

[我的主页](#) [个人资料](#)

[我的闪存](#) [发短消息](#)

<	2010年1月						>
日	一	二	三	四	五	六	
27	28	29	30	31	1	2	
3	4	5	6	7	8	9	
10	11	12	13	14	15	16	
17	18	19	20	21	22	23	
24	25	26	27	28	29	30	
31	1	2	3	4	5	6	

搜索

[寻找18-28岁待业者](#)  
经实训后直接进入IT企业工  
作，薪资2500-3500，1年  
后年薪6万  
[www.isep.com.cn](#)

常用链接

[我的随笔](#)

[我的空间](#)

[我的短信](#)

[我的评论](#)

[更多链接](#)

我参与的团队

[博客园](#) [首页](#) [社区](#) [新随笔](#) [联系](#) [订阅XML](#) [管理](#)

随笔-64 评论-92 文章-43 trackbacks-0

## “蚂蚁”(NAnt)爬进.NET

### 一、引言

在Visual Studio.NET中，我们只要用一个简单的菜单的命令，就可以构造和编译一个包含大量子项目（例如，相互关联的Web页面集合、执行文件、DLL程序集，等等）的.NET项目。但是，对于大型、复杂的软件项目来说，依靠某个程序员去点击“编译”按钮有时是行不通的。如果有人不乐意在每一台机器上安装VS.NET该怎么办？要是能够自动执行软件的构造过程，永远不必有人去点击“编译”按钮，那该多好！让构造过程自动化的好处很多，不过要做到这一点，必须要有适当的构造工具。

构造工具解决的是软件编译过程相关的问题。小型开发组编写一些简单的软件可能用不上构造工具——你只要启动编译器，将代码编译成二进制执行文件，就算大功告成。

但是，现在的软件一般都是组件化的，依赖于一个或多个子项目；而这些子项目又可能由许多不同的人编写，他们随时可能签入（Check In）代码的另一个版本。只要有一个组件编译失败，或者在构造时使用了一个过时的组件，整个项目就会被拖累。因此，在构造复杂的项目时，人们往往用构造工具来解决多人协作开发中面临的问题。

编译器会针对代码中存在的问题发出错误警告，但是，如果项目包含了一些二进制执行模块以及多个相互紧密依赖的组件，要找出真正引发错误的根源就相当困难。最好有一个工具能够理清应用对外部模块的依赖关系，一旦发现问题就提出警告。

这个工具就是Ant。Ant原先由Apache Jakarta Project设计，解决了许多现有构造工具（例如Unix环境中常见的make工具）存在的不足，其中很重要的一点是，常规的构造工具总是限定于特定的操作系统、开发环境或开发语言，与此相对，Ant是平台中立的。为保持平台中立，用来告诉Ant如何编译项目的文件（构造文件）也是XML格式。这意味着使用Ant工具不局限于某一特定的平台（当然，开发平台——Java、.NET等，以及Ant本身除外）。

除了平台中立这一优点之外，Ant的另一个优点是它的构造文件是声明式的（而不是过程式的）。这意味着我们不必编写一行代码就可以完成大量工作——大多数看来繁重的任务，都可以通过在XML文件中加入声明的方式完

深圳.NET 俱乐部(0/0)  
Ajax&Atlas 技术团队(0/0)  
O/R Mapping 团队(0/0)








随笔分类(82)

ASP.NET(1)   
C#(1)   
NHibernate(9)   
技术文章(5)   
生存轨迹(45)   
文化之旅(2)   
羊群生活(19) 

随笔档案(64)

2006年8月 (1)  
2006年2月 (15)  
2006年1月 (15)  
2005年10月 (10)  
2005年9月 (11)  
2005年8月 (12)

文章分类(58)

.NET(1)   
ASP.NET(1)   
C#(1)   
C/C++(10)   
NHibernate   
图形编程(5)   
转载文章(40) 

相册

文章附图

收藏夹(7)

成。（如果构造过程极其复杂，必须借助程序逻辑才能完成，Ant 也提供这方面的支持，允许用户编写代码扩展这个工具）。另外，由于构造文件是XML 格式的，我们可以用任何XML 编辑工具创建和修改。

Ant 为.NET 实现的版本叫NAnt。NAnt 本身用C# 写成，但可以用于任何.NET 语言（NAnt 的发行包中就有好几个C#、VB.NET 和JScript.NET 的范例），甚至一个项目中可以结合运用多种.NET 语言，如果你要构造一个VB.NET 客户端应用，即使它依赖于多个由VB.NET 和C# 编写的程序集，NAnt 也能够轻而易举地完成。如果你觉得这还不够，NAnt 还可以运行多个编译器，例如，如果你想同时使用Microsoft 的工具和Mono C# 编译器（一个在Linux 上运行C# 和.NET 软件的工具，参见《在Linux 上运行C# 和.NET》），NAnt 也照办不误。

要使用NAnt，最好能够了解一些.NET 命令行编译器的知识。本文以C# 的编译器csc 为例说明，但是，你可以方便地改用vbc 或其他编译器——或者，如果你乐意的话，同时使用多个编译器。

## 二、NAnt 入门

要使用NAnt，第一步当然是从NAnt 网站下载这个工具。正在用NAnt 作为整合和构造工具的开发者优先注意一下，最新的NAnt 集成了优秀的单元测试工具NUnit 2.0。NUnit 2.0 在1.0 的基础上作了重大的改进，如果你使用NUnit 2，应该使用最新的NAnt 以充分发挥它的优势。

下面我们通过一个最简单的例子看看NAnt 的使用过程——构造一个由单个执行文件组成的C# 控制台程序。应用程序的代码如下：

```
public class HelloWorld {  
    static void Main() {  
        System.Console.WriteLine("Hello world.");  
    }  
}
```

当然，对于这样一个简单的项目，用C# 命令行编译器也可以很方便地编译，只要执行一个“csc \*.cs”命令就可以了。编译得到的结果是一个二进制可执行文件HelloWorld.exe。要用NAnt 完成同样的任务，首先要创建一个扩展名为.build 的XML 构造文件，下面是一个NAnt 构造文件的例子default.build，它完成的任务与执行一个简单的csc 编译命令一样：

### 【Listing 1：创建单个执行文件的简单NAnt 构造脚本】

```
<?xml version="1.0"?>  
<project name="Hello World" default="build" basedir=".">  
    <target name="build">  
        <csc target="exe" output="HelloWorld.exe">  
            <sources>  
                <includes name="HelloWorld.cs"/>  
            </sources>  
        </csc>  
    </target>  
</project>
```

创建好构造文件之后，执行nant 命令就可以构造项目。只要当前目录包含default.build 文件，且NAnt 本身的执行

C/C++(5) XML

计算机学习(1) XML

其他(1) XML

技术站点

aiut's blog

CSDN

ScottGu's Blog XML

SourceForge

TheCodeProject

TheServerSide.NET XML

友情链接

花露水史贝史

罗记拔甩

三水拔甩

我的史贝史

淫虫拔甩

积分与排名

积分 - 43260

排名 - 1690

最新评论 XML

1. Re:C语言图形编程(三、绘图函数-01)

谢谢你的文章!! 很好!!!

--呼呼睡大觉

2. Re:八皇后问题 C 语言代码

写这儿让人抄不太好哦

--onelovelypig

3. re: C语言图形编程(三、绘图函数-02)

谢谢博主分享!!!!

--hailoong

文件在操作系统的PATH环境变量中,执行nant命令后,NAnt就会分析default.build文件,完成default.build文件中指定的任务。

当然,对于这样一个只有一个类的项目,使用NAnt之类的工具实在是牛刀杀鸡、大材小用了。但是,如果我们要先构造执行文件然后立即执行它,或者先构造一个或多个相关模块,然后构造主执行程序,又该如何?对于这样的任务,运用Ant之类的构造工具就能节省大量的时间。

NAnt构造文件主要由目标(target)、任务(task)、相关性(dependency)三部分内容构成。一个task就是要求NAnt执行的一个任务,举例来说,NAnt支持的任务包括运行编译器、复制/删除文件、发送email,甚至还能够压缩一组文件(关于NAnt支持的完整任务清单,请参见[这里](#))。

目标描述了一组要求NAnt执行的任务,它是一种将任务分成逻辑组的手段。例如,假设我们要求NAnt删除bin目录的内容、编译5个执行文件、把编译得到的二进制文件复制到某个位置,可以把这些动作组织成一个target。

相关性可以看作是两个target之间的关系。不过Listing 1只有一个target,它的名称是build,它的任务是运行编译器编译指定的源文件。把标记的default属性设置为build,NAnt就会处理名称为build的target。

在csc任务内有一个子节点,它指定了要编译的源文件。

三、定义相关性

现在我们加入第二个target——编译好HelloWorld.exe文件后立即执行。修改后的构造文件如Listing 2所示。

【Listing 2: 包含两个相关target的构造脚本】

```
<?xml version="1.0"?>
<project name="Hello World" default="run" basedir=".">
  <target name="build">
    <csc target="exe" output="HelloWorld.exe">
      <sources>
        <includes name="HelloWorld.cs"/>
      </sources>
    </csc>
  </target>

  <target name="run" depends="build">
    <exec program="HelloWorld.exe"/>
  </target>
</project>
```

新添加的target名叫run,只包含一个用来执行程序的动作exec,此外它还有一个对build目标的相关性。这个相关性表示,在执行run这个target之前,必须先实现build这个target且必须执行成功。注意在节点中,我们把default属性由原来的build改成了run。由于run依赖于build,因此确保了在运行应用之前先编译好应用。

如果由于某种原因build目标没有达到(通常是由于编译器发现了代码存在的错误),run目标也不会执行。你可以试验一下:先在HelloWorld的代码中故意加入一个语法错误,然后再次运行NAnt,NAnt将把编译器的错误信息显示到控制台,可以方便地看出哪里出现了错误。

四、从头开始构造

4. re: C语言图形编程(一、字符屏幕)

```
/* Note:Your choice is C IDE */ #include "graphics.h" int main() { int gdriver=DETECT,gm...
```

--匿名

5. re: 张老三的站因为没有备案给封了,真郁闷。。。

学得咋样了啊

--荔枝伊珊雨

## 阅读排行榜

1. 千难万难询问试验终于得出 Quid way WA1003A 的端口映射方法(4000)
2. C#下载远程服务器文件到本地服务器(1980)
3. 关于 Hibernate 中的“脏数据”、“脏对象”(1228)
4. 今天开始了《离散数学及其应用》的阅读(1165)
5. 今天用 hbm2ddl 生成数据库脚本时,不明不白地遇到了一个问题又糊里糊涂解决了(1027)

## 评论排行榜

1. 我的偶像不是马化腾(9)
2. 千难万难询问试验终于得出 Quid way WA1003A 的端口映射方法(7)
3. 今天用 hbm2ddl 生成数据库脚本时,不明不白地遇到了一个问题又糊里糊涂解决了(6)
4. 开学到现在,宿舍一直处于断网状态,总认为背后有些阴谋。。。 (5)
5. 关于 Hibernate 中的“脏数据”、“脏对象”(5)

如果有一个编译好的二进制文件比源文件还新,NAnt不会再执行编译操作——换句话说,NAnt不会编译任何无需编译的文件。此外,如果构造文件定义了多重相关性(即,二个或二个以上的组件依赖于另一个组件),NAnt很“聪明”,它只构造被依赖的组件一次,不会重复构造同一个组件。这种处理方式大大提高了构造大型项目所需的时间,但有的时候,人们需要能够说“不管我有什么,你都编译不误”的权利,也就是说,要能够清除所有已经编译好的二进制文件,从头开始构造。

为此,许多构造文件会包含一个clean目标,开发者可以利用它来清除上一次编译留下的所有文件。下面是一个包含clean目标的构造文件例子:

### 【Listing 3: 包含clean目标的构造脚本】

```
<?xml version="1.0"?>
<project name="Hello World" default="run" basedir=".">
  <target name="build">
    <mkdir dir="bin" />

    <csc target="exe" output="bin\HelloWorld.exe">
      <sources>
        <includes name="HelloWorld.cs"/>
      </sources>
    </csc>
  </target>

  <target name="clean">
    <delete dir="bin" failonerror="false"/>
  </target>

  <target name="run" depends="build">
    <exec program="bin\HelloWorld.exe"/>
  </target>
</project>
```

clean目标并不是每次构造时都要运行,只是偶尔需要运行一下。要运行clean目标,只需执行nant clean命令即可。nant clean命令要求NAnt只执行clean目标(也就是说,不会执行构造项目的操作,只是清除一下bin目录的内容)。另外还可以看到,这个修改之后的构造脚本包含了一个mkdir动作,用来创建bin子目录以存放编译好的二进制文件。如果既要清除bin目录,又要构造项目,执行命令:nant clean build。

## 五、执行单元测试

如果要将构造过程和其他操作结合,例如email提醒和自动化的单元测试,NAnt也能够很好地完成。详细讨论NUnit单元测试框架已经超出了本文的范围,不过NAnt与NUnit确实协作得很好。Listing 4就是这样一个构造文件的例子,它构造一个应用,并把执行NUnit也作为构造过程中很自然的一部分。

### 【Listing 4: 集成了单元测试的构造文件】

```

<?xml version="1.0"?>
<project name="NUnit Integration" default="test">
  <property name="build.dir" value="\dev\src\myproject\" />

  <target name="build">
    <csc target="library" output="account.dll">
      <sources>
        <includes name="account.cs" />
      </sources>
    </csc>
  </target>

  <target name="test" depends="build">
    <csc target="library" output="account-test.dll">
      <sources>
        <includes name="account-test.cs" />
      </sources>
      <references>
        <includes name="nunit.framework.dll" />
        <includes name="account.dll" />
      </references>
    </csc>

    <nunit2>
      <test assemblyname="${build.dir}account-test.dll" />
    </nunit2>
  </target>
</project>

```

构造文件首先以一个**property**标记的形式指定项目文件的位置。把一些可能改变或可能要再次使用的值放入属性变量很有用，但不是必需的；属性通常在构造文件的开头声明，但如有必要，也可以改为通过命令行参数提供。在这个例子中，以属性的形式指定项目文件带来不少方便，因为在后面的构造过程中我们要把这些信息传递给NUnit。

接下来，构造文件依次构造出account.dll组件和测试工具account-test.dll。这两个构造过程都包含target="library"选项，这是告诉编译器我们要构造的是一个组件程序集，而不是一个.exe文件。另外，从Listing 4还可以看出，测试工具还通过references节点引用了两个它依赖的程序集——被测试的业务逻辑组件account.dll和NUnit框架。当我们构造的项目依赖于外部库时，就要用到这个节点。

构造好测试工具和业务逻辑组件后，构造脚本调用NUnit，并指定了包含测试组件的程序集的名称，要求生成一个XML格式的文件记录测试结果。

关于NUnit集成，有一点必须注意：如果你正在用NUnit 2.0，必须使用最新的NAnt版本，这是因为NUnit最近作了重大的修改，某些“稳定”的NAnt根本不能与NUnit 2.0一起运行，但最新的NAnt对NUnit 2.0的支持相当稳

定。

希望本文介绍的NAnt知识对你有用。要想了解更多有关NAnt功能的信息，请参见NAnt的文档，特别是task reference，其中包括一个NAnt能够完成哪些任务的简清单。

0

0

(请您对文章做出评价)

« 上一篇: [C#下载远程服务器文件到本地服务器](#)

» 下一篇: 昨晚第一次使用了 NUnit，方才发现它是个好东西，似乎好多好东西其实一直在身边，只是没有勇气去尝试而一直无法体会。

[寻找18-28岁待业者](#)

权威: 华浦ISEP国际软件工程师 高薪: 半年IT认证+高薪就业  
[www.isep.com.cn](http://www.isep.com.cn)

posted on 2005-10-04 20:49 [rocman](#) 阅读(446) [评论\(2\)](#) [编辑](#) [收藏](#) [网摘](#) 所属分类: [转载文章](#)

评论:

[#1楼](#) 2006-07-25 17:43 | [terryehco](#)[未注册用户]

谢谢

[回复](#) [引用](#)

[#2楼](#) 2006-07-25 17:43 | [terryehco](#)[未注册用户]

谢谢

[回复](#) [引用](#)

注册用户登录后才能发表评论，请[登录](#)或[注册](#)。

网站导航:

[博客园首页](#) [IT新闻](#) [个人主页](#) [闪存](#) [程序员招聘](#) [社区](#) [博问](#) [网摘](#)

IT新闻:

- [Hotmail发展历程回顾](#)
- [谷歌要用Nexus One重演IBM的故事](#)
- [上海市民即日起可刷手机进地铁 需120元换卡费](#)
- [.NET 2.0仍是主流 4.0成未来主力军](#)
- [赶集网收购263.com 分类信息网站开始洗牌](#)



[China-pub](#) 计算机图书网上专卖店! 6.5万品种2-8折!

[China-Pub](#) 计算机绝版图书按需印刷服务

链接: [切换模板](#)

相关搜索:

[转载文章](#)

在知识库中查看:

[“蚂蚁”\(NAnt\)爬进.NET](#)