

# Hadoop Deployment Layout

## Table of contents

1 Introduction .....	2
2 Packages .....	2
3 Deployment .....	3
4 Path Configurations .....	4

## 1. Introduction

This document describes the standard deployment layout for Hadoop. With increased complexity and evolving Hadoop ecosystem, having standard deployment layout ensures better integration between Hadoop sub-projects. By making the installation process easier, we can lower the barrier to entry and increase Hadoop adoption.

## 2. Packages

We need to divide Hadoop up into packages that can be independently upgraded. The list of packages should include:

- Hadoop Common - Common including the native code and required jar files.
- HDFS Client - HDFS jars, scripts, and shared libraries.
- HDFS Server - jsvc executable
- Yarn Client - Yarn client jars and scripts
- Yarn Server - Yarn server jars and scripts
- MapReduce - MapReduce jars, scripts, and shared libraries
- LZ0 - LZ0 codec from [github.com/omally/hadoop-gpl-compression](https://github.com/omally/hadoop-gpl-compression)
- Metrics - Plugins for Chukwa and Ganglia

Packages from other teams will include:

- Pig
- Hive
- Oozie client
- Oozie server
- Howl client
- Howl server

These packages should be deployable with RPM on RedHat. We also need a package that depends on a version of each of these packages. In general, we can generate tarballs in the new deployment layout.

Note that some packages, like Pig, which are user facing, will have 2 versions installed in a given deployment. This will be accomplished by modifying the package name and the associated binaries to include the version number.

All of the following paths are based on a prefix directory that is the root of the installation. Our packages must support having multiple Hadoop stack installation on a computer at the same time. For RPMs, this means that the packages must be relocatable and honor the `--prefix` option.

### 3. Deployment

It is important to have a standard deployment that results from installing the packages regardless of the package manager. Here are the top level directories and a sample of what would be under each. Note that all of the packages are installed "flattened" into the prefix directory. For compatibility reasons, we should create "share/hadoop" that matches the old HADOOP\_HOME and set the HADOOP\_HOME variable to that.

```
$PREFIX/ bin / hadoop
          |   | mapred
          |   | pig -> pig7
          |   | pig6
          +   + pig7
          |
          + etc / hadoop / core-site.xml
          |           | hdfs-site.xml
          |           + mapred-site.xml
          |
          + include / hadoop / Pipes.hh
          |           |   + TemplateFactory.hh
          |           + hdfs.h
          |
          + lib / jni / hadoop-common / libhadoop.so.0.20.0
          |           | libhdfs.so -> libhdfs.so.0.20.0
          |           + libhdfs.so.0.20.0
          |
          + libexec / task-controller
          |
          + man / man1 / hadoop.1
          |           | mapred.1
          |           | pig6.1
          |           + pig7.1
          |
          + share / hadoop-common
          |         | hadoop-hdfs
          |         | hadoop-mapreduce
          |         | pig6
          |         + pig7
          |
          + sbin / hdfs-admin
          |       | mapred-admin
          |
          + src / hadoop-common
          |       | hadoop-hdfs
          |       + hadoop-mapreduce
          |
          + var / lib / data-node
          |           + task-tracker
```

```

| log / hadoop-datanode
|   + hadoop-tasktracker
+ run / hadoop-datanode.pid
|   + hadoop-tasktracker.pid

```

Note that we must continue to honor `HADOOP_CONF_DIR` to override the configuration location, but that it should default to `$prefix/etc`. User facing binaries and scripts go into `bin`. Configuration files go into `etc` with multiple configuration files having a directory. JNI shared libraries go into `lib/jni/$tool` since Java does not allow to specify the version of the library to load. Libraries that aren't loaded via `System.loadLibrary` are placed directly under `lib`. 64 bit versions of the libraries for platforms that support them should be placed in `lib64`. All of the architecture-independent pieces, including the jars for each tool will be placed in `share/$tool`. The default location for all the run time information will be in `var`. The storage will be in `var/lib`, the logs in `var/log` and the pid files in `var/run`.

## 4. Path Configurations

Path can be configured at compile phase or installation phase. For RPM, it takes advantage of the `--relocate` directive to allow path reconfiguration at install phase. For Debian package, path is configured at compile phase.

Build phase parameter:

- `package.prefix` - Location of package prefix (Default `/usr`)
- `package.conf.dir` - Location of configuration directory (Default `/etc/hadoop`)
- `package.log.dir` - Location of log directory (Default `/var/log/hadoop`)
- `package.pid.dir` - Location of pid directory (Default `/var/run/hadoop`)

Install phase parameter:

```

rpm -i hadoop-[version]-[rev].[arch].rpm \
    --relocate /usr=/usr/local/hadoop \
    --relocate /etc/hadoop=/usr/local/etc/hadoop \
    --relocate /var/log/hadoop=/opt/logs/hadoop \
    --relocate /var/run/hadoop=/opt/run/hadoop

```