|  |
| --- |
|  |
| Jenkins部署与使用 |
|  |
|  |
| ©MetaShare Inc. |
| **7/25/2019** |

|  |
| --- |
|  |

**修订记录**

|  |  |  |
| --- | --- | --- |
| **日期** | **修改人** | **修改说明** |
| 2019-07-25 | 雷宇 | 初稿 |
| 2019-11-02 | 许志尧 | Rev1 |
|  |  |  |

目录

[1 简介 1](#_Toc23597403)

[1.1 虚拟机服务器登录信息 1](#_Toc23597404)

[2 安装 1](#_Toc23597405)

[2.1 使用.war文件安装 1](#_Toc23597406)

[2.2 使用.exe文件安装 1](#_Toc23597407)

[3 系统配置 1](#_Toc23597408)

[3.1 修改端口 2](#_Toc23597409)

[3.2 全局工具 3](#_Toc23597410)

[3.2.1 MSBuild.exe路径 3](#_Toc23597411)

[3.2.2 MSBuild命令行参数 3](#_Toc23597412)

[3.3 插件管理 4](#_Toc23597413)

[3.3.1 安装插件 4](#_Toc23597414)

[3.3.2 重启服务 4](#_Toc23597415)

[4 项目部署 4](#_Toc23597416)

[4.1 新建项目New Item 4](#_Toc23597417)

[4.2 源码管理Source Code Management 5](#_Toc23597418)

[4.3 构建触发器Build Triggers 7](#_Toc23597419)

[4.3.1 构建触发器语法的规则 7](#_Toc23597420)

[4.3.2 构建触发器语法的实例 8](#_Toc23597421)

[4.4 构建Build 8](#_Toc23597422)

[4.4.1 Build a Visual Studio project or solution using MSBuild 8](#_Toc23597423)

[4.4.2 Execute Windows batch common 9](#_Toc23597424)

[4.5 构建后操作Post-build Actions 9](#_Toc23597425)

[5 运行构建Build Now 10](#_Toc23597426)

[6 查看构建结果 11](#_Toc23597427)

[7 参考资料 12](#_Toc23597428)

[8 遇到的问题 12](#_Toc23597429)

[8.1 运行Jenkins的机器需要安装.NET Framwork 12](#_Toc23597430)

[8.2 .NET Framework和 .NET Core工程需要用不同的编译工具 12](#_Toc23597431)

[8.3 Error MSB4057: The target “GetTargetPath” does not exist in the project 12](#_Toc23597432)

[8.4 忘记管理员密码 13](#_Toc23597433)

[8.5 需检查代码完整 13](#_Toc23597434)

# 简介

Jenkins（持续集成Continue Integration工具）是一个独立的开源自动化服务器，可用于自动化各种任务，如构建，测试和部署软件。支持多种插件。可定时编译，运行测试用例并输出测试结果。

Sanjel项目的Jenkins服务器是部署在虚拟机上，有完整的Windows Server 2012操作系统，并配备了完整版本的Visual Studio 2019。如Jenkins上编译报错，检查过并不是Jenkins配置问题的话，可以远程登录上虚拟机做运行环境的设置，或者用Visual Studio检查从svn下载的代码的完整度。

## 虚拟机服务器登录信息

url：117.34.118.114:10029

账号：administrator 密码： Meta123$%^

# 安装

有多种安装方式，实测判断，每次新安装以早上7点半到9点之间，Jenkins安装时连接插件服务器比较稳定，其他时间段会出现无法下载插件导致安装不完全的情况。下载页面：<https://jenkins.io/zh/download/>

## 使用.war文件安装

Jenkins的Web应用程序ARchive（WAR）文件版本可以安装在任何支持Java的操作系统或平台上。

1. 安装JDK并设置环境变量，版本为Java8或Java11，不支持其他版本
2. 将最新的稳定Jenkins WAR包 下载到您计算机上的相应目录
3. 在下载的目录内打开一个终端/命令提示符窗口，使用命令运行war文件：java -jar jenkins.war
4. 浏览器打开http://localhost:8080并等到\*Unlock Jenkins\*页面出现
5. 继续使用Post-installation setup wizard后面步骤设置向导

## 使用.exe文件安装

1. 下载安装文件并运行
2. 根据提示设置

# 系统配置

访问地址：<http://117.34.118.114:7511/>

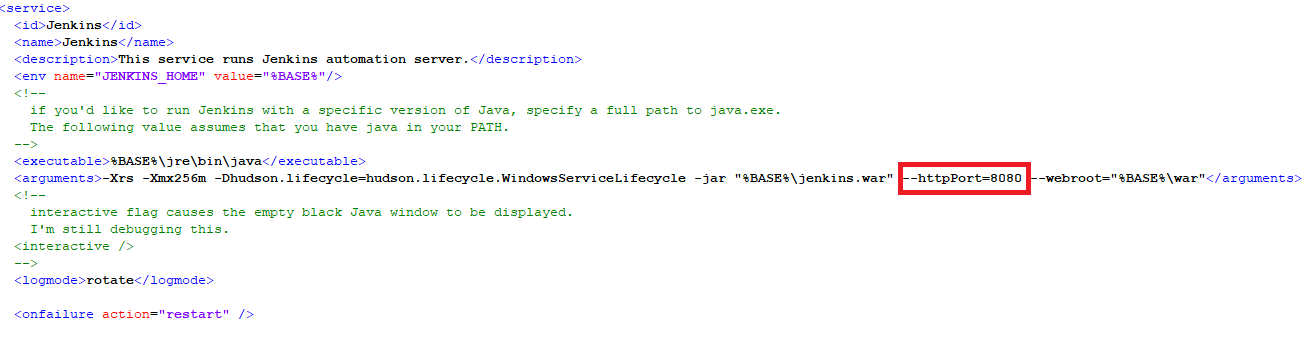
账号：metashare 密码：Meta20!9$ummer



## 修改端口

在安装完成后即可修改Jenkins系统端口。在Jenkins根目录下jenkins.xml里，设置

—httpPort={修改的端口}，重新启动Jenkins，即可生效。



修改完后，Jenkins后的主界面可能会提示反向代理设置出错，原因是Jenkins的URL设置没有完成。进入Manage Jenkins（系统管理）-> Configure System（系统设置）->Jenkins Location一项，修改Jenkins URL为<http://117.34.118.114:7511>

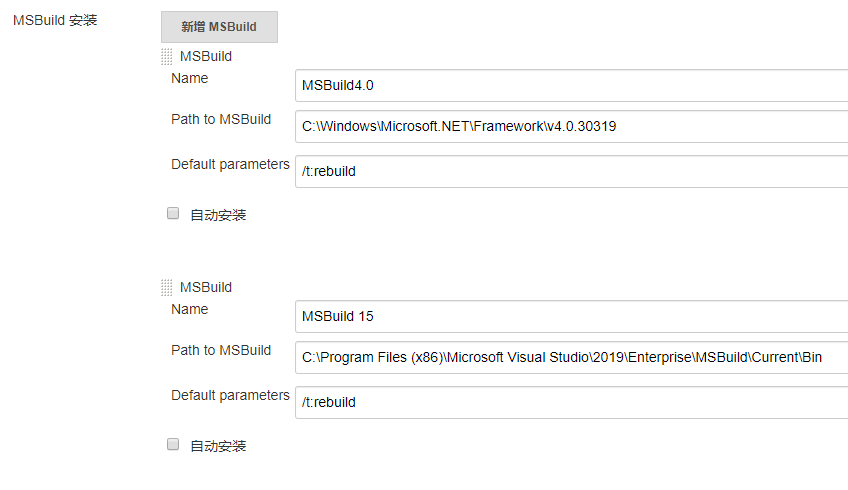


## 全局工具

添加MSBuild：Manage Jenkins系统管理 -> Global Tool Configuration全局工具配置 -> MSBuild -> 新增MSBuild。



分别填写Name，Path to MSBuild，Default parameters。名称可随意填写，Path为服务器上MSBuild.exe文件的路径，默认参数可省略。可添加多个不同路径下的MSBuild.exe。



对于新发布的项目，一般使用Visual Studio版本的MSBuild，而非Windows自带的。

### MSBuild.exe路径

C:\Windows\Microsoft.NET\Framework\v4.0.30319

Enterprise版本C:\Program Files (x86)\Microsoft Visual Studio\2019\Enterprise\MSBuild\Current\Bin

Professional 版本

C:\Program Files (x86)\Microsoft Visual Studio\2019\Professional\MSBuild\Current\Bin

### MSBuild命令行参数

/property:name=value或者/p:name=value 设置或重写指定项目级属性，其中 name 是属性名称，value 是属性值。 单独指定每个属性，或使用分号或逗号分隔多个属性。

如-property:WarningLevel=2;OutDir=bin\Debug

/t:Rebuild 每次都重建，不使用增量编译

/p:Configuration=Release 编译Release版本

/p:TargetFrameworkVersion=v4.5 编译的目标版本为.NET4.5

## 插件管理

插件是增强Jenkins环境功能以满足组织或用户特定需求的主要手段。使用MSBuild插件编译VS工程，使用NUint插件输出测试结果。

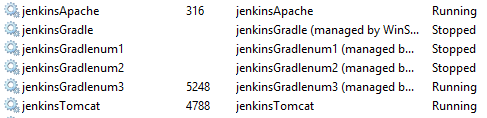
### 安装插件

Manage Jenkins 系统管理 -> 插件管理Plugin Manager -> 可选插件Available，搜索MSBuild和NUint，勾选插件，下载安装并**重启服务**。可在已安装中查看是否已安装。



### 重启服务

可在资源管理器中查看服务状态，直接用右键操作启动、停止、重启。



# 项目部署

## 新建项目New Item

新建任务New Item -> 输入任务名称 -> 选择 构建一个自由风格的软件项目 -> 确定

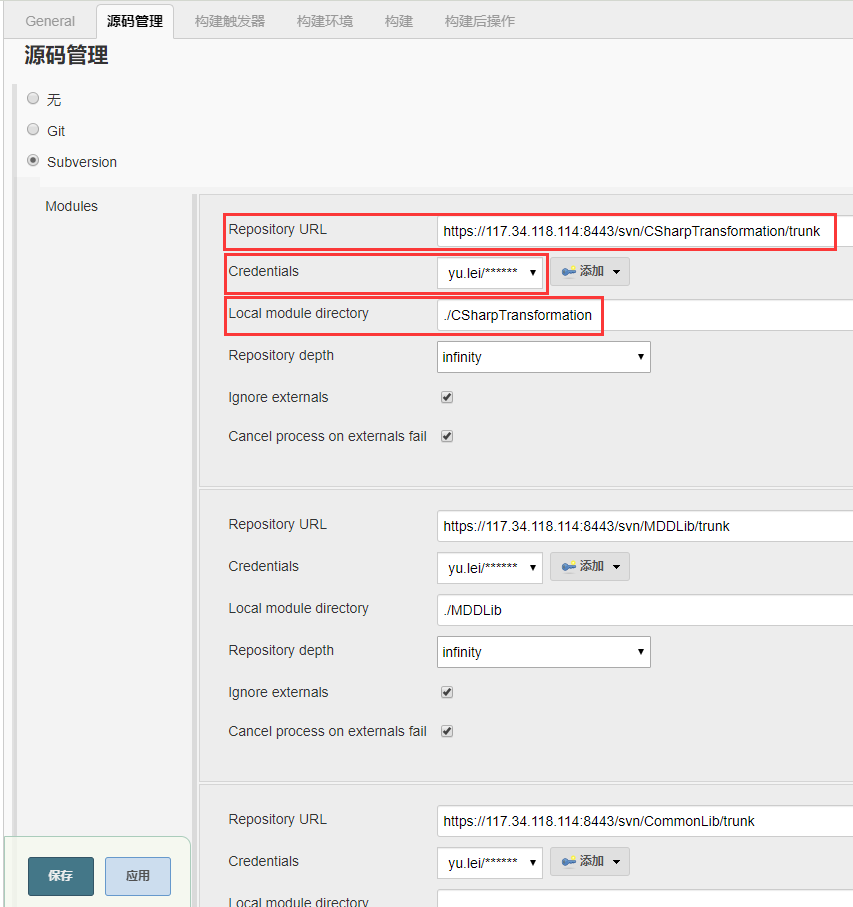


新建项目后就可以进行相关的设置，若设置完成后需要再次修改，可点击该项目左侧配置Configuration进行。工作空间Workspace可查该项目的所有代码和文件。



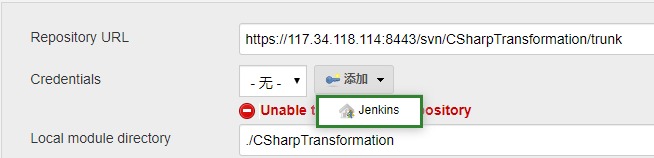
## 源码管理Source Code Management

进入 源码管理 -> 勾选 Subversion -> 填写Repository URL，Credentials，Local module directory。本地模块目录Local module directory，即保存代码的目录，可使用相对路径，可添加多个。



本地路径（Local module directory）的根目录路径是{Jenkins的安装目录}/workspace/{项目目录}，所以在该目录的各个下载目录其实就是直接在当前目录下定义目录名即可，例如 ./{下载目录名}

证书Credentials，即访问SVN的用户权限，第一次需添加新的，后续可直接选取。



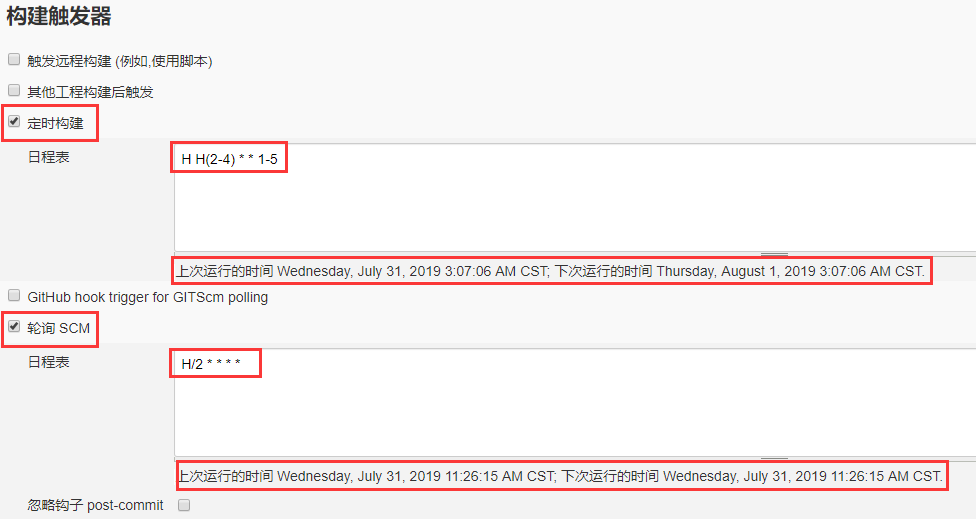
输入可访问上述SVN的有效用户名密码。



## 构建触发器Build Triggers

可以使用以下两种方式：

1. 定时构建Build periodically：在指定时间构建
2. 轮询SCM Poll SCM：定时检查源代码是否有新提交，如果有就执行构建，如H/30 \* \* \* \*表示每半小时检查一次



### 构建触发器语法的规则

每个规则包含用**空格**或**tab**分开的五个部分：MINUTE HOUR DOM MONTH DOW（注意空格）

|  |  |
| --- | --- |
| MINUTE | 分钟 (0–59) |
| HOUR | 小时 (0–23) |
| DOM | 天 (1–31) |
| MONTH | 月份 (1–12) |
| DOW | 星期 (0–7) 0和7是周日 |

可以使用以下操作符

|  |  |
| --- | --- |
| \* | 所有有效值 |
| M-N | 在M和N之间的值 |
| M-N/X | MN之间每隔X |
| A, B, C | 所列出的值，即A和B和C |
| H | H符号可理解为是随机时间 |

H即HASH是一个哈希值，可以理解为是一个随机的时刻，即在满足条件的时间内的任意时刻。H后可以跟括号表示该范围内的任意时刻。

### 构建触发器语法的实例

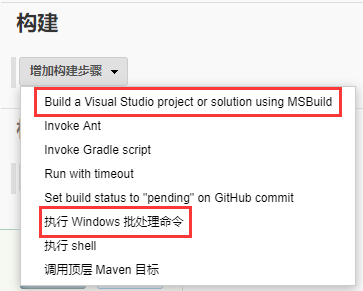
|  |  |
| --- | --- |
| \* \* \* \* H(6-7) | 每个周末触发一次，具体时间未知 |
| H/15 \* \* \* \* | 每隔15分钟触发一次，可能是:07, :22, :37, :52 |
| H H(2-5) 5,15,25 1-3 \* | 1到3月的每个5、15、25号的凌晨2到5点中的任意时刻 |
| H/30 9-18/2 H(1-9) 6 1-5 | 6月1到9号中的某个工作日的 9到18点中每隔2.5小时 |

## 构建Build

可以添加多个构建流程以按顺序构建不同的工程，可以使用以下两种方式进行构建：

1. 通过使用MSBuild构建VS工程Build a Visual Studio project or solution using MSBuild配置
2. 通过运行Windows批处理命令Execute Windows batch common配置

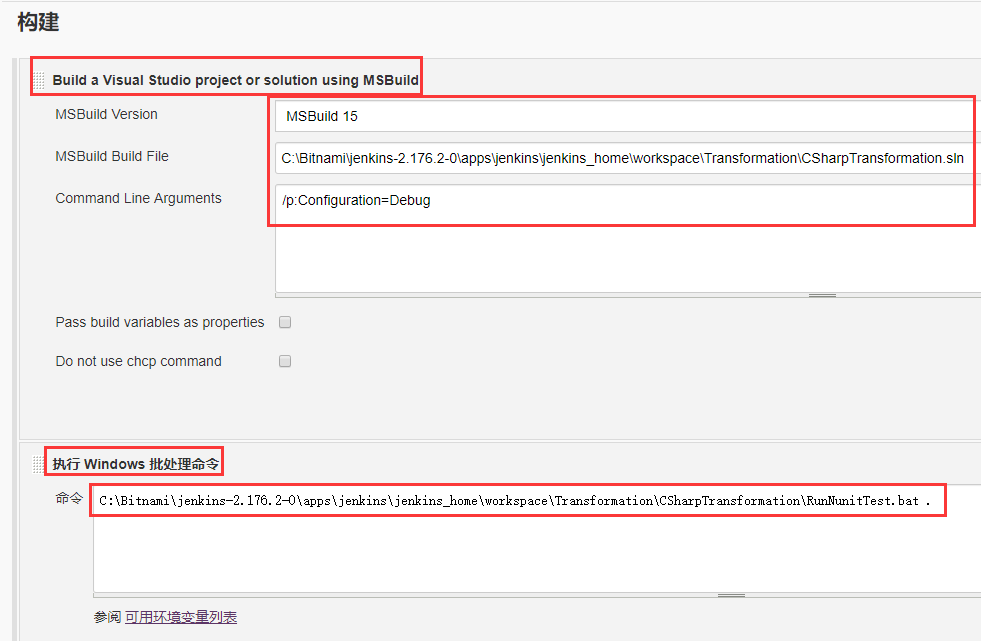
对于Sanjel这个特定项目，除了eServiceMDM是使用.Net Core开发，需使用bat批处理命令编译之外，其他统一使用MSBuild进行编译。



### Build a Visual Studio project or solution using MSBuild

通过使用MSBuild.exe构建VS工程。

MSBuild Version选择在之前全局工具里配置好的MSBuild，MSBuild Build File为编译的工程文件，需要使用**绝对路径**，编译参数参考3.1.2



### Execute Windows batch common

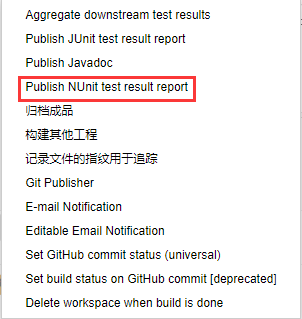
通过运行Windows批处理命令进行构建。

新建.bat文件，内容为：dotnet build xxx\xxx.sln（注意空格），此路径为构建文件的路径，可使用相对路径。

执行批处理命令：yyy\yyy.bat .（注意空格），此处的路径是批处理文件的路径。

## 构建后操作Post-build Actions

发布NUnit测试结果报告Publish NUnit test result report。

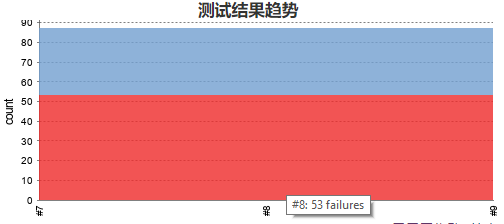


新建.bat文件，内容：%NUnit工具路径% %test工程目录%.dll /result:%测试结果%.xml（注意空格）。使用NUint工具运行.dll中的测试用例并将测试结果输出到.xml里。

Test report XMLs填入上面.bat文件里的测试报告xml文件名。

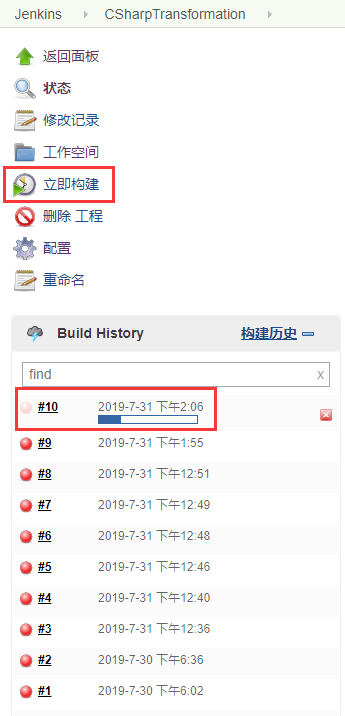


NUnit插件会自动获取测试结果.xml文件里的信息并进行展示。如下图，第8次构建中有53个测试用例失败。



# 运行构建Build Now

可等待触发器自动进行编译，也可点击 立即构建 进行编译，如图，本次构建是第10次，正在进行中等待完成

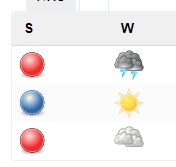


# 查看构建结果

点击某个构建查看构建的信息信息。变更记录为代码库提交记录；控制台输出为编译时的所有输出信息，包括编译失败的信息；Test Tesult为测试结果



红色圆球表示编译失败，蓝色圆球表示编译成功，灰色圆球表示编译终止；太阳表示近期成功率高，阵雨表示成功率低，多云表示成功率一般。



# 参考资料

官方主页：<https://jenkins.io/zh/>

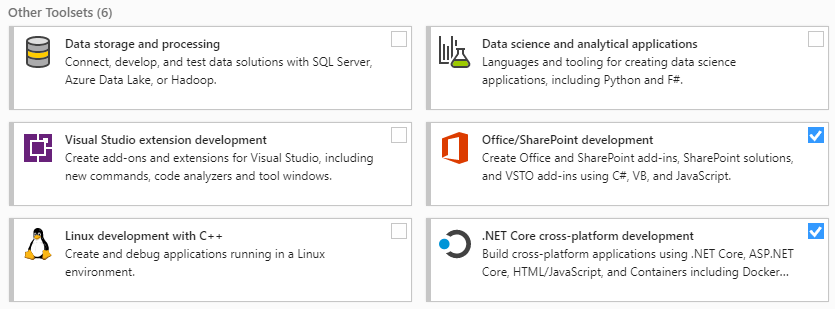
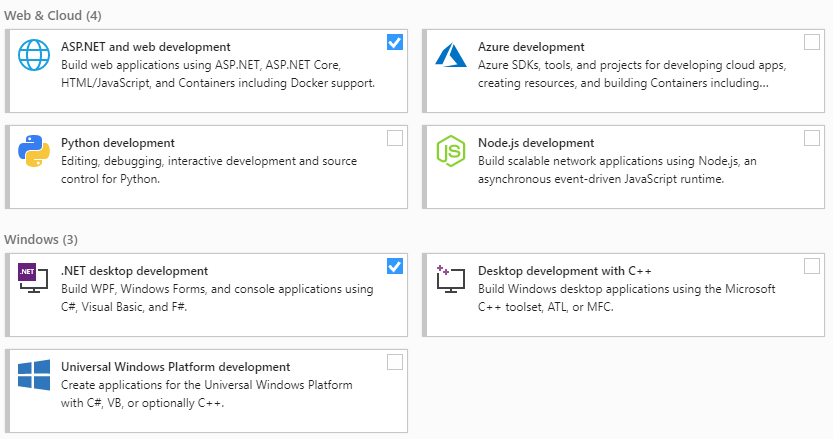
说明文档：<https://jenkins.io/zh/doc/>

GitHub：<https://github.com/jenkinsci/jenkins>

# 遇到的问题

## 运行Jenkins的机器需要安装.NET Framwork

安装Visual Studio与相关的组件，比如：



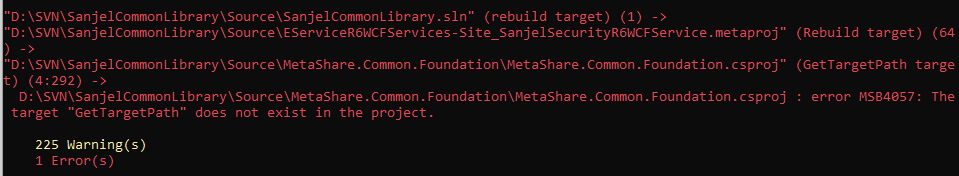
## .NET Framework和 .NET Core工程需要用不同的编译工具

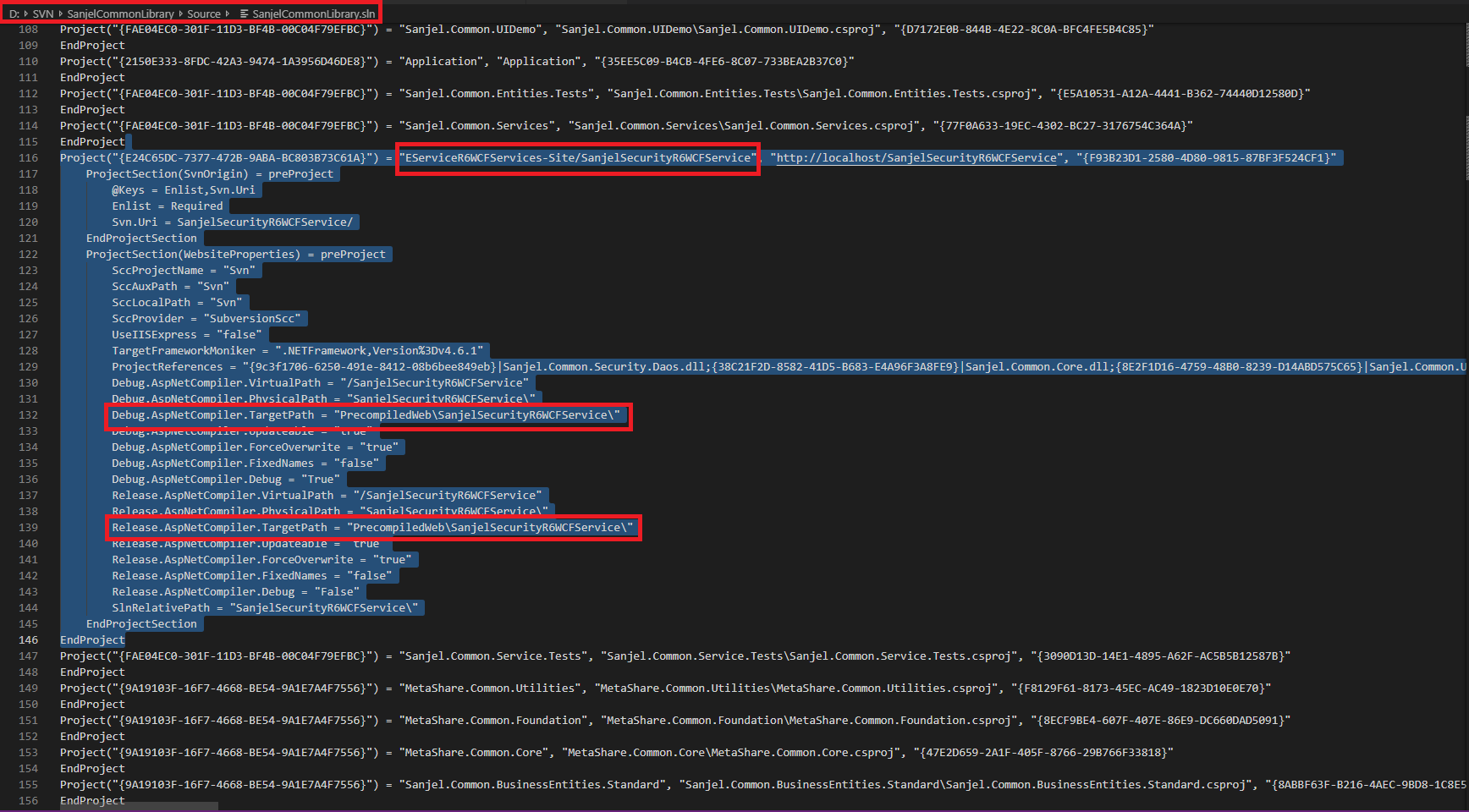
.NET Framework工程使用在全局工具配置好的MSBuild编译

.NET Core工程需要编写.bat脚本执行命令来编译：dotnet build xxx/xxx.sln

## Error MSB4057: The target “GetTargetPath” does not exist in the project

为解决





## 忘记管理员密码

1. 删除Jenkins目录下config.xml文件中的如下代码，保存文件，重启Jenkins服务 <useSecurity>true</useSecurity>

<authorizationStrategy class="hudson.security.FullControlOnceLoggedInAuthorizationStrategy">

<denyAnonymousReadAccess>true</denyAnonymousReadAccess>

</authorizationStrategy>

<securityRealm class="hudson.security.HudsonPrivateSecurityRealm">

<disableSignup>true</disableSignup>

<enableCaptcha>false</enableCaptcha>

</securityRealm>

1. Manage Jenkins系统管理 -> Configure Global Security全局安全配置 -> 勾选 启用安全 -> 勾选 Jenkins专有用户数据库 -> 点击 保存
2. Manage Jenkins系统管理 -> 管理用户 -> 用户列表 -> 点击右侧进入修改密码页面

修改后即可重新登录。

## 需检查代码完整

如缺少引用或代码库包或代码本身报错的情况下，编译会失败。除了检查Jenkins设置之外，还可使用Visual Studio检查代码本身的完整度是否可以编译。