

東方通 | TongTech®

400-650-7088



东方通消息中间件 TongLINK/Q V9.0 快速使用手册

版本 V9-1

北京东方通科技股份有限公司



目录

版权声明	5
版本变更说明	6
前言	7
术语说明	8
第1章 产品介绍	9
1.1 概述	9
1.2 整体架构.....	9
1.3 特别说明.....	10
第2章 基本概念	11
2.1 管理节点 (Nameserver)	11
2.2 工作节点 (Broker)	11
2.3 客户端节点 (Client)	11
2.4 通信域 (Domain)	11
2.5 主题 (Topic)	11
2.6 消息 (Message)	12
2.7 生产者 (Producer)	12
2.8 消费者 (Consumer)	12
2.9 发布订阅 (Pub-Sub)	12
第3章 安装准备	13
3.1 运行环境.....	13
3.2 安装包.....	13
第4章 系统安装配置及启动	14
4.1 管理节点安装配置及启动.....	14
4.1.1 安装步骤	14
4.1.2 目录及文件	14
4.1.3 配置环境变量	15
4.1.4 修改主配置文件	15
4.1.5 启停节点进程	16
4.1.6 查看日志	17
4.2 工作节点安装配置及启动.....	18

4.2.1	安装步骤	18
4.2.2	目录及文件	18
4.2.3	配置环境变量	19
4.2.4	修改主配置文件	19
4.2.5	启停节点进程	20
4.2.6	查看日志	20
4.3	客户端安装配置.....	23
4.3.1	安装步骤	23
4.3.2	目录及文件	23
4.3.3	配置环境变量	24
4.3.4	修改配置文件	24
4.3.5	查看日志	24
第5章	应用场景示例	25
5.1	发布与订阅流程.....	25
5.2	服务端配置及启动.....	25
5.3	客户端安装及演示.....	26
5.3.1	客户端安装	26
5.3.2	demo_c编译及演示	26
5.3.3	demo_java编译及演示	29

图片目录

图 1.2-1: TLQ整体架构	9
图 5.1-1: 发布与订阅流程	25
图 5.2-1: 发布与订阅拓扑图	26

表格目录

表 3.1-1: 产品运行环境	13
表 3.2-1: 产品安装包	13
表 4.1-1: 管理节点目录及文件	14
表 4.2-1: 工作节点目录及文件	18
表 4.2-2: 工作节点日志目录	21
表 4.3-1: 客户端目录及文件	23

版权声明

版权

©2006–2021东方通科技版权所有。

版权保护

未经东方通科技公司书面许可，本文档不得整体或部分地复印、复制、翻译或缩减成任何电子介质或计算机可以阅读的格式。本文档中的信息可能不加通知进行修改。

商标

©2006–2021东方通科技版权所有。**东方通**、*TongTech*、*TongLINK*、*TongLINK/Q*、*TongIntegrator*、*TongSEC*、*TongWeb*、*TongEASY*、*TongWorkflow*是东方通科技的注册商标。其他所有的公司和产品名称可能为第三方所有。

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V9-1适用于TongLINK/Q V9.0.0.0版本。

手册版本	更新内容
V9-1	1. 统一使用TongLINK/Q用户使用手册模板。

前言

本文档是TongLINK/Q V9.0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：

1. Linux 系统基本操作；
2. C 或 Java 开发。

用户使用手册集

TongLINK/Q V9.0为您提供的用户使用手册集包含的文档有：

1. 000_TongLINKQ9.0用户手册导读：提供手册集的目录及手册版本说明。
2. 001_TongLINKQ9.0快速使用手册：提供各版本客户快速安装，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。
3. 002_TongLINKQ9.0系统安装及配置手册：提供产品安装及系统配置步骤和参数的详细说明。
4. 003_TongLINKQ9.0系统功能及应用指南：介绍客户端基本概念及系统所具备的其他功能，并提供C接口、Java接口示例程序的详细说明。
5. 004_TongLINKQ9.0管理控制台安装手册：提供管理控制台的安装、启动及登录说明。
6. 005_TongLINKQ9.0管理控制台使用手册：提供管理控制台资源管理、业务配置、告警管理、系统配置等功能使用说明。
7. 006_TongLINKQ9.0C客户端编程手册：介绍C接口应用编写中用到的相关宏、数据结构及函数定义等。
8. 007_TongLINKQ9.0Java客户端编程手册：对JAVA编程中用到的BaseClass相关的类和使用，以及BaseClass编程等进行说明。
9. 008_TongLINKQ9.0RESTful服务API编程手册：介绍对外提供的API接口列表以及接口调用方式。

技术支持

东方通产品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：

网址：www.tongtech.com

Support Tel：400-650-7088

邮箱：pqmo@tongtech.com

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
TongLINK/Q (简称TLQ)	东方通消息中间件
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 概述

TongLINK/Q V9.0是面向分布式应用的消息中间件产品，主要功能是在应用程序之间进行实时、高效和可靠地传递消息，使得消息可以在不同的网络协议、不同的计算机系统和不同的应用软件之间进行网络传输。系统支持IPV4协议，TCP协议以及UDP协议。

TongLINK/Q应用程序可灵活地运行在多平台的多节点上。当应用程序之间要传送消息时，应用程序只需将消息目的地、消息的属性和内容以及消息的控制信息通过API接口传递给TongLINK/Q，TongLINK/Q会根据应用提供的信息对消息进行处理，并且利用TongLINK/Q节点组成的虚拟网，将消息传送到消息接收者所在的节点上，最后提交给消息的接收者。

TongLINK/Q V9.0具有如下优点：

1. 提供亿级消息堆积能力
2. 提供低时延的接入传输服务
3. 提供百万级业务并发传输能力
4. 保证数据传输的高可靠性
5. 多种安全防护策略保证业务数据的安全性
6. 支持灵活多变的网络部署模式，方便用户在不同的业务场景下使用

1.2 整体架构

TongLINK/Q整体架构由客户端（client）、服务端、管理控制台（web）三部分组成，服务端又分为管理节点（namesvr）和工作节点（broker）。产品整体架构如下图所示：

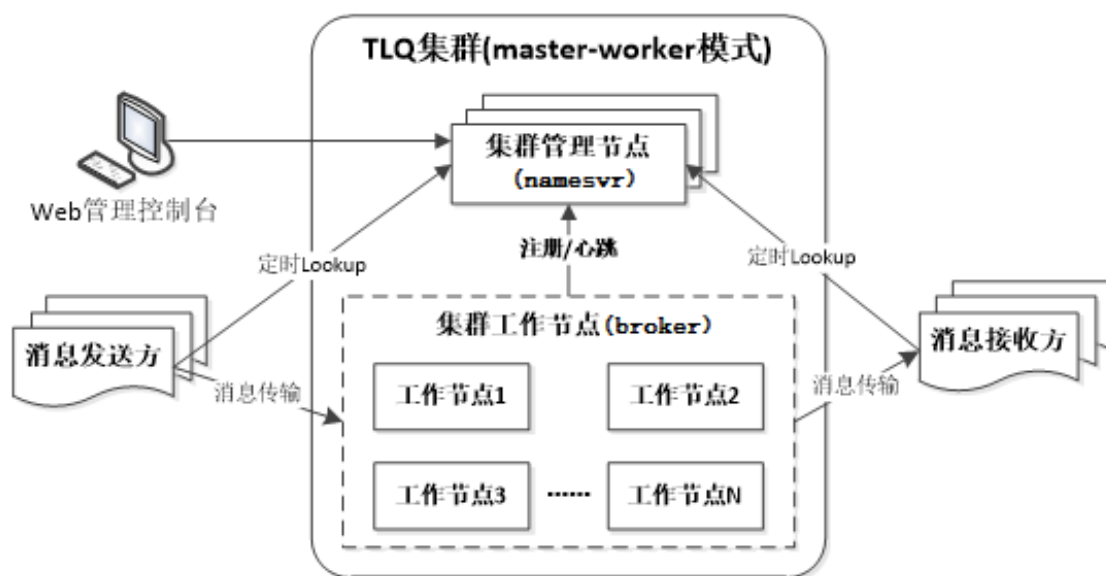


图 1.2-1: TLQ整体架构

1.3 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。
- TongLINK/Q 的 JAVA 客户端由纯 JAVA 语言编制而成，可以直接跟服务端通信，暂不支持 JMS 规范。
- TongLink/Q 目前只支持本地存储，工作节点接收到消息存放在本地文件系统，如使用网盘设备进行消息的持久保存，可靠性相对较低，目前暂无计划支持。
- 查询设置 TongLink/Q 系统功能当前支持通过 Web 管理控制台进行，暂不支持命令行方式使用。
- 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。更多的功能及使用说明可参考《003_TongLINKQ9.0系统功能及应用指南》。

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)

TongLINK/Q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TongLINK/Q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)

TongLINK/Q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)

TongLINK/Q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。

TongLINK/Q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。

2.9 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，只接收感兴趣的消息，无需了解哪些发布者存在。

第3章 安装准备

3.1 运行环境

TongLINK/Q需运行在Linux系统环境下，系统要求满足以下条件：

表 3.1-1：产品运行环境

系统组件	系统要求
硬件	Linux 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
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
磁盘	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

3.2 安装包

TongLINK/Q产品为不同的架构平台提供不同的安装包，安装之前请查看机器属于哪种架构。如果机器是aarch64架构服务器，用TongLINK/Q的aarch64架构安装包；x86_64架构服务器则用TongLINK/Q的x86_64架构安装包。产品包分别由namesvr、broker和client三部分组成，用户需要在主机上部署以下安装包。

表 3.2-1：产品安装包

产品包名（xxx代表不同的架构平台）	对应程序名	解压后目录名
Install_TLQMgr_Standard_Linux4.19.90_xxx_9.0.0.0.tar.gz	管理节点 (namesvr)	TLQ9_Mgr
Install_TLQBok_Standard_Linux4.19.90_xxx_9.0.0.0.tar.gz	工作节点 (broker)	TLQ9_Bok
Install_TLQCli_Standard_Linux4.19.90_xxx_9.0.0.0.tar.gz	客户端 (client)	TLQ9_Cli

第4章 系统安装配置及启动

TongLINK/Q的管理节点、工作节点及客户端可单台部署也可分布式部署，网络中可存在一个或多个集群配置，使用灵活且方便。

4.1 管理节点安装配置及启动

4.1.1 安装步骤

1. 使用命令**uname -m**查看系统架构，确认为aarch64或是x86_64架构，例如：

```
$ uname -m
aarch64
```

或

```
$ uname -m
x86_64
```

2. 把对应架构的产品安装包上传到新建用户或已存在的用户目录下, 例如: **/home/xxx**
3. 解压TongLINK/Q管理节点安装包: (xxx代表不同的架构平台)
tar xvf Install_TLQMgr_Standard_Linux4.19.90_xxx_9.0.0.0.tar.gz
4. 购买的TLQ产品中提供有license文件。将**license.dat**文件复制到管理节点的**TLQ9_Mgr/config**目录下。

4.1.2 目录及文件

解压安装完成后，管理节点（namesvr）会生成如下目录及文件。

表 4.1-1：管理节点目录及文件

目录	目录下文件及说明
TLQ9_Mgr/bin	tlq_namesvr: 管理节点服务程序
	tlq_procd: 管理节点进程程序
TLQ9_Mgr/config	nameServer.xml: 管理节点主配置文件
	zone.xml: 集群通信域配置文件
	topic.xml: 集群主题配置文件
	queue.xml: 集群队列配置文件
	*.pem *.key: RESTful接入认证相关密钥文件
	key_pub.pem license.dat: license认证所需文件 注意：安装时需要用户手动复制license.dat到该目录下
TLQ9_Mgr/logs	日志文件目录
TLQ9_Mgr/scripts	namesvr_setenv: 管理节点环境变量设置脚本

TLQ9_Mgr/sample TLQ9_Mgr/sample/scripts TLQ9_Mgr/sample/scripts/config	restful接口测试脚本及相关测试使用到的配置文件
--	----------------------------

4.1.3 配置环境变量

1. 执行以下命令对环境变量进行自动配置。

```
cd TLQ9_Mgr/scripts
```

```
source namesvr_setenv
```

2. 检测以下环境变量是否设置成功。

```
echo $NAMESERVER_CONFIG_FILE
```

```
echo $TOPIC_CONFIG_FILE
```

```
echo $TLQ9HOMEDIR
```

验证结果如下所示:

```
[xxx@tstsvr scripts]$ source namesvr_setenv
[xxx@tstsvr scripts]$ echo $NAMESERVER_CONFIG_FILE
/home/xxx/TLQ9_Mgr/scripts/./config/nameServer.xml
[xxx@tstsvr scripts]$ echo $TOPIC_CONFIG_FILE
/home/xxx/TLQ9_Mgr/scripts/./config/topic.xml
[xxx@tstsvr scripts]$ echo $TLQ9HOMEDIR
/home/xxx/TLQ9_Mgr/scripts/..
```

4.1.4 修改主配置文件

管理节点配置文件在**TLQ9_Mgr/config**目录下，分别为以下4个配置文件：

- **nameserver.xml**：主配置文件
- **zone.xml**：通信域配置文件
- **topic.xml**：主题配置文件
- **queue.xml**：队列配置文件

以上配置文件中具体的参数属性说明可参考《002_TongLINKQ9.0系统安装及配置手册》。

完成环境变量配置后，按以下步骤对管理节点的主配置文件进行修改。

1. 进入配置文件目录 **TLQ9_Mgr/config**

```
cd TLQ9_Mgr/config
```

2. 修改主配置文件 **nameServer.xml**

```
vi nameServer.xml
```

必须配置项有以下4处：（可修改不可删除）

- 1) <ListenPort>: 服务监听端口（默认为 9888）
- 2) <IpAddress>: 当前网络接口获取的 IP 地址（**必须修改**，默认为 10.10.20.30）
- 3) <LogPath>: log 日志存储目录的绝对路径（默认为 “.” 即进程启动所在目录，建议修改为 “/home/xxx/TLQ9_Mgr/logs”）
- 4) <HttpPort>: Restful 服务监听端口（默认为 52011）

注意：

ListenPort 及 HttpPort 建议使用 1024 及以后的端口，避免跟系统常用端口号冲突。

3. 保存主配置文件并退出。

4.1.5 启停节点进程

进入 **TLQ9_Mgr/bin** 目录下，运行以下命令可对管理节点的进程进行查询、启停、清理等操作。

- 清理进程管理工具后台进程和它所管理的所有进程（确保旧有的程序不再运行）

```
tlq_procd -f
```

```
[xxx@tstsvr bin]$ tlq_procd -f
Success to flush the daemon!
```

说明：如果tlq_daemon后台管理进程不存在，会报[Flush error]: there is no daemon exist under current user!的错误，此时该错误可以忽略。

- 开启管理节点的进程

```
tlq_procd -p tlq_namesvr -s
```

```
[xxx@tstsvr bin]$ tlq_procd -p tlq_namesvr -s
...
Process has been started successfully!
```

- 关闭管理节点的进程

```
tlq_procd -p tlq_namesvr -t
```



```
[xxx@tstsvr bin]$ tlq_procd -p tlq_namesvr -t
Process has been stopped successfully!
```

- 重启管理节点的进程

tlq_procd -p tlq_namesvr -r

```
[xxx@tstsvr bin]$ tlq_procd -p tlq_namesvr -r
...
Process has been reloaded successfully!
```

- 查看管理节点的进程运行状态

tlq_procd -l

```
[xxx@tstsvr bin]$ tlq_procd -l
tlq_namesvr      <Running>
```

- 查看管理节点进程版本号

tlq_namesvr -V

```
[xxx@tstsvr bin]$ tlq_namesvr -V
Version: 9.0.0.0
```

- 查看管理节点进程号

ps -ef |grep tlq

```
[xxx@tstsvr bin]$ ps -ef |grep tlq
xxx   9971   17927   0 15:12 pts/2    00:00:00 grep tlq
xxx   32316         1   0 14:21 ?        00:00:00 tlq_daemon
xxx   32364   32316   0 14:21 ?        00:00:01 /home/xxx/TLQ9_Mgr/bin/tlq_namesvr
```

说明：tlq_daemon为守护进程，tlq_namesvr出现异常会主动拉起进程，保证服务运行的持续性。

4.1.6 查看日志

管理节点的进程启动后，在日志目录 **TLQ9_Mgr/logs** 下会生成以用户名命名的 **dbg** 和 **sys** 文件：dbg 存储的是调试日志信息，sys 文件存放系统信息的日志内容。运行如下命令查看生成的日志文件：

cd TLQ9_Mgr/logs

ls

```
[xxx@tstsvr TLQ9_Mgr]$ cd logs
[xxx@tstsvr logs]$ ls
tlq_namesvr_xxx.dbg.0  tlq_namesvr_xxx.sys.0
[xxx@tstsvr logs]$ more tlq_namesvr_xxx.sys.0
```

```
S 2021-08-13 10:18:45.602|4241:4241|TNMainBody.c|798|0:0|Server started!!!
S 2021-08-13 10:18:45.604|4241:4337|TNNetworkTcp.c|303|0:0|TCP service started!!!
S 2021-08-13 10:18:45.604|4241:4338|TNNetworkUdp.c|216|0:0|UDP service started!!!
```

4.2 工作节点安装配置及启动

4.2.1 安装步骤

1. 使用命令**uname -m**查看系统架构，确认为aarch64或是x86_64架构，例如：

```
$ uname -m
aarch64
```

或

```
$ uname -m
x86_64
```

2. 如果工作节点需要处理大并发连接，会占用一定数量的文件句柄。
可检查系统参数后，调整Linux最大文件打开数为65535（默认为1024），无需重启服务器。

注意：需使用有修改权限的帐号（root或者sudo已授权的帐号）进行修改

编辑/etc/security/limits.conf最后一行，添加如下内容：

```
* soft nofile 65535
* hard nofile 65535
```

3. 把对应架构的产品安装包上传到新建用户或已存在的用户目录下，例如：**/home/xxx**
4. 解压TongLINK/Q工作节点安装包：（xxx代表不同的架构平台）
tar xvf Install_TLQBok_Standard_Linux4.19.90_xxx_9.0.0.0.tar.gz

4.2.2 目录及文件

解压安装完成后，工作节点（broker）会生成如下目录及文件。

表 4.2-1：工作节点目录及文件

目录	目录下文件及说明
TLQ9_Bok/bin	tlq_broker_server: 工作节点服务程序
	tlq.sh: 工作节点服务启停控制脚本
TLQ9_Bok/config	broker_server.xml: 工作节点主配置文件
	broker_queue.xml: 队列配置文件
	tlq9_broker_error.xml: 错误码文件
TLQ9_Bok/	setp: 工作节点环境变量设置脚本

4.2.3 配置环境变量

进入 **TLQ9_Bok/**目录下，运行以下命令对工作节点的环境变量进行自动配置。

source setp

或

./setp (两个点中间有空格)

4.2.4 修改主配置文件

工作节点配置文件在**TLQ9_Bok/config**目录下，分别为以下2个配置文件：

- **broker_server.xml**：主配置文件
- **broker_queue.xml**：队列配置文件

以上配置文件中具体的参数属性说明可参考《002_TongLINKQ9.0系统安装及配置手册》。

按以下步骤可对工作节点的主配置文件进行修改。

1. 进入工作节点配置文件目录 **TLQ9_Bok/config**
cd TLQ9_Bok/config

2. 修改该目录下的配置文件 **broker_server.xml**

vi broker_server.xml

必须配置项有以下8处：（可修改不可删除）

- 1) **<system>**小节下的**<broker_name>**：工作节点名称（默认即可）
- 2) **<system>**小节下的**<brokerid>**：工作节点编号（默认即可）
- 3) **<system>**小节下的**<homedir>**：工作节点工作根目录及日志存放目录（默认为./）
- 4) **<system>**小节下的**<log_level>**：日志等级（默认为 ERROR，设置为 DEBUG 时，日志输出到 dbg 文件中）
- 5) **<listen>**小节下的**<listen_port>**：服务的监听端口（默认为 9876）
- 6) **<register>**小节下的**<nameserver_url_list>**：注册管理节点的 IP 和端口（**必须修改**，需和 TLQ9_Mgr/config/nameServer.xml 中的**<IpAddress>**及**<ListenPort>**相同）
- 7) **<register>**小节下的**<register_listen_ipv4>**：上报管理节点本地服务所用的监听地址（**必须修改**）
- 8) **<register>**小节下的**<register_listen_port>**：上报管理节点本地服务所用的监听端口（默认为 9876）

4.2.5 启停节点进程

进入 **TLQ9_Bok/bin** 目录下，运行以下命令可对工作节点的进程进行查询、启停等操作。

- 启动工作节点进程且后台运行

tlq.sh start

```
[xxx@tstsvr bin]$ tlq.sh start
Start /home/xxx/TLQ9_Bok/bin/tlq_broker_server
Using config [/home/xxx/TLQ9_Bok/config/broker_server.xml]
TongLink/Q Broker Server is starting ...

-----TongLink/Q Broker Server Start OK!!!-----
```

注意：在统信平台，需将**tlq.sh**中第一行“#!/bin/sh”修改为“/bin/bash”后，执行命令**./tlq.sh start**

- 停止工作节点进程

tlq.sh stop

```
[xxx@tstsvr bin]$ tlq.sh stop
Stop /home/xxx/TLQ9_Bok/bin/tlq_broker_server
Using config [/home/xxx/TLQ9_Bok/config/broker_server.xml]
TongLink/Q Broker Server is stopping ...
Tlq Broker Service Not Runing

-----TongLink/Q Broker Server Stop!!!-----
```

- 查看工作节点进程号

ps -ef |grep broker

```
[xxx@tstsvr bin]$ ps -ef |grep broker
xxx 20030      1   3 15:07 ?           00:00:00 /home/xxx/TLQ9_Bok/bin/tlq_broker_server -c
/xxx/TLQ9_Bok/config/broker_server.xml
xxx 24240 17927   0 15:07 pts/2    00:00:00 grep broker
```

4.2.6 查看日志

- 进入配置文件 **TLQ9_Bok/config/broker_server.xml** 中<homedir>设置的 log 日志存放目录
- 工作节点进程启动后，日志目录下（默认为 TLQ9_Bok/）会新产生以下 5 个文件夹：

表 4.2-2：工作节点日志目录

文件目录名	描述信息
etc	存放运行时pid文件和pipe文件
msg	消息存储和主题队列目录
files	大文件存放目录
history	消费者历史记录存放目录
log	默认的日志存放路径，存放调试日志和系统日志

```
[xxx@tstsvr TLQ9_Bok]$ ls
```

```
bin    config  etc    files  history log    msg    setp
```

3. 与管理节点和客户端交互日志一般存放在 **log** 目录下，工作节点启动成功后，查看 **tlq9_broker.sys.0**:

```
[xxx@tstsvr log]$ ls
```

```
tlq9_broker.dbg.0  tlq9_broker.sys.0
```

```
[xxx@tstsvr log]$ more tlq9_broker.sys.0
```

```
I 2021-08-13 10:30:11.645|9250:9250|BNConfig.c|423|0:2|Create ./log success
```

```
S 2021-08-13 10:30:11.646|9251:9251|BNBrokerStart.c|205|0:2|
```

```
-----TongLink/Q Broker Server Start OK!!!-----
```

```
I 2021-08-13 10:30:12.080|9251:9371|BNStoreConsumequeue.c|224|0:0|pBatchData malloc 4198400 size success
```

tlq9_broker.dbg.0文件中的日志信息的主要格式如下：

- 信息类型标志包括：
 - I：表示一般信息
 - W：表示警告信息，不影响系统运行
 - E：表示出错信息，系统停止
 - D：调试信息
 - S：提示启动停止的输出
- 当前时间串包括：年、月、日、时、分、秒、微秒

- 打印该行信息的进程号
- 打印该行信息的源代码文件名
- 打印该行信息的源代码行数
- 工作节点错误号
- 操作系统错误号
- 信息描述

例如：

```
D 2021-05-10 15:40:57.294|977427:977443|BNMessageRegHeartbeatProcess.c|41|BnMessageRegHeartbeatReqSerialize:topicTotalmsg[42584], queueTotalmsg[53763]
D 2021-05-10 15:40:57.294|977427:977443|BNMessageRegHeartbeatProcess.c|51|B_BNPackRequestHeartbeat:[00 00 07 d3 00 00 00 1b 0a 03 18 d3 0f 10 16 1a 04 31 30 30 30 28 a8 83 90 cb 03 60 d8 cc 02 68 83 a4 03 ]
D 2021-05-10 15:41:27.295|977427:977443|BNRemotingRegisterService.c|200|Register service heartbeat timer
D 2021-05-10 15:41:27.295|977427:977443|BNRemotingRegisterService.c|207|Register session remoting ip:168.1.100.126 | port:2222
D 2021-05-10 15:41:27.295|977427:977443|BNMessageRegHeartbeatProcess.c|41|BnMessageRegHeartbeatReqSerialize:topicTotalmsg[42584], queueTotalmsg[53763]
D 2021-05-10 15:41:27.295|977427:977443|BNMessageRegHeartbeatProcess.c|51|B_BNPackRequestHeartbeat:[00 00 07 d3 00 00 00 1b 0a 03 18 d3 0f 10 16 1a 04 31 30 30 30 28 a8 83 90 cb 03 60 d8 cc 02 68 83 a4 03 ]
D 2021-05-10 15:41:57.295|977427:977443|BNRemotingRegisterService.c|200|Register service heartbeat timer
D 2021-05-10 15:41:57.295|977427:977443|BNRemotingRegisterService.c|207|Register session remoting ip:168.1.100.126 | port:2222
D 2021-05-10 15:41:57.295|977427:977443|BNMessageRegHeartbeatProcess.c|41|BnMessageRegHeartbeatReqSerialize:topicTotalmsg[42584], queueTotalmsg[53763]
D 2021-05-10 15:41:57.295|977427:977443|BNMessageRegHeartbeatProcess.c|51|B_BNPackRequestHeartbeat:[00 00 07 d3 00 00 00 1b 0a 03 18 d3 0f 10 16 1a 04 31 30 30 30 28 a8 83 90 cb 03 60 d8 cc 02 68 83 a4 03 ]
D 2021-05-10 15:42:27.296|977427:977443|BNRemotingRegisterService.c|200|Register service heartbeat timer
D 2021-05-10 15:42:27.296|977427:977443|BNRemotingRegisterService.c|207|Register session remoting ip:168.1.100.126 | port:2222
D 2021-05-10 15:42:27.296|977427:977443|BNMessageRegHeartbeatProcess.c|41|BnMessageRegHeartbeatReqSerialize:topicTotalmsg[42584], queueTotalmsg[53763]
D 2021-05-10 15:42:27.296|977427:977443|BNMessageRegHeartbeatProcess.c|51|B_BNPackRequestHeartbeat:[00 00 07 d3 00 00 00 1b 0a 03 18 d3 0f 10 16 1a 04 31 30 30 30 28 a8 83 90 cb 03 60 d8 cc 02 68 83 a4 03 ]
D 2021-05-10 15:42:57.297|977427:977443|BNRemotingRegisterService.c|200|Register service heartbeat timer
D 2021-05-10 15:42:57.297|977427:977443|BNRemotingRegisterService.c|207|Register session remoting ip:168.1.100.126 | port:2222
D 2021-05-10 15:42:57.297|977427:977443|BNMessageRegHeartbeatProcess.c|41|BnMessageRegHeartbeatReqSerialize:topicTotalmsg[42584], queueTotalmsg[53763]
D 2021-05-10 15:42:57.297|977427:977443|BNMessageRegHeartbeatProcess.c|51|B_BNPackRequestHeartbeat:[00 00 07 d3 00 00 00 1b 0a 03 18 d3 0f 10 16 1a 04 31 30 30 30 28 a8 83 90 cb 03 60 d8 cc 02 68 83 a4 03 ]
D 2021-05-10 15:43:27.298|977427:977443|BNRemotingRegisterService.c|200|Register service heartbeat timer
D 2021-05-10 15:43:27.298|977427:977443|BNRemotingRegisterService.c|207|Register session remoting ip:168.1.100.126 | port:2222
D 2021-05-10 15:43:27.298|977427:977443|BNMessageRegHeartbeatProcess.c|41|BnMessageRegHeartbeatReqSerialize:topicTotalmsg[42584], queueTotalmsg[53763]
D 2021-05-10 15:43:27.298|977427:977443|BNMessageRegHeartbeatProcess.c|51|B_BNPackRequestHeartbeat:[00 00 07 d3 00 00 00 1b 0a 03 18 d3 0f 10 16 1a 04 31 30 30 30 28 a8 83 90 cb 03 60 d8 cc 02 68 83 a4 03 ]
D 2021-05-10 15:43:57.299|977427:977443|BNRemotingRegisterService.c|200|Register service heartbeat timer
```

4.3 客户端安装配置

4.3.1 安装步骤

1. 使用命令**uname -m**查看系统架构，确认为aarch64或是x86_64架构，例如：

```
$ uname -m
aarch64
```

或

```
$ uname -m
x86_64
```

2. 把对应架构的产品安装包上传到新建用户或已存在的用户目录下, 例如: **/home/xxx**
3. 解压TongLINK/Q客户端安装包: (xxx代表不同的架构平台)
tar xvf Install_TLQCli_Standard_Linux4.19.90_xxx_9.0.0.0.tar.gz

4.3.2 目录及文件

解压安装完成后，客户端（client）会生成如下目录及文件。

表 4.3-1：客户端目录及文件

目录	目录下文件及说明
TLQ9_Cli/bin	tlq9version: 显示客户端版本号
TLQ9_Cli/etc	tlq_config_client.xml: 客户端配置文件
TLQ9_Cli/incl	TLQ.h: C客户端对外提供的头文件
TLQ9_Cli/java/conf	Logback.xml: java客户端日志配置文件
TLQ9_Cli/java/lib	tlq9client.jar: java客户端jar包程序
TLQ9_Cli/lib	libtlq9client.so: C客户端动态库
TLQ9_Cli/log	日志文件目录
TLQ9_Cli/samples/bin	*: 客户端示例程序
TLQ9_Cli/samples/demo_c	Makefile: 编译客户端示例源码的makefile readme.txt: 使用说明 *.c: C客户端示例的源码
TLQ9_Cli/samples/demo_java/base	compleie: java samples 编译shell readme.txt: 使用说明 *.java: JAVA客户端的示例程序
TLQ9_Cli/	setp: 客户端环境变量设置脚本

4.3.3 配置环境变量

进入 **TLQ9_Cli** 目录下设置环境变量：

source setp 或 **./setp**（两个点之间有空格）

4.3.4 修改配置文件

客户端配置文件为 **TLQ9_Cli/etc/tlq_config_client.xml** 中具体的参数属性说明可参考《002_TongLINKQ9.0安装及系统配置手册》。

按以下步骤可对客户端的配置文件进行修改。

1. 进入客户端配置文件目录 **TLQ9_Cli/etc**
cd TLQ9_Cli/etc
2. 修改该目录下的配置文件 **tlq_config_client**

vi tlq_config_client

必须修改的配置项有以下1处，例如：

<managerAddress>：注册管理节点的传输协议，IP 和端口

说明： IP 和端口必须修改，需和 **TLQ9_Mgr/config/nameServer.xml** 中的**<IpAddress>**及**<ListenPort>**相同。

4.3.5 查看日志

客户端日志目录 **TLQ9_Cli/log** 下会生成 **dbg** 和 **syslog** 文件：**dbg** 存储的是调试日志信息，**syslog** 文件存放系统信息的日志内容。运行如下命令查看生成的日志文件：

```
[xxx@tstsvr log]$ ls  
  
client.dbg.0  client.syslog.0
```


第5章 应用场景示例

本章将以第2章节中介绍过的**发布与订阅**的概念，对其应用场景进行举例说明，以便用户更快、更好、更直观地了解TongLINK/Q产品。

5.1 发布与订阅流程

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，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_{消息}，无需了解哪些发布者存在。

发布者/订阅者应用模式支持向一个特定的消息主题生产消息。0或多个订阅者可能对接收来自特定消息主题的消息感兴趣。每条消息可以有0或多个订阅者；订阅者只有订阅后才能接收到消息。

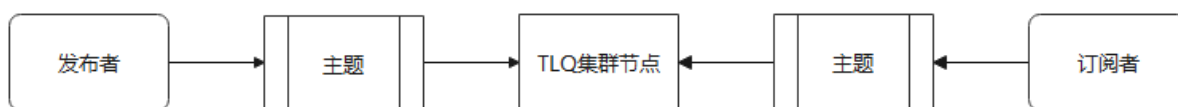


图 5.1-1：发布与订阅流程

1. 消息发布者数据流向：发布者携带主题消息发布到TLQ集群节点
2. 消息订阅者数据流向：订阅者向TLQ集群节点订阅已发布的主题消息

5.2 服务端配置及启动

部署 1 个**管理节点**，示例如下：

用户	目录	IP	端口
tlq9	/home/tlq9/TLQ9_Mgr	<ip1>	<port1>默认9888

1. 安装管理节点：参考“管理节点安装配置及启动”中的[安装步骤](#)
2. 配置及启动管理节点，参考[管理节点安装配置及启动](#)

部署 1 个**工作节点**，示例如下：

工作节点部署示例：

用户	目录	IP	端口
tlq9	/home/tlq9/TLQ9_Bok	<ip2>	<port2>默认9876

1. 安装工作节点：参考“工作节点安装配置及启动”中的[安装步骤](#)
2. 配置及启动工作节点，参考[工作节点安装配置及启动](#)
3. 保证工作节点向管理节点注册成功并保持常连接，拓扑图如下所示：

说明：管理节点和工作节点可配置在同一主机上

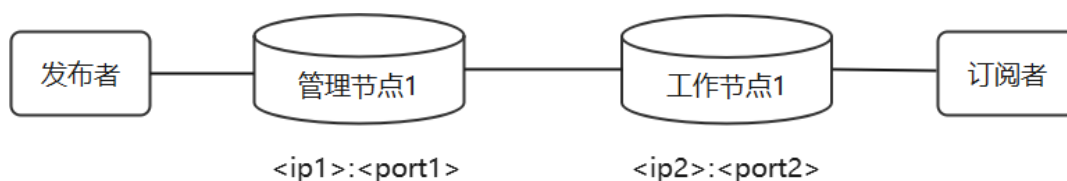


图 5.2-1：发布与订阅拓扑图

5.3 客户端安装及演示

5.3.1 客户端安装

部署 1 个客户端，示例如下：

用户	目录
tlq9	/home/tlq9/TLQ9_Cli

1. 安装客户端：参考“客户端安装配置”中的[安装步骤](#)
2. 进入 **TLQ9_Cli** 目录下设置环境变量：
source setp 或 **./setp**（两个点之间有空格）

5.3.2 demo_c编译及演示

5.3.2.1 编译C示例程序

本章节测试使用到的示例程序都需要基于示例代码进行编译后才能使用。

示例程序编译生成方法如下：

1. 进入 **TLQ9_Cli/samples/demo_c** 目录下编译可执行程序：

make

```

[xxx@tstsvr demo_c]$ make
gcc TLQPairProducerAsyncSendFile.c -g -Wall -Werror -fPIC -I../incl -L../lib -ltlq9client -pthread -ldl -o ../bin/TLQPairProducerAsyncSendFile
gcc TLQPairConsumerAsync.c -g -Wall -Werror -fPIC -I../incl -L../lib -ltlq9client -pthread -ldl -o ../bin/TLQPairConsumerAsync
gcc TLQProducerSampleFile.c -g -Wall -Werror -fPIC -I../incl -L../lib -ltlq9client -pthread -ldl -o ../bin/TLQProducerSampleFile
gcc TLQConsumerPullFile.c -g -Wall -Werror -fPIC -I../incl -L../lib -ltlq9client -pthread -ldl -o ../bin/TLQConsumerPullFile
gcc TLQProducerAsync.c -g -Wall -Werror -fPIC -I../incl -L../lib -ltlq9client -pthread -ldl -
    
```

```
o ../bin/TLQProducerAsync
gcc TLQPairConsumerAsyncPullFile.c -g -Wall -Werror -fPIC -I../incl -L../lib -ltlq9client -pthread -ldl -
-o ../bin/TLQPairConsumerAsyncPullFile
gcc TLQPairConsumerBuff.c -g -Wall -Werror -fPIC -I../incl -L../lib -ltlq9client -pthread -ldl -
o ../bin/TLQPairConsumerBuff
gcc TLQConsumerSampleFile.c -g -Wall -Werror -fPIC -I../incl -L../lib -ltlq9client -pthread -ldl -
o ../bin/TLQConsumerSampleFile
gcc TLQProducerSendFile.c -g -Wall -Werror -fPIC -I../incl -L../lib -ltlq9client -pthread -ldl -
o ../bin/TLQProducerSendFile
gcc TLQConsumerAsync.c -g -Wall -Werror -fPIC -I../incl -L../lib -ltlq9client -pthread -ldl -
o ../bin/TLQConsumerAsync
gcc TLQProducerBatchAsync.c -g -Wall -Werror -fPIC -I../incl -L../lib -ltlq9client -pthread -ldl -
o ../bin/TLQProducerBatchAsync
gcc TLQProducerAsyncSendFile.c -g -Wall -Werror -fPIC -I../incl -L../lib -ltlq9client -pthread -ldl -
o ../bin/TLQProducerAsyncSendFile
gcc TLQPairProducerSendFile.c -g -Wall -Werror -fPIC -I../incl -L../lib -ltlq9client -pthread -ldl -
o ../bin/TLQPairProducerSendFile
gcc TLQPairProducerBatch.c -g -Wall -Werror -fPIC -I../incl -L../lib -ltlq9client -pthread -ldl -
o ../bin/TLQPairProducerBatch
gcc TLQConsumerAsyncPullFile.c -g -Wall -Werror -fPIC -I../incl -L../lib -ltlq9client -pthread -ldl -
o ../bin/TLQConsumerAsyncPullFile
gcc TLQProducerSample.c -g -Wall -Werror -fPIC -I../incl -L../lib -ltlq9client -pthread -ldl -
o ../bin/TLQProducerSample
gcc TLQPairProducerAsync.c -g -Wall -Werror -fPIC -I../incl -L../lib -ltlq9client -pthread -ldl -
o ../bin/TLQPairProducerAsync
gcc TLQConsumerSample.c -g -Wall -Werror -fPIC -I../incl -L../lib -ltlq9client -pthread -ldl -
o ../bin/TLQConsumerSample
gcc TLQPairProducerBuff.c -g -Wall -Werror -fPIC -I../incl -L../lib -ltlq9client -pthread -ldl -
o ../bin/TLQPairProducerBuff
gcc TLQPairConsumerPullFile.c -g -Wall -Werror -fPIC -I../incl -L../lib -ltlq9client -pthread -ldl -
o ../bin/TLQPairConsumerPullFile
gcc TLQProducerBatch.c -g -Wall -Werror -fPIC -I../incl -L../lib -ltlq9client -pthread -ldl -
o ../bin/TLQProducerBatch
gcc TLQPairProducerBatchAsync.c -g -Wall -Werror -fPIC -I../incl -L../lib -ltlq9client -pthread -ldl -
o ../bin/TLQPairProducerBatchAsync
```

2. 进入 **TLQ9_Cli/samples/bin** 下查看成功编译的示例程序：

ls -ls

```
[xxx@tstsvr bin]$ ls -ls
总用量 500
24 -rwxr-xr-x. 1 tlq9c users 22504 8月 6 18:40 TLQConsumerAsync
20 -rwxr-xr-x. 1 tlq9c users 18192 8月 6 18:40 TLQConsumerAsyncPullFile
20 -rwxr-xr-x. 1 tlq9c users 17912 8月 6 18:40 TLQConsumerPullFile
24 -rwxr-xr-x. 1 tlq9c users 22760 8月 6 18:40 TLQConsumerSample
```

20 -rwxr-xr-x. 1 tlq9c users 17816 8月	6 18:40 TLQConsumerSampleFile
20 -rwxr-xr-x. 1 tlq9c users 17776 8月	6 18:40 TLQPairConsumerAsync
20 -rwxr-xr-x. 1 tlq9c users 17480 8月	6 18:40 TLQPairConsumerAsyncPullFile
20 -rwxr-xr-x. 1 tlq9c users 17704 8月	6 18:40 TLQPairConsumerBuff
20 -rwxr-xr-x. 1 tlq9c users 17240 8月	6 18:40 TLQPairConsumerPullFile
24 -rwxr-xr-x. 1 tlq9c users 23800 8月	6 18:40 TLQPairProducerAsync
20 -rwxr-xr-x. 1 tlq9c users 18336 8月	6 18:40 TLQPairProducerAsyncSendFile
28 -rwxr-xr-x. 1 tlq9c users 24736 8月	6 18:40 TLQPairProducerBatch
28 -rwxr-xr-x. 1 tlq9c users 25304 8月	6 18:40 TLQPairProducerBatchAsync
24 -rwxr-xr-x. 1 tlq9c users 23312 8月	6 18:40 TLQPairProducerBuff
20 -rwxr-xr-x. 1 tlq9c users 17712 8月	6 18:40 TLQPairProducerSendFile
24 -rwxr-xr-x. 1 tlq9c users 23792 8月	6 18:40 TLQProducerAsync
20 -rwxr-xr-x. 1 tlq9c users 18256 8月	6 18:40 TLQProducerAsyncSendFile
28 -rwxr-xr-x. 1 tlq9c users 24608 8月	6 18:40 TLQProducerBatch
28 -rwxr-xr-x. 1 tlq9c users 25176 8月	6 18:40 TLQProducerBatchAsync
24 -rwxr-xr-x. 1 tlq9c users 23128 8月	6 18:40 TLQProducerSample
24 -rwxr-xr-x. 1 tlq9c users 23096 8月	6 18:40 TLQProducerSampleFile
20 -rwxr-xr-x. 1 tlq9c users 17776 8月	6 18:40 TLQProducerSendFile

5.3.2.2 C示例程序演示步骤

1. 运行以下命令**发布**主题+域的消息：

TLQProducerSample tcp://<IP>:<Port>

说明：IP 地址、端口改为管理节点服务对应的 IP 地址、端口，示例程序默认使用的主题：**topic1**，域：**zone1**

发布主题+域的消息成功：通过客户端打印出日志 “**PUT MESSAGE END.**”。

```
PUT MESSAGE END.
Destroy producer END.
```

2. 运行以下命令**订阅**主题+域的消息：

TLQConsumerSample tcp://<IP>:<Port>

说明：IP 地址、端口改为管理节点服务对应的 IP 地址、端口，示例程序默认使用的主题：**topic1**，域：**zone1**

订阅主题+域的消息成功：通过客户端打印出消息日志，表明订阅了这条主题+域的消息。

```
attrValue:value
consumer is destroyed.
```

5.3.3 demo_java编译及演示

5.3.3.1 编译Java示例程序

本章节测试使用到的示例程序都需要基于示例代码进行编译后才能使用。
示例程序编译生成方法如下：

1. 进入 **TLQ9_Cli/samples/demo_java/base** 目录下编译可执行程序：
source compile 或 **./compile**（两个点之间有空格）
2. 进入 **TLQ9_Cli/samples/bin** 下查看成功编译的示例程序：

ls -ls

```
[xxx@tstsvr bin]$ ls -ls
总用量 108
8 -rw-rw-r-- 1 hui hui 5708 8月 9 17:24 TLQLightPairProducerTest.class
4 -rw-rw-r-- 1 hui hui 735 8月 9 17:24 'TLQLightPairPullConsumerTest$1.class'
4 -rw-rw-r-- 1 hui hui 3646 8月 9 17:24 TLQLightPairPullConsumerTest.class
8 -rw-rw-r-- 1 hui hui 5643 8月 9 17:24 TLQLightTopicProducerTest.class
4 -rw-rw-r-- 1 hui hui 738 8月 9 17:24 'TLQLightTopicPullConsumerTest$1.class'
8 -rw-rw-r-- 1 hui hui 4504 8月 9 17:24 TLQLightTopicPullConsumerTest.class
4 -rw-rw-r-- 1 hui hui 1715 8月 9 17:24 'TLQPairProducerTest$1.class'
4 -rw-rw-r-- 1 hui hui 1586 8月 9 17:24 'TLQPairProducerTest$2.class'
4 -rw-rw-r-- 1 hui hui 1690 8月 9 17:24 'TLQPairProducerTest$3.class'
8 -rw-rw-r-- 1 hui hui 5784 8月 9 17:24 TLQPairProducerTest.class
4 -rw-rw-r-- 1 hui hui 1757 8月 9 17:24 'TLQPairPullConsumerTest$1.class'
4 -rw-rw-r-- 1 hui hui 1341 8月 9 17:24 'TLQPairPullConsumerTest$2.class'
4 -rw-rw-r-- 1 hui hui 3874 8月 9 17:24 TLQPairPullConsumerTest.class
4 -rw-rw-r-- 1 hui hui 1771 8月 9 17:24 'TLQTopicProducerTest$1.class'
4 -rw-rw-r-- 1 hui hui 1632 8月 9 17:24 'TLQTopicProducerTest$2.class'
4 -rw-rw-r-- 1 hui hui 1696 8月 9 17:24 'TLQTopicProducerTest$3.class'
8 -rw-rw-r-- 1 hui hui 6170 8月 9 17:24 TLQTopicProducerTest.class
4 -rw-rw-r-- 1 hui hui 2347 8月 9 17:24 'TLQTopicPullConsumerTest$1.class'
4 -rw-rw-r-- 1 hui hui 1401 8月 9 17:24 'TLQTopicPullConsumerTest$2.class'
4 -rw-rw-r-- 1 hui hui 2255 8月 9 17:24 'TLQTopicPullConsumerTest$3.class'
8 -rw-rw-r-- 1 hui hui 6604 8月 9 17:24 TLQTopicPullConsumerTest.class
```

5.3.3.2 Java示例程序演示步骤

1. 运行以下命令**发布**主题+域的消息：

java -Dtype="topicProducerSendMessageAsync" -DnameSrv="<IP>:<Port>" TLQTopicProducerTest

说明：IP 地址、端口改为管理节点服务对应的 IP 地址、端口，示例程序默认使用的主题：**topic1**，域：**zone1**

发布主题+域的消息成功：通过客户端打印出发送消息的结果。

胖客户端Topic模式异步发生消息 SendResult [sendStatus=SEND_OK, msgId=808000015ADA2A139A552CFF99460000, brokerId=1010]
 发送消息总数量： 1
 花费时间： 503毫秒
 发送消息成功数量： 1
 发送消息失败数量： 0
 发送异常： 0

2. 运行以下命令订阅主题+域的消息：

java -Dtype="topicPullMessageAsync" -DnameSrv="<IP>:<Port>" TLQTopicPullConsumerTest

说明：IP 地址、端口改为管理节点服务对应的 IP 地址、端口，示例程序默认使用的主题：**topic1**，域：**zone1**

订阅主题+域的消息成功：通过客户端打印出消息结果，表明订阅了这条主题+域的消息。

已成功拉取消息数量： 1 ，胖客户端Topic模式异步拉取消息返回：PullResult{pullStatus=FOUND, clientId='23675', consumerId='U7q9sVi2vCDLDbKG', groupName='consumer_topic_pullAsync_001', topic='topic1', queueId=-1, minConsumeQueueOffset=0, maxConsumeQueueOffset=0, consumeHistoryOffset=-1, statueCode=0, domain='zone1', msgFoundList=[MessageExt{time=1628477711,reconsumeTimes=0,srcNode='', proofIndex=0, consumeQueueOffset=0, queueID=0, cluster=BrokerCluster, domain=zone1, brokerId=1010, commitLogOffset=0, Message{topic='topic1', attr=null, body=LKZ3XG3OW3, transactionId='null', msgId='808000017FE058644D462B8AB8580004', properties='{PERSISTENCE=0, EXPIRY=0, PRIORITY=0}}}}]
 拉取消息总数量： 1
 花费时间： 1014毫秒
 拉取消息成功数量： 1
 拉取消息为空数量： 0
 拉取消息异常数量： 0

