



Product Introduction

Apusic Distributed Message Queue V2.0.6_for_mqtt

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 Overview
 - 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 for Apusic Distributed Message Queue for MQTT (Apusic Distributed Message Queue, referred to as: ADMQ for MQTT), helping users quickly understand the features, technical characteristics, and typical application scenarios of the ADMQ for MQTT 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

To learn more about ADMQ for MQTT, please refer to the following product documentation set:

No.	Document	Description
1	Apusic Distributed Message Queue for MQTT Quick Start Guide	Briefly introduces how to quickly get started with ADMQ for MQTT.
2	Apusic Distributed Message Queue for MQTT Installation Guide	Details how to install ADMQ for MQTT on various operating systems, as well as service start/stop operations.
3	Apusic Distributed Message Queue for MQTT Message Engine User Manual	Details the use, configuration, and management of ADMQ for MQTT message engine features and supporting tools.
4	Apusic Distributed Message Queue for MQTT Console User Manual	Details the use and operation of the ADMQ for MQTT console.
5	Apusic Distributed Message Queue for MQTT Development Guide	Details how to develop ADMQ for MQTT client applications in various development languages.
6	Apusic Distributed Message Queue for MQTT Migration Guide	Details the migration from MQTT Broker to ADMQ for MQTT.
7	Apusic Distributed Message Queue for MQTT Operations Manual	Details monitoring, operations, and security hardening for ADMQ for MQTT.
8	Apusic Distributed Message Queue for MQTT Performance Optimization Guide	Details performance tuning for ADMQ for MQTT.

1.3 Technical Support

ADMQ for MQTT 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 seeking technical support, please provide the following information:

1. Your name
2. Company information and contact details
3. Operating system and its version
4. Product version number
5. Detailed information such as logs and screenshots of the exception or error

2 Product Overview

2.1 Product Positioning

Apusic Distributed Message Queue for MQTT (Apusic Distributed Message Queue, referred to as: ADMQ for MQTT) is an enterprise-grade MQTT messaging middleware designed for IoT and mobile internet scenarios. It fully complies with the core MQTT protocol specification, achieving full compatibility with protocols, clients, ecosystem tools, and business scenarios. It can replace existing MQTT Broker clusters without modification, meeting the requirements for domestic 信息化 innovation (信创) projects.

ADMQ for MQTT maintains native MQTT protocol compatibility while providing enterprise-grade capabilities such as security authentication, permission control, visual management, rule engine, data bridge, and operations monitoring.

2.2 Core Value

Adhering to the lightweight, efficient, and reliable core characteristics of MQTT, while adapting to domestic 信息化 innovation hardware and software environments, it provides massive device access capabilities, more stable operation guarantees, more convenient operations support, and more comprehensive security capabilities, making it the preferred solution for domestic 信息化 innovation replacement of IoT messaging middleware.

2.3 Core Capabilities

Core Capability	Description
Full MQTT Protocol Compatibility	Fully compatible with MQTT 3.1, 3.1.1, and 5.0 protocols, behavior consistent with native MQTT Broker.
Full Version Client Adaptation	Supports mainstream MQTT clients such as Eclipse Paho and MQTT.js, with no code modification required for integration.
Seamless Ecosystem Tool Reuse	Compatible with MQTT native tools, monitoring systems (Prometheus+Grafana), and other ecosystem components.
Deep Adaptation to Domestic 信息化创新 Environment	Adapts to mainstream domestic 信息化 innovation operating systems (Kylin, UnionTech, etc.), domestic chips (ARM, RISC-V, etc.), and is compatible with domestic databases and middleware ecosystems.
MQTT Core Feature Inheritance	Supports publish/subscribe model, QoS 0/1/2, retained messages, will messages, session persistence, topic wildcards, and other native core features.

Enterprise-Grade Enhancement

Supports rule engine, data bridge, authentication and authorization, ACL access control, TLS/National Secret SSL encryption, and other enterprise-grade capabilities.

3 Product Architecture

3.1 Architecture Overview

ADMQ for MQTT adopts a distributed cluster architecture, fully complying with the MQTT protocol and publish/subscribe 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 status monitoring of MQTT cluster nodes.
- User Permissions: Role-based access control (RBAC) to ensure operational security.
- Client Management: View online clients, subscription relationships, and inflight messages.
- Topic Management: Manage topics, topic rewriting, auto subscriptions, and topic monitoring.
- Authentication and Authorization: Manage authenticators, authorization sources, and ACL permission rules.
- Message Management: Send test messages and manage retained messages.
- Monitoring and Alerts: View node status, runtime monitoring, slow subscriptions, and alert information.

3.1.2 Core Engine

The Core Engine is the core of message processing, compatible with the MQTT protocol, ensuring reliable message transmission and distribution:

MQTT Protocol Access & Security Access Control

- Provides a standard MQTT protocol compatibility layer, supporting TCP, SSL, WebSocket, WebSocket over SSL, QUIC, and other access methods.
- Integrates authentication (username/password, JWT, SCRAM), authorization (ACL), and transport encryption (SSL/TLS, National Secret SSL) to ensure access security.

Message Processing Service

- Message Publish Reception: Receives messages from publishers, performs validation and topic matching.
- Message Subscription Service: Pushes messages to subscribers, supporting QoS 0/1/2 message quality levels.

- Retained Message Service: Retains the last retained message for a topic, immediately delivered to new subscribers.
- Will Message Service: Automatically publishes will messages when a client disconnects abnormally.
- Session Persistence Service: Saves subscription relationships, unacknowledged messages, and offline messages, recoverable upon client reconnection.

High Availability and Load Management

- Stateless Node Cluster: Stateless sharing between nodes, supports failover for any node.
- Automatic Failure Recovery: Automatically removes abnormal nodes upon failure, clients can reconnect to other nodes.
- Load Balancing: Evenly distributes client connections across cluster nodes to improve overall access capacity.
- Status Monitoring: Real-time monitoring of node, connection, and topic health status, providing basis for failure recovery.

3.1.3 Distributed Coordination

The Distributed Coordination module is responsible for maintaining cluster consistency and status:

- Cluster Member Management: Maintains online/offline status of all MQTT nodes in the cluster.
- Session State Synchronization: Synchronizes client session state across cluster nodes.
- Route Table Management: Maintains routing relationships from topics to subscribers.
- Cluster State Monitoring Notification: When cluster state changes (such as node online/offline, subscription relationship changes), sends notifications to relevant components to trigger corresponding processing.

4 Product Advantages

- Zero-Modification Domestic信息化创新 Replacement: No need to modify business code or adjust client configuration (only need to change the connection address) to directly replace the existing MQTT Broker, with migration costs approaching zero.
- Full Ecosystem Compatibility: MQTT native tools, monitoring systems (Prometheus+Grafana), etc., can be directly reused without additional adaptation development.
- Massive Device Access: Supports millions of concurrent connections, meeting device access requirements for large-scale IoT scenarios.
- Domestic信息化创新 Environment Adaptation: Fully compatible with domestic信息化 innovation software and hardware ecosystems, meeting the landing requirements of government and enterprise信息化 innovation projects, providing end-to-end信息化 innovation solutions.
- Enhanced Security Compliance: On the basis of retaining MQTT's original security capabilities, adds ACL, TLS/National Secret SSL, audit logs, and other security features, meeting信息化 innovation project security compliance standards.
- Flexible Extensibility: Supports rule engine and data bridge, can integrate with databases, message queues, stream computing engines, and other systems.

5 Application Scenarios

- Domestic信息化 Innovation Project Replacement: In government and enterprise信息化 innovation transformation, directly replace the original MQTT Broker without refactoring business systems, quickly meeting信息化 innovation landing requirements.
- IoT Platform Scenario: Supports massive sensor and smart device access, enabling device data collection, command delivery, and status monitoring.
- Vehicle Networking Scenario: Supports vehicle terminal access, enabling vehicle status reporting, remote control, OTA upgrades, etc.
- Mobile Message Push Scenario: Supports mobile app message push, instant messaging, real-time notifications, etc.
- Industrial Internet Scenario: Supports industrial device data collection, predictive maintenance, production monitoring, etc.
- Smart Home Scenario: Supports smart home device联动, voice control, scene automation, etc.

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

