Preparing the System

System Recommendations

General	Symas OpenLDAP will run on any system that meets the minimum requirements for the operating system where it is installed. Additional disk space and memory may be required to accommodate the application's data.
CPU	Symas OpenLDAP makes very efficient use of the system CPUs. In most cases, systems with 1 to 4 processors are adequate for all but the heaviest workloads. For those, up to 8 processors may be appropriate, but beyond that the factors that affect performance the most are the amount of memory and the bandwidth between it and the processors, network bandwidth, and disk I/O bandwidth.
Disk	The Symas OpenLDAP packages will require approximately 25MB. In addition, there should be enough space to accommodate the projected database size, the index files, and the transaction logs. These will be in separate file systems.
Memory	As with any database-type application, having sufficient memory is the key to performance. The system should have approximately 10% more memory than is needed to accommodate the entire database and its index files, plus the base amount of memory recommended by the OS vendor.

Preparing the System

To prepare your system to host Symas OpenLDAP directory services we recommend the following:

Disks & File Systems

LDAP databases serviced by the back-mdb (preferred), back-hdb and back-bdb backends can be stored on direct-attached disks (best) or iSCSI network-attached storage (your mileage will vary).

- For optimum performance, Symas recommends the use of at least three separate disk spindles (drives) for bdb/hdb backends and 2 spindles (drives) for mdb backends, each with its own independent file system. The first of these file systems should be dedicated to the database and its environment files, the second should contain the operating system and applications, and the final is for the database transaction logs (bdb/hdb only). The swap partition can be placed on the same drive as the operating system. This arrangement reduces drive head contention to the lowest possible point. RAID arrays do not enhance performance, but can be used to improve fault tolerance.

```
df -h
```

Example Output:

```
Filesystem              Size  Used Avail Use% Mounted on
/dev/mapper/centos-root  17G  2.2G   15G  13% /
/dev/sda1               472M  169M  279M  38% /boot
/dev/sdb1               470G   91G  356G  21% /var/symas
/dev/sdc1               470G  429G   17G  97% /var/log
```

- Test disk write speed (time it takes to sync data from cache to disk) by running the following command multiple times (to get an average). Requires BASH.

```
sync; dd if=/dev/zero of=1g.bin bs=1G count=1 >/dev/null 2>&1;
TIMEFORMAT="%Es Cache to Disk Sync Time"; time sync; rm -f
1g.bin
```

Example Output:

```
1.275s Cache to Disk Sync Time
1.173s Cache to Disk Sync Time
```

```
3.833s Cache to Disk Sync Time
```

Consider how many write operations per minute will be expected. If the Cache to Disk Sync Time when multiplied by 60 (seconds per minute) exceeds the expected writes per minute ***and*** the database backend to be used is MDB, see SLAPD.conf Customization → Database Section → Configuration → MDB section for optimization instructions.

Warning: If MDB optimizations are required skip the following step. **DO NOT** disable journaling.

- **NOTE:** For EXT4 partitions which are a journaling file system, disable the journaling for the file system that holds the database (MDB, HDB and BDB) and transaction log (HDB and BDB only) files. Since the Berkeley DB performs its own transaction management and journaling, file-system-based journaling only slows things down without adding any gains in reliability.

```
mkfs –t ext4 –O ^has_journal <device>

tune2fs –O ^has_journal <device>
```

Example:

```
mkfs –t ext4 –O ^has_journal /dev/xvdf/1

tune2fs –O ^has_journal /dev/xvdf/1
```

NOTE: Because journaling cannot be disabled for xfs file systems, we do NOT recommend using XFS. However, if XFS must be used, we recommend externalizing journaling to reduce the performance impact on the file system.

To reserve an external journal with a specified size when you create an XFS file system, specify the `-l logdev=device,size=size` option to the `mkfs.xfs` command. If you omit the `size` parameter, `mkfs.xfs` selects a journal size based on the size of the file system. To mount the XFS file system so that it uses the external journal, specify the `-o logdev=device` option to the `mount` command.

- Add the “noatime” (preferred) or “relatime” flag ONLY to the LDAP database partition

```
sudo vi /etc/fstab
      UUID=foo     /var   ext4   noatime,errors=continue    1 1
sudo mount -a
```

Swappiness

Tune the OS disk cache for best performance

Add new values to /etc/sysctl.conf:

```
sudo vi /etc/sysctl.conf
      vm.dirty_background_ratio = 50
      vm.dirty_ratio = 90
      vm.swappiness = 0
      vm.dirty_writeback_centisecs = 500
      vm.dirty_expire_centisecs = 60000
```

For more information on Disk caching and Performance tuning, please see the following:
https://lonesysadmin.net/2013/12/22/better-linux-disk-caching-performance-vm-dirty_ratio/
https://www.kernel.org/doc/Documentation/sysctl/vm.txt

Processes

Minimize the number of unneeded processes running on the system, particularly any that use a lot of memory or processor bandwidth, as these will have an adverse impact on directory server performance. In particular, be on the lookout for Java processes, as these tend to use large quantities of processing time and available memory. Additionally, ensure any processes using network ports required for OpenLDAP are disabled.

Network & Firewall

The default ports used by Symas OpenLDAP are 389/TCP for LDAP, 636/TCP for LDAP over SSL, and 88/TCP and 88/UDP for Kerberos key distribution (kdc). On many systems the netstat command can tell you about the ports that are in use and the processes that are using them.

IPTables

First, let's ensure IPTables is installed

```
which iptables
```

For newer versions it is important to disable FirewallD before installing IPTables

```
sudo systemctl stop firewalld

sudo systemctl disable firewalld

sudo systemctl mask firewalld

sudo yum install iptables-services -y

sudo systemctl enable iptables
```

Adjust iptables to save changes on stop/restart by inserting the following:

```
sudo vi /etc/sysconfig/iptables-config
IPTABLES_SAVE_ON_STOP="yes"
IPTABLES_SAVE_ON_RESTART="yes"
```

Adjust iptables allow **LDAP** and **LDAPS** protocol. This is done by adding the rules in red lines below.

```
sudo vi /etc/sysconfig/iptables
```

```
sudo apt-get update

sudo apt-get install iptables-persistent
```

Adjust iptables to allow **LDAP** and **LDAPS** protocol. This is done by exporting the current settings and adding the rules in red below.

```
sudo iptables-save > iptables.rules
```

```
sudo vi iptables.rules
```

```
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [192:19802]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 389 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 636 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

Be aware that the order is critical in the rules.
Port 22 is for SSH
Port 389 is for LDAP and LDAP over TLS
636 is for LDAPS

Save the Changes to IPTables

Note: The IPTables config is stored in /etc/sysconfig/iptables, and this is the file you have to update, otherwise the info will not be kept.

Restart of the **IPTables** service.

```
sudo systemctl restart iptables.service
```

Or

```
sudo service iptables restart
```

Save the changes

```
sudo service iptables save
```

Verify the changes

```
iptables –L -nv
```

Import the iptables.rules file and save the changes

```
sudo iptables-restore < iptables.rules

sudo service iptables-persistent save
```

Restart the **IPTables** Service

```
sudo service iptables-persistent reload
```

Verify the changes

```
iptables -S
```

FirewallD

FirewallD is frontend controller for iptables used to implement persistent network traffic rules. It provides command line and graphical interfaces and is available in the repositories of most Linux distributions. Working with FirewallD has two main differences compared to directly controlling iptables:

1. FirewallD uses zones and services instead of chain and rules.
2. It manages rulesets dynamically, allowing updates without breaking existing sessions and connections.

Note: FirewallD is a wrapper for iptables to allow easier management of iptables rules-it is not an iptables replacement. While iptables commands are still available to FirewallD, it's recommended to use only FirewallD commands with FirewallD.

FirewallD is included by default with RedHat 7+ but it's inactive. Controlling it is the same as with other systemd units.

To start the service and enable FirewallD on boot:

```
sudo systemctl start firewalld

sudo systemctl enable firewalld
```

To stop and disable it:

```
sudo systemctl stop firewalld

sudo systemctl disable firewalld
```

To check the firewall status. The output should say either running or not running.

```
sudo firewall-cmd --state
```

To view the status of the FirewallD daemon:

```
sudo systemctl status firewalld
```

Example output:

```
firewalld.service - firewalld - dynamic firewall daemon
       Loaded: loaded
       (/usr/lib/systemd/system/firewalld.service;
       disabled)
       Active: active (running) since Wed 2015-09-02
       18:03:22 UTC; 1min 12s ago
Main PID: 11954 (firewalld)
       CGroup: /system.slice/firewalld.service
            └─11954 /usr/bin/python -Es /usr/sbin/firewalld
       --nofork --nopid
```

To reload a FirewallD configuration:

```
sudo firewall-cmd reload

sudo systemctl enable firewalld
```

Configuring FirewallD

Firewalld is configured with XML files. Except for very specific configurations, you won't have to deal with them and the firewall-cmd should be used instead.

Configuration files are located in two directories:

`/usr/lib/FirewallD`

Holds default configurations like default zones and common services. Avoid updating them because those files will be overwritten by each firewalld package update.

`/etc/firewalld`

Holds system configuration files. These files will overwrite a default configuration.

Configuration Sets

Firewalld uses two configuration sets: Runtime and Permanent. Runtime configuration changes are not retained on reboot or upon restarting FirewallD whereas permanent changes are not applied to a running system. By default, firewall-cmd commands apply to runtime configuration but using the --permanent flag will establish a persistent configuration. To add and activate a permanent rule, you can use one of two methods.

1. Add the rule to both the permanent and runtime sets for ldap and ldaps.

```
sudo firewall-cmd --zone=public --add-service=ldap

sudo firewall-cmd --zone=public --add-service=ldaps
```

2. Add the rule to the permanent set and reload FirewallD.

```
sudo firewall-cmd --zone=public --add-service=ldap --permanent

sudo firewall-cmd --zone=public --add-service=ldaps --permanent

sudo firewall-cmd --reload
```

Note: to view all available services run the following:

```
sudo firewall-cmd --get-services
```

Note: To view only enabled/active services run the following:

```
sudo firewall-cmd --list-services
```

UFW

Uncomplicated Firewall

One other potential blocker is the UFW configuration.

Install the graphical UFW interface (Optional):

```
sudo apt-get install gufw -y
```

We may need to adjust it to allow **LDAP** and **LDAPS** protocol. This is done by adding these rules from the command line:

```
sudo ufw allow 389

sudo ufw allow 636
```

```
sudo ufw reload
```

These rules call also be set in the GUFW interface (if installed)

```
sudo gufw
```

PF

ORACLE
SOLARIS

Solaris Firewall

By default PF Firewall is not installed.

To install:

```
sudo pkg install firewall
```

Ensure IPFilter is disabled (will prevent PF from loading)

```
svcadm disable ipfilter
```

Add rules to Packet Filter configuration file

```
sudo vi /etc/firewall/pf.conf
```

```
anchor "_auto/*"
set skip on lo0
set reassemble yes no-df
block log
pass in from any to any port = 22
pass in proto tcp quick from any to any port = 389
pass in proto tcp in from any to any port = 636
pass out
```

Refresh the PF service

```
svcadm refresh firewall
```

FreeBSD

FreeBSD has three firewalls built into the base system: PF, IPFW, and IPFILTER, also known as IPF. FreeBSD also provides two traffic shapers for controlling bandwidth usage: altq(4) and dummynet(4). We recommend reviewing the FreeBSD handbook for greater detail on configuring the firewall that suits your needs. However, by default no firewall is configured during the Operating System installation.

https://www.freebsd.org/doc/handbook/firewalls.html

Time Synchronization

It is good practice under all circumstances, and essential if configuring replication or using Kerberos, to make sure the system clocks are synchronized with each other and one of the standard time servers. The Network Time Protocol (NTP) is widely supported and relatively easy to set up. Consult your operating system documentation for further information.

Disable SELINUX

Check to see if SELinux is installed

```
rpm -qa |grep selinux
```

```
dpkg --list |grep selinux
```

Check to see if SELinux (Security-Enhanced Linux) is enabled

```
sestatus
```

No response indicates SELinux is not installed/enabled

If enabled, edit the /etc/sysconfig/selinux file. This file is a symlink to /etc/selinux/config. The configuration file is self-explanatory. Changing the value of SELINUX or SELINUXTYPE changes the state of SELinux and the name of the policy to be used the next time the system boots.

```
sudo vi /etc/selinux/config
    SELINUX=permissive or enforcing
```

Change “permissive” or “enforcing” to “disabled”

Configure System Logging

In a production environment the best performance is obtained when the log file directory is on a separate disk from the database directory.

Enable rsyslog in RedHat/CentOS 7+

Symas OpenLDAP adds logs to the OS Syslog. However, for RedHat 7+, rsyslog was implemented instead of syslog and it does not come automatically installed. Determine if rsyslog is installed:

```
rpm -qa | grep rsyslog
```

If not, install it using the following command:

```
sudo yum install rsyslog -y
```

Disable the syslog/system bridge in RedHat/CentOS 7+

With RedHat 7, RedHat made a bridge between syslog and systemd's binary logging. This bridge destroys performance due to serious deficiencies with systemd. For reasonable performance on RedHat 7 then, it is necessary to remove this bridge from the rsyslog configuration.

Modify /etc/rsyslog.conf and comment out (#) the following lines:

```
sudo vi /etc/rsyslog.conf
    $ModLoad imjournal # provides access to the system journal
    $OmitLocalLogging on
    $IMJournalStateFile imjournal.state
```

Remove the /etc/rsyslog.d/listen.conf file

```
sudo rm -rf /etc/rsyslog.d/listen.conf
```

Configure syslog/rsyslog

Note: Most Linux OSes will have rsyslog installed or available for installation. Some OSes such as older versions of RedHat, FreeBSD and Solaris still use syslog. For any of the following commands that contain `"rsyslog"`, `"syslog"` can be used instead, if it is installed.

For all Linux OSes

SLAPD automatically logs to local4, so we recommend the `/etc/rsyslog.conf` file or `/etc/rsyslog.d/slapd.conf` file have the following RULE added:

Rsyslog.conf Log Location (option 1 – static configuration)

Modify /etc/rsyslog.conf or /etc/syslog.conf with the following just below the *RULES* section header

```
sudo vi /etc/rsyslog.conf
```

```
local4.*            -/var/log/slapd.log
```

If additional SLAPD processes are configured be sure to add rules that match the slapd arguments used in `/opt/symas/etc/openldap/symas-openldap.conf`

```
EXTRA_SLAPD_ARGS="-n slapd2 -l LOCAL5 -f /<path to>/slapd2.conf"
```

by setting the log location in /etc/rsyslog.conf to

```
local5.*            -/var/log/slapd2.log
```

Rsyslog.d Log Location (option 2 – dynamic configuration)

Modify rsyslog.conf to include the /etc/rsyslog.d directory in the GLOBAL DIRECTIVES section

```
sudo vi /etc/rsyslog.conf
```

```
IncludeConfig /etc/rsyslog.d/*.conf
OR
include      /etc/rsyslog.d
```

Create /etc/rsyslog.d directory

```
sudo mkdir /etc/rsyslog.d
```

Create configuration file for each SLAPD process

```
sudo vi slapd.conf
```

Add the following contents:

```
local4.*                    -/var/log/slapd.log
```

Create additional slapd(2)(3)... configuration files with matching local(5)(6)... log file paths.

Create log files in directories specified

```
sudo touch /var/log/slapd.log
```

Set Permissions

```
sudo chown -R syslog:adm /var/log/slapd.log
```

Restart Rsyslog

After all rules have been added, restart RSYSLOG

```
sudo systemctl restart rsyslog.service
```

Or

```
sudo service rsyslog restart
```

To view Syslog or the slapd log real-time use the following command

```
tail -f /var/log/syslog
```

Or

```
tail -f /var/log/slapd.log
```

Log Levels

The content saved to the slapd.log file is determined by the loglevel(s) set in the slapd.conf file. Symas recommends log levels in a MMR or MSR environment be set to "stats sync". See the Logging Section of SLAPD.conf Customization for more details.

Rotation/Retention

It is important the /var/log directory be located on a separate disk from your OpenLDAP installation and database files. For Linux operating systems Symas recommends logrotate be used.

http://linux.die.net/man/8/logrotate

1. Verify logrotate is installed. If so, skip to step 4.

```
which logrotate
```

2. If not, install it using the following command:


```
sudo yum install logrotate -y
```


```
sudo apt-get install logrotate -y
```

```
sudo zypper in logrotate
```

```
sudo pkg install logrotate
```

3. Verify the `logrotate.conf` file includes the `logrotate.d` directory



```
sudo vi /etc/logrotate.conf
```

If not, add the following:

```
include /etc/logrotate.d
```

```
sudo vi /usr/local/etc/logrotate.conf
```

If not, add the following:

```
include /usr/local/etc/logrotate.d
```

4. Create the logrotate configuration for your database log

```
sudo vi /etc/logrotate.d/slapd
```

OR

```
sudo vi /usr/local/etc/logrotate.d/slapd
```

See the logrotate man page for specific options to use in this configuration

Note: Follow existing company retention guidelines.

Note: The path to the log on the first line must be the absolute path and match what is configured in rsyslog.conf. Spacing is also important so following the spacing guidelines in *italics.*

```
/var/log/slapd.log
{
  rotate 7    (indented 2 spaces)
  daily
  missingok
  notifempty
  compress
  postrotate
    kill -HUP `cat /var/run/syslog*.pid 2>/dev/null` || true
              (indented 4 spaces)
  endscript
}
```

```
/var/log/slapd.log
{
  rotate 7    (indented 2 spaces)
  daily
  missingok
  notifempty
  compress
  postrotate
    service rsyslog rotate>/dev/null 2>&1 ||true   (indented 4 spaces)
  endscript
}
```

debian 9 ubuntu 18 and newer

```
{
        rotate 7
        daily
        missingok
        notifempty
        compress
        postrotate
                /usr/lib/rsyslog/rsyslog-rotate
        endscript
}
```

```
sudo vi /usr/lib/rsyslog/rsyslog-rotate
```

Verify the following content:

```
#!/bin/sh

if [ -d /run/systemd/system ]; then
    systemctl kill -s HUP rsyslog.service
else
    invoke-rc.d rsyslog rotate > /dev/null
fi
```

This rotate script performs the following:

rotate 7: 7 copies of the /var/log/slapd.log file are kept
daily: The /var/log/slapd.log file is copied daily
missingok: Missing log files do not stop process or issue an error
notifempty: An empty log will not be rotated
compress: Old logs are compressed with gzip
postrotate: Executes the rotate command

5. Create an /etc/cron/daily file or verify it exists and is configured as follows which will execute the logrotate command daily:

redhat LINUX CentOS SUSE

```
sudo vi /etc/cron.daily/logrotate
```

Add or verify the following contents:

```
#!/bin/sh
/usr/sbin/logrotate /etc/logrotate.conf
EXITVALUE=$?
if [ $EXITVALUE != 0 ]; then
    /usr/bin/logger -t logrotate "ALERT exited abnormally with
[$EXITVALUE]"
fi
exit 0
```

FreeBSD

POWERED BY symas LMDB

```
sudo crontab -e
## Logrotate at 1 AM in the morning

0 01 * * * root /usr/local/sbin/logrotate
/usr/local/etc/logrotate.d/slapd > /dev/null 2>&1
sudo service cron restart
```

6. Test LogRotate

   Once you have created a new logrotate configuration file within /etc/logrotate.d create a matching log file if it does not already exist:

   ```
   sudo echo "rotate my log file" > /var/log/slapd.log
   ```

   Once your log file is in place force logrotate to rotate all logs with -f option.

   ```
   sudo logrotate -f /etc/logrotate.conf
   ```

   Warning: The above command will rotate all your logs defined in the /etc/logrotate.d directory. To avoid this add -d to the command (dry run).

7. Now visit again your /var/log/ directory and confirm that your log file was rotated and new log file was created.

   ```
   ll /var/log/
   -rw-r--r--   1 syslog adm          0 Aug 13 17:12 slapd.log
   -rw-r--r--   1 syslog adm         39 Aug 13 16:56 slapd.log.1.gz
   ```

ORACLE

SOLARIS

Add rotation rules to the logs by running the following command:

```
sudo /usr/sbin/logadm -C7 -N -p 1d -w /var/log/slapd.log -z 1
```

Test LogAdm Log Rotation

Once you have created a new logrotate configuration file within /etc/logrotate.d create a matching log file if it does not already exist:

```
sudo echo "rotate my log file" > /var/log/slapd.log
```

Once your log file is in place force logrotate to rotate all logs with -f option.

```
sudo /usr/sbin/logadm -p now /var/log/slapd.log
```

Now visit again your /var/log/ directory and confirm that your log file was rotated and new log file was created.

Install Recommended Programs

Recommended Packages:

TMUX, TelNet, GCC, GDB, WGET, NCurses-Devel, Perl, Perl-Core, Python, Python-DateUtil

redhat LINUX CentOS

```
sudo yum install epel-release -y

sudo yum repolist
```

```
sudo yum install tmux telnet gcc gdb wget ncurses-devel perl
perl-core python python-dateutil -y
```


```
sudo apt-get update

sudo apt-get install tmux telnet build-essential gcc gdb wget
libncurses5-dev libncursesw5-dev perl python python-dateutil -y
```

```
sudo zypper refresh

sudo zypper update

sudo zypper install tmux telnet gcc-c++ gdb wget ncurses-devel
python-dateutil
```

```
sudo pkg refresh --full

sudo pkg update --accept

sudo pkg install tmux pkg://solaris/service/network/telnet gcc
gdb wget ncurses python/dateutil
```

For Solaris 11 add existing Perl installation to path

```
sudo export PATH=$PATH: /usr/perl5/bin
```

Additional for Solaris 10 **only**

```
sudo pkgadd -d http://get.opencsw.org/now

/opt/csw/bin/pkgutil -U

/opt/csw/bin/pkgutil -y -i perl

/usr/sbin/pkgchk -L CSWperl # list files
```

```
sudo pkg install tmux gcc gdb wget devel/ncurses py27-dateutil
```

Optional: Download and install Java Runtime Environment (for Apache Directory Studio):
https://java.com/en/download/help/linux_x64rpm_install.xml

Download the Symas OpenLDAP package(s) for your operating system and processor architecture from https://downloads.symas.com and place them in a directory where they are accessible. Note that you need to have a support subscription in place to download the Gold packages. Contact Symas or send email to sales@symas.com to obtain subscription information.

Set Environment Paths Automatically

To set LDAPCONF, PATH and MANPATH environment variables to be created automatically on boot-up on Linux Operating Systems, create a shell script that creates/updates these environment variables. Depending on how your system is configured, you may want to set the /opt/symas paths to follow the existing environment variables.

This script sets the executable path and manpath. It also sets the LDAPCONF environment variable which is needed for the ldap* commands if you need to use SSL/TLS.





```
sudo vi /etc/profile.d/symasenv.sh
```

Put this in the symasenv.sh file:

```
if  [ -d "/opt/symas" ]; then
export LDAPCONF=/opt/symas/etc/openldap/ldap.conf
export PATH=/opt/symas/bin:$PATH
export MANPATH=/opt/symas/share/man:$MANPATH
fi
```

Make the file executable:

```
sudo chmod +x /etc/profile.d/symasenv.sh
```

ORACLE

SOLARIS

Mozilla OpenLDAP is integrated into Solaris OS functionality and cannot be removed. Therefore it is important to set the Path to use Symas executables instead.

Modify /etc/profile

```
sudo vi /etc/profile
```

Add the following

```
export LDAPCONF=/opt/symas/etc/openldap/ldap.conf
export PATH=/opt/symas/bin:$PATH:/usr/perl5/bin
export SUPATH=/opt/symas/bin:/usr/bin:/usr/sbin
export MANPATH=/opt/symas/share/man:$MANPATH
```

```
sudo vi /etc/profile
if  [ -d "/opt/symas" ]; then
export LDAPCONF=/opt/symas/etc/openldap/ldap.conf
export PATH=/opt/symas/bin:$PATH
```

```
export MANPATH=/opt/symas/share/man:$MANPATH
fi
```

Core Retention

Sometimes a bug in a piece of software will cause it to crash, and, if enabled, create a core file. This core file can often be used by Symas Technical Support to examine what problem occurred in order to fix it.

Check Core Dump Permission

On most Unix/Linux systems, the creation of core dumps by non-admin/root users is disabled. However, since slapd runs as root (unless otherwise configured in /opt/symas/etc/openldap/symas-openldap.conf) this is normally not applicable. If using a non-root account to run slapd, continue with the following. Core file creation is controlled by the ulimit command. A few things to note about ulimit are:

- Desired ulimit settings can be set in default shell profiles and/or the /etc/security/limits.conf file (depending on the system).
- Any change to ulimit settings during a login session are valid only for the login session and will not apply to new or parallel sessions. Permanent changes must be made to the user/global login profile or limits.conf file.
- There are two types of limits: soft and hard. Soft limits are the maximum ulimit values that may be set for a user. Hard limits are the maximum ulimit values allowed system-wide and cannot be overridden by the user.
- When the ulimit command is used, the limits that are listed or set are for that login process only. Ulimit cannot be used to list or update the limits for other system/user processes.

In the following example, the ulimit command is used to check the hard and soft limits for core files:

Checking Soft (-Sc) and Hard (-Hc) ulimits for Core File sizes

```
ulimit -Sc
        0
ulimit -Hc
        unlimited
```

There are three values that may be used for the core file setting in ulimit:

Value	Function
0	Setting the value to zero disables the creation of core files
N	Setting the value to any number above zero will be the maximum size in blocks. Core files that are greater than N blocks will be truncated which renders the core file useless for troubleshooting.
unlimited	There are no restrictions to the size of the core file. The full core file will be created, however, if the size of the core file is greater than the amount of available disk space, the filesystem may be filled causing further issues.

ORACLE

SOLARIS

Use the coreadm command to view current settings.

```
sudo coreadm
        global core file pattern:
```

```
global core file content: default
kernel zone core file pattern:
init core file pattern: core
init core file content: default
global core dumps: disabled
kernel zone core dumps: disabled
per-process core dumps: enabled
global setid core dumps: disabled
per-process setid core dumps: disabled
global core dump logging: disabled
```

(Optional) You can set a specific path where core files will be saved automatically by using the following command:

```
sudo coreadm -e global -g /var/core/core.%f.%p
```

Enabling Core Dumps

Warning: Depending on the size of the LDAP database, core dumps can be extremely large. Before enabling core dumps, make sure that your server has enough disk space to accommodate the core file.

RedHat/CentOS 6+ Users

Core files may be handled differently in RedHat 6+ by the abrtd daemon, if installed, which causes core dumps to be automatically deleted. See Preventing Core Dump Deletion for more information and instructions.

Older Linux Operating Systems

In order for core dumps to be taken, the ulimit core size needs to be set to “unlimited” before slapd is started.

On RedHat/CentOS 6

```
sudo echo "DAEMON_COREFILE_LIMIT='unlimited'" >>
/etc/sysconfig/init

reboot
```

The best way to do this is to add the ulimit command to the /opt/symas/etc/openldap/symas-openldap.conf file:

```
sudo vi /opt/symas/etc/openldap/symas-openldap.conf
# Symas OpenLDAP Configuration file
# Copyright (c) 2015 Symas Corporation. All Rights Reserved.
#
# This file contains configuration information for Symas OpenLDAP.
# Refer to the comments just before each variable to determine proper
# settings.
# Set ulimit to allow core dumps
ulimit -c unlimited
…
```

It is also possible to enable core dumps at the OS level by editing the limits.conf file.

```
sudo vi /etc/security/limits.conf
```

Change

```
# *                soft   core   0
```

To

```
(slapd uid or *)    soft   core   unlimited
```

Add

```
(slapd uid or *)    hard   core   unlimited
```

Logout/Login is required for change to take effect.

All SystemD-Based Operating Systems

Warning: RedHat/CentOS 7+, Ubuntu 16+, Debian 8+, SLES 12+ all use systemd which ignores `/etc/security/limits.conf` and `/etc/security/limits.d/*` settings. To permit core files without size limits on systemd-based systems use the following process instead.

Note that these steps must be done as root:

```
sudo -s

cd /etc/systemd/system

mkdir solserver.service.d

cd solserver.service.d

printf "[Service]\nLimitCORE=infinity\n" > override.conf

systemctl daemon-reload
```

Now you can verify the change has taken effect for future slapd startup:

```
systemctl cat slapd
```

This should show the updated limits:

```
# /etc/systemd/system/solserver.service.d/override.conf
[Service]
LimitCORE=infinity
```

Once Symas OpenLDAP is installed you can view the process's limits file to see the Max open files limit:

```
pgrep slapd

cat /proc/<slapd PID>/limits | grep "Max core file"
```

```
Max core file size     unlimited     unlimited            bytes
```

If adjusting these settings after Symas OpenLDAP is installed and slapd is running, a restart of solserver is required to pick up the change.

```
systemctl restart solserver
```

FreeBSD

```
sudo vi /etc/rc.conf
dumpdev="AUTO"
dumpon_flags=""
```

```
dumpdir="/var/crash"
savecore_enable="YES"
savecore_flags="-m 10"
```

```
sudo mkdir /var/crash
```

Note: savecore_flags="-m 10" - only the 10 most recent kernel dumps are saved.

```
sudo vi /etc/login.conf
```

```
default:\
        :coredumpsize=unlimited:\
```

Generating Usable Core Dumps in Debian/Ubuntu

Unfortunately, generating usable core files on Ubuntu requires disabling the apport process. This is likely not a big deal unless you want various crashes reported upstream to Ubuntu. To do this, edit /etc/default/apport, and set enabled to 0, then stop the services:

```
service apport stop

sudo vi /etc/sysctl.d/60-slapd-core.conf
```

```
kernel.core_uses_pid=1
fs.suid_dumpable=2
kernel.core_pattern=/opt/symas/data/tmp/core-%e-%s-%u-%g-%p-%t
```

Then run

```
sudo service procps start
```

Creating Core Dumps in Linux

Create Core Dump by Killing Slapd

If the process is still running and you need to kill it, send the slapd process a SIGQUIT signal:

1. Get the slapd process ID

```
ps -C slapd
```

```
            PID TTY      TIME   CMD
            23348 ?    15:35:33  slapd
```

2. Kill the process with a SIGQUIT signal

```
sudo kill -3 23348
```

After you kill the process, there should be a file with the name “core” in it in the / (root) directory.

Create a Core Dump without Killing Slapd

If you need a core dump but don't want or need to kill slapd, you can obtain the core dump by using the gcore utility. This will create a core without disturbing the slapd process. The gcore utility is not installed by default on some systems. To ensure its availability, install gdb (see Prerequisites → Preparing the System → Install Recommended Programs).

To obtain a core with gore, get the process id of slapd, then call gcore:

1. Get the slapd process ID

```
ps -C slapd
```

```
  PID TTY          TIME CMD
23348 ?        15:35:33 slapd
```

2. Run gcore with the slapd process id

```
sudo gcore 23348
```

The resulting core file should be in your *current* working directory and should have the id of the process that was dumped in the filename.

Set Default Save-To Location for Core Files

To cause all core files to get created in /tmp/, /opt/symas/tmp/ or any other directory, set the following settings in /etc/sysctl.conf.

The default setting kernel.core_pattern=core means the core file gets created in whatever the current working directory was at the time the process started, which is less useful, as that could be anywhere on the system and may not be writable by the executing process.

Note: This setting gets overridden on RHEL6/CentOS6 by default if ABRT is installed.

```
sudo vi /etc/sysctl.conf
```

```
kernel.core_uses_pid=1
```

This appends the PID to the generated core file, allowing multiple core dumps.

```
fs.suid_dumpable=2
```

This parameter allows binaries that are setuid (like slapd) to drop core.

```
kernel.core_pattern = /tmp/core-%e-%s-%u-%g-%p-%t
```

Or for the GNR release and later

```
kernel.core_pattern = /opt/symas/tmp/core-%e-%s-%u-%g-%p-%t
```

Filename variables:

%e is the filename
%g is the gid the process was running under
%p is the pid of the process
%s is the signal that caused the dump
%t is the time the dump occurred
%u is the uid the process was running under

Create the /opt/symas/tmp directory

```
sudo mkdir /opt/symas/tmp
```

After the modifications are finished execute the following command so that a reboot is not required (applies to most OSes)

```
sudo /sbin/sysctl -p /etc/sysctl.conf
```

Preventing Core Dump Deletion

After a slapd crash, the core dump generated is deleted automatically by abrtd on RedHat/CentOS 6, if abrtd is installed. The OpenGPGCheck setting in /etc/abrt/abrt-action-save-package-data.conf is set to “yes” (this is the default setting). When set to “yes”, abrtd will only process packages that are signed with a GPG key located in /etc/abrt/gpg_keys. Any core that isn't generated from a signed package is automatically deleted.

Solution: Change the value for the OpenGPGCheck to “no”.

1. Open /etc/abrt/abrt-action-save-package-data.conf
```
sudo vi /etc/abrt/abrt-action-save-package-data.conf
```
2. Change the value of OpenGPGCheck to no.
3. Save the changes and close the editor

Note: It is not necessary to restart the abrt daemon or slapd for the change to go in to effect.

Solaris, SuSE and FreeBSD do not automatically delete core dumps, thus they can, over time, accumulate and waste valuable storage space. Because they can be saved in any random directory, use the following command to remove them.

```
find /* -name core -exec rm {} \;
```

Or

```
find / -name core -exec du -hsc {} ;
```

Upload Core Dumps to Symas

Follow the instructions in the Support → Ticket Creation section to request SFTP access to a shared folder where you can upload the core dump file.

Files Open per Process

Older Linux Operating Systems

The default number of files (nofile) allowed to be opened by any specific process can vary depending on the operating system. Defaults are typically 1024 - 4096. For large customers with high traffic volume on OpenLDAP servers these limits can cause slapd to return PANIC errors rather than completing the requested operations.

Desired nofile settings can be set in default shell profiles and/or the /etc/security/limits.conf file (depending on the system).

Any change to nofile settings during a login session are valid only for the login session and will not apply to new or parallel sessions. Permanent changes must be made to the user/global login profile or limits.conf file.

There are two types of limits: soft and hard. Soft limits are the maximum values that may be set for a user. Hard limits are the maximum values allowed system-wide and cannot be overridden by the user.

Use the following ulimit command to check the soft (-Sn) and hard (-Hn) limits for the number of files (nofile) allowed per process:

```
ulimit -Sn
        1024
```

```
ulimit -Hn
        4096
```

Setting nofile limits in Limits.conf

```
sudo vi /etc/security/limits.conf
```

Insert the following at the bottom of the file:

```
        (slapd uid or *)     soft  nofile 524288
        (slapd uid or *)     hard  nofile 524288
```

Logout/Login is required for change to take effect.

SystemD-Based OSes

Warning: RedHat/CentOS 7+, Ubuntu 16+, Debian 8+, SLES 12+ all use systemd which ignores `/etc/security/limits.conf` and `/etc/security/limits.d/*` settings. To increase nofile limits on systemd-based systems use the following process instead.
Note that these steps must be done as root:

```
sudo -s

cd /etc/systemd/system/solserver.service.d

printf "LimitNOFILE=524288\n" >> override.conf

systemctl daemon-reload
```

Now you can verify the change has taken effect for future slapd startup:

```
systemctl cat slapd
```

This should show the updated limits:

```
        # /etc/systemd/system/solserver.service.d/override.conf
        [ [Service]
        LimitCORE=infinity
        LimitNOFILE=524288
```

Once Symas OpenLDAP is installed you can view the process's limits file to see the Max open files limit:

```
pgrep slapd

cat /proc/<slapd PID>/limits | grep "Max open files"
        Max open files      524288 524288       files
```

If adjusting these settings after Symas OpenLDAP is installed and slapd is running, a restart of solserver is required to pick up the change.

```
systemctl restart solserver
```

ORACLE
SOLARIS

Increase Solaris file open limits
Append the following to the end of `/etc/system`

```
sudo vi /etc/system
        set rlim_fd_max=4096
        set rlim_fd_cur=1024
```

```
sudo vi /etc/login.conf
default:\
        :openfiles=unlimited:\
```

Virtual Environment Options

Virtual Memory

Older Linux OSes

Configuring VMs to utilize all available virtual memory in symas-openldap.conf

```
sudo vi /opt/symas/etc/openldap/symas-openldap.conf
# Symas OpenLDAP Configuration file
# Copyright (c) 2015 Symas Corporation. All Rights Reserved.
#
# This file contains configuration information for Symas OpenLDAP.
# Refer to the comments just before each variable to determine proper
# settings.
# Set ulimit to allow core dumps
ulimit -v unlimited
…
```

Note: -v allows for unlimited use of virtual memory (particularly useful in virtual environments).

SystemD-Based OSes

RedHat/CentOS 7+, Ubuntu 16+, Debian 8+, SLES 12+ all use systemd which ignores `/etc/security/limits.conf`, `/etc/security/limits.d/*` and `/opt/symas/etc/openldap/symas-openldap.conf` settings. To configure your VM to utilize all available virtual memory on systemd-based systems use the following process instead.

Note that these steps must be done as root:

```
sudo -s

cd /etc/systemd/system/solserver.service.d

printf "LimitAS=infinity\n" >> override.conf

systemctl daemon-reload
```

Now you can verify the change has taken effect for future slapd startup:

```
systemctl cat slapd
```

This should show the updated limits:

```
# /etc/systemd/system/solserver.service.d/override.conf
[Service]
LimitCORE=infinity
LimitNOFILE=524288
LimitAS=infinity
```

Once Symas OpenLDAP is installed you can view the process's limits file to see the Max open files limit:

```
pgrep slapd
```

```
cat /proc/<slapd PID>/limits | grep "Max address space"
```

```
Max address space   unlimited    unlimited            bytes
```

If adjusting these settings after Symas OpenLDAP is installed and slapd is running, a restart of solserver is required to pick up the change.

```
systemctl restart solserver
```

I/O Scheduler for Linux Operating Systems

For all Linux OSes running on virtual machines (VMWare, Xen, KVM and VirtualBox) it is important to set the default I/O Scheduler to "noop".

The Noop scheduler is a unique scheduler. Rather than prioritizing specific I/O operations, it simply places all I/O requests into a FIFO (First in, First Out) queue. While this scheduler does try to merge similar requests, that is the extent of the complexity of this scheduler. This scheduler is optimized for systems that essentially do not need an I/O scheduler. This scheduler can be used in numerous scenarios such as environments where the underlying disk infrastructure is performing I/O scheduling on Virtual Machines. Since a VM is running within a Host Server/OS, that host already may have an I/O scheduler in use. In this scenario, each disk operation is passing through two I/O schedulers: one for the VM and one for the VM Host.

```
sudo -s

echo noop > /sys/block/sda/queue/scheduler

cat /sys/block/sda/queue/scheduler
```

Should return the following:

```
[noop] deadline cfq
```

With the above, the scheduler has been changed to the Noop scheduler. Below are benchmarking results to measure the impact of this I/O scheduler.

```
starting vacuum...end.
transaction type: TPC-B (sort of)
scaling factor: 50
query mode: simple
number of clients: 100
number of threads: 2
number of transactions per client: 1000
number of transactions actually processed: 100000/100000
latency average: 46.364 ms
tps = 2156.838618 (including connections establishing)
tps = 2157.102989 (excluding connections establishing)
```

From the above, we can see that we were able to reach 2,156 transactions per second which is a slight improvement over the Deadline scheduler and far superior to the CFQ scheduler.

VMWare Tools (if using VMWare)

If VMware tools were installed previously via the local vSphere, remove them using the following command:

```
sudo /usr/bin/vmware/vmware-uninstall-tools.pl
```

Search for the respective VMWare repository from https://packages.vmware.com/tools/esx/5.1/repos/index.html

Install VMware and/or VMWare Tools

```
sudo yum install
http://packages.vmware.com/tools/esx/5.1/repos/vmware-tools-
repo-RHEL6-9.0.0-2.x86_64.rpm -y
```

```
sudo yum install vmware-tools-esx-nox -y
```

```
sudo apt-get install vmware-tools-esx-nox -y
```

Right-click the VMWare guest in the VMWare client, and click Install VMWare tools. This may also be under a Guest tab after right-clicking on the guest VMWare

```
mount /dev/cdrom /media

cp /media/*.tar.gz /tmp

cd /tmp

tar -zxvf VM*.tar.gz

/tmp/vmware-tools-distrib/vmware-install.pl --default
```

In the VSphere Client window click Virtual Machine → VM → Guest → Install/Upgrade VMWare Tools → OK

```
cp  /cdrom/vmwaretools/vmware-solaris-tools.tar.gz /tmp

cd /tmp

gunzip vmware-solaris-tools.tar.gz

tar xvf vmware-solaris-tools.tar

cd vmware-tools-distrib

./vmware-install.pl
```

Press ENTER to accept default values

Startup service by default

```
ln -s /etc/init.d/vmware-tools /etc/rc3.d/S98vmware-tools
```

1. Power on the virtual machine.
2. Select **VM** > **Install VMware Tools**.

The remaining steps take place inside the virtual machine, not on the host computer.

3. Be sure the guest operating system is running in text mode. You cannot install VMware Tools while X is running.
4. As root (su -), mount the VMware Tools virtual CD-ROM image, change to a working directory (for example, /tmp), unzip the installer, then unmount the CD-ROM image.

   Note: *You do not use an actual CD-ROM to install VMware Tools, nor do you need to download the CD-ROM image or burn a physical CD-ROM of this image file. The VMware Workstation software contains an ISO image that looks like a CD-ROM to your guest operating system. This image contains all the files needed to install VMware Tools in your guest operating system.*

   Note: *Some FreeBSD distributions automatically mount CD-ROMs. If your distribution uses automounting, do not use the mount and umount commands below. You still must untar the VMware Tools installer to /tmp.*

```
mount /cdrom
cd /tmp
```

5. Untar the VMware Tools tar file:

```
tar zxpf /cdrom/vmware-freebsd-tools.tar.gz
umount /cdrom
```

6. Run the VMware Tools installer.

```
cd vmware-tools-distrib
./vmware-install.pl
```

7. Log out of the root account.

```
exit
```

8. Start X and your graphical environment
9. In an X terminal, launch the VMware Tools background application.

```
vmware-toolbox &
```

   Note: You may run VMware Tools as root or as a normal user. To shrink virtual disks, you must run VMware Tools as root (su -).

Note: In a FreeBSD 4.5 guest operating system, sometimes VMware Tools does not start after you install VMware Tools, reboot the guest operating system or start VMware Tools on the command line in the guest. An error message appears:

```
Shared object 'libc.so.3' not found.
```

The required library was not installed. This does not happen with full installations of FreeBSD 4.5, but does occur for minimal installations. To fix the problem of the missing library, take the following steps:

1. Insert and mount the FreeBSD 4.5 installation CD or access the ISO image file.
2. Change directories and run the installation script.

```
cd /cdrom/compat3x
./install.sh
```

Reboot

```
sudo reboot
```

Verify service status

```
/etc/init.d/vmware-tools status
```

VirtualBox Guest Additions (if using Oracle VM VirtualBox)

```
sudo yum install kernel-devel dkms -y

sudo yum groupinstall "Development Tools"
```

```
sudo apt-get install module-assistant dkms; -y

m-a prepare
```

Click on Install Guest Additions… from the Devices menu and then run the following in a terminal window:

```
mount /media/cdrom

cd /media/cdrom

sh ./VBoxLinuxAdditions.run
```

Follow the instructions on screen and reboot system

```
sudo reboot
```

(Optional) After rebooting the system, if access to shared folders is denied, run the following command:

```
sudo usermod -aG vboxsf $(whoami)
```

Install gcc and make

```
sudo zypper in gcc make
```

Find the kernel type

```
uname -r
```

e.g. `3.4.28-2.20-desktop`

Update the kernel

```
sudo zypper update kernel-desktop
```

Install the kernel development files

```
sudo zypper in kernel-devel
```

Reboot

```
init 6
```

Click Devices -> Install Guest Additions in VirtualBox window.
Browse to the Guest Addition CD from a terminal.

```
cd /run/media/VBOXADDITIONS_#.#.#_#####
```

Run the Linux installer as root.

```
sudo ./VBoxLinuxAdditions.run
```

Note: These commands are run in the FreeBSD guest.

1. First, install the emulators/virtualbox-ose-additions (https://www.freebsd.org/cgi/url.cgi?ports/emulators/virtualbox-ose-additions/pkg-descr) package or port in the FreeBSD guest. This will install the port:

```
cd /usr/ports/emulators/virtualbox-ose-additions && make
install clean
```

2. Add these lines to /etc/rc.conf:

```
vboxguest_enable="YES"
vboxservice_enable="YES"
```

3. If ntpd(8) or ntpdate(8) is used, disable host time synchronization:

```
vboxservice_flags="--disable-timesync"
```

4. Xorg will automatically recognize the vboxvideo driver. It can also be manually entered in /etc/X11/xorg.conf:

```
Section "Device"
    Identifier "Card0"
    Driver "vboxvideo"
    VendorName "InnoTek Systemberatung GmbH"
    BoardName "VirtualBox Graphics Adapter"
EndSection
```

5. To use the vboxmouse driver, adjust the mouse section in /etc/X11/xorg.conf:

```
Section "InputDevice"
    Identifier "Mouse0"
    Driver "vboxmouse"
EndSection
```

Reboot the server

```
sudo reboot
```