# DAOS Software Installation

DAOS runs on both Intel 64 and ARM64 platforms and has been successfully tested on CentOS7, OpenSUSE 42.2 and Ubuntu 18.04 distributions.

## Software Dependencies

DAOS requires a C99-capable compiler, a golang compiler, and the scons build tool. Moreover, the DAOS stack leverages the following open source projects:

* [CaRT](https://github.com/daos-stack/cart) for rank-based transport services that rely on both [Mercury](https://mercury-hpc.github.io/) and [Libfabric](https://ofiwg.github.io/libfabric/) for lightweight network transport and [PMIx](https://github.com/pmix/master) for process set management. See the CaRT repository for more information on how to build the CaRT library.
* [PMDK](https://github.com/pmem/pmdk.git) for persistent memory programming.
* [SPDK](http://spdk.io/) for userspace NVMe device access and management.
* [FIO](https://github.com/axboe/fio) for flexible testing of Linux I/O subsystems, specifically enabling validation of userspace NVMe device performance through fio-spdk plugin.
* [ISA-L](https://github.com/01org/isa-l) for checksum and erasure code computation.
* [Argobots](https://github.com/pmodels/argobots) for thread management.

The DAOS build system can be configured to download and build any missing dependencies automatically.

## Distribution Packages

DAOS RPM packaging is under development and will be available for DAOS v1.0. Integration with the [Spack](https://spack.io/) package manager is also under consideration.

## DAOS Source Code

To check out the DAOS source code, run the following command:

git clone https://github.com/daos-stack/daos.git

This command clones the DAOS git repository (path referred as ${daospath} below). Then initialize the submodules with:

cd ${daospath}

git submodule init

git submodule update

## Building DAOS from Scratch

The below instructions have been verified with CentOS. Installations on other Linux distributions might be similar with some variations. Developers of DAOS may want to check additional sections below before beginning for suggestions related specifically to development. Please contact us in our [forum](https://daos.groups.io/g/daos) if running into issues.

### Build Prerequisites

Please install the following software packages (or equivalent for other distros):

**On CentOS and OpenSUSE:**

yum install -y epel-release

yum install -y git gcc gcc-c++ make cmake golang libtool scons boost-devel

yum install -y libuuid-devel openssl-devel libevent-devel libtool-ltdl-devel

yum install -y librdmacm-devel libcmocka libcmocka-devel readline-devel

yum install -y doxygen pandoc flex patch nasm yasm

yum install -y ninja-build meson libyaml-devel

# Required SPDK packages for managing NVMe SSDs

yum install -y CUnit-devel libaio-devel astyle-devel python-pep8 lcov

yum install -y python clang-analyzer sg3\_utils libiscsi-devel

yum install -y libibverbs-devel numactl-devel doxygen mscgen graphviz

# Required IpmCtl packages for managing SCM Modules

yum install -y yum-plugin-copr epel-release

yum copr -y enable jhli/ipmctl

yum copr -y enable jhli/safeclib

yum install -y libipmctl-devel

**On Ubuntu and Debian:**

apt-get install -y git gcc golang make cmake libtool-bin scons autoconf

apt-get install -y libboost-dev uuid-dev libssl-dev libevent-dev libltdl-dev

apt-get install -y librdmacm-dev libcmocka0 libcmocka-dev libreadline6-dev

apt-get install -y curl doxygen pandoc flex patch nasm yasm

apt-get install -y ninja-build meson libyaml-dev python2.7-dev

# Required SPDK packages for managing NVMe SSDs

apt-get install -y libibverbs-dev librdmacm-dev libcunit1-dev graphviz

apt-get install -y libaio-dev sg3-utils libiscsi-dev doxygen mscgen libnuma-dev

# Required IpmCtl packages for managing SCM Modules

apt-get install -y software-properties-common

add-apt-repository ppa:jhli/libsafec

add-apt-repository ppa:jhli/ipmctl

apt-get update

apt-get install -y libipmctl-dev

Verify that all the auto tools listed below are at the appropriate versions:

* m4 (GNU M4) 1.4.16
* flex 2.5.37
* autoconf (GNU Autoconf) 2.69
* automake (GNU automake) 1.13.4
* libtool (GNU libtool) 2.4.2

### Protobuf Compiler

The DAOS control plane infrastructure will be using protobuf as the data serialization format for its RPC requests. The DAOS proto files use protobuf 3 syntax which is not supported by the platform protobuf compiler in all cases. Not all developers will need to build the proto files into the various source files. However, if changes are made to the proto files, they will need to be regenerated with a protobuf 3.\* or higher compiler. To set up support for compiling protobuf files, download the following precompiled package for Linux and install it somewhere accessible by your PATH variable.

https://github.com/google/protobuf/releases/download/v3.5.1/protoc-3.5.1-linux-x86\_64.zip

### Building DAOS & Dependencies

If all the software dependencies listed previously are already satisfied, then type the following command in the top source directory to build the DAOS stack:

scons --config=force install

If you are a developer of DAOS, we recommend following the instructions in Section 4.4.4 below.

Otherwise, the missing dependencies can be built automatically by invoking scons with the following parameters:

scons --config=force --build-deps=yes USE\_INSTALLED=all install

By default, DAOS and its dependencies are installed under ${daospath}/install. The installation path can be modified by adding the PREFIX= option to the above command line (e.g., PREFIX=/usr/local).

### Environment setup

Once built, the environment must be modified to search for binaries and header files in the installation path. This step is not required if standard locations (e.g. /bin, /sbin, /usr/lib, ...) are used.

CPATH=${daospath}/install/include/:$CPATH

PATH=${daospath}/install/bin/:${daospath}/install/sbin:$PATH

export CPATH PATH

If using bash, PATH can be set up for you after a build by sourcing the script scons\_local/utils/setup\_local.sh from the daos root. This script utilizes a file generated by the build to determine the location of daos and its dependencies.

If required, ${daospath}/install must be replaced with the alternative path specified through PREFIX. The network type to use as well the debug log location can be selected as follows:

export CRT\_PHY\_ADDR\_STR="ofi+sockets",

OFI\_INTERFACE=eth0, where eth0 is the network device you want to use.

For infiniband you could use ib0 or whichever label points to IB device.