metasphinx Documentation

Release 0.1

Morten Bo Nielsen

September 13, 2015

Contents

1 Control structure **3**
- 1.1 foreach . . . 3
- 1.2 if and else . . . 4
- 1.3 Troubleshooting . . . 4

2 Configuration files **5**
- 2.1 Pound sign . . . 5
- 2.2 Defining variables . . . 5
- 2.3 Commands . . . 5

3 Config variables **7**
- 3.1 if and else . . . 7

4 Config files **9**
- 4.1 import . . . 9
- 4.2 modify . . . 9

5 Special commands and variables **11**
- 5.1 unix users . . . 11
- 5.2 file ownership and permissions . . . 11
- 5.3 Variables . . . 11
- 5.4 Triggers . . . 12
- 5.5 Unix users and groups . . . 12

6 Apt **13**
- 6.1 Adding and removing packages . . . 13
- 6.2 Adding apt keys . . . 13

7 Howto **15**
- 7.1 Setting static IP . . . 15

8 Stuff missing **17**

This documentation is all about metaconfig

CHAPTER 1

Control structure

Control structures and how they are used in data files

1.1 foreach

The minimal example

in _config_

```
[settings]
nameservers = ["192.168.1.1", "8.8.8.8"]
```

in an included file, say files/etc/resolv.conf

```
%header("##")

#foreach ns in nameservers
   nameservers = %(ns)
#end
```

A foreach example using a dictionary

```
crontab["example"] = dict(
             enabled = 0,
             min     = "0",
             hour    = "*",
             day     = "*",
             month   = "*",
             weekday = "*",
             user    = "nobody",
             command = "echo foo"
         )
```

using the variable

```
%header("##")
SHELL="%(shell)"
PATH="%(path)"
MAILTO="%(mailto)"
%# min  hour  day  month  weekday  user    command
#foreach ct in crontab
  #if crontab[ct]["enabled"]
      %(crontab[ct]["min"])    %(crontab[ct]["hour"])     %(crontab[ct]["day"])    %(crontab[ct]["mon
```

```
  #end
#end
```

1.2 if and else

if-else branching is done like this

```
[settings]
enable_logging = 1
```

in an included file, say files/etc/resolv.conf

```
%header("##")

#if enable_logging
   logging = 1
#else
   logging = 0
#end
```

or, if _logging_ is default disabled, leave out the explicit _logging_ line.

```
%header("##")

#if enable_logging
   logging = 1
#end
```

To test if a dictionary contains a specific key

```
#if "max_days" in params
   SomethignWithDays = %(params["max_days"])
#end
```

1.3 Troubleshooting

```
# metaconfig
[E] Lexer error at (<some file name>: line n, column m): Command token expected but EOL found
[F] Too many errors, aborting compilation
```

CHAPTER 2

Configuration files

As part of the configuration, it is possible to add files.

It must be place in the files subdirectory

example from the ntp package

```
$ tree ntp
ntp
-- config
-- files
    -- etc
        -- ntp.conf
```

in the files adding %header("##") adds the metaconfig warning header.

The "##" part is the comment signs to use. For some config files ";" or "–" would be appropriate

2.1 Pound sign

To show the sign # in a file, use %#.

To have a comment in the file parsed by metaconfig, use ##

2.2 Defining variables

define a variable to be used later in the config

example usage

```
#define vname "drupal.example.org"
SomeVar = %(vname)
```

2.3 Commands

define, import, method

CHAPTER 3

Config variables

In the config files different variable type are allowed.

example

```
[settings]
AnInteger = 12
AString = "This is a string"
AnArray = [1,2,3,4,5]
ADictionary = dict( a = 2, b = "C" )
```

in an included file, say files/etc/resolv.conf

```
%header("##")

#foreach ns in nameservers
   nameservers = %(ns)
#end
```

3.1 if and else

if-else branching is done like this

```
[settings]
enable_logging = 1
```

in an included file, say files/etc/resolv.conf

```
%header("##")

#if enable_logging
   logging = 1
#else
   logging = 0
#end
```

or, if _logging_ is default disabled, leave out the explicit _logging_ line.

```
%header("##")

#if enable_logging
   logging = 1
#end
```

CHAPTER 4

Config files

As part of the configuration, for all nodes and modules, a file named _config_ is used.

4.1 import

Imports a module from /etc/metaconfig/res

Changing setting from e.g. config/network is done in the namespace [settings.config/network]

TODO: quick example

4.2 modify

TODO: quicl example and explanation

CHAPTER 5

Special commands and variables

5.1 unix users

TDB

5.2 file ownership and permissions

(from profile/minimal)

Files permissions and ownership can be explicitly set.

```
[files]
# Set executable permission for standard dirs for binaries
permission["/etc/initramfs-tools/hooks/*"] = "0755"
permission["/etc/init.d/*"] = "0755"
permission["/etc/cron.hourly/*"] = "0755"
permission["/etc/cron.daily/*"] = "0755"
permission["/etc/cron.weekly/*"] = "0755"
permission["/etc/cron.monthly/*"] = "0755"
permission["/usr/local/bin/*"] = "0755"
permission["/root"] = "0700"
```

5.3 Variables

Example of build-in variables

```
from profile/standard
if PLATFORM_OS_NAME in ["debian", "ubuntu"]
  import "public/config/debian-logrotate"
  import "public/config/kernel-img"
  import "public/config/hostname"
  import "public/config/locale"
end
```

TODO: which one are there and how to find out

5.4 Triggers

(from config/backup)

Triggers are used to run certain commands when metaconfig change files.

```
[trigger.generate-backup-keys]
command = "DIR=/etc/spye/backup/main/ssh; PRIV=$DIR/private.key; PUB=$DIR/public.key; mkdir -p $DIR;
single = True
```

TODO: single = true? what does that mean?

```
[trigger.restart-exim]
command = ["service", "exim4", "restart"]
files += "/etc/exim4/*"
files += "/etc/exim4/conf.d/*/*"
```

5.5 Unix users and groups

(from mozrepo/config/sysuser)

Metaconfig can enforce uid and gid values, as well as the ohter paramteres for a given user.

```
[unixuser.sysuser]
uid = 1500
gid = 1500
gecos = ",,,"
home = "/home/sysuser"
shell = "/bin/bash"

groups_include += "adm"
groups_include += "sudo"
groups_include += "sysuser"

[unixgroup.sysuser]
gid=1500
```

CHAPTER 6

Apt

Apt is the pacakge handling system of debian and its derivatives, like ubuntu.

TODO: add ref

6.1 Adding and removing packages

In the config file, instruct metaconfig to install certain pacakages, and to remove certain packages (if they are installed).

```
[apt]
install += "somepackage"
install += ["a", "list", "of", "packages"]

remove += "Someotherpackage"
remove += ["a", "list", "of", "packages", "no", "to", "be", "on", "the", "system"]
```

If bad package names are used, metaconfig will return with an error.

Note that, before installing, metaconfig will update the repositories (as in run "apt-get update")

6.2 Adding apt keys

Apt repositories has signing keys, and metaconfig will normalæy fail if packages cannot be validated.

```
[apt.source.debian_mozilla]
url = "http://mozilla.debian.net/"
dist = "wheezy-backports"
components = ["iceweasel-esr"]
keys += "A6AA8C72"
```

(from Metaconfig_mozrepo/service/iceweasel_latest_esr/config)

TODO: how to find these keys? or a link to some debian documentation related to this.

CHAPTER 7

Howto

The follwing contains metaconfig snippets for doing specific things.

7.1 Setting static IP

The following will set a static IP on interface eth0.

```
import "mozrepo/config/networks"

[settings.mozrepo/config/network]
iface["eth0"] = dict( type="static", address="10.0.0.123", netmask="255.255.0.0", gateway="10.0.0.1"
```

This is work-in-progress

CHAPTER 8

Stuff missing

- Examples/howtos
- Adaptation of obsolete pdf
- **rundown of**
  - Settings and template files
  - import/modify
  - apt
  - unixuser
- contact info
- FAQ
- some sort of overview and proper introduction
- installation instructions
- Troubleshooting