# Hortonworks Data Platform for W

## System Administration Guide

(Aug 13, 2013)

docs.hortonworks.com

# Hortonworks Data Platform for Windows : System Administration Guide



The Hortonworks Data Platform, powered by Apache Hadoop, is a massively scalable and 100% open source platform for storing, processing and analyzing large volumes of data. It is designed to deal with data from many sources and formats in a very quick, easy and cost-effective manner. The Hortonworks Data Platform consists of the essential set of Apache Hadoop projects including MapReduce, Hadoop Distributed File System (HDFS), HCatalog, Pig, Hive, HBase, Zookeeper and Ambari. Hortonworks is the major contributor of code and patches to many of these projects. These projects have been integrated and tested as part of the Hortonworks Data Platform release process and installation and configuration tools have also been included.

Unlike other providers of platforms built using Apache Hadoop, Hortonworks contributes 100% of our code back to the Apache Software Foundation. The Hortonworks Data Platform is Apache-licensed and completely open source. We sell only expert technical support, training and partner-enablement services. All of our technology is, and will remain free and open source.

Please visit the Hortonworks Data Platform page for more information on Hortonworks technology. For more information on Hortonworks services, please visit either the Support or Training page. Feel free to Contact Us directly to discuss your specific needs.

# Table of Contents

# 1. Configuring HDFS Compression

This document is intended for system administrators who need to configure HDFS compression on Windows platform.

Windows supports `GzipCodec`, `DefaultCodec`, and `BZip2Codec`. Typically, `GzipCodec` is popularly used for HDFS compression.

Ensure that `zlib1.dll` is installed in the above mentioned locations on all the nodes of the cluster.

Use the following instructions to use GZipCodec

- **Option I:** To use GzipCodec with a one-time only job:

  1. On the NamNode host machine, execute the following commands as `hdfs` user:

     ```
     hadoop jar hadoop-examples-1.1.0-SNAPSHOT.jar sort "-Dmapred.compress.map.
     output=true" "-Dmapred.map.output.compression.codec=org.apache.hadoop.
     io.compress.GzipCodec" "-Dmapred.output.compress=true" "-Dmapred.output.
     compression.codec=org.apache.hadoop.io.compress.GzipCodec" -outKey org.
     apache.hadoop.io.Text -outValue org.apache.hadoop.io.Text input output
     ```

- **Option II:** To enable GzipCodec as the default compression:

  1. Edit the `core-site.xml` file on the NameNode host machine:

     ```
     <property>
       <name>io.compression.codecs</name>
        <value>org.apache.hadoop.io.compress.GzipCodec,org.apache.hadoop.io.
     compress.DefaultCodec,org.apache.hadoop.io.compress.BZip2Codec</value>
        <description>A list of the compression codec classes that can be used
      for compression/decompression.</description>
     </property>
     ```

  2. Edit `mapred-site.xml` file on the JobTracker host machine:

     ```
     <property>
       <name>mapred.compress.map.output</name>
       <value>true</value>
     </property>

     <property>
        <name>mapred.map.output.compression.codec</name>
        <value>org.apache.hadoop.io.compress.GzipCodec</value>
     </property>

     <property>
        <name>mapred.output.compression.type</name>
        <value>BLOCK</value>
     </property>
     ```

  3. [Optional] - Enable the following two configuration parameters to enable job output compression.

     Edit `mapred-site.xml` file on the JobTracker host machine:

```
<property>
  <name>mapred.output.compress</name>
  <value>true</value>
</property>

<property>
   <name>mapred.output.compression.codec</name>
   <value>org.apache.hadoop.io.compress.GzipCodec</value>
</property>
```

4. Restart the cluster using instructions provided here.

# 2. WebHDFS Administrator Guide

Use the following instructions to set upo WebHDFS:

1. Set up WebHDFS.

   Add the following property to the `hdfs-site.xml` file:

```
<property>
   <name>dfs.webhdfs.enabled</name>
   <value>true</value>
</property>
```

2. [Optiona] - If running a secure cluster, follow the steps listed below.

   a. Create an HTTP service user principal using the command given below:

```
kadmin: addprinc -randkey HTTP/$<Fully_Qualified_Domain_Name>@
$<Realm_Name>.COM
```

   where:

   - `Fully_Qualified_Domain_Name`: Host where NameNode is deployed
   - `Realm_Name`: Name of your Kerberos realm

   b. Create keytab files for the HTTP principals.

```
kadmin: xst -norandkey -k /etc/security/spnego.service.keytab HTTP/
$<Fully_Qualified_Domain_Name>
```

   c. Verify that the keytab file and the principal are associated with the correct service.

```
klist –k -t /etc/security/spnego.service.keytab
```

   d. Add the following properties to the `hdfs-site.xml` file.

```
<property>
   <name>dfs.web.authentication.kerberos.principal</name>
   <value>HTTP/$<Fully_Qualified_Domain_Name>@$<Realm_Name>.COM</value>
</property>
```

```
<property>
   <name>dfs.web.authentication.kerberos.keytab</name>
   <value>/etc/security/spnego.service.keytab</value>
</property>
```

   where:

   - `Fully_Qualified_Domain_Name`: Host where NameNode is deployed
   - `Realm_Name`: Name of your Kerberos realm

3. Restart the NameNode and DataNode services using the instructions provided here.