

# Setting up Puppet and Fusion Inventory

BV

*[2014-05-23 Fri 08:51]*

The number of computers in my name is TOO DAMN HIGH.

## 1 What I Want

I want:

- a list all of "my" computers at BNL.
- a detailed description of characteristics of each computer
- have this be generated as automatically as possible
- be able to manually augment the description
- link to other system databases at BNL (netreg, nessus)

I started out looking into just putting up some kind of inventory management system. This led me to Fusion Inventory and from that to GLPI. I've also been meaning to move my systems over to the Puppet system that RACF supplies to the Physics department as the one I set up a while ago has been neglected and rotting. I was going to use the time needed to setup GLPI/FI as a chance to also migrate to RACF's Puppet. But then, I was pleasantly surprised that RACF already has GLPI/FI integrated into Puppet so no need for me to put up any new services!

## 2 Setting up a client

Here describes adding a Debian-like client to the system. Debian is not yet officially supported by the RACF team so some work on the service side is expected to be needed. So, this starts slowly with just getting a box

to report via GLPI/FI and with Puppet applying the `bnl_banner` module. I also put on `etckeeper` because it's cool and because it's integrated with Debian's Puppet package.

## 2.1 Installation

Install required packages:

```
# apt-get install etckeeper
# cd /etc/ && etckeeper init && etckeeper commit -m "First commit"
# apt-get install puppet fusioninventory-agent smartmontools read-edid
```

### 2.1.1 Debian Squeeze

The version of FI-agent in Debian Squeeze, proper is broken. Follow the instructions for using `squeeze-backports`:

```
# echo "deb http://backports.debian.org/debian-backports squeeze-backports main" >> /e
# apt-get update
# apt-get install -t squeeze-backports fusioninventory-agent
# apt-get install smartmontools read-edid libio-socket-ssl-perl
```

## 2.2 Introductions

### 2.2.1 Fusion Inventory

Edit `/etc/fusioninventory/agent.cfg` and make sure these are set:

```
server = https://puppet.phy.bnl.gov/glpi/plugins/fusioninventory/front/plugin_fusionin
user = inventory
password = redacted
no-ssl-check = 1
```

Note, some are turned on by default, check the whole file.

Edit or create `/etc/default/fusioninventory-agent` to have

`MODE=daemon`

One can also set this to `"cron"` but with `"daemon"` it will listen on a port and allow for remotely triggered updates and a status display from GLPI.

Run the agent explicitly:

```
# fusioninventory-agent
[info] task WakeOnLan execution not requested
```

Once it runs without errors information about the machine should show up in the Physics GLPI web site and it can then be started and left in daemon-mode:

```
# /etc/init.d/fusioninventory-agent restart
Restarting fusioninventory-agent: fusioninventory-agent.
```

### 2.2.2 GLPI setup

After the host has made an initial connection through `fusioninventory-agent`, log int to the GLPI site, select its entry from "Inventory->Computers" and assure these settings:

**Domain** `phy.bnl.gov`

**Status** `production`

Other items can be filled at this time such as

**Location** where is it? Try to maintain a hierarchy of locations, (manage via Setup->Dropdowns->Locations)

**Inventory number** BNL barcode

**Tech. in charge** who to blame

**Group in charge** who pays the one to blame

Note, one can click on "Links" tab and follow links to NegReg entry to find the barcode assuming it has already been entered there.

After changes, remember to click the **Update** button.

### 2.2.3 Puppet

Prior to registering the client with Puppet for the first time, make sure the **Domain** and **Status** is set in GLPI as above.

Add to the `[main]` section of `/etc/puppet/puppet.conf`

```
server=puppet.phy.bnl.gov
```

Then, make sure this returns a fully qualified domain name which is **not** just `NODE.bnl.gov`.

```
# factor fqdn
NODE.phy.bnl.gov
```

If this fails try:

**DHCP** edit the "search" variable in `/etc/resolv.conf` to be `phy.bnl.gov`

**Static IP** assure `/etc/hosts` has a line "`<IP> <FQDN> <NODE>`" (note, `<IP>` can be `127.0.1.1`)

Now, run a test which as a side-effect introduces the client to the puppetmaster:

```
# puppet agent --test
Info: Creating a new SSL key for casey.phy.bnl.gov
Info: csr_attributes file loading from /etc/puppet/csr_attributes.yaml
Info: Creating a new SSL certificate request for casey.phy.bnl.gov
Info: Certificate Request fingerprint (SHA256): B7:BF:E8:31:0C:29:57:C6:5F:A2:A3:33:CB
Exiting; no certificate found and waitforcert is disabled

# puppet agent --fingerprint --digest SHA1
(SHA1) 5C:02:B4:B7:64:0C:71:03:21:33:F3:8A:2E:9C:8D:A9:31:E8:28:39
```

Make sure the correct FQDN is shown.

Visit the puppetca.pl web service, **make sure the request includes the correct domain** and matches the fingerprint. Select it and click "sign" and confirm "yes".

If the request does not show up or does not show up with the correct info (FQDN, fingerprint), it may be due to a premature attempt to register locally.

```
# puppet agent --enable
# rm /var/lib/puppet/ssl/certificate_requests/<hostname>.pem
# puppet agent --test
```

After registration is successful repeat the test:

```
# puppet agent --enable
# puppet agent --test
```

This should cause many files to be copied to `/var/lib/puppet/lib/puppet/`. It may fail if no classes have yet been defined in GLPI. See the next section.

## 2.3 Configuration

Any given computer's configuration is governed by settings in its GLPI page as:

**Status** sets the git branch of Puppet manifests to use (eg, "production", "development", "testing")

**Puppet Classes** lists the Puppet classes to apply to the computer

The Puppet Classes are listing as a "Custom Field" in GLPI and will require special GLPI permissions. Start by adding:

**bnl\_banner**

Repeating the `puppet agent --test` command should now succeed.

Some Puppet classes have been written to take arguments. These are specified in the "Puppet Parameters" text box by prepending the class name to the argument. For example, the default `bnl_banner` value for the `file` argument is `/etc/motd`. To set it to `/etc/issue` one would type into the Puppet Parameters text entry:

`bnl_banner_file=/etc/issue`

Actually, this banner needs to be in three places and this can be done with an array:

`bnl_banner_file=[/etc/motd,/etc/issue,/etc/issue.net]`

Note: this example is somewhat contrived. Just having the default `/etc/motd` should suffice.

## 3 Developing Puppet modules

### 3.1 The modules

Puppet modules exist in git repositories. The Physics repository is found at:

**git** `https://puppet-phy.racf.bnl.gov/git/puppet/catalog/`

**web** `https://puppet-phy.racf.bnl.gov/cgit/puppet/catalog/`

RACF authentication is required. The `common/` directory is managed with `git subtree` to allow any commits that touch it to be incorporated into the corresponding directory in RACF's repository via a shared common repository, and vice versa. Examples of this cross-repository sharing are in `check_common_updates.sh`. Directories next door (like `astro/`) may provide Physics-specific modules which override any in `common/`. The order of these overrides is determined by Puppet's `modulepath`

### 3.2 Not developing

The common modules are well written and parameterized. One may not need to modify them in order to make them do what you want. Instead, read the manifests to see what and how parameters are used.

For example, Ubuntu 14.04 desktops use `dnsmasq` instead of directly finding DNS servers via `/etc/resolve.conf`. Reading `desktop/manifests/init.pp` one finds the use of `desktop_use_dnsmasq` switching between a `dns::dnsmasq` and the base `dns` module. If the full `desktop` class is to be used, asserting `desktop_use_dnsmasq` in the GLPI Puppet Parameters entry would cause `dnsmasq` to be used. Or, just to try out that one class one can write in the entry: `dns::dnsmasq`.

### 3.3 Testing

To test a client using the GLPI as the External Node Classifier (ENC) edit the "Puppet Classes" and "Puppet Parameters" entries to taste and run:

```
# puppet agent --test
```

### 3.4 Local development

You can run Puppet locally w/out using GLPI as the ENC. To do this checkout a copy of the Puppet manifests

```
# cd /path/to/work
# git clone https://puppet-phy.racf.bnl.gov/git/puppet/catalog/
```

One needs a way to "inject" the "Puppet Classes" and "Puppet Parameters" that are normally set in GLPI. To do this create a simple Puppet manifests that wraps these settings, the `site.pp` file and which includes the desired modules.

```
# cat << EOF > test.pp
import "/path/to/work/site.pp"

$dns_domain = 'phy.bnl.gov'
$desktop_display_manager='lightdm'
$desktop_use_dnsmasq=1
$desktop_disable_kerberos=1

include desktop
```

To run:

```
# puppet apply --debug --modulepath='pwd'/common 'pwd'/test.pp
```

## **3.5 Useful tips**

### **3.5.1 More feedback**

Include `--verbose` and/or `--debug` options to `puppet apply` or `puppet agent --test`.

### **3.5.2 Not testing**

Include `--noop` to test without actually applying anything. Note, this can cause some false errors if, for example, a script must be created before it is (not) run.