

Jenkins 多节点简介

简介.....	1
新建从节点	2
配置从节点	2
Name	3
Description	3
# of executors.....	3
Labels.....	3
Usage	3
Launch method	4
Availability	4
Node Properties	5
Environment variables	5
Tool locations	5
Save	5
启动从节点	6
Have master launch slave agent via ssh.....	6
Have master launch slave agent on Windows	7
Write your own script to launch Jenkins slaves	8
Launch slave agent via Java Web Start	8
多节点构建.....	13
并发构建.....	16

简介

Jenkins 支持“主/从”模式，可以将构建任务分配到不同的从节点中，使得只需安装一个 jenkins 就可以管理多个构建任务，或者为构建或者测试提供不同的环境。

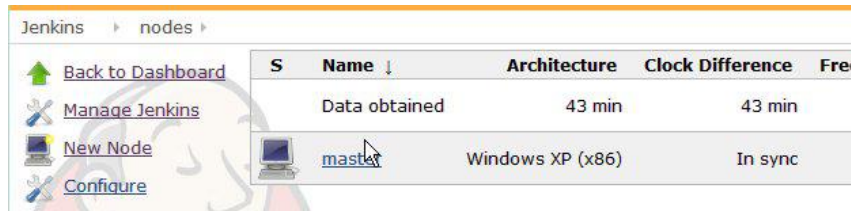
主节点是安装有 jenkins 的机器，如果不使用“主从”模式，则主节点就是一切。即使使用了“主从”模式，主节点同样重要。它会响应所有的 http 请求，自身也能执行构建任务。

从节点是那些为主节点执行构建任务的计算机。Jenkins 在从节点上运行一个叫做“从代理”的独立任务。换句话说，没有必要在从节点上安装完整的 jenkins。现在有很多的方法可以启动从代理，但无论如何，最后主从节点都需要建立一个双向的字节流，例如说 TCP/IP 套接字。

当从节点向主节点注册了之后，主节点就会为从节点分配构建任务。具体的分配策略取决于每个工程的具体配置。有些任务可能一直在某些从节点上运行，有些可能随机地在所有从节点中运行。这些主从节点之间的交互对于通过网页访问 jenkins 的用户来说都是透明的。用户仍然可以查看 javadoc 的结果，从主节点中下载构建结果，但不会意识到这些构建任务是由从节点完成的。换句话说，主节点可以认为是整个构建工厂的门户。

新建从节点

1 Manage Jenkins > Manage Nodes

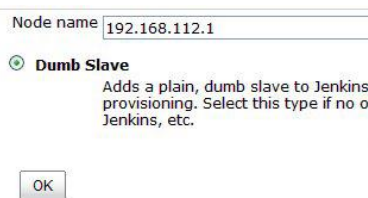


The screenshot shows the Jenkins 'Manage Nodes' page. On the left, there are links: 'Back to Dashboard', 'Manage Jenkins', 'New Node', and 'Configure'. The main table lists nodes with columns: 'S', 'Name', 'Architecture', 'Clock Difference', and 'Free Space'. One node named 'master' is listed with 'Windows XP (x86)' architecture and 'In sync' clock difference.

S	Name ↓	Architecture	Clock Difference	Free Space
	Data obtained	43 min	43 min	
	master	Windows XP (x86)	In sync	

2 点击“New Node”，输入节点名称，可以是 IP 地址或者机器名称，或其他，但要唯一。

3 选中“Dumb Slave”，点击“OK”



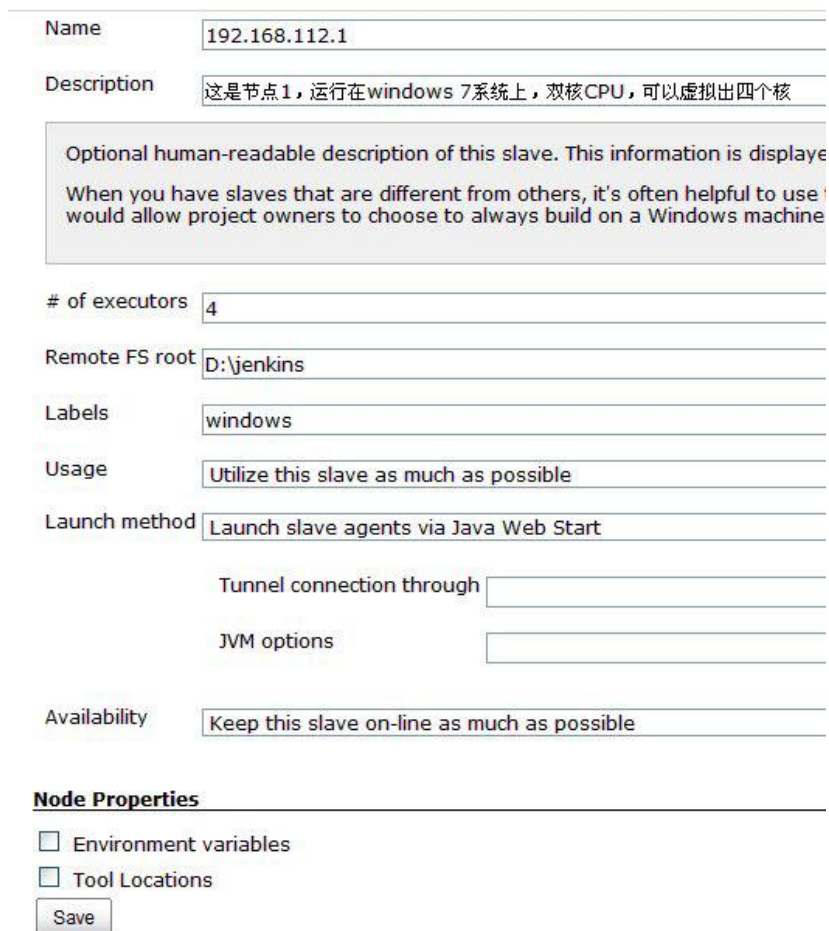
The screenshot shows the 'New Node' form in Jenkins. The 'Node name' field contains '192.168.112.1'. The 'Dumb Slave' radio button is selected. Below it, a description reads: 'Adds a plain, dumb slave to Jenkins provisioning. Select this type if no other type is appropriate for Jenkins, etc.' An 'OK' button is at the bottom.

Node name: 192.168.112.1

☒ **Dumb Slave**
Adds a plain, dumb slave to Jenkins provisioning. Select this type if no other type is appropriate for Jenkins, etc.

OK

配置从节点



The screenshot shows the 'Configure Slave' form in Jenkins. The 'Name' field contains '192.168.112.1'. The 'Description' field contains '这是节点1，运行在windows 7系统上，双核CPU，可以虚拟出四个核'. Below the description is a text box with the following text: 'Optional human-readable description of this slave. This information is displayed on the Jenkins dashboard. When you have slaves that are different from others, it's often helpful to use a descriptive label. This would allow project owners to choose to always build on a Windows machine'. The '# of executors' field contains '4'. The 'Remote FS root' field contains 'D:\jenkins'. The 'Labels' field contains 'windows'. The 'Usage' field contains 'Utilize this slave as much as possible'. The 'Launch method' field contains 'Launch slave agents via Java Web Start'. The 'Tunnel connection through' field is empty. The 'JVM options' field is empty. The 'Availability' field contains 'Keep this slave on-line as much as possible'. At the bottom, there are checkboxes for 'Environment variables' and 'Tool Locations', and a 'Save' button.

Name: 192.168.112.1

Description: 这是节点1，运行在windows 7系统上，双核CPU，可以虚拟出四个核

Optional human-readable description of this slave. This information is displayed on the Jenkins dashboard. When you have slaves that are different from others, it's often helpful to use a descriptive label. This would allow project owners to choose to always build on a Windows machine

of executors: 4

Remote FS root: D:\jenkins

Labels: windows

Usage: Utilize this slave as much as possible

Launch method: Launch slave agents via Java Web Start

Tunnel connection through:

JVM options:

Availability: Keep this slave on-line as much as possible

Node Properties

☐ Environment variables

☐ Tool Locations

Save

Name

slave 节点显示名字

Description

对该节点进行描述，这个信息会显示在 jenkins 工程配置屏幕上。当某个从节点和其他从节点有区别时，使用这个属性来解释怎么不同是非常方便的。例如，这里输入“windows slave”可以提示创建工程的用户这个从节点是运行在 windows 系统上的，因为有时候某些工程是特定需要 windows 系统的。

of executors

这个值控制着 Jenkins 并发构建的数量。因此这个值会影响 Jenkins 系统的负载压力。使用处理器个数作为其值会是比较好的选择。增大这个值会使每个构建的运行时间更长，但是这能够增大整体的构建数量，因为当一个项目在等待 I/O 时它允许 CPU 去构建另一个项目。r 控制 jenkins 在此 slave 上并行执行的最大数量。一般设置为此 slave 上 cpu 的核数。当 jenkins 在主从模式的时候，也可以把 master 的这个值设为 0 从而使 master 上不做任何构建，slave 上不允许 executor 数为 0，但是可以在从节点状态设置里使从节点暂时不能执行构建，从而不会被分配到任务。

Remote FS root

slave 节点默认的 workspace 目录，在界面上配置后会在 slave 机器上自动生成该文件夹路径。存放被解析文件，包括代码、配置的目录。应使用绝对路径，在 linux 中可以使用“/var/jenkins”，在 windows 中可以使用“D:\jenkins”等。这个路径应该是从节点自己的，在正常情况下这个路径对主节点可以不可见。

Labels

设置 slave 唯一的标识，在 job 中输入这个 Labels，说明这个 job 在这个 slave 机器上做分布式运行。用来将多个从节点进行逻辑分组。使用空格将标号隔开，“regression java6”，将给从节点分配两个标号“regression”和“java6”。例如如果你有很多 windows 从节点并且一些 jobs 一定要运行在 windows 系统上，你可以将所有 windows 从节点分配一个 label “windows”，然后在任务配置中使用“windows”，这样任务就可以运行在任何 windows 从节点上。

Usage

控制 Jenkins 如何在这台机器上安排构建。

Utilize this slave as much as possible(尽可能使用这个节点)

这是默认和常用的设置。在这种模式下,Jenkins 会尽可能的使用这个节点. 任何时候如果一个构建能使用这个节点构建, 那么 Jenkins 就会使用它.

Leave this machine for tied jobs only(只分配给特定的 jobs)

这种模式下,Jenkins 只会构建那些分配到这台机器的 Job. 这允许一个节点专门保留给某种类型的 Job. 例如, 在 Jenkins 上连续的执行测试, 你可以设置执行者数量为 1, 那么同一时间就只会会有一个构建, 一个实行者不会阻止其它构建, 其它构建会在另外的节点运行.

Launch method

控制 jenkins 怎么开始这个 slave:

Launch slave agents on Unix machines via SSH

master 通过 ssh 来启动 slave

Launch slave agents via Java Web Start

通过 JNLP 来启动一个代理。这种启动方式初始化是由 slave 自己完成的, 因此 master 可以不访问 slave 的 IP (例如被防火墙阻隔)。

Launch slave via execution of command on the Master

通过在 master 中执行命令了来启动 slave, 当 master 可以通过 ssh/rsh 在 slave 中执行远程命令可以使用这中方式。

Let Jenkins control this Windows slave as a Windows service

master 通过 WMI+DCOM 来启动 windows slave, 适合于启动 windows 从节点。

Availability

控制 jenkins 启动和停止 slave

Keep this slave on-line as much as possible

默认选项, 尽可能多的在线。如果 jenkins 可以不通过用户操作而启动从节点, 那么它会周期性地重启不可用的从节点, 从而保证从节点不会掉线。

Take this slave on-line according to a schedule

根据调度启动这个从节点

Take this slave on-line and off-line at specific times

在这种模式下，如果 jenkins 可以不通过用户操作而启动从节点，而且有没有正在执行的任务满足下面的条件，它会启动从节点：

- 1 至少在特定的启动周期的等待队列中
- 2 可以在这个从节点执行

满足下面条件时从节点将 off-line：

- 1 没有激活任务在这个从节点运行
- 2 在特定的空闲周期里

Node Properties

Environment variables

可以在所有运行于此从节点的任务中使用的环境变量，键值对，同名会覆盖全局环境变量。可以在 jenkins 配置中通过 `$key` 或 `${key}` 的形式使用。

Tool locations

你可以在这个从节点中设定某些工具的位置，会覆盖全局设置。

Save

保存新建的节点，offline，未知从节点结构

The screenshot shows the Jenkins 'nodes' page. On the left, there are navigation links: 'Back to Dashboard', 'Manage Jenkins', 'New Node', and 'Configure'. Below these are sections for 'Build Queue' (showing 'No builds in the queue.') and 'Build Executor Status' (showing 'master' as idle). The main table lists nodes:

S	Name ↓	Architecture
	Data obtained	30 min
	192.168.112.1	
	master	Windows XP (x86)

In the 'Build Executor Status' table, the node '192.168.112.1' is listed with a status of '(offline)'. Red circles highlight the 'offline' status in the table and the '192.168.112.1' node name in the main table.

启动从节点

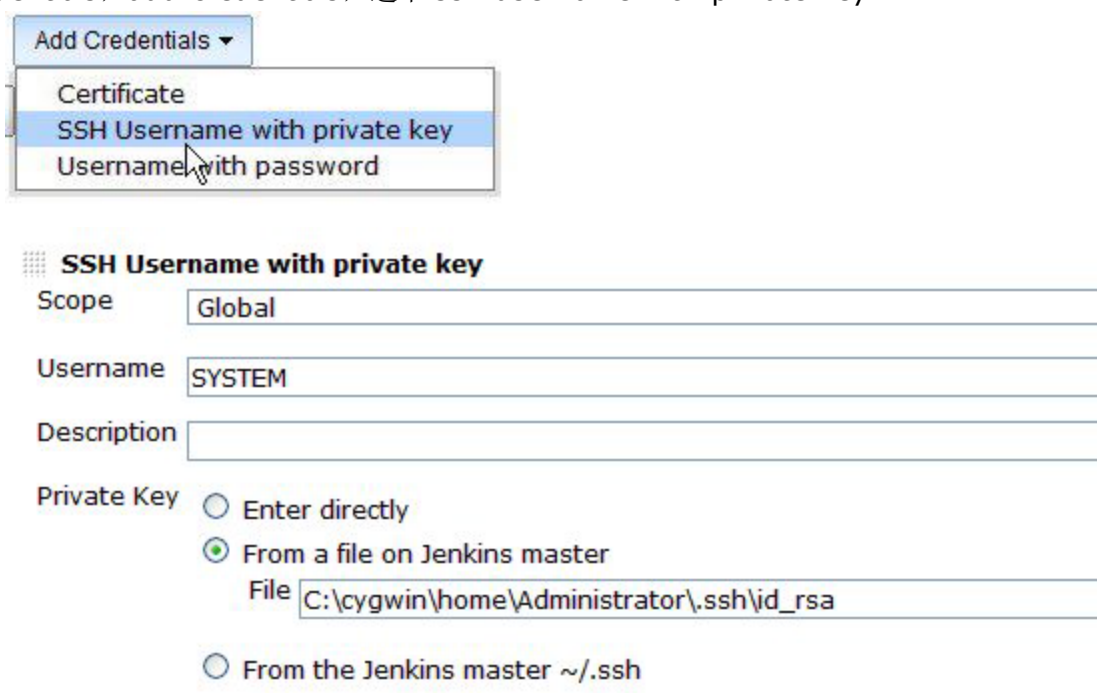
可以在这个网址获取一些帮助

<https://wiki.jenkins-ci.org/display/JENKINS/Distributed+builds>

Have master launch slave agent via ssh

Jenkins 内置有 ssh 客户端实现,可以用来与远程的 sshd 通信,从而启动 slave agent。这是对*unix 系统的 slave 最方便的方法,因为*unix 系统一般默认安装有 sshd。在创建 ssh 连接的 slave 的时候,你需要提供 slave 的 host 名字,用户名和 ssh 证书。创建 public/private keys,然后将 public key 拷贝到 slave 的 ~/.ssh/authorized_keys 中,将 private key 保存到 master 上某 ppk 文件中。jenkins 将会自动地完成其他的配置工作,例如 copy slave agent 的 binary,启动和停止 slave。但是你的 job 运行所依赖其他的项目需要你自己设置。

使用这个方法要有 credentials,在 jenkins 中点击 manage jenkins, manage credentials, add credentials, 选中 ssh username with private key



The image shows the Jenkins interface for adding credentials. At the top, there is a blue button labeled 'Add Credentials' with a dropdown arrow. The dropdown menu is open, showing three options: 'Certificate', 'SSH Username with private key' (which is highlighted in blue), and 'Username with password'. Below this, the configuration form for 'SSH Username with private key' is visible. It includes a 'Scope' field set to 'Global', a 'Username' field set to 'SYSTEM', and an empty 'Description' field. Under the 'Private Key' section, there are three radio buttons: 'Enter directly' (unselected), 'From a file on Jenkins master' (selected), and 'From the Jenkins master ~/.ssh' (unselected). Below the selected option, there is a 'File' field containing the path 'C:\cygwin\home\Administrator\.ssh\id_rsa'.

在 windows 中生成 ssh key 可以使用 cygwin, 命令如下:

Ssh-keygen -t rsa 回车

Enter file in witch...回车

Enter passphrase...回车

Enter same passphrase again:回车

```
Administrator@yhluo ~
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/Administrator/.ssh/id_rsa):
Created directory '/home/Administrator/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/Administrator/.ssh/id_rsa.
Your public key has been saved in /home/Administrator/.ssh/id_rsa.pub.
The key fingerprint is:
95:c0:de:58:9e:70:b4:7a:bd:79:9b:31:47:1f:66:86 Administrator@yhluo
The key's randomart image is:
+--[ RSA 2048 ]-----+
  ....
  o..oo
  . B+.
  oo+.
  S . .E =.
  . o+.o
  . o + o
  . +
  . o
+-----+

Administrator@yhluo ~
$
```

不管是 Enter directly, From a file on Jenkins master 还是 From the jenkins master ~/.ssh, 我试了一下, 可是都失败了, 没有检查出原因。

Have master launch slave agent on Windows

对于 Windows 的 Slave, Jenkins 可以使用 Windows2000 及以后内置的远程管理功能 (WMI+DCOM), 你只需要提供对 slave 有管理员访问权限的用户名和密码, jenkins 将远程地创建 windows service 然后远程地启动和停止他们。对于 windows 的系统, 这是最方便的方法, 但是此方法不允许运行有显示交互的 GUI 程序。

Launch method Let Jenkins control this Windows slave as a Windows service

This launch method relies on DCOM and is often associated with [subtle problems](#). Consider using **Launch slave agents using Java Web Start** instead, which also permits installation as a Windows service but is generally considered more reliable.

Administrator user name

Password

Host

Run service as Use Local System User

Path to java executable

JVM options

这里也尝试了好多次, 关闭防火墙, 启用 DCOM 服务等, 都还是不行, 下面是一些可能的错误。

<https://wiki.jenkins-ci.org/display/JENKINS/Windows+slaves+fail+to+start+via+DCOM>

Write your own script to launch Jenkins slaves

如果上面成套的方法不够灵活，你可以实现自己的脚本来启动 **slave**。你需要将启动脚本放到 **master**，然后告诉 **jenkins master** 在需要的时候调用此脚本来启动 **slave**。

典型地，你的脚本使用远程程序执行机制，例如 **SSH**, **RSH**, 或类似的方法（在 **windows**，可以通过 **cygwin** 或 **psexec** 来完成），

在脚本的最后需要执行类似 `java -jar slave.jar` 来启动 **slave**。**slave.jar** 可以从 `http://yourjenkinsserver:port/jnlpjars/slave.jar` 下载，也可以在脚本的开始先下载此 **slave.jar** 从而保证 **slave.jar** 正确的版本。但是如果使用 **ssh slave plugin** 的话，此 **plugin** 将自动地更新 **slave.jar**。

Launch slave agent via Java Web Start

在从节点中打开 **master** 的 **jenkins server URL**，**Manage Jenkins**，**Manage Nodes**，点击刚才新建的从节点：



Jenkins

Jenkins » nodes » 192.168.112.1

[Back to List](#)
[Status](#)
[Delete Slave](#)
[Configure](#)
[Build History](#)
[Load Statistics](#)
[Script Console](#)
[Log](#)
[System Information](#)

Slave 192.168.112.1 (这是节点1, 运行在windows 7系统上, 双核CPU, 可以虚拟出四个核)

Connect slave to Jenkins one of these ways:

- [Launch](#) Launch agent from browser on slave
- Run from slave command line:

```
java -jar _slave.jar -jnlpUrl http://yhluo.iflytek.com:8080/jenkins/computer/192.168.112.1/slave-agent.jnlp -secret e7c9feefd000e5527fc39310bd6c1ff7f9665359289b564849335ea6e855052
```

Created by [Yuanhao Luo](#)
Labels: [windows](#)

Build Executor Status

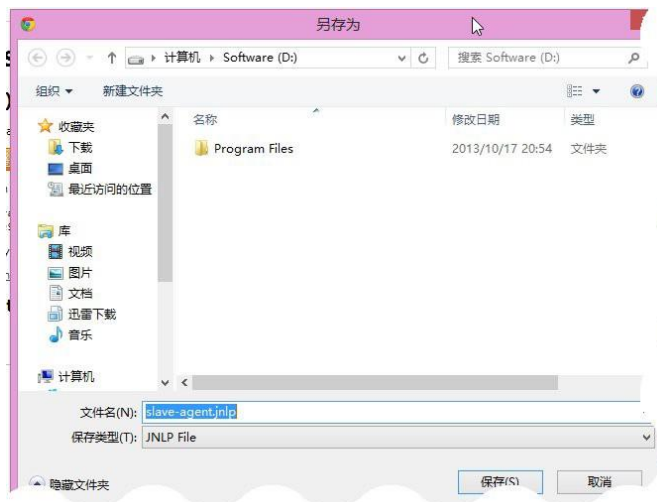
#	Status

Projects tied to 192.168.112.1

None

如上图所示，没有启动时状态显示为空，还没有任何 **project** 绑定到这个节点。下面有两种方法启动 **slave**

第一种：点击 **Launch**，弹出保存文件对话框，将 **slave-agent.jnlp** 保存在 **D** 盘



双击打开 slave-agent.jnlp，使用 java web start launcher 打开，如果 java 版本过低，可能需要先更新 jre 和 jdk。若弹出安全警告，选择仍然运行即可。



运行后会弹出如下图所示对话框，可以注意到 Build Executor status 下面已经有 4 个 idle，表明已经启动，且有 4 个执行器处于空闲状态。如果关闭弹出的对话框，连接也会终止。

Jenkins nodes 192.168.112.1

Back to List

Status

Jenkins slave ...

File

Connected

Log

System Information

Build Executor Status

#	Status
1	Idle
2	Idle
3	Idle
4	Idle

Slave 192.168.112.1 (四个核)

Connect slave to Jenkins one of these ways:

- Launch Launch agent from browser
- Run from slave command line:

```
java -jar slave.jar -jnlpUrl http://e7c9feefdf000e5527fc39310bd6c1ff7f5
```

Created by Yuanhao Luo

Labels: windows

Projects tied to 192.168.112.1

None

再在 master 上查看，如下图，显示 slave 已经连接且有 4 个执行器

Jenkins > nodes > 192.168.112.1

[Back to List](#)
[Status](#)
[Build History](#)
[Load Statistics](#)

Slave 192.168.112.1 (这是节点1，运行在windows 7系统上，双核CPU，可以虚拟出四个核)

Connected via JNLP agent.
Created by [Yuanhao Luo](#)
Labels: [windows](#)

Build Executor Status

#	Status
1	Idle
2	Idle
3	Idle
4	Idle

Projects tied to 192.168.112.1
None

如果希望 jenkins 作为一个 windows 服务运行在 slave 上，可以在 slave 中弹出的窗口点击“File”，“Install as windows service”。



弹出如下提示：这将安装从代理为 windows 服务，这样 jenkins 从节点将随着机器的启动而启动。如果不安装为服务的话，每次都需要在 slave 中双击 salve-agent.jnlp 启动 jenkins。这里点击“确定”安装。



弹出 jenkins 根目录不存在的提示

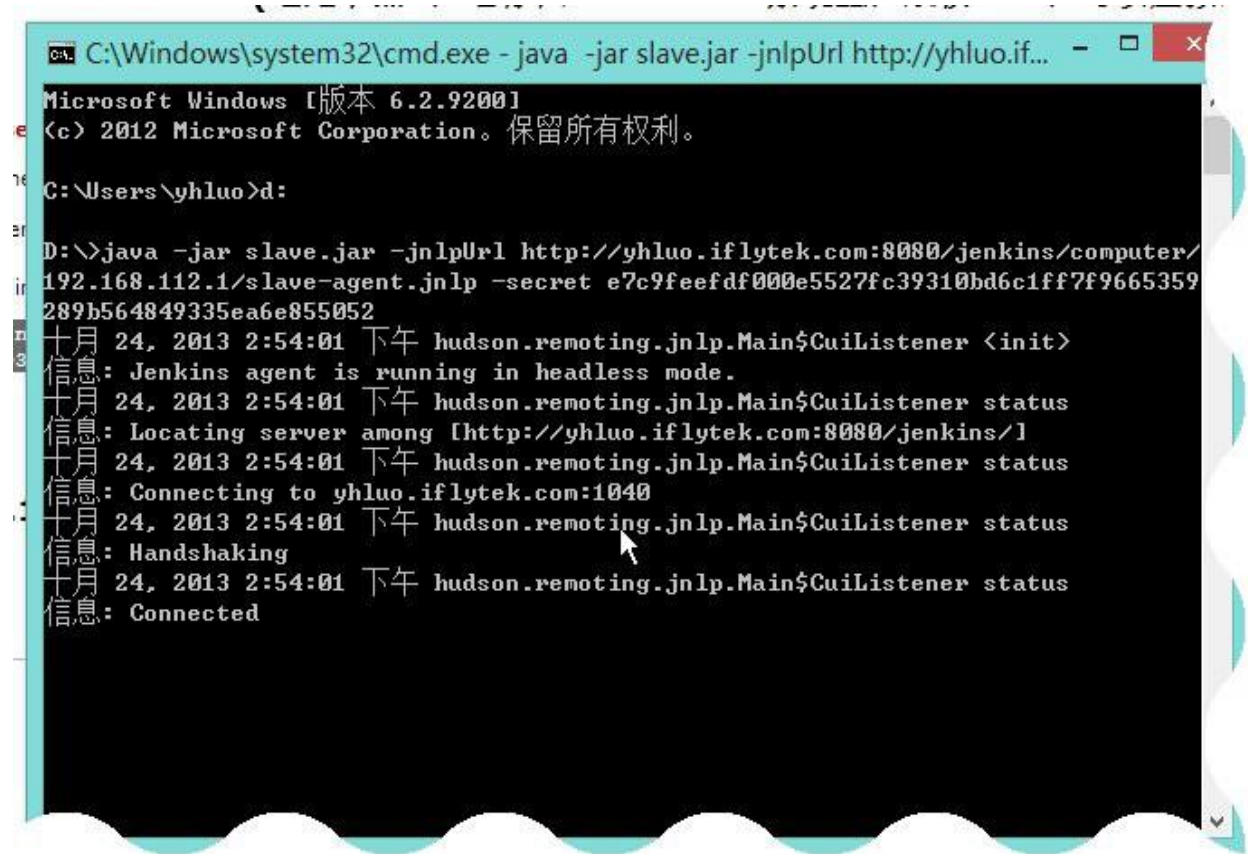
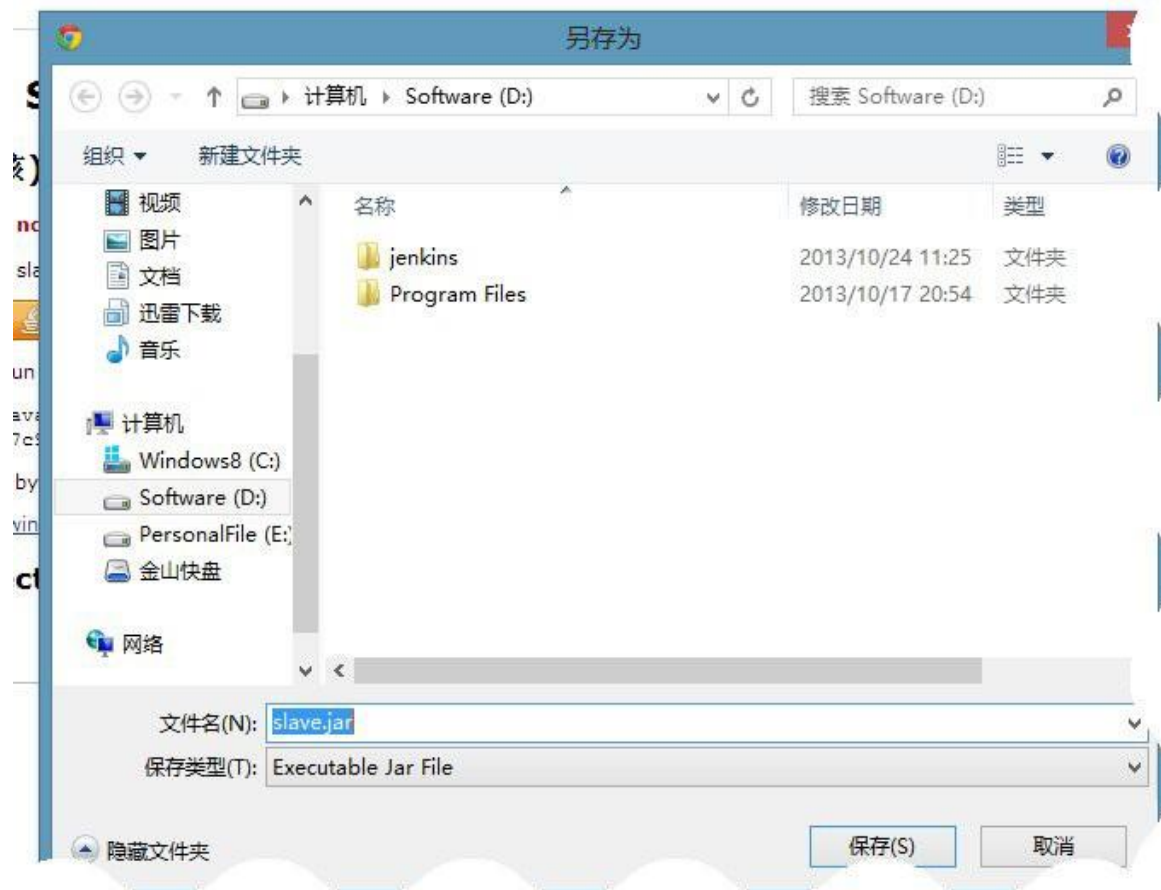


在 D 盘新建文件夹 jenkins，再次点击“File”，“Install as windows service”。然后提示要安装“.NET FRAMEWORK2 3”啥的，太麻烦，还是每次双击打开吧。

第二种，点击“slave.jar”，弹出保存对话框，将 slave.jar 保存到 D 盘。打开 cmd，cd 到 D 盘，输入提示的命令：

```
java -jar slave.jar -jnlpUrl  
http://yhluo.iflytek.com:8080/jenkins/computer/192.168.112.1/slave-agent.jnlp -secret  
e7c9feefdf000e5527fc39310bd6c1ff7f9665359289b564849335ea6e855052
```

也可以启动 slave 并连接。



如果要断开连接，只需要点击“disconnect”：

Jenkins ▸ nodes ▸ 192.168.112.1

 [Back to List](#)

 [Status](#)

 [Delete Slave](#)

 [Configure](#)

 [Build History](#)

 [Load Statistics](#)

 [Script Console](#)

 [Log](#)

 [System Information](#)

 [Disconnect](#)

1

C

2

C

3

L

4

F

5

N

Build Executor Status

#	Status
1	Idle
2	Idle
3	Idle
4	Idle

它会提示为什么要断开连接，在下面的文本框中输入信息点击“yes”即可断开连接。

192.168.112.1

Are you sure about disconnecting?

Yes

You can optionally explain why you are taking this node offline, so that others can see why:

我要重启系统，临时关闭这个节点

多节点构建

连续点击 4 次构建，可以看到在 master 上执行了两次构建，在 slave 上执行了两次构建。如果构建次数多于所有 executors 的总数，总会将多余的放入等待队列。

Build Queue

No builds in the queue.

Build Executor Status

#	Status
master	
1	Building HandWriteEffectPluginDefault #52
2	Building HandWriteEffectPluginDefault #53
192.168.112.1	
1	Building HandWriteEffectPluginDefault #55
2	Idle
3	Building HandWriteEffectPluginDefault #54
4	Idle

Build History (trend)

#55

Oct 24, 2013 3:04:32 PM

#54

Oct 24, 2013 3:04:27 PM

#53

Oct 24, 2013 3:04:23 PM

#52

Oct 24, 2013 3:04:19 PM

并发的 4 次构建中，只有第一次成功，其余三次失败。查看构建日志第 53 号构建错误如下：

```
updateTable:
[sql] Executing resource: D:\database\AfterBuild.sql
[sql] Failed to execute: update AIMBUILD set IsBuild="已构建" where id in (select id from VIEW_MAX_ID)
```

BUILD FAILED

是数据库加锁后不能共享写，即竞争资源导致构建失败。

第 54 和 55 号构建是在 slave 上的，错误原因都是 svn 授权失败。



Console Output

```
Started by user Yuanhao Luo
Building remotely on 192.168.112.1 in workspace D:\jenkins\workspace\HandWriteEffectPluginDefault@2
Checking out a fresh workspace because there's no workspace at D:\jenkins\workspace\HandWriteEffectPluginDefault@2
Cleaning local Directory .
Checking out https://yhluo.iflytek.com/svn/HandWriteEffectPlugin/branches/HandWrite@2 at revision 2
ERROR: Failed to check out https://yhluo.iflytek.com/svn/HandWriteEffectPlugin/branches/HandWrite@2
org.tmatesoft.svn.core.SVNCancelException: svn: E200015: OPTIONS /svn/HandWriteEffectPlugin/branches/HandWrite failed
    at org.tmatesoft.svn.core.internal.io.dav.http.HTTPConnection.request(HTTPConnection.java:384)
    at java.lang.Thread.run(Thread.java:750)
Caused by: org.tmatesoft.svn.core.SVNCancelException: svn: E200015: No credential to try. Authentication failed
    at org.tmatesoft.svn.core.internal.wc.SVNErrorManager.cancel(SVNErrorManager.java:37)
```

解决方法：

在配置 job 设定 SVN URL 的时候，点击后面的蓝色问号，会提示一些授权的信息。如下图，点击 “This link”，弹出一个新的页面，在里面填入 URL，选中

“Username/password authentication” 输入 SVN 中的 username 和 password。保存，重新构建。如下图，授权成功，正在构建。

Subversion
Modules

Repository URL

https://yhluo.iflytek.com/svn/HandWriteEffectPlugin/branches/HandWrite

Specify the subversion repository URL to check out, such as "http://svn.apache.org/repos/asf/ant/". You can also add "@NNN" at the end of the URL to check out a specific revision number, if that's desirable. This works for [Subversion Revision Keywords and Dates](#) like e.g. "HEAD", too.
When you enter URL, Jenkins automatically checks if Jenkins can connect to it. If access requires authentication, it will ask you the necessary credential. If you already have a working credential but would like to change it for other reasons, click [this link](#) and specify different credential.
During the build, revision number of the module that was checked out is available through the environment variable `SVN_REVISION`, provided that you are only checking out one module. If you have multiple modules checked out, use the [svnversion command](#). If you have multiple modules checked out, you can use the [svnversion command](#) to get the revision information, or you can use the `SVN_REVISION_<n>` environment variables, where <n> is a 1-based index matching the locations configured. The URLs are available through similar `SVN_URL_<n>` environment variables.



Subversion Authentication

Enter the authentication information needed to connect to the Subversion repository. This information will be stored in Jenkins.

Repository URL

☒ Username/password authentication

User name

Password

☐ SSH public key authentication (svn+ssh)

☐ HTTPS client certificate

OK

Build Queue		
No builds in the queue.		
Build Executor Status		
#	Status	
<u>master</u>		
1	Building HandWriteEffectPluginDefault #61	
2	Building HandWriteEffectPluginDefault #62	
<u>192.168.112.1</u>		
1	Building HandWriteEffectPluginDefault #64	
2	Building HandWriteEffectPluginDefault #65	
3	Building HandWriteEffectPluginDefault #63	
4	Building HandWriteEffectPluginDefault #66	

之后构建失败，提示 ant.bat 脚本问题，原来是在 slave 上没有安装 ant，安装完后开始构建，又在编译的时候出现错误，报找不到包 android.content， android.os 等。可见 slave 上要有和 master 相同的环境配置，需要在本地有 ant， mysql， findbugs 等支持，且构建文件 build.xml 等资源路径也要相同。

Console Output

```
Started by user Yuanhao Luo
Building remotely on 192.168.112.1 in workspace D:\jenkins\workspace\HandWriteEffectPluginDefault
Updating https://yhluo.iflytek.com/svn/HandWriteEffectPlugin/branches/HandWrite at revision '2013-10-25T16:35:47.953 +0800'
At revision 2
no change for https://yhluo.iflytek.com/svn/HandWriteEffectPlugin/branches/HandWrite since the previous build
No emails were triggered.
[HandWriteEffectPluginDefault] $ cmd.exe /C "ant.bat -file build.xml -DSVNBRANCH=HandWrite -DSVNREVISION=2 && exit %%ERRORLEVEL%%"
'ant.bat' 不是内部或外部命令，也不是可运行的程序
或批处理文件。
Build step 'Invoke Ant' marked build as failure
[FINDBUGS] Skipping publisher since build result is FAILURE
Email was triggered for: Failure
Sending email for trigger: Failure
Sending email to: yhluo@iflytek.com
Finished: FAILURE
```

```
\HandwriteEffectManager.java:5: 错误: 程序包android.content不存在

\HandwriteEffectManager.java:6: 错误: 程序包android.graphics不存在

\HandwriteEffectManager.java:7: 错误: 程序包android.os不存在

\HandwriteEffectManager.java:8: 错误: 程序包android.text不存在

\HandwriteEffectManager.java:9: 错误: 程序包android.view不存在

\HandwriteEffectManager.java:10: 错误: 程序包android.view不存在

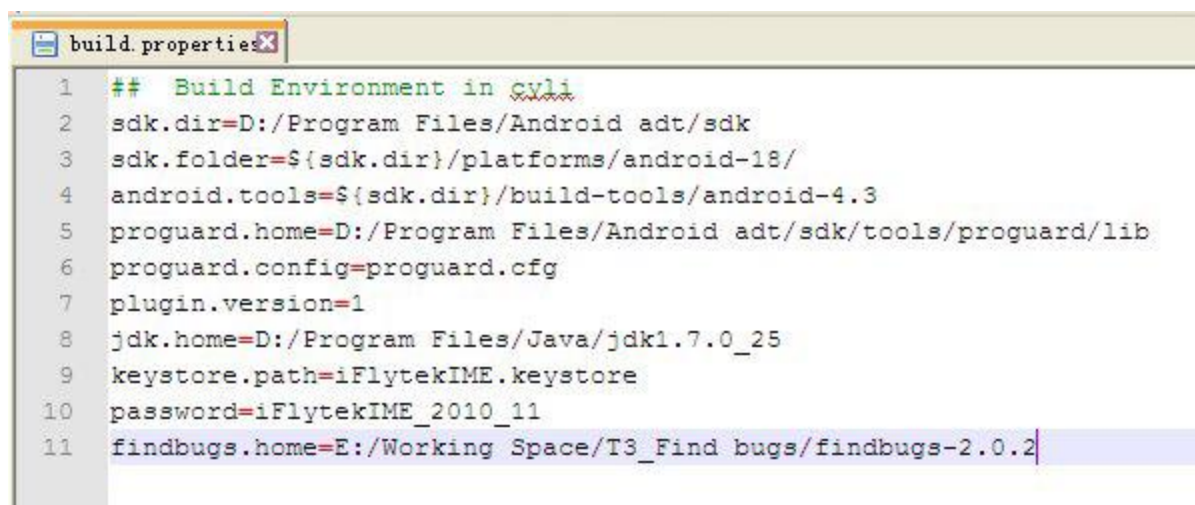
\control\EffectSettingViewManager.java:3: 错误: 程序包android.app不存在

\control\EffectSettingViewManager.java:4: 错误: 程序包android.app不存在

\control\EffectSettingViewManager.java:5: 错误: 程序包android.content不存在

\control\EffectSettingViewManager.java:6: 错误: 程序包android.content不存在

\control\EffectSettingViewManager.java:7: 错误: 程序包
```



```
1  ## Build Environment in Cyli
2  sdk.dir=D:/Program Files/Android adt/sdk
3  sdk.folder=${sdk.dir}/platforms/android-18/
4  android.tools=${sdk.dir}/build-tools/android-4.3
5  proguard.home=D:/Program Files/Android adt/sdk/tools/proguard/lib
6  proguard.config=proguard.cfg
7  plugin.version=1
8  jdk.home=D:/Program Files/Java/jdk1.7.0_25
9  keystore.path=iFlytekIME.keystore
10 password=iFlytekIME_2010_11
11 findbugs.home=E:/Working Space/T3_Find bugs/findbugs-2.0.2
```

在 build.properties 中有如上图配置，但 slave 本地没有。

并发构建

在配置 job 的时候，如下图所示，有一个 Execute concurrent builds if necessary 选项：

- ☐ Disable Build (No new builds will be executed)
- ☐ Execute concurrent builds if necessary
- ☐ Restrict where this project can be run

如果需要，执行并发构建，即同时执行几个构建任务。点击后面的蓝色问号，有一些解释：如果这个选项被选中，jenkins 会自动调度并并发执行几个构建（假设你有足够的执行器和构建申请）。这对于每次构建和测试需要花费很长时间，但是构建之间差异很小的任务非常有利，由于它不需要等待之前的构建先完成，能极大地降低总的时间消耗。这对于相互独立的参数化构建任务也非常有利。对于有些问题，例如多个构建之间如果会使用共同的资源，比如数据库，则允许并发构建也许会引起其它问题。另外，如果你使用定制的工作目录并且勾选了这个功能，除非足够的小心，否则很容易引起冲突。如果不是定制的工作目录，即使是在同一个节点上运行，jenkins 也会使用不同的工作目录使各个构建之间相互独立。Jenkins 通过给工作目录添加后缀，例如@num 来创建不同的目录。分隔符@可以在系统属性“ hudson.slaves.WorkspaceList ”中设置，例如在启动 jenkins 时添加参数“-Dhudson.slaves.WorkspaceList=-”，则会使用“-”作为分隔符。能同时执行多少个构建主要取决于系统设备，一般和 CPU 的核数相同。例如对于双核 CPU，一般能同时执行两个构建。

例如下图所示，最多能同时执行两个构建：

Build Queue	
No builds in the queue.	
Build Executor Status	
#	Status
1	Idle
2	Idle

执行后的结果如下图所示：可以看出 jenkins 通过在后面添加@num 生成不同的工作目录。

