GNUstep Meetup 2015 notes

[Day 1](#_rpwrj68glbj7)

[Fred's Story](#_w2fryv3its2m)

[State of GNUstep](#_z6xhklf3qm3x)

[Future of GCC's Objective-C support (and our support for it)](#_esfofon5zfdm)

[gnustep-make, libobjc2 and new features in GNUstep](#_hegv31wgm9h9)

[Future of Make system and alternatives](#_1fzggtgfbtu8)

[Garbage collection](#_o9nmmii4bzc3)

[Criteria for new release](#_pu3rk1owquzh)

[GUI Backends](#_dre9kqgqdh1z)

[Dropping non-Cairo Backends](#_pgzzw0yag3ks)

[Opal backend](#_o54hjsslqv1p)

[Wayland](#_ihuirr5pry68)

[Git, Gerrit and other tools](#_o46jj6id0jb6)

[Packaging](#_i6koxjf1ldhe)

[OpenCSW (rm)](#_61cht2jp41io)

[OpenSUSE (fred)](#_iuiqmzrb7lka)

[Debian/Ubuntu (ivucica)](#_rrxpl2d7eou4)

[David's FreeBSD ports](#_s1p6de33b8jh)

[Do we compile xib files?](#_1ceb4wt4rfwo)

# Day 1

10:00

[fred] Let's specify schedule.

[fred] Let's discuss future of gcc's objective-c support.

[fred] Let's discuss state of GNUstep.

[fred] Let's tell a story. And other people's stories on why they joined and use GNUstep!

[rfm] I'll look at the changelog to see how long ago I joined.

[fred] Do you trust the changelog? :-)

[rfm] I first contributed in September 1996 - 19 years!

[rm] Let's discuss the make system

[rfm] Let's discuss libobjc2 and all the new features.

[fred] Let's discuss dropping all backends except cairo!

[rm] Disagree!

[rfm] and dropping garbage collection

[fred] Let's also discuss opal backend!

[ivucica] Git, Gerrit, automated test on push and even on commit.

[rfm] New release should be on agenda.

[fred] We should not necessarily make a release based just on bugs, they make us visible. I'd like not to ignore patches that came with PikoPixel, but they are complicated.

[fred] Let's talk about PikoPixel, looking at code and talking about motivation of someone who creates all the GNUstep-specific patches but does not contact the mailing list?

[rm] Let's talk about who are our target users? And priorities? Is the target user the Ubuntu user, the Mac developer, etc?

[fred] Still, everyone will work on things everyone is interested in. Even if we specify target user.

[rm] I'd like to show demo of a "cool GNUstep application"!

[fred] Window manager indirection, borders, stacking order and visibility.

[everyone] PikoPixel is doing something interesting with tooltips, let's explore.

[rfm] We should fix tooltips to support multiple lines/paragraphs and limit their width

[rm] PikoPixel Preview nib windowStyle broken

[rm] Let's discuss theming and new features I need.

[rm] I want better sliders (behavior and themes)

* **Fred's Story**
* **State of GNUstep**
* **Future of GCC's Objective-C support**
* **gnustep-make, libobjc2 and new features in GNUstep**
* **Future of Make system and alternatives**
* **Dropping garbage collection support**
* **Criteria for New Release**
* **GUI Backends**
  + **Dropping non-Cairo Backends**
  + **Opal backend**
  + **Wayland**
* **Git, Gerrit and other tools**
* **Packaging**
  + **OpenSUSE (fred)**
  + **Debian/Ubuntu (ivucica)**
  + **David's FreeBSD ports**
* Who is our target user?
* Riccardo's "Cool GNUstep Application"
* Window manager indirection, borders, stacking order and visibility
* PikoPixel and related patches
  + PikoPixel tooltips
  + PikoPixel Preview nib windowStyle
* Theming and requests for new features
  + Slider improvements
* Filip's experiences: GSoC and new user experiences

## Fred's Story

[fred] Here's a GNUstep for Windows CD from September 1996 I ordered from Scott Christley for $30! Scott could never cash the check, so -- who owns the CD? :-)

[everyone] The old code! Wow!

## State of GNUstep

[fred] There's only five people here. It reflects the situation of GNUstep, judging from mailing list and commits. GNUstep should be publicly visible, so mailing list and website are important, but what counts for me is code (and commits). Looking at this year's changes, there's changes from Richard, then me, and not much more.

[rfm] Most of the changes have not been really GNUstep -- mainly peripheral libraries and such. Very dispiriting. But we've been saying it over the decade.

[fred] But it's been getting worse.

[rm] Last meeting was 2 years ago. Fred, you made a joke: "Now all the bugs I find are hard." It used to be that you could easily add new things. Now, bugfixes and features take days and you yak-shave a lot.

[fred] It used to be that people (e.g. Alexander Malmberg) would spend days to implement hard stuff in GNUstep.

[rm] But fixing things is difficult. And new features are hard: you look at Apple docs and you wonder what's going on.

[ivucica] Core Animation a good example.

[rfm] I have a problem with lack of enthusiasm. We try to follow Apple, but I don't like what they are doing. I have no incentive to add things they're adding. But that leaves me without lack of focus because project objective is to be compatible and provide cross platform alternative -- and I don't want to be compatible. And if not that, then what?

[fred] Most of base -- old stuff at least -- is complete, with just a few bugs with NSString case ignore stuff, that we should talk about this weekend. There was a private email, a forwarding of a testplant patch which you may have not noticed as an SVN commit message. It was 2wk ago.

[ivucica] Core Data?

[fred] Didn't we have an implementation?

[ivucica] It cannot serialize the object graph.

[rfm] What do we want it for?

[ivucica] <thoughts about how i use it>

[rfm] Is Core Object the same?

[ivucica] No. It has versioning and sync over network. It's different, more complex and I'm not sure how stable it is. It's very interesting however!

[ivucica] I would have liked it Etoile was more alive.

[fred] There is lack of activity in GNUstep and closely related projects. What's really active is GAP.

[rm] I checked activity and …

[fred] You made releases! Something is going on! But not in most projects. PikoPixel is interesting: people outside are actually using GNUstep. 3-4yr ago, it was an Austrian guy who used GNUstep to lunch his car racing game. He wrote it for Mac, and ported it over to Linux with GNUstep. At that time, he really helped me implement a lot of missing code in GNUstep. This is what keeps me going: if people from outside are interested in stuff.

[rm] If you check GAP project, it says it has 2yr 3 contributors, and in last 30d, 1 contributor -- me. I'm continuing working, because I like it, but… it suffers a similar fate. Maybe stronger. We have tendency that core devs are not interested in applications. Nobody really writes applications. If you write a project, make it more or less complete, it's there but it just stays there, possibly unmaintained. I try to correct this and maintain apps so they don't break. But I can't maintain everything.

[filip] I want to talk about this, and I'd like to prepare a presentation to talk about this.

[ivucica] I want to promote applications by packaging GNUstep.

[rm] We had live CD, but it was not done well. You need to maintain it. Updates to the live CD were even worse than live CD. Person who made it used official packages for Debian, which didn't really work out. Debian packages are strange and things are missing. It very soon became bad. Richard Stonehouse did it better, by using his own packages.

[fred] We could do it better by coordinating releases of core, and the applications.

[rm] To do this, we should have a stable and dev branch. I cannot however say that GAP could always be released at arbitrary points.

[ivucica] Git makes things easier here.

[rm] But it doesn't change things: you still have to do it.

[rfm] We should focus on one system for showcase.

[rm] Yes; it should be an easy-to-use live CD.

[ivucica] I could work on this.

[rfm] We ought to be able to automatically.

[rfm] My main objection: I never got Ubuntu to work

[rm] We should reach out to Richard Stonehouse to avoid duplicating effort.

[ivucica] <explanation of debian source packages being pushed to launchpad>

[rfm] We should build our own stuff because of bad reputation we get from Debian packaging.

[fred] I think we should push things into Debian itself.

[fred] Let's stop packaging discussion here. Let's talk about state and future. Do we want the future?

[rm] Yes, otherwise we would not be here.

[ivucica] I want a window manager.

[rfm] We could try to push for changes to WindowMaker, we could try writing a WM but in Objective-C people have failed.

[rm] We still have to respect other apps.

[ivucica] I can still do it.

[rfm] Why would you want this?

[ivucica] Reference system should be representative of what you can do with GNUstep.

## Future of GCC's Objective-C support (and our support for it)

[fred] I brought this up because of dynamic properties. You are using this in Core Animation, and I wanted to compile this with GCC, and the runtime that comes with it. I was able to compile it, it was just not working because there is no support for dynamic properties in the GCC compiler, and they are missing from the old runtime. So I was unable to use some parts of CA where you use this.

[fred] I even managed to get PikoPixel working with GCC, even though it was not supported for this environment. There was a small change that was necessary. I'd like a GCC version that supports all the features that we need, but if that's not doable, we may want to drop GCC support in the future. Which would be hard for me.

[rm] For me too, and I can explain.

[fred] There are real limitations in GCC's support for Objective-C, and I don't see how we are going to address this.

[rm] As long as those features don't go into core libraries, I'm fine with it. If you want a framework, you need a feature.

[fred] But the list of features that are missing is larger and larger. We have a workaround for blocks, but it's doable. It's horrible.

[rm] I don't like blocks.

[rfm] They don't fit into the design ethos.

[fred] I grew up with LISP, so having functions as first class citizens is just natural to me. I like it that the "modern" languages learned from LISP.

[rfm] They're just not object oriented. You push around blocks instead of designing objects. They're abstract and move you away from OO design.

[fred] SmallTalk had blocks from beginning, and ObjC would be.

[rfm] ST is not a paradigm for OO design.

[fred] I thought it was!

[rfm] It was an early implementation of it. This depends on your view.

[fred] Am I only to see the limitations?

[rm] I see them too. People use the new features.

[ivucica] There are a bunch of features that we need for Mac developers. Esp. @dynamic in core animation.

[rm] GCC support is important to me. We are GNU. And there are environments where you can't get clang. I have solaris machine where I cannot install clang. I need GCC for that.

[fred] Last time we ran into this problem with GCC, Nicola Pero implemented a bunch of these features over a week.

[rfm] What features do we need in the first place?

[fred/rfm] Dynamic properties seem important. Maybe ARC if we can get it in.

[rfm] We should talk to current maintainers. They are not involved with GNUstep.

[fred] I could write to the current maintainers.

[rfm] I believe we can easily get things into the runtime library. But we lack technical knowledge to hack on the compiler.

[fred] What I would be offering to do is do the stuff that's missing in the runtime. It's fairly simple. GCC runtime would not be used much, but as long as something doesn't break it would be good enough and I could do this. I'm just not sure how we could convince GCC people to call these.

[rfm] I'm sure Nicola would at least make sure we get the runtime changes in place. Half the job is under our control…

[rfm] Could we get this as a GSOC project? This is a beefy, but doable project.

[rfm] If we could persuade maintainers to act as mentors, this would be interesting.

[fred] I'll try to talk to them.

[rfm] We have a plan! \o/

## gnustep-make, libobjc2 and new features in GNUstep

[rfm] I specified working on use of different library combo [triplet], which implies use of clang, libobjc2 with blocks, etc.

[rfm] We need an easy way to switch off ARC, because a lot of libraries don't know about ARC.

[ivucica] I would default to ARC being off, and easily being turned on.

[rfm] All I want is we should have a library combo that says 'everything new' rather than having to turn everything on.

[fred] Even if we don't use ARC, there is still autorelease implemented in libraries. Would we be using this in the library?

[rfm] It's in David's library used only for ARC.

[fred] Only for ARC?

[rfm] Anyway, the idea is to tie compiler and library. People have trouble getting all 'new' features working.

[rm] When you have both compilers installed, you can have trouble. E.g. on FreeBSD you can end up with 3 compilers.

[rfm] Some people suggest you specify environment variables. I think we should have a different combo so that we change the defaults.

[fred] How would the make system get all the flags incl. include path set up correctly?

[rfm] Normal autoconf tests.

[fred] But isn't include path already set up at that point?

[rfm] No, it's done during autoconf.

[fred] But you have to be very specific about which options to use when you're calling configure script.

[rfm] If you are building it for new version, if it knows it's going to find clang, it can try to work it out at build time.

[rm] You "do the hard part once".

[rfm] And at this point, you can still override the guesses.

[ivucica] So like now, but with different defaults.

## Future of Make system and alternatives

[ivucica] I dislike gnustep-make, but there is no good alternative.

[everyone] Discussion of different systems and issues.

[ivucica] Is Python acceptable if we were ever to decide to replace gnustep-make?

[rm] No, it's broken on many systems I use.

## Garbage collection

15:05

[fred] Does anyone oppose?

[everyone] No.

## Criteria for new release

[fred] Go through PikoPixel patches and apply them. E.g. image conversion code; when saving images, we need to undo premultiplication of alpha. Or, when drawing with cairo, we always have to swap bytes. We had a patch to speed up raspberry pi, but that was slower on my machine. It would be good if we could do this conversion just once per image instead of when drawing. I'd also like to make it possible to use bitmap images as targets of graphics context. We don't have that; PikoPixel works around it.

[fred] What is the state of base? Is it useful to release it?

[rfm] I think we should release it real soon. It's not /required/.

[fred] When was the last release?

[rfm] Cca half a year ago. We had a bunch of tiny fixes in between.

[fred] Let's release in a month or two?

[fred] Gregory, what are your plans for developing the core?

[greg] I want to fix printing, including on Windows side. Over last few months, I've been thinking about writing a browser using CEF, so that we have a working webview.

[fred] Isn't Riccardo working on SimpleWebKit?

[greg] I think it's good for a light HTML parser, but when you consider the work that went into WebKit, expecting SWK to match CEF's functionality is hard. CEF is also resource hungry. It's not realistic that we would catch up with SWK.

[fred] Should we wait on CEF-based webview for release?

[greg] No.

[fred] Richard and I would like to agree on the release. I'd like this to happen in next 2 months.

[rfm] I will be away for a month. Then we have a gap of week or two before Christmas. Is it realistic that we do a "Christmas release"?

[greg] (Looking back) 19 years! I think my first contribution was NSOutlineView for gorm.

## GUI Backends

### Dropping non-Cairo Backends

[fred] I made suggestion of not just deprecating non-Cairo backend, but really moving away from them. I know this is not popular with Riccardo. I wanted to step down as -gui and -back maintainer, and has more energy and time to spend on it, but it didn't happen so it's just me. And looking back, it's even more just me than a year before. So I'd like to reduce effort and bring GNUstep forward with features that are easy to support with Cairo, but not otherwise.

[rfm] I supported you on FOSDEM. I didn't change opinion

[greg] I agree

[rm] I disagree. We are reducing portability.

[greg] Issue is it's hard to support features Cairo supports. There is no value supporting other backends, there is waste of effort.

[fred] The other option is that I implement new features, and ignore other backends. As long as new methods are not called, it should be fine.

[rm] I'm fine with removing libart. Whenever I use libart, I'm astonished how nicely it renders fonts.

[fred] Cairo at the moment uses just the toy API.

[rm] If you take any other application, Cairo is not nice.

[rfm] Suggestion: why not drop support for anything but cairo, and prioritizes better support using other than toy API?

[rm] I was actually proposing that if you can make Cairo use NSFont, I'd be in favor of that. I'd be okay dropping libart, but keeping xlib and winlib.

[greg] Problem is winlib is using GDI, instead of GDI+. Remote desktop is using GDI over the wire, so when you use winlib, sometimes you can't see it because remote desktop is not sending all the GDI calls over network.

[rm] There are many applications with same problem.

[greg] But should ours be counted among them? I would say no. I'd rather use Cairo, if it solves this problem. If we keep winlib, we don't know when you'll run into this problem for important applications. E.g. Testplant's whole business is based on remote viewing, so if we fail doing what we need to do, we're failing them.

[rfm] We don't have many people who want to use -gui, so few businesses that do, we should take care of them.

[fred] Is it ok if I leave the code in there, but only do new stuff in Cairo? If someone wants to add these new things, they're free to. I'd promise not to break stuff.

[rm] I'm for removing one backend.

[fred] That doesn't help.

[rm] I think we will become Cairo dependant.

[ivucica] I think we will be okay if we keep thinking of using multiple backends like Opal and Cairo.

[greg] There is no plan to move away from backends architecture.

[ivucica] (aware :-))

[fred] I don't want to get rid of other backends, I want to just not support them. A possible change is that we change architecture so we can mix Cairo and Opal calls. Opal is currently unusable, but we should be able to use it in the future.

### Opal backend

[fred] Is anyone willing to work on Opal backend and Opal itself?

[ivucica] At the moment I'd like to focus on packaging, but after that, yes :-)

[fred] Could Cairo backend provide content for Core Animation?

[ivucica] It would be a big hack. I'd prefer not to do it.

### Wayland

[fred] I put it on the list not because I want to work on it, but currently we support Windows and X11 - and many Linux distributions are moving away from X11 to Mir or Wayland or whatever. Last time discussed Wayland, David said it's the only system he could imagine that's worse than X11.

[rfm] Was that a recommendation? :-)

[fred] Things changed a lot for Wayland since then. We should reconsider what to do with it. Our backends are split into two parts: window + event handling, and drawing part. Aside from x11/win32, do we want a wayland implementation of window + event handling.

[ivucica] Mobile is wayland on Jolla.

[fred] Is there compat with X11?

[ivucica] Not that I'm aware of (on Jolla), but there could be an X server that serves that. Xwayland?

[fred] Do we want Wayland backend?

[rm] Will that involve drawing with Cairo?

[greg] Probably yes.

[rfm] I like the description.

[fred] Now that you have event handling, yes.

[fred] Who wants to do this?

[ivucica] Maybe I'd like to do it, at some point in the future. Same point, packaging.

## Git, Gerrit and other tools

[everyone] Git, generally (albeit a bit reluctantly) yes. Gerrit, no, except as an optional tool. Savannah as the main hosting system.

[everyone] We are fine with staying with FSF, as long as bureaucratic admin stuff is way, way simplified. If it's already simplified, we're fine.

[fred] Who's going to talk to Savannah guys about the actual move?

[greg] I will.

other remarks:

[greg] We have [Jenkins](http://gnustep.dnsalias.org:8080/).

[ivucica] Travis-CI exists.

## Packaging

### OpenCSW (rm)

[rm] I want to showcase OpenCSW. See GORM! <https://www.opencsw.org/packages/CSWgorm/>

### OpenSUSE (fred)

[fred] <https://build.opensuse.org/project/show/X11:GNUstep> includes Fedora and SLES.

### Debian/Ubuntu (ivucica)

[ivucica] <https://bitbucket.org/ivucica/gnustep-ubuntu/>

### David's FreeBSD ports

[everyone] Let's skip this!

## Do we compile xib files?

[greg] We should discourage reading xibs directly and primarily compile them into gorm files.

[fred] Microsoft compiles xibs and storyboards, too, into nib files.

[fred] At the moment, you will be losing a lot of information compiling xib into gorm.

# Day 2

10:00

## Who is our target user?

[rm] There are many. This has always been the project. This comes up with the question "What is GNUstep?" and everyone has different ideas. I have "this box" which I want to make usable. I want to, for example, make my Thinkpad usable with GNUstep. I want workspace and a desktop environment (term which I don't like). For other people, it's a way of porting applications.

[fred] Target user is mostly you, and everyone else who wants a full desktop environment.

[rm] I want my applications to remain compatible with a Mac. I love when I run Graphos, I can extend the same app to the Mac. I don't require that every Mac app runs on my computer, but I want the other way around. I think this is a legitimate goal. I can imagine other users would be people just having application and making it cross-platform, which is similar to what I want. For me it's easier, I can start with GNUstep; others don't care about GNUstep, but want to port to Windows (most people don't care about Linux). Many people interested in GNUstep are interested not in Linux but in Windows.

[ivucica] Are we interested in improving tools and IDEs?

[rm] IDEs are very complex, this would be farfetched.

[rm] Target users I see are not interested in GNUstep or anything around GNUstep.

[rfm] There is a whole category of OS X users who would want to use GNUstep to port things to Windows and Linux, but have no idea any GNUstep and don't want to put any effort into it, nor into the project. I feel a lot of people will not even produce patches for their own uses, and I don't know why would we care about catering to them.

[rm] Gregory is not here, tends to care more about these users. But what do these contribute?

[rm] We want to support these users to, because I myself want to do this type of things.

[rm] I had many problems with newer MinGW and newer GCC.

[ivucica] Let's consider exploring how Microsoft is supporting Objective-C.

[fred] Josh Freeman is the target user.

[rfm] Anyone who's producing a good app and patches.

[fred] Anyone who wants to work with us.

[rm] Josh is the best we can get: he mentions GNUstep on his website, he wants to work with us. He's not a lazy programmer. He is the best of what we can get out of the target users. Testplant used to avoid the term GNUstep.

[rfm] Companies try to avoid saying they use free software, especially GPL. They want to use it, but not say they use it.

[fred] Even though we have LGPL.

[rfm] Probably a lot of companies use GNUstep without telling us.

[rm] We might be able to interest people in computing by thinking about small devices (Raspberry PI etc).

[fred] Josh Freeman seems to be using Raspberry PI for his GNUstep experience.

## Riccardo's "Cool GNUstep Application"

[rm] Demo!

## Filip's experiences: GSoC and new user experiences

[filip] Suggestion: advertise GSoC on homepage while it's running

[rfm] "Are you using GNUstep? Get in touch!" on homepage

[filip] Presentation is in GDrive!

## PikoPixel and related patches

### PikoPixel tooltips

[fred] We should support multiline tooltips, and consider wrapping into multiple lines.

### PikoPixel Preview nib windowStyle

[fred] rm and I are looking at this.

## Window manager indirection, borders, stacking order and visibility

## Theming and requests for new features

[fred] Richard, what are you missing about sliders?

[rm] Let's instead ask a bigger question. I want to be able to change icons across the application.

[fred] What do you mean?

[rm] We have a set of common images, called 'named images', which you can theme. But not every application uses these. You have a set of images provided, but what we can theme are decorations (anything that is a GNUstep control, so you can make buttons different), you can supply some base images, but applications may have 'new' button, 'save' button etc. Those applications use their own images. I want to be able to theme them.

[fred] Why don't they use name?

[rm] They don't have them.

[ivucica] We are missing a lot of named images that other platforms have.

[rm] I still want to hack them. Hypothetical PikoPixel exists, I don't have access to its code. I want to make things look consistent.

[fred] You should use named images.

[rfm] if you talk about particular application, you don't theme app by changing the theme, but by adding support for theming to the app.

[rm] I accept I want to make app theme-aware.

[fred] You should use named images!

[rm] I want to provide app specific themes. We can require theme-awareness so apps can use them.

[fred] I don't see how you can work around this. Current issue is just the order in which images will be loaded.

[ivucica] Who should have priority? App and its developer, or theme and its developer?

[fred] Theme.

[rm] I want to theme across apps.

[fred] What is not possible with current setup?

[rm] If I have a mail client with file "New.tiff" representing 'compose', and a text editor with file "New.tiff" representing 'create new document', themes should be able to provide different files.

[ivucica] How about first looking inside theme's bundle at 'app.bundle.something/New.tiff' and only then at 'New.tiff'?

[ivucica] Is GTK theme using the systemwide images for NSImage?

[everyone] We don't know.

[fred] It will definitely replace internal images for drawing borders and controls with GTK-provided images. It will not replace named images with GTK-specific named images for that theme; I would be surprised if it does.

[rfm] We should not count on app having bundle id, we should ALSO namespace apps based on app name.

[rm] This is also important for theming document icons. (Each app specifies its own document icons, which is unthemable right now.)

### Slider improvements

## OPML editor

[ivucica] My OPML editor's NSOutlineView behaves strangely if two nodes in the XML tree have same value.