Puppet Configuration - GUI

[**Introduction:**](#_yjh1f5d02pw1) **1**

[The yum group command:](#_yvmsn7boacct) 1

[**The yum Puppet Forge Module:**](#_jofvy7762a8) **2**

[**Creating a wrapper module for GUI Desktop**](#_1gje65cb95i0) **2**

# Introduction:

This document explains how to create and configure a Puppet module to install the GUI desktop. By changing a parameter in the init.pp file you can either enable or disable the GUI at boot. The module will even try to remove the software if you like. In order to do this you **would** need to use:

|  |
| --- |
| yum group install |

The command above is used to install software collections.

## The yum group command:

On **CentOS/RHEL**, you can either install packages individually or install multiple packages in a single operation in a group. Package group contain packages that perform related tasks such as development tools, web server (for example LAMP or LEMP), desktop (a minimal desktop that can as well be employed as a thin client) and many more.

You can get a list of the packages available using:

|  |
| --- |
| yum grouplist |

You can get an even bigger list using:

|  |
| --- |
| yum grouplist hidden |

The package we are going to install is:

|  |
| --- |
| GNOME Desktop |

If we were working from the command line we would issue:

**DO NOT DO THIS:**

|  |
| --- |
| yum grouplist ‘GNOME Desktop’ |

# The yum Puppet Forge Module:

The stock version of **Puppet** **does not** support “**yum group**” commands. But there is a **community Puppet Module** that will handle this for us. The Puppet Module is **called** **“yum”.**

There is a document that covers community modules:

|  |
| --- |
| [The Puppet Forge - Community Modules:](https://docs.google.com/document/d/1HywiZtcQHdTMwl_XF0TFtzfZTKHuSdvbc5vq219N4JU/edit?usp=sharing) |

# 

|  |
| --- |
| For more information see the HTC Class document about using [Puppet Forge](https://docs.google.com/document/d/1HywiZtcQHdTMwl_XF0TFtzfZTKHuSdvbc5vq219N4JU/edit?usp=sharing), in particular, about the  [yum](https://docs.google.com/document/d/1HywiZtcQHdTMwl_XF0TFtzfZTKHuSdvbc5vq219N4JU/edit#heading=h.w6th734ysq8p) community module or go directly to the [Puppet Forge yum module](https://forge.puppet.com/puppet/yum). |

You can install the Community Puppet Module using the command:

|  |
| --- |
| cd /etc/puppetlabs/code/environments/production/modules  puppet module install puppet-yum --version 2.2.1 |

# Creating a wrapper module for GUI Desktop

Now we will create a Puppet Module to install the GUI. We do the usual commands to create a module:

|  |
| --- |
| cd /etc/puppetlabs/code/environments/production/modules  puppet module generate htcclass-gnome\_desktop |

Now modify the **init.pp** file and add the following code. The code uses new features of the Puppet language:

* **define**
* **Enum**
* **case**

|  |
| --- |
| class gnome\_desktop {  }  define gnome::desktop (  Enum['present', 'installed', 'absent', 'purged'] $ensure = 'present',  ) {  notify{'gnome\_desktop': message => "Module: GNOME Desktop"}  include yum  yum::group { 'GNOME Desktop':  ensure => "$ensure",  timeout => 900,  }  file { 'gnome\_login\_screen':  path => '/etc/dconf/db/gdm.d/00-login-screen',  ensure => 'present',  } ->  file\_line { 'org\_gnome\_login\_screen':  path => '/etc/dconf/db/gdm.d/00-login-screen',  line => '[org/gnome/login-screen]',  } ->  file\_line { 'disable\_user\_list':  path => '/etc/dconf/db/gdm.d/00-login-screen',  line => 'disable-user-list=true',  after => '\[org/gnome/login-screen\]',  multiple => false,  notify => Exec['dconf'],  }  exec {'dconf':,  command => '/usr/bin/dconf update',  }  case $ensure {  'present', 'installed', default: {  file { "/etc/systemd/system/default.target":  ensure => link,  target => '/lib/systemd/system/runlevel5.target',  }  }  'absent', 'purged': {  file { "/etc/systemd/system/default.target":  ensure => link,  target => '/lib/systemd/system/runlevel3.target',  }  }  }  } |

Now we use this definition to create an instance of gnome::desktop. This is very similar to what happens when you use the Puppet commands; file, package, service.

**Modify:**

|  |
| --- |
| modules/role/manifests/init.pp |

And the following to the role::htcXXX class:

|  |
| --- |
| include gnome\_desktop  gnome::desktop{"present": ensure => "present"} |

Your code should look similar to:

|  |
| --- |
| include motd  include autofs  include sshkey  include condor  include gnome\_desktop  gnome::desktop{"present": ensure => "present"} |

Now on the Puppet Client run:

|  |
| --- |
| puppet agent -t |

It may take some time to finish. It **may** even **timeout** during the installation. If it times out, **try running puppet agent again**. If you are able to run **puppet agent** **without** any **errors**, you are ready to **reboot** the computer and have a **GUI workstation**.