Config5

Systems Administrator's Guide

Stefan Walter

Config5: Systems Administrator's Guide

Stefan Walter

Publication date March 2013

Table of Contents

List of Examples

Preface

Config5 is a configuration management utility for UNIX systems. It is particularly well suited to manage large numbers of systems in an enterprise where the configuration varies in the various organizational units. Configuration data can be verified before deployment which makes deployment more resilient to human errors and facilitates teamwork. Config5 is highly customizable and extensible to support a broad variation of UNIX and UNIX-like operating systems.

1. Background

The IT Service Group (*isginf*) of the Department of Computer Science at ETH Zurich is managing UNIX systems since the early nineties. Since then we have always used custom configuration management utilities to provision servers and workstations. Config5 is the *fifth* utility that we developed, hence its name. We believe that Config5 may be useful for others too and therefore release Config5 as open source project and make it publicly available.

2. Motivation

Having a way to provision computer systems with configuration data over the whole life cycle is important to any systems administrator who manages two computers or more.

Some enterprise operating systems provide a way to provision a configuration during the initial installation but usually provide none to update the configuration later. Many programs exist to fill in this gap. Utilities like Bcfg2 [1], Cfengine [2], LCFG [3], Puppet [4] and many others were all developed to solve one problem: To consistently manage the configuration of many computers. Each of them has a different philosophy, different design requirements and different strengths and weaknesses. A systems administrator can already now pick the one that fits his or her needs, policies or work flow best.

At the end of 2010 *isginf* was preparing to roll out Red Hat Enterprise Linux 6. Because of some important missing functionality and other problems with our previous configuration management utilities we once more looked for alternatives. Unfortunately none of the existing utilities were suitable for our requirements and we developed Config5.

2.1. Requirements

The way IT is organized for us (in an educational institution with thirty-odd research groups) defines an interesting set of general conditions:

- Our people are free to use our service but are not required to. As a consequence we never know for sure what machines are around. A system that we install today may be reinstalled by its owner with his favorite distro the next day. Workstations are generally installed with both Windows and Linux and either of them may not be booted into for months. The difference between a reinstalled system and one that is always booted in the other operating system is difficult to track.

- We provide installations on laptops too. These may be outside the organization for weeks but must be able to update themselves without access to our infrastructure.

- Every research group has its own set of configuration data that is unique to it. Configured printers, authentication settings, firewall rules are typically all different.

  Total uniformity of the deployed systems even within a research group is difficult to enforce. There are always exceptions or special configurations requested for a project. Such exceptions must be tracked and documented.

- We are organized as a team with individual responsibilities. Parts of the configuration may be changed by several team members at the same time.

Based on the above and a few lessons learned we defined the following set of requirements for a suitable configuration management utility:

- The configuration must be pulled by client systems rather than pushed on by a server.

- The configuration must be selected using information that may be only known to the client system but is not maintained in a central database. This implies that the client system selects the configuration that it needs rather than having a server unilaterally decide what it has to have.

- Configuration files must be constructible from templates and fragments.

- The deployment should be fast and scalable. The configuration may be applied often and on many systems.

- Changes to the configuration must be traceable and revertible. This can usually be achieved with a version control management system.

- The system should be resilient to human error as much as possible. In combination with a version control management system it should prevent invalid, conflicting or incomplete configuration data.

3. Acknowledgments

Config5 bases on configng (our *fourth* configuration management utility) done by Klaus Ethgen (<klaus@ethgen.ch>) while he was a member of *isginf* . Some concepts and ideas used in configng were taken from Template Tree II [6].

Chapter 1. Introduction

In this chapter you will get a quick overview of how Config5 works and how it is used.

1.1. Overview

Configuration data that is managed and deployed with Config5 is maintained in a simple directory structure. A main directory (the *feature set*) contains a set of sub directories (*features*), each of which contains configuration files and a specification (the *spec* file) that contains instructions.

By default the main directory is named `features` and located in the directory `deploy` of the Config5 distribution:

```
$ ls
deploy  doc  example
$ ls deploy
bin  etc  features  lib
```

A feature must at least contain the specification file which is by default named `spec`. A fictive feature named `test` would look like this:

```
$ ls deploy/features
test
$ ls deploy/features/test
spec
```

1.1.1. The spec File

The spec file describes the changes to do by grouping them in *classes* and is processed from top to bottom. Classes start at the beginning of a line and are followed by a colon. *Changes* are identified by a unique keyword with leading whitespace. Each change belongs to the most recent class declaration:

```
# This is a comment

class1: …
  change1 …
  change2 …
  change3 …

class2: …
  change1 …
```

1.1.2. Changes

Config5 has nine core changes built in: `copy`, `truncate`, `append`, `directory`, `link`, `symlink`, `properties`, `remove` and `execute`. Additional changes can be added with extensions.

Each change supports a number of optional flags right after the keyword which modify the behavior of the change. The arguments after each keyword vary:

```
  copy       [-bintT] src path user group mode [context] [key=value...]
  truncate   [-intT]  path user group mode [context]
  append     [-b]     src path [key=value...]
  directory  [-ntT]   path user group mode [context]
  link       [-ntT]   src_path dest_path
  symlink    [-bntT]  value path user group [context]
  properties [-ntT]   path user group mode [context]
  remove     [-ntT]   path
  execute    [-bntT]  shell_line
```

Changes are only applied on the *target system* (the computer on which Config5 is executed) if necessary. The only exceptions is the `execute` change which is always applied.

For each change Config5 will check if applying the change would actually modify the state of the target system and skip it if does not.

1.1.3. Classes

A class may be a host name, an operating system identifier or some other property. A target system that configures itself with Config5 will use the available information about itself (the *system information*) to choose the best matching class and perform the changes of this class:

```
foobar:
```

```
  copy hosts.allow.foobar /etc/hosts.allow root root 644
  copy hosts.deny /etc/hosts.deny root root 644

RedHatEnterpriseServer:
  copy hosts.allow /etc/hosts.allow root root 644
  copy hosts.deny /etc/hosts.deny root root 644

RedHatEnterpriseWorkstation_6.1:
  copy hosts.allow.61 /etc/hosts.allow root root 644
  copy hosts.deny /etc/hosts.deny root root 644
```

Identifying the best matching class is customizable but generally the class that matches the target system in the most specific way is used. For example, if there is a class for the operating system identifier and one for the host name of the target system, the later is chosen.

Classes can also include other classes and override changes to files done in the included classes:

```
foobar: RedHatEnterpriseServer
  copy hosts.allow.foobar /etc/hosts.allow root root 644

RedHatEnterpriseServer:
  copy hosts.allow /etc/hosts.allow root root 644
  copy hosts.deny /etc/hosts.deny root root 644

RedHatEnterpriseWorkstation_6.1: RedHatEnterpriseServer
  copy hosts.allow.61 /etc/hosts.allow root root 644
```

1.1.4. File Generation

Files can be constructed from fragments using the `append` change which can be distributed over multiple features.

All files that are copied and all fragments that are appended are by default processed with Template Toolkit [5]. Using *substitution* (replacing parts of the template with generated content) it is then very easy to generate customized configuration files for a set of system from one template. The substitutions can be specified in the class declaration or per change. Substitutions in the class declaration are passed down to all included classes and all changes of the class. Substitutions that are passed down override substitutions with the same name in included classes and changes:

```
foobar: RedHatEnterpriseServer ACCESS=snafu

RedHatEnterpriseServer:
  copy hosts.allow /etc/hosts.allow root root 644 ACCESS=ALL
```

1.2. Examples

To illustrate how Config5 is used we will create a feature to configure `ntpd` and deploy it to the host `sample` which is running `Red Hat Enterprise Linux 6` (`RHEL6`). In subsequent steps we will enhance the configuration step by step.

First we start by downloading and unpacking the Config5 distribution:

```
$ cd /tmp
$ wget http://oss.isg.inf.ethz.ch/pub/Main/ProjectsConfig5/config5-latest.tgz
$ tar zxf config5-latest.tgz
$ cd config5-?.?
```

1.2.1. The Basic `ntp` feature

First we create a new feature called `ntp` and `spec` file:

```
$ cd deploy/features
$ mkdir ntp
$ cd ntp
$ vi spec
```

The content of the `spec` file should look like this:

```
RedHatEnterpriseWorkstation:
  copy ntp.conf /etc/ntp.conf root root 644
```

This change will copy the file `ntp.conf` to `/etc/ntp.conf` for all target systems that are running `Red Hat Enterprise Linux Workstation` and set the permissions as indicated.

The only missing part is the file `ntp.conf` which is created in the same place as the `spec` file:

```
$ vi ntp.conf
```

We will use a simplified version of the default `/etc/ntp.conf` that ships with `RHEL6`:

```
restrict default kod nomodify notrap nopeer noquery
restrict -6 default kod nomodify notrap nopeer noquery
restrict 127.0.0.1
restrict -6 ::1

server 0.rhel.pool.ntp.org
server 1.rhel.pool.ntp.org
server 2.rhel.pool.ntp.org
```

Now we can deploy the configuration to the host `sample`:

```
$ ( cd ../../.. ; tar cf /tmp/c5.tar deploy )
$ scp /tmp/c5.tar root@sample:/root/c5.tar
$ ssh root@sample "tar xf c5.tar ; deploy/bin/config5"
```

The output of the last command will look like this:

```
Date:               Fri Aug 24 09:54:47 2012
Host name:          sample
Architecture:       x86_64
OS:                 RedHatEnterpriseWorkstation
Release:            6.3
IP:                 127.0.0.1
FQHN:               sample.example.com
FQDN:               example.com
Root:               /
Stage:              production

Pass 1:
/etc/ntp.conf: file (ntp:RedHatEnterpriseWorkstation) - CHANGED

Done.
```

If the last command is run again it will not change the file a second time and display the following output:

```
$ ssh root@sample deploy/bin/config5
…
Root:               /
Stage:              production

Done.
```

1.2.2. Best Matching Class

When the configuration in a feature is applied, Config5 will select the class that it applies using a list of classes generated from the system information. The first class that exists in the feature will be used.

The list of all classes that are checked on a system can be obtained by running **config5 --report classorder**. For host sample the output will look like this:

```
$ ssh root@sample "deploy/bin/config5 --report classorder"
sample_RedHatEnterpriseWorkstation_6.3_x86_64
sample_RedHatEnterpriseWorkstation_6.3
sample_RedHatEnterpriseWorkstation_x86_64
sample_RedHatEnterpriseWorkstation
sample_x86_64
sample
RedHatEnterpriseWorkstation_6.3_x86_64
RedHatEnterpriseWorkstation_6.3
RedHatEnterpriseWorkstation_x86_64
RedHatEnterpriseWorkstation
```

The class `RedHatEnterpriseWorkstation` from our example is the last one in the list. We could modify the `spec` file to look like this:

```
sample:
  copy ntp.conf /etc/ntp.conf root root 644
```

When this configuration is on the host sample the result is the same. When the same configuration is applied on any other system the feature would not be applied because no class matches (assuming that the system is not named `sample`).

1.2.3. Adding Service Management to the `ntp` Feature

The `ntp` feature so far only copies the file. This is not enough because the `ntpd` service is not enabled by default and is not running. We will extend the feature to enable the service and start it:

```
RedHatEnterpriseWorkstation:
  copy ntp.conf /etc/ntp.conf root root 644
  execute chkconfig ntpd on
  execute service ntpd start
```

This will additionally execute the two shell commands **chkconfig ntpd on** and **service ntpd start**.

If we deploy this to `sample` like before the output will look like this:

```
$ ssh root@sample "tar xf c5.tar ; deploy/bin/config5"
…

Pass 1:
run (ntp:RedHatEnterpriseWorkstation): chkconfig ntpd on
run (ntp:RedHatEnterpriseWorkstation): service ntpd restart

Done.
```

The output remains the same if the **config5** command is run a second time. The service is enabled and started every time but we actually only want that to happen when `/etc/ntp.conf` changes. To change that we add a trigger flag to the lines that execute the shell commands:

```
RedHatEnterpriseWorkstation:
  copy ntp.conf /etc/ntp.conf root root 644
  execute -t root root chkconfig ntpd on
  execute -t root root service ntpd restart
```

Now the commands are only executed if the configuration file changes. When we deploy this to `sample` the output again shows no changes because the correct `/etc/ntp.conf` is already in place:

```
$ ssh root@sample "tar xf c5.tar ; deploy/bin/config5"
…
Root:               /
Stage:              production

Done.
```

1.2.4. Improving the `ntp` Feature with Extensions

Executing commands to enable and start the `ntpd` service works but there is a better way. Config5 ships with a set of extensions and one of them is the Services extension which will manage system services using the **chkconfig** and the **service** commands. It does follow the overall principle of Config5 that it only applies a change if necessary, i.e., only enable or start a service if it is not enabled or started yet.

The installation of the extension is simple:

```
$ ( cd ../.. ; tar zxf ../extensions/service.tgz )
```

We modify the spec file and use the new `service` change provided by the extension:

```
RedHatEnterpriseWorkstation:
  copy ntp.conf /etc/ntp.conf root root 644
  service enable ntpd
  service start ntpd
```

The trigger flags are gone which makes subtle difference for the systems administrator: Any time **config5** is run it will enable and start the service, even if an administrator has intentionally shut it down on a specific target system. For `ntpd` this is probably the right choice but for services that require maintenance such as `mysqld`, `httpd` or `samba` an administrator will probably not want these services to be suddenly started, especially if **config5** is run from `cron`. For these services it is better to at least trigger the service start on configuration file changes.

There is still one issue with the above spec file though. The `ntpd` service will not be restarted if the configuration file changes while the `ntpd` service is already running. The Services extension does intentionally not provide a `restart` action but the original **service ntpd restart** command will do the job:

```
RedHatEnterpriseWorkstation:
  copy ntp.conf /etc/ntp.conf root root 644
  execute -t root root service ntpd restart
  service enable ntpd
  service start ntpd
```

The `execute` change is inserted before the service changes to avoid a restart of the service because one of the `service` changes needs to be applied.

1.2.5. Installation vs. Production

We will typically install Red Hat Enterprise Linux Workstation on a target system over the network using kickstart and at that time also run **config5** from a post script section to apply the configuration. However, the environment during installation is not exactly the same as when the system is running with the installed operating system. The installer has already configured the local time so we actually do not want to start or restart `ntpd` at this point.

Config5 predefines two *stages* for a target system, `installation` and `production`, the later being the default. We will tag the service start and restart changes to only be applied in the production stage:

```
RedHatEnterpriseWorkstation:
  copy ntp.conf /etc/ntp.conf root root 644
  execute -tP root root service ntpd restart
  service enable ntpd
  service -P start ntpd
```

To make this work we will run **config5** with the option `--set stage=installation` in the kickstart post script.

1.2.6. Adding Time Servers to the `ntp` Feature

We are actually running two time servers `ts1.example.com` and `ts2.example.com` that we would prefer to the default servers. Workstations will only use `ts1` while servers will use `ts2`. We could create two files, one for servers and one for workstations but we can do better using a template. We modify the file `ntp.conf` as follows:

```
restrict default kod nomodify notrap nopeer noquery
restrict -6 default kod nomodify notrap nopeer noquery
restrict 127.0.0.1
restrict -6 ::1

[%- IF server %]
server [% server %]
[%- ELSE %]
server 0.rhel.pool.ntp.org
server 1.rhel.pool.ntp.org
server 2.rhel.pool.ntp.org
[%- END %]
```

To generate two distinct files for servers and workstations we change the `spec` file to this:

```
RedHatEnterpriseWorkstation: _ntp server=ts2.example.com
RedHatEnterpriseServer: _ntp server=ts1.example.com

_ntp:
  copy ntp.conf /etc/ntp.conf root root 644
  execute -tP root root service ntpd restart
  service enable ntpd
  service -P start ntpd
```

1.2.7. Dealing with Exceptions

As time passes we encounter a few exceptions that our configuration has to cover:

- The laptop `foobar` is configured as workstation but cannot contact the time servers because of the company firewall.
- The user of workstation `snafu` wants it to be configured to use `ts2`.
- A bug in `ntpd` on 64 bit systems causes it to crash all the time. Until the bug is fixed we need a workaround.

We change the `spec` file by adding the following two host-specific lines:

```
foobar: RedHatEnterpriseWorkstation
snafu: RedHatEnterpriseWorkstation server=ts2.example.com
```

The generated `/etc/ntp.conf` on `foobar` will contain the three time servers in the `ntp.org` pool. The same file on `snafu` will refer to `ts2.example.com`.

A suitable workaround for the crashes of `ntpd` could be to set the time with **ntpdate** and run it whenever **config5** runs: Adding the following snippet to the `spec` file does this:

```
RedHatEnterpriseWorkstation_x86_64: _ntpdate server=ts1.example.com
RedHatEnterpriseServer_x86_64: _ntpdate server=ts2.example.com
```

```
_ntpdate:
  service -P stop ntpd
  service disable ntpd
  execute ntpdate [% server %]
```

The workarounds for `foobar` and `snafu` need to be verified again at this point. Since `foobar` is an old laptop it will remain a 32 bit installation. The workstation `snafu` on the other hand could be reinstalled with either 32 or 64 bit so we make sure both work fine and add this line to the `spec` file:

```
snafu_x86_64: RedHatEnterpriseWorkstation_x86_64 server=ts2.example.com
```

Chapter 2. Configuration Data

The following sections provide in-depth explanation of the various concepts and the structure of configuration data that is deployed using Config5.

2.1. Feature Set

Configuration data that is applied using Config5 is organized in a *feature set*. A feature set is nothing more than a directory containing features (see Section 2.2, “Features”).

By default Config5 has one feature set that is located in a directory named `features` in the same directory as the `bin` directory containing the **config5** script:

```
$ ls
deploy  doc  example  util
$ ls deploy/
bin  etc  features  lib
$ ls deploy/bin
config5
```

There are two ways to customize the location of the default feature set:

- The directory `features` can be replaced by a symbolic link that points to the real location of the feature set directory. The directory the symbolic link links to must be available on all target systems that are going to be managed.
- The directory name can be changed by customizing the Config5 settings (see Section 7.3, “Settings Reference”).

2.2. Features

The actual configuration data is organized in one or more *features* in the feature set (see Section 2.1, “Feature Set”). Each feature is again a directory containing a *specification file*. The default name for the file is `spec` but that can be changed (see Section 7.3, “Settings Reference”):

```
$ ls deploy/features
autofs  firewall  ssh  sshd
$ ls deploy/features/sshd
sshd_config spec
```

The specification file contains a number of class definitions.

2.3. Classes

Classes group together individual the changes (see Section 2.4, “Changes”). The general syntax is the following:

```
class: [referenced_class…] [key=value…]
```

Any string of character excluding white space is suitable as a class name. Classes that are not intended to match (see Section 7.3.8, “Class Match Order”) can still be referenced by other classes. Typically these classes have a naming convention that distinguishes them from classes that can be selected as best match, for instance by prefixing them with an underscore.

2.3.1. Substitutions

One of the strengths of Config5 is its capability to create config files from templates using the Template Toolkit [5]. Command lines run by the `execute` change are also processed using Template Toolkit.

Substitutions are always specified as `key=value` pair. If `value` contains spaces the pair must be quoted. Substitutions can be specified in two places:

- In class definitions. Substitutions are passed to all changes of the class and all included classes.
- In `copy` or `append` changes.

Substitutions that are passed down override substitutions in changes or included classes. Each substitution is a variable in Template Toolkit with the name `key` and `value` as value.

Example 2.1. Substitution Specification

```
RedHatEnterpriseServer: _sshd permitroot=1
RedHatEnterpriseWorkstation: _sshd permitroot=0

_sshd:
  copy sshd_config /etc/sshd/sshd_config root root 600

sample:
  copy sshd_config.test /etc/sshd/sshd_config root root 600 permitroot=1
```

2.3.1.1. Predefined Substitutions

A set of substitutions are predefined and relate to information about the target system and the configuration:

Variable	Description
`auto_base`	The base directory of Config5 (parent directory of the directory containing the running **config5** script)
`auto_class`	The name of the matching class of the `copy` or `append` change
`auto_class_lc`	Same as `auto_class` but all lower case
`auto_feature`	The name of the feature of the `copy` or `append` change
`auto_dynamic_rnd`	32 bit random number (changes every time **config5** is run)
`auto_static_rnd`	32 bit random number (changed only when the `host` system item changes)
`auto_program_version`	The version of the Config5

Additionally the value of all system information items (see Section 6.2, “System Information Items”) is available as `info_`*`item`* variable. For list type system information items a variable info_*`item_entry`* is set to `1` for each entry of the list. Both *`item`* and *`entry`* are converted to lower case.

All predefined substitutions can be overloaded at class or change level.

2.3.2. Best Match

For each feature a class is selected and its changes are processed. The class is selected according to a prioritized list of class names that are generated from the available system information items. The first class that is found is used. The feature is ignored if no class matches.

See Section 7.3.8, “Class Match Order” for information about the default list.

2.4. Changes

The *changes* specify individual changes to the system. Each change is identified by a keyword, usually a verb or noun and followed by parameters.

Extensions can define additional changes with other keywords.

2.4.1. Flags

Some aspects of a change can be modified using a set of *flags*. Flags are indicated using a leading minus sign. The flags are optional but if present must be directly after the keyword.

The following common flags are handled the same way for all changes, both core changes and changes from extensions:

Flag	Description
`-0` to `-9`	Sets the pass in which the change is applied
`-b`	Data is binary and should not be processed with the Template Toolkit
`-i`	Files are changed in-place
`-n`	All errors are ignored
`-t`	Change is only performed if another change in the same class was applied
`-T`	Change is only performed if another change in the same feature was applied

Stages (see Section 3.2.2, “Stages”) are also specified as flags.

2.4.2. Parameters

Every change has its own set of mandatory and optional parameters. The core changes share some common parameters.

The following subsections describe common parameters that are handled the same way for all changes, both core changes and changes from extension.

2.4.2.1. Source File

The *source* file may be specified as path relative to the feature directory or an absolute path on the target system.

2.4.2.2. Destination Path

The *destination* path for changes related to the file system must always be absolute.

2.4.2.3. User

The *user* for a file (symbolic link, etc.) or directory can be specified as a uname or a numeric uid.

For an existing file or directory it is possible to use a minus sign as user in which case the current user of the file is unchanged. If the file does not exist the user of the file is set to root (uid 0).

2.4.2.4. Group

The *group* for a file (symbolic link, etc.) or directory can be specified as a group name or a numeric gid.

For an existing file or directory it is possible to use a minus sign as group in which case the current group of the file is unchanged. If the file does not exist the group of the file is set to root (gid 0).

2.4.2.5. Mode

The *mode* for a file or directory is specified as an octal number.

If a minus sign is used as mode then the current mode of the file is unchanged. If the file does not exist then the mode is set to 0644.

2.4.2.6. SELinux Context

For files (directories, symbolic links, etc.) the core changes accept an optional SELinux context if SELinux support is activated (see Section 7.3, “Settings Reference”). A context can be specified in three formats:

type	SELinux type, expands to system_u:object_r:*type*:s0.
user:role:type	SELinux user, role and type, expands to *user*:*role*:*type*:s0.
user:role:type:range	Complete SELinux context.

The core changes that accept an optional SELinux context will keep the current context of an existing file. If a file is created then the context is defined by the SELinux policy.

2.4.3. Core Changes

The following subsections explain the core changes that are natively supported by Config5, i.e., are not provided by an extension.

2.4.3.1. Copy File

The *copy* change is used to copy a regular file.

Synopsis	copy [-bintT] *source* *destination* *user* *group* *mode* [*context*] [*key*=*value*…]

The parameters *source*, *destination*, *user*, *group*, *mode* and *context* are common parameters (see Section 2.4.2, “Parameters”).

One or more substitutions can be appended as *key*=*value* pairs (see Section 2.3.1, “Substitutions”). If the -b flag is used the file is treated as binary and no substitutions are done.

The content of a file (after applying all append changes for the same file), *user*, *group*, *mode* and *context* are individually compared to the destination file and copied or set when different.

If *destination* exists and is not a file then it will be deleted before creating the file.

Example 2.2. Copy File

```
sample:
  copy sshd_config /etc/sshd/sshd_config root root 600 permitroot=1
  copy authorised_keys /root/.ssh/authorized_keys root root 644 ssh_home_t
```

2.4.3.2. Truncate File

The *truncate* change is used to create a empty file.

Synopsis	`truncate [-intT] destination user group mode [context]`

The parameters *source*, *destination*, *user*, *group*, *mode* and *context* are common parameters (see Section 2.4.2, “Parameters”).

The content of a file (after applying all `append` changes for the same file), *user*, *group*, *mode* and *context* are individually compared to the destination file and copied or set when different.

If *destination* exists and is not a file then it will be deleted before creating the file.

Example 2.3. Truncate File

```
sample:
  truncate /etc/rc.d/rc.local root root 755
```

2.4.3.3. Append to File

The *append* change is used to append data to a file.

Synopsis	`append [-b] source destination [key=value...]`

The parameters *source* and *destination* are common parameters (see Section 2.4.2, “Parameters”).

One or more substitutions can be appended as *key=value* pairs (see Section 2.3.1, “Substitutions”). If the `-b` flag is used the appended data is treated as binary and no substitutions are done.

Important

To append to a file it must be copied or truncated. A corresponding copy or truncate change must be present in exactly one of the features.

Example 2.4. Append to File

```
sample:
  append rc.local /etc/rc.d/rc.local
```

2.4.3.4. Create Directory

The *directory* change is used to create a directory.

Synopsis	`directory [-ntT] destination user group mode [context]`

The parameters *source*, *destination*, *user*, *group*, *mode* and *context* are common parameters (see Section 2.4.2, “Parameters”).

The *user*, *group*, *mode* and *context* are individually compared to the destination path and copied or set when different.

If *destination* exists and is not a directory then it will be deleted before creating the file.

Example 2.5. Create Directory

```
sample:
  directory /local root root 1777
```

2.4.3.5. Create Link

The *link* change is used to create a (hard) link.

Synopsis	`link [-ntT] source destination`

The parameter *destination* is a common parameter (see Section 2.4.2, “Parameters”). The *source* parameter is like the *destination* parameter an absolute path in the file system of the target system.

If *destination* exists and its inode or device id is different then it will be deleted before creating the link.

Example 2.6. Create Link

```
sample:
  link /tmp/a /tmp/b
```

2.4.3.6. Create Symbolic Link

The *symlink* change is used to create a symbolic link.

Synopsis	symlink [-bntT] *value destination user group* [*context*]

The parameters *destination*, *user*, *group* and *context* are common parameters (see Section 2.4.2, “Parameters”). The parameter value is usually a path but may be set to any string.

The substitutions defined for the class or inherited from including classes are applied to value unless the -b flag is used.

The content of the symbolic link, *user*, *group* and *context* are individually compared to the destination path and set when different.

If *destination* exists and is not a symbolic link then it will be deleted before creating the symbolic link.

Example 2.7. Create Symbolic Link

```
sample:
  copy ca.pem /etc/openldap/cacerts/ca.pem root root 644
  symlink ca.pem /etc/openldap/cacerts/12345678.0 root root
```

2.4.3.7. Change Properties

The *properties* change is used to change the properties of an existing file.

Synopsis	properties [-ntT] *destination user group mode* [*context*] [*key=value*...]

The parameters *destination*, *user*, *group*, *mode* and *context* are common parameters (see Section 2.4.2, “Parameters”).

The *user*, *group*, *mode* and *context* are individually compared to the destination file and set when different.

Example 2.8. Change Properties

```
sample:
  properties /var/log/messages - - 644
```

2.4.3.8. Remove

The *remove* change is used to delete a file, directory, symbolic link, link or special file.

Synopsis	remove [-ntT] *destination*

The parameters *destination* is a common parameter (see Section 2.4.2, “Parameters”).

Important

It is currently not possible to recursively remove a directory with the remove change. Run **rm -rf** using the execute change instead (see Section 2.4.3.9, “Execute Shell Command Line”).

Example 2.9. Remove

```
sample:
  remove /etc/openldap/cacerts/ca.pem
  remove /etc/openldap/cacerts/12345678.0
  remove /etc/openldap/cacerts
```

2.4.3.9. Execute Shell Command Line

The *execute* change is used to execute a shell command line.

Synopsis	`execute` [`-bntT`] ...

The entire rest of the line is executed as a shell command line. The line is executed as the user and group of the running **config5** process which is usually both `root`.

The substitutions defined for the class or inherited from including classes are applied to the command line before executing it unless the `-b` flag is used.

The execute change is always performed unless the `-t` or `-T` flag is used but no earlier change triggers it.

Example 2.10. Execute Shell Command Line

```
sample:
  execute service sssd stop ; rm -f rm -f /var/lib/sss/db/* ; service sssd start
```

Chapter 3. Configuration Application

Understanding how the configuration is applied to a system is essential and outlined in this chapter.

3.1. Configuration Processing

The information in the feature set is processed in a well defined order:

- All features are processed in alphabetical order.
- Starting with the class that matches best in each feature (if there is any, see Section 2.3.2, “Best Match”) the referenced classes are processed first from left to right. Classes are processed once every time they are referenced which may be more than once.
- In each class that is processed the changes are processed from top to bottom.

Some changes are applied in the order they are processed (see Section 3.3.3, “Changes”).

3.2. Environment

3.2.1. Root Directory

The root directory that is used when Config5 applies the configuration on a target system is typically /. The root directory can be changed by setting the root system information item, for instance using the `--set` command line option of **config5**.

All core changes support a different root directory. All changes except the execute change will automatically prepend the root directory to all absolute source or destination paths. The execute change is executed but requires the use of substitution by inserting *`[% info_root %]`* where needed.

3.2.2. Stages

A target system may require different configurations throughout its life cycle. Config5 supports this in a generic fashion by defining a set of *stages*. The default configuration defines the following two stages:

Stage	Flag	Description
`installation`	`-I`	System is being installed.
`production`	`-P`	System is up and running (default).

Individual changes can be selectively enabled for only some stages by specifying the stages as flags (see Section 2.4.1, “Flags”). If no stage is specified this means that the change is applied in any stage.

The stage can be set using the `--set` command line option of **config5**.

3.3. Configuration Application

The configuration is applied after processing in a structured manner.

3.3.1. Passes

The configuration is applied in ten *passes*. The pass in which a change is applied can be specified with the flags `-0` to `-9` for each change. The default pass is `1`.

Applying the configuration in multiple passes can be helpful in a couple of scenarios:

- Changes that must be done before or after everything else can be placed in an earlier or later pass.
- Configuration files that are generated by a command that writes a temporary file which is copied to the final destination.

3.3.2. Phases

In each pass the relevant changes are applied in different phases. Currently the following phases are defined and processed in this order:

1. Install or remove packages
2. Delete files and directories
3. Create files and directories
4. Modify files and directories

5. Configure system
6. Execute commands

Each type of change is applied in exactly one phase:

Install or remove packages	(not used by the core changes)
Delete files and directories	`remove`
Create files and directories	`append`, `copy`, `directory`, `link`, `symlink`, `truncate`
Modify files and directories	(not used by the core changes)
Configure system	(not used by the core changes)
Execute commands	`execute`, `service`

For each phase the order in which changes are applied is also defined:

Install or remove packages	All changes are applied in the order the changes were processed (see Section 3.1, “Configuration Processing”).
Delete files and directories	All changes are ordered according to the *reverse alphabetical order* of the destination path and applied in this order. Files are deleted before their parent directory.
Create files and directories	All changes are ordered according to the *alphabetical order* of the destination path and applied in this order. Directories are created before files they contain.
Modify files and directories	All changes are ordered according to the *alphabetical order* of the destination path and applied in this order. Directories are created before files they contain.
Configure system	All changes are applied in the order the changes were processed (see Section 3.1, “Configuration Processing”).
Execute commands	All changes are applied in the order the changes were processed (see Section 3.1, “Configuration Processing”).

3.3.2.1. Data Generation

Before the files are copied in phase `3` of each pass, the content of files are generated. This is done on two steps:

1. The source data of all append and copy changes that are not marked as binary using the `-b` flag is processed using Template Toolkit.
2. The processed data from all `append` changes is appended to the processed data of the respective `copy` or `truncate` changes.

The processed data is not written to the destination files yet but kept in memory for use during the actual configuration application.

Important

Copying large files may use up a lot of memory because of the intermediate copy in memory.

3.3.3. Changes

For each change that is applied **config5** will first check if there are modifications needed. Changes that require no modifications on the target system are not shown in the default verbosity level.

The following subsections describe how each change type is applied.

3.3.3.1. `copy` and `truncate`

When applying a `copy` or `truncate` change **config5** will do it with as few side effects as possible. A file is only replaced or created if it is missing or its content (or file type) changes in which case **config5** does the following:

1. If the file already exists but is not a plain file then it will be deleted first.
2. It will create a temporary file with the new content and set all permissions as specified. If SELinux is enabled and no SELinux context is specified and the file is replaced then the context from the current file is copied to the temporary file.
3. The temporary file is renamed to the real file name which also deletes the old file.

If *in-place* modification is explicitly enabled using the `-i` flag then the existing file is modified directly instead and **config5** does the following:

1. If the file already exists but is not a plain file then it will be deleted first.

2. It will write the new content to the file.

If the file is already present and the content is the same then only the properties such as user, group, mode or SELinux context (if enabled) are modified if necessary.

3.3.3.2. `directory`

If a file exists with the same path as the directory it is removed.

3.3.3.3. `execute`

The command line is executed by the default shell of the user who is running **config5**.

3.3.3.4. `link`

If the link is missing or does not have the same device and inode number as the file it links to **config5** does the following:

1. It will create a temporary link to the source file

2. The temporary link is renamed to the real file name which also deletes the old link.

3.3.3.5. `remove`

If the file or directory exists it will be removed.

Important

Recursively deleting directories is not supported at this time. Use an `execute` change to run **rm** instead.

3.3.3.6. `symlink`

If the symbolic link is missing or does not have the right content **config5** does the following:

1. If the file already exists but is not a symbolic link then it will be deleted first.

2. It will create a temporary symbolic link with the new content and set all permissions as specified. If SELinux is enabled and no SELinux context is specified and the symbolic link is replaced then the context from the current symbolic link is copied to the temporary symbolic link.

3. The temporary symbolic link is renamed to the real file name which also deletes the old symbolic link.

3.3.3.7. properties

The file or directory must exist. The properties such as user, group, mode or SELinux context (if enabled) are modified if necessary.

3.3.4. Triggered Changes

A simple mechanism exists to make the application of changes depend on the modifications of previous changes. The most common case where this is used is to restart a service after updating its configuration files.

Each class and each feature has a *triggered* flag which are cleared before configuration application. Every change that requires a relevant modification (see Section 3.3.4.1, “Relevant Modifications”) to the target system will set the triggered flags of its class and feature. Each change can be flagged to be triggered using either the `-t` or `-T` flag. In this case the change is only applied if the triggered flag of the class (`-t`) or the feature (`-T`) of the change is set.

The following facts are also relevant when working with triggered changes:

- The flags are kept across all phases and passes
- Triggered changes never set the triggered flags

3.3.4.1. Relevant Modifications

Not every modification that is done when a change is applied sets the triggered flags. The following table lists the modifications that set or do not set the triggered flags:

Change	Triggered Flag Set	No Change
`append`	Never	
`copy`	File missing or not a plain file,content changed	Change of user, group, mode or other properties

Change	Triggered Flag Set	No Change
`directory`	Directory missing or not a directory	Change of user, group, mode or other properties
`execute`	Never	
`link`	Link missing or to different inode	
`properties`	Change of user, group, mode or other properties	
`remove`	File or directory existing	
`symlink`	Symbolic link missing or not a symbolic link, content changed	Change of user, group or other properties
`truncate`	File missing or not a plain file,content changed	Change of user, group, mode or other properties

Chapter 4. The config5 Script

This chapter describes the **config5** script and how to use it for configuration verification, application and various management tasks.

4.1. Requirements

The **config5** script is written in Perl and therefore requires a Perl installation with the core modules as well as a number of additional modules.

4.1.1. Perl Modules

The following non-core Perl modules are required to run **config5**:

- `Module::Load`
- `Module::Load::Conditional`
- `Template`

The following modules are only required if a given feature is enabled:

`File::ExtAttr`	Required for SELinux support (see Section 7.3.2, “Operating System Settings”).
`Socket6`	Required for IPv6 support if the IP6 address is a mandatory system information item (see Section 7.3.3, “System Information Settings”).

4.1.2. Other Requirements

These external programs are required if a given feature or component of **config5** is used:

lsb_release	Required for the LSB system information module.

4.2. Running config5

The **config5** script will determine its location and automatically find settings, libraries and the feature set relative to its path.

The libraries and settings are expected to be found in the directories `../lib` and `../etc` respectively. The feature set is expected in `../features` unless the path of the feature set is explicitly set using the `--features` command line option.

4.2.1. Arguments

The arguments to **config5** are a list of features. Only the features listed on the command line are processed in alphabetical order.

If no features are specified then all features of the feature set are processed.

4.3. Applying the Configuration

The configuration is applied by running the config5 script with the `--apply` command line option.

```
# /usr/config5/bin/config5 --apply
```

This is also the default action (the other two are `--check` and `--report`) so the option can be omitted:

```
# /usr/config5/bin/config5
```

While the configuration is applied config5 will use a lock file to make sure that no other instance of config5 is running at the same time. The file is relative to the root directory system item (see Section 6.2.3, “Root”), the default path for the file is `/var/lock/config5` but can be changed (see Section 7.3.1, “General Settings”).

4.3.1. Changing System Information Items

The `--set` command line options can be used to override the system information items that are automatically determined. The argument to `--set` is of the form `key=value` where `key` is the identifier of the system information item and `value` is the value to set:

```
# /usr/config5/bin/config5 --set host=foobar
```

The list of identifiers of all configured system information items can be displayed using `--report systemitems`.

4.3.2. SELinux Support

SELinux support is a built-in feature and typically configured via the global configuration (see Section 7.3.2, “Operating System Settings”). In some situations, such as when running **config5** on diskless clients, it may be required to override this setting via command line. SELinux support is also automatically disabled when the root directory is not `/` but the command line options take precedence.

The `--selinux` command line option enables SELinux support, the `--no-selinux` command line option disables it:

```
# /usr/config5/bin/config5 --no-selinux
```

4.3.3. Dry Run

The `--dummy` command line options will force **config5** to simulate configuration application:

```
# /usr/config5/bin/config5 --apply --dummy
```

No changes to the system will be done. The changes that would be done are printed.

Important

The dry run uses the current system state for the simulation which may not be enough to reflect the changes that would really be done.

For instance a `service start` change that follows a `service stop` change for a service would usually be executed but in dry run mode the current state is not the same as after the `service stop` change was executed.

4.3.4. Disable Configuration Application

The configuration application for a single system can be disabled by creating an empty *disable file*. This is usually helpful when manual maintenance is done on a system and running **config5** would interfere. The file is relative to the root directory system item (see Section 6.2.3, “Root”).

The default path for the file is `/etc/config5-disabled` but can be changed (see Section 7.3.1, “General Settings”).

4.4. Verifying the Configuration

With the `--check` command line option the configuration will only be checked for correctness. Currently the following static checks are performed:

- Syntax errors in the `spec` files or missing spec files.
- Missing source files for copy or append.
- File or directory modified in more than one feature.
- Missing referenced class.
- Missing `copy` or `truncate` change for `append` changes.
- Duplicate classes in a feature.

Ensuring that every append change has a corresponding copy or truncate change cannot be verified statically. To enable this check do the following:

1. Collect the output of **config5 --report rawsystem** from all systems the configuration will be applied to and store each output in an individual file in some directory, typically a network share. The directory must be readable from where you will run config5 --check (if you integrate config5 with Subversion such as described in Chapter 5, *Deployment* then the directory must also be accessible on your Subversion server). The best is to do this periodically.
2. Set the setting `$self->{system}->{variations_directory}` (see Section 7.3.3, “System Information Settings”) to the directory containing the above files.

4.5. Reports

The `--report` command line option can be used to produce a variety of reports. The argument to `--report` is a report type:

`changes`	Report the keywords of all changes.
`classes`	Report all classes of all features.
`classorder`	Report all classes that can be matched in a `spec` file, in the order they are matched (see Section 2.3.2, “Best Match”).

`features`	Produce a normalized output of all changes, classes and features.
`files`	Report all files that are copied or truncated in all classes of all features. This information can be used to manage configuration file upgrades when a new release of an operating system is available.
`matches`	Report the matching class for each feature based on the available system information.
`rawsystem`	Report the current values for all system information items for the system in a form that can be parsed by a shell or used for configuration verification.
`stages`	Report all configured stages.
`system`	Report the current values for all system information items for the system. This is also printed first when applying the configuration.
`systemitems`	Report all configured system information items.

Chapter 5. Deployment

This chapter provides information and tips how to actually deploy Config5 in your organization.

Deploying Config5 to a system is done in two steps:

1. Distributing the Config5 deployment kit including the configuration data on the system

2. Running the **config5** script

Config5 is very flexible and allows for a wide range of approaches for both.

5.1. Distribution of the Configuration

The **config5** script along with its settings and libraries are typically distributed along with the feature set. This set comprises the deployment kit (the `deploy` folder in the distribution). Some options for distributing the deployment kit include:

Network File system	The files of the deployment kit are exported via CIFS or NFS and mounted read-only on the system to configure. If applicable the authentication scheme of the network file system can be used to restrict access.
Version Control System	A version control system such as Subversion or GIT can be used as a distribution point. A system will check out the latest version of the deployment kit.
Remote Copy	The deployment kit is pushed on a system via **rsync**, **scp** or any other tool that does the job.

The right choice of the deployment method largely depends on your work flow, policies and security requirements.

5.1.1. Integration with Subversion

Config5 integrates well with Subversion. The configuration check function of **config5** (see Section 4.4, “Verifying the Configuration”) can easily be used in a pre-commit hook to guard against common errors such as syntax errors and missing configuration files:

```
#!/bin/sh

REPOSITORY="$1"
TRANSACTION="$2"

TEMPDIR=`/bin/mktemp -d`
export-commit "$REPOSITORY" "$TRANSACTION" trunk "$TEMPDIR" 1>&2

"$TEMPDIR/trunk/bin/config5" --check 1>&2

RC=$?
rm -rf "$TEMPDIR"
exit $RC
```

This script relies on an additional script **export-commit** which performs an export of the repository *including the pending commit*:

```
#!/bin/sh

REPOSITORY="$1"
TRANSACTION="$2"
SUBDIR="$3"
DIRECTORY="$4"

svnlook tree --full-paths "$REPOSITORY" -t "$TRANSACTION" "$SUBDIR" | while read path
do
   if [[ $path =~ "\/$" ]]
   then
      mkdir -p "$DIRECTORY/$path"
   else
      svnlook cat "$REPOSITORY" -t "$TRANSACTION" "$path" > "$DIRECTORY/$path";
      chmod 755 "$DIRECTORY/$path"
   fi
done
```

Both scripts can be found in the directory `examples/svn` in the distribution.

5.2. Running config5

Running the **config5** script at the right time is an important decision for a successful deployment and depends on several factors such as the operating system, work flows, policies, etc. Below is a non-conclusive list of possible places from where to run **config5**:

cron	Running **config5** periodically via **cron** is the simplest way to make sure that all running systems receive a new configuration.
anacron	Configuring **anacron** on systems such as laptops and desktops that may be put in sleep mode or hibernation can be a meaningful complement to **cron**. If cron runs config5 often (like, every hour) this may not be necessary.
on boot	After the operating system has booted it is usually best to run **config5** once to get the latest configuration updates immediately, possibly before the services it configures start up.
after package updates	After packages are updated it may be required to overwrite configuration files by running **config5** immediately afterward.

When **config5** is running it uses a lock file to make sure that only one instance of **config5** is running at a time. It is therefore save to run **config5** from different places without additional synchronization means.

5.2.1. Random Delay

If many system are likely to update their configuration at the same time, it may be meaningful to add a random delay before running **config5**. This is usually the case if **config5** is run via **cron** on managed systems. If the deployment kit is on a network file system then all managed systems will otherwise access the file server at the same time. Depending on the file server and file system this may cause all kinds of errors.

The following **bash** code will perform a random five minute delay before running **config5**:

Example 5.1. Random Delay with bash

```
RND=$RANDOM ; let "RND\%=300" ; sleep $RND; /usr/config5/bin/config5
```

If the deployment kit is pulled from a central repository via download or checkout the delay should be placed before the download or checkout commands for similar reasons.

Chapter 6. System Information

Config5 uses an approach where the configuration is selected on the system that is being configured. This chapter explains the mechanism by which **config5** discovers information about the system it runs on.

6.1. The System Information Module

The module used for gathering the system information is exchangeable. Config5 ships with two modules for immediate productive use: `LSB` (the default) and `Hardwired`.

Custom modules can also be added. A sample module (the one that we at *isginf* use in our productive environment) can be found in `examples/modules/System/SystemInfo.pm` in the distribution.

6.2. System Information Items

The system information is organized in *items*. Each item is determined somehow by the system information module and can be used in the class matching process and is available as a variable for substitution.

Each item has a unique name and the following relevant information associated with it:

- Priority for displaying the system information summary (low priority means display first).
- Whether or not the item is optional (`0`) or mandatory (`1`).
- The type of the system information item (`0` for scalar, `1` for ordered list).
- A conversion method for the gathered value of the item (see below).
- The human readable description displayed in the system information summary.

The conversion is useful to modify the raw value gathered by the module. No conversions are used in the default configuration. The following conversions are supported:

none	No conversion.
toupper	Convert all characters to upper case.
tolower	Convert all characters to lower case.
nows	Strip all white space characters.
wstounderscore	Convert all white space characters to underscore.

The available system information items are configured via settings (see Section 7.3.3, “System Information Settings”). The following items are configured by default:

Item	Priority	Mandatory	Type	Conversion	Description
`date`	`10`	no	scalar	none	Architecture
`host`	`20`	yes	scalar	none	Host name
`arch`	`30`	yes	scalar	none	Architecture
`os`	`40`	yes	scalar	none	OS
`release`	`50`	yes	scalar	none	Release
`ip`	`60`	no	scalar	none	IP
`ip6`	`70`	no	scalar	none	IP6
`fqhn`	`80`	no	scalar	none	Fully qualified host name
`fqdn`	`90`	no	scalar	none	Fully qualified domain name
`properties`	`100`	no	list	none	Properties
`root`	`110`	yes	scalar	none	Root directory
`stage`	`120`	yes	scalar	none	Stage

6.2.1. Common System Information Items

The following system information items are determined independent of the chosen module and are always available and are determined from the running operating system:

- `arch`
- `host`
- `ip` (if configured)
- `ip6` (if configured)
- `properties`
- `root`
- `stage`

If the root directory is changed (see Section 3.2.1, “Root Directory”) then only the following items are determined.

- `properties`
- `root`
- `stage`

The remaining mandatory modules must be provided by command line argument or by an alternative system information module such as the `Hardwired` module (see Section 6.3.2, “Hardwired”).

6.2.2. Properties

The properties are read from a file that contains space- or comma-separated words. This file can be provisioned during installation.

6.2.3. Root

The root directory (see Section 3.2.1, “Root Directory”) can be modified with the `--set` command line option when running **config5**. Changing the root directory only makes sense when the configuration is applied to something other than the system where **config5** is run, for instance a diskless client image on the file server that serves it.

The default path of the root directory is taken from the `$self->{os}->{root}` setting (see Section 7.3.2, “Operating System Settings”) and is typically `/`.

6.2.4. Stage

The stage (see Section 3.2.1, “Root Directory”) can be modified with the `--set` command line option when running **config5**.

The default change can be changed with the following setting:

Setting	Default	Description
`$self->{system}->{defaults}->{stage}`	`production`	The default stage.

6.3. System Information Modules

The system information module to use can be changed in the settings (see Section 7.3.3, “System Information Settings”).

6.3.1. LSB

The `LSB` module can determine the OS name and release using **lsb_release**.

The following settings can be used to control the information gathering:

Setting	Default	Description
`$self->{system}->{lsb}->{append_code}`	`0`	Append the OS code name to the OS id with an underscore in between.

6.3.2. Hardwired

The `Hardwired` module can read constants for the active system information items from the Config5 settings. Supplied constants overload the automatically determined information (architecture, host name, etc.):

Setting	Default	Description
`$self->{system}->{hardwired}->{`*`item`*`}`	*`undefined`*	Set the constant value for system information item *`'item'`*.

Chapter 7. Customization

Config5 ships with a reasonable set of default settings for the **config5** script for a normal deployment using the core features.

Systems administrators with special needs or policies can tweak many settings that affect the file system structure or behavior of Config5.

7.1. Files

All files are located under the directory `etc` next to the `bin` directory containing the **config5** script. In the official distribution the files are in `deploy/etc`:

```
$ ls deploy/etc
custom.d  extensions.d  settings.pl
```

All configuration files are basically perl code that modifies the settings data structure which is passed in `$_` by reference.

7.1.1. Order of Processing

The various files are processed in the following order:

1. The main settings file `settings.pl`.
2. All files ending in `.pl` in `extensions.d` in alphabetical order.
3. All files ending in `.pl` in `custom.d` in alphabetical order.

Settings in one file can be removed, replaced or extended by files that are loaded afterward.

7.2. Custom Settings

Custom settings should be placed in a file ending in `.pl` in the directory `etc/custom.d`. A custom settings file should start with the following code:

```
my $self = $_;
```

This is not absolutely necessary but the other settings file all use `$self` and settings can be customized easier using copy-paste.

Each file should also end with the following line:

```
1;
```

The file `examples/settings/custom.pl.example` in the distribution of Config5 can be used as template for a custom configuration file.

7.3. Settings Reference

The following reference lists the various settings. All examples assume that `$self` is defined and holds the reference to the settings structure.

7.3.1. General Settings

The following settings are of general nature:

Setting	Default	Description
`$self->{disable}->{file}`	`/etc/config5-disabled`	The path of the disable file. If this file is present then **config5** will not run. Dry runs via dummy mode are still possible.
`$self->{lock}->{file}`	`/var/lock/config5`	The path of the lock file used to prevent concurrent execution of **config5**. *This file must be on the local file system and not on a network share.*
`$self->{lock}->{timeout}`	`5`	The timeout after which **config5** will abort if the lock file cannot be locked. A value of `0` will block indefinitely until the lock file can be locked.
`$self->{features}->{name}`	`features`	The directory name of the feature set.

Setting	Default	Description
`$self->{spec}->{name}`	`spec`	The name of the specification file in each feature directory.

7.3.2. Operating System Settings

The following settings control additional support for features of the operating system:

Setting	Default	Description
`$self->{os}->{env}->{`*`variable`*`}`	undefine `LANG`	Set environment variable to a given string. To undefine an environment variable set it to `undef`.
`$self->{os}->{root}`	`/`	The root directory.
`$self->{os}->{selinux}->{enable}`	`0`	Enables support for SELinux if set to `1` and disables it if set to `0`.
`$self->{os}->{syslog}->{enable}`	`1`	Enables logging via `syslog` if set to 1 and disables it if set to `0`.
`$self->{os}->{syslog}->{identifier}`	`config5`	The identifier for log entries if logging via `syslog` is enabled.
`$self->{os}->{syslog}->{facility}`	`local5`	The facility for log entries if logging via `syslog` is enabled.
`$self->{os}->{umask}`	`022`	The file and directory creating mask for shell commands.

7.3.3. System Information Settings

The following settings control the system information collection.

Setting	Default	Description
`$self->{system}->{class}`	`Config5::System::LSB`	The class that collects the system information.
`$self->{system}->{items}->{`*`item`*`}`	`see Section 6.2, “System Information Items”`	The known system information items.
`$self->{system}->{properties}->{file}`	`/etc/config5-properties`	The file name of the properties file.
`$self->{system}->{variations_directory}`	*`undefined`*	Directory containing files variations of system information (see Section 4.4, “Verifying the Configuration”).

7.3.4. Built-in Changes

The following settings control the behavior of the built-in changes.

Setting	Default	Description
`$self->{builtin}->{execute}->{allow_chroot}`	`0`	Apply execute changes if root is not `/`.

7.3.5. Verbosity

The amount of information that is printed at which verbosity level can be configured by changing the following settings.

Setting	Default	Description
`$self->{display}->{entities}->{system}`	`1`	Print the system information.
`$self->{display}->{entities}->{pass}`	`1`	Print the pass numbers.
`$self->{display}->{entities}->{status}`	`1`	print the status.

The default verbosity level is `1` which means that by default all three items are printed.

7.3.6. Stages

The stages and the associated flags can be altered by defining `$self->{stage}->{stages}` like this:

```
# Stages of the target system
$self->{stage}->{stages} =
{
   'installation' => 'I',
   'startup' => 'S'
   'production' => 'P'
};
```

The flags defined for the stages must not conflict with the regular flags (see Section 2.4.1, “Flags”).

7.3.7. Change Keywords

The keywords for changes can be altered by modifying `$self->{change}->{keywords}` like this:

```
# Replace the keyword 'directory' by 'mkdir'
delete $self->{change}->{keywords}->{directory};
$self->{change}->{keywords}->{mkdir} = 'directory';

# Add 'package' as keyword for 'yum'
$self->{change}->{keywords}->{package} = 'yum';
```

Similarly the identifier used for changes when displaying them can be changed by modifying `$self->{display}->{keywords}`:

```
# Replace the keyword 'directory' by 'mkdir'
$self->{display}->{keywords}->{directory} = 'mkdir';

# Add 'package' as keyword for 'yum'
$self->{display}->{keywords}->{yum} = 'package';
```

The names for the core changes are:

- `copy`
- `truncate`
- `append`
- `directory`
- `link`
- `symlink`
- `properties`
- `remove`
- `execute`

7.3.8. Class Match Order

The search order for the best class match in each feature can be customized by modifying `the list referenced by $self->{class}->{match}`. Each list entry is a string containing placeholders of the form *`{item}`* which are replaced by the value of the respective system information item.

The default search order is the following:

1. `{host}_{os}_{release}_{arch}`
2. `{host}_{os}_{release}`
3. `{host}_{os}_{arch}`
4. `{host}_{os}`
5. `{host}_{arch}`
6. `{host}`
7. `{properties}_{os}_{release}_{arch}`

8. `{properties}_{os}_{release}`
9. `{properties}_{os}_{arch}`
10. `{properties}_{os}`
11. `{properties}_{arch}`
12. `{properties}`
13. `{os}_{release}_{arch}`
14. `{os}_{release}`
15. `{os}_{arch}`
16. `{os}`

Appendix A. Man Pages

A.1. config5

```
NAME
    config5 - 5th Generation Unix Configuration Management Utility

SYNOPSIS
    config5 [-vqdnhmV] [--selinux|--no-selinux] [--set *key=value*]
    [--apply|--check|--report *type*] [--features dir] [feature...]

DESCRIPTION
    config5 is the main script of the *Config5* Unix Configuration
    Management Kit. It can apply a configuration (feature set), check
    configuration metadata for consistency and give various reports on the
    configuration and the system it is run on.

    When run without arguments it will perform all operations with the full
    set of features. Otherwise it will only process the features specified.
    The feature set location defaults to 'features' in the *Config5* base
    directory and can be altered with the --features otion.

    Information about the system a configuration is appled to is collected
    from data on the system where config5 is run. The collected values can
    be altered using the --set option.

    For detailed documentation consult the "Config5 Systems Administrator's
    Guide".

OPTIONS
    --apply
        Apply the configuration. Either the entire feature set or only the
        features specified on the command line are applied.

    --check
        Perform consistency checks of the entire configuration. Checks
        include syntax checks of the specification files in all features,
        checks for references to missing files and more.

    --report *type*
        Print various reports. The report *type* can be one of:

        *changes*
            Report the keywords for all changes.

        *classes*
            Report all classes in alphabetical order.

        *classorder*
            Report the classes used to match the best class in each feature.
            The classes are reported in the order they are checked and are
            generated using the gathered system information and the values
            set using --set.

        *features*
            Report all classes and changes for all features in a normalized
            form.

        *files*
            Report all files that are modified by any class of all features.

        *matches*
            Report the matching class for each feature using the gathered
            system information and the values set using --set.

        *stages*
            Report all defined target system stages.

        *system*
            Show the various values of the system information.
```

```
        *rawsystem*
            Show the various values of the system information as parsable
            output. The output can be sourced by a shell.

        *systemitems*
            Report all configured system information items with description,
            identifier and other meta information.

        For all reports relating to features the report comprises either all
        features or only the features specified on the command line.

    --set *key=value*
        Change the value of one of the system information items.

    --features dir
        Sets the directory containing the feature set.

    --set *key=value*
        Change the value of one of the system information items.

    --selinux or --no-selinux
        Activates or deactivates SELinux support, overrides the
        configuration in the settings.

    --help, -h
        Print a brief help message and exit.

    --man, -m
        Print this manual page and exit.

    --version, -V
        Print the version number of the script and exit.

    --verbose, -v
        Enable verbose messages on the process. Can be specified multiple
        times to increase verbosity.

    --quiet, -q
        Only print errors and warning. This option overrides --verbose.

    --debug, -d
        Enable debugging output. Can be specified multiple times to increase
        the debug level.

    --dummy, -n
        Activate dummy mode where no actual changes are made.

EXIT VALUES
    If there were no errors the program exists with return code 0. Otherwise
    the return code is 1.

EXAMPLES
    "config5"
        Apply the entire configuration.

    "config5 --set root=/var/diskless/host1 --set host=host1"
        Apply the configuration for a diskless client on the server.

    "config5 --report info"
        Report information about the system.

    "config5 --dummy"
        Simulate application of the configuration.

    "config5 --check"
        Check the feature set for consistency and syntax errors.

    "config5 --set host=sample --report matches cifs nfs"
        Report the matching class of features 'nfs' and 'cifs' for host
        'sample'.
```

Appendix B. Error Messages

B.1. Runtime Errors

Disabled via '*file*'	Configuration application is disabled because the named disable file is present (see Section 4.3.4, “Disable Configuration Application”). Remove the file to enable configuration application.
Unable to lock 'file'	A lock could not be obtained for the lock file. Verify that the lock file can be created, is writable and on a local file system. Increase the wait time for obtaining the lock (see Section 7.3.1, “General Settings”).

B.2. System Errors

System errors are fatal, no changes will be done to the system.

Cannot run 'lsb_release -…' to determine …	The **lsb_release** command could not be run. Install or deploy the package containing the **lsb_release** command on all systems where **config5** is run with the same settings.
Perl module '*module*' is not installed …	The named perl module is not installed but either generally required or required because of an optional functionality that is enabled. Install or deploy the named module on all systems where **config5** is run with the same settings.

B.3. Setting Errors

Setting errors are fatal, no changes will be done to the system.

Cannot load module '*module*' for keyword in …	The module of the handling class for the change failed to load.
Cannot read feature list in '*directory*'	The features directory could not be read.
Mandatory system information item '*item*' could not be determined	The settings list the system information item as mandatory but the system information module did not set a value for it.
No handling class for keyword '*keyword*' in …	The settings do not specify a class to handle changes with this keyword.
Undefined system information item '*item*'	An undefined system information item was used in the class match list (see Section 7.3.8, “Class Match Order”.)

B.4. Configuration Data Errors

Configuration data errors are fatal, no changes will be done to the system.

Change '*change*' in feature '*feature*' without class	A change was correctly parsed but no class was declared.
Conflict for '*file*' between …	The same file or directory is configured in different features.
Duplicate class '*class*' in feature '*feature*'	The feature contains two declarations of the same class.
Feature '*feature*' does not contain a '…' file	The feature does not contain a spec file.
'*file*' is of the wrong type for …	An `append` change was used for a path that is not a plain file.
'*file*' needs to be created for …	An `append` change for a file was found for which there is no matching `touch` or `copy` change in the same pass.
Garbage '…' in …	Unexpected arguments were found in a change.
Insufficient arguments in …	The change requires more arguments.
Invalid flag '*flag*' for …	The flag is invalid for the change.
Invalid mode '*mode*' in …	The mode is invalid. See Section 2.4.2, “Parameters” for a description of the mode parameter.

Invalid SELinux context '*context*' in …	The SELinux context is invalid. See Section 2.4.2.6, “SELinux Context” for a list of supported context formats.
Invalid substitution '*substitution*' in …	The substitution is invalid.
No file '*file*' in …	The file could not be found in the file system of the target system or in the feature.
Path '*path*' is not absolute in …	The path of the file or directory is not absolute.
Problem with template …	Template Toolkit failed to process the content of the source file.
Recursive reference to class '*class*' in class '*parent*'	A class declaration references an undefined class.
Reference to undefined class '*class*' in class '*class*' in feature '*feature*'	A class declaration references an undefined class.
Unable to append data in '*source*' to '*file*'	The data from the source file could not be read. Verify that the source file exists and is readable.
Unknown change keyword '*keyword*' in …	The change with this keyword is unknown.
Unknown group '*group*' in …	The group is unknown or invalid. See Section 2.4.2, “Parameters” for a description of the group parameter.
Unknown user '*user*' in …	The user is unknown or invalid. See Section 2.4.2, “Parameters” for a description of the user parameter.
Unparsable line *number* in '…' file for feature '*feature*': …	The line is not a valid class declaration or change.

B.5. Configuration Application Errors

The following errors are reported for changes as they are applied but do not interrupt the application of other changes.

Command '*command*' returned *error* in …	The command returned a non-zero return value. Increase the verbosity level to see the output and error messages of the command.
Failed to correctly apply change in …	The change could not be correctly applied.
Unable to chcon '*path*' to *context* in …	The SELinux context of the file or directory could not be set. Verify that the given path is on a file system that has SELinux support.
Unable to chmod '*file*' to mode *mode* in …	The mode bits of the file or directory could not be set.
Unable to chown '*path*' to *uid*:*gid* in …	The user and group of the file or directory could not be set.
Unable to create directory '*directory*' in …	The directory could not be created. Verify that the directory containing the file exists and that it is in a local, writable file system.
Unable to create link '*link*' to '*source*' in …	The link to the source path could not be created. Verify that the source file exists and is in the same file system as the link. Verify that the directory containing the link exists and that it is in a local, writable file system.
Unable to create symbolic link 'link' in …	The symbolic link could not be created. Verify that the directory containing the symbolic link exists and that it is in a local, writable file system.
Unable to get properties of '*file*' in …	The properties of an existing file could not be read.
Unable to open '*file*' for writing in …	The file could not be opened for writing. Verify that the directory containing the file exists and that it is in a local, writable file system.
Unable to remove 'path' in …	The file or directory could not be removed Verify that the file or directory is in a local, writable file system. When a directory is removed it must be empty.
Unable to rename temporary to '*file*' in …	The temporary file could not be renamed to the file.
Unable to write to '*file*' in …	An error occurred while writing data to the file. Verify that the file system containing the file has enough free space.

Appendix C. GNU General Public License version 3

Version 3, 29 June 2007

Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program—to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS

0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To “convey” a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays “Appropriate Legal Notices” to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

1. Source Code.

The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The “System Libraries” of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work’s System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

3. Protecting Users’ Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work’s users, your or third parties’ legal rights to forbid circumvention of technological measures.

4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program’s source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

a. The work must carry prominent notices stating that you modified it, and giving a relevant date.

b. The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to “keep intact all notices”.

c. You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

d. If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an “aggregate” if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation’s users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

a. Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b. Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c. Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d. Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e. Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or

for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

“Additional permissions” are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a. Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b. Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c. Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or

d. Limiting the use for publicity purposes of names of licensors or authors of the material; or

e. Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or

f. Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party’s predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

11. Patents.

A “contributor” is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor’s “contributor version”.

A contributor’s “essential patent claims” are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, “control” includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor’s essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a “patent license” is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To “grant” such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. “Knowingly relying” means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient’s use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is “discriminatory” if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others’ Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if

you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

```
one line to give the program's name and a brief idea of what it does.
Copyright (C) year name of author

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program.  If not, see http://www.gnu.org/licenses/.
```

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
program Copyright (C) year name of author
This program comes with ABSOLUTELY NO WARRANTY; for details type ‘show w’.
This is free software, and you are welcome to redistribute it
under certain conditions; type ‘show c’ for details.
```

The hypothetical commands ‘`show w`’ and ‘`show c`’ should show the appropriate parts of the General Public License. Of course, your program’s commands might be different; for a GUI interface, you would use an “about box”.

You should also get your employer (if you work as a programmer) or school, if any, to sign a “copyright disclaimer” for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see http://www.gnu.org/licenses/.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read http://www.gnu.org/philosophy/why-not-lgpl.html.

Appendix D. GNU Free Documentation License

Version 1.3, 3 November 2008

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “publisher” means any person or entity that distributes copies of the Document to the public.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as

“Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document’s license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled “History”, Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled “History” in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section.

You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties — for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this

License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See Copyleft [http://www.gnu.org/copyleft/].

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright © YEAR YOUR NAME

Permission is granted to copy, distribute and/or modify this document under the
terms of the GNU Free Documentation License, Version 1.3 or any later version
published by the Free Software Foundation; with no Invariant Sections, no
Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in
the section entitled “GNU Free Documentation License”.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with… Texts.” line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts
```

```
being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Glossary

Config5 Concepts

Change	A change defines a single modification to the configuration. Changes always belong to a class. See Also Class.
Class	A class is part of a feature and defines a set of changes. When applying the configuration Config5 searches for the best matching class in each feature. A class can include other classes of the same feature. Substitutions for files defined at class level are available in all changes of the class or any included class. See Also Feature, Change.
Deployment Kit	The deployment kit is that part of the Config5 distribution that needs to be present on all managed systems. By default this is everyting in the `deploy` folder in the distribution and contains scripts, library files, settings and the feature set.
Feature	A feature groups together changes that belong together according to some criteria, including all files needed for these changes. Features are structured into classes. See Also Class, Change.
Feature Set	The feature set groups together all features of a deployable configuration. See Also Feature.
Pass	The configuration is applied in ten passes, in each pass all phases are performed once. Changes are flagged to run in a given pass. See Also Feature, Feature.
Phase	The application of the configuration of one pass is structured into phases. Each phase groups all changes that affect a given configurable aspect of a system such as files, packages, services, etc. See Also Pass.
Specification File	The specification file is part of a feature and contains the classed and changes. See Also Feature.
Stage	The stage defines the state a target system is currently in when the configuration is applied. Only the changes that are intended to be applied at this stage are applied. See Also Change.
Substitution	Substitutions are keywords that are substituted by a value using Template Toolkit. They are specified in class definitions and in changes that support it. See Also Class, Change.

Bibliography

Web Resources

[1] *Bcfg2 Project Website*. *http://trac.mcs.anl.gov/projects/bcfg2*.

[2] *Cfengine Product Website*. *http://www.cfengine.org/*.

[3] *LCFG Product Website*. *http://www.lcfg.org/*.

[4] *Puppet Product Website*. *http://www.puppetlabs.com/*.

[5] *Template Toolkit Home Page*. *http://template-toolkit.org/*.

Papers

[6] *TemplateTree II: The Post-Installation Setup Tool*. Tobias Oetiker. Copyright © 2001. 179-185 of the Proceedings of LISA ’01: Fifteenth Large Installation Systems Administration Conference.