

# KIP-370: Remove Orphan Partitions

- [Status](#)
- [Motivation](#)
- [Public Interfaces](#)
- [Proposed Changes](#)
- [Compatibility, Deprecation, and Migration Plan](#)
- [Rejected Alternatives](#)

## Status

**Current state:** *"Under Discussion"*

**Discussion thread:** [here](#)

**JIRA:** [KAFKA-7362](#)

## Motivation

When partition reassignment removes topic partitions from a offline broker, those removed partitions become orphan partitions to the offline broker. When the offline broker comes back online, it is not able to clean up both data and folders that belong to orphan partitions. The retention policy on a topic partition will not be kicked out until the partition is known to the broker as either leader partition or follower partition (i.e., until "replicaHighWatermark" is set). Orphan partitions will not become leader or follower partitions and thus will not have retention. In addition, we do not have logic to delete folders that belong to orphan partition today. This KIP provides a mechanism for brokers to remove orphan partitions automatically.

## Public Interfaces

- Add two metrics:
  - 1) kafka.log:type=LogManager,name=OrphanLogPartitionCount  
type = gauge  
value = the number of orphan partitions
  - 2) kafka.log:type=LogManager,name=OrphanLogPartitionSize  
type = gauge  
value = the size of orphan partitions

## Proposed Changes

a) Provide a mechanism to remove orphan partition automatically. The orphan partitions removal works as follows:

1. Collect orphan partitions.  
The orphan partition set will be determined based on the first leaderandISR request when a broker starts. Topic Partitions that are present in the broker but not in the leaderandISR request are determined as orphan partitions.
2. Schedule the orphan partition removal  
Schedule the orphan partition removal after a certain time interval (such as 2 hours, defined internally in the broker). This time interval enables the broker to revive orphan partitions for pending partition reassignment in the cluster. Since partition reassignment request might come as separated leaderandISR request, the delayed orphan partition removal makes it possible for the broker to re-host the same topic partition without actually fetching entire toptic partition data from partition leader.
3. Remove the orphan partitions.  
The broker removes orphan partitions (including partition folders) whose log segments are all older than the broker default retention period. Broker will not distinguish between the log compacted topic and time-retention topics for those partitions in orphan partition set. The default retention period of a broker is used for all orphan partitions. Broker only removes orphan partitions whose log segments are all older than the default retention period. This is to ensure broker will not try to delete new data. If some orphan partitions cannot be removed immediately because the retention period has not been reached, a new deletion will be scheduled again to perform deletion.

b) Adding metrics to keep track of the number of orphan partitions and the size of these orphan partitions as described in the public Interface section.

## Compatibility, Deprecation, and Migration Plan

- *There is no compatibility issue.*

## Rejected Alternatives

1 ) Manual deletion of orphan partitions via provided API. Kafka provides an API that the user can specify what topic partitions that he wants to delete and what is the time retention rule. Kafka only remove a partition if all the following conditions are met:

- The current broker is neither a follower nor a leader for the partition.
- The partition satisfied the specify time retention rule. For example, all log segments are more than certain days old.
- The current broker has received the first leaderAndIsr request.

The problem of this solution is it requires human interaction. We need to use admin tool to find out what are orphan partitions and then use provided API to perform partition deletion.