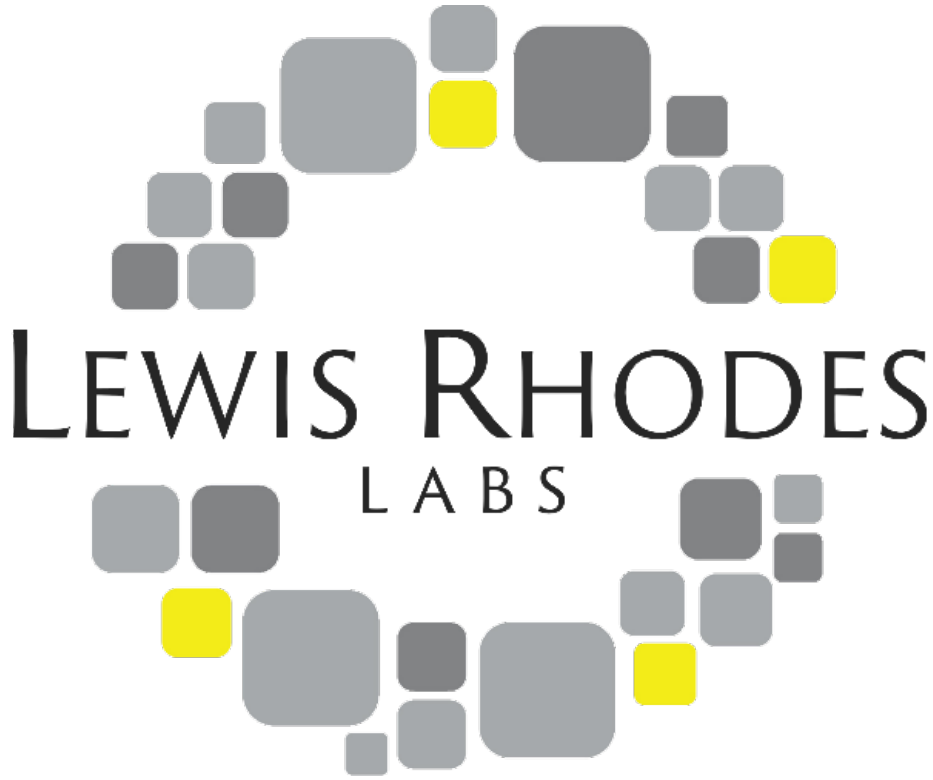

Extreme Search® Manual

Version 2024.04

https://lewis-rhodes.com

support@lewis-rhodes.com

---

4 Common tasks

4.1 Installation

The NPU*search* software is distributed as a tarball containing one or more packages (`.deb` or `.rpm`), an extra copy of the `npusearch` python for easier access/inspection without installing, and a setup script, `npusearch-install.sh` which will install extra dependencies and the npusearch packages, and then system install the `npusearch` python using `pip3`.

An up to date copy of the python **MUST** be system installed for an npusearch host to function correctly. This can be done either by installing via the setup script or by running `sudo pip3 install /opt/lrl/share/python/npusearch` if the `npusearch deb` or `rpm` package has already been installed.

Both the `npusearch-install.sh` script and manually installing the python with `pip` will install python packages from the `pip` repositories in addition to installing dependencies from the package manager. Installing the `deb` or `rpm` packages will not do this, and will only install dependencies which the package manager knows about and can uninstall.

By default `npusearch-install.sh` will install gluster from `ppa:gluster/glusterfs-${GLUSTER_VERSION}`. If this is not desired, it can be run with the environment variable `SKIP_GLUSTER=1` set.

The default version of gluster installed by `npusearch-install.sh` is currently 11. GlusterFS version 8 or higher is required, and version 10 or higher is recommended.

4.2 Storage Setup

The npusearch backends will look for files under the directories they have been assigned. If mount points for npusearch SSDs change, the npusearch backends on that host will need to be restarted for those changes to take effect. If npusearch partitions are mounted, unmounted, or moved around, or if the npusearch SSDs are reformatted in any way, a restart of the backends or of `npusearch.service` will be required.

The npusearch backends reread the files under `/var/lib/glusterd/vols` periodically. Gluster volumes can be created, destroyed, reconfigured, and renamed without restarting the backends. These changes may take up to 30 seconds for the backends to notice them.

If SED hardware encryption is in use on the npusearch SSDs, they must be unlocked before their partitions will be available and backends will come up. SED drives will also need to be unlocked, and their partitions mounted, before any gluster volumes they provide storage for will function.

If npusearch backends do not come up properly, ensure that all expected drives and partitions are visible, mounted, and accessible, and then try bringing the backends up again.

4.3 Controlling Backends

In general, the backends should be started and restarted using `systemctl [ start | restart | stop] npusearch.service`.

The method `NPUGlusterClient.quit` can be used to bring down backends remotely for the entire cluster. Passing `hostname=` to `quit` will bring down only backends on a particular host.

By default `NPUGlusterClient.quit` will cause backends to exit with code `0`, after which they will **NOT** be automatically restarted by systemd. An integer can be passed to `NPUGlusterClient.quit( ..., returncode=)`, which will cause the backends to exit with that code. A nonzero exit code will be default cause systemd to restart the backends.

A commandline wrapper of `NPUGlusterClient.quit` is provided as `npuearch_quit`.

4.4 Status and logs

A logfile can be specified in `/opt/lrl/etc/npusearch.conf` as `LOGFILE=` to cause backend and startup script logs to be written to that destination. By default `LOGFILE` is `/dev/null`.

When running under systemd, by default logs can be accessed via `sudo journalctl -u npusearch.service` independent of the `LOGFILE`.

Basic information about status of the unit can be retrieved with `sudo systemctl status npusearch.service`, which gives running state, uptime, the list of running processes for the service, memory usage, and some additional information.

Information about backends in the cluster can be retrieved via `NPUGlusterClient.info`, which returns hardware information about each backend that is reachable.

A commandline wrapper of `NPUGlusterClient.info` is provided as `npusearch_info`.

4.5 License Management

NPU*search* uses RLM for license management. Each backend needs to be able to check out a license in order to come up. Typically, licenses should be placed in `/opt/lrl/lib/npusearch/<file>.lic`. If desired, it may be possible to store license files in one central location made accessible under a local license server. In this case, a license for each node will still be required.

When requesting a license from LRL, the output of `/opt/lrl/bin/getLrlLicInfo` should be provided. This utility will collect and print basic system information as well a list of network interfaces which the license can be locked to.

4.6 Running Searches

Access to the NPU*search* backends is controlled by the `npusearch` python module. See the Python Documentation section for details.

4.7 Troubleshooting

See the Common Issues section for troubleshooting help.