**JXcore,** which is an open source project, introduces a unique feature for packaging and encryption of source files and other assets into JX packages.

Consider you have a large project consisting of many files. JXcore can pack them all into a single file to simplify the distribution. This chapter provides a quick overview of the whole process starting from installing JXcore.

JXcore Installation

Installing JXcore is quite simple. Here we have provided step-by-step instructions on how to install JXcore on your system. Follow the steps given below −

Step 1

Download the JXcore package from <https://github.com/jxcore/jxcore>, as per your operating system and machine architecture. We downloaded a package for Cenots running on 64-bit machine.

$ wget https://s3.amazonaws.com/nodejx/jx\_rh64.zip

Step 2

Unpack the downloaded file **jx\_rh64.zip**and copy the jx binary into /usr/bin or may be in any other directory based on your system setup.

$ unzip jx\_rh64.zip

$ cp jx\_rh64/jx /usr/bin

Step 3

Set your PATH variable appropriately to run jx from anywhere you like.

$ export PATH=$PATH:/usr/bin

Step 4

You can verify your installation by issuing a simple command as shown below. You should find it working and printing its version number as follows −

$ jx --version

v0.10.32

Packaging the Code

Consider you have a project with the following directories where you kept all your files including Node.js, main file, index.js, and all the modules installed locally.

drwxr-xr-x 2 root root 4096 Nov 13 12:42 images

-rwxr-xr-x 1 root root 30457 Mar 6 12:19 index.htm

-rwxr-xr-x 1 root root 30452 Mar 1 12:54 index.js

drwxr-xr-x 23 root root 4096 Jan 15 03:48 node\_modules

drwxr-xr-x 2 root root 4096 Mar 21 06:10 scripts

drwxr-xr-x 2 root root 4096 Feb 15 11:56 style

To package the above project, you simply need to go inside this directory and issue the following jx command. Assuming index.js is the entry file for your Node.js project −

$ jx package index.js index

Here you could have used any other package name instead of **index.** We have used **index** because we wanted to keep our main file name as index.jx. However, the above command will pack everything and will create the following two files −

* **index.jxp** This is an intermediate file which contains the complete project detail needed to compile the project.
* **index.jx** This is the binary file having the complete package that is ready to be shipped to your client or to your production environment.

Launching JX File

Consider your original Node.js project was running as follows −

$ node index.js command\_line\_arguments

After compiling your package using JXcore, it can be started as follows −

$ jx index.jx command\_line\_arguments