

Part 3. napp-it ZFS Server: advanced user (work on progress)

(c) Günther Alka, 12.09.2017 - Licence: CC-BY-SA see <http://creativecommons.org/licenses/by-sa/2.0/>

Thanks to all helping with the manual. You can send an email with suggestions, technical and syntax errors to community@napp-it.org

The screenshot shows a vSphere Client window with a browser displaying the napp-it web interface. The browser address bar shows `http://localhost:81/cgi-bin/napp-it/admin.pl?id=admin,!UAlu/!NnMdeM!UGUacm`. The page title is "napp-it // network & nas appliance - Mozilla Firefox". The page content includes a navigation menu and a table of Pools and Volumes.

Pool	VER	SIZE	ALLOC	RES	FRES	FREE	CAP	DEDUP	FAILMODE	AUTOEXPAND	AUTOREPLACE	ALTROOT	GUID	HEALTH	ENCRYPT	ACTION
a	-	1.40T	581M	-	40.9G	1.40T	0%	1.00x	wait	off	off	-	11947425550452163933	ONLINE	n.a.	clear errors
dev5	28	928G	382G	-	100G	546G	41%	1.00x	wait	off	off	-	7327265687587497166	DEGRADED	n.a.	clear errors
rpool	28	29.8G	13.0G	-	-	16.7G	43%	1.00x	wait	off	off	-	595541815522571375	ONLINE	n.a.	clear errors
test2	28	408G	10.2G	13.4G	13.4G	398G	2%	1.00x	wait	off	off	-	15687756182106061174	ONLINE	n.a.	clear errors

zpool status

```
pool: a
state: ONLINE
scan: scrub repaired 0 in 0h0m with 0 errors on Sun Sep 2 23:00:04 2012
config:
NAME                STATE      READ WRITE CKSUM    CAP      Product
a
c3t50014EE057FCA8D2d0  ONLINE    0    0    0      450.10 GB  WDC WD4500HLHX-0
c6t5001E67002AFCFE0d0  ONLINE    0    0    0      250.06 GB  ST3250820NS
c6t5001E67002AFCFE1d0  ONLINE    0    0    0      250.06 GB  ST3250820NS
c3t50014EE0AD523124d0  ONLINE    0    0    0      450.10 GB  WDC WD4500HLHX-0
c3t5000C5000AAB27AFd0  ONLINE    0    0    0      146.82 GB  ST3146855SS

errors: No known data errors

pool: dev5
state: DEGRADED
status: One or more devices could not be opened. Sufficient replicas exist for
the pool to continue functioning in a degraded state.
```

Part 1: Setup napp-it

Part 2: All-In-One (ESXi with embedded virtual SAN) - download newest from: <http://www.napp-it.org/manuals>

Content: This file is always „work on progress„

1. Scripting with Perl
Perl basics
2. Perl and CGI
3. Install napp-it
- file structure and setup modifications
4. napp-it FREE // napp-it PRO // extensions // Add-Ons // Evaluation
5. napp-it subsystems
6. Localization or modifications for bundlings (use your logo, hide/add/force menu, edit messages)
7. Customize menu (pure css menu up from 16.11 dev)
8. Customize appliance maps

1. Scripting with Perl

Perl is a scripting language mainly used for system administration. Main advantage is that it is included with nearly every Unix or Linux installation out of the box, offers much more comfort than shell scripting and is best suited to manage system files and call system commands.

In the early days of the Internet it was also the main language to generate dynamic Web-applications via CGI scripts. This part is today mostly done with PHP, a scripting language very very similar to Perl in usage and syntax. PHP combines HTML pages with dynamic content while Perl strictly separates the scripts in an separate restricted cgi-bin folder, independant from regular HTML pages. So if you have some scripting knowledgs in PHP or similar languages, you should not have a problem with Perl when you know some basics:

1.1 Perl is an interpreter

Perl uses simple textfiles to write scripts. At the beginning of such a file, you need a reference to the Perl interpreter like `#!/usr/bin/perl`

```
# your code like
print "hello world";
```

You can write these files with any texteditor. You need Unix like line-breaks (only needed for the first line with the reference to Perl). I use a Windows editor (DZSoft Perl editor) but you can use any, best are those with syntax highlight, variable and function overview. Save the file under a name like firstscript.pl - Easiest way to execute such a file is: `perl firstscript.pl`

1.2 Perl datatypes

Perl variables come in three types: scalars, arrays and hashes. Each type has its own sigil: `$`, `@` and `%` respectively. Variables are declared using `my`, and remain in scope until the end of the enclosing block or file. After each command you must place a `;` You should predeclare variables like `my $var="1";` Variables can hold characters, numbers and boolean values without predeclaring type.

examples:

```
$name="Paul";
@names=("Peter","Paul","Mary");          # !! If you want to acess Paul: $name=$names[0]; (The second element of a list is a scalar)
```

```
%adress = (
    "name" => "Peter" , "age" => "23" ,
)
```

You can get elements like: `$age=$adress{"age"};`

1.3 napp-it is a Perl application, you need some very basic Perl knowledge to extend napp-it

Perl in about 2 hours 30 minutes: read: <http://qntm.org/perl>

Learn Perl Interactively: <http://www.guru99.com/perl-tutorials.html>

2. Perl and CGI (call Perlscrippts via a webserver)

If you want to call a Perlscript via your Browser, you need:

- a Perl interpreter (you do not need to install. Nearly every Linux/Unix comes with Perl preinstalled, huge advantage over other languages)
- a CGI capable webserver (napp-it includes mini-httpd as embedded webserver). You must allow executing Perl-files.
- a Perl CGI-library (there are many, I use cgi-lib.pl due to its easyness, available from <http://cgi-lib.berkeley.edu/>)

The CGI library is used to parse form and url parameters from browsers like `http://server.com/cgi-bin/perlscript.pl?var1=abc&var2=123` to a hash %in
Within your perlscript, you get these hash-values like:

```
$in{"var1"}    -> holds abc  
$in{"var2"}    -> holds 123
```

The CGI library must be included at beginning of a script via `require ("cgi-lib.pl");` values are parsed with the `&ReadParse;` command.
Part of napp-it is the library `admin-lib.pl` that combines this basic lib with a lot of napp-it functions.

example (CGI-script to display zpool status of a given pool), write it with a texteditor and save as `/var/web-gui/_my/wwwroot/test.pl`

Set permissions of this file example via WinSCP to executable (755) or you get error 500.

Execute the script from your Browser: `"http://ip:81/_my/test.pl?pool=poolname"` (replace poolname with your pool, ex tank)

```
#!/usr/bin/perl  
  
use strict;                                # all vars must be predeclared  
use vars qw(%in $pool $r);                 # predeclare public vars for use strice  
  
use CGI::Carp qw (fatalToBrowser);         # output script errors to browser  
  
require "/var/web-gui/data/wwwroot/cgi-bin/admin-lib.pl"; # cgilib and common used functions of napp-it  
&ReadParse;                               # parse parameters from browser to Hash %in  
  
$pool=$in{"pool"};                          # get poolname from browser: test.pl?pool=poolname  
if ($pool eq "") { $pool="rpool"; }         # default: show rpool  
  
my $r=`sudo /usr/sbin/zpool status $pool`;  # call zpool with root permissions  
  
##### output #####  
print „Content-type: text/html\n\n“;       # print html header  
  
print <<EOF;  
<hr>zpool status $pool<hr>  
<pre>$r</pre>  
EOF  
  
# /multiline output up to EOF at beginning of a line  
#####
```

3. napp-it Setup: file structure

Napp-it is mainly a copy and run application, what means that the installer just copies all needed files to `/var/web-gui/` with subfolders:

<code>/_log</code>	Logfiles, jobs, settings, update-safe
<code>/_my</code>	User -settings, -menus, -actions, -www-files settings are update-safe
<code>/data</code>	Menus, actions and napp-it tools of current version
<code>/data/napp-it/data/zfsos/</code>	Menus (each menu is a folder) Within each menu you find an <code>action.pl</code>
<code>/var/web-gui/data/tools/</code>	Tools and embedded webserver <code>minihttpd</code>
<code>/data_xx.yy</code>	Other versions of napp-it (switch to this version = copy to <code>/data</code>)

3.1 Menus

Without language modification (`/var/web-gui/data/napp-it/zfsos/_lib/lang/*`) a menu name is the folder name without `nn_` at the beginning. (`nn_` is only used to sort entries). Private language files are in `/var/web-gui/_my/zfsos/_lib/lang/MY/` (or other uppercase folders)

3.2 Menu actions

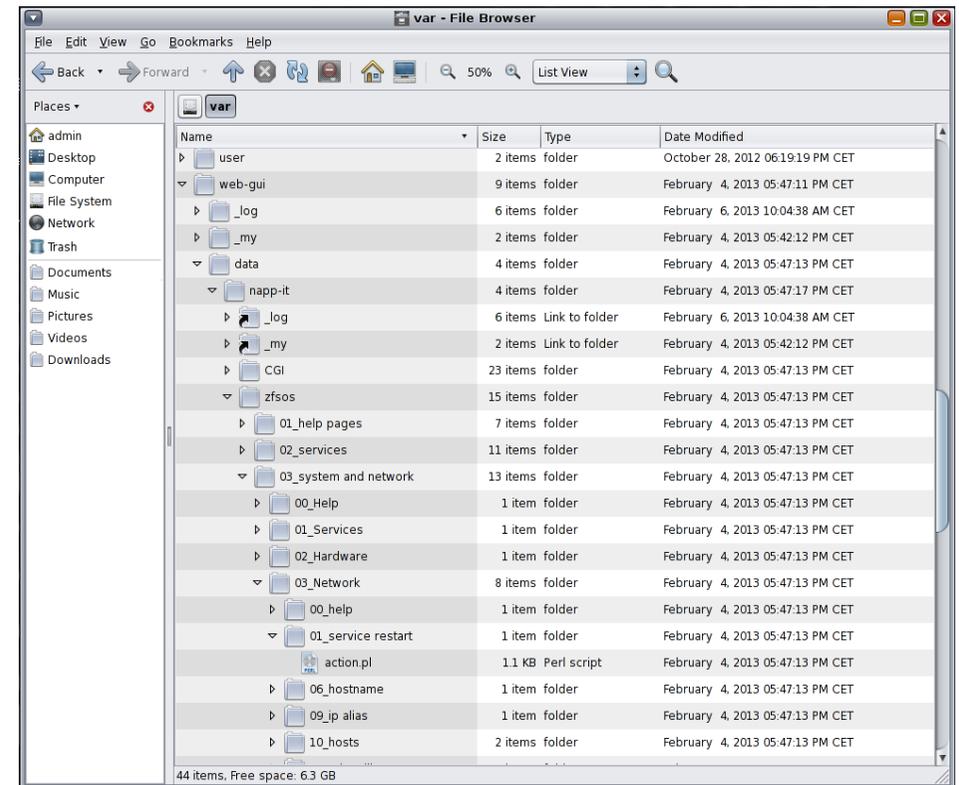
Within every menufolder, you find a Perlscript `action.pl` with all actions that are executed when the menu is called. In simple cases, this script can hold all needed actions optionally with sub-functions in this file.

If you need functions in different menus you can use library files with common used functions and include them in your action file - either from napp-it or you can write your own and include ex via `require "/var/web-gui/_my/zfsos/_lib/myfunctions.pl";`

All napp-it libraries are in `/var/web-gui/data/napp-it/zfsos/_lib/`

Default private library:

If you rename `/var/web-gui/_my/zfsos/_lib/mylib.pl.disabled` to `/var/web-gui/_my/zfsos/_lib/mylib.pl` it is included automatically



3.3 Other system-modifications during napp-it wget installation

If you run the napp-it wget installer, it creates the `/var/web-gui` folder and:

- creates a user napp-it
- creates a initfile to start napp-it `/etc/init.d/napp-it` with a link in `/etc/rc3.d/`
You can use this initfile to start/stop napp-it example
`/etc/init.d/napp-it stop` (options are start, stop, restart)
- install all needed file server services (CIFS, Comstar etc)
- setup Solaris CIFS server

- install tools like smartmontools, bonnie, mc etc

Minimal customisation

If you only want to display your logo and favicon

Copy your logo.png to `/var/web-gui/_my/wwwroot/` and your favicon.ico to `/var/web-gui/_my/wwwroot/`

They will be displayed automatically

The above is allowed without a bundling/redistribution contract.
(Your own servers or napp-it Pro configured on customers demand)

Bundling/ Reseller customisation

Rename `/var/web-gui/_my/zfsos/_lib/lang/MY/` to MY!
This makes this menu set mandatory

Create/Edit `/var/web-gui/_my/zfsos/_lib/lang/MY!/about_menus.txt`
Use `/var/web-gui/data/napp-it/zfsos/_lib/lang/app/about_menus.txt` as reference

Restore the settings

All modifications in `/var/web-gui/_my/` are update save.
If you want to save/restore your modifications, simply restore `/var/web-gui/_my/`

4. napp-it FREE // napp-it PRO // extensions // Add-Ons

4.1 napp-it FREE

Napp-it is a software package with open sources. You can read and manipulate the sources for inhouse use unless you do not give away the modifications, keep license infos intact and use only FREE options. You are not allowed to use PRO features beside evaluation.

In this sense napp-it is not OpenSource. You are not allowed to give away or distribute napp-it without our agreement. You are not even allowed to install or configure napp-it on behalf of others if you are not an employee. But if you are an end-user (commercial or private), you are allowed to install and use napp-it FREE installed online with the napp-it wget installer without costs and other restrictions (example time or storage capacity) for your inhouse use. The napp-It Web-GUI and the underlying OS is "Use as it is, use at own risk"

4.2 napp-it Add-Ons

Free Add-Ons are free Extras that can be installed without fee and without support. Examples are XAMPP, Netatalk or Mediatomb

4.3 napp-it PRO bundling

If you like to distribute napp-it, install or configure a ZFS server on behalf of a customer, you need to acquire a PRO/ Bundling licence for every server. This PRO edition enables access bugfixe editions and includes an improved GUI performance and adds features like the Tuning and Security Panel. With bundlings, you can use your logo, hide/ add menus or edit messages or the shown appliance name with a menu-control file. For a first test select menu/lang „app“ in About >> Settings. You can edit this menu control-file `/var/web-gui/data/napp-it/zfsos/_lib/lang/app/about_menus.txt` with a texteditor or WinSCP. If you want to force this as the only menu option, copy it to `/var/web-gui/_my/zfsos/_lib/lang/MY!/about_menus.txt` (Use MY for a regular menu and MY! as the only one)

4.4 napp-it Pro with Extensions

Extensions are nonfree Extras under a different licence - either delivered from napp-it or others. You can acquire usage permissions for one year, without time limits and a special release for limited or unlimited noncommercial home-use. Napp-it PRO features are always includes as well as bugfixes / access to bugfix releases for the first year. For users of the complete commercial extension, we will answer technical emails for the first year

napp-it Extensions:

- ACL and advanced user settings
- Monitor and advanced network settings
- Replication with remote control of source machines (Key covers one backup machine and unlimited sources)
- Complete: all of above.

4.5 Evaluation

If you install napp-it for the first time, a 30 day PRO Evaluation period with all extensions is included. After 30 days, you have a regular FREE version. You can extend the evaluation online at napp-it.org/extensions. All Pro features beside update are available.

5. napp-it Subsystems

5.1 CGI - Webapplication

Basically, napp-it is a Perl-CGI application (admin.pl) to generate a Web-UI, something like a tiny CMS application framework to manage a server. You can create menu entries and assign actions to that menu. You can rename, translate or arrange the menus. A menu action can call CLI commands with root permissions to setup something or display a system state. Whenever you click on a menu, alle needed systeminfos are requested and modifications are done via CLI Unix commands on demand. After the commands have finished, a new webpage with the result is displayed.

This can introduce problems with CLI command that may need a long time to run. Diskinfos like smartmontools or detection of SAS slots can last up to several seconds per disk to deliver the needed information. Such informations are buffered to display them immediately as long as the basic diskstate has not changed.

5.2 Background Agents

A pure CGI - Webapplication cannot process any information independent from a menu action that is clicked by a user. Control over actions that lasts longer than a few seconds is not possible. For this reason napp-it includes background agents to start and control jobs, replications, monitor system state and things like that. Background agents are executed and managed by a always running shellsript taskserver.sh.

5.3 HTML-5 Mojolicious Websocket server (up to napp-it 2016.11)

CGI Applications are quite static. You call a menu and get an answer. If something happens afterwards, you will get informed only at next menu click. For realtime actions example update of job or disk state or display the server overall state in realtime, you need bidirectional datalinks that keeps intact after a page is fully loaded with the option to update page content in realtime. Included with napp-it is the Perl-based Mojolicious webserver (socketserver.pl) used for websocket transfers.

Currently not every browser supports websockets. Best are Google Chrome and newer Firefox releases.

Up from napp-it 2017.01 napp-it removes Mojolicious to reduce complexity. and use the default mini_httpd webserver for Realtime monitoring and acceleration

6. Localization

6.1 Menus

6.1.2 Menu Layout

napp-it creates two independent menus. One is a javascript rollover-menu build with the help of the javascript jQuery menu library mbmenu http://pupunzi.open-lab.com/mb-jquery-components/mb-_menu/. The other is a static menu that shows the selected menu item and allows navigation as well. They work independently. Most napp-it menu creation functions are in the file `/var/web-gui/data/napp-it/zfsos/_lib/interface.pl`. If you copy this file to `/var/web-gui/_my/zfsos/_lib/interface.pl` this private copy is used. (This file is kept even on napp-it updates)

6.1.2 Default behaviour

napp-it reads in all folder below `/data/napp-it/data/zfsos`. Each folder example `01_disk` is a menu item. The numbers are not shown and are used to sort the entries while the text is the default menu text. In every folder is a file named `action.pl` that creates the content and actions for this menu item. This file can include all actions or it can call functions from common used libraries. They are either loaded within this file via `require library.pl` or they are preloaded based on menus. The default menu text and visibility can be manipulated.

6.2 Manipulating menus

On every page-load, napp-it reads the default english language files to a hash txt (%txt):

<code>/var/web-gui/data/napp-it/zfsos/_lib/lang/en/about_menus.txt</code>	(Alternative menu text, hide menus, use icons etc)
<code>/var/web-gui/data/napp-it/zfsos/_lib/lang/en/about_basics.txt</code>	(Text that is used in several menus/submenus)
<code>/var/web-gui/data/napp-it/zfsos/_lib/lang/en/current menu.txt</code>	(Text for current menu)

napp-it supports multiple languages. You can select a language in napp-it settings (from folders `/var/web-gui/data/napp-it/zfsos/_lib/lang/`). If you have selected a language other than `en`, the according language files are loaded. A translated entry overwrites the default english entry in %txt.

Example: language `de` (german)

<code>/var/web-gui/data/napp-it/zfsos/_lib/lang/de/about_menus.txt</code>	(Alternative language menu text, hide menus, use icons etc)
<code>/var/web-gui/data/napp-it/zfsos/_lib/lang/de/about_basics.txt</code>	(Language Text that is used in several menus/submenus)
<code>/var/web-gui/data/napp-it/zfsos/_lib/lang/de/current menu.txt</code>	(Language text for current menu)

Options in `about-menus.txt`:

Each entry has the following syntax `<<key>> value` where the value can be a multiline value up to the next key.

You can edit the menu-options and translations within napp-it (monitor extension required) in menu `About-Translations` and you can check the current values in this hash %txt when you activate editing in `topmenu` and click to %txt.

If you want to create a translation to a new language, just create a new language folder, select the new language and translate the english messages

Force a single menu

<code>/var/web-gui/_my/zfsos/_lib/lang/MY!/about_menus.txt</code>	(A file „ <code>about_menus.txt</code> “ in the folder <code>MY!</code> is forced as the only menu option)
---	--

6.3 Options in about-menus.txt

Default behaviour: <<key>> value	(examples)
<<app_css>> /_my/style.css	(use a private css)
<<app_script>> /_my/script.js	(use a private script file, example for a different rollover-menu)
<<app_title>> your text	(different menu title, html header value)
<<app_updateserver>> www.napp-it.org	(server to load list of available online updates, ex update-0.9.txt)

<<m05>> Disks	Menu description for menu 05
<<m05.02>> Hotswap	Menu description for submenu 02 below disks

Switches are Options added to the value after=

<<m05>> Disks =H	Hide this menu
<<m05>> Disks =A	Menu only for admin (not operator)
<<m05>> Disks =N	Not editable within napp-it
<<m05>> Disks =I	use an icon ex 05.png or 05.02.png, Icon must be in Icon-Folder defined in script.js (iconPath)
<<m05>> Disks =sol	Show this menu only in OS=Solaris (not in others like OI)
<<m05>> Disks =-sol	Show this menu in all OS but Solaris

6.4 Options to show another logo, hide menus or edit messages

<<app_editmenu>> no
<<app_editmenu_inFo>> show top level edit menu (no)

<<app_about_header>> Your ZFS Appliance
<<app_about_header_inFo>> Menu About: header text (on green)

<<app_about_show_license>> no
<<app_about_show_license_inFo>> Menu About: show license information, disable=no

<<app_about_appliance>> DataApp (example)
<<app_about_appliance_inFo>> Appliance Info ex var1, var2, var3

<<app_css>>
<<app_css_inFo>> Path to your css, example: /_my/style.css

<<app_text_encoding>>
<<app_text_encoding_inFo>> different html charset ex: utf-8

<<app_logo>>
<<app_logo_inFo>> example:

<<app_os>> DataApp OS
<<app_os_inFo>> Override OS name

<<app_script>>
<<app_script_inFo>> Path to your js Menu file ex /_my/script.js

<<app_title>> DataApp appliance
<<app_title_inFo>> Your appliance page title

<<app_hide_zfs_services>> afp, ftp, www
<<app_hide_zfs_services_inFo>> Hide these options in menu ZFS Filesystems example: afp, ftp, rsync, www

<<app_about_pools_product>> appliance vendor
<<app_about_pools_product_inFo>> Menu pools, productname above pool status, 16 char or less, default: Product /napp-it

<<app_about_show_running_on>> no
<<app_about_show_license_inFo>> Menu About: show running on SunOS., disable=no otherwise enter a new text

6.5 Update safe modifications

Any files in the folders
/var/web-gui/_log/ (napp-it settings) and
/var/web-gui/_my/ (private menus and settings) are update safe

If you create a menu control file /var/web-gui/_my/zfsos/_lib/lang/MY!/about_menus.txt
this is mandatory and update safe. A user cannot select another menu.

If you rename the folder MY! to MY you can switch between this menu and the default en Menu.

6.5 How to start with user modifications ex for bundlings

1. copy /var/web-gui/data/zfsos/_lib/lang/app/about_menus.txt to /var/web-gui/_my/zfsos/_lib/lang/MY/about_menus.txt
2. select MY as menu in About >> settings

You can now switch between app and en in to toplevel menu (upper right, next to logout)

3. Edit /var/web-gui/_my/zfsos/_lib/lang/MY/about_menus.txt according to your needs

4. If everything is ok, rename /var/web-gui/_my/zfsos/_lib/lang/MY to /var/web-gui/_my/zfsos/_lib/lang/MY! to force this menu
If you need the default en menu ex for updates, rename /var/web-gui/_my/zfsos/_lib/lang/MY! to /var/web-gui/_my/zfsos/_lib/lang/MY and select en (topmenu)

7. Custom Menu

napp-it use a pure css driven menu to reduce complexity

HTML Structure

```
<div id="menupos">
<ul id="menu">
  <li><a href="#">Main menu1</a></li>
  <li>
    <a href="#">Main menu 2</a>
    <ul>
      <li><a href="#">Menu 2.1</a></li>
      <li><a href="#">Menu 2.2</a></li>
      <li><a href="#">Menu 2.3</a></li>
      <li><a href="#">Menu 2.4</a></li>
    </ul>
  </li>
  <li><a href="#">Main menu 2</a></li>
</ul>
</div>
```

CSS Structure

```
/* css menu style */

/* Main menu */

#menupos { position:absolute; top:30px; left: 0; z-index: 99990; width: 99%; min-width: 1280px }

#menu
{
  width: 100%;
  margin: 0;
  padding: 10px 0 0 0;
  list-style: none;

  /* no border or background
  highbackground: #111;
  background:-moz-linear-gradient(#444, #111);
background:-webkit-gradient(linear,left bottom,left top,color-stop(0, #111),color-stop(1, #444));
ackground:-webkit-linear-gradient(#444, #111);
background:-o-linear-gradient(#444, #111);
background:-ms-linear-gradient(#444, #111);
background: linear-gradient(#444, #111);
-moz-border-radius: 5px;
border-radius: 5px;
-moz-box-shadow: 0 2px 1px #9c9c9c;
```

```
-webkit-box-shadow: 0 2px 1px #9c9c9c;
box-shadow: 0 2px 1px #9c9c9c;
*/
```

```
}

#menu li
{
  float: left;
  padding: 0 0 10px 0;
  position: relative;

  height:16px;
  z-index: 99990;
}

#menu a
{
  float: left;
  height: 10px;
  /* space between main menu items*/
  padding: 10px 8px;
  color: #555555;
  /* text-transform: uppercase; */
  font: 14px/16px SourceSansProBold,Arial,Helvetica;
  text-decoration: none;

  text-shadow: 1px 1px #eeeeee;
}

#menu li:hover > a
{
  color: #fafafa;

  border-radius: 5px;
  text-shadow: unset;

  /*Link hover*/
  background: url(../_doc/menu/images/bgnd_sel_2.jpg");
  color: #333;
  text-shadow: 1px 1px #eeeeee;
}

*html #menu li a:hover /* IE6 */
{
  color: #fafafa;
}
```

```

#menu li:hover > ul
{
    display: block;
}

/* Sub-menu */

.icons { position: relative; top: 5px; margin: 0px 10px 0 5px }
.subbg { background: url(../_doc/menu/images/bg1.png"); }
.subbgmore { background: url(../_doc/menu/images/bg1more.png") }

/*first line of submenu1, repeat main*/
.subbg0 { background: url(../_doc/menu/images/bg1.png"); border-bottom: 1px solid #cccccc; z-index: 99993; background-size: contain; }

/* level1 submenu*/
#menu ul
{
    list-style: none;
    margin: 0;
    padding: 0;
    display: none;
    position: absolute;
    top: 30px;
    left: 0;
    z-index: 99991;

    background: #444;
    background:-moz-linear-gradient(#444, #111);
    background:-webkit-gradient(linear,left bottom,left top,color-stop(0, #111),color-stop(1, #444));
    background:-webkit-linear-gradient(#444, #111);
    background:-o-linear-gradient(#444, #111);
    background:-ms-linear-gradient(#444, #111);
    background: linear-gradient(#444, #111);

    -moz-box-shadow: 0 0 2px rgba(255,255,255,.5);
    -webkit-box-shadow: 0 0 2px rgba(255,255,255,.5);
    box-shadow: 0 0 2px rgba(255,255,255,.5);
    -moz-border-radius: 5px;
    border-radius: 5px;
    border-left: 1px white;
}

/* level2 submenu*/
#menu ul ul
{
    top: 0;

```

```

/* align l2 menu */
left: 200px;
border-left: 2px solid #eee;
z-index: 99992;
}

#menu ul li
{
    float: none;
    margin: 0;
    padding: 0;
    display: block;

    /* */
    -moz-box-shadow: 0 1px 0 #111111, 0 2px 0 #777777;
    -webkit-box-shadow: 0 1px 0 #111111, 0 2px 0 #777777;
    box-shadow: 0 1px 0 #111111, 0 2px 0 #777777;

    opacity: 0.95;
    border-radius: 5px;
    border-bottom: 2px #888888 solid;

    width: 200px;
    height:35px;
}

#menu ul li:last-child
{
    -moz-box-shadow: none;
    -webkit-box-shadow: none;
    box-shadow: none;
}

#menu ul a
{
    padding: 0px;
    height: 10px;

    height: auto;
    line-height: 1;
    display: block;
    white-space: nowrap;
    float: none;
    text-transform: none;

```

```

/* text */
color: #eeeeee;
font: 14px/25px SourceSansProRegular,Arial,Helvetica;
text-shadow: unset;

*html #menu ul a /* IE6 */
{
    height: 10px;
}

*:first-child+html #menu ul a /* IE7 */
{
    height: 10px;
}

#menu ul a: hover
{
    background: #0186ba;
    background: -moz-linear-gradient(#04acec, #0186ba);
    background: -webkit-gradient(linear, left top, left bottom, from(#04acec), to(#0186ba));
    background: -webkit-linear-gradient(#04acec, #0186ba);
    background: -o-linear-gradient(#04acec, #0186ba);
    background: -ms-linear-gradient(#04acec, #0186ba);
    background: linear-gradient(#04acec, #0186ba);

    color: white;
    background: url(../_doc/menu/images/box_menu_over.png");
}

#menu ul li: first-child > a
{
    -moz-border-radius: 5px 5px 0 0;
    border-radius: 5px 5px 0 0;
}

/* creates the arrow up or left
#menu ul li: first-child > a: after
{
    content: '↑';
    position: absolute;
    left: 30px;
    top: -8px;
    width: 0;
    height: 0;
    border-left: 5px solid transparent;
    border-right: 5px solid transparent;
    border-bottom: 8px solid #444;
}

```

```

}
*/

#menu ul ul li: first-child a: after
{
    left: -8px;
    top: 12px;
    width: 0;
    height: 0;
    border-left: 0;
    border-bottom: 5px solid transparent;
    border-top: 5px solid transparent;
    border-right: 8px solid #444;
}

#menu ul li: first-child a: hover: after
{
    border-bottom-color: #04acec;
}

#menu ul ul li: first-child a: hover: after
{
    border-right-color: #04acec;
    border-bottom-color: transparent;
}

#menu ul li: last-child > a
{
    -moz-border-radius: 0 0 5px 5px;
    border-radius: 0 0 5px 5px;
}

/* Clear floated elements */
#menu: after
{
    visibility: hidden;
    display: block;
    font-size: 0;
    content: " ";
    clear: both;
    height: 0;
}

* html #menu { zoom: 1; } /* IE6 */
*: first-child+html #menu { zoom: 1; } /* IE7 */

/* end menu */

```

8. Customize appliance maps

1. Copy a mapfolder ex `/var/web-gui/data/wwwroot/_doc/map/blueline/` to `/var/web-gui/_my/wwwroot/_doc/map/`
(Folders under `_my` are update save and private)
2. rename `/var/web-gui/_my/wwwroot/_doc/map/blueline` to a name that contains an uppercase character ex `Myblue`
(Maps with an uppercase character are treated as private)
3. You can now select `Myblue` as a map
Edit the background images and set positions of led, buttons and descriptions in the included `pos.txt` file

You can crosscheck `„/var/web-gui/data/napp-it/zfsos/05_Disks and controller/12_Appliance_Maps=-lin/action.pl“`
in sub `my_print_map` for possible values.