



# Pentaho MapReduce with MapR Client

# HITACHI

## Inspire the Next

Change log (if you want to use it):

Date	Version	Author	Changes

# Contents

Overview .....	1
Before You Begin.....	1
Use Case: (Title of Use Case) .....	1
Set Up Your Environment.....	2
Get MapR Server Information .....	2
Set Up Host Environment.....	2
Install and Configure MapR Client .....	4
Download MapR Client Tools .....	4
Set Up Environment Variables.....	4
Configure MapR Client to Connect into HDFS.....	5
Modify <code>core-site.xml</code> for MapR Client.....	5
Modify <code>mapred-site.xml</code> for MapR Client .....	7
Connect to HDFS Using MapR Client.....	7
Configure Hadoop Cluster Environment for PDI Jobs .....	9
Select Hadoop Distribution for MapR .....	9
Modify <code>config.properties</code> in Pentaho Shim Folder .....	9
Run PDI PMR from Samples .....	11
Related Information .....	12
Finalization Checklist.....	13

This page intentionally left blank.

# Overview

This document covers some best practices on setting up Pentaho Data Integration (PDI) to work with MapR. In it, you will learn how to set up and install the MapR client tool that is required by PDI to run Pentaho MapReduce (PMR) jobs.

The intention of this document is to speak about topics generally; however, these are the specific versions covered here:

Software	Version(s)
Pentaho	7.x, 8.0

The [Components Reference](#) in Pentaho Documentation has a complete list of supported software and hardware.

## Before You Begin

Before beginning, use the following information to prepare for the procedures described in the main section of the document.

This document assumes that you have knowledge of Pentaho and that you have already set up your basic environment and operating system.

### *Use Case: (Title of Use Case)*

---

*Fabiola has set up her Windows environment and needs to run MapReduce jobs. She knows that the only way to set this up is to run the MapReduce jobs on her MapR cluster.*

*Fabiola will install the MapR client and configure it for her environment to get this going.*

---

## Set Up Your Environment

In this document, the example we use involves a MapR Virtual Machine (VM), downloaded from [MapR Sandbox](#), running on Windows. This section covers the steps you need to take to set up your environment.

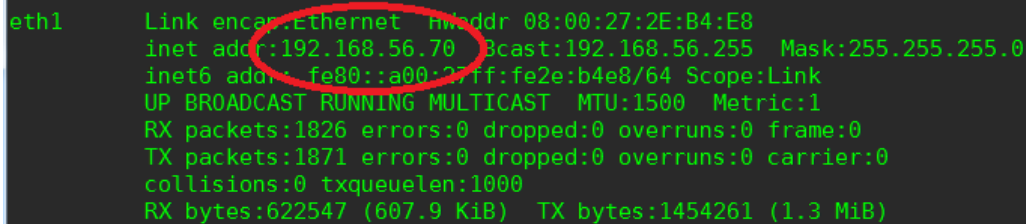
You can find details on these topics in the following sections:

- [Get MapR Server Information](#)
- [Set Up Host Environment](#)
- [Install Pentaho Shim](#)

## Get MapR Server Information

Once you have the MapR VM downloaded, and imported into VMWare or VirtualBox, then you must add a host-only adapter.

Once the VM starts, you will be able to find its IP address, which will vary based on your environment. This screenshot shows an example IP address.



```

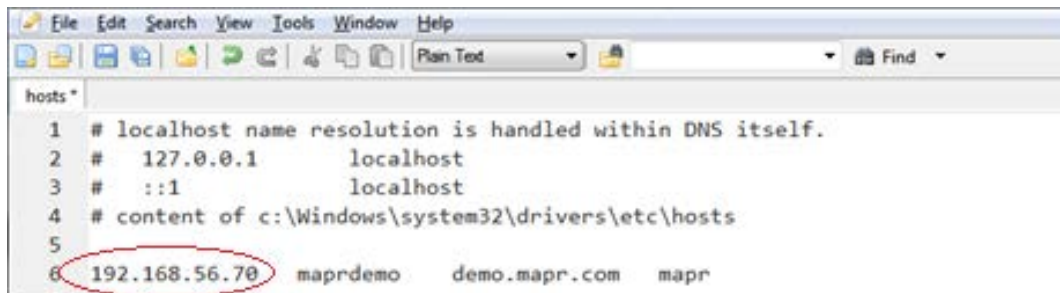
eth1    Link encap:Ethernet HWaddr 08:00:27:2E:B4:E8
        inet addr:192.168.56.70 Bcast:192.168.56.255 Mask:255.255.255.0
        inet6 addr: fe80::a00:27ff:fe2e:b4e8/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
        RX packets:1826 errors:0 dropped:0 overruns:0 frame:0
        TX packets:1871 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:622547 (607.9 KiB) TX bytes:1454261 (1.3 MiB)
  
```

Figure 1: Windows Development Environment

## Set Up Host Environment

Whether your development environment is Windows, macOS, or Unix, you will need to set it up to reflect the host environment's IP address, and connect to the MapR administration page.

1. Update your `hosts` file to reflect the host environment's IP address, which you found in the VM. Replace the IP address in Figure 2 with whatever the IP address is for your environment.



```

File Edit Search View Tools Window Help
Plan Text Find
hosts *
1 # localhost name resolution is handled within DNS itself.
2 # 127.0.0.1 localhost
3 # ::1 localhost
4 # content of c:\Windows\system32\drivers\etc\hosts
5
6 192.168.56.70 maprdemo demo.mapr.com mapr
  
```

Figure 2: Host Environment

2. Use your browser to connect to the MapR administration page using either of these URLs:
  - a. <http://demo.mapr.com:8443/mcs>
  - b. <http://localhost:8443/mcs>
3. Log in using the username/password combination of `root/mapr` or `mapr/mapr` to get to the main console.
4. At this point, make note of the cluster name, which in this example is `demo.mapr.com`. You will need the cluster name when you configure the MapR client.

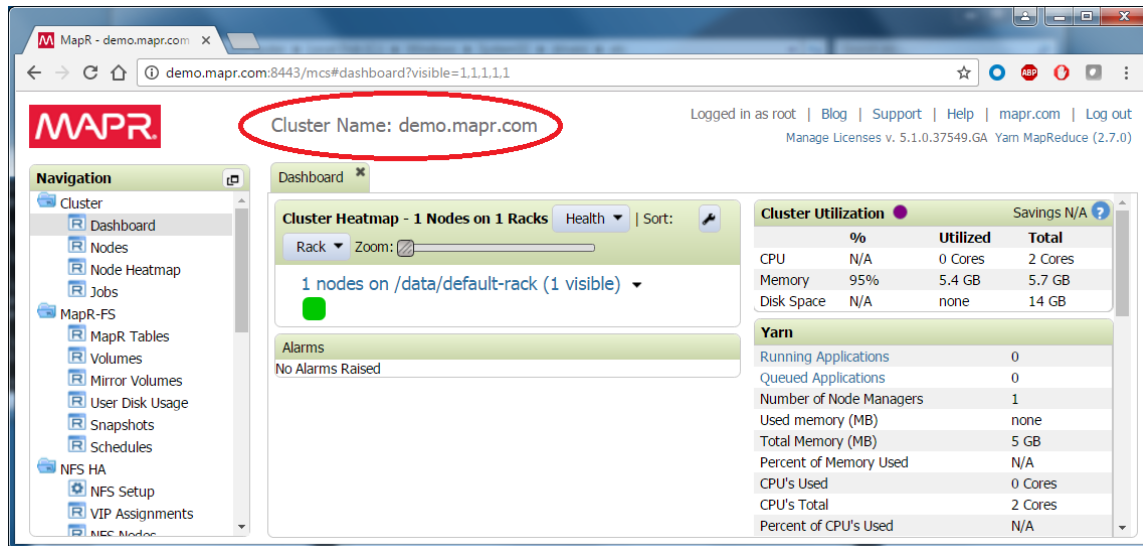


Figure 3: MapR Cluster

# Install and Configure MapR Client

This section shows you how to download and configure the MapR client tools, as well as how to set up your environment variables.

You can find details on these topics in the following sections:

- [Download MapR Client Tools](#)
- [Set Up Environment Variables](#)
- [Configure MapR Client to Connect to HDFS](#)

## Download MapR Client Tools

The [MapR Archive Index](#) has the latest MapR client tools version for various operating systems. The images in this demonstration are from an installation of MapR client tool version 5.1 onto Windows.

1. Download the correct version of the MapR client tools from the [MapR Archive Index](#).
2. Extract the downloaded client tool zip file into `c:\opt\mapr` or another appropriate location for your environment.
3. The folder structure of the client will show the following directories:

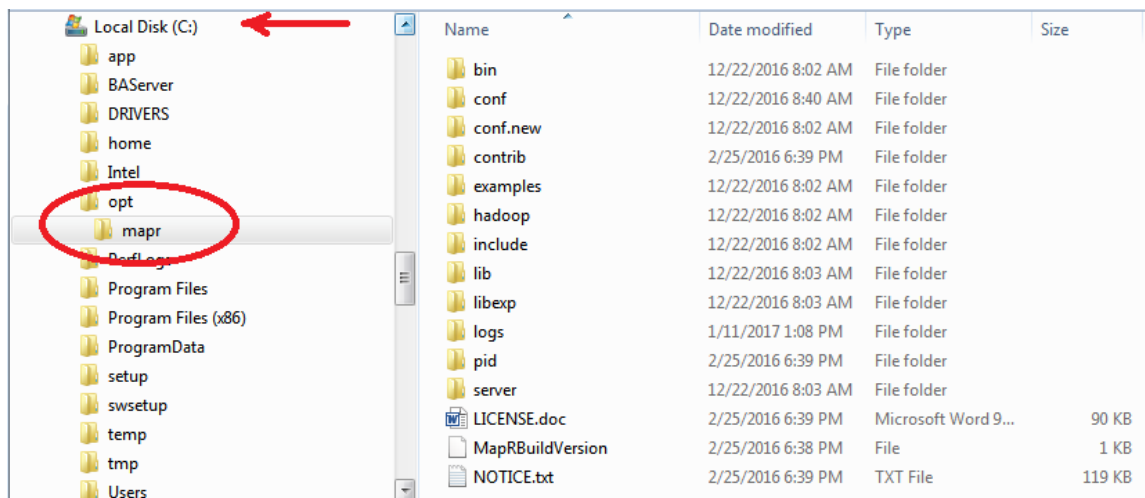


Figure 4: MapR Client Installation

## Set Up Environment Variables

The MapR client requires some configuration changes which will allow you to connect to the Hadoop file system using the client tools. MapR's documentation has specific information about the [ports used by MapR](#).

1. Open the command prompt.
2. Change to the directory to which you unzipped the client tools. In this example, it is `c:\opt\mapr\server`:

---

```
cd c:\opt\mapr\server
```

---



3. If you are using Windows, set `MAPR_HOME` globally to that same directory where you unzipped the client tools. For Unix, set `.bash_profile` or `.bash_rc` using `set-pentaho-env.sh`. For Windows, use `set-pentaho-env.bat` to set `MAPR_HOME`:

---

```
set MAPR_HOME=c:\opt\mapr
```

---

4. Run `configure.bat` with the parameters shown.
  - a. `-N` is the cluster name ([which you got earlier from the browser screen](#)).
  - b. `-C` is the node running Container Location Databases (CLDB) services and its port.
  - c. `-HS` is the history server name – hostname found in the `hosts` file.

---

```
configure.bat -N demo.mapr.com -c -C maprdemo:7222 -HS maprdemo
```

---

5. Set the `JAVA_HOME` variable if it is not already set in your environment.

## Configure MapR Client to Connect into HDFS

You will need to configure a couple of files in the MapR client in order to connect to the Hadoop Distributed File System (HDFS). Make sure you have the user ID for these files, and then follow these steps:

- [Modify `core-site.xml` for MapR Client](#)
- [Modify `mapred-site.xml` for MapR Client](#)
- [Connect to HDFS Using MapR Client](#)

### *Modify `core-site.xml` for MapR Client*

To modify `core-site.xml`:

1. Get the `userid (uid)` value in `hadoop.spoofer.user.uid` from the `/etc/passwd` file found in the MapR VM for the Hadoop user.
2. Get the `group ID (gid)` value in `hadoop.spoofer.user.gid` from the `/etc/group` file found in the MapR VM for the user who will be the Hadoop user.

- Use these commands to find the user ID and group ID for that user. In this example, they are the same (2000), but this may not always be the case:

```

root@maprdemo ~
[root@maprdemo ~]#
[root@maprdemo ~]# grep mapr /etc/passwd
maprdev:x:500:500::/home/maprdev:/bin/bash
mapr:x:2000:2000::/home/mapr:/bin/bash
[root@maprdemo ~]#
[root@maprdemo ~]# grep mapr /etc/group
wheel:x:10:maprdev,vagrant
maprdev:x:500:
mapr:x:2000:mapr
shadow:x:2001:mapr
[root@maprdemo ~]#
  
```

The terminal output shows the results of two commands. The first command, `grep mapr /etc/passwd`, lists the `maprdev` and `mapr` users. The `mapr` user has a UID of 2000 and a GID of 2000. The second command, `grep mapr /etc/group`, lists the `wheel`, `maprdev`, `mapr`, and `shadow` groups. The `mapr` group has a GID of 2000. Red circles and arrows highlight the UID and GID values for the `mapr` user and group.

Figure 5: MapR VM

- Once you have the group ID and user ID, put them into `core-site.xml` in `c:\opt\mapr\hadoop\hadoop-(version)\etc\hadoop` directory (or wherever it is found in your environment): in the correct spaces, indicated in red in this code block:

---

```

<property>
  <name>hbase.table.namespace.mappings</name>
  <value>*:/tables</value>
</property>
<property>
  <name>hadoop.proxyuser.mapr.hosts</name>
  <value>*</value>
</property>
<property>
  <name>hadoop.proxyuser.mapr.groups</name>
  <value>*</value>
</property>
<property>
  <name>hadoop.spoofed.user.uid</name>
  <value>2000</value>
</property>
<property>
  <name>hadoop.spoofed.user.gid</name>
  <value>2000</value>
</property>
  
```

```
<property>
  <name>hadoop.spoofed.user.username</name>
  <value>mapr</value>
</property>
```

---

### *Modify mapred-site.xml for MapR Client*

Add a cross-platform parameter to the `mapred-site.xml` to be able to run PDI in the Hadoop cluster. Detailed information on this process is available at [Set Up Pentaho to Connect to a MapR Cluster](#).

Add this property to the `mapred-site.xml` file in the `c:\opt\mapr\hadoop\hadoop-(version)\etc\hadoop` directory:

```
<property>
  <name>mapreduce.app-submission.cross-platform</name>
  <value>true</value>
</property>
```

---

### *Connect to HDFS Using MapR Client*

Once you have configured your environment with the minimum configurations in this document, you will be able to test the MapR client:

1. Open the command prompt and navigate to the `c:\opt\mapr\hadoop\hadoop-(version)\bin` directory:

```
cd c:\opt\mapr\hadoop\hadoop-(version)\bin
```

---

2. Make sure the `MAPR_HOME` environment variable is properly set:

```
hadoop fs -ls /
```

```
c:\opt\mapr\hadoop\hadoop-0.20.2\bin>hadoop fs -ls /
```

---

3. If it is set up incorrectly, you may see UID 2000:GID 2000 instead of `mapr root` as shown in Figure 7 in the ownership of HDFS's directories:

```
C:\opt\mapr\hadoop\hadoop-2.7.0\bin>hadoop fs -ls /  
Found 8 items  
drwxr-xr-x - mapr root 1 2016-03-16 13:49 /apps  
drwxr-xr-x - mapr root 0 2016-03-16 13:33 /hbase  
drwxrwxrwx - mapr root 2 2016-03-16 13:49 /oozie  
drwxr-xr-x - mapr root 1 2017-01-12 04:49 /opt  
drwxr-xr-x - root root 0 2016-03-16 13:45 /tables  
drwxrwxrwx - mapr root 0 2016-03-16 13:33 /tmp  
drwxr-xr-x - mapr root 7 2016-03-16 13:49 /user  
drwxr-xr-x - mapr root 1 2016-03-16 13:33 /var  
  
C:\opt\mapr\hadoop\hadoop-2.7.0\bin>
```

Figure 6: MapR for HDFS

# Configure Hadoop Cluster Environment for PDI Jobs

The last stage of the configuration is to verify that the libraries for the MapR client are loaded when PDI MapReduce is executed.

You can find details on these topics in the following sections:

- [Select Hadoop Distribution for MapR](#)
- [Modify `config.properties` in Pentaho Shim Folder](#)
- [Run PDI PMR from Samples](#)

## Select Hadoop Distribution for MapR

To select the correct Hadoop distribution, use PDI.

1. From the **Tools** menu of PDI, select **Hadoop Distribution....**
2. In the selection box, choose the correct MapR from the list (this example shows MapR5.1.0):

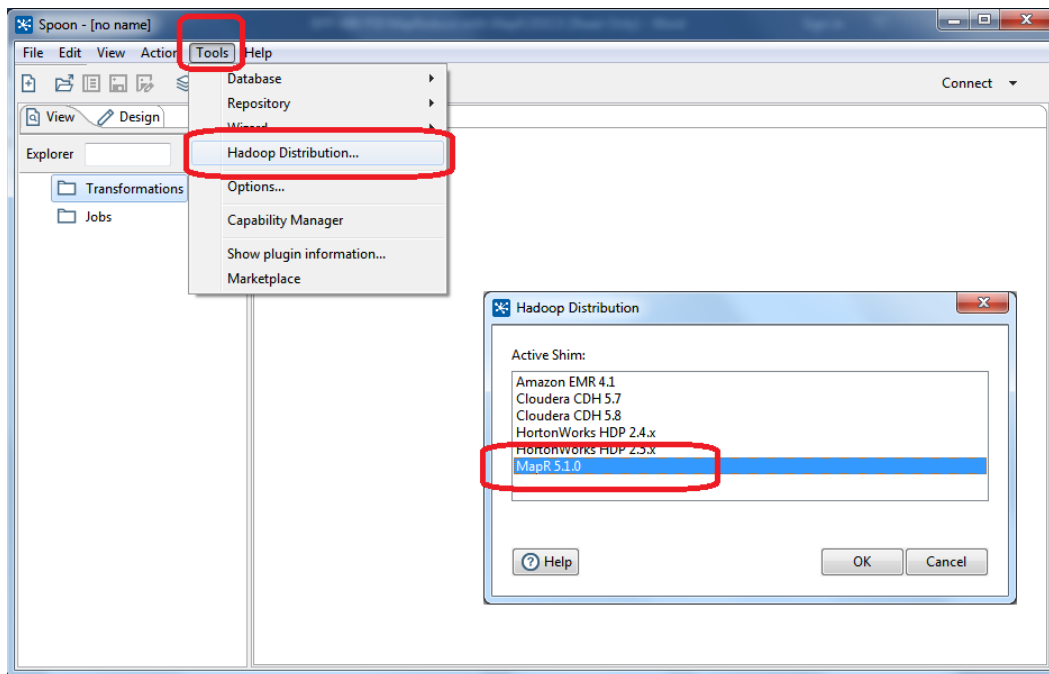


Figure 7: Hadoop Distribution for MapR

3. Restart the PDI client after you choose your Hadoop distribution.

## Modify `config.properties` in Pentaho Shim Folder

In this document, we use version 5.1 as our example. We can find the `config.properties` folder in `<pentaho_folder>\data-integration\plugins\pentaho-big-data-plugin\hadoop-`

configurations\mapr510. More information on this topic is available at [Set Up Pentaho to Connect to a MapR Cluster](#).

1. Open the `config.properties` file found in the Pentaho Shim folder.
2. Edit the following values. The `windows.classpath` and `windows.librarypath` values depend on the version of the MapR client you installed. Make sure to keep the triple slashes:

```

windows.classpath=lib/hadoop2-windows-patch-
08072014.jar, file:///C:/opt/mapr/hadoop/hadoop-
2.7.0/etc/hadoop, file:///C:/opt/mapr/hadoop/hadoop-
2.7.0/share/hadoop/common/lib, file:///C:/opt/mapr/hadoop/hadoop-
2.7.0/share/hadoop/common, file:///C:/opt/mapr/hadoop/hadoop-
2.7.0/share/hadoop/hdfs, file:///C:/opt/mapr/hadoop/hadoop-
2.7.0/share/hadoop/hdfs/lib, file:///C:/opt/mapr/hadoop/hadoop-
2.7.0/share/hadoop/yarn/lib, file:///C:/opt/mapr/hadoop/hadoop-
2.7.0/share/hadoop/yarn, file:///C:/opt/mapr/hadoop/hadoop-
2.7.0/share/hadoop/mapreduce/lib, file:///C:/opt/mapr/hadoop/hadoop-
2.7.0/share/hadoop/mapreduce, file:///C:/Pentaho/design-tools/data-
integration/plugins/pentaho-big-data-plugin/hadoop-
configurations/mapr510, file:///C:/Pentaho/design-tools/data-
integration/plugins/pentaho-big-data-plugin/hadoop-
configurations/mapr510/lib, file:///C:/opt/mapr/lib

windows.librarypath=C:///opt///mapr///lib

```

3. Start or restart the PDI tool by running `spoon.bat` or `spoon.sh`.
4. Create a new Hadoop cluster for MapR and check the box to use the MapR client.
5. Test the connection as shown in Figure 9. You can ignore the Shim Configuration Verification warning since the value of `fs.defaultFS` does not exist in `core-site.xml`.

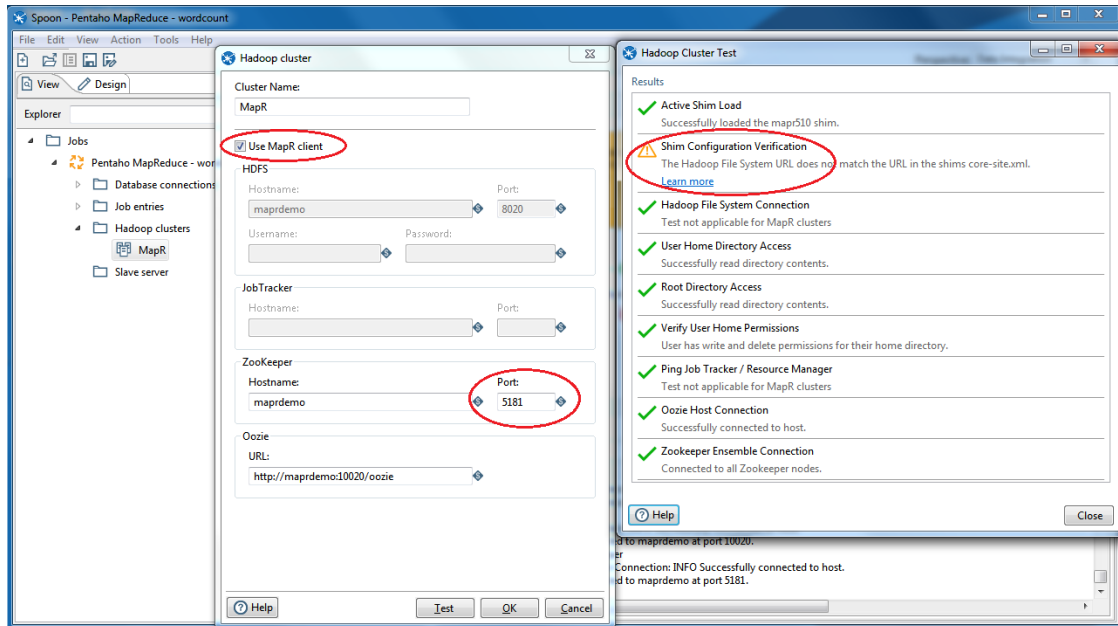


Figure 8: Hadoop Cluster Environment

## Run PDI PMR from Samples

To run PMR from the samples:

1. Open the sample MapReduce job called Pentaho MapReduce-wordcount.kjb from the samples\jobs\hadoop directory.
2. Open the mapping for **Copy Files to HDFS** and change the **Destination File/Folder** from `hdfs://maprdemo:8020/wordcount/input` to `maprfs://maprdemo:8020/wordcount/input`:

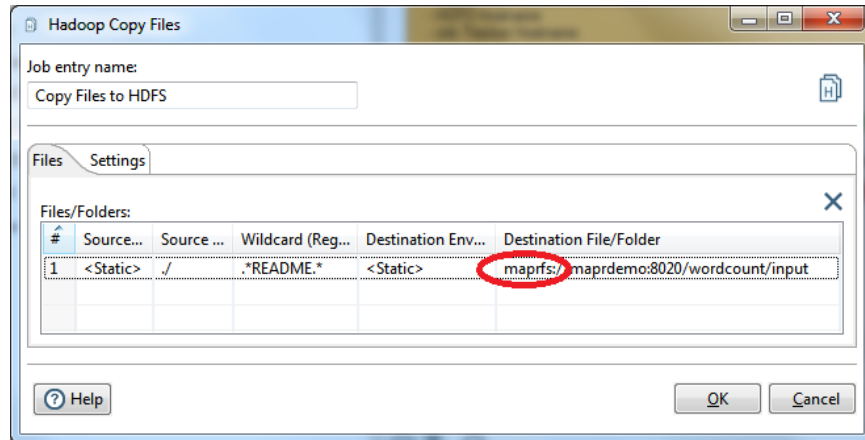


Figure 9: Hadoop Copy Files

3. Run the job and you will see a successful completion:

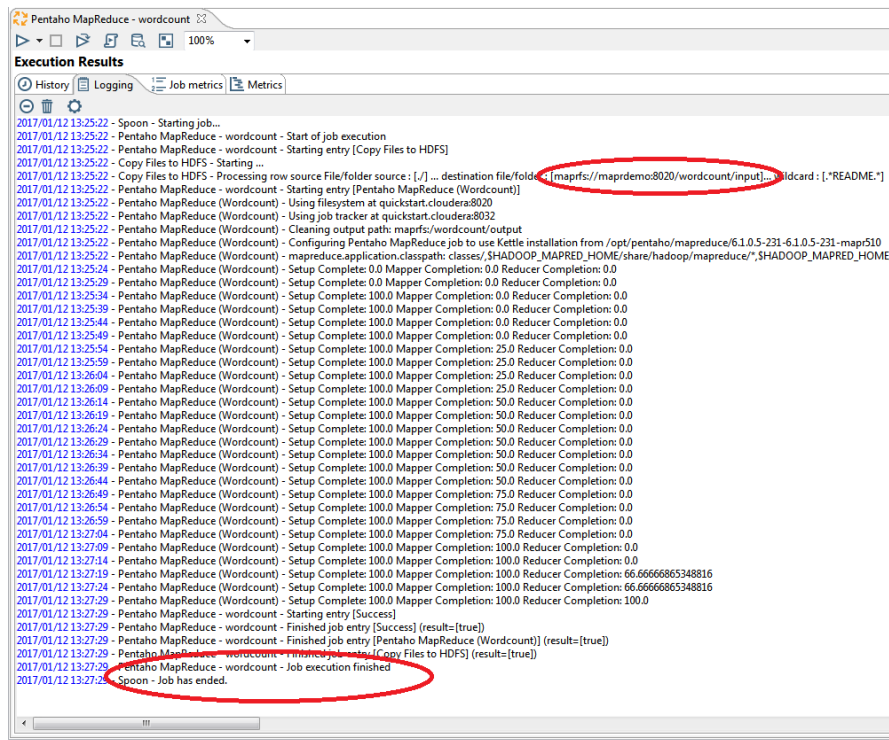


Figure 10: Execution Results

4. Confirm the job has run successfully by listing the contents of the `hdfs` directory where the files were generated.
5. If you encounter the error in the following figure, make sure the entry `fs.defaultFS` does *not* exist in the `core-site.xml` file (remove this property, if it exists):

---

```
<property>
  <name>fs.defaultFS</name>
  <value>maprfs://maprdemo:8020</value>
  <final>true</final>
</property>
```

---

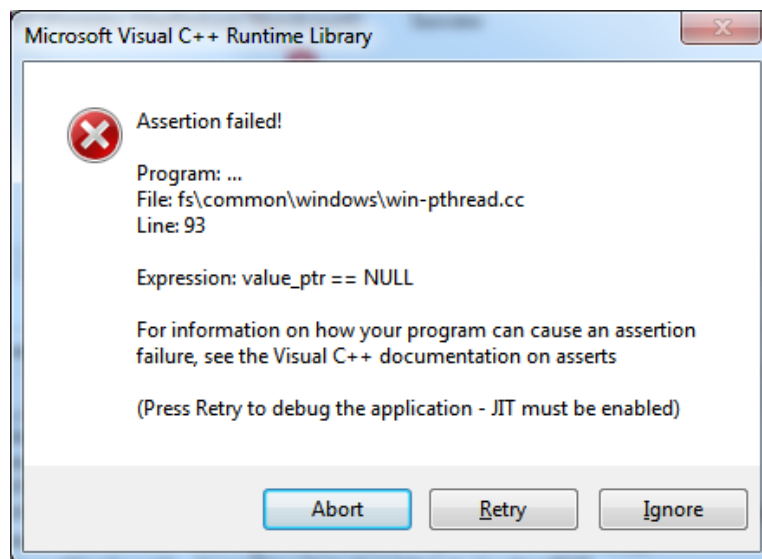


Figure 11: Assertion Failed Error

## Related Information

Here are some links to information that you may find helpful while using this best practices document:

- MapR
  - [Client Support Matrix](#)
  - [MapR Archive](#)
  - [MapR Download Sandbox](#)
  - [Ports Used by MapR](#)
- Pentaho
  - [Components Reference](#)
  - [Set Up Pentaho to Connect to a MapR Cluster](#)



## Finalization Checklist

This checklist is designed to be added to any implemented project that uses this collection of best practices, to verify that all items have been considered and reviews have been performed.

Name of the Project: \_\_\_\_\_

Date of the Review: \_\_\_\_\_

Name of the Reviewer: \_\_\_\_\_

Item	Response	Comments
Did you obtain the MapR server information?	YES_____ NO_____	
Did you set up your host environment?	YES_____ NO_____	
Did you download and install the Pentaho Shim for MapR, if necessary?	YES_____ NO_____	
Did you download the latest MapR client tools from the index?	YES_____ NO_____	
Have you set up the environment variables?	YES_____ NO_____	
Have you configured MapR to connect to the HDFS?	YES_____ NO_____	
Did you modify <code>core-site.xml</code> for the MapR client?	YES_____ NO_____	
Did you modify <code>mapred-site.xml</code> for the MapR client?	YES_____ NO_____	
Did you connect to the HDFS using the MapR client?	YES_____ NO_____	
Did you modify <code>config.properties</code> in the Pentaho Shim folder?	YES_____ NO_____	
Did you run PDI PMR from samples?	YES_____ NO_____	