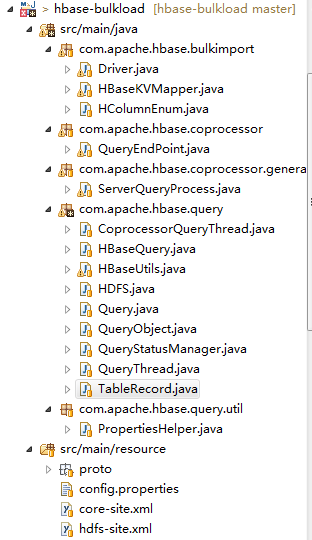
HBase查询API接口文档

# 工程结构

HBase批导入库程序



HBase 查询API所在的包

HBase Coprocessor程序，主要是为了支撑模糊查询，大批量数据范围，提高查询效率

## config.properties文件说明

##zookeeper的主机名

hbase.zookeeper.quorum=n11,n2,n10

##zookeeper服务的端口，如果没有改动就是默认的2181

hbase.zookeeper.property.clientPort=2181

##hbase在zookeeper上注册的节点名称，如果没有改动，默认就是下面的值

zookeeper.znode.parent=hbase-unsecure

##zip文件在hdfs上的路径前缀，这里我们约定所有数据在hdfs上放置的根目录就是/data\_hbase

hdfs.prefix=hdfs://julong/data\_hbase/zip/

##HBase表的前缀，这里我们预定一个所有HBase表都是以某个前缀开始的

hbas.table.prefix=FSN\_

## hdfs-site.xml及core-site.xml

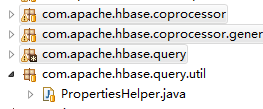
将Hadoop环境中的core-site.xml和hdfs-site.xml文件放到自己的classpath中

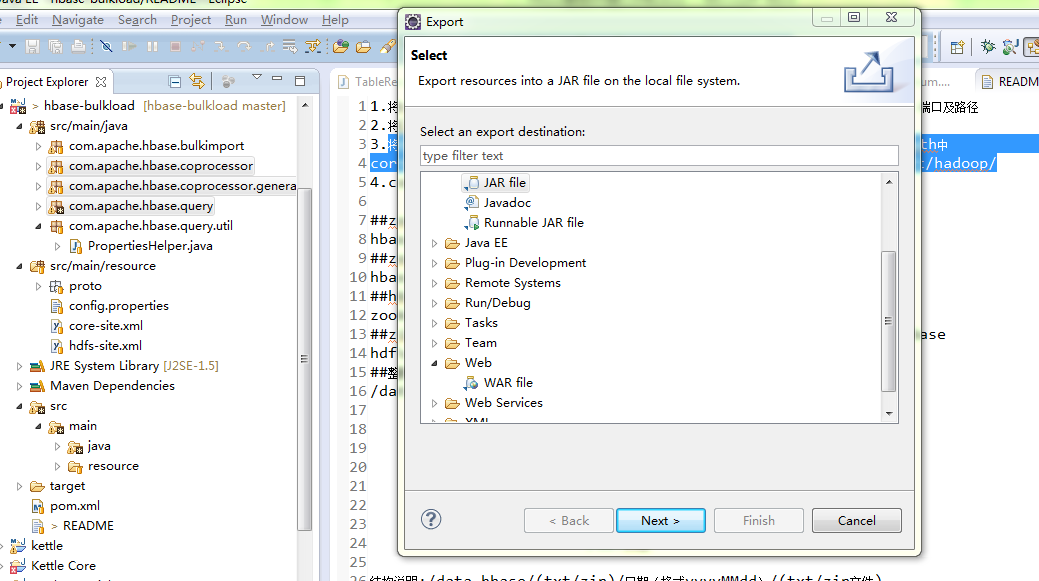
core-site.xml和hdfs-site.xml在hadoop环境中的路径/usr/lib/hadoop/etc/hadoop/

# 打包部署

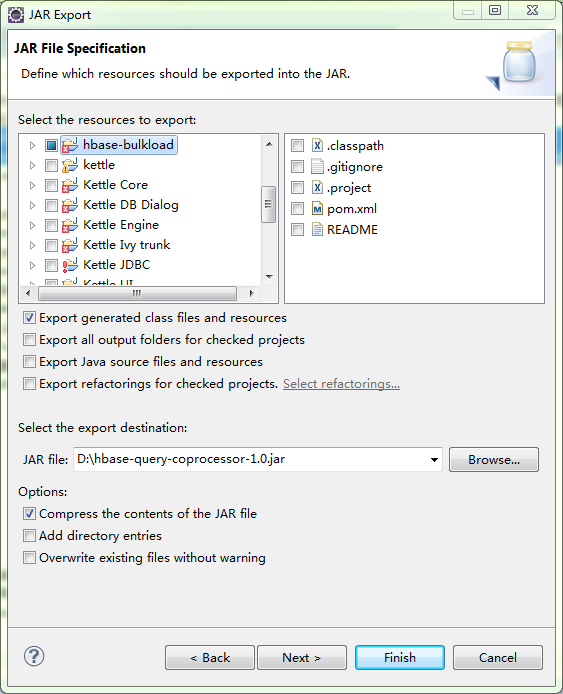
## Coprocessor打包部署

选中下面的四个包，单击右键选择export->JAR file导出jar





下一步：

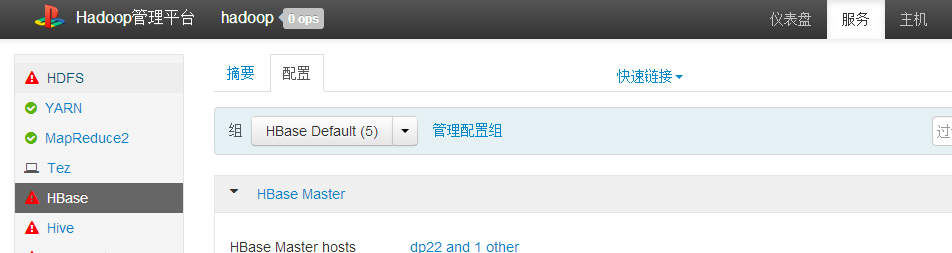


点击完成即可

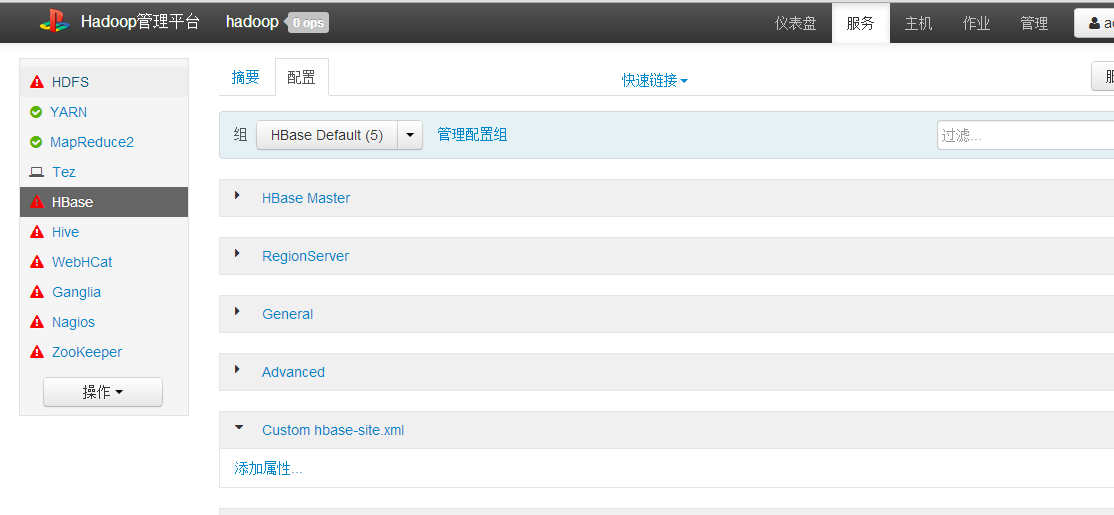
## 部署自定义coprocessor

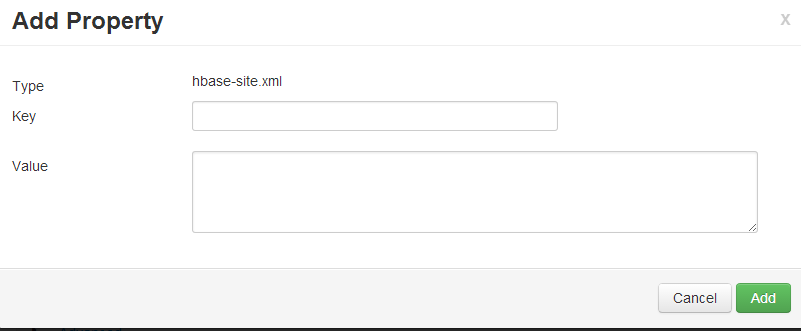
首先将hbase-query-coprocessor-1.0.jar 拷贝到每一台机子上的/usr/lib/hbase/lib目录下

然后配置Hbase Coprocessor



选择Custom hbase-site.xml文件，添加添加属性





在这里key填入: hbase.coprocessor.region.classes

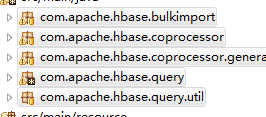
Value：com.apache.hbase.coprocessor.QueryEndPoint

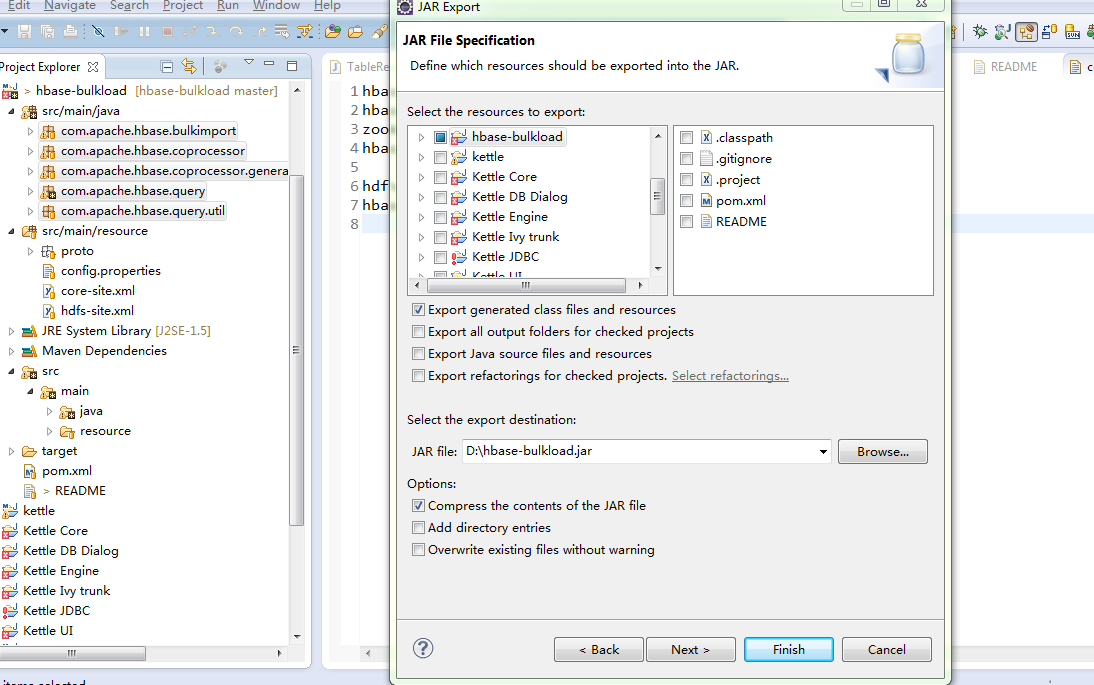
最后保存Hbase 配置。在选择重启所有操作

**备注：客户端查询使用的API和Coprocessor使用的是同一个jar包，为了避免引起误会，将jar该名称hbase-query-api-1.0.jar即可**

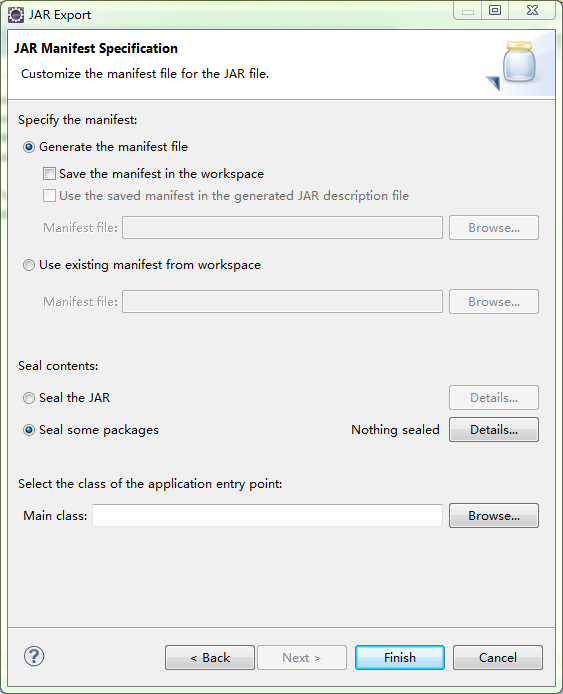
## HBase批导程序的打包

选择下面的包，单击右键选择export->JAR file导出jar

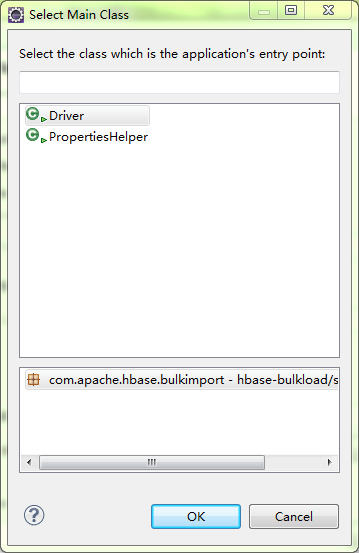




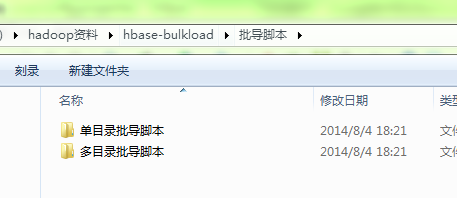
下一步，直到出现下面界面



单击Browse，选择Driver，点击完成即可



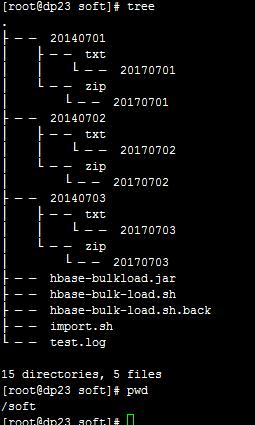
# HBase数据导入



有两个脚本：

* 一个是可以支持多个目录连续导入的，目录结构如下所示：

这个导入脚本支持对指定目录下的多个目录自动导入，自动创建HBase表，表名就是指定的表前缀+下划线+指定目录下的一级目录名（以日期命名），要求每个日期目录下面必须存在txt和zip两个目录，这个两个目录下面分别是日期（和前面那个日期一致）

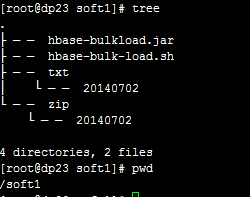


然后执行./hbase-bulk-load.sh <数据所在目录>

例如：./hbase-bulk-load.sh /soft

* 单目录导入，目录结构如下

这个只支持单次导入，要求指定目录下必须存在txt和zip两个目录，这个两个目录下面分别是日期（zip和txt目录下的日期要一致），这个脚本要指定表名，脚本会判断表是否存在，如果不存在，会自动创建表。



执行命令：

./hbase-bulk-load.sh <表名> <数据所在目录>

例如：./hbase-bulk-load.sh FSN\_20140728 /soft1

# API使用

## 查询API（HBaseQuery.java）

查询对外只提供一个接口，用户只需要将查询api的jar包引入到自己的工程中，同时将config.properties,hdfs-site.xml,core-site.xml（这三个文件的说明见本文档第一部分）文件放到自己的classpath中，按照下面的示例编写自己的查询程序即可。返回的结果是List<String>

示例：

HBaseQuery query = **new** HBaseQuery();

String start = "2014-07-01 00:00:00";

String end = "2014-07-12 23:59:29";

String gzh = "ZAAG781194";//冠字号至少要输入前四位字母

QueryObject object = **new** QueryObject();

object.setEnd(end);//结束时间

object.setGzh(gzh);//冠字号

object.setFr("01");//法人

object.setQy("00");//区域

object.setCzr("");//操作人

object.setSbbm("");//设备编码

object.setWd("");//网点

object.setStart(start);//开始时间

Date t1 = **new** Date();

List<String> list = query.query(object);//调用查询接口获取结果

**for**(String record : list ){

System.*out*.println(record);

}

## 查询表的记录总数（HBaseQuery.java）

该API接口主要提供根据表名查询该表中的记录总数，及异常记录数据，目前提供三种记录数：正确记录数、分隔符错误记录数、字段数错误记录数。

示例：

HBaseQuery query = **new** HBaseQuery();

TableRecord record = query.getRecordForTable("FSN\_20140728");

**if**(record != **null**){

System.*out*.println("分隔符错误记录数： "+record.getErrorRecord());

System.*out*.println("字段数不正确记录数： "+record.getInvalRecord());

System.*out*.println("正确记录总数数： "+record.getTotalRecord());

}

## 删除表（HBaseUtils.java）

该接口提供对表删除操作，删除表的同时删除该表数据对应的zip文件，同时删除该表对应的就总数记录数据。参数：日期（到天，格式yyyyMMdd）

示例：

HBaseUtils.*getInstance*().deleteTable("20140728");

## HDFS文件读取（HDFS.java）

这个API（HDFS.java）读取hdfs上的文件，并保存到本地。

//这里是你的要在的HDFS上的额文件路径

String hdfsPath = "hdfs://julong/data/0028\_0.txt";

String localPath = "d:\\0028\_0.txt";

*downFileToLocal*(hdfsPath,localPath);