

LibHDFS

libhdfs

libhdfs is a [JNI](#) based C api for Hadoop's DFS.

It provides a simple subset of C apis to manipulate DFS files and the filesystem. libhdfs is available for [download](#) as a part of Hadoop itself. The source for libhdfs is available for browsing [here](#).

Table Of Contents

1. [Overview](#)
2. [Setup](#)
3. [APIs](#)
 - a. [FileSystem Manipulation](#)
 - b. [File Manipulation](#)
 - c. [Support for Threads](#)
4. [Examples](#)
5. [Assistance/Bugs](#)
6. [Conclusion](#)

Overview

[Here](#) is an overview of Hadoop's DFS. The javadocs for DFS are available [here](#) and [here](#). **libhdfs** is a simple JNI based C api for accessing and manipulating Hadoop's DFS from native code. It offers a simple subset of the same functionality.

Setup

It is necessary to setup Hadoop's DFS itself first. The information to setup Hadoop is available [here](#). Once you have a working setup, you will need to get into the **src/c++/libhdfs** directory and use the Makefile to build libhdfs (in case of issues use [this](#)). Once you have successfully built libhdfs you can link it into your programs and are good to go.

APIs

This section describes the various apis provided by **libhdfs** to manipulate the DFS. It is classified into apis which manipulate individual files and those which manipulate the filesystem itself. (Please see the doxygen documentation [[# here](#)] for details of individual apis.)

FileSystem APIs

libhdfs provides apis for both **generic** manipulation of the filesystem (create directories, copy/move files etc.) and also some very DFS specific functionality (get information on file replication etc.).

At startup one should use the *hdfsConnect* api to connect to the DFS before any operations can be performed (on files or the filesystem); the analogous *hdfsDisconnect* performs a clean teardown of the connection.

Generic operations:

- *hdfsCopy* (across filesystems also)
- *hdfsMove* (across filesystems also)
- *hdfsRename*
- *hdfsDelete*

libhdfs also provides apis for manipulating directories on the DFS:

- *hdfsCreateDirectory*
- *hdfsSetWorkingDirectory*
- *hdfsGetWorkingDirectory*
- *hdfsListDirectory* / *hdfsGetPathInfo* / *hdfsFreeFileInfo*

The apis to query the filesystems for various properties:

- *hdfsGetHosts*
- *hdfsGetDefaultBlockSize*
- *hdfsGetUsed* / *hdfsGetCapacity*

File APIs

libhdfs provides posix-like apis to manipulate individual files (create, read/write, query etc.) listed below:

- *hdfsOpenFile* / *hdfsCloseFile*
- *hdfsRead* / *hdfsWrite*
- *hdfsTell* / *hdfsSeek*
- *hdfsFlush*
- *hdfsAvailable*

Using libhdfs in Threaded Applications

libhdfs can be used in threaded applications using the Posix Threads. However to carefully interact with JNI's global/local references the user has to explicitly call the *hdfsConvertToGlobalRef* / *hdfsDeleteGlobalRef* apis.

Examples

The [test cases](#) for libhdfs provide some good examples on how to use libhdfs.

Contact Information

Please drop us an email at users@hadoop.apache.org if you have any questions or any suggestions. Use [Jira](#) (component: hdfs) to report bugs.

Conclusion

Thank you for your interest in Hadoop and libhdfs!