System Tools

The source code for Apache Kafka System Tools are located https://github.com/apache/kafka/tree/0.8/core/src/main/scala/kafka/tools

If you are looking for the replication related tools then please check out the wiki page on Replication tools

System tools can be run from the command line using the run class script (i.e. bin/kafka-run-class.sh package.class --options)

Consumer Offset Checker

This tool has been removed in Kafka 1.0.0. Use kafka-consumer-groups.sh to get consumer group details.

Displays the: Consumer Group, Topic, Partitions, Offset, logSize, Lag, Owner for the specified set of Topics and Consumer Group

```
bin/kafka-run-class.sh kafka.tools.ConsumerOffsetChecker
```

required argument: [group]
Option Description
------ -----------
--broker-info Print broker info
--group Consumer group.
--help Print this message.
--topic Comma-separated list of consumer
topics (all topics if absent).
--zkconnect ZooKeeper connect string. (default: localhost:2181)

Dump Log Segment

This can print the messages directly from the log files or just verify the indexes correct for the logs

```
bin/kafka-run-class.sh kafka.tools.DumpLogSegments
```

required argument "[files]"

Option Description
------ -----------
--deep-iteration if set, uses deep instead of shallow iteration
--files <file1, file2, ...> REQUIRED: The comma separated list of data and index log files to be dumped
--max-message-size <Integer: size> Size of largest message. (default: 5242880)
--print-data-log if set, printing the messages content when dumping data logs
--verify-index-only if set, just verify the index log without printing its content

Export Zookeeper Offsets

A utility that retrieves the offsets of broker partitions in ZK and prints to an output file in the following format:

/consumers/group1/offsets/topic1/1-0:286894308
/consumers/group1/offsets/topic1/2-0:284803985

```
bin/kafka-run-class.sh kafka.tools.ExportZkOffsets
```

required argument: [zkconnect]
Option Description
------ -----------
--group Consumer group.
--help Print this message.
--output-file Output file
--zkconnect ZooKeeper connect string. (default: localhost:2181)

Get Offset Shell

get offsets for a topic

```
bin/kafka-run-class.sh kafka.tools.GetOffsetShell
```

required argument [broker-list], [topic]
Option Description
------ -----------
--broker-list <hostname:port,..., REQUIRED: The list of hostname and hostname:port> port of the server to connect to.
--max-wait-ms <Integer: ms> The max amount of time each fetch request waits. (default: 1000)
--offsets <Integer: count> number of offsets returned (default: 1)
--partitions <partition ids> comma separated list of partition ids. If not specified, will find offsets for all partitions (default)
--time <Long: timestamp in milliseconds / -1(latest) / -2 (earliest) timestamp; offsets will come before this timestamp, as in getOffsetsBefore >
--topic <topic> REQUIRED: The topic to get offsets from.

Import Zookeeper Offsets

can import offsets for a topic partitions

file format is the same as for the export

/consumers/group1/offsets/topic1/1-0:286894308
/consumers/group1/offsets/topic1/2-0:284803985

```
bin/kafka-run-class.sh kafka.tools.ImportZkOffsets
```

required argument: [input-file]
Option Description
------ -----------
--help Print this message.
--input-file Input file
--zkconnect ZooKeeper connect string. (default: localhost:2181)

JMX Tool

prints metrics via JMX

```
bin/kafka-run-class.sh kafka.tools.JmxTool
```

Option Description
------ -----------
--attributes <name> The whitelist of attributes to query. This is a comma-separated list. If no attributes are specified all objects will be queried.
--date-format <format> The date format to use for formatting the time field. See java.text. SimpleDateFormat for options.
--help Print usage information.
--jmx-url <service-url> The url to connect to to poll JMX data. See Oracle javadoc for JMXServiceURL for details. (default: service:jmx:rmi:///jndi/rmi://: 9999
/jmxrmi)
--object-name <name> A JMX object name to use as a query. This can contain wild cards, and this option can be given multiple times to specify more
than one query. If no objects are specified all objects will be queried.
--reporting-interval <Integer: ms> Interval in MS with which to poll jmx stats. (default: 2000)

Kafka Migration Tool

Migrates a 0.7 broker to 0.8

```
bin/kafka-run-class.sh kafka.tools.KafkaMigrationTool
```

Missing required argument "[consumer.config]"
Option Description
------ -----------
--blacklist <Java regex (String)> Blacklist of topics to migrate from the 0.7 cluster
--consumer.config <config file> Kafka 0.7 consumer config to consume from the source 0.7 cluster. You man specify multiple of these.
--help Print this message.
--kafka.07.jar <kafka 0.7 jar> Kafka 0.7 jar file
--num.producers <Integer: Number of Number of producer instances (default: producers> 1)
--num.streams <Integer: Number of Number of consumer streams (default: 1)consumer threads>
--producer.config <config file> Producer config.
--queue.size <Integer: Queue size in Number of messages that are buffered terms of number of messages> between the 0.7 consumer and 0.8 producer (default: 10000)
--whitelist <Java regex (String)> Whitelist of topics to migrate from the 0.7 cluster
--zkclient.01.jar <zkClient 0.1 jar zkClient 0.1 jar file file required by Kafka 0.7>

Mirror Maker

Provides mirroring of one Kafka cluster to another, for more info check out the wiki page on Kafka mirroring (MirrorMaker)

```
bin/kafka-run-class.sh kafka.tools.MirrorMaker
```

required argument [consumer.config]
Option Description
------ -----------
--blacklist <Java regex (String)> Blacklist of topics to mirror.
--consumer.config <config file> Consumer config to consume from a source cluster. You may specify multiple of these.
--help Print this message.
--num.producers <Integer: Number of Number of producer instances (default: producers> 1)
--num.streams <Integer: Number of Number of consumption streams. threads> (default: 1)
--producer.config <config file> Embedded producer config.
--queue.size <Integer: Queue size in Number of messages that are buffered terms of number of messages> between the consumer and producer (default: 10000)
--whitelist <Java regex (String)> Whitelist of topics to mirror.

Replay Log Producer

Consume from one topic and replay those messages and produce to another topic

```
bin/kafka-run-class.sh kafka.tools.ReplayLogProducer
```

required argument [broker-list], [input-topic], [output-topic], [zookeeper]
Option Description
------ -----------
--async If set, messages are sent asynchronously.
--batch-size <Integer: batch size> Number of messages to send in a single batch. (default: 200)
--broker-list <hostname:port> REQUIRED: the broker list must be specified.
--compression-codec <Integer: If set, messages are sent compressed compression codec > (default: 0)
--delay-btw-batch-ms <Long: ms> Delay in ms between 2 batch sends. (default: 0)
--inputtopic <input-topic> REQUIRED: The topic to consume from.
--messages <Integer: count> The number of messages to send. (default: -1)
--outputtopic <output-topic> REQUIRED: The topic to produce to
--reporting-interval <Integer: size> Interval at which to print progress info. (default: 5000)
--threads <Integer: threads> Number of sending threads. (default: 1)
--zookeeper <zookeeper url> REQUIRED: The connection string for the zookeeper connection in the form host:port. Multiple URLS can be given to allow fail-over. (default: 127.0.0.1:2181)

Simple Consumer Shell

Dumps out consumed messages to the console using the Simple Consumer

```
bin/kafka-run-class.sh kafka.tools.SimpleConsumerShell
```

required argument [broker-list], [topic]
Option Description
------ -----------
--broker-list <hostname:port,..., REQUIRED: The list of hostname and hostname:port> port of the server to connect to.
--clientId <clientId> The ID of this client. (default: SimpleConsumerShell)
--fetchsize <Integer: fetchsize> The fetch size of each request. (default: 1048576)
--formatter <class> The name of a class to use for formatting kafka messages for display. (default: kafka.consumer. DefaultMessageFormatter)
--max-messages <Integer: max-messages> The number of messages to consume (default: 2147483647)
--max-wait-ms <Integer: ms> The max amount of time each fetch request waits. (default: 1000)
--no-wait-at-logend If set, when the simple consumer reaches the end of the Log, it will stop, not waiting for new produced messages
--offset <Long: consume offset> The offset id to consume from, default to -2 which means from beginning; while value -1 means from end (default: -2)
--partition <Integer: partition> The partition to consume from. (default: 0)
--print-offsets Print the offsets returned by the iterator
--property <prop>
--replica <Integer: replica id> The replica id to consume from, default -1 means leader broker. (default: -1)
--skip-message-on-error If there is an error when processing a message, skip it instead of halt.
--topic <topic> REQUIRED: The topic to consume from.

State Change Log Merger

A utility that merges the state change logs (possibly obtained from different brokers and over multiple days).

```
bin/kafka-run-class.sh kafka.tools.StateChangeLogMerger
```

Provide arguments to exactly one of the two options "[logs]" or "[logs-regex]"
Option Description
------ -----------
--end-time <end timestamp in the The latest timestamp of state change format java.text. log entries to be merged (default: SimpleDateFormat@f17a63e7>
9999-12-31 23:59:59,999)
--logs <file1,file2,...> Comma separated list of state change logs or a regex for the log file names
--logs-regex <for example: /tmp/state- Regex to match the state change log change.log*> files to be merged
--partitions <0,1,2,...> Comma separated list of partition ids whose state change logs should be merged
--start-time <start timestamp in the The earliest timestamp of state change format java.text. log entries to be merged (default:
SimpleDateFormat@f17a63e7> 0000-00-00 00:00:00,000)
--topic <topic> The topic whose state change logs should be merged

Update Offsets In Zookeeper

A utility that updates the offset of every broker partition to the offset of earliest or latest log segment file, in ZK.

```
bin/kafka-run-class.sh kafka.tools.UpdateOffsetsInZK
```

USAGE: kafka.tools.UpdateOffsetsInZK$ [earliest | latest] consumer.properties topic

Verify Consumer Rebalance

Make sure there is an owner for every partition. A successful rebalancing operation would select an owner for each available partition.
This means that for each partition registered under /brokers/topics/[topic]/[broker-id], an owner exists under /consumers/[consumer_group]/owners/[topic]/
[broker_id-partition_id]

```
bin/kafka-run-class.sh kafka.tools.VerifyConsumerRebalance
```

required argument: [group]
Option Description
------ -----------
--group Consumer group.
--help Print this message.
--zookeeper.connect ZooKeeper connect string. (default: localhost:2181)