



APUSIC
固若长城
睿比世界

核心引擎使用者手冊

金蝶Apusic分布式消息隊列V2.0.6_for_rocketmq

版權所有 © 深圳市金蝶天燕雲計算股份有限公司2026。保留所有權利。

版權聲明

本文件所涉及的軟件著作權、版權等知識產權已依法進行了註冊，由金蝶天燕雲計算股份有限公司合法擁有。受《中華人民共和國著作權法》《計算機軟件保護條例》《知識產權保護條例》和相關國際版權條約、法律、法規以及其它知識產權法律和條約的保護。未經授權許可，不得非法使用。

免責聲明

本文件包含的版權資訊由金蝶天燕雲計算股份有限公司合法擁有，受法律的保護，金蝶天燕雲計算股份有限公司對本文件可能涉及到的非金蝶天燕雲計算股份有限公司的資訊不承擔任何責任。在法律允許的範圍內，您可以查閱並僅能夠在《中華人民共和國著作權法》規定的合法範圍內複製和打印本文件。任何單位和個人未經金蝶天燕雲計算股份有限公司書面授權許可，不得使用、修改、再發佈本文件的任何部分和內容，否則將被視為侵權，金蝶天燕雲計算股份有限公司有依法追究其責任的權利。

本文件如有更新，不另行通知。對本文件中的問題您可向金蝶天燕雲計算股份有限公司告知或查詢。未經本公司明確授予的任何權利均予保留。

商標聲明

 是深圳市金蝶天燕雲計算股份有限公司向中華人民共和國國家商標局申請註冊的註冊商標，註冊商標專用權由金蝶天燕合法擁有，受法律保護。未經金蝶天燕的書面許可，任何單位及個人不得以任何方式或理由對該商標的任何部分進行使用、複製、修改、傳播、抄錄或與其它產品捆綁使用銷售。凡侵犯金蝶天燕商標權的，金蝶天燕將依法追究其法律責任。本文件提及的其他所有商標或註冊商標，由各自的所有人擁有。

目錄

- 1 前言
 - 1.1 適用對象
 - 1.2 相關文檔
 - 1.3 技術支持
- 2 核心引擎目錄說明
- 3 Nameserver 配置說明
 - 3.1 基礎配置 (Basic Configuration)
 - 3.2 TLS配置
- 4 Broker 配置說明
 - 4.1 基礎配置 (Basic Configuration)
 - 4.2 NameServer相關配置
 - 4.3 存儲配置
 - 4.4 Prometheus配置
- 5 核心概念
 - 5.1 RocketMQ 架構
 - 5.2 消息模型
- 6 常見命令
 - 6.1 Topic 管理
 - 6.1.1 Topic 屬性
 - 6.1.2 創建 Topic
 - 6.1.3 查看 Topic
 - 6.1.4 刪除 Topic
 - 6.2 Group 管理
 - 6.2.1 創建 Group
 - 6.2.2 查看 Group
 - 6.2.3 刪除 Group
 - 6.3 消息查詢
 - 6.4 配置管理
 - 6.5 消息軌跡
 - 6.6 MessageQueue
 - 6.6.1 隊列分配策略
- 7 生產者 (Producer)
 - 7.1 發送方式
 - 7.2 消息屬性
 - 7.3 消息路由
- 8 消費者 (Consumer)
 - 8.1 消費模式
 - 8.1.1 集群消費 (Clustering)

- 8.1.2 廣播消費 (Broadcasting)
- 8.2 消費方式
- 8.3 消費位點 (Offset)
 - 8.3.1 位點類型
 - 8.3.2 位點管理
- 8.4 消息過濾
 - 8.4.1 Tag 過濾
 - 8.4.2 SQL92 過濾
- 9 高級消息特性
 - 9.1 順序消息
 - 9.1.1 全局順序
 - 9.1.2 分區順序
 - 9.2 事務消息
 - 9.3 定時/延時消息
 - 9.4 批量消息
- 10 集群與高可用
 - 10.1 集群狀態
 - 10.2 集群運維命令
- 11 安全配置
 - 11.1 ACL (1.0-適用4.x版本) 配置
 - 11.1.1 配置文件修改
 - 11.1.2 啟用 ACL
 - 11.2 ACL (2.0-適用5.x版本) 配置
 - 11.2.1 配置文件修改
 - 11.2.2 配置mqadmin工具
 - 11.2.3 創建業務用戶和授權
 - 11.2.4 驗證配置
 - 11.3 TLS 配置

1 前言

本文檔為金蝶Apusic分布式消息队列for RocketMQ（Apusic Distributed Message Queue for RocketMQ，簡稱：ADMQ for RocketMQ）消息引擎的用戶手冊，詳細介紹了ADMQ for RocketMQ消息引擎的功能使用、配置方法及管理操作等內容。

1.1 適用對象

本文檔適用於IT信息化業務負責人、研發經理、軟件項目經理、軟件架構師、運維工程師。

1.2 相關文檔

了解更多ADMQ for RocketMQ產品相關的信息，請參閱以下ADMQ for RocketMQ產品手冊文檔集：

序號	手冊文檔	說明
1	金蝶Apusic分布式消息队列for RocketMQ 快速使用手冊	簡單介紹了如何快速上手使用ADMQ for RocketMQ。
2	金蝶Apusic分布式消息队列for RocketMQ 安裝手冊	詳細介紹如何在各操作系統上安裝ADMQ for RocketMQ，以及ADMQ for RocketMQ服務啟停等操作。
3	金蝶Apusic分布式消息队列for RocketMQ 消息引擎用戶手冊	詳細介紹 ADMQ for RocketMQ 消息引擎相關功能的使用、配置、管理及配套工具的使用方法。
4	金蝶Apusic分布式消息队列for RocketMQ 管控制台用戶手冊	詳細介紹ADMQ for RocketMQ管控制台相關功能的使用和操作說明。
5	金蝶Apusic分布式消息队列for RocketMQ 開發手冊	詳細介紹基於各開發語言進行ADMQ for RocketMQ客戶端應用開發的說明。
6	金蝶Apusic分布式消息队列for RocketMQ 遷移手冊	詳細介紹從RocketMQ遷移到ADMQ for RocketMQ的說明。
7	金蝶Apusic分布式消息队列for RocketMQ 運維手冊	詳細介紹ADMQ for RocketMQ的監控、運維、安全加固等運維說明。
8	金蝶Apusic分布式消息队列for RocketMQ 性能優化手冊	詳細介紹ADMQ for RocketMQ性能調優的說明。

1.3 技術支持

ADMQ for RocketMQ產品提供全面的技術支持服務，您可以通過以下方式獲得技術支持：

- 網址：www.apusic.com
- 電話：400-855-5800
- 郵箱：support@apusic.com
- 金蝶雲社區：<https://vip.kingdee.com/?productId=73&productLineId=14&lang=zh-CN>

您在取得技術支持時，請提供如下信息：

1. 您的姓名
2. 公司信息與聯繫方式
3. 操作系統及其版本
4. 產品版本號
5. 出現異常及錯誤的日誌、截圖等詳細信息

2 核心引擎目錄說明

目錄名	說明
bin	存放啟動、停止及管理腳本
conf	存放配置文件 (如 <code>broker.conf</code> , <code>nameserver.properties</code>)
lib	存放 RoekctMQ 依賴包
logs	存放運行日誌 (Broker 日誌、NameServer 日誌等)

3 Nameserver 配置說明

3.1 基礎配置 (Basic Configuration)

參數	說明	生產環境建議
listenPort	Nameserver的監聽端口	默認使用9876
deployServerAddress	Nameserver的地址配置參數	默認使用127.0.0.1，可以更新成對應機器地址
clusterName	集群名稱	默認defaultCluster
managerUrl	管控台的接入地址	根據實際部署的地址IP填寫，如果沒有管控台，默認參數可不修改

3.2 TLS配置

核心作用：tls認證加密

參數	說明	生產環境建議
tls.enable	TLS認證開啟參數	默認false
tls.server.keyPath	存放服務器證書密鑰的地址	根據實際存放文件地址填寫
tls.server.keyPassword	服務器證書密鑰密碼	根據實際設置的密碼填寫
tls.server.certPath	存放服務器證書的地址	根據實際存放文件地址填寫
tls.server.trustCertPath	存放服務器信任證書的地址	根據實際存放文件地址填寫
tls.client.keyPath	存放客戶端密鑰的地址	根據實際存放文件地址填寫
tls.client.keyPassword	客戶端證書密鑰密碼	根據實際設置的密碼填寫
tls.client.certPath	存放客戶端證書的地址	根據實際存放文件地址填寫
tls.client.trustCertPath	存放客戶端信任證書的地址	根據實際存放文件地址填寫

4 Broker 配置說明

4.1 基礎配置 (Basic Configuration)

參數	說明	生產環境建議
brokerClusterName	集群名稱	默認集群名稱defaultCluster
brokerName	broker節點名稱	默認節點名稱broker-a
brokerRole	broker角色	默認ASYNC_MASTER
brokerId	broker id	根據實際部署的broker設置
brokerIP1	broker 映射IP	broker映射的IP
listenPort	監聽端口	默認端口10911
haListenPort	主從同步的監聽端口	默認端口9888
managerUrl	管控台的接入地址	根據實際部署的地址IP填寫，如果沒有管控台，默認參數可不修改

4.2 NameServer相關配置

參數	說明	生產環境建議
namesrvAddr	nameserver連接地址	nameserver組件連接地址

4.3 存儲配置

核心作用：存儲管理

參數	說明	生產環境建議
storePathRootDir	存儲路徑	根據實際使用設置
storePathCommitLog	commitlog存儲路徑	根據實際使用設置
storePathConsumerQueue	消費隊列存儲路徑	根據實際使用設置
storePathIndex	index存儲路徑	根據實際使用設置
log.dir	日誌存儲地址	根據實際使用設置

4.4 Prometheus配置

核心作用：監控運維

參數	說明	生產環境建議
enableMultiDispatch	是否開啟多維度消息分發的參數	建議開啟，設置成true

enablemq	是否開啟多維度消息分發的參數	建議開啟，設置成true
metricsPromExporterHost	prometheus監控地址	默認0.0.0.0
metricsExporterType	監控類型	默認PROM

5 核心概念

5.1 RocketMQ 架構

ADMQ for RocketMQ 完全兼容 RocketMQ 核心架構，主要包含以下組件：

組件	說明
NameServer	輕量級服務註冊與發現中心，維護 Broker 路由信息
Broker	消息存儲和轉發服務節點，分 Master 和 Slave
Producer	生產者，負責發送消息到 Broker
Consumer	消費者，負責從 Broker 接收消息
Topic	消息主題，消息的分類標識
MessageQueue	消息隊列，Topic 的分片單位

5.2 消息模型

```
Producer -> Topic -> MessageQueue -> Consumer
```

1. 生產者發送消息到指定 Topic
2. Topic 被劃分為多個 MessageQueue，分布在不同 Broker 上
3. 消費者訂閱 Topic，從 MessageQueue 拉取消息消費
4. 消費進度以 Offset 形式記錄

6 常見命令

6.1 Topic 管理

6.1.1 Topic 屬性

屬性	說明
讀寫隊列數	每個 Broker 上的隊列數量
權限	讀寫權限控制
順序類型	是否順序消息
消息類型	普通/事務/定時消息

6.1.2 創建 Topic

創建一個名為 `test-topic` 的主題，包含8個讀隊列，8個寫隊列

```
# 使用 mqadmin 創建 Topic
bin/mqadmin updateTopic -n localhost:9876 -c DefaultCluster -t test-topic -r 8 -w
8

# 參數說明
# -n: NameServer 地址
# -c: 集群名稱
# -t: Topic 名稱
# -r: 讀隊列數
# -w: 寫隊列數
```

6.1.3 查看 Topic

```
# 查看 Topic 列表
bin/mqadmin topicList -n localhost:9876

# 查看 Topic 路由信息
bin/mqadmin topicRoute -n localhost:9876 -t test-topic

# 查看 Topic 狀態
bin/mqadmin topicStatus -n localhost:9876 -t test-topic
```

6.1.4 刪除 Topic

```
bin/mqadmin deleteTopic -n localhost:9876 -c DefaultCluster -t test-topic
```

6.2 Group 管理

6.2.1 創建 Group

```
bin/mqadmin updateSubGroup -n localhost:9876 -b localhost:10911 -g groupName
```

6.2.2 查看 Group

查看所有訂閱組列表及進度

```
bin/mqadmin consumerProgress -n localhost:9876
```

查看訂閱組的連接

```
bin/mqadmin consumerConnection -n localhost:9876 -g groupName
```

6.2.3 刪除 Group

```
bin/mqadmin deleteSubGroup -n localhost:9876 -b localhost:10911 -g groupName
```

6.3 消息查詢

根據key查詢消息

```
bin/mqadmin queryMsgByKey -n localhost:9876 -t test-topic -k test-key
```

根據消息id查詢消息

```
bin/mqadmin queryMsgById -n localhost:9876 -i message-id
```

6.4 配置管理

獲取broker配置

```
# 配置管理
bin/mqadmin getBrokerConfig -n localhost:9876 -b broker-a:10911
```

更新broker配置

```
bin/mqadmin updateBrokerConfig -n localhost:9876 -b broker-a:10911 -k key -v  
value
```

6.5 消息軌跡

```
# 查看消息軌跡  
bin/mqadmin printMsg -n localhost:9876 -t test-topic -s "2026-01-01#00:00:00:000"  
-e "2026-01-02#00:00:00:000"
```

6.6 MessageQueue

MessageQueue 是 Topic 的邏輯分片，每個 Topic 在 Broker 上劃分為多個隊列。

6.6.1 隊列分配策略

策略	說明
輪詢	消息依次發送到各隊列
哈希	按 Key 哈希選擇隊列
順序	按指定規則選擇隊列

7 生產者 (Producer)

7.1 發送方式

發送方式	說明	適用場景
同步發送	發送後等待服務端響應	需要可靠性的場景
異步發送	發送後立即返回，通過回調獲取結果	需要高吞吐的場景
單向發送	發送後不等待響應	可靠性要求低的場景
批量發送	批量發送多條消息	需要高吞吐的場景
事務發送	兩階段提交發送	需要事務一致性的場景

7.2 消息屬性

屬性	說明
Topic	消息主題
Tag	消息標籤，用於過濾
Key	業務唯一標識
Body	消息體
Properties	擴展屬性

7.3 消息路由

生產者發送消息時，根據以下規則選擇 MessageQueue：

1. **未指定 Key**：輪詢選擇隊列
2. **指定 Key**：按 Key 哈希選擇隊列（同一 Key 的消息進入同一隊列）
3. **順序消息**：按消息隊列選擇器選擇隊列

8 消費者 (Consumer)

8.1 消費模式

8.1.1 集群消費 (Clustering)

- 同一消費者組內消息僅被消費一次
- 消息被組內某個消費者消費
- 適用場景：大部分業務場景

8.1.2 廣播消費 (Broadcasting)

- 同一消費者組內每條消息都被所有消費者消費
- 適用場景：本地緩存更新、配置同步

8.2 消費方式

消費方式	說明
推模式 (Push)	Broker 主動推送消息到消費者
拉模式 (Pull)	消費者主動拉取消息

8.3 消費位點 (Offset)

8.3.1 位點類型

類型	說明
集群位點	消費進度存儲在 Broker 上，集群共享
廣播位點	消費進度存儲在消費者本地，各自維護

8.3.2 位點管理

```
# 查看消費者組消費進度
bin/mqadmin consumerProgress -n localhost:9876 -g test-group

# 重置消費位點
bin/mqadmin resetOffsetByTime -n localhost:9876 -g test-group -t test-topic -s
"2026-01-01#00:00:00:000"
```

8.4 消息過濾

8.4.1 Tag 過濾

消費者訂閱時指定 Tag，只接收匹配的消息：

```
// 訂閱所有 Tag
consumer.subscribe("test-topic", "*");

// 訂閱指定 Tag
consumer.subscribe("test-topic", "tag-a");

// 訂閱多個 Tag (用 || 分隔)
consumer.subscribe("test-topic", "tag-a || tag-b");
```

8.4.2 SQL92 過濾

使用 SQL92 語法進行複雜過濾：

```
// 訂閱滿足 SQL 條件的消息
consumer.subscribe("test-topic", MessageSelector.bySql("age > 18 AND city = 'Beijing'"));
```

9 高級消息特性

9.1 順序消息

9.1.1 全局順序

所有消息按發送順序消費，性能較低。

9.1.2 分區順序

同一分區（MessageQueue）內的消息按順序消費，不同分區間無序。

```
// 發送順序消息
Message msg = new Message("order-topic", "tag-a", "order-1",
    "content".getBytes());
producer.send(msg, new MessageQueueSelector() {
    @Override
    public MessageQueue select(List<MessageQueue> mqs, Message msg, Object arg) {
        // 按訂單 ID 選擇隊列
        Long orderId = (Long) arg;
        long index = orderId % mqs.size();
        return mqs.get((int) index);
    }
}, orderId);

// 消費順序消息
consumer.registerMessageListener(new MessageListenerOrderly() {
    @Override
    public ConsumeOrderlyStatus consumeMessage(List<MessageExt> msgs,
        ConsumeOrderlyContext context) {
        // 按順序消費
        return ConsumeOrderlyStatus.SUCCESS;
    }
});
```

9.2 事務消息

基於兩階段提交實現分布式事務：

```
// 創建事務生產者
TransactionMQProducer producer = new TransactionMQProducer("tx-group");
producer.setTransactionListener(new TransactionListener() {
    @Override
```

```

public LocalTransactionState executeLocalTransaction(Message msg, Object arg)
{
    // 執行本地事務
    try {
        // 執行業務邏輯
        return LocalTransactionState.COMMIT_MESSAGE;
    } catch (Exception e) {
        return LocalTransactionState.ROLLBACK_MESSAGE;
    }
}

@Override
public LocalTransactionState checkLocalTransaction(MessageExt msg) {
    // 回查本地事務狀態
    if (isTransactionSuccess(msg)) {
        return LocalTransactionState.COMMIT_MESSAGE;
    }
    return LocalTransactionState.ROLLBACK_MESSAGE;
}
});

```

9.3 定時/延時消息

支持指定時間投遞消息：

```

// 發送延時消息 (延時級別)
Message msg = new Message("delay-topic", "tag-a", "content".getBytes());
// 設置延時級別 (1s 5s 10s 30s 1m 2m 3m 4m 5m 6m 7m 8m 9m 10m 20m 30m 1h 2h)
msg.setDelayTimeLevel(3); // 延時 10 秒
producer.send(msg);

// RocketMQ 5.x 支持任意時間
msg.setDelayTimeMs(3600000); // 延時 1 小時

```

9.4 批量消息

批量發送消息提高吞吐：

```

List<Message> messages = new ArrayList<>();
for (int i = 0; i < 100; i++) {
    messages.add(new Message("batch-topic", "tag-a", ("msg-" + i).getBytes()));
}

```

```
}  
producer.send(messages);
```

10 集群與高可用

10.1 集群狀態

```
# 查看 Broker 狀態
bin/mqadmin brokerStatus -n localhost:9876 -b broker-a:10911
```

10.2 集群運維命令

```
# 查看集群狀態
bin/mqadmin clusterList -n localhost:9876

# 查看 Broker 狀態
bin/mqadmin brokerStatus -n localhost:9876 -b broker-a:10911

# 查看 Topic 列表
bin/mqadmin topicList -n localhost:9876

# 查看消費者組連接
bin/mqadmin consumerConnection -n localhost:9876 -g test-group

# 查看生產者連接
bin/mqadmin producerConnection -n localhost:9876 -t test-topic -g test-group
```

11 安全配置

11.1 ACL (1.0-適用4.x版本) 配置

11.1.1 配置文件修改

```
# conf/plain_acl.yml
globalWhiteRemoteAddresses:
  - 10.*.*.*
  - 192.168.*.*

accounts:
  - accessKey: test-user
    secretKey: test-password
    whiteRemoteAddress:
      admin: false
    defaultTopicPerm: DENY
    defaultGroupPerm: SUB
    topicPerms:
      - test-topic=CREATE+WRITE+READ
    groupPerms:
      - test-group=READ
```

11.1.2 啟用 ACL

```
# 啟用 ACL 插件
bin/mqadmin updateBrokerConfig -n localhost:9876 -b broker-a:10911 -k aclEnable -
v true
```

11.2 ACL (2.0-適用5.x版本) 配置

11.2.1 配置文件修改

修改broker.conf

```
# 啟用認證
authenticationEnabled = true
authenticationMetadataProvider =
org.apache.rocketmq.auth.authentication.provider.LocalAuthenticationMetadataProvider

# 啟用授權
```

```

authorizationEnabled = true
authorizationMetadataProvider =
org.apache.rocketmq.auth.authorization.provider.LocalAuthorizationMetadataProvider

# 初始化管理員用戶（首次啟動自動創建）
initAuthenticationUser = {"username":"rocketmq","password":"12345678"}

# 組件間認證憑證（用於Broker主從同步、集群內部通信等）
innerClientAuthenticationCredentials =
{"accessKey":"rocketmq","secretKey":"12345678"}

```

通過修改broker.conf，需要重啟集群

11.2.2 配置mqadmin工具

啟用ACL後，需要配置mqadmin工具的認證憑證才能執行管理命令。

編輯 `conf/tools.yml` 文件：

```

# 使用初始化的管理員用戶憑證
accessKey: rocketmq
secretKey: 12345678

```

11.2.3 創建業務用戶和授權

生產者

```

# 創建生產者用戶
bin/mqadmin createUser -n 127.0.0.1:9876 -c DefaultCluster \
  -u producer_user \
  -p producer123 \
  -t Normal

# 授予Topic發送權限
bin/mqadmin createAcl -n 127.0.0.1:9876 -c DefaultCluster \
  -s User:producer_user \
  -r Topic:TestTopic \
  -a Pub \
  -d Allow

```

消費者

```

# 創建消費者用戶
bin/mqadmin createUser -n 127.0.0.1:9876 -c DefaultCluster \

```

```
-u consumer_user -p consumer123 -t Normal

# 授予消費者消費權限
bin/mqadmin createAcl -n 127.0.0.1:9876 -c DefaultCluster \
  -s User:consumer_user -r Topic:TestTopic,Group:TestGroup -a Sub -d Allow
```

11.2.4 驗證配置

使用Java客戶端發送一條測試消息：

```
SessionCredentials credentials = new SessionCredentials("producer_user",
"producer123");
StaticSessionCredentialsProvider credentialsProvider =
    new StaticSessionCredentialsProvider(credentials);

ClientConfiguration clientConfiguration = ClientConfiguration.newBuilder()
    .setEndpoints("127.0.0.1:10911")
    .setCredentialProvider(credentialsProvider)
    .build();
// ... 創建Producer並發送消息
```

11.3 TLS 配置

```
# broker.conf
tls.server.mode=enforcing
tls.server.certPath=/path/to/server.crt
tls.server.keyPath=/path/to/server.key
tls.server.trustCertPath=/path/to/ca.crt

tls.client.keyPath=conf/auth/client.key
tls.client.keyPassword=123456
tls.client.certPath=conf/auth/client.pem
tls.client.trustCertPath=conf/auth/ca.pem
```

全國統一服務熱線
4008-555-800

金蝶天燕雲計算股份有限公司（簡稱“金蝶天燕”）成立於2000年，前身為“金蝶中間件公司”，是金蝶集團旗下基礎軟件平台提供商，雲計算國家標準制定企業，國家信創產業核心軟件企業。金蝶天燕是國家863重點研發計劃與核高基重大專項承接企業，也是“兩網一站四庫十二金”國家重點工程的基礎平台提供商，產品廣泛應用於政府、軍工、金融、能源等關鍵行業，累計服務客戶總數超過10萬家。



雲計算國家標準制定企業
金蝶集團旗下基礎軟件企業
資訊技術應用創新核心企業
官網：www.apusic.com

