



Symas OpenLDAP

How-To Guides

Using Symas OpenLDAP Devel Build to Create Source Files

Overlays, backends, and modules intended for use with Symas OpenLDAP must be compiled and linked using the `symas-openldap-devel` package. That package contains all relevant include files and libraries and has been used successfully by many of our customers to do exactly what you are attempting to do. By using the `-devel` package, you ensure that exactly the same headers and libraries that the Symas OpenLDAP package was built with are used for your custom-built modules.

To do this, you will need to take the following steps on your development system:

1. As a safety measure, make sure all other OpenLDAP or LDAP development packages or self-compiled LDAP packages have been removed from your development system.
2. Install the package 'symas-openldap-gold'
3. Install the package 'symas-openldap-gold-devel'
4. Make sure you have a modern gcc compiler installed
5. When compiling your overlay file(s), add the following flags to the gcc command line:

`-I /opt/symas/include`

6. When linking your overlay, add the following flags to the command line:

`-L <libdir> -l ldap -l ssl`

Where `<libdir>` is replaced with the directory where the library files reside. This is usually `/opt/symas/lib` or `/opt/symas/lib64`, but may be something else, depending on the OS/architecture you are using.

Please also keep in mind the following:

- Some versions of gcc allow a space between `-I` and its arguments and others do not. The same goes for the `-I` and `-L` flags.
- The order in which the parameters appear is important. Make sure that your `-I`, `-L`, and `-l` flags and their arguments appear before any other `-I`, `-L`, or `-l` flags. This will cause Symas OpenLDAP's libraries to be used in preference to those that might have shipped with your OS.
- Make sure your programs and modules contain a runpath by specifying the appropriate flag (`-r`, `-R`, `--runpath`, etc) and library directory on the link line so that they load the Symas OpenLDAP libraries from the correct locations. We do not recommend depending on the `LD_LIBRARY_PATH` environment variable to achieve this.

If you follow the steps above, you will end up with modules and executables that match slapd internal interfaces and the libraries that ship with Symas OpenLDAP.

