# Beyond the file Component

Stephen Quinney
<squinney@inf.ed.ac.uk>

# Why do we need to manage more than just configuration file contents?

# Possible Reasons

- Stop/Start daemons
- Packages
- Improve usability
- Clarify intentions
- Validation of resources
- Allow components to compute the “howto”
- Manage relationships

# Example 1 - client

- Special component which contains a daemon which watches for profile changes.
- Calls other components when the relevant resources have been changed.

# Example 2 - nsswitch

- Very simple component
- Only manages /etc/nsswitch.conf

# Parts of a component - Server

- Schema files
  - */usr/lib/lcfg/defaults/server/*
- Header files
  - */var/lcfg/conf/server/include*

# Parts of a component - Client

- The component script
  - */usr/lib/lcfg/components/*
- Templates
  - */usr/lib/lcfg/conf/*
- Log files
  - */var/lcfg/log/*
- Manual pages
  - e.g. “man lcfg-client”

# LCFG tools - qxprof

- View the current resources for a component
- Useful options:
  - *-v* Full info including **where** a resource was set
  - *-a* See all resources include those which are null

# *Hands On* - nsswitch

1) Look at constituent parts in server and client
2) qxprof nsswitch – not present in profile
3) View /etc/nsswitch.conf
4) *example2a* -> localhost
5) qxprof nsswitch – now it's there
6) View /etc/nsswitch.conf – no change
7) om nsswitch start
8) View /etc/nsswitch.conf – now changed

```
/* example2a – nsswitch – source profile */

#include <local/site.h>
#include <lcfg/os/minimal.h>
#include <lcfg/hw/vmware_ws5.h>
#include <lcfg/options/lcfg-server.h>
#include <lcfg/defaults/nsswitch.h>

/* eof */
```

# *Hands On* - nsswitch

1) Try “getent passwd root”
2) Try “getent passwd squinney”
3) Want to add LDAP access for passwd,group:
    i. Modify some nsswitch resources
    ii. example2b -> localhost
    iii. Watch changes in server, client and nsswitch logs
    iv. View /etc/nsswitch.conf

```
/* example2b - nsswitch - source profile */

#include <local/site.h>
#include <lcfg/os/minimal.h>
#include <lcfg/hw/vmware_ws5.h>
#include <lcfg/options/lcfg-server.h>
#include <lcfg/defaults/nsswitch.h>

!nsswitch.mods_group  mEXTRA(ldap)
!nsswitch.mods_passwd mEXTRA(ldap)

/* eof */
```

# LCFG tools – Status pages

- http://localhost/
- Allows browsing of:
  - System status
  - Inventory information (if enabled)

# Example 3 - logserver

- Provides ability to browse host information:
  - Log files
  - Component documentation
  - Current resource settings

# *Hands On* - logserver

1) example3 -> localhost
2) om logserver start
3) “ps aux | grep logserver” - now running
4) Browse online status and docs

```
/* example3 - logserver */

#include <local/site.h>
#include <lcfg/os/minimal.h>
#include <lcfg/hw/vmware_ws5.h>
#include <lcfg/options/lcfg-server.h>
#include <lcfg/defaults/nsswitch.h>
#include <lcfg/defaults/logserver.h>
#include <lcfg/defaults/inv.h>

/* eof */
```

# Example 4 - xfree

- Used to manage /etc/X11/xorg.conf
- Uses a template shipped as part of the package
- Can be configured to merge resources from the profile with data detected from the hardware, i.e.
  - monitor
  - video card

# Example 5 - updaterpms

- Utilises the special *profile.packages* resource
- Can be run manually or regularly via cron
- Add, removes and upgrades packages according to the package list.

# Example 6 - DHCP

- Client MAC address specified in client source profile (along with other information such as that for PXE booting).
- Data automatically gets added to the profile of the relevant DHCP server

# Example 7 - openssh

- Manages the files */etc/ssh/sshd_config* and */etc/ssh/ssh_config*
- Starts, stops, restarts the daemon
- Daemon is automatically restarted after configuration.

# *Hands On* - openssh

1) example7a -> localhost
2) View /etc/ssh/sshd_config & /etc/ssh/ssh_config
3) Look at status pages (note cross as not started)
4) om openssh start
5) View /etc/ssh/sshd_config & /etc/ssh/ssh_config
6) Try stop, start, restart methods – check the sshd process really did what you wanted

```
/* example 7a - openssh */

#include <local/site.h>
#include <lcfg/os/minimal.h>
#include <lcfg/hw/vmware_ws5.h>
#include <lcfg/options/lcfg-server.h>
#include <lcfg/defaults/nsswitch.h>
#include <lcfg/defaults/logserver.h>
#include <lcfg/defaults/inv.h>
#include <lcfg/defaults/openssh.h>

/* eof */
```

# *Hands On* - openssh

- Return to the port changing example
  1) “ps aux | grep sshd”
  2) example7b -> localhost
  3) Check server, client and openssh logs
  4) “ps aux | grep sshd” - did daemon restart?

```
/* example 7b - openssh */

#include <local/site.h>
#include <lcfg/os/minimal.h>
#include <lcfg/hw/vmware_ws5.h>
#include <lcfg/options/lcfg-server.h>
#include <lcfg/defaults/nsswitch.h>
#include <lcfg/defaults/logserver.h>
#include <lcfg/defaults/inv.h>
#include <lcfg/defaults/openssh.h>

!openssh.sshdopts mADD(Port)
openssh.sshdopt_Port   222

/* eof */
```

# Conclusion

- Reasons to go beyond just managing files:
  - Stop/Start daemons
  - Packages
  - Improve usability
  - Clarify intentions
  - Validation of resources
  - Allow components to compute the “howto”
  - Manage relationships