

See the [Pre-requisites](#) and the [Setting up the flake registry](#) guide before getting started.

IOCs

For an introduction into IOC development, read [Creating a StreamDevice IOC](#).

IOC creation

<code>nix</code> flake new -t epnix myTop	Create an EPNix EPICS top
<code>makeBaseApp.pl</code> -t ioc name	Initialize an EPICS app in an EPICS top
<code>makeBaseApp.pl</code> -a linux-x86_64 -i -t ioc -p name Name	Initialize an <code>iocBoot</code> folder in an EPICS top

IOC development

<code>propagatedBuildInputs = [epnix.support.StreamDevice];</code>	In <code>ioc.nix</code> , add an EPICS support module to the build environment
<code>nativeBuildInputs = [myLib];</code> <code>buildInputs = [myLib];</code>	In <code>ioc.nix</code> , add a native library to the build environment
<code>myApp_DBD = stream.dbd</code> <code>myApp_LIBS = stream</code>	In <code>myApp/src/Makefile</code> , add an EPICS support module to an EPICS app

Important files

See [Template files](#) for more detailed information.

<code>flake.nix</code>	Nix project file
<code>ioc.nix</code>	Defines the EPICS top build
<code>checks/simple.nix</code>	Defines the <code>simple</code> integration check

IOC building

<code>nix</code> build -L	Build the IOC, showing compilation logs
<code>nix</code> develop	Enter the development shell
<code>epicsConfigurePhase</code>	<i>In the development shell</i> , configure the EPICS build
<code>make</code>	<i>In the development shell</i> , manually build the EPICS top

Flake input overrides

<code>nix</code> build -L \ --override-input \ supportModule \ /path/to/supportModule	Build the IOC, but with a custom version of a support module
<code>nix</code> develop \ --override-input \ supportModule \ /path/to/supportModule	Enter the development shell, but with a custom version of a support module

IOC testing

<code>nix</code> flake check -L	Run IOC checks
Adding integration tests to your IOC	IOC testing tutorial documentation
Testing	IOC testing guides

IOC inspection

For IOCs deployed on NixOS system by using the [IOC services](#) options.

<code>systemctl</code> status myIoc	Check whether an IOC is running
<code>systemctl</code> restart myIoc	Restart an IOC
<code>systemctl</code> stop myIoc	Stop an IOC
<code>journalctl</code> -xefu myIoc	Follow the logs of an IOC
<code>telnet-myIoc</code>	Connect to the command-line of an IOC

NixOS services

For an introduction into how to deploy EPICS-related services, read the [Creating an Archiver Appliance instance](#) tutorial.

Applying NixOS changes

<code>nixos-rebuild</code> test	Apply changes now, but revert them on reboot
<code>nixos-rebuild</code> switch	Apply changes now, and keep them on reboot
<code>nixos-rebuild</code> boot	Apply changes for the next reboot