

附录 A：基于 Cygwin 的 Hadoop 环境搭建

1. 安装和配置 Cygwin

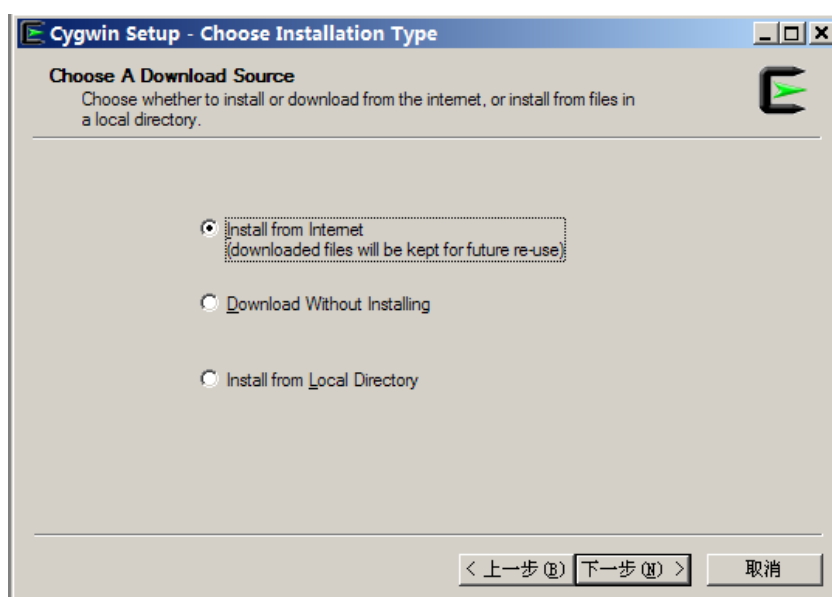
为 Hadoop 准备的 Windows 下 Cygwin 环境安装过程如下：

➤ 下载安装文件

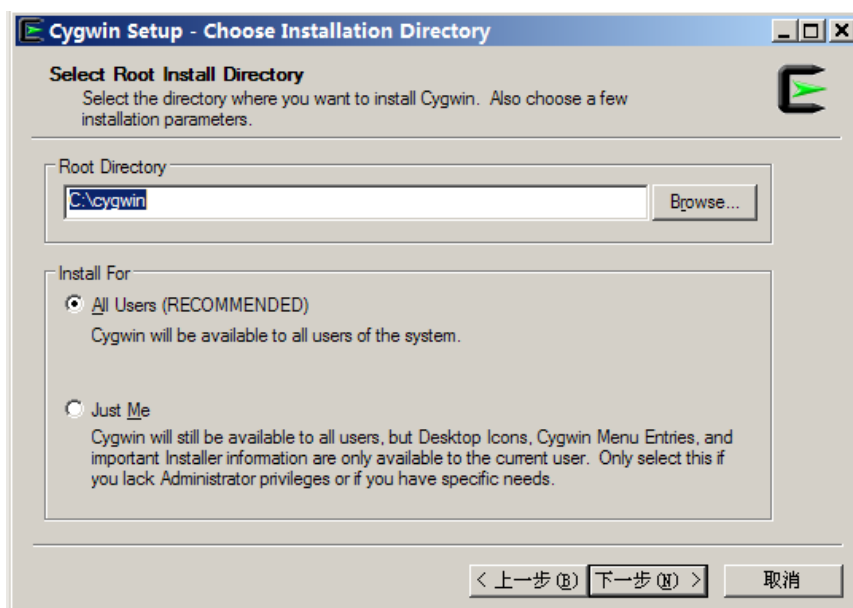
最新的 cygwin 安装文件 setup.exe 下载地址在这里：<http://cygwin.com/install.html>。用最新版本的 cygwin 就可以，我用的是 2.774 版本的安装程序。

➤ 安装 cygwin

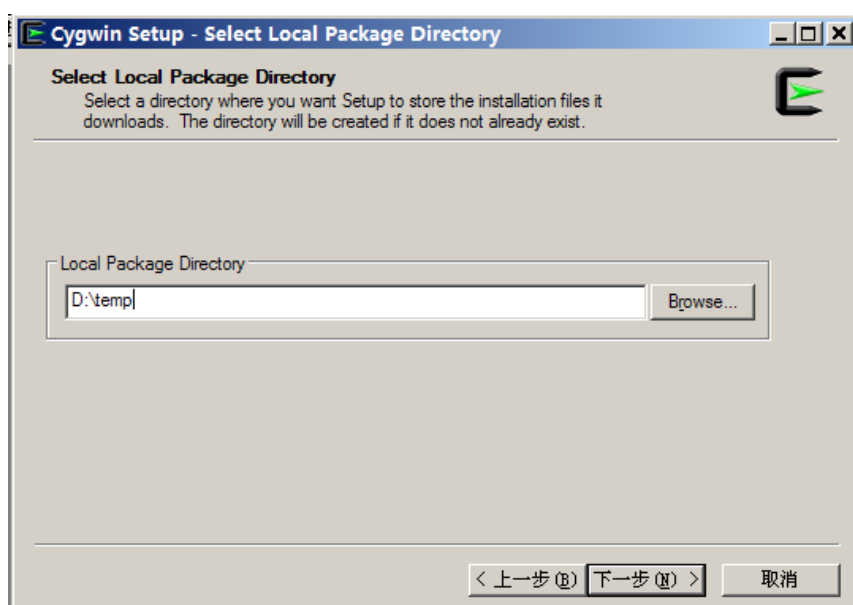
在上一步下载的 setup.exe 文件实际上只是一个引导安装和下载过程的执行文件，真正的下载安装过程是通过网络进行的，然后执行下载后的 setup.exe 文件，并点击下一步进入安装模式引导界面。



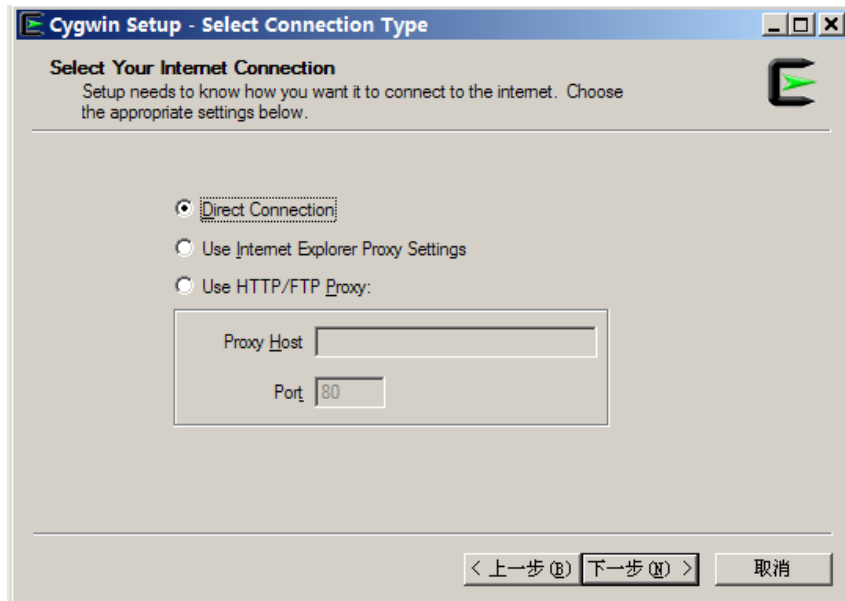
这里面的三个选项是“从网络下载并安装”、“只下载不安装”、“从本地下载文件安装”，简单粗暴地选择第一个“从网络下载并安装”，点击下一步后进入选择安装目录界面。



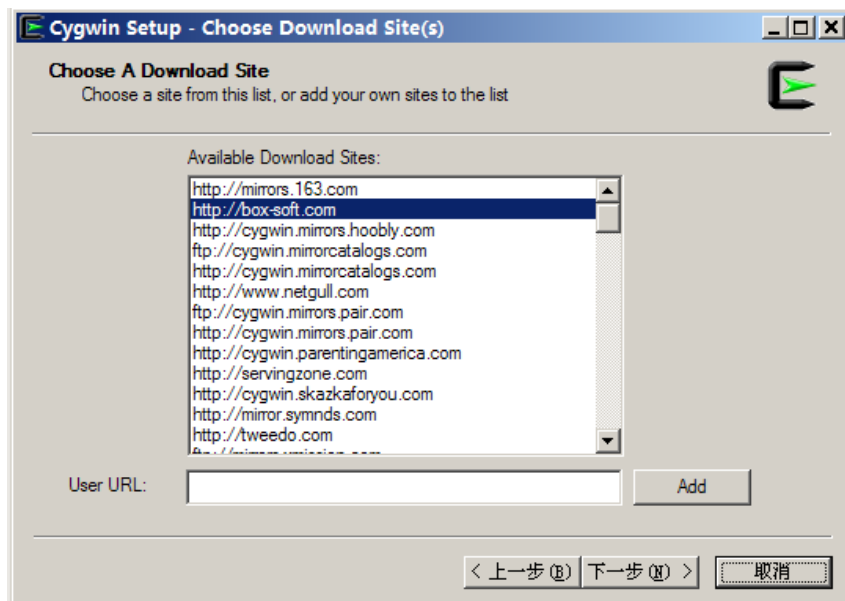
使用默认的 Cygwin 安装目录，并在下面选项中选择默认的所有本机用户都能使用。点击下一步进入下载文件存放目录选择界面。



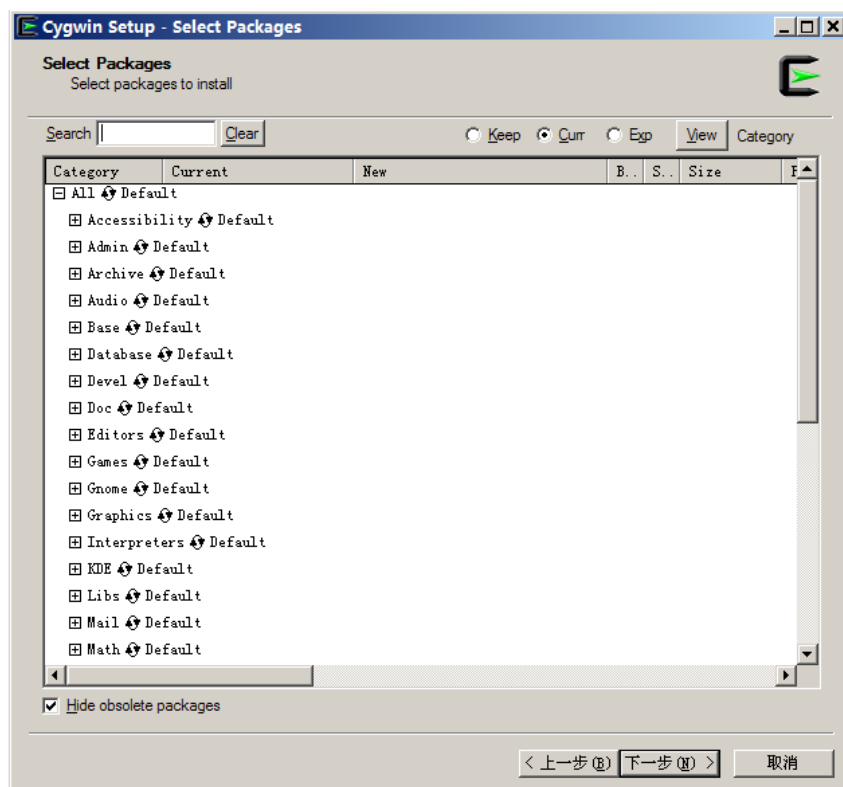
可以选择让下载文件放到常用的临时文件目录下，点击下一步（如果输入的是一个不存在的目录，程序会提示你是否要创建这个目录，选 yes 就 ok），进入网络链接选择界面。



选择 Direct Connection 直接连接网络，点击下一步后会有一个下载安装文件镜像服务器列表的短暂过程，然后会出现选择安装文件镜像服务器的界面。



按照正常情况选择第一个 mirrors.163.com 的服务器应该是最快的，但是有可能会出现 setup 文件下载出错的情况，导致安装不能完成。建议选择了第二个 box-soft.com 服务器，击下一步后，仍然会有一个短暂的从所选服务器下载安装组件列表的过程，然后出现选择安装组件的界面。



选择安装组件的步骤比较重要，需要仔细选择以下组件：

- Base 组件的全部，操作方法是点击 Base 后面的 Default，变为 Install
- Devel 组件下的 subversion 及其他将来开发需要用到的组件，例如 autoconf 等，操作方法是展开 Devel 组件，点击各个小组件前的 Keep 文字，变为相应的版本号。
- Net 组件下的 openssh 和 openssl 组件，用于 hadoop 需要的 ssh 访问，操作方法同上。
- System 组件下的 util-linux 组件，用于使用一些常用的 more 等功能进行调试，操作方法同上。
- 其他一些可能用到的组件，可以单独选择 Perl、Python、Ruby、Science 等组件。

选择组件完成后，点击下一步即开始进行下载、安装等自动步骤，一路选择下一步即可。

➤ 配置 cygwin 的 ssh 服务

Cygwin 安装完成后，需要对 ssh 服务进行配置，以运行 hadoop 环境进行 ssh 无密码登录，过程如下：

- 使用安装后生成的 cygwin 启动快捷方式，启动 cygwin 环境。
- 执行 cygwin 的 ssh-host-config。

```
liujun@pc07 ~  
$ ssh-host-config  
  
*** Info: Generating /etc/ssh_host_key  
*** Info: Generating /etc/ssh_host_rsa_key  
*** Info: Generating /etc/ssh_host_dsa_key  
*** Info: Generating /etc/ssh_host_ecdsa_key  
*** Info: Creating default /etc/ssh_config file  
*** Info: Creating default /etc/sshd_config file  
*** Info: Privilege separation is set to yes by default since OpenSSH 3.3.  
*** Info: However, this requires a non-privileged account called 'sshd'.  
*** Info: For more info on privilege separation read /usr/share/doc/openssh/README.privsep.  
*** Query: Should privilege separation be used? (yes/no) no  
*** Info: Updating /etc/sshd_config file  
*** Info: Added ssh to C:\WINDOWS\system32\drivers\services  
  
*** Query: Do you want to install sshd as a service?  
*** Query: (Say "no" if it is already installed as a service) (yes/no) yes  
*** Query: Enter the value of CYGWIN for the daemon: []  
  
*** Info: The sshd service has been installed under the LocalSystem  
*** Info: account (also known as SYSTEM). To start the service now, call  
*** Info: 'net start sshd' or 'cygrunsrv -S sshd'. Otherwise, it  
*** Info: will start automatically after the next reboot.  
  
*** Info: Host configuration finished. Have fun!  
  
liujun@pc07 ~  
$ |
```

在第一步询问“Should privilege separation be used? (yes/no)”时，输入 no；

在第二步询问“(Say 'no' if it is already installed as a service) (yes/no)”时，输入 yes；

在第三步询问“Enter the value of CYGWIN for the daemon: []”，直接回车。

看到“Host configuration finished. Have fun!”后此步即完成。

- 使用 Windows 的管理工具中的服务管理，将“CYGWIN sshd”服务启动。
- 回到 Cygwin 环境，执行 ssh localhost 命令。

```
$ ssh localhost  
The authenticity of host 'localhost (127.0.0.1)' can't be established.  
ECDSA key fingerprint is e2:9a:5d:c7:b8:a9:dc:42:d6:77:55:45:ee:30:bc:3b.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.  
liujun@localhost's password:
```

在第一步询问中输入 yes，在第二步要求输入密码时，输入用户密码。

- 在 cygwin 中输入 ssh-keygen，一路回车即可。

```
liujun@pc07 ~
$ ssh localhost
liujun@localhost's password:
Last login: Sat Jun 30 19:43:18 2012 from localhost

liujun@pc07 ~
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/liujun/.ssh/id_rsa):
/home/liujun/.ssh/id_rsa already exists.
overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/liujun/.ssh/id_rsa.
Your public key has been saved in /home/liujun/.ssh/id_rsa.pub.
The key fingerprint is:
8a:4c:d7:54:4b:6d:d3:19:cf:f2:9b:13:fa:a2:bb:8e liujun@pc07
The key's randomart image is:
+--[ RSA 2048 ]-----+
  o . . . o
  o . + . oo
  . . . . o
  o . . . o
  o . . . S
  o o . . . +
  o . . . +
  . . . . .
  . . . . .
  E . = + . .
+-----+

liujun@pc07 ~
$ |
```

- 然后在 cygwin 下依次执行如下命令：

```
cd ~/.ssh
```

```
cp id_rsa.pub authorized_keys
```

完成后连续使用 exit 命令退出 Cygwin 环境,再打开 Cygwin 环境,执行 ssh localhost,发现不需要密码即可进入,就代表成功了。

至此,我们为 Hadoop 安装准备的 Cygwin 环境即已搭建完成

2. 安装和配置 Hadoop

在上一步安装好 cygwin 环境后,下面我们进入 hadoop 安装环节。

➤ 安装 JDK

Hadoop 运行需要 jdk 环境,我下载了最新的 JDK 7u5 版本,具体地址请搜索 JDK 官方网站,然后在下面的列表中选择合适的版本,我们需要选择的是 Windows X86 版本。

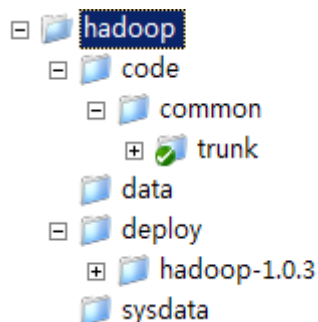
Java SE Development Kit 7u5		
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.		
Product / File Description	File Size	Download
Linux x86	64.1 MB	jdk-7u5-linux-i586.rpm
Linux x86	79.1 MB	jdk-7u5-linux-i586.tar.gz
Linux x64	64.93 MB	jdk-7u5-linux-x64.rpm
Linux x64	77.67 MB	jdk-7u5-linux-x64.tar.gz
Macosx-x64	97.3 MB	jdk-7u5-macosx-x64.dmg
Solaris x86	137.41 MB	jdk-7u5-solaris-i586.tar.Z
Solaris x86	82.01 MB	jdk-7u5-solaris-i586.tar.gz
Solaris SPARC	140.43 MB	jdk-7u5-solaris-sparc.tar.Z
Solaris SPARC	86.72 MB	jdk-7u5-solaris-sparc.tar.gz
Solaris SPARC 64-bit	16.45 MB	jdk-7u5-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	12.55 MB	jdk-7u5-solaris-sparcv9.tar.gz
Solaris x64	14.92 MB	jdk-7u5-solaris-x64.tar.Z
Solaris x64	9.54 MB	jdk-7u5-solaris-x64.tar.gz
Windows x86	87.95 MB	jdk-7u5-windows-i586.exe
Windows x64	92.36 MB	jdk-7u5-windows-x64.exe

下载完成后，执行安装，建议安装目录不要实用默认的 c:\Program Files 下的目录，可以选择相对简洁的路径，例如 c:\java 目录作为安装路径。因为在后面的 Hadoop 配置过程中，需要配置 JDK 的路径，如果安装在 c:\Program Files 下配置过程会相对复杂。

➤ 下载及安装 hadoop

安装好 JDK 后，去 Apache Hadoop 网站下载一个合适的 Hadoop 版本，因为在本书编写时，2.0 版本刚刚发布的，且是 Alpha 版本，因此我们选择了较为稳定的 1.0.3 版本下载，下载文件为 hadoop-1.0.3.tar.gz。

下载完成后，需要先仔细规划下 Hadoop 部署目录，建议目录部署如下：



Hadoop 目录位于 D 盘根目录下；code 目录用于存放后面建立 Eclipse 开发环境时需要用到的代码，code 下的 common/trunk 目录为 svn checkout 的 Hadoop 核心组件代码；data 目录用于存放将来用到的分析数据和输出结果；deploy 目录用于存放 Hadoop 运行文件，下载的 hadoop-1.0.3.tar.gz 文件解压后的文件即放在此目录下；sysdata 目录用于存放 hadoop 运行时需要产生和用到的系统数据，例如 namenode、tmp 目录等。

建立目录后，即可将下载的 hadoop-1.0.3.tar.gz 文件解压后的文件放在 deploy 目录下。

➤ 配置 **hadoop-env.sh**

hadoop-env.sh 文件中，需要配置 JDK 的安装路径，配置内容如下：

```
# Set Hadoop-specific environment variables here.

# The only required environment variable is JAVA_HOME. All others are
# optional. When running a distributed configuration it is best to
# set JAVA_HOME in this file, so that it is correctly defined on
# remote nodes.

# The java implementation to use. Required.
export JAVA_HOME=/cygdrive/c/Java/jdk1.7.0
```

➤ 配置 **core-site.xml**

core-site.xml 配置文件中可配置的内容很多，可以慢慢研究，我们先使用代码中自带的模版文件进行少量修改以满足基本的运行条件，步骤如下：

- 拷贝 **D:\hadoop\deploy\hadoop-1.0.3\src\core\core-default.xml** 文件到 **D:\hadoop\deploy\hadoop-1.0.3\conf** 目录下，并重命名为 **core-site.xml**。

- 修改临时文件存放路径

```
<property>
  <name>hadoop.tmp.dir</name>
  <value>/hadoop/sysdata/1.0.3/tmp</value>
  <description>A base for other temporary directories.</description>
</property>
```

- 修改文件系统的默认名称

```
<property>
  <name>fs.default.name</name>
  <value>hdfs://localhost:9000</value>
  <description>The name of the default file system. A URI whose
scheme and authority determine the FileSystem implementation. The
uri's scheme determines the config property (fs.SCHEME.impl) naming
the FileSystem implementation class. The uri's authority is used to
determine the host, port, etc. for a filesystem.</description>
</property>
```

➤ 配置 **hdfs-site.xml**

同样使用代码中自带的模版文件进行少量修改以满足基本的运行条件。

- 拷贝 **D:\hadoop\deploy\hadoop-1.0.3\src\hdfs\hdfs-default.xml** 文件到 **D:\hadoop\deploy\hadoop-1.0.3\conf** 目录下，并重命名为 **hdfs-site.xml**。

- 修改 DFS 文件系统 namenode 存放 name 表的目录

```
<property>
```



```

<name>dfs.name.dir</name>
<value>/hadoop/sysdata/1.0.3/name</value>
<description>Determines where on the local filesystem the DFS name node
    should store the name table(fsimage). If this is a comma-delimited list
    of directories then the name table is replicated in all of the
    directories, for redundancy. </description>
</property>

```

- 修改 DFS 文件系统 datanode 存放数据的目录

```

<property>
  <name>dfs.data.dir</name>
  <value>/hadoop/sysdata/1.0.3/data</value>
  <description>Determines where on the local filesystem an DFS data node
    should store its blocks. If this is a comma-delimited
    list of directories, then data will be stored in all named
    directories, typically on different devices.
    Directories that do not exist are ignored.
  </description>
</property>

```

- 修改数据存放副本数量为 1（因为我们要部署的是伪分布式单节点）

```

<property>
  <name>dfs.replication</name>
  <value>1</value>
  <description>Default block replication.
    The actual number of replications can be specified when the file is created.
    The default is used if replication is not specified in create time.
  </description>
</property>

```

➤ 配置 **mapred-site.xml**

同样使用代码中自带的模版文件进行少量修改以满足基本的运行条件。

- 拷贝 D:\hadoop\deploy\hadoop-1.0.3\src\ mapred\mapred-default.xml 文件到 D:\hadoop\deploy\hadoop-1.0.3\conf 目录下，并重命名为 mapred-site.xml。

- 修改 jobtracker 运行的服务器和端口

```

<property>
  <name>mapred.job.tracker</name>

```

<value>localhost:9001</value>

<description>The host and port that the MapReduce job tracker runs at. If "local", then jobs are run in-process as a single map and reduce task.

</description>

</property>

- 修改 mapreduce 运行存放的即时数据文件目录

<property>

<name>mapred.local.dir</name>

<value>/hadoop/sysdata/1.0.3/temp</value>

<description>The local directory where MapReduce stores intermediate data files. May be a comma-separated list of directories on different devices in order to spread disk i/o. Directories that do not exist are ignored.

</description>

</property>

- Mapreduce 存放临时文件的目录

<property>

<name>mapred.child.tmp</name>

<value>/hadoop/sysdata/1.0.3/temp</value>

<description> To set the value of tmp directory for map and reduce tasks. If the value is an absolute path, it is directly assigned. Otherwise, it is prepended with task's working directory. The java tasks are executed with option -Djava.io.tmpdir='the absolute path of the tmp dir'. Pipes and streaming are set with environment variable,

TMPDIR='the absolute path of the tmp dir'

</description>

</property>

➤ 格式化 namenode

在以上配置文件完成后，我们可以开始对 namenode 进行格式化了，这是 hadoop 开始使用的第一步，就像我们对硬盘进行格式化操作一样，指令执行过程如下图：

```
/cygdrive/d/hadoop/deploy/hadoop-1.0.3/bin
liujun@pc07 ~
$ cd /cygdrive/d/hadoop/deploy/hadoop-1.0.3/bin
liujun@pc07 /cygdrive/d/hadoop/deploy/hadoop-1.0.3/bin
$ ls
hadoop          slaves.sh          start-mapred.sh    stop-mapred.sh
hadoop-config.sh start-all.sh       stop-all.sh        task-controller
hadoop-daemon.sh start-balancer.sh  stop-balancer.sh
hadoop-daemons.sh start-dfs.sh        stop-dfs.sh
rc               start-jobhistoryserver.sh stop-jobhistoryserver.sh

liujun@pc07 /cygdrive/d/hadoop/deploy/hadoop-1.0.3/bin
$ ./hadoop namenode -format
12/07/01 16:48:22 INFO namenode.NameNode: STARTUP_MSG:
*****
STARTUP_MSG: Starting NameNode
STARTUP_MSG: host = pc07/192.168.1.10
STARTUP_MSG: args = [-format]
STARTUP_MSG: version = 1.0.3
STARTUP_MSG: build = https://svn.apache.org/repos/asf/hadoop/common/branches/branch-1.0 -r 1335192
; compiled by 'hortonfo' on Tue May 8 20:31:25 UTC 2012
*****
12/07/01 16:48:23 INFO util.GSet: VM type = 32-bit
12/07/01 16:48:23 INFO util.GSet: 2% max memory = 19.33375 MB
12/07/01 16:48:23 INFO util.GSet: capacity = 2^22 = 4194304 entries
12/07/01 16:48:23 INFO util.GSet: recommended=4194304, actual=4194304
12/07/01 16:48:23 INFO namenode.FSNamesystem: fsowner=liujun
12/07/01 16:48:23 INFO namenode.FSNamesystem: supergroup=supergroup
12/07/01 16:48:23 INFO namenode.FSNamesystem: isPermissionEnabled=true
12/07/01 16:48:23 INFO namenode.FSNamesystem: dfs.block.invalidate.limit=100
12/07/01 16:48:23 INFO namenode.FSNamesystem: isAccessTokenEnabled=false accessKeyUpdateInterval=0 m
in(s), accessTokenLifetime=0 min(s)
12/07/01 16:48:23 INFO namenode.NameNode: Caching file names occurring more than 10 times
12/07/01 16:48:24 INFO common.Storage: Image file of size 112 saved in 0 seconds.
12/07/01 16:48:24 INFO common.Storage: Storage directory \hadoop\sysdata\1.0.3\name has been success
fully formatted.
12/07/01 16:48:24 INFO namenode.NameNode: SHUTDOWN_MSG:
*****
SHUTDOWN_MSG: Shutting down NameNode at pc07/192.168.1.10
*****
liujun@pc07 /cygdrive/d/hadoop/deploy/hadoop-1.0.3/bin
$ |
```

基本过程就是进入 Hadoop 安装目录的 bin 目录，执行格式化指令，具体指令流程如下：

```
cd /cygdrive/d/hadoop/deploy/hadoop-1.0.3/bin
./hadoop namenode -format
```

➤ 启动 hadoop

格式化 namenode 后，我们就可以执行 ./start-all.sh 开始启动 hadoop 了。

```
liujun@pc07 ~  
$ cd /cygdrive/d/hadoop/deploy/hadoop-1.0.3/bin  
liujun@pc07 /cygdrive/d/hadoop/deploy/hadoop-1.0.3/bin  
$ ls  
hadoop          slaves.sh          start-mapred.sh    stop-mapred.sh  
hadoop-config.sh start-all.sh       stop-all.sh        task-controller  
hadoop-daemon.sh start-balancer.sh  stop-balancer.sh  
hadoop-daemons.sh start-dfs.sh        stop-dfs.sh  
rcc             start-jobhistoryserver.sh stop-jobhistoryserver.sh  
liujun@pc07 /cygdrive/d/hadoop/deploy/hadoop-1.0.3/bin  
$ ./start-all.sh  
starting namenode, logging to /cygdrive/d/hadoop/deploy/hadoop-1.0.3/libexec/../logs/hadoop-liujun-namenode-pc07.out  
localhost: starting datanode, logging to /cygdrive/d/hadoop/deploy/hadoop-1.0.3/libexec/../logs/hadoop-liujun-datanode-pc07.out  
localhost: starting secondarynamenode, logging to /cygdrive/d/hadoop/deploy/hadoop-1.0.3/libexec/../logs/hadoop-liujun-secondarynamenode-pc07.out  
starting jobtracker, logging to /cygdrive/d/hadoop/deploy/hadoop-1.0.3/libexec/../logs/hadoop-liujun-jobtracker-pc07.out  
localhost: starting tasktracker, logging to /cygdrive/d/hadoop/deploy/hadoop-1.0.3/libexec/../logs/hadoop-liujun-tasktracker-pc07.out  
liujun@pc07 /cygdrive/d/hadoop/deploy/hadoop-1.0.3/bin  
$
```

如果使用 hadoop 完成，或需要关闭 hadoop 了，执行 ./stop-all.sh 即可。

至此，Hadoop 的基本运行环境就已经部署好了，但是要验证是否可以正常实用还需要运行一下 Hadoop 中自带的 wordcount 示例程序，具体请看下一节。

3. 运行示例程序验证 Hadoop 安装

在前面的步骤中，我们已经建立了一个基本的 Hadoop 运行环境，下面实用 Hadoop 自带的一个 wordcount 程序验证运行环境是否可以正常运行，步骤如下：

➤ 建立本地数据文件

在我们准备的 Hadoop 本地文件夹的 data 目录下建立一个 data_in 文件夹，并在此目录下创建两个数据文件，分别是 file1.txt 和 file2.txt。

file1.txt 中保存一个句子：Hello world!

file2.txt 中保存一个句子：I am the king of the world!

➤ 上传数据文件至 dfs 文件系统

下面我们要将本地建立的两个数据文件上传到 hdfs 文件系统中。以下过程，如果没有启动 hadoop 环境，请参考 hadoop 安装过程中启动 hadoop 的方法先启动 hadoop 环境，否则会看到“Retrying connect to server”的错误。

- Windows 下启动 cygwin 环境

- 进入 hadoop 的 bin 目录

```
cd /cygdrive/d/hadoop/deploy/hadoop-1.0.3/bin
```

- 在 hdfs 上建立 data_in 目录

```
./hadoop dfs -mkdir data_in
```

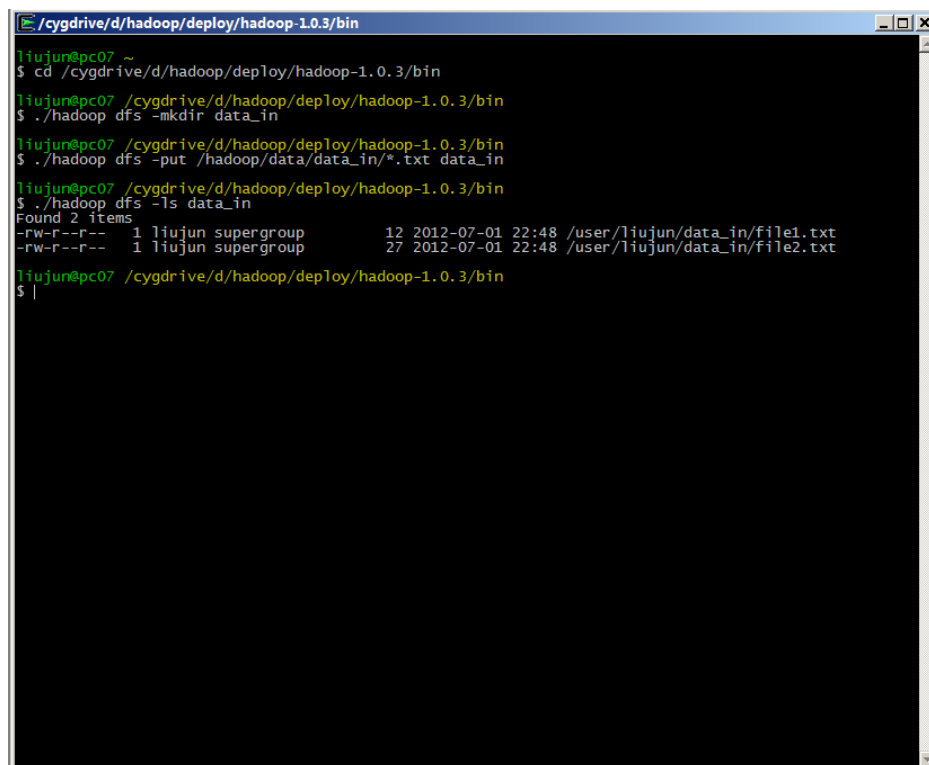
- 上传数据文件

```
./hadoop dfs -put /hadoop/data/data_in/*.txt data_in
```

- 查看文件上传成功

```
./hadoop dfs -ls data_in
```

整个过程如下图所示：



```
liujun@pc07 ~  
$ cd /cygdrive/d/hadoop/deploy/hadoop-1.0.3/bin  
liujun@pc07 /cygdrive/d/hadoop/deploy/hadoop-1.0.3/bin  
$ ./hadoop dfs -mkdir data_in  
liujun@pc07 /cygdrive/d/hadoop/deploy/hadoop-1.0.3/bin  
$ ./hadoop dfs -put /hadoop/data/data_in/*.txt data_in  
liujun@pc07 /cygdrive/d/hadoop/deploy/hadoop-1.0.3/bin  
$ ./hadoop dfs -ls data_in  
Found 2 items  
-rw-r--r-- 1 liujun supergroup      12 2012-07-01 22:48 /user/liujun/data_in/file1.txt  
-rw-r--r-- 1 liujun supergroup      27 2012-07-01 22:48 /user/liujun/data_in/file2.txt  
liujun@pc07 /cygdrive/d/hadoop/deploy/hadoop-1.0.3/bin  
$ |
```

➤ 执行 wordcount 程序

数据文件准备好了，我们就可以执行 wordcount 程序了。在 hadoop 的 bin 目录下执行：

```
./hadoop jar ../hadoop-examples-1.0.3.jar wordcount data_in data_out
```

当看到成功执行的输出信息时即说明 Hadoop 环境已经可以正常运行了。

4. 安装和配置 Eclipse 下的 Hadoop 开发环境

在学习和使用 Hadoop 的过程中，我们可能需要对 Hadoop 代码进行改进以实现一些特殊

的需求，这就要求我们必须建立一个可修改 Hadoop 代码的开发环境，下面的过程，我们就来建立一个基于 Eclipse 开发工具的 Hadoop 开发环境。

➤ 安装 Ant

Hadoop 的编译需要 Ant 的支持，从这里下载并安装最新的 Ant：

<http://ant.apache.org/bindownload.cgi>。

安装完成后，别忘了将 Ant 的 bin 目录路径加入到 windows 系统的 PATH 环境变量中。

➤ 安装 TortoiseSVN

Hadoop 代码是以 SVN 的形式存放在 apache 服务器上，因此我们需要先安装一个 SVN 客户端，推荐使用较为广泛的工具的 TortoiseSVN。

从这里下载并安装最新的 TortoiseSVN：<http://tortoisesvn.net/downloads.html>

➤ Checkout hadoop 代码

在我们前面建立的代码目录 D:\hadoop\code\common\chunk 目录下，checkout hadoop 代码。我们选用的是 1.0.3 版本的 hadoop，所以远程服务器代码的 URL 填入的是：

<http://svn.apache.org/repos/asf/hadoop/common/tags/release-1.0.3/>。

➤ 安装 Eclipse

代码 checkout 完成后，就该安装 Eclipse 工具了。

从这里下载并安装 Eclipse Classic 4.2：<http://www.eclipse.org/downloads/>。

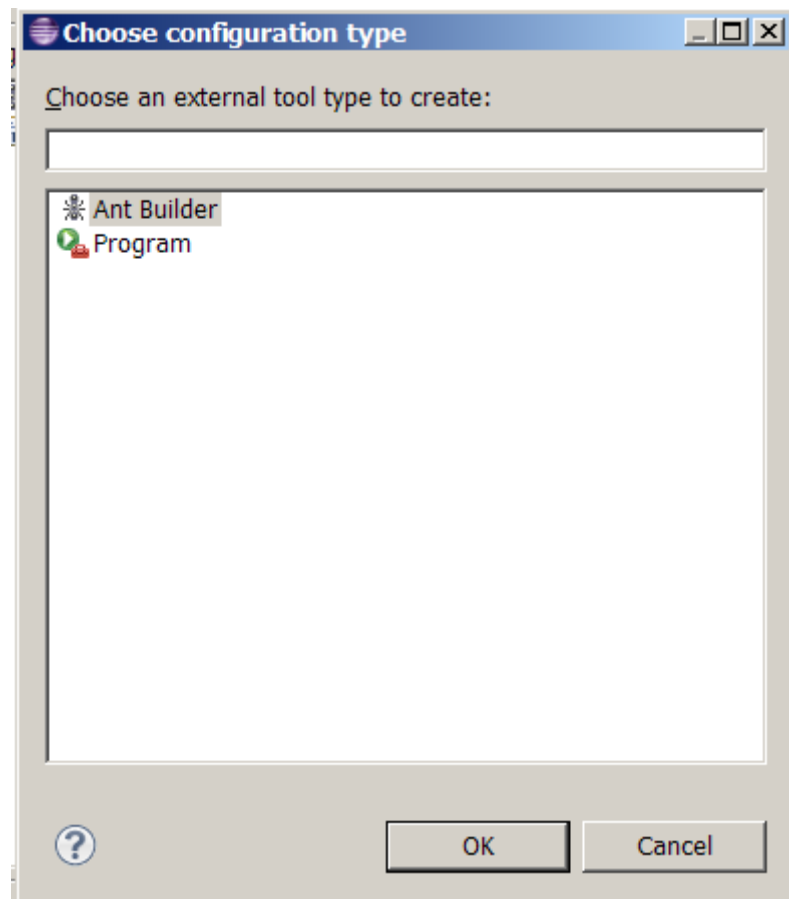
➤ 建立 hadoop 工程

在 Eclipse 中，点击 File 菜单的 New->Java project，在打开的界面中输入以下信息：

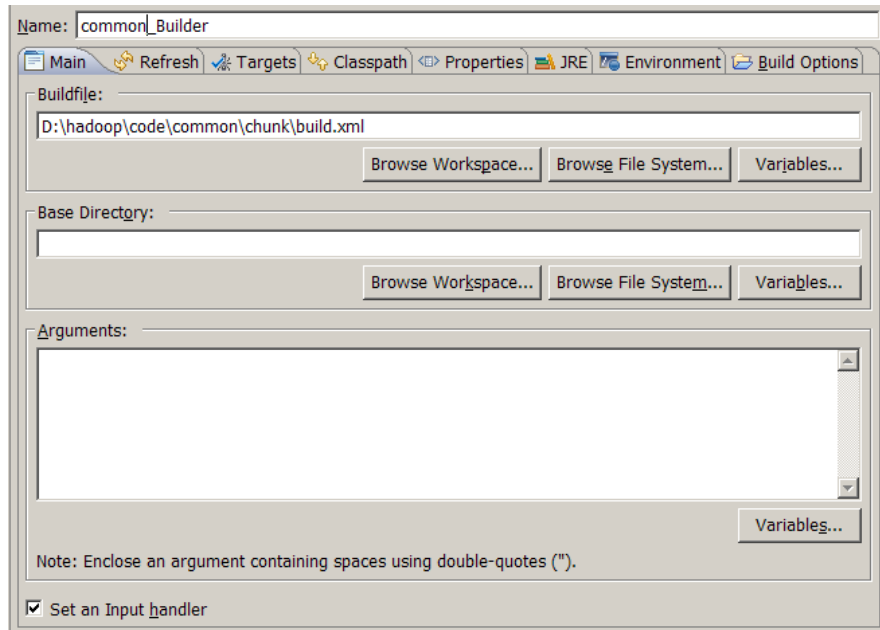
然后点击 **Finish**，即可导入我们已经 checkout 的 Hadoop 基础组件代码。

导入完成后，你会看见左侧工程上会有很多小红叉，这是因为 Hadoop 是需要用 Ant 进行编译，而不是 java，所以我们要配置启用 Ant 编译：

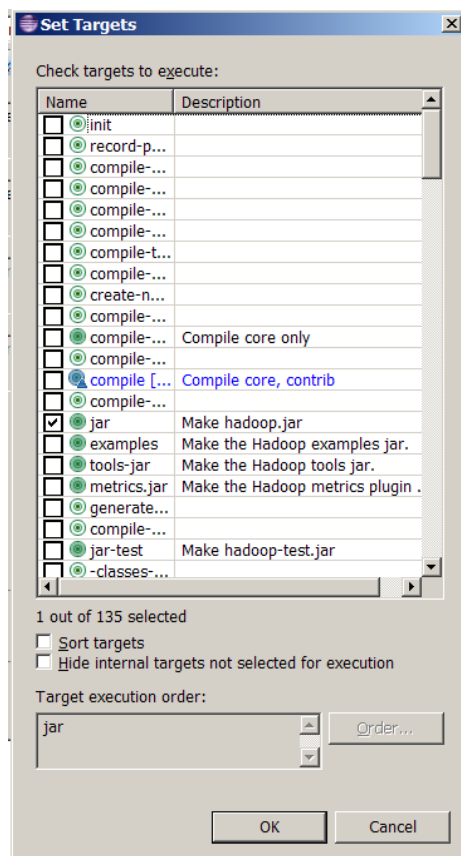
- 左键点击 **common** 工程，选择 **Properties**
- 选择左侧树形列表的 **Builders**
- 点击右侧的 **New** 按钮，在弹出的对话框中选择 **Ant Builder**，然后点击 **OK**



- 在弹出的对话框中，将 **Name** 输入为：**common_builder**；并点击 **Browser File System** 按钮，选择 **D:\hadoop\code\common\chunk\build.xml** 文件。

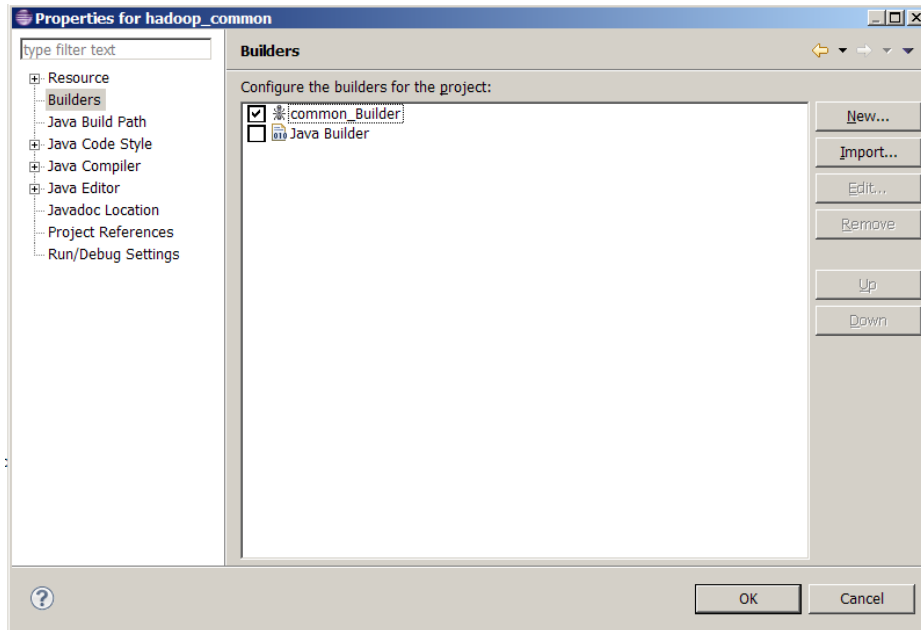


在点击上面的 Target 标签页，在第二项 Manual Build 右侧点击 Set Targets 按钮，将原来的勾选去掉，选择 jar 选项，如下图：

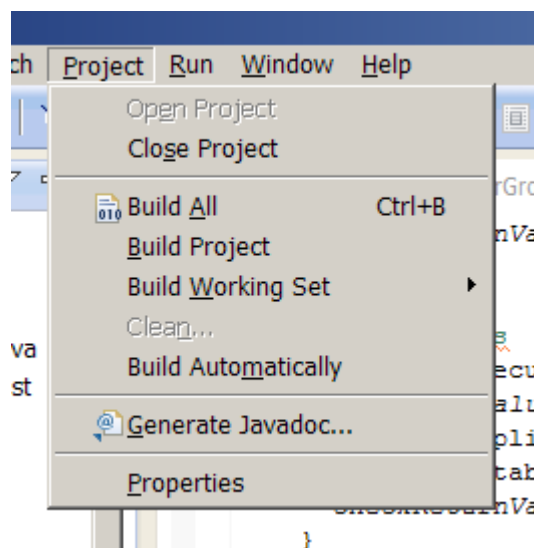


然后点击 OK 按钮

- 再回到对话框中再点击 OK 按钮，即可保存 common_Builder 编译器配置，然后将原来的 Java Building 下移并去掉勾选，如下图后点击 OK 即可：



回到 eclipse 视图，点击菜单项 Project，去掉 Build Automatically 选项。



➤ 修改 Hadoop 代码

按照需要对 Hadoop 工程中的代码文件进行修改。

➤ 编译 hadoop 程序

在菜单项 windows->preferences->ant-> Runtime->Classpath 中，点击 Add Jars，添加 jdk 目录下的\lib\tools.jar。

然后点击菜单项 Project->Build project，等待编译完成。

➤ 使用编译后的代码

编译完成后，会看到 Console 输出中提示：

[jar] Building jar:

D:\hadoop\code\common\chunk\build\hadoop-core-1.0.4-SNAPSHOT.jar

这就说明编译成功了，我们已经得到了新的 `hadoop jar` 文件。将这个 `jar` 文件拷贝到 `D:\hadoop\deploy\hadoop-1.0.3` 目录下，并重命名为 `hadoop-tools-1.0.3.jar`。

然后进入 `cygwin` 环境，重新 `stop-all.sh` 和 `start-all.sh` 启动 `hadoop`，这样就使用了新的编译代码了。

至此，我们就成功建立了 `Cygwin` 环境下的 `Hadoop 1.0.3` 的运行和开发环境，后面就可以按照自己的想法使用和修改 `Hadoop` 了。