四、寫Code 環境準備

4.A : Console 端
編譯與執行

4.B : 透過 Eclipse 開發與測試運算

Java 之編譯與執行

1. 編譯
   - javac △ -classpath △ hadoop-*-core.jar △ -d △ MyJava △ MyCode.java
2. 封裝
   - jar △ -cvf △ MyJar.jar △ -C △ MyJava △ .
3. 執行
   - bin/hadoop △ jar △ MyJar.jar △ MyCode △ HDFS_Input/ △ HDFS_Output/

- 所在的執行目錄為Hadoop_Home
- ./MyJava = 編譯後程式碼目錄
- My jar. jar = 封裝後的編譯檔
- 先放些文件檔到HDFS上的input目錄
- ./input; ./ouput = hdfs的輸入、輸出目錄

WordCount1 練習 (I)

1. cd $HADOOP_HOME; mkdir input_local
2. echo "I like NCHC Cloud Course." > input_local/input1
3. echo "I like nchc Cloud Course, and we enjoy this crouse." > input_local/input2
4. bin/hadoop dfs -put input_local input
5. bin/hadoop dfs -ls input

```
waue@vPro:/opt/hadoop$ bin/hadoop dfs -ls input
Found 2 items
-rw-r--r--   1 waue supergroup         26 2009-03-22 12:15 /user/waue/input/input1
-rw-r--r--   1 waue supergroup         52 2009-03-22 12:15 /user/waue/input/input2
waue@vPro:/opt/hadoop$
```

WordCount1 練習 (II)

1. 編輯WordCount.java
   http://trac.nchc.org.tw/cloud/attachment/wiki/jazz/Hadoop_Lab6/WordCount.java?format=raw
2. mkdir MyJava
3. javac -classpath hadoop-*-core.jar -d MyJava WordCount.java
4. jar -cvf wordcount.jar -C MyJava .
5. bin/hadoop jar wordcount.jar WordCount input/ output/

- 所在的執行目錄為Hadoop_Home（因為hadoop-*-core.jar）
- javac編譯時需要classpath, 但hadoop jar時不用
- wordcount. jar = 封裝後的編譯檔，但執行時需告知class name
- Hadoop進行運算時，只有 input 檔要放到hdfs上，以便hadoop分析運算，執行檔（wordcount. jar）不需上傳，也不需每個node都放，程式的載入交由java處理

WordCount1 練習(III)

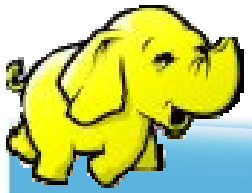
```
waue@vPro:/opt/hadoop$ bin/hadoop dfs -put input input
waue@vPro:/opt/hadoop$ mkdir MyJava
waue@vPro:/opt/hadoop$ javac -classpath hadoop-*-core.jar -d MyJava WordCount.java
waue@vPro:/opt/hadoop$ jar -cvf wordcount.jar -C MyJava .
新增 manifest
新增：WordCount.class (讀=1516)(寫=740)(壓縮 51%)
新增：WordCount$Reduce.class (讀=1591)(寫=642)(壓縮 59%)
新增：WordCount$Map.class (讀=1918)(寫=795)(壓縮 58%)
waue@vPro:/opt/hadoop$ bin/hadoop jar wordcount.jar WordCount input/ output/
09/03/22 11:39:01 WARN mapred.JobClient: Use GenericOptionsParser for parsing the argu
ments. Applications should implement Tool for the same.
09/03/22 11:39:01 INFO mapred.FileInputFormat: Total input paths to process : 1
09/03/22 11:39:01 INFO mapred.FileInputFormat: Total input paths to process : 1
09/03/22 11:39:02 INFO mapred.JobClient: Running job: job_200903201526_0007
09/03/22 11:39:03 INFO mapred.JobClient:  map 0% reduce 0%
09/03/22 11:39:08 INFO mapred.JobClient:  map 100% reduce 0%
09/03/22 11:39:15 INFO mapred.JobClient: Job complete: job_200903201526_0007
09/03/22 11:39:15 INFO mapred.JobClient: Counters: 16
09/03/22 11:39:15 INFO mapred.JobClient:   File Systems
09/03/22 11:39:15 INFO mapred.JobClient:     HDFS bytes read=320950
09/03/22 11:39:15 INFO mapred.JobClient:     HDFS bytes written=130568
09/03/22 11:39:15 INFO mapred.JobClient:     Local bytes read=168448
09/03/22 11:39:15 INFO mapred.JobClient:     Local bytes written=336932
09/03/22 11:39:15 INFO mapred.JobClient:   Job Counters
09/03/22 11:39:15 INFO mapred.JobClient:     Launched reduce tasks=1
09/03/22 11:39:15 INFO mapred.JobClient:     Launched map tasks=1
09/03/22 11:39:15 INFO mapred.JobClient:     Data-local map tasks=1
09/03/22 11:39:15 INFO mapred.JobClient:   Map-Reduce Framework
09/03/22 11:39:15 INFO mapred.JobClient:     Reduce input groups=9284
09/03/22 11:39:15 INFO mapred.JobClient:     Combine output records=18568
09/03/22 11:39:15 INFO mapred.JobClient:     Map input records=7868
09/03/22 11:39:15 INFO mapred.JobClient:     Reduce output records=9284
09/03/22 11:39:15 INFO mapred.JobClient:     Map output bytes=445846
09/03/22 11:39:15 INFO mapred.JobClient:     Map input bytes=320950
09/03/22 11:39:15 INFO mapred.JobClient:     Combine input records=47227
09/03/22 11:39:15 INFO mapred.JobClient:     Map output records=37943
09/03/22 11:39:15 INFO mapred.JobClient:     Reduce input records=9284
waue@vPro:/opt/hadoop$
```

WordCount1 練習(IV)

```
waue@vPro:/opt/hadoop$ bin/hadoop dfs -cat output/part-00000
Cloud   2
Course, 1
Course. 1
I       2
NCHC    1
and     1
course. 1
enjoy   1
like    2
nchc    1
this    1
we      1
```

BTW …

- 雖然Hadoop框架是用Java實作，但Map/Reduce應用程序則不一定要用 Java來寫
- Hadoop Streaming ：
  - 執行作業的工具，使用者可以用其他語言（如：PHP）套用到Hadoop的mapper和reducer
- Hadoop Pipes：C++ API



四、寫Code 環境準備

4.A : Console 端編譯與執行

4.B：透過 Eclipse 開發與測試運算

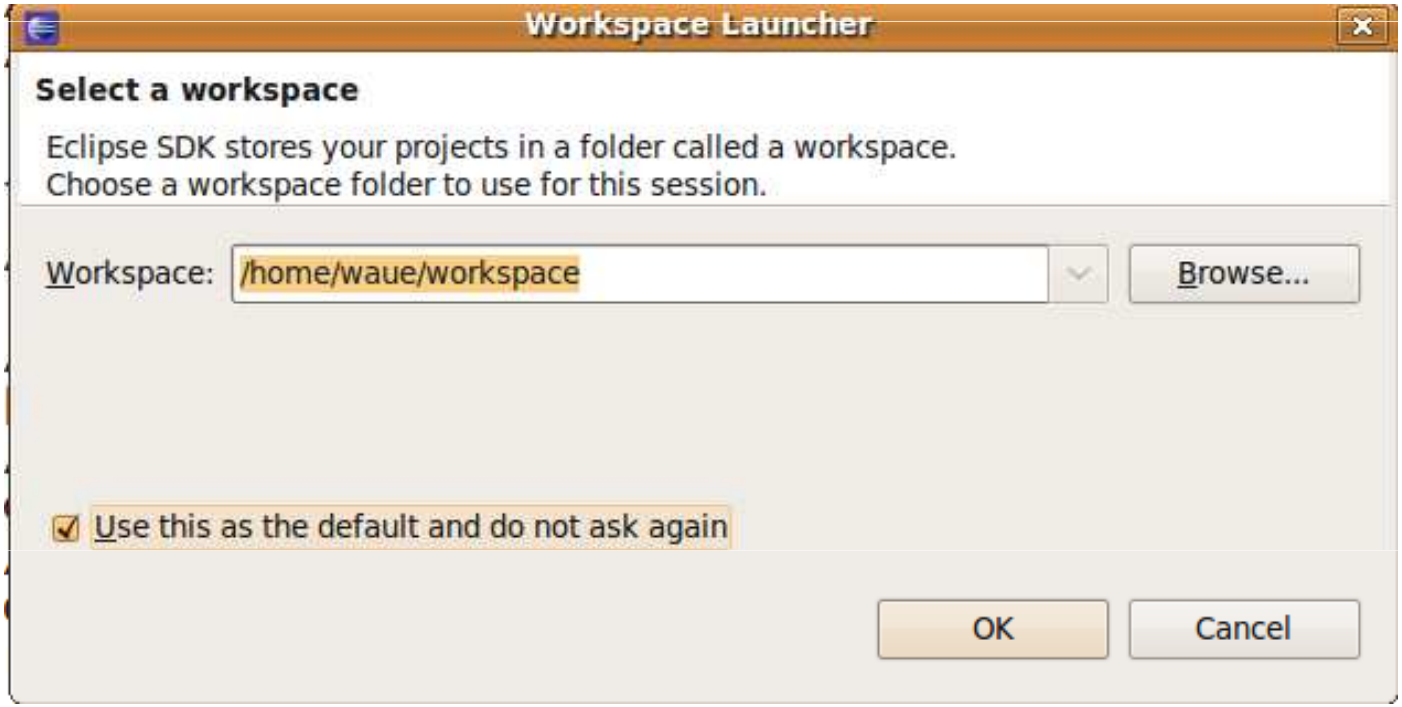
Requirements

- Hadoop 0.20.0 up
- Java 1.6
- Eclipse 3.3 up
- Hadoop Eclipse Plugin 0.20.0 up

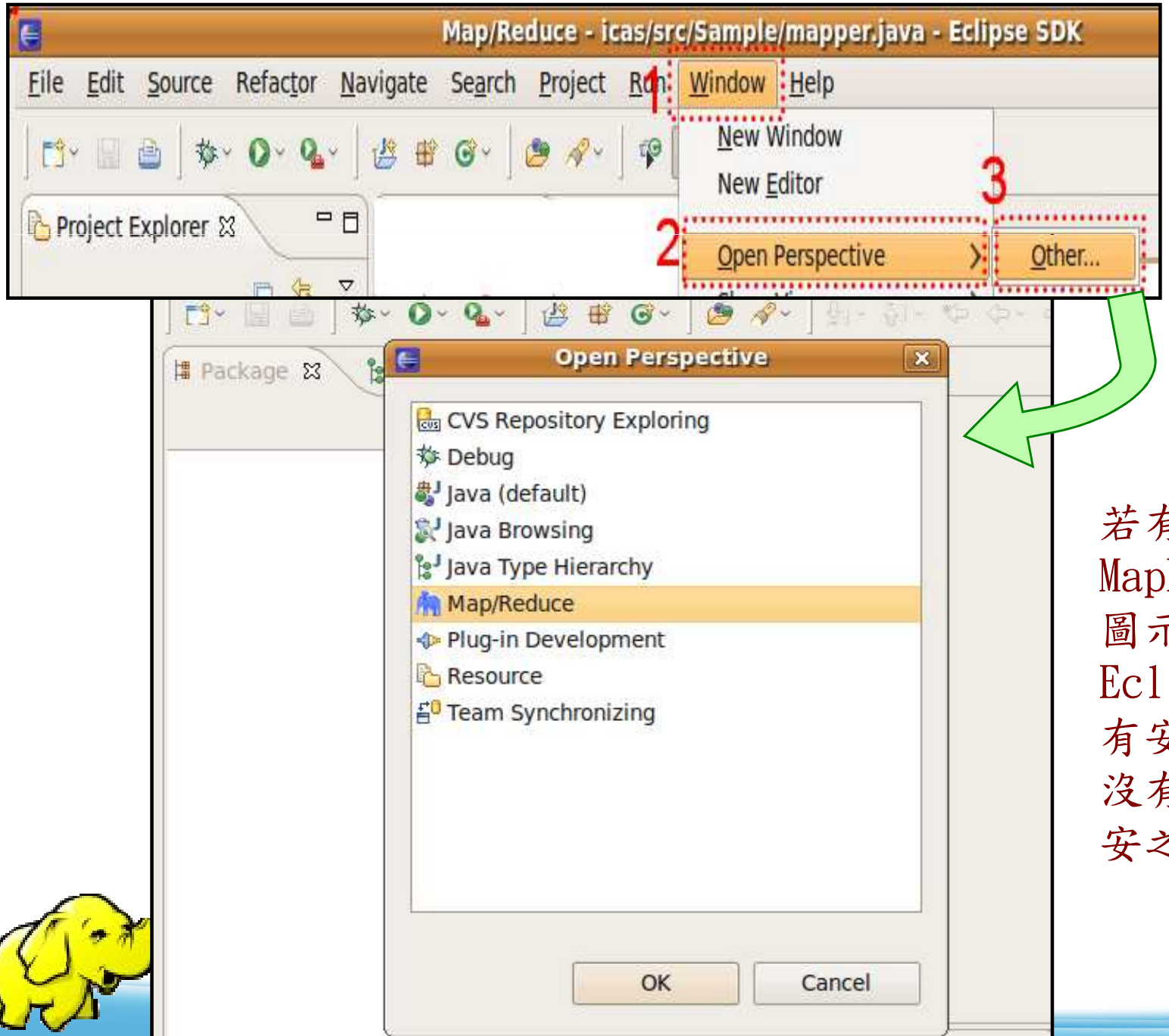
安裝Hadoop Eclipse Plugin

- Hadoop Eclipse Plugin 0.20.0
  - From $Hadoop_0.20.0_home/contrib/eclipse-plugin/hadoop-0.20.0-eclipse-plugin.jar
- Hadoop Eclipse Plugin 0.20.1
  - Compiler needed
  - Or download from http://hadoop-eclipse-plugin.googlecode.com/files/hadoop-0.20.1-eclipse-plugin.jar
- copy to $Eclipse_home/plugins/

1 打開Eclipse, 設定專案目錄



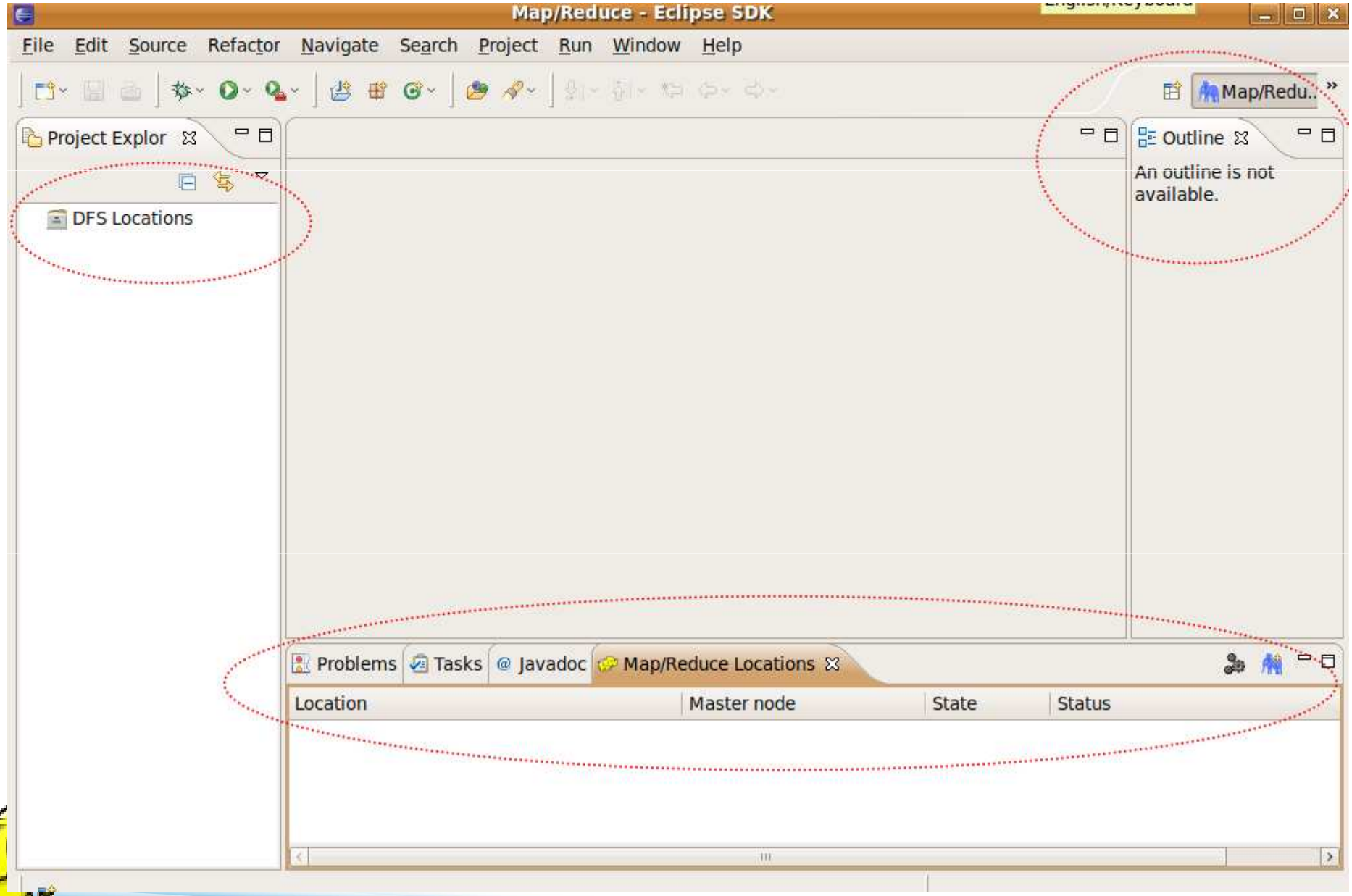
2. 使用Hadoop mode視野



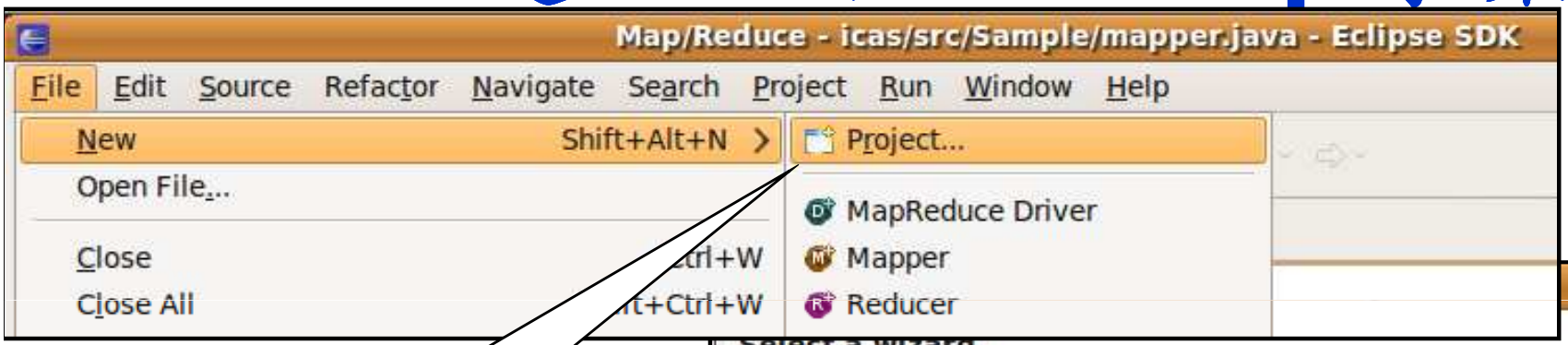
1. Window
2. Open Perspective
3. Other

若有看到MapReduce的大象圖示代表Hadoop Eclipse plugin有安裝成功， 若沒有請檢查是否有安之裝正確

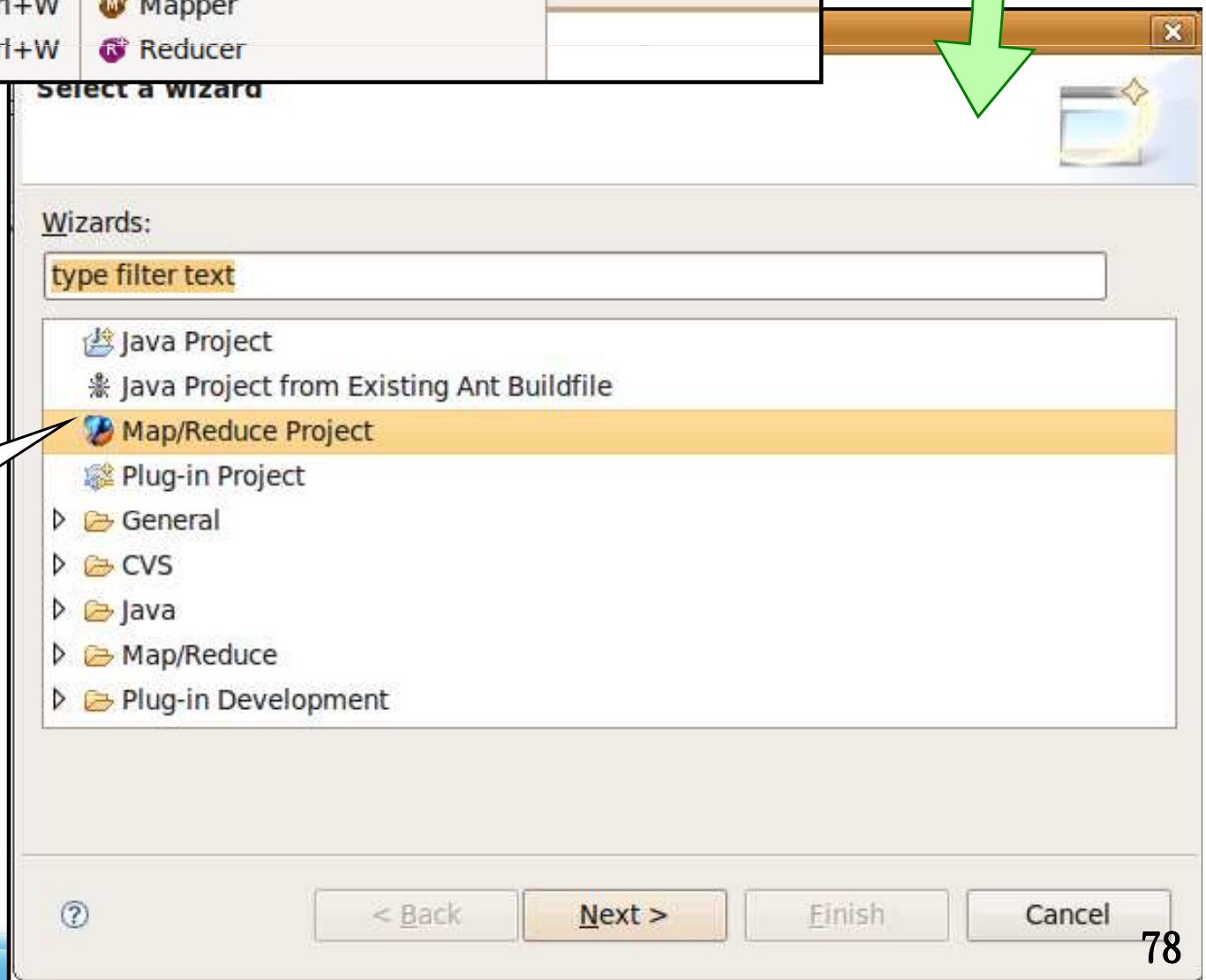
3. 使用Hadoop視野，主畫面將出現三個功能



4.建立一個Hadoop專案

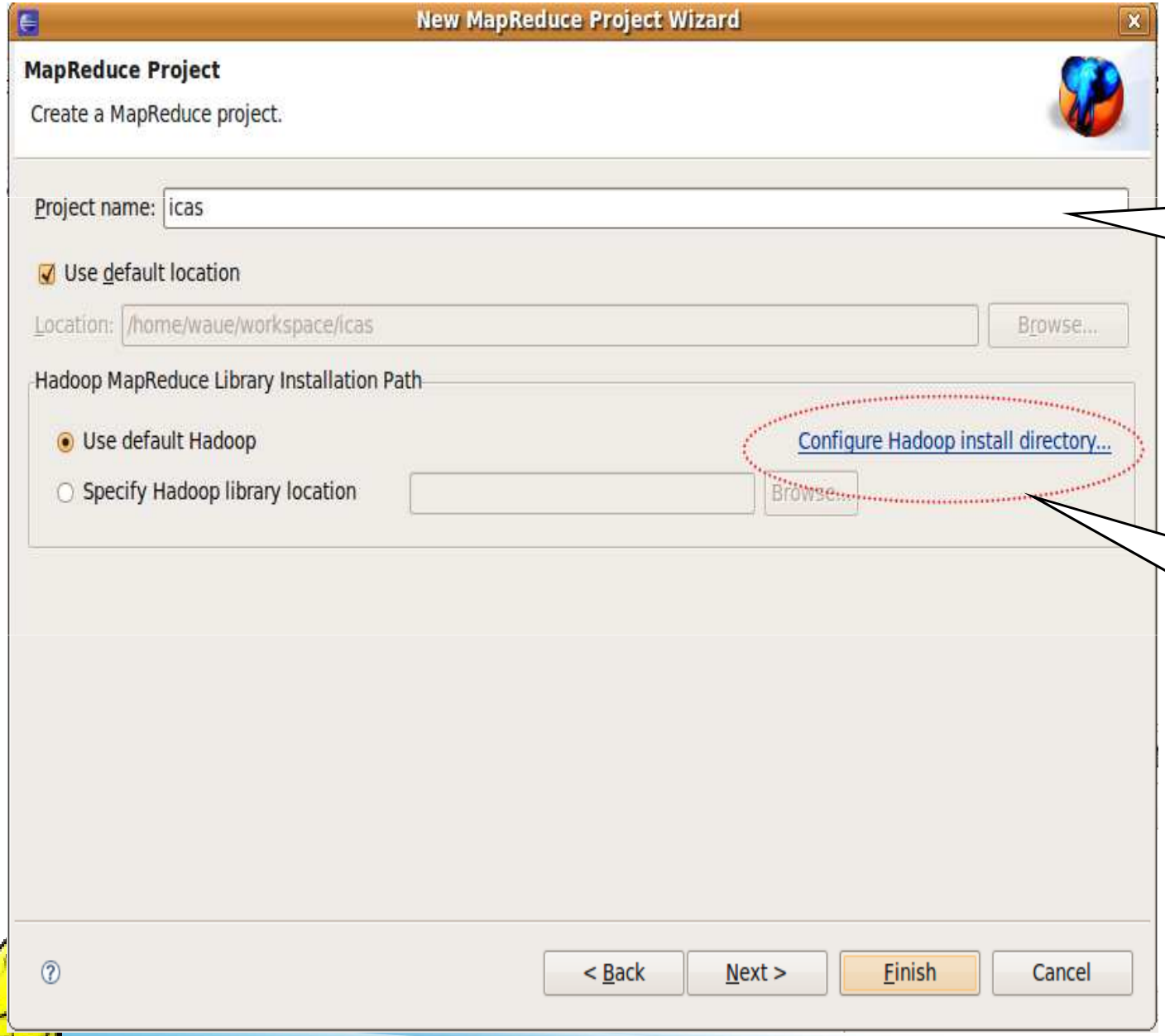


開出新專案



選擇Map/Reduce專案

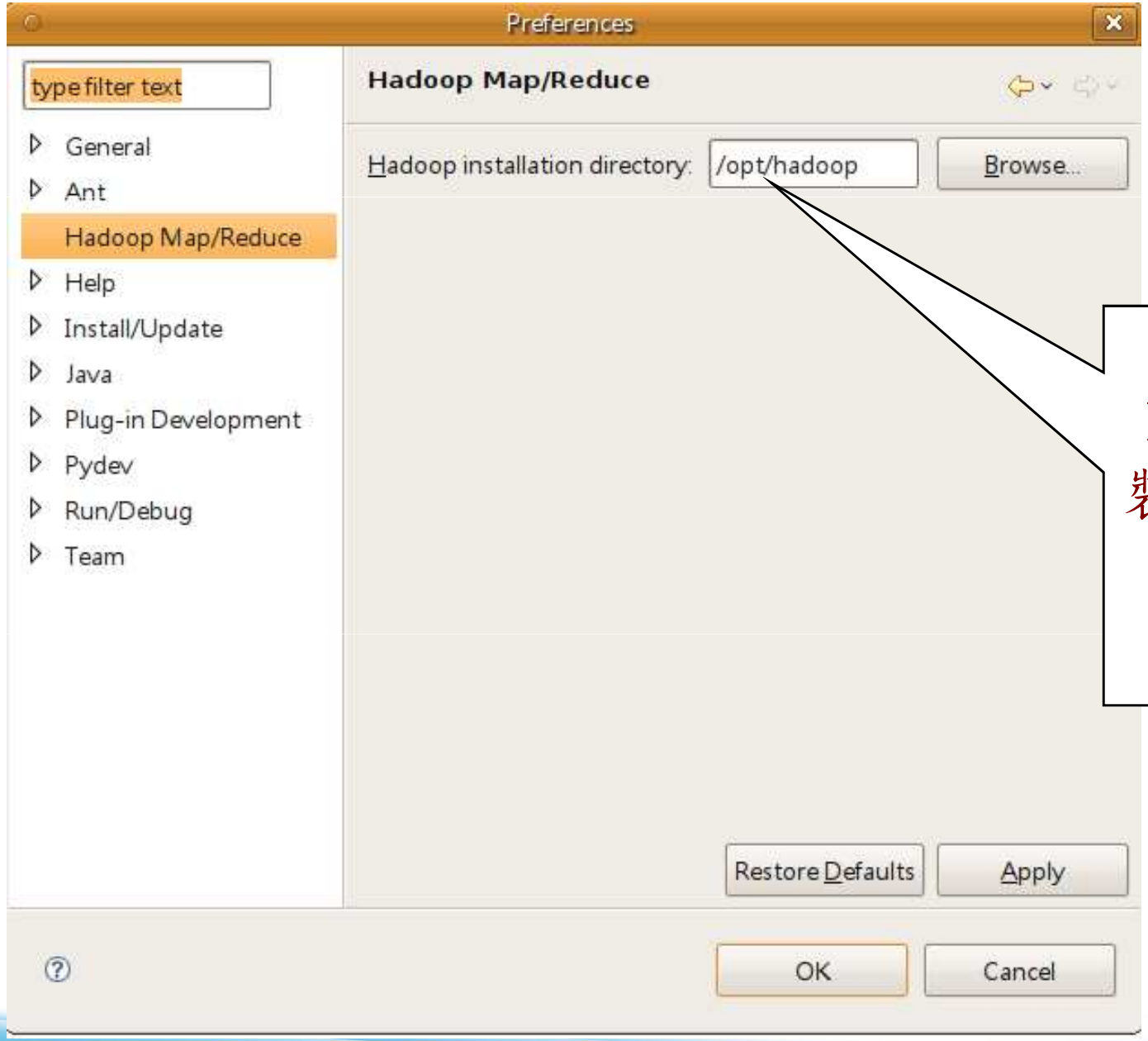
4-1. 輸入專案名稱並點選設定Hadoop安裝路徑



由此設定專案名稱

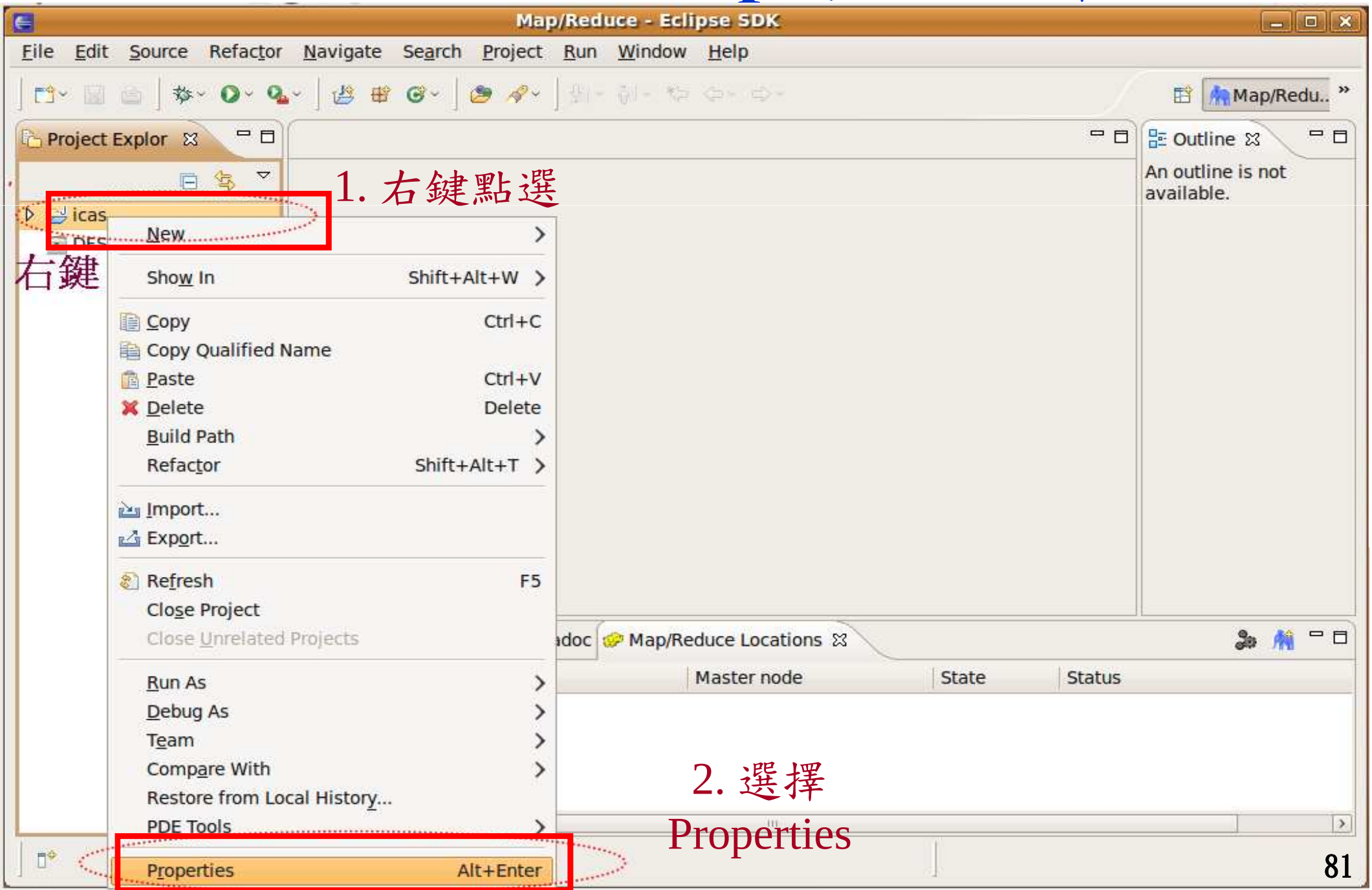
由此設定Hadoop的安裝路徑

4-1-1. 填入Hadoop安裝路徑



於此輸入您Hadoop的安裝路徑，之後選擇 ok

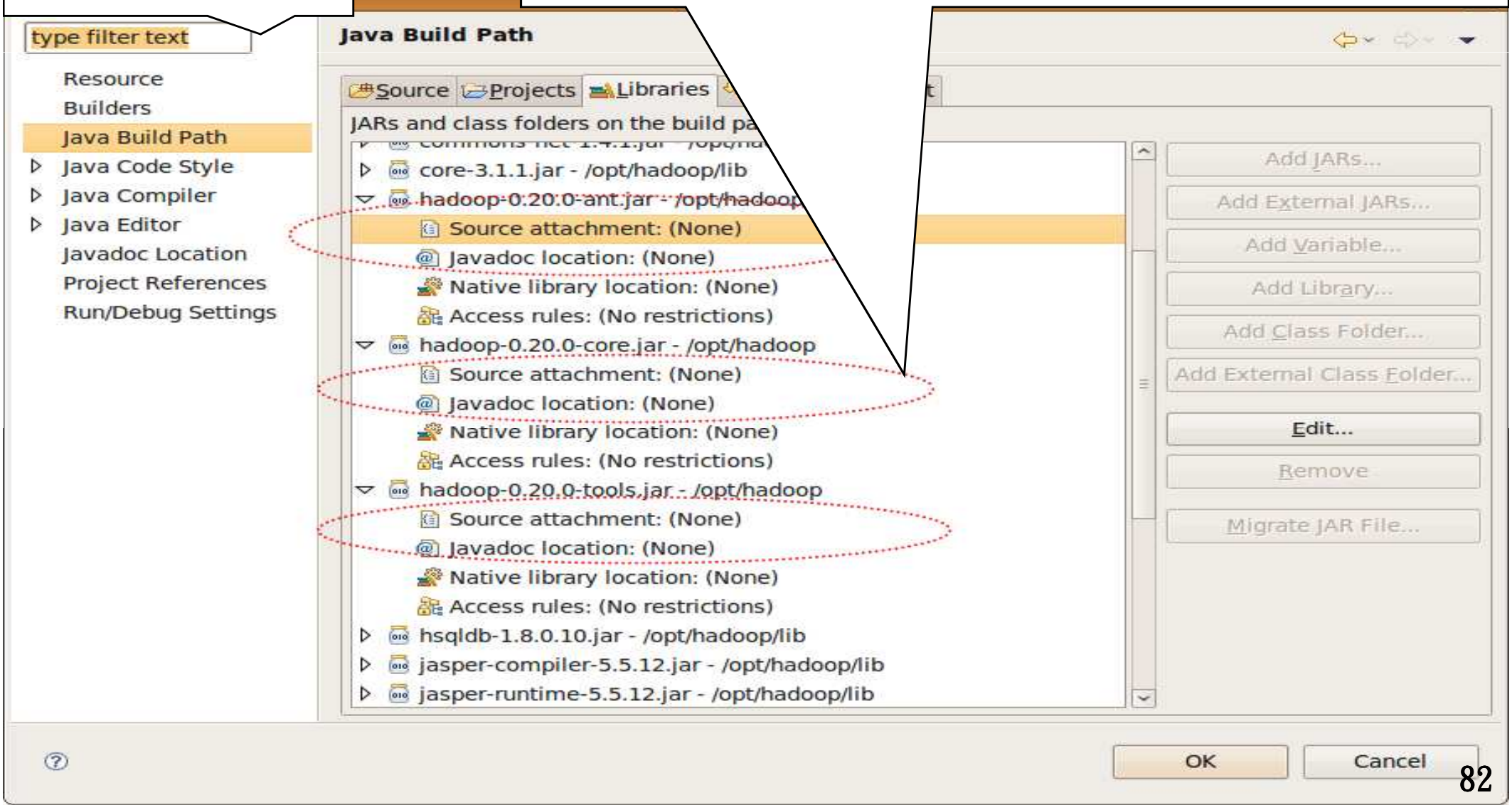
5. 設定Hadoop專案細節



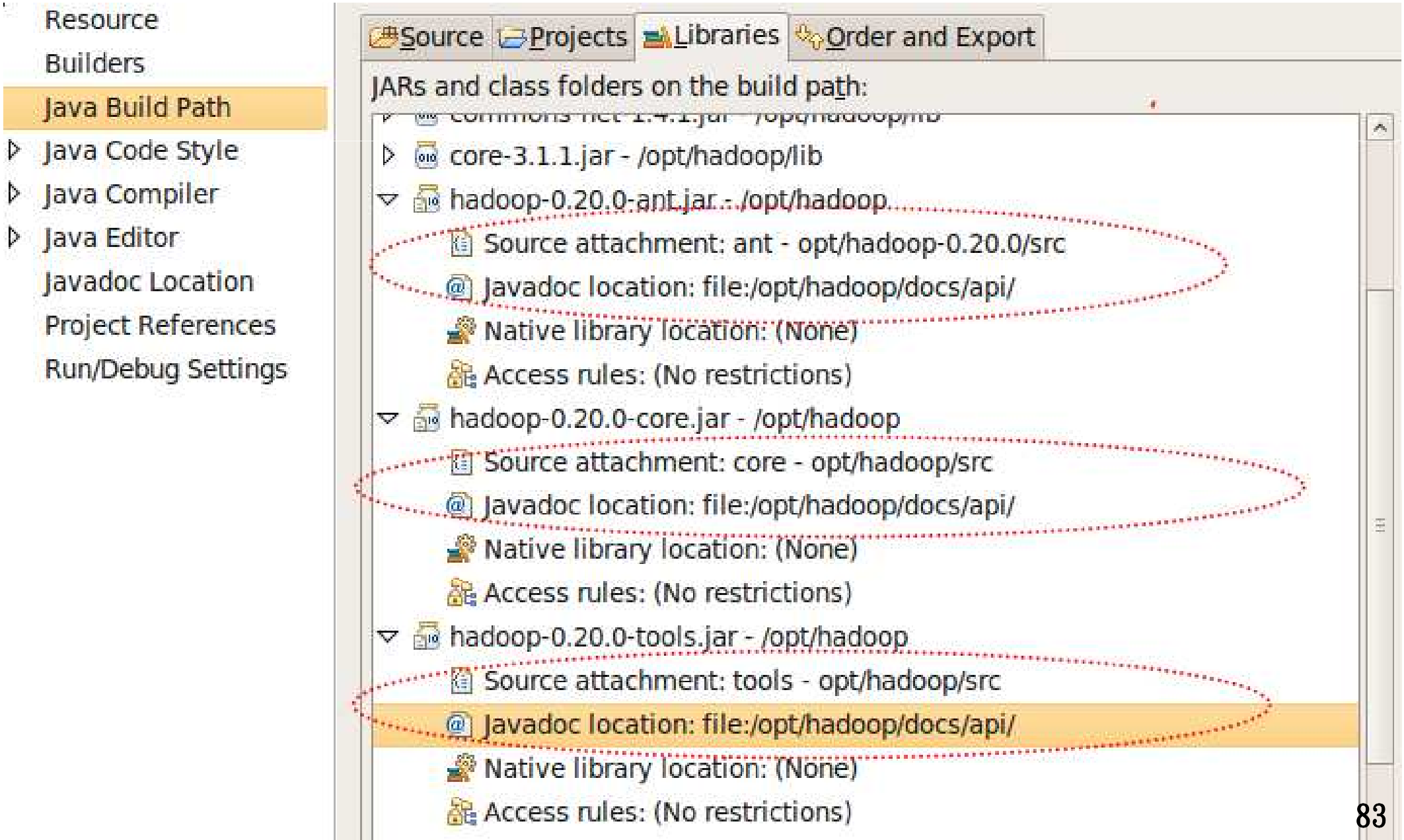
5-1. 設定原始碼與文件路徑

選擇 Java Build Path

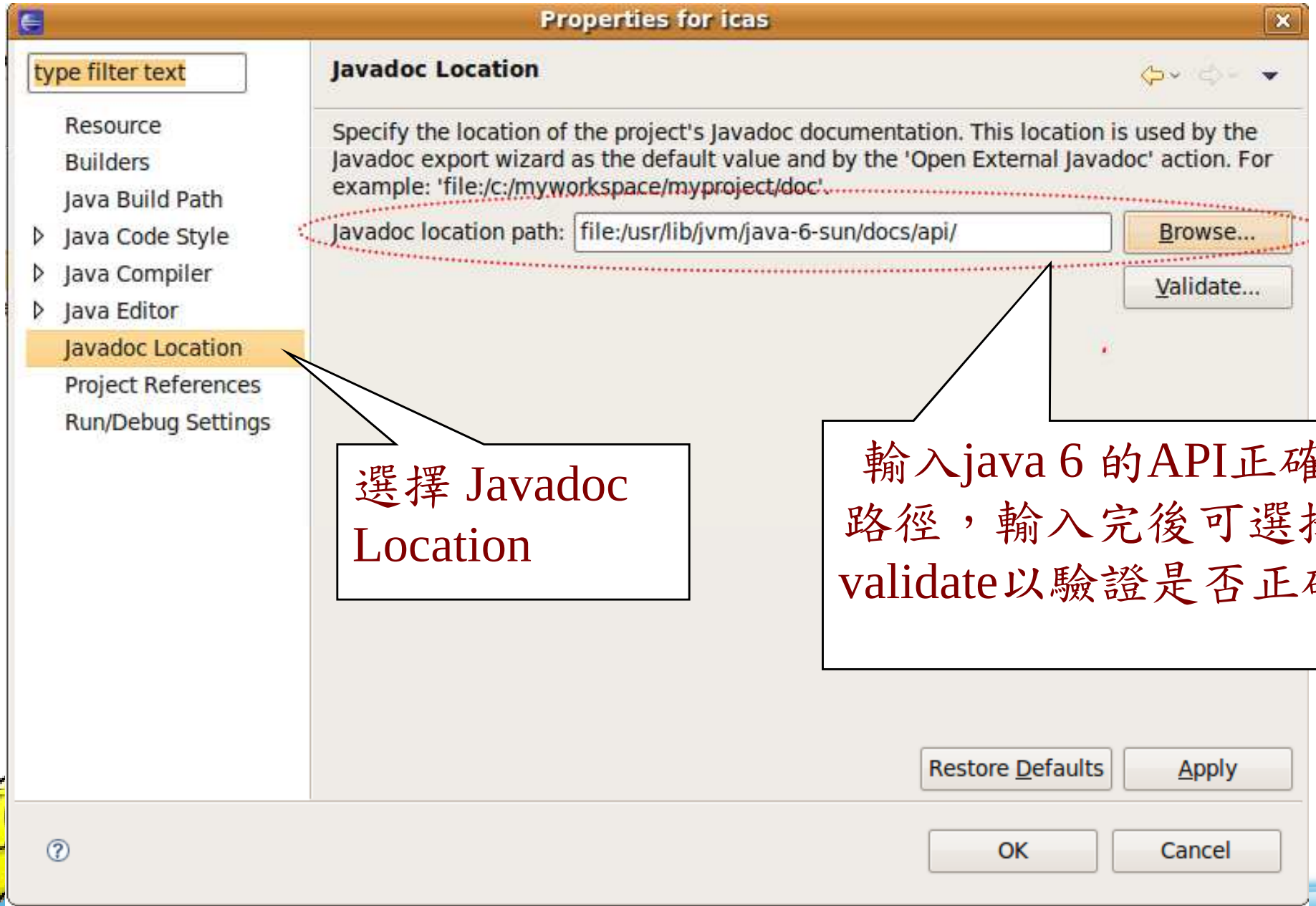
以下請輸入正確的Hadoop原始碼與API文件檔路徑，如
source ：/opt/hadoop/src/
javadoc：file:/opt/hadoop/docs/api/



5-1-1. 完成圖


Resource
Builders
Java Build Path
Java Code Style
Java Compiler
Java Editor
Javadoc Location
Project References
Run/Debug Settings
Source
Projects
Libraries
Order and Export
JARs and class folders on the build path:
core-3.1.1.jar - /opt/hadoop/lib
hadoop-0.20.0-ant.jar - /opt/hadoop
Source attachment: ant - opt/hadoop-0.20.0/src
Javadoc location: file:/opt/hadoop/docs/api/
Native library location: (None)
Access rules: (No restrictions)
hadoop-0.20.0-core.jar - /opt/hadoop
Source attachment: core - opt/hadoop/src
Javadoc location: file:/opt/hadoop/docs/api/
Native library location: (None)
Access rules: (No restrictions)
hadoop-0.20.0-tools.jar - /opt/hadoop
Source attachment: tools - opt/hadoop/src
Javadoc location: file:/opt/hadoop/docs/api/
Native library location: (None)
Access rules: (No restrictions)

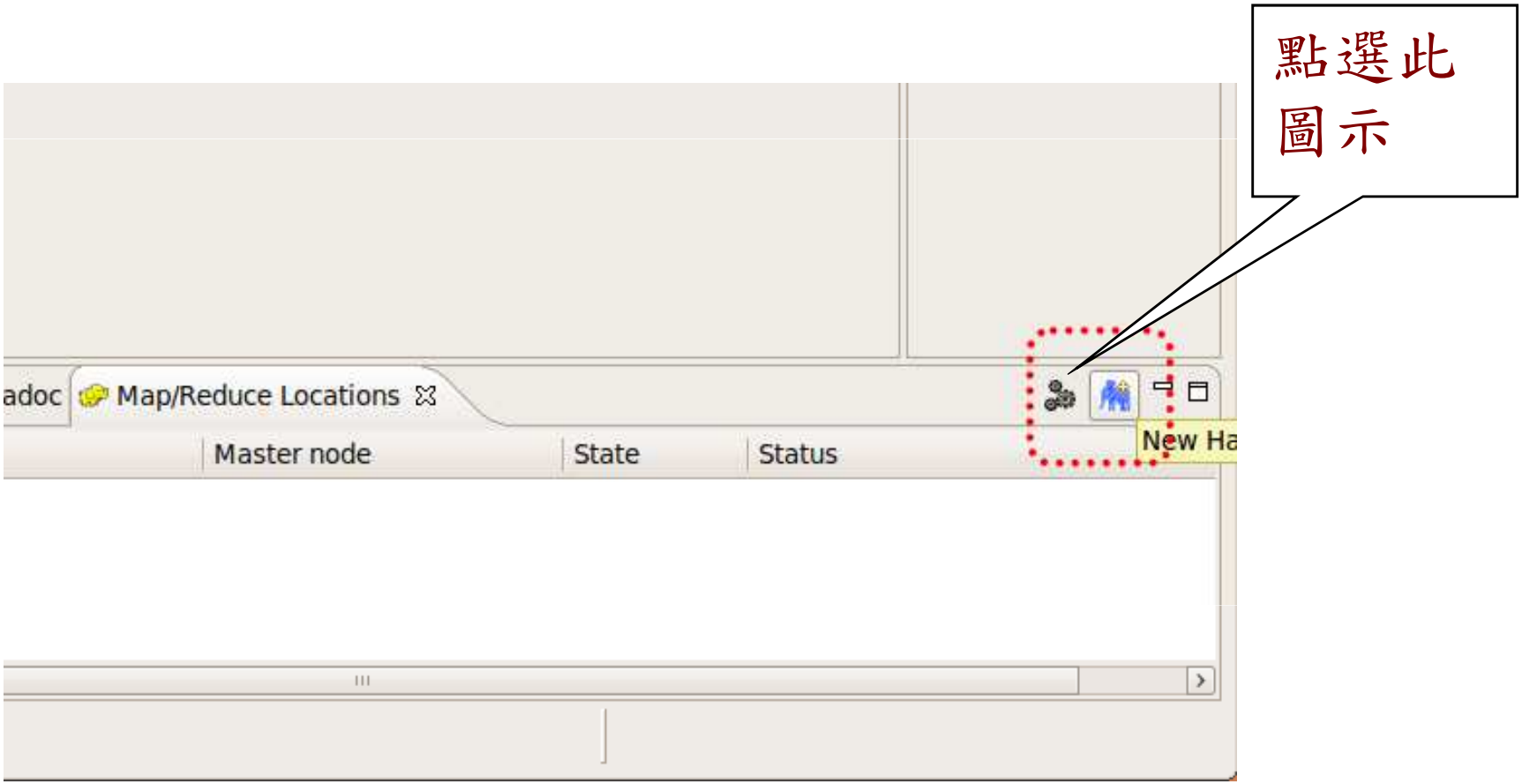
5-2. 設定java doc的完整路徑



選擇 Javadoc Location

輸入java 6 的API正確路徑，輸入完後可選擇validate以驗證是否正確

6. 連結Hadoop Server與Eclipse



6-1. 設定你要連接的Hadoop主機

任意填一個名稱

輸入主機位址或domain name

MapReduce 監聽的Port (設定於mapred-site.xml)

New Hadoop location...

Define Hadoop location

Define the location of a Hadoop infrastructure for running MapReduce applications.

General | Advanced parameters

Location name: hadoop

Map/Reduce Master
Host: localhost
Port: 9001

DFS Master
☑ Use M/R Master host
Host: localhost
Port: 9000

User name: waue

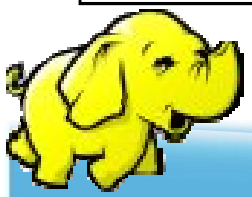
SOCKS proxy
☐ Enable SOCKS proxy
Host: host
Port: 1080

Load from file | Validate location

Finish | Cancel

HDFS監聽的Port (設定於core-site.xml)

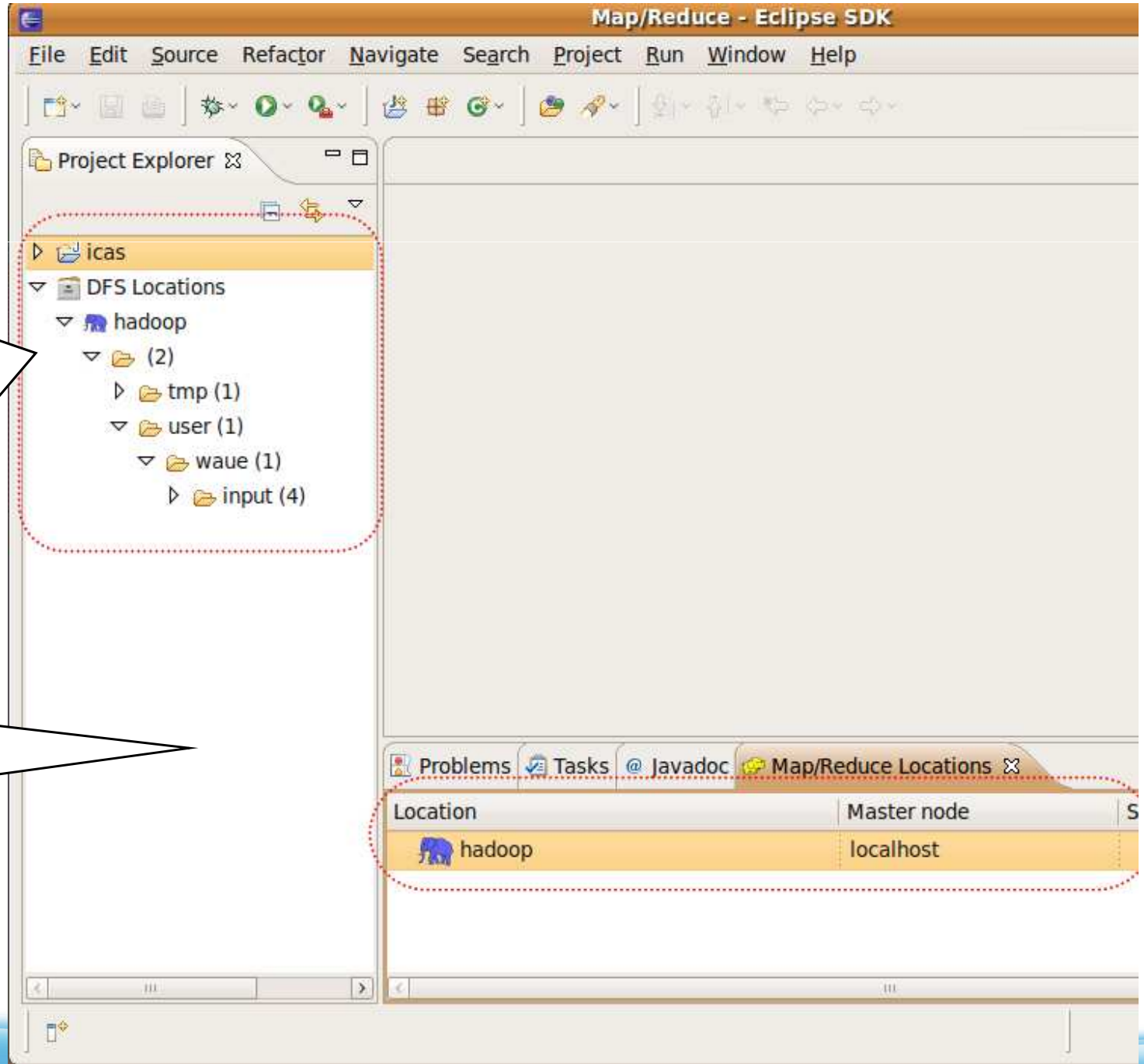
你在此Hadoop Server上的Username



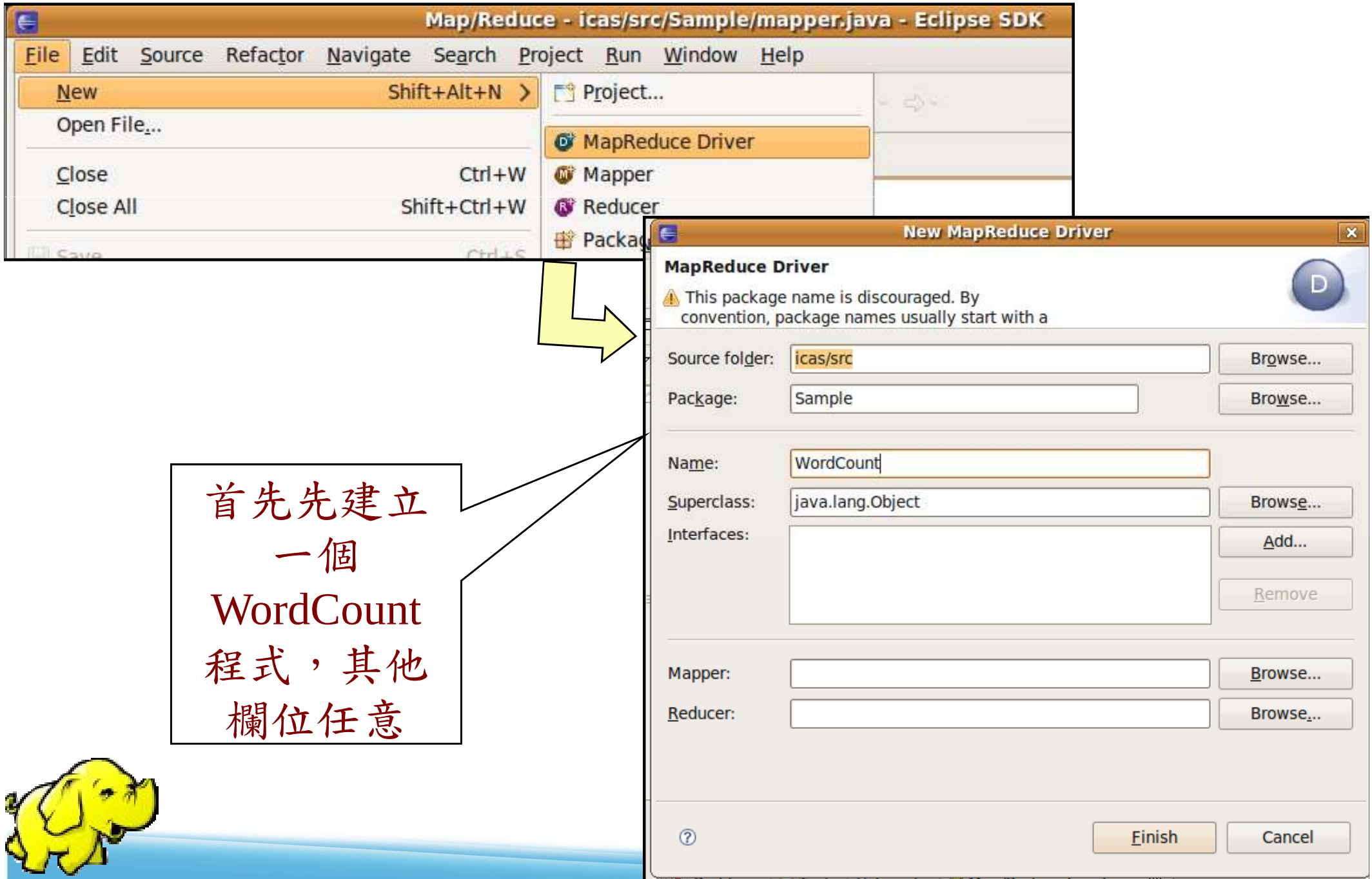
6-2 若正確設定則可得到以下畫面

HDFS的資訊，可直接於此操作檢視、新增、上傳、刪除等命令

若有Job運作，可於此視窗檢視



7. 新增一個Hadoop程式



首先先建立一個WordCount程式，其他欄位任意

7.1 於程式窗格內輸入程式碼

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer

DFS Locations
- secuse
  - (2)
    - user (1)
      - waue (3)
    - var (1)
- hadoop020
  - src
    - (default package)
      - CheckAndDelete
      - GetFromHdfs.jav
      - HelloHadoop.jav
      - HelloHadoopV2.
      - HelloHadoopV3.
      - HelloMapperV2.
      - HelloReducerV2.
      - PutToHdfs.java
      - TsmMenu.java
      - WordCount.java
      - WordCountV02(
      - WordCountV2.ja

WordCount.java

```
//1.    在hdfs 上來源檔案的路徑為 你所指定的 <input>
//請注意必須先放資料到此hdfs上的資料夾內，且此資料夾內只能放檔案，不可再放資料夾
//2.    運算完後，程式將執行結果放在hdfs 的輸出路徑為 你所指定的 <output>
//
public class WordCount {

    public static class TokenizerMapper extends
            Mapper<Object, Text, Text, IntWritable> {

        private final static IntWritable one = new IntWritable(1);
        private Text word = new Text();

        public void map(Object key, Text value, Context context)
                throws IOException, InterruptedException {
            StringTokenizer itr = new StringTokenizer(value.toString());
            while (itr.hasMoreTokens()) {
                word.set(itr.nextToken());
                context.write(word, one);
            }
        }
    }

    public static class IntSumReducer extends
            Reducer<Text, IntWritable, Text, IntWritable> {
        private IntWritable result = new IntWritable();

        public void reduce(Text key, Iterable<IntWritable> values,
                Context context) throws IOException, InterruptedException {
            int sum = 0;
            for (IntWritable val : values) {
                sum += val.get();
            }
            result.set(sum);
            context.write(key, result);
```

此區為程式窗格

Outline
- import declarations
- WordCount
  - TokenizerMapper
  - IntSumReducer
  - main(String[])

Problems | Tasks | Javadoc | Console | Map/Reduce Locations

Location	Master node	State	Status
secuse	secuse.nchc.org.tw		

7.2 補充

若之前doc部份設定正確，則滑鼠移至程式碼可取得API完整說明

mapper.java | reducer.java | WordCount.java

```
package Sample;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.util.GenericOptionsParser;
```

org.apache.hadoop.util.GenericOptionsParser

GenericOptionsParser is a utility to parse command line arguments generic to the Hadoop framework. GenericOptionsParser recognizes several standarad command line arguments, enabling applications to easily specify a namenode, a jobtracker, additional configuration resources etc.

Generic Options

The supported generic options are:

```
-conf <configuration file>     specify a configuration file
-D <property=value>            use value for given property
-fs <local|namenode:port>      specify a namenode
```

Press 'F2' for focus

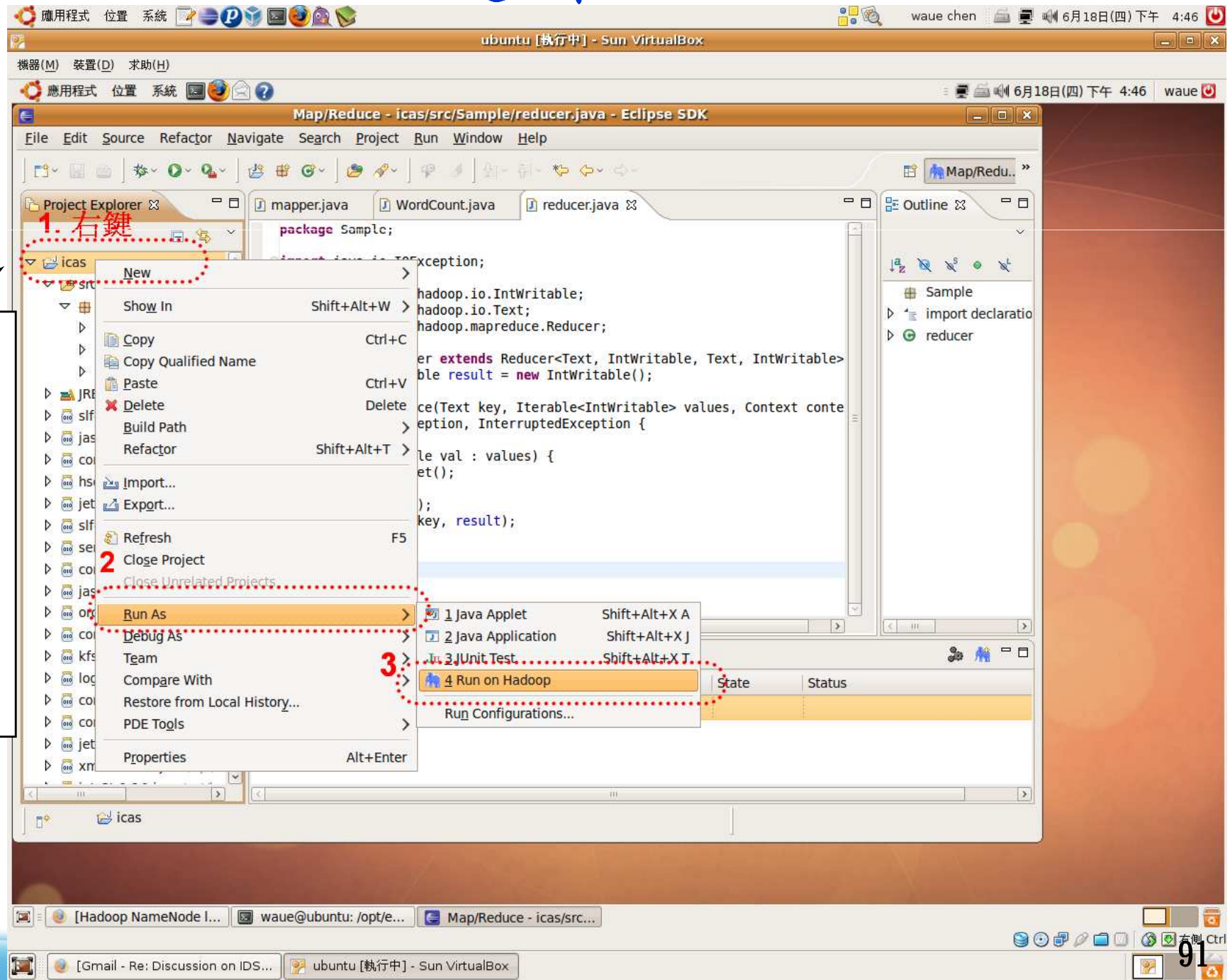
Outline: Sample, import declaratio, org.apache.ha, org.apache.ha, org.apache.ha

Problems | Tasks | Javadoc | Map/Reduce Locations | Console

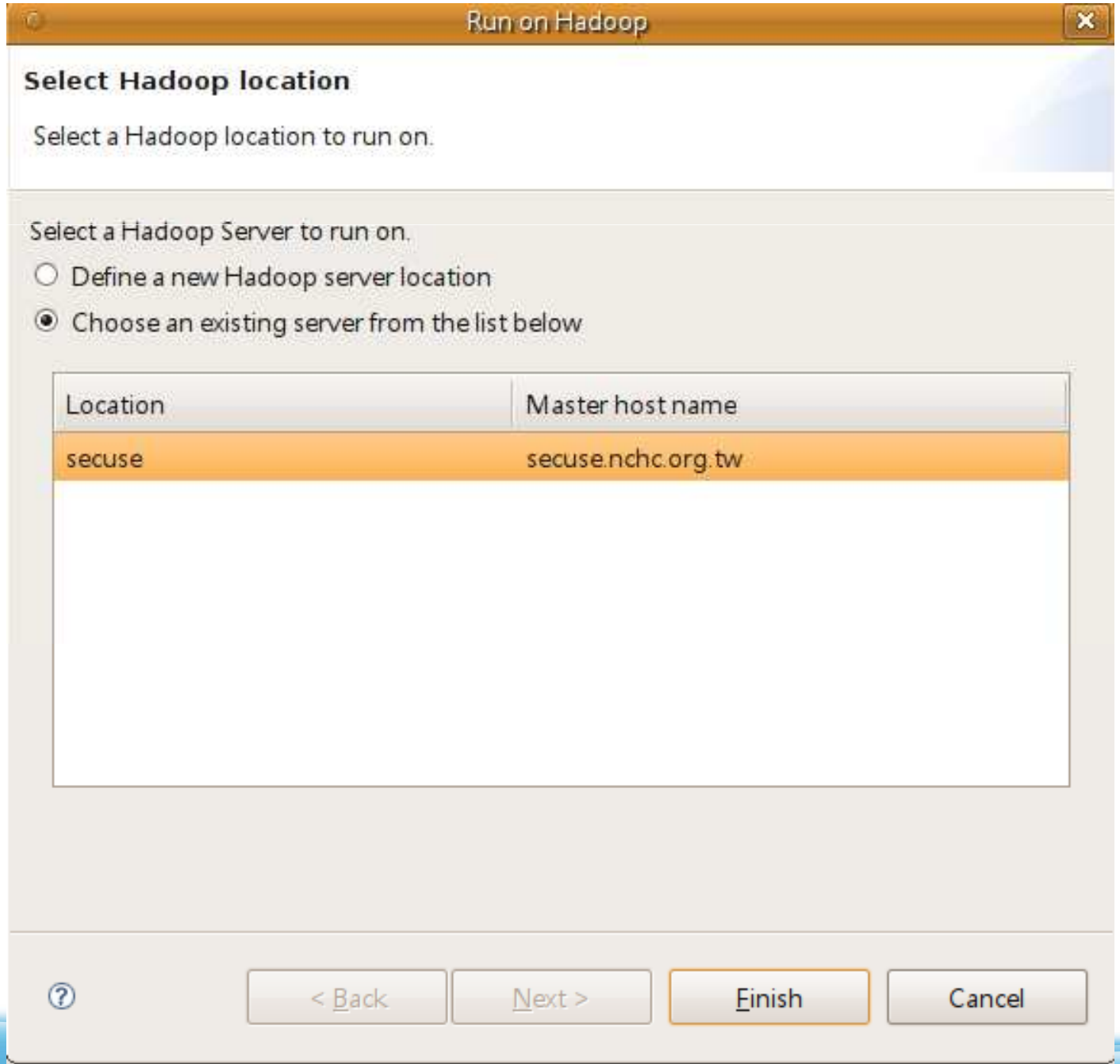
Location	Master node	State	Status
job_200906161902_0005		FAILED	Maps : 2/2 (1.0) Reduces : 1/1 (1.0)

8. 運作

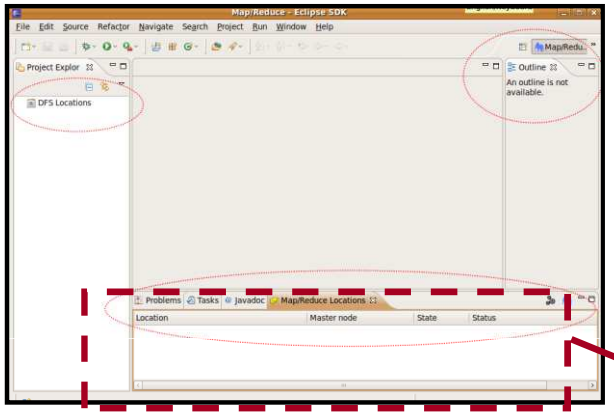
於欲運算的程式碼處點選右鍵 → Run As → **Run on Hadoop**



8-1 選擇之前設定好所要運算的主機



8.2 運算資訊出現於Eclipse 右下方的Console 視窗

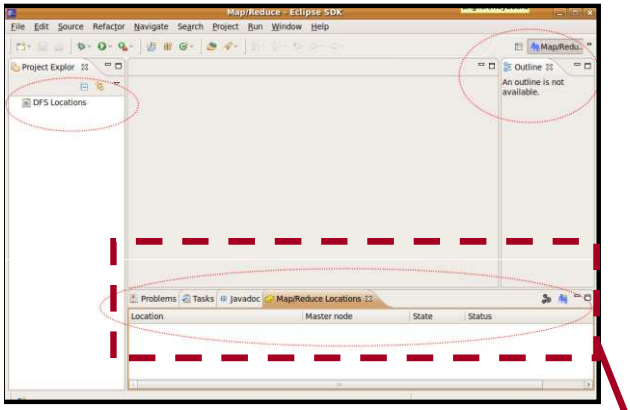


放大

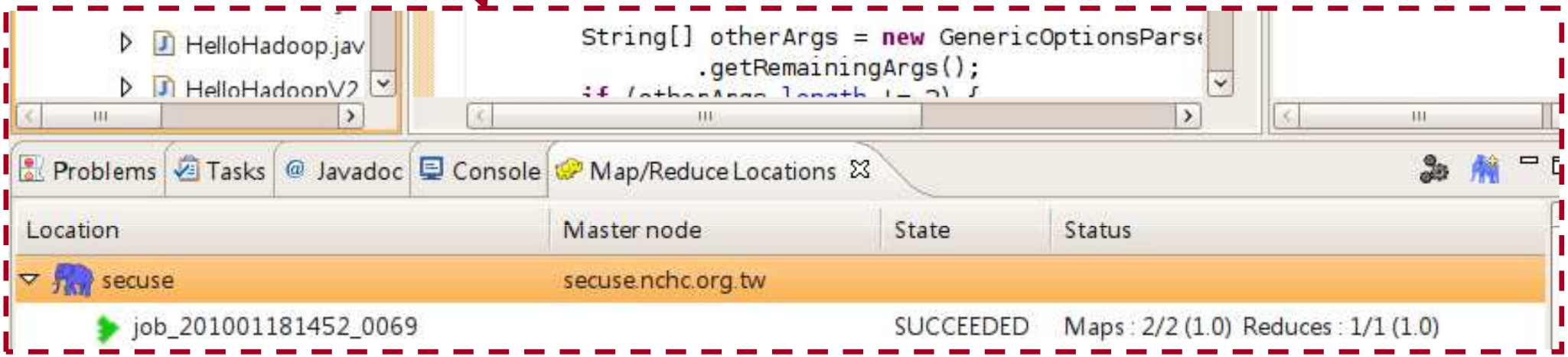
Problems | Tasks | Javadoc | Console | Map/Reduce Locations

```
<terminated> WordCount (1) [Java Application] /usr/lib/jvm/java-6-sun-1.6.0.16/bin/java (2010/1/20 下午 6:15:07)
10/01/20 18:15:08 WARN conf.Configuration: DEPRECATED: hadoop-site.xml found in the classpath. Usage of hadc
10/01/20 18:15:08 WARN mapred.JobConf: The variable mapred.task.maxvmem is no longer used. Instead use mapre
10/01/20 18:15:08 WARN mapred.JobConf: The variable mapred.task.maxvmem is no longer used. Instead use mapre
10/01/20 18:15:08 INFO input.FileInputFormat: Total input paths to process : 2
10/01/20 18:15:08 INFO mapred.JobClient: Running job: job_201001181452_0078
10/01/20 18:15:09 INFO mapred.JobClient:  map 0% reduce 0%
10/01/20 18:15:16 INFO mapred.JobClient:  map 100% reduce 0%
10/01/20 18:15:28 INFO mapred.JobClient:  map 100% reduce 100%
10/01/20 18:15:30 INFO mapred.JobClient: Job complete: job_201001181452_0078
10/01/20 18:15:30 INFO mapred.JobClient: Counters: 17
10/01/20 18:15:30 INFO mapred.JobClient:   Job Counters
10/01/20 18:15:30 INFO mapred.JobClient:     Launched reduce tasks=1
10/01/20 18:15:30 INFO mapred.JobClient:     Launched map tasks=2
10/01/20 18:15:30 INFO mapred.JobClient:     Data-local map tasks=2
10/01/20 18:15:30 INFO mapred.JobClient:   FileSystemCounters
10/01/20 18:15:30 INFO mapred.JobClient:     FILE_BYTES_READ=153
```

8.3 剛剛運算的結果出現如下圖



放大



Conclusions

- 優點
  - 快速開發程式
  - 易於除錯
  - 智慧尋找函式庫
  - 自動鍊結API
  - 直接操控 HDFS 與 JobTracker
  - …
- 缺點
  - Plugin 並會因Eclipse 版本而有不同的狀況