

RepairAsyncAPI

Repair Async API

Repair used to be invoked through sync JMX interface, but since repair takes time to finish, JMX connection timeout happens sometimes. So [CASSANDRA-4767](#) added asynchronous repair API which once invoked users can track repair progress through JMX notification.

Repair JMX Notification

Repair JMX Notification is sent from [StorageService](#) MBean(org.apache.cassandra.db:type=StorageService).

Before you run repair, you should subscribe to receive JMX notification otherwise you may miss some of messages.

Repair JMX Notification contains the following.

type	"repair"
message	repair status message
user data	int array containing command number and repair status

message is repair status message like "Starting repair ..." or error message.

user data is int array of 2 elements. The first element is *command number* which is assigned uniquely when repair is invoked through async API. You can obtain command number as return value of async APIs. The second element is repair status number as shown below.

0	STARTED	repair command started
1	SESSION_SUCCEEDED	repair session (repair for one range in a keyspace) succeeded
2	SESSION_FAILED	repair session failed
3	FINISHED	repair command finished

(In the code, these are defined as [ActiveRepairService](#).Status enum.)

nodetool repair command also uses these status to track repair progress.

Further improvement

Still, the granularity of tracking repair status is large. Repair involves several nodes who do validation compaction and file streaming. Each of those are monitored through *nodetool compactionstat* and *nodetool netstat* on each node.

Possible solution to track the whole repair process is to [Repair tracing](#).