



核心引擎使用者手冊

金蝶Apusic分布式消息队列V2.0.6_for_rabbitmq

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 相關文件
 - 1.3 技術支援
- 2 核心概念
 - 2.1 AMQP 協定
 - 2.2 訊息模型
- 3 交換器 (Exchange)
 - 3.1 交換器類型
 - 3.1.1 Direct Exchange
 - 3.1.2 Fanout Exchange
 - 3.1.3 Topic Exchange
 - 3.1.4 Headers Exchange
 - 3.2 交換器屬性
- 4 佇列 (Queue)
 - 4.1 佇列類型
 - 4.1.1 Classic Queue
 - 4.1.2 Quorum Queue
 - 4.1.3 Stream Queue
 - 4.2 佇列屬性
 - 4.3 死信佇列
- 5 訊息機制
 - 5.1 訊息確認
 - 5.1.1 自動確認 (Auto Ack)
 - 5.1.2 手動確認 (Manual Ack)
 - 5.2 訊息持久化
 - 5.2.1 佇列持久化
 - 5.2.2 訊息持久化
 - 5.3 訊息優先級
 - 5.4 延遲佇列
- 6 叢集與高可用
 - 6.1 叢集模式

- 6.1.1 普通叢集
- 6.1.2 鏡像佇列
- 6.1.3 Quorum Queue
- 6.2 叢集運維命令
- 7 安全配置
 - 7.1 用戶認證
 - 7.1.1 建立用戶
 - 7.1.2 配置權限
 - 7.2 SSL/TLS 配置
 - 7.3 存取控制
- 8 效能調校參數
 - 8.1 記憶體管理
 - 8.2 磁碟管理
 - 8.3 連線與通道
 - 8.4 佇列參數
- 9 常用工具
 - 9.1 bin/admq rabbitmq admin ctl
 - 9.2 bin/admq rabbitmq admin diagnostics
 - 9.3 rabbitmq-plugins

1 前言

本文件為金蝶 Apusic 分散式訊息佇列 for RabbitMQ（Apusic Distributed Message Queue，簡稱：ADMQ for RabbitMQ）訊息引擎的用戶手冊，詳細介紹了 ADMQ for RabbitMQ 訊息引擎的功能使用、配置方法及管理操作等內容。

1.1 適用對象

本文件適用於 IT 資訊化業務負責人、研發經理、軟體專案經理、軟體架構師、運維工程師。

1.2 相關文件

了解更多 ADMQ for RabbitMQ 產品相關的資訊，請參閱以下 ADMQ for RabbitMQ 產品手冊文件集：

序號	手冊文件	說明
1	金蝶 Apusic 分散式訊息佇列 for RabbitMQ 快速使用手冊	簡單介紹了如何快速上手使用 ADMQ for RabbitMQ。
2	金蝶 Apusic 分散式訊息佇列 for RabbitMQ 安裝手冊	詳細介紹如何在各作業系統上安裝 ADMQ for RabbitMQ，以及 ADMQ for RabbitMQ 服務啟停等操作。
3	金蝶 Apusic 分散式訊息佇列 for RabbitMQ 訊息引擎用戶手冊	詳細介紹 ADMQ for RabbitMQ 訊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 分散式訊息佇列 for RabbitMQ 管控台用戶手冊	詳細介紹 ADMQ for RabbitMQ 管控台相關功能的使用和操作說明。
5	金蝶 Apusic 分散式訊息佇列 for RabbitMQ 開發手冊	詳細介紹基於各開發語言進行 ADMQ for RabbitMQ 用戶端應用開發的說明。
6	金蝶 Apusic 分散式訊息佇列 for RabbitMQ 遷移手冊	詳細介紹從 RabbitMQ 遷移到 ADMQ for RabbitMQ 的說明。
7	金蝶 Apusic 分散式訊息佇列 for RabbitMQ 運維手冊	詳細介紹 ADMQ for RabbitMQ 的監控、運維、安全強化等運維說明。
8	金蝶 Apusic 分散式訊息佇列 for RabbitMQ 效能最佳化手冊	詳細介紹 ADMQ for RabbitMQ 效能調校的說明。

1.3 技術支援

ADMQ for RabbitMQ 產品提供全面的技術支援服務，您可以透過以下方式獲得技術支援：

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

您在取得技術支援時，請提供如下資訊：

1. 您的姓名
2. 公司資訊與聯絡方式
3. 作業系統及其版本
4. 產品版本號
5. 出現異常及錯誤的日誌、截圖等詳細資訊

2 核心概念

2.1 AMQP 協定

ADMQ for RabbitMQ 完全相容 AMQP 0-9-1 協定，支援以下核心概念：

概念	說明
Producer	生產者，負責傳送訊息到 Exchange
Consumer	消費者，負責從 Queue 接收訊息
Exchange	交換器，負責接收和路由訊息到 Queue
Queue	佇列，負責儲存訊息
Binding	綁定關係，定義 Exchange 到 Queue 的路由規則
Routing Key	路由鍵，用於訊息路由匹配
Channel	通道，輕量級的連線，建立在真實連線之上

2.2 訊息模型

```
Producer -> Exchange -> [Binding/Routing Key] -> Queue -> Consumer
```

1. 生產者傳送訊息到 Exchange
2. Exchange 根據類型和 Routing Key 決定如何路由訊息
3. 訊息透過 Binding 路由到匹配的 Queue
4. 消費者從 Queue 接收訊息

3 交換器 (Exchange)

3.1 交換器類型

3.1.1 Direct Exchange

- 路由方式：精確匹配 Routing Key
- 適用場景：點對點訊息、簡單路由
- 範例：Routing Key 為 "order.create" 的訊息只會路由到綁定鍵為 "order.create" 的佇列

3.1.2 Fanout Exchange

- 路由方式：廣播到所有綁定的佇列，忽略 Routing Key
- 適用場景：廣播通知、群發訊息、快取重新整理
- 範例：傳送一條訊息到 Fanout Exchange，所有綁定的佇列都會收到

3.1.3 Topic Exchange

- 路由方式：模式匹配 Routing Key，支援萬用字元
 - * 匹配一個單詞
 - # 匹配零個或多個單詞
- 適用場景：日誌分類、業務事件路由、多層級訂閱
- 範例：
 - 綁定鍵 `order.*` 匹配 `order.create`、`order.update`
 - 綁定鍵 `order.#` 匹配 `order.create`、`order.create.success`

3.1.4 Headers Exchange

- 路由方式：基於訊息頭 (Headers) 進行匹配
- 適用場景：需要複雜條件路由，不依賴 Routing Key
- 範例：匹配 header 中 `x-match=all` 且 `type=order`、`status=paid`

3.2 交換器屬性

屬性	說明
持久化 (Durable)	交換器是否在服務重啟後保留
自動刪除 (Auto-delete)	當最後一個佇列解綁後自動刪除
內部 (Internal)	只允許其他交換器綁定，不允許直接發布訊息

4 佇列 (Queue)

4.1 佇列類型

4.1.1 Classic Queue

- **特點：**傳統佇列類型，支援鏡像模式
- **適用場景：**一般業務場景，對吞吐量要求不是極高的場景
- **配置選項：**
 - 鏡像模式：在多個節點間同步佇列資料
 - 持久化：訊息持久化到磁碟

4.1.2 Quorum Queue

- **特點：**基於 Raft 共識演算法的高可用佇列，自動實現多副本
- **適用場景：**對高可用和資料安全要求高的場景
- **優勢：**
 - 自動故障轉移
 - 強一致性保證
 - 無需配置鏡像策略
- **限制：**
 - 不支援獨佔佇列
 - 不支援訊息 TTL (佇列級 TTL 支援)
 - 不支援優先級佇列

4.1.3 Stream Queue

- **特點：**只附加日誌類型佇列，支援多次讀取
- **適用場景：**高吞吐量、訊息回放、日誌收集場景
- **優勢：**
 - 極高的寫入吞吐量
 - 支援多個消費者獨立讀取
 - 訊息不會自動刪除，可配置保留策略
- **特性：**
 - 基於偏移量 (offset) 消費
 - 支援時間戳定位
 - 訊息持久化儲存

4.2 佇列屬性

屬性	說明
持久化 (Durable)	佇列是否在服務重啟後保留
自動刪除 (Auto-delete)	當最後一個消費者中斷後自動刪除
排他性 (Exclusive)	只允許建立者連線使用
訊息 TTL	訊息在佇列中的存活時間
佇列 TTL	佇列在空閒多久後自動刪除
最大長度	佇列最大訊息數限制
最大位元組	佇列最大位元組數限制
死信交換器 (DLX)	訊息被拒絕或過期後轉發的交換器
死信路由鍵 (DLK)	死信訊息使用的路由鍵

4.3 死信佇列

死信佇列用於處理無法正常消費的訊息。

訊息成為死信的幾種情況：

1. 訊息被消費者拒絕 (basic.reject 或 basic.nack) ， 且 requeue=false
2. 訊息過期 (TTL 到期)
3. 佇列達到最大長度，新訊息被擠出

配置死信佇列：

1. 建立死信交換器 (DLX)
2. 建立死信佇列並綁定到 DLX
3. 在業務佇列中設定 `x-dead-letter-exchange` 和 `x-dead-letter-routing-key`

5 訊息機制

5.1 訊息確認

5.1.1 自動確認 (Auto Ack)

- 訊息傳送給消費者後立即從佇列刪除
- 優點：效能高，延遲低
- 缺點：消費者處理失敗時訊息遺失

5.1.2 手動確認 (Manual Ack)

- 消費者處理完成後傳送確認 (basic.ack)
- 支援批次確認 (multiple=true)
- 支援否定確認 (basic.nack) ，可選擇是否重新入隊
- 優點：保證訊息不遺失
- 缺點：效能略低

5.2 訊息持久化

5.2.1 佇列持久化

建立佇列時設定 durable=true，佇列元資料會持久化到磁碟。

5.2.2 訊息持久化

傳送訊息時設定 delivery_mode=2，訊息內容會持久化到磁碟。

```
channel.basicPublish(EXCHANGE, ROUTING_KEY,  
    MessageProperties.PERSISTENT_TEXT_PLAIN,  
    message.getBytes());
```

5.3 訊息優先級

支援為訊息設定優先級 (0-255) 。

1. 建立佇列時設定 x-max-priority
2. 傳送訊息時設定 priority

```
Map<String, Object> args = new HashMap<>();
args.put("x-max-priority", 10);
channel.queueDeclare(Queue, true, false, false, args);

AMQP.BasicProperties props = new AMQP.BasicProperties.Builder()
    .priority(5)
    .build();
channel.basicPublish(EXCHANGE, ROUTING_KEY, props,
    message.getBytes());
```

5.4 延遲佇列

支援延遲訊息投遞。

1. 使用死信交換器 + TTL 實現延遲佇列
2. 或使用外掛支援的延遲交換器 (x-delayed-message)

6 叢集與高可用

6.1 叢集模式

6.1.1 普通叢集

- 節點間共享元資料 (Exchange、Binding、Queue 定義)
- 佇列資料僅儲存在宣告佇列的節點上
- 優點：資源佔用低
- 缺點：節點故障時該節點上的佇列不可用

6.1.2 鏡像佇列

- Classic Queue 支援鏡像模式
- 佇列資料同步到多個節點
- 配置策略：

```
# 設定所有佇列鏡像到所有節點
bin/admq rabbitmq admin ctl set_policy ha-all "^" '{"ha-mode":"all"}'

# 設定佇列鏡像到指定數量節點
bin/admq rabbitmq admin ctl set_policy ha-two "^" '{"ha-mode":"exactly","ha-params":2}'
```

6.1.3 Quorum Queue

- 基於 Raft 共識演算法自動實現多副本
- 無需配置鏡像策略
- 自動 Leader 選舉和故障轉移

6.2 叢集運維命令

```
# 檢視叢集狀態
bin/admq rabbitmq admin ctl cluster_status

# 新增節點到叢集
bin/admq rabbitmq admin ctl join_cluster rabbit@node1
```

移除節點

```
bin/admq rabbitmq admin ctl forget_cluster_node rabbit@node2
```

檢視佇列詳情

```
bin/admq rabbitmq admin ctl list_queues name pid slave_pids  
synchronised_slave_pids
```

檢視連線

```
bin/admq rabbitmq admin ctl list_connections peer_host peer_port  
state
```

7 安全配置

7.1 用戶認證

7.1.1 建立用戶

```
bin/admq rabbitmq admin ctl add_user username password
bin/admq rabbitmq admin ctl set_user_tags username administrator
```

7.1.2 配置權限

```
bin/admq rabbitmq admin ctl set_permissions -p /vhost username ".*"
".*" ".*"
```

權限格式：

- 第一個參數：Configure 權限（建立/刪除資源）
- 第二個參數：Write 權限（傳送訊息）
- 第三個參數：Read 權限（接收訊息）

7.2 SSL/TLS 配置

```
# 配置 SSL 監聽器
listeners.ssl.default = 5671
ssl_options.cacertfile = /path/to/ca_certificate.pem
ssl_options.certfile = /path/to/server_certificate.pem
ssl_options.keyfile = /path/to/server_key.pem
ssl_options.verify = verify_peer
ssl_options.fail_if_no_peer_cert = false
```

7.3 存取控制

支援基於 Vhost 的資源隔離，每個用戶在不同 Vhost 下可擁有不同權限。

8 效能調校參數

8.1 記憶體管理

參數	說明	預設值
vm_memory_high_watermark	記憶體使用上限比例	0.4
vm_memory_high_watermark_paging_ratio	開始分頁的記憶體比例	0.5

8.2 磁碟管理

參數	說明	預設值
disk_free_limit.absolute	磁碟空間下限	50MB
disk_free_limit.relative	磁碟空間下限比例	2.0

8.3 連線與通道

參數	說明	預設值
channel_max	每個連線的最大通道數	2047
connection_max	最大連線數	infinity
heartbeat	心跳間隔（秒）	60

8.4 佇列參數

參數	說明	預設值
queue_master_locator	佇列 Leader 分配策略	min-masters
lazy_queue_explicit_gc_run_operation_threshold	惰性佇列 GC 閾值	1000

9 常用工具

9.1 bin/admq rabbitmq admin ctl

RabbitMQ 命令列管理工具。

服務管理

```
bin/admq rabbitmq admin ctl start_app
```

```
bin/admq rabbitmq admin ctl stop_app
```

```
bin/admq rabbitmq admin ctl status
```

用戶管理

```
bin/admq rabbitmq admin ctl list_users
```

```
bin/admq rabbitmq admin ctl add_user user password
```

```
bin/admq rabbitmq admin ctl delete_user user
```

```
bin/admq rabbitmq admin ctl change_password user newpassword
```

佇列管理

```
bin/admq rabbitmq admin ctl list_queues
```

```
bin/admq rabbitmq admin ctl list_queues name messages consumers
```

交換器管理

```
bin/admq rabbitmq admin ctl list_exchanges name type
```

連線管理

```
bin/admq rabbitmq admin ctl list_connections peer_host state
```

9.2 bin/admq rabbitmq admin diagnostics

診斷工具。

檢查節點健康

```
bin/admq rabbitmq admin diagnostics health_check
```

檢視節點狀態

```
bin/admq rabbitmq admin diagnostics status

# 檢視記憶體使用
bin/admq rabbitmq admin diagnostics memory_breakdown

# 檢視叢集狀態
bin/admq rabbitmq admin diagnostics cluster_status
```

9.3 rabbitmq-plugins

外掛管理工具。

```
# 列出外掛
rabbitmq-plugins list

# 啟用外掛
rabbitmq-plugins enable rabbitmq_management
rabbitmq-plugins enable rabbitmq_shovel
rabbitmq-plugins enable rabbitmq_federation

# 停用外掛
rabbitmq-plugins disable rabbitmq_management
```

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

