# TECHNOIDS

The free BeOS magazine

**Pro Edition | PE Edition | Zeta | OpenBeOS | B.E. OS | Cosmoe**

## EVOLUTION

**The rise and fall of the "PC"**

## OpenBeOS

**Interview with Axel Dörfler**

**Article**

Raycone back on track

Overview- graphic software for BeOS

**WorkShops**

Package Builder - quick reference

cifsmount - access windows shares

We hope that you had a good time since the release of the first issue. We were really surprised that Techniods was listed on BeBits as one of the most popular downloads (admittedly, for a very short time ;) -- over 400 downloads total. We were also honored that some BeOS community members contacted us to aid our project. We received good suggestions from many individuals, in particular Bernd Korz. Thank you, guys! We want to thank some other geeks, too.  They've translated Technoids into several languages on their own, so that you can read this magazine in German, English, French, Polish and even Greek. Greetings to Remi Grumeau, Adam Szczech and Thanassis Anastassiou for their great work.

We also thank Joao Carvallho (Portugal) for the excellent layout.  This layout is now used in the English version, and (in part) in the German one. Thank you Joao.

And, finally, thanks go to our proof-readers. They have put much effort into correcting various errors throughout the magazine. We hope that the magazine has more "professional" feel as a result.

Now, enjoy the new issue and keep the excellent feedback coming :)

Best Regards,

Matthias, Florian and Lelldorin

**"They've translated Technoids into several languages on their own, so that you can read this magazine in German, English, French, Polish and even Greek"**

# In This Issue

# OpenBeOS

www.openbeos.org

# Interview with Axel Dörfler from openbeos.org

**Axel Dorfler spent a bit of his valuable spare time to answer some interesting questions about OpenBeOS for Technoids. As a leading programmer and primary contact for the OpenBeOS project, Axel earns our thanks.**

**technoids** It would be nice if you could introduce yourself to our readers.

**axel** My name is Axel Dörfler, born in 1976 and at the moment I'm studying "Computing Linguistic & Artificial Intelligence" at the university of Osnabrück.

**technoids** When (and how) did you get into BeOS?

**axel** Mainly this was started with my personal anger against Windows. I've been addicted to the AMIGA and all versions of Windows older than NT aren't "serious" operating systems.
Until 2000 I didn't even own an "IBM PC". I tried a BeOS 3.x release on my father's PC, but this version was only running in b/w and in 640*480- not really amazing. Even the featured applications crashed often while I was testing the system. So, I haven't followed BeOS for a long time.
I was registered as a developer at Be, but some day they stopped sending me newsletters- OK, I wasn't active at this time, so I think that was the reason. This happened even before the first BeBox was released.

Three months after I got my PC, BeOS R5 PE was released, and I tried BeOS again. My PC felt lonely before, because I was still using my old AMIGA 600 - it feels faster than the PC with Windows 98.
BeOS' amazing speed was not the only reason to use it: everything worked fine and all my hardware was supported. From that day on, I'm a BeOS Fellow.

**technoids** What pushed you to initiate Open BeOS?

**axel** Well, I didn't initiated it. In those "early" days Michael Phipps founded it with a small group of other enthusiasts. Back then I wasn't in the project, cause I didn't thought that Open BeOS could be a serious alternative in the next few years. This all changed when Be was acquired by Palm. The official development was stopped and Yellowtab's ZETA won't be available soon.
Because I don't like the other alternative OSes too, I've decided to take the advance with Open BeOS to create my own "up-to-date" BeOS - Version. Together with Bruno G. Albuquerque (also known as BGA) I've joined the Open BeOS Project. And I'm still there, because

I've never met such nice and skilled programmers before.

**technoids** Are you still satisfied with the programming possibilities (API, Interfaces ...) in BeOS today?

**axel** More or less I am. OK, there are some mistakes and errors in some implementations (like the MediaKit). What I'm mostly missing is an easy and efficient way to create a GUI (have a look at Marco Nelissens amazing liblayout.so), a LocaleKit (still in the making) and a modern Network- Stack.
As you can see, I'm not expecting too much. Never the less it would make sense to improve some details and add some current design patterns from modified class libs. But after all the current state of BeOS is not limiting my work.
This is also the point where Open BeOS starts:
in the first step, we're creating new implementations of the existing functionality. Details are added later. For example, who likes to work with an easy to use MediaKit, if it's not able to play a movie synchronized with it's sound?

**technoids** If you also know other Operating Systems, would you say that Be reached the target of an easy way for developing software?

**axel** Yes, it is still competitive to other modern systems like Java or NextStep/Mac OS X. Windows (at least, in the pre .NET era) is a good example of how you shouldn't create an OS from a programmer's point of view.

**technoids** Back to Open BeOS: will you improve or add functionality or will you simply "reprogram" BeOS in a free way?

**axel** Our goal is to implement the BeOS R5 API in an optimized way. The API itself won't change. This could be done in later releases. From an inside view we change a lot but the API remains. Developers and users will benefit of this, but the API will

still "feel" like it always did.
This also provides the advantage to change things inside the API and instantly see, how existing work is responding to the changes.

**technoids** Unlike other free BeOS Projects you're using a "real" BeOS Kernel instead of a Linux/UNIX System. Why?

**axel** Well, there are several reasons. At first the GPL. Don't get me wrong, I like the GPL but it's sometimes in the wrong place. Imagine what would happen if XFree was licensed under the GPL. I think all users would need a VESA- Graphics-Adaptor even today. And also the driver



support within Linux isn't perfect. Especially since most manufacturers never release an official driver. Also the GPL creates the not always welcome atmosphere that everything has to be distributed for free. For a small market like the BeOS- Community this could be a wrong signal. But even by a technical aspect I would
prefer something like FreeBSD or Darwin (Apple's MacOS X Source -- Ed.).
Back to your question on why we've chosen a BeOS styled kernel (NewOS). Of course, if we had chosen FreeBSD for example, our kernel could be more

completed today. Now we have to work for years to reach all of our goals.

But it's not as bad as it seems. If you take an existing Kernel it is much harder to create every function that you want it to have. It's more simple and more efficient to create a new one.

We can create BeOS again "from scratch" and when it is completed, everything will be implemented more clearly and from the same source.

If you compare Linux- or BSD- Source code with our source, you will see that our is more efficient and even more easy to understand. I like this method of creating a well organized project, and it allows us to create a really modern and transparent system.

That's what keeps me motivated- and motivation is THE vital artefact if you are working on a free project.

Of course B.E.OS can present results earlier. But transparency and efficiency(simplicity) couldn't have been primary objectives when Linux was created. I'm not trying to deny the incredible things Linux can do for power users.

**technoids** Will Open BeOS be 100% BeOS compatible or is recompiling necessary?

**axel** Open BeOS will be 99.8% compatible. This means that any application based on the *public* API will be working from the start. Programs based on undocumented or unofficial APIs won't work, because we don't implement them or implement them in another way. The original DriveSetup for example is such an application- but for such essential tasks we will provide our own software.

All those features allows you to run compatible software (99.8%) from the beginning. Incompatible apps (0.2%) will never run, even recompiling can't change that.

An exception is PowerPC Software which needs to be recompiled, because the PPC is not binary compatible with the x86.

**technoids** How far would real compatibility make sense from a technical

aspect? I'm thinking of software OpenGL for example.

**axel** That's a question of implementing, again. The current version of MesaGL can be used as drop-in-replacement (it is buggy at the moment) and older software won't be affected. But in this case no one can expect, that we won't use a better solution (OK, nothing in range so far).

Or another example: BeOS has big problems with handling primary and extended partitions. There is no reason not to fix it. We won't let us limit only to assume that such things behave like in BeOS R5. And such implementations won't affect compatibility at all.

**technoids** What you think about Bernd Korz's ZETA OS which is not Open Source?

**axel** The only thing I don't like is that ZETA isn't already completed and in stores. I hope that ZETA will bring a bit more life to the BeOS Community. We are part of it too and it would be great for all of us. Bernd, I wish you good luck!

**technoids** Does it make sense that we get 3 free BeOS- Flavors and one commercial?

**axel** No, but it's not so bad. If different programmers are not able to find a unique direction or target, it's only a

consequence that different projects appear. None of our projects (except ZETA) is commercial, neither we have to reach business targets nor to reach fixed release dates. A programmer shouldn't join a project that doesn't meet his targets, especially if he isn't paid for it. Most programmers also don't do this to make everyone happy.

But if you look besides the different targets, all projects will benefit from Marco Overhagens implementation of the MediaKit for example. We let the other projects profit from his work cause our license allows the use of such improvements by all BeOS Friends.

Although different targets led us into different projects and some work will be done twice, basic elements are still the same. So co-operation has already been there and will still exceed while our projects are growing. For example Bill Hayden from COSMOE has also a writing permission to our repository.

Long answer, short sense:

our common users won't profit from this development but I'm sure that some day our different projects reach the same level and targets again.

**technoids** From a historical point of view: would you say that only closed products with their own operating system were successful (AMIGA, Commodore, Apple, IBM PC + Microsoft)?

**axel** No, it shows us the opposite way. A "well known system" was widely spread because it didn't brought it's own hardware. Linux is heading this way too.

**technoids** Would you say that a computer like the BeBox with higher sales could have succeeded better? Or is BeOS still alive because it runs on a "normal" PC?

**axel** Last one. If the BeOS Community would only exist of BeBox- Owners then it could have already been buried.

**technoids** Can you tell us how far the OpenBeOS Development is so far? Are there any functions that are sill working?

**axel** In my opinion we are in a good and fast development. We have working things from all topics by now. Sometimes only our own unit tests. But those are working more stable than the original implementations. And even complex technologies as the BFS or the MediaKit are still working in early versions. Marcus just released the MediaKit as an alpha and the work we've done right after this event is simply amazing.

**technoids** You are working on a PPC-Core. Will there be special PowerPC Software again or is OpenBeOS able to run applications on any platform?

**axel** All software has to be recompiled for this purpose but there are no further requirements. In comparison to the original BeOS the same compiler is used on every computer. This provides a much easier handling with different systems.

The PPC Kernel is a port of our kernel to the Pegasos, a new and state of the art PowerPC- System which is developed by Genesi. The PPC Version could be the first one able to boot from a harddisk without "external help". This depends on how "buggy" my Openfirmware is.

**technoids** Will your PPC- Port allow the use of modern PPC's? The "old" BeOS was limited to the 604.

**axel** It's even worse. My port will only support modern PowerPC CPUs. Support for older ones could be done after that, but I can only develop versions for the hardware I own.

**technoids** Are you satisfied by the support of the community? How often your team is working on OpenBeOS?

**axel** These are 2 different questions, aren't they? Of course we're working on BeOS at least every day. Thanks' to the world-wide spread of our developers we are working around the clock, and that's not a joke!

I'm proud to be a part of a great Community!

**technoids** Do you think that BeUnited can control all BeOS Flavors?

**axel** We will see. We will also see if there's a need for that anyway. I hope that we can reach such a solution after all.

**technoids** Short question: are you working on Open BeOS full-time or in your spare time?

**axel** *lol, if someone is going to pay for my work, I would be programming all the time. But I'm a student and I work only in my spare time on OpenBeOS.

**technoids** What do you expect from BeOS?

**axel** I'm just hoping that BeOS can be an alternative to the existing systems we know today. For myself it's not important how many users will realize that- a small market provides it's advantages.

**technoids** Thank you very much for this interview!

**axel** No problem.

*Interview was recorded in German via email. Recording and Translation by Matthias Breiter.*

- Open Source Operating System

- MIT Freeware License (like BSD)

- R1 targeted on BeOS R5 Compatibility

- Contains updated drivers

- Features corrected APIs like a new MediaKit

- Work is contributed by enthusiast programmers

- Current status is Pre- Alpha or Alpha (like MediaKit)

- New features are planned for versions after R1- no known details

further information provides
http://www.openbeos.org

# Every new Generation

## Evolution of the computing industry

**In the last issue I was talking about the immediate future and how we can profit from this evolution. Since the first Technoids I get some new ideas to show what we're talking about. The problem with the PC (includes every kind of OS like Windows, MacOS, BeOS, Linux) may not be clear. Some people will also ask "Is there a problem?".**
**The key is buried (like it's often) in the past, in the young history of computers. While it's a short history, we can still learn a lot out of it.**

### the competition

No other technology has been developed so fast like the computers' one. Yesterday's vision becomes reality today and is obsolete tomorrow. This fate belongs to us since April 1st 1976. Apple was starting their operational business and sold their first home brew computer (a kit to build one). Sine then we have war. New models were released like on an assembly line. One year later Apple introduced the Apple II, the first real commercial success in computing history. Computers were the new "Klondike", a license to print money.

Almost every day new companies appeared and some of them grow big. Like Commodore, ATARI, AMIGA, Microsoft, Compaq and many other.

Until 1994 every year appeared new computers and new operating systems, delivering new ideas, becoming new "every-day-devices" or being buried as fast as they have appeared. This was the heart of the IT- Market for years. New systems provide new innovations or combine existing ideas to something new. Also applications profit from this developments. Programmers use ideas from their opponents or from older software of their own- the user got most out of this. Documents became compatible and networks crossed systems and ideologies.

And things good for your computer can't be bad for your video game, can't they? So videogames are in the same way of competition since 1977. Companies appear and disappear, ideas become games, new games deliver new ideas and so on ...

This competition has been reinventing computers and videogames, has been redefining them and has been reaching new customers.

### does it always went like this?

At first you should think about what has happened. Look on the table of the IT-History. Yes, it is like this, cause it happened this way- that's the reality. Well, we can learn form this development a few basic rules:

- manufacturer XY develops new product
- uses his own experience and ideas
- uses ideas from other products

- creates new, better product

Don't care if the manufacturer is successful with his product- he gives us his knowledge. Cause his ideas remain in his product. And the more brilliant his ideas are, the more often they are used. Sometimes huge things like the graphical user interface of the macintosh, sometimes only small and lovely details.

Of course you can say now "Hey, Matthias, we aren't that stupid. It always wents like this." Really? We will discuss this later.

But at first we have to figure out another important topic about the IT Industry. Success of a computer or videogame system not depends only on one guy but on many guys.

## all for one- One for all

Every new system is a new platform. An unwritten story, blank piece of paper. The developer delivers first ideas, a little hint of the platforms direction. Maybe some apps and some games also. But he can't do more than that. Ok, some companies release now and then some of their own work, but the main part is filled by 3rd Party Developers.

On every new system these consist of veteran companies and new studios or even single freelancers. Because new systems are like a blank sheet, small studios can make a hit as good as the big players.

New Platforms are always a chance for small developers and young talents. Ideas can be realized without "mind barriers" and you benefit from the fact, that the big ones also need time to create well optimized software. We can also proof that by viewing the history. New machines raised new companies, who've not even existed before.

And even the whole system runs bankrupt- skilled developers are welcome on any platform. And their ideas "walk" to other products, will be picked up by other guys and all remains in the circle of evolution- with every new generation.

## really?

Returning to our topic from above "Does it always went like this?" please have a look on our little time- table. As you can see, the last new computer was introduced in 1994- the Acorn RiscPC. The videogames on the other hand remained in their natural evolution and developed further and further. 1993/94 was the time of the big bang when Microsoft were overrunning the computer market. Since 10 years there has been no real innovation and the basic techniques of the PC are from the early 80s. The PC missed the vital process of renewing for over 10 years now.

With the begin of the 90s the PC was an amazing platform. Games like X-Wing, Wing Commander III, the several Lucas adventures and many other pushed the PC far beyond it's limitations. But then it ran into a stagnancy. No new system- no new ides- no new software and no new developers.

What the world needs is a new computer. Doesn't matter if it wins or fails. Only importance- to be here. Then (and only then) new companies will appear, new ideas will be born, new software will be written and everything will be inherited to every new generation.

### anything to do with BeOS?

Normally, technoids don't want to publish such ideological texts. But we want to show you what can be done.

yellowTab owns a mighty OS- Zeta. It's the most competitive OS available and the most dangerous for the PC. But only an OS can not run into battle- people buy computers- not operating systems.

Apple or Acorn for example. These companies have no big market share but they are very successful and they are rich. Ok, Acorn doesn't even exist any more but the computers are still popular and many companies deliver new hardware and software. Same with AMIGA.

I don't say "Hey, lets go to manufacturer xy and ask them to build us a computer." but when Microsoft wasn't much bigger than yT today, they've created their successful MSX Platform. And by the way, the MSX contained only available hardware.

Why no one does such things today is clear, too. After using the same PC for years and years your horizon gets a bit ... small.

*by Matthias Breiter*

| | TIMESCALE computers & videogames | |
|------|------|------|
| 1976 | Apple I Commodore PET | |
| 1977 | Apple II | ATARI VCS (2600) |
| 1978 | ATARI 400/800 | |
| 1980 | Apple III ACORN Atom Sinclair ZX80 | Colecovision |
| 1982 | ACORN BBC Micro | |
| 1983 | Apple LISA ASCII / MS MSX | ATARI 5200 |
| 1984 | Apple Macintosh | |
| 1985 | ATARI ST AMIGA A (1000, 500...) MSX 2 Robotron KC-85 | Nintendo NES |
| 1986 | Apple IIgs NeXT | ATARI 7800 SEGA Mastersystem SNK Neogeo |
| 1987 | ACORN Archimedes | |
| 1988 | NeXT Cube ASCII/MS MSX 2+ | SEGA Megadrive NEC PC Engine |
| 1989 | | ATARI LYNX Nintendo Gameboy |
| 1990 | AMIGA A+ (3000) | Nintendo SNES SEGA GameGear |
| 1992 | AMIGA AGA (1200,4000) Linux | |
| 1993 | Apple PowerMacintosh | AMIGA CD32 ATARI JAGUAR |
| 1994 | ACORN RiscPC | SONY Playstation 3DO FZ-1 |
| 1995 | BeBox/BeOS | SEGA SATURN Nintendo N64 |
| 1996 | | Apple Pipin |
| 1998 | | SEGA Dreamcast |
| 2000 | | Playstation 2 |
| 2001 | | Microsoft X-BOX |
| 2002 | | Nintendo GameCube |

# Visual Dreams

## Selection of fine BeOS graphics dedicated Apps

**BeOS has got much to offer, also on the area of image viewing and editing. Unfortunately the lack of drivers is the key restraining factor in really exploiting the power of these applications. It would be nice to see BeOS support more peripheral devices such as scanners, digital cameras and web cams. The applications to work on the input by this hardware are often already there under BeOS. Some of them have even gained a position in my BeOS poll (see http://beos.holgerwendenburg.de/), and I will mention their position if they have.**

### image viewing

Let's head over to the image viewers. Have you ever scrolled through a digital photo album containing far more than 100 images, without having to reload them? If not, you should check out an application called "Butterfly". It is a great example of how fast BeOS is.
Another application worth your while is Peek, which perfectly does what it's called. Both applications can be categorized as ImageViewer and ImageMounter.

### image editing and manipulation

Now for the applications to draw and edit images with. I have put them in alphabetical order. We will start with ArtPaint (25th place) which has now moved on to a second version and in features has now cut itself loose from its' predecessors.
Subsequently there is BeCasso, which shares the 25th place with ArtPaint and gives honor to its name.
BePhoto Magic can also be used with much ease.
An application quite different from the ones above, but still a real killer, is Easel.

For many professional designers their real h(e)aven on Earth. We now get to a true highlight within the image editors which also got a good rating in the poll (17th
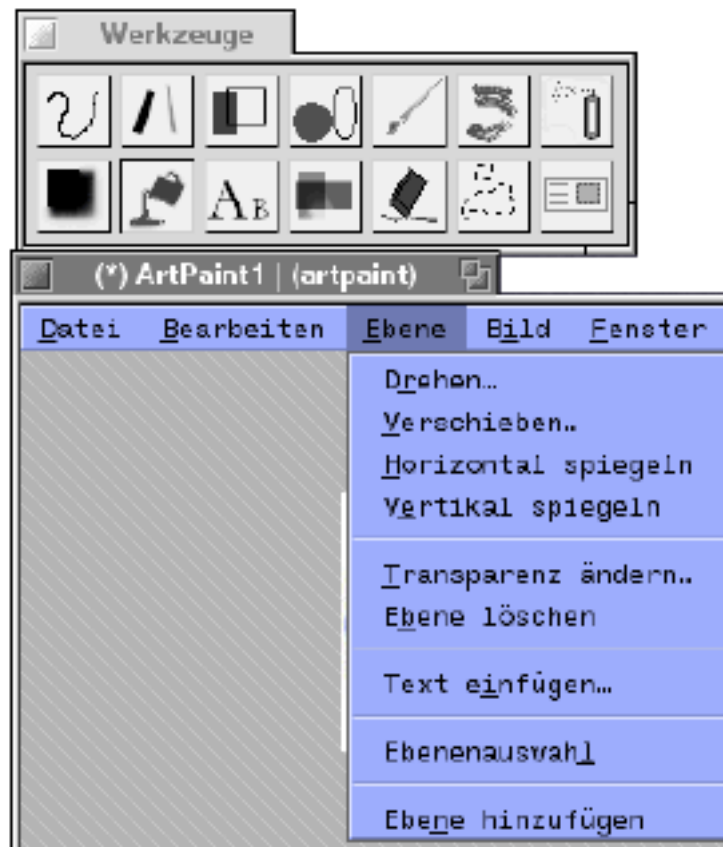
**ArtPaint 2 - every Art got its' purpose**

place). Refraction (formerly known as Inferno) ends up far above all other comparable applications in the customer evaluation.
Update: Xentronix Software delivers a new beta program. You register a Copy for around 40 EURO and you will receive all updates till final version 1.5 for free. A good way to support the developer, provide feedback, use the software now

(already better than the other ones) and save a lot of money.

Last but not least: BeOS can also show its funny side, as proven by the application TuxPaint, which is ported from Linux and works great as a drawing tool. It comes with supportive sounds and other extra features. This definitely will encourage younger BeOS users to make their own drawings.

### special forces

Now we are getting to applications that perform special tasks, such as converting images into different formats. ImageGrinder shows little thumbnails of your pictures and is easy to work with.
To convert massive amounts, the next application is your weapon of choice, explaining its purposeful name: MassivePictureConverter. It uses features from both PicConvert and TrackerAdd-On.



**BePhotoMagic - it's magic !**
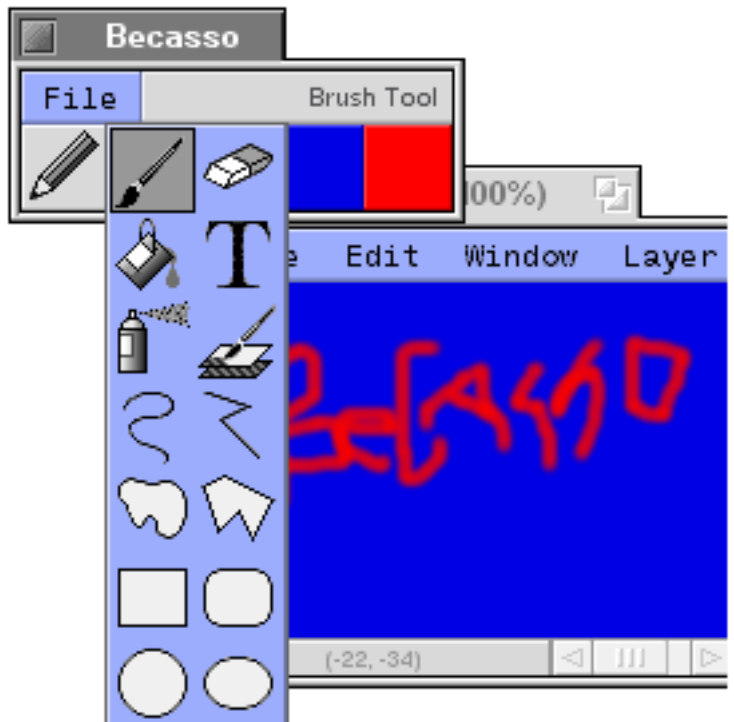


**BeCasso - Picasso would be happy**

### the others

Of course this is only the tip of the iceberg in the variated world of BeOS, as there are yet so many other image applications. Think for example of Blender (17th place) or bmodeler which allows you to create 3D-images. To run Blender, here's a little hint: on startup it will need libpython1.5.so which is supposed to be in the /home/config/lib folder. The best thing to do is install Python 2.2.2 which has an .so file included with the right functionality (but since it involves a newer version of this file, it's not the exact file Blender is looking for). Copy this file and rename it into libpython1.5.so.

If you have any questions, please do not hesitate to contact me or Technoids. post@holgerwendenburg.de
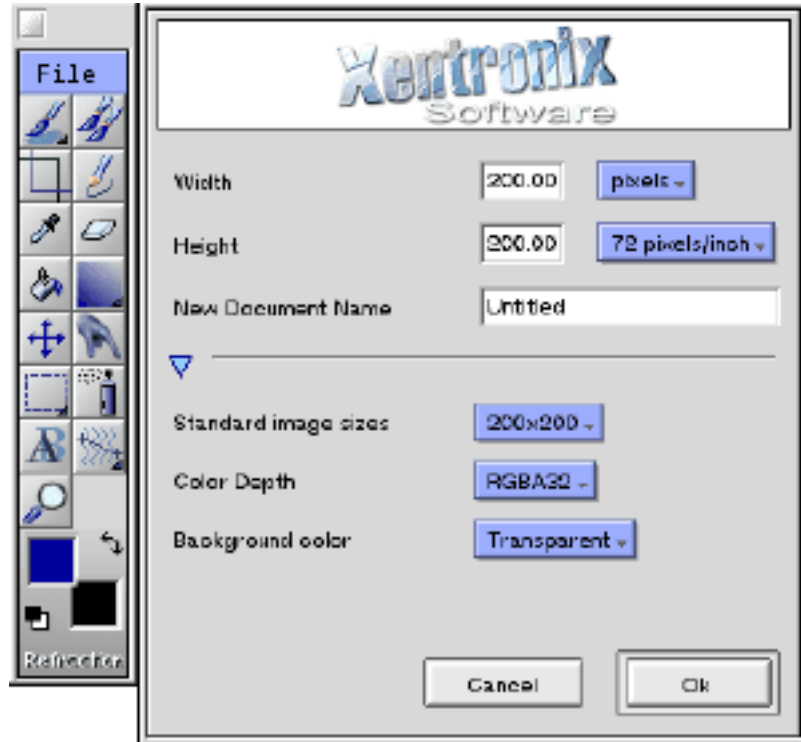
*Article was written in German by Holger Wendenburg. Translated by Eddy Zeeder. Screenshots were taken by Florian Thaler.*

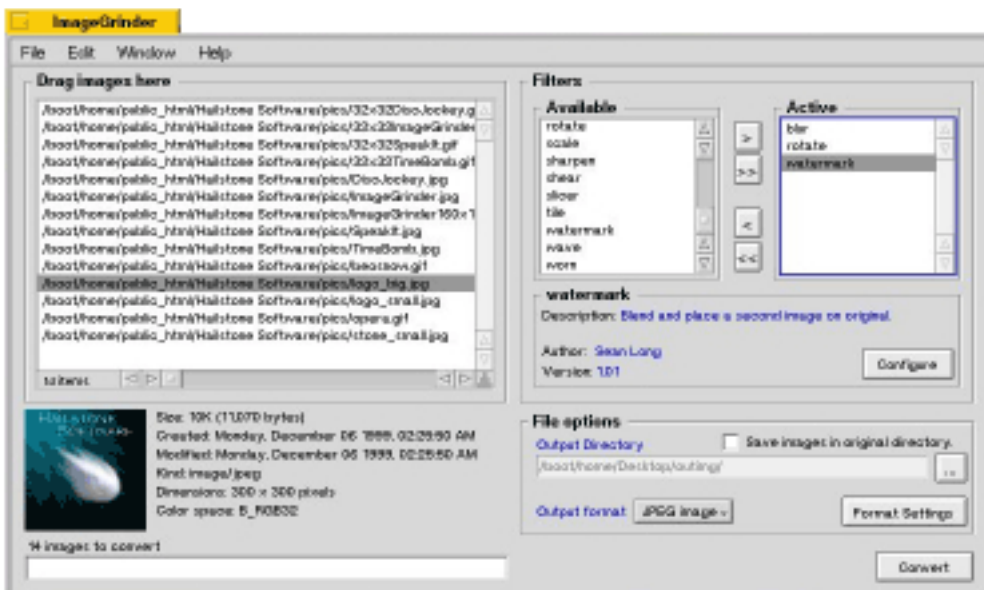Have a look on the next page for some more impressions.

EDEN
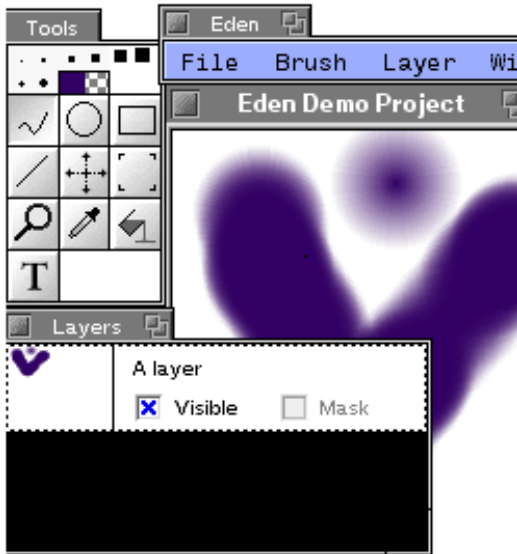


Good for Kids - TuxPaint



Xentronix critically acclaimed software: Refraction



Manipulate all pictures with one click - ImageGrinder

# Raycone is Back!

## Raycone Software is back on the track with their HTML Software

After a break in the development of the "Insite" HTML- Suite Raycone is back with good news. 8 months ago the sales climbed down to 0, but have increased since then. So Raycone is back on the track. How far the evolution of InsiteConstructor and InsiteDesigner will grow, depends on how many licenses will be sold in the future.

On our question, if Raycone will develop a whole new BeOS Product they've commended that much bigger sales would be required. But never the less Ales Nedomlel reported in his email, that new versions of InsiteConstructor and InsiteDesigner are in progress. Creator could appear in July, August or Septembre and Designer is announced for late Fall this year.
Constructor is extended with new tools and all the features a good HTML Editor should contain. Many changes are only targeting details which will make editors life a bit more comfortable. Now you can toggle line count, hide the left panel and

so on. The most of the features were suggested by the buyers of the current version- so we're expecting a good optimized product.
The programs are already available in several languages like Czech, English, German, Russian and so on. Creating a new language is easiest- the developers included a localization utility (localedit).
After all, both apps will be available for around 50$- so no changes in this direction. This seems a bit much for BeOS Software, but competitive apps on other platforms are even more expansive. And by the way, you can get a free demo from Raycone on their web site. So try them out and if you are satisfied- you can buy one.

Raycone Website with lot of more info, screenshots and free trial editions:
http://www.raycone.com

*This article was provided by Florian Thaler. English translation by Matthias Breiter. Special thanks to Raycone.*
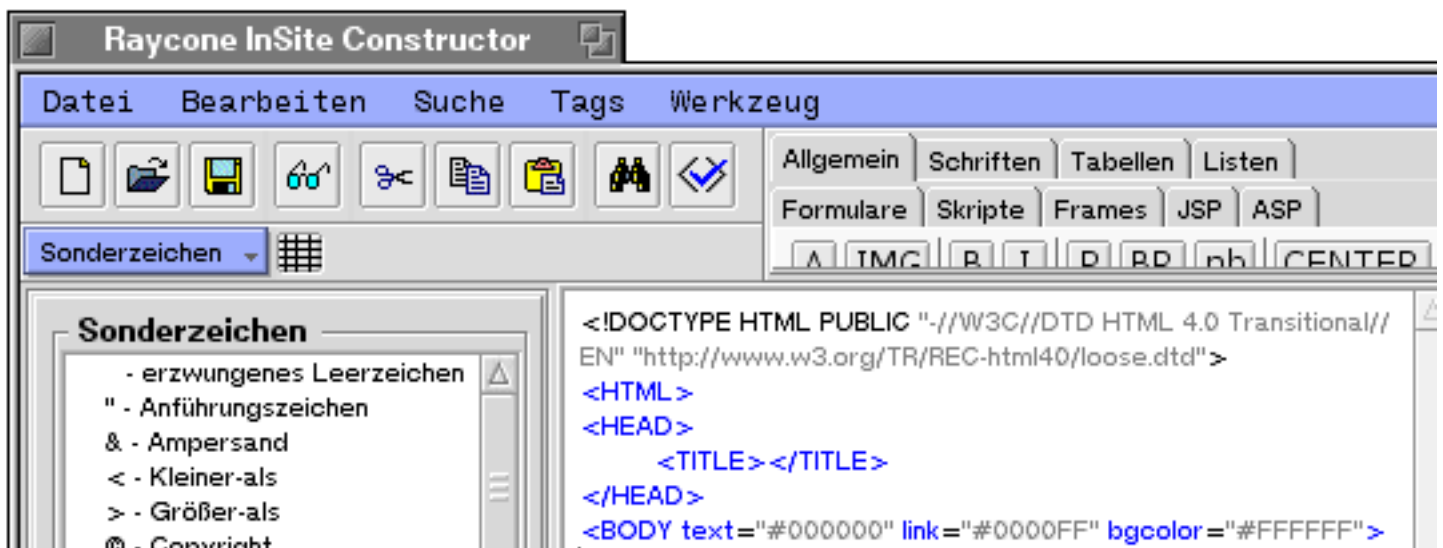
# Package Builder

## A short introduction to an useful app

**This program allows you to create installer apps. This means that all used files are collected in one package and that all files will be installed where they are needed. Please remind the libraries- without them no program will work. While this is a manual step when using standard packages (like Zip or RAR) Package Builder does this automatically. Also PB compresses the content of the package to reduce the file's size. The Package Builder is included in every BeOS Pro Edition but has to be installed separately.**

### let's have a detailed look on the functionality (use the top menu):

**- New Package**
Opens a new plain package without overwriting any open project

**- Open ...**
With this entry you can open existing packages

**- Edit Groups**
Here you can create specific groups which are later responsible for allowing several packages to be installed during installation. Without this declaration the installer doesn't know what to install later.

**- Edit Destination**
Here you can select the destination where the installer put the files. You can choose from standard destinations or you can type a specific path manually. This is important if the files need to be placed in special system folders.

**- Save Package**
Saves the project under it's name. This doesn't creates the installer- this have to be done with the Builder which opens every time you save the project.

**- Save Package as...**
Allows you to save the package under a different name. This is a good way for debugging your package. You can easily switch back to earlier versions stored under another name.

**- Close**
Closes the current project. Asks if unsaved work should be saved now. Doesn't quit the application itself.

**- Preferences**
You can make some settings for the Package Builder

**- About PB...**
Shows some information about the Package Builder.

**- Quit**
Almost the same as close, but quits the application ;)

### items Menu

**- Add Files**
You can add single files to your project...

**- Add Folders**
... or include whole directories.

**- Add Patch**
You can create an update function with this one. Old versions of a file will be replaced with the newer one. Simply specify the old file and the new file. This keeps your whole package up to date.

**- Add Shell Script**
You can select a shell script (#!bin/sh) here. This script will be started just after the installer finished his work. With the function you can launch your app after

installation or show a read me or whatever.

### - Extract Items
You can decompress files directly from a package. This makes sense if you don't have another backup of the file any more.

### - Delete
Deletes a specified file from your project.

### - New Folder
Creates a new folder inside of your package.

### - Rename Item
What does this function do? ;)

### - Select All
Any questions ?

## Installation Menu

### - Generate R4...
If you select this one the package will also work within BeOS R4.

### - Splash Screen
You can integrate a graphic (logo, artwork, welcome...) in the installer. This graphic is easy to include. Simply open SplashScreen and put a graphic on it by using drag&drop. All kinds of graphics recognized by your BeOS Translators are supported.

### - Install
Here you can perform a lot of settings. Because these settings are vital we'll explain every of them.
*/Install Folder*
If you mark this one, all files and

subfolders of your project will be placed inside the "Install Folder" (can be selected from the Edit Destination Menu). You shouldn't check this option if you have specified special destinations (or different for several files).
*/Display Folder Selection Menu*
If checked the user can specify his own destination. Not recommended if your files have to be placed in a special (system) location.
*/Install Description*
You can enter a short description about what is going to be installed here.
*/Display Pkg Help*
Allows the user to display a help text in the installer. You can use a generic text or you can type your own.

*/Display Text at Installer Open*
If this is selected, the Install Description will appear instantly when your installer is launched

- Package Settings
Again a lot of interesting settings.
*/Package Name*
Specify a name of the package.
*/Version*
You can add a note about the version of your software (e.g. 1.0 or 0.0.1a)
*/Developer*
You can enter the name of the developer here
*/Release Date*
Enter the date when the package is/was released
*/Description*
A short description of the delivered software
*/Software type*
You can add information about the software's license, like "commercial", "freeware" and so on...

Of course there are many other options, but the ones mentioned here should be enough to start working with the Package Builder. It's a great and mighty piece of software and a good way to deliver your applications.

*Written by Christian Albrecht. Adapted from the German version by Matthias Breiter.*

# CIFS Mount

## How to connect to Windows and SMB Shares

**In this article I will try to explain the beginner users of BeOS operating systems, how we can see and use shared folders from Windows and other systems (SAMBA).**

Most of us know, that in BeOS, similarly like UNIX and other operating systems we must mount partitions, if we want to see it and work on it. If other partition or disk is in our computer it is no problem, mounting partitions in BeOS is very easy, we click right mouse button and select "mount". But what to do, if we want to mount partition (or logical disk) of our friend from network ? And what to do that logical disk of our friend from network will be like our disk ? In windows this similarly operation is called "mapping". But how to do it within BeOS ?

### requirements

We must to know three basic things:

1. Address IP of this person.

2. Name of computer( this visible in network for other computers e.g. JOHN, TOMMY, etc.)

3. Name of shared folder( like GAMES, VIDEOS, MUSIC etc)

Our friend working under Windows operating system has to to realize also following conditions:

1. He must working on TCP / IP protocol, he cannot work for example only on NetBeui protocol.

2. He must have static IP address, he cannot work on DHCP because we have to knew his IP number

3. Of course he must enable "Files and folders sharing" under Windows, and he must share someone Folder for us.

### general information about 'cifsmount'

We can see other network computers by using Common Internet File System. CIFS (Common Internet File System) is called also SMB (Server Message Block) or NetBIOS in Windows operating system

To remote mounting folders in BeOS serves just command cifsmount, this small program is standard in every BeOS.

We can find it in folder:"/boot/beos/bin/" of our system. If from some reasons does not have it there, we can download it from www.bebits.com.

It is proper also to check, whether in catalogues:

/boot/home/config/bin/ and

/boot/beos/bin/ is found file ksocketd.

it is important file to working cifsmount! To mount by remote control disc or folder we will have to to use BeOS Terminal, Terminal we will open clicking on icon BeOS (equivalent of button like 'Start' from Windows system), then we choose Applications, and then Terminal.

## commands

In Terminal uses oneself only of textual commands, so will not be of use us one self here already mouse.
Syntax of command cifsmount looks as follows:

```
cifsmount -I numberIp -W WORKGROUPNAME -d \\\ computer_name\\SHAREDFOLDERNAME login password /mountFolderPath
```

Or more clean:

```
cifsmount -I str1 -W str2 -d \\\\str3\\STR4 str5 str6 / str7
```

Str1-7, I was called seven strings which we must know and write in command. Now I will say what every string mean:

string1 - this is number IP of computer our friend, which we want to join, we must know this number!.

string2 - this is name of WORKGROUP, in of which works our friend. If we are mounting Windows folders, workgroup name is no valid (we can write everyone workgroup name)

string3 - this is name of computer which we are mounting. This name our friend puts in Network Preferences (under Windows). There are names like example: Adam, Tom, Tommy, George, Server etc.).

string4 - this if name of shared Folder like GAMES, PROGRAMMES, MUSIC etc. Attention! This word we must necessarily write with big letters. If our colleague looks like:" music"," Music", or " mUsIc". We must write "MUSIC".

string5 - this is user login name - if we are mounting Windows folders it is no valid what we write here. But if we are mounting UNIX or LINUX folder shared through SAMBA where login is needed, we must write login here.

string6 - this is password of share. If our friend give password for his folder, we must know it. It is no valid that password is to 'readonly' or to full access. We must know this password, but if we are mounting Windows folders it is no valid that we writ big or small fonts. But if we are mounting folders from other systems (SAMBA) big and small fonts have meaning.

string7 - this is path of catalog in which we will mount. Example if we write:

/files - then cifsmount will mount files to folder 'flies' in main catalogue.

/boot/home/files/ - then cifsmount will mount files to folder 'files' in our 'home' catalogue.

Or if we want to mount to folder 'Music' on our Desktop, we must write:

/boot/home/Desktop/Music

Attention: We must earlier create this folder. We cannot use 'cifsmount' if we haven't create folder to mount.

Let us sum up:

- Name of shared folder we must write capital letters.

- We must have created catalogue to which we want to mount files from network. (big and small letters of all path are valid).

- size of letters in remaining words (without string 4 and 7) is no valid. (does not have meaning), of course if we are mounting Windows folders.

- login and password, are not required if computer from which we mount do not want it. If we are not required password and login we can write anything.

### examples of using cifsmount

Here several of examples. Attention: we want to mount folder GAMES from computer called TOM from Microsoft Network (msnetworks). We know that number IP. of computer called TOM is 192.168.2.44. And we want to mount this folder to our folder 'files' on our Desktop under BeOS. Examples:

cifsmount -I 192.168.2.44 -W GROUP -d \\\\TOM\\GAMES login passwd /boot/home/Desktop/files - it is all right.

cifsmount -I 192.168.2.44 -W noValid -d \\\\tOm\\GAMES noValid noValid /boot/home/Desktop/files - also is ok. Name 'tOm' cannot be example 'tommy' but it is no valid which fonts (small or big) we will use.

cifsmount -I 192.168.2.44 -W noValid -d \\\\ToMMY\\GAMES noValidLogin notValidPasswd /boot/home/Desktop/files - now is badly, computer name must be 'TOM' or 'tom' or 'tOm', but cannot be 'TOMMY'. You understand...?

cifsmount -I 192.168.2.44 -W noValid -d \\\\tom\\Games noValidLogin notValidPasswd /boot/home/Desktop/files - no good !, Windows shared folder 'Games' must be written 'GAMES', always big letters !

Now this same user gave us the password 'Elena' to read his folder Games. Examples:

cifsmount -I 192.168.2.44 -W GROUP -d \\\\TOM\\GAMES noValid Elena /boot/home/Desktop/files - it is ok, not important which one fonts we will write'elena'.

cifsmount -I 192.168.2.44 -W noValid -d \\\\tom\\GAMES noValid eLeNa /boot/home/Desktop/files - also ok.

cifsmount -I 192.168.2.44 -W xXx -d \\\\ToM\\GAMES xXx eLenA /boot/home/Desktop/files - also good!

cifsmount -I 192.168.2.44 -W xXx -d \\\\ToM\\GAMES xXx susan /boot/home/Desktop/files -Bad password

### other possibilities

There are other possibilities of usage this command, but exists one condition:

In folder /boot/beos/etc/ in file hosts, we must write numbers IP known hosts and names of computers in our nets( will suffice only this of computer, which we assemble).

Part of file /boot/beos/etc/hosts can look like this:

192. 168. 2. 22 tom

192. 168. 2. 34 george

217. 117. 128. 4 elena

Now our system know that Tom's IP number is 192.168.2.22.

Now we can mount folders in this way:

cifsmount \\\\TOM\\GAMES login passwd /boot/home/Desktop/files

not easier ?!

But how cifsmount know where is workgroup name and IP number ?

Cifsmount take IP from /boot/beos/etc/hosts .And workgroup name take from /boot/home/config/settings/network. But workgroup name is no valid in our chance.

We can also use script ezmount(it use cifsmount).

ezmount 192.168.2.44 GAMES login/pwd

But if we use ezmount we must know one thing. Ezmount is always mounting to folder '/ezmount' in main catalogue.

### how to unmount folder?

There is command 'unmount' if we want to unmount folder...

After end of work with given folder it is good to unmount this.

Syntax this commands looks as follows:

unmount path_of_folder

Example- use of command 'unmount':

unmount /boot/home/Desktop/files

### conclusion

Exist programmes for BeOS simplifying remote assembling of discs, such how CifsMounter, EasyMounter and Be2Win but rather I do not recommend ussage them. Exists also experimental programme World O' Networking(WON)- using da_hood server, at help of which We can freely look through working groups, users and thrown open catalogues, but I met on problems from copying of files. This program is experimental and do not working good yet.

And If you want to using 'cifsmount', best solution is creation of script, which automatically will mount what we want by one command. Example we can do text file called 'mount_tommy'. In this script we can write:

#!/bin/sh

echo I am mounting Tommy Folder......

cifsmount -I 192.168.2.110 -W BEOS -d \\\\TOM\\FILES login passwd /boot/home/Desktop/Tom's_Folder

echo Ready...

Save this file. After, we can write in terminal:

chmod +x mount_tommy [Enter]

Now, we can run our file:

mount_tommy [Enter]

And ready. Tom's folder FILES is on our desktop as Tom's_Folder in which are some Tom's files from network.

*This article was delivered by our friend Adam Szczech from Poland. So please be patient - his English is not so go but I've understood everything and tested this HowTo on my own- it works!*

Important: some BeOS Versions out of the R6 development use a newer version of cifsmount. The parameters are the same, but given in a different order. Type cifsmount without any arguments in your terminal to get a quick description.

# BeKaffe

## This short Tutorial offers an easy entry into the world of Java for BeOS

**With this tutorial, I want to help those of you who want to program with Java using BeOS. For doing this, the Java Compiler "BeKaffe" is needed which you´ll find on www.bebits.com**

### installation

After downloading, you´ll need to unpack the file bekaffeDR.zip into /boot/home. With doing this, the CLASSPATH will be set automatically. From now on you can call "javac" and "java" with the terminal application from everywhere in the system. If you have BeIDE installed (which is recommended), you can download and install the JikesDevKit (approx. 981 kB) as well.

With this, you can choose the plug-in JavaApp or JavaApplet while opening the BeIDE. Syntax highlighting while coding is possible now and the source code is by far easier to read. It´s sad to say that we´re still in the last century concerning the Java version, the last version available is 1.1 from 1999. This in fact sets some limits now and then, but it won´t stop us to write a small program.

### working with BeIDE

We write the source code in BeIDE and save the file with the ".java" ending in a folder of our choice. Next, we start the terminal application and cd to the location where we saved our small script (example: MyJavaProg.java). Now we run "javac MyJavaProg.java". Watch out for the right spelling, small and large letters are not handled the same as in Windows. The script "javac" compiles the actual

source code and changes him into byte code which is common for the Java Virtual Machine (JVM). This JVM is called BeKaffe on BeOS.

Now we can call our program with "java MyJavaProg" and it will be started via the JVM. In /boot/home/bekaffe, you also can start BeKaffe and open your file with "file" and "open". Always watch out that you open a file with a -class ending because other files will cause no effect at all. BeKaffe now should open your program. If you don´t have any file with a -class ending means that the ".java" file wasn´t complied properly.

For short, there are the steps to have your own java program:

- Install the used programs (BeIDE, bekaffe and JikesDevKit)
- Write your source code
- Save your source code with the ".java" suffix
- Open the terminal application, cd to the place of your source code
- Compile your source code with "javac MyJavaProg.java"
- run byte code with "java MyJavaProg"

Now we´re building a small program that generates a window and a button to stop the program. Just open BeIDE and chose JavaApp from the list (see picture on the next page).
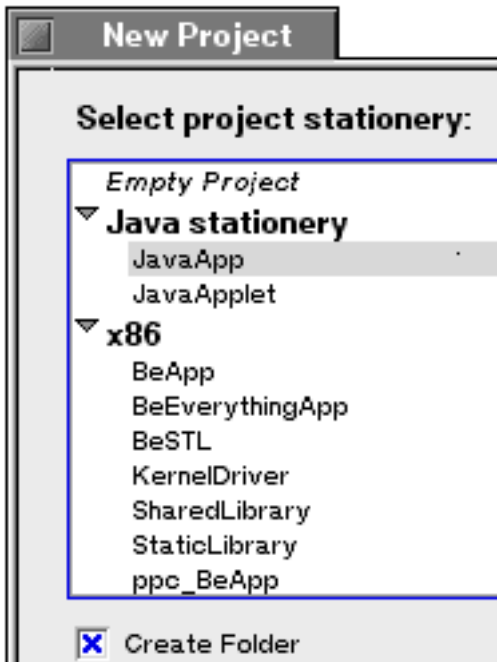
After clicking on "create" we´re called to give the project an name. Let us call it "CloseButton". After we´re saving, a window appears with the name CloseButton.proj included. We´re trashing the already generated ".proj" file under "files" and new file called "CloseButton.java" with "file" and "new text". And please don´t forget to mark "add to project"!



Now we write the source code (it's straight behind this article) into the already generated file. (green text is "just" comments, the source- sample is

included at the end of this article).

With clicking the right mouse button on the files from our CloseButton- Window we can choose "compile". Our project will be compiled now, and if we made a mistake it will show us some errors. Make sure that you`ve mad no typing mistakes. Now we open the file launch.sh and change the line HelloWorldApp into CloseButton. Compile launch.sh and our program will start. Other method to compile and run java apps is to run "javac" and then "java" from the terminal as described before.

After compiling and launching it, our application should look like this:



If you press the close button, the program will quit.
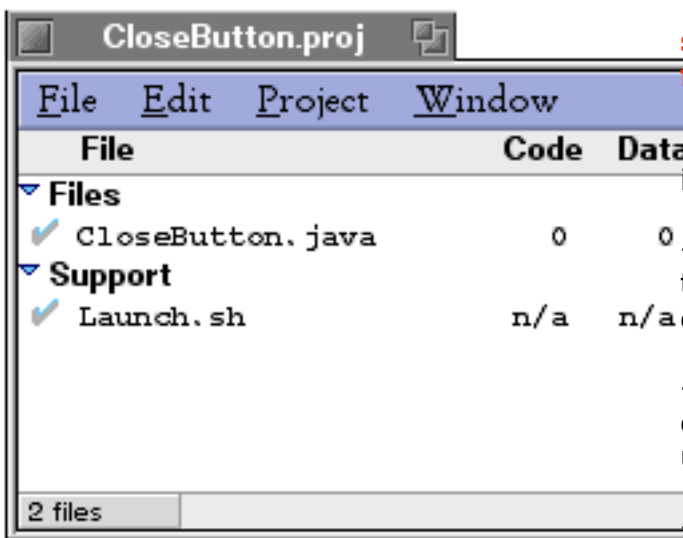
**some explanations**
**for the source code**

// = all text after // will be ignored, so it is useful to write a comment

/* <COMMENT> */ = all text within those "brackets" will be ignored. Can also cross whole paragraphs

- import = like include in C. Tells the compiler which external libraries we will use.

- void = initializes a new function call.

- private/public = defines wherever the function is visible from other files or not. In our simple app you don`t have to care about that.

- public static void main (String [] args) {...} = basic java function "main". Every java app needs one of this. Your source is written between the { and }

## a last short example

The easiest possible java app is this one:

```
class Simple
{
        public static void main (String []
args)
        {
                System.out.println ("Hello
World!");
        }
}
```

Simply type it in a text editor like StyleEdit or Globe (not Gobe!) and save is as a .java file, so Simple.java for example. O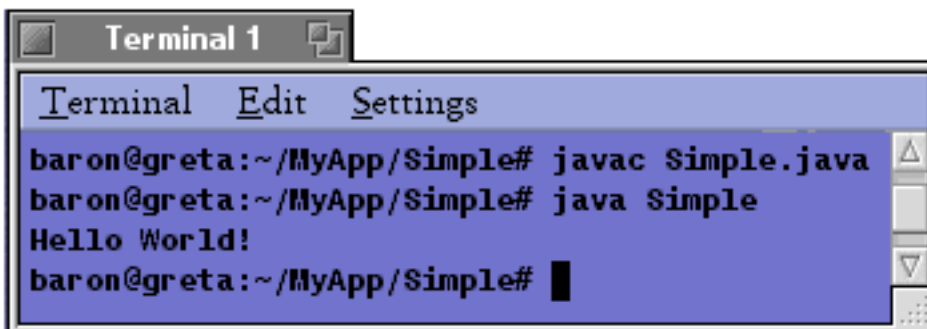nce the file is saved, switch to the directory where the file is located from terminal. Run the java compiler javac and then java itself, like in our example here. After running the program the terminal will write "Hello World!" and then you will be right back with the input cursor.

- cd /home/programming/java
- javac Simple.java
- java Simple
- Hello World!

I hope you get everything and you can use this editorial as a first step into the world of Java.

Florian Thaler

*This article was submitted by Florian Thaler and has been translated by "choulth".*

```java
import java.awt.*;
import java.awt.event.*;

public class CloseButton extends Frame
{
        Frame f = new Frame("CloseButton");
        Button button = new Button("Exit");

        public CloseButton()
        {
                System.out.println ("Starting CloseButton...");
                components();
        }

        public void components ()
        {
                f.setLayout (null);
                f.setBounds (200,200,180,100);
                button.setBounds (40,40,100,40);
                f.setResizable (false);
                f.add(button);
                button.addActionListener ( new ActionListener()
                {
                        public void actionPerformed (ActionEvent aev)
                        {
                                System.exit (0);

                        }
                });

                f.setVisible (true);
        }
}

public static void main (String [] args)
{
        new CloseButton();
}
```

# Tips & Tricks
## Some nice hints to make your life easier

### windows and workspaces

If you want to "unite" 2 different applications (windows) from 2 different workspaces into one workspace without closing one of the apps, simply select one of the apps`title with the mouse, hold it and switch to the workspace in which you want it.
Example: the BeShare Window in Workspace 1 (WS1) should appear with Net+ on WS 2. Simply click on the titlebar oft the BeShare window, hold the mousebutton down and press [alt]+[F2].

### beos radio

This is an internet radio station, moderated by Dane Scott (LeBuzz). He's playing original music created by BeOS Users or friends of BeOS Users. Every hour BeOS Radio sends general and BeOS dedicated news. You can reach them on www.beosradio.com.

Dane runs the system on Zeta (yes, Zeta) with TuneTracker, a radio software that competes with much more expansive equipment. TuneTracker is available from 150 US$ and a BeOS Only Product.

If you want to enjoy this unique radio, you can also use a small app called "Probe for BeOS Radio". You simply select the kind of connection (64k stereo for example) and after a few seconds CL-Amp should appear and play the stream.

### tracker

If you right- click on the desktop, a menu appears, where you can select "New...". Normally it only features "New Folder". But you can create your owntemplate for this menu.

Image that you would have something like a new textfile in this menu. Just open StyleEdit.
Save the empty document within "/boot/home/config/settings/Tracker/ Tracker New Templates/". Save the file es "Textfile" for example.
Now you can create a new empty textfile from the "New..." Menu.
Of course the same procedure can be used with almost any other program.

*These hints were submitted by Florian Thaler and were translated by Matthias Breiter.*

# Welcome Community!

## The Brand new section in our magazine

### the idea

When I was thinking about what can be placed here in Technoids, I was thinking about this section. Because many articles are provided by community members, layout, ideas and translations were also created by the communiy.
The honor you've shown me is now given back to you.

### what's inside?

Well, we have at first letters to the editors. Thanks a lot to Lelldorin who has answered questions from our forum. If you have any ideas or any questions just write an email to technoids@morgentau.org or (because of all the trouble I had) to technoids@web.de.

Next thing we have is community advertising. This means if you're running a community page or a free software "company" or a BUG (BeOS User Group) you can have a small advertising for free. Please contact us if you want one. An example is printed on the next page.

### things to come

What we want to implement (and that will be in the 3rd issue) is "Community Status". Think of projects like OpenBeOS or COSMOE or OpenOffice. We are asking the developers how far the development has grown and if there are any vital news. Also small interviews for this purpose will be included.

And what we want from you. If you`re running a BUG or website you have the chance to write something about that here. Just write some imformation, like members or user meetings or if you have barbeque or something like that. If you don't have a BUG you can also write about BeOS in your country. We got many requests in the past "Hey, what's going on out in the world. We all know about BeOS in our country- but we also want to know about BeOS in foreign contries."
And even if you are only a single BeOS user you can send in a personal BeOS-Experience report. I mean all the things you do with BeOS.

### private trading

I was thinking abou some private trades within here (like almost every magazine offers). But then we would hav the following problem: we have German ones, Austrian ones, Swiss ones, English ones, US ones ...
So if you are interested in buy or sell things for personal purposes we could create a part for that in our Forum.

### an unique place for you

As you can see, we want to provide an unique section for all flavors of the community.
What we need know is your feedback, you comments and your experience reports. Feel free to contact us at technoids@morgentau.org.

Best Regards,

Matthias Breiter

# Letters to the Editors
## Taken from our forum and answered by Lelldorin

**Programming C++ with BeIDE**
Technoids Forum - June 21st 2003

**michael lotz** Some of the statements are not correct at all. For example not every programming language recognises small letters and capitals like BASIC or PASCAL.

**technoids** While writing the article I've already known that this statement will provoke feedback. In deed, my text is not correct. With this assertion I wanted to show an important fact of UNIX and UNIX- Like Systems. On the other hand it can help you to keep your source code more understandable. And, of course, it helps you also when you switch to other languages. By the way, in C++ it is important to remind the correct spelling.

**PDF or HTML**
Technoids Forum - June 21st 2003

**jürgen** I don't know if PDF is the correct answer to the topic of visualisations (its' driver). You can see by yourself that the source code is not readable with ease- I didn't have fun by trying the examples by myself. Why don't you use HTML instead? It's also cross platform but creates less problems.
- Jürgen

**technoids** We've chosen PDF cause this delivers a good quality for the included text and graphics. On the other hand it also features a bit more security because it's not so easy to create a faked Technoids. Technoids is a free magazine of course but the editors and the writers put a lot of work in the articles and the presentation.
We feel sorry that some parts were not easy to read. One aspect is the BeOS PDF- Writer which can't compete with more professional solutions like Adobe Acrobat. Another aspect is that we've chosen the wrong font and we will fix that in the future.

*Answered by Lelldorin (Christian Albrecht), translated by Matthias Breiter*

### in the nex issue

- Project
  Accelerating your old PC with
  just an Operating System ;)

- Audiosoftware
  Holger Wendenburg give us a look at
  different BeOS Audio Applications

- Thoughts
  Have Computers anything to deal with
  love ?

- Games
  We will test a game or two

And a lot of more (these were only the
boring ones ;)

### your article in technoids ?

if you want to submit any kind of article,
please contact us at
technoids@morgentau.org

### next Technoids (No. 3) will be available around Octobre 1st, 2003

## TECHNOIDS
### The free BeOS magazine

**Impressum**

**Editor in Chief**
Matthias Breiter

**Team Members**
Florain Thaler
Christian Albrecht

**Layout**
Joao Carvalho

**Guest - Authors in this issue**
Holger Wendenburg
Adam Szczech

**Translations**
Eddy Zeeder
"choulth"
Matthias Breiter

**Proof Reading**
Jon Pimble
Tony Paul
Jared Eldredge

**International Translations**
Remi Grumeau
Thanassis Anastassiu
Adam Szczech

**contact information**
Matthias Breiter - Technoids -
Zollernstrasse 24
D- 86154 Augsburg
www.technoids.tk
technoids@morgentau.org

**Special Thanks**
Axel Dörfler
Be Inc. for BeOS

### www.technoids.tk
made with BeOS