

# CASSANDRA - SHELL COMMANDS

[http://www.tutorialspoint.com/cassandra/cassandra\\_shell\\_commands.htm](http://www.tutorialspoint.com/cassandra/cassandra_shell_commands.htm)

Copyright © tutorialspoint.com

Cassandra provides documented shell commands in addition to CQL commands. Given below are the Cassandra documented shell commands.

## Help

The HELP command displays a synopsis and a brief description of all cqlsh commands. Given below is the usage of help command.

```
cqlsh> help

Documented shell commands:
=====
CAPTURE COPY DESCRIBE EXPAND PAGING SOURCE
CONSISTENCY DESC EXIT HELP SHOW TRACING.

CQL help topics:
=====
ALTER          CREATE_TABLE_OPTIONS    SELECT
ALTER_ADD      CREATE_TABLE_TYPES      SELECT_COLUMNFAMILY
ALTER_ALTER    CREATE_USER             SELECT_EXPR
ALTER_DROP     DELETE                  SELECT_LIMIT
ALTER_RENAME   DELETE_COLUMNS          SELECT_TABLE
```

## Capture

This command captures the output of a command and adds it to a file. For example, take a look at the following code that captures the output to a file named **Outputfile**.

```
cqlsh> CAPTURE '/home/hadoop/CassandraProgs/Outputfile'
```

When we type any command in the terminal, the output will be captured by the file given. Given below is the command used and the snapshot of the output file.

```
cqlsh:tutorialspoint> select * from emp;
```



```
emp_id | emp_city | emp_name | emp_phone | emp_sal
-----+-----+-----+-----+-----
      1 | Hyderabad |      ram | 9848022338 | 50000
      2 |      Delhi |     robin | 9848022339 | 50000
      4 |      Pune |    rajeev | 9848022331 | 30000
      3 |      Chennai |    rahman | 9848022330 | 50000

(4 rows)
```

You can turn capturing off using the following command.

```
cqlsh:tutorialspoint> capture off;
```

## Consistency

This command shows the current consistency level, or sets a new consistency level.

```
cqlsh:tutorialspoint> CONSISTENCY  
Current consistency level is 1.
```

## Copy

This command copies data to and from Cassandra to a file. Given below is an example to copy the table named **emp** to the file **myfile**.

```
cqlsh:tutorialspoint> COPY emp (emp_id, emp_city, emp_name, emp_phone,emp_sal) TO  
'myfile';  
4 rows exported in 0.034 seconds.
```

If you open and verify the file given, you can find the copied data as shown below.



## Describe

This command describes the current cluster of Cassandra and its objects. The variants of this command are explained below.

**Describe cluster** - This command provides information about the cluster.

```
cqlsh:tutorialspoint> describe cluster;  
  
Cluster: Test Cluster  
Partitioner: Murmur3Partitioner  
  
Range ownership:  
-658380912249644557 [127.0.0.1]  
-2833890865268921414 [127.0.0.1]  
-6792159006375935836 [127.0.0.1]
```

**Describe Keyspaces** - This command lists all the keyspaces in a cluster. Given below is the usage of this command.

```
cqlsh:tutorialspoint> describe keyspaces;  
  
system_traces system tp tutorialspoint
```

**Describe tables** - This command lists all the tables in a keyspace. Given below is the usage of this command.

```
cqlsh:tutorialspoint> describe tables;
emp
```

**Describe table** - This command provides the description of a table. Given below is the usage of this command.

```
cqlsh:tutorialspoint> describe table emp;

CREATE TABLE tutorialspoint.emp (
  emp_id int PRIMARY KEY,
  emp_city text,
  emp_name text,
  emp_phone varint,
  emp_sal varint
) WITH bloom_filter_fp_chance = 0.01
   AND caching = '{"keys":"ALL", "rows_per_partition":"NONE"}'
   AND comment = ''
   AND compaction = {'min_threshold': '4', 'class':
'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy',
'max_threshold': '32'}

   AND compression = {'sstable_compression':
'org.apache.cassandra.io.compress.LZ4Compressor'}

   AND dclocal_read_repair_chance = 0.1
   AND default_time_to_live = 0
   AND gc_grace_seconds = 864000
   AND max_index_interval = 2048
   AND memtable_flush_period_in_ms = 0
   AND min_index_interval = 128
   AND read_repair_chance = 0.0
   AND speculative_retry = '99.0PERCENTILE';
CREATE INDEX emp_emp_sal_idx ON tutorialspoint.emp (emp_sal);
```

## Describe Type

This command is used to describe a user-defined data type. Given below is the usage of this command.

```
cqlsh:tutorialspoint> describe type card_details;

CREATE TYPE tutorialspoint.card_details (
  num int,
  pin int,
  name text,
  cvv int,
  phone set<int>,
  mail text
);
```

## Describe Types

This command lists all the user-defined data types. Given below is the usage of this command. Assume there are two user-defined data types: **card** and **card\_details**.

```
cqlsh:tutorialspoint> DESCRIBE TYPES;

card_details card
```

## Expand

This command is used to expand the output. Before using this command, you have to turn the expand command on. Given below is the usage of this command.

```
cqlsh:tutorialspoint> expand on;
```

```
cqlsh:tutorialspoint> select * from emp;
```

```
@ Row 1
```

```
-----+-----  
emp_id | 1  
emp_city | Hyderabad  
emp_name | ram  
emp_phone | 9848022338  
emp_sal | 50000
```

```
@ Row 2
```

```
-----+-----  
emp_id | 2  
emp_city | Delhi  
emp_name | robin  
emp_phone | 9848022339  
emp_sal | 50000
```

```
@ Row 3
```

```
-----+-----  
emp_id | 4  
emp_city | Pune  
emp_name | rajeev  
emp_phone | 9848022331  
emp_sal | 30000
```

```
@ Row 4
```

```
-----+-----  
emp_id | 3  
emp_city | Chennai  
emp_name | rahman  
emp_phone | 9848022330  
emp_sal | 50000  
(4 rows)
```

**Note:** You can turn the expand option off using the following command.

```
cqlsh:tutorialspoint> expand off;  
Disabled Expanded output.
```

## Exit

This command is used to terminate the cql shell.

## Show

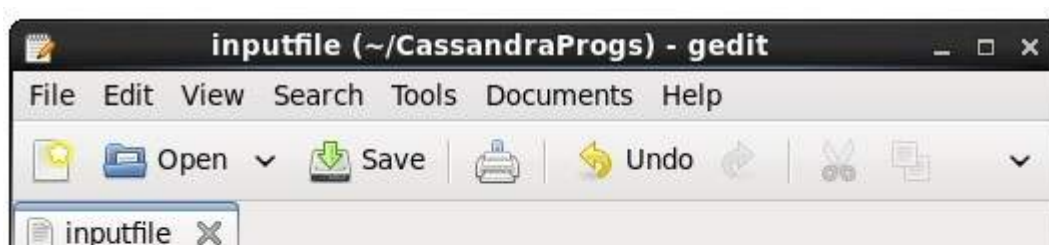
This command displays the details of current cqlsh session such as Cassandra version, host, or data type assumptions. Given below is the usage of this command.

```
cqlsh:tutorialspoint> show host;  
Connected to Test Cluster at 127.0.0.1:9042.
```

```
cqlsh:tutorialspoint> show version;  
[cqlsh 5.0.1 | Cassandra 2.1.2 | CQL spec 3.2.0 | Native protocol v3]
```

## Source

Using this command, you can execute the commands in a file. Suppose our input file is as follows:



```
use tutorialspoint;
select * from emp;
```

Plain Text ▾ Tab Width: 8 ▾ Ln 1, Col 17 INS

Then you can execute the file containing the commands as shown below.

```
cqlsh:tutorialspoint> source '/home/hadoop/CassandraProgs/inputfile';
```

emp_id	emp_city	emp_name	emp_phone	emp_sal
1	Hyderabad	ram	9848022338	50000
2	Delhi	robin	9848022339	50000
3	Pune	rajeev	9848022331	30000
4	Chennai	rahman	9848022330	50000

(4 rows)