DIY Zoning: The Complete Configuration Guide

by Vadim Tkachenko

1. Introduction

Note:

Work in progress, verify everything!

FIXME (VT):

This page is almost useless without screenshots. As usual, there's a tradeoff between having the code completed, and having the documentation written - so no screenshots until after `0.1p7dev2` release. If I forget about it, remind me.

Proper configuration is one of essential pieces of DZ's operation. The following components must be configured in order to make system operational:

- 1-Wire® network hardware;
- 1-Wire® network adapter[s];
- RRD Logger;
- Plug-and-Play capabilites discovery broadcast and listener services;
- Temperature zones;
- Mapping temperature sensors to temperature zones;
- Servo controller[s] (damper actuators);
- Mapping damper actuators to temperature zones;
- HVAC unit drivers;
- Mapping temperature zones to HVAC units;
- Temperature zone schedule.

This section is intended to be used as a context-sensitive help media for the configuration wizards, but you may get a better understanding of the way the system is configured if you just read through all of it.

2. Configuration Policy

A rule of thumb for the configuration readers and writers is: *Be liberal in what you accept, be*

conservative in what you produce.

In other words, the configuration wizards will apply their best efforts to produce a working configuration, and the DZ modules (DAC and CORE) will try their best to work with whatever configuration they're given.

Note:

In particular, this means that it is quite possible to get DZ up and running with no hardware whatsoever - you'll get a working model, though, and a feel for what it's like.

3. Common Configuration Elements

There are several configuration elements that are common to all the DZ modules.

3.1. New vs. Existing configuration file

Both DAC and CORE configuration wizards will present a choice of using existing configuration file, or creating a new one.

If an existing configuration file is selected, the configuration that may be contained in it is read and used thereinafter. Otherwise, a configuration is created from scratch, using reasonable default assumptions.

Default configuration file locations are determined by DZ compilation and deployment options. Most probably, you want to use them as they are.

FIXME (VT):

Currently, DAC configurator is not able to read existing configuration files, only produce a new configuration from scratch. This is planned to be fixed in `0.1p7dev3` release.

3.2. Probing USB bus and serial ports

In order to successfully probe both the USB bus and serial ports, you have to have sufficient permissions. If you don't, you will either see the exception trace (for USB bus), or no serial ports (or not all of them) will show up in the serial port probe result panel.

3.3. PnP announcer and listener configuration

By default, PnP announcer and listeners are configured to use:

- Unicast TCP ports 5000 (for temperature sensor data), 5002 (for actuator control) and 5014 (used by VIEW module);

- Unicast UDP ports 5001 (for temperature sensor data) and 5003 (for actuator control).

If you don't have any network-enabled daemons using these ports already, you may simply use the default values.

Note:
If you change settings in one place (such as DAC announcer), you must change settings to the same values in another (such as CORE listener). Otherwise, the listener will not be able to receive capabilities discovery announcements.

Note:
One of possible reasons for announcers or listeners to fail is that you are already running the modules you're trying to configure. Make sure that when you're running `dz_dac_wizard`, the DAC is not running at the same box. Same for `dz_core_wizard` and DZ CORE.

3.4. Mapping Entities

There are some places where entities of one kind have to be mapped to entities of another:

- Temperature sensors to temperature zones;
- Damper actuators to temperature zones;
- Temperature zones to units;
- etc.

All these cases are handled in a more or less uniform manner - using a dual list panel with some buttons in between.

FIXME (VT):
Include screenshot.

4. DAC Configuration

Note:
Information regarding DAC configuration is current as of 0.1p7dev1 release.

4.1. Selecting 1-Wire® Driver

4.1.1. OWAPI vs. OWFS

OWAPI (1-Wire® Java API by Dallas Semiconductors is a native 1-Wire® device driver. Therefore, it is possible to get more from it since the abstraction level is lower, and commands can be used more efficiently. On the other hand, it depends on RxTx, which may

be funky sometimes. RxTx is the weakest link in the chain, though it seems to have improved recently.

OWFS (1-Wire® virtual file system) seems to be more stable than RxTx, but, on the other hand, its level of indirection is higher, therefore it'll be difficult to make the OWFS based driver as efficient as OWAPI based one.

Verdict: *If you can compile RxTx, go for OWAPI, otherwise OWFS.*

4.2. Configuring RRD Logger

RRD (stands for Round Robin Database) logger allows to log the data using constant amount of space. You definitely want this feature - it allows to keep statistics, estimate the system performance and create temperature, humidity and pressure graphs for extended periods of time.

Default file and directory locations are determined by DZ compilation and deployment options. Most probably, you want to use them as they are.

4.2.1. JRobin vs. RRDTool

JRobin is a native Java tool. RRDTool is not. JRobin is slower (so far; but it's improving rapidly) - but RRDTool has more of a call overhead. RRDTool's graphs are nicer - but JRobin is catching up, fast.

Verdict: *JRobin is the way to go for beginners. Use RRDTool only if you know what you're doing.*

4.2.2. Should I enable RRD autocreation?

Short answer: only for a short period of time, until your 1-Wire network configuration stabilizes.

Long answer: currently, RRD regeneration code is pretty slow - it may take tens of minutes, hours or even days on big RRD databases (several years) to regenerate the database. Debugging this code is pretty difficult - in particular because it is taking a long time to make just one test run.

However, on small samples the regeneration is pretty tolerable. Therefore, enable it when you start playing with the DAC, let it autoconfigure the RRD, and then disable the autocreation.

4.2.3. Should I keep the raw trace file?

The raw trace file is a text file where all the data samples are recorded as string literals, in human readable and easily parseable form.

Short answer: usually, yes.

Long answer: RRD is lossy - the further back in time we go, the less is the data precision. Trace file allows to recreate the RRD database in case it was lost/damaged, and/or in case when a change in the RRD structure is required (for example, a new sensor has been added).

Amount of space taken by a trace file is negligible by modern standards.

5. Core Configuration

Note:

Information regarding CORE configuration is current as of current CVS development version. It is expected to last until, and including, `0.1p7dev2` release.

Note:

Make sure you have the DAC instance you've just configured up and running, or you will not be able to benefit from PnP autoconfiguration abilities.

5.1. Mapping Temperature Sensors to Temperature Zones

This is an entity mapping page.

Left panel (with "Temperature Sensors" heading) contains list of sensor addresses discovered on the network. If it is empty, either your DAC instance is not running (start it), or it is not reachable (check your firewall settings - see PnP announcer and listener configuration).

Right panel (with "Temperature Zones" heading") will eventually contain a list of temperature zones you've created. It is not possible to proceed unless at least one zone exists.

Note:

It is possible to have zones without sensors assigned to them. This is intentional - sometimes, some troubleshooting may be required to make the sensors discovered, so this is a shortcut to avoid being cornered. It is possible to rerun the configurator later, or to modify the configuration by hand.

5.2. Selecting Servo Controller[s]

Problems you may run into are described in Probing USB bus and serial ports section.

You will not be able to proceed until you test the devices you've discovered and then

selected. If you don't have your hardware connected yet, skip this page - a `NullDamper` entity will be used instead of a real device.

5.3. Mapping Servos to Temperature Zones

Note:
If you don't have your hardware connected, skip this page.

This is an entity mapping page.

The "Test" button allows to test the servo currently selected in the left list.

5.4. Adding HVAC Units

This page allows you to add or delete HVAC units and configure them.

In order to properly configure your HVAC unit, you will have to know what it is - examine your hardware, manuals, talk to HVAC contractors, do whatever it takes to make a correct identification.

Note:
Failure to correctly identify your unit may render it inoperable.

5.5. Mapping Temperature Zones to Units

This is an entity mapping page.

You must map all the entities - there must be no temperature zones left not assigned to units, and no units with no zones assigned to them.

If you have any extra entities left, go back and delete them.

6. View Configuration

FIXME (VT):
This is not implemented yet (planned for `0.1p7dev3` release) - currently, there is a `./etc/console.conf.xml` configuration file. The syntax is extremely simple, though - all you have to modify is a hostname and port, and even that only in the case when your configuration is different from default - in other words, if you run CORE and VIEW on different boxes.

The VIEW module does not have any configurable elements. It uses PnP to discover running CORE instances, connects to them and uses the capabilities discovery protocol features to

configure itself.

> **Note:**
> It is possible that in the future the VIEW module will have configurable settings, but even in that case the configuration mechanism will be seamless and integrated into the VIEW module itself.

7. Advanced Configuration

Both DAC and CORE configurators, out of necessity, will be able to read and render only limited (though quite functional) subsets of configuration files. Unfortunately, all the WISYWIG tools have a common problem: they must be **explicitly** coded to support something, whereas working with the XML configuration files will give you much more freedom.

Even now, with the minimal functionality, the configuration wizards take almost a fifth part of the DZ code base, and since the demand for configurators is still unknown at this point, all the work on them (except bugfixes and maintenance) is planned to be stopped with the `0.1p7 final` release, unless there's a significant demand for it.

As for advanced configuration - well, open the files with your favorite editor and take a look. If it doesn't make sense, revert to configurators and submit a bug report, request for enhancement or support request. Otherwise, you know what to do.

> **Note:**
> Keep in mind that the configuration wizards are unable to preserve the configuration information they don't understand, and your changes to configuration will be lost next time you run them.