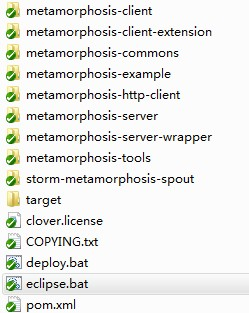
## 如何导入eclipse工程

前提：已经在自己的maven库install或deploy了通信库gecko

* 主工程。命令行进入工程根目录，输入mvn -U eclipse:eclipse或者运行eclipse.bat
* 其他独立工程。extension、example、http-client、tools、spout属于独立工程，因此需要进入相应的工程目录运行mvn -U eclipse:eclipse命令



## 打包

前提：已经在自己的maven库install或deploy了通信库gecko

主工程mvn -U -Dtest -DfailIfNoTests=false clean install package assembly:assembly，将在根目录的target目录下生成服务端所需要的全部东西。

其中taobao-metamorphosis-server-wrapper.dir就是服务端的目录。也可以使用不带扩展的taobao-metamorphosis-server.dir。

把里面的metamorphosis-server-wrapper拷出来就可以部署服务端了

其他独立工程进入相应目录下mvn -U -Dtest -DfailIfNoTests=false clean install package

如果需要发布到maven库就加deploy命令

## 服务器部署

### 前提

* **安装zookeeper**

首先你需要搭建自己的zookeeper集群，meta利用zookeeper做服务的注册和发现，以及默认情况下offset的存储。

* 安装java运行环境

### 第一步：下载安装包

从[这里](http://code.taobao.org/p/metamorphosis/file/2153/taobao-metamorphosis-server-wrapper.tar.gz)下载最新的metamorphosis服务器可运行包并在某个目录解压缩（或者checkout源码按上面的步骤打包出来）。解压出来的结构大概是这样

-你的目录

-taobao

-metamorphosis-server

-bin

-meta-server-start.sh

-meta-server-stop.sh

-conf

-server.properties

-logs

metaServer.log

-lib

其中bin目录包含启动脚本（目前只有linux下的启动脚本），conf下为配置文件，lib下为meta服务器的所有jar包,logs为日志所在目录。

### 第二步：配置server.ini

利用文本编辑器编辑conf/server.ini，这是meta服务器的配置文件，主要关注这几个配置项：

* brokerId 服务器ID，必须是集群内唯一
* numPartitions 默认每个topic的分区数目
* dataPath 数据文件的存放路径，默认在user.home/meta下
* zookeeper配置：

;以下为zk配置，可以为空，为空将从diamond获取（目前不提供），不为空则优先使用下列配置

zk.zkConnect=localhost:2181

;zk心跳超时，单位毫秒，默认30秒

zk.zkSessionTimeoutMs=30000

;zk连接超时时间，单位毫秒，默认30秒

zk.zkConnectionTimeoutMs=30000

;zk数据同步时间，单位毫秒，默认5秒

zk.zkSyncTimeMs=5000

zookeeper的地址也可以通过diamond管理，如果本地不明确配置zookeeper，则设置diamond的dataId和group即可自动从diamond获取zookeeper配置（目前不提供，要注释掉）：

;zk在diamond中配置存储的dataId

;diamondZKDataId=metamorphosis.zkConfig

;zk在diamond中配置存储的group

;diamondZKGroup=DEFAULT\_GROUP

配置topic列表

一份默认的文件如下：

;系统属性

[system]

;必须，服务器唯一标志

brokerId=0

;服务器hostname，可以为空，默认将取本机IP

hostName=

;默认每个topic的分区数目，默认为1

numPartitions=1

;服务器端口，必须

serverPort=8123

;数据文件路径，默认在user.home/meta下

dataPath=

;日志数据文件路径，默认跟dataPath一样

dataLogPath=

;最大允许的未flush消息数，超过此值将强制force到磁盘，默认1000

unflushThreshold=1000

;最大允许的未flush间隔时间，毫秒，默认10秒

unflushInterval=10000

;单个文件的最大大小,实际会超过此值，默认1G

maxSegmentSize=1073741824

;传输给客户端每次最大的缓冲区大小，默认1M

maxTransferSize=1048576

;处理get请求的线程数,默认cpus\*10

getProcessThreadCount=80

;处理put请求线程数，默认cpus\*10

putProcessThreadCount=80

;数据删除策略，默认超过7天即删除,这里的168是小时，10s表示10秒，10m表示10分钟，10h表示10小时，默认为小时

deletePolicy=delete,168

;删除策略的执行时间，cron表达式

deleteWhen=0 0 6,18 \* \* ?

;事务相关配置

;最大保存事务checkpoint数目，默认为3

maxCheckpoints=3

;事务checkpoint时间间隔，单位毫秒，默认1小时

checkpointInterval=3600000

;最大事务超时事件数，用于监控事务超时

maxTxTimeoutTimerCapacity=30000

;最大事务超时时间，单位秒

maxTxTimeoutInSeconds=60

;事务日志的刷盘设置，0表示让操作系统决定，1表示每次commit都刷盘，2表示每隔1秒刷盘一次

flushTxLogAtCommit=1

;zk配置

[zookeeper]

;是否注册到zk，默认为true

;zk.zkEnable=true

;以下为zk配置，可以为空，为空将从diamond获取，不为空则优先使用下列配置

;zk的服务器列表

zk.zkConnect=localhost:2181

;zk心跳超时，单位毫秒，默认30秒

zk.zkSessionTimeoutMs=30000

;zk连接超时时间，单位毫秒，默认30秒

zk.zkConnectionTimeoutMs=30000

;zk数据同步时间，单位毫秒，默认5秒

zk.zkSyncTimeMs=5000

;zk在diamond中配置存储的dataId

;diamondZKDataId=metamorphosis.zkConfig

;zk在diamond中配置存储的group

;diamondZKGroup=DEFAULT\_GROUP

;topic列表

[topic=test]

;是否启用统计

stat=true

;这个topic指定分区数目，如果没有设置，则使用系统设置

numPartitions=10

;topic的删除策略，默认使用系统策略

deletePolicy=

unflushInterval=

unflushThreshold=

;删除策略的执行时间，cron表达式

deleteWhen=0 0 6,18 \* \* ?

[topic=wuhua-test]

stat=true

numPartitions=10

unflushInterval=50

unflushThreshold=10

### 第三步：启动服务器

cd bin

sh meta-server-start.sh -f ../conf/server.properties

其中-f选项用于指定配置文件所在完整路径。 启动meta服务器后，你可以telnet到8123端口测试

telnet localhost 8123

stats

8123是meta服务器的默认端口，我们telnet上去并敲一个stats命令看看。敲quit命令可以退出telnet交互。

启动后，可以查看metaServer.log。

### 第四步：关闭服务器

关闭服务器通过meta-server-stop.sh脚本即可关闭

sh meta-server-stop.sh

## 其他

### 服务器的快速启动

Meta服务器在启动的时候会一个一个地校验所有文件，如果文件数目较多，那么这个启动过程会非常慢，如果想加快启动过程，可以使用fast\_boot选项，在meta-run-class.sh脚本里添加环境变量

META\_OPTS="-Xmx512m -server -Dcom.sun.management.jmxremote -Dlog4j.configuration=$base\_dir/bin/log4j.properties -Dmeta.fast\_boot=true"

将meta.fast\_boot设置为true即可跳过校验环节快速启动。

### 消息的删除策略

目前服务端支持两种删除策略：

* 定期删除，保存消息一定时间，超过指定时间就无条件删除。例如

deletePolicy=delete,72

的配置就是指使用删除策略，保存至少72个小时，超过即删除。

* 定期压缩归档，保存消息一定时间，超过指定时间就将消息压缩归档，例如

deletePolicy=archive,72

的配置就是指使用归档策略，保存至少72个小时，超过即归档。归档后的文件名前缀不变（也就是start offset)，后缀变为arc。归档策略还可以指定是否压缩：

deletePolicy=archive,72,true

第三个参数true指定归档策略使用压缩，meta使用zip压缩算法，压缩后的文件后缀即为zip，前缀不变。

### 不重启新增topic

新增一个topic或重新配置一些参数以后，运行tools里面的reloadconfig.sh，前提是已经部署了tools

Tools工程的打包部署：假设已经打包好了其他所有子工程，进入tools目录运行mvn -U -Dtest -DfailIfNoTests=false clean install package assembly:assembly，然后在tools的target目录里把跟服务端差不多结构的这个目录拷过去就可以了

或者直接使用jconsole的方式reload配置文件

### 如何启动异步复制slave

前提是使用metamorphosis-server-wrapper这个server。一个master可以挂上多个异步复制slave

1. 配置slave文件。编辑conf/async\_slave.properties

#异步复制的slave节点配置

#有问题联系无花 ([wq163@163.com](mailto:wq163@163.com))

#slave编号,大于等于0表示作为slave启动,同一个master下的slave编号应该设不同值.

slaveId=0

#作为slave启动时向master订阅消息的group,如果没配置则默认为meta-slave-group

#建议使用默认值

slaveGroup=meta-slave-group

#slave数据同步的最大延时,单位毫秒

slaveMaxDelayInMills=500

1. 确保conf/server.ini这个文件的brokerId跟master相同，这个表示作为哪个master的slave
2. 启动slave

sh meta-server-start.sh -f ../conf/server.properties -Fmetaslave=../conf/async\_slave.properties

**如何启用同步复制**

前提是使用metamorphosis-server-wrapper这个server。同步复制的master和slave是一对一的关系

1. 进入slave机器，确保conf/ gregor\_slave.properties文件的存在，不需要配置
2. 启动同步复制slave

sh meta-server-start.sh -f ../conf/server.properties -Fgregor =../conf/gregor\_slave.propertie

1. 进入master机器，配置conf/samsa\_master.properties

#slave节点ip，配置异步复制slave的地址

slave=localhost:8124

#应答回调线程池大小，默认3\*cpus

callbackThreadCount=10

#是否recover offset,只有在slave作为master启动的时候需要设置为true

recoverOffset=false

1. 启动同步复制master

sh meta-server-start.sh -f ../conf/server.properties -Fsamsa =../conf/samsa\_master.properties