

## EXP 4: Create UDF (User Defined Functions) in Apache Pigand

### execute it in MapReduce / HDFS mode

#### AIM:

To create UDF in Apache Pig and execute it in MapReduce/HDFS mode.

#### PROCEDURE:

##### Pig Download and installation:

##### 1. Download Pig:

Download Pig from “<https://downloads.apache.org/pig/pig-0.17.0/>”

**Index of /pig/pig-0.17.0**

| Name                      | Last modified    | Size | Description |
|---------------------------|------------------|------|-------------|
| Parent Directory          |                  | -    |             |
| README.txt                | 2017-06-16 18:10 | 1.4K |             |
| RELEASE_NOTES.txt         | 2017-06-16 18:10 | 1.9K |             |
| pig-0.17.0-src.tar.gz     | 2017-06-16 18:11 | 15M  |             |
| pig-0.17.0-src.tar.gz.asc | 2017-06-16 18:11 | 488  |             |
| pig-0.17.0-src.tar.gz.md5 | 2017-06-16 18:11 | 56   |             |
| pig-0.17.0.tar.gz         | 2017-06-16 18:10 | 220K |             |
| pig-0.17.0.tar.gz.asc     | 2017-06-16 18:11 | 488  |             |
| pig-0.17.0.tar.gz.md5     | 2017-06-16 18:11 | 52   |             |

##### 2. Add the environment variable for Pig:

Edit System Variable

Variable name: PIG\_HOME

Variable value: C:\pig-0.17.0

Browse Directory... Browse File... OK Cancel

Edit environment variable

C:\Program Files (x86)\Common Files\Oracle\Java\javapath  
 C:\Program Files (x86)\Common Files\Oracle\Java\javapath  
 C:\Program Files\Python311\Scripts\  
 C:\Program Files\Python311\  
 %SystemRoot%\system32  
 %SystemRoot%  
 %SystemRoot%\System32\Wbem  
 %SYSTEMROOT%\System32\WindowsPowerShell\v1.0\  
 %SYSTEMROOT%\System32\OpenSSH\  
 C:\Users\Admin\AppData\Roaming\Python\Python311\Scripts  
 C:\Program Files\nodejs\  
 D:\Admin\Git\cmd  
 C:\Java\jdk-1.8\bin  
 C:\Hadoop\bin  
 C:\Hadoop/sbin  
 C:\Python39\  
 %PIG\_HOME%\bin

New Edit Browse... Delete Move Up Move Down Edit text... OK Cancel

3. Go to C:\pig-0.17.0\bin and open pig (Windows Command Script)

```
set HADOOP_BIN_PATH=%HADOOP_HOME%\libexec
```

4. Open Windows Powershell and type “pig –x local” and check whether pig grunt appears.

**Pig is successfully installed.**

**Create UDF:**

**1. Start Hadoop services:**

Open command prompt as an administrator

```
start-dfs.cmd
start-yarn.cmd
```

2. Open the browser and go to the URL “localhost:9870”

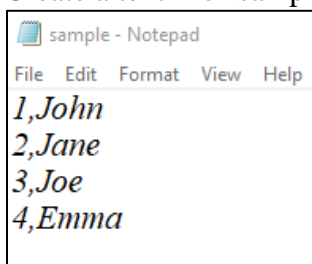
The screenshot displays the Hadoop Overview page for 'localhost:9800' (active). The page has a green header with navigation tabs: Hadoop, Overview, Datanodes, Datanode Volume Failures, Snapshot, Startup Progress, and Utilities. The main content area shows the Overview section with a table of cluster details:

|                |  |
|----------------|--|
| Started:       | Mon Aug 26 17:20:52 +0530 2024   |
| Version:       | 3.3.6, r1be78238728da9266a4f98195058f08fd012bf9c                                   |
| Compiled:      | Sun Jun 18 13:52:00 +0530 2023 by ubuntu from (HEAD detached at release-3.3.6-RC1) |
| Cluster ID:    | CID-a23ce25d-ee9d-4000-ac1f-044f436c4c8a   |
| Block Pool ID: | BP-934656018-192.168.56.1-1723971050909  |

Below the table is a Summary section with the following information:

- Security is off.
- Safemode is off.
- 119 files and directories, 51 blocks (51 replicated blocks, 0 erasure coded block groups) = 170 total filesystem object(s).
- Heap Memory used 85.02 MB of 294.5 MB Heap Memory. Max Heap Memory is 889 MB.
- Non Heap Memory used 70.43 MB of 71.88 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.
- Configured Capacity: 118.63 GB

3. Create a text file “sample.txt”:



4. Create a Directory in HDFS and copy the Input File to HDFS

```
hdfs dfs -mkdir /UDF
```

```
hadoop fs -put C:/Users/user/Documents/Pig/sample.txt /UDF
```

```
C:\hadoop\sbin>hdfs dfs -mkdir /UDF
```

```
C:\hadoop\sbin>hadoop fs -put C:/Users/user/Documents/Pig/sample.txt /UDF
```

### 5. Create a Python file “uppercase\_udf.py”:

```
# uppercase_udf.py
def uppercase(text):
    return text.upper()
if __name__ == "__main__":
    import sys
    for line in sys.stdin:
        line = line.strip()
        result = uppercase(line)
        print(result)
```

### 6. Create a Directory in HDFS and copy the Input File to HDFS

```
hdfs dfs -mkdir /UDF/udfs
```

```
hadoop fs -put C:/Users/user/Documents/Pig/Uppercase_udf.py /UDF/udfs
```

```
C:\hadoop\sbin>hdfs dfs -mkdir /UDF/udfs
```

```
C:\hadoop\sbin>hadoop fs -put C:/Users/user/Documents/Pig /Uppercase_udf.py /UDF/udfs
```

### 7. Create pig file “UDF.pig”:

```
UDF - Notepad
File Edit Format View Help
-- udf_example.pig

-- Register the Python UDF script
REGISTER 'hdfs:///UDF/udfs/Uppercase_udf.py' USING jython AS udf;

-- Load some data
data = LOAD 'hdfs:///UDF/sample.txt' USING PigStorage(',') AS (id:int, name:chararray);

-- Use the Python UDF to convert names to uppercase
uppercased_data = FOREACH data GENERATE id, udf.uppercase(name) AS uppercase_name;

-- Store the result
STORE uppercased_data INTO 'hdfs:///UDF/output' USING PigStorage(',');
```

### 8. Execute Pig file

```
pig -x mapreduce C:/Users/user/Documents/Pig/UDF.pig
```

```
2024-08-26 19:03:11,501 [JobControl] INFO org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input files to process : 1
2024-08-26 19:03:11,502 [JobControl] INFO org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1
2024-08-26 19:03:11,540 [JobControl] INFO org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths (combined) to process : 1
2024-08-26 19:03:12,073 [JobControl] INFO org.apache.hadoop.mapreduce.JobSubmitter - number of splits:1
```

```
C:\hadoop\sbin>pig -x mapreduce C:/Users/user/Documents/Pig/UDF.pig
2024-09-08 18:57:35,502 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL
2024-09-08 18:57:35,502 INFO pig.ExecTypeProvider: Trying ExecType : MAPREDUCE
2024-09-08 18:57:35,502 INFO pig.ExecTypeProvider: Picked MAPREDUCE as the ExecType
2024-09-08 18:57:36,298 [main] INFO org.apache.pig.Main - Apache Pig version 0.17.0 (r1797386) compiled Jun 02 2017, 15:41:58
2024-09-08 18:57:36,298 [main] INFO org.apache.pig.Main - Logging error messages to: C:\hadoop\logs\pig_1725802056283.log
2024-09-08 18:57:37,033 [main] INFO org.apache.pig.impl.util.Utils - Default bootstrap file C:\Users\user/.pigbootstrap not found
2024-09-08 18:57:37,142 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2024-09-08 18:57:37,142 [main] INFO org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop file system at: hdfs://localhost:9000
2024-09-08 18:57:38,095 [main] INFO org.apache.pig.PigServer - Pig Script ID for the session: PIG-UDF.pig-8ebf5171-17af-4a4d-9312-0f483160f591
2024-09-08 18:57:38,095 [main] WARN org.apache.pig.PigServer - ATS is disabled since yarn.timeline-service.enabled set to false
2024-09-08 18:57:40,377 [main] ERROR org.apache.pig.tools.grunt.Grunt - ERROR 2997: Encountered IOException. Call From DESKTOP-DKU83QW/192.168.56.1 to localhost:9000 failed on connection except
tion: java.net.ConnectException: Connection refused: no further information; For more details see: http://wiki.apache.org/hadoop/ConnectionRefused
Details at logfile: C:\hadoop\logs\pig_1725802056283.log
2024-09-08 18:57:40,455 [main] INFO org.apache.pig.Main - Pig script completed in 5 seconds and 156 milliseconds (5156 ms)

C:\hadoop\sbin>hdfs dfs -cat /UDF/output/part-m-00000
1,JOHN
2,JANE
3,JOE
4,EMMA
```

## 9. View the Output

```
hdfs dfs -ls /UDF/output
```

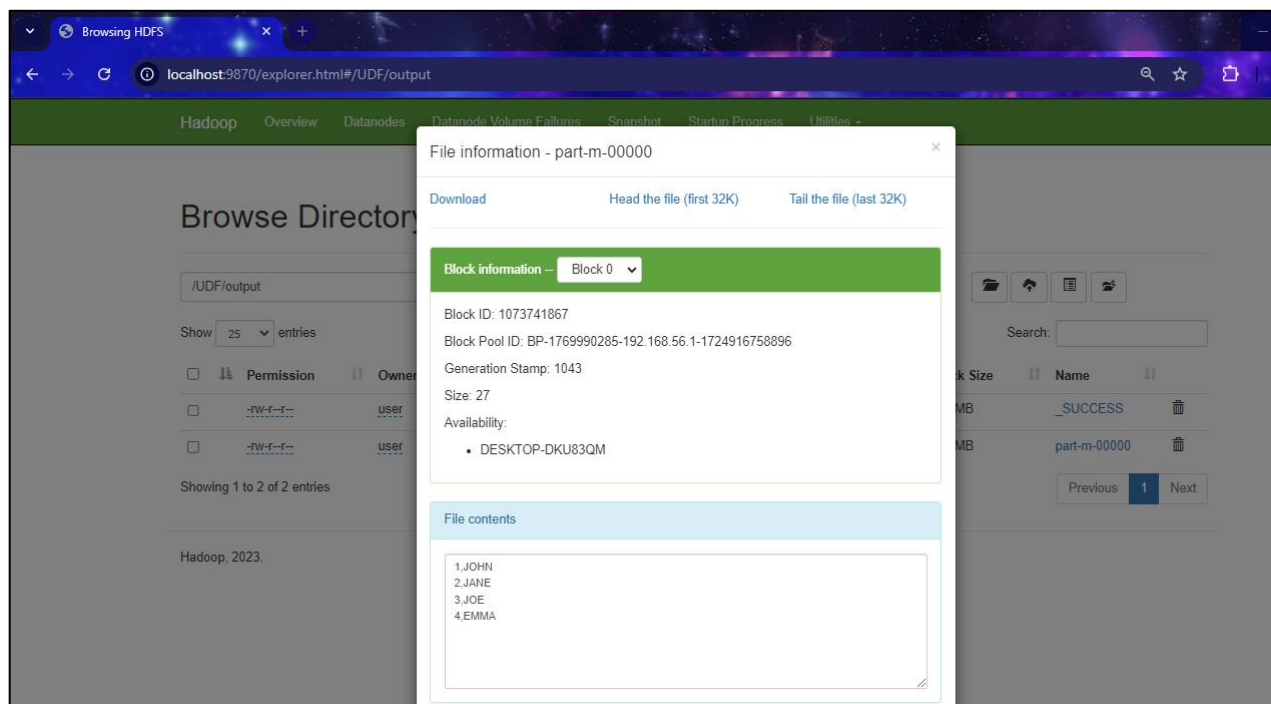
```
hdfs dfs -cat /UDF/output/part-m-00000
```

```
C:\hadoop\sbin>hdfs dfs -ls /UDF/output
Found 2 items
-rw-r--r--  1 user supergroup      0 2024-08-29 22:12 /UDF/output/_SUCCESS
-rw-r--r--  1 user supergroup    27 2024-08-29 22:12 /UDF/output/part-m-00000

C:\hadoop\sbin>hdfs dfs -cat /UDF/output/part-m-00000
1,JOHN
2,JANE
3,JOE
4,EMMA
```

10. Once the map reduce operations are performed successfully, the output will be present in the specified directory.

“/UDF/output/part-m-00000”



## RESULT:

Thus, UDF in Apache Pig has been created and executed in MapReduce/HDFS mode successfully.