自动化打包编译说明

|  |  |
| --- | --- |
| 编制 | 孟小庆 |
| 审批 |  |

**目 录**

[1. 简介 3](#_Toc383702726)

[1.1 术语和缩写 3](#_Toc383702727)

[1.2 参考资料 3](#_Toc383702728)

[2. 设计约束 3](#_Toc383702729)

[2.1 遵循的规范和标准 3](#_Toc383702730)

[2.2 运行环境 3](#_Toc383702731)

[2.2.1 硬件环境 3](#_Toc383702732)

[2.2.2 软件环境 3](#_Toc383702733)

[3. 流程描述 3](#_Toc383702734)

[4. 操作说明 4](#_Toc383702735)

[4.1 操作Jenkins 4](#_Toc383702736)

[4.1.1 配置SVN 5](#_Toc383702737)

[4.1.2 脚本执行 5](#_Toc383702738)

[4.2 代码运行环境提交 6](#_Toc383702739)

[4.3 打包脚本修改 6](#_Toc383702740)

[4.4 客户端代码编译（全编译、增量编译） 7](#_Toc383702741)

[4.5 控件数字签名 7](#_Toc383702742)

[4.6 客户端安装程序打包 8](#_Toc383702743)

[4.7 控件CAB包数字签名 8](#_Toc383702744)

[4.8 下载器版本修改 8](#_Toc383702745)

[4.9 程序发布 8](#_Toc383702746)

[5. 异常处理 9](#_Toc383702747)

[6. 修订记录 10](#_Toc383702748)

# 简介

本文主要介绍了使用jenkins持续集成系统，搭建客户端自动构建、自动部署、发通知布的整套平台。用于提高软件发布效率、保障软件开发质量。

## 术语和缩写

| **术语/缩写** | **含 义** |
| --- | --- |
| 持续集成 | 是一种软件开发实践，对于提高软件开发效率并保障软件开发质量提供了理论基础。 |
| jenkins | 是一个开源项目，提供了一种易于使用的持续集成系统。 |
| InstallShield | 是一款可以制作安装程序的开发套件 |

## 参考资料

1. [《基于 Jenkins 快速搭建持续集成环境》](http://www.ibm.com/developerworks/cn/java/j-lo-jenkins/)

# 设计约束

## 遵循的规范和标准

客户端持续集成依照jenkins系统规范，使用版本控制库保证提交的代码合法性。使用独立的集成服务器执行集成构建，开发人员代码提交到版本控制库。程序编译从版本控制库更新到本地。

## 运行环境

### 硬件环境

CPU：Intel/AMD 双核 2G及以上

内存：2GB 及以上

硬盘：80GB及以上

网卡：1000M网卡及以上

显示器：22英寸及以上

### 软件环境

Jenkins持续集成系统

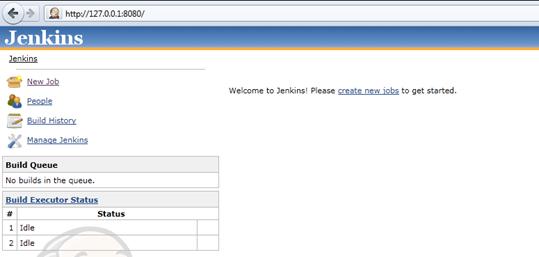
# 流程描述

依托jenkins系统自动定时、脚本执行模块，将自动化编译、打包流程分为：

1. 代码及运行环境文件提交。（可选）
2. 打包脚本修改（可选）
3. 客户端代码编译（全编译、增量编译）
4. 控件数字签名
5. 客户端安装程序打包
6. OCX安装程序打包
7. OCX安装程序数字签名
8. 下载器版本修改
9. 程序发布

# 操作说明

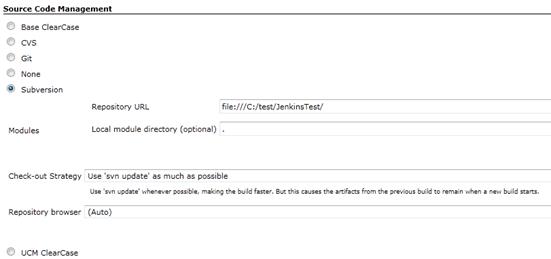
## 操作Jenkins



Jenkins主界面

Jenkins 是一个开源项目，提供了一种易于使用的持续集成系统，使开发者从繁杂的集成中解脱出来，专注于更为重要的业务逻辑实现上。同时 Jenkins 能实施监控集成中存在的错误，提供详细的日志文件和提醒功能。

### 配置SVN



根据 SVN 服务器服务器信息配置 Source Code Management，将需要编译的代码下载到本地。

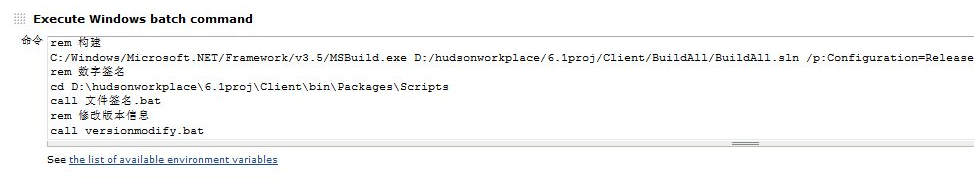
### 脚本执行

客户端构建使用批处理命令执行方式，将代码编译并生成EXE程序。

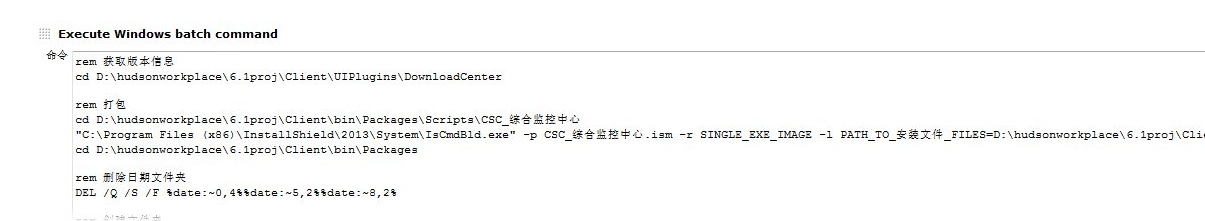
脚本执行流程：



1. SVN清理本地目录
2. 将服务器文件更新到本地。



1. 编译本地客户端工程，build模式。
2. 将编译后的控件执行控件签名。
3. 修改下载器版本注册信息。



1. 执行客户端InstallShiled打包
2. 将打包后的文件发布。

## 代码运行环境提交

客户端目前编译的代码版本是SVN trunk目录内，更新代码都需要提交到trunk。

客户端程序运行环境需要提交到bin\release内。相关DLL、配置文件。

编译生成的lib提交到: public\lib\release。Release目录不再保留。

## 打包脚本修改

**打包脚本路径：bin\Packages\Scripts。**

如果打包脚本需要新增、修改文件，需要在本地先使用InstallShiled修改脚本。

**客户端脚本:bin\Packages\Scripts\CSC\_综合监控中心\ CSC\_综合监控中心.ism**

**OCX脚本：bin\Packages\Scripts\ Ocx.ism**

将修改后的文件上传至SVN。重新执行打包流程。

## 客户端代码编译（全编译、增量编译）

客户端程序编译支持：全编译、增量编译模式。全编译使用visual studio 程序rebuild模式只支持编译客户端工程。不涉及打包。如果代码上传比较多建议先使用全编译模式，在使用增量编译。

全编译连接：<http://192.0.7.161:8080/jenkins/job/SIXPOTONEClientbuild/build?token=SIXPOTONEClientbuild>

增量编译链接：

<http://192.0.7.161:8080/jenkins/job/%E5%9F%BA%E7%BA%BF%E5%AE%A2%E6%88%B7%E7%AB%AF/build?token=clientbuildandpackage>

## 控件数字签名



客户端程序编译完毕会自动执行控件签名脚本。如果有需要新增、删除文件。请修改a.txt文件。增加删除控件名称。

格式如下：

**控件名称 控件描述**

FilePlayBackOcx.dll FilePlayBackOcx

AlmSinglePlaybackOcx.dll AlmSinglePlaybackOcx

## 客户端安装程序打包

目前打包程序使用的是InstallShiled编译提供，打包文件引用目录为: bin\release。

**在本地修改打包脚本不要跨目录引用文件。上传到服务器会导致路径找不到文件丢失。**

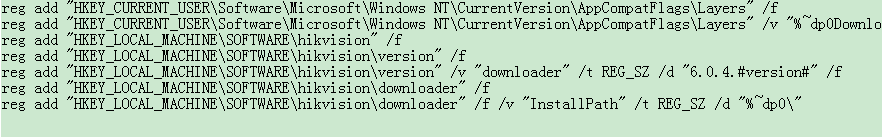
## 控件CAB包数字签名



在Scripts\CAB目录存放了需要生成CAB所需的文件。Cab签名批处理会将OCX.exe拷贝到该目录。执行数字签名操作。

## 下载器版本修改

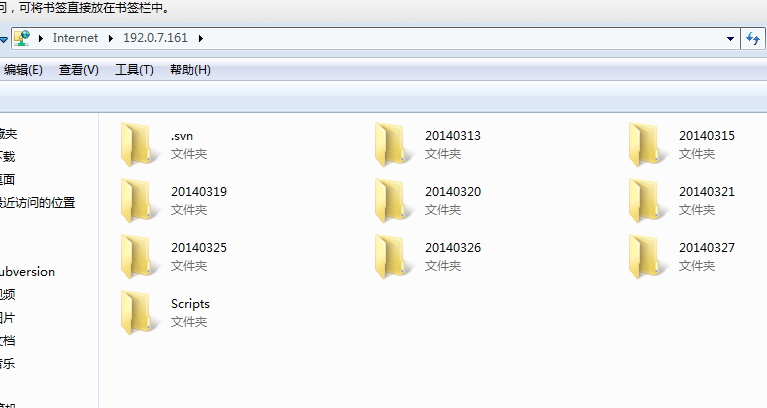


下载器版本信息安装时需要注册到客户机。为了每次都能获取到正确的版本信息。先使用批处理获取下载器BuildVersion。再替换目录bin\release\dlcenterreg.bat文件特殊字符串:#version#。

## 程序发布

客户端程序、OCX安装程序编译完成后，jenkins系统自动将程序发布到:

<ftp://192.0.7.161/>,用户名：密码。 admin:admin

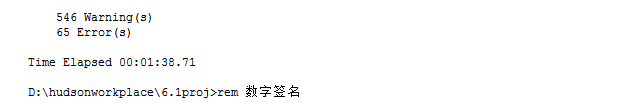


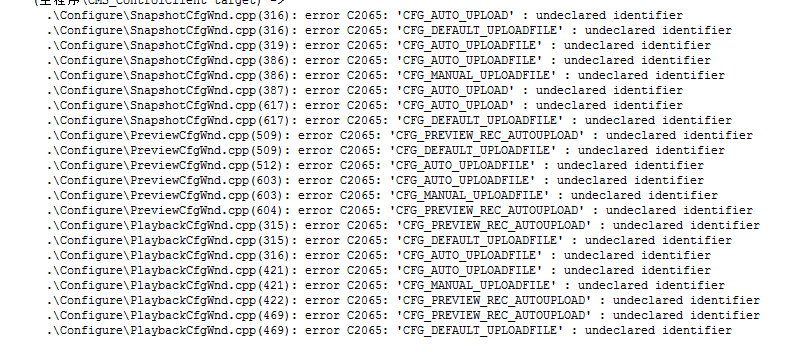
# 异常处理

1. 如果编译代码中出现异常jenkins不会停止脚本继续执行。这需要每次都检测控制台输出。

<http://192.0.7.161:8080/jenkins/job/%E5%9F%BA%E7%BA%BF%E5%AE%A2%E6%88%B7%E7%AB%AF/>







如果Error(s)大于零，表示提交的代码存在编译错误。需要重新检查提交。

1. 下载器文件dlcenterreg.bat版本信息不用手动替换。已使用脚本自动获取版本信息。
2. 由于服务器存在C++ 代码覆盖率工具Bullseye Coverage。会跟客户端编译存在冲突，请确保编译时代码覆盖率已停止，并使用全编译模式清理目录。

# 修订记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **变更时间** | **版本** | **变更人** | **变更说明** |
| 1 | 2014-03-27 | V0.0.1 | 孟小庆 | 创建文档 |
|  |  |  |  |  |