**RobotFramework 自动化测试框架安装手册**

**--- create by shifr at 2013/08/15**

目录

[1 RobotFramework介绍 1](#_Toc362857076)

[1.1. 特性 1](#_Toc362857077)

[1.2. 跨平台性 2](#_Toc362857078)

[2. 软件安装介质 2](#_Toc362857079)

[2.1. python安装介质 2](#_Toc362857080)

[2.1.1. LINUX、Windows 2](#_Toc362857081)

[2.1.2. AIX 3](#_Toc362857082)

[2.1.3. HPUX 3](#_Toc362857083)

[2.2. RobotFramework安装介质 4](#_Toc362857084)

[2.3. Ride安装介质 4](#_Toc362857085)

[3. 软件安装 5](#_Toc362857086)

[3.1. IBM AIX环境安装 5](#_Toc362857087)

[3.1.1. Python安装 5](#_Toc362857088)

[3.1.2. RobotFramework 安装 7](#_Toc362857089)

[3.1.3. 测试用例验证 8](#_Toc362857090)

[3.1.4. 执行测试用例 10](#_Toc362857091)

[3.1.5. 测试用例输出 11](#_Toc362857092)

[3.2. HPUX环境安装 12](#_Toc362857093)

[3.2.1. python安装 12](#_Toc362857094)

[3.2.2. RobotFramework安装 12](#_Toc362857095)

[3.2.3. 测试用例验证 12](#_Toc362857096)

[3.2.4. 执行测试用例 13](#_Toc362857097)

[3.2.5. 测试用例输出 13](#_Toc362857098)

[3.3. LINUX环境安装 13](#_Toc362857099)

[3.3.1. python安装 13](#_Toc362857100)

[3.3.2. RobotFramework安装 14](#_Toc362857101)

[3.3.3. 测试用例验证 14](#_Toc362857102)

[3.3.4. 执行测试用例 14](#_Toc362857103)

[3.3.5. 测试用例输出 14](#_Toc362857104)

[3.4. Windows环境安装 15](#_Toc362857105)

# RobotFramework介绍

Robot Framework是一款python编写的功能自动化测试框架。具备良好的可扩展性，支持关键字驱动，可以同时测试多种类型的客户端或者接口，可以进行分布式测试执行。主要用于轮次很多的验收测试和验收测试驱动开发（ATDD）。

Robot Framework是开源软件，由Nokia Siemens Networks开发并提供支持。

## 特性

1.测试用例使用文本文件（TXT或者TSV文件）保存，使用制表符分隔数据。可以方便的使用任何文本编辑器，或者EXCEL编辑测试用例。也可以使用HTML格式创建用例。

2.测试用例中支持变量使用，可以使用IF语句和FOR循环语句。

3.可以利用“标签”功能对测试用例进行分类和有选择执行。

4.支持关键字驱动、数据驱动和行为驱动。

5.利用已有的关键字，测试人员可以创建自己需要的关键字，形成更高级别的行为。

6.测试执行报告和日志是HTML格式，容易阅读。

7.Robot Framework并不是一款像QTP一样的自动化测试工具。它是一个自动化测试框架，或者说是一个自动化测试平台。

8.提供了测试执行事件的监听接口，并且可以自定义接口中的脚本。例如，某个用例执行前，“start\_test”接口中的脚本就会被执行；用例执行结束后，“end\_test”接口中的脚本就会被执行。而测试人员可以自定义“start\_test”、“end\_test”这两个接口的脚本。

9.提供了命令行接口和XML格式的输出。可以与版本管理工具结合，进行持续集成。

10.Robot Framework是通过测试库识别被对象、操纵被测对象，有很多自带的或者第三方的开源测试库。例如，使用“selenium2Library”库测试web客户端。此外还可以测试java客户端、Win32客户端、SSH协议的字符终端等。

11.测试人员可以使用Python和java创建自己需要的测试库。

12.提供了远程测试执行接口，可以进行分布式测试执行。

## 跨平台性

RobotFramework框架具备跨平台部署，可以运行于IBM AIX，HPUX，LINUX、Windows等多种不同的操作系统平台。

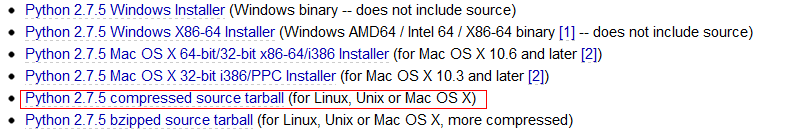
# 软件安装介质

由于RobotFramework是基于Python语言开发的，故在安装该介质前需要先安装Python软件。

## python安装介质

### LINUX、Windows

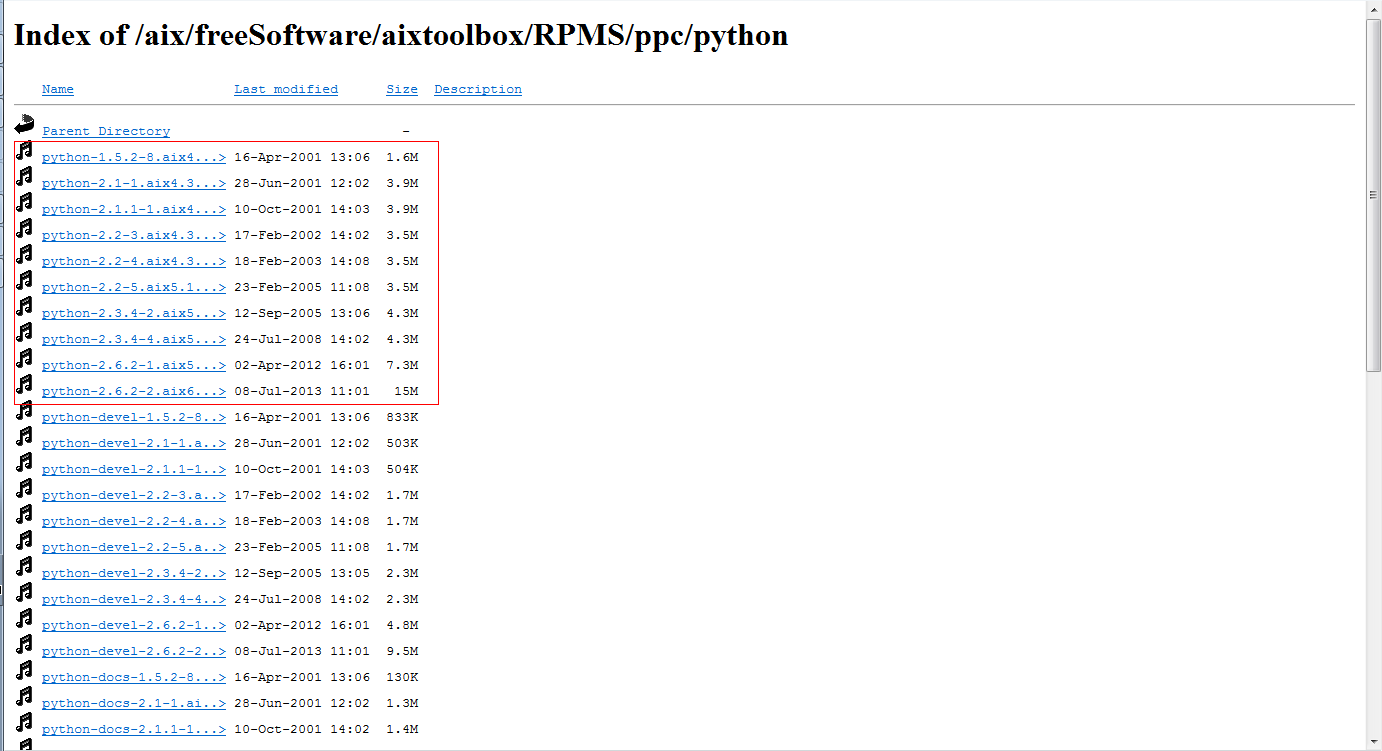
下载网址：http://www.python.org/download/



### AIX

AIX平台的安装比较特殊，需要从IBM官方获取，直接从python官方网站获取的无法安装，请特别注意。下载的时候请根据操作系统的版本选择对应的版本。

下载网址：ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/python/



### HPUX

HPUX环境也是需要从操作系统官方网站进行下载，如果是从Python官方网站下载的版本无法正常成功，需要特别注意。

下载网址：<http://hpux.connect.org.uk/hppd/hpux/Languages/python-2.7.5/>



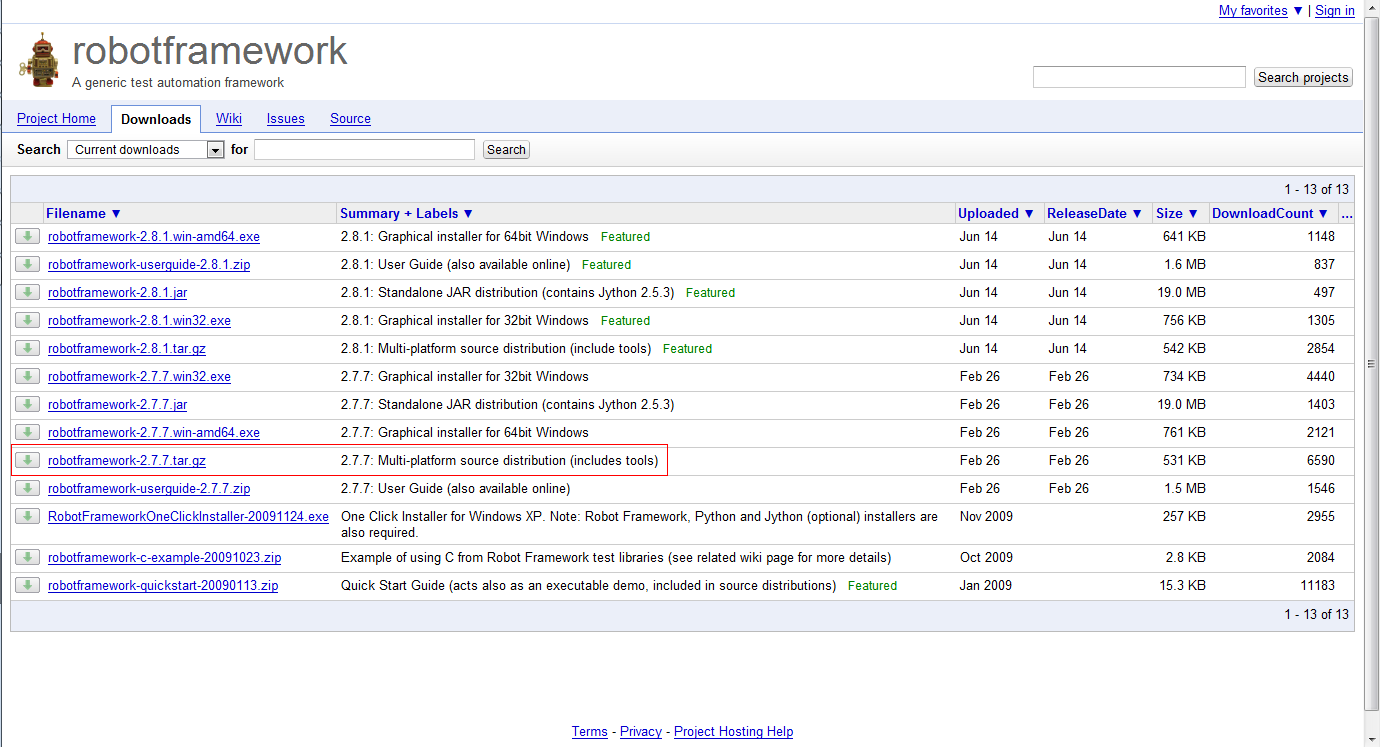
需要根据HPUX操作系统的具体版本选择对应的Python版本。

安装和运行时所需要的包，如果在安装和运行过程中有发现缺少库文件则需要到HP官方网站进行包文件下载并安装。



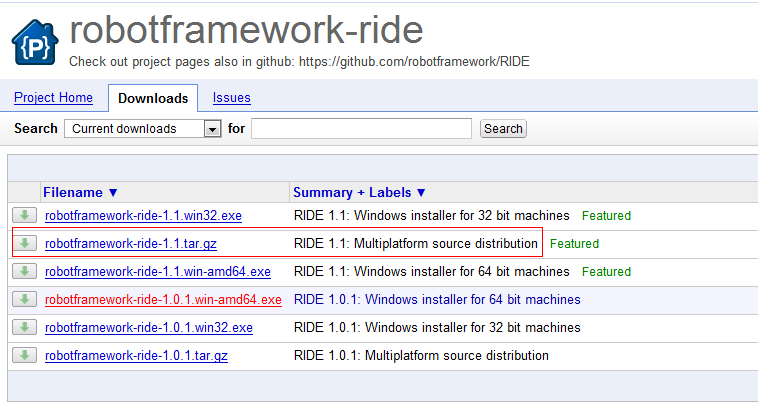
## RobotFramework安装介质

下载网址：http://code.google.com/p/robotframework/downloads/list



AIX、HPUX、LINUX系统的安装请选择多平台源代码发布的版本，windows系统的安装请根据操作系统的位数（32/64）选择具体的版本。

## Ride安装介质

下载网址：<http://code.google.com/p/robotframework-ride/downloads/list>

AIX、HPUX、LINUX系统的安装请选择多平台源代码发布的版本，windows系统的安装请根据操作系统的位数（32/64）选择具体的版本。

# 软件安装

## IBM AIX环境安装

### Python安装

* **需要使用root用户安装**
* **安装步骤**

#rpm -ivh python-2.6.2-2.aix6.1.ppc.rpm

error: failed dependencies:

libexpat.a(libexpat.so.1) is needed by python-2.6.2-2

libgdbm.a(libgdbm.so.3) is needed by python-2.6.2-2

libreadline.a(libreadline.so.4) is needed by python-2.6.2-2

安装python rpm包过程中出现以上错误，该错误是由于所需要的库文件不存在导致安装过程出错，解决办法如下：

从<ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/>网站上下载如下安装包

gdbm-1.8.3-5.aix5.2.ppc.rpm

readline-4.3-2.aix5.1.ppc.rpm

expat-2.0.1-2.aix5.3.ppc.rpm

安装库文件rpm包

#rpm -ivh gdbm-1.8.3-5.aix5.2.ppc.rpm

gdbm ##################################################

#rpm -ivh readline-4.3-2.aix5.1.ppc.rpm

readline ##################################################

#rpm -ivh expat-2.0.1-2.aix5.3.ppc.rpm

expat ##################################################

#

重新安装python

#rpm -ivh python-2.6.2-2.aix6.1.ppc.rpm

python ##################################################

#

安装完以上三个库文件后，python正常安装成功。

* **验证是否安装成功**

#python

Python 2.6.2 (r262:71600, Jun 1 2012, 22:50:06) [C] on aix6

Type "help", "copyright", "credits" or "license" for more information.

>>>

出现以上信息表示python已安装成功并正常运行。

### RobotFramework 安装

* **需要使用root用户安装**
* **安装步骤**

#python setup.py install

running install

error: invalid Python installation: unable to open /opt/freeware/lib/python2.6/config/Makefile (No such file or directory)

==》解决办法如下：

#mkdir config

#cd config

# > Makefile

#python setup.py install

running install

error: invalid Python installation: unable to open /opt/freeware/include/python2.6/pyconfig.h (No such file or directory)

==》解决办法如下：

#cd /opt/freeware/include/python2.6

#ls py\*

ls: 0653-341 The file py\* does not exist.

#

#> pyconfig.h

再次安装

#python setup.py install

running install

running build

running build\_py

running build\_scripts

running install\_lib

creating /opt/freeware/lib/python2.6/site-packages/robot

copying build/lib/robot/\_\_init\_\_.py -> /opt/freeware/lib/python2.6/site-packages/robot

...............................（中间部分省略）

running install\_scripts

copying build/scripts-2.6/pybot -> /opt/freeware/bin

copying build/scripts-2.6/rebot -> /opt/freeware/bin

changing mode of /opt/freeware/bin/pybot to 755

changing mode of /opt/freeware/bin/rebot to 755

running install\_egg\_info

Writing /opt/freeware/lib/python2.6/site-packages/robotframework-2.7.7-py2.6.egg-info

#

* **验证安装是否成功**

#pybot --version

Robot Framework 2.7.7 (Python 2.6.2 on aix6)

出现以上信息表示pybot已安装成功并正常运行。

### 测试用例验证

本文档的重点在于描述测试框架及所需要软件的安装，至于测试用例如何构造、测试逻辑开发等相关内容则在测试用例编写文档体现。

* **编写要验证的函数**

#cat hello.cpp

#include <stdlib.h>

#include <stdio.h>

int main(int argc, char \*\* argv)

{

if (1 == atoi(argv[1]))

{

printf("hello");

return 0;

}

else

printf("goodbye");

return -1;

}

编译：

xlc –g –o hello hello.cpp

* **测试用例**

#more hello.tsv

\*Settings\*

Library hello.py

\*Test Cases\*

hello helloworld

* **测试用例验证逻辑**

#more hello.py

#coding:gb18030

import time

import commands

import os

import sys

def helloworld():

dir = os.environ['HOME']

propath = dir+ '/' + 'shifr/autotest'

os.chdir(propath)

(status,output)=commands.getstatusoutput('hello 2')

if output == "hello":

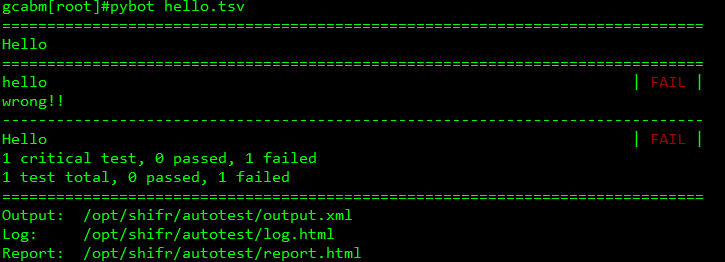
print ("hello is started successfully!")

else:

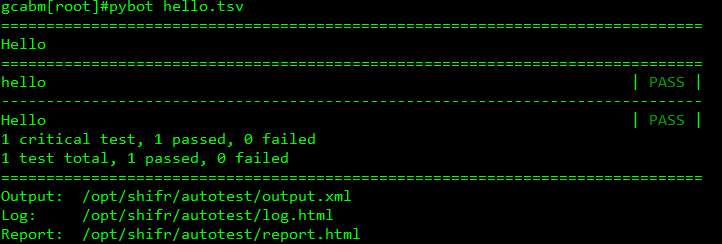
raise AssertionError("wrong!!")

### 执行测试用例

* **测试失败案例**

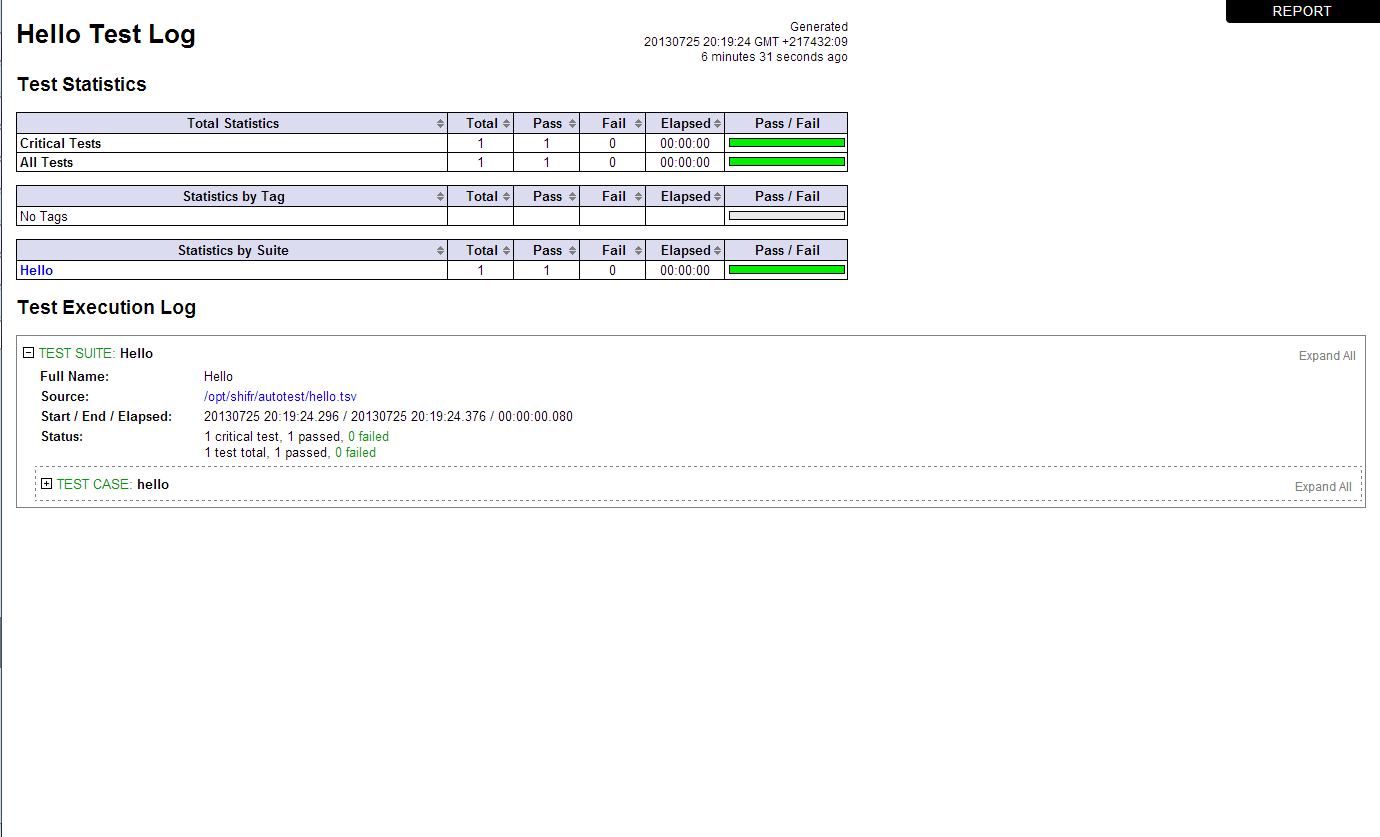
****

* **测试成功案例**

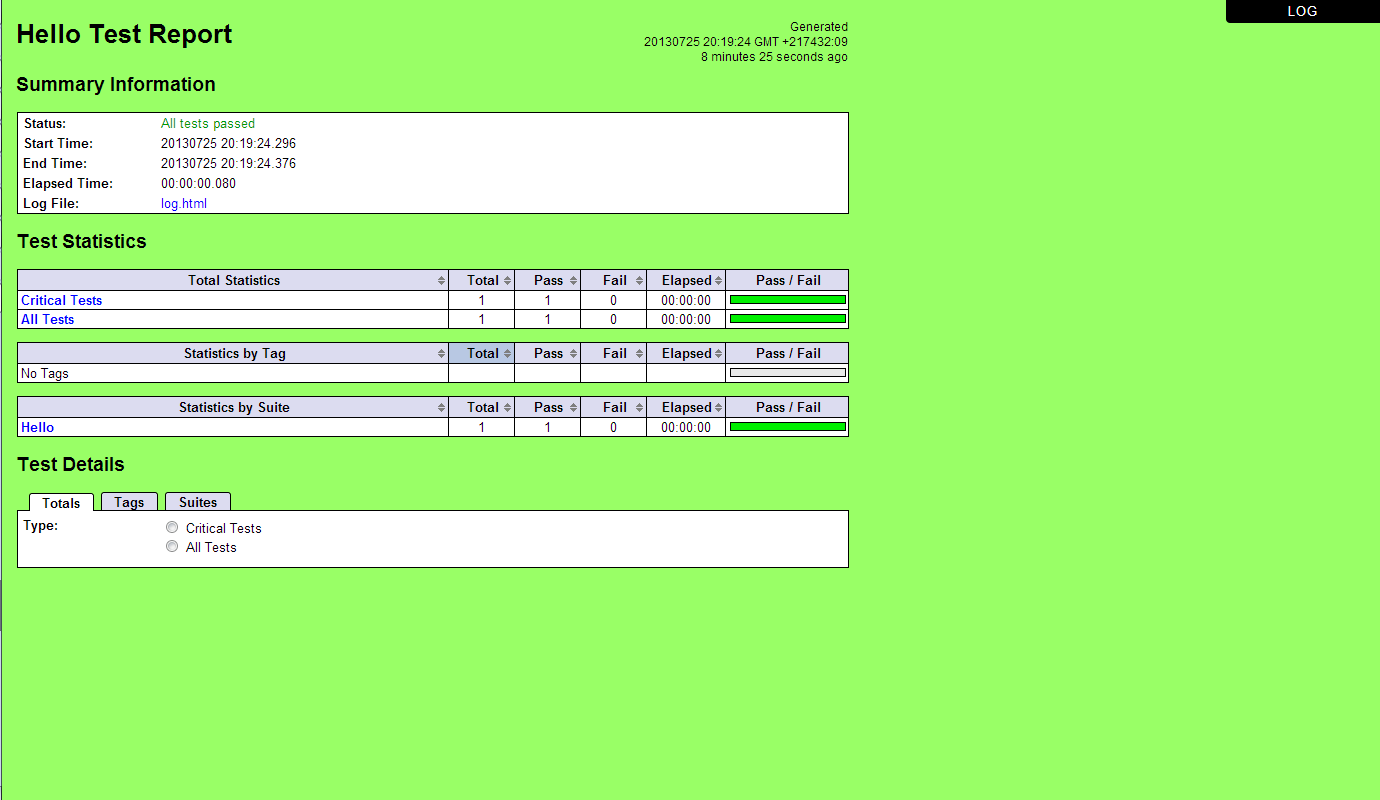


### 测试用例输出

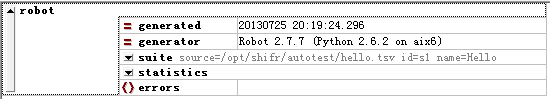
* **测试日志**



* **测试报告**



* **测试输出（output.xml）**



## HPUX环境安装

### python安装

* **需要使用root用户进行安装**
* **安装步骤**

swinstall -s /home/test/python-2.7.3-ia64-11.31.depot

在出现的界面中按空格选中python-2.7.3，执行action菜单中的install.

* **验证安装是否成功**

# /usr/local/bin/python  
Python 2.5.2 (r252:60911, Mar 12 2008, 18:22:03) [C] on hp-ux11  
Type "help", "copyright", "credits" or "license" for more information.  
>>> print "hello HPUX"  
hello HPUX

出现以上信息表示python已在hpux上安装成功并正常运行。

### RobotFramework安装

安装步骤同AIX环境一致，详见 [RobotFramework 安装](#_RobotFramework_安装)

### 测试用例验证

具体详见 [测试用例验证](#_测试用例验证_1)

HPUX环境下的C++源代码编译方式如下：

aCC –o hello hello.cpp

### 执行测试用例

具体详见 [执行测试用例](#_执行测试用例)

### 测试用例输出

具体详见 [测试用例输出](#_测试用例输出)

## LINUX环境安装

### python安装

* **需要使用root用户进行安装**
* **安装步骤**

1. 解压tar包

tar -xvf Python-2.7.5.tgz

1. 配置

cd Python-2.7.5.tgz

# ./configure --without-pymalloc --prefix=/opt/freeware --with-cxx=xlC\_r --enable-shared --with-threads --disable-ipv6

### 其中--prefix该选项来指定python的安装目录，需要安装的时候

1. 编译源代码

# make

1. 安装

# make install

* **验证安装是否成功**

[root@localhost ~]# python

Python 2.7.5 (default, Jul 26 2013, 14:16:32)

[GCC 4.1.2 20071124 (Red Hat 4.1.2-42)] on linux2

Type "help", "copyright", "credits" or "license" for more information.

>>>

出现以上信息表示python已在linux上安装成功并正常运行。

### RobotFramework安装

安装步骤同AIX环境一致，详见 [RobotFramework 安装](#_RobotFramework_安装)

### 测试用例验证

具体详见 [测试用例验证](#_测试用例验证_1)

HPUX环境下的C++源代码编译方式如下：

g++ –o hello hello.cpp

### 执行测试用例

具体详见 [执行测试用例](#_执行测试用例)

### 测试用例输出

具体详见 [测试用例输出](#_测试用例输出)

## Windows环境安装

Windows环境的安装都是图形界面，一路 next，基本都可以完成，本文档不再详细说明。

需要注意，在Win环境下，Ride软件安装前需要先安装wxPython，否则会出现安装失败。