

Supporting Puppet External Nodes

[Puppet](#) is an open source data center automation and configuration management framework. Puppet provides system administrators with a simplified platform that allows for consistent, transparent, and flexible systems management.

A Puppet 'puppetmaster' host can query Ubersmith for configuration details used during a puppet client deployment. This functionality is referred to within the context of Puppet as an '[external node](#)'. You can define the Puppet modules that are available for a device, as well as its environment. Ubersmith will also pass along the device's Custom Device Data as Puppet parameters.

Configuration of Puppet is outside the scope of this document, however once you have your puppetmaster host up and running, you can follow these instructions to get it communicating with Ubersmith.

Puppet Node Control Device Module

For every device type within the Device Manager that you would like to specify Puppet classes and parameters for, you will need to configure the 'Puppet Node Control' device module. This module is configured like any other device module within the system. Configuration of device modules is done from the 'Device Types' section of *Settings*. Select the device type group or individual device type(s) that you'd like to configure the module for, and click the 'modules' link. From this page, click the 'add module' link. This will load a popup. The 'Details' tab should be configured with 'Enabled' to 'Yes', 'Module' to 'Puppet Node Control', and 'Client Access' to 'No Access'. The Config tab is a bit more complicated.

The 'Available Classes/Modules' section allows you to define the classes and [modules](#) known to your puppetmaster, and available for application to a puppet client. The format for an entry is as follows:

```
module_name:Module Name
```

To the left of the colon is the actual module/class as you would use in a standard Puppet node definition. To the right of the colon is a 'friendly name' that will appear in Ubersmith and will be easier for your technicians and administrators to work with. The value on the right can be anything you like, but the value on the left **MUST** be a valid puppet module /class.

The 'Available Environments' section allows you to specify your [puppetmaster environments](#). In most cases, this may simply be:

```
production:Production
```

The naming convention here is the same as above for modules. To the left of the colon is the actual environment name as you would use in a node definition. To the right of the colon is a 'friendly name'.

Once this module is configured, you can then set up devices in the Device Manager to take advantage of it. By default, Ubersmith will provide an empty YAML node definition with no classes, and the custom data fields and values as parameters. If you edit the configuration for the Puppet Node Control module, you'll be able to select the applicable classes for that device, as well as the appropriate environment.

NOTE: The puppetmaster uses the host's fully qualified domain name (FQDN) -- in most cases, the output of the 'hostname' on the system to query Ubersmith. You will want to ensure that the Device Manager entry has that hostname configured as the device label in order for the configuration to be pulled properly from Ubersmith.

On the Puppetmaster

On your puppetmaster host, place the file '`external_nodes.php`' (available in the 'Downloads' section of the Ubersmith Knowledge Base) in a secure location. In the example below, we've selected

```
/opt/puppet-custom
```

This is a directory we created specifically for this purpose. Note that your puppetmaster host will have to have the PHP command line interface (CLI) as well as the PHP curl module installed in order for the external nodes script to function properly.

Edit the

[main]

section of your `puppet.conf` file to include the following lines:

```
# External node definitions found here
external_nodes = /opt/puppet-custom/external_nodes.php
node_terminus = exec
```

The external nodes script queries the Ubersmith API, and Ubersmith returns a YAML statement which the puppetmaster uses as the node definition.

That's it for your puppetmaster configuration! Your puppetmaster host will now query Ubersmith for the details for any node details. **NOTE:** The puppetmaster uses the host's fully qualified domain name (FQDN) -- in most cases, the output of 'hostname' to query Ubersmith. You will want to ensure that the Device Manager entry has that hostname configured as the device label in order for the configuration to be pulled properly from Ubersmith.