**Title:** Redis XTRIM

***Excerpt:*** Redis XTRIM command is used to trim a stream based on a threshold value which can be a maximum length of the final stream or a minimum id that should be le after trimming a stream. The MAXLEN and MINID arguments can be used to decide the threshold value type. Also, the ~ operator can be used to perform a nearly exact trimming on a given Redis stream which is more efficient than the default exact trimming approach. In addition, the LIMIT argument can be used to limit the number of entries to be evicted by the XTRIM command.

**Permalink:** redis-xtrim

**Category:** Redis

# 

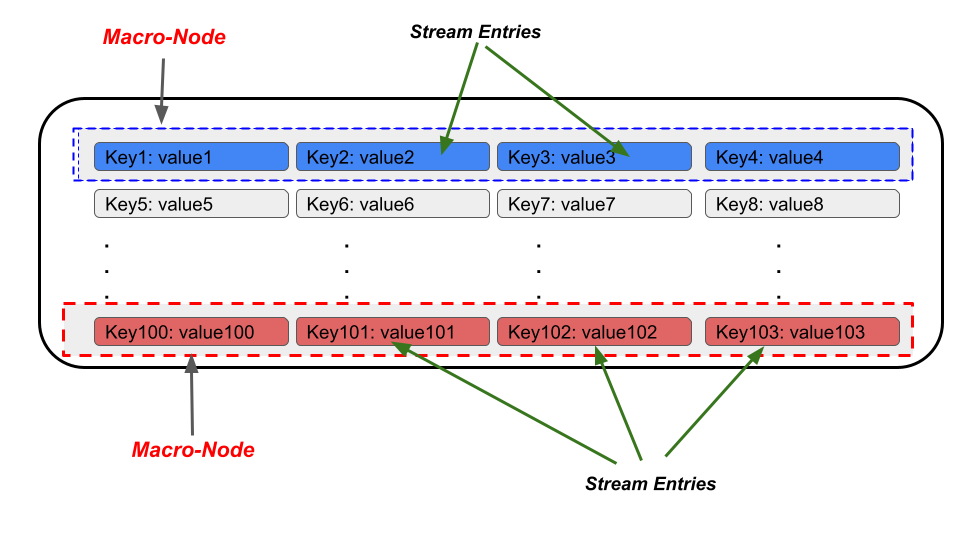
# 

# 

# Low Level Details of Redis Streams

Redis streams are an append-only data structure that provides a set of highly efficient read and insert operations with a memory-efficient storage mechanism. Internally, the Redis streams use a Radix tree data structure which is a space-optimized tree with high memory efficiency.

Redis streams store data as a list of entries where each entry consists of key-value pairs. At low level, these entries are packed into macro-nodes as shown in the following.



In this article, we are focusing on the removal of stream entries using the XTRIM command and the above-mentioned concept closely affects the efficiency of the removal operation. Usually, Redis stream entry removal is very efficient if it's done on the macro-node level but not at the entry-level. This mechanism is implemented with the XTRIM command that we will be discussing in the following section.

# The XTRIM Command

The XTRIM command is used to cut off the entries of a stream based on a given threshold value. The threshold can be a maximum number of entries per stream or an older entry id. The XTRIM command accepts the type of the threshold as a command argument. The syntax of the XTRIM command is as follows.

| XTRIM stream\_key MAXLEN | MINID [= | ~] threshold\_value [LIMIT count] |
| --- |

***stream\_key***: The key of the Redis stream.

***MAXLEN:*** The maximum length of the stream after trimming entries. All the entries will be removed that exceed the length of the stream which is specified by the threshold value. This argument is a positive integer.

***MINID***: The minimum stream id that should remain after trimming the entries. All the entries will be removed that are having IDs lower than the specified threshold. Whenever you specify this parameter, the threshold becomes a stream ID.

***= operator :*** When specified, exact trimming will be performed based on the threshold value.

***~ operator:*** When specified, nearly exact trimming will be performed based on the threshold value and macro-node size.

***threshold\_value:*** The threshold value based on the argument MAXLEN or MINID.

***LIMIT count***: The maximum number of entries to be removed.

## Use Case 01 - Removing Stream Entries Based on Maximum Length

Let’s assume that a tourism company maintains a Redis store to keep track of tourists information. The Redis streams data structure has been used to store each tourists information as key-value pairs. With time, the stream has grown and they are planning to keep only the latest 1000 entries. So they have identified the maximum length of the stream should be 1000 at any time. The XTRIM command has been used to achieve this.

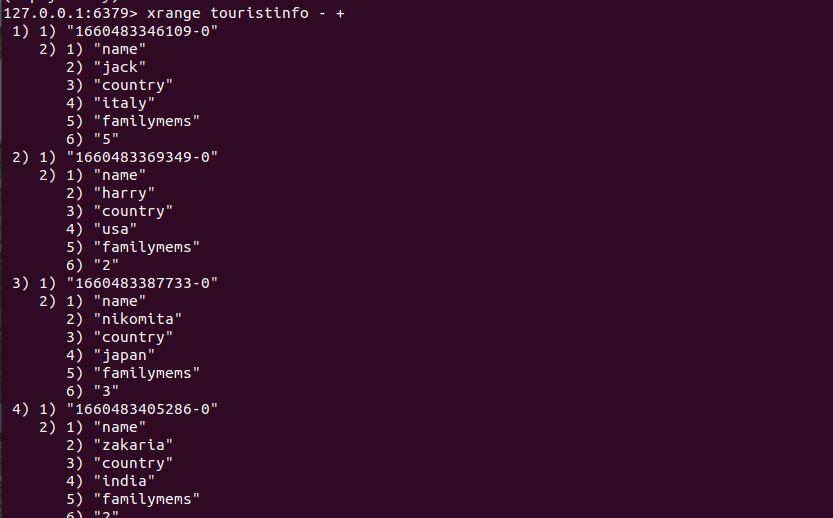
For the demonstration purposes, first we will create a stream with 10 entries as follows. The XADD command has been used to perform the insertion.

| xadd touristinfo \* name jack country italy familymems 5 xadd touristinfo \* name harry country usa familymems 2 xadd touristinfo \* name nikomita country japan familymems 3 xadd touristinfo \* name zakaria country india familymems 2 xadd touristinfo \* name redmond country brazil familymems 6 xadd touristinfo \* name nakita country japan familymems 3 xadd touristinfo \* name maryjohn country usa familymems 2 xadd touristinfo \* name liza country italy familymems 5 xadd touristinfo \* name nimshikaa country japan familymems 3 xadd touristinfo \* name nisha country italy familymems 5 |
| --- |

Let’s use the XRANGE command to inspect the stream touristinfo as follows.

| xrange touristinfo - + |
| --- |

Output:

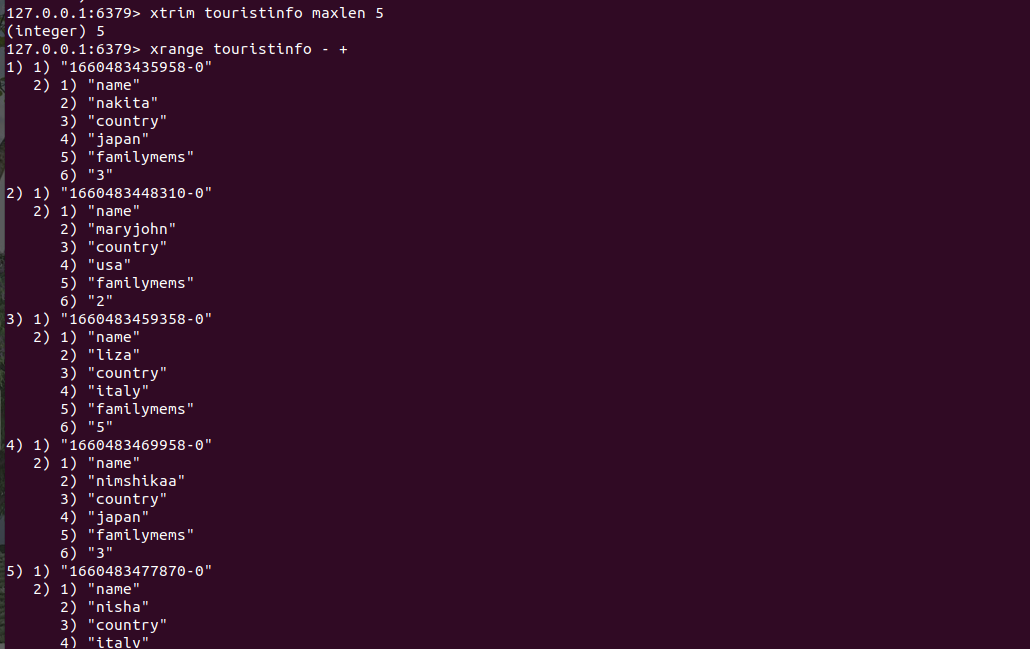


As expected, the 10 stream entries have been displayed by this command.

For demonstration purposes, we will be trimming the stream where its maximum length would be 5.

| xtrim touristinfo maxlen 5 |
| --- |

Let’s inspect the stream entries again with the XRANGE command.



As expected, five entries have been evicted from the stream and its length is 5.

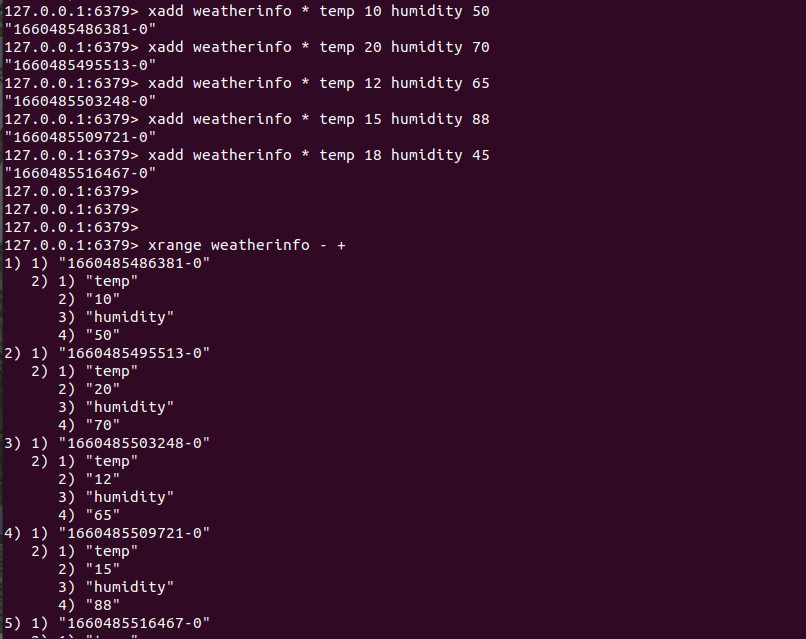
## Use Case 02 - Removing Stream Entries Based on Stream IDs

Let’s take an example where a weather company keeps track of weather information of a given location using Redis streams. Now, they want to delete older entries that are no longer needed. We can use the XTRIM command using the MINID strategy as shown in the following.

First we will be creating a stream called *weatherinfo* and add 5 entries to it as follows.

| xadd weatherinfo \* temp 10 humidity 50 xadd weatherinfo \* temp 20 humidity 70 xadd weatherinfo \* temp 12 humidity 65 xadd weatherinfo \* temp 15 humidity 88 xadd weatherinfo \* temp 18 humidity 45 |
| --- |

Output:

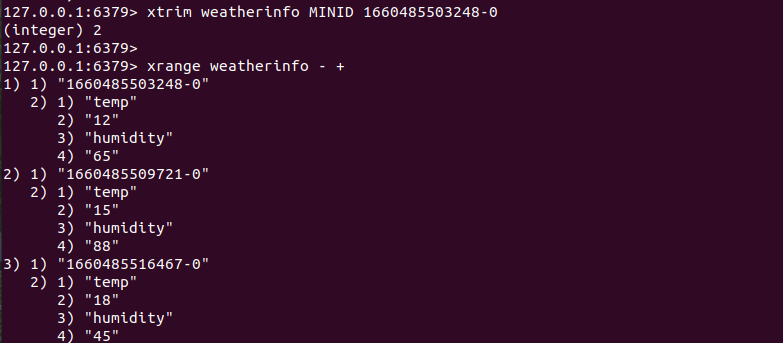


Let’s use the XTRIM command to remove the entries that are having IDs lower than the specified threshold value.

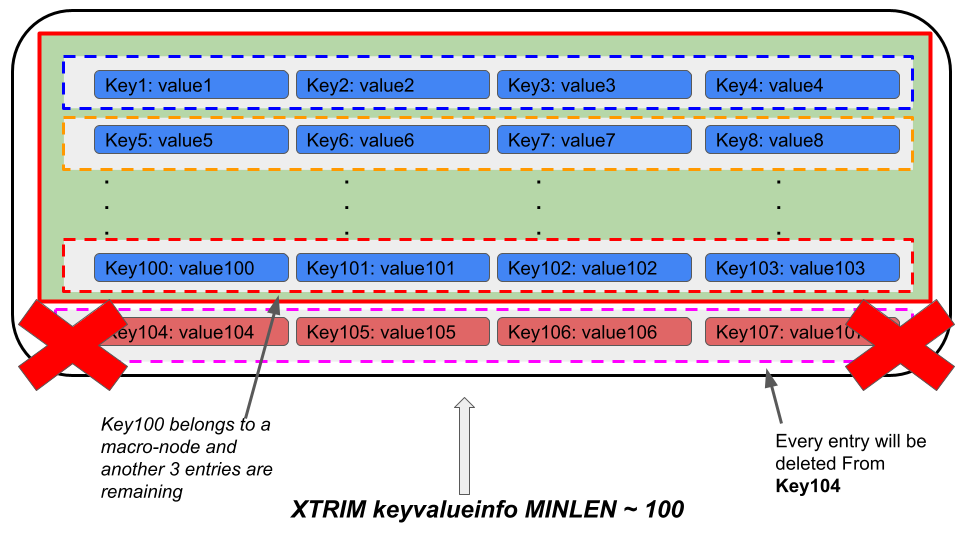
| xtrim weatherinfo MINID 1660485503248-0 |
| --- |

The minimum id specified is associated with the third entry. Hence, the entries after the third entry which are having lower ids will be deleted.

Output:



Since we have not specified the *=* or *~* arguments explicitly, the command uses the *=* operator by default. Hence, the exact trimming has been done in both use cases. If you have specified the ~ operator explicitly, nearly exact trimming will be done as shown in the following.



As shown in the above figure, the XTRIM command has been used with the *~* operator. We ask the command to remove all the entries that exceed the length of 100. Since we do not force the XTRIM command to do an exact trimming, it will focus on the efficiency of the trimming operation. So, it will not remove the immediate entries which belong to the same macro-node. It will keep the next three entries that are in the same macro-node and will remove all the macro-nodes after those. It gives a considerable amount of performance improvement than in the exact trimming approach which forces the command to put extra effort. The same procedure takes place when the threshold is based on the entry ids.

The *LIMIT* argument limits the number of evicted entries from the specified stream which can be used to gain a bit more performance improvement.

# Conclusion

In short, the XTRIM command is used to remove stream entries based on a threshold value. The type of the threshold value can be changed with the context which is being specified using the MAXLEN and MINID command arguments. As discussed, the trimming can be done in two ways where the trimming operation will consider the threshold as an exact or approximate limit. By default, the command uses *the =* operator that is used for exact trimming. With the *~* operator, you can do an approximate trimming as mentioned in the above section. Overall, the approximate trimming approach and the *LIMIT* argument help you to gain a considerable performance gain with the XTRIM command.