



APUSIC
固若长城
睿比世界

Product Introduction

金蝶Apusic分布式消息队列 V2.0.6_for_rabbitmq

Copyright Notice

All intellectual property rights including software copyrights involved in this document have been legally registered and duly owned by Kingdee Apusic Cloud Computing Co., Ltd. Such rights are protected by Copyright Law of the People's Republic of China, Regulations on Protection of Computer Software, Regulations on Intellectual Property Protection, relevant international copyright conventions, laws, regulations and other intellectual property legislations. Unauthorized use is strictly prohibited.

Disclaimer

All copyright information contained herein is legally owned by Kingdee Apusic Cloud Computing Co., Ltd. and protected by applicable laws. The Company shall bear no liability for any information unrelated to Kingdee Apusic contained in this document. You may view, copy and print this document only within the scope permitted by Copyright Law of the People's Republic of China. No entity or individual may use, modify or republish any part or content of this document without written authorization from Kingdee Apusic Cloud Computing Co., Ltd. Any violation shall constitute infringement, and the Company reserves the right to pursue legal liabilities accordingly.

Updates to this document shall not be notified separately. You may consult the Company regarding issues related to this document. All rights not expressly granted by the Company are reserved.

Trademark Notice

 is a registered trademark filed with the National Intellectual Property Administration of PRC by Kingdee Apusic Cloud Computing Co., Ltd. Shenzhen. Exclusive rights to the registered trademark are legally owned by Kingdee Apusic and protected by law. Without written consent of Kingdee Apusic, no entity or individual shall use, copy, modify, disseminate, transcribe, or bundle and sell any part of this trademark with other products in any manner or for any reason. Kingdee Apusic shall pursue legal liabilities against any infringement of its trademark rights. All other trademarks or registered trademarks mentioned in this document belong to their respective owners.

Contents

- 1 Preface
 - 1.1 Target Audience
 - 1.2 Related Documents
 - 1.3 Technical Support
- 2 Product Introduction
 - 2.1 Product Positioning
 - 2.2 Core Value
 - 2.3 Core Capabilities
- 3 Product Architecture
 - 3.1 Architecture Overview
 - 3.1.1 Console
 - 3.1.2 Core Engine
 - 3.1.3 Distributed Coordination
- 4 Product Advantages
- 5 Application Scenarios

1 Preface

This document is a product introduction to Kingdee Apusic Distributed Message Queue for RabbitMQ (Apusic Distributed Message Queue, referred to as: ADMQ for RabbitMQ), helping users quickly understand the features, technical characteristics, and typical application scenarios of the ADMQ for RabbitMQ product.

1.1 Target Audience

This document is intended for IT information business leaders, R&D managers, software project managers, software architects, and operations engineers.

1.2 Related Documents

For more information about ADMQ for RabbitMQ, please refer to the following product documentation set:

No.	Document	Description
1	Kingdee Apusic Distributed Message Queue for RabbitMQ Quick Start Guide	Briefly introduces how to get started with ADMQ for RabbitMQ.
2	Kingdee Apusic Distributed Message Queue for RabbitMQ Installation Guide	Details how to install ADMQ for RabbitMQ on various operating systems, including starting and stopping the ADMQ for RabbitMQ service.
3	Kingdee Apusic Distributed Message Queue for RabbitMQ Message Engine User Guide	Details the usage, configuration, management, and companion tools of the ADMQ for RabbitMQ message engine.
4	Kingdee Apusic Distributed Message Queue for RabbitMQ Console User Guide	Details the usage and operation of the ADMQ for RabbitMQ console.
5	Kingdee Apusic Distributed Message Queue for RabbitMQ Development Guide	Details how to develop ADMQ for RabbitMQ client applications in various languages.
6	Kingdee Apusic Distributed Message Queue for RabbitMQ Migration Guide	Details the migration process from RabbitMQ to ADMQ for RabbitMQ.
7	Kingdee Apusic Distributed Message Queue for RabbitMQ Operations Guide	Details the monitoring, operations, and security hardening of ADMQ for RabbitMQ.
8	Kingdee Apusic Distributed Message Queue for RabbitMQ Performance	Details performance tuning for ADMQ for RabbitMQ.

1.3 Technical Support

ADMQ for RabbitMQ provides comprehensive technical support services. You can obtain technical support through the following channels:

- Website: www.apusic.com
- Phone: 400-855-5800
- Email: support@apusic.com
- Kingdee Cloud Community: <https://vip.kingdee.com/?productId=73&productLineId=14&lang=zh-CN>

When requesting technical support, please provide the following information:

1. Your name
2. Company information and contact details
3. Operating system and version
4. Product version number
5. Detailed logs, screenshots, and other information regarding the error or exception

2 Product Introduction

2.1 Product Positioning

Kingdee Apusic Distributed Message Queue for RabbitMQ (Apusic Distributed Message Queue, referred to as: ADMQ for RabbitMQ) is a cloud-native message middleware designed as a replacement for RabbitMQ in domestic IT (Xinchuang) environments. It fully follows the core architecture design of RabbitMQ and the AMQP protocol specification, achieving full compatibility with protocols, clients, ecosystem tools, and business scenarios. It can replace existing RabbitMQ clusters without modification, meeting the requirements of Xinchuang projects.

2.2 Core Value

It inherits RabbitMQ's core features of distribution, high reliability, and flexible routing, while adapting to Xinchuang software and hardware environments. It provides more stable operation guarantees, more convenient operational support, and more comprehensive security capabilities, making it the preferred solution for replacing RabbitMQ in Xinchuang projects.

2.3 Core Capabilities

Core Capability	Description
Full AMQP Protocol Compatibility	Fully implements the AMQP 0-9-1 protocol, supporting core protocols such as production/consumption, queue binding, and message acknowledgment, with behavior identical to native RabbitMQ.
Full-Version Client Adaptation	Supports Java, Python, Go, Node.js, .NET and other RabbitMQ official clients as well as third-party open-source clients, with zero code modification required.
Seamless Ecosystem Tool Reuse	Compatible with native tools such as RabbitMQ Management Plugin, bin/admq rabbitmq admin ctl, bin/admq rabbitmq admin diagnostics, and supports data synchronization tools such as Shovel and Federation.
Deep Xinchuang Environment Adaptation	Adapts to mainstream Xinchuang operating systems (Kylin, UOS, etc.), Xinchuang chips (ARM, RISC-V, etc.), and is compatible with Xinchuang databases and middleware ecosystems.
RabbitMQ Core Feature Inheritance	Supports multiple exchange types (Direct, Fanout, Topic, Headers), multiple queue types (Classic, Quorum, Stream), message acknowledgment, persistence, priority queues, and other native core features.

3 Product Architecture

3.1 Architecture Overview

The ADMQ for RabbitMQ product adopts the classic RabbitMQ distributed architecture, fully following the AMQP protocol and message routing model. The overall product architecture is divided into three modules: Console, Core Engine, and Distributed Coordination, featuring high reliability, high availability, and scalability.

3.1.1 Console

- The Console is a visual operation interface for operations personnel and administrators, providing full lifecycle management capabilities:
- Cluster management: Access and monitor the status of message cluster nodes.
- User permissions: Role-based access control (RBAC) for operational security.
- Vhost management: Create, delete, and configure virtual hosts (Vhost) for resource isolation.
- Exchange management: Create, delete, and configure exchanges, and manage routing rules.
- Queue management: Create, delete, and configure queues, and manage queue attributes and bindings.
- Message search: Search historical messages by time, routing key, content, and other criteria.

3.1.2 Core Engine

The Core Engine is the heart of message processing, compatible with the AMQP protocol, ensuring reliable message transmission and storage:

AMQP Protocol Access & Security Access Control

- Provides a standard AMQP protocol compatibility layer, supporting seamless migration of existing RabbitMQ clients.
- Integrates authentication (e.g., SASL/PLAIN, SCRAM), authorization, and transport encryption (SSL/TLS) for access security.

Message Processing Service

- Message production: Receives messages from producers, performs validation and routing.
- Message consumption: Pushes messages to consumers, supporting manual and automatic acknowledgment.
- Message persistence: Writes messages to disk, supporting high-reliability writes and efficient retrieval, ensuring no data loss.

- Metadata service: Manages metadata such as Vhost, Exchange, Queue, and Binding, ensuring cluster state consistency.

High Availability & Load Management

- Multi-replica synchronization: Ensures data redundancy through Quorum Queue and Stream Queue multi-replica mechanisms, preventing single points of failure.
- Failure recovery and replica migration: Automatic failover upon node failure, supporting replica migration within the cluster for load balancing.
- Load balancing: Distributes production and consumption requests evenly across cluster nodes, improving overall throughput.
- Status monitoring: Real-time monitoring of node, queue, and connection health status, providing a basis for failure recovery.

3.1.3 Distributed Coordination

The Distributed Coordination module (based on Kingdee Apusic Distributed Configuration Center) is responsible for maintaining cluster consistency and state:

- Cluster membership management: Maintains the online/offline status of all Broker nodes in the cluster.
- Metadata persistent storage: Stores and synchronizes core metadata such as Vhost, Exchange, Queue, and Binding.
- Leader election: Elects a Leader node in the cluster, responsible for queue management and failure handling.
- Queue Leader election: When a queue Leader node fails, automatically elects a new Leader from Followers.
- Cluster status monitoring and notification: When cluster status changes (e.g., nodes going online/offline, Leader changes), sends notifications to relevant components to trigger corresponding actions.

4 Product Advantages

- Zero-modification Xinchuang replacement: Directly replaces existing RabbitMQ clusters without modifying business code or adjusting client configuration (only the connection address needs to be changed), with near-zero migration cost.
- Full ecosystem compatibility: Native RabbitMQ tools, monitoring systems (Prometheus+Grafana), data synchronization tools, etc., can all be directly reused without additional adaptation development.
- Xinchuang environment adaptation: Fully compatible with the Xinchuang software and hardware ecosystem, meeting the requirements of government and enterprise Xinchuang projects, providing end-to-end Xinchuang solutions.
- Authentic feature experience: Fully follows RabbitMQ core feature specifications, supporting all business scenarios familiar to users (such as flexible routing, asynchronous processing, peak load leveling, etc.), with business-unaware switching.
- Enhanced security and compliance: Retains RabbitMQ's original security capabilities while adding new security features, meeting Xinchuang project security compliance standards.

5 Application Scenarios

- Xinchuang project replacement scenario: In government and enterprise Xinchuang transformations, directly replaces existing RabbitMQ clusters without refactoring business systems, quickly meeting Xinchuang requirements.
- Asynchronous communication decoupling scenario: Based on RabbitMQ's native publish-subscribe model, enables asynchronous communication between distributed systems, reducing system coupling.
- Peak load leveling scenario: Uses RabbitMQ's queue buffering capability to smooth burst traffic and protect backend service stability.
- Reliable message transmission scenario: Based on message acknowledgment mechanisms and persistent storage, ensures critical business messages are not lost.
- Flexible routing scenario: Leverages multiple exchange types such as Direct, Fanout, Topic, and Headers to meet complex message routing requirements.

UNIFIED SERVICE HOTLINE

4008-555-800

Kingdee Apusic Cloud Computing Co., Ltd. (referred to as “Kingdee Apusic”) was founded in 2000. Its predecessor was “Kingdee Middleware Company.” It is a basic software platform provider under Kingdee Group, a national standards-setting enterprise for cloud computing, and a core software enterprise in the national IT application innovation industry. Kingdee Apusic is an enterprise undertaking the National 863 Key R&D Program and the National Major Science and Technology Project for Core Electronic Components, High-end General-purpose Chips and Fundamental Software, and is also a basic platform provider for the national key projects of “Two Networks, One Website, Four Databases, and Twelve Golden Projects.” Its products are widely used in key industries such as government, military, finance, and energy, and it has cumulatively served more than 100,000 customers.



National standard setter for cloud computing
Basic software enterprise under Kingdee Group
Core enterprise for IT application innovation
Website: www.apusic.com

