

信璞科技消息中间件  
XinPuMQ V1.0  
使用手册

南京信璞科技服务有限公司

## 目录

版本变更说明 .....	3
前言 .....	4
术语说明 .....	5
第 1 章 产品介绍 .....	6
第 2 章 基本概念 .....	7
第 3 章 安装准备 .....	9
第 4 章 系统安装配置及启动 .....	10

# 版本变更说明

本手册的更新是累积的。因此，最新的手册版本包含对以前版本所做的所有更改。

本手册版本 V1.0 适用于 XinPuMQ V1.0 版本。

手册版本	更新内容
V1.0	统一使用 XinPuMQ 用户使用手册模板。

# 前言

本文档是 XinPuMQ V1.0 产品的用户使用手册之一，提供客户快速安装、基本概念及服务端通用配置 的说明，并提供了发布及订阅的应用场景示例。

## 适合的对象

本手册主要适用于生产环境中的系统管理员，部分内容同样适用于应用开发人员和应用部署人员。

本手册假定您已经具备如下技能

Linux 系统基本操作。

C 或 Java 开发能力。

## 技术支持

南京信璞科技服务有限公司将为您提供全方位的技术支持，您可以通过以下方式获得技术支持：

Support Tel: 15366101014

邮箱: [chenzekai@njxinpu.com.cn](mailto:chenzekai@njxinpu.com.cn)

您在取得技术支持时，请提供如下信息：

1. 您的姓名
2. 您的公司信息
3. 您的联系方式
4. 操作系统及其版本
5. 产品版本号
6. 日志等错误的详细信息

## 术语说明

名称	说明
XinPuMQ (简称 XPMQ)	信璞科技消息中间件
Namesvr	服务端管理节点
Broker	服务端工作节点
Client	客户端
Domain	通信域
Topic	主题
Message	消息
Producer	生产者, 即消息的提供者
Consumer	消费者, 即消息的消费者
Pub-Sub	发布及订阅

# 第 1 章 产品介绍

## 1.1 产品概述

信璞科技消息中间件 XinPuMQ(以下简称 XinPuMQ)是一款专门为现代分布式应用程序设计的高性能、可靠的消息传递系统。系统采用轻量级的消息队列模型和高效的传递机制,能够支持应用程序的解耦、弹性伸缩、流量削峰、异步通信和数据同步,从而确保数据的安全可靠传输。

XinPuMQ 具有以下特点:

### 1. 支持多语言和平台

开发人员可以利用 C 语言、Java 语言等多种语言接口快速集成 XinPuMQ,而其底层基于 C++ 语言开发,确保了 100% 的自主可控性。全面兼容国产化生态,包括各类国产芯片和操作系统。作为促进企业数字化转型和云原生应用程序开发的优选方案,XinPuMQ 为各个领域提供了技术领先、稳定可靠的数据交互解决方案。

### 2. 性能优越,高可用

XinPuMQ 不仅支持单节点超过 100 万 TPS(每秒事务处理量)的卓越性能,而且通过对等集群技术和交叉保护策略,提供了高可用性和无限扩展能力。

### 3. 先进队列消息及路由机制

高效发挥软硬件资源性能;支持集群,并提供多种负载均衡算法,包括线路备份、动态和静态负载、基于内容的路由、客户端连接池集群等

### 4. 安全可靠传输

通过使用 XinPuMQ,应用系统完全不需要担心消息传递过程中可能遇到的各种障碍(机器故障、网络故障等)和异常。

### 5. 安装简单

通过安装脚本,可实现一键快速安装,避免繁琐的安装过程,降低了安装门槛。

无论是在性能、可靠性、可扩展性还是兼容性方面,XinPuMQ 消息中间件都展现出了卓越的表现。通过优化设计, MQ 实现了快速消息处理和低资源占用,是微服务架构中理想的消息传递解决方案。它可以帮助企业构建高效、灵活、安全的分布式系统,满足现代应用程序对数据传输的各种需求。

## 1.2 特别说明

- 客户端 C 接口提供动态库支持, JAVA 接口提供 jar 包支持。
- XinPuMQ 的 JAVA 客户端由纯 JAVA 语言编制而成,可以直接跟服务端通信。
- 使用 JAVA 客户端要求 JDK 8 及以上版本。

## 第 2 章 基本概念

本章列举了消息中间件产品及客户端功能涉及的基本概念。

### 2.1 管理节点 (Nameserver)

管理节点提供服务注册、服务发现、对集群中各工作节点的运行状态和负载情况的监控、集中存储集群 状态信息、统一管理集群工作节点配置、服务负载均衡以及连接安全、日志和本地进程监控(以下简称 Nameserver)。管理节点的功能设计包括监控管理子系统、安全保障子系统、数据存储子系统、名字服务管理子系统、 集群调度子系统、远程管理子系统。

### 2.2 工作节点 (Broker)

工作节点提供消息传输和转发、消息存储、消息消费、消息事件生成、发送队列管理、节点运行信息上报管理节点以及动态配置更新、日志和本地进程监控(以下简称 Broker)。工作节点的功能设计包括数据传输子系统、数据存储子系统、消息处理子系统、监控管理子系统。

### 2.3 客户端节点 (Client)

XinPuMQ 客户端提供对多种开发工具和开发环境的支持，提供 C、Java 方式的接口，方便应用程序的开发。应用程序可以通过调用这些接口实现消息的传递、系统管理等功能，应用程序只需要关心业务处理逻辑，而不必关心复杂的底层网络传输。应用程序可以和 XinPuMQ 服务端工作节点、管理节点部署在同一台机器上，也可以部署在不同的机器上。

### 2.4 通信域 (Domain)

XinPuMQ 客户端采用通信域+主题模式来唯一确定某一业务标识。在同一个通信域下，主题是唯一的。在不同通信域中，主题可能会有重名情况。通信域与主题的具体设置是由客户自己规划与设置(以下简称 Domain)。

### 2.5 主题 (Topic)

XinPuMQ 客户端是采用围绕主题模式来进行业务区分和开展通讯的，各个通信模块可以配置相应的主题进行发布，订阅者可以根据主题进行订阅。主题命名：字符和数字的组合来创建主题标识符命名。

## 2.6 消息 (Message)

“消息”是在应用进程之间或一个应用的不同部分之间交换的数据单位，应用可分布在多台相同或不同的平台上，也可分布在一台计算机上。消息可非常简单，例如只包含文本字符串；也可更复杂，包含嵌入对象或者一个文件。

XinPuMQ 的消息分为两部分：

- 应用数据：应用数据的内容和结构由用户程序来定义。
- 消息描述：消息描述给出消息的特定属性，包含消息类型、长度、优先级等。

## 2.7 生产者 (Producer)

消息的提供者。生产者通过指定管理节点 IP 地址和端口服务、客户端标识、业务域属性、主题这些要素，组织要传递的消息内容，把该消息发送到生产者要传送的目的地。

## 2.8 消费者 (Consumer)

消息的消费者。消费者通过指定管理节点 IP 地址和端口服务、客户端标识、业务域属性、主题这些要素进行消息消费。

## 2.8 发布订阅 (Pub-Sub)

发布订阅是一种消息范式，消息的发送者（称为发布者）不会将消息直接发送给特定的接收者（称为订阅者）。而是将发布的消息分为不同的类别，无需了解哪些订阅者可能存在。同样的，订阅者可以表达对一个或多个类别的兴趣，只接收感兴趣的新闻，无需了解哪些发布者存在。

## 第 3 章 安装准备

### 3.1 运行环境

XinPuMQ 可运行在 Linux,Unix,Window 环境下，系统要求满足以下条件：

表 3-1 产品运行环境

系统组件	系统要求
硬件	Linux,Unix,Window 操作系统所支持的硬件平台
操作系统	Linux 内核版本不低于 3.10 Window 版本不低于 Windows7
网络协议	L3: IPv4 ; L4: TCP、UDP
内存	建议内存 8G 或以上
网卡	100M 或 1000M 以上
硬盘	支持机械盘，推荐使用 SSD 固态硬盘，大小不低于 128G
Java 客户端	JDK8 及以上

## 第 4 章 系统安装配置及启动

### 4.1 客户端安装指南

#### 4.1.1 安装步骤

1. 配置好系统环境变量 WORK\_ROOT
2. 在 Java 端使用代码初始化 MQ 客户端
3. 设置好消息发布代码和消息回调处理代码
4. 启动 MQ 客户端

#### 4.1.2 环境配置

在 Java 应用中初始化 MQ 客户端之前，请确保设置了工作根目录环境变量 WORK\_ROOT。可以通过在用户的 .profile 文件中添加如下行实现：`export WORK_ROOT=$HOME/work` 确认环境变量指向系统的整合安装目录。

#### 4.1.3 初始化过程

在 Java 代码中使用以下原生方法初始化 MQ 客户端：

```
public native int init(String workRoot, int clientId, int[] topicIds);
```

- workRoot: 工作根目录，通过上一步配置，`System.getenv("WORK_ROOT")` 获取。
- clientId: 分配给客户端的唯一标识符，保证其在客户端生命周期内的唯一性。
- topicIds: 要订阅的主题 ID 数组，例如 `int[] topicIds = {5, 55};`。根据需求选择合适的主题 ID。

#### 4.1.4 消息发布

负载均衡：通过 `publishTopicPollMessage` 方法发布消息，系统将自动根据负载均衡选择订阅者。

广播：使用 `publishTopicBroadcastMessage` 方法发布消息至所有订阅者。

定向消息：利用 `publishAddressedMessage` 方法直接发送消息给指定订阅者。

## 4.1.5 回调处理

回调接口有以下接口，供用户选择：

onReady: MQ 准备就绪，可投递消息时触发。

onNotReady: 所有 MQ 服务器连接断开时触发。

onExit: 客户端退出时触发，如响应 Ctrl+C。

onPublishAddressedMessage: 消息投递确认时触发。

## 4.1.6 设计建议

优先实现请求/响应模式，以确认消息成功投递至接收方。

## 4.2 服务端安装指南

### 4.2.1 安装步骤

1. 在待安装的设备上运行提供的 `collect_matchine_info` 程序，将运行后显示的信息发送给客服获取 `license` 文件用于激活。
2. 将 `license` 文件放到指定文件位置后，运行客服提供的安装脚本。安装完毕后即可直接运行。

### 4.2.2 设备激活

用户将我们提供的 `collect_matchine_info` 程序，放置到待安装的设备上运行。运行完毕后屏幕会显示相应的信息。此时联系我们的客服人员，将显示的信息提供给我们客服人员。客服人员会进行激活操作，激活成功后客服人员会提供一份唯一的 `license` 文件用于此设备的激活。

### 4.2.3 脚本安装

将 客服提供的 `license` 文件放到指定文件位置后,运行客服提供的安装脚本。

如果没有正确放置 `license` 文件,或者安装脚本的权限不足,很可能会导致软件无法正常安装和运行。如若出现无法正常安装的情况,请及时联系客服人员。

安装完成后,用户就可以立即运行该软件,开始正常使用了。