Scientific Linux 7 LCFG port diary
A diary on progress of the LCFG port to SL7

Running daily tasks

Posted on April 1, 2015 by squinney

As we replaced the boot component with systemd in EL7 we lost the ability to schedule daily tasks using the `boot.run` resources. This has now been replaced with the specialist *runner* component. I have some plans for how we can make this new component a lot more useful, particularly with machines which we want to sleep whenev idle, but for now it is just a more or less straight replacement for the boot component run method. This has bee running for a few weeks on our EL7 machines and happily managing the daily updaterpms task. Documentatio now available on the LCFG wiki with examples of how it can be used. I also took the chance to document the ol boot component run method since it looks like that was lacking a basic users guide.

Posted in Uncategorized | Leave a comment |

Logging under EL7

Posted on March 12, 2015 by Alastair Scobie

In the *good old days*, prior to EL7 and systemd, the syslog daemon (and, later, the rsyslog daemon) would listen the unix socket /dev/log for messages sent from daemons. The syslog daemon would then decide on which messages to record on the console, to local text files or to a remote syslog host.

With the introduction of systemd, an additional logging daemon has been added to the mix – journald. Journal provides much of the functionality of syslog – eg listening for messages from daemons – but it also adds the abil to receive :-

- Structured system log messages via the native Journal API
- Standard output and standard error of system services
- Audit records, via the audit subsystem

Journald stores messages in structured, indexed binary journals rather than in text files. The authors argue that this makes it easier to make queries of local log files. Whether that is true or not, one definite advantage of journald is that it can create per-user journal files. Many daemons, eg sshd, mate-session, gnome-keyring-daem etc, log user specific information to /dev/log : this was previously unavailable to the user as the syslog file was protected. Per-user journal files allow individual users to read log entries specific to their account.

The syslog daemon has not disappeared from the scene. Journald does not, yet, have the ability to forward messages to remote logging hosts. Under EL7, journald listens on the /dev/log socket, stores messages in its journals and then passes on the messages to syslog to process. Syslog can then forward messages to remote logg hosts.

For LCFG, we have decided to minimally change the syslog configuration. Log messages will continue to be logg to text files in /var/lcfg/log, but this will be in addition to the journald journals. This gives us some time to bec used to querying the journald journals. It may be that we may decide, in the future, to drop logging to text file

Posted in Uncategorized | Leave a comment |

internet-online.target

Posted on March 5, 2015 by Alastair Scobie

Stock EL7 (and SL7) has a target called *network-online.target*. This is used to delay services until the network interface is configured and online. Whilst this is suitable for a scenario where the network gateway is statically configured or obtained via DHCP, it is not suitable for systems which run router discovery (eg *rdisc*). On such systems, *network-online.target* is reached before any network gateway is configured, so services which require access to off-wire resources (eg *updaterpms*) can fail to start correctly.

To solve this, an extra target called *internet-online.target* has been added. This target is only reached once *netwo online.target* and any services required to provide full internet access have been started. Services that require off-wire resources to start correctly should depend on this target (using '*after*') instead of the *network-online.tar*

Posted in Uncategorized | Leave a comment |

New installer feature

Posted on March 5, 2015 by squinney

One of the problems we've encountered with systemd is the handling of interaction with the user on the console particular, we have always had the kerberos component run the kdcregister tool after the final reboot of the ins process. This kdcregister tool asks the user to authenticate with their admin principle and then uses that to add hostclient for the machine into the keytab. Although possible when booting with systemd the problem comes fr having lots of components starting simultaneously and all writing to the console, it becomes very difficult to spo that an interactive prompt is waiting for user input! The solution was to move this registration step from after t final reboot to before the end of the initial install phase.

Similarly we want to keep using the same SSH host keys even after a reinstall. Consequently we needed to run local script to restore the SSH host keys from our *wallet* repository. This has to happen before the first time the daemon is started or it will generate new keys which are not overwritten by our scripts. Previously this was quit simple as the ssh daemon was managed directly by the LCFG component but it is now managed by systemd. Ag the simplest solution is to move this step to the end of the initial install phase.

One thing that has become clear with the move to systemd on EL7 is that we have lots of this type of Informatic specific configuration which has to be done during the install process. If we move all that to the LCFG installer then we necessarily have to add lots of local scripts and packages to the installer ISO image (referred to as the *installroot*). This is not really manageable and requires other sites to carry lots of software which they don't nee It's also not very flexible, if we decide to add new scripts to the install process we end up having to rebuild the ISOs and update the PXE installer.

An alternative solution is to use the software installed onto the machine in the first phase (referred to as the *installbase*. The list of packages for the *installbase* comes from an LCFG profile and, although there is a default profile, each site typically has their own. This is much more dynamic and flexible, we can add any local package we like to our local *installbase* profiles and change them whenever necessary.

To actually use the software recently installed on the machine at this stage requires a *chroot* call as the new roo

partition is mounted as `/root`. Rather than hardcode all this into the existing `install` component I added a new `baseinstall` component. This is designed to work just like the main `install` component but uses software from th *installbase*. It does various things to sanitise the calling environment (e.g. setting the `PATH` variable and tweaking the tty settings). Commands are called in a full shell so that it's possible to use all the features you might need. example:

```
!baseinstall.installmethods     mADD(kdcr)
baseinstall.imethod_kdcr        %oneshot% /usr/bin/kdcregister_wrapper -f -a -r <%kerberos.realm%>
```

As with the `install` component, the `%oneshot%` indicates a script is to be called and otherwise it's assumed to be a LCFG component which should be passed to `om`.

Another advantage of this approach is that the code does not need to be modified to deal with an alternative ro Previously many LCFG components have had to have an extra `install` method which deals with the new config being stored in `/root`. With the baseinstall component the standard `configure` method can be called without modifications.

By default the baseinstall component is included only on EL7 but it has no methods. This should work on SL6 b so far it has only really been tested on EL7.

Posted in Uncategorized | Leave a comment |

openafs client changes

Posted on March 5, 2015 by squinney

As part of the move to EL7 we are trying to use systemd where possible to manage daemons rather than doing s through LCFG components. One service we want to move over to this model is openafs. Sadly the existing open component wasn't really playing along and the changes necessary where likely to be extensive and incompatible with what was required to manage the daemon via upstart on SL6.

We've had a longstanding desire to make the management of openafs much simpler by splitting the client and server parts of this component into two separate components. With this in mind I have split out the client functionality into a new openafs_client component. As it shares some templates it's still part of the lcfg-openafs source package but the files are in a separate sub-package so that we do not need to install it on servers which a not clients.

This new component does not make any attempt to start or stop the daemon when the component is started or stopped. That is now left to the init-system. It will still restart the daemon when appropriate to apply changes.

At the same time I took the chance to make the configuration changes more dynamic by using more of the functions provided by the `AFS::Command::FS` Perl module. This module helpfully provides a much more complete s of functions compared to the old `AFS` module. There should now be much less need to reboot to apply changes t running openafs client configuration.

The new component also tries very hard not to *Fail* if problems occur. It will now log errors and may not compl the configuration but the component will still be in a started state and ready to recieve and apply further chang

which will rectify the sitation. This is a problem we see with many components, we are slowly working through them and restructuring the code to ensure they start at boot time whenever possible.

Posted in Uncategorized | Leave a comment |

Service function improvements

Posted on March 5, 2015 by squinney

In May 2014 version 1.6.0 of lcfg-ngeneric was released which provided a new *Service* function for LCFG components (see this blog article for more details). The aim of this new feature was to make it possible to call daemon methods (e.g. stop, start, restart) in a platform-independent manner.

Originally the functionality in the Shell and Perl versions was implemented separately. This led to there being some inconsistencies in behaviour (e.g. only the Perl version supported setting timeouts). It also meant that add new features was more difficult than necessary since everything had to be implemented twice.

Recently (version 1.15.0 of lcfg-ngeneric) we have revisited this code and split out the Perl implementation fron the `LCFG::Component` module into a separate `LCFG::Utils::Service` module. This new module can be used in an entirely standalone manner from any Perl module or script. The new module has been designed to work in a similar way to the `LCFG::Om::Command` module. Here are a few examples:

```
use LCFG::Utils::Service;

my ( $status, $stdout, $stderr ) =
    LCFG::Utils::Service::Run( "crond", "status" );

my $status =
    LCFG::Utils::Service::Run( "sshd", "stop" );

LCFG::Utils::Service::Run( "httpd", "start",
                           { timeout => 20 } );
```

Alongside this new module has been added a wrapper script – `lcfg-service` which can be used from any script or the command line to call the daemon methods in a platform-independent way. This script is now used for the Service function in ngeneric. It supports setting a timeout, will exit with the status returned by the daemon met and also helpfully passes through any output to stderr or stdout so they can be captured in the calling script (fo example, to redirect into a component log file).

To avoid some problems with timeout handling (see Bug#57277) we have raised the minimum required versio the `IPC::Run` to 0.91.

The actual API of the Service method in ngeneric is unchanged by the addition of this module and script. Howe it's worth noting that a couple of small changes in behaviour were made. There is now a default timeout of 10 minutes, in nearly all cases this will be more than adequate and should help avoid machines becoming hung. Al for systemd, when a *stop* action is being called the `--no-block` option will be added to the arguments list. This avoids deadlock problems when a machine is being shut down. Without this we see problems when both system and an LCFG component request that a service be stopped. This has, in particular, affected the sshd service and associated openssh LCFG component.

Posted in Uncategorized | Leave a comment |

Dynamic image generation for login screens

Posted on February 24, 2015 by Chris Cooke

Having decided to use LightDM as the default display manager for EL7, the next problem was to control the appearance of the greeter or login screen. On SL6 we do this using custom KDM themes distributed by RPM. Fo EL7 we've provided a way of configuring the greeter appearance with LCFG.

The new `lcfg-webpic` component combines text and image resources with an HTML and CSS template into a sing image. This can then be used as the background image of a LightDM greeter screen using the `lightdm.greeterbg` resource.

The image is made using the very useful PhantomJS scriptable headless web browser. For more details of webpi see the LoginScreens page on the DICE wiki and the lcfg-webpic source and man page.

Posted in lightdm | Leave a comment |

grub2 – sanity prevails

Posted on February 18, 2015 by squinney

Back in April 2014 I wrote a new LCFG component to configure the grub2 bootloader. At the time I blogged abo the problems with restricting edit access for menu items. The issue was that once you had a list of "super users" access to BOTH editing and booting menu items was completely restricted to those users. There was no way to allow normal users to boot a particular item without also giving them the ability to edit the menu items (which really do not want to do…).

Thankfully it appears that sometime since I last looked the situation has vastly improved and sanity has prevaile Now the behaviour is that when there are super-users specified the editing and booting of menu items is restrict to those users except where a menu item is marked as *unrestricted*.

For the LCFG component this is as simple as this:

```
!grub2.users_lcfg_kernel mSET(unrestricted)
```

In the case of the standard lcfg kernel item that's now the default behaviour so normal users will always be able boot that item.

At the same time I also took the chance to slightly improve how the list of super users is specified in the grub configuration so that it is now applied to all menu items not just those managed by the LCFG component.

Posted in Uncategorized | Leave a comment |

Local firefox configuration

Posted on February 13, 2015 by squinney

For various reasons we need to apply some local configuration to firefox on DICE machines. In particular we set

the `negotiate_auth` options so that we get single-sign-on to our various Cosign protected web services. Recent testing of our SL7 desktop environment revealed that although the necessary configuration files were being generated by the LCFG ffox component the settings were not having any effect. A little investigation revealed th was because the main configuration file (`all-lcfg.js`) was being created in the top-level `/usr/lib64/firefox/` directory when it actually needed to be in `/usr/lib64/firefox/defaults/preferences/`. This was a bit puzzling as it worked fine on SL6 but then it was noticed that the versions of the component and schema were different. A lit bit of further checking revealed that although we had schema 2 we did not have the latest version (`1.6.2`) so the necessary `preferences_path` resource, which is used to specify the particular sub-directory, did not exist. Once the schema had been updated on our LCFG servers everything began working correctly and our local configuration now has the desired effect.

Posted in Uncategorized | Leave a comment |

jabber GSSAPI SASL problems

Posted on February 13, 2015 by squinney

A number of Computing Officers in Informatics are now using SL7 for their standard desktop. This is good beca it means various bugs are being flushed out of the woodwork. A recent issue we have found is that the pidgin messaging client could not talk to our local jabber server. We have GSSAPI support enabled so that we can have single-sign-on, that all works fine on SL6 but the client would not work at all on SL7. It could talk fine to other jabber servers where simple username/password was used though. The error message we have been seeing is:

```
SASL error: SASL(-7): invalid parameter supplied: Parameter error in client.c near line 961
```

There was a lot of head-scratching over this problem, after much investigation by various people we pinned it down to an issue with the `cyrus-sasl` package in SL7 (`2.1.26-17.el7`). It seems to be related to Redhat bug #984(and also bug 3480 in the cyrus bug tracker. The latest version of the package (`2.1.26-20`) taken from Fedora was built for SL7 and that works perfectly. The differences between the two versions of the package are not huge, it appears that the relevant patch is `cyrus-sasl-2.1.26-revert-upstream-080e51c7fa0421eb2f0210d34cf0ac48a228b1e9.patc` suspect we could just apply this single patch to the SRPM from SL7 but we probably want the other patches anyway so we'll just run with the rebuilt version from Fedora for now.

Posted in Uncategorized | Leave a comment |

Trusting a local CA

Posted on February 13, 2015 by squinney

For some of our services we use SSL certificates which are signed using the local university CA. On Redhat syste to get applications such as firefox to trust this CA we have always had to hackily patch and rebuild the nss packages. This has always been a rather undesirable situation since it means we need to ensure we keep up-to-c with security releases. Also, every time there is a new release the package tends to change just enough to mean patches need reworking, this all means that our solution to this problem has been rather fragile.

Thankfully Redhat have now provided a nice solution which makes it trivial to solve this problem. They have enhanced the `ca-certificates` package to provide an `update-ca-trust` script, see the Redhat announcement for ful

details. What it now boils down to is the simple process of placing your `pem` or `DER` file into the `/etc/pki/ca-trust/source/anchors/` directory and then running the `update-ca-trust` script.

To make it even easier the local `eucs-sslcerts` package will be updated to store the CA file into the correct locati and run the script from its post-install script.

As an added bonus, the `eucs-sslcerts` package will now be included in the LCFG installer so that it is possible to trust LCFG servers using https with a locally signed certificate.

Posted in Uncategorized | Leave a comment |

Describing features of a system

Posted on February 13, 2015 by squinney

Recently I have added a new concept to the sysinfo component which allows admins to describe general *feature* a system. These are simple arbitrary flags which indicate the presence (or possibly lack) of some feature on a system. The aim of the sysinfo component has always been to provide a place in which all general information c be gathered. Previously we had support for an extensible list of standard paths along with a static set of parame which describe things such as the operating system, the architecture and various inventory information. Notably there was a lack of a locally extensible way of describing other general features.

To understand why this is useful it helps to see where this requirement came from in the SL7 project. A while ba we became aware of a problem with the way we generate logrotate configuration files for LCFG components on SL7 (see bug#812). Whenever a component `configure` method is called the ngeneric framework attempts to buil an associated logrotate configuration file using a template. Annoyingly, due to the way we have the permissions on our log directories (which grant write access to the `lcfg` group) the latest version of logrotate requires the us a new `su` directive. There are various ways we could have approached this problem. We could have made this optional based on the version of the OS (e.g. on for SL7 but off for SL6) but this runs the risk of the newer versi of logrotate being added to the next minor release of SL6 (it also reduces platform-independence if not done carefully). We could also have made this optional based on the version of logrotate, this seems quite sensible bu we've learnt the hard way that Redhat like to backport software features without changing the major version (ju incrementing the package release string instead). This makes it very hard to spot that some new feature is required. So, we introduced the new sysinfo concept of *features* and the first *feature* is `logrotate_needs_su`. The ngeneric framework queries sysinfo to check for that feature and the template is then processed appropriately to include (or not) that configuration directive. A nice bonus here is that if we ever decide to stop granting write access for the `lcfg` group on that directory we can just remove the *feature* and everything will work correctly.

Further information is provided on the LCFG wiki describing the sysinfo resources and how to query sysinfo fro LCFG components and scripts.

Posted in Uncategorized | Leave a comment |

Systemd and buffered output

Posted on January 28, 2015 by Alastair Scobie

Kenny MacDonald submitted LCFG bug #799 reporting that the output of updaterpms appeared to be buffered boot time. He is correct, and this behaviour is particularly noticeable when installing a large-ish number of

substantial RPMs – it is not at all obvious that the system is doing anything productive. There is a real danger th a desktop user may decide to power-cycle/reboot his/her desktop should they encounter this behaviour.

By default, systemd units are configured to output to '*journal+console*'. This means that output is sent to both journald and to */dev/console*. For some reason, with this configuration, output to */dev/console* is buffered (multi lines). If a unit is configured to output to '*tty*' (which is equivalent to '*console*'), lines are only singly buffered. I c only guess that systemd is blocking on logging to journald, and the output to */dev/console* is blocked behind this

The only obvious solution is for lcfg-updaterpms to output just to '*tty*'. This does mean that we lose the ability to log useful error messages that are produced during the updaterpms run (by install scripts), but we haven't captured those error messages in the past.

The current solution (work-around, really) is to output to 'tty' by default, but allow buffered output to be enable by use of

#define LCFG_UPDATERPMS_BUFFERED_OUTPUT

at the top of a machine's profile. Buffered output could usefully be enabled on machines used for release testing

Posted in systemd | Leave a comment |

Which display manager? A decision... for now

Posted on January 28, 2015 by Alastair Scobie

An earlier post described the dilemma facing us with respect to the choice of display manager for EL7.

Well, the decision has been made much easier by the existence of a bug in GDM described at https://access.redhat.com/solutions/960583 and https://bugzilla.redhat.com/show_bug.cgi?id=1092274. Und certain circumstances, GDM will prompt twice for the user's username. This can easily result in a user typing his password in clear text into a field expecting a username. This is such an obvious security flaw that it is quite depressing that Redhat have sat on this bug since July 2014.

So, we will default to LightDM for at least the School of Informatics – for now.

Posted in lightdm | Leave a comment |

Sleep

Posted on January 16, 2015 by Chris Cooke

The LCFG sleep component has now been adapted for EL7 platforms. Here's a summary of the main changes.

The ever-growing init software systemd has its own sleep capabilities, replacing those of pm-utils. While pm-uti still available in EL7, its use is deprecated and it will be removed in a future release.

For the sleep component the shift from pm-utils means two changes: the command to initiate sleep changes (fr `/usr/sbin/pm-suspend` to `/usr/bin/systemctl suspend`) and a different mechanism is used to provide the `sleep.suspendcommands` and `sleep.resumecommands` hooks which run arbitrary commands at suspend-time or resume-time. Although systemd presents several possibilities for integrating the hooks, putting an LCFG-built sc

into the `/usr/lib/systemd/system-sleep` directory seems the best option, both because it's the simplest (no need to concoct any systemd configuration files) and because it's really very similar to the pm-utils method already in u (a shell script in `/usr/lib64/pm-utils/sleep.d/`). This solution also means that commands can be run one at a tim avoiding the usual aggressive parallelisation of systemd.

Since the sleep component still has to work on both SL6 and SL7 platforms it now creates whichever hook scrip applicable on the platform on which it finds itself. It uses a templating system to make the script. The LCFG templating system is also now deprecated, so the component has been altered to use the recommended Templa Toolkit.

The sleep component is affected by the advance of systemd in another way: the systemd tentacle logind has tak over management of user logins. This replaces ConsoleKit as the service used by the sleep component when making enquiries about machine idleness. ConsoleKit is queried using its DBus interface. Although logind also h a DBus API it proved more straightforward for the component to use its command line interface.

While systemd has brought a lot of change, it hasn't changed everything (yet). The kernel wake alarm is still th same, and seems to work just as well. This is the facility which is used to wake machines at a certain time – usu in order to run a particularly important cron job. The cron jobs themselves are also to be found in the same plac and in the same formats, and the sleep component recognises and parses them as correctly on EL7 as on SL6.

Giving devices the ability to wake a sleeping machine is similar too. The new OS version still uses `/proc/acpi/wak` to list devices which may be wake-enabled. However on EL7 some devices cannot be wake-enabled at all, so th sleep component now merely does its best to enable as many devices as it can, rather than insisting on enabling them all.

The new lcfg-sleep – now grandly numbered 1.0.0 – is available for EL7 machines, and for SL6 machines on the develop release. After a period of testing it will be rolled out to other SL6 machines.

Posted in Uncategorized | Tagged sleep | Leave a comment |

Which display manager to use for EL7?

Posted on January 13, 2015 by Alastair Scobie

We chose to use KDM, rather than GDM, for Scientific Linux 6 because it allowed us to tweak the behaviour of t login screen in a number of ways :

1. Change background image (used, for example, to distinguish between exam and non-exam mode in student labs)
2. Disable/enable power/shutdown/hibernate options (to stop users shutting down PCs in AV podiums)
3. Hooks to run scripts, as root, at login and logout (to change device permissions)
4. Disable user login list (we have far too many users for this to be practical)

Unfortunately, development of KDM has ceased and there is no version available for EL7, so we had to find an alternative.

We decided to investigate whether the current EL7 version of GDM provides the above required functionality. W found that 4) was simple to disable using dconf and 2) is possible using polkit configuration. We expect that

changing device permissions at user login/logout, 3), should be possible using udev configuration – we have yet confirm this. As far as we can see, 1) is not possible. Whilst it is possible to change various elements of the login screen using dconf resources, it is not possible to change the background image. The GDM login ("greeter") screen is built up using CSS elements – the background image is hard-wired to be noise-texture.png in /usr/share/gnome-shell/theme/gnome-shell.css. It appears that the only way to modify the the "greeter" screen is to modify this file owned by the gnome-shell RPM. GDM also has some "usability" issues. Many of the prompts are displayed in too small a font – impossible to change without making changes to the CSS as above. Tab completion for user name has been deprecated, which results too easily in one typing one's password in the Username box.

The LightDM display manager is available in EPEL for EL7. This provides all the required functionality listed above – an LCFG component has been produced to configure these. However, LightDM has its own problems. Gnome 3 has done away with a separate screen lock program – locking is now provided jointly by GDM, gnome-shell and gnome-session. LightDM does not itself support screen locking – later versions do, with LightLocker – so a separate screen lock program is required. We have built xscreensaver (version from Fedora18) for use with LightDM – it is not yet clear how we configure sensible defaults (eg blankscreen for screensaver) for this without modifying the RPM. Another concern with LightDM is power-management – this needs further investigation.

Summary – it appears that we can do more with GDM, under EL7, than we could under EL6. Using GDM feels like a more sustainable approach as it is the official out-the-box solution, but can we live without the ability to change the background login screen and those usability issues?

Posted in lightdm | Leave a comment |

Replacing boot.run

Posted on December 17, 2014 by squinney

With the demise of the LCFG boot component we have lost the `boot.run` facility. This was a useful place to hang jobs which needed to be run on a daily basis, for example, the daily run of updaterpms to update the installed set of software packages. It provides a very simple and convenient way of running a list of LCFG component methods and shell commands in the specified order on a daily basis at a time which is appropriate for the machine.

Although we could just replace this with a simple cron job for each task, such as updaterpms, we see this as an opportunity to enhance the facility to provide other functionality, in particular, it would be nice if we could have an anacron-like service. The anacron tool ensures that jobs run with a particular frequency (e.g. daily, weekly or monthly), to do this it keeps a timestamp of when the job was last run and then compares that to the current time to see if the interval has been exceeded. We cannot easily just use anacron as we also wish to limit when certain jobs will run based on the time of day, for instance, we do not usually want to have updaterpms run during office hours as it can inconvenience users.

It is likely that most of our regular system-maintenance cron jobs could be grouped so that they only run at certain times of day. They rarely ever need to run at a particular, specific time, just stating something like "morning", "afternoon" or "evening" might be sufficient. If the distribution of cron jobs could be reduced to something like 3 groups per day then there is the opportunity for lightly used machines to sleep for long periods rather than the

current situation where they have to wake up for cron jobs quite a lot more frequently.

At the moment I'm just at the stage of thinking about what functionality we can provide so now is the time to cl in with your thoughts.

Posted in Uncategorized | Leave a comment |

Repository changes and yum/mock config

Posted on December 8, 2014 by squinney

I have finally finished restructuring our internal packages repository so that we now have paths to the SL6 and l local packages directories which include the architecture as an element. So we now have:

```
pkgs/rpms/os/sl6/i386/{devel,inf,uoe,world,lcfg}
pkgs/rpms/os/sl6/x86_64/{devel,inf,uoe,world,lcfg}
pkgs/rpms/os/el7/x86_64/{devel,inf,uoe,world,lcfg}

pkgs/srpms/os/sl6/i386/{devel,inf,uoe,world,lcfg}
pkgs/srpms/os/sl6/x86_64/{devel,inf,uoe,world,lcfg}
pkgs/srpms/os/el7/x86_64/{devel,inf,uoe,world,lcfg}
```

It's a bit weird having the architecture in the srpms directory path but there's no better way to do this at the moment due to the limitations of the pkgsubmit tool. Ideally there would be one shared directory for all SRPMs a distro.

This makes it possible to hugely simplify our yum and mock configurations, in particular we now have a single S configuration for each "bucket" rather than one per-architecture, so the number of repository configurations required has halved. The new configuration also makes much greater use of the `$basearch` and `$releasever` yum variables rather than trying to achieve everything through LCFG resources, sometimes it's better to use the supp built-in to the standard tools…

I have also taken the opportunity to push the yum configuration for SL and EPEL repositories into the lcfg level headers so that everyone gets these automatically. In the short term this might break a few configurations but tl fixes should be simple and the long-term benefit definitely outweighs the short-term costs.

Posted in Uncategorized | Leave a comment |

Systemd – LCFG documentation

Posted on December 5, 2014 by Alastair Scobie

A start has been made on LCFG systemd documentation.

The Systemd Cookbook documents common queries / recipes.

The lcfg-systemd document is a more full description of LCFG systemd.

Posted in systemd | Leave a comment |

Systemd – fun with targets dependencies

Posted on December 5, 2014 by Alastair Scobie

If you have a target B.target which wants/requires S.service and you want S.service to start after A.target – it is sufficient to state B.target 'requires' and 'after' A.target. If you just do that, S.service will start based on its own requires/afters – usually much earlier than after A.target. You also need to state that S.service 'requires' and 'aft A.target.

Posted in systemd | Leave a comment |

Systemd – issues with overriding dependencies

Posted on December 5, 2014 by Alastair Scobie

It does not appear possible to override dependencies defined through a unit's .wants or .requires directories.

A unit can declare a 'want' or 'require' either through directives in its unit file or through links created in a .wan or .requires directory.

You can easily override dependencies specified in a unit's unit file, by creating a file of the same name in /etc w the offending directives removed. This file is simply used instead of the file in /lib. For example, unit sshd.servic might specify that it 'Wants' kerberos.service by means of a Wants directive in its unit file /lib/systemd /sshd.service. One can override this by creating an /etc/systemd/sshd.service with the offending Wants directiv removed.

One might expect that creating unit's .wants or .requires directories in /etc would override the equivalent directories in /lib, but that is not the case. For example, ntpd.service may specify that it 'Requires' network.targ by means of a symlink, in /lib/systemd/ntpd.service.requires, to the network.target unit file. One might expect you could remove this dependency by creating a directory /etc/systemd/ntpd.service.requires without the offending symlink. Unfortunately this isn't what happens – instead, systemd uses the union of /etc/systemd /ntpd.service.requires and /lib/systemd/ntpd.service.requires.

Posted in systemd | Leave a comment |

Repository management

Posted on December 3, 2014 by squinney

In Informatics we manage our local package repositories with a tool called refreshpkgs. Each package directory separate AFS volume and whenever a file is submitted the repository metadata is updated and a new read-only snapshot is created by the refreshpkgs script.

As previously noted we have changed parts of the repository layout for EL7 to include the architecture as a sub-directory in the path. The aim is to eventually make our paths more standard and drop the use of the LCFG platform name which may include an architecture specific suffix (e.g. `sl6_64`). To achieve this the scripts need re-educating, sadly the paths are currently built in various places without reference to a common function so th first step is to try and merge the code.

We also had a long-standing aim to switch entirely to using the AFS::Command Perl module instead of the badl supported AFS module. This thankfully proved to be fairly straightforward although I did find a few bugs along way which we had to patch (the patches have been sent back upstream).

As well as merging the path building functionality I have taken the opportunity to merge a lot of other common code so that it is much more maintainable.

I have also come up with a solution for a bug we occassionally encounter where a package is submitted whilst a release is in progress and the refreshpkgs tool does not spot the change. We were caching a timestamp which indicated the time of the end of the last run but this could differ from the time the AFS snapshot was created th causing a race condition. Instead we now compare the creation time of the read-only snapshot with the last upd time of the read-write copy.

Posted in Uncategorized | Leave a comment |

LCFG client and Systemd

Posted on December 3, 2014 by squinney

One of the jobs of the LCFG client is to send back status information to the LCFG server. A few people had notic that the CGI status pages were showing components with a blue box instead of the “all normal” green box. This eventually became puzzling enough that I decided to take a look at the client code. It turned out that as part of building the status information the client was inspecting the `boot.services` resource to see if it contained a reference to a component with an `lcfg_` prefix (e.g. `lcfg_cron`). As we have now switched to systemd on EL7 that was clearly not going to work so I’ve reworked it to look at the `systemd.units` resource instead. Annoyingly this h a slightly different prefix without an underscore (e.g. `lcfgcron`) but it does make the status pages appear correct again. It’s questionable as to whether it’s really the job of the LCFG client to be poking around in these resource but that’s something to think about another day…

Posted in Uncategorized | Leave a comment |

Package List management

Posted on December 3, 2014 by squinney

Although many of our package lists can now be managed using yummy we still have some which are handled using our older pkglist-tools scripts as they need special support. In particular, the updates and kernel lists don’t quite work in the necessary way for yummy to be useful.

Having made some changes to the way we store our package repositories for EL7 these older scripts no longer worked. The main issue is that for SL6 we had the packages stored in AFS but on EL7 they are stored locally on server with general access only possible via http.

To resolve this problem the list_packages and list_updates scripts in pkglist-tools have been reworked so that th same code can handle both directory and http access. In the process I switched from scanning the directory inde to using the rpmlist files which are provided for updaterpms. This makes things more efficient and means we ar working with the repository in the same way as updaterpms.

Much of what these scripts do is similar to the functionality already available in the LCFG::PkgTools suite of Per modules. To avoid unnecessary code duplication I’ve spent some time updating those Perl modules to provide some additional features.

I also took the time to fix an issue with the regular expression used in the lcfg-pkgtools C library for parsing LCI

package specifications. I was surprised to discover that this had never been properly documented so I've created new page on the LCFG wiki which has all the gory details on LCFG package specifications.

Another small benefit of all this work was that I spotted in updaterpms a hardwired default architecture of "i38 I've now reworked the code to use a macro which takes the architecture from CMake at build time. This means updaterpms is more likely to work "out of the box" without additional configuration.

Posted in Uncategorized | Leave a comment |

Display managers and the "burning issues" page

Posted on November 20, 2014 by Chris Cooke

In keeping with LCFG tradition we've added an SL7 Burning Issues page to keep track of local issues which you might want to be aware of.

The first and so far only entry is on display managers. On SL6 we use KDM, but since that's not available on SL7 we tried GDM. We found it a pig to get running (but eventually we did achieve a compatible PAM configuration and horrible to configure (but these GNOME manuals for administrators may help). Once it was up and running the interface seemed both weirdly unhelpful and distinctly fragile.

The umpteenth web search for solutions to these problems turned up a recommendation that GDM simply be binned and replaced by LightDM. So far LightDM seems robust and straightforward to use. It's also easier to configure – it eschews the fashionable "poorly documented mystery database with custom tools" approach in favour of outmoded text-based configuration files; you may remember them with fondness.

Our LightDM configuration isn't yet perfect: it's very basic, we haven't yet provided the planned lightdm component, and it seems to periodically lose its permission to run; but we're working on these problems with h in our hearts.

Posted in lightdm | Leave a comment |

Developers FAQ

Posted on November 3, 2014 by squinney

We have begun maintaining an FAQ page for LCFG SL7 development. If you have any questions that you would like adding to the list (whether or not you know the answer), please let us know.

Posted in Uncategorized | Leave a comment |

Network device naming

Posted on November 3, 2014 by squinney

Last week a couple of my colleagues installed SL7 onto desktop machines so that they could begin testing. They immediately stumbled across a problem with the naming of network devices which we had not seen before. Instead of being named like `ethX` the network interface had a completely different style of name. This resulted in the LCFG installer failing at the end of the first stage.

Traditionally network devices have been named like `ethX`. Although typically the allocation is stable it has long

been known that the kernel can allocate these randomly so a new scheme has been developed which does the naming based on attributes of the device. This naming is done via udev using a tool called `biosdevname`. Full deta are in the Chapter 8 of the RHEL7 Networking Guide.

On the VMs (both KVM and Virtualbox) and also HP desktops we had not seen this in action, we still had the ol legacy style. We were somewhat surprised to see this feature enabled on certain machines when there was seemingly no difference in configuration. The important paragraph in the RHEL7 guide is:

> *If the system has biosdevname enabled, it will be used. Note that enabling biosdevname requires passing biosdevname=1 as a command-line parameter except in the case of a Dell system, where biosdevname will be used by default as long as it is installed.*

Yes, both the machines which failed to install were Dells. I'm not a big fan of exceptions like that, probably ther a good reason but none is mentioned!

Moving on from this discovery the next challenge was to find out how to disable this feature. Setting the `biosdevname=0` kernel parameter was not sufficient. It's not particularly well stated but the way to disable this naming style is to disable the biosdevname support **and** set the `net.ifnames=0` kernel parameter. For now in LCF(for EL7 we are doing the following:

```
!grub2.default_cmdline_linux    mEXTRA(biosdevname=0 net.ifnames=0)
```

This means we consistently use the legacy naming style which is expected by the LCFG network component. We really do need to move towards the safer modern naming style but that's a challenge for another day…

Posted in Uncategorized | Leave a comment |

polkit support

Posted on November 3, 2014 by squinney

Recently I have been working on creating an LCFG component which can configure polkit ACLs. In EL7 polkit is used by quite a few applications to do authorisation checks which were previously done with PAM (for instance halt, poweroff and reboot commands). There are also applications such as NetworkManager which by default w permit a normal user to alter various settings that we wish to protect on our managed machines. With a persona laptop it is typically sensible for a user to be able to alter the network settings but with a managed desktop it's a recipe for bizarre support calls.

The new LCFG polkit component supports creating whatever ACLs are required. The ACLs use the *polkit localauthority authorization (pkla)* style which is supported by the pkla-check-authorization script. The compone can also manage the list of administrators (which is used when the result value is `auth_admin` or `auth_admin_keep`).

polkit ACLs can be applied to identities which are either users, groups or netgroups with each identity having th relevant prefix (`unix-user:`, `unix-group:` or `unix-netgroup:`). For convenience the LCFG component also supports th pam_access style where netgroups have an '`@`' prefix and groups are encased in brackets. This allows the current value of the `auth.users` resource to be mapped into ACLs without any changes.

Here's a fairly trivial policy to deny access to the various NetworkManager commands:

```
!polkit.policies                mADD(netman)
polkit.identity_netman          default
polkit.action_netman            org.freedesktop.NetworkManager.*
polkit.resultany_netman         no
polkit.resultactive_netman      no
polkit.resultinactive_netman    no
```

My main concern with this increased use of polkit is that when new applications are added we might not immediately spot that they use polkit for authorisation. If the default policy for an application is wide-open ther we could be inadvertantly allowing users to alter settings which we really want to remain fixed.

Posted in Uncategorized | Leave a comment |

Desktop support

Posted on October 29, 2014 by squinney

Basic desktop support for SL7 has now been completed. At this stage we are only providing Gnome3 which is based on the packages in the "GNOME Desktop" yum group with all optional packages included. We have gdm logins working, it needs some theming to make it look like a DICE machine, that will come later. This means it's now possible to begin work on developing the "DICE Desktop" environment.

Posted in Uncategorized | Leave a comment |

SL7 installer

Posted on October 20, 2014 by squinney

The installbase and installroot package lists and LCFG profiles have now been created so that it is possible to ins SL7 machines directly without having to go via EL7. Again, the use of yummy has made the creation of the package lists really simple and efficient. A few scripts had to be tweaked to use SysInfo resources to cope with t new "OS base" versus "OS id" situation but nothing needed to change dramatically. The PXE installer has also b built and appears to work nicely. Porting from our EL7 release-candidate based platform to the final release of S doesn't seem to be any more work than a minor platform upgrade.

Posted in Uncategorized | Leave a comment |

Package repository layout changes

Posted on October 10, 2014 by squinney

To simplify the way we build the URLs for the `updaterpms.rpmpath` resource I've altered the local package reposito layout for EL7. Where we used to have `rpms/os/<platform>/<bucket>` we now have `rpms/os/<os_base>/<arch>/<bucket` This means that `rpms/os/el7/lcfg` becomes `rpms/os/el7/x86_64/lcfg`. This brings us more into line with how upstre (and other sites) structure their repositories and does away with the compound platform name which is based o the OS and architecture (e.g. we have `sl6` which is for SL6 32bit and `sl6_64` for SL6 64bit). This makes it easier f any/all EL7 derivatives to use the same repository in the way we already do for EPEL.

This has had a few consequences for the way we manage our software builds internally but mostly it just works.

a nice bonus I've added a new `LOCAL_PACKAGE_BUCKET` cpp macro which can be used to hide the details of including local buckets. It works like:

```
LOCAL_PACKAGE_BUCKET(lcfg)
LOCAL_PACKAGE_BUCKET(devel)
```

this is handy to avoid having to know about the different layouts we have for SL6 and EL7.

Posted in Uncategorized | Leave a comment |

Handling EL7 and SL7

Posted on October 10, 2014 by squinney

For the new EL7/SL7 platform we've been working on recognising the fact that most of LCFG will work on EL7 any derivative (e.g. Scientific Linux 7, Centos 7, Oracle Linux7). As part of this I have introduced a new concept the "OS base" into the system information – *sysinfo*. This is intended to indicate the base OS from which all are derived, so if the platform is SL7 the base is EL7, if the platform is centos7 the base is also EL7. This is useful, fo example, for shared package lists or where there are configuration file templates which can be reused on any EL derivative. I have updated the System Information page on the LCFG wiki to reflect the changes.

Currently the following package lists are considered to work on any EL7 derivative:

- `lcfg/lcfg_el7_lcfg.rpms`
- `lcfg/lcfg_el7_devel.rpms`
- `lcfg/lcfg_el7_epel.rpms`
- `ed/ed_el7_env.rpms`

The base and desktop lists are different on each version of EL7 since any derivative needs to have all the Redhat branding removed (e.g. we have `dhclient-4.2.5-27.sl7` on SL7 and `dhclient-4.2.5-27.el7` on EL7). The number of differences are small (8 packages in the base list) so it might be that we can avoid complete duplication of the f contents with cpp conditionals but that might make things more complicated, simplicity is always good. The management of the package lists will need a bit of thought and might evolve over the lifetime of the EL7 platfor

Yet again we've seen how useful the lcfg-yummy tool is for managing big package lists. Having created the necessary templates for EL7 a while ago there wasn't too much work required to build the SL7 lists using yumm

Posted in Uncategorized | Leave a comment |

Systemd gains start_on_add resource

Posted on September 23, 2014 by Alastair Scobie

The systemd component has gained the ability to auto-start service units when they are added to the units resource, without requiring a reboot. Setting the start_on_add_{tag} resource to true will enable this functional for the unit associated with {tag}.

Posted in systemd | Leave a comment |

Mass specfile change

Posted on September 19, 2014 by squinney

As announced some time ago the specfiles for LCFG components stored in our central subversion repository hav been modified en-masse. This is to convert the post-install scripts over to the new way of testing if an LCFG component has been started. The new standard post-install method should look like this:

```
%post
if [ $1 -eq 2 ] && @LCFGOM@ @LCFG_NAME@ isstarted -q; then
  echo reconfiguring @LCFG_NAME@ component
  /usr/sbin/daemon @LCFGOM@ @LCFG_NAME@ configure
fi
exit 0
```

We have not tagged new releases for any of the components, the authors can do that when they next make changes.

Posted in Uncategorized | Leave a comment |

Changes to package mirrors

Posted on September 19, 2014 by squinney

For the last few years we have stored our mirrors of the SL6 and EPEL package repositories in AFS. This has bee useful since it allows us to access the packages via the filesystem as well as through http and rsync. It has howe recently begun to cause us a number of headaches, particularly with EPEL which is now so large we cannot stor in a single AFS directory. Partially this space problem is because we have a policy of never deleting a package fr our mirror. The upstream sites regularly delete old versions but we may have a need to keep a particular versio around for a long period of time (for example, we aim to not change important packages during a teaching session). We do not use the yum approach of always just using the latest version available. It's also useful to be able to revert to an older version if a newer version has introduced a critical bug.

Having come to the conclusion that AFS cannot be used to store the mirror for EPEL7 we decided to take the opportunity to revisit the whole way in which we handle the synchronisation of our package repositories with upstream.

In EPEL6 all the packages for an architecture are stored in a single directory which is very convenient for using with updaterpms but in EPEL7 the packages are stored in sub-directories based on the first character of the package name. If we took the previous approach we would end up having to add every single sub-directory into the `updaterpms.rpmpath` resource which would be horrible. We thus decided to create a new top-level `lcfg` sub-directory within the directory for each architecture which contains links to the real files, updaterpms could thus be pointed at a single location. The decision to use links has also inspired a better solution to keeping old versions (thanks to Kenny MacDonald). With the use of hardlinks we can ensure that a copy will always be kept the `lcfg` directory even after it has been removed upstream. This is really simple to manage and it does not wast disk space when the file exists in both locations.

Also with EPEL6 we deliberately delete all the upstream yum metadata and generate our own which includes th old versions. This is done so that the yum metadata contains the details of all the old package versions which w have kept. With a large repository this is a very time-consuming business because we end up building the files fi

scratch every time. With the new approach we can generate our own yum metadata within the `lcfg` directory ar leave the upstream metadata untouched. This makes it possible to avoid rebuilding the entire dataset every tim which saves a huge amount of time.

This means we can now provide an EPEL7 package repository which is usable from LCFG and could be used via yum by anyone who is manually maintaining EL7 machines.

Posted in Uncategorized | Leave a comment |

Systemd presets

Posted on September 18, 2014 by Alastair Scobie

Once we had managed to install EL7 machines using the LCFG installroot, we discovered that the systemd confi for nslcd, which had been created by the LCFG systemd component whilst running the installroot, was removed the first reboot. On investigation we discovered that when the nslcd RPM was being installed, the RPM postinst script was calling 'systemctl preset'.

Systemd preset files (description) are used to encode policy as to which units should be enabled by default and which should be disabled. The preset files, which live under /usr/lib/systemd/system-preset, are provided by th distribution.

Redhat's default preset policy for nslcd is 'disabled', so when 'systemctl preset' was called as part of the RPM install, systemd removed the nslcd configuration that the LCFG systemd component had previously created.

Fortunately you can disable individual preset files by creating a softlink (to /dev/null) with the same filename i /etc/systemd/system-preset.

In the LCFG world, we'll have no use for the preset mechanism as we'll be using the LCFG systemd component t control which systemd units are enabled. The LCFG systemd component can be configured to disable individual preset files.

Posted in systemd | Leave a comment |

Fun with systemd targets

Posted on September 18, 2014 by Alastair Scobie

The existing LCFG boot component allows one to configure components such that they are not started, at syste boot time, until after those components which may ask for a system reboot have completed. Simply put, the boo component starts all components with a start ID < 100. It then reboots the system if any of those components h requested a reboot. If no reboot has been requested, it then starts those components with a start ID >= 100. Th is so that user fronting services such as web, getty and ssh aren't started just for them to be taken down should reboot be requested.

We wanted to replicate this behaviour with systemd. It appeared obvious(?) that targets would assist us in doin this.

Our first attempt was to create a new target lcfg-multi-user-stable.target to be the end 'default' target, and have this require the normal multi-user.target. The following diagram shows the basic structure :-

- lcfg-multi-user.target (as end ‘default’ target)
  - multi-user.target
    - lcfg-auth.service
    - lcfg-client.service
    - ….
    - lcfg-updaterpms.service
    - getty.service
    - other stock units
  - lcfg-rebootcheck.service (after multi-user.target)
  - lcfg-clearbootctx.service (after multi-user.target)
  - lcfg-cron.service (after multi-user.target)
  - lcfg-apacheconf.service (after multi-user.target)
  - lcfg-openssh.service (after multi-user.target)

The lcfg-rebootcheck tool reboots the system if any component started whilst reaching multi-user.target has requested a reboot.

The above configuration appeared to work fine, but for the fact that getty was being started before the reboot check was being performed. From the above table, we can see that we want to start getty.service after multi-user.target (or lcfg-rebootcheck.service). Unfortunately it proved impossible to achieve this. Getty is, by default, pulled in as a “wants” of multi-user.target. This dependency is hard-coded in /usr/lib/systemd. Amazingly, whil it is possible to override a unit file in /usr/lib/systemd/system by creating a file with the same name in /etc/systemd/system, or to disable a unit file by creating a symlink of the same name to /dev/null, it is not possible to override dependency files in .wants or .requires directories.

For our second attempt, we decided to keep the default multi-user.target as the end ‘default’ target, adding new LCFG targets as “wants” of multi-user.target.

- multi-user (as end ‘default’ target)
  - lcfg-multi-user-stable
    - lcfg-multi-user
      - lcfg-auth
      - lcfg-client
      - …
      - lcfg-updaterpms
    - lcfg-rebootcheck (after lcfg-multi-user)
  - lcfg-clearbootctx (after lcfg-multi-user-stable)
  - lcfg-cron (after lcfg-multi-user-stable)
  - lcfg-apacheconf (after lcfg-multi-user-stable)
  - lcfg-openssh (after lcfg-multi-user-stable)
  - getty (after lcfg-multi-user-stable)

The above configuration achieves what we wanted.

Whilst we have worked round the inability to override dependencies in this case, we’re convinced we’ll trip up c this problem again in the future.

Posted in systemd | Leave a comment |

Poor updaterpms performance

Posted on August 13, 2014 by Alastair Scobie

Once we had RHEL7 LCFG installs working, it quickly became apparent that updaterpms performance was poor under RHEL7. In particular, the phase where updaterpms flags up RPMs for install was very slow. During this phase updaterpms is loading, over HTTP, the package meta data stored in the header files.

After some debugging (and adding timestamps to updaterpms debug output), it became clear that the problem with the library *libcurl* which updaterpms uses to HTTP fetch the header files. Measurements showed that the ti taken by *libcurl* to fetch a page have substantially increased, under RHEL7. Even an attempt at fetching a non-existent page has increased from 10-20ms to around 150ms. This with just plain HTTP.

My first thought was that the API has changed in some subtle way and the fault must be in the updaterpms cod but then I thought to try measuring the performance of the *curl* tool itself. To my surprise, I found the same performance disparity.

Interestingly…

* Adding the DNS address of the HTTP server to /etc/hosts (and specifying files,dns in /etc/nsswitch.conf) doe not improve things

* but, performance when using an IP address in the URL is fine (~ 10-20ms) for both RHEL6 and RHEL7

… which kind of suggests that the delay is in libcurl's DNS code rather than in the upstream resolver.

We did notice that when using DNS names in the URL, curl forks off a process to do the name lookup and then poll()s. There's no such fork when using an IP address in the URL. The poll() timesout after a period of time – v suspect this is the cause of the delay.

An email to the Curl mailing list resulted in a quick response from Daniel Stenberg – *"Redhat switched their build the threaded resolver and in 7.29.0 we switched some libcurl internals which made us have that performance issue"*. quick (ha!) rebuild of the curl tool and library, with the threaded resolver disabled, demonstrated that this was indeed the cause of the problem.

Daniel reckons that the problem is fixed in a later version of curl. Until RHEL ships with the fixed version (Redh bug 1130239), we will run with our local patched version.

Posted in Uncategorized | Leave a comment |

grub2 install issues resolved

Posted on July 24, 2014 by squinney

Having put a truly epic amount of effort in I finally have a grub2 LCFG component which can make a machine bootable at install time. In the end to generate a sane configuration file I had to make a copy of the standard `gr mkconfig` script which is provided as `lcfg-grub2-mkconfig`. This script adds support for a `--root` command-line optic which is used to modify the paths to files in the `/etc/grub.d/` and `/boot/grub2/` directories. It's also used to modify

the paths which are passed to the `grub2-probe` script for looking up the root and boot filesystem devices (which a exposed via the `GRUB_DEVICE` and `GRUB_DEVICE_BOOT` environment variables). Finally, to ensure a simple configuratic is generated, the script only runs two config-generation scripts – `00_header` and `41_lcfg`. This means that only kernels specified via LCFG resources will be listed as grub menu entries. This seems fairly reasonable during the install process and makes the process work in a similar way to what we have had previously with legacy grub in SL6. By setting the `grub2.mkconfig_cmd` resource back to `/usr/sbin/grub2-mkconfig` the standard behaviour can be restored.

Further to the changes in the grub2 component, changes have had to be made to the LCFG fstab component and the `mkparts` script (part of the lcfg-hackparts package). I have discovered that when using grub2 with a GPT partition layout an extra partition is required for the 2nd stage of the bootloader. The partition can be fairly sma 4MB is apparently more than sufficient, the only restriction is that it must be within the first 2TB of disk space. partition must be flagged as `bios_grub`. No filesystem type needs to be specified, although depending on the tool used to create the partition you might need to set one anyway, it doesn't matter what type you choose, grub2 w just write the data directly into the partition. The `mkparts` script has gained support for creating these partitions similar manner to how we handle `lvm` partitions. The fstab component will just ignore partitions with a type of `bios_grub` so that it doesn't get formatted or added into the `/etc/fstab` file.

This change means that we will have to completely reinstall all machines when they are upgraded to EL7. Normally we have the option of keeping the old partition layout and not touching any data stored on the disk b that won't work when we need to introduce this new partition.

We will probably add the new partition as the 2nd partition and move swap to 3 (with the rest also shuffling up Many scripts expect root (`/`) to be the first partition so we definitely don't want it there.

Posted in Uncategorized | Leave a comment |

grub2 install issues

Posted on July 21, 2014 by squinney

We have had support for configuring grub2 in EL7 for quite some time now but up until this week we had not tackled the install-time support that we require.

When installing a machine we boot either via PXE or using an ISO image. Either way we mount the root filesyst for the target system at `/root`. With the legacy grub this has never posed a problem but with the new grub2 this given us a lot of difficulties. The biggest problem is with the design of the tool which is provided to generate the configuration file (`grub2-mkconfig`). This script runs all the executable scripts stored within `/etc/grub.d` and then u the text sent to stdout to generate the configuration file `/boot/grub2/grub.cfg`. This script has hardwired paths to files in `/etc` and `/boot` which is useless for anyone with their target root mounted elsewhere.

My first thought was to try a slightly hacky approach by using the `chroot` system call. This resolved the issue wit writing to the wrong file locations but broke the `grub2-probe` script which needs to be able to poke around in `/pr` really did not want to have to mount that partition (and possibly others) inside the chroot location so that approach was dropped.

My second idea was to just hack the `grub2-mkconfig` script to take additional options. This should be a technically

simple solution but this failed due to the slightly crazy packaging of grub2 as an RPM by Redhat. They already apply a lot of patches and these have been done using git. In fact, the SRPM will only build when git is installed can find no way to apply one additional simple patch. You are forced to do it the very complicated Redhat way not at all. So, that option has now been dropped as well.

My third approach, and the most promising so far, is to create my own simple alternative script which provides functionality in the `grub2-mkconfig` script which we actually require. All it has to provide is enough capabilities to configure the system so that it is bootable after the install phase has completed. On the next boot we can rerun full script when the grub2 component is started.

As part of this work I've extended the functionality of the lcfg script which generates the menu entries to more closely match those provided by default (e.g. `10_linux`). It is now possible to use the locations and types of the r and boot filesystems which are detected by the standard grub2 probe scripts. This means it is now very easy to a an additional menu item, something like this:

```
!grub2.menuitems     mADD(linux)
!grub2.id_linux      mADD(dice)
!grub2.title_linux   mSET(DICE Linux Kernel)
!grub2.kernel_linux  mSET(/vmlinuz)
!grub2.initrd_linux  mSET(/initrd.img)
!grub2.classes_linux mSET(red gnu-linux gnu os)

!grub2.defaultboot   mSET(dice)
```

This closely resembles the way we use the legacy grub component in SL6. The filesystem UUID will be automatically added to the kernel command line and the necessary grub2 modules will be loaded.

Posted in Uncategorized | Leave a comment |

Improved locale support

Posted on July 21, 2014 by squinney

The hardware component has traditionally been used to configure the keyboard map and this has slowly evolve to include further locale configuration support. For EL7 we've extended this further to add support for configuri the default locale via the new `/etc/locale.conf` file (see `locale.conf(5)` manual page for full details).

The default behaviour is to just configure the `LANG` setting, it has been set to `en_GB.UTF-8` which should suit most users. This is done in the following way:

```
!hardware.locale               mADD(lang)
!hardware.locale_val_lang      mSET(en_GB.UTF-8)
```

Other supported variables can be configured in a similar way.

Posted in Uncategorized | Leave a comment |

LCFG installer

Posted on July 21, 2014 by Alastair Scobie

Porting the LCFG installer to a new platform has always been an “interesting” task. The port to EL7 has been n different.

The first stage of the LCFG installer runs from an initramfs image. (See the LCFG installer documentation for a explanation of what this stage does). The initramfs image is created by a tool called lcfg-mkbootpkg. This tool builds the image from RPMs using updaterpms, but with dependency checking disabled to avoid pulling in load unnecessary RPMs – the initramfs needs to be as small as possible. Unfortunately, under EL7 rpmlib now insists that an RPM owning a top-level directory such as /lib is installed before any other RPM that owns files within t directory. This is normally sorted by dependencies providing RPM install ordering, but as lcfg-mkbootpkg was n using dependencies updaterpms was not installing the RPMs in the order required to keep rpmlib happy. This affected directories owned by the filesystem rpm – the solution was to remove this rpm from the initramfs list a create any required toplevel directories by hand.

The initramfs image makes heavy use of busybox. Neither EL7 distro nor EL7 EPEL currently ship busybox, so t busybox from Fedora 20 was built and deployed. Unfortunately, the busybox developers have chosen to disable DHCP ‘userclass’ option because [some] options .. do not make sense or not supported. In Informatics, we use t ‘userclass’ option to pass the LCFG url to the LCFG installer.

LCFG SL6 uses upstart as the init subsystem for the second stage of the LCFG installer. The obvious choice for E was to use systemd. There were concerns that this might be overly heavyweight, particularly wrt. configuration, and require lots of additional software, but these concerns proved unfounded. It proved relatively straightforw after referring to the EL7 single-user-mode configuration, to produce systemd configuration for the LCFG install

Posted in Uncategorized | Leave a comment |

ngeneric changes

Posted on June 27, 2014 by squinney

Over the last month a lot of work has been done on the ngeneric (and LCFG::Component) framework which is used by LCFG components. This work has been done as part of the porting of LCFG to RHEL7. Here are outlines the work, more details will appear on the wiki in the near future. Some of these changes will not appear until version 1.11.2 of lcfg-ngeneric in the stable release next Wednesday (2nd July).

Service Command

We now provide a convenient method, named Service, for calling actions on managed daemons (e.g. apache

restart). This automatically uses the correct tool for the init daemon on the machine (e.g. upstart, systemd or SysV).

We recommend that all component authors convert their components to using this new method so that when th upgrade to EL7 it "just works". Full details at the LCFG wiki.

IsStarted Component Method

All components have gained a new method named IsStarted which can be called like "om foo isstarted". This wi return 0 (zero) when the component is running and 1 (one) when it is stopped.

Methods can specify exit code

We now allow authors to return an integer value from the method function (e.g. Method_Configure, Method_Install), if the end is reached without errors that value will be set as the exit status.

Previously there was no control over the exit code of a component method. If you reached the end it was zero, i you failed it was non-zero.

Along with this new behaviour we now have a consistent exit status of 255 if a component fails or records an er (via Fail() or Error()). Note that the Fail method aborts immediately and is rarely a good idea since it can leave component in a bad state.

We have seen a few side-effects from this change, particularly during the install process. We believe we have solved all the issues with the "core" components, we recommend that component authors who have added additional methods audit their code and check they are always returning sensible values.

Change in location of run, status and lock files

When a component is started there exists a "run file" which is used to indicate that it is "running" and a "status file" which records the state of the resources after the last successful call of the configure method. When a component method is actively doing work there is also usually a lock file to avoid concurrency issues.

These files have always been stored in directories within /var/lcfg and are all deleted by the lcfginit script befor the boot process is begun. This is rather non-standard behaviour and is awkward to handle when using systemd To improve the situation on EL7 we have moved the files to directories under /run. This is a tmpfs partition, it i now the standard location for these types of files and it is guaranteed to be empty at the start of each boot proc

This change has led to a big change in how we handle the LCFGSTATUS, LCFGRUN and LCFGLOCK paths. Thes were previously hardwired into the code when the software package was built. We now look up the paths dynamically at runtime using LCFG SysInfo.

Component authors rarely need to know the locations of these paths but in case it is necessary there are now convenience functions named RunFile and StatusFile (see also HasRunFile and HasStatusFile). Locking is now handled entirely by LCFG::Lock (or lcfglock), see that code for details if necessary.

To make life easier a new option was added to the qxprof command line tool. It is no longer necessary to specif

the full path to the status file for a component with '-r', you can now just specify the component name with '-s' a qxprof will work out the correct location.

If you need to look up standard LCFG paths and SysInfo is not available (e.g. early in the install process) you m be able to use the LCFG::Client::FileLocator module instead. The values of SysInfo resources will always reflect values returned by methods in that module. This allows component authors to avoid boot-strapping problems.

Minor Bug Fixes

A number of minor bugs were discovered in the locking code which have been resolved. It's unlikely that these were causing anyone major issues.

A number of problems were found which were due to the reuse of the same variable name in different methods the ngeneric shell code. This has been resolved with the liberal application of the 'local' function.

Posted in Uncategorized | 1 Comment |

Building the cron component

Posted on April 21, 2014 by Chris Cooke

lcfg-cron needed a number of changes to its packaging in order to get it to build to an RPM on both F20 and SL

When a perl-based LCFG component is packaged into a Linux RPM, a copy of its documentation is translated fr Perl's "pod" format into "man" format, to make it accessible from the "man" command. The conversion and packaging are done by `rpmbuild` using the `pod2man` and `podselect` utilities.

On SL6 these utilities are included in the `perl` RPM so are automatically available on any machine which has pe installed. However on F20 the large monolithic perl package has been split into a large number of smaller packages. To get the pod conversion working this has to be added to the specfile:

```
BuildRequires:  /usr/bin/pod2man, /usr/bin/podselect
```

On SL6 the pod conversion process has the charming side-effect of generating a bogus empty man page in the 'man3′ directory for the component's perl modules. When porting to or developing for SL6 we coped with this b adding an extra line to the `%files` list:

```
%doc %{_mandir}/man3/*
```

That is no longer necessary on F20 – in fact it triggers an RPM build error – so that line has to be *removed* from specfile. However for now the RPM still needs to be buildable on SL6 too, so needs a better fix: delete any bogu empty man pages before they get packaged, by adding this line to the `%install` section of the specfile:

```
find $RPM_BUILD_ROOT -type f -empty -print0 | xargs --null --no-run-if-empty rm
```

That's a no-op on F20 as no empty files are generated in the build.

The splitting of the monolithic perl RPM on F20 can lead to other bits of perl suddenly not being available, and having to be specified as requirements. The cron component is tested using the perl testing framework as part o the build process. This now has to be listed explicitly in the specfile:

```
BuildRequires: perl(Test::Harness)
BuildRequires: perl(Test::Simple)
```

With these changes lcfg-cron builds for F20 but also still builds for SL6.

Edit: although the cron component *builds* on F20 it can't manage `crond` yet because bug 703.

Posted in Uncategorized | Leave a comment |

grub2

Posted on April 18, 2014 by squinney

I've started work on writing a new LCFG component to manage the grub2 configuration on EL7. To assist with t switch over from the legacy version my aim is to be consistent with the previous grub component where possibl Some parts of the interface do badly need improvement though, particularly the support for multiple grub installations will be completely dropped. I am also writing the new component in Perl, another shell componen bites the dust…

I've come up against one annoying issue. We have always used password protection to prevent against users editting the entries in the boot menu. It appears with grub2 that this also makes it impossible to select alternate menu entries that are password protected. This was first reported to Redhat/Fedora on 14th July 2012 (bug#840204) for Fedora 17 and it still applies to RHEL7 in April 2014 (bug#853430). Clearly Redhat has ver little interest in "fixing" this issue which makes me think we will have to carry local patches. It looks like this change in behaviour is a result of the upstream Gnu developers wanting menu options to be inaccessible by default. It does appear that the patches in the Redhat bugzilla entry will at least make grub2 work the way we need.

Posted in Uncategorized | 1 Comment |

file component

Posted on April 3, 2014 by squinney

We all love the LCFG file component, how would we survive without it? In a lightweight lcfg installation the file component is actually truly essential. Sadly it has a few bad quirks, the worst of which is that the component wi not start if there are errors building any of the managed files. Ideally we would alter this behaviour so that it lo the errors and bravely continues but that's not going to happen right now. Consequently to get the file compone working I've also ported lcfg-etcservices (a simple drop-in of the `/etc/services` file from F20 as a new template) lcfg-nsu (which thankfully just built). We can now splat files into the filesystem to our heart's content.

Posted in Uncategorized | Leave a comment |

om access for non-root users

Posted on April 3, 2014 by squinney

It's always nice to be able to issue `om` commands as a normal user rather than having to be permanently logged i as root. In the fully managed DICE world we use the `DICE::Authorize` Perl module to do the authorisation. In a lightweight installation we have to fallback to using the much more simple `LCFG::Authorize` module. The selectio of the authorization module is typically done by altering the `profile.authorize` module although it can also be dc a per-component basis using the `om_authorization` resource.

When using the `LCFG::Authorize` module there is normally a default group named *superusers* which has the capability to call all om commands. Thus the simplest way to give yourself om super-powers is to add yourself t the list of users for that group, that can be done like this:

```
!authorize.users_superusers mADD(rod)
!authorize.users_superusers mADD(jane)
!authorize.users_superusers mADD(freddy)
```

The authorization model can be made much more sophisticated than this example but right now we're just port to a new platform so there are just a few of us who require super-powers.

Posted in Uncategorized | Leave a comment |

Bootstrapping the client

Posted on April 3, 2014 by squinney

To work properly the LCFG client really needs the lcfg user and group to exist on the system. With a fully mana system we achieve this by using the LCFG auth component. On a "lightweight" system or one to which we are ir the early stages of porting we need another approach. To solve this bootstrapping issue I've dealt with bug#252 adding a pre-install script which will add the necessary user and group if they do not already exist.

Posted in Uncategorized | Leave a comment |

Firewall

Posted on April 3, 2014 by squinney

I have seen a number of problems in F20 with daemons not being able to receive data even though they have successfully acquired a network port (e.g. the lcfg client and logserver). I had assumed this was something we h done wrong but this morning I discovered the true cause when I couldn't get SSH access to my F20 VM. The iss is actually related to the use of FirewallD. For now, the simplest solution is to do `systemctl disable firewalld.service` and then reboot. I've also removed FirewallD from our small "base" package list as we are unlikely to need it there. It might be that at some point in the future someone writes an LCFG component to manage FirewallD, until then we'll stick with good old iptables.

Posted in Uncategorized | Leave a comment |

Mass change to specfiles

Posted on March 27, 2014 by squinney

As announced a while ago to be able to build packages on EL7/F20 we needed to remove the build-requirement the existence of the `/etc/rpm/macros.cmake` file. This requirement was only necessary on SL5 where we had CMak packages from two different sources available and only one of them provided that essential file. On more recent systems the directory for rpm macros files has changed. Today I made the mass change to all specfiles in the LC source subversion repository so that they are now all buildable on both SL6 and EL7. You probably want to run "svn update" on your working copy now.

Posted in Uncategorized | Leave a comment |

core-prereq but not lcfg-logserver

Posted on March 19, 2014 by Chris Cooke

The dependencies for most of the "core" packages are now in the "core-prereq" section of `lcfg_f20_lcfg.rpms`. (Summary: lots of perl modules.) I got them by wiping and reinstalling my F20 VM with "Minimal Install", addi the local yum repo definition, and using yum to install each core RPM.

I've noticed another package that's missing from the core section. It's **lcfg-logserver**. I've tried building it – mak the usual specfile changes – but it won't build. It seems that the RPM build is failing to find a man page, even though the %files section of the specfile just specifies a glob. Something to do with gzipping (or not) the man pages, I'm guessing. I haven't figured it out yet. The build logs are all online for anyone with a DICE account, ar if you don't have one of those then make yourself an iFriend account which is almost as shiny. If you think you know what might be going wrong here, please get in touch.

Oh, and **lcfg-file** has joined the happy F20 throng, at version 1.2.1-1.

Posted in Uncategorized | Leave a comment |

Package lists remade.

Posted on March 18, 2014 by Chris Cooke

As threatened I've now remade the package lists, this time excluding the Fedora 20 Updates repository. I took th opportunity to change the composition of one or two lists.

lcfg_f20_installbase.rpms
As before this is based on the *Minimal Install* group.

lcfg_f20_base.rpms
The previous version of this list was based on the *Infrastructure Server* group but I based this version on the *Basi Desktop* group instead. This may not be ideal for servers but in retrospect I don't think Infrastructure Server was exactly ideal for desktops. In addition I can't identify a suitable group to be shared by desktops and servers, and this port is currently aimed at desktops: servers will come later. In fact strictly speaking this Fedora 20 port is, I hope, a dress rehearsal for the SL7 port which will be aimed at desktops. The closest thing I can see (in the outp of `yum grouplist`) to a group shared by everything is *Minimal Install*, which has been used as the basis of the installbase list.

lcfg_f20_desktop.rpms
This version adds the following groups to the *Basic Desktop* group used in the base package list:

- *GNOME Desktop*
- *KDE Plasma Workspaces*
- *Xfce Desktop*
- *LXDE Desktop*
- *Cinnamon Desktop*
- *MATE Desktop*
- *Sugar Desktop Environment*

The remade package lists have improved matters. Running updaterpms on my test F20 machine now produces a mere 31 conflicts, all of which appear to be from dependencies not yet listed in `lcfg_f20_lcfg.rpms`.

I also noticed in passing that I've missed a core LCFG package: I haven't yet ported lcfg-file.

For the record, here is the output of `yum grouplist` on my Fedora 20 machine:

```
Available environment groups:
   GNOME Desktop
   KDE Plasma Workspaces
   Xfce Desktop
   LXDE Desktop
   Cinnamon Desktop
   MATE Desktop
   Sugar Desktop Environment
   Development and Creative Workstation
   Web Server
   Infrastructure Server
   Basic Desktop
   Minimal Install
Available Groups:
   3D Printing
   Administration Tools
   Authoring and Publishing
   Books and Guides
   C Development Tools and Libraries
   Cloud Infrastructure
   D Development Tools and Libraries
   Design Suite
   Development Tools
   Editors
   Educational Software
   Electronic Lab
   Engineering and Scientific
   Fedora Eclipse
   FreeIPA Server
   Games and Entertainment
   LibreOffice
   Medical Applications
   Milkymist
   Network Servers
   Office/Productivity
   RPM Development Tools
   Robotics
   Security Lab
   Sound and Video
   System Tools
   Text-based Internet
```

```
Window Managers
```

Posted in Uncategorized | Leave a comment |

Two steps forward, one step back.

Posted on March 17, 2014 by Chris Cooke

As hoped, today I got the F20 VM running updaterpms, using a list of packages obtained from the machine's ow LCFG profile. The updaterpms problem mentioned at the end of last week's episode was easily solved by being reminded where Stephen had put our local mirror of the Fedora 20 repository, and altering the machine's `updaterpms.rpmpath` resource to match. So, updaterpms ran! Using a list of packages derived from the machine's LCFG profile! Hooray!

That's the good news.

The result: lots of updaterpms error messages and 149 package conflicts. This on a machine with only 324 installed packages.

The conflicts were of various types.

Some of them were to be expected – for instance I haven't yet got around to adding the prerequisites of the core LCFG packages to the `lcfg_f20_lcfg.rpms` list, so they were being listed as "scheduled for deletion", which was triggering off conflicts since they were still required by updaterpms and the LCFG client software.

Another repeated complaint was of missing RPM header files, which I took to mean missing RPMs – missing fro the repository, that is.

However one set of errors was more interesting. A number of packages turned out to have more recent versions specified in the LCFG package lists I'd made than were installed on the "minimal install" VM I was using. The versions were also more recent than those in our local mirror of the Fedora repository. But my other VM, the les minimal one, already had the more recent versions installed.

It seems that when I made my F20 package lists I used yum's standard F20 configuration, which takes packages from the Fedora 20 repository and also from the Fedora 20 Updates repository. The more recent versions were from the latter.

So, what to do? Do we start mirroring the updates repository too, so my package lists work, and the missing RP are found by updaterpms? Or do we remake the package lists, this time excluding updates? The answer is that we'll have to do both. The `lcfg_f20_base.rpms` list should use packages solely from the "base" Fedora 20 repositor The `lcfg_f20_updates.rpms` list, when we get around to putting something in it, should use packages solely from t Fedora 20 Updates repository.
Stephen is obligingly making a local mirror of the Updates repository as I write, and I'll tackle the package lists again tomorrow.

It's interesting to note in passing that my "minimal install" F20 VM does not appear to have updated its RPMs tc those in the Updates repository, but my slightly less minimal (lightweight desktop plus RPM tools, I think) F20 has.

The “yum” commands I used to make the package lists the first time around did not specify repositories in any way. This time round I’ll try disabling the updates repository by adding the `--disablerepo=updates*` option to the yum command line. This seems to work:

```
# yum repolist
repo id                         repo name                                   status
fedora/20/x86_64                Fedora 20 - x86_64                          38,597
inf-devel                       Informatics devel bucket                        52
updates/20/x86_64               Fedora 20 - x86_64 - Updates                14,053
repolist: 52,702
# yum --disablerepo=updates* repolist
repo id                           repo name                                 status
fedora/20/x86_64                  Fedora 20 - x86_64                        38,597
inf-devel                         Informatics devel bucket                      52
repolist: 38,649
#
```

Posted in Uncategorized | Leave a comment |

Getting close

Posted on March 13, 2014 by Chris Cooke

Today I got an F20 VM up on our networked KVM servers. (Until now I’ve been running F20 on VirtualBox on n laptop, which is great for me but not so good for others who want to follow along and use my work for their ow F20 development.) My new F20 VM was installed with the ‘Minimal Install’ set. Our KVM servers now have basi support for the creation of Fedora 20 VMs.

I’ve also made a basic LCFG profile for the F20 VM. To get that to build I had to add a few package lists I’d forgotten to create. I also had to change round the network component’s lcfg defaults slightly so that when usin on a non-SL OS you get the same schema as on SL6.

I then set about bootstrapping updaterpms on the new F20 VM. The first stage of this was deliciously easy: I co the configuration of a local yum repository to my new F20 VM, making sure that it pointed at the local F20 buil bucket. Then I was simply able to:

```
# yum install updaterpms
# yum install lcfg-client
```

Both of these yum commands *worked first time*. I’m very pleased.

Now that I have a proper profile generated for the VM, the client component starts and properly downloads the profile.

The next stage of the bootstrap was to try the updaterpms component. I’ve built that for F20, with the usual specfile BuildRequires adjustments. It also installed cleanly with yum.
Starting it up produces the following message:

```
LCFG updaterpms: failed to fetch rpmlist
```

… followed by a slightly bogus-looking URL. That'll be the next problem to tackle.
It feels like we're getting close to having a bootstrapped basic LCFG system. Exciting!

Posted in Uncategorized | Leave a comment |

Coming up …

Posted on March 12, 2014 by Chris Cooke

The next step will be to get updaterpms to manage all of the RPMs on an F20 machine, using data from the machine's LCFG profile, downloaded and interpreted by the LCFG client. I've already done much of the heavy lifting involved in that. One thing I hadn't yet done was to make the basic infrastructure headers for the new OS so I've made and committed them. I've also created and committed the rest of the package lists that the platform may need.

Posted in Uncategorized | Leave a comment |

#pragma LCFG category "core"

Posted on March 11, 2014 by Chris Cooke

The 'core' section of the as yet nonexistent `lcfg_f20_lcfg.rpms` list should now be complete, and here it is:

```
lcfg-authorize-1.1.1-1/noarch                   /* MPU */
lcfg-client-3.1.3-1/noarch                      /* MPU */
  perl-LCFG-Client-3.5.8-1/noarch               /* MPU */
lcfg-file-1.2.0-1/noarch                        /* MPU */
lcfg-logserver-1.5.0-1/noarch                   /* MPU */
lcfg-ngeneric-1.4.2-1/noarch                    /* MPU */
lcfg-om-0.9.2-1                                 /* MPU */
lcfg-sysinfo-1.4.1-1/noarch                     /* MPU */
lcfg-utils-1.3.8-1                              /* MPU */
lcfg-utils-devel-1.3.8-1                        /* MPU */
lcfg-pkgtools-1.0.13-1                          /* MPU */
lcfg-pkgtools-devel-1.0.13-1                    /* MPU */
perl-LCFG-PkgTools-1.1.2-1                      /* MPU, gives us qxpack */
perl-LCFG-PkgUtils-1.0.4-1                      /* MPU */
perl-LCFG-Utils-1.7.0-1                         /* MPU */
```

It's worth mentioning the problems I had building lcfg-pkgtools. After making the usual change to BuildRequire (removing `/etc/rpm/macros.cmake`) the RPM build threw up these:

line 12: prereq is deprecated: Prereq: /sbin/ldconfig
This just seems to be a warning but you can fix it by renaming `Prereq:` to the newer `Requires(pre):`.

File not found by glob: /builddir/build/BUILDROOT/lcfg-pkgtools-1.0.12-1.x86_64/usr/lib64/liblcfg_pkgtools.so.*
Stephen found the solution to this one: edit all the `CMakeLists.txt` files in the project (subdirectories may have th too) to change any occurrences of `CMAKE_INSTALL_LIBDIR` to `LIB_INSTALL_DIR`.

Posted in Uncategorized | Leave a comment |

lcfg_f20_desktop.rpms

Posted on March 7, 2014 by Chris Cooke

The latest package list for F20 is the desktop list. Here it is in the repository. I based it on Fedora 20's "Basic Desktop" yum environment group. I'm not sure how good an idea this was as I haven't yet tried making an F20 machine using just "Basic Desktop", but I've committed the new list anyway. If anyone has tried "Basic Desktop' already please let me know whether or not it's usable – thanks.

Posted in Uncategorized | Leave a comment |

lcfg_f20_base.rpms

Posted on March 7, 2014 by Chris Cooke

… is now made and committed (view it in the repository). It's based on the "Infrastructure Server" Fedora 20 yu environment group, this being the closest thing I could see to a "Basic Server" group.

Posted in Uncategorized | Leave a comment |

lcfg-client

Posted on March 7, 2014 by Chris Cooke

… is now built and installed, together with its prereqs. The new versions are as follows:

```
lcfg-ngeneric-1.4.2-1/noarch
lcfg-sysinfo-1.4.1-1/noarch
lcfg-client-3.1.3-1/noarch
```

Posted in Uncategorized | Leave a comment |

lcfg_f20_installbase.rpms

Posted on March 7, 2014 by Chris Cooke

I've just committed an F20 installbase list. See it in svn.lcfg.org.

Like the earlier F19 list it's based on the Fedora "Minimal Install" yum environment group and it was made in tl same way.

Posted in Uncategorized | Leave a comment |

updaterpms and rdxprof package list

Posted on March 6, 2014 by Chris Cooke

Not much progress to report today, but meanwhile here's a complete list of the extra packages I ended up addin (to my F20 VM) to get updaterpms and perl-LCFG-Client installed:

```
lcfg-om-0.9.2-1
```

```
perl-Business-ISBN-2.06-4.fc20/noarch
perl-Business-ISBN-Data-20120719.001-4.fc20/noarch
perl-CGI-3.64-1.fc20/noarch
perl-Compress-Raw-Bzip2-2.062-2.fc20
perl-Compress-Raw-Zlib-2.062-2.fc20
perl-DB_File-1.831-1.fc20
perl-Encode-Locale-1.03-7.fc20/noarch
perl-FCGI-0.74-10.fc20
perl-File-Listing-6.04-7.fc20/noarch
perl-HTML-Parser-3.71-4.fc20
perl-HTML-Tagset-3.20-18.fc20/noarch
perl-HTTP-Cookies-6.01-7.fc20/noarch
perl-HTTP-Daemon-6.01-7.fc20/noarch
perl-HTTP-Date-6.02-8.fc20/noarch
perl-HTTP-Message-6.06-7.fc20/noarch
perl-HTTP-Negotiate-6.01-7.fc20/noarch
perl-IO-Compress-2.062-2.fc20/noarch
perl-IO-HTML-1.00-3.fc20/noarch
perl-IO-Socket-IP-0.26-1.fc20/noarch
perl-IO-Socket-SSL-1.955-1.fc20/noarch
perl-IO-Tty-1.10-12.fc20
perl-IPC-Run-0.92-4.fc20/noarch
perl-LCFG-Client-3.5.8-1/noarch
perl-LCFG-Utils-1.6.0-1
perl-LWP-MediaTypes-6.02-4.fc20/noarch
perl-List-MoreUtils-0.33-11.fc20
perl-Net-HTTP-6.06-4.fc20/noarch
perl-Net-LibIDN-0.12-16.fc20
perl-Net-SSLeay-1.55-4.fc20
perl-Readonly-1.03-24.fc20/noarch
perl-Readonly-XS-1.05-16.fc20
perl-Sub-Name-0.05-11.fc20
perl-Test-Simple-1.001002-1.fc20/noarch
perl-TimeDate-2.30-3.fc20/noarch
perl-Try-Tiny-0.18-1.fc20/noarch
perl-UNIVERSAL-require-0.13-14.fc20/noarch
perl-URI-1.60-11.fc20/noarch
perl-W3C-SAX-XmlParser-0.99-4
perl-W3C-Util-Basekit-0.91-4
perl-WWW-RobotRules-6.02-8.fc20/noarch
perl-XML-Parser-2.41-11.fc20
perl-YAML-Syck-1.27-3.fc20
perl-libwww-perl-6.05-3.fc20/noarch
```

Posted in Uncategorized | Leave a comment |

We have achieved rdxprof.

Posted on March 4, 2014 by Chris Cooke

Let's see if I can reconstruct today's trail of package building. When we left off yesterday I was trying to build `perl-LCFG-Client` but this was failing for want of `perl-HTTP-Server-Simple-Static` which wouldn't build for lack of `perl-ExtUtils-MakeMaker`. Today:

- Adding a BuildRequires on `perl(ExtUtils::MakeMaker)` got `perl-HTTP-Server-Simple-Static` to build. The new version is `perl-HTTP-Server-Simple-Static-0.09-3.inf.noarch.rpm`.
- `perl-LCFG-Client` still wouldn't build. This time it wanted `perl-LCFG-Om-Command`.
- `lcfg-om` (which provides `perl-LCFG-Om-Command`) wouldn't build because of a build requirement for `/etc/rpm`

`/macros.cmake`. Removing this (in `lcfg-om-0.9.1`) got the build process failing at a later stage: the man page file weren't being made. Stephen worked this one out: the man pages are made by `/usr/bin/podselect`. On SL6 th is part of the core perl distribution but on Fedora 20 it's not. Adding a build requirement on `/usr/bin/podsele` is harmless on SL6 but allows the package to build on Fedora 20. The successful package was `lcfg-om-0.9.2-1.x86_64.rpm`.

- perl-LCFG-Client now failed for want of the old `W3c::Sax::XmlParser`. That built first time, refreshingly.
- However it turned out to need `W3C::Util::Exception`, provided by `perl-W3C-Util-Basekit`. That built too.
- `perl-LCFG-Client` next needed `/usr/bin/podselect`. Doh! Adding a build requirement for this brought perl-LCFG Client up to 3.5.8 and allowed it to build on Fedora 20.
- It even installs, given a bunch of perl module installations. (I'll produce the package list later so you can try at home.)
- And it seems to work too; it successfully retrieved an XML profile both from a local file and from an LCFG server via HTTP, and turned it into a DBM file, allowing me to query resources with `qxprof`.

There are two changes here which will probably be needed by lot of our local packages' `.spec` files:

1. Any mention of `/etc/rpm/macros.cmake` in the `BuildRequires` section should be removed. That file doesn't exist c Fedora 20, and isn't needed on SL6.
2. Any package which makes man pages from pod source will need to have `/usr/bin/podselect` *added* to its `BuildRequires`. That's also safe for SL6.

Posted in Uncategorized | Leave a comment |

updaterpms, and starting on rdxprof

Posted on March 3, 2014 by Chris Cooke

Today I've been trying to test updaterpms and to test the basic functionality of the LCFG client on Fedora 20 (we've upgraded from Fedora 19).
Updaterpms went OK. I tweaked its build dependencies and released version 3.3.2, which built cleanly for f20. I removed /etc/rpm/macros.cmake from its BuildRequires and also removed an inclusion of an obsolete C header /usr/include/curl/types.h. Simple testing went OK. When run on my f20 test VM it behaves as follows:

- It complains that it can't find a load of RPMs which are in the list (quite right, I haven't got local copies, I jus added their names to the rpmlist file).
- It complains that the same RPMs don't have header files either (right again).
- It installs a few packages when I add them to the list it's looking at.
- It deletes a few packages when I remove them from the list it's looking at.

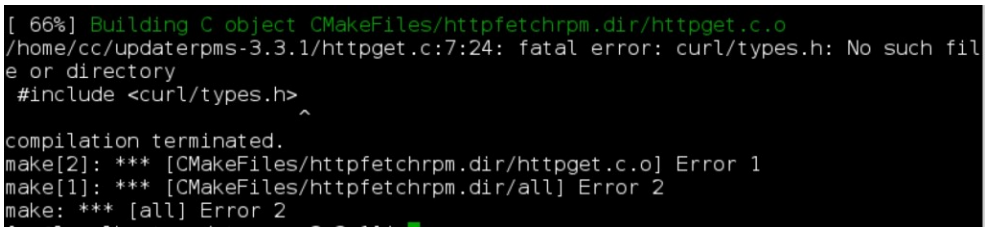
However the next part of today's work, the installation of rdxprof, is turning into a trip down the rabbit hole. Firstly it took me a while, and incidentally several failed build attempts involving a chain of dependencies, to realise that I wanted to build RPM perl-LCFG-Client rather than lcfg-client – my fault, I should have checked. Th a build attempt on perl-LCFG-Client on f20 failed for lack of a sufficiently up to date perl-HTTP-Server-Simple-Static. (We need >= 0.09 but Fedora 20 offers 0.07.) A build attempt on that failed for lack of – well, I *think* fo lack of perl-ExtUtils-MakeMaker, because that's what the pkgforge build.log tells me, but there's no dependency it in the perl-HTTP-Server-Simple-Static specfile. Still trying to sort that out.

Posted in Uncategorized | Leave a comment |

Building updaterpms on F19

Posted on February 18, 2014 by Chris Cooke

Today I tackled the error I mentioned in my last post. After spending a while looking for complicated cmake problems I realised that the problem could be fixed simply by installing the missing library with `yum`. (Head, mee desk.) Several other libraries were also missing from my F19 VM, but `yum` was able to install them. That is, until got to `liblcfgutils`. I got that by transferring a tarball of `lcfg-utils` from the subversion repository, and building It built and installed cleanly, first time. Having done that `updaterpms` still isn't building but at least I've come up against a less silly error:



The missing `curl/types.h` file is presumably `/usr/include/curl/types.h`. In SL6 this is provided by the package `libcurl-devel` but in F19 it isn't. Let's take a closer look –

```
% cat /usr/include/curl/types.h
/* not used */
%
```

Commenting out the inclusion of `curl/types.h` in `httpget.c.cin`, **updaterpms builds**!

Posted in Uncategorized | Leave a comment |

Fedora 19 installbase

Posted on February 12, 2014 by Chris Cooke

Today I got a Fedora 19 VM up and running, made a first pass at an installbase RPM list for it, and had a cursor go at building updaterpms.

The output of `yum group list` suggests that the base system group is called *Minimal Install*. I installed that in a chrooted directory with:

```
yum groupinstall 'Minimal Install' --installroot=/var/tmp/minimal --releasever=/
```

The resulting RPMs were massaged into `lcfg_f19_installbase.rpms` which has been tentatively committed to `<subversion>/packages/lcfg`.

Having done that I naturally wanted to try it out with updaterpms, but:

```
[cc@localhost ~]$ cd updaterpms-3.3.1/
[cc@localhost updaterpms-3.3.1]$ cmake .
-- The C compiler identification is GNU 4.8.1
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Found Perl: /usr/bin/perl (found version "5.16.3")
CMake Error at CMakeLists.txt:13 (message):
  Could not locate rpm library


-- Configuring incomplete, errors occurred!
See also "/home/cc/updaterpms-3.3.1/CMakeFiles/CMakeOutput.log".
[cc@localhost updaterpms-3.3.1]$
```

Incidentally, I have a full recording of the yum groupinstall, showing exactly why each dependency was pulled i We ought to keep that somewhere for posterity…?

Posted in Uncategorized | Leave a comment |

LCFG and systemd

Posted on February 11, 2014 by Alastair Scobie

Alastair gave a quick introduction to *systemd* to the Informatics COs and explained some of the problems we're going to face with integrating LCFG with *systemd*.

In summary, to integrate LCFG with *systemd* properly will require some fundamental changes to the LCFG framework. Unfortunately, we do not have time to properly consider such changes before we have to deploy RH 7 desktops, so we have no option but to find some way of using the existing framework with *systemd*. It's likely this will involve taking different approaches for different components. Once we have finished the desktop port, will take stock and consider what changes need to be made to the LCFG framework – before we start work on tl server port.

(Alastair's slides are here)

Posted in systemd | Leave a comment |

Project officially starts

Posted on January 20, 2014 by Alastair Scobie

The project to port LCFG to an RHEL 7 based distribution has officially started!

Posted in Uncategorized | Leave a comment |

Package lists and os headers

Posted on January 16, 2014 by Chris Cooke

I've created package list files and 'os' headers for SL7. They're based on those currently in use for SL6. I had to make a few changes:

- Since RHEL7beta is 64 bit only the base architecture is **x86_64**.
- The `_64` files are no longer needed. For instance `lcfg/os/sl7.h` will give you 64 bit SL7.

- The package lists are all empty as yet, apart from a little bit of internal structure.
- Some of our package lists and headers are specific to a particular minor version of the operating system. I've taken a wild bet on the first version of SL7 being SL7.0, giving for instance **`lcfg_sl70_kernel.rpms`**.
- Although x86_64 is the only one of the current RHEL7beta architectures which we expect to support, there i for instance the possibility of a future ARM version (there's already an ARM port of Fedora) so there's room a future **`lcfg/os/sl7_arm.h`** if need be.

Posted in Uncategorized | Leave a comment |

kdcregister

Posted on December 17, 2013 by Kenny MacDonald

As part of testing my krb5-1.11 RPMs on SL6 (for EASE upgrade) I found that kdcregister needed patched to wo at all, bug submitted. This will also be required for Scientific Linux 7 as it will use the same version of the Kerbe libraries.

I haven't used kdcregister before, and also noted that I had to use both the `-p` and `-k` options to get it to use the keytab. I mentioned this in the bug too, so it won't be forgotten.

Posted in Uncategorized | Leave a comment |

Bugs

Posted on December 12, 2013 by squinney

I've made a start on filing bugs for work that needs doing on various components for SL7. So far I've just done a simple search for references to scripts in `/etc/init.d` or `/etc/rc.d/init.d`. These all need updating to use the `servi` command which works fine with upstart in SL6 and systemd in SL7.

We also spotted that the grub component is probably going to need a complete rewrite for SL7 as it uses grub version 2. It has needed the schema reworking for a long time now to simplify the resources anyway.

Posted in Uncategorized | 2 Comments |

Automatic home-directory creation

Posted on December 12, 2013 by squinney

It's nice being able to login to our RHEL7 VM but it would be even nicer if we all had home directories.

This can be done using a PAM module to create home directories whenever they are required. I've previously us pam_mkhomedir but apparently this has now been replaced by a plugin for something called *oddjob*.

Install the package:

```
yum install oddjob-mkhomedir
```

Enable the *oddjob* service:

```
chkconfig oddjobd on
service oddjobd start
```

and then enable the PAM module:

```
authconfig --enablemkhomedir --update
```

A quick grep for `oddjob` in the main PAM auth file (`/etc/pam.d/system-auth`) shows that this has had an effect:

```
session optional pam_oddjob_mkhomedir.so skel=/etc/skel/ umask=0022
```

Hey presto, now when you login you will get a home directory created if you didn't already have one.

Posted in Uncategorized | Leave a comment |

Authentication and Authorization

Posted on December 12, 2013 by squinney

In Informatics we use Kerberos for authentication and LDAP for authorisation, so to allow users to login to our RHEL7 VM we need to get those both properly configured. This is all fairly simple since Redhat provide scripts which will do a lot of the work for us.

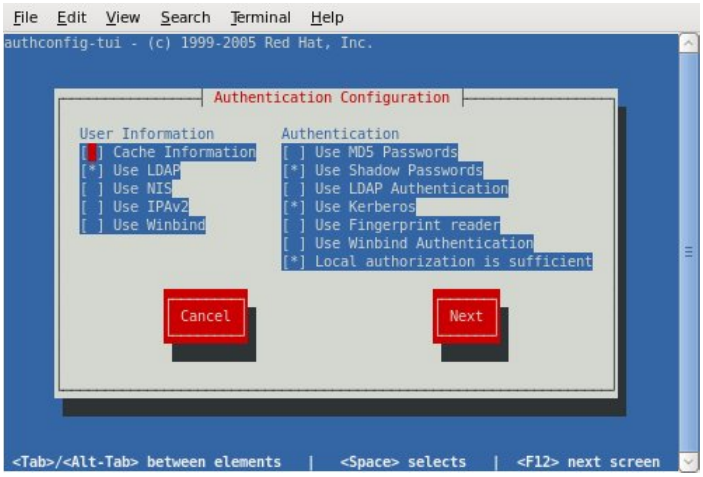
The first step is to install some useful packages:

```
yum install nss-pam-ldapd pam_krb5 \
            openldap-clients krb5-workstation
```

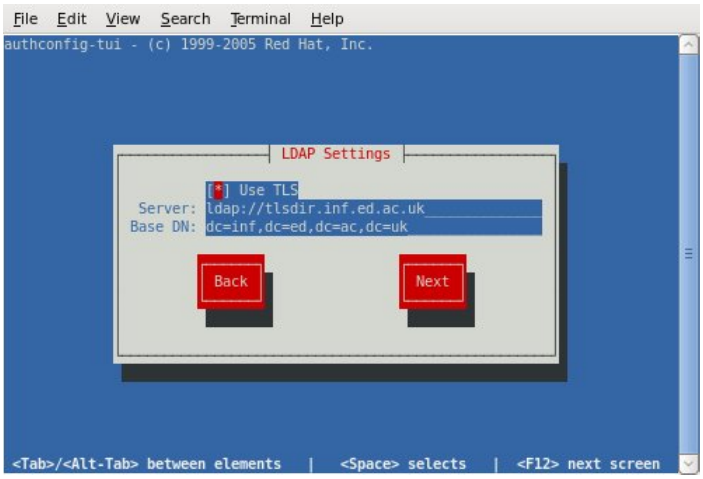
Not sure I needed all of the following, for simplicity, I just copied the standard DICE setup:

```
yum install cyrus-sasl-plain cyrus-sasl-gssapi cyrus-sasl-md5
```

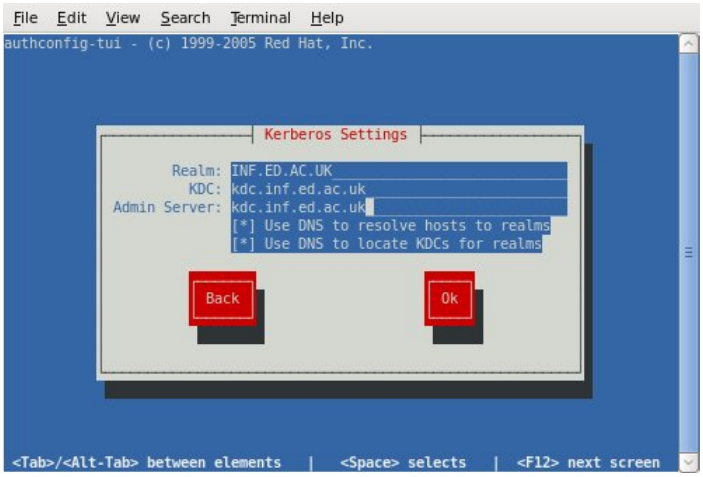
To configure the various bits (e.g. PAM and nsswitch.conf) the easiest approach is to use the `authconfig-tui` command line tool which looks something like this:



Here is the LDAP configuration:



and here is the Kerberos configuration:



This got things almost but not quite working. I discovered that nslcd wasn't actually running, this required:

```
chkconfig nslcd on
service nslcd start
```

To properly match our requirements on DICE I configured them to use our TLS-enabled LDAP server. This invol copying the `/etc/pki/tls/certs/dice-sixkts.crt` certificate from my DICE desktop into the same location on the RHEL7 machine. I then made the `/etc/nslcd.conf` file look like this (which is almost identical to the DICE versio but lacks the homeDirectory attribute mapping for AFS):

```
uid nslcd
gid ldap
uri ldap://tlsdir.inf.ed.ac.uk
base dc=inf,dc=ed,dc=ac,dc=uk
ssl start_tls
tls_cacertfile /etc/pki/tls/certs/dice-sixkts.crt
tls_reqcert demand
bind_timelimit 60
reconnect_sleeptime 5
reconnect_retrytime 60
nss_initgroups_ignoreusers ALLLOCAL
```

and made the `/etc/openldap/ldap.conf` file look like identical to the DICE version.

```
HOST tlsdir.inf.ed.ac.uk
URI ldap://tlsdir.inf.ed.ac.uk
BASE dc=inf,dc=ed,dc=ac,dc=uk
TLS_CACERT /etc/pki/tls/certs/dice-sixkts.crt
TLS_REQCERT demand
```

At this point I was able to login using my normal DICE username and password.

Posted in Uncategorized | Leave a comment |

Getting Started

Posted on December 12, 2013 by squinney

To get started with the shiny, new RHEL7 beta I have installed one as a KVM guest. This was mostly fairly straightforward, the installer had correct values for most of the options so there was little effort required to do a minimal install. The only problem was that for some reason it wouldn't enable the network on bootup. In the process of attempting to debug that problem we noted the lack of the `ifconfig` and `netstat` tools, they are in the net-tools package which, surprisingly is not part of the minimal install. Eventually we gave up on network-mana and went back to a manual call to the `dhclient` tool which worked just fine.

Moving on we needed to make the system actually useful so we installed emacs which pulled in 102 packages (200MB) including Perl. So, the even bigger surprise is that Perl is **not** part of the minimal install! This still left installation at just a tad over 1GB so that's not bad for a basic system.

Posted in Uncategorized | Leave a comment |

Redhat Enterprise 7 beta announced

Posted on December 12, 2013 by Alastair Scobie

As an early Xmas present, Redhat have announced the availability of RHEL 7 beta.

This means we can start looking at what LCFG changes we'll need to make for SL7

Posted in Uncategorized | Leave a comment |

Scientific Linux 7 LCFG port diary
Proudly powered by WordPress.