

from Tapani, December 2013

Here's a brief list of regular, eh, "e-tasks" that someone should take responsibility of, from the top of my head (meaning I've probably forgotten some):

(1) Website content & appearance.

This doesn't require any real technical expertise, just simple Wordpress content editing, but some things should be done in a timely fashion.

Looking at the website, here're some subtasks that should be done regularly (and as noted, these don't require any serious technical skills):

* **Banner photo and CD date should be changed after each meeting**, preferably as soon as possible (or as soon as a photo of the next meeting is available). Three times a year, that is. And indeed this should be done right now (volunteers welcome). Wilson, you've done this in the past, it might be helpful if you explained how it's done.

* **EC member page**, <http://www.ncuc.org/governance/executive-committee/>, should be update yearly after each election and possibly in between if a member resigns &c. (Unless someone beats me to it, I will do this as soon as the election results are in, although some photos will probably need to be added later.)

* **EC meetings page**, <http://www.ncuc.org/governance/ec-meetings/>, should be updated after each meeting, as soon as minutes or a transcript is available. (It's just a list of links to mailing list archives, very easy to edit.)

* **Elections pages**, like <http://www.ncuc.org/governance/elections-2013/>, can perhaps be left alone but should at least be removed from the menu after the election is over, and of course a new one should be created when the next election is at hand.

* **Forthcoming events**, <http://www.ncuc.org/events/forthcoming-events/>, should be maintained at least to the extent of moving events away from there once they're over. (Right now someone might want to remove BA meeting and add Singapore.)

* **Adding new users**. At least all EC members should have logins to Wordpress. (Roles may require some thought. Perhaps "author" might be appropriate initial role, changed to "editor" when need arises; "contributor" could be used for people who need some looking after, maybe even for all NCUC members by default. And "guest" can be used for people who should see hidden menu entries but not write anything themselves.)

Those are perhaps most critical to keep the site from appearing derelict. Member list at <http://www.ncuc.org/participate/members/> as well as all (?) pages with member statistics,

including the map image, are automatically updated from the member database, they do not need to be manually maintained.

Of course it would be nice if new content was also added regularly, blog entries, additions to Statements and Letters, &c, but those don't fall under maintenance by any standard.

(2) Mailing list management.

Again, no expertise required, although some tasks are fairly frequent.

Managing Mailman is perhaps the most important of the e-tasks. I listed, given how much NCUC work relies on email, and it needs attention frequently and often rapid response. On the other hand it's also very easy, technically anyone can do it, the primary requirement is conscientiousness, or reliability if you will.

It is also easy to share between several people in many ways, but the very least there should be two people who can do "everything", or at least one all-purpose admin and one moderator. Different lists could have different managers, now they're all managed by same people (I and Wilson) and have same password (ncuc1:/root/mailman.passwd, which Brenden can also access).

Mailing list management tasks are probably familiar enough, but here's a quick summary with rough estimate of frequency:

(1) *Allowing or disallowing messages held because of some rule.* Common situations are members using a different sender address than the one they're subscribed with, too many Cc: addresses, too big messages, cross-postings resulting in non-members posting, &c. These happen weekly, sometimes more often, even several in one day, and usually should be handled quickly (within the same day).

(2) *Approving subscription requests.* This is much less frequent and tends to happen in bursts a couple of times a year, especially the EC list after elections and the discuss list after admitting new members, occasionally at other times, like when someone changes their address. (The EC list is exceptional in that unsubscribe also needs moderator approval.) And of course if working teams or the like start getting more active using their lists they may increase this, but then it might make sense to delegate subscription approvals to whoever coordinates the team.

Mailman has distinct administrator and moderator roles, moderator basically just doing the two tasks above, i.e., approving or disapproving messages and (un)subscriptions, and it might well make sense to have, say, two administrators for the lot and in addition moderator rights to others in some lists (whenever someone complains about too slow moderation in some list, offer to make them moderator...)

(3) Mass subscriptions. I'll mention this separately from the above, as it cannot be done with the moderator password and is technically somewhat different. Not difficult though, usually it's simply cut'n'pasting list of addresses.

(4) Creating new lists. Infrequent, never really urgent, easy. The mail system has been set up to recognize new lists automatically as long as they use addresses of the form something@lists.ncuc.org, so all it takes is the Mailman list creation form (<http://lists.ncuc.org/cgi-bin/mailman/create>) - no need to worry about editing alias files or whatever various documents talk about.

(5) Adjusting list settings, like maximum messages size &c. Rare, usually only needed after list creation, but sometimes needed to handle spam attacks or temporarily moderating some list member who's gone crazy &c.

That's about it. There are also some list-related tasks that I've omitted, but they're either one-off things that have been done already (general mail system settings) or such that they usually aren't done at all (like messing with archives), and also technically belong to linux admin (cannot be done with the web GUI).

There're already multiple admins in most lists, in most lists Wilson, in some also Brenden and Dave. It might make sense if you guys coordinated a bit who does what and when.

(3) Member database management. Unfortunately I never got around to creating a pretty interface for this, but there's phppgadmin, a basic database web GUI that's simple enough to use with a little documentation about the database content. The only hard part is getting the data...

Here's some basic documentation of the member database.

Maintaining the database here means simply keeping member data up to date: adding new members, correcting and updating data for old, deleting resigned members.

(Unfortunately there's no interface for members themselves to update their own data directly. I was going to write one but... perhaps ICANN labs will come up with something.)

The database is quite simple, with PostgreSQL as db engine. As noted earlier there's no custom interface, but phppgadmin is easy enough even if not exactly pretty.

Access details including URL, login &c can be found in the ncuc1 server in `/root/member-database-access.txt` (presently accessible to me, Wilson and Brenden; just let one of us know who should be added).

Phppgadmin's main drawback is the lack of any content-specific help or safety features like undo, and its error messages are sometimes rather cryptic, but I've peppered the database with a heavy dose of checks, constraints and triggers that make it somewhat hard to do much irreversible damage.

There is only one database, 'ncuc', and only one table, 'ncucers', that normally need to be touched, with the following columns:

id	internal index (autogenerated, never change)
name	name (no separation of first and last name, by convention "firstname lastname" but that's not checked in any way).
role	'OffRep' for an organization's official representative, 'AddRep' for additional (non-voting) representatives, 'Ind' for individual members (must be spelled exactly like that).
organization name	organization name of the organization. Note, organizations are not listed separately, only implied by their official representatives; there should be exactly one official representative per organization. Also, organization probably should be null for individual members but isn't in all cases (these should be checked).
country	obvious (blank for several members, unfortunately)
domain website	copied from NCSG data, not presently used anywhere ditto
email	primary email address (used for ballots); must be unique (but can be null if unknown), forced to lower case
email2	alternate email (to be used in case primary fails)
orgsize	'SMALL' or 'LARGE', null for individuals
state	membership status, possible values: 0 = new member, no ballot (not included in voter roll)

- 1 = old member, contact info not confirmed, no ballot
- 2 = member in good standing, included in voter roll
- 3 = removed (ex-member due to be deleted but kept in db pending verification; no ballot, not shown in member list in the web)

xkey random identifier used in data confirmation emails
(should be regenerated if needed again, no need to edit)

Adding new members, updating member data and deleting old ones, as well as simply browsing member data can be done with phpPgadmin GUI, which has buttons like Browse, Search, Edit, Insert, Delete &c.

Various web pages showing member data use the database and are automatically updated when it changes.

Mailing lists, however, are not linked to the db in any way, in particular ncuc-discuss subscriptions must be handled separately.

That's basically all there is to it - easy as pie. If you managed to read this far and understood most of it, you are qualified to take over. :-)

For completeness, here are other tables that may be of interest (generally not editable):

ncucer_history history of changes made to ncucers (starting now):
 usable for undoing mistakes,
 reviewing changes, making statistics
 (in the future this could even be used to plot
 membership development graphs or something)

ncucer_states valid values for state

orgsizes valid values for orgsize
 (no need to change unless bylaws are changed
 adding medium size organizations or something)

roles valid values for role

There're also a couple of tables that have been used for temporary purposes like comparing data with NCSG, but they're not being used anymore (should really be removed, I may do it yet).

(4) Linux system administration. There's very little that needs doing on an ongoing basis, but someone should read root's emails and be able to react to things like disk filling up &c.

To complete the series, a few words about Linux system administration.

Once the system is up and running there is not much that needs to be done regularly.

At the very least, however, someone should read root's email and react to problems. It is redirected simply by `/root/.forward` and now comes to me - I want someone to take this over ASAP. (Bill, sorry, but if nobody volunteers I'll direct it to you and let you find someone else to take over at your leisure.)

Splitting this among several people would also be possible (as long as they don't both assume the other(s) will take care of everything...)

In particular disk space is likely to run out sometime soon(ish). Also, RAM is fairly low and some services (especially etherpad) tend to die when they run out of memory (and need to be restarted).

System software updates don't need much attention, security patches are installed automatically, other updates can be installed easily enough (`apt-get update`; `apt-get dist-upgrade`) but generally aren't urgent (a couple of times a year is enough).

Wordpress needs to be upgraded separately, as it has been manually installed (system-supplied version being too old) and fairly heavily customized. Skipping a version or two is generally no big deal, but running old version for years would probably not be a good idea.

Upgrading the operating system (Ubuntu LTS) to new release should be done every two years on the average, but it is generally not urgent (old versions are supported several years anyway). The process is nowadays usually very easy, especially on servers, but of course it's recommended to start by checking that backups are up to date (including databases).

Backups are now taken to two locations, in the 2nd disk at Gandi and in my personal backup server. I may leave the latter running for the time being, as it costs me basically nothing, and it's only for real emergencies (including a failure of the Gandi backup). The only care and feeding the former generally needs is making sure it doesn't run out of disk space, and it will start sending emails to root when that happens (easiest solution is purchasing more disk space from Gandi, although some could be saved by reducing backup frequency). Backups are done with `rsnapshot` and can be restored by simply copying from `/backup/...`

Finally, a reminder: there's a list of things that could and perhaps should be done at <http://pad.ncuc.org/p/website-todo> - not exactly a neat presentation, rather a random collection of ideas, but perhaps still useful.

(5) Wordpress administration. Upgrading Wordpress itself, possibly installing new plugins &c.

I listed Wordpress administration as a separate task because it falls somewhere between website content management and Linux administration. This is stuff that's perhaps a bit more technically demanding but infrequent (not even monthly, and rarely anything urgent).

Roughly, it's things that can be done from the web GUI by users with Wordpress 'administrator' role, presently me, Brenden, David and Wilson (who has been using "admin" account for this rather than his own; I also have two accounts, an admin one and a regular one ("author")). In practice Wilson has done most of the WP customization, theme editing &c.

With that division, one task I already listed under website maintenance is actually WP admin task, namely adding (and removing) users. In addition their roles (privileges) may need to be changed from time to time.

Other admin tasks are ***managing (installing and updating) plugins and themes*** and some site-wide appearance settings like menu tree &c. Note that there're a two custom-made (by me or Wilson) plugins, changing them requires rudimentary programming (php) skills. (Specifically, they're "NCUC Members" and "NCUC Ballots".)

One detail perhaps worth mentioning is that a number of pages in the menu have been made visible to logged-in users only, this is done by specifying "Visibility" as "is_user_logged_in()" (uses "Menu Items Visibility Control" plugin).

Upgrading Wordpress itself can be considered a WP admin task or a Linux admin task, depending on the people doing them (likely to be same people which would make this moot, but in principle WP admin doesn't need to know anything about Linux). This is actually something that should be done pretty soon (we're running 3.5.1, the latest is 3.8). The main difficulty here is making sure all customizations (especially things done to the theme) survive the upgrade (and if not, being ready to fix or redo them, or to roll back if it fails; a paranoid might install a new instance of WP just to test it out...)

(6) Domain, VPN resources and SSL certificate renewal. Mainly, forwarding bills to the treasurer in time, once a year. Easy but rather embarrassing if forgotten. (Now we're actually using only self-signed SSL certificate, we should get a proper one ASAP.)

One fairly natural set of tasks is those that are done with our hosting provider, Gandi:

* Renewing domain registration, once a year (although it could be paid for several years at once if desired).

* Editing zone file as needed. Rarely needed, probably not even every year (might want to add IPv6 address at some point though).

* Editing email aliases. Necessary only when @ncuc.org addresses need to be changed or added (and they aren't used much).

* Managing VPS resources: renewing VPS yearly, purchasing more disk space, RAM, CPU power or bandwidth when needed (probably not often - disk space is likely to run out before next year is over though).

* Buying and renewing SSL certificate, if it is decided to use https properly (should've done this long ago but... the cost is some \$12/year, first year free).

All of those are done with Gandi's web interface, which is pretty easy as such things go. Brenden has mostly taken care of these.

(7) Election management. Yearly process, actually very easy if member database is up to date (I wrote a simple tool for producing voter roll in the format Glen &co want it automatically from the db).

As mentioned before, I wrote a couple of tools to help in elections.

First, if you log on to Wordpress (<http://www.ncuc.org/login>), a new menu entry appears under "Participate", namely "Ballots".

(A few others will also appear but they're empty for now.) It's actually a bit of a misnomer as there's only voter list, both as human-readable web page as well as a downloadable text file in the format Glen &co need it to generate actual ballots.

(The login requirement is simply to avoid address harvesting by spammers. It doesn't provide any real security, indeed the pages in question can be accessed without login if one knows their URLs - so please don't post them here.)

When the next election comes, what's needed is simply ensuring that member database is up to date, and in particular that "members in good standing" are marked with state=2, then going to the above page, downloading the weighted voter list and sending it to Glen. (If the voter weights need to be changed, edit the "NCUC Ballots" plugin in Wordpress.)

The hard part is keeping the database up to date, that is, verifying people & their addresses.

The tool I wrote for this is a bit less polished, it needs command-line access to the server. It wouldn't be all that hard to make web GUI for this, but I never got around to doing that - this was a very quick hack. So, I'm afraid the following is a bit more technical, but it should be easy

enough to anyone who can deal with command line. (Note though that there's no absolute need to use this, any means of updating the database (like manually reading emails and using phppgadmin) will do fine for people who like manual labour. Me, I am lazy enough to automate things as much as I can.)

Basically it's a script that sends a customized form letter to all members with state=1, which contains a custom URL, clicking which is logged and yet another script uses the log file to change state of members who replied to 2.

If someone wants to use it for next election: log on to ncuc1 as membadm (or use "sudo su - membadm") and:

- (1) review and edit the form letter template "confirmation1.txt" and the "thank you" message in the reply web page /var/www/members/confirm.php as desired
- (2) ensure all uncertain members have state=1 in the member database; might even do this to all members by
psql ncuc -c 'update ncucers set state=1 where state<3;'
- (3) generate new xkey for all members, e.g., by
psql ncuc -c 'update ncucers set xkey=trunc(1000000000*random()) where state=1;'
- (4) remove old replies from /var/local/ncuc/confirm.log
("rm /var/local/ncuc/confirm.log" will do)
- (5) run "passivecheck.sh", which sends the email to members.
- (6) once deadline for replies is over, run "check-confirmations.sh" to update the database.
(can be run again if replies keep coming - replies can be seen in /var/local/ncuc/confirm.log)
- (7) if you want to send another message to those who haven't confirmed, just do (5) and (6) again (possibly editing the letter template first).

That's all.

Plus, of course, handling people who reply to the letter with corrections or whatever instead of just clicking the link. (A development idea is adding a form where they can update their data directly.) Replies go to [memberadmin at ncuc.org](mailto:memberadmin@ncuc.org), which should be directed to whoever is handling this.

ROLE ADDRESSES

Still cleaning places up, I went through all @ncuc.org addresses and removed myself from them. All of them still go to someone now, as most had two or more recipients, and the rest are directed to root which Dave graciously offered to take over.

For reference, here they are, including who gets them now:

admin	Brenden, Wilson
memberadmin	Wilson
ncuc-memberadmin	Wilson
membadm	Wilson
ncuc	Brenden, Rafik, Wilson
postmaster	Dave
webmaster	Dave
listmaster	Dave
twitter	Brenden, Wilson
youtube	Brenden, Wilson

Some (like the extra variants of memberadmin) could probably be removed; listmaster hasn't actually ever been used but could be (it was intended to be shared alias for list admins, like Joly mentioned earlier).

Anyway, you might want to review the above and consider who should be getting what. In particular memberadmin and ncuc (which is used in printer brochures) might need more people to handle them.

For the technical people among us, a note about email setup:

@ncuc.org -addresses are handled by Gandi and managed by DNS admins (of which I'm still one, besides Brenden and Wilson - who wants to replace me there?). At present they're all aliases pointing to [something at ncuc1.ncuc.org](mailto:something@ncuc1.ncuc.org).

ncuc1.ncuc.org is the actual name of our VPS and it handles addresses ending @ncuc1.ncuc.org, but those should never be used publicly, only in internal aliases. This mechanism allows changing @ncuc.org recipients inside ncuc1, without changing the aliases in Gandi. (They're generally either account names in ncuc1 and redirected with .forward, or aliases defined in /etc/aliases.)

@lists.ncuc.org addresses are directed directly to ncuc1 and to Mailman (using postfix transport rule). This makes things very easy, everything list-related can be done from within Mailman without messing up with aliases &c (ignore things like that in Mailman documentation, should you read such).