**Reconfiguring Packages**  
Debian packages often provide more-extensive initial setup options than do their RPM  
counterparts. Frequently, the install script included in the package asks a handful of  
questions, such as querying for the name of an outgoing mail relay system for a mail server  
program. These questions help the package system set up a standardized configuration that  
has nonetheless been customized for your computer.  
In the course of your system administration, you may alter the configuration files for a  
package. If you do this and find that you’ve made a mess of things, you may want to revert  
to the initial standard configuration. To do so, you can use the dpkg-reconfigure program,  
which runs the initial configuration script for the package you specify:  
# **dpkg-reconfigure samba**  
This command reconfigures the samba package, asking the package’s initial installation  
questions and restarting the Samba daemons. Once this is done, the package should be in  
something closer to its initial state.

**Debian Packages Compared to Other Package Formats**  
The overall functionality of Debian packages is similar to that of RPMs, although there are  
differences. Debian source packages aren’t single files; they’re groups of files—the original  
source tarball, a patch file that’s used to modify the source code (including a file that controls the building of a Debian package), and a .dsc file that contains a digital “signature” to help verify the authenticity of the collection. The Debian package tools can combine these and compile the package to create a Debian binary package. This structure makes Debian source packages slightly less convenient to transport because you must move at least  
two files (the tarball and patch file; the .dsc file is optional) rather than just one.

Debian source packages also support just one patch file, whereas RPM source packages may contain multiple patch files. Although you can certainly combine multiple patch files into one,  
doing so makes it less clear where a patch comes from, thus making it harder to back out of  
any given change.  
These source package differences are mostly of interest to software developers. As a  
system administrator or end user, you need not normally be concerned with them unless  
you must recompile a package from a source form—and even then, the differences between  
the formats need not be overwhelming. The exact commands and features used by each  
system differ, but they accomplish similar overall goals.  
Because all distributions that use Debian packages are derived from Debian, they tend  
to be more compatible with one another (in terms of their packages) than are RPM-based  
distributions. In particular, Debian has defined details of its system startup scripts and  
many other features to help Debian packages install and run on any Debian-based system.  
This helps Debian-based systems avoid the sorts of incompatibilities in startup scripts that  
can cause problems using one distribution’s RPMs on another distribution. Of course, some  
future distribution could violate Debian’s guidelines for these matters, so this advantage  
isn’t guaranteed to hold over time.  
As a practical matter, it can be harder to locate Debian packages than RPM packages for  
some exotic programs. Debian maintains a good collection at www.debian.org/distrib/  
packages, and some program authors make Debian packages available as well. If you can  
find an RPM but not a Debian package, you may be able to convert the RPM to Debian  
format using a program called alien, as described shortly in “Converting between Package  
Formats.” If all else fails, you can use a tarball, but you’ll lose the advantages of the Debian  
package database.