# MOB User Guide

Data comes to HBase in many sizes, and it is convenient to store the binary data such as images, documents in HBase. The normal read and write paths in HBase are optimized for values smaller than 100KB in size. When HBase handles large numbers of values up to 10M, performance can be degraded because of write amplification caused by splits and compactions.

HBase 2.0+ has included the implementation for better storing medium objects (MOBs) directly in HBase.

To enable the MOB feature, you must use HFile version 3, and declare the MOB-enabled column families in tables. No changes in client are required.

## Enable HFile Version 3

Edit the hbase-site.xml, add the following property into it to enable HFile version 3.

|  |
| --- |
| <property>  <name>hfile.format.version</name>  <value>3</value> </property> |

## Enable MOB in Column Families

There are two ways to enable MOB in column families.

Using HBase Shell.

|  |
| --- |
| hbase> create 't1', {NAME => 'f1', IS\_MOB => true, MOB\_THRESHOLD => 102400} |

Using Java APIs.

|  |
| --- |
| HColumnDescriptor hcd = new HColumnDescriptor(“f”);  hcd.setMobEnabled(true);  hcd.setMobThreshold(102400); |

## Enable and Configure the MOB File Cache

Edit hbase-site.xml, add the following properties to enable the MOB file cache.

|  |
| --- |
| <property>  <name>hbase.mob.file.cache.size</name>  <value>1000</value>  <description>  Number of opened file handlers to cache.  A larger value will benefit reads by provinding more file handlers per mob  file cache and would reduce frequent file opening and closing.  However, if this is set too high, this could lead to a "too many opened file handers"  The default value is 1000.  </description>  </property>  <property>  <name>hbase.mob.cache.evict.period</name>  <value>3600</value>  <description>  The amount of time in seconds before the mob cache evicts cached mob files.  The default value is 3600 seconds.  </description>  </property>  <property>  <name>hbase.mob.cache.evict.remain.ratio</name>  <value>0.5f</value>  <description>  The ratio (between 0.0 and 1.0) of files that remains cached after an eviction  is triggered when the number of cached mob files exceeds the hbase.mob.file.cache.size.  The default value is 0.5f.  </description>  </property> |

## MOB File Management

The MOB feature introduces a new read and write path to HBase, the MOB files are cleaned and compacted in two ways.

One is ExpiredMobFileCleaner, it cleans the MOB files that are expired by TTLs. The other is MobCompactor, it cleans up the deleted MOB data and compacts small MOB files into larger ones.

1. **ExpiredMobFileCleaner**

There are two ways to run the ExpiredMobFileCleaner.

One is to run the following command line.

|  |
| --- |
| hbase org.apache.hadoop.hbase.mob.ExpiredMobFileCleaner tableName familyName |

The other is the periodical execution in HMaster. The period can be set in the configuration. The default value is one day.

|  |
| --- |
| <property>  <name>hbase.master.mob.ttl.cleaner.period</name>  <value>86400</value>  <description>  The period that ExpiredMobFileCleanerChore runs. The unit is second.  The default value is one day.  The MOB file name users only the data part of the file creation time in it. We use this time for deciding TTL expiry of the files. So the removal of TTL expired files might be delayed. The max delay might be 24 hrs.  </description>  </property> |

1. **MobCompactor**

There are two ways to run MobCompactor.

One is to run it in hbase shell.

|  |
| --- |
| **hbase> compact ‘t1’, ‘f1’, ‘MOB’**  or  **hbase> major\_compact ‘t1’, ‘f1’, ‘MOB’** |

The other is to enable the periodical execution in HMaster. The period can be set in the configuration. The default value is one week. And if this value is set as 0 or a negative value, the periodical execution is disabled.

|  |
| --- |
| <property>  <name>hbase.mob.compaction.mergeable.threshold</name>  <value>1342177280</value>  <description>  If the size of a mob file is less than this value, it's regarded as a small  file and needs to be merged in mob compaction. The default value is 1280MB.  </description>  </property>  <property>  <name>hbase.mob.delfile.max.count</name>  <value>3</value>  <description>  The max number of del files that is allowed in the mob compaction.  In the mob compaction, when the number of existing del files is larger than  this value, they are merged until number of del files is not larger this value.  The default value is 3.  </description>  </property>  <property>  <name>hbase.mob.compaction.batch.size</name>  <value>100</value>  <description>  The max number of the mob files that is allowed in a batch of the mob compaction.  The mob compaction merges the small mob files to bigger ones. If the number of the  small files is very large, it could lead to a "too many opened file handlers" in the merge.  And the merge has to be split into batches. This value limits the number of mob files  that are selected in a batch of the mob compaction. The default value is 100.  </description>  </property>  <property>  <name>hbase.mob.compaction.chore.period</name>  <value>604800</value>  <description>  The period that MobCompactionChore runs. The unit is second.  The default value is one week.  </description>  </property>  <property>  <name>hbase.mob.compactor.class</name>  <value>org.apache.hadoop.hbase.mob.compactions.PartitionedMobCompactor</value>  <description>  Implementation of mob compactor, the default one is PartitionedMobCompactor.  </description>  </property>  <property>  <name>hbase.mob.compaction.threads.max</name>  <value>1</value>  <description>  The max number of threads used in MobCompactor.  </description>  </property> |

## Run MOB Integration Test

|  |
| --- |
| sudo -u hbase hbase org.apache.hadoop.hbase.IntegrationTestIngestWithMOB -threshold 1024 -minMobDataSize 512 -maxMobDataSize 5120 |

* threshold: the threshold to classify cells to MOB data, default is 1KB.
* minMobDataSize: Minimum value size for MOB data, default is 512B.
* maxMobDataSize: Maxmum value size for MOB data, default is 5KB.