

RocketMQ在Windows平台下环境搭建

一. 环境搭建

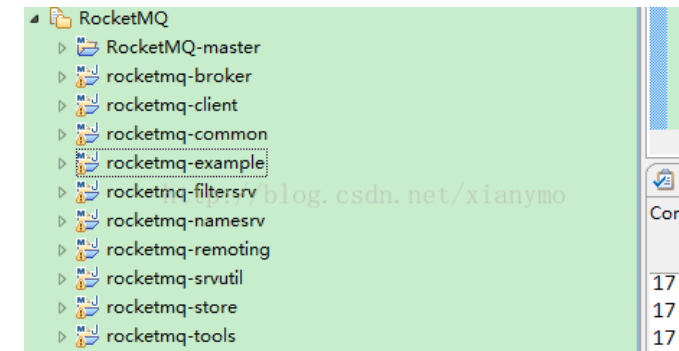
需要jdk1.6(以上) 64bit, maven, eclipse

二. RocketMQ项目下载

项目地址: <https://github.com/alibaba/RocketMQ>, 将下载的RocketMQ-master放到eclipse工作空间中

三. 将RocketMQ-master导入到eclipse中

1. 将项目导入eclipse, 如下图



2. 在我下载的RocketMQ-master的pom.xml文件的parent有个问题, 默认如下:

```
1. <parent>
2.   <groupId>com.taobao</groupId>
3.   <artifactId>parent</artifactId>
4.   <version>1.0.2</version>
5. </parent>
6.
7. <!-- <parent>
8.   <groupId>org.sonatype.oss</groupId>
9.   <artifactId>oss-parent</artifactId>
10.  <version>7</version>
11. </parent> -->
12.
```

但是编译时总报错parent找不到, 而用下面的parent, 则编译通过。

```
1. <!--<parent>
2.   <groupId>com.taobao</groupId>
3.   <artifactId>parent</artifactId>
4.   <version>1.0.2</version>
5. </parent>-->
6.
7. <parent>
8.   <groupId>org.sonatype.oss</groupId>
9.   <artifactId>oss-parent</artifactId>
10.  <version>7</version>
11. </parent>
```

3. 由于我用的是jdk1.7, 故修改RocketMQ-master.pom的jdk版本

```
1. <properties>
2.   <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
3.   <!--maven properties -->
4.   <maven.test.skip>true</maven.test.skip>
5.   <maven.jdoc.skip>true</maven.jdoc.skip>
6.   <downloadSources>true</downloadSources>
7.   <!-- compiler settings properties -->
8.   <java_source_version>1.7</java_source_version>
```

昵称: marco_tan
园龄: 1年8个月
粉丝: 3
关注: 0
[+加关注](#)

<2016年10月>

日	一	二	三	四	五	六
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

搜索

常用链接

[我的随笔](#)

[我的评论](#)

[我的参与](#)

[最新评论](#)

[我的标签](#)

随笔分类

[css\(1\)](#)

[Hibernate\(10\)](#)

[IO\(4\)](#)

[Java\(10\)](#)

[java8\(1\)](#)

[jetty\(4\)](#)

[js\(1\)](#)

[JSP](#)

[JUnit\(4\)](#)

[jvm\(5\)](#)

[linux\(3\)](#)

[MINA\(10\)](#)

[MVC\(1\)](#)

[mvn\(5\)](#)

[netty](#)

[NIO\(10\)](#)

[RocketMQ\(6\)](#)

[servlet\(2\)](#)

[Spring\(7\)](#)

[SQL\(2\)](#)

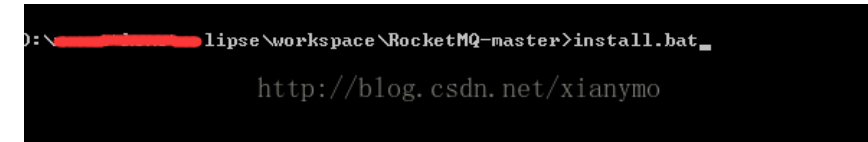
[Tomcat\(4\)](#)

[web\(1\)](#)

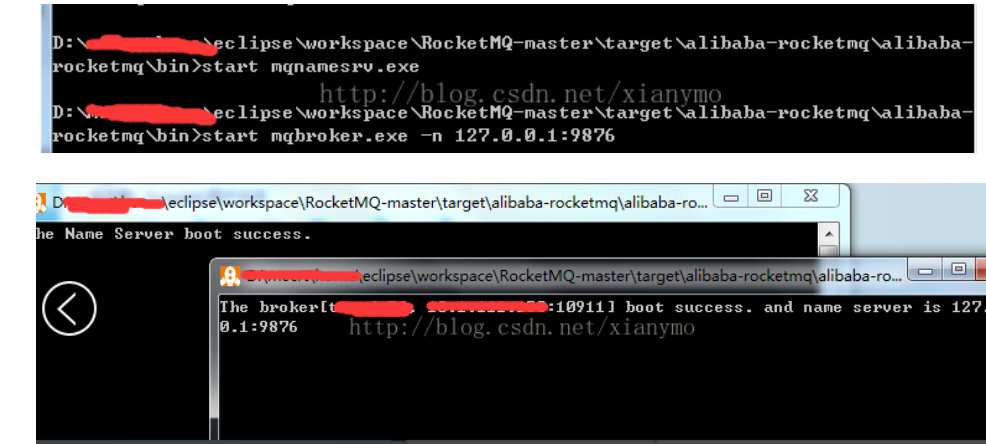
```
9.         <java_target_version>1.7</java_target_version>
10.        <file_encoding>UTF-8</file_encoding>
11.    </properties>
```

四. 编译RocketMQ项目

1. 在命令行执行在RocketMQ-master文件夹下的install.bat批处理



2. 该命令会编译整个项目，并在RocketMQ-master目录下生成一个target文件夹
3. 进入刚生成的target文件夹下的bin目录，在命令行中执行：start mqnamesrv.exe，会弹出一个信息窗口，显示The name Server boot success 说明启动成功了，接着启动borker，在命令行中执行：start mqbroker.exe -n 127.0.0.1:9876 同样的弹出一个窗口，看到success表示成功了。



五. 启动Producer和Customer

1. 在RocketMQ-example项目中加入Producer.java

```
1. public class Producer {
2.     public static void main(String[] args) throws MQClientException,
3.         InterruptedException{
4.         /**
5.          * 一个应用创建一个Producer，由应用来维护此对象，可以设置为全局对象或者单例<br>
6.          * 注意：ProducerGroupName需要由应用来保证唯一<br>
7.          * ProducerGroup这个概念发送普通的消息时，作用不大，但是发送分布式事务消息时，比较关键。
8.          * 因为服务器会回查这个Group下的任意一个Producer
9.          */
10.        final DefaultMQProducer producer = new DefaultMQProducer("ProducerGroupName");
11.        producer.setNamesrvAddr("127.0.0.1:9876");
12.        producer.setInstanceName("Producer");
13.
14.        /**
15.         * Producer对象在使用之前必须要调用start初始化，初始化一次即可<br>
16.         * 注意：切记不可以在每次发送消息时，都调用start方法
17.         */
18.        producer.start();
19.
20.        /**
21.         * 下面这段代码表明一个Producer对象可以发送多个topic，多个tag的消息。
22.         * 注意：send方法是同步调用，只要不抛异常就标识成功。但是发送成功也可会有多种状态，<br>
23.         * 例如消息写入Master成功，但是Slave不成功，这种情况消息属于成功，但是对于个别应用如果对消息可靠性要求极高，<br>
24.         * 需要对这种情况做处理。另外，消息可能会存在发送失败的情况，失败重试由应用来处理。
25.         */
26.        for (int i = 0; i < 10; i++){
27.            try {
28.                {
29.                    Message msg = new Message("TopicTest1",// topic
30.                        "TagA",// tag
31.                        "OrderID001",// key
32.                        ("Hello MetaQA").getBytes());// body
```

XML(3)
多线程(6)
数据结构(9)
算法(1)
系统架构(2)
性能优化(1)
正则表达式(1)

随笔档案
2015年1月 (15)
2014年12月 (1)
2014年11月 (13)
2014年10月 (25)
2014年9月 (18)
2014年8月 (31)
2014年7月 (7)
2014年6月 (6)
2014年4月 (14)

最新评论
1. Re:RocketMQ在Windows平台下环境搭建
执行完编译之后没有.exe文件阿，请楼主解答
--灵魂盗猎者
2. Re:RocketMQ入门（1）
很不错的文章，赞一个！
--灵魂盗猎者
3. Re:RocketMQ在Windows平台下环境搭建
请问在进入刚生成的target文件夹下的bin目录，在命令行中执行：start mqnamesrv.exe，会弹出一个信息窗口这一步中 bin目录下并没有mqnamesrv.exe 只有mqnames.....
--小野人~kobe
4. Re:RocketMQ在Windows平台下环境搭建
你好，运行start mqnamesrv时出现window s版本不兼容怎么办？我是win7 64位。
--Leo_璇哥

阅读排行榜
1. RocketMQ在linux平台下环境搭建(1313)
2. RocketMQ在Windows平台下环境搭建(723)
3. RocketMQ入门（1）(304)
4. RocketMQ入门（3）拉取消息(237)
5. RocketMQ入门（2）最佳实践(216)

评论排行榜
1. RocketMQ在Windows平台下环境搭建(3)
2. RocketMQ入门（1）(1)

推荐排行榜

```

33.         SendResult sendResult = producer.send(msg);
34.         System.out.println(sendResult);
35.     }
36.
37.     {
38.         Message msg = new Message("TopicTest2", // topic
39.             "TagB", // tag
40.             "OrderID0034", // key
41.             ("Hello MetaQB").getBytes()); // body
42.         SendResult sendResult = producer.send(msg);
43.         System.out.println(sendResult);
44.     }
45.
46.     {
47.         Message msg = new Message("TopicTest3", // topic
48.             "TagC", // tag
49.             "OrderID0061", // key
50.             ("Hello MetaQC").getBytes()); // body
51.         SendResult sendResult = producer.send(msg);
52.         System.out.println(sendResult);
53.     }
54. } catch (Exception e) {
55.     e.printStackTrace();
56. }
57. TimeUnit.MILLISECONDS.sleep(1000);
58. }
59.
60. /**
61.  * 应用退出时，要调用shutdown来清理资源，关闭网络连接，从MetaQ服务器上注销自己
62.  * 注意：我们建议应用在JBoss、Tomcat等容器的退出钩子里调用shutdown方法
63.  */
64. //producer.shutdown();
65. Runtime.getRuntime().addShutdownHook(new Thread(new Runnable() {
66.     public void run() {
67.         producer.shutdown();
68.     }
69. }));
70. System.exit(0);
71. }
72. }

```

[1. RocketMQ入门（2）最佳实践\(1\)](#)
[2. RocketMQ入门（1）\(1\)](#)

2. 加入Customer.java

```

1. public class Consumer {
2.     /**
3.      * 当前例子是PushConsumer用法，使用方式给用户感觉是消息从RocketMQ服务器推到了应用客户端。<br>
4.      * 但是实际PushConsumer内部是使用长轮询Pull方式从MetaQ服务器拉消息，然后再回调用户Listener方法
5.      <br>
6.      */
7.     public static void main(String[] args) throws InterruptedException,
8.         MQClientException{
9.         /**
10.          * 一个应用创建一个Consumer，由应用来维护此对象，可以设置为全局对象或者单例<br>
11.          * 注意：ConsumerGroupName需要由应用来保证唯一
12.          */
13.         DefaultMQPushConsumer consumer = new DefaultMQPushConsumer(
14.             "ConsumerGroupName");
15.         consumer.setNamesrvAddr("127.0.0.1:9876");
16.         consumer.setInstanceName("Consumer");
17.
18.         /**
19.          * 订阅指定topic下tags分别等于TagA或TagC或TagD
20.          */
21.         consumer.subscribe("TopicTest1", "TagA || TagC || TagD");
22.         /**
23.          * 订阅指定topic下所有消息<br>
24.          * 注意：一个consumer对象可以订阅多个topic
25.          */
26.         consumer.subscribe("TopicTest2", "*");
27.
28.         consumer.registerMessageListener(new MessageListenerConcurrently() {
29.             public ConsumeConcurrentlyStatus consumeMessage(
30.                 List<MessageExt> msgs, ConsumeConcurrentlyContext context)

```

```

{
31.
32.         System.out.println(Thread.currentThread().getName()
33.             + " Receive New Messages: " + msgs.size());
34.
35.         MessageExt msg = msgs.get(0);
36.         if(msg.getTopic().equals("TopicTest1")) {
37.             //执行TopicTest1的消费逻辑
38.             if(msg.getTags() != null && msg.getTags().equals("TagA"))
39.
40.             {
41.
42.                 //执行TagA的消费
43.                 System.out.println(new String(msg.getBody()));
44.
45.             }else if (msg.getTags() != null
46.                 &&msg.getTags().equals("TagC")) {
47.                 //执行TagC的消费
48.                 System.out.println(new String(msg.getBody()));
49.
50.             }else if (msg.getTags() != null
51.                 &&msg.getTags().equals("TagD")) {
52.                 //执行TagD的消费
53.                 System.out.println(new String(msg.getBody()));
54.
55.             }
56.             }else if (msg.getTopic().equals("TopicTest2")) {
57.                 System.out.println(new String(msg.getBody()));
58.             }
59.
60.             return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
61.
62.         }
63.     });
64.
65.     /**
66.      * Consumer对象在使用之前必须要调用start初始化，初始化一次即可<br>
67.      */
68.     consumer.start();
69.
70.     System.out.println("ConsumerStarted.");
71. }
72. }

```

3. 运行Producer

```

1. 19:50:57.233 [main] DEBUG i.n.u.i.l.InternalLoggerFactory - Using SLF4J as the default logging framework
2. 19:50:57.259 [main] DEBUG i.n.c.MultithreadEventLoopGroup - -Dio.netty.eventLoopThreads: 4
3. 19:50:57.274 [main] DEBUG i.n.util.internal.PlatformDependent0 - java.nio.Buffer.address: available
4. 19:50:57.274 [main] DEBUG i.n.util.internal.PlatformDependent0 - sun.misc.Unsafe.theUnsafe: available
5. 19:50:57.274 [main] DEBUG i.n.util.internal.PlatformDependent0 - sun.misc.Unsafe.copyMemory: available
6. 19:50:57.275 [main] DEBUG i.n.util.internal.PlatformDependent0 - java.nio.Bits.unaligned: true
7. 19:50:57.275 [main] DEBUG i.n.util.internal.PlatformDependent - Platform: Windows
8. 19:50:57.276 [main] DEBUG i.n.util.internal.PlatformDependent - Java version: 7
9. 19:50:57.276 [main] DEBUG i.n.util.internal.PlatformDependent - -Dio.netty.noUnsafe: false
10. 19:50:57.276 [main] DEBUG i.n.util.internal.PlatformDependent - sun.misc.Unsafe: available
11. 19:50:57.277 [main] DEBUG i.n.util.internal.PlatformDependent - -Dio.netty.noJavassist: false
12. 19:50:57.506 [main] DEBUG i.n.util.internal.PlatformDependent - Javassist: available
13. 19:50:57.507 [main] DEBUG i.n.util.internal.PlatformDependent - -Dio.netty.tmpdir: C:\Users\tannj\AppData\Local\Temp (java.io.tmpdir)
14. 19:50:57.507 [main] DEBUG i.n.util.internal.PlatformDependent - -Dio.netty.bitMode: 64 (sun.arch.data.model)
15. 19:50:57.507 [main] DEBUG i.n.util.internal.PlatformDependent - -Dio.netty.noPreferDirect: false
16. 19:50:57.544 [main] DEBUG io.netty.channel.nio.NioEventLoop - -Dio.netty.noKeySetOptimization: false
17. 19:50:57.544 [main] DEBUG io.netty.channel.nio.NioEventLoop - -Dio.netty.selectorAutoRebuildThreshold: 512
18. 19:50:57.875 [main] DEBUG i.n.util.internal.ThreadLocalRandom - -Dio.netty.initialSeedUniquifier: 0xfaf4f653b3d8be50 (took 52 ms)
19. 19:50:57.927 [main] DEBUG io.netty.buffer.ByteBufUtil - -Dio.netty allocator.type: unpooled
20. 19:50:57.927 [main] DEBUG io.netty.buffer.ByteBufUtil - -Dio.netty.threadLocalDirectBufferSize: 65536

```

```
21. 19:50:57.958 [NettyClientSelector_1] DEBUG i.n.u.i.JavassistTypeParameterMatcherGenerator - Generated: io.netty.util.internal.__matchers__.com.alibaba.rocketmq.remoting.protocol.RemotingCommandMatcher
22. 19:50:58.006 [main] DEBUG io.netty.util.Recycler - - Dio.netty.recycler.maxCapacity.default: 262144
23. 19:50:58.027 [NettyClientWorkerThread_1] DEBUG io.netty.util.ResourceLeakDetector - - Dio.netty.leakDetectionLevel: simple
24. 19:50:58.255 [NettyClientSelector_1] DEBUG io.netty.util.internal.Cleaner0 - java.nio.ByteBuffer.cleaner(): available
25. SendResult [sendStatus=SEND_OK, msgId=0A016F9600002A9F000000000272A36A, messageQueue=MessageQueue [topic=TopicTest, brokerName=tannj-PC, queueId=0], queueOffset=73578]
26. SendResult [sendStatus=SEND_OK, msgId=0A016F9600002A9F000000000272A3F2, messageQueue=MessageQueue [topic=TopicTest, brokerName=tannj-PC, queueId=1], queueOffset=73577]
27. SendResult [sendStatus=SEND_OK, msgId=0A016F9600002A9F000000000272A47A, messageQueue=MessageQueue [topic=TopicTest, brokerName=tannj-PC, queueId=2], queueOffset=73577]
28. SendResult [sendStatus=SEND_OK, msgId=0A016F9600002A9F000000000272A502, messageQueue=MessageQueue [topic=TopicTest, brokerName=tannj-PC, queueId=3], queueOffset=73576]
29. SendResult [sendStatus=SEND_OK, msgId=0A016F9600002A9F000000000272A58A, messageQueue=MessageQueue [topic=TopicTest, brokerName=tannj-PC, queueId=0], queueOffset=73579]
30. SendResult [sendStatus=SEND_OK, msgId=0A016F9600002A9F000000000272A612, messageQueue=MessageQueue [topic=TopicTest, brokerName=tannj-PC, queueId=1], queueOffset=73578]
31. SendResult [sendStatus=SEND_OK, msgId=0A016F9600002A9F000000000272A69A, messageQueue=MessageQueue [topic=TopicTest, brokerName=tannj-PC, queueId=2], queueOffset=73578]
32. SendResult [sendStatus=SEND_OK, msgId=0A016F9600002A9F000000000272A722, messageQueue=MessageQueue [topic=TopicTest, brokerName=tannj-PC, queueId=3], queueOffset=73577]
33. SendResult [sendStatus=SEND_OK, msgId=0A016F9600002A9F000000000272A7AA, messageQueue=MessageQueue [topic=TopicTest, brokerName=tannj-PC, queueId=0], queueOffset=73580]
34. SendResult [sendStatus=SEND_OK, msgId=0A016F9600002A9F000000000272A832, messageQueue=MessageQueue [topic=TopicTest, brokerName=tannj-PC, queueId=1], queueOffset=73579]
```

4. 运行Customer

```
1. 19:51:49.059 [main] DEBUG i.n.u.i.l.InternalLoggerFactory - Using SLF4J as the default logging framework
2. 19:51:49.069 [main] DEBUG i.n.c.MultithreadEventLoopGroup - -Dio.netty.eventLoopThreads: 4
3. 19:51:49.083 [main] DEBUG i.n.util.internal.PlatformDependent0 - java.nio.Buffer.address: available
4. 19:51:49.084 [main] DEBUG i.n.util.internal.PlatformDependent0 - sun.misc.Unsafe.theUnsafe: available
5. 19:51:49.084 [main] DEBUG i.n.util.internal.PlatformDependent0 - sun.misc.Unsafe.copyMemory: available
6. 19:51:49.085 [main] DEBUG i.n.util.internal.PlatformDependent0 - java.nio.Bits.unaligned: true
7. 19:51:49.085 [main] DEBUG i.n.util.internal.PlatformDependent - Platform: Windows
8. 19:51:49.086 [main] DEBUG i.n.util.internal.PlatformDependent - Java version: 7
9. 19:51:49.086 [main] DEBUG i.n.util.internal.PlatformDependent - -Dio.netty.noUnsafe: false
10. 19:51:49.086 [main] DEBUG i.n.util.internal.PlatformDependent - sun.misc.Unsafe: available
11. 19:51:49.086 [main] DEBUG i.n.util.internal.PlatformDependent - -Dio.netty.noJavassist: false
12. 19:51:49.299 [main] DEBUG i.n.util.internal.PlatformDependent - Javassist: available
13. 19:51:49.300 [main] DEBUG i.n.util.internal.PlatformDependent - - Dio.netty.tmpdir: C:\Users\tannj\AppData\Local\Temp (java.io.tmpdir)
14. 19:51:49.300 [main] DEBUG i.n.util.internal.PlatformDependent - - Dio.netty.bitMode: 64 (sun.arch.data.model)
15. 19:51:49.300 [main] DEBUG i.n.util.internal.PlatformDependent - -Dio.netty.noPreferDirect: false
16. 19:51:49.338 [main] DEBUG io.netty.channel.nio.NioEventLoop - - Dio.netty.noKeySetOptimization: false
17. 19:51:49.338 [main] DEBUG io.netty.channel.nio.NioEventLoop - - Dio.netty.selectorAutoRebuildThreshold: 512
18. 19:51:49.617 [main] DEBUG i.n.util.internal.ThreadLocalRandom - - Dio.netty.initialSeedUniquifier: 0x51d30e47b2203417 (took 19 ms)
19. 19:51:49.685 [main] DEBUG io.netty.buffer.ByteBufUtil - -Dio.netty allocator.type: unpooled
20. 19:51:49.685 [main] DEBUG io.netty.buffer.ByteBufUtil - - Dio.netty.threadLocalDirectBufferSize: 65536
21. 19:51:49.728 [NettyClientSelector_1] DEBUG i.n.u.i.JavassistTypeParameterMatcherGenerator - Generated: io.netty.util.internal.__matchers__.com.alibaba.rocketmq.remoting.protocol.RemotingCommandMatcher
22. 19:51:49.779 [main] DEBUG io.netty.util.Recycler - - Dio.netty.recycler.maxCapacity.default: 262144
23. 19:51:49.811 [NettyClientWorkerThread_1] DEBUG io.netty.util.ResourceLeakDetector - - Dio.netty.leakDetectionLevel: simple
24. 19:51:50.034 [NettyClientSelector_1] DEBUG io.netty.util.internal.Cleaner0 - java.nio.ByteBuffer.cleaner(): available
25. Consumer Started.
26. ConsumeMessageThread_2 Receive New Messages: [MessageExt [queueId=0, storeSize=136, queueOffset=73578, sysFlag=0, bornTimestamp=1421236258344, bornHost=/10.1.111.150:53985, storeTimestamp=1421236258374, storeHost=/10.1.111.150:10911, msgId=0A016F9600002A9F000000000272A36A, commitLogOffset=4106737]
```

```
0, bodyCRC=613185359, reconsumeTimes=0, preparedTransactionOffset=0, toString()=Message [topic=Topi
cTest, flag=0, properties={TAGS=TagA, WAIT=true, MAX_OFFSET=73581, MIN_OFFSET=0}, body=16]]]
27. ConsumeMessageThread_1 Receive New Messages: [MessageExt [queueId=1, storeSize=136, queueOffset=735
77, sysFlag=0, bornTimestamp=1421236258376, bornHost=/10.1.111.150:53985, storeTimestamp=1421236258
377, storeHost=/10.1.111.150:10911, msgId=0A016F9600002A9F000000000272A3F2, commitLogOffset=4106750
6, bodyCRC=1401636825, reconsumeTimes=0, preparedTransactionOffset=0, toString()=Message [topic=Top
icTest, flag=0, properties={TAGS=TagA, WAIT=true, MAX_OFFSET=73580, MIN_OFFSET=0}, body=16]]]
28. ConsumeMessageThread_4 Receive New Messages: [MessageExt [queueId=1, storeSize=136, queueOffset=735
78, sysFlag=0, bornTimestamp=1421236258387, bornHost=/10.1.111.150:53985, storeTimestamp=1421236258
387, storeHost=/10.1.111.150:10911, msgId=0A016F9600002A9F000000000272A612, commitLogOffset=4106805
0, bodyCRC=1424393152, reconsumeTimes=0, preparedTransactionOffset=0, toString()=Message [topic=Top
icTest, flag=0, properties={TAGS=TagA, WAIT=true, MAX_OFFSET=73580, MIN_OFFSET=0}, body=16]]]
29. ConsumeMessageThread_7 Receive New Messages: [MessageExt [queueId=3, storeSize=136, queueOffset=735
76, sysFlag=0, bornTimestamp=1421236258382, bornHost=/10.1.111.150:53985, storeTimestamp=1421236258
383, storeHost=/10.1.111.150:10911, msgId=0A016F9600002A9F000000000272A502, commitLogOffset=4106777
8, bodyCRC=1032136437, reconsumeTimes=0, preparedTransactionOffset=0, toString()=Message [topic=Top
icTest, flag=0, properties={TAGS=TagA, WAIT=true, MAX_OFFSET=73578, MIN_OFFSET=0}, body=16]]]
30. ConsumeMessageThread_9 Receive New Messages: [MessageExt [queueId=2, storeSize=136, queueOffset=735
77, sysFlag=0, bornTimestamp=1421236258379, bornHost=/10.1.111.150:53985, storeTimestamp=1421236258
381, storeHost=/10.1.111.150:10911, msgId=0A016F9600002A9F000000000272A47A, commitLogOffset=4106764
2, bodyCRC=1250039395, reconsumeTimes=0, preparedTransactionOffset=0, toString()=Message [topic=Top
icTest, flag=0, properties={TAGS=TagA, WAIT=true, MAX_OFFSET=73579, MIN_OFFSET=0}, body=16]]]
31. ConsumeMessageThread_3 Receive New Messages: [MessageExt [queueId=0, storeSize=136, queueOffset=735
79, sysFlag=0, bornTimestamp=1421236258384, bornHost=/10.1.111.150:53985, storeTimestamp=1421236258
385, storeHost=/10.1.111.150:10911, msgId=0A016F9600002A9F000000000272A58A, commitLogOffset=4106791
4, bodyCRC=601994070, reconsumeTimes=0, preparedTransactionOffset=0, toString()=Message [topic=Topi
cTest, flag=0, properties={TAGS=TagA, WAIT=true, MAX_OFFSET=73581, MIN_OFFSET=0}, body=16]]]
32. ConsumeMessageThread_5 Receive New Messages: [MessageExt [queueId=0, storeSize=136, queueOffset=735
80, sysFlag=0, bornTimestamp=1421236258394, bornHost=/10.1.111.150:53985, storeTimestamp=1421236258
395, storeHost=/10.1.111.150:10911, msgId=0A016F9600002A9F000000000272A7AA, commitLogOffset=4106845
8, bodyCRC=710410109, reconsumeTimes=0, preparedTransactionOffset=0, toString()=Message [topic=Topi
cTest, flag=0, properties={TAGS=TagA, WAIT=true, MAX_OFFSET=73581, MIN_OFFSET=0}, body=16]]]
33. ConsumeMessageThread_8 Receive New Messages: [MessageExt [queueId=3, storeSize=136, queueOffset=735
77, sysFlag=0, bornTimestamp=1421236258392, bornHost=/10.1.111.150:53985, storeTimestamp=1421236258
393, storeHost=/10.1.111.150:10911, msgId=0A016F9600002A9F000000000272A722, commitLogOffset=4106832
2, bodyCRC=988340972, reconsumeTimes=0, preparedTransactionOffset=0, toString()=Message [topic=Topi
cTest, flag=0, properties={TAGS=TagA, WAIT=true, MAX_OFFSET=73578, MIN_OFFSET=0}, body=16]]]
34. ConsumeMessageThread_10 Receive New Messages: [MessageExt [queueId=2, storeSize=136, queueOffset=73
578, sysFlag=0, bornTimestamp=1421236258390, bornHost=/10.1.111.150:53985, storeTimestamp=142123625
8391, storeHost=/10.1.111.150:10911, msgId=0A016F9600002A9F000000000272A69A, commitLogOffset=410681
86, bodyCRC=1307562618, reconsumeTimes=0, preparedTransactionOffset=0, toString()=Message [topic=To
picTest, flag=0, properties={TAGS=TagA, WAIT=true, MAX_OFFSET=73579, MIN_OFFSET=0}, body=16]]]
35. ConsumeMessageThread_6 Receive New Messages: [MessageExt [queueId=1, storeSize=136, queueOffset=735
79, sysFlag=0, bornTimestamp=1421236258398, bornHost=/10.1.111.150:53985, storeTimestamp=1421236258
397, storeHost=/10.1.111.150:10911, msgId=0A016F9600002A9F000000000272A832, commitLogOffset=4106859
4, bodyCRC=1565577195, reconsumeTimes=0, preparedTransactionOffset=0, toString()=Message [topic=Top
icTest, flag=0, properties={TAGS=TagA, WAIT=true, MAX_OFFSET=73580, MIN_OFFSET=0}, body=16]]]
```

至此，已把RocketMQ在window平台下运行起来。

分类: RocketMQ

好文要顶

关注我

收藏该文

marco_tan
关注 - 0
粉丝 - 3
[+加关注](#)

0

1

« 上一篇: [linux命令之cat](#)

» 下一篇: [RocketMQ在linux平台下环境搭建](#)

posted @ 2015-01-14 19:03 marco_tan 阅读(723) 评论(3) 编辑 收藏

#1楼 2015-07-18 10:33 Leo_競哥

“你好，运行start mqnamesrv时出现windows版本不兼容怎么办？我的是win7 64位。

支持(1) 反对(0)

#2楼 2015-12-30 11:54 小野人~kobe

“请问在
进入刚生成的target文件夹下的bin目录，在命令行中执行：**start mqnamesrv.exe**，会弹出一个信息窗口
这一步中 bin目录下并没有mqnamesrv.exe 只有mqnamesrv， 我将其改
为.exe 但是报错 说不支持64位win7系统，这应该怎么办？
还有我在淘宝上看到
开放消息服务ONS(Open Notification Service)是基于阿里开源消息中间件MetaQ（RocketMQ）
目前支持Java,C++,.NET, PHP四种语言，Linux平台下支持C++,JAVA和PHP三种语言，Windows平台支持C++和.NET两种语言。W
indows不支持java,怎么用java测试？在windows系统中

支持(1) 反对(0)”

#3楼 2016-04-21 15:45 灵魂盗猎者

“执行完编译之后没有.exe文件阿，请楼主解答

支持(0) 反对(0)”

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库

【活动】优达学城正式发布“无人驾驶车工程师”课程

【推荐】移动直播百强八成都在用融云即时通讯云

【推荐】别再闷头写代码！找对工具，事半功倍，全能开发工具包用起来

【推荐】网易这个云产品做了15年才面世，1年吸引10万+开发者



广告

最新**IT**新闻：

- **Netflix**创始人：取代电视电影的可能是一种药物
- 美互联网集体宕机事件，我们该反思什么？
- 谷歌宣布暂停光纤互联网接入业务的扩张：裁员9%并撤换CEO
- 小米MIX、小米Note2双色图赏
- 天宫二号伴星回传首批高清图像 可看组合体全貌

» 更多新闻...

极光 智能推送全面升级 更快、更稳定、更成熟

了解更多

最新知识库文章：

- 循序渐进地代码重构
- 技术的正宗与野路子
- 陈皓：什么是工程师文化？
- 没那么难，谈CSS的设计模式
- 程序猿媳妇儿注意事项

» 更多知识库文章...