**ATS自动化测试系统**

**Version 1.0**

目录

[修订记录 3](#_Toc498676084)

[一、 概述 4](#_Toc498676085)

[Nose 简介 4](#_Toc498676086)

[ATS简介 4](#_Toc498676087)

[二、 测试用例命名规则 5](#_Toc498676088)

[三、 系统工作原理示意图 6](#_Toc498676089)

[四、 系统组成示意图 7](#_Toc498676090)

[五、 ATS使用方法及CLI命令说明 7](#_Toc498676091)

[配置： 7](#_Toc498676092)

[CLI登录： 9](#_Toc498676093)

[ATS命令说明： 9](#_Toc498676094)

[1. Help 9](#_Toc498676095)

[2. Listcases 9](#_Toc498676096)

[3. Listcase 10](#_Toc498676097)

[4. Crun 10](#_Toc498676098)

[5. Listsuites 11](#_Toc498676099)

[6. Listsuite 11](#_Toc498676100)

[7. Srun 11](#_Toc498676101)

[8. Runagain 12](#_Toc498676102)

[9. Createsuite 13](#_Toc498676103)

[10. Addcase 13](#_Toc498676104)

[11. Delcase 13](#_Toc498676105)

[12. Givereport 13](#_Toc498676106)

[13. Exit 13](#_Toc498676107)

[六、 测试用例开发demo 13](#_Toc498676108)

[七、 安装 13](#_Toc498676109)

# 修订记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **日期** | **版本** | **修改内容** | **作者** | **审核** |
| 8/30/2017 | v1.0 | 初稿 | 宫勋 |  |
| 10/09/2017 | v1.1 | 更新用例编号 | 宫勋 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# 概述

本测试系统是以nose为基本组件， 主要目的是针对公司的智能路由器产品进行自动化测试。主要功能特点如下：

* 简单易用
* 完善的日志功能
* 一目了然的测试报告
* 人性化的命令行界面
* 方便的用例与测试集合管理

## Nose 简介

nose是一个用于python单元测试的工具。nose 项目是于 2005 年发布的，也就是 py.test 改名后的一年。它是由 Jason Pellerin 编写的，支持与 py.test 相同的测试习惯做法，但是这个包更容易安装和维护，有更加丰富的插件。

nose的工作方式为：自动从指定目录下的python 文件，目录， 和类中搜索测试用例。任何文件，目录或者类只要满足正则表达式： *(?:\b|\_)[Tt]est* 都会被nose作为搜索对象， 而匹配它的函数或者方法会被为用例执行：

nosetests only\_test\_this.py

nosetests testdir

nosetests another.test:TestCase.test\_method

nosetests a.test:TestCase

nosetests /path/to/test/file.py:test\_function

## ATS简介

ATS是基于nose的自动化测试框架， 同时也是一个集成了一些测试辅助工具的工具箱。

ATS是通过nose来实现自动化测试的。具体来说是在内部调用了nose提供的命令来执行用例。ATS负责用例的管理，用例的执行则委托给nose。测试用例的实现是按照nose可识别的类中的方法来编写。ATS对该类做了4层抽象，分别为：

1. Framework：定义了测试用例的接口
2. Base: 通过充填实例化方法，实现测试用例的最基本的功能
3. Common\_methods：一些可复用的方法， 需要慢慢累积， 用以减少用例的开发周期
4. Case：由具体的测试用例提供， 需要实现Framework定义的接口

# 测试用例命名规则

‘ats\_’ + 8位用例编号 + 用例描述 + ‘\_test.py’

1.‘ats\_’与 ‘\_test.py’为固定格式

2. 8位用例编号：

前4位为功能模块编号， 后四位为用例编号。

2.1 目前已有的模块编号：

0000：一些辅助测试脚本，被ATS管理，并不是测试用例

0001：系统管理

0002：设置向导

0003：路由器上网

0004：无线设置

0005：子设备操作

0006：局域网管理

0007：风扇控制

0008：硬盘存储

0009：高级功能

2.2 用例编号：

第1位为用例分组， 后3位用于用例编号。比如子设备操作模块的用例编号，第一位为1为硬件方式添加子设备， 2为 软件方式添加子设备：

0005 1001：硬件方式添加子设备的第1个用例

0005 1002：硬件方式添加子设备的第2个用例

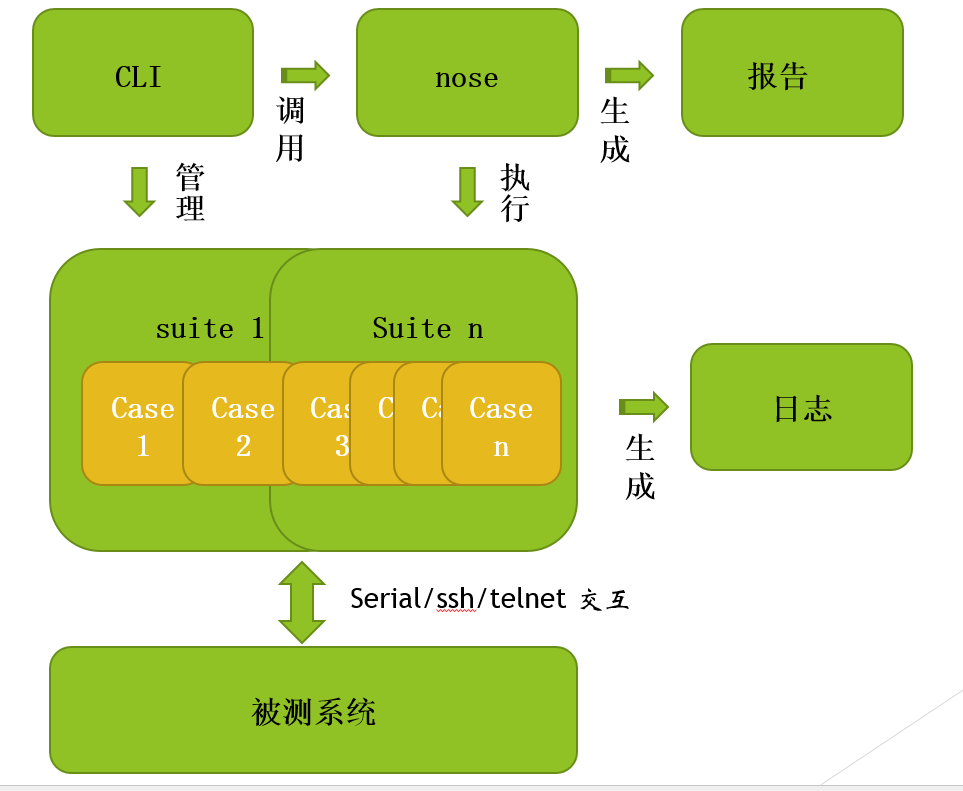
0005 2001：软件方式添加子设备的第1个用例

0005 2002：软件方式添加子设备的第2个用例

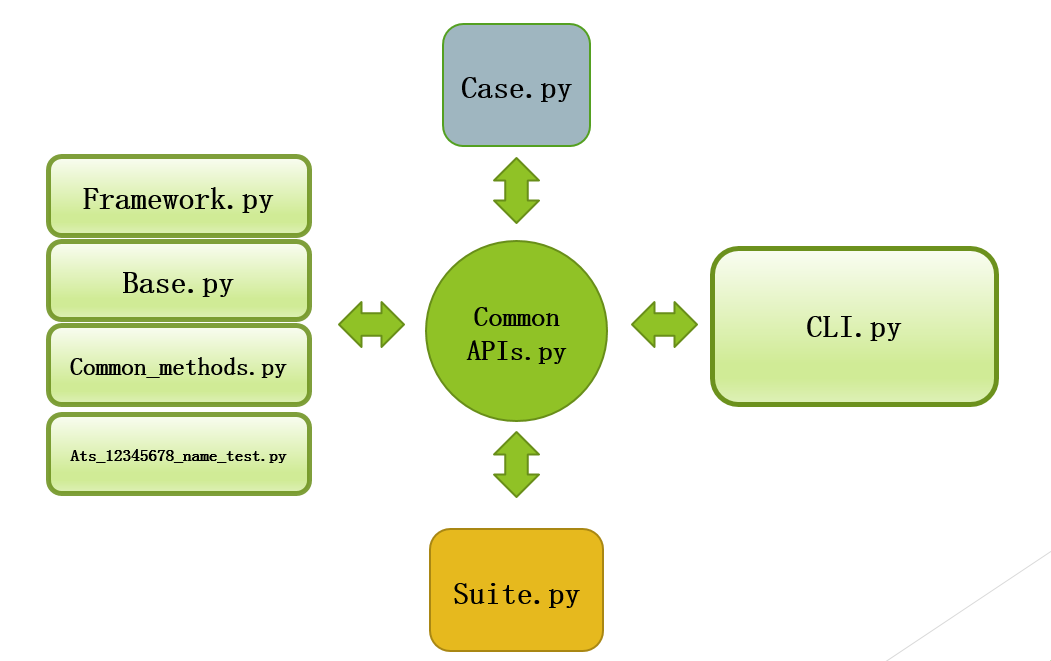
3. 用例描述

简单的字符串描述下用例的功能，字符串需满足正则表达式‘\w’

# 系统工作原理示意图



# 系统组成示意图



# ATS使用方法及CLI命令说明

## 配置：

系统的配置文件为ATS\config.ini， 可根据具体部署情况做修改：

[system]

#ATS系统所在目录

ats\_dir = C:\ATS\

#用例所在目录

case\_dir = C:\ATS\testcases\

#日志及报告目录

result\_dir = C:\ATS\testcases\ats\_result\

#below no need to care

ats\_path = C:\Python27\Lib\site-packages\

html\_tool\_path = C:\Python27\Lib\site-packages\htmloutput\

[serial]

#串口名字：COM1/COM2， 配置成any会自动选第一个可用的

port = any

baudrate = 115200

[network]

#智能路由器IP

host = 192.168.10.1

#智能路由器登陆用户名

user = root

#智能路由器密码

password = admin

prompt = ^.\*OpenWrt:.\*\#

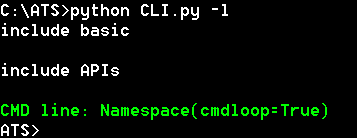
#智能路由器ssid name

ssid = BeeBox\_1414

ssid\_5G = BeeBox\_5G\_wifi\_test

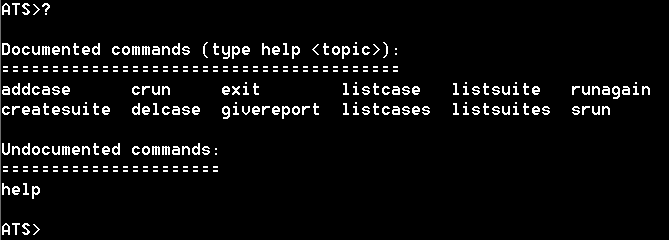
## CLI登录：

用命令行工具进入ATS目录， 运行命令python CLI.py –l 进入ATS系统命令行：



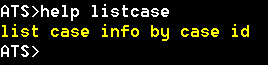
## ATS命令说明：

在ATS命令行敲入？可查看ATS支持的所有命令：



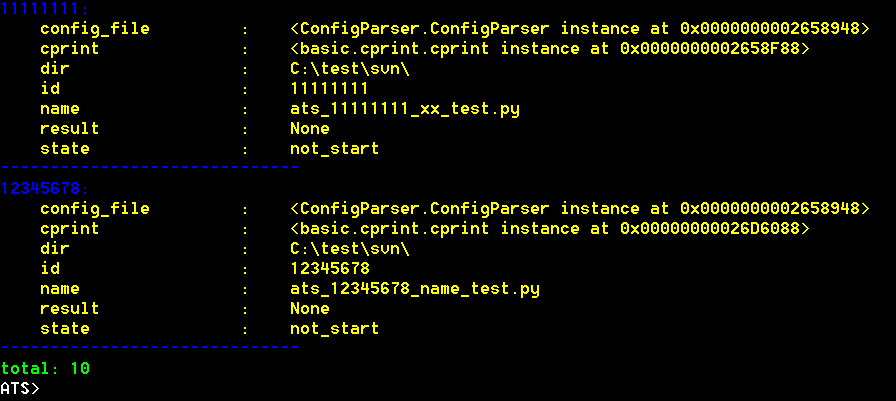
## Help

可以通过help命令查看命令的用法：



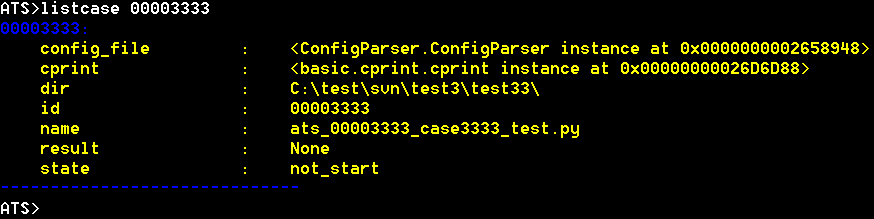
## Listcases

查看所有用例信息：



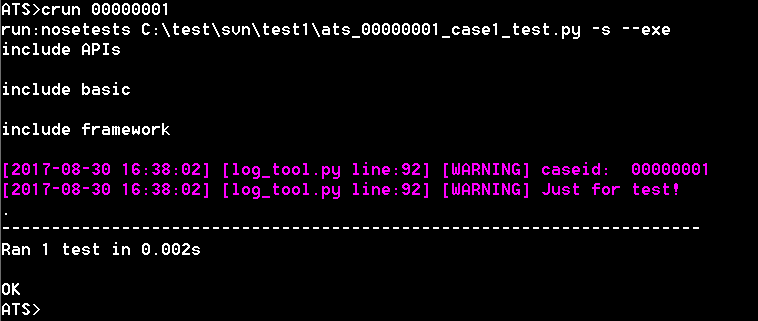
## Listcase

查看某个用例信息：



## Crun

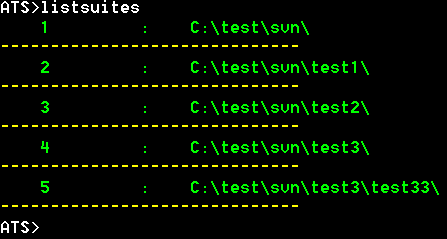
运行测试用例：



用例执行日志：C:\test\ats\_result\default\20170830\_163802-00000001\

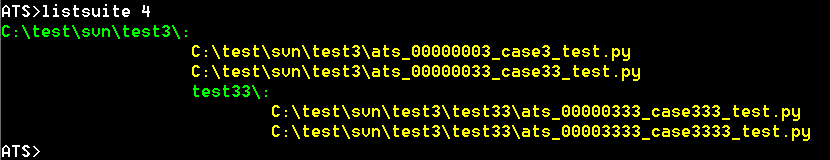
## Listsuites

查看当前所有测试集：



## Listsuite

查看某个测试集信息：

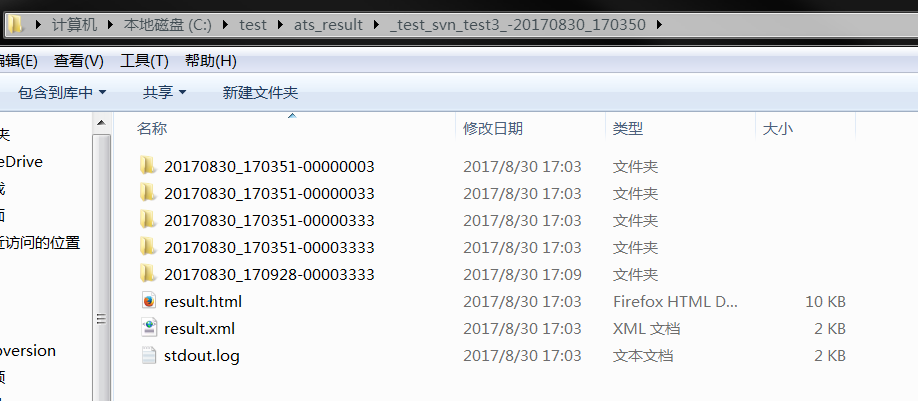


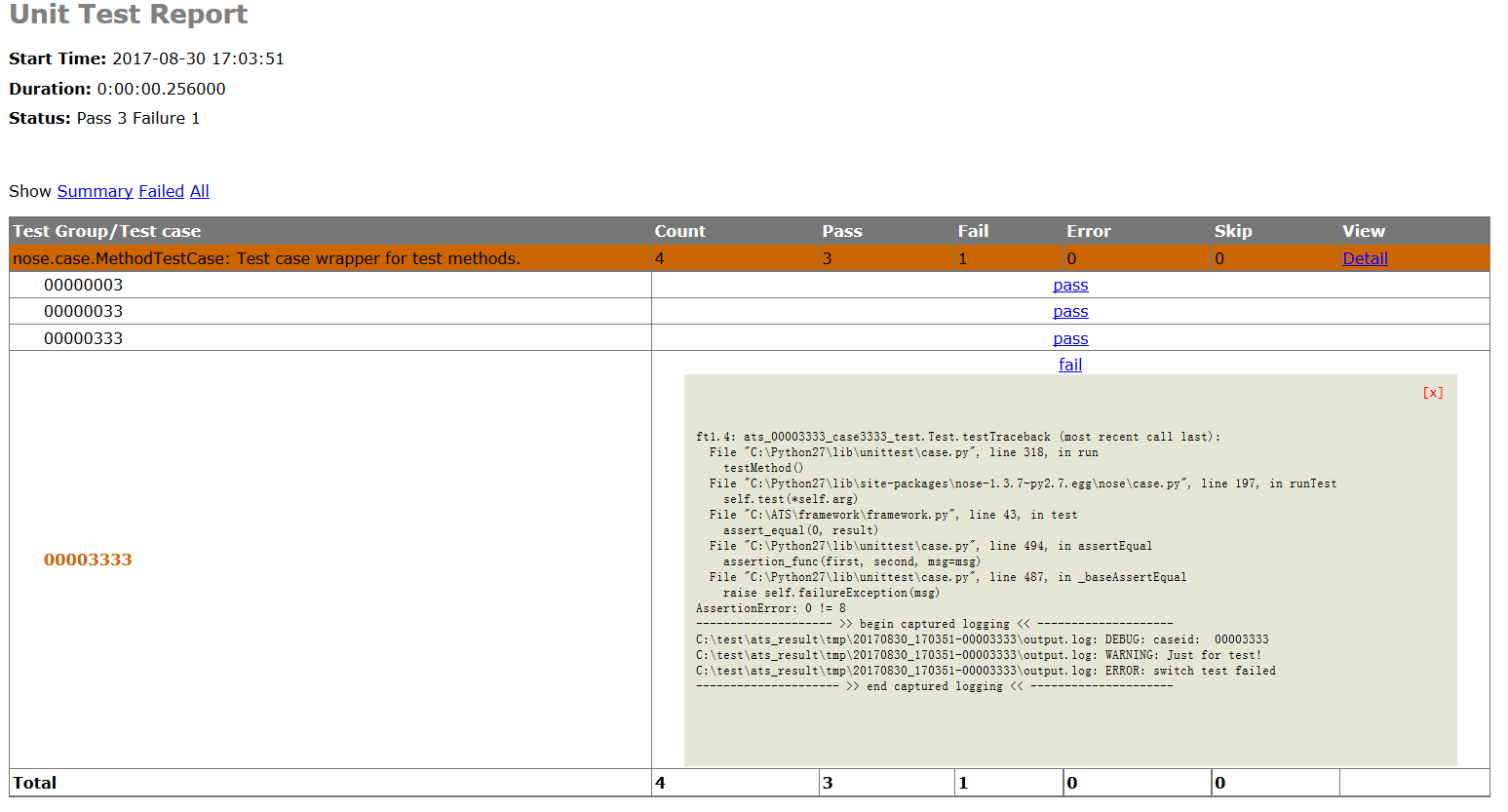
## Srun

用测试集id运行测试集, id 可用lsitsuites命令查看：



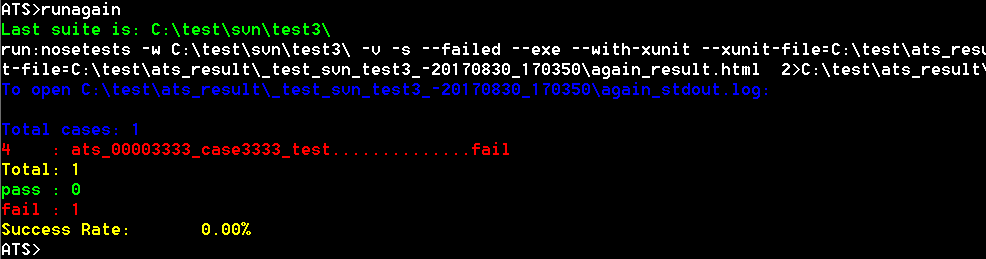
日志及测试报告目录：C:\test\ats\_result\\_test\_svn\_test3\_-20170830\_170350\





## Runagain

重新执行上一个测试集中所有的失败用例：



日志及报告仍在原目录

## Createsuite

创建新的测试集：

## Addcase

添加用例到新创建的测试集：

## Delcase

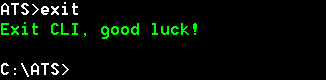
重新创建的测试集删除用例：

## Givereport

生成测试报告：

## Exit

退出系统：



# 测试用例开发demo

例如要开发一个wifi配置的用例：

1. 确定用例名字：

假定wifi模块的编号为： 0001， 用例的编号为： 0001

所以用例的名字为： ‘ats\_00010001\_wifidemo\_test.py’

1. 用例编写：

参考‘ats\_88888888\_demo\_test.py‘

1. 将用例放入相应测试集：

test0001\_wifi\_config\

1. 合入svn, 并提交审核

# 安装

前提：

1). 已安装python并正确设置了环境变量

2). 从SVN获取ATS代码

步骤：

1．双击安装ATS目录下的pycrypto-2.6.win-amd64-py2.7.exe 与 pywin32-221.win-amd64-py2.7.exe

2．命令行进入ATS目录运行pip install -r requirements.txt命令

3．根据具体情况修改ATS目录下的配置文件（config.ini）