**Module 1, Exercise 1:** This section focuses on developing the Java code and making use of HDFS Java API.

**Note: Please find the java files required for this exercise in this path: /home/mapr/Desktop/mapr\_training/java\_files**

1. Copy the file from the local file system to HDFS

2. Copy the file from HDFS to the local file system.

**Solution 1:**

import java.io.IOException;

import java.io.InputStream;

import java.io.OutputStream;

import java.net.URI;

import java.net.URISyntaxException;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.FileStatus;

import org.apache.hadoop.fs.FileSystem;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IOUtils;

import org.apache.hadoop.util.Progressable;

public class CopyToHDFS {

public static void main(String[] args) throws IOException, URISyntaxException {

Configuration conf = new Configuration();

conf.addResource(new Path("/opt/mapr/hadoop/hadoop-2.4.1/etc/hadoop/core-site.xml"));// adding the necessary configuration values

conf.addResource(new Path("/opt/mapr/hadoop/hadoop-2.4.1/etc/hadoop/hdfs-site.xml"));// adding the necessary configuration values

FileSystem hdfs = FileSystem.get(conf);

FileSystem local = FileSystem.get(new URI("file:///"), new Configuration());

Path localGlob = new Path("/home/mapr/java\_inp/inp.txt");// source directory

Path hdfsRoot = new Path("/user/mapr/java\_test");//Target HDFS directoy

hdfs.mkdirs(hdfsRoot);

FileStatus [] files = local.globStatus(localGlob);

for (FileStatus file : files ){

Path from = file.getPath();

Path to = new Path(hdfsRoot, file.getPath().getName());

copy(local, from, hdfs, to);//copying from source to target

}

}

// The “copy” helper function

private static void copy(FileSystem fromFs, Path fromPath, FileSystem toFs, Path toPath) throws IOException {

System.out.println("Copying [" + fromPath + "] to [" + toPath + "]");

OutputStream out = null;

InputStream in = null;

try {

in = fromFs.open(fromPath);

out = toFs.create(toPath, new Progressable() {

@Override

public void progress() {

System.out.print("..");

}

});

IOUtils.copyBytes(in, out, 10, false);

} finally {

IOUtils.closeStream(out);

IOUtils.closeStream(in);

}

System.out.print("\n");

}

}

1. Navigate to the folder where CopyToHDFS.java file exists. To Compile the program : **javac –cp $(Hadoop classpath) CopyToHDFS.java**
2. To Run the program :  **java –cp .:$(hadoop classpath) CopyToHDFS**

**Solution 2:**

import java.io.IOException;

import java.io.InputStream;

import java.io.OutputStream;

import java.net.URI;

import java.net.URISyntaxException;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.FileStatus;

import org.apache.hadoop.fs.FileSystem;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IOUtils;

public class CopyFromHDFS1 {

public static void main(String[] args) throws IOException, URISyntaxException {

Path glob = new Path("/user/mapr/java\_test/inp2.txt");//source directory on HDFS

String localRoot = "/home/mapr/java\_inp/"; // target local directory

Configuration conf = new Configuration();

conf.addResource(new Path("/opt/mapr/hadoop/hadoop-2.4.1/etc/hadoop/core-site.xml"));// adding the necessary configuration values

conf.addResource(new Path("/opt/mapr/hadoop/hadoop-2.4.1/etc/hadoop/hdfs-site.xml"));// adding the necessary configuration values

FileSystem hdfs = FileSystem.get(conf);

FileSystem localFs = FileSystem.get(new URI("file:///"), new Configuration());

FileStatus [] files = hdfs.globStatus(glob);

for (FileStatus file : files ){// copy each file

copyToLocal(hdfs, localFs, file.getPath(),new Path(localRoot + file.getPath().getName()));

}

}

// Helper CopyToLocal function

private static void copyToLocal(FileSystem fromFs, FileSystem toFs, Path fromPath, Path toPath) throws IOException {

System.out.println("Copying [" + fromPath + "] to [" + toPath + "]");

OutputStream out = null;

InputStream in = null;

try {

in = fromFs.open(fromPath);

out = toFs.create(toPath);

IOUtils.copyBytes(in, out, 10, false);

} finally {

IOUtils.closeStream(out);

IOUtils.closeStream(in);

}

}

}

1. Navigate to the folder where CopyFromHDFS1.java exists. To Compile the program : **javac –cp $(Hadoop classpath) CopyFromHDFS1.java**
2. To Run the program :  **java –cp .:$(hadoop classpath) CopyFromHDFS1**

**Module 1, Exercise 1 Using Eclipse:** This section focuses on developing the Java code and making use of HDFS Java API using Eclipse.

1. Copy the file from the local file system to HDFS

2. Copy the file from HDFS to the local file system.

To execute the exercise in eclipse:

1. Type the command : eclipse.
2. Create a new project.
   1. Click **File** -> **New** -> **Java Project**.
   2. Enter the Project name
   3. Click **Finish**.

To add the jars before executing:

1. Right click **src** -> **Build Path** -> **Configure Build Path** -> **Libraries** -> **Add external Jars** -> **Navigate**.
2. Add the jars under /opt/mapr/hadoop/hadoop-2.4.1/etc/hadoop:/opt/mapr/hadoop/hadoop-2.4.1/share/hadoop/common/lib/\*:/opt/mapr/hadoop/hadoop-2.4.1/share/hadoop/common/\*:/opt/mapr/hadoop/hadoop-2.4.1/share/hadoop/hdfs:/opt/mapr/hadoop/hadoop-2.4.1/share/hadoop/hdfs/lib/\*:/opt/mapr/hadoop/hadoop-2.4.1/share/hadoop/hdfs/\*:/opt/mapr/hadoop/hadoop-2.4.1/share/hadoop/yarn/lib/\*:/opt/mapr/hadoop/hadoop-2.4.1/share/hadoop/yarn/\*:/opt/mapr/hadoop/hadoop-2.4.1/share/hadoop/mapreduce/lib/\*:/opt/mapr/hadoop/hadoop-2.4.1/share/hadoop/mapreduce/\*
3. Click **OK**.

Create a class name.

* 1. On the **Projects** tab, click on the arrow next to the name of the project -> Right click  **-> New** -> **Class**.
  2. Enter the class name
  3. Click **Finish**.
  4. Enter the below code in the class file

Note: Class Name and the program main class should be same.

1. **Solution**

import java.io.IOException;

import java.io.InputStream;

import java.io.OutputStream;

import java.net.URI;

import java.net.URISyntaxException;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.FileStatus;

import org.apache.hadoop.fs.FileSystem;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IOUtils;

import org.apache.hadoop.util.Progressable;

public class CopyToHDFS {

public static void main(String[] args) throws IOException, URISyntaxException {

Configuration conf = new Configuration();

conf.addResource(new Path("/opt/mapr/hadoop/hadoop-2.4.1/etc/hadoop/core-site.xml"));// adding the necessary configuration values

conf.addResource(new Path("/opt/mapr/hadoop/hadoop-2.4.1/etc/hadoop/hdfs-site.xml"));// adding the necessary configuration values

FileSystem hdfs = FileSystem.get(conf);

FileSystem local = FileSystem.get(new URI("file:///"), new Configuration());

Path localGlob = new Path("/home/mapr/java\_inp/inp.txt");// source directory

Path hdfsRoot = new Path("/user/mapr/java\_test");//Target HDFS directoy

hdfs.mkdirs(hdfsRoot);

FileStatus [] files = local.globStatus(localGlob);

for (FileStatus file : files ){

Path from = file.getPath();

Path to = new Path(hdfsRoot, file.getPath().getName());

copy(local, from, hdfs, to);//copying from source to target

}

}

// The “copy” helper function

private static void copy(FileSystem fromFs, Path fromPath, FileSystem toFs, Path toPath) throws IOException {

System.out.println("Copying [" + fromPath + "] to [" + toPath + "]");

OutputStream out = null;

InputStream in = null;

try {

in = fromFs.open(fromPath);

out = toFs.create(toPath, new Progressable() {

@Override

public void progress() {

System.out.print("..");

}

});

IOUtils.copyBytes(in, out, 10, false);

} finally {

IOUtils.closeStream(out);

IOUtils.closeStream(in);

}

System.out.print("\n");

}

}

To run the program, click **Run** -> **Run** or press **Ctrl+F11**.

1. **Solution:**

**This can be achieved in two ways:**

**Approach 1:**

import java.io.IOException;

import java.io.InputStream;

import java.io.OutputStream;

import java.net.URI;

import java.net.URISyntaxException;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.FileStatus;

import org.apache.hadoop.fs.FileSystem;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IOUtils;

public class CopyFromHDFS1 {

public static void main(String[] args) throws IOException, URISyntaxException {

Path glob = new Path("/user/mapr/java\_test/inp2.txt");//source directory on HDFS

String localRoot = "/home/mapr/java\_inp/"; // target local directory

Configuration conf = new Configuration();

conf.addResource(new Path("/opt/mapr/hadoop/hadoop-2.4.1/etc/hadoop/core-site.xml"));// adding the necessary configuration values

conf.addResource(new Path("/opt/mapr/hadoop/hadoop-2.4.1/etc/hadoop/hdfs-site.xml"));// adding the necessary configuration values

FileSystem hdfs = FileSystem.get(conf);

FileSystem localFs = FileSystem.get(new URI("file:///"), new Configuration());

FileStatus [] files = hdfs.globStatus(glob);

for (FileStatus file : files ){// copy each file

copyToLocal(hdfs, localFs, file.getPath(),new Path(localRoot + file.getPath().getName()));

}

}

// Helper CopyToLocal function

private static void copyToLocal(FileSystem fromFs, FileSystem toFs, Path fromPath, Path toPath) throws IOException {

System.out.println("Copying [" + fromPath + "] to [" + toPath + "]");

OutputStream out = null;

InputStream in = null;

try {

in = fromFs.open(fromPath);

out = toFs.create(toPath);

IOUtils.copyBytes(in, out, 10, false);

} finally {

IOUtils.closeStream(out);

IOUtils.closeStream(in);

}

}

}

**Approach 2:**

import java.io.IOException;

import java.net.URISyntaxException;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.FileStatus;

import org.apache.hadoop.fs.FileSystem;

import org.apache.hadoop.fs.Path;

public class CopyFromHDFS2 {

public static void main(String[] args) throws IOException, URISyntaxException {

Path glob = new Path("/user/mapr/java\_test/inp2.txt");//source directory on HDFS

String localRoot = "/home/mapr/java\_inp”; // target local directory

Configuration conf = new Configuration();

conf.addResource(new Path("/opt/mapr/hadoop/hadoop-2.4.1/etc/hadoop/core-site.xml "));

conf.addResource(new Path("/opt/mapr/hadoop/hadoop-2.4.1/etc/hadoop/hdfs-site.xml"));

FileSystem hdfs = FileSystem.get(conf);

FileStatus [] files = hdfs.globStatus(glob);

for (FileStatus file : files ){

Path from = file.getPath();

Path to = new Path(localRoot, file.getPath().getName());

System.out.println("Copying hdfs file [" + from + "] to local [" + to + "]");

hdfs.copyToLocalFile(from, to);// no loop required as in 1st approach , we make use of the built in function

}

}

}

To run the program, click **Run** -> **Run** or press **Ctrl+F11**.