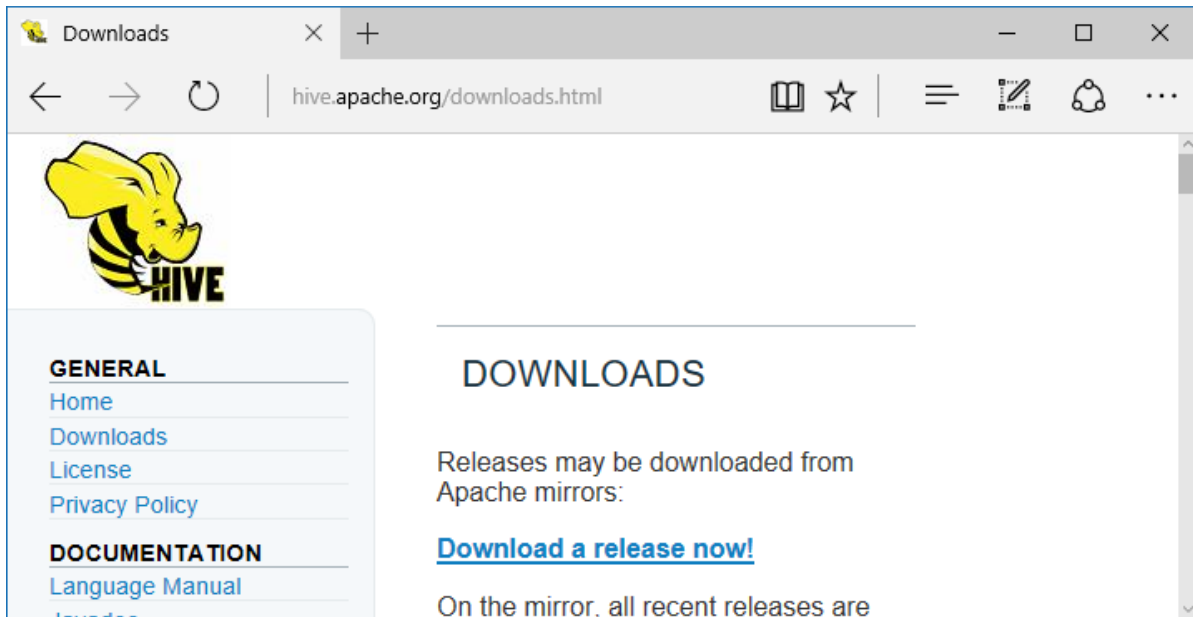


## HIVE\_Version\_2.1.0 INSTALLATION

1. Download HIVE, and EXTRACT to the folder of your choice

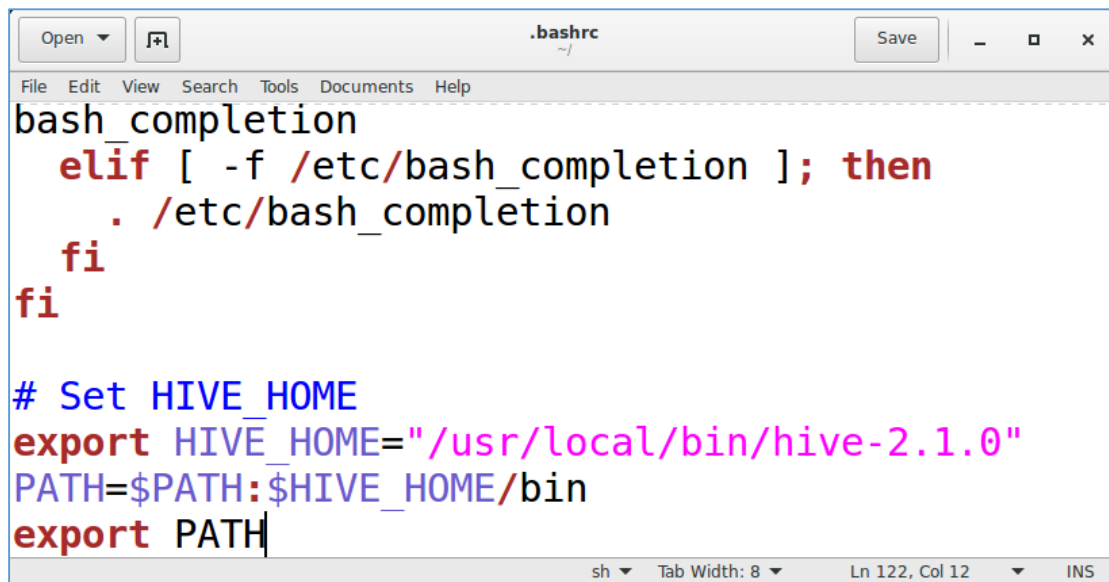


2. Setting Hive environment variable:

```
user@ubuntu:~$ sudo gedit ~/.bashrc
```

Copy and paste the following lines at end of the file

```
# Set HIVE_HOME
export HIVE_HOME="/usr/lib/hive/apache-hive-0.13.0-bin"
PATH=$PATH:$HIVE_HOME/bin
export PATH
```



3. Setting HADOOP\_PATH in HIVE config.sh

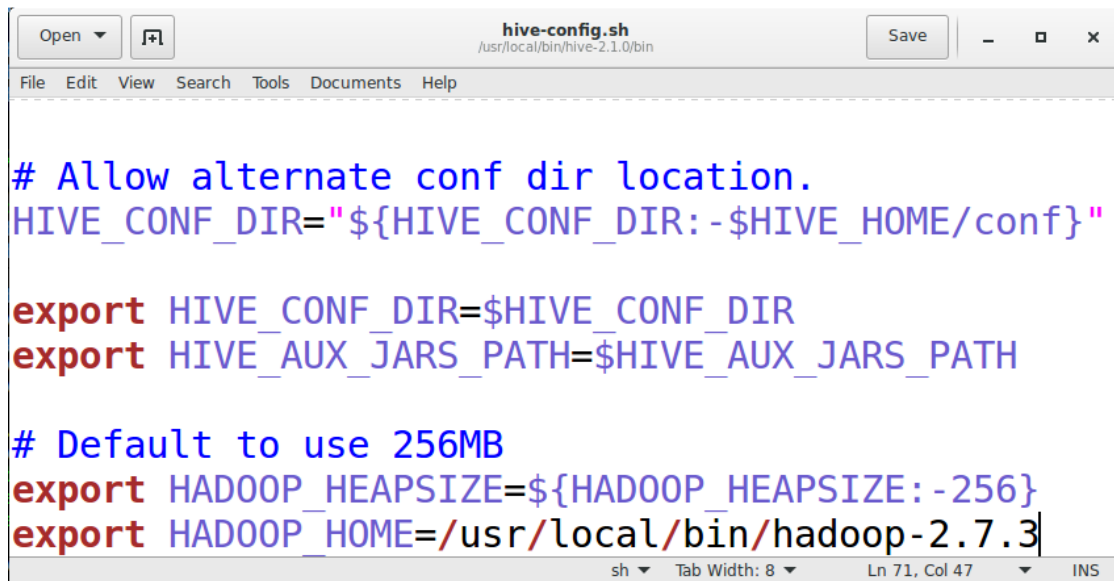
```
user@ubuntu:~$ cd /hive_install_dir/bin
user@ubuntu:~$ sudo gedit hive-config.sh
```

Go to the line where the following statements are written

```
# Allow alternate conf dir location.
HIVE_CONF_DIR="${HIVE_CONF_DIR:-$HIVE_HOME/conf}"
export HIVE_CONF_DIR=$HIVE_CONF_DIR
export HIVE_AUX_JARS_PATH=$HIVE_AUX_JARS_PATH
```

Below this write the following

```
export HADOOP_HOME=/usr/local/hadoop (write the path where hadoop file is there)
```



The screenshot shows a text editor window titled "hive-config.sh" with the following content:

```
# Allow alternate conf dir location.
HIVE_CONF_DIR="${HIVE_CONF_DIR:-$HIVE_HOME/conf}"

export HIVE_CONF_DIR=$HIVE_CONF_DIR
export HIVE_AUX_JARS_PATH=$HIVE_AUX_JARS_PATH

# Default to use 256MB
export HADOOP_HEAPSIZE=${HADOOP_HEAPSIZE:-256}
export HADOOP_HOME=/usr/local/bin/hadoop-2.7.3
```

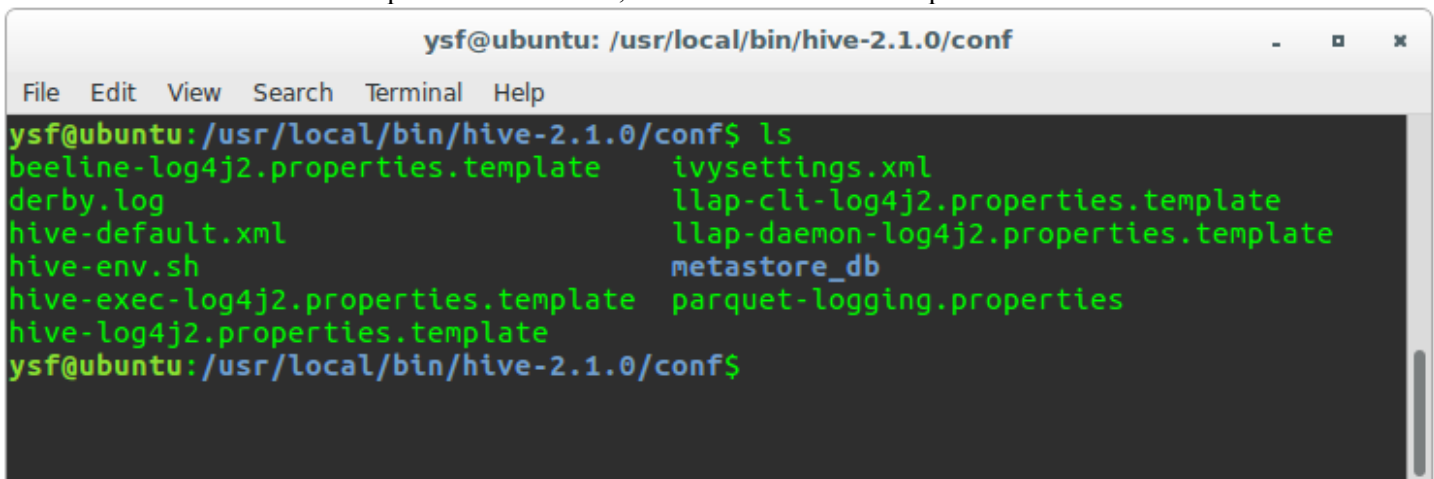
4. Create Hive directories within HDFS

```
hadoop fs -mkdir /hive/warehouse
```

Setting READ/WRITE permission for table

```
hadoop fs -chmod g+w /hive/warehouse
```

5. Rename hive-env.sh.template to hive-env.sh, and hive-default.xml.template to hive-default.xml



The screenshot shows a terminal window with the following output:

```
ysf@ubuntu: /usr/local/bin/hive-2.1.0/conf
ysf@ubuntu:/usr/local/bin/hive-2.1.0/conf$ ls
beeline-log4j2.properties.template  ivysettings.xml
derby.log                           llap-cli-log4j2.properties.template
hive-default.xml                    llap-daemon-log4j2.properties.template
hive-env.sh                         metastore_db
hive-exec-log4j2.properties.template  parquet-logging.properties
hive-log4j2.properties.template
ysf@ubuntu:/usr/local/bin/hive-2.1.0/conf$
```

6. Edit hive-env.sh

```
ysf@ubuntu: /usr/local/bin/hive-2.1.0/conf
File Edit View Search Terminal Help
GNU nano 2.5.3 File: hive-env.sh

# Set HADOOP_HOME to point to a specific hadoop install directory
HADOOP_HOME=/usr/local/bin/hadoop-2.7.3/

# Hive Configuration Directory can be controlled by:
# export HIVE_CONF_DIR=

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Linter ^_ Go To Line
```

7. Edit hive-default.xml

```
ysf@ubuntu: /usr/local/bin/hive-2.1.0/conf
File Edit View Search Terminal Help
GNU nano 2.5.3 File: hive-default.xml

</property>
<property>
  <name>hive.exec.stagingdir</name>
  <value>.hive-staging</value>
  <description>Directory name that will be created inside table locations in $
</property>
<property>
  <name>hive.exec.scratchdir</name>
  <value>/hive/tmp</value>
  <description>HDFS root scratch dir for Hive jobs which gets created with wr$
</property>
```

```
</property>
  <name>fs.har.impl</name>
  <value>org.apache.hadoop.hive.shims.HiveHarFileSystem</value>
  <description>The implementation for accessing Hadoop Archives. Note that th$
</property>
<property>
  <name>hive.metastore.warehouse.dir</name>
  <value>/hive/warehouse</value>
  <description>location of default database for the warehouse</description>
</property>

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

8. When you run hive from \$HIVE\_HOME/bin directory, you may get the following error:

```
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hive/lib/slf4j-log4j12-1.6.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
```

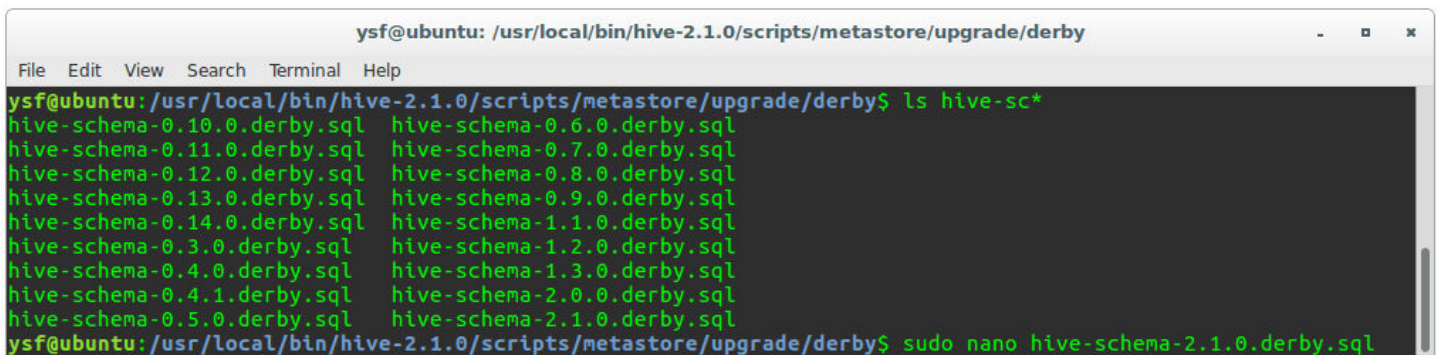
You are getting such warning message because of conflicts sl4j.jar which is being used from HIVE and HADOOP path. In order to get rid of this, you need to delete these jar files binding between Hadoop and Hive

```
rm lib/hive-jdbc-2.0.0-standalone.jar
rm lib/log4j-slf4j-impl-2.4.1.jar
```

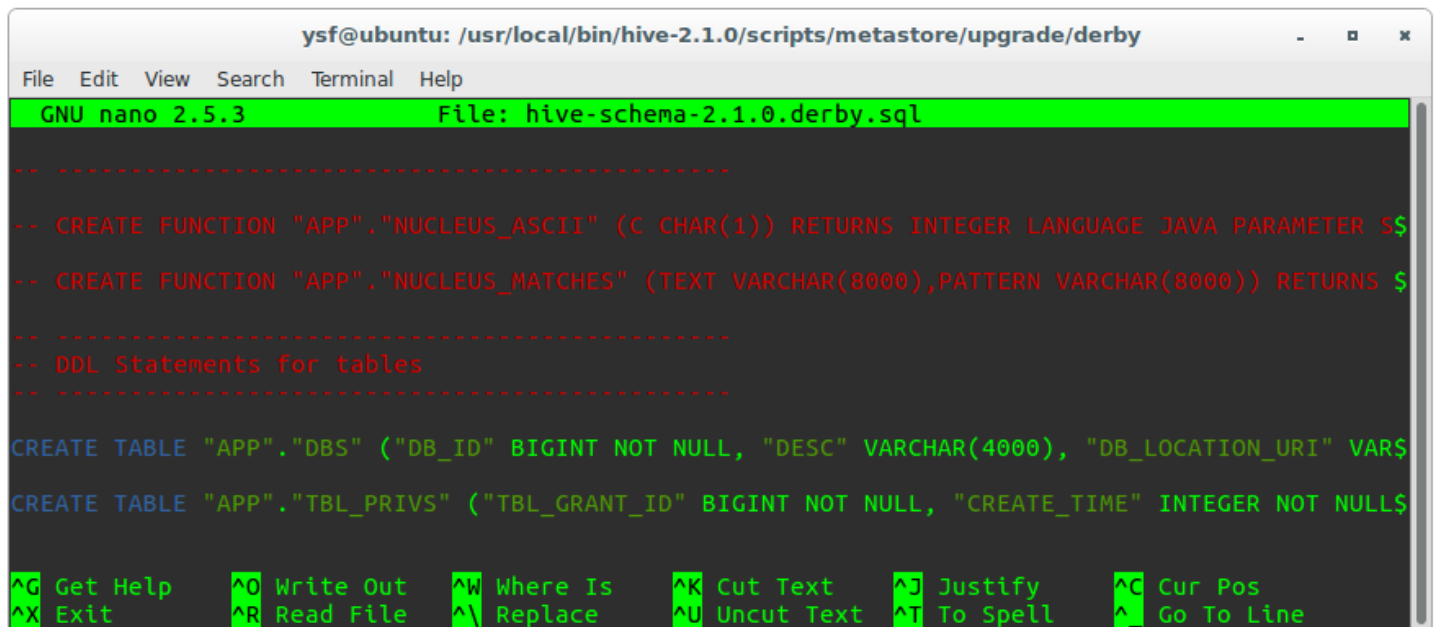
9. When you run hive from \$HIVE\_HOME/bin directory, you may get the following error:

Starting metastore schema initialization to 2.0.0 Initialization script hive-schema-2.0.0.derby.sql Error: FUNCTION 'NUCLEUS\_ASCII' already exists. (state=X0Y68,code=30000) org.apache.hadoop.hive.metastore.HiveMetaException: Schema initialization FAILED! Metastore state would be inconsistent !!

To fix this, edit hive-schema-2.1.0.derby.sql



```
ysf@ubuntu: /usr/local/bin/hive-2.1.0/scripts/metastore/upgrade/derby
File Edit View Search Terminal Help
ysf@ubuntu:/usr/local/bin/hive-2.1.0/scripts/metastore/upgrade/derby$ ls hive-sc*
hive-schema-0.10.0.derby.sql  hive-schema-0.6.0.derby.sql
hive-schema-0.11.0.derby.sql  hive-schema-0.7.0.derby.sql
hive-schema-0.12.0.derby.sql  hive-schema-0.8.0.derby.sql
hive-schema-0.13.0.derby.sql  hive-schema-0.9.0.derby.sql
hive-schema-0.14.0.derby.sql  hive-schema-1.1.0.derby.sql
hive-schema-0.3.0.derby.sql   hive-schema-1.2.0.derby.sql
hive-schema-0.4.0.derby.sql   hive-schema-1.3.0.derby.sql
hive-schema-0.4.1.derby.sql   hive-schema-2.0.0.derby.sql
hive-schema-0.5.0.derby.sql   hive-schema-2.1.0.derby.sql
ysf@ubuntu:/usr/local/bin/hive-2.1.0/scripts/metastore/upgrade/derby$ sudo nano hive-schema-2.1.0.derby.sql
```



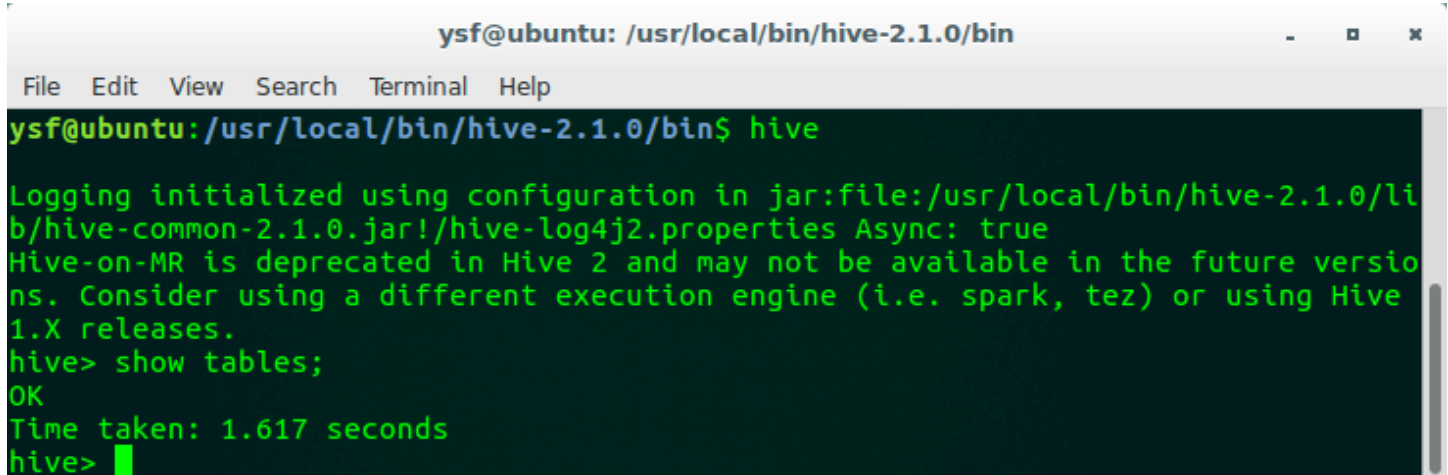
```
ysf@ubuntu: /usr/local/bin/hive-2.1.0/scripts/metastore/upgrade/derby
File Edit View Search Terminal Help
GNU nano 2.5.3 File: hive-schema-2.1.0.derby.sql
-- .....
-- CREATE FUNCTION "APP"."NUCLEUS_ASCII" (C CHAR(1)) RETURNS INTEGER LANGUAGE JAVA PARAMETER $$
-- CREATE FUNCTION "APP"."NUCLEUS_MATCHES" (TEXT VARCHAR(8000),PATTERN VARCHAR(8000)) RETURNS $
-- .....
-- DDL Statements for tables
-- .....
CREATE TABLE "APP"."DBS" ("DB_ID" BIGINT NOT NULL, "DESC" VARCHAR(4000), "DB_LOCATION_URI" VARS
CREATE TABLE "APP"."TBL_PRIVS" ("TBL_GRANT_ID" BIGINT NOT NULL, "CREATE_TIME" INTEGER NOT NULL$
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

Then, run the following command from \$HIVE\_HOME/bin

```
schematool -dbType derby -initSchema
```

10. Run the following command to start the HIVE shell:

`$HIVE_HOME/bin/hive`

A terminal window titled 'ysf@ubuntu: /usr/local/bin/hive-2.1.0/bin' with a menu bar (File, Edit, View, Search, Terminal, Help). The prompt is 'ysf@ubuntu: /usr/local/bin/hive-2.1.0/bin\$'. The command 'hive' has been entered. The output shows logging initialization, a deprecation warning for Hive-on-MR, and the successful execution of the 'show tables;' command, which returned 'OK' and took 1.617 seconds. The prompt is now 'hive>' with a cursor.

```
ysf@ubuntu: /usr/local/bin/hive-2.1.0/bin$ hive
Logging initialized using configuration in jar:file:/usr/local/bin/hive-2.1.0/lib/hive-common-2.1.0.jar!/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
hive> show tables;
OK
Time taken: 1.617 seconds
hive>
```