



# Nexus Application Development - ISO

---

This chapter contains the following sections:

- [About ISO, on page 1](#)
- [Installing the ISO, on page 1](#)
- [Using the ISO to Build Applications, on page 2](#)
- [Using RPM to Package an Application, on page 3](#)

## About ISO

The ISO image is a bootable Wind River 5 environment that includes the necessary tools, libraries, and headers to build and RPM-package third-party applications to run natively on a switch.

The content is not exhaustive, and it might be required that the user download and build any dependencies needed for any particular application.



---

**Note** Some applications are ready to be downloaded and used from the Cisco devhub website and do not require building.

---

## Installing the ISO

The ISO image is available for download at: [http://devhub.cisco.com/artifactory/simple/open-nxos/7.0-3-12-1/x86\\_64/satori-vm-intel-xeon-core.iso](http://devhub.cisco.com/artifactory/simple/open-nxos/7.0-3-12-1/x86_64/satori-vm-intel-xeon-core.iso).

The ISO is intended to be installed as a virtual machine. Use instructions from your virtualization vendor to install the ISO.

---

**Step 1** (Optional) VMware-based installation.

The ISO image installation on a VMWare virtual machine requires the virtual disk to be configured as SATA and not SCSI.

**Step 2** (Optional) QEMU-based installation.

Enter the following commands:

```
bash$ qemu-img create satori.img 10G
bash$ qemu-system-x86_64 -cdrom ./satori-vm-intel-xeon-core.iso -hda ./satori.img -m 8192
```

Once the ISO starts to boot, a menu is displayed. Choose the **Graphics Console Install** option. This installs to the virtual HD. Once the install is complete, the virtual machine must be rebooted.

---

### What to do next

To login to the system, enter **root** as the login and **root** as the password.

Using the ISO to Build Applications Most of the build procedures that work with the SDK, and Linux in general, also apply to the ISO environment. However, there is no shell environment script to run. The default paths should be fine to use the tools installed. The source code for applications needs to be obtained through the usual mechanisms such as a source tar file or git repository.

Build the source code:

```
bash$ tar --xvzf example-lib.tgz
bash$ mkdir example-lib-install
bash$ cd example-lib/
bash$ ./configure --prefix=/path/to/example_lib_install
bash$ make

bash$ make install
```

## Using the ISO to Build Applications

Most of the build procedures that work with the SDK, and Linux in general, also apply to the ISO environment. However, there is no shell environment script to run. The default paths should be fine to use the tools installed. The source code for applications needs to be obtained through the usual mechanisms such as a source tar file or git repository.

Build the source code.

- a) **tar -xvzf example-lib.tgz**
- b) **mkdir example-lib-install**
- c) **cd example-lib/**
- d) **./configure -prefix=***path\_to\_example-lib-install*
- e) **make**
- f) **make install**

The steps are normal Linux.

### Example:

The following example shows how to build the source code:

```
bash$ tar -xvzf example-lib.tgz
bash$ mkdir example-lib-install
bash$ cd example-lib/
bash$ ./configure -prefix=<path_to_example-lib-install>
bash$ make
```

```
bach$ make install
```

---

## Using RPM to Package an Application

If the application successfully builds using "make", then it can be packaged into an RPM.



---

**Note** RPM and spec files

The RPM package format is designed to package up all files (binaries, libraries, configurations, documents, etc) that are needed for a complete install of the given application. The process of creating an RPM file is therefore somewhat non-trivial. To aid in the RPM build process, a .spec file is used that controls everything about the build process.

---



---

**Note** Many third-party applications are available on the internet in the form of source code packaged into tarballs. In many cases, these tarballs will include a .spec file to help with RPM build process. Unfortunately, many of these .spec files are not updated as frequently as the source code itself. Even worse, sometimes there is no spec file at all. In these cases the spec file may need editing or even creating from scratch so that RPMs can be built.

