

# Lab 01: HDFS, MapReduce, Pig, Hive, and Jaql

*Hands-On Lab*



## Table of Contents

<b>1</b>	<b>Introduction</b> .....	<b>3</b>
<b>2</b>	<b>About this Lab</b> .....	<b>3</b>
<b>3</b>	<b>Environment Setup Requirements</b> .....	<b>3</b>
3.1	Getting Started .....	3
<b>4</b>	<b>Exploring Hadoop Distributed File System (HDFS)</b> .....	<b>6</b>
4.1	Using the command line interface.....	6
<b>5</b>	<b>MapReduce</b> .....	<b>13</b>
5.1	Running the WordCount program .....	13
<b>6</b>	<b>Working with Pig</b> .....	<b>15</b>
<b>7</b>	<b>Working with Hive</b> .....	<b>18</b>
<b>8</b>	<b>Working with Jaql</b> .....	<b>22</b>
<b>9</b>	<b>Summary</b> .....	<b>26</b>

## 1 Introduction

The overwhelming trend towards digital services, combined with cheap storage, has generated massive amounts of data that enterprises need to effectively gather, process, and analyze. Techniques from the data warehousing and high-performance computing communities are invaluable for many enterprises. However, often times their cost or complexity of scale-up discourages the accumulation of data without an immediate need. As valuable knowledge may nevertheless be buried in this data, related scaled-up technologies have been developed. Examples include Google's MapReduce, and the open-source implementation, Apache Hadoop.

Hadoop is an open-source project administered by the Apache Software Foundation. Hadoop's contributors work for some of the world's biggest technology companies. That diverse, motivated community has produced a collaborative platform for consolidating, combining and understanding data.

Technically, Hadoop consists of two key services: data storage using the Hadoop Distributed File System (HDFS) and large scale parallel data processing using a technique called MapReduce

## 2 About this Lab

After completing this hands-on lab, you'll be able to:

- Use Hadoop commands to explore the HDFS on the Hadoop system
- Use Hadoop commands to run a sample MapReduce program on the Hadoop system
- Explore Pig, Hive and Jaql

## 3 Environment Setup Requirements

To complete this lab you will need the following:

1. InfoSphere BigInsights Bootcamp VMware® image
2. VMware Player 2.x or VMware Workstation 5.x or later

For help on how to obtain these components please follow the instructions specified in VMware Basics and Introduction from module 1.

### 3.1 Getting Started

To prepare for the contents of this lab, you must go through the process of getting all of the Hadoop components started.

1. Start the VMware image by clicking the  button in VMware Workstation if it is not already on.
2. Log in to the VMware virtual machine using the following information:
  - User: biadmin
  - Password: password

3. Open Gnome Command Prompt Window by right-clicking on the Desktop and selecting "Open in Terminal".

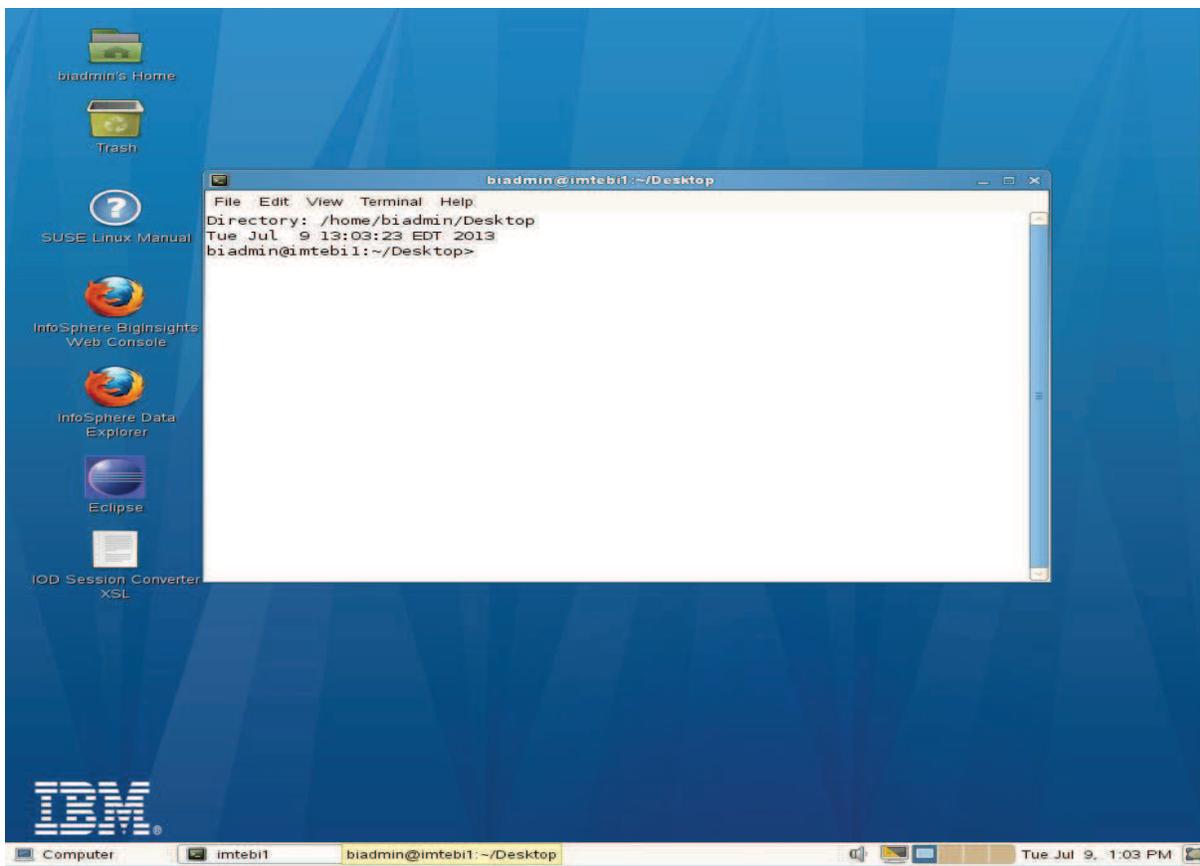


Figure 1 - Open a new terminal window

4. Change to the **\$BIGINSIGHTS\_HOME** (which by default is set to `/opt/ibm/biginsights`).

```
cd $BIGINSIGHTS_HOME/bin
```

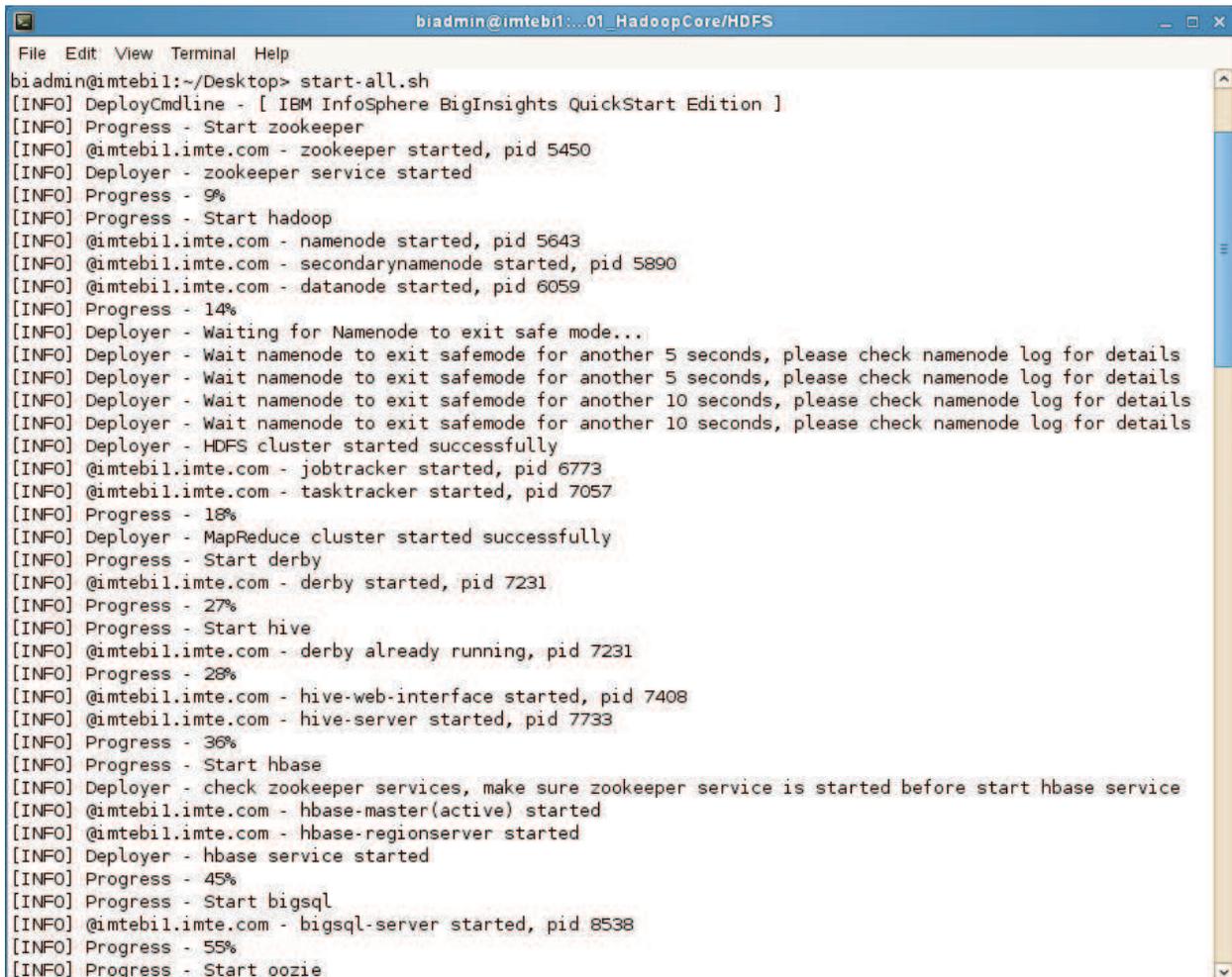
or

```
cd /opt/ibm/biginsights/bin
```

5. Start the Hadoop components (daemons) on the BigInsights server. You can practice starting all components with these commands. Please note they will take a few minutes to run:

```
./start-all.sh
```

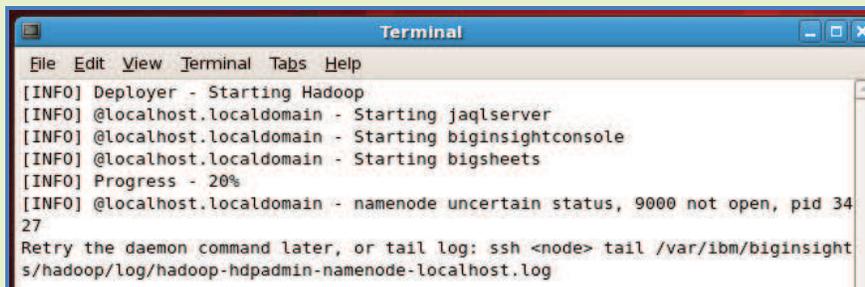
The following figure shows the different Hadoop components starting.



```
biadmin@imtebil:~/Desktop> start-all.sh
[INFO] DeployCmdline - [ IBM InfoSphere BigInsights QuickStart Edition ]
[INFO] Progress - Start zookeeper
[INFO] @imtebil.imte.com - zookeeper started, pid 5450
[INFO] Deployer - zookeeper service started
[INFO] Progress - 9%
[INFO] Progress - Start hadoop
[INFO] @imtebil.imte.com - namenode started, pid 5643
[INFO] @imtebil.imte.com - secondarynamenode started, pid 5890
[INFO] @imtebil.imte.com - datanode started, pid 6059
[INFO] Progress - 14%
[INFO] Deployer - Waiting for Namenode to exit safe mode...
[INFO] Deployer - Wait namenode to exit safemode for another 5 seconds, please check namenode log for details
[INFO] Deployer - Wait namenode to exit safemode for another 5 seconds, please check namenode log for details
[INFO] Deployer - Wait namenode to exit safemode for another 10 seconds, please check namenode log for details
[INFO] Deployer - Wait namenode to exit safemode for another 10 seconds, please check namenode log for details
[INFO] Deployer - HDFS cluster started successfully
[INFO] @imtebil.imte.com - jobtracker started, pid 6773
[INFO] @imtebil.imte.com - tasktracker started, pid 7057
[INFO] Progress - 18%
[INFO] Deployer - MapReduce cluster started successfully
[INFO] Progress - Start derby
[INFO] @imtebil.imte.com - derby started, pid 7231
[INFO] Progress - 27%
[INFO] Progress - Start hive
[INFO] @imtebil.imte.com - derby already running, pid 7231
[INFO] Progress - 28%
[INFO] @imtebil.imte.com - hive-web-interface started, pid 7408
[INFO] @imtebil.imte.com - hive-server started, pid 7733
[INFO] Progress - 36%
[INFO] Progress - Start hbase
[INFO] Deployer - check zookeeper services, make sure zookeeper service is started before start hbase service
[INFO] @imtebil.imte.com - hbase-master(active) started
[INFO] @imtebil.imte.com - hbase-regionserver started
[INFO] Deployer - hbase service started
[INFO] Progress - 45%
[INFO] Progress - Start bigsql
[INFO] @imtebil.imte.com - bigsql-server started, pid 8538
[INFO] Progress - 55%
[INFO] Progress - Start oozie
```

Figure 2 - Starting Hadoop components

**Note:** You may get an error that the server has not started, please be patient as it does take some time for the server to complete start.



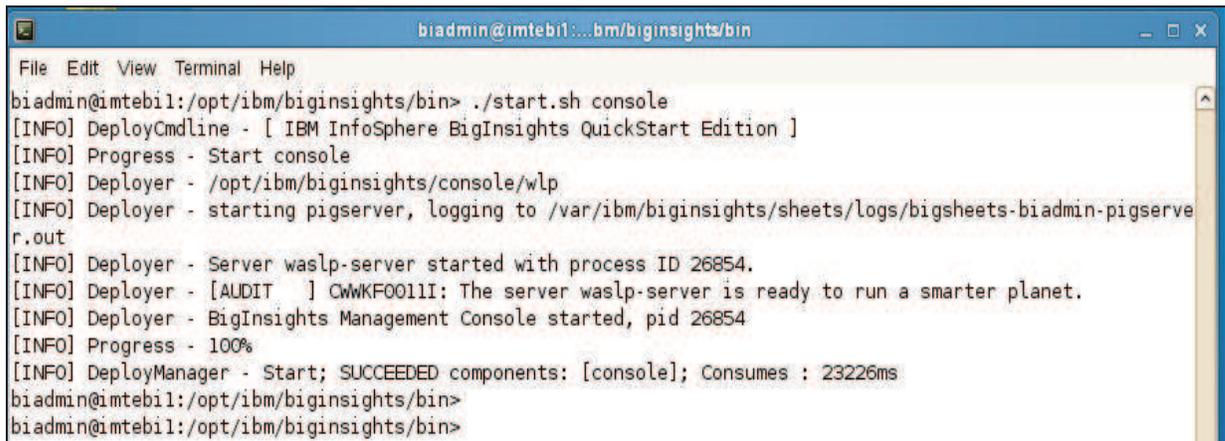
```
Terminal
File Edit View Terminal Tabs Help
[INFO] Deployer - Starting Hadoop
[INFO] @localhost.localdomain - Starting jaqlserver
[INFO] @localhost.localdomain - Starting biginsightconsole
[INFO] @localhost.localdomain - Starting bigsheets
[INFO] Progress - 20%
[INFO] @localhost.localdomain - namenode uncertain status, 9000 not open, pid 3427
Retry the daemon command later, or tail log: ssh <node> tail /var/ibm/biginsights/hadoop/log/hadoop-hdpadmin-namenode-localhost.log
```

Figure 3 - Hadoop component error

6. Sometimes certain hadoop components may fail to start. You can start and stop the failed components one at a time by using **start.sh** or **stop.sh** respectively. For example, to start and stop Hadoop use:

```
./start.sh hadoop
./stop.sh hadoop
```

In the following example, the console component failed. The particular component was then started again using the `./start.sh console` command. It then succeeded without any problems. This approach can be used for any failed components.



```
biadmin@imtebil:~/opt/ibm/biginsights/bin
File Edit View Terminal Help
biadmin@imtebil:/opt/ibm/biginsights/bin> ./start.sh console
[INFO] DeployCmdline - [ IBM InfoSphere BigInsights QuickStart Edition ]
[INFO] Progress - Start console
[INFO] Deployer - /opt/ibm/biginsights/console/wlp
[INFO] Deployer - starting pigserver, logging to /var/ibm/biginsights/sheets/logs/bigsheets-biadmin-pigserver.out
[INFO] Deployer - Server waslp-server started with process ID 26854.
[INFO] Deployer - [AUDIT  ] CwWKF0011I: The server waslp-server is ready to run a smarter planet.
[INFO] Deployer - BigInsights Management Console started, pid 26854
[INFO] Progress - 100%
[INFO] DeployManager - Start; SUCCEEDED components: [console]; Consumes : 23226ms
biadmin@imtebil:/opt/ibm/biginsights/bin>
biadmin@imtebil:/opt/ibm/biginsights/bin>
```

Figure 4 - Starting a specific component

Once all components have started successfully you can then move to the next section.

## 4 Exploring Hadoop Distributed File System (HDFS)

Hadoop Distributed File System (HDFS), allows user data to be organized in the form of files and directories. It provides a command line interface called *FS shell* that lets a user interact with the data in HDFS accessible to Hadoop MapReduce programs.

There are two methods to interact with HDFS:

1. You can use the command-line approach and invoke the FileSystem (fs) shell using the format: `hadoop fs <args>`. This is the method we will use in this lab..
2. You can also manipulate HDFS using the BigInsights Web Console. You will explore the BigInsights Web Console on another lab.

### 4.1 Using the command line interface

In this part, we will explore some basic HDFS commands. All HDFS commands start with **hadoop** followed by **dfs** (distributed file system) or **fs** (file system) followed by a dash, and the command. Many HDFS commands are similar to UNIX commands. For details, refer to the *Hadoop Command Guide* and *Hadoop FS Shell Guide*.

We will start with the `hadoop fs -ls` command which returns the list of files and directories with permission information.

Ensure the Hadoop components are all started, and from the same Gnome terminal window as before (and logged on as *biadmin*), follow these instructions:

1. List the contents of the root directory.

```
hadoop fs -ls /
```



```
biadmin@imtebil:~/bm/biginsights/bin
File Edit View Terminal Help
biadmin@imtebil:/opt/ibm/biginsights/bin> hadoop fs -ls /
Found 5 items
drwxr-xr-x - biadmin biadmgrp      0 2013-06-19 22:56 /biginsights
drwxr-xr-x - biadmin supergroup    0 2013-07-09 13:48 /hadoop
drwxr-xr-x - biadmin supergroup    0 2013-07-09 13:50 /hbase
drwxrwxrwx - biadmin supergroup    0 2013-06-19 22:45 /tmp
drwxrwxrwx - biadmin supergroup    0 2013-07-09 13:33 /user
biadmin@imtebil:/opt/ibm/biginsights/bin>
```

Figure 5 - List directory command

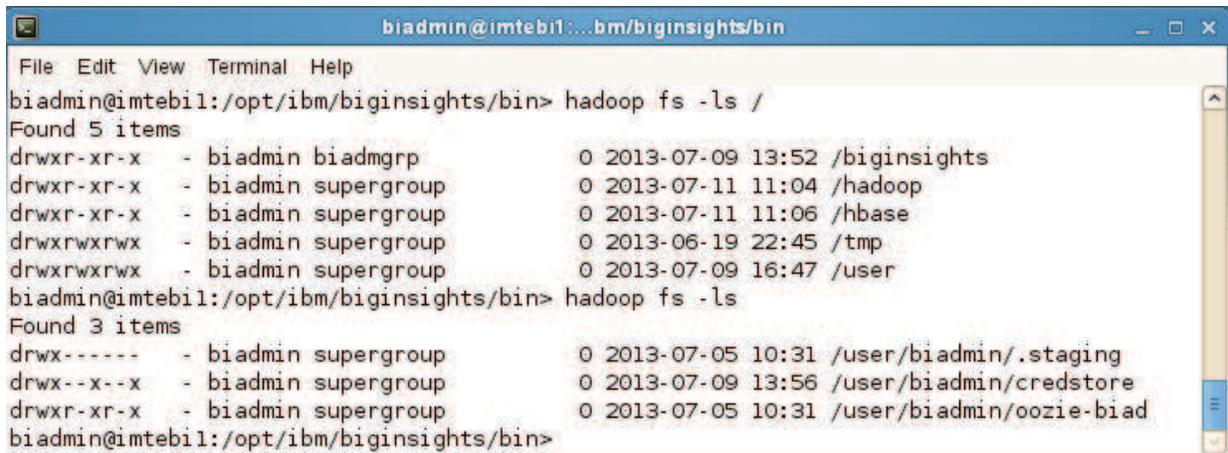
2. To list the contents of the */user/biadmin* directory, execute:

```
hadoop fs -ls
```

or

```
hadoop fs -ls /user/biadmin
```

Note that in the first command there was no directory referenced, but it is equivalent to the second command where */user/biadmin* is explicitly specified. Each user will get its own home directory under */user*. For example, in the case of user *biadmin*, his home directory is */user/biadmin*. Any command where there is no explicit directory specified will be relative to the user's home directory.



```
biadmin@imtebil:~/bm/biginsights/bin
File Edit View Terminal Help
biadmin@imtebil:/opt/ibm/biginsights/bin> hadoop fs -ls /
Found 5 items
drwxr-xr-x - biadmin biadmgrp      0 2013-07-09 13:52 /biginsights
drwxr-xr-x - biadmin supergroup    0 2013-07-11 11:04 /hadoop
drwxr-xr-x - biadmin supergroup    0 2013-07-11 11:06 /hbase
drwxrwxrwx - biadmin supergroup    0 2013-06-19 22:45 /tmp
drwxrwxrwx - biadmin supergroup    0 2013-07-09 16:47 /user
biadmin@imtebil:/opt/ibm/biginsights/bin> hadoop fs -ls
Found 3 items
drwx----- - biadmin supergroup    0 2013-07-05 10:31 /user/biadmin/.staging
drwx--x-x-x - biadmin supergroup    0 2013-07-09 13:56 /user/biadmin/credstore
drwxr-xr-x - biadmin supergroup    0 2013-07-05 10:31 /user/biadmin/oozie-biad
biadmin@imtebil:/opt/ibm/biginsights/bin>
```

Figure 6 - hadoop fs -ls command outputs

3. To create the directory *myTestDir* you can issue the following command:

```
hadoop fs -mkdir myTestDir
```

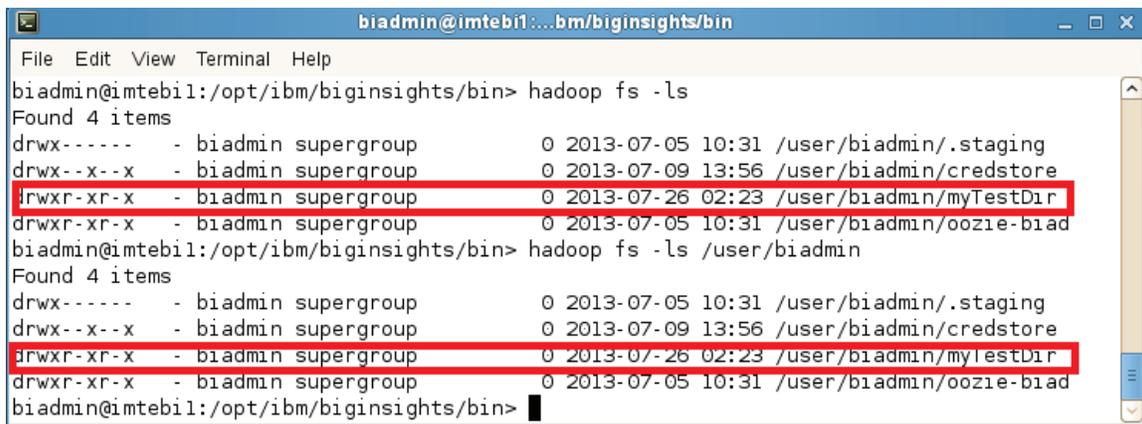
Where was this directory created? As mentioned in the previous step, any relative paths will be using the user's home directory.

4. Issue the ls command again to see the subdirectory myTestDir:

```
hadoop fs -ls
```

or

```
hadoop fs -ls /user/biadmin
```



```
biadmin@imtebi1:~/opt/ibm/biginsights/bin
File Edit View Terminal Help
biadmin@imtebi1:/opt/ibm/biginsights/bin> hadoop fs -ls
Found 4 items
drwx----- - biadmin supergroup      0 2013-07-05 10:31 /user/biadmin/.staging
drwx--x--x - biadmin supergroup      0 2013-07-09 13:56 /user/biadmin/credstore
drwxr-xr-x - biadmin supergroup      0 2013-07-26 02:23 /user/biadmin/myTestDir
drwxr-xr-x - biadmin supergroup      0 2013-07-05 10:31 /user/biadmin/oozie-biadmin
biadmin@imtebi1:/opt/ibm/biginsights/bin> hadoop fs -ls /user/biadmin
Found 4 items
drwx----- - biadmin supergroup      0 2013-07-05 10:31 /user/biadmin/.staging
drwx--x--x - biadmin supergroup      0 2013-07-09 13:56 /user/biadmin/credstore
drwxr-xr-x - biadmin supergroup      0 2013-07-26 02:23 /user/biadmin/myTestDir
drwxr-xr-x - biadmin supergroup      0 2013-07-05 10:31 /user/biadmin/oozie-biadmin
biadmin@imtebi1:/opt/ibm/biginsights/bin>
```

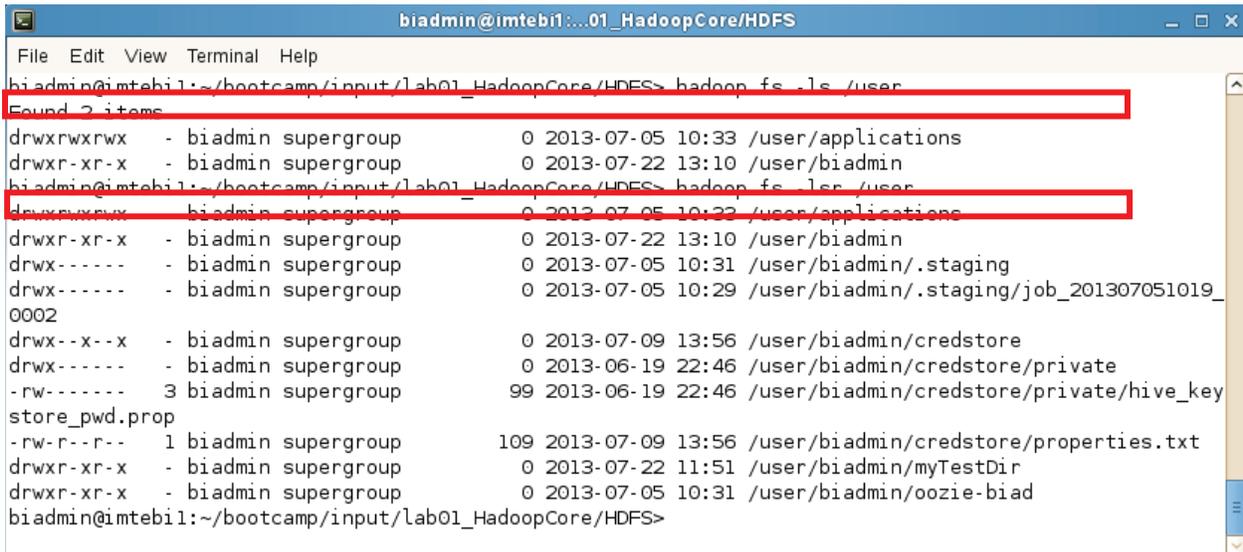
Figure 7 - hadoop fs -ls command outputs

**Note:** If you specify a relative path to hadoop fs commands, they will implicitly be relative to your user directory in HDFS. For example when you created the directory myTestDir, it was created in the /user/biadmin directory.

To use HDFS commands recursively generally you add an “r” to the HDFS command (In the Linux shell this is generally done with the “-R” argument).

5. For example, to do a recursive listing we'll use the -lsr command rather than just -ls, like the examples below:

```
hadoop fs -ls /user
hadoop fs -lsr /user
```

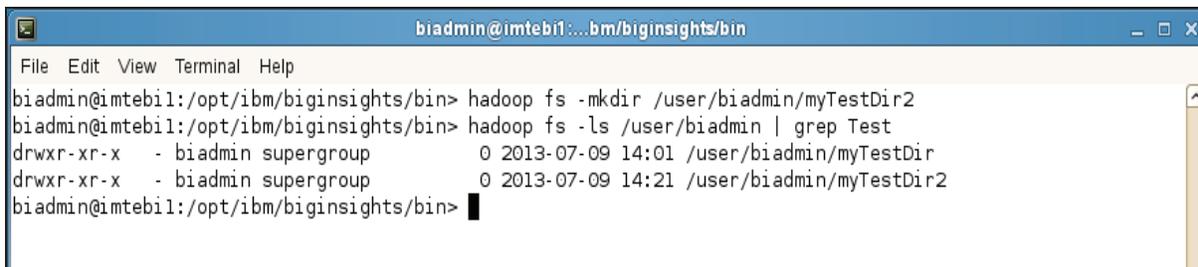


```
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/HDFS
File Edit View Terminal Help
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/HDFS> hadoop fs -ls /user
Found 2 items
drwxrwxrwx - biadmin supergroup 0 2013-07-05 10:33 /user/applications
drwxr-xr-x - biadmin supergroup 0 2013-07-22 13:10 /user/biadmin
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/HDFS> hadoop fs -lsr /user
drwxrwxrwx - biadmin supergroup 0 2013-07-05 10:33 /user/applications
drwxr-xr-x - biadmin supergroup 0 2013-07-22 13:10 /user/biadmin
drwx----- - biadmin supergroup 0 2013-07-05 10:31 /user/biadmin/.staging
drwx----- - biadmin supergroup 0 2013-07-05 10:29 /user/biadmin/.staging/job_201307051019_0002
drwx--x--x - biadmin supergroup 0 2013-07-09 13:56 /user/biadmin/credstore
drwx----- - biadmin supergroup 0 2013-06-19 22:46 /user/biadmin/credstore/private
-rw----- 3 biadmin supergroup 99 2013-06-19 22:46 /user/biadmin/credstore/private/hive_keystore_pwd.prop
-rw-r--r-- 1 biadmin supergroup 109 2013-07-09 13:56 /user/biadmin/credstore/properties.txt
drwxr-xr-x - biadmin supergroup 0 2013-07-22 11:51 /user/biadmin/myTestDir
drwxr-xr-x - biadmin supergroup 0 2013-07-05 10:31 /user/biadmin/oozie-biadmin
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/HDFS>
```

Figure 8 - hadoop fs-lsr command output

6. You can pipe (using the | character) any HDFS command to be used with the Linux shell. For example, you can easily use *grep* with HDFS by doing the following:

```
hadoop fs -mkdir /user/biadmin/myTestDir2
hadoop fs -ls /user/biadmin | grep Test
```



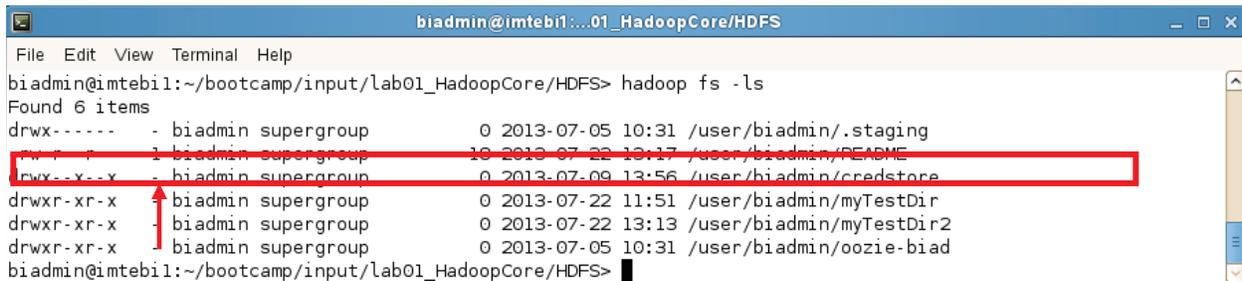
```
biadmin@imtebil:~/opt/ibm/biginsights/bin
File Edit View Terminal Help
biadmin@imtebil:/opt/ibm/biginsights/bin> hadoop fs -mkdir /user/biadmin/myTestDir2
biadmin@imtebil:/opt/ibm/biginsights/bin> hadoop fs -ls /user/biadmin | grep Test
drwxr-xr-x - biadmin supergroup 0 2013-07-09 14:01 /user/biadmin/myTestDir
drwxr-xr-x - biadmin supergroup 0 2013-07-09 14:21 /user/biadmin/myTestDir2
biadmin@imtebil:/opt/ibm/biginsights/bin>
```

Figure 9 - hadoop fs -ls using grep to filter the output

As you can see the *grep* command only returned the lines which had test in them (thus removing the “Found x items” line and the *.staging* and *oozie-biadmin* directories from the listing

7. To move files between your regular Linux filesystem and HDFS you can use the *put* and *get* commands. For example, move the text file *README* to the hadoop filesystem.

```
hadoop fs -put /home/biadmin/bootcamp/input/lab01_HadoopCore/HDFS/README
README
hadoop fs -ls /user/biadmin
```



```
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/HDFS> hadoop fs -ls
Found 6 items
drwx----- - biadmin supergroup          0 2013-07-05 10:31 /user/biadmin/.staging
-rw-r--r-- 1 biadmin supergroup          10 2013-07-22 13:17 /user/biadmin/README
-rwx--x--x - biadmin supergroup          0 2013-07-09 13:56 /user/biadmin/credstore
drwxr-xr-x  biadmin supergroup          0 2013-07-22 11:51 /user/biadmin/myTestDir
drwxr-xr-x  biadmin supergroup          0 2013-07-22 13:13 /user/biadmin/myTestDir2
drwxr-xr-x  biadmin supergroup          0 2013-07-05 10:31 /user/biadmin/oozie-biad
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/HDFS>
```

Figure 10 - README file inside HDFS

You should now see a new file called /user/biadmin/README listed as shown above. Note there is a '1' highlighted in the figure. This represents the replication factor. By default, the replication factor in a BigInsights cluster is 3, but since this laboratory environment only has one node, the replication factor is 1.

8. In order to view the contents of this file use the `-cat` command as follows:

```
hadoop fs -cat README
```

You should see the output of the README file (that is stored in HDFS). We can also use the linux `diff` command to see if the file we put on HDFS is actually the same as the original on the local filesystem.

9. Execute the commands below to use the `diff` command:

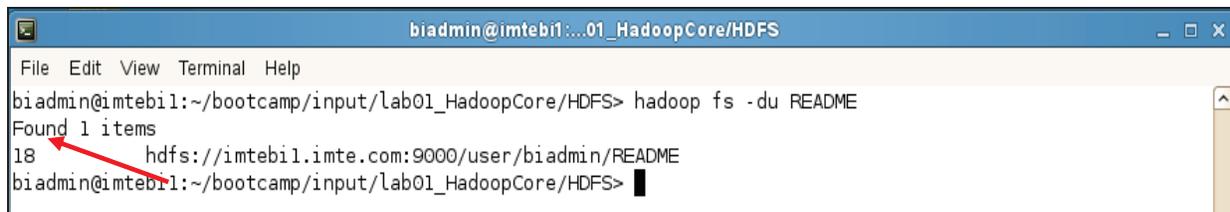
```
cd /home/biadmin/bootcamp/input/lab01_HadoopCore/HDFS/
diff <( hadoop fs -cat README ) README
```

Since the `diff` command produces no output we know that the files are the same (the `diff` command prints all the lines in the files that differ).

To find the size of files you need to use the `-du` or `-dus` commands. Keep in mind that these commands return the file size in bytes.

10. To find the size of the README file use the following command:

```
hadoop fs -du README
```



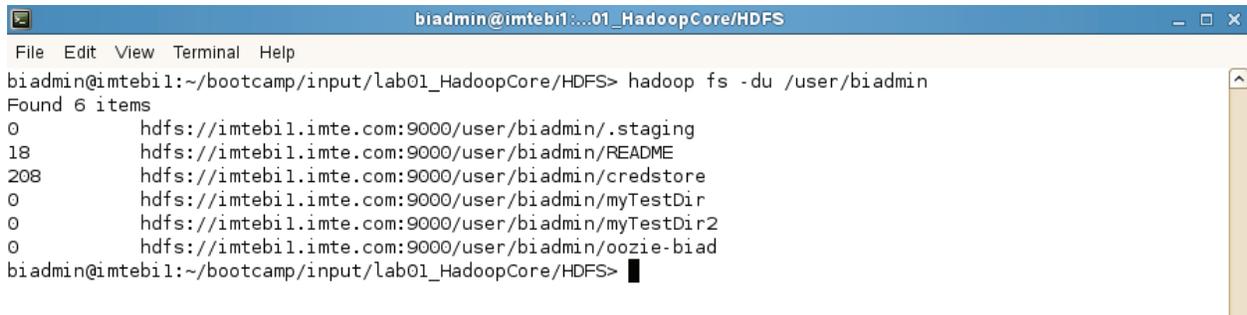
```
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/HDFS> hadoop fs -du README
Found 1 items
18      hdfs://imtebil.imte.com:9000/user/biadmin/README
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/HDFS>
```

Figure 11 - Inspecting README file size

In this example, the README file has 18 bytes.

11. To find the size of all files individually in the /user/biadmin directory use the following command:

```
hadoop fs -du /user/biadmin
```

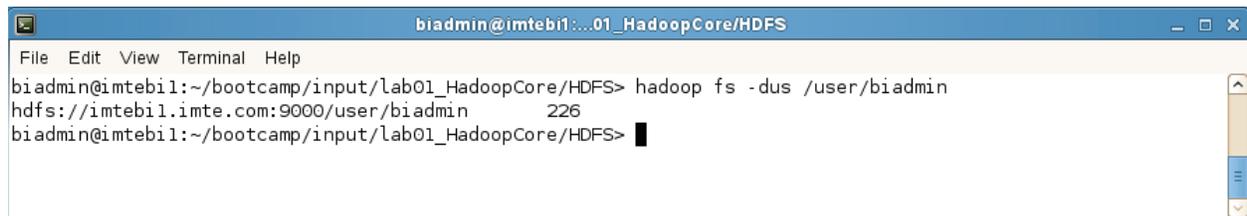


```
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/HDFS
File Edit View Terminal Help
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/HDFS> hadoop fs -du /user/biadmin
Found 6 items
0          hdfs://imtebil.imte.com:9000/user/biadmin/.staging
18         hdfs://imtebil.imte.com:9000/user/biadmin/README
208       hdfs://imtebil.imte.com:9000/user/biadmin/credstore
0         hdfs://imtebil.imte.com:9000/user/biadmin/myTestDir
0         hdfs://imtebil.imte.com:9000/user/biadmin/myTestDir2
0         hdfs://imtebil.imte.com:9000/user/biadmin/oozie-biad
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/HDFS>
```

Figure 12 - Inspecting files size in a specific directory

12. To find the size of all files in total of the /user/biadmin directory use the following command:

```
hadoop fs -dus /user/biadmin
```

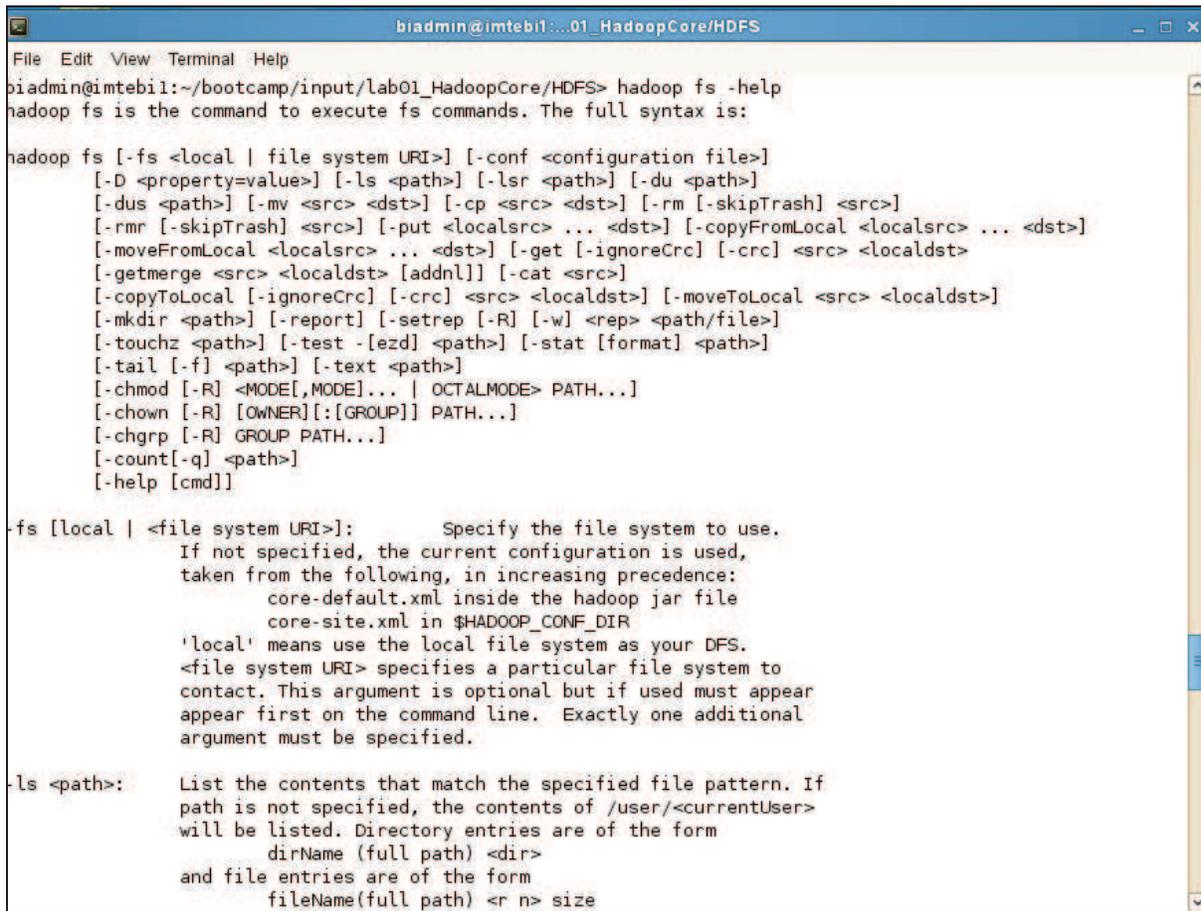


```
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/HDFS
File Edit View Terminal Help
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/HDFS> hadoop fs -dus /user/biadmin
hdfs://imtebil.imte.com:9000/user/biadmin      226
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/HDFS>
```

Figure 13 - Inspecting the size of directories

13. If you would like to get more information about hadoop fs commands, invoke `-help` as follows:

```
hadoop fs -help
```



```
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/HDFS
File Edit View Terminal Help
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/HDFS> hadoop fs -help
hadoop fs is the command to execute fs commands. The full syntax is:

hadoop fs [-fs <local | file system URI>] [-conf <configuration file>]
  [-D <property=value>] [-ls <path>] [-lsr <path>] [-du <path>]
  [-dus <path>] [-mv <src> <dst>] [-cp <src> <dst>] [-rm [-skipTrash] <src>]
  [-rmr [-skipTrash] <src>] [-put <localsrc> ... <dst>] [-copyFromLocal <localsrc> ... <dst>]
  [-moveFromLocal <localsrc> ... <dst>] [-get [-ignoreCrc] [-crc] <src> <localdst>]
  [-getmerge <src> <localdst> [addnl]] [-cat <src>]
  [-copyToLocal [-ignoreCrc] [-crc] <src> <localdst>] [-moveToLocal <src> <localdst>]
  [-mkdir <path>] [-report] [-setrep [-R] [-w] <rep> <path/file>]
  [-touchz <path>] [-test [-ezd] <path>] [-stat [format] <path>]
  [-tail [-f] <path>] [-text <path>]
  [-chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH...]
  [-chown [-R] [OWNER][:[GROUP]] PATH...]
  [-chgrp [-R] GROUP PATH...]
  [-count[-q] <path>]
  [-help [cmd]]

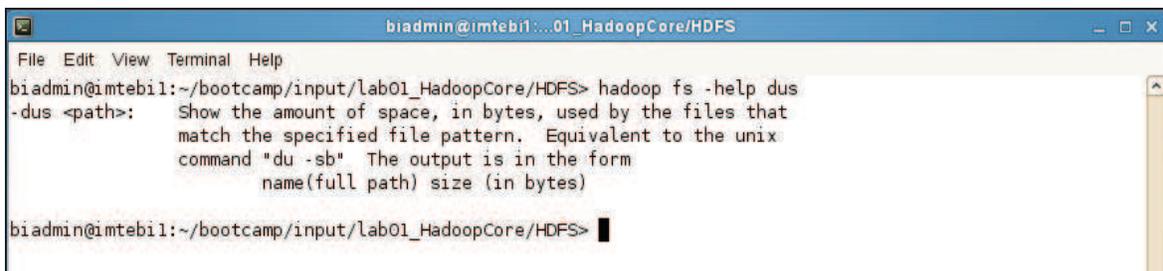
-fs [local | <file system URI>]:      Specify the file system to use.
  If not specified, the current configuration is used,
  taken from the following, in increasing precedence:
    core-default.xml inside the hadoop jar file
    core-site.xml in $HADOOP_CONF_DIR
  'local' means use the local file system as your DFS.
  <file system URI> specifies a particular file system to
  contact. This argument is optional but if used must appear
  first on the command line. Exactly one additional
  argument must be specified.

-ls <path>:      List the contents that match the specified file pattern. If
  path is not specified, the contents of /user/<currentUser>
  will be listed. Directory entries are of the form
    dirName (full path) <dir>
  and file entries are of the form
    fileName(full path) <r n> size
```

Figure 14 - Hadoop help command

14. For specific help on a command, add the command name after help. For example, to get help on the dus command you'd do the following:

```
hadoop fs -help dus
```



```
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/HDFS
File Edit View Terminal Help
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/HDFS> hadoop fs -help dus
-dus <path>:      Show the amount of space, in bytes, used by the files that
  match the specified file pattern. Equivalent to the unix
  command "du -sb" The output is in the form
    name(full path) size (in bytes)

biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/HDFS>
```

Figure 15 - Help for specific Hadoop commands

## 5 MapReduce

Now that we've seen how the FileSystem (fs) shell can be used to execute Hadoop commands to interact with HDFS, the same fs shell can be used to launch MapReduce jobs. In this section, we will walk through the steps required to run a MapReduce program. The source code for a MapReduce program is contained in a compiled .jar file. Hadoop will load the JAR into HDFS and distribute it to the data nodes, where the individual tasks of the MapReduce job will be executed. Hadoop ships with some example MapReduce programs to run. One of these is a distributed WordCount program which reads text files and counts how often words occur.

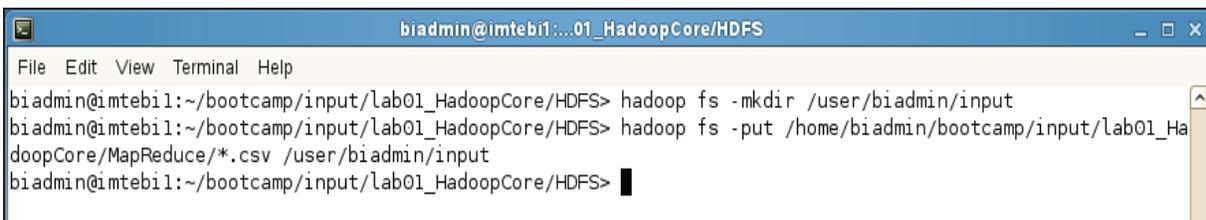
### 5.1 Running the WordCount program

First we need to copy the data files from the local file system to HDFS.

1. Execute the commands below to copy the input files into HDFS.

```
hadoop fs -mkdir /user/biadmin/input

hadoop fs -put /home/biadmin/bootcamp/input/lab01_HadoopCore/MapReduce/*.csv
/user/biadmin/input
```

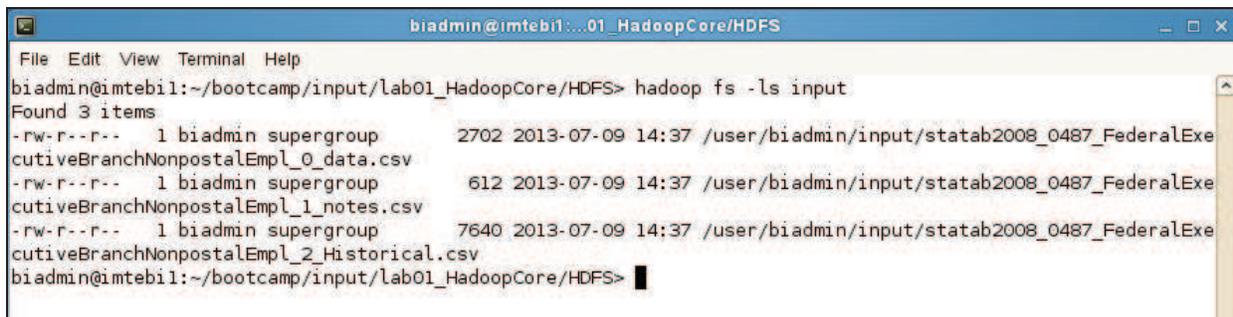


```
biadmin@imtebil:...01_HadoopCore/HDFS
File Edit View Terminal Help
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/HDFS> hadoop fs -mkdir /user/biadmin/input
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/HDFS> hadoop fs -put /home/biadmin/bootcamp/input/lab01_Ha
dooopCore/MapReduce/*.csv /user/biadmin/input
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/HDFS> █
```

Figure 16 - Copy input files into HDFS

2. Review the files have been copied with the following command:

```
hadoop fs -ls input
```

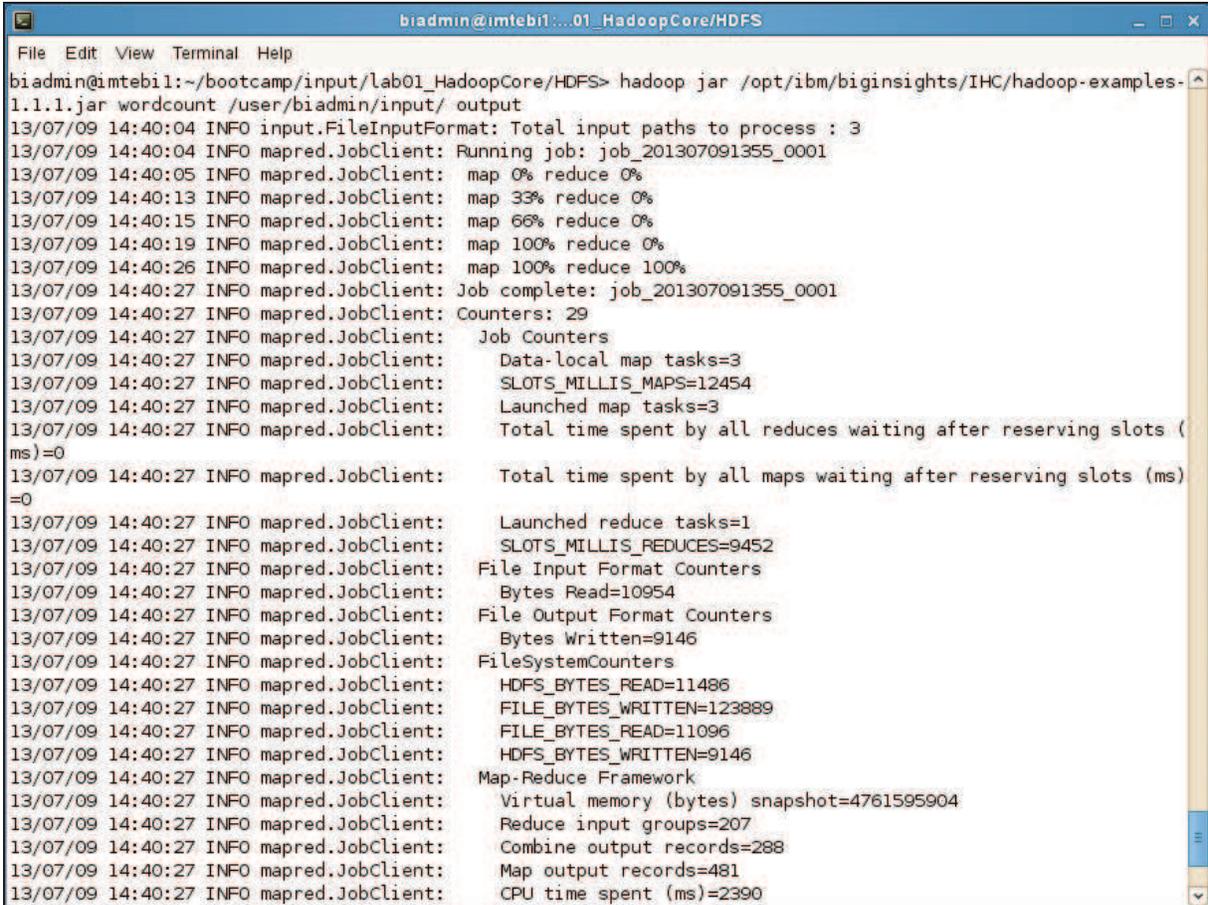


```
biadmin@imtebil:...01_HadoopCore/HDFS
File Edit View Terminal Help
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/HDFS> hadoop fs -ls input
Found 3 items
-rw-r--r-- 1 biadmin supergroup      2702 2013-07-09 14:37 /user/biadmin/input/statab2008_0487_FederalExe
cutiveBranchNonpostalEmpl_0_data.csv
-rw-r--r-- 1 biadmin supergroup       612 2013-07-09 14:37 /user/biadmin/input/statab2008_0487_FederalExe
cutiveBranchNonpostalEmpl_1_notes.csv
-rw-r--r-- 1 biadmin supergroup      7640 2013-07-09 14:37 /user/biadmin/input/statab2008_0487_FederalExe
cutiveBranchNonpostalEmpl_2_Historical.csv
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/HDFS> █
```

Figure 17 - List copied files into HDFS

3. Now we can run the wordcount job with the command below, where “/user/biadmin/input/” is where the input files are, and “output” is the directory where the output of the job will be stored. The “output” directory will be created automatically when executing the command below.

```
hadoop jar /opt/ibm/biginsights/IHC/hadoop-examples-1.1.1.jar wordcount /user/biadmin/input/ output
```

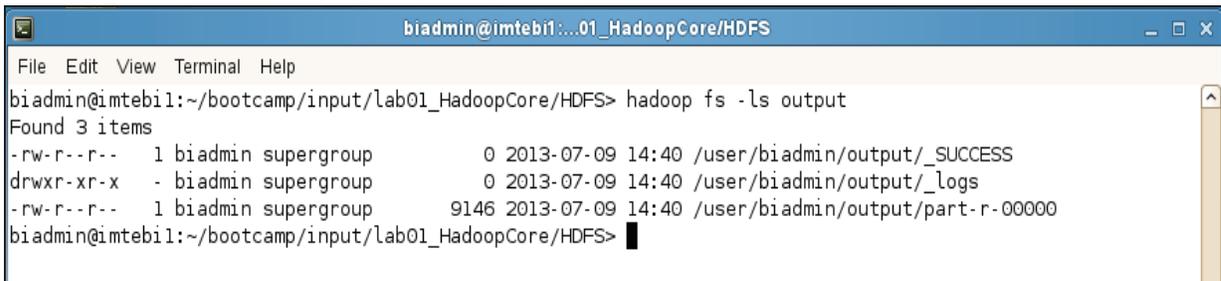


```
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/HDFS> hadoop jar /opt/ibm/biginsights/IHC/hadoop-examples-1.1.1.jar wordcount /user/biadmin/input/ output
13/07/09 14:40:04 INFO input.FileInputFormat: Total input paths to process : 3
13/07/09 14:40:04 INFO mapred.JobClient: Running job: job_201307091355_0001
13/07/09 14:40:05 INFO mapred.JobClient: map 0% reduce 0%
13/07/09 14:40:13 INFO mapred.JobClient: map 33% reduce 0%
13/07/09 14:40:15 INFO mapred.JobClient: map 66% reduce 0%
13/07/09 14:40:19 INFO mapred.JobClient: map 100% reduce 0%
13/07/09 14:40:26 INFO mapred.JobClient: map 100% reduce 100%
13/07/09 14:40:27 INFO mapred.JobClient: Job complete: job_201307091355_0001
13/07/09 14:40:27 INFO mapred.JobClient: Counters: 29
13/07/09 14:40:27 INFO mapred.JobClient: Job Counters
13/07/09 14:40:27 INFO mapred.JobClient: Data-local map tasks=3
13/07/09 14:40:27 INFO mapred.JobClient: SLOTS_MILLIS_MAPS=12454
13/07/09 14:40:27 INFO mapred.JobClient: Launched map tasks=3
13/07/09 14:40:27 INFO mapred.JobClient: Total time spent by all reduces waiting after reserving slots (ms)=0
13/07/09 14:40:27 INFO mapred.JobClient: Total time spent by all maps waiting after reserving slots (ms)=0
13/07/09 14:40:27 INFO mapred.JobClient: Launched reduce tasks=1
13/07/09 14:40:27 INFO mapred.JobClient: SLOTS_MILLIS_REDUCE=9452
13/07/09 14:40:27 INFO mapred.JobClient: File Input Format Counters
13/07/09 14:40:27 INFO mapred.JobClient: Bytes Read=10954
13/07/09 14:40:27 INFO mapred.JobClient: File Output Format Counters
13/07/09 14:40:27 INFO mapred.JobClient: Bytes Written=9146
13/07/09 14:40:27 INFO mapred.JobClient: FileSystemCounters
13/07/09 14:40:27 INFO mapred.JobClient: HDFS_BYTES_READ=11486
13/07/09 14:40:27 INFO mapred.JobClient: FILE_BYTES_WRITTEN=123889
13/07/09 14:40:27 INFO mapred.JobClient: FILE_BYTES_READ=11096
13/07/09 14:40:27 INFO mapred.JobClient: HDFS_BYTES_WRITTEN=9146
13/07/09 14:40:27 INFO mapred.JobClient: Map-Reduce Framework
13/07/09 14:40:27 INFO mapred.JobClient: Virtual memory (bytes) snapshot=4761595904
13/07/09 14:40:27 INFO mapred.JobClient: Reduce input groups=207
13/07/09 14:40:27 INFO mapred.JobClient: Combine output records=288
13/07/09 14:40:27 INFO mapred.JobClient: Map output records=481
13/07/09 14:40:27 INFO mapred.JobClient: CPU time spent (ms)=2390
```

Figure 18 - WordCount MapReduce job running

4. Now review the output of step 3:

```
hadoop fs -ls output
```



```
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/HDFS> hadoop fs -ls output
Found 3 items
-rw-r--r-- 1 biadmin supergroup          0 2013-07-09 14:40 /user/biadmin/output/_SUCCESS
drwxr-xr-x - biadmin supergroup          0 2013-07-09 14:40 /user/biadmin/output/_logs
-rw-r--r-- 1 biadmin supergroup    9146 2013-07-09 14:40 /user/biadmin/output/part-r-00000
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/HDFS>
```

Figure 19 - MapReduce result files





```
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/PigHiveJaql
File Edit View Terminal Help
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/HDFS> cd /home/biadmin/bootcamp/input/lab01_HadoopCore/PigHiveJaql/
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/PigHiveJaql> head -5 googlebooks-1988.csv
#      1988      94000      45770      9585
$0.000 1988      4          4          2
$0.0006 1988      3          3          3
$0.0027 1988      1          1          1
$0.003  1988      4          4          2
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/PigHiveJaql>
```

Figure 21 – Googlebooks-1988.csv file

The columns these data represent are the word, the year, the number of occurrences of that word in the corpus, the number of pages on which that word appeared, and the number of books in which that word appeared.

2. Copy the data file into HDFS.

```
hadoop fs -put googlebooks-1988.csv pighivejaql/googlebooks-1988.csv
```

Note that directory `/user/biadmin/pighivejaql` is created automatically for you when the above command is executed.

3. Start pig. If it has not been added to the PATH, you can add it, or switch to the `$PIG_HOME/bin` directory

```
cd $PIG_HOME/bin
```

```
./pig
```



```
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/PigHiveJaql
File Edit View Terminal Help
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/PigHiveJaql> cd $PIG_HOME/bin
biadmin@imtebil:/opt/ibm/biginsights/pig/bin> ./pig
grunt>
```

Figure 22 – Pig command line

4. We are going to use a Pig UDF to compute the absolute value of each integer. The UDF is located inside the `piggybank.jar` file (This jar file was created from the source, following the instructions in <https://wiki.apache.org/confluence/display/PIG/PiggyBank>, and copied to the `piggybank` directory). We use the `REGISTER` command to load this jar file:

```
REGISTER /opt/ibm/biginsights/pig/contrib/piggybank/java/piggybank.jar;
```

The first step in processing the data is to `LOAD` it.

- Execute the step below to load data.

```
records = LOAD 'pighivejaql/googlebooks-1988.csv' AS (word:chararray,  
year:int, wordcount:int, pagecount:int, bookcount:int);
```

This returns instantly. The processing is delayed until the data needs to be reported.

- To produce a histogram, we want to group by the length of the word:

```
grouped = GROUP records by  
org.apache.pig.piggybank.evaluation.string.LENGTH(word);
```

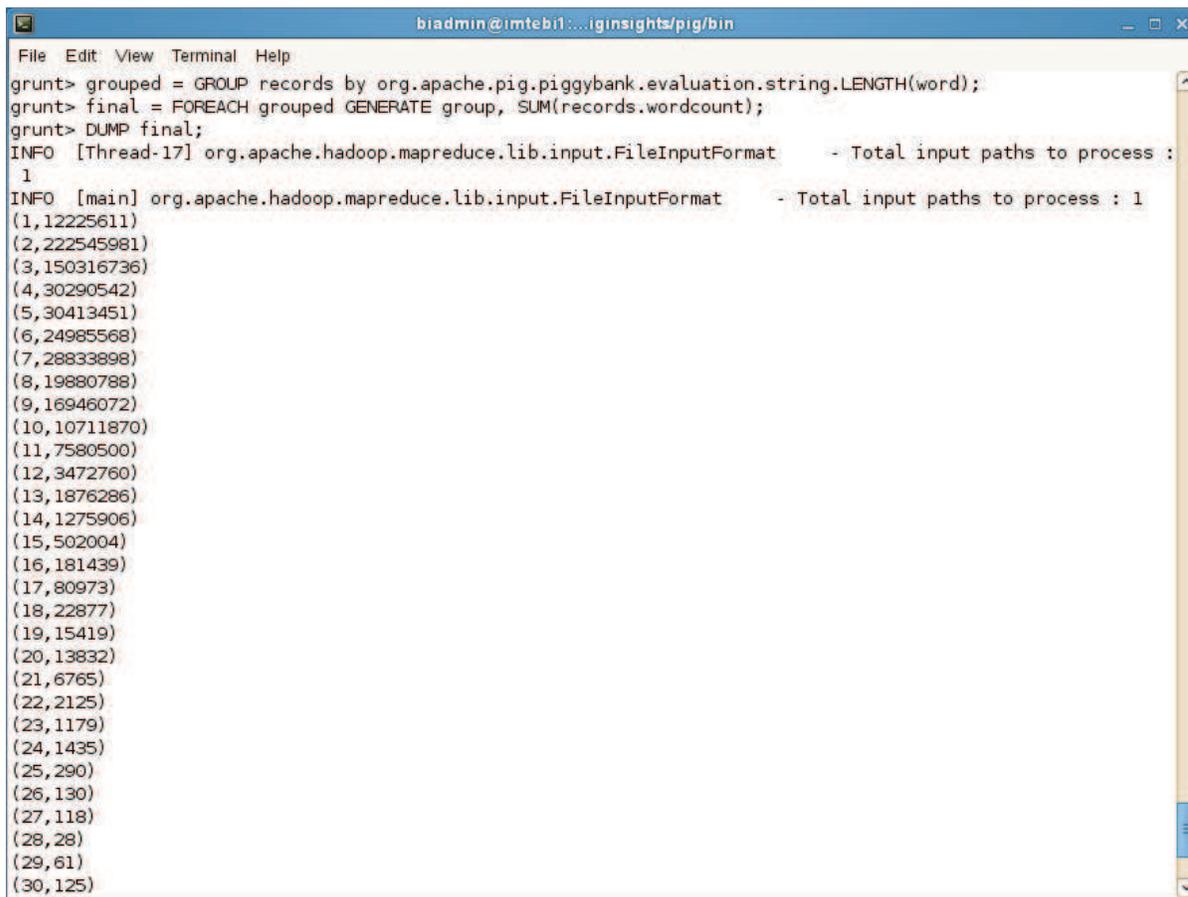
- Sum the word counts for each word length using the SUM function with the FOREACH GENERATE command.

```
final = FOREACH grouped GENERATE group, SUM(records.wordcount);
```

- Use the DUMP command to print the result to the console. This will cause all the previous steps to be executed.

```
DUMP final;
```

This should produce output like the following:



```
biadmin@imtebil...iginsights/pig/bin
File Edit View Terminal Help
grunt> grouped = GROUP records by org.apache.pig.piggybank.evaluation.string.LENGTH(word);
grunt> final = FOREACH grouped GENERATE group, SUM(records.wordcount);
grunt> DUMP final;
INFO [Thread-17] org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process :
1
INFO [main] org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
(1,12225611)
(2,222545981)
(3,150316736)
(4,30290542)
(5,30413451)
(6,24985568)
(7,28833898)
(8,19880788)
(9,16946072)
(10,10711870)
(11,7580500)
(12,3472760)
(13,1876286)
(14,1275906)
(15,502004)
(16,181439)
(17,80973)
(18,22877)
(19,15419)
(20,13832)
(21,6765)
(22,2125)
(23,1179)
(24,1435)
(25,290)
(26,130)
(27,118)
(28,28)
(29,61)
(30,125)
```

Figure 23 - Wordcount application output

9. Quit pig.

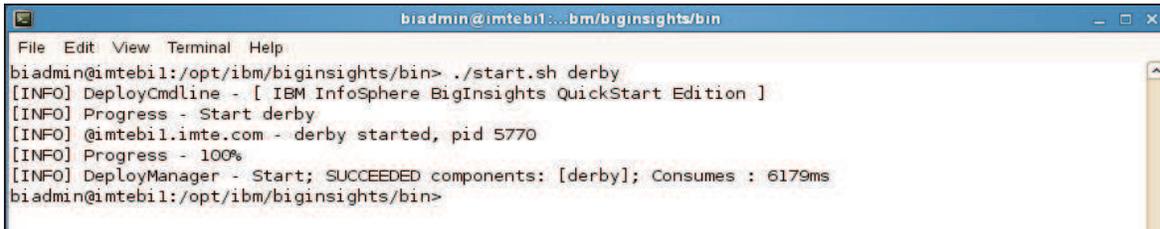
```
grunt> quit
```

## 7 Working with Hive

In this tutorial, we are going to use Hive to process the 1988 subset of the Google Books 1-gram records to produce a histogram of the frequencies of words of each length. A subset of this database (0.5 million records) has been stored in the file `googlebooks-1988.csv` under `/home/biadmin/bootcamp/input/lab01_HadoopCore/PigHiveJaql` directory.

1. Ensure the Apache Derby component is started. Apache Derby is the default database used as metastore in Hive. A quick way to verify if it is started, is to try to start it using:

```
start.sh derby
```



```
biadmin@imtebil:/opt/ibm/biginsights/bin
File Edit View Terminal Help
biadmin@imtebil:/opt/ibm/biginsights/bin> ./start.sh derby
[INFO] DeployCmdline - [ IBM InfoSphere BigInsights QuickStart Edition ]
[INFO] Progress - Start derby
[INFO] @imtebil.imte.com - derby started, pid 5770
[INFO] Progress - 100%
[INFO] DeployManager - Start; SUCCEEDED components: [derby]; Consumes : 6179ms
biadmin@imtebil:/opt/ibm/biginsights/bin>
```

Figure 24 - Start Apache Derby

2. Start hive interactively. Change the directory to the \$HIVE\_HOME/bin first, and execute from there using ./hive

```
cd $HIVE_HOME/bin
./hive
```

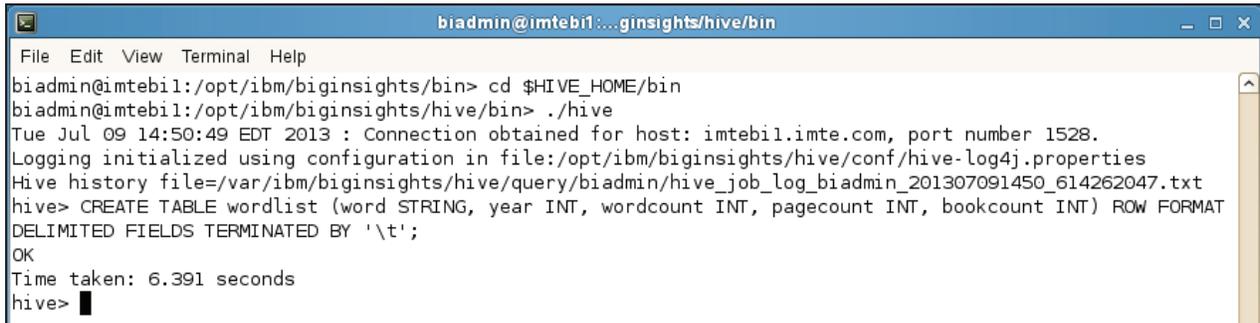


```
biadmin@imtebil:/opt/ibm/biginsights/hive/bin
File Edit View Terminal Help
biadmin@imtebil:/opt/ibm/biginsights/bin> cd $HIVE_HOME/bin
biadmin@imtebil:/opt/ibm/biginsights/hive/bin> ./hive
Tue Jul 09 14:50:49 EDT 2013 : Connection obtained for host: imtebil.imte.com, port number 1528.
Logging initialized using configuration in file:/opt/ibm/biginsights/hive/conf/hive-log4j.properties
Hive history file=/var/ibm/biginsights/hive/query/biadmin/hive_job_log_biadmin_201307091450_614262047.txt
hive>
```

Figure 25 - Start Apache Hive

3. Create a table called wordlist.

```
CREATE TABLE wordlist (word STRING, year INT, wordcount INT, pagecount INT,
bookcount INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t';
```



```
biadmin@imtebil:...ginsights/hive/bin
File Edit View Terminal Help
biadmin@imtebil:/opt/ibm/biginsights/bin> cd $HIVE_HOME/bin
biadmin@imtebil:/opt/ibm/biginsights/hive/bin> ./hive
Tue Jul 09 14:50:49 EDT 2013 : Connection obtained for host: imtebil.imte.com, port number 1528.
Logging initialized using configuration in file:/opt/ibm/biginsights/hive/conf/hive-log4j.properties
Hive history file=/var/ibm/biginsights/hive/query/biadmin/hive_job_log_biadmin_201307091450_614262047.txt
hive> CREATE TABLE wordlist (word STRING, year INT, wordcount INT, pagecount INT, bookcount INT) ROW FORMAT
DELIMITED FIELDS TERMINATED BY '\t';
OK
Time taken: 6.391 seconds
hive>
```

Figure 26 - Create wordlist table

4. Load the data from the googlebooks-1988.csv file into the wordlist table.

```
LOAD DATA LOCAL INPATH
'/home/biadmin/bootcamp/input/lab01_HadoopCore/PigHiveJaql/googlebooks-
1988.csv' OVERWRITE INTO TABLE wordlist;
```

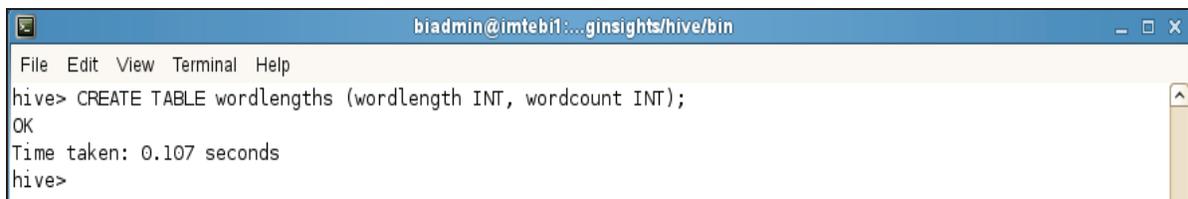


```
biadmin@imtebil:...ginsights/hive/bin
File Edit View Terminal Help
hive> LOAD DATA LOCAL INPATH '/home/biadmin/bootcamp/input/lab01_HadoopCore/PigHiveJaql/googlebooks-1988.csv
' OVERWRITE INTO TABLE wordlist;
Copying data from file:/home/biadmin/bootcamp/input/lab01_HadoopCore/PigHiveJaql/googlebooks-1988.csv
Copying file: file:/home/biadmin/bootcamp/input/lab01_HadoopCore/PigHiveJaql/googlebooks-1988.csv
Loading data to table default.wordlist
Deleted hdfs://imtebil.imte.com:9000/biginsights/hive/warehouse/wordlist
OK
Time taken: 1.011 seconds
hive>
```

Figure 27 - Load data into wordcount table

5. Create a table named wordlengths to store the counts for each word length for our histogram.

```
CREATE TABLE wordlengths (wordlength INT, wordcount INT);
```

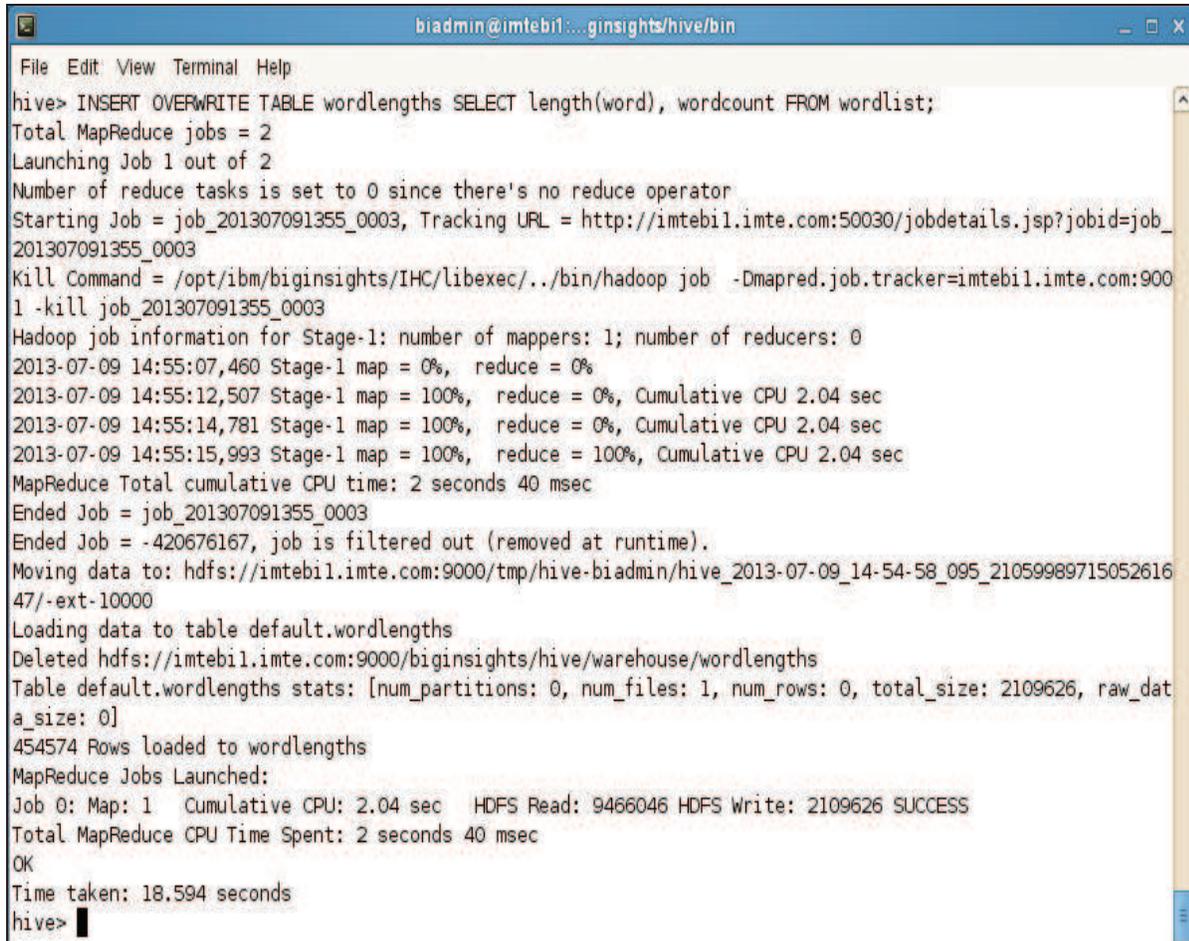


```
biadmin@imtebil:...ginsights/hive/bin
File Edit View Terminal Help
hive> CREATE TABLE wordlengths (wordlength INT, wordcount INT);
OK
Time taken: 0.107 seconds
hive>
```

Figure 28 - Create wordlength table

6. Fill the wordlengths table with word length data from the wordlist table calculated with the length function.

```
INSERT OVERWRITE TABLE wordlengths SELECT length(word), wordcount FROM wordlist;
```

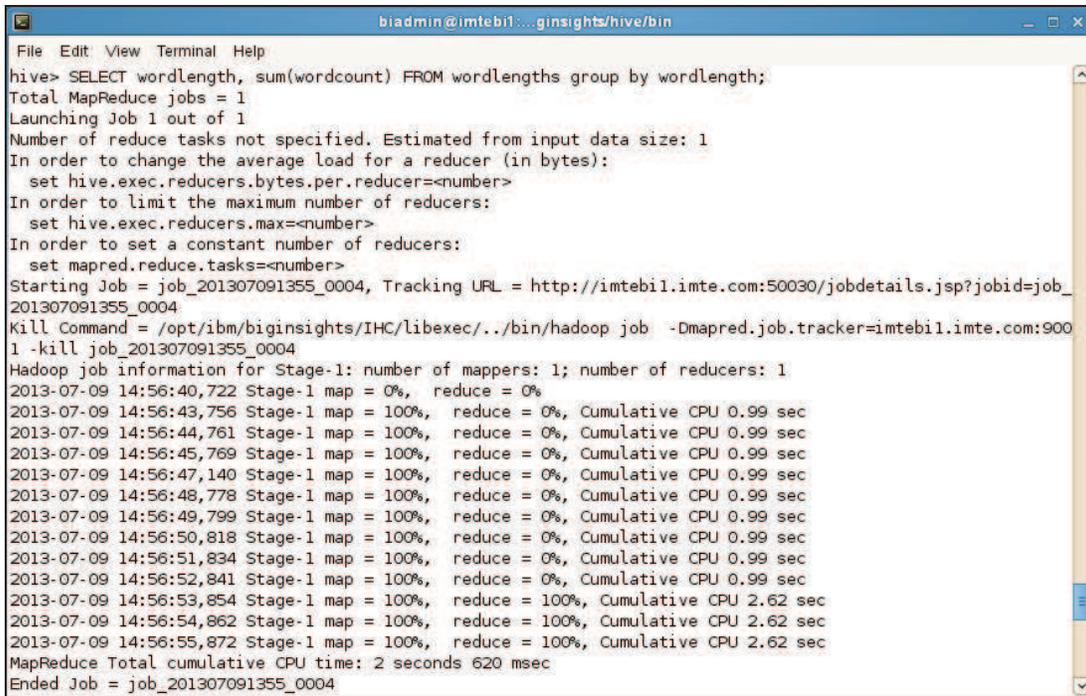


```
biadmin@imtebil...ginsights/hive/bin
File Edit View Terminal Help
hive> INSERT OVERWRITE TABLE wordlengths SELECT length(word), wordcount FROM wordlist;
Total MapReduce jobs = 2
Launching Job 1 out of 2
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_201307091355_0003, Tracking URL = http://imtebil.imte.com:50030/jobdetails.jsp?jobid=job_201307091355_0003
Kill Command = /opt/ibm/biginsights/IHC/libexec/./bin/hadoop job -Dmapred.job.tracker=imtebil.imte.com:9000 1 -kill job_201307091355_0003
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2013-07-09 14:55:07,460 Stage-1 map = 0%, reduce = 0%
2013-07-09 14:55:12,507 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.04 sec
2013-07-09 14:55:14,781 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.04 sec
2013-07-09 14:55:15,993 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 2.04 sec
MapReduce Total cumulative CPU time: 2 seconds 40 msec
Ended Job = job_201307091355_0003
Ended Job = -420676167, job is filtered out (removed at runtime).
Moving data to: hdfs://imtebil.imte.com:9000/tmp/hive-biadmin/hive_2013-07-09_14-54-58_095_2105998971505261647/-ext-10000
Loading data to table default.wordlengths
Deleted hdfs://imtebil.imte.com:9000/biginsights/hive/warehouse/wordlengths
Table default.wordlengths stats: [num_partitions: 0, num_files: 1, num_rows: 0, total_size: 2109626, raw_data_size: 0]
454574 Rows loaded to wordlengths
MapReduce Jobs Launched:
Job 0: Map: 1 Cumulative CPU: 2.04 sec HDFS Read: 9466046 HDFS Write: 2109626 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 40 msec
OK
Time taken: 18.594 seconds
hive>
```

Figure 29 - Fill wordlengths table

7. Produce the histogram by summing the word counts grouped by word length.

```
SELECT wordlength, sum(wordcount) FROM wordlengths group by wordlength;
```



```
biadmin@imtebil:...ginsights/hive/bin
File Edit View Terminal Help
hive> SELECT wordlength, sum(wordcount) FROM wordlengths group by wordlength;
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapred.reduce.tasks=<number>
Starting Job = job_201307091355_0004, Tracking URL = http://imtebil.imte.com:50030/jobdetails.jsp?jobid=job_201307091355_0004
Kill Command = /opt/ibm/biginsights/IHC/libexec/./bin/hadoop job -Dmapred.job.tracker=imtebil.imte.com:9000 1 -kill job_201307091355_0004
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2013-07-09 14:56:40,722 Stage-1 map = 0%, reduce = 0%
2013-07-09 14:56:43,756 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 0.99 sec
2013-07-09 14:56:44,761 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 0.99 sec
2013-07-09 14:56:45,769 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 0.99 sec
2013-07-09 14:56:47,140 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 0.99 sec
2013-07-09 14:56:48,778 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 0.99 sec
2013-07-09 14:56:49,799 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 0.99 sec
2013-07-09 14:56:50,818 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 0.99 sec
2013-07-09 14:56:51,834 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 0.99 sec
2013-07-09 14:56:52,841 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 0.99 sec
2013-07-09 14:56:53,854 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 2.62 sec
2013-07-09 14:56:54,862 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 2.62 sec
2013-07-09 14:56:55,872 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 2.62 sec
MapReduce Total cumulative CPU time: 2 seconds 620 msec
Ended Job = job_201307091355_0004
```

Figure 30 - Executing MapReduce job

- 8. Quit hive.

```
quit;
```



```
biadmin@imtebil:...ginsights/hive/bin
File Edit View Terminal Help
Time taken: 22.421 seconds
hive> quit;
biadmin@imtebil:/opt/ibm/biginsights/hive/bin>
```

Figure 31 - Quit Apache Hive

## 8 Working with Jaql

In this tutorial, we are going to use Jaql to process the 1988 subset of the Google Books 1-gram records to produce a histogram of the frequencies of words of each length. A subset of this database (0.5 million records) has been stored in the file googlebooks-1988.csv under /home/biadmin/bootcamp/input/lab01\_HadoopCore/PigHiveJaql directory.

- 1. Let us examine the format of the Google Books 1-gram records:

```
cd /home/biadmin/bootcamp/input/lab01_HadoopCore/PigHiveJaql  
  
head -5 googlebooks-1988.del
```



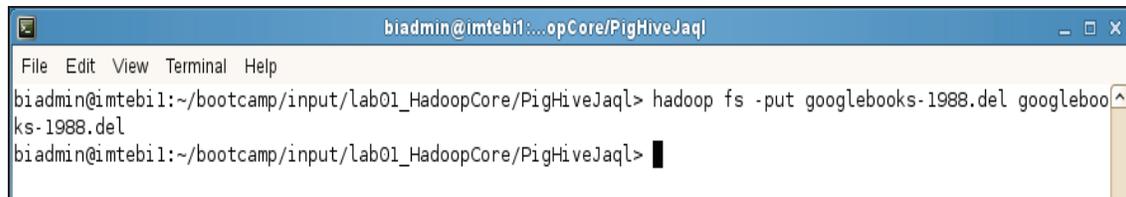
```
biadmin@imtebil:...opCore/PigHiveJaql  
File Edit View Terminal Help  
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/PigHiveJaql> cd /home/biadmin/bootcamp/input/lab01_HadoopCore/PigHiveJaql  
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/PigHiveJaql> head -5 googlebooks-1988.del  
#,1988,94000,45770,9585  
$0.000,1988,4,4,2  
$0.0006,1988,3,3,3  
$0.0027,1988,1,1,1  
$0.003,1988,4,4,2  
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/PigHiveJaql>
```

Figure 32 - googlebooks-1998.csv file format

The columns these data represent are the word, the year, the number of occurrences of that word in the corpus, the number of pages on which that word appeared, and the number of books in which that word appeared.

2. Copy the googlebooks-1988.del file to HDFS.

```
hadoop fs -put googlebooks-1988.del googlebooks-1988.del
```



```
biadmin@imtebil:...opCore/PigHiveJaql  
File Edit View Terminal Help  
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/PigHiveJaql> hadoop fs -put googlebooks-1988.del googlebooks-1988.del  
biadmin@imtebil:~/bootcamp/input/lab01_HadoopCore/PigHiveJaql>
```

Figure 33 - Copy googlebooks-1988.csv file to HDFS

3. Change directory to \$JAQL\_HOME/bin, and then execute ./jaqlshell to start the JaqlShell.

```
cd $JAQL_HOME/bin  
./jaqlshell
```



Figure 34 - Start Jaqlsell

4. Read the comma delimited file from HDFS. Note that this operation might take a few minutes to complete.

```
$wordlist = read(del("googlebooks-1988.del", { schema: schema { word: string, year: long, wordcount: long, pagecount: long, bookcount: long } }));
```



Figure 35 - Read googlebooks-1988.del from HDFS

5. Transform each word into its length by applying the `strLen` function.

```
$wordlengths = $wordlist -> transform { wordlength: strLen($.word), wordcount: $.wordcount };
```

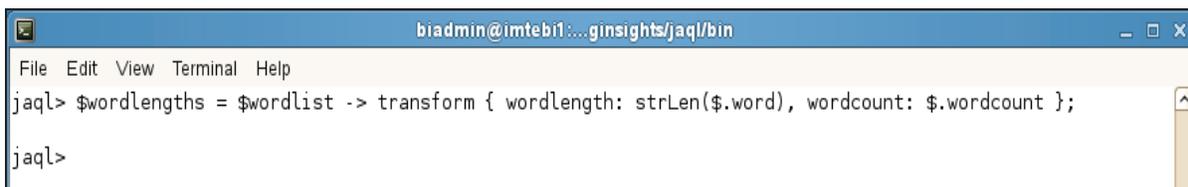
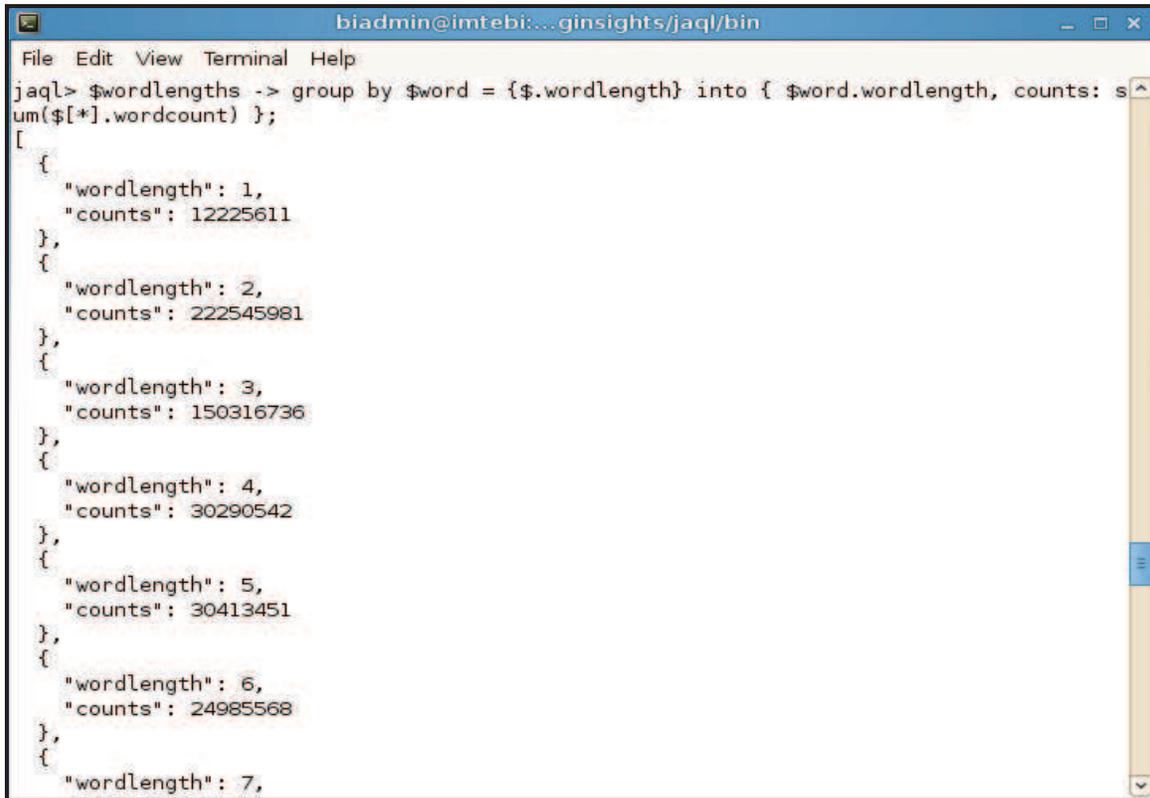


Figure 36 - Applying `strLen()` function

6. Produce the histogram by summing the word counts grouped by word length.

```
$wordlengths -> group by $word = { $.wordlength } into { $word.wordlength, counts: sum($[*].wordcount) };
```

This should produce output like the following:



A terminal window titled "biadmin@imtebi:...ginsights/jaql/bin" showing the execution of a Jaql command. The command is: `jaql> $wordlengths -> group by $word = {$word.wordlength} into { $word.wordlength, counts: sum($[*].wordcount) };`. The output is a JSON array of objects, each representing a word length and its count. The visible output is:

```
[
  {
    "wordlength": 1,
    "counts": 12225611
  },
  {
    "wordlength": 2,
    "counts": 222545981
  },
  {
    "wordlength": 3,
    "counts": 150316736
  },
  {
    "wordlength": 4,
    "counts": 30290542
  },
  {
    "wordlength": 5,
    "counts": 30413451
  },
  {
    "wordlength": 6,
    "counts": 24985568
  },
  {
    "wordlength": 7,
```

Figure 37 - Wordcount output

7. Quit Jaql.

```
quit;
```



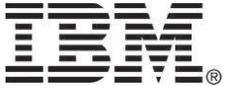
A terminal window titled "biadmin@imtebi:...ginsights/jaql/bin" showing the execution of the `quit;` command. The output is: `Shutting down jaql.` followed by the shell prompt `biadmin@imtebi1:/opt/ibm/biginsights/jaql/bin>`.

Figure 38 - Quit Jaql shell

## 9 Summary

You have just completed Lab 1 which focused on the basics of the Hadoop platform, including HDFS, MapReduce, Pig, Hive, and Jaql. You should now know how to perform the following basic tasks on the platform:

- Start/Stop the Hadoop components
- Interact with the data in the Hadoop Distributed File System (HDFS)
- Navigate within HDFS
- Run MapReduce programs
- Use Pig, Hive, and Jaql languages to interact with Hadoop



---

© Copyright IBM Corporation 2013  
All Rights Reserved.

IBM Canada  
8200 Warden Avenue  
Markham, ON  
L6G 1C7  
Canada

IBM, the IBM logo, ibm.com and Tivoli are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at [ibm.com/legal/copytrade.shtml](http://ibm.com/legal/copytrade.shtml)

Other company, product and service names may be trademarks or service marks of others.

References in this publication to IBM products and services do not imply that IBM intends to make them available in all countries in which IBM operates.

No part of this document may be reproduced or transmitted in any form without written permission from IBM Corporation.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT.

IBM products are warranted according to the terms and conditions of the agreements (e.g. IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided.