

配置Ant环境

下载Ant: <http://ant.apache.org/bindownload.cgi>

在windows上应该选择zip压缩包，将zip压缩包解压到一个目录。

打开系统环境变量，在系统变量栏点击新建，变量名输入“ANT_HOME”，变量值为Ant的根目录，如“D:\Android\apache-ant-1.9.0”，注意不要带双引号。

在系统变量中找到Path变量，点击编辑，在变量值的最后添加“%ANT_HOME%\bin”，注意不要带双引号，并且要使用“;”和之前的变量值隔开。

打开一个cmd窗口，输入“ant”，如果显示一下信息，说明Ant的环境配置成功，如果显示：'ant' 不是内部或外部命令，也不是可运行的程序或批处理文件。则要检查一下路径是否有问题。

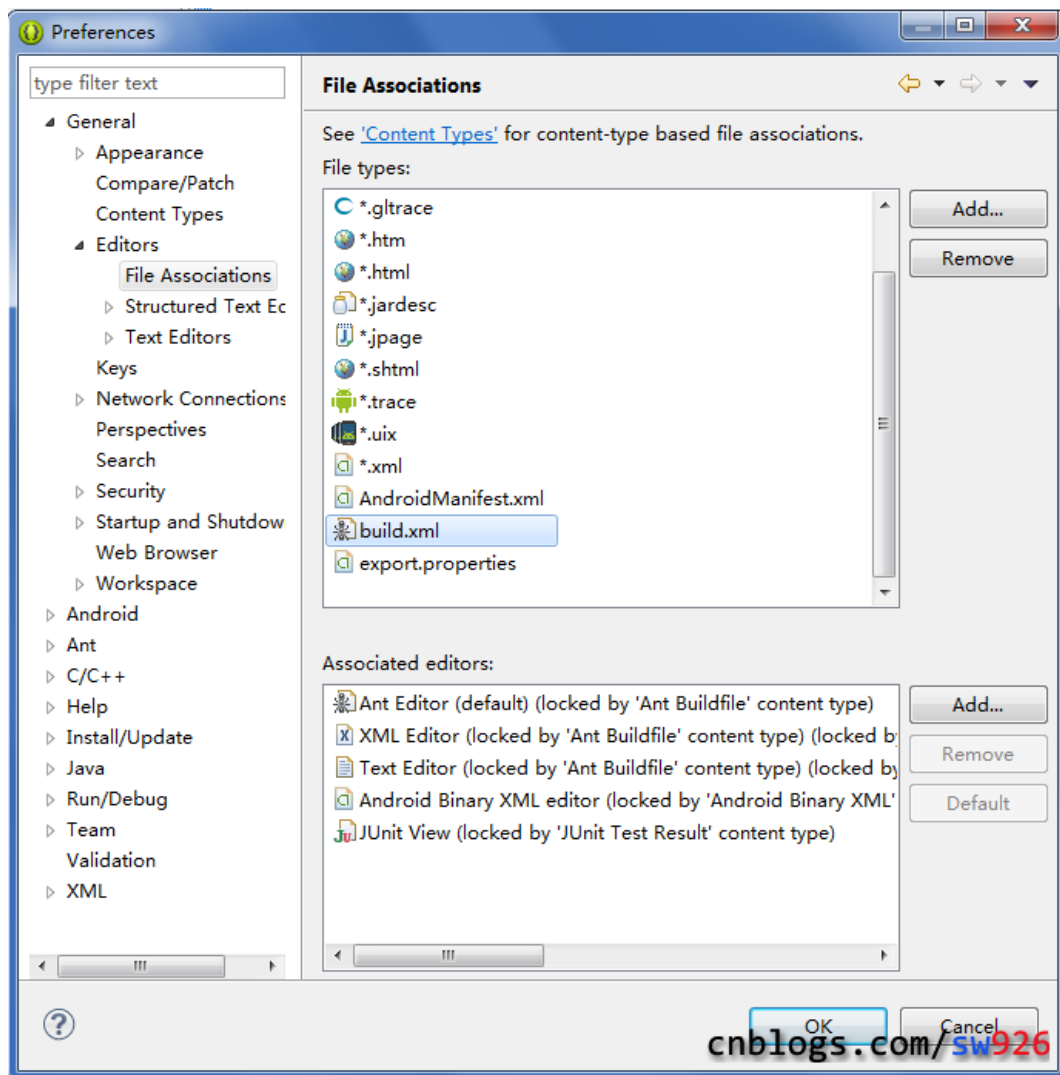
```
C:\Users\Sun>ant
Buildfile: build.xml does not exist!
Build failed
```

在Eclipse中配置Ant

在eclipse中使用Ant之前，为了使Ant的build.xml文件能够安装制定的格式进行缩进和高亮显示，并能够进行代码提示，首先要简单的设置一下。

打开Windows - Preferences，依次展开General，Editors，选中File Associations，点击Add...，在Add File Type对话框中输入build.xml，点击Ok。

接下来在File type：栏选中build.xml，在Associated Editor：栏选中Ant Editor，点击Default，build.xml的图标变成了一个小蚂蚁，配置完毕。



使用Ant编译Android的java代码和native代码

新建一个Android工程TestAnt，在工程的根目录下新建一个build.xml文件

输入以下内容：

```
<?xml version="1.0" encoding="UTF-8"?>
<project name="TestAnt" default="init">

    <target name="init">
        <fail message="Ant 1.7.0 or higher is required.">
            <condition>
                <not>
                    <antversion property="ant.version" atleast="1.7.0" />
                </not>
            </condition>
        </fail>
    </target>
```

```
</project>
```



打开cmd，切换到工程根目录，输入ant init

```
D:\Android\workspace\TestAnt>ant init
Buildfile: D:\Android\workspace\TestAnt\build.xml

init:

BUILD SUCCESSFUL
Total time: 1 second
```

编译成功，来解析这个build.xml:

```
<project name="TestAnt" default="init">
```

project是Ant工程的根节点，name属性是工程的名称，default是默认执行的target，init为默认的target，当我们输入Ant的时候和Ant init是一样的效果。



```
<target name="init">
  <fail message="Ant 1.7.0 or higher is required.">
    <condition>
      <not>
        <antversion property="ant.version" atleast="1.7.0" />
      </not>
    </condition>
  </fail>
</target>
```



target指定了要执行的操作，init是我们为这个target所起的名字，也可以是build，clean等等。在这个target中，执行的是检查Ant的版本，如个小于1.7.0的话会输出报错信息。

例如我们可以添加一个clean的target



```
<target name="clean">
  <echo message="Deleting temporary files..." />
  <delete dir="gen" />
  <delete dir="bin" />
  <delete dir="out" />
  <delete dir="obj" />
  <echo message="DONE (Deleting temporary files)" />
</target>
```



执行的操作是删除所有临时目录，在cmd窗口中输入Ant clean，这四个临时目录就会被删除，和在eclipse中执行清理是一个效果。

使用Ant编译Android工程

在SDK中，Google已经为我们写好了一个build.xml文件，就是sdk根目录\tools\ant\build.xml，所以我们只要把这个build.xml引入就可以编译Android工程了。

在这之前首先要新建一个local.properties文件，引入sdk和ndk的路径

输入一下内容：

sdk.dir=D:\\Android\\android-sdk

ndk.dir=D:\\Android\\android-ndk

分别为sdk和ndk的路径，要安装自己的实际路径进行配置。

在测试工程的build.xml中输入一下代码：

```
<?xml version="1.0" encoding="UTF-8"?>
<project name="TestAnt" default="release">
    <loadproperties srcFile="local.properties" />
    <loadproperties srcFile="project.properties" />

    <fail message="sdk.dir is missing. Make sure to generate local.properties using
'android update project'" unless="sdk.dir" />
    <fail message="ndk.dir is missing. Make sure to generate local.properties using
'android update project'" unless="ndk.dir" />

    <import file="${sdk.dir}/tools/ant/build.xml" />
</project>
```

在命令行中执行ant release或ant debug，就会执行对应的编译。

以上只是进行Java代码的编译，如何编译native代码呢，在NDK中Google可没有提够build.xml，这就需要我们自己实现。在新版NDK中，我们只要在命令行中切换到工程的根目录，然后执行

“D:\Android\android-ndk\ndk-build.cmd”（红色部分要替换成自己的路径）

就可以，例如，在TestAnt工程中添加native代码，参考<http://www.cnblogs.com/sw926/p/3232311.html>

执行ndk-build

```
D:\Android\workspace\TestAnt>D:\Android\android-ndk\ndk-build.cmd
Android NDK: WARNING: APP_PLATFORM android-14 is larger than android:minSdkVersi
on 8 in ./AndroidManifest.xml
"Compile++ thumb : TestAnt <= TestAnt.cpp
StaticLibrary   : libstdc++.a
SharedLibrary   : libTestAnt.so
Install         : libTestAnt.so => libs/armeabi/libTestAnt.so
```

所以我们只要在Ant中执行一个cmd命令就可以了，在build.xml中添加

```
<target name="native">
    <echo message="Building native libraries..." />
    <exec executable="${ndk.dir}/ndk-build.cmd" failonerror="true" />
</target>
```

```
<echo message="DONE (Building native libraries)" />
</target>
```

然后在命令行中执行ant native

```
D:\Android\workspace\TestAnt>ant native
Buildfile: D:\Android\workspace\TestAnt\build.xml

native:
    [echo] Building native libraries...
    [exec] Android NDK: WARNING: APP_PLATFORM android-14 is larger than android
:minSdkVersion 8 in ./AndroidManifest.xml
    [exec] Install      : libTestAnt.so => libs/armeabi/libTestAnt.so
    [echo] DONE (Building native libraries)

BUILD SUCCESSFUL
Total time: 1 second
```

执行了ndk-build。

以上介绍的都是使用Ant进行进步的Android编译操作，在此基础上可以实现更复杂的操作，比如批量替换资源文件，批量打包等等。

build.xml参考了FBReader。