ldconfig – configure dynamic linker run-time bindings

Program loading needs to be fast, so use the ldconfig command to process the ld.so.conf file and all the included files from ld.so.conf.d and libraries from the trusted directories, /lib and /usr/lib, and any others supplied on the command line. The ldconfig command creates the necessary links and cache to recently used shared libraries in /etc/ld.so.cache. The dynamic loader uses the cached information from ld.so.cache to locate files that are to be dynamically loaded and linked. If you change ld.so.conf (or add new included files to ld.so.conf.d), you must run the ldconfig command (as root) to rebuild your ld.so.cache file.

Dynamic linker ld.so uses ld.so.cache file to determine the location of the libs. But you can also use LD\_LIBRARY\_PATH instead of ld.so.cache to provide the libs ‘s location to dynamic linker.

/etc/ld.so.conf is only read by ldconfig (the cache generation program), not by ld.so (the dynamic loader). You can change the location of the configuration file if you want to generate a cache that's different from the system default.

/etc/ld.so.cache is read by ld.so

Program loading needs to be fast, so use the ldconfig command to process the ld.so.conf file and all the included files from ld.so.conf.d and libraries from the trusted directories, /lib and /usr/lib, and any others supplied on the command line. The ldconfig command creates the necessary links and cache to recently used shared libraries in /etc/ld.so.cache. The dynamic loader uses the cached information from ld.so.cache to locate files that are to be dynamically loaded and linked. If you change ld.so.conf (or add new included files to ld.so.conf.d), you must run the ldconfig command (as root) to rebuild your ld.so.cache file.

LDCONFIG   
( Where are the Libs ? )  
  
  
Sometimes when you install a program from source it can complain that a certain library is missing . . . and still you know that the lib it is complaining about is actually installed on your system. But most likely it is not on the default place ( /usr/lib ) where the program looks for the lib.  
  
There is a file on your system where all the paths to the libraries are mentioned: the **/etc/ld.so.conf** file. Here is an example of the **/etc/ld.so.conf** file on Slackware:

|  |
| --- |
| **QUOTE** |
| /usr/local/lib /usr/X11R6/lib /usr/i486-slackware-linux/lib /usr/lib /opt/kde/lib |

So, what's the solution ?   
  
**1).** First locate the lib the program is complaining about, maybe it is in /usr/lib/qt/lib or in /usr/include or any other odd location.  
  
**2).** Next add the path to that lib in the **/etc/ld.so.conf** file. So, for our example the **/etc/ld.so.conf** file would look like:

|  |
| --- |
| **QUOTE** |
| /usr/local/lib /usr/X11R6/lib /usr/i486-slackware-linux/lib /usr/lib /opt/kde/lib /usr/lib/qt/lib  /usr/include |

**3).** Finally to let the system know that you updated the **/etc/ld.so.conf** file and make it use the new values give the command:

|  |
| --- |
| **CODE** |
| # ldconfig |

Now you can run the program that was complaining when we started this Tip and you will see that this time it will find the library . . . . have fun !