

Cloud Message Queue

Message Queue Model

Product Introduction



Tencent
Cloud

Copyright Notice

©2013-2017 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

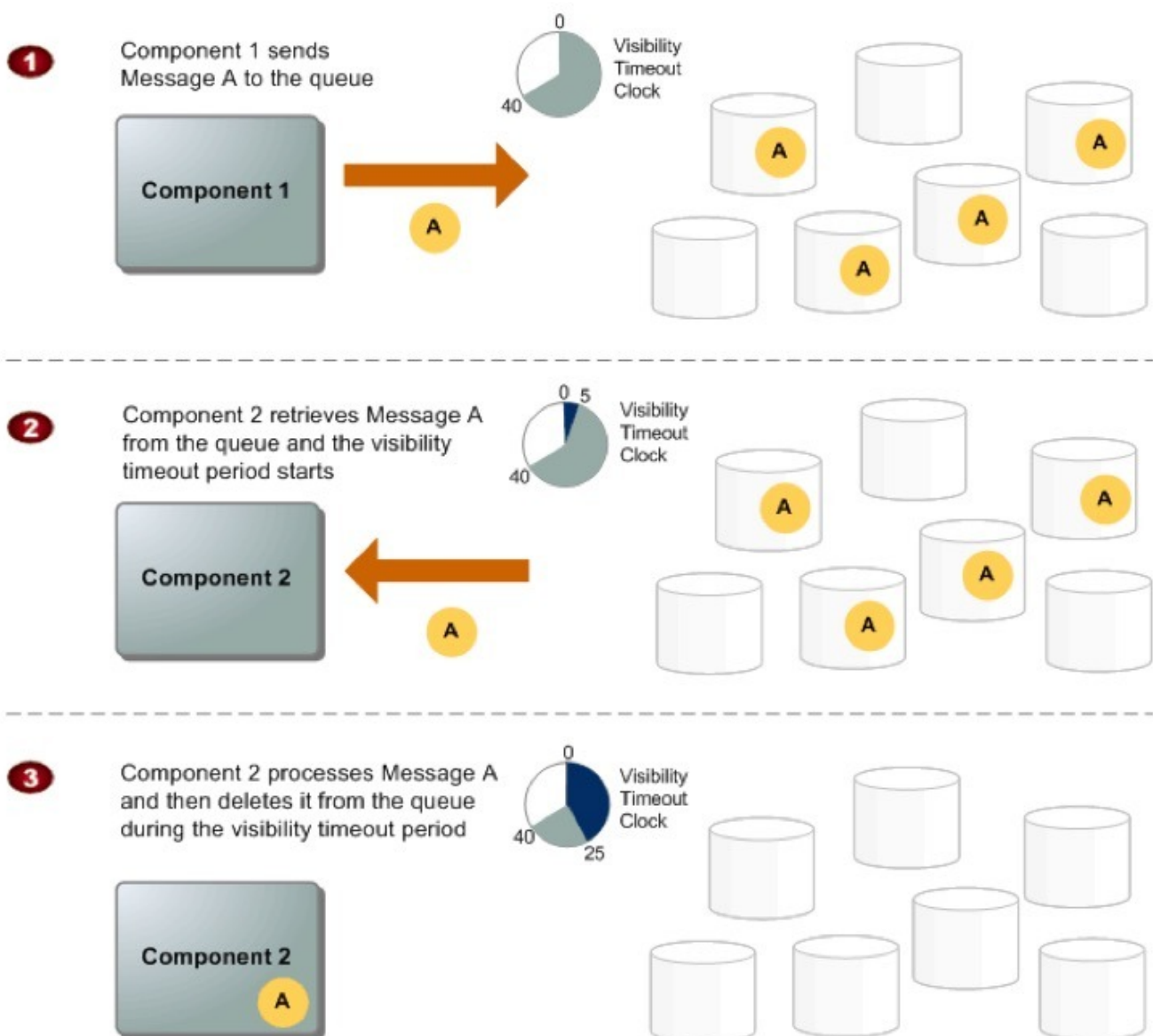
Documentation Legal Notice	2
Message Queue Model	4
Message Lifecycle	4
Queue and Message Identifiers.....	6
Delayed Message	7
Message Rewinding.....	8
Creating a Queue.....	12
Producing Messages.....	13

Message Queue Model

Message Lifecycle

A common message that has been sent to the common message queue is originally in an Active status. If retrieved, this message will be in an Inactive status within VisibilityTimeout. If the message has not been deleted within the VisibilityTimeout, it will return to Active status. The status of the message will become Deleted if it is deleted within VisibilityTimeout. MessageRetentionPeriod specified upon the creation of a queue determines the maximum retention time of the message. It will become Expired and be reclaimed when the retention time expires.

Consumers can only retrieve the messages in an Active status. This ensures that a message cannot be repeatedly consumed in the same period of time but can be repeatedly consumed in sequence.



- Component 1 sends Message A to a queue, and the message provides multiple copies of redundant data across the CMQ servers.
- When Component 2 is ready to process a message, it retrieves messages from the queue, and Message A is returned. While Message A is being processed, it remains in the queue and cannot be received by other business for the duration of the visibility timeout
- Component 2 deletes Message A from the queue to prevent the message from being received and processed again once the visibility timeout expires. If Message A is not deleted, it can be consumed repeatedly by other business

Queue and Message Identifiers

□ Before using Tencent Cloud CMQ, you need to get familiar with the following identifiers: queue name, message ID and receipt handle.

1. Queue Name

When creating a new queue, you must specify a unique queue name within the current region. The same queue name can be used among different regions. Tencent Cloud CMQ uses region plus queue name to identify a queue. You need to provide both parameters whenever you want to work with the queue.

2. Message ID

Each message receives a system-assigned message ID that Tencent Cloud returns to you in the `SendMessage` response. This identifier is used to identify messages. However, you need to provide the message's receipt handle rather than the message ID to delete a message. Example of a message ID format: "Msg-XXXXXXX".

3. Receipt Handle

Every time you receive a message from a queue, you receive a receipt handle for that message. This handle is associated with the action of receiving the message, not with the message itself. To delete the message or to change the message visibility, you must provide the receipt handle (not the message ID). Thus, you must always receive a message before you can delete/change it.

Note: If you receive a message more than once, each time you receive it, you get a different receipt handle. You must provide the most recently received receipt handle when you request to delete the message. Otherwise, the message might not be deleted.

Delayed Message

CMQ Message Timers allow you to specify an initial invisibility period for a message that you add to a queue, which is called "in flight". For example, if you send a message with the DelaySeconds parameter set to 45, the message isn't visible to consumers for the first 45 seconds during which the message stays in the queue. The default value for DelaySeconds is 0.

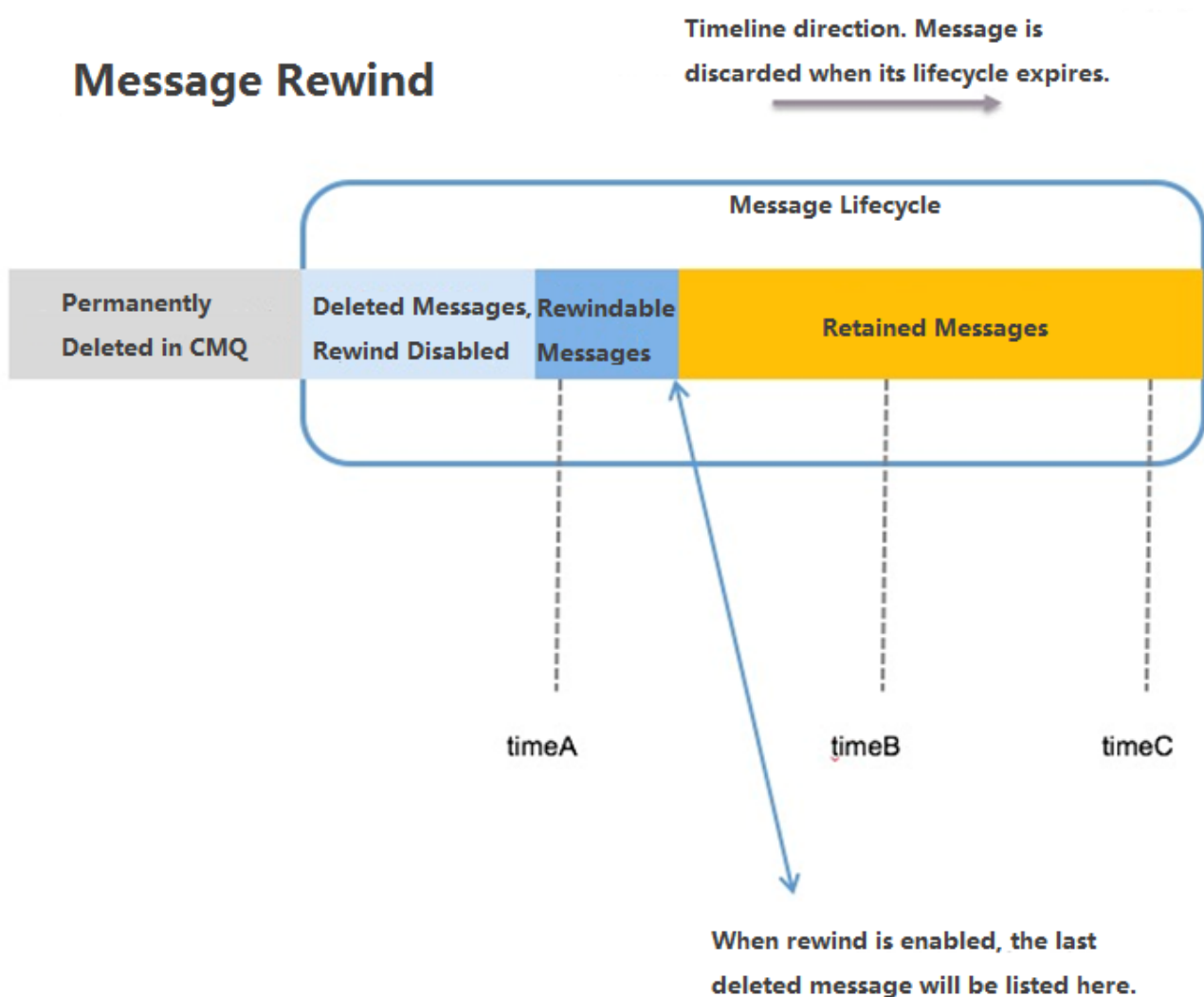
Configuration range for delay messages: When you specify a queue to produce messages, you can configure the QueueDelaySeconds parameter with a value range of 0-3600 seconds, which means the maximum invisibility period for a message is 1 hour. If this parameter is left empty, there is no delay.

Note: There can be a maximum of 1,000,000 inflight messages per queue. If you reach this limit, the extra messages will be invisible in the queue. Currently, this feature is not supported under the Topic mode.

Message Rewinding

CMQ provides a message rewind feature similar to kafka. With this feature, you can re-consume the message that you have consumed and deleted before, and easily perform business reconciliation, business system reboot, and other operations for your core financial businesses.

Feature Description

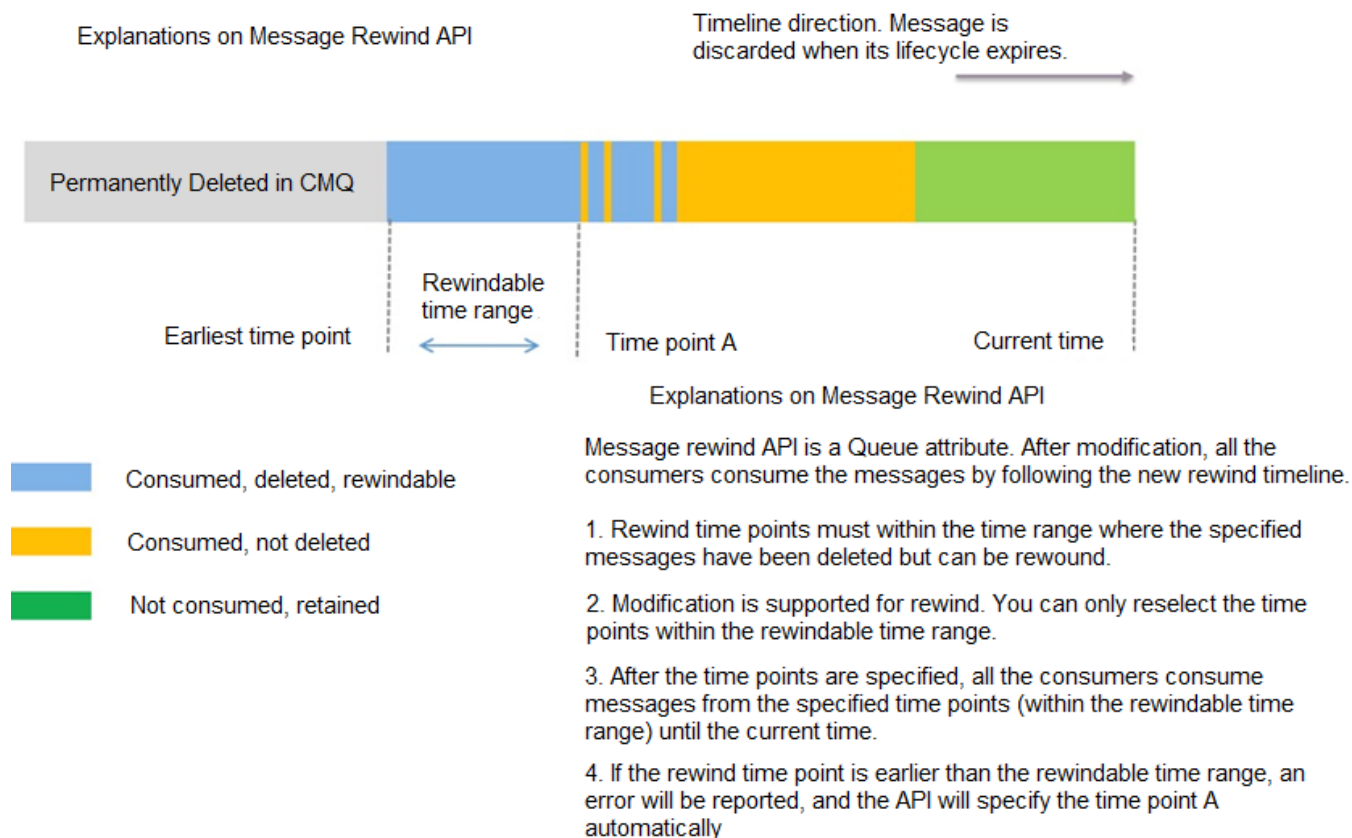


As shown above, the fragments in the blue box represents the message lifecycle. After the message rewind is enabled, the message that has been consumed and deleted by a consumer will enter message for rewind zone. CMQ backend will still retain this message. However, once exceeding the message lifecycle specified in Queue (if it is set to 1 day), the message will be deleted automatically and cannot be rewound. Specific product logic is as follows:

- **Enabling:** If the message rewind is not enabled, the message that has been consumed and deleted by a consumer will be deleted immediately. If it is enabled, you must specify the rewindable cycle for message rewind with a range no larger than that of the message lifecycle.
- **Milestone:** Based on the above policy, after the message rewind is enabled, the number of rewindable messages will increase as messages are repeatedly consumed and deleted by the consumer.
- **Disabling:** If the message rewind is disabled, the messages in the "message for rewind" zone will be deleted immediately and cannot be rewound.
- **Queue attribute:** Message rewind is a Queue attribute, which can be configured when a queue is created or in the configuration modification section. After specifying the rewind time, all the consumers will consume the messages produced after the specified time.
- **Billing:** After the message rewind is enabled, rewindable message will generate some retention fees. The unit price will be calculated together with the message retention fee.
- **Specified rewind time:** To rewind consumption, consumers need to specify the Queue Name and the specific rewind time. You need to rewind back from the furthest point of time. Time is key, and the message cannot be consumed retroactively. As shown above, the messages can only be consumed from timeA to timeB/timeC, and cannot be consumed in a reverse way.
- **Specified rewind time range:** 0 to 15 days. Only if the message rewind is enabled in the console, the deleted messages can be rewound. It is recommended that you keep the message rewind enabled for key applications. In addition, the message rewind cycle should be aligned with the message lifecycle.
- **The retained messages cannot be specified for rewind:** If a message is retained and not consumed, you cannot specify a specific location for consumption.

Rewindable Range

The maximum rewindable time = Current time - the configured rewind time
The messages cannot be rewound if produced before this time; otherwise, they can be rewound, as shown below:



Timeline

Message rewind The messages for rewind will be sorted based on their production time rather than the deletion time.

As shown in the figure below:



Creating a Queue

□When creating a queue, you need to specify the following attributes:

- 1) QueueName: Name of the queue
- 2) CreateTime: Creation time of the Queue
- 3) LastModifyTime: The last time when Queue attributes are modified.
- 4) PollingWaitSeconds: The number of waiting seconds for the messages to be received when using long-polling. When you are waiting for a response using long polling, a message consumption request will only return a response when a valid message is obtained or a long polling timeout is occurred, which is similar to the long polling used for Ajax request. Valid values: an integer from 0 to 30 seconds. Default is 0 (second).
- 5) VisibilityTimeout: The visibility timeout for the queue. Valid values: an integer from 0 to 43,200 seconds (12 hours). Each Message has a default VisibilityTimeout which will be counted after a worker receives the message. If the Message cannot be handled by the Worker within timeout, it may be received and handled by another Worker. Default is 30 (seconds).
- 6) MaxMsgSize: The maximum message size for the queue. The limit of how many bytes a message can contain when it is allowed to be sent to the queue. Valid values: An integer from 1,024 bytes (1 KB) up to 65,536 bytes (64 KB). Default is 64 (KB).
- 7) MsgRetentionSeconds: The number of seconds for which the message can be retained in the queue. The message will be deleted after this time has elapsed since the message has been sent to this queue, regardless of whether the message has been pulled or not. Valid values: An integer representing seconds, from 60 (1 minute) to 1,296,000 (15 days).
- 8) Queue API request address: This is an access address for the subscriber to pull messages in the CMQ message queue. It should be used together with a key. You can also use the address to configure queue attributes and modify parameters.

Producing Messages

□The producer needs to specify the following attributes when producing a message:

1) Queue name

2) Queue resource ID

3) Published content: Main body of the message, customers may enter the content according to their own needs. CMQ will not encode or modify this content