

摩诘

我思故我在 常辨而常新

博客园 :: 首页 :: 新随笔 :: 联系 :: 订阅  :: 管理 ::  20 随笔 :: 2 文章 :: 383 评论 :: 14 引用

< 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

公告

要对我的朋友们说抱歉了,因为去年转向从事SAP的技术,所以很少有精力写.NET的新文章和回答朋友们的问题了。希望以后还能慢慢再写一些!

被访问次数:

239344

访客人数:

194422

[我的主页](#) [个人资料](#)

[我的闪存](#) [发短消息](#)

搜索

[免费学习asp.net课程](#)

本市人员可享受50-100%政府补贴 合格颁发国家职业资格和微软双认证

www.zili.cn

在.NET环境中实现每日构建(Daily Build)--NAnt篇

在.NET环境中实现每日构建--NAnt篇

前言

关于每日构建这个话题,也已经有很多很好的文章讨论了。本文的写作过程中也参考了这些文章。本文之所以继续这个题目,是因为在查阅了网上的资源后,发现没有一个比较通用的过程。所以本文就主要讨论了利用NAnt构建一个通用日编译的方案。利用这个方案,日编译的维护者可以不需要对每个要编译的方案都要做很多维护。只要定义一个属性文件就可以了。

关键词: **Daily Build, NAnt**

1. 简介

1.1. 每日构建的优点:

每日构建(Daily Build)也可称为持续集成(Continuous Integration),强调完全自动化的、可重复的创建过程,其中包括每天运行多次的自动化测试。每日构建的作用日益显得重要。它让开发者可以每天进行系统集成,从而减少了开发过程中的集成问题。

持续集成可以减少集成阶段"捉虫"消耗的时间,从而最终提高生产力。它使得绝大多数bug在引入的同一天就可以被发现。而且,由于一天之中发生变动的部分并不多,所以可以很快找到出错的位置。

1.2. 每日构建完成的任务

实现自动化每日构建需要做以下几部分的工作:

- 使创建过程完全自动化,让任何人都可以只输入一条命令就完成系统的创建。
- 使测试完全自动化,让任何人都可以只输入一条命令就运行一套完整的系统测试。
- 确保所有人都可以得到最新、最好的可执行文件。

2. 每日构建所使用的工具

在.NET环境下建立每日构建可以使用一系列开源工具:

Nant: 完成代码的自动编译,自动运行测试工具。<http://nant.sourceforge.net/builds/>

NantContrib: 自动从源码库中获取源代码。<http://nantcontrib.sourceforge.net/nightly/builds/>

NUnit2Report: 将NUnit测试工具产生的XML报告转换为HTML报告形式。<http://NUnit2Report.sourceforge.net>

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

- 上海.NET 俱乐部(0/0)
- BluePrint 翻译团队(0/0)

我的标签

- 多次触发 (1)
- FileSystemWatcher (1)
- Timer (1)
- Multipul (1)
- log4net (1)

随笔分类(20)

- 技术研究(17) (rss)
- 生活手记 (rss)
- 似水流年(3) (rss)
- 小说收藏 (rss)

随笔档案(20)

- 2008年7月 (1)
- 2007年6月 (1)
- 2006年9月 (1)
- 2006年5月 (2)
- 2005年7月 (1)
- 2005年6月 (1)
- 2005年4月 (2)
- 2005年3月 (6)
- 2004年10月 (5)

相册

- 123
- life

技术站点

朋友的blog

- Debbie's blog

收藏的站点

- 我的生活
- 记述我的工作及生活感受

积分与排名

- 积分 - 241106

VSS：Visual Source Safe，微软源码管理工具

Draco.NET: 用于自动检测VSS中源代码变动情况，调用Nant完成自动编译

http://sourceforge.net/projects/draconet/

下载所需的工具后，按照如下步骤进行安装：

在服务器上安装VSS源码管理工具

安装下载的Draco Server 和 Draco Web,修改安装后的Draco Web 目录下的web.config文件，设置正确的Draco Server安装路径
将NAnt、NAntContrib、NUnit2Report压缩包解压，将三个Bin目录中的内容复制到一个公用目录，比如D:\DailyBuildTools，然后将该路径加入系统的Path路径列表中，具体为“控制面板-> 系统属性-> 环境变量-> Path”

3. NAnt 自动脚本

NAnt脚本实现了每日构建的主体功能，它具体分为下面几部分

- 定义每日构建所需的一些环境变量，比如从VSS上下载源码的保存目录，发布目录等
- 清除旧的代码并从VSS源码库中下载最新源代码
- 编译源代码并运行测试代码集
- 将编译后的目标代码拷贝到发布目录进行发布

为了尽可能少的改动NAnt的脚本文件，简化日常维护的工作量，我们把一些对所有项目都基本相同的过程抽取出来，如环境变量定义，清除旧代码获取新代码，编译源代码，对目标代码进行发布的过程都可以写成通用的脚本，而一个具体项目的每日构建脚本则调用通用过程完成

本文采取的目录体系如下所示：

- D： \DailyBuild\
 - <project1>\Source: 存放<project1>源代码的目录
 - <project1>\Build: 存放<project1>编译后的目标代码的目录
 - <project1>\Publish: 存放<project1>的WEB发布文件的目录
 - <project1>\log: 存放<project1>的日志文件

3.1. Nant的基础知识

- Nant脚本代码文件的基本结构

```
<?xml version="1.0" encoding="gb2312"?>
<project name="Projects" default="prebuild">
    <target name="prebuild" depends="namecheck,clean " description="...">
        .....
    </target>
    <target name="namecheck" >
        .....
    </target>
</project>
```

说明： encoding="gb2312"使得脚本文件可以支持中文

<project> 标签定义了项目属性，一个脚本文件只能有一个项目定义 default="prebuild"说明该项目缺省从prebuild任务开始执行

<target> 标签定义了一项任务，任务是Nant脚本具体执行动作的最小单元

最新评论 XML

1. Re: 在C# 中利用自动化模型操纵Word
谢谢

--吉文
2. Re: 自动向网页Post信息并提取返回的信息
写得很好

--eking
3. Re: 自动向网页Post信息并提取返回的信息
地方告诉对方公司的股份大幅vxcv

--12334534563465
4. Re: 自动向网页Post信息并提取返回的信息
和很多黄金时代刻录机撒旦克里夫回家啊都是

--12334534563465
5. Re: 自动向网页Post信息并提取返回的信息
抽刀断水水更流

--12334534563465

阅读排行榜

1. 深入浅出之正则表达式（一）(72740)
2. Log4Net使用指南(38230)
3. 深入浅出之正则表达式（二）(25891)
4. 自动向网页Post信息并提取返回的信息(14533)
5. VS.NET 2003集成环境插件开发指南(一)----操纵菜单(12623)

评论排行榜

1. Log4Net使用指南(89)
2. 深入浅出之正则表达式（一）(54)
3. 自动向网页Post信息并提取返回的信息(45)
4. 在.NET环境中实现每日构建(Daily Build)--NAnt篇(38)
5. 深入浅出之正则表达式（二）(32)

depends="namecheck,clean "说明该任务执行前需要namecheck和clean任务先执行
description描述了该任务的一些说明性信息

- 定义变量
`<property name="<变量名>" value="$<变量值>"/>`
如上所示，定义变量使用<property>标签，name属性定义了变量的名称,value属性定义变量的值，其中name属性可以使用字母、数字、点号、下划线等符号，而value属性可以使用字符串或是已经定义的变量，Nant内建的函数等，要使用已经定义的变量，可以用\${<变量名>}，要使用内建函数，可以使用\${<函数名称>}
如: `<property name="solution.basedir" value="${core.basedir}\${solution.name}"/>`
使用了已定义变量core.basedir和solution.name来定义变量solution.basedir;
`<property name="curdir" value="${directory::get-current-directory()}/>`
使用了NAnt内建函数directory::get-current-directory()来定义curdir变量

3.2. 定义环境变量

定义环境变量的脚本代码写在Common。Config文件里
主要有以下几类信息的定义：

- 每日构建所在的根目录
`<property name="curdir" value="${directory::get-current-directory()}/>`
`<property name="core.basedir" value="${curdir}"/>`
说明：\${directory::get-current-directory()}内建函数获取当前文件所在路径信息
- 被编译的解决方案的目录结构，和前面提到的目录体系一致
`<property name="solution.basedir" value="${core.basedir}\${solution.name}"/>`
`<property name="solution.source" value="${solution.basedir}\source"/>`
`<property name="solution.build" value="${solution.basedir}\build"/>`
`<property name="solution.log" value="${solution.basedir}\log"/>`
说明：以上代码是定义了要编译的解决方案的目录结构信息，其中\${solution.name}是由外部传入的解决方案的名称，后面的代码将根据该名称在日编译的根目录下生成和solution.name指定的名称同名的目录，并在该目录下生成source,buld,log等子目录
- VSS源代码管理系统的基本信息
`<!--vss数据库登录信息-->`
`<property name="vss.username" value="autobuild"/>`
`<property name="vss.password" value="autobuild"/>`
`<!--vss数据库所在的位置-->`
`<property name="vss.dbpath" value="\\10.136.238.231\vss\srcsafe.ini"/>`
`<!--vss中工程的根目录-->`
`<property name="vss.basepath" value="$"/>`
说明：定义了和VSS源码管理系统相关的一些信息，其中VSS数据库所在位置可以是网络路径，也可以是本地路径
- <编译时的一些参数

```

<!--编译版本号-->
<property name="build.number" value="1.0"/>
<!--决定编译是Debug版本还是Release版本-->
<property name="build.configuration" value="Release"/>

```

3.3. 建立目录结构，获取源代码

脚本代码写在CheckSource.build.xml文件里

- 包含在Common.config文件里定义的公共变量

```
<include buildfile="common.config"/>
```

- 检查是否存在solution.name变量

```

<target name="namecheck" description="检查solution.name变量是否设置">
  <!--检查解决方案名称是否已经定义-->
    <ifnot test="${property::exists('solution.name')}">
      <fail message="未定义解决方案名称solution.name"/>
    </ifnot>
    <!--去掉可能的空格字符-->
    <property name="solution.name" value="${string::trim(solution.name)}/>
    <!--检查solution.name变量是否为空字符-->
    <if test="${string::get-length(solution.name)==0}">
      <fail message="未定义解决方案的名称solution.name"/>
    </if>
  </target>

```

说明：\${property::exists('<变量名>')}是NAnt内建函数，用于测试某变量是否存在

\${string::get-length(<字符串变量>)==0}测试字符串的长度是否为0

<ifnot test=<逻辑表达式> ... </ifnot>:如果test表达式值为假，执行<ifnot>标签内的代码

<if test=<逻辑表达式> ... </if>:如果test表达式值为假，执行<if>标签内的代码

- 建立解决方案的目录结构

```

<target name="clean" depends="namecheck" description="移除旧目录,建立新目录">
  <!--删除旧的解决方案代码所在目录-->
  <delete dir="${solution.basedir}" failonerror="false"/>
  <!--重新建立目录-->
  <mkdir dir="${solution.basedir}\" failonerror="false"/>
  <mkdir dir="${solution.source}" failonerror="false"/>
  <mkdir dir="${solution.build}" failonerror="false"/>
  <mkdir dir="${solution.log}" failonerror="false"/>
</target>

```

说明:delete和mkdir标签内的failonerror属性表示即使操作文件夹的过程中出现了错误，也忽略错误向下执行

- 获取源代码:

从VSS上获取解决方案<solution.name>的源代码

```
<target name="getsourcecode">
  <!--检查从VSS上下载解决方案的路径是否设定-->
  <!-- 如果不定义vss.projectpath,则缺省为solution.name -->
  <ifnot test="${property::exists('vss.projectpath')}">
    <property name="vss.projectpath" value="${solution.name}"/>
  </ifnot>
  <vssget
    user="${vss.username}"
    password="${vss.password}"
    localpath="${solution.source}"
    recursive="true"
    replace="true"
    dbpath="${vss.dbpath}"
    path="${vss.basepath}${vss.projectpath}"
  />
</target>
```

说明: <vssget> 标签是NAntContrib的语法, 用来从VSS源码管理器上下载源代码, user和password属性表示登录VSS服务器的信息; Localpath属性是指下载的源代码存放的路径; recursive="true"表示递归获取代码; replace="true"表示如果本地有重复文件, 则进行覆盖; dbpath定义VSS的srcsafe.ini文件的路径信息, 包括srcsafe.ini文件名; path定义了要获取的源代码在VSS数据库中的路径, 一般都是以\$/为根目录。

3.4. 编译源代码

- 编译命令

编译解决方案的命令为

```
<solution solutionfile="..." configuration="..." outputdir="...">
  <webmap>
    <map url="..." path="..." />
    <map url="..." path="..." />
  </webmap>
</solution>
```

其中solutionfile属性表明了要编译的解决方案文件的路径信息, 即以"sln"为扩展名的文件,

configuration属性表明要编译的是发行版还是调试版, 取值为"Release"或"Debug"

outputdir表明了编译后的动态链接库或可执行文件存放的目录

solution中的嵌套标签<webmap>用于当解决方案含有WEB项目的情况, 有几个WEB项目, 就有几项<map>标签, map标签中的url属性为WEB项目的*.csproj文件的WEB路径, path则为该*.csproj文件所在磁盘上的物理路径, 例如, 解决方案中有WEB项目exam, 则map标签为 <map url="http://localhost/exam/exam.csproj" path="c:\exam\exam.csproj"

- 根据解决方案名称获取解决方案文件的路径信息

```
<target name="build" description="编译解决方案">
```

```

<!-- 查找解决方案文件名 -->
<foreach item="File" property="filename">
    <in>
        <items>
            <include name="**\${solution.name}.sln"/>
        </items>
    </in>
    <do>
        <!--根据文件名设置解决方案的名称-->
        <property name="solution.file" value="\${filename}"/>
    </do>
</foreach>

```

说明：<foreach>标签是NAnt中处理循环的命令,item="File"说明foreach进行循环处理的对象是文件，<include>中的name变量表示要查找的文件信息，"**\"表示查找路径包括子目录。Foreach的属性property="变量名"表示查找到的文件路径信息保存在该变量中，可以在<do>标签中引用.foreach每查找到一项符合条件的Item，都会执行<do>标签中的代码，以上代码执行的结果就是查找到指定名称的解决方案文件，供后面编译代码使用

- 获取解决方案中WEB项目的路径信息

如果解决方案中含有WEB项目，则其编译命令和不含WEB项目的解决方案编译有所区别，所以要区别对待。如果解决方案含有多个WEB项目，则可以让用户将多个WEB项目的名称放在一个变量中，如solution.webprojects，以逗号或分号或空格做分隔符。然后将项目名称分别提取出来，根据Web项目的个数决定solution命令的形式，代码如下

```

<!--将solution.webprojects中用",";"或" "分隔的Web工程名提取出来，
分别设为webproject1,webproject2 -->
<if test="\${property::exists('solution.webprojects')}">
    <foreach item="String" in="\${solution.webprojects}" delim=";;, " property="project">
        <if test="\${property::exists('webproject1')}">
            <property name="webproject2" value="\${project}"/>
        </if>
        <ifnot test="\${property::exists('webproject1')}">
            <property name="webproject1" value="\${project}"/>
        </ifnot>
    </foreach>
</if>

```

以上代码中foreach标签的属性item="String" in="\\${solution.webprojects}" delim=";;, " property="project"表明循环对象是字符串，对in所代表的字符串

如果设定solution.webprojects="webprj1;webprj2"，则以上代码执行的结果是定义了两个变量webproject1="webproj1"和webproject2="webproj2"

- 查找WEB工程名

根据前面从solution.webprojects中提取出来的webproj1和webproj2变量，查找该WEB工程的文件名

```

<!-- 查找WEB工程文件名 -->
<if test="\${property::exists('webproject1')}">
    <echo message="test \${webproject1}" />

```

```

    <foreach item="File" property="filename">
    <in>
        <items>
            <include name="**\${webproject1}.csproj"/>
        </items>
    </in>
    <do>
        <!--根据Web项目的名称获取Web项目文件路径,可以处理两个Web项目的情况-->
        <echo message="WebProject file=${filename}"/>
        <property name="webproject1.file" value="${filename}"/>
    </do>
    </foreach>
</if>

```

同理可以处理存在第二个WEB工程项目的情况，设置webproject2.file变量

- 编译解决方案

最后是编译解决方案，分别根据无WEB项目，有2个WEB项目，有一个WEB项目的三种情况处理

下面仅列出有两个WEB项目的情况

```

    <!-- 存在2个Web工程 -->
    <if test="${property::exists('webproject2')}">
        <solution
            solutionfile="${solution.file}"
            configuration="${build.configuration}"
            outputdir="${solution.build}"
        >
            <webmap>
                <map
                    url="http://localhost/${webproject1}/${webproject1}.csproj"
                    path="${webproject1.file}"
                />
                <map
                    url="http://localhost/${webproject2}/${webproject2}.csproj"
                    path="${webproject2.file}"
                />
            </webmap>
        </solution>
    </if>

```

3.5. 运行测试代码

- 测试命令

NAnt中关于测试的命令是<NUnit2>标签

```
<nunit2>
  <formatter type="Xml" usefile="true"
    extension=".xml" outputdir="..."
  />

  <test assemblyname="..." haltonfailure="false" />
</nunit2>
```

说明: <formatter> 标签中, type="Xml"表明了根据测试结果生成XML结构化信息, usefile="true"表明使用文件保存测试结果, extension=".xml"表明生成的文件扩展名为xml, outputdir指出了文件将被保存到哪个目录

Test标签中的assemblyname表明了被测试的dll程序集的路径信息, haltonfailure="false"表明即使测试没有通过仍然继续执行脚本文件

这样在测试命令完成后,会在outputdir指出的目录下生成一个XML形式的报告文件,为了增加测试结果的可读性,可以使用另一个工具NUnit2Report,将测试结果转换为直观的HTML文件。具体命令如下

```
<nunit2report out="<文件名>" todir="<输出目录>" >
  <fileset>

    <includes name="<文件匹配符>" />

  </fileset>
</nunit2report>
```

说明: includes 标签用来搜索符合条件的XML文件,转换出来的HTML文件保存为out指出的文件名, todir指出了HTML文件将保存的目录信息

```
<if test="${property::exists('solution.testprojects')}">
  <foreach item="String" in="${solution.testprojects}" delim=";" " property="project">
    <property name="testfile" value="${solution.build}\${project}.dll"/>
    <nunit2>
      <formatter type="Xml" usefile="true"
        extension=".xml" outputdir="${solution.build}" />
      <test assemblyname="${testfile}" haltonfailure="false" />
    </nunit2>
    <nunit2report out="${project}.html" todir="${solution.log}" >
      <fileset>
        <includes name="${solution.build}\*.xml" />
      </fileset>
    </nunit2report>
  </foreach>
</if>
```

3.6. 进行WEB发布

WEB发布主要针对有WEB工程项目的解决方案,其实现原理为利用NAnt的拷贝命令,将WEB工程下除了源代码,资源代码, VSS信息文件外的其他文件和编译后的程序集拷贝到发布目录,最后设置WEB虚拟路径以供WEB访问的过程。

设置WEB虚拟路径的命令为

<mkiisdir dirpath="<物理路径>" vdirname="<虚拟路径>"/>
说明:设WEB项目发布在C:\Intepub\wwwroot\Exam，访问该WEB项目用地址http://127.0.0.1/Example/default.aspx，则<物理路径>为"C:\Intepub\wwwroot\Exam"，虚拟路径为"Example"（此处略去详细代码）。

[下载示例代码](#)

00

(请您对文章做出评价)

« 上一篇: 自动向网页Post信息并提取返回的信息
» 下一篇: 深入浅出之正则表达式（一）

[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](#)

posted on 2005-07-29 22:36 [sema](#) 阅读(9222) [评论\(38\)](#) [编辑](#) [收藏](#) [网摘](#) 所属分类: [技术研究](#)

评论

- #1楼 2005-07-29 22:57 [hudan](#)

好，正打算看看每日构建。

字体是不是小了一号，看起来有些不习惯
[回复](#) [引用](#) [查看](#)
- #2楼[楼主] 2005-07-29 23:10 [sema](#)

用word直接贴上去的，要改字体太麻烦，抱歉了
你可以下载我的示例代码，参考这篇文章看，这样可能看起来容易些
[回复](#) [引用](#) [查看](#)
- #3楼 2005-07-30 00:55 [木野狐](#)

你可以先从 word 复制到记事本，然后再再从记事本复制到这里就好看了。

这个字体实在难受阿，老大。
[回复](#) [引用](#) [查看](#)
- #4楼 2005-07-30 01:37 [春鱼](#)

博客园其实应该多出现像本文这样的文章。写出自己的技术体会或者best practices. 这才是blog所需要的。比那些只知道哗众取宠的所谓高手强。

小小的建议：作者可以在标题里加上Daily Build，可以使搜索方便一些。

[回复](#) [引用](#) [查看](#)

#5楼 2005-07-30 08:20 双鱼座

反对来自java的“每日构建”这样与IDE脱节的概念，更反对NAnt这样的垃圾。

Java的世界是一个分裂的世界，而.net的世界是一个全面统一的世界。几乎没有人使用vs.net以外的开发工具，包括使用MS提供的Web Matrix的都几乎没有。

使用“每日构建”或者单元测试必须依赖IDE的支持，所以最聪明的作法还是等待MS来提供，而不是依赖一些垃圾产品，例如NAnt、NUnit或者Log4Net这样完全不考虑.Net中的一些特殊性的、纯粹从Java移植过来的东西。几乎可以肯定地讲，这些NXXX的东西是不可能成功的，完全没有可能象Java世界中一样成为标准。学习它们只是浪费时间和精力，甚至污染头脑中的纯洁。

[回复](#) [引用](#) [查看](#)

#6楼[楼主] 2005-07-30 08:49 sema

To 春鱼:

谢谢你的建议，已采纳！

To 双鱼座:

你的观点太极端了，我不敢苟同，呵呵！

每日构建这个概念不是与IDE脱节，即使是微软，也是在使用每日构建的。另外我并不觉得NAnt是“垃圾”，应该说他是个很优秀的开源项目。而且我个人认为，我们应该鼓励多样性和创造性。如果没有NAnt，NUnit，Log4net这样的优秀开源项目，微软会不会考虑在他的产品里加入这些相应的特性，也是个值得怀疑的问题。一个典型的例子就是，自从Netscape公司被微软用浏览器绑定的方法挤垮后，微软在浏览器软件上基本就没有大的创新了。而Firefox流行起来后，微软又准备研发下一版新的浏览器了。所以如果等待微软主动提供，估计你会把眼睛望穿的。

Anyway，这个问题是仁者见仁，智者见智，每个人都有自己的观点。所以我觉得这个问题深入讨论下去也没什么意义。所以，就让这个话题到此为止吧。

[回复](#) [引用](#) [查看](#)

#7楼 2005-07-30 09:18 四海为家[未注册用户]

双鱼座的观点，看似好激进哦，其实好保守，他是什么都想着微软来解决的，呵呵，可是要是没nunit，微软就不会在vs2005中集成msunit，微软其实从.net开源世界里汲取了好多营养，没有n***，要得到微软更好的服务，就要等更长的时间，还能指望微软来解决么？

[回复](#) [引用](#)

#8楼 2005-07-30 10:06 双鱼座

@四海为家:

我还没有看到vs2005中的msunit，我无权评论。但是我对NUnit是有权保留的。虽然我目前为止一直用NUnit做单元测试，但是我已经可以肯定NUnit不可能象JUnit一样成为标准。

我只能猜测一下。象NUnit这样的东西，MS应该会做得更合理一些，至少不会将TestFixtureAttribute、TestAttribute和Assert这些在扩展中必须用到的类型封闭了。因为这样的封闭不会对NUnit GUI带来半点好处。你应该知道，JUnit成功的原因并不是JUnit这样一个简单的程序，而是一种标准、一种体系。JUnit吸引众多开发者的原因是因为JUnit有了数不清的扩充，而NUnit根本没有提供任何扩充的可能。

微软或者说世界上任何一家大的软件公司都会从开源汲取营养，这是肯定的，但我更肯定不会是从NXXXX这些垃圾产品。

我不知道你从我100%肯定JUnit、Ant、Hibernate、Log4J的同时，又100%否定NUnit、NAnt、NHibernate、Log4Net，到底是觉得我开放还是觉得我保守？

[回复](#) [引用](#) [查看](#)

#9楼 2005-07-30 10:24 双鱼座

@sema:

我以为你的观点才过于主观。MS是不是会吸收NUnit、NAnt、Log4Net这些有人认为是垃圾的东西，我觉得你是无法确定的。你怎么就认为MS不是从JUnit、Ant和Log4J获取的营养呢？毕竟所有在.net中实现的java中的开源项目比在java中的实现相差一大截是一个显而易见的客观事实。每日构建在某些情况下是需要的。但现在在.net中不具备你所谓的“每日构建”的环境。每日构建是一种开发模式的特征，而RAD是另外一种开发模式的特征。

不明白你为什么不能接受别人对你的话题进行评论。难道这样才是开放而不是保守吗？

[回复](#) [引用](#) [查看](#)

#10楼[楼主] 2005-07-30 11:20 sema

To 双鱼座:

我想对一个问题，每个人有每个人的看法，这是很正常的。我觉得我们可以发表自己的观点，也没必要强求别人接受自己的观点，求同存异是王道，呵呵。

我想我也没有“不能接受别人对你的话题进行评论”。我只是在这个问题上有不同观点而已，而不同意你的观点并不等于不能接受你的评论，是吗？但是，请大家注意，在我的版面上是绝对禁止相互之间进行人身攻击的！相互间有不同的观点碰撞，这是正常的，甚至这种碰撞比较激烈，这也是允许的。但请大家保持对彼此的相互尊重（以上这段话并不针对任何人）。我这里欢迎任何技术方面的讨论

[回复](#) [引用](#) [查看](#)

#11楼 2005-07-30 12:58 蛙蛙池塘

讨论具体问题，对事不对人哦

[回复](#) [引用](#) [查看](#)

#12楼 2005-07-30 13:28 双鱼座

@sema:

“所以我觉得这个问题深入讨论下去也没什么意义。所以，就让这个话题到此为止吧。”

在网上好象大多数都是这样的。因为我提出了不同的意见，令别人不快，所以急于停止话题。事实上我也只是说明我的观点，从来没有强迫别人来接受，不是吗？相反我倒真的希望能够有机会领教哪位高手举出NXXXX有哪些闪光点。

[回复](#) [引用](#) [查看](#)

#13楼 2005-07-30 13:52 idior

NXXXX的闪光点就在于你可以在.net下方便的发挥XXXX的思想。

[回复](#) [引用](#) [查看](#)

#14楼[楼主] 2005-07-30 14:14 sema

To 双鱼座:

“所以我觉得这个问题深入讨论下去也没什么意义。所以，就让这个话题到此为止吧。”--如果这句话让你感到不快，我向你道歉。我说这句话没有其他意思，只是觉得像这样的讨论很难达到一个共识，讨论到后来也不会有什么结果，所以才这样说。并不是针对你的评论。

[回复](#) [引用](#) [查看](#)

#15楼 2005-07-30 15:19 双鱼座

@idior:

你说的应该是NXXXX的本意，的确，XXXX的思想值得推崇。遗憾的是，NXXXX不仅没有很好地实现，反而对理解和发扬XXXX的思想造成破坏。并且，这种现象不是偶然，而是普遍存在。我渴望在.net的世界里出现真正体现XXXX思想的东西，相信大多数人都和我一样渴望。但是NXXXX的每一个项目都令我极端地失望。

@sema:

很难达到共识的东西很多，但是我不认为不能达到共识就不要去讨论，至少这不是一种Open的态度。不过我还是必须尊重你的态度，我退出你这个话题的讨论。

[回复](#) [引用](#) [查看](#)

#16楼[楼主] 2005-07-30 22:45 sema

To 双鱼座:

我没有不愿你讨论，我只是发表我个人的观点而已。当然我的观点不一定完全正确，你可以仅把我当作一个话题的参与者。

[回复](#) [引用](#) [查看](#)

#17楼 2005-07-31 12:31 surodansu[未注册用户]

to 双鱼座

象你这样的人，在这里有很多，我看到过很多象你这样的回复。这和我想象中的博客园是不一样的。一篇文章，它哪怕只给一个人带来了帮助，那就够了。不需要象你这样的人来颠覆它，你知道你说的那些根本就是没意义的。也许你这样的人在技术方面真的很好，那么我希望你不是在这里说一些无意义的话，而是为.net多做出些实质的东西。

另外，你知道sema为什么说这个问题到此为止吗？是因为你的回复用词不当，请你在技术方面无限提高的同时，别忘了本，中国是个谦逊的民族，而从你的言语中看不出来。

[回复](#) [引用](#)

#18楼[楼主] 2005-07-31 13:14 sema

to surodansu:

请别这样说双鱼座，我想每个人都有自己的观点。作为技术方面的看法，我觉得不同的观点是可以保留的。我不愿意继续就这个

问题讨论下去，是担心讨论到后来就成了口水仗，没有实质性的结果。我只有这个意思，没有更多的意思。我想双鱼座也只是表达了他对N****的看法，也没有反对我发这样的文章来帮助一些需要这方面知识的人。大家之间讨论问题，如果伤了和气，就是我不愿意看到的了。

谢谢大家的参与！

[回复](#) [引用](#) [查看](#)

#19楼 2005-08-08 15:51 alittlefish

个人觉得双鱼座虽然看起来激进，但是讨论就是需要有激烈的碰撞。

在前段时间，也包括最近，园子里有很多关于nxxx的介绍，或者是需要用的nxxx的一些opensource的东西，我的感觉是两个字：饱和。

对于写文章的人，是为了表达自己的观点态度，但是对看文章的人，如果了解甚至熟悉作者的文章内容的话，可以有自己的不同观点，但是很多想来了了解作者所述的内容的人，作者的文章对他们有一定导向作用，当园子里的“个人观点”铺天盖地的时候，就是要有一些争论，与其只讲怎么应用，不如就该内容是否有价值做些大胆的争论，亦能加深旁人的印象，有能让参与者巩固自己的认识。

双鱼座算起来是前辈了，当然我不反对他的这种表达方式并不是因为这个，只是经常看到他的一些所谓激进的回复，偶尔象强心剂一样让我对一些老套的文章有了新的认识，并产生关注的兴趣。

[回复](#) [引用](#) [查看](#)

#20楼 2005-08-16 13:52 世玉[未注册用户]

看了您的文章之后，自己动手做了个试验，遇到一个问题：
我的解决方案里只有一个winform的项目，项目里引用了word，在编译时出现一个错误：找不到文件或程序集名称“Microsoft.Office.Interop.Word”，或找不到它的一个依赖项。
如果我去掉word引用，编译就没问题。请问像这种引用com组件的情况下，build文件应该怎么写？

[回复](#) [引用](#)

#21楼[楼主] 2005-08-16 22:04 sema

这个问题一般来说是因为你的引用设置不正确引起的。你最好把那个引用的程序集拷贝到你的项目文件夹下面，然后在VS的解决方案资源管理器里面将该程序集添加到解决方案中来。这样NAnt将你的源代码从VSS中拉出来的时候，就会把该被引用的程序集一起Copy出来的。不然的话，你光有引用，但是却没有该文件，NAnt怎么能找到那个被引用的文件呢？自然会说找不到依赖项了。你试试看吧！

[回复](#) [引用](#) [查看](#)

#22楼[楼主] 2005-08-16 22:08 sema

To 世玉
发完了前面的消息才看到原来是你啊！你写的太客气了，什么“您”啊，搞得我还以为认错人了。呵呵！

[回复](#) [引用](#) [查看](#)

#23楼 2005-08-18 08:04 世玉[未注册用户]

还是不行，等你哪天有时间我再向你仔细请教请教。

[回复](#) [引用](#)

#24楼 2006-02-07 15:31 MaxBoX

TO:sema

我最近在研究这块

看了你的文章对我的帮助很大

现在正在借助你这文章的帮助在做一些的事例

可能还会运到一些困难

不知道sema你最近有没有空

我遇到的问题发出来 可不可以帮助我下

[回复](#) [引用](#) [查看](#)

#25楼[楼主] 2006-02-08 13:23 sema

@ MaxBoX

我只能说尽量了。去年换了工作，现在转做SAP了。而且这个题目我也是一年半前做过一些研究。好长时间不摸，有些生疏了。不到之处还请谅解！

[回复](#) [引用](#) [查看](#)

#26楼 2006-02-09 09:02 MaxBoX

To: sema

首先谢谢你对我的留言回复，我借用了你的例子，现在就是在<nunit2>这个标签中做测试的时候他所编译的.xml文件没有在我指定的目录下生成。我的代码如下

```
<?xml version="1.0" encoding="gb2312"?>
<project name="Projects" default="test">
<include buildfile="common.config"/>

<target name="test" description="查找并测试代码">
<if test="${property::exists('solution.testprojects')}">
<foreach item="String" in="${solution.testprojects}" delim=";" "property="project">
<property name="testfile" value="F:\fw\DailyBuild\bin\netshop.dll"/>

<nunit2>
<formatter type="Xml" usefile="true" extension=".xml" outputdir="f:\fw\NUnit\" />
<test assemblyname="${testfile}" tonfailure="false" />
</nunit2>

<nunit2report out="netshop.html" todir="F:\fw\DailyBuild\NUnit" >

<fileset>

<includes name="netshop\*.xml" />

</fileset>

</nunit2report>
```

```
</foreach>
```

```
</if>
```

```
</target>
```

```
</project>
```

请你抽空的时候帮我看看 谢谢~~

[回复](#) [引用](#) [查看](#)

#27楼 2006-02-26 09:13 劲舞团[未注册用户]

不错。。。收藏了。。。。

[回复](#) [引用](#)

#28楼 2006-03-02 16:05 qixiafeng[未注册用户]

请问可有办法将所有的操作用TXT文件记录下来，以供查看呢？

[回复](#) [引用](#)

#29楼 2006-03-08 14:43 qixiafeng[未注册用户]

用一个bat文件运行nant ,将信息保存到TXT文件即可将所有的操作日志保存下来

[回复](#) [引用](#)

#30楼[楼主] 2006-03-08 14:50 sema

谢谢qixiafeng!

[回复](#) [引用](#) [查看](#)

#31楼 2006-03-21 15:33 smallmousel[未注册用户]

在VS2005开发的Web站点中不存在CSPROJ文件，该如何编译呢。

[回复](#) [引用](#)

#32楼 2006-04-14 10:23 jzp[未注册用户]

很好，好久以前就看到这篇文章了。

[回复](#) [引用](#)

#33楼 2006-11-21 16:49 特里猪

谁知道maillogger怎么使用

有问题想请教大家

[回复](#) [引用](#) [查看](#)

#34楼 2007-07-01 19:04 小鱼儿[未注册用户]

nant不支持编译VS2005的解决方案和工程项目，有什么方法解决了。。
问:NANT，有没有支持MSBUID的方案呢？
[回复](#) [引用](#)

#35楼 2007-10-23 13:06 Koy

强列支持,正想学下这东西!
[回复](#) [引用](#) [查看](#)

#36楼 2007-10-31 11:03 尊敬楼主[未注册用户]

楼主的一丝不苟的精神令我辈敬佩有加。向你学习知识和为人
[回复](#) [引用](#)

#37楼 2008-12-01 14:47 wind.hong

先谢谢楼主的文章，提一点点小建议 把文章的字体改改，好让大家看的舒服点。
[回复](#) [引用](#) [查看](#)

#38楼 2009-07-10 15:22 荷塘月色

谢谢楼主，我正准备学习这个，不过有些链接都失效了，能麻烦更新一下吗？万分感谢
[回复](#) [引用](#) [查看](#)

注册用户登录后才能发表评论，请[登录](#)或[注册](#)。

[免费学习asp.net课程](#)

本市人员可享受50-100%政府补贴 合格颁发国家职业资格和微软双认证

[www.zili.cn](#)

[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.isen.com.cn](#)

[免费测评孩子的潜能](#)

权威：测评准确找出孩子的优势和弱势 免费：0—6岁孩子潜能测试 因材施教

[www.61smart.com](#)

[边唱歌 边学英语](#)

老外教你一起唱！轻松英语 就要这样学

[www.MarsEnglish.cn](#)

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:
[技术研究](#)

在知识库中查看:
[在.NET环境中实现每日构建\(Daily Build\)--NAnt篇](#)

Powered by:

[博客园](#)

Copyright © sema