

So The World May Know

## Words From The Top



Vince Pfeifer

As I sit here writing this editorial, I realize that six months ago I went through a birthing process: I had my first child, a baby boy. With Amiga, I am getting another chance to witness a birth. While I am not going to start

gushing about my new baby boy, I am going to gush about what we are working on at Amiga.

Nine months ago when Bill McEwen approached me about my thoughts on what Amiga wanted to do, I was rather skeptical. I didn't grasp the concept of a single application code base using the same code to run on multiple platforms. I thought of Java and the challenges that it faced as different implementers of Java Virtual Machines made small changes to the system that required modifications to code to run properly. I thought of the power required by machines to run Java Code. I saw a mountain of technical problems that were impeding Bill's vision.

At that time, the Amiga vision was a work in process and wasn't fully fleshed out or presentable. I didn't understand the underlying technology and its abilities. And most of all, I didn't know about the continued Amiga loyalist community that exists in the computing world.

You see I am a heathen. While I owned Commodore and Atari computers as a kid, my first real

powerful machines were IBM PC clones of the mid 80s. For 12 years, I sold, supported, developed, designed, and managed hardware or software for MS-DOS, Windows, or Novell Netware systems. I would laugh at the Apple Macintosh people whose products never made the numbers that their PC counterparts made and usually required changing Heap size or something. In all honesty, Commodore didn't market their products in such a fashion as to convince the companies I worked for to invest in selling to their platform. I was (and am) a product of the market driven computing world. I make products for platforms that have large user bases so that I can have a product line that doesn't require an unrealistic buy-in from the group.

Which brings me back to Bill's mountain of problems when he came to talk to me. If the new Amiga computer were to be recreated in its old form, Amiga Inc. would have a tough time. Creating a new Operating System was tough enough, but to build a new hardware line that runs the OS in today's market seemed an insurmountable task for an unestablished company. However, if the Operating System worked on X86, PPC and other off-the-shelf hardware AND smaller, less powerful computing devices AND they

“ When you can build a system like the one Amiga has envisioned you have an exciting new computing paradigm opening up to you ”

## Hyperion: Ready For The New Amiga

Upstart European games house Hyperion is throwing its resources behind the new Amiga, promising to port a whole range of top-class PC hits.

The firm will play a vital role in creating new Amiga compatibility with OpenGL, the industry standard for displaying fast and smooth 3D graphic sequences.

Hyperion's boss Ben Hermans is a Belgian lawyer and Amiga enthusiast who has proved in the space of one year that it is possible to license top-grade PC titles for the Amiga. For years, Amiga games players waited in vain for major PC ports, but in most cases they never came (Quake and Myst by ClickBOOM were notable exceptions). PC game publishers didn't seem really interested in the waning Amiga market, and too stringent in their licensing conditions.

Herman's experience in negotiating helped his young company to succeed where others failed, porting major PC titles not only to Amiga but also to Linux and Mac. The first product, Heretic II, will be released shortly, and has already earned itself an Amiga Format Gold Medal and AmigaActive's Editor's Choice Award. Other games are Sin, The Heretic Fortress, Freespace: The Great War, Worms Armageddon, and Shogo. "We are going to take as many as possible of the current high-end PC games that we have licenses for and port them to run natively on the new Amiga," says Ben.

"We are not looking at developing brand new games at the moment. That would take two to three years, and the budgets are atrocious. Games development in that respect is moving in the same direction as movie production! We will continue to license high-end products



on the PC platform and get them over as fast as possible to the new Amiga machines."

Amiga Inc. just recently sent two Hyperion developers to a three day training course with Tao, one of Amiga's strategic partners providing the foundation software for the Amiga environment. Upon their return, these programmers started work on a native implementation of OpenGL. Initially, Hyperion's work will be based on MESA, an OpenGL-compatible, open-source 3D system developed on Unix/Linux machines.

Obtaining the OpenGL license has traditionally been a difficult task strewn with pitfalls, partly because of a host of tests, high licensing fees, and a certain wariness on the part of the owners, SGI. However, the situation is improving, according to Hermans.

"Silicon Graphics has become a lot more lenient since the Linux days. They are apparently working with NVidia to get an official SGI OpenGL on the Linux platform, and becoming more favorable towards open source and open standards in general."

If Amiga ever does decide to pursue an OpenGL license from SGI, Hyperion will be ready. "If you're doing MESA you're 99% there," says Hermans.

All ports by Hyperion are first developed on the Amiga by a team

of three people, reworking code so that it is no longer dependent on Windows and Intel x86 functions—replacing, for example, Microsoft Direct3D with OpenGL.

"It's a lot of work, but nothing can be done about that," says Ben. "We don't have Direct3D, and we don't



want it!" Technically, the process of first preparing an Amiga port, and then versions for Mac and Linux, has proved viable. Once the Amiga version is finalized only another month of work is necessary to produce the Mac version.

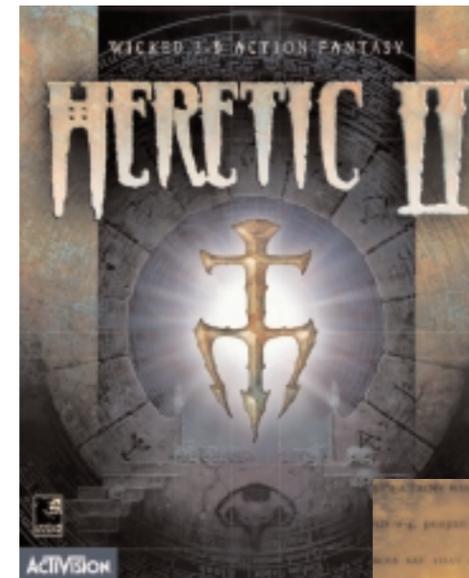
So what does Hermans think of the possibilities that will be opened up by the new Amiga?

"Almost endless. The problem is that right now we're being held back by outdated hardware.

Even the current [Amiga] PPC boards are three years old, with bottlenecks all over the place. We could start out with a PPC Boxer or PoP motherboards."

PoP is a public domain motherboard design based on the former Common Hardware Reference Platform (CHRP) of IBM, Apple and Motorola. "Looking at the Motorola roadmap it's clear that the G4- G5 generations are going to be light-years ahead of anything that even AMD will be able to offer," says Hermans.

"Combine that with industry-leading graphics capabilities, such as from NVidia, and you have an extremely capable platform. The whole issue of the new Amiga's virtual processor makes almost irrelevant what kind of hardware you are using," says Hermans. "The good thing is that if I compile something for the new Amiga running on PoP motherboards, it will run on x86 or whatever platform Amiga chooses to support. All the code is being compiled to a virtual processor, which doesn't exist, and the result is a completely platform-independent technology. This is very nice, because if something goes horribly wrong with, say, IBM, and they decide to cancel PPC development, you're not stuck to a specific object-code or whatever. You can say well, OK, let's use something else! That along with the inherently high Java execution speed are major advantages."



Thanks in part to contract work with Monolith Inc., Hyperion has built up considerable expertise in one of today's most advanced (if not *the* most advanced) 3D engine, LithTech. This is currently going into overdrive, being licensed to Fox Interactive, 3DO, Interplay, and others.

"We are porting that engine to current and new Amiga systems, and this will allow anything based on that engine to be ported by us in a very short time, maybe two weeks," affirms Hermans. "We've also ported Heretic II, which means that we have a very good understanding of the Quake II engine, and this means that we can license games using this engine and get them ported in a matter of weeks instead of months."

"Our good relations with games publishers would enable us to get a license before a game hits the market and a week or two after the PC launch we can have it on the new platform as well," says Hermans. "That's why we're focusing on specific engines,

because if you have that know-how, you can move really quickly. We shall be demonstrating that in the second quarter of this year, when we'll get a few Monolith titles and make sure that they are released [on the Amiga, Mac and Linux] nearly simultaneously with the PC version."

Hermans is talking about ports to the classic Amiga here, but he points out that there is no reason the same procedure cannot be used for the new Amigas as soon as they become available.

Hyperion's proactive approach to preparing for the new Amiga is what Amiga is looking for in a partner. This type of commitment to the future will pay off for companies that recognize the possibilities of the new Amiga. Hyperion surely envisions a bright future with the new Amiga.

## Making User Groups Work

When Amiga was acquired from Gateway last year, Bill McEwen cited the main reason for doing so was the Amiga community. However, that community would be far less valuable if it weren't for the efforts of user groups throughout the world. These worldwide user groups accomplished great things over the last few years, and are a big reason the Amiga continues to this day.

While Amiga user group numbers have fallen, it doesn't mean that the user group experience has faltered. Some of the ingenious things Amiga user groups have done have enhanced the Amiga experience into a tight knit community. User groups filled in the gaps left behind by the mainstream PC community. For example: Gateway Amiga sponsors an annual Amiga show in St. Louis. Amiga Atlanta is noted for their charity work with the Special Olympics. SEAL of the UK puts together a very nice magazine, as do the French groups AFLE and TRIPLE A. NCAUG has sponsored the blimp-cam and IRC show coverage. And the international virtual group Team AMIGA is a veritable customer support group. No other platform has user groups so involved and so responsible for maintaining a platform's viability.

So what does it take to make your user group an enjoyable experience? Not much actually. It just takes some smart planning and organization to get things going. You'll find that there are many folks eager to see the latest software and hardware, get their questions answered, or shoot the breeze with like-minded people. The good news is that you don't have to go it alone either. There are many companies, resources, and other user groups just waiting to lend a helping hand, but first let's cover the formalities.

Amiga user groups vary in size and organization. Some of the smaller groups have 10 or fewer members, while some of the larger groups (particularly those in Europe) have hundreds of members. Some user groups have no formal charters in place while others have achieved official not-for-profit status with their local governments.

After your group has decided whether to incorporate or not, and has elected or appointed some key officers (President, Treasurer, etc.) you will need to find a place to meet and to get the word out. If your group is

small, then any member's home will suffice for a meeting place. This can be a nice arrangement. Not only will you have access to an Amiga for meeting topics, you won't have to use the group's funds to rent a room. There should be various community centers, schools, libraries, offices, and universities that have rooms to offer for free or for a



small fee.

Getting the word out about your new user group shouldn't be hard in this day of the Internet. Setting up a web page and registering it with some of the many Internet search engines is a big plus. Posting announcements to Amiga forums and newsgroups is another good way to get the word out. Many localities have free computer publications like 'Computer User' that list groups in their area for free. Lastly, many cable companies and radio stations offer free air time that you can take advantage of to promote your non-commercial group.

With your user group all set up and ready to go, your next task is to consider the meeting topics and activities your group will undertake. Not every meeting has to be an elaborate multimedia event, but some good planning can make for a better experience for all.

While hardware and software offerings for the classic Amiga are a bit thin right now, there are still many releases (commercial and shareware) that users would be interested in seeing demonstrated at a meeting. Naturally, when the new Amiga is released, there will be a great deal

of attention placed on new products. Another important area for the Amiga is networking. Showing how you can network your Amiga to PCs and Macs and other cross platform topics can help folks find value in using their Amigas. Presenting tutorials of application software, like Pagestream, ImageFX, Wordworth, Imagine, Tornado, etc., or showing hands-on how to install a graphic card or other hardware will certainly keep members interested. Of course, playing the latest game release is always a good choice for a user group meeting.

Occasional special meetings can also help keep up morale and attendance. Annual swap meets and cookouts break up the normal routine. Swap meets can help you drum up some cash as well and lessen the clutter in your closet, and cookouts are always a crowd pleaser. For example, once a year members of the National Capital Amiga Users group drive up to Software Hut in Philadelphia and have a day of food, fun, and special discounts. Many Amiga dealers are more than happy to participate in these types of user group events.

Anyone who knows Lamar Morgan and the Amiga Atlanta user group understands the great work they have done in supporting the Special Olympics. When IBM dropped their support of the Special Olympics, Mr. Morgan saw an opportunity to increase the Amiga's exposure for a good cause. Because of Mr. Morgan's efforts, Amiga is now the exclusive platform of the Atlanta Special Olympics. This is something most user groups could accomplish. Mr. Morgan invites all Amiga user groups to see how they can participate with their local organizer. If interested, please call Norm Sterling of the Special Olympics at 770-414-9390 ext. 102 and tell him Lamar Morgan sent you.

But it doesn't end there. Through Amiga Atlanta's hard work and involvement with the Special

Olympics, Sprint will now offer all Amiga groups in the US fee-free usage of their special long distance conferencing accounts. Lamar hopes that one day a group of Amiga user groups link up in a large meeting using this special account. Those interested should contact Lamar Morgan at 770-209-4011 or go to the Amiga Atlanta website at [www.amigaatlanta.org](http://www.amigaatlanta.org).

With the new Amigas to arrive soon, now is the time to start planning a new user group or to gear-up your current user group for the next generation. Until then, there are many things you can do to keep the fire going at your user meetings. Whether your group joins in a picnic chatting about Amigas, reviews the latest software, or participates in a local charity event, your user group experience can still be enjoyable.

No matter what size user group you're working with, the first place you should visit to get information on how to operate a user group is the User Group Network, or UGN (<http://ugn.amiga.org>). The UGN provides vital information for all Amiga user groups. A search from the Amiga.org homepage (<http://amiga.org>) for user groups you will yield a complete listing of Amiga user groups worldwide. In addition to the great info on user groups, you will find a wealth of Amiga knowledge and support from the Amiga.org website.

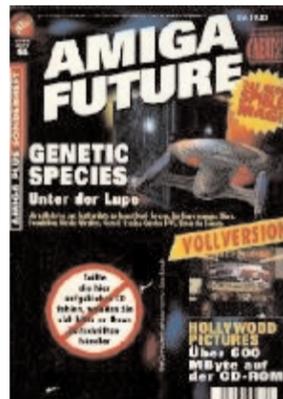
If you would like Amiga to advertise in your group's newsletter, or to sponsor a local sports team, contact Kari at [kari@amiga.com](mailto:kari@amiga.com). Amiga would also like to hear suggestions or comments about user group activities and events at the same address. ♦



## Amiga Resource Center

Amiga World wants to let readers know about the wealth of information available in both print and electronic format. If you know of an Amiga resource that should be printed here, be it an Amiga-related magazine, website, mailing list, or other resource, please let us know by emailing [amigaworld@amiga.com](mailto:amigaworld@amiga.com). This month we present to you even more print magazines.

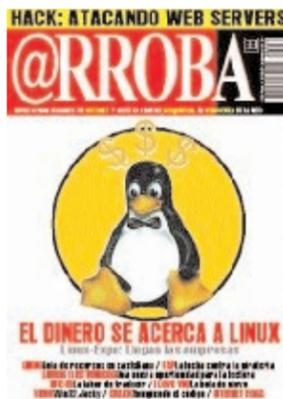
**Magazine name:** Amiga Future  
 Editor: Andreas Magerl  
 Country published: Germany  
 How often published: Bimonthly (6 issues per year)  
 Format: Color, Amiga only  
 Cover disk: CD included  
 Language(s): German  
 Website: [www.amigafuture.de](http://www.amigafuture.de)  
 Email contact: [redaktion@amigafuture.de](mailto:redaktion@amigafuture.de) or [Andreas@chiemgau.org](mailto:Andreas@chiemgau.org)  
 Time published: Two years  
 Subscription price: DM 19.80  
 Distribution: Worldwide  
 Comments: We can be reached at +49-8642-89995 or Faxed +49-8642-895004



**Magazine name:** Anews  
 Editor: Thierry Sillis and Johann Girard-Cheron  
 Country published: France  
 How often published: Monthly  
 Format: Color & B&W, Amiga 80%, Atari and BeOS 20%  
 Cover disk: CD  
 Language(s): French  
 Website: [www.amiganews.com](http://www.amiganews.com)  
 Email contact: [abonnements@amiganews.com](mailto:abonnements@amiganews.com)  
 Time published: 13 years  
 Subscription price: 390 Fr  
 Distribution: Worldwide



**Magazine name:** Arroba  
 Editor: Andres Carlos B. Gomez Miranda  
 Country published: Spain, with sales in Latin America  
 How often published: Monthly  
 Format: A4, full color, 80 pages, much Amiga.  
 Cover disk? CD included  
 Language(s): Spanish

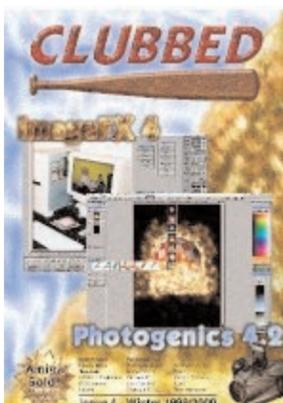


Website: [www.megamultimedia.com/arroba/](http://www.megamultimedia.com/arroba/)  
 Email contact: [arroba@activanet.es](mailto:arroba@activanet.es)  
 Time published: Since October 1997  
 Subscription price: 995 PTS per issue  
 Distribution: Spanish speaking countries worldwide  
 Comments: It's not an easy task to define our magazine. Arroba is an underground computer magazine with few commercial points of views. We publish lots of information which gets little coverage in other magazines...security, alternative platforms, etc.

**Magazine name:** Amiga.it  
 Editor: Giorgio Signori  
 Country published: Italy  
 How often published: Bimonthly (6 per year)  
 Format: CD-ROM Magazine, Amiga, LinuxPPC/Linux68k  
 Cover disk: The magazine is a CD  
 Language(s): Italian  
 Website: [www.skylink.it/ear](http://www.skylink.it/ear)  
 Email contact: [Yurex@tin.it](mailto:Yurex@tin.it)  
 Time published: Since Nov. 1999  
 Subscription price: 165.000 ITL  
 Distribution: Italy  
 Comments: The first Italian Amiga CD-based magazine that we know about.



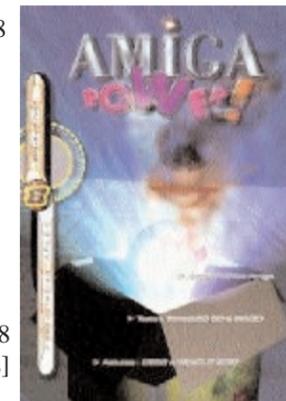
**Magazine name:** Clubbed  
 Editor: Robert Williams  
 Country published: United Kingdom  
 How often published: Quarterly  
 Format: A4 (8"x11" approx), Color covers, mono interior, 40 pages, Amiga only.  
 Cover disk: No  
 Language(s): English  
 Website: [www.seal-amiga.co.uk](http://www.seal-amiga.co.uk)  
 Email contact: [clubbed@seal-amiga.co.uk](mailto:clubbed@seal-amiga.co.uk)  
 Time published: Since October 1998  
 Subscription price: 4 issues for UK £10, Europe £12, USA \$14, RoW £15  
 Distribution: Worldwide  
 Comments: We cover as many aspects of the Amiga as we can, including, news, features, reviews, and tutorials. We welcome contributions from our readers.



**Magazine name:** Amiga plus  
 Editor: Thomas Raukamp  
 Country published: Germany, Switzerland, Austria, and Luxemburg  
 How often published: Monthly  
 Format: Color, Amiga only  
 Cover disk: For subscribers  
 Language(s): German  
 Website: [www.amigaos.de](http://www.amigaos.de)  
 Email Contact: [falkeabo@aol.com](mailto:falkeabo@aol.com)  
 Time published: 8 years  
 Subscription price: Germany, DM 99; foreign countries, DM 120; Worldwide, DM 159  
 Distribution: Worldwide  
 Comments: Created on an Amiga 060/PPC, using PageStream software. Phone (+49) 4331 84 93 37.

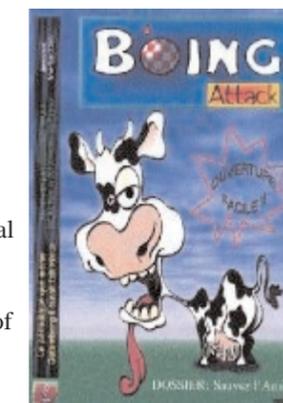


**Magazine Name:** aMiGa=PoWeR  
 Editor: Association AFLE [First French Amiga Association, more than 230 members]  
 Country published: France  
 How often published: Bimonthly  
 Format: A4, Laser Color & B/W, 28 pages, 100% Amiga  
 Cover disk: No  
 Language: French  
 Websites: [www.netlinker.com/amigapower](http://www.netlinker.com/amigapower) [www.multimania.com/afle](http://www.multimania.com/afle)  
 Email contact: [amigapower@pacwan.fr](mailto:amigapower@pacwan.fr) or [chipset@wanadoo.fr](mailto:chipset@wanadoo.fr)  
 Time published: Since October 1998  
 Subscription price: 150Frs [6 issues] France - 220Frs [6issues] Foreign country  
 Distribution: Worldwide  
 Comments: News, Games, Internet, Music, Hard & Software tests, 3D, Assistance, Blitz, etc. We are supported by French Amiga retailer [[www.aps.fr](http://www.aps.fr)-[www.sl-diffusion.com](http://www.sl-diffusion.com) - [www.sparadise.com](http://www.sparadise.com)]

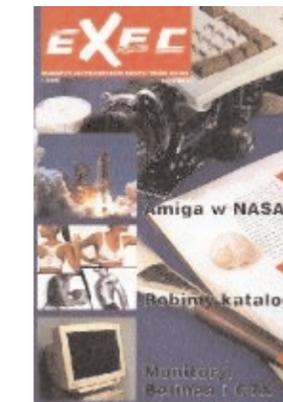


**Magazine name :** BOING ATTACK  
 Editor: TRIPLE A (French Amiga User Group) - We have the only French Amiga Merchandising License.  
 Country published: France  
 How often published: Bimonthly  
 Format: B&W with color cover, 24 pages, Amiga only  
 Cover disk: No  
 Language(s): French

Website: [www.triplea26.com](http://www.triplea26.com)  
 Email contact: [triplea@triplea26.com](mailto:triplea@triplea26.com)  
 Subscription price: 20 FF (\$USD=6,5FF) per issue.  
 Subscriptions available for 1 or 6 issues  
 Distribution: Worldwide. Additional postage outside of France.  
 Comments: Boing Attack is a magazine about Amiga with tests of games, public domain, news, development (HTML), hardware and software tests. The subscription is the inscription to TRIPLE A Association and we have a lot of discounts with French Amiga retailers and Amiga developers.



**Magazine name:** eXec  
 Country published: Poland  
 Editor: Grzegorz Juraszek  
 How often published: Monthly  
 Format: B&W, 48 pages, color cover, Amiga with a Linux section  
 Cover disk: CD  
 Language(s): Polish  
 Website: [www.amiga.pl](http://www.amiga.pl)  
 E-mail contact for subscriptions: [sprzedaz@amiga.pl](mailto:sprzedaz@amiga.pl)  
 Time published: We've just started  
 Subscription price: 22 ZL (1 USD ~4 ZL)  
 Distribution: Poland only, for now  
 Comments: We have just started publishing our magazine. The second issue is in the works



**Magazine name:** Mir Amiga  
 Country published: Russia  
 Editor: Boris Kondratyev  
 How often published: Bimonthly (6 per year)  
 Format: A4, 64 pages, color cover, B&W inside, Amiga only  
 Cover disk: A special CD-Rom about once a year  
 Language(s): Russian  
 Website: <http://www.forcefield.net/woarus>  
 Email contact: [amiline@chat.ru](mailto:amiline@chat.ru)  
 Time published: Since 1999  
 Subscription price: \$20 for 6 issues  
 Distribution: Russia and all former USSR countries



## The Amiga Foundation Layer

With the impending first release of the Software Development Kit (SDK), we will be launching the first new Amiga product in over 5 years. The SDK will provide all the software elements required to develop for the new Amiga. We want to share with you some of the features and concepts our pathfinding developers will encounter, some insight into why Amiga has made some of its choices, and some idea of where we intend to go over the short, medium and long terms.

The Amiga has always been about enjoyment, and its secret weapon has always been the smile. Too often these days people and technology collide in a multiple pile-up of confusion, frustration and frowns, whether it be someone wanting to record a program, add a new purchase to their house or developing a new audio codec.

The eastern philosophy of Taoism teaches the principle of Wu Wei, which is (poorly) translated as effortless effort. This has always been and will continue to be the guiding principle of the Amiga philosophy--the creation of products that empower the user, customer and developer while being highly tuned to the tasks that they want to perform. They are thus able to do what they want to do without having to concern themselves with the underlying system. Because they are able to succeed in what they want to do, they find themselves smiling.

The flow of Tao into the Amiga is now more than just a philosophical reference. We looked at many potential partners and technologies before finally settling on the provider of the Amiga Foundation Layer. At issue was ensuring that we found a product that matched our very different technical and cultural view of the digital future. Since we see part of the problem as originating in traditional architectures, it was very important to us that we picked a partner who not only saw the future as we see it but who "walked the walk" as well as they "talked the talk;" in other words, a partner who not only got the future but who had built a product designed for that future.

The Tao Group's story starts with Chris Hinsley, a games developer throughout the 1980s, the golden age of

the home computer. He would develop a game for the Amiga, then be asked to port it to the Atari ST or to the PC. After one or two ports it occurred to Chris that there had to be a better way, so he began experimenting. First he came up with a heavily macroed assembler; this soon evolved into something more--A virtual processor that could be simply and quickly implemented on top of multiple CPU families to provide the holy grail of

software development: Binary portability without sacrificing high performance or memory efficiency, both prerequisites given the specification of hardware in his market space.

With such a cutting edge concept Chris was able to attract a lot of talented developers, and from the virtual processor rolled out other powerful features, including effective dynamic binding and transparent multiprocessing capabilities. This was rolled into TAOS, introduced to most of us for the first time in the excellent Byte article by Dick Pountain

[www.byte.com/art/9407/sec6/art1.htm](http://www.byte.com/art/9407/sec6/art1.htm).

That was five years ago. Since then the Tao Group has been improving, adding to and spreading its product. Its second-generation is called intent, and combines a wealth of functionality that includes a series of multimedia libraries and the intent Java Technology Edition. The latter is fully Sun certified and the first implementation of the PersonalJava specification. It is this product we have selected to be the Amiga Foundation Layer.

The core of the Amiga Foundation Layer (AFL from now on) and of the whole new Amiga architecture is the Virtual Processor. As befits something that is virtual, it doesn't actually exist.

The virtual processor is a 32 bit little endian RISC processor which provides 5 banks of registers: integer (i0-i(n) - 32 bit), long (l0-l(n) - 64 bit), float (f0-f(n) - 32 bit), double (d0-d(n) - 64 bit) and pointer (p0-p(n) - 32 bit). The developer accesses these registers using the macro assembler language VP, which assembles into VP Code, the binary portable object code.

“The core of the Amiga Foundation Layer and of the whole new Amiga architecture is the Virtual Processor. As befits something that is virtual, it doesn't actually exist”

VP Code has been designed to allow the developer to take the small instruction set and grow it into powerful expression trees, giving the benefit of compact binaries while also allowing for the application of powerful optimization techniques. VP provides not only register and memory access through the basic instruction set but also provides a rich and powerful set of macros covering assignment, condition, iteration and some higher level functionality that will be instantly familiar to C programmers.

Most attempts at the creation of portable binaries have ended up sacrificing portability for speed through the use of interpreters. VP Code is NOT interpreted. It follows the same pipeline from source to machine code as do compiled architectures, with a single modification: It stretches the pipeline and inserts VP Code between the compiler back end and the creation of native (i.e. CPU specific) machine code. VP Code thus becomes the distributable rather than the native machine code.

The final piece of the compilation pipeline takes place not on the developer's machine but on the user's machine. When the user process requires that piece of code be executed, the VP Code is loaded and compiled dynamically into the machine code of the host processor. This native machine code, which is then executed, provides the same speed as classic compiled code... and sometimes more.

How is that possible? VP Code has been designed to be as small as possible, and often the VP version is much smaller than the native code version. While there is the obvious overhead of translating VP Code to machine code, this is much less than the overhead of reading the larger machine code version from storage (perhaps a hard drive). Adding to this the fact that the translators themselves are often less than 100K in size (and therefore stay in the cache on many processors), and when the total cost equation is calculated, the VP Code solution is often faster.

Because it is translated dynamically on the target environment, VP Code can make intelligent decisions in

real time--something far beyond the scope of traditional architectures which must prepare a general distributable and add code for all possible variants. For example, the translator may take the VP Code and generate different machine code versions on an i386 system as opposed to a Pentium system because the translators for each are smart enough to provide optimizations.

Thus the path from development to execution is:

1. High Level Language is compiled into...
2. Low Level Assembly Language (VP), which is assembled into...
3. Binary Portable VP Code, which is distributed and installed on a system that calls for the execution of that piece of VP Code, which is then loaded from storage into...
4. The Translator, which then optimizes and translates the VP Code into native machine code for the host CPU, which is then...
5. Executed as normal on the host CPU.

Note: Java is compiled into Java byte code, which is translated at load time into 3.

The Amiga advantage is thus both binary portability and speed. What was always seen as a compromise can now be a dual benefit.

The more perceptive of you may have seen a potential flaw in this architecture. If you have to translate a whole library or class every time it is required, then the overhead will be tremendous. The answer to this question leads nicely into another key feature of the Amiga Foundation Layer, one with which VP programmers will become very familiar.

The AFL utilizes a tool-based architecture. A tool is a small piece of code that performs a single operation. To understand this better, consider classic C or C++ development, which leads to the creation of libraries, each containing a set of

functions, or to the creation of classes, each containing a set of methods. When VP is used, libraries and classes are replaced by a set of tools, one for each function or method. Thus, the AFL provides a very finely grained environment with tools often fitting into less than a kilobyte of memory

“Another powerful feature of VP Code is that, because it is translated dynamically on the target environment, it can make intelligent decisions in real time - something far beyond the scope of traditional architectures”

and enabling toolkits (sets of tools targeted at a specific functional area) to offer a lot of power in a small size. For instance, the standard kernel is 62 KB, the AVE is 152 KB, the complete ANSI/POSIX library is 80KB and the Pentium translator is 88KB.

When a tool is called, the AFL first checks to see if it is already available in memory as a previously translated tool. If it is, then execution flows directly into it, otherwise the tool is pulled from storage, translated and then placed as a native executable into memory. The advantage of this dynamic binding is clear. Not only can targeting be done at run time using state information, but the AFL only has in memory the tools it needs. Compared to this granularity, the loading of an entire library or class just to use a few functions or methods seems positively archaic.

Those who asked the first question may now ask a second question: "Ahh, if this virtual processor is so portable, what about processor specific features such as Altivec, SSE or 3DNow? The virtual processor doesn't know about them." Well, not only do the translators have the intelligence built into them to be able to take advantage of these extra instructions, but a developer can code both a VP Code tool for general release and then a native tool, both with the same name and providing the same function. If a particular tool is called, then the system will first look for a native tool specific to that system (which of course requires no translation and is optimized to that system) and only if one isn't available will it utilize the default VP Code tool. The AFL does this by using the naming convention, toolname.nn where nn represents a specific processor target and 00 represents the virtual processor.

Tools have a name, an entry point, one or more exit points and their own set of registers, and are fully re-entrant. There are two type of tools: Main tools and secondary tools. Main tools are directly startable, and secondary tools are callable only from a main or another secondary tool.

One of the strengths of the Amiga Foundation Layer is that it fully supports both static and dynamic binding. Static binding is provided through the SYSGEN

utility and is mainly used for compiling to embedded systems, where all references are known before time. For the Amiga Foundation Layer, we will concentrate on using dynamic binding. This is where tools are loaded as they are referenced and it can happen either when a process starts or when another tool makes a call. In general, the binder performs this function.

Tools call each other via qcalls or quick calls, of which there are 3 types.

1. Normal qcall - the referenced tool is found at load time and the calling tool is fixed up so that execution jumps directly to the called tool's entry point. This is the fastest possible call. The tool remains in memory until the calling tool is removed.
2. Virtual qcall - at bind time the calling tool is fixed up to call a kernel function with the name of the tool required. At run time the kernel function finds the referenced tool, loads it and then calls it. The referenced tool is de-referenced when the call returns. This is slower than a straight qcall but is very efficient on memory.
3. Virtual + Fixup qcall - at bind time the calling tool is fixed up to call a kernel function with the name of the tool required. At run time the kernel function finds the referenced tool, loads it and then calls it. The tool then remains in memory for further calls. Thus, the first call is as slow as a virtual qcall on its first call but as fast as a normal qcall on subsequent calls.

Tools are the building blocks of the Amiga Foundation Layer and of applications written for the new Amiga. At run time they are accessed via processes and threads. A process is a set of threads and a thread is a route through the code in a set of tools. This is different from the common UNIX model and care should be taken to understand the difference.

In the AFL, processes are very lightweight and memory efficient. Threads can share system resources such as the file table, the signal table, etc., but they are effectively just processes with some additional attributes. In addition, process sovereignty is not enforced; thus

processes running on the same processor can access one another's data.

We at Amiga feel that memory protection is an overhead that punishes good code because bad code exists, and this is an attitude that is becoming increasingly prevalent. Indeed, the classic Amiga lasted for 15 years without any memory protection at all and has been praised as one of the most stable systems ever created, a credit to both the system and the developers themselves.

Instead of placing this overhead on the system, we intend to push it up to its rightful place; at the code level itself. Protection should be at the language level, giving a choice of safe and unsafe languages. Those who are comfortable writing in unsafe languages such as VP and C should be able to take advantage of the extra speed and features. Those who aren't should use a "safe" language such as Java, or restrict their development to those niches where stability is a secondary consideration

Amiga foresees markets for both digital consumers who want stable systems with software that works safely, and pathfinders who like to get down and dirty with the best system possible and consider the occasional instability a price worth paying.

We will thus work hard to promote VP as our system level language of choice, but will encourage the use of Java as the safe development language. This does not mean that those who use Java will be restricted to Java. Far from it, in fact. Not only is the Tao Group's J-Engine one of the fastest personal Java implementations in existence but it also offers a very fast and lightweight JNI solution. Thus, all of the functionality written in VP will be made available to the Java developer through Amiga packages.

So, for our digital consumer markets, we will closely monitor who is allowed to develop in VP, while offering Java as the safe development environment for all. For our pathfinder markets, users are free to develop and use any applications they like in any language they like with the caveat "buyer beware!"

The advantages of Java are already well documented and, over the last two years, the disadvantages have grown fewer to the point where it is now a very safe, powerful and simple language to use. With the special Amiga packages that we will create to encapsulate our system level functionality, Java developers no longer have to be the poor relatives.

The advantages of VP should hopefully be apparent. While it is an assembler level language it is far

more than just assembler, and a good C developer will feel at home in a very short time. Its chief advantages over other languages are simple. It is blazingly fast and it is the language that the Amiga Foundation Layer speaks. There is a learning curve, as there is with any language, but our development team and those who have been kind enough to test the AFL for us have gone from pain to pleasure in a very short time. Of course, C is also provided via the GNU compiler but this simply compiles source down into VP Code anyway. Why buy TV dinners when you can cook yourself? We see the C route as being used primarily as a porting and learning path, with developers turbo charging their applications via VP. Since C programs can call VP tools and vice versa, this will become a powerful feature for accessing the AFL functionality as it grows.

We at Amiga are as excited as those thousands of developers out there who have been emailing us constantly for the last 4 months asking us when they can get hold of the new Amiga. Indeed, a recent semi official poll on the Be website asking what system developers want to get their hands on next found the new Amiga the clear winner. We are excited because we are all standing together at the beginning of a new path, a new adventure, just as the original Amiga team was in 1985. There is a lot more to do, a lot to build and a lot of work involved but it is also going to be a lot of fun, and we are going to do it together.

As a teaser, we are pleased to present the famous but slightly modified Hello World program in VP in such a form that you can code it when you get your SDK home

“The advantages of Java are already well documented and, over the last two years, the disadvantages have grown fewer to the point where it is now a very safe, powerful and simple language to use”

and have it installed on your development machine.

The first thing that you will notice is that we are shipping the SDK with the Amiga Foundation Layer running on top of Linux. Why? Ignoring the fatuous answer, "because we can," Linux at the present time provides a far richer driver and applications base for developers. While we are looking forward to Amiga native products appearing soon, we feel it would have been intensely cruel to limit developers to VI and clawhand syndrome.

But why can we? Because the AFL allows for multiple levels of abstraction. Not only can it sit on top of multiple processor families, it can also sit painlessly on top of other operating systems and provide added value to them while taking advantage of features within that host OS. Yes, there is a penalty, and we are looking forward to creating our first pure Amiga system, but as an exercise in demonstrating the flexibility of our foundation layer, it succeeds admirably.

A point I should make is that the AFL can run either on its own on top of hardware (the pure version), as an OS on top of another underlying OS with only the AFL visible (dedicated hosted), or as an application on top of another OS (application hosted), in which case you can use both the host OS and the AFL at the same time. For the SDK we are shipping the AFL as an application.

I have been using PFE as an editor in Linux to type my programs. The AFL uses a mirrored filing system that allows the Linux filing system to see the Amiga file system and so on. To create my Hello World program, I simply wrote the following VP program and saved it into my developer directory within the Amiga file system. See Figure 1.

As you can see we have an include file for the VP macros as the first line. Then we define the tool proper, in a tool/toolend structure. In the tool initiation statement we give the tool location and name, the language it is in (VP), the fact that it is a main tool, the stack size and a global data area size.

The next statement defines the entry point, the registers to be passed in and the registers to be passed out.

Multiple values can pass in and out, providing a lot of flexibility for the system. Remember that each tool gets its own bank of registers specifically for it, as can subroutines within tools (but that is a more advanced feature). In this example, nothing is passed in or out.

Then we have 3 loops and some pretty standard code that should warm the hearts of C developers everywhere. (i0) is the first integer register, and we can have as many as we like. VP also supports register

Fig. 1

```
.include 'taort'

tool
'demo/example/HelloAmiga',vp,f_main,8192,0
;tool written in vp, main tool, stack size,global
data

ent -:-
  for 5,i0
    printf "Hello World, Amiga is back !!!
(%d)\n",i0
  next i0

  cpy.i 5,i0
  while i0 != 0
    printf "Hello World, Amiga is back !!!
(%d)\n",i0
    dec.i i0
  endwhile

  cpy.i 5,i0
  loop
    breakif i0 == 0
    printf "Hello World, Amiga is back !!!
(%d)\n",i0
    sub 1,i0
  endloop

ret

toolend
```

naming, where (i0) could be called (count) for added readability.

Finally, a tool must always have at least one exit point as specified by the ret command.

With this program saved in the Amiga file system as /demo/example/HelloAmiga.asm, I can flip to the Amiga shell and do the following. See Figure 2.

As you can see, most of this is pretty self-explanatory. (-v) on the assembler (asm) gives the verbose

Fig. 2

```
Amiga0.1:/$
Amiga0.1:/$ cd /demo/example$
Amiga0.1:/demo/example$
Amiga0.1:/demo/example$ ls H*
HelloAmiga.asm
Amiga0.1:/demo/example$ asm -v
/demo/example/HelloAmiga
Elate Assembler 2.93
demo/example//demo/example/HelloAmiga.asm
Written 212 bytes to demo/example/HelloAmiga.00
Amiga0.1:/demo/example$
/demo/example/HelloAmiga
Hello World, Amiga is back !!! (5)
Hello World, Amiga is back !!! (4)
Hello World, Amiga is back !!! (3)
Hello World, Amiga is back !!! (2)
Hello World, Amiga is back !!! (1)
Hello World, Amiga is back !!! (5)
Hello World, Amiga is back !!! (4)
Hello World, Amiga is back !!! (3)
Hello World, Amiga is back !!! (2)
Hello World, Amiga is back !!! (1)
Hello World, Amiga is back !!! (5)
Hello World, Amiga is back !!! (4)
Hello World, Amiga is back !!! (3)
Hello World, Amiga is back !!! (2)
Hello World, Amiga is back !!! (1)
Hello World, Amiga is back !!! (5)
Hello World, Amiga is back !!! (4)
Hello World, Amiga is back !!! (3)
Hello World, Amiga is back !!! (2)
Hello World, Amiga is back !!! (1)
Amiga0.1:/demo/example$ ls H*
HelloAmiga.00 HelloAmiga.asm
Amiga0.1:/demo/example$
```

mode, and writes out demo/example/HelloAmiga.00. Remember, the (.nn) represents the processor family, with 00 meaning that my tool is targeted at the virtual processor. It's as easy as that. In one fell swoop, I now have a tool I can run on any processor for which there is a VP translator (over 20 and counting).

Hopefully this example will mean that anyone buying the SDK will be able to get at least one new Amiga program running. Of course, this isn't the full extent of our developer support program. Gary Peake, our Director of Developer Support is constructing a comprehensive set of resources including web resources, forums and technical books. We encourage all developers to register, take advantage of, and more importantly become a part of the new Amiga developer community. We know that we won't succeed on our own, or by telling developers what they can and cannot do. We will only succeed in partnership.

We are on a new adventure. There is much here already, and more to come: Full audio, streaming, 3D, some great partner announcements and a pure system. In addition, there is our next generation architecture, codenamed Amie, which is being built as we speak and which will, we hope, be an important player in the digital world of tomorrow.

We look forwards to working with you, playing with you, and smiling with you.

Fleecy Moss (VP Technology) on behalf of the whole Amiga Team,  
Amiga Inc,  
Snoqualmie, Washington USA



## Amiga Nameplate Competition

So, you noticed the new nameplate used for this issue of Amiga World. It looks different from the one used in the last issue. Well, get used to change, gentle readers, because you'll see a different nameplate used in most future issues of Amiga World.

We know that the community is loaded with talent and we'd like to display some of that talent in each issue. What better place to display such talent than on the cover of Amiga World?



If you are interested in having your work considered for the nameplate of Amiga World you should email your creation to [amigaworld@amiga.com](mailto:amigaworld@amiga.com). One winner will be chosen by a panel of Amiga VIPs to have their work displayed.

The guidelines are simple. Create your nameplate using the Amiga World title and subtitle (So The World May Know). You can use any software you like. It should be created at 72 dpi and be between 500 to 520 pixels wide and between 155 to 165 pixels high. The size in inches should be between 7 to 7.5 inches wide and 2.2 to 2.5 inches high. It should be submitted as a TIFF, GIF or JPEG. You will be contacted if you are chosen. That's it!

The designer of this issue's nameplate is Brian Huebert, a freelance graphic designer ([www.icenter.net/~huebs/](http://www.icenter.net/~huebs/)). Brian used Photoshop 4 on a frequently crashing Windows95 Pentium to create this nameplate, although he honed most of his artistic skills using Photogenics and PPaint on the Amiga. Nice work Brian! ♦

Vince, continued from page 1

could talk to each other, well, this had real potential.

That's when the excitement starts. When you can build a system like the one Amiga Inc. has envisioned, you have an exciting new computing paradigm opening up to you. The last year has seen several companies offer new solutions for home networking through phone lines, electrical outlets, and other non-traditional network connectivity methods. Why? Because users want and need to be able to share data and devices between multiple PCs. Amiga is going to go beyond that. I can't fully tell you about it today, but two years from now Amiga will be the forerunner in home digital data access.

Since the Amiga assets were bought from Gateway in December 1999, Amiga employees have been working very hard to release their first product, and this first product is now ready to be released as the Amiga 1 Software Developers Kit. This is our first step in creating new digital content for the Amiga platform. This release will be primarily for developers who will make development tools for the platform. Other releases this summer will continue to expand the system and will contain additional capabilities by the Operating System and additional tools. Things will only grow.

As a community, our success will depend on us all creating quality content for the different Amiga devices that will be available to the market. And judging by past community efforts at producing quality content, I think we are all going to be quite successful. Good Luck to us all. ♦

Vince Pfeifer

## Credits and Contacts

### Publisher

Amiga Inc.

### Layout and Design

Equinox Design

### Editor

Fletcher Haug

### Assitant Editor

Robert Bryant

### Senior Writers

Robert Bryant  
Ted Wallingford

### Contributing Writers

Fabian Jimenez  
Steve Folberg  
Bruce Lepper  
Fleecy Moss

### This Issue's Writing Credits

#### Hyperion: Ready For The New

##### Amiga

Bruce Lepper

#### Making User Groups Work

Fabian Jimenez

#### Amiga Resource Center

Steve Folberg

#### The Amiga Foundation Layer

Fleecy Moss

Mission Statement: Amiga World is here to provide readers with news, information and insights on the New Amiga and to chronicle the events, activities and projects of Amiga Inc.

Copyright 2000 Amiga Inc. All rights reserved. Nothing may be reproduced in whole or in part in any way without the expressed consent of Amiga Inc. Amiga is a registered trademark of Amiga Inc. The mention or reference to other companies or trademarks is not a challenge to the trademarks or copyrights concerned. All other trademarks are property of their respective owners. Amiga Inc is solely responsible for the content of Amiga World newsletter.

Amiga Inc: Phone: 425-396-5660; Fax: 425-396-5661

General Inquiries: [friend@amiga.com](mailto:friend@amiga.com); Developer Inquiries: [developer@amiga.com](mailto:developer@amiga.com);

Amiga World Inquiries: [amigaworld@amiga.com](mailto:amigaworld@amiga.com)

World Headquarters AMIGA, Inc., 34935 SE Douglas Street, Suite 210, Snoqualmie WA 98065