of your support staff's quick responses.

Sincerely yours,



Charles Hodgson

Customer letter to TRIUS

I am the registered user of your above-named products which I to date have found to be excellent programs, notwithstanding one or two defects which I understand are being looked at by your software development division.

However, my main purpose of writing to you is to extend my most sincere thanks to your Technical Support staff who have been outstanding in providing me with solutions to the problems that I have experienced so far. What has really impressed me has been the promptness of their replies, never later than 24 hours after my FAX enquiry. I take my hat off to them. I think a few other shareware developers should take note of your support staff's quick responses.

Are there any features that you wish you had added to As-Easy-As for DOS?

I can't really think of any feature in particular, though I'm sure we had a few pages worth of features for future consideration, when we stopped development of As-Easy-As for DOS. I know that at some point, we decided to branch off development and create a lighter version (Alite), where we added the capability of TSR (Terminate and Stay Resident). Users could load the program and then have access to it over any running application using a hot key combination - a breakthrough at the time, for the single task operating system.

One thing that comes to mind is text processing features. I have to put it in context... In those days and running a single task operating system, if you were working in a spreadsheet or a database program and you needed to write a letter, you'd have to exit the current application, start your text editor (or word processor), write the letter, print or save it, exit the app and restart the app you were previously working on to continue your work - a very time consuming process. So, what a lot of us that spent most of our time in As-Easy-As would do, is to write short letters and notes and print them from within the program, so we would waste time exiting and restarting multiple applications. To do that effectively, we needed "some" formatting options, like embed CR/LF, right justify, bold and italicize words, word wrap, etc. We started down that road, but never really had an opportunity to implement those features. By the way, Alite was developed primarily to eliminate the exiting/restarting of all these apps.



Shareware authors have stepped in here to fill the gap left by the bigger vendors. I found three shareware spreadsheet programs small enough to fit on a 720K floppy disk. The smallest is a program called *ALite* (Trius Inc., P.O. Box 249, North Andover, MA 01845; 800-468-7487, 508-794-9377; \$20 plus \$6 shipping). Taking only about 200K of disk space and requiring only 256K of RAM, *ALite* still gives you 2,048 rows by 128 columns, EGA support, and five types of graphs (see Figure 1). It directly supports *Lotus 1-2-3* .WKS and .WK1 files, and uses .WKS as its own format.

ALite is a scaled-down version of Trius's original spreadsheet, *As-Easy-As*, Version 5.0 (\$69 plus \$6 shipping), which provides 8,192 rows by 256 columns in about 320K of disk space. It is also .WKS- and .WK1-compatible, offers ten different graphs, has EMS and mouse support, and can record macros. You can download the latest shareware versions of *ALite* and *As-Easy-As* from Trius's BBS at 508-794-0762 or from Library 8/ASP Applications of the PC MagNet Utilities/Tips Forum as ALITE.ZIP and ASEASY.ZIP, respectively.

InstaCalc (FormalSoft, P.O. Box 1913, Sandy, UT 84091; 801-565-0971; \$49 plus \$5 shipping) uses its own file format but can import and export .WK1, *dBASE*, *DIF*, and *ASCII* files. It provides 4,096 rows by 256 columns; supports EMS, EGA graphics, a mouse, and file encryption; and uses only 390K of disk space.

Alite in PC Magazine, 1992

Shareware authors have stepped in here to fill the gap left by the bigger vendors. I found three shareware programs small enough to fit on a 720K floppy disk. The smallest is a program called Alite ... Taking only about 200K of disk space and requiring only 256K of

RAM, Alite still gives you 2,048 rows by 128 columns, EGA support, and five types of graphs (see Figure 1). It directly supports Lotus 1-2-3 .WKS and .WK1 files, and uses .WKS as its own format.

Alite is a scaled-down version of Trius's original spreadsheet, As-Easy-As, version 5.0 ... which provides 8,192 rows by 256 columns in about 320K of disk space. It is also .WKS- and .WK1-compatible, offers ten different graphs, has EMS and mouse support, and can record macros. You can download the latest shareware versions of Alite and As-Easy-As from Trius's BBS at ...

Programmers are the best at spotting their own bugs after-the-fact. Are there any bugs in As-Easy-As that you look back on and think "I wish we'd fixed that"?

I don't want to say there are no bugs in the program (which I have not used in a while). I'm sure someone can find some. However, believe it or not, we spent a lot of time testing and correcting and re-testing before a new release.

As you most likely realize, our testing was not just the user interface - that was the easy part. In those early days we didn't have the luxury of "function

libraries,” in the context of what’s available today. For example, If we wanted to add a function to calculate mortgage payments, we couldn’t just call some Payment function from some library. We had to come up with the mathematical formula from base principles, program the calculations needed directly into Turbo Pascal and once implemented, we had to then use the function in As-Easy-As multiple times and perform hand-calculations to verify the results. Lots of very long nights, in particular for some of the more complex functions.

We had a small group of users that had been using the program from the early days and they’d help us with beta testing every new release. They were a great asset! I know this was not part of your question, but even beta testing was executed as a process. Every beta tester had a signed agreement with us outlining their responsibilities, their benefits, included an NDA, etc.

We wouldn’t use this group for minor releases, we would use them for major version releases that included testing over a few weeks and they would make a significant time/effort investment. For minor maintenance releases, we used a sub-set of the group

made up of users that were willing to intensively test over 2-3 days. And, for complete transparency, if the release only included a couple of minor changes, we'd sometimes rely only on our internal testing. Luckily, this did not get us into any major trouble (which it had the potential to).

About the manual

Registered users got a printed manual. How was this written?

I wrote the first manual using PC-Write (another shareware program by a good friend, Bob Wallace). No pictures, no graphics, just ASCII characters. Even the name of As-Easy-As on the front cover was written using the ASCII line characters! I printed it on a HP LaserJet II printer on 8.5"x11" paper, then had it reduced, printed and bound by a local printer. We printed 100 of them in the first run and they were depleted in about a month. The next version was also written in PC-Write and we produced 500 this time, which lasted us about 3 months.

I also wrote the next couple versions of the manuals, this time using WordPerfect and we had cover art designed professionally for us. After that, the manuals were written by a couple of employees, and I would review and approve them. When we ported the program to Windows, I took over writing the manuals again.

What system or word processor did you use to produce the manual?

The early user manuals were written using PC-Write.

We started using WordPerfect with subsequent versions and then moved to Corel and Aldus PageMaker before moving to Microsoft Word.

It was always a balance game. We liked PC-Write because we had absolute control over it (embedded control codes, etc.), but it was not a true word processor, so as the look and feel of the manuals changed, it became harder to get the desired results.

We moved to WordPerfect and liked it for its embedded character mode, where you could see what was happening under the surface (remember, we

were all hackers, in the good sense of the word). When layout demands became even more needed, we moved to the other products which were more “publishing apps.”

What was the process to write the first manual?

Well, we have to take things in context. Back in the old days, there used to be at least two or three manuals, a user’s manual, a reference manual and (for hardware) a service manual. Each was laid out differently since they served different purposes. For software, there were two; a user’s manual that took the user step-by-step to achieving certain tasks, and a reference manual that documented what each section of the software did.

In writing the first As-Easy-As manual, I took a hybrid approach. Menus structure in the program already provided the outline of a “Reference Manual” for me, so I started with that, explaining what each menu command did, filling in with some general spreadsheet sections and use cases for the menus, where appropriate.

Writing manuals for subsequent versions of As-Easy-As, and other software products, I used different approaches with my favorite being to indeed write an outline and then follow it to write the manual, while still providing use-cases as examples of solved problems.

What fonts did you use for the manual? What's your favorite font for print?

Font selection in the early days was not great, in particular within the confines of the PC. If I recall correctly, Times New Roman and Courier were the most common ones with Helvetica starting to gain some ground at the time. I believe I used Helvetica for the first manual, and a mix of fonts later on, including Garamond at some point.

Nowadays, I personally don't generate a lot of material for print, but I find myself using a couple of fonts more than others. I mostly use the default fonts (Arial, Calibri, Aptos,...). For printed matter readability, I like Times New Roman, but most of the people I know hate it - it's not one of the modern

fonts. I guess that's personal preference, just like ice cream flavors.

What was the editing and publishing process like?

For the first manual, I wrote it, Dave quickly scanned it and that was it. I did go over it a few times for errors, but we wanted to get the information out to the users who were already paying for our product, so we released it, knowing that I'd have to go back and re-do some of the work.

Once we had some employees and we could share the workload, we did have some of them review the manual (even some that had no computer knowledge/experience) and provide feedback on language, and ease-of-understanding. I was so close to the program that whatever I wrote already made sense to me, but I might be the only one it made sense to. These reviewers were instrumental in improving the user manual.

When it got to versions 5.2, 5.7, etc. we employed a small agency to proofread what we wrote, but I have to be honest, I don't think their contribution was

significant. Sure, they may have caught a few grammatical errors, but no substantive contribution.

What was the printing process like? How did you print the manual?

I mentioned that I wrote the first As-Easy-As manual using the PC-Write shareware program, written by Bob Wallace, a friend and the 9th employee of Microsoft. Once it was completed, I printed it on 8.5"x11" paper, single sided, on a LaserJet II printer. I took the printed copy to a local printer, they photo-copied it at 50% size reduction, so they could generate 8.5"x11" double sided spreads, then they cut them in half and perfect bound them with an 80# glossy cover. As you can tell, they spent a bunch of time to lay out the half-size spreads and the final product didn't look like much. However, it's hard to describe the pride and euphoria when I picked up 2 boxes of manuals (I believe 50 manuals in each box) and brought them home. It felt as if I had achieved something significant.

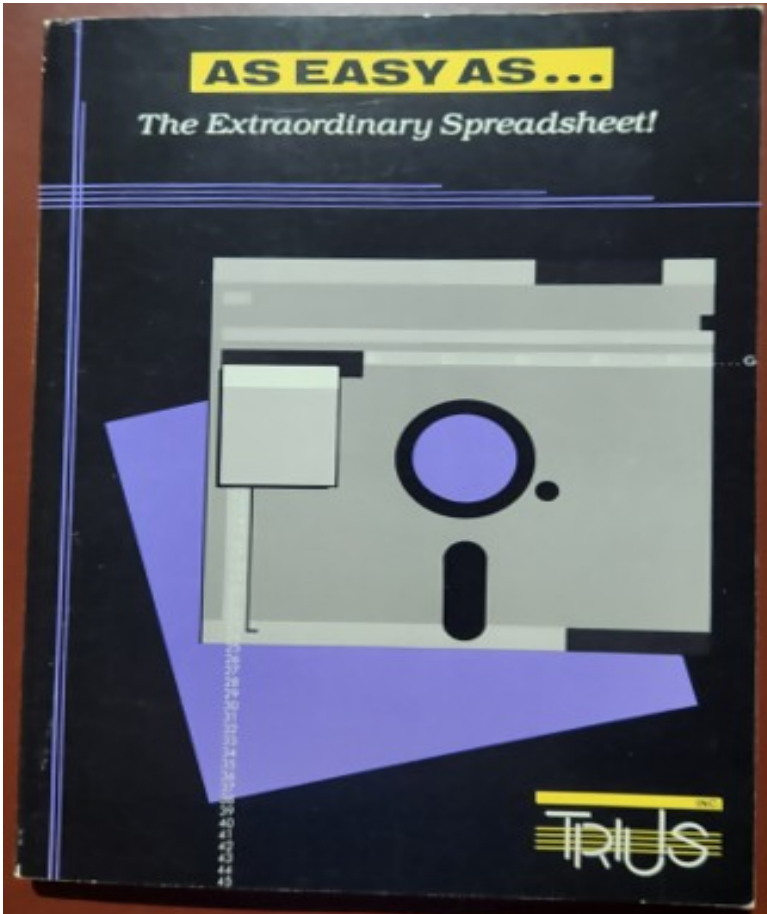
We followed the same process for a few of future user manual versions, and then we moved to providing

digital files to a larger print house we had started using, who also had in-house Aldus Pagemaker and Corel capabilities that made it a lot easier to accept and lay out the material we gave them.

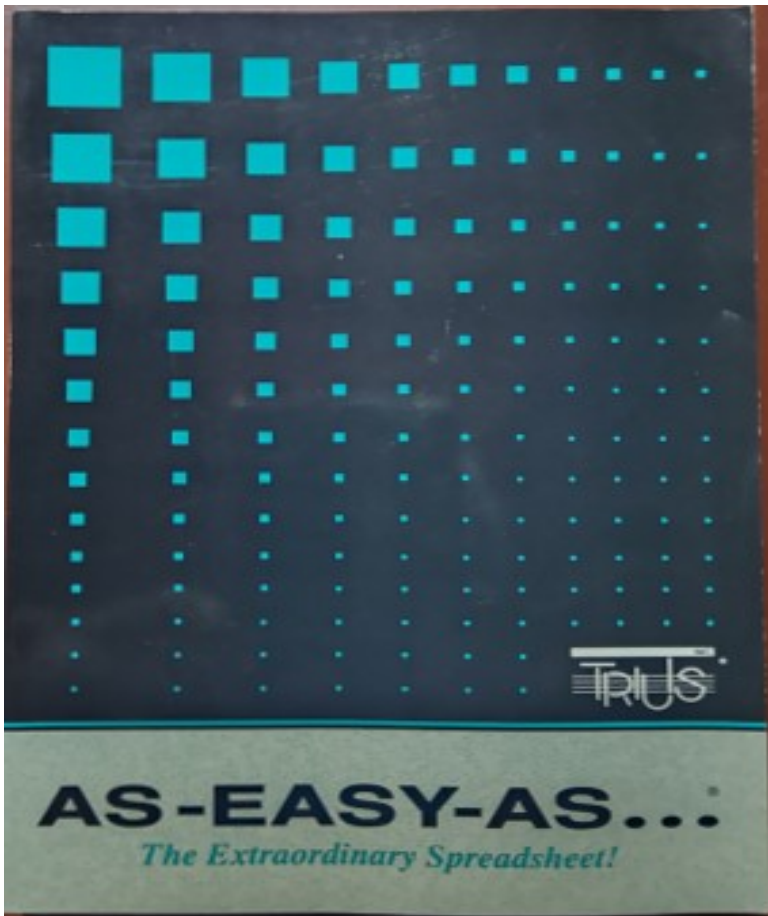
What did the manual look like?

I have some of the manuals and have attached photos. I may have some earlier ones, as well, somewhere in the basement, but I'll have to really look for them. I wish I had kept one of the first batch of the manuals we produced.

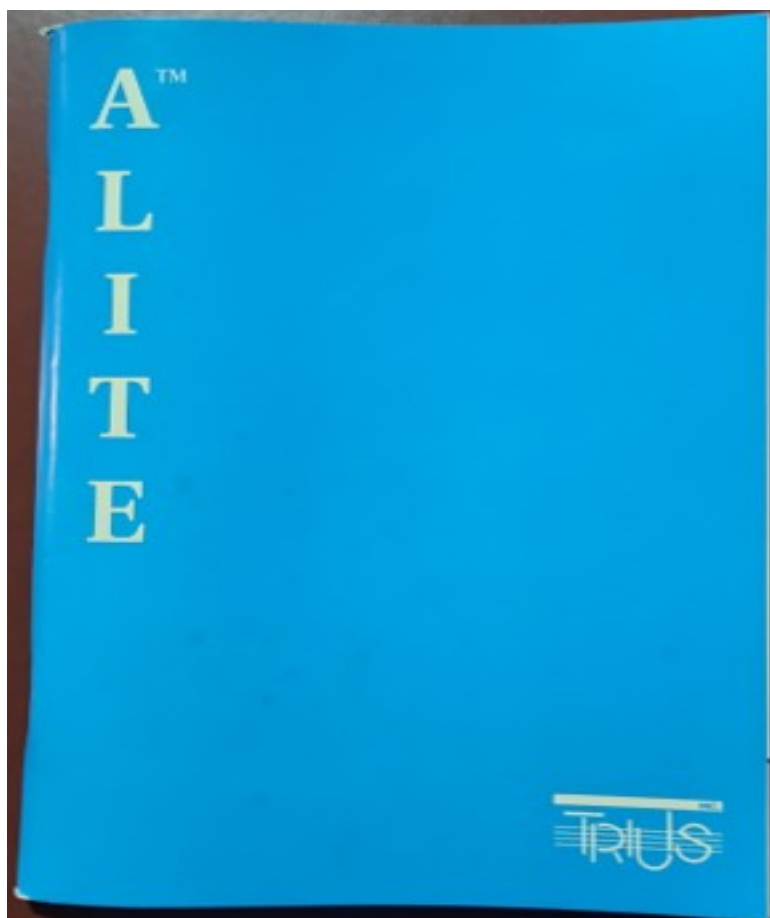
Here are a couple of versions of the As-Easy-As and Alite (the lighter, TSR version) user manuals:



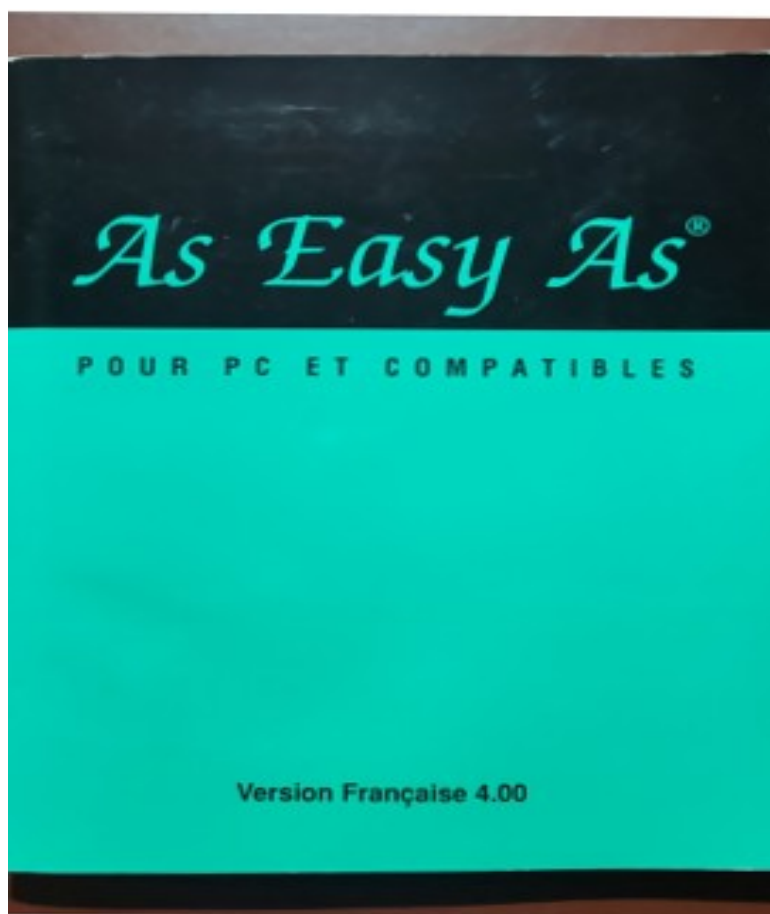
As-Easy-As 4.0 manual



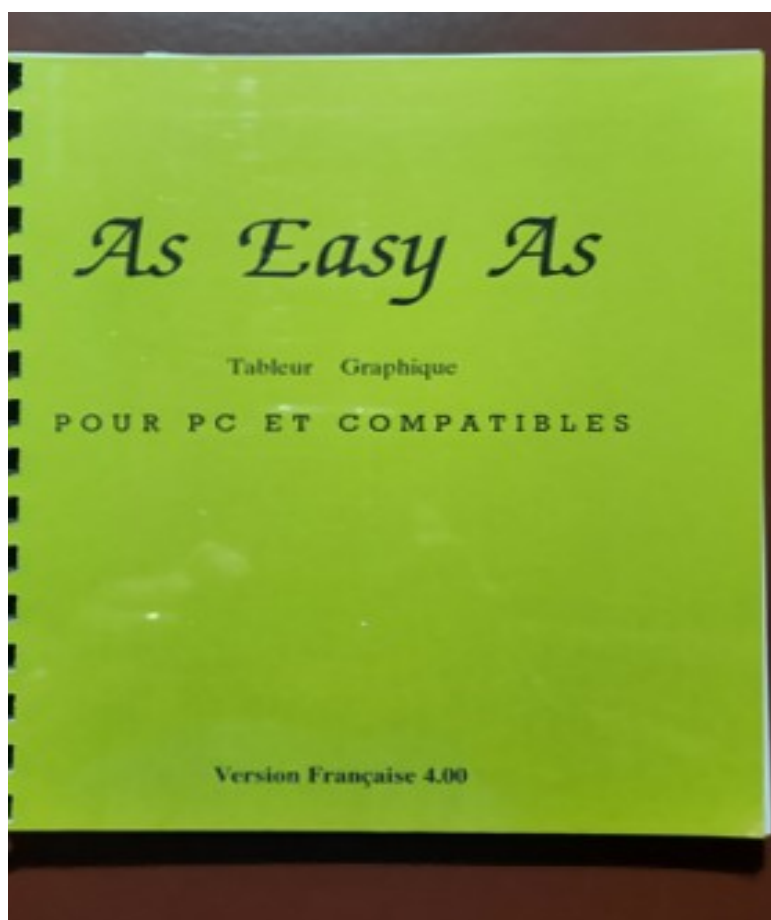
As-Easy-As 5.7 manual



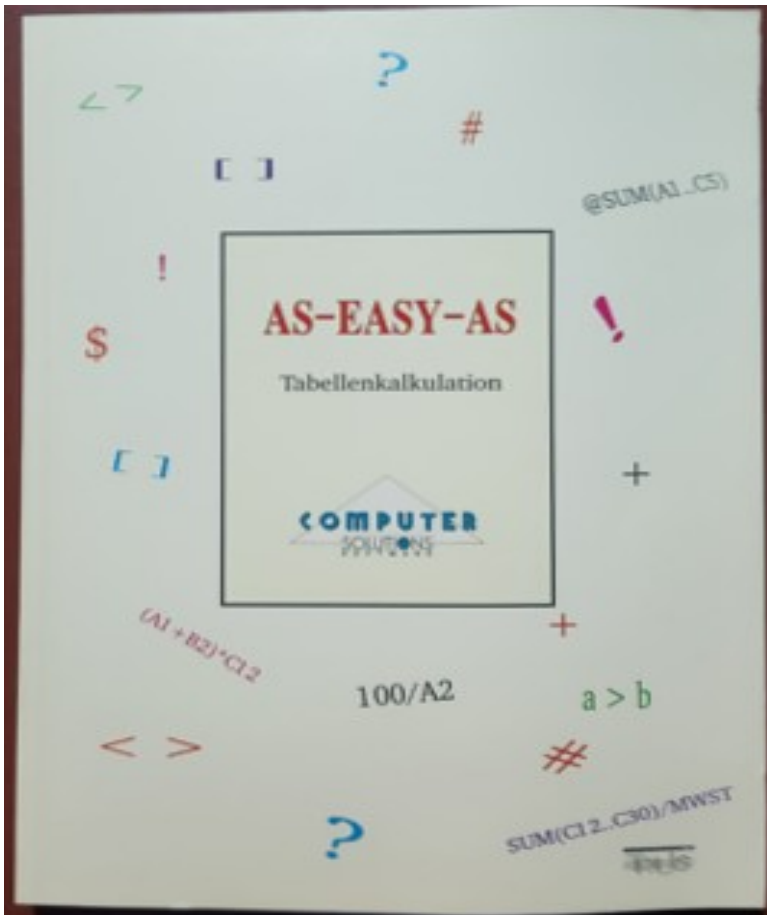
ALITE manual



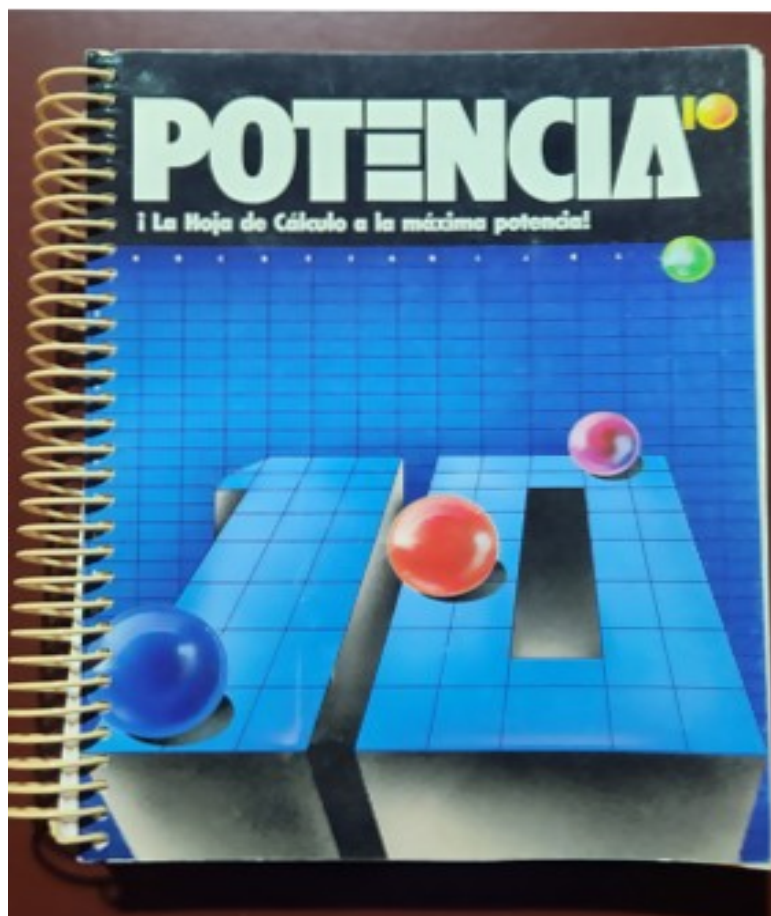
As-Easy-As 4.00 manual, French edition, published in France by SIR Informatique



As-Easy-As 4.00 manual, French edition



As-Easy-As manual, German edition, published in Germany by Computer Solutions GmbH



"Potencia 10" manual, Spanish edition



Eccecutive manual, Italian edition



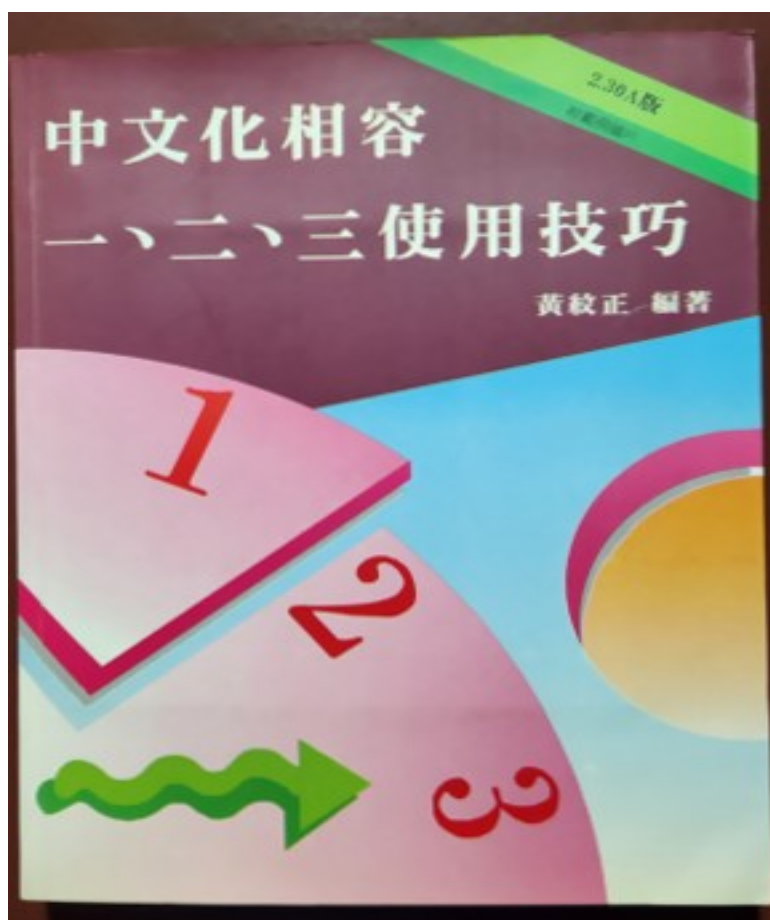
As-Easy-As books in Chinese, used in teaching spreadsheets in Taiwan (required course in Computer Technical Institutes). Published by Race International LTD



As-Easy-As books in Chinese, used in teaching spreadsheets in Taiwan (required course in Computer Technical Institutes). Published by Race International LTD



As-Easy-As books in Chinese, used in teaching spreadsheets in Taiwan (required course in Computer Technical Institutes). Published by Race International LTD



As-Easy-As books in Chinese, used in teaching spreadsheets in Taiwan (required course in Computer Technical Institutes). Published by Race International LTD



As-Easy-As books in Chinese, used in teaching spreadsheets in Taiwan (required course in Computer Technical Institutes). Published by Race International LTD



As-Easy-As books in Chinese, used in teaching spreadsheets in Taiwan (required course in Computer Technical Institutes). Published by Race International LTD



As-Easy-As books in Chinese, used in teaching spreadsheets in Taiwan (required course in Computer Technical Institutes). Published by Race International LTD

Thank you!

Jim Hall (editor)

I loved reading about the history of As-Easy-As spreadsheet! Thanks to Paris Karahalios for this outstanding interview about As-Easy-As and TRIUS Inc.

I hope you enjoyed reading this interview collection about the history, development, and impact of As-Easy-As spreadsheet. This interview focused on As-Easy-As for DOS, because that was the version I loved using in the 1990s, and still use today on FreeDOS.

It's hard to convey in words how As-Easy-As made my work easier, but a demonstration might make it more clear. Read on for a demonstration of how I used As-Easy-As to perform data analysis in undergraduate physics labs.

How I used As-Easy-As

When I was an undergraduate physics student, I used As-Easy-As for all of my lab analysis. It didn't take long for me to learn how to navigate the application using the keyboard. Like other DOS spreadsheets, As-Easy-As used the / key to enter the menu, displayed in the upper left corner of the screen. As-Easy-As highlighted key letter for each new menu action, making it easy to tap the next key for the submenu or action that I needed. For example, to save a spreadsheet file, you pressed the / key to enter the menu, then **F** for **File**, then **S** for **Store**; the As-Easy-As manual uses **/File Store** as shorthand notation for the same action.

Let's look at an example analysis for a lab I might have done as an undergraduate physics student. A classic "physics 101" experiment is the ball drop, where we calculated the acceleration due to gravity by dropping a steel ball from different heights and measuring the time it takes for the ball to reach "zero." In our labs, we used two photo "gates" to measure the time: when the ball passed through the first gate, it started a timer. Passing through the

second gate stopped the timer, with a resolution to 0.001 seconds.

We can use the equation of motion to determine how to calculate the acceleration due to gravity:

$$y(t) = y_0 + v_0 t + \frac{1}{2} a t^2$$

The y_0 and v_0 terms are both “zero” in this experiment, leaving a linear relationship between distance and time-squared:

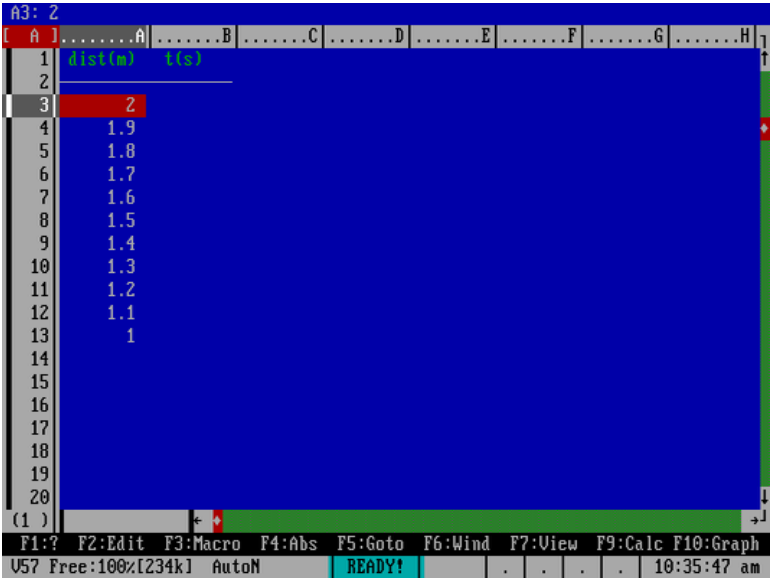
$$y(t) = \frac{1}{2} a t^2$$

Simulating the data

I don’t have my notebooks from my old physics labs, but we can generate suitable data by factoring a random “error” value for the time measurement. This simulates a small delay in the photo gates, or undergraduate students not paying too close attention to setting up the photo gates at each new height.

Let’s start a new spreadsheet in As-Easy-As. After typing in the titles for two columns and formatting a line with **/Range Format bOx**, we can use the **/Data**

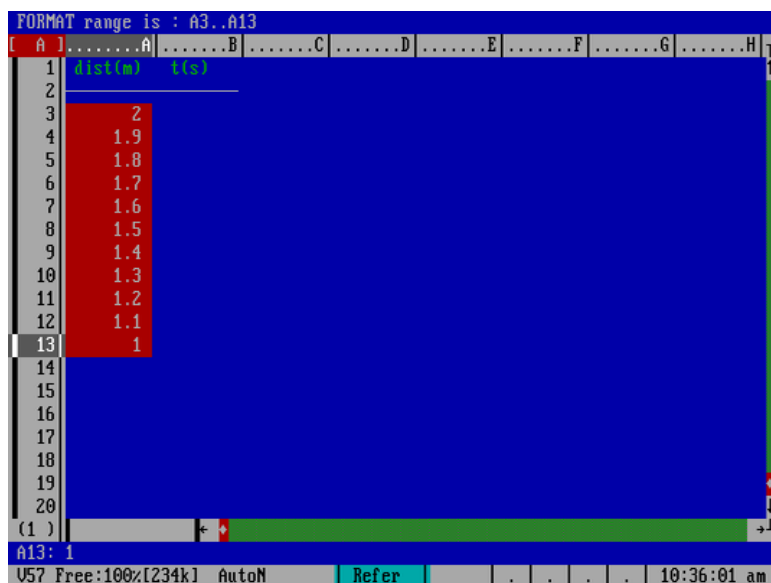
Fill action to enter data in a range. Select the range from A3 to A13, and specify “2” for the starting value and “-0.1” for the step value. This fills the specified range from values counting down from 2 to 1 at regular intervals.



Entering sample data in a spreadsheet

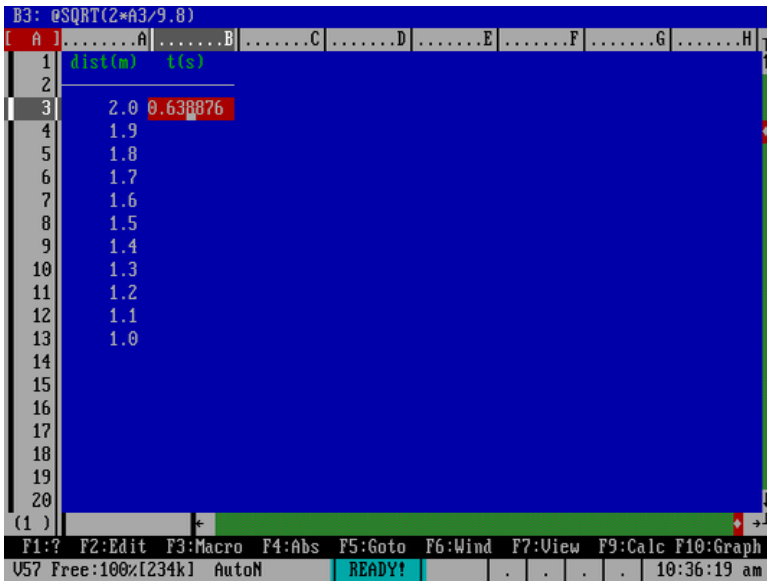
As undergraduate physics students, we might have performed the ball drop experiment from heights ranging from one to two meters, at every 0.1 meter increment distance. We can make this range look like actual measurements by formatting the values to one

decimal place with **/Range Format Fixed** and entering “1” for the number of digits.



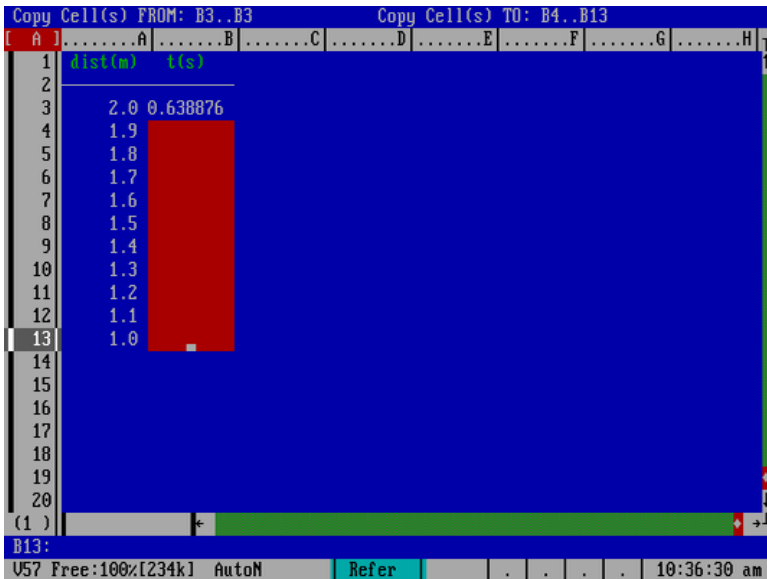
Formatting to one decimal place

We can use the equation of motion to determine the time a ball would take to drop from each height. Solving for t gives the spreadsheet calculation $@SQRT(2*A3/9.8)$. This uses the expected value of the acceleration due to gravity of about 9.8, and the first value in the “dist(m)” column.



Calculating the time

After we have this calculation in cell B3, we can copy the calculation into the other cells in that column using **/Copycell**, selecting the source cell, and specifying the output cells.



Copying the calculation

To simulate actual lab conditions, we should add some “error” values to each measurement. The **@RAND** function generates a random value between zero and one, although it never really gets to 1.0. Divide the random number by 100, and use the **@ROUND** function to truncate the final value to three decimal places. This effectively adds some random “delay” to the time measurement in the last decimal place.

C3: @ROUND(B3+@RAND/100,3)

	A	B	C	D	E	F	G	H
1		dist(m)	t(s)	t+err				
2								
3		2.0	0.638876	0.64				
4		1.9	0.622699					
5		1.8	0.606091					
6		1.7	0.589015					
7		1.6	0.571428					
8		1.5	0.553283					
9		1.4	0.534522					
10		1.3	0.515078					
11		1.2	0.494871					
12		1.1	0.473803					
13		1.0	0.451753					
14								
15								
16								
17								
18								
19								
20								

(1)

F1: ? F2: Edit F3: Macro F4: Abs F5: Goto F6: Wind F7: View F9: Calc F10: Graph
 V57 Free: 99%(233k) AutoN READY! 10:36:53 am

Adding delay to the time measurement

These values depend on a random number, which will change if the spreadsheet performs a global recalculation, such as with the **F9** key. To get static values, we can use **/Range Copy Value** to copy the column of numbers as values, not calculations. The result gives us a column of numbers that will never change, even during a recalculation.

D3: 0.64

	A	B	C	D	E	F	G	H
1	dist(m)	t(s)	t+err	time(s)				
2								
3	2.0	0.638876	0.64	0.64				
4	1.9	0.622699	0.626	0.626				
5	1.8	0.606091	0.614	0.614				
6	1.7	0.589015	0.589	0.589				
7	1.6	0.571428	0.573	0.573				
8	1.5	0.553283	0.559	0.559				
9	1.4	0.534522	0.541	0.541				
10	1.3	0.515078	0.516	0.516				
11	1.2	0.494871	0.503	0.503				
12	1.1	0.473803	0.479	0.479				
13	1.0	0.451753	0.455	0.455				
14								
15								
16								
17								
18								
19								
20								

(1)

F1: ? F2: Edit F3: Macro F4: Abs F5: Goto F6: Wind F7: View F9: Calc F10: Graph
 V57 Free: 99%[232k] AutoN READY! 10:37:29 am

Copying the values from the calculations

The acceleration due to gravity

The equation of motion predicts a linear relationship between distance and time-squared, with $a/2$ for the slope and the intercept at zero. Let's start the analysis by calculating time-squared in the next column. This is a simple calculation to multiply the "time" value by itself.

E3: +D3*D3

	A	B	C	D	E	F	G	H
1	dist(m)	t(s)	t+err	time(s)	t^2			
2								
3	2.0	0.638876	0.64	0.64	0.4096			
4	1.9	0.622699	0.626	0.626				
5	1.8	0.606091	0.614	0.614				
6	1.7	0.589015	0.589	0.589				
7	1.6	0.571428	0.573	0.573				
8	1.5	0.553283	0.559	0.559				
9	1.4	0.534522	0.541	0.541				
10	1.3	0.515078	0.516	0.516				
11	1.2	0.494871	0.503	0.503				
12	1.1	0.473803	0.479	0.479				
13	1.0	0.451753	0.455	0.455				
14								
15								
16								
17								
18								
19								
20								

(1)

F1: ? F2: Edit F3: Macro F4: Abs F5: Goto F6: Wind F7: View F9: Calc F10: Graph
 V57 Free: 99%(232k) AutoN READY! 10:37:47 am

Calculating time-squared

With this calculation, we have all that we need to calculate the acceleration due to gravity, as “measured” in this experiment. Before we take the final steps, we might clean up the spreadsheet by using **/Range Format Fixed** to set the time values to three decimal places, and time-squared values to four places.

FORMAT range is : E3..E13

	A	B	C	D	E	F	G	H
1	dist(m)	t(s)	t+err	time(s)	t^2			
2								
3	2.0	0.638876	0.64	0.640	0.4096			
4	1.9	0.622699	0.626	0.626	0.391876			
5	1.8	0.606091	0.614	0.614	0.376996			
6	1.7	0.589015	0.589	0.589	0.346921			
7	1.6	0.571428	0.573	0.573	0.328329			
8	1.5	0.553283	0.559	0.559	0.312481			
9	1.4	0.534522	0.541	0.541	0.292601			
10	1.3	0.515078	0.516	0.516	0.266256			
11	1.2	0.494871	0.503	0.503	0.253009			
12	1.1	0.473803	0.479	0.479	0.229441			
13	1.0	0.451753	0.455	0.455	0.207025			
14								
15								
16								
17								
18								
19								
20								

(1)

E13: +D13*D13

V57 Free: 99%[232k] AutoN Refer 10:38:22 am

Formatting the time-squared values

The intermediate calculations for “t(s)” and “t+err” aren’t needed here. We could delete these columns with **/Sheet Delete Column**, but let’s instead use **/Sheet Colwidth Hide** to hide them. We actually need to perform this action once per column.

Assign a width of zero to column

	..A	B	C	D	E	F	G	H
SHEET	m)	t(s)	t*err	time(s)	t^2			
COLWIDTH								
	0	0.638876	0.64	0.640	0.4096			
Setwidth	9	0.622699	0.626	0.626	0.3919			
Reset	8	0.606091	0.614	0.614	0.3770			
Global	7	0.589015	0.589	0.589	0.3469			
Hide	6	0.571428	0.573	0.573	0.3283			
Range	5	0.553283	0.559	0.559	0.3125			
	4	0.534522	0.541	0.541	0.2927			
10	3	0.515078	0.516	0.516	0.2663			
11	2	0.494871	0.503	0.503	0.2530			
12	1	0.473803	0.479	0.479	0.2294			
13	0	0.451753	0.455	0.455	0.2070			
14								
15								
16								
17								
18								
19								
20								
(1)								
F1: ? F2: Edit F3: Macro F4: Abs F5: Goto F6: Wind F7: View F9: Calc F10: Graph								
V57 Free: 99%(232k) AutoN -Sheet- 10:38:32 am								

Hiding columns B and C

The final action is to perform a linear regression on the time-squared and distance data. Use **/Data Regress Xdata** to specify the time-squared range for the regression, and **/Data Regress Ydata** for the distance.

INDEPENDANT or X Data Range: E3..E13

	A	D	E	F	G	H	I	J
1	dist(m)	time(s)	t^2					
2								
3	2.0	0.640	0.4096					
4	1.9	0.626	0.3919					
5	1.8	0.614	0.3770					
6	1.7	0.589	0.3469					
7	1.6	0.573	0.3283					
8	1.5	0.559	0.3125					
9	1.4	0.541	0.2927					
10	1.3	0.516	0.2663					
11	1.2	0.503	0.2530					
12	1.1	0.479	0.2294					
13	1.0	0.455	0.2070					
14								
15								
16								
17								
18								
19								
20								
(1)								

E13 [F4]: +D13*D13
 V57 Free: 99%[232k] AutoN Refer . . . 10:39:06 am

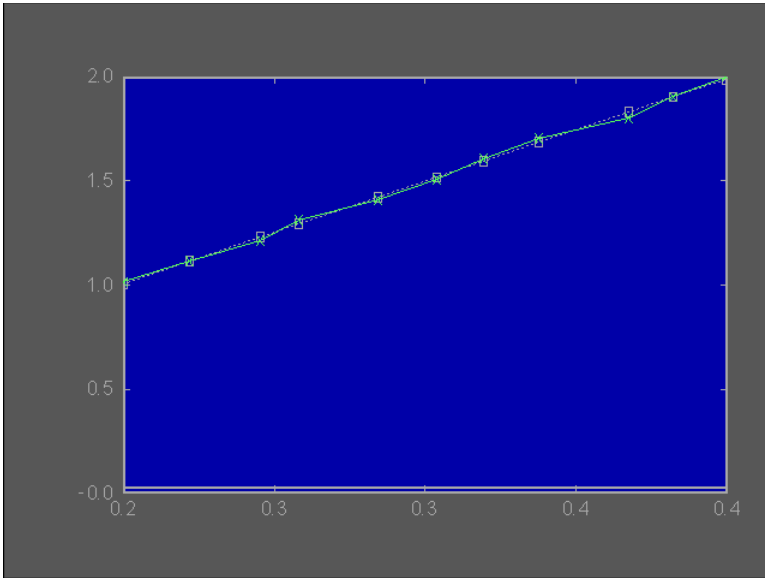
The x data range

To generate the regression and display a chart at the same time, use **/Data Regress View**. As-Easy-As will prompt for where to save the regression output, then will use that data to generate a chart.



Specify the output range

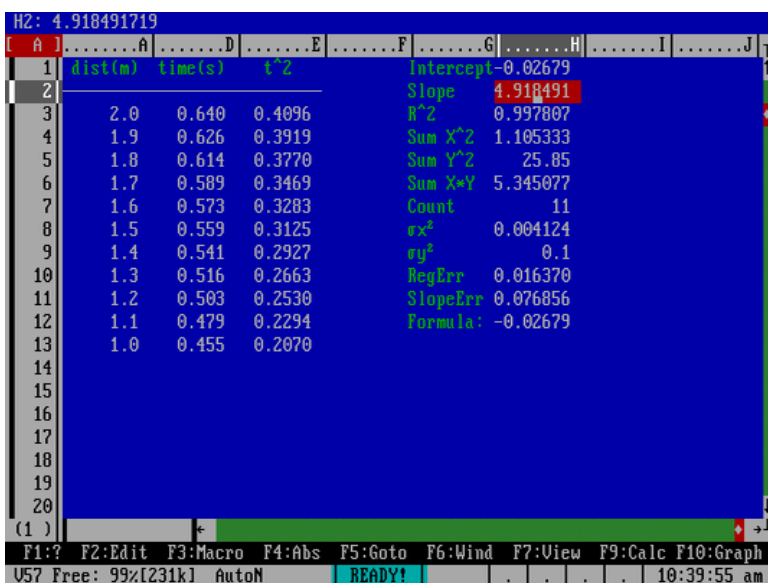
As-Easy-As uses default values to draw the chart:
 data points are “x” marks joined by a green line, the
 line fit is a dotted white line with squares at each
 value.



As-Easy-As also generates a chart

The final analysis

The linear regression shows several important values: the calculated intercept and slope are at the top, and the “error” values for both are towards the bottom. In this calculation, the linear regression calculated -0.02679 for the intercept and 4.918491 for the slope.

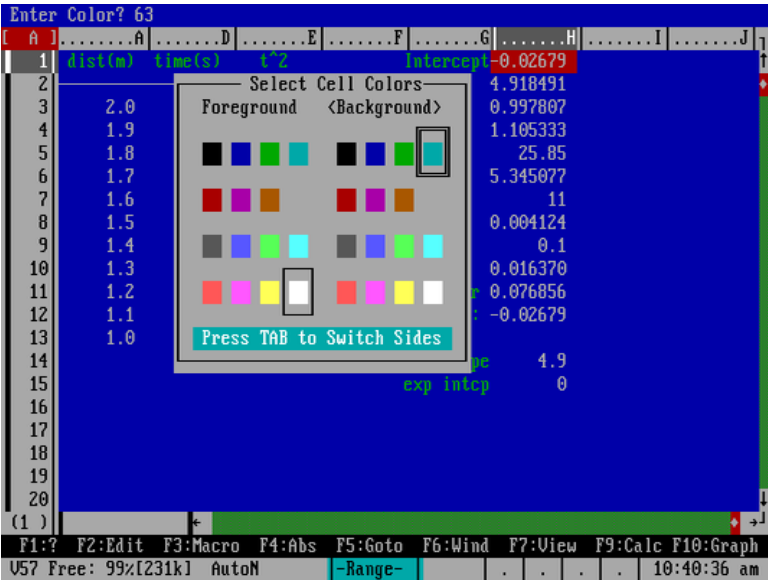


Highlighting the calculated slope value

We can't really "claim" these values in a final analysis. The "error" values determine the actual precision: the intercept value is *plus or minus* 0.016370 and the calculated slope is *plus or minus* 0.076856.

As an undergraduate physics student, I preferred to only display the final values to the number of decimal places indicated by the precision. Let's highlight the important values, and set them to three decimal places; the first significant digit is actually in the second decimal place, but our undergraduate lab analysis typically went "out" one more decimal place,

and we rounded the values to the first significant digit in the “error” when reporting the result.



Bright white on a cyan background

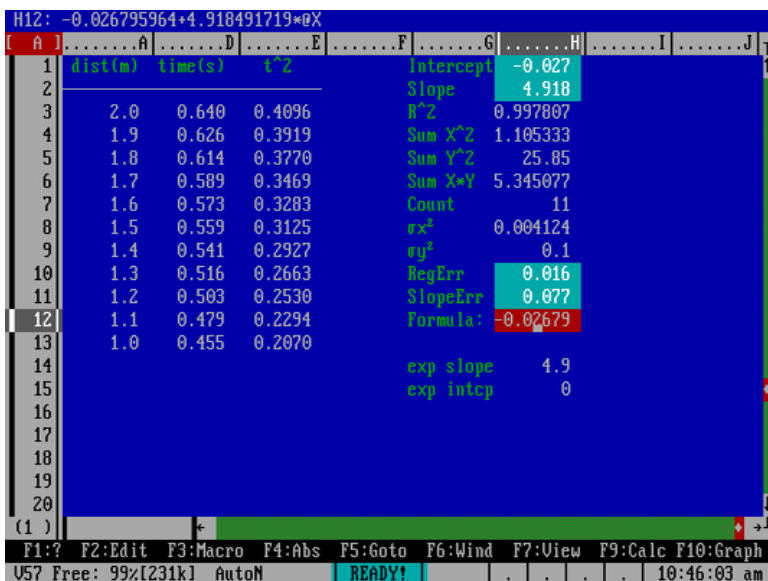
To compare, I’ve also entered the expected slope and intercept. Using the equation of motion, we expect a line with time-squared and distance to have a slope of $9.8/2$, or 4.9 (half the acceleration due to gravity) and a zero intercept (a ball at zero height requires no time to fall).

In this analysis, we calculated the slope at 4.92 *plus or minus* 0.08, and an intercept at -0.02 *plus or minus* 0.02.

H14: 9.8/2										
	A	D	E	F	G	H	I	J		
1	dist(m)	time(s)	t^2		Intercept	-0.027				
2					Slope	4.918				
3	2.0	0.640	0.4096		R^2	0.997807				
4	1.9	0.626	0.3919		Sum Y^2	1.105333				
5	1.8	0.614	0.3770		Sum Y^2	25.85				
6	1.7	0.589	0.3469		Sum X*Y	5.345077				
7	1.6	0.573	0.3283		Count	11				
8	1.5	0.559	0.3125		xx^2	0.004124				
9	1.4	0.541	0.2927		yy^2	0.1				
10	1.3	0.516	0.2663		RegErr	0.016				
11	1.2	0.503	0.2530		SlopeErr	0.077				
12	1.1	0.479	0.2294		Formula:	-0.02679				
13	1.0	0.455	0.2070							
14					exp slope	4.9				
15					exp intcp	0				
16										
17										
18										
19										
20										
(1)										
F1: ? F2: Edit F3: Macro F4: Abs F5: Goto F6: Wind F7: View F9: Calc F10: Graph										
V57 Free: 99%(231k) AutoN READY! . . . 10:41:27 am										

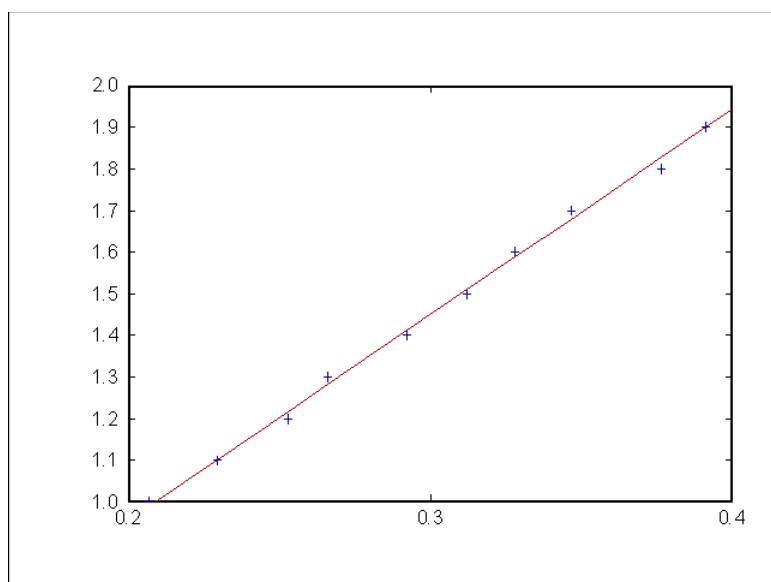
The final result

One reason that charting the results was so easy is that As-Easy-As can generate a chart using a “formula” calculation, using @X as a placeholder for the x value in the line. This formula is the last value in the linear regression output, in cell H12.



The linear regression formula

As a last step, I would usually print a copy of the chart to attach in my lab notebook. As-Easy-As supported many options to generate a printed copy of a chart, including PIC or PCX image files, PSP for printing on a PostScript laser printer, and PLT files for plotting to a pen plotter. For example, here's a PCX copy of my chart, after applying some additional formatting such as colors, ranges, marks, and line styles.



Final linear regression