Presenting eti

- How i learned to stop worrying and created megawidgets -



Axel Nagelschmidt http://axn.dyndns.org



Modula-2

How it all began ...

Started programming on a calculator TI-59

Built the Sinclair ZX-80 from a soldering kit after a holiday in England

Advanced to ZX-81, ZX Spectrum, Atari ST 520, Macintosh SE

Studied Computer Science and Mathematics in Erlangen

Worked for several institutes and clinical departments during years of study, designed PC Card for measuring tasks with DMA Started business to assemble PCs according to customer requirements

Used CP/M, GEM, DOS, Windows, Mac OS, Minix, IRIX, Solaris, Linux, Mac OS X and other systems

Together with a friend founded a computer company positioned in trainings and consulting, mainly on java and OO techniques Preparing course materials, administrating Solaris and Linux machines for classes and developers, first contact to TCL

Working since 9 years in a company for biomedical technology, as developer, database developer and Solaris administrator

Today running Ubuntu on PCs, Mac OS X on several Macs (PPC), Solaris on some SUNs, being forced to use Win-XP at work

Until now met John Mc Carthy, Niklaus Wirth, Douglas Adams (SIGGRAPH Keynote in New Orleans 1996), happy to meet more experts here !

Writing a complete editor is possible after one month of learning TCL!

I wrote something like this around the year 1997:

 $\Theta \Theta \Theta$ Wish text .t pack .t entry .f button .o -text Open -command {.t delete 0.0 end; set f [open [.f get]]; .t inse rt end [read \$f]; close \$f} button .s -text Save -command {set f [open [.f get] w]; puts \$f [.t get 0.0 endlcl; close \$f} button .q -text Quit -command exit pack .f .o .s .q eti_0.1.tcl Open Save Quit 1

eti 0.1, nearly feature complete

Even better version 0.3:

Has menubar (not shown here, screenshot from Mac)

Options to save / save as / open / close file, toolbar, find and replace

Options to change font family / size / background colour

Status bar with status info, cursor position and clock

Can mail it's content

Ported to Solaris (because no change necessary)

No widespread use, probably because never deployed

000	et3.tcl	
ē 💾		
Find:	Replace:	
	edittool 3 v 0.5 running under TCL 8.4.12. Copyright AxN 1998-2004 Not for distribution. OK	
	Program edittool 3 v 0.5 loaded Line 1, Column 0 20.05.2006, 14:02	1

Started database programming with TCL and oratcl, network programming cross platform

Found TCL be able to connect to a database very easily

Created an online transaction processing application, processing tens of thousands messages per day from cardiac pacemakers

Needed to write a multiplexing front end to an SMS sender/receiver that allowed only one connect from a client

Added monitoring tools to watch processes, checks for availability of hosts, DB listeners, Webservers, disk space, load ... Running several instances of tclhttpd on production and test servers to watch processing and resources, lightweight, easy to use

Using tequila to synchronize processes and have a look at process figures from several clients

Also created more and more tools for me and collegues

- simple application
- single window
- some buttons and entries
- database connection
- monitoring tools
- text transformation
- database repair tasks
- XML parsing and creation of TCL and SQL scripts from description of data modules

Each of these sharing many properties with other tools, having it's own method for persistant storage like .files, ini-files etc.

A tool is born ...

Collected these into a common tabnotebook, without any interaction between the modules yet:

Is	ool beastle Samstag	15.09.2007 15:3	1:16, Wuche 37	- I I X
File DB3 TCL St	tyle			Help
Actions Ampel	Calendor Console Los	d Log Ping	Projects Server Sha	ared Systems Wiki
	gnome28 Webserver FAIL 15.09.2007 15:31:20	gnome28 Docserv not tested	ver gnome28 toolserv not tested	er
	hmer Webserver OX 15.09.2007 15:31:28	Dator Doescryer FA 15.09.2007 15:31:3	IL Jamer toolserver 28 not tested	
	mini Webserver not tested	mini Docserver not tested	mini toolserver not tested	-
	ultra5 Webserver not testud	ultra5 Docserver not tested	ultra5 toolserver not tosted	
	ips Webserver not tested	ipx Docserver not tested	ipx toolserver net tested	
	kS Webserver not tested	k5 Docserver not tested	kS toolserver not tested	
	kő Webserver not tested	k6 Docserver not tested	kő toolserver not tested	
	indy Webserver not tested	indy Decserver not tested	indy toolsorver not tested	
	next Webserver not tested	ness Docserver not tested	next toolserver not tested	
	g4 Webserver not tested	g4 Docserver not tested	g4 toolserver not tested	
	30b Webserver OX 18.09.2007 15:31:33	wh Docserver FAIL 15.09.2007 15:31:35	xuh toolserver FAIL 15.09.2007 15:31:36	
	bløde Webserver not tested	blade Docserver not tested	blade toolserver not tested	

Usage of object orientation seemed to be a good idea:

◊ Have a mini editor for any kind of text. Derive HTML editor, TCL editor etc. from this.

♦ Have a set of classes needed for graphic programs, derive different graphical drawings / simulations from this.

◊ Specify a set of common used actions to include into the menu or a toolbar

♦ Have multiple instances of the same type, so can use a notebook with several tabs

◊ incr TCl offering classes and incr TK offering megawidgets like notebook seemed natural

But (1) ...

◇ incr TK seemed to be more complicated than necessary

◊ incr TK seemed to be slow because of code overhead

Solution I:

- \diamond create one superclass drawing the toolbar, the notebook and the status bar
- ◊ create one superclass in incr TCL (not TK!) for all instances (called etimodules)
- ◊ have each etimodul draw it's own frame
- ◊ inherit superclass etimodul to take care of common needed methods

But (2) ...

- creating multiple windows with notebooks needed another superclass
- inheritance for widgets was not really used
 - tcledit or diffedit could not really inherit the class miniedit, different count and layout of widgets
- \diamond learned about tile and increasing / ongoing support for Mac OS X

Solution II:

- ◇ recreated some widgets using tile components
- ◊ added menu to change style to any existing tile theme
- ◇ some usage of mkWidgets, because better creation of megawidgets was anticipated

But (3) ...

- \diamond mixing iwidgets with tile widgets was no good idea
- MkWidgets usable for megawidgets, but not really an OO system
 A system
- MkWidgets not being supported any longer?
- ◇ not much time to try something new

Then ...

- ◇ visited the 5th european TCL User meeting (and organized the 6th one)
- ◊ got more involved using tequila, thus making shared apps an easy possibility
- ◊ learned about integrating OpenGL to create apps using 3D graphics
- \diamond learned more about tile, the notebook api has some differences
- ◊ learned more on creating fullfeatured apps
- ◊ learned more on mixing text and graphics to create integrated learning experiences
- ◊ learned more on tablelist and it's integration into tile

Other influences happening in this time:

- \diamond More usage of Java and especially Eclipse at work
- ◊ Our company introducing Lotus Notes, me visiting a developers course
- ◊ Seeing that TCL can control much of windows through twapi
- \diamond .NET and mono appearing in widespread use, new IDEs anjuta and monodevelop
- ◊ Applications becoming more complex, each one developing their (own) makro language
- ◊ Seeing that TCL can integrate calls to graphic libraries like ImageMagic or OpenGL
- Our of Understanding that integration in big consistent frameworks like Oberon or Squeak makes usage much easier
- ♦ Anticipating to learn that TCL can integrate calls to control OpenOffice.org
- Learning about snit and it's ongoing support and integration with TCL 8.5
- ◊ Writing more applications as helpers or for database access, needing a usable and robust framework to do so
- ◊ Expecting user interfaces to allow users much more choice and flexibility "graphical programming"
- ◊ Not wanting to learn a new P*-language every year
- ◊ Seeing that TCL is here to stay, easy to use, easy to teach or use as description language, cross platform

=> Decided to base framework and apps on TCL, integrating good usage examples into an usable IDE

(OK, did so before with Pascal, Oberon and Delphi ...)

Moving from incr TCL to snit

Торіс	incr TCL	snit
Header	itcl::class <name></name>	snit::widget <name></name>
Inheritance vs. Delegation	inherit <upperclass></upperclass>	delegate <method> to <otherclass></otherclass></method>
Building widgets	frame \$w, delivered as argument	set w \$win, frame already created
Reference to instance	\$this	\$self
Calling own methods	<method></method>	\$self <method></method>
declaring method	method <name> <args> <body></body></args></name>	method <name> <args> <body></body></args></name>
Destroying instance	itcl::class delete object \$this	destroy \$self
Overwrite class definition	no, only after deletion of class	yes, like the TCL way
Inherit methods	yes	no, but delegation helps

Rewrote about 5 major classes and the core of the framework in about 3 days.

Need to rewrite toolbar, and some handy megawidgets used from mkWidgets to snit.

First results seem to be very promising.

Skeleton of an etimodule:

```
snit::widget somepage {
```

```
typecomponent super
delegate method * to super
typeconstructor {set super [etimodul %AUTO%]}
variable w
constructor args {
     # create the modules widgets
      # initialize variables
method start {} {
      # perform actions, start timers etc.
}
method onshow {} {
      # perform actions when module is shown again
}
method onhide {} {
      # perform actions when module is hidden
}
```

```
method f9 \{\} {
      # refresh, reload actions when requested to do so
}
method clone {} {
      # create a clone of the same type of module
}
method close {} {
      # stop timers, free resources, close this module
}
method whatever {} {
      # do something
}
# .....
```

}

All methods not implemented in a specific etimodule are called in the super class by delegation.

Some of them are empty, others do the easy and safe default action, e.g. clone, aboutme

A digression on network and monitoring:

Needed a small TCP/IP connected client / server tool for cross platform monitoring and maintenance applications

Synchronous reliable remote calls to hosts or services that are down lead to sometimes intolerable delays in client tools

Until now we could live with tequila for process synchronization, carefully dividing into modules which may encounter delays and those that need short response times

Some problems remain, will explore more about this later. Maybe UDP can help?

Also using tclhttpd modules as application server, after integrating SQLite database into the webserver.

 bloserver 0.1 	
Life	
Letelved 6 Bytes on sock sock?: "date" (sereved 6 Dytes on sock sock?: "bain"	Server Status
cepeived 6 Býteo oz bock back7: "pinĝ"	
Status of socket-server is: serv(socspts) = 3	Users
<pre>cerv(lastbaye) = 14 09.2007 22:25:24 serv(port) = 12937</pre>	3
<pre>serv(firststart) = 11.08.2007 18:19:51 serv(tankn) = 3</pre>	Connects
Serv(minis) = 8 serv(runs) = 40	3
<pre>Serv(os) = LIMIX 8.6.81-16-generic serv(users) = sock5 sock5 sock7 </pre>	Tasks
perv(status) = 0 nerv(thistory) = 2 nerv(thistory) = 2	5
<pre>berv(slres) = 144 secu(slres) = 15 02 0007 10.17 10</pre>	
<pre>perv(lister) = sector serv(lister) = sector</pre>	
received 4 Bytes on such suchT: show (core yed) Dites of such such, such, "stat"	
	Guit

Biter command:	Biter command:	Biter command:		biodient b.1 connected to localbost	
Biter command: 2011	Biter command:	Biter command:	Notechi (Fourbal) (Type 'nell Servia (Cluber) (Cl	o bisserviz (11 ku serviz bizstne, tsrt 1180) iz 1 out of U o to git a list of allevin communis) seizhis off (15:31:17	
Biter command: Seni fi	Biter command: Seni 1	Biter command:			
	localnost 🚽	Disconnect			_
Disconnect	Exists data		Biter command:	liccirox: v Discament Extributes	3cmi 1
Disconnect Fefcir data	Fetch stata		Biter command:	lice al root	3cml 1
Disconnect Fetch riata randori 🖉	Fetch stata. To record	ro 'dzrı 👻	Biter command:	lice II rost v Disconnect Feich Nata rolidera v	3cm 1

Webclient with SQLite database should make deployment of application and upgrades very easy ...

#!/usr/local/bin/tclsh

```
# idea from my wiki 147, thanks to DKF
# stripped down, no interp, working and tested
```

package require Tk package require http

```
proc http_source url {
    set token [http::geturl $url]
    set script [http::data $token]
    http::cleanup $token
    eval $script
}
```

http_source http://deployment.local/tclapps/initapp.tcl

(Remember the first version? 7 lines of code is all that is needed, the possibilities are endless ...)

Upgrading and migrating through tests is a reliable way ...

Also a good way to document bugs and show when they are fixed

Itcltest::test achterbahn test2 -body {math3::achterbahn 39} -result 34
Itcltest::test fibonacci test1 -body {math3::fibonacci 14} -result 377
Itcltest::test fibonacci test2 -body {math3::fibonacci 19} -result 4181
Itcltest::test roman test1 -body {math3::roman 25} -result XXV

Lately had to follow strict QM standard procedures, parts of TCL application now certified for medical usage in clinical studies

Requirement No.	Description of requirement
REQ-628-003.012	The new module shall be called routines3 and conform to the standard calls for etimodules. It shall be placed in the etilib2 module library.
REQ-628-003.016	The module shall have a method demo to start a small demo showing it's features.
REQ-628-003.018	The module shall have a method to reliably predict the weather for the next 20 days.

Test-driven development helps to structure requirements into code and to prove that all usecases have been considered:

It cltest::test routines3 exists -body {file exists \$tclpath/etilib2/routines3.tcl} -result 1
It cltest::test routines3 demo -body {routines3 .a; expr {[lsearch [.a info methods] demo] >= 0}} result 1
It cltest::test routines3 test2 -body {# not so easy to write the test} -result {# the programmer

will implement whatever he thinks is correct}

Examples of etimodules:

Startpage, allowing to start other modules, showing most used documents, todo and calendar Older framework for helper applications (terminal, consoles for sqlite and oratcl) shown below

000		Startpage		
÷ 🗙 i 🥶				
Startpage \ Termin 15 \ diffsync \ buyestch \ miniedit.e	ti (buywaich.sti)			
Welcome to eti 0.4 and TCL! voor last usage of eti was at 11.05.3006 21:57:55, this is 20 hours 49 minutes 56 seconds appill since the first start on 15.09.2000 15:26:43 you have started eti 543 times. Choose a module, most used item or a todo item: this cat't happen!	Modules at 3 bench cal calc dame db2 debug demo diffecit diffsync filebox filebox filebox filebox filebox filebox filebox filebox filebox filevew gif graphobj htmMaw lindenmeyer linpage macpage minied t myora nac projtool skel sopage starustool	Most used	Todo	Calendar
-Console Debug Find Log Oratel Shell S	OLite			
<pre>rale=prode Allocativ reallocativelocativession_id reallocative > allocativening_id idi</pre>	runiq_id reallor3:rkey			
TCL-Command:				

tcledit, editor with syntax colouring based on ctext Allows working in projects and libraries, highlights requirements and proc / method headers

<pre>Surpage Termin 5% diffyre { hypertech i miniadizat } suywatch i miniadizat } suywatch i miniadizat ; suywatch i miniadiza</pre>	000	tcledit: "/Users/axn/tcl/etilib/buyw	atch.eti*
Surjage Termin 15 dffsync buywatch ministias buywatch]] set type Duywatch]] set type Duywatch]] set type Duywatch]] set type Duywatch]] set the list (alloc3::uniq_id)']]]] 	48	1 N N 1 X 1 🔿 1 🖌	
<pre>4 33</pre>	Startpag	e \ Termin 15 \ diffsync \ buywatch \ miniedit.eti \ buywatch.eti \	
TCL-Command:	 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 465 411ce 141 	<pre>set type buywatch set file 'Untitled [alloc3::uniq_id]' pack ['mark \$w] -fill both -expand yes puck ['markbeckground #fifTbb] Sw : insert end 'muywatch, add auction number or UKL to watch' set the list \$::env/(HOME)/doc/buywatch load_list tab_select } destructor { debugsay 'destructor buywatch' } method loadlist {} { foreach line [aplit [file3::textfromfile \$the_list]] \n] { file foreach line [aplit [file3::textfromfile \$the_list]] \n] { foreach line [aplit [file3::textfromfile \$the_list]] \n] { for a seve is [aplit [file3::textfromfile \$the_list]] \n] { for a seve is [aplit [file3::textfromfile \$the_list]] \n] { for a seve array to file1!! } nethod onlide {} { for use default toolbar or examples below!!! Debug Find log Oracl Shell SQLite proce alloc3::session_id :selloc3::runiq_id :selloc3::text desumiq_id</pre>	3:# note 7:# legi 13:# requ 16:# paste-button!!! requ 16:# paste-button!!! requ 18:# entry for bid, button for refresh, remove!!! 19:# show #, seller, sofort-kauf too!!! 21:#
	TCL-Com	mand:	

nac, (not another commander), because this is my main productivity tool Using tablelist, but need more intelligent bindings for keyboard operation and cleaning of layout

				/Users/axn		
Name	Size	Date	Attr	* Name	Size	Date Attr
gplications	<112 k	12.15.2006 17:05:50	Attributes	1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.	<0105	29.10.2005 19:45:56 Attra
plictics (the CS S)	(118)	(B.10.2005 14:07:59)	85.tributes	10	(IIII)	17.01.2006 23.18:05 Attes
atos oust	(3.18)	25.11.2008 18:54:45	Monral Stokes	bin.	(115)	05.05.2006 BL: 19:47 Aster
10 7 5 6 7	172	10.10.0000.00.41.50	Attributes Attributes	his 22.5 2005	(013)	20.05.2005.16:57:50 WHEE
edeten Fulder	1118.	E1 18 2008 19 50 14	Alsher holes	hin in to some later	(113)	12 05 5005 28 25 45 ALL
er.	(122)	12, 15, 2006 18: 45: 55	Attributes	0	(113)	22.01.2005 15:01:03 Attr
eveloper	<112>	12.11.2005 16:12:52	Attributes	cal	(000)>	20.04.2006 17:29:50 Atter
okaeente-	(TIR)	18.18.2008.20:39:20	Attributes	conftal	(0.03)	14.05.2006 11:24:52 Attai
to	(318)	15.14.2006 10:40:84	ShiriDobas	da.su	(1035)	01.05.2006 12:37:22 Aster
ibrary	<120>	23.11.2006 23:10:13	<u>Attributes</u>	451	<0005	24.03.2005 10:06:37 Atter
a.e.	(112)	61.11.2005 16:40:55	Attributes	defs	4005	09.00.0005 01:22:42 Attri
etwork	118	14.15.2006 09:28:48	853ribites	Desktoy	000	14.15.2006 UN 38.43 ATC/
rivete Ma	1100	18,10,2006,10,40,07	No1r1300es	Dies Diese Deux Delder	(105)	10.07.0000 10.00 01.06 Miles
IID.	172	16 14 2006 12:40:07	Attributes Attributes	dee	(0.00)	14 05 2005 09:49:35 Attac
e staan en troee	11184	IR 10 2005 15 56 58	Altri hulasi	Documents	(IIII)	14 DR 2006 19 25 50 AUE
anograry Itawa	(122)	02, 15, 2004 20: 59: 00	Attributes	earter	(III)	27.03.2005 22:19:53 Attr
heVolumeSettingsTolder	<112>	20.12.2003 17:06:50	Attributes	easter3	(010)>	01.04.2005 21:50:12 Attr/
ab	(11R)	14.15.2006 09:51:06	Attributes	caster nech	(DIR)	21.02.2005 21:29:55 Attac
rash.	(318)	01.08.2008 20:19:28	66-hr dirobera	wite	(1135)	23.04.2006 15:19:43 Atter
pisters	<322>	10.14.2004 17:40:50	Attributes	etc	<dd5< td=""><td>10.05.2006 19:10:25 Atter</td></dd5<>	10.05.2006 19:10:25 Atter
ie eone	<1128>	20.10.2005 10:45:56	Attributes	htal	<0.020>	10.01.2006 21:16:50 Attri
SF	(118)	24.15.2005 28:15:28	85 tributes	10005_for_tol	d100	03.02.2006 33:04:24 Attes
9F	(319)	12.15.2006 18:45:57	Motributes	1478	(115)	13.00.2000 01:46:43 Attes
0100000 12402 41 4 4000	(112)	13.15.2006 10:55:25	ACTF13CC+8	layoot_econgle	(DDS	23.04.2000 14:24:45 Attes
halling IR	100 CT (100 CT	11.11.11.1106 34:46:20 15 14 9005 90.49 02	ALL CARDEN		1110	221.02.2006 21.44:49 Webra
kesktop IS kesktop IF	1250220	14 13 0006 19:02:05	Robrid Sectors	ratiab	(113)	22.05.2000 01:16:00 Attes
ach	500050	12.15.2006 10:45:57	Attributes	1000	(111)	17.04.2005 12:40:40 Attr
anh ann	500052	10 85 9006 10 45 50	State Control	7 House	4172	05 04 0005 01 57 42 2+ker
2 files in dir /				75 files in dir /Users/axn		
cha	nge dir	(info) sh	ow (miniedit) (cop	w)	(r	nv/ren (mkdir) delete
leftdir/right	dir Command					
Console Debug Find Log	Oratel Shell	SQLite				
> into proce allocativ	assion id ++allor	Second id scalloc3ecker				
seal loote widness seal better						

Using a database in the application, using the application in the database ...

Many possibilities with metakit, starkits etc. exist

Coming from Oracle, i found SQLite the most easy method to use a lightweight though very powerful database in my apps

Already had a personal wiki inside of eti, using structure of this to migrate the project into the database, shown in demo now

sqlite>
sqlite> .tables
addr cal code prefs todo wiki
sqlite> .schema wiki
CREATE TABLE wiki (id id, item string, text string, first date, count integer, last date);
sqlite> .schema code
CREATE TABLE code (id id, type string, value string, name string, first date, count number, last date);
sqlite> select distinct type from code;
app lib module
sqlite>

Still using cvs in filesystem, but will move versioning into database soon, have code archive on other servers in net.

There is still more to come and more todo ...

TCL 8.5 should be ready soon (or may be it is already?)

Check tcl modules (see TIP #189) and convert my libraries to new format

API of etimodules has to be frozen at some point for deployment

Update webserver and have code repository there

Solve some problems with timeout waits on TCP/IP

Have a single application framework for easier deployment of apps, perhaps using freewrap?

Follow development of Eclipse, Lotus and the Komodo Project ...

info, to show information about the application Thanks again, this is really a tool made possible by the community!

🜵 🕙 🕌 🔚 🗶 🚺 (Merge Wiki in) (Make de	defpages Back Forward Delete Rename Find	
	Thanks to the TCL community	1
Startpage \ Termin 15 \ diffsync \ buywatch \ miniedit.eti \ buywatch.eti	ti This program would not have been possible without suggestions and ideas from the TCL community	
	Manufactor for the state of a large second state of a flat the state of a second state of a state o	
A	Many thanks for inventing the language, creating complete distributions and many wonderful packages and extension	ns to:
CL/TK: 8.4.12/8.4.12 threaded, System encoding: utf-8		
C TCL-lb: (Library/Frameworks/Tcl framework/Versions/8.4/Resources/Scripts TK-lb: (Library/Frameworks/Tc.framework/Versions/8.4/Resources/Scripts)	Alexios Andreas Arjen Artur Brian Bryan Cameron Csaba David Dennis Detlef Don Donal George Georgios Holger Jean-Claude Jeff Jeffrey John Jörg Jos Larry Laurent Mac Michael Paul Richard Rolf Steve Todd William	
C lib-path: d Albrary/Frameworks/Tk.framework/Versions/8.4/Resources/Wish.app/Contents/lib/d8.4 /L/brary/Frameworks/Tk.framework/Versions/8.4/Resources/Wish.app/lib/d8.4 d Albrary/Frameworks/Tk.framework/Versions/8.4/Resources/Wish.app/cl/brary /L/brary/Frameworks/Tk.framework/Versions/8.4/Resources/Wish.app/cl/brary d.ibrary/Frameworks/Tk.framework/Versions/8.4/Resources/Wish.app/cl/brary d.ibrary/Frameworks/Tk.framework/Versions/8.4/Resources/Wish.app/cl/brary	Many thanks also to you (unknown TCL author), though i forgot to mention you! Thanks to you, for using this program!	
4. Dorary/Frameworks/Tk.framework/Versions/8.4/Resources/tel8.4.12/library Library/Frameworks/Tcl.framework/Versions/8.4/Resources/Scripts		
d Libraries	OK	
D ActiveTcl	O O O Machine info	TT.
d addressbook 1.1	Eostname luxor	11.
e aes 1.0.0	Machine Power Macintosh	1.0
e anoco 0.1	Operating System Darwin	1.0
ascenc 0.11	Version 8.6.0	1.0
e asn 0.4	Systemtype unix	1.8
e autoproxy 1.2.1	syteorger bigendian	11.
e autoscroll 1.0 1.1	Displaysystem aqua Recolution 1440x800	18.0
e base64 2.3 2.3.2 2.3.1	Colordepth 24	1.8
e Dee 0.1		1.0
pench 0.1	x0	1.8
bench:putticsy 0.1		1.8
bench::out::text 0.1		
bgexec 3.0		
bibtex 0.5		
binhex3 0.1	Loc3::key	2
blowfish 1.0.0 0.11		
BL1 3.0		-
<u></u>		- Noch 3