[Derived from http://docs.sun.com/app/docs/doc/816-0559?q=+Solaris+Linker+and+Libraries+Guide&s=t .]

When should I set LD_LIBRARY_PATH?

The short answer is **never**.

Why?

Some users seem to set this environment variable because of bad advice from other users or badly linked code that they do not know how to fix.

Libraries in UNIX

Linking is the process of joining one or more object code files together with one or more of the system libraries. The C library `libc` is by default linked in. Other libraries may be specified with `-l`*lib* options to the compiler (e.g. `-lm` for the Math library).

Static and dynamic libraries

Libraries are linked either *statically* or *dynamically* (also called *shared* libraries). Libraries live in `/usr/lib` amongst other places. Static libraries (`.a` suffix) are incorporated into the binary at link time, whereas dynamic ones (`.so` suffix) are referenced by location. There is a move in Solaris and less so in UNIX in general to using dynamic libraries wherever possible (Solaris ships with less static libraries at each release). Once a library has been linked with statically, if the same version of the library is updated (eg. minor upgrade, bug fix) the binary would need to be manually relinked to use the new library. Other tricks with dynamic libraries are possible, such as using a per-architecture copy of the library optimised for the processor/architecture on the current host.

Specifying the location of dynamic libraries not in `/usr/lib`

In UNIX the location of a library can be specified with the `-L` *dir* option to the compiler. Furthermore, in Solaris you need to specify the run time location with a corresponding `-R` *dir* option. For example, suppose we need to use the freely available `libz` in our C program, `squash`, built from `squash.c`. This is not part of Solaris so it is installed in `/usr/local/lib`. Here's a typical command to build it with Sun's C compiler - `cc`(1):

```
$ cc -o squash -L/usr/local/lib -R/usr/local/lib -lz squash.c
```

We could check what dynamic libraries are used with `ldd`(1) (list dynamic dependencies):

```
$ ldd squash
        libz.so =>      /usr/local/lib/libz.so
        libc.so.1 =>    /usr/lib/libc.so.1
```

```
        libdl.so.1 =>    /usr/lib/libdl.so.1
        /usr/platform/SUNW,Ultra-5_10/lib/libc_psr.so.1
```

Note, the other libraries are incorporated by default, and are not of interest to us. The last one would be different on a different type of Sun host. If we missed out the `-R` option, the binary would still build, however we could set `LD_LIBRARY_PATH` for it to find `libz`. `ldd` would give something like:

```
$ ldd squash
        libz.so =>       (file not found)
        libc.so.1 =>     /usr/lib/libc.so.1
        libdl.so.1 =>    /usr/lib/libdl.so.1
        /usr/platform/SUNW,Ultra-5_10/lib/libc_psr.so.1
```

However, this is a mistake and you should not set `LD_LIBRARY_PATH`; reasons why not are given below. As an alternative to using the `-L` and `-R` options, you can set the environment variable `LD_RUN_PATH` before compiling the code.

Using C++ compilers

To use Gnu's `g++` compiler to compile `foo.cc` which uses the Standard C++ library, you must do:

```
$ gcc -o foo -L/gnu/lib -R/gnu/lib -lstdc++ foo.cc
```

This is because it is not in a standard location as far as Solaris is concerned. This may not be needed in the future because of new functionality in Solaris 8. To build `foo` using Sun's C++ compiler, you can do:

```
$ CC -o foo foo.cc
```

This is because Solaris has its own C++ library which is in a standard location.

Changes in Solaris 8

In Solaris 8, functionality is added for the administrator to easily add default link locations to a host's environment with the `crle`(1) command. This will lead to less confusion with linking.

Why setting LD_LIBRARY_PATH is considered harmful

For the following reasons:

- `LD_LIBRARY_PATH` is used in preference to any run time or default system linker path. If (God forbid) you had it set to something like `/dcs/spod/baduser/lib`, if there was a hacked version of `libc` in that directory (for example) your account could be compromised. It is for this reason that set-uid programs completely ignore `LD_LIBRARY_PATH`.
- When code is compiled and depends on this to work, it can cause confusion where different versions of a library are installed in different directories, for example there is a `libtiff` in `/usr/openwin/lib` and `/usr/local/lib`. In this case, the former library is an older one used by some programs that come with Solaris.
- Sometimes when using precompiled binaries they may have been built with 3rd party libraries in specific locations; ideally code should either ship with the libraries and install into a certain location or link the code

as a pre-installation step. Solaris 7 introduces `$ORIGIN` which allows for a *relative* library location to be specified at run time (see the **Solaris Linker and Libraries Guide**). The alternative is to set `LD_LIBRARY_PATH` on a per-program basis, either as a wrapper program to the real program or a shell alias. Note however, that `LD_LIBRARY_PATH` may be inherited by programs called by the wrapped one ...

Further information

The **Solaris Linker and Libraries Guide**

Some library locations and their contents

First some Solaris library paths; 64-bit libraries typically exist in a `sparcv9` subdirectory:

Library path	Contents	Notes
`/usr/lib`	Standard system libraries	Default search path
`/usr/openwin/lib`	X11 libraries	none
`/usr/dt/lib`	CDE libraries and Motif (Xm)	none
`/usr/4lib`	SunOS 4 binary compatibility	DO NOT USE
`/usr/ucblib`	BSD compatibility	DO NOT USE

Some local library path conventions:

Library path	Contents	Notes
`/usr/local/lib`	3rd party libraries	none
`/gnu/lib`	Gnu libraries	Gnu's C++ compiler uses the libstdc++ here
`/usr/lang/lib`	Sun compiler extras	also exists as /opt/SUNWspro/lib
`/usr/local/kde/lib`	KDE libraries	standard ones exist elsewhere
`/gnu/gnome/lib`	GNOME libraries	standard ones exist elsewhere

[Local mirror site's addendum: Usenet poster Dan Espen recommends, instead: "LDFLAGS='-L/my/strange/path/lib -Wl,-rpath /my/strange/path/lib'"