



Bacula® Bacula Miscellaneous Guide

It comes in the night and sucks the essence from your computers.

Kern Sibbald

August 26, 2015

This manual documents Bacula version 5.0.0 (26 January 2010)

Copyright © 1999-2010, Free Software Foundation Europe e.V.

Bacula ® is a registered trademark of Kern Sibbald.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 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".

Contents

1	Python Scripting	1
1.1	Python Configuration	1
1.2	Bacula Events	2
1.3	Python Objects	2
1.4	Python Console Command	5
1.5	Debugging Python Scripts	5
1.6	Python Example	6
2	Variable Expansion	9
2.1	General Functionality	9
2.2	Bacula Variables	9
2.3	Full Syntax	10
2.4	Semantics	11
2.5	Examples	11
3	Using Stunnel to Encrypt Communications	13
3.1	Communications Ports Used	13
3.2	Encryption	13
3.3	A Picture	14
3.4	Certificates	14
3.5	Securing the Data Channel	14
3.6	Data Channel Configuration	15
3.7	Stunnel Configuration for the Data Channel	15
3.8	Starting and Testing the Data Encryption	16
3.9	Encrypting the Control Channel	16
3.10	Control Channel Configuration	17
3.11	Stunnel Configuration for the Control Channel	17

3.12	Starting and Testing the Control Channel	18
3.13	Using stunnel to Encrypt to a Second Client	18
3.14	Creating a Self-signed Certificate	19
3.15	Getting a CA Signed Certificate	20
3.16	Using ssh to Secure the Communications	20
4	DVD Volumes	21
4.1	DVD Specific SD Directives	21
4.2	Edit Codes for DVD Directives	22
4.3	DVD Specific Director Directives	23
4.4	Other Points	23
5	Bacula Projects	25
6	The internal database is not supported, please do not use it.	27
6.1	Internal Bacula Database	27
7	Bacula Copyright, Trademark, and Licenses	29
7.1	FDL	29
7.2	GPL	29
7.3	LGPL	29
7.4	Public Domain	29
7.5	Trademark	30
7.6	Fiduciary License Agreement	30
7.7	Disclaimer	30
8	GNU Free Documentation License	31
8.1	Table of Contents	37
8.2	GNU GENERAL PUBLIC LICENSE	37
8.3	Preamble	37
8.4	TERMS AND CONDITIONS	38
8.5	How to Apply These Terms to Your New Programs	41
8.6	Table of Contents	42
8.7	GNU LESSER GENERAL PUBLIC LICENSE	42
8.8	Preamble	42
8.9	TERMS AND CONDITIONS	43

Chapter 1

Python Scripting

You may be asking what Python is and why a scripting language is needed in Bacula. The answer to the first question is that Python is an Object Oriented scripting language with features similar to those found in Perl, but the syntax of the language is much cleaner and simpler. The answer to why have scripting in Bacula is to give the user more control over the whole backup process. Probably the simplest example is when Bacula needs a new Volume name, with a scripting language such as Python, you can generate any name you want, based on the current state of Bacula.

1.1 Python Configuration

Python must be enabled during the configuration process by adding a `--with-python`, and possibly specifying an alternate directory if your Python is not installed in a standard system location. If you are using RPMs you will need the `python-devel` package installed.

When Python is configured, it becomes an integral part of Bacula and runs in Bacula's address space, so even though it is an interpreted language, it is very efficient.

When the Director starts, it looks to see if you have a **Scripts Directory** Directive defined (normal default `/etc/bacula/scripts`, if so, it looks in that directory for a file named **DirStartUp.py**. If it is found, Bacula will pass this file to Python for execution. The **Scripts Directory** is a new directive that you add to the Director resource of your `bacula-dir.conf` file.

Note: Bacula does not install Python scripts by default because these scripts are for you to program. This means that with a default installation with Python enabled, Bacula will print the following error message:

```
09-Jun 15:14 bacula-dir: ERROR in pythonlib.c:131 Could not import
Python script /etc/bacula/scripts/DirStartUp. Python disabled.
```

The source code directory **examples/python** contains sample scripts for `DirStartUp.py`, `SDStartUp.py`, and `FDStartUp.py` that you might want to use as a starting point. Normally, your scripts directory (at least where you store the Python scripts) should be writable by Bacula, because Python will attempt to write a compiled version of the scripts (e.g. `DirStartUp.pyc`) back to that directory.

When starting with the sample scripts, you can delete any part that you will not need, but you should keep all the Bacula Event and Job Event definitions. If you do not want a particular event, simply replace the existing code with a **noop = 1**.

1.2 Bacula Events

A Bacula event is a point in the Bacula code where Bacula will call a subroutine (actually a method) that you have defined in the Python StartUp script. Events correspond to some significant event such as a Job Start, a Job End, Bacula needs a new Volume Name, ... When your script is called, it will have access to all the Bacula variables specific to the Job (attributes of the Job Object), and it can even call some of the Job methods (subroutines) or set new values in the Job attributes, such as the Priority. You will see below how the events are used.

1.3 Python Objects

There are four Python objects that you will need to work with:

The Bacula Object The Bacula object is created by the Bacula daemon (the Director in the present case) when the daemon starts. It is available to the Python startup script, **DirStartup.py**, by importing the Bacula definitions with **import bacula**. The methods available with this object are described below.

The Bacula Events Class You create this class in the startup script, and you pass it to the Bacula Object's **set_events** method. The purpose of the Bacula Events Class is to define what global or daemon events you want to monitor. When one of those events occurs, your Bacula Events Class will be called at the method corresponding to the event. There are currently three events, JobStart, JobEnd, and Exit, which are described in detail below.

The Job Object When a Job starts, and assuming you have defined a JobStart method in your Bacula Events Class, Bacula will create a Job Object. This object will be passed to the JobStart event. The Job Object has a good number of read-only members or attributes providing many details of the Job, and it also has a number of writable attributes that allow you to pass information into the Job. These attributes are described below.

The Job Events Class You create this class in the JobStart method of your Bacula Events class, and it allows you to define which of the possible Job Object events you want to see. You must pass an instance of your Job Events class to the Job Object **set_events()** method. Normally, you will probably only have one Job Events Class, which will be instantiated for each Job. However, if you wish to see different events in different Jobs, you may have as many Job Events classes as you wish.

The first thing the startup script must do is to define what global Bacula events (daemon events), it wants to see. This is done by creating a Bacula Events class, instantiating it, then passing it to the **set_events** method. There are three possible events.

JobStart This Python method, if defined, will be called each time a Job is started. The method is passed the class instantiation object as the first argument, and the Bacula Job object as the second argument. The Bacula Job object has several built-in methods, and you can define which ones you want called. If you do not define this method, you will not be able to interact with Bacula jobs.

JobEnd This Python method, if defined, will be called each time a Job terminates. The method is passed the class instantiation object as the first argument, and the Bacula Job object as the second argument.

Exit This Python method, if defined, will be called when the Director terminates. The method is passed the class instantiation object as the first argument.

Access to the Bacula variables and methods is done with:

```
import bacula
```

The following are the read-only attributes provided by the bacula object.

Name

ConfigFile

WorkingDir

Version string consisting of "Version Build-date"

A simple definition of the Bacula Events Class might be the following:

```
import sys, bacula
class BaculaEvents:
    def JobStart(self, job):
        ...
```

Then to instantiate the class and pass it to Bacula, you would do:

```
bacula.set_events(BaculaEvents()) # register Bacula Events wanted
```

And at that point, each time a Job is started, your BaculaEvents JobStart method will be called.

Now to actually do anything with a Job, you must define which Job events you want to see, and this is done by defining a JobEvents class containing the methods you want called. Each method name corresponds to one of the Job Events that Bacula will generate.

A simple Job Events class might look like the following:

```
class JobEvents:
    def NewVolume(self, job):
        ...
```

Here, your JobEvents class method NewVolume will be called each time the Job needs a new Volume name. To actually register the events defined in your class with the Job, you must instantiate the JobEvents class and set it in the Job **set_events** variable. Note, this is a bit different from how you registered the Bacula events. The registration process must be done in the Bacula JobStart event (your method). So, you would modify Bacula Events (not the Job events) as follows:

```
import sys, bacula
class BaculaEvents:
    def JobStart(self, job):
        events = JobEvents()          # create instance of Job class
        job.set_events(events)         # register Job events desired
        ...
```

When a job event is triggered, the appropriate event definition is called in the JobEvents class. This is the means by which your Python script or code gets control. Once it has control, it may read job attributes, or set them. See below for a list of read-only attributes, and those that are writable.

In addition, the Bacula **job** object in the Director has a number of methods (subroutines) that can be called. They are:

set_events The set_events method takes a single argument, which is the instantiation of the Job Events class that contains the methods that you want called. The method names that will be called must correspond to the Bacula defined events. You may define additional methods but Bacula will not use them.

run The run method takes a single string argument, which is the run command (same as in the Console) that you want to submit to start a new Job. The value returned by the run method is the JobId of the job that started, or -1 if there was an error.

write The write method is used to be able to send print output to the Job Report. This will be described later.

cancel The cancel method takes a single integer argument, which is a JobId. If JobId is found, it will be canceled.

DoesVolumeExist The DoesVolumeExist method takes a single string argument, which is the Volume name, and returns 1 if the volume exists in the Catalog and 0 if the volume does not exist.

The following attributes are read/write within the Director for the **job** object.

Priority Read or set the Job priority. Note, that setting a Job Priority is effective only before the Job actually starts.

Level This attribute contains a string representing the Job level, e.g. Full, Differential, Incremental, ... if read. The level can also be set.

The following read-only attributes are available within the Director for the **job** object.

Type This attribute contains a string representing the Job type, e.g. Backup, Restore, Verify, ...

JobId This attribute contains an integer representing the JobId.

Client This attribute contains a string with the name of the Client for this job.

NumVols This attribute contains an integer with the number of Volumes in the Pool being used by the Job.

Pool This attribute contains a string with the name of the Pool being used by the Job.

Storage This attribute contains a string with the name of the Storage resource being used by the Job.

Catalog This attribute contains a string with the name of the Catalog resource being used by the Job.

MediaType This attribute contains a string with the name of the Media Type associated with the Storage resource being used by the Job.

Job This attribute contains a string containing the name of the Job resource used by this job (not unique).

JobName This attribute contains a string representing the full unique Job name.

JobStatus This attribute contains a single character string representing the current Job status. The status may change during execution of the job. It may take on the following values:

C Created, not yet running

R Running

B Blocked

T Completed successfully

E Terminated with errors

e Non-fatal error

f Fatal error

D Verify found differences

A Canceled by user

F Waiting for Client

S Waiting for Storage daemon

m Waiting for new media

M Waiting for media mount

s Waiting for storage resource

- j** Waiting for job resource
- c** Waiting for client resource
- d** Waiting on maximum jobs
- t** Waiting on start time
- p** Waiting on higher priority jobs

Priority This attribute contains an integer with the priority assigned to the job.

CatalogRes tuple consisting of (DBName, Address, User, Password, Socket, Port, Database Vendor) taken from the Catalog resource for the Job with the exception of Database Vendor, which is one of the following: MySQL, PostgreSQL, SQLite, Internal, depending on what database you configured.

VolumeName After a Volume has been purged, this attribute will contain the name of that Volume. At other times, this value may have no meaning.

The following write-only attributes are available within the Director:

JobReport Send line to the Job Report.

VolumeName Set a new Volume name. Valid only during the NewVolume event.

1.4 Python Console Command

There is a new Console command named **python**. It takes a single argument **restart**. Example:

```
python restart
```

This command restarts the Python interpreter in the Director. This can be useful when you are modifying the DirStartUp script, because normally Python will cache it, and thus the script will be read one time.

1.5 Debugging Python Scripts

In general, you debug your Python scripts by using print statements. You can also develop your script or important parts of it as a separate file using the Python interpreter to run it. Once you have it working correctly, you can then call the script from within the Bacula Python script (DirStartUp.py).

If you are having problems loading DirStartUp.py, you will probably not get any error messages because Bacula can only print Python error messages after the Python interpreter is started. However, you may be able to see the error messages by starting Bacula in a shell window with the **-d1** option on the command line. That should cause the Python error messages to be printed in the shell window.

If you are getting error messages such as the following when loading DirStartUp.py:

```
Traceback (most recent call last):
  File "/etc/bacula/scripts/DirStartUp.py", line 6, in ?
    import time, sys, bacula
ImportError: /usr/lib/python2.3/lib-dynload/timemodule.so: undefined
symbol: PyInt_FromLong
bacula-dir: pythonlib.c:134 Python Import error.
```

It is because the DirStartUp script is calling a dynamically loaded module (timemodule.so in the above case) that then tries to use Python functions exported from the Python interpreter (in this case PyInt_FromLong). The way Bacula is currently linked with Python does not permit this. The solution to the problem is to put such functions (in this case the import of time into a separate Python script, which will do your calculations and return the values you want. Then call (not import) this script from the Bacula DirStartUp.py script, and it all should work as you expect.

1.6 Python Example

An example script for the Director startup file is provided in `examples/python/DirStartup.py` as follows:

```
#
# Bacula Python interface script for the Director
#

# You must import both sys and bacula
import sys, bacula

# This is the list of Bacula daemon events that you
# can receive.
class BaculaEvents(object):
    def __init__(self):
        # Called here when a new Bacula Events class is
        # is created. Normally not used
        noop = 1

    def JobStart(self, job):
        """
        Called here when a new job is started. If you want
        to do anything with the Job, you must register
        events you want to receive.
        """
        events = JobEvents()          # create instance of Job class
        events.job = job               # save Bacula's job pointer
        job.set_events(events)         # register events desired
        sys.stderr = events            # send error output to Bacula
        sys.stdout = events            # send stdout to Bacula
        jobid = job.JobId; client = job.Client
        numvols = job.NumVols
        job.JobReport="Python Dir JobStart: JobId=%d Client=%s NumVols=%d\n" % (jobid,client,numvols)

# Bacula Job is going to terminate
def JobEnd(self, job):
    jobid = job.JobId
    client = job.Client
    job.JobReport="Python Dir JobEnd output: JobId=%d Client=%s.\n" % (jobid, client)

# Called here when the Bacula daemon is going to exit
def Exit(self, job):
    print "Daemon exiting."

bacula.set_events(BaculaEvents()) # register daemon events desired

"""
    These are the Job events that you can receive.
"""
class JobEvents(object):
    def __init__(self):
        # Called here when you instantiate the Job. Not
        # normally used
        noop = 1

    def JobInit(self, job):
        # Called when the job is first scheduled
        noop = 1

    def JobRun(self, job):
        # Called just before running the job after initializing
        # This is the point to change most Job parameters.
        # It is equivalent to the JobRunBefore point.
        noop = 1

    def NewVolume(self, job):
        # Called when Bacula wants a new Volume name. The Volume
        # name returned, if any, must be stored in job.VolumeName
        jobid = job.JobId
        client = job.Client
        numvol = job.NumVols;
        print job.CatalogRes
        job.JobReport = "JobId=%d Client=%s NumVols=%d" % (jobid, client, numvol)
        job.JobReport="Python before New Volume set for Job.\n"
```

```
Vol = "TestA-%d" % numvol
job.JobReport = "Exists=%d TestA-%d" % (job.DoesVolumeExist(Vol), numvol)
job.VolumeName="TestA-%d" % numvol
job.JobReport="Python after New Volume set for Job.\n"
return 1

def VolumePurged(self, job):
    # Called when a Volume is purged. The Volume name can be referenced
    # with job.VolumeName
    noop = 1
```


Chapter 2

Variable Expansion

Please note that as of version 1.37, the Variable Expansion is deprecated and replaced by Python scripting (not yet documented).

Variable expansion is somewhat similar to Unix shell variable expansion. Currently (version 1.31), it is used only in format labels, but in the future, it will most likely be used in more places.

2.1 General Functionality

This is basically a string expansion capability that permits referencing variables, indexing arrays, conditional replacement of variables, case conversion, substring selection, regular expression matching and replacement, character class replacement, padding strings, repeated expansion in a user controlled loop, support of arithmetic expressions in the loop start, step and end conditions, and recursive expansion.

When using variable expansion characters in a Volume Label Format record, the format should always be enclosed in double quotes ("").

For example, `${HOME}` will be replaced by your home directory as defined in the environment. If you have defined the variable `xxx` to be `Test`, then the reference `${xxx:p/7/Y/r}` will right pad the contents of `xxx` to a length of seven characters filling with the character `Y` giving `YYYTest`.

2.2 Bacula Variables

Within Bacula, there are three main classes of variables with some minor variations within the classes. The classes are:

Counters Counters are defined by the **Counter** resources in the Director's conf file. The counter can either be a temporary counter that lasts for the duration of Bacula's execution, or it can be a variable that is stored in the catalog, and thus retains its value from one Bacula execution to another. Counter variables may be incremented by postfixing a plus sign (+ after the variable name).

Internal Variables Internal variables are read-only, and may be related to the current job (i.e. Job name), or maybe special variables such as the date and time. The following variables are available:

- Year – the full year
- Month – the current month 1-12
- Day – the day of the month 1-31
- Hour – the hour 0-24
- Minute – the current minute 0-59

Second – the current second 0-59

WeekDay – the current day of the week 0-6 with 0 being Sunday

Job – the job name

general – the Director's name

Level – the Job Level

Type – the Job type

JobId – the JobId

JobName – the unique job name composed of Job and date

Storage – the Storage daemon's name

Client – the Client's name

NumVols – the current number of Volumes in the Pool

Pool – the Pool name

Catalog – the Catalog name

MediaType – the Media Type

Environment Variables Environment variables are read-only, and must be defined in the environment prior to executing Bacula. Environment variables may be either scalar or an array, where the elements of the array are referenced by subscripting the variable name (e.g. `${Months[3]}`). Environment variable arrays are defined by separating the elements with a vertical bar (`—`), thus `set Months="Jan—Feb—Mar—Apr—..."` defines an environment variable named **Month** that will be treated as an array, and the reference `${Months[3]}` will yield **Mar**. The elements of the array can have differing lengths.

2.3 Full Syntax

Since the syntax is quite extensive, below, you will find the pseudo BNF. The special characters have the following meaning:

```
::=      definition
( )      grouping if the parens are not quoted
|        separates alternatives
'/'      literal / (or any other character)
CAPS     a character or character sequence
*        preceding item can be repeated zero or more times
?        preceding item can appear zero or one time
+        preceding item must appear one or more times
```

And the pseudo BNF describing the syntax is:

```
input      ::= ( TEXT
                | variable
                | INDEX_OPEN input INDEX_CLOSE (loop_limits)?
                ) *
variable   ::= DELIM_INIT (name|expression)
name       ::= (NAME_CHARS)+
expression ::= DELIM_OPEN
                (name|variable)+
                (INDEX_OPEN num_exp INDEX_CLOSE)?
                (':' command)*
                DELIM_CLOSE
command    ::= '-' (TEXT_EXP|variable)+
                | '+' (TEXT_EXP|variable)+
                | 'o' NUMBER ('-'|'|') (NUMBER)?
                | '#'
                | '*' (TEXT_EXP|variable)+
                | 's' '/' (TEXT_PATTERN)+
                | '/' (variable|TEXT_SUBST)*
                | '/' ('m'|'g' | 'i' | 't')*
```

```

| 'y' '/' (variable|TEXT_SUBST)+
| '/' (variable|TEXT_SUBST)*
| '/'
| 'p' '/' NUMBER
| '/' (variable|TEXT_SUBST)*
| '/' ('r'|'l'|'c')
| '%' (name|variable)+
| '(' (' (TEXT_ARGS)? ')?
| 'l'
| 'u'
num_exp ::= operand
| operand ('+'| '-'| '*'| '/'| '%') num_exp
operand ::= ('+'| '-'| '*')? NUMBER
| INDEX_MARK
| '(' num_exp ')'
| variable
loop_limits ::= DELIM_OPEN
| (num_exp)? ' ' (num_exp)? (' ' (num_exp)?)?
| DELIM_CLOSE
NUMBER ::= ('0'|...|'9')+
TEXT_PATTERN ::= ('('|')')+
TEXT_SUBST ::= ('(DELIM_INIT|'/'|')')+
TEXT_ARGS ::= ('(DELIM_INIT|'')')+
TEXT_EXP ::= ('(DELIM_INIT|DELIM_CLOSE|' ':'|'+')+)+
TEXT ::= ('(DELIM_INIT|INDEX_OPEN|INDEX_CLOSE))+
DELIM_INIT ::= '$'
DELIM_OPEN ::= '{'
DELIM_CLOSE ::= '}'
INDEX_OPEN ::= '['
INDEX_CLOSE ::= ']'
INDEX_MARK ::= '#'
NAME_CHARS ::= 'a'|...|'z'|'A'|...|'Z'|'0'|...|'9'

```

2.4 Semantics

The items listed in **command** above, which always follow a colon (:) have the following meanings:

```

- perform substitution if variable is empty
+ perform substitution if variable is not empty
o cut out substring of the variable value
# length of the variable value
* substitute empty string if the variable value is not empty,
  otherwise substitute the trailing parameter
s regular expression search and replace. The trailing
  options are: m = multiline, i = case insensitive,
               g = global, t = plain text (no regexp)
y transpose characters from class A to class B
p pad variable to l = left, r = right or c = center,
  with second value.
% special function call (none implemented)
l lower case the variable value
u upper case the variable value

```

The **loop_limits** are start, step, and end values.

A counter variable name followed immediately by a plus (+) will cause the counter to be incremented by one.

2.5 Examples

To create an ISO date:

```
DLT-${Year}-${Month:p/2/0/r}-${Day:p/2/0/r}
```

on 20 June 2003 would give **DLT-2003-06-20**

If you set the environment variable **mon** to

```
January|February|March|April|May|...  
File-${mon[${Month}]}/${Day}/${Year}
```

on the first of March would give **File-March/1/2003**

Chapter 3

Using Stunnel to Encrypt Communications

Prior to version 1.37, Bacula did not have built-in communications encryption. Please see the TLS chapter if you are using Bacula 1.37 or greater.

Without too much effort, it is possible to encrypt the communications between any of the daemons. This chapter will show you how to use **stunnel** to encrypt communications to your client programs. We assume the Director and the Storage daemon are running on one machine that will be called **server** and the Client or File daemon is running on a different machine called **client**. Although the details may be slightly different, the same principles apply whether you are encrypting between Unix, Linux, or Win32 machines. This example was developed between two Linux machines running stunnel version 4.04-4 on a Red Hat Enterprise 3.0 system.

3.1 Communications Ports Used

First, you must know that with the standard Bacula configuration, the Director will contact the File daemon on port 9102. The File daemon then contacts the Storage daemon using the address and port parameters supplied by the Director. The standard port used will be 9103. This is the typical server/client view of the world, the File daemon is a server to the Director (i.e. listens for the Director to contact it), and the Storage daemon is a server to the File daemon.

3.2 Encryption

The encryption is accomplished between the Director and the File daemon by using an stunnel on the Director's machine (server) to encrypt the data and to contact an stunnel on the File daemon's machine (client), which decrypts the data and passes it to the client.

Between the File daemon and the Storage daemon, we use an stunnel on the File daemon's machine to encrypt the data and another stunnel on the Storage daemon's machine to decrypt the data.

As a consequence, there are actually four copies of stunnel running, two on the server and two on the client. This may sound a bit complicated, but it really isn't. To accomplish this, we will need to construct four separate conf files for stunnel, and we will need to make some minor modifications to the Director's conf file. None of the other conf files need to be changed.

3.3 A Picture

Since pictures usually help a lot, here is an overview of what we will be doing. Don't worry about all the details of the port numbers and such for the moment.

```
File daemon (client):
    stunnel-fd1.conf
    |=====|
Port 29102 >----| Stunnel 1 |-----> Port 9102
    |=====|
    stunnel-fd2.conf
    |=====|
Port 9103 >----| Stunnel 2 |-----> server:29103
    |=====|
Director (server):
    stunnel-dir.conf
    |=====|
Port 29102 >----| Stunnel 3 |-----> client:29102
    |=====|
    stunnel-sd.conf
    |=====|
Port 29103 >----| Stunnel 4 |-----> 9103
    |=====|
```

3.4 Certificates

In order for stunnel to function as a server, which it does in our diagram for Stunnel 1 and Stunnel 4, you must have a certificate and the key. It is possible to keep the two in separate files, but normally, you keep them in one single .pem file. You may create this certificate yourself in which case, it will be self-signed, or you may have it signed by a CA.

If you want your clients to verify that the server is in fact valid (Stunnel 2 and Stunnel 3), you will need to have the server certificates signed by a CA (Certificate Authority), and you will need to have the CA's public certificate (contains the CA's public key).

Having a CA signed certificate is **highly** recommended if you are using your client across the Internet, otherwise you are exposed to the man in the middle attack and hence loss of your data.

See below for how to create a self-signed certificate.

3.5 Securing the Data Channel

To simplify things a bit, let's for the moment consider only the data channel. That is the connection between the File daemon and the Storage daemon, which takes place on port 9103. In fact, in a minimalist solution, this is the only connection that needs to be encrypted, because it is the one that transports your data. The connection between the Director and the File daemon is simply a control channel used to start the job and get the job status.

Normally the File daemon will contact the Storage daemon on port 9103 (supplied by the Director), so we need an stunnel that listens on port 9103 on the File daemon's machine, encrypts the data and sends it to the Storage daemon. This is depicted by Stunnel 2 above. Note that this stunnel is listening on port 9103 and sending to server:29103. We use port 29103 on the server because if we would send the data to port 9103, it would go directly to the Storage daemon, which doesn't understand encrypted data. On the server machine, we run Stunnel 4, which listens on port 29103, decrypts the data and sends it to the Storage daemon, which is listening on port 9103.

3.6 Data Channel Configuration

The Storage resource of the bacula-dir.conf normally looks something like the following:

```
Storage {
    Name = File
    Address = server
    SDPort = 9103
    Password = storage_password
    Device = File
    Media Type = File
}
```

Notice that this is running on the server machine, and it points the File daemon back to server:9103, which is where our Storage daemon is listening. We modify this to be:

```
Storage {
    Name = File
    Address = localhost
    SDPort = 9103
    Password = storage_password
    Device = File
    Media Type = File
}
```

This causes the File daemon to send the data to the stunnel running on localhost (the client machine). We could have used client as the address as well.

3.7 Stunnel Configuration for the Data Channel

In the diagram above, we see above Stunnel 2 that we use stunnel-fd2.conf on the client. A pretty much minimal config file would look like the following:

```
client = yes
[29103]
accept = localhost:9103
connect = server:29103
```

The above config file does encrypt the data but it does not require a certificate, so it is subject to the man in the middle attack. The file I actually used, stunnel-fd2.conf, looked like this:

```
#
# Stunnel conf for Bacula client -> SD
#
pid = /home/kern/bacula/bin/working/stunnel.pid
#
# A cert is not mandatory here. If verify=2, a
# cert signed by a CA must be specified, and
# either CAfile or CPath must point to the CA's
# cert
#
cert = /home/kern/stunnel/stunnel.pem
CAfile = /home/kern/ssl/cacert.pem
verify = 2
client = yes
# debug = 7
# foreground = yes
[29103]
accept = localhost:9103
connect = server:29103
```

You will notice that I specified a pid file location because I ran stunnel under my own userid so I could not use the default, which requires root permission. I also specified a certificate that I have as well as verify level 2 so that the certificate is required and verified, and I must supply the location of the CA (Certificate Authority) certificate so that the stunnel certificate can be verified. Finally, you will see that there are two lines commented out, which when enabled, produce a lot of nice debug info in the command window.

If you do not have a signed certificate (stunnel.pem), you need to delete the cert, CAfile, and verify lines.

Note that the stunnel.pem, is actually a private key and a certificate in a single file. These two can be kept and specified individually, but keeping them in one file is more convenient.

The config file, stunnel-sd.conf, needed for Stunnel 4 on the server machine is:

```
#
# Bacula stunnel conf for Storage daemon
#
pid = /home/kern/bacula/bin/working/stunnel.pid
#
# A cert is mandatory here, it may be self signed
# If it is self signed, the client may not use
# verify
#
cert  = /home/kern/stunnel/stunnel.pem
client = no
# debug = 7
# foreground = yes
[29103]
accept = 29103
connect = 9103
```

3.8 Starting and Testing the Data Encryption

It will most likely be the simplest to implement the Data Channel encryption in the following order:

- Setup and run Bacula backing up some data on your client machine without encryption.
- Stop Bacula.
- Modify the Storage resource in the Director's conf file.
- Start Bacula
- Start stunnel on the server with:

```
stunnel stunnel-sd.conf
```

- Start stunnel on the client with:

```
stunnel stunnel-fd2.conf
```

- Run a job.
- If it doesn't work, turn debug on in both stunnel conf files, restart the stunnels, rerun the job, repeat until it works.

3.9 Encrypting the Control Channel

The Job control channel is between the Director and the File daemon, and as mentioned above, it is not really necessary to encrypt, but it is good practice to encrypt it as well. The two stunnels that are used in

this case will be Stunnel 1 and Stunnel 3 in the diagram above. Stunnel 3 on the server might normally listen on port 9102, but if you have a local File daemon, this will not work, so we make it listen on port 29102. It then sends the data to client:29102. Again we use port 29102 so that the stunnel on the client machine can decrypt the data before passing it on to port 9102 where the File daemon is listening.

3.10 Control Channel Configuration

We need to modify the standard Client resource, which would normally look something like:

```
Client {
    Name = client-fd
    Address = client
    FDPort = 9102
    Catalog = BackupDB
    Password = "xxx"
}
```

to be:

```
Client {
    Name = client-fd
    Address = localhost
    FDPort = 29102
    Catalog = BackupDB
    Password = "xxx"
}
```

This will cause the Director to send the control information to localhost:29102 instead of directly to the client.

3.11 Stunnel Configuration for the Control Channel

The stunnel config file, stunnel-dir.conf, for the Director's machine would look like the following:

```
#
# Bacula stunnel conf for the Directory to contact a client
#
pid = /home/kern/bacula/bin/working/stunnel.pid
#
# A cert is not mandatory here. If verify=2, a
# cert signed by a CA must be specified, and
# either CAfile or CApath must point to the CA's
# cert
#
cert = /home/kern/stunnel/stunnel.pem
CAfile = /home/kern/ssl/cacert.pem
verify = 2
client = yes
# debug = 7
# foreground = yes
[29102]
accept = localhost:29102
connect = client:29102
```

and the config file, stunnel-fd1.conf, needed to run stunnel on the Client would be:

```
#
# Bacula stunnel conf for the Directory to contact a client
#
```

```

pid = /home/kern/bacula/bin/working/stunnel.pid
#
# A cert is not mandatory here. If verify=2, a
# cert signed by a CA must be specified, and
# either CAfile or CPath must point to the CA's
# cert
#
cert = /home/kern/stunnel/stunnel.pem
CAfile = /home/kern/ssl/cacert.pem
verify = 2
client = yes
# debug = 7
# foreground = yes
[29102]
accept = localhost:29102
connect = client:29102

```

3.12 Starting and Testing the Control Channel

It will most likely be the simplest to implement the Control Channel encryption in the following order:

- Stop Bacula.
- Modify the Client resource in the Director's conf file.
- Start Bacula
- Start stunnel on the server with:

```
stunnel stunnel-dir.conf
```

- Start stunnel on the client with:

```
stunnel stunnel-fd1.conf
```

- Run a job.
- If it doesn't work, turn debug on in both stunnel conf files, restart the stunnels, rerun the job, repeat until it works.

3.13 Using stunnel to Encrypt to a Second Client

On the client machine, you can just duplicate the setup that you have on the first client file for file and it should work fine.

In the bacula-dir.conf file, you will want to create a second client pretty much identical to how you did for the first one, but the port number must be unique. We previously used:

```

Client {
  Name = client-fd
  Address = localhost
  FDPort = 29102
  Catalog = BackupDB
  Password = "xxx"
}

```

so for the second client, we will, of course, have a different name, and we will also need a different port. Remember that we used port 29103 for the Storage daemon, so for the second client, we can use port 29104, and the Client resource would look like:

```
Client {
    Name = client2-fd
    Address = localhost
    FDPort = 29104
    Catalog = BackupDB
    Password = "yyy"
}
```

Now, fortunately, we do not need a third stunnel to on the Director's machine, we can just add the new port to the config file, stunnel-dir.conf, to make:

```
#
# Bacula stunnel conf for the Directory to contact a client
#
pid = /home/kern/bacula/bin/working/stunnel.pid
#
# A cert is not mandatory here. If verify=2, a
# cert signed by a CA must be specified, and
# either CAfile or CPath must point to the CA's
# cert
#
cert = /home/kern/stunnel/stunnel.pem
CAfile = /home/kern/ssl/cacert.pem
verify = 2
client = yes
# debug = 7
# foreground = yes
[29102]
accept = localhost:29102
connect = client:29102
[29104]
accept = localhost:29102
connect = client2:29102
```

There are no changes necessary to the Storage daemon or the other stunnel so that this new client can talk to our Storage daemon.

3.14 Creating a Self-signed Certificate

You may create a self-signed certificate for use with stunnel that will permit you to make it function, but will not allow certificate validation. The .pem file containing both the certificate and the key can be made with the following, which I put in a file named **makepem**:

```
#!/bin/sh
#
# Simple shell script to make a .pem file that can be used
# with stunnel and Bacula
#
OPENSSL=openssl
umask 77
PEM1="/bin/mktemp openssl.XXXXXX"
PEM2="/bin/mktemp openssl.XXXXXX"
${OPENSSL} req -newkey rsa:1024 -keyout $PEM1 -nodes \
    -x509 -days 365 -out $PEM2
cat $PEM1 > stunnel.pem
echo "" >>stunnel.pem
cat $PEM2 >>stunnel.pem
rm $PEM1 $PEM2
```

The above script will ask you a number of questions. You may simply answer each of them by entering a return, or if you wish you may enter your own data.

3.15 Getting a CA Signed Certificate

The process of getting a certificate that is signed by a CA is quite a bit more complicated. You can purchase one from quite a number of PKI vendors, but that is not at all necessary for use with Bacula.

To get a CA signed certificate, you will either need to find a friend that has setup his own CA or to become a CA yourself, and thus you can sign all your own certificates. The book OpenSSL by John Viega, Matt Mesier & Pravir Chandra from O'Reilly explains how to do it, or you can read the documentation provided in the Open-source PKI Book project at Source Forge: <http://ospkibook.sourceforge.net/docs/OSPki-2.4.7/OSPki-html/ospki-book.htm>. Note, this link may change.

3.16 Using ssh to Secure the Communications

Please see the script **ssh-tunnel.sh** in the **examples** directory. It was contributed by Stephan Holl.

Chapter 4

DVD Volumes

Bacula allows you to specify that you want to write to DVD. However, this feature is implemented only in version 1.37 or later. You may in fact write to DVD+RW, DVD+R, DVD-R, or DVD-RW media. The actual process used by Bacula is to first write the image to a spool directory, then when the Volume reaches a certain size or, at your option, at the end of a Job, Bacula will transfer the image from the spool directory to the DVD. The actual work of transferring the image is done by a script **dvd-handler**, and the heart of that script is a program called **growisofs** which allows creating or adding to a DVD ISO filesystem.

You must have **dvd+rw-tools** loaded on your system for DVD writing to work. Please note that the original **dvd+rw-tools** package does **NOT** work with Bacula. You must apply a patch which can be found in the **patches** directory of Bacula sources with the name **dvd+rw-tools-5.21.4.10.8.bacula.patch** for version 5.21 of the tools, or patch bf dvd+rw-tools-6.1.bacula.patch if you have version 6.1 on your system. Unfortunately, this requires you to build the dvd+rw-tools from source.

Note, some Linux distros such as Debian dvd+rw-tools-7.0-4 package already have the patch applied, so please check.

The fact that Bacula cannot use the OS to write directly to the DVD makes the whole process a bit more error prone than writing to a disk or a tape, but nevertheless, it does work if you use some care to set it up properly. However, at the current time (version 1.39.30 – 12 December 2006) we still consider this code to be BETA quality. As a consequence, please do careful testing before relying on DVD backups in production.

The remainder of this chapter explains the various directives that you can use to control the DVD writing.

4.1 DVD Specific SD Directives

The following directives are added to the Storage daemon's Device resource.

Requires Mount = *Yes—No* You must set this directive to **yes** for DVD-writers, and to **no** for all other devices (tapes/files). This directive indicates if the device requires to be mounted using the **Mount Command**. To be able to write a DVD, the following directives must also be defined: **Mount Point**, **Mount Command**, **Unmount Command** and **Write Part Command**.

Mount Point = *directory* Directory where the device can be mounted.

Mount Command = *name-string* Command that must be executed to mount the device. Although the device is written directly, the mount command is necessary in order to determine the free space left on the DVD. Before the command is executed, %a is replaced with the Archive Device, and %m with the Mount Point.

Most frequently, you will define it as follows:

```
Mount Command = "/bin/mount -t iso9660 -o ro %a %m"
```

However, if you have defined a mount point in `/etc/fstab`, you might be able to use a mount command such as:

```
Mount Command = "/bin/mount /media/dvd"
```

Unmount Command = *name-string* Command that must be executed to unmount the device. Before the command is executed, `%a` is replaced with the Archive Device, and `%m` with the Mount Point.

Most frequently, you will define it as follows:

```
Unmount Command = "/bin/umount %m"
```

Write Part Command = *name-string* Command that must be executed to write a part to the device. Before the command is executed, `%a` is replaced with the Archive Device, `%m` with the Mount Point, `%e` is replaced with 1 if we are writing the first part, and with 0 otherwise, and `%v` with the current part filename.

For a DVD, you will most frequently specify the Bacula supplied **dvd-handler** script as follows:

```
Write Part Command = "/path/dvd-handler %a write %e %v"
```

Where `/path` is the path to your scripts install directory, and `dvd-handler` is the Bacula supplied script file. This command will already be present, but commented out, in the default `bacula-sd.conf` file. To use it, simply remove the comment (`#`) symbol.

Free Space Command = *name-string* Command that must be executed to check how much free space is left on the device. Before the command is executed, `%a` is replaced with the Archive Device.

For a DVD, you will most frequently specify the Bacula supplied **dvd-handler** script as follows:

```
Free Space Command = "/path/dvd-handler %a free"
```

Where `/path` is the path to your scripts install directory, and `dvd-handler` is the Bacula supplied script file. If you want to specify your own command, please look at the code in `dvd-handler` to see what output Bacula expects from this command. This command will already be present, but commented out, in the default `bacula-sd.conf` file. To use it, simply remove the comment (`#`) symbol.

If you do not set it, Bacula will expect there is always free space on the device.

In addition to the directives specified above, you must also specify the other standard Device resource directives. Please see the sample DVD Device resource in the default `bacula-sd.conf` file. Be sure to specify the raw device name for **Archive Device**. It should be a name such as `/dev/cdrom` or `/media/cdrecorder` or `/dev/dvd` depending on your system. It will not be a name such as `/mnt/cdrom`.

Finally, for **growisofs** to work, it must be able to lock a certain amount of memory in RAM. If you have restrictions on this function, you may have failures. Under **bash**, you can set this with the following command:

```
ulimit -l unlimited
```

4.2 Edit Codes for DVD Directives

Before submitting the **Mount Command**, **Unmount Command**, **Write Part Command**, or **Free Space Command** directives to the operating system, Bacula performs character substitution of the following characters:

```
%% = %
%a = Archive device name
%e = erase (set if cannot mount and first part)
%n = part number
%m = mount point
%v = last part name (i.e. filename)
```

4.3 DVD Specific Director Directives

The following directives are added to the Director's Job resource.

Write Part After Job = <yes—no> If this directive is set to **yes** (default **no**), the Volume written to a temporary spool file for the current Job will be written to the DVD as a new part file will be created after the job is finished.

It should be set to **yes** when writing to devices that require a mount (for example DVD), so you are sure that the current part, containing this job's data, is written to the device, and that no data is left in the temporary file on the hard disk. However, on some media, like DVD+R and DVD-R, a lot of space (about 10Mb) is lost everytime a part is written. So, if you run several jobs each after another, you could set this directive to **no** for all jobs, except the last one, to avoid wasting too much space, but to ensure that the data is written to the medium when all jobs are finished.

This directive is ignored for devices other than DVDs.

4.4 Other Points

- Please be sure that you have any automatic DVD mounting disabled before running Bacula – this includes auto mounting in `/etc/fstab`, `hotplug`, ... If the DVD is automatically mounted by the OS, it will cause problems when Bacula tries to mount/unmount the DVD.
- Please be sure that you the directive **Write Part After Job** set to **yes**, otherwise the last part of the data to be written will be left in the DVD spool file and not written to the DVD. The DVD will then be unreadable until this last part is written. If you have a series of jobs that are run one at a time, you can turn this off until the last job is run.
- The current code is not designed to have multiple simultaneous jobs writing to the DVD. As a consequence, please ensure that only one DVD backup job runs at any time.
- Writing and reading of DVD+RW seems to work quite reliably provided you are using the patched `dvd+rw-mediainfo` programs. On the other hand, we do not have enough information to ensure that DVD-RW or other forms of DVDs work correctly.
- DVD+RW supports only about 1000 overwrites. Every time you mount the filesystem read/write will count as one write. This can add up quickly, so it is best to mount your DVD+RW filesystem read-only. Bacula does not need the DVD to be mounted read-write, since it uses the raw device for writing.
- Reformatting DVD+RW 10-20 times can apparently make the medium unusable. Normally you should not have to format or reformat DVD+RW media. If it is necessary, current versions of `growisofs` will do so automatically.
- We have had several problems writing to DVD-RWs (this does NOT concern DVD+RW), because these media have two writing-modes: **Incremental Sequential** and **Restricted Overwrite**. Depending on your device and the media you use, one of these modes may not work correctly (e.g. **Incremental Sequential** does not work with my NEC DVD-writer and Verbatim DVD-RW).

To retrieve the current mode of a DVD-RW, run:

```
dvd+rw-mediainfo /dev/xxx
```

where you replace `xxx` with your DVD device name.

Mounted Media line should give you the information.

To set the device to **Restricted Overwrite** mode, run:

```
dvd+rw-format /dev/xxx
```

If you want to set it back to the default **Incremental Sequential** mode, run:

```
dvd+rw-format -blank /dev/xxx
```

- Bacula only accepts to write to blank DVDs. To quickly blank a DVD+/-RW, run this command:

```
dd if=/dev/zero bs=1024 count=512 | growisofs -Z /dev/xxx=/dev/fd/0
```

Then, try to mount the device, if it cannot be mounted, it will be considered as blank by Bacula, if it can be mounted, try a full blank (see below).

- If you wish to blank completely a DVD+/-RW, use the following:

```
growisofs -Z /dev/xxx=/dev/zero
```

where you replace xxx with your DVD device name. However, note that this blanks the whole DVD, which takes quite a long time (16 minutes on mine).

- DVD+RW and DVD-RW support only about 1000 overwrites (i.e. don't use the same medium for years if you don't want to have problems...).

To write to the DVD the first time use:

```
growisofs -Z /dev/xxx filename
```

To add additional files (more parts use):

```
growisofs -M /dev/xxx filename
```

The option **-use-the-force-luke=4gms** was added in growisofs 5.20 to override growisofs' behavior of always checking for the 4GB limit. Normally, this option is recommended for all Linux 2.6.8 kernels or greater, since these newer kernels can handle writing more than 4GB. See below for more details on this subject.

- For more information about DVD writing, please look at the dvd+rw-tools homepage.
- According to bug #912, bscan cannot read multi-volume DVDs. This is on our TODO list, but unless someone submits a patch it is not likely to be done any time in the near future. (9 Sept 2007).

Chapter 5

Bacula Projects

Once a new major version of Bacula is released, the Bacula users will vote on a list of new features. This vote is used as the main element determining what new features will be implemented for the next version. Generally, the development time for a new release is between four to nine months. Sometimes it may be a bit longer, but in that case, there will be a number of bug fix updates to the currently released version.

For the current list of project, please see the projects page in the CVS at: http://cvs.sourceforge.net/viewcvs.py/*checkout*/bacula/bacula/projects see the **projects** file in the main source directory. The projects file is updated approximately once every six months.

Separately from the project list, Kern maintains a current list of tasks as well as ideas, feature requests, and occasionally design notes. This list is updated roughly weekly (sometimes more often). For a current list of tasks you can see **kernstodo** in the Source Forge CVS at http://cvs.sourceforge.net/viewcvs.py/*checkout*/bacula/bacula/kernstodo.

Chapter 6

The internal database is not supported, please do not use it.

6.1 Internal Bacula Database

Previously it was intended to be used primarily by Bacula developers for testing; although SQLite is also a good choice for this. We do not recommend its use in general.

This database is simplistic in that it consists entirely of Bacula's internal structures appended sequentially to a file. Consequently, it is in most cases inappropriate for sites with many clients or systems with large numbers of files, or long-term production environments.

Below, you will find a table comparing the features available with SQLite and MySQL and with the internal Bacula database. At the current time, you cannot dynamically switch from one to the other, but must rebuild the Bacula source code. If you wish to experiment with both, it is possible to build both versions of Bacula and install them into separate directories.

Feature	SQLite or MySQL	Bacula
Job Record	Yes	Yes
Media Record	Yes	Yes
FileName Record	Yes	No
File Record	Yes	No
FileSet Record	Yes	Yes
Pool Record	Yes	Yes
Client Record	Yes	Yes
JobMedia Record	Yes	Yes
List Job Records	Yes	Yes
List Media Records	Yes	Yes
List Pool Records	Yes	Yes
List JobMedia Records	Yes	Yes
Delete Pool Record	Yes	Yes
Delete Media Record	Yes	Yes
Update Pool Record	Yes	Yes
Implement Verify	Yes	No
MD5 Signatures	Yes	No

In addition, since there is no SQL available, the Console commands: **sqlquery**, **query**, **retention**, and any other command that directly uses SQL are not available with the Internal database.

Chapter 7

Bacula Copyright, Trademark, and Licenses

There are a number of different licenses that are used in Bacula. If you have a printed copy of this manual, the details of each of the licenses referred to in this chapter can be found in the online version of the manual at <http://www.bacula.org>.

7.1 FDL

The GNU Free Documentation License (FDL) is used for this manual, which is a free and open license. This means that you may freely reproduce it and even make changes to it. However, rather than distribute your own version of this manual, we would much prefer if you would send any corrections or changes to the Bacula project.

The most recent version of the manual can always be found online at <http://www.bacula.org>.

7.2 GPL

The vast bulk of the source code is released under the GNU General Public License version 2..

Most of this code is copyrighted: Copyright ©2000-2009 Free Software Foundation Europe e.V.

Portions may be copyrighted by other people. These files are released under different licenses which are compatible with the Bacula GPLv2 license.

7.3 LGPL

Some of the Bacula library source code is released under the GNU Lesser General Public License. This permits third parties to use these parts of our code in their proprietary programs to interface to Bacula.

7.4 Public Domain

Some of the Bacula code, or code that Bacula references, has been released to the public domain. E.g. md5.c, SQLite.

7.5 Trademark

Bacula[®] is a registered trademark of Kern Sibbald.

We have trademarked the Bacula name to ensure that any program using the name Bacula will be exactly compatible with the program that we have released. The use of the name Bacula is restricted to software systems that agree exactly with the program presented here. If you have made modifications to the Bacula source code that alter in any significant way the way the program functions, you may not distribute it using the Bacula name.

7.6 Fiduciary License Agreement

Developers who have contributed significant changes to the Bacula code should have signed a Fiduciary License Agreement (FLA), which guarantees them the right to use the code they have developed, and also ensures that the Free Software Foundation Europe (and thus the Bacula project) has the rights to the code. This Fiduciary License Agreement is found on the Bacula web site at:

<http://www.bacula.org/en/FLA-bacula.en.pdf>

and if you are submitting code, you should fill it out then sent to:

Kern Sibbald
Cotes-de-Montmoiret 9
1012 Lausanne
Switzerland

When you send in such a complete document, please notify me: kern at sibbald dot com.

7.7 Disclaimer

NO WARRANTY

BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, 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.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE 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.

Chapter 8

GNU Free Documentation License

Version 1.2, November 2002

Copyright ©2000,2001,2002 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

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. Secondly, 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.

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 for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

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 <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.

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 1.2 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.

GNU General Public License

image of a Philosophical GNU

- What to do if you see a possible GPL violation
- Translations of the GPL

8.1 Table of Contents

- GNU GENERAL PUBLIC LICENSE
 - Preamble
 - TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION
 - How to Apply These Terms to Your New Programs

8.2 GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

8.3 Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) 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 this service 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 make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. 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.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its

recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

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

8.4 TERMS AND CONDITIONS

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute 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 and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- **a)** You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- **b)** You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- **c)** If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- **a)** Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- **b)** Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- **c)** Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), 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 distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the 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 a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, 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.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE 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.

END OF TERMS AND CONDITIONS

8.5 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 convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
{\em one line to give the program's name and an idea of what it does.}
Copyright (C) {\em yyyy} {\em 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 2
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, write to the Free Software
Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA
02110-1301 USA
```

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

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

```
Gnomovision version 69, Copyright (C) {\em year} {\em name of author}
Gnomovision 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, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright
interest in the program 'Gnomovision'
(which makes passes at compilers) written
by James Hacker.
{\em signature of Ty Coon}, 1 April 1989
Ty Coon, President of Vice
```

This 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 Library General Public License instead of this License. Return to GNU's home page.

FSF & GNU inquiries & questions to gnu@gnu.org. Other ways to contact the FSF.

Comments on these web pages to webmasters@www.gnu.org, send other questions to gnu@gnu.org.

Copyright notice above. Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Updated: 3 Jan 2000 rms

GNU Lesser General Public License

image of a Philosophical GNU [English — Japanese]

- Why you shouldn't use the Lesser GPL for your next library
- What to do if you see a possible LGPL violation
- Translations of the LGPL
- The GNU Lesser General Public License as a text file
- The GNU Lesser General Public License as a Texinfo file

This GNU Lesser General Public License counts as the successor of the GNU Library General Public License. For an explanation of why this change was necessary, read the Why you shouldn't use the Lesser GPL for your next library article.

8.6 Table of Contents

- GNU LESSER GENERAL PUBLIC LICENSE
 - Preamble
 - TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION
 - How to Apply These Terms to Your New Libraries

8.7 GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
[This is the first released version of the Lesser GPL. It also counts
as the successor of the GNU Library Public License, version 2, hence
the version number 2.1.]

8.8 Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, 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 this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

8.9 TERMS AND CONDITIONS

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- **a)** The modified work must itself be a software library.
- **b)** You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- **c)** You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- **d)** If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- **a)** Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library

and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

- **b)** Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.
- **c)** Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
- **d)** If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- **e)** Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

- **a)** Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
- **b)** Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), 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 distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser 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 Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "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 LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY 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 LIBRARY (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 LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

8.10 How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
{\it one line to give the library's name and an idea of what it does.}
Copyright (C) {\it year} {\it name of author}
This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.
This library 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
Lesser General Public License for more details.
You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301
USA
```

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

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in
the library "Frob" (a library for tweaking knobs) written
by James Random Hacker.
{\it signature of Ty Coon}, 1 April 1990
Ty Coon, President of Vice
```

That's all there is to it! Return to GNU's home page.

FSF & GNU inquiries & questions to gnu@gnu.org. Other ways to contact the FSF.

Comments on these web pages to webmasters@www.gnu.org, send other questions to gnu@gnu.org.

Copyright notice above. Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301
USA USA

Updated: 27 Nov 2000 paulv

General Index

- Bacula Copyright, Trademark, and Licenses, 29
- Bacula Events, 2
- Bacula Projects , 25
- Bacula Variables , 9
- bacula-dir.conf
 - Modification for the Data Channel , 15
- Certificate
 - Creating a Self-signed , 19
 - Getting a CA Signed , 20
- Certificates , 14
- Channel
 - Encrypting the Control , 16
 - Securing the Data , 14
 - Starting and Testing the Control , 18
- Client
 - Using stunnel to Encrypt to a Second , 18
- Communications
 - Using ssh to Secure the , 20
- Communications Ports Used , 13
- CONDITIONS
 - TERMS AND , 38, 43
- Config Files for stunnel to Encrypt the Control Channel , 17
- Configuration
 - Python, 1
- Console Command
 - Python, 5
- Contents
 - Table of , 37, 42
- Control Channel Configuration , 17
- Counters , 9
- Creating a Self-signed Certificate , 19
- Database
 - Internal Bacula , 27
- Debugging Python Scripts, 5
- Directives
 - DVD, 21, 23
 - DVD Edit Codes, 22
- Disclaimer , 30
- Domain
 - Public , 29
- DVD Specific Director Directives , 23
- DVD Specific SD Directives , 21
- DVD Volumes, 21
- DVD Writing, 21
- Edit Codes for DVD Directives , 22
- Encrypting the Control Channel , 16
- Encryption
 - Starting and Testing the Data , 16
- Encryption , 13
- Environment Variables , 10
- Events, 2
- Example
 - Python, 6
- Examples , 11
- Expansion
 - Variable , 9
- FDL , 29
- Fiduciary License Agreement , 30
- Free Space Command , 22
- Full Syntax , 10
- Functionality
 - General , 9
- General Functionality , 9
- Getting a CA Signed Certificate , 20
- GNU Free Documentation License, 31
- GNU GENERAL PUBLIC LICENSE , 37
- GNU General Public License , 37
- GNU LESSER GENERAL PUBLIC LICENSE , 42
- GNU Lesser General Public License , 42
- GPL , 29
- How to Apply These Terms to Your New Libraries , 48
- How to Apply These Terms to Your New Programs , 41
- Internal Bacula Database , 27
- Internal Variables , 9
- JobStart, 2
- LGPL , 29
- Libraries
 - How to Apply These Terms to Your New , 48
- LICENSE
 - GNU GENERAL PUBLIC , 37
 - GNU LESSER GENERAL PUBLIC , 42
- License
 - GNU Free Documentation, 31
 - GNU General Public , 37
 - GNU Lesser General Public , 42
- Licenses
 - Bacula Copyright Trademark, 29
- Modification of bacula-dir.conf for the Data Channel , 15

- Mount Command, 21
- Mount Point, 21
- Objects
 - Python, 2
- Other Points , 23
- Picture , 14
- Points
 - Other , 23
- Preamble , 37, 42
- Programs
 - How to Apply These Terms to Your New , 41
- Projects
 - Bacula , 25
- Public Domain , 29
- Python Configuration, 1
- Python Console Command, 5
- Python Example, 6
- Python Objects, 2
- Python Scripting, 1
- Requires Mount , 21
- Scripting
 - Python, 1
- Securing the Data Channel , 14
- Semantics , 11
- Starting and Testing the Control Channel , 18
- Starting and Testing the Data Encryption , 16
- Stunnel Configuration for the Data Channel , 15
- Syntax
 - Full , 10
- Table of Contents , 37, 42
- TERMS AND CONDITIONS , 38, 43
- The internal database is not supported, please do not use it. , 27
- Trademark , 30
- Unmount Command, 22
- Use it
 - The internal database is not supported please do not , 27
- Used
 - Communications Ports , 13
- Using ssh to Secure the Communications , 20
- Using Stunnel to Encrypt Communications to Clients , 13
- Using stunnel to Encrypt to a Second Client , 18
- Variable Expansion , 9
- Variables
 - Bacula , 9
- Volumes
 - DVD, 21
- Write Part After Job , 23
- Write Part Command , 22
- Writing DVDs, 21