Installing LCG Software with Quattor

C. Loomis
Laboratoire de l'Accélérator Linéaire (LAL), Orsay, France

February 13, 2005

Abstract

The Quattor Working Group maintains and supports the pan templates and configuration components necessary to install LCG software with Quattor. This manual describes how to use these and how to adapt them to your needs.

Contents

1 Overview

Quattor is a modular toolkit which facilitates the centralized installation, configuration, and management of a computer cluster. The LCG Quattor Working Group (QWG) maintains and supports the software necessary to use Quattor to install and maintain a site participating in the LHC Computing Grid (LCG). The full mandate of the group can be found starting from the Grid Deployment Board web page[1] and further information, on the QWG's home page[2]. The QWG strongly encourages sites using Quattor to participate actively in its activities.

The Quattor LCG configuration consists of two parts: a set of "pan templates" and a corresponding set of configuration components. The templates contain the configuration parameters and the components act as client-side agents to affect configuration changes on the clients.

The supplied configuration currently covers all of the LCG services on the UI, MON, PX, RB, BDII, CE, WN, and SE machine types. These are the services typically installed on most sites. Configurations for VO-specific services like a VOMS server or replica catalog have not yet been created. Some manual configuration of required external services (e.g. MySQL) is still necessary.

2 Prerequisites

This manual only describes the specifics of the LCG Quattor configuration. Before trying to install the LCG software with Quattor, you should already be familiar with using the pan language to describe machine configurations and with using Quattor to install and configure clients. The Quattor web site[3] contains a number of manuals[4] describing it's design and use.

Similarly you should understand what services are provided by the LCG software and the requirements for running the LCG software. More information can be obtained from the LCG grid deployment web pages[5].

3 Configuration Templates

The configuration templates form a database of a site's configuration information. These templates are compiled by the pan compiler (**panc**) into a set of XML files—one XML machine profile per client machine. The pan templates define: the schema of the machine profiles, the actual configuration parameters, and validation functions for the information.

3.1 Template Organization

The organization of the templates is actually independent of the defined schema. The templates have been divided into the following categories based on common functionality or common evolution. To the largest extent possible, the

[1] http://lcg.web.cern.ch/LCG/PEB/gdb/default.htm
[2] http://svn.lal.in2p3.fr/LCG/QWG/web/index.html
[3] http://quattor.org
[4] http://quattor.org/documentation
[5] http://lcg.web.cern.ch/LCG/peb/grid_deployment/

template design minimizes the number of templates to modify when adapting the configuration to a site.

3.1.1 Standard Templates

These templates serve three separate functions: they extend the number of basic atomic types, define standard utility functions, and define the machine profile schema.

Pan's validation features effectively keep many simple configuration mistakes from actual deployment. Strong-typing plays a large role in this; however, pan itself only defines four basic atomic types: boolean, long, double, and string. The standard templates extend the available types to include those generally encountered in machine configurations. Table 1 completely lists the available types, each with a short description.

Within these templates, a set of composite types or structures are defined. These define the schema of the machine profiles. The top-level of the schema is defined by the type `structure_profile` in the file `pro_declaration_structures.tpl`. The full schema can be traced down by following the type definitions in the structures.

Lastly, there are a set of low-level utility functions which are useful, for example, for interacting with lists and named lists (nlists). Unless you are writing a new complex configuration you are unlikely to need these. One generally useful utility is the `default()` function. The example,

```
"/path/to/value" = default(20);
```

will set the given path to the value 20 *if and only if* the value is not already set. This is useful for setting default values of parameters without having to worry about the order that templates are included or about multiple inclusion of the same template.

3.1.2 Site Templates

These templates contain a site-specific functions and information. Some can be used verbatim, but each should be reviewed to ensure that it is appropriate for your site and that the information is correct. These templates tend to contain information which will be common to all of the machines on your site. (E.g. DNS servers, domain name, time servers, etc.)

3.1.3 Hardware

The default schema for the configuration information requires some information about each machine's hardware. This information is built from standard templates describing typical processors, disks, network cards, and RAM. This subdirectory contains a generic collection of these templates. Your site's machine descriptions must still be "assembled" from the parts. If you have hardware not described in these templates, please create new templates and submit them via the bug tracking system. They will be included in future releases to minimize the effort for others. Please do follow the naming conventions for these templates.

Table 1: Available Atomic Types

Type	Function	Description
boolean	is_boolean	boolean value (pan)
double	is_double	64-bit floating point value (pan)
long	is_long	64-bit signed integer value (pan)
string	is_string	sequence of characters (pan)
type_absoluteURI	is_absoluteURI	valid absolute URI
type_asndate	is_asndate	ASN date format
type_email	is_email	valid email address
type_fqdn	is_fqdn	valid fully-qualified domain name
type_hostname	is_hostname	valid complete hostname
type_hostport	is_hostport	host port combination
type_hostURI	is_hostURI	valid host-based URI
type_hwaddr	is_hwaddr	MAC address
type_ipv4	is_ipv4	IPv4 address
type_ipv6	is_ipv6	IPv6 address
type_isodate	is_isodate	ISO date format
type_port	is_port	valid port number
type_shorthostname	is_shorthostname	valid short hostname
type_ip	is_ip	valid IP address
type_URI	is_URI	valid URI

3.1.4 Quattor

These templates consist of package lists and configuration for the Quattor clients. Review the default parameters for the Quattor updates to ensure that the latency is appropriate for your site. In general smaller sites can handle lower latencies without overloading the Quattor servers.

3.1.5 LCG2

These templates describe the configuration necessary for the LCG services. There are two subdirectories: `rpmlist_sl3` and `source`. The `rpmlist` directory contains the package lists necessary for each LCG service or machine type. The `source` directory contains the corresponding software configuration. Normally, one only needs to include the appropriate software configuration for a service; the necessary package list is included automatically.

3.1.6 Components

The templates describing the configuration components are managed in the Quattor CVS repository along with the configuration components themselves. Copies of these templates are kept here for two reasons. First, it is more convenient to download from a single location rather than from many different component packages. Second, this ensures that the exact component versions necessary for an particular LCG release are documented.

3.2 Site Adaptation

The template design minimizes the number of changes necessary to adapt the configuration to your site. Nonetheless, there are still a fair number of changes which must be made. The following sections describe the typical changes necessary when adapting the configuration to a site.

3.2.1 Non-LCG Templates

To state the obvious, you need to add one object template for each machine you want to manage.

For the distributed templates, you need to review the templates in the `site`, `quattor`, and `hardware` directories. If you have already tested a Quattor installation of a client, these changes should already be done. Ensure that you are using the standard templates distributed with the LCG templates; they extend the standard schema to include LCG-specific informaiton. These have not yet converged with those distributed with Quattor itself.

3.2.2 LCG Site Configuration

The LCG site configuration template (`pro_lcg2_config_site`) contains many global variables used throughout the rest of the LCG templates. These variables encapsulate common configuration changes and localize the corresponding changes to this site configuration template. However, radical changes in configuration may require modifying the individual service templates. The format of this template is intentionally very similar to the site configuration file for YAIM.

LCG defines a set of standard "machine types", i.e. machines with a predefined set of services. The set of pan templates which define these machine types can be used directly. If the standard machine types do not meet your needs, you can create new machine types with different mixtures of services. Use the standard machine type templates as examples to create the new machine types. If you use non-standard mixes of services, pay particular attention to possible security implications. For example, it is a bad idea to put a service which requires MySQL on a machine with user accounts or running user jobs.

3.2.3 Batch System

The default batch system is torque (pbs). Currently the templates necessary to use other batch systems are not available in the distribution. Support of other batch systems requires creating a set of templates which parallel those for torque (pbs) and a set of configuration components. If you do the work to support a new batch system, please contribute the templates and components.

3.3 Downloads

The pan templates are managed through a subversion repository. They are available via the web for download; however, checking out the templates (for write access) requires a subversion client. Subversion clients are a standard part of recent linux releases, but older releases may require you to find and install the

client manually. The sources as well as a list of available binaries are available from the subversion web site[6].

3.3.1 Versioning

It is important to get the version of the templates appropriate for the version of the LCG software you want to run. To facilitate this, the template repository is branched and tagged according to the LCG releases.

The following types of releases are available:

trunk This contains the absolute latest version of the pan templates. This is not tied to a specific LCG version nor is it guaranteed to be functional. This is equivalent to the HEAD release for CVS.

branch For each LCG release, there is a correspondingly named branch. For example for the LCG 2.3.0 release, there is a branch named `branches/LCG-2.3.0`. The templates in this branch are tied to the given LCG release, but may evolve. The branch contains the latest, experimental templates for that version.

tagged For each LCG release, there will be one or more tagged versions. These versions are fixed and have names like `tags/LCG-2.3.0-1`. The first part corresponds to the LCG release and the last is a serial tag number. In general, these releases are appropriate for production. The latest tag for a release is likely to be the most reliable.

3.3.2 Repository Location

To checkout a version of the templates, a subversion client is needed. See the above description to decide what type of release you want to check out.

To check out the trunk version use the command:

```
svn checkout http://svn.lal.in2p3.fr/LCG/QWG/templates/trunk/
```

To checkout a branch version use the command:

```
svn checkout http://svn.lal.in2p3.fr/LCG/QWG/templates/branches/LCG-2.3.0/
```

where you must replace the "LCG-2.3.0" with the LCG version you are interested in. Finally, to check out a tagged release, use the command:

```
svn checkout http://svn.lal.in2p3.fr/LCG/QWG/templates/tags/LCG-2.3.0-0/
```

replacing "LCG-2.3.0-0" with the LCG version number and tag number you want. To find the available tags, browse

```
http://svn.lal.in2p3.fr/LCG/QWG/templates/tags/
```

with a web browser. All of the available tags will be listed. New tags are announced on the QWG mailing list.

[6] http://subversion.tigris.org/

3.3.3 Packages

As the templates are only used within the Quattor configuration database (CDB or equivalent), no binary packages are created. However, tarballs of the various branches and tags are available via WebSVN. For example, to obtain a tarball for the LCG-2.3.0-0 tag, browse from

`http://svn.lal.in2p3.fr/WebSVN/LCG/QWG/templates/`

to the LCG-2.3.0-0 tag. Then click the "tarball" link to download a tarball of the corresponding templates. These can then be unpacked and inserted into your configuration database as usual.

3.4 Bug Reports & RFE

Bug reports for the components should be submitted via the Quattor project's bug tracking system[7]; use the "Quattor Pan Templates" category. Anonymous submission of bug reports is not permitted. Please register with the savannah system[8].

Requests for enhancements (RFE) should be submitted via the bug tracking as well. RFEs with patches are especially welcome.

4 Configuration Components

The pan templates by themselves are not sufficient to manage machine configurations. A client-side agent is necessary to read the machine profile generated from the pan templates and to transform that into a real configuration on the client. A set of configuration components fills this role.

Actually these agents are just perl scripts which access the configuration information (via the NVA-API), write configuration files, and restart services as necessary. For information on writing configuration components see the documentation on the Quattor web site[9].

Each component defines a schema for the configuration parameters it uses. You can find the definition of these parameters in the associated man page or by reading the corresponding pan template. For example, the template:

`pro_declaration_component_cron.tpl`

contains the configuration schema for the cron component.

4.1 Core Components

The "core" components manage standard system services and configuration, for example, the ntp daemon or user accounts. Table 2 lists the available core components each with a short description.

The available components cover the most common system services, but not all. If you need a component which does not exist, send a RFE through the bug tracking system. RFEs with code and detailed specifications are greatly appreciated.

[7] http://savannah.cern.ch/bugs/?group=elfms
[8] https://savannah.cern.ch/account/register.php
[9] http://quattor.org/documentation

Table 2: Core Components

ncm-access_control	machine access control
ncm-accounts	user accounts
ncm-altlogrotate	logrotation configuration
ncm-authconfig	authorization configuration
ncm-autofs	autofs configuration
ncm-ccm	Quattor CCM configuration
ncm-cdp	Quattor CDP configuration
ncm-cron	cron entries
ncm-dirperm	directory and file permissions
ncm-filecopy	copies information to a file
ncm-grub	GRUB configuration
ncm-interactivelimits	limits.conf configuration
ncm-iptables	firewall configuration
ncm-ldconf	ld.so.conf configuration
ncm-lmsensors	LEMON sensor configuration
ncm-logrotate	logrotation configuration
ncm-mailaliases	sendmail aliases
ncm-netdriver	network driver configuration
ncm-nfs	NFS configuration
ncm-ntpd	NTP (time server) client configuration
ncm-portmap	portmapper configuration
ncm-profile	profile.d scripts for environment
ncm-serialclient	root access over serial line
ncm-smartd	SMART daemon configuration
ncm-spma	Quattor SPMA configuration
ncm-ssh	ssh daemon configuration
ncm-state	desired state of machine
ncm-sysctl	runtime kernel parameters

Table 3: LCG Configuration Components

ncm-bdiicfg	BDII configuration
ncm-ceinfo	CE (Gatekeeper) information provider
ncm-cliconfig	Workload mgt. CLI configuration
ncm-cmnconfig	Workload mgt. common configuration
ncm-condorconfig	Workload mgt. condor configuration (RB)
ncm-dblbconfig	Logging & Bookkeeping database configuration
ncm-edglcg	EDG and LCG sysconfig files
ncm-gip	Generic information provider configuration
ncm-globuscfg	Globus configuration
ncm-gridmapdir	Creation and mgt. of the gridmapdir
ncm-guiconfig	Workload mgt. GUI configuration
ncm-lbconfig	Logging & Bookkeeping configuration
ncm-lcas	LCAS (gatekeeper authorization) configuration
ncm-lcgbdii	LCG BDII configuration
ncm-lcmaps	LCMAPS (gatekeeper mapping service) configuration
ncm-mkgridmap	Grid mapfile configuration
ncm-myproxy	MyProxy server configuration
ncm-pbsclient	PBS (Torque) client configuration
ncm-pbsknownhosts	PBS (Torque) configuration
ncm-rgmaproducer	R-GMA configuration
ncm-rm	Replica Manager configuration
ncm-sshkeys	SSH key distribution
ncm-uicmnconfig	Workload mgt. UI common configuration
ncm-wlconfig	Workload mgt. daemon configuration
ncm-yaim	YAIM configuration

4.2 LCG Components

Each service in the LCG release has a corresponding configuration component. The pan templates reference these components and provide the necessary information for them. Table 3 lists the LCG components for LCG 2.3.0, each with a short description.

The default LCG configuration currently uses only direct Quattor configuration components to manage the clients. However, there is a YAIM component to allow use of LCG's scripted installation of the machines. There is an effort to ensure that the Quattor and YAIM configurations evolve in parallel. It is expected that some sites will use the YAIM configuration, some the Quattor configuration, and some will mix the two. The QWG and YAIM developers intend to support these different scenarios.

4.3 Downloads

The Quattor CVS repository contains the NCM components necessary to configure the LCG services using Quattor. The source from for these components can be checked out directly from the CVS repository or the pre-packaged binaries

can be obtained from the QWG package repository.

4.3.1 Versioning

Currently the components are tagged according to individual (and somewhat arbitrary) version numbers. The component versions appropriate for a particular release of the pan templates can be extracted from the included "components" templates. If you use, SPMA for package installation; there is a check that the correctly-versioned component package is installed.

4.3.2 Repository Location

The source code can be obtained from the Quattor CVS repository via a standard CVS checkout. For anonymous (read-only) access, set an environmental variable:

```
export CVSROOT=:pserver:anonymous@isscvs.cern.ch:/local/reps/elfms
```

then simply check out the quattor code:

```
cvs checkout quattor
```

The code for all of the NCM components will be in the subdirectory `quattor/ncm-components`. The LCG-specific components are in the `lcg-2` subdirectory and those for general system services in `core`.

To create RPM packages, simply type **make rpm**. There is not yet a target to make a binary distribution, but an install directory can be supplied to the make command:

```
make PREFIX=/full/installation/path install
```

Creating a binary tarball from the installation directory is easy.

4.3.3 Packages

As of this writing, all of the components are written in perl and do not depend on the architecture. RPM binaries of these components can be obtained from the QWG package repository[10].

4.4 Bug Reports & RFE

Bug reports for the components should be submitted via the Quattor project's bug tracking system[11]; use the "Quattor NCM Components" category. Anonymous submission of bug reports is not permitted. Please register with the savannah system[12].

Requests for enhancements (RFE) should be submitted via the bug tracking as well. RFEs with patches are especially welcome.

[10] http://quattor.web.lal.in2p3.fr/QWG/ncm-components/
[11] http://savannah.cern.ch/bugs/?group=elfms
[12] https://savannah.cern.ch/account/register.php

5 Summary

The Quattor Working Group provides a set of configuration templates and configuration components to permit LCG sites to use Quattor. People are encouraged to contribute to this effort by using the provided configuration, feeding back problems, and writing components. More information on the group's activites can be obtained from the group's web page[13].

Bug reports and RFEs should be made through the Quattor savannah project[14]. Support can be obtained through the Quattor[15] and QWG[16] mailing lists.

[13] http://svn.lal.in2p3.fr/LCG/QWG/web/index.html
[14] http://savannah.cern.ch/projects/elfms/
[15] mailto:project-quattor@cern.ch
[16] project-lcg-gdb-quattor-wg@cern.ch