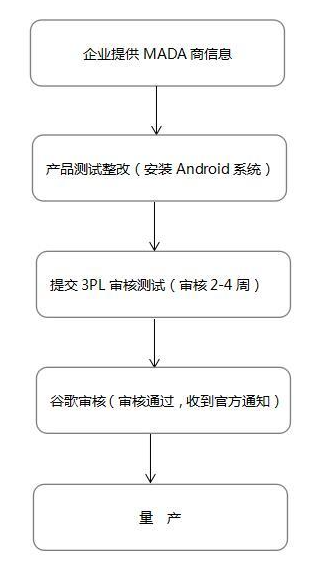
# GMS认证流程：

1. 测试机品牌商是否能够签MADT协议
2. 需要确定签好MADT协议或Goole同意签MADT协议，我们才做，一般报价3-4万美金，签MADT协议目前需要2-3个月时间；MADT协议时品牌商与谷歌签订的协议；
3. 根据CTS、GTS、加上谷歌的应用软件（APK）做正式测试

时间：6周

样品：10台样机、2台谷歌留着

1. 只有完全测试通过的测试报告才会提交给谷歌认证中心
2. 获取谷歌法律授权：

step4提交通过审核后，产品会获得谷歌的徽标logo授权，产品也会被谷歌放在其认证通过的全球列表中

1. MADA协议申请流程：

MADA：Mobile Application Distribution Agreement，移动应用发布协议，谷歌跟使用Android系统的厂商签定的一个协议

详细测试内容安排：

1. GMS认证测试主要包含：CTS、GTS、CTS Verifier测试；

(总的一共有：CTS、CTS Verify（手工测试用例）、GTS、VTS、STS、GVS、BTS，尽量用18.04的Ubuntu，如果目前22的无法测试通过，重新安装18.04的试试，22的容易因为网络原因Fail)

CTS:最高兼容最低，64位测试机，只能用64位的arm包跑，32位的测试机，需要用64位和x86的都跑。

1. CTS测试：

Compatibility TestSuite，兼容性测试，主要测试API的兼容性，保证API的兼容性，最终达到各种应用程序可以兼容的在不同平台上运行；

1. GTS测试：

Google Mobile Services Test Suite，谷歌移动服务，当用户使用谷歌的时候，谷歌可以把广告嵌入各种谷歌的服务中，保障谷歌的服务能够在产品上正常运行；

1. 手动CTS Verifier测试：

CTS Verifier是CTS测试中剥离出来的手动测试部分，主要用于测试那些自动测试系统无法测试的功能，相机、传感器等等，因为硬件配置原因，不同手机上部分测试项目不会显示出来，CTS Verifier中测试用例的总数，取决于测试机支持的功能；

# GMS基础环境变量的配置

### 1、兼容32bit版本

CTS测试需要兼容32bit版本，而Ubuntu系统大多只有64bit

先需要进行如下操作：

控制台输入

sudo apt-get update

然后控制台输入（兼容adb）

sudo apt-get install lib32stdc++6

然后控制台输入（兼容aapt）

sudo apt-get install zlib1g:i386

注意这个过程中需要按Y确认

### java环境配置

根据测试机器的Android版本选择对应的Java版本：



1、Jdk可到oracle的官网下载：

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

2、或可以直接使用命令安装Java：

sudo apt install -y openjdk-8-jdk

sudo apt install -y openjdk-11-jdk

3、添加用户环境变量

在控制台输入

sudo gedit ~/.bashrc

在打开的文档末尾加入如下内容：

（直接复制粘贴，然后保存关闭，需要切换Java版本时将蓝色处更换为目标Java版本所在的文件夹名即可）

export JAVA\_HOME=/usr/lib/jvm/jdk1.8.0\_121  
export JRE\_HOME=${JAVA\_HOME}/jre   
export CLASSPATH=.:${JAVA\_HOME}/lib:${JRE\_HOME}/lib   
export PATH=${JAVA\_HOME}/bin:$PATH

配置后，保存，在命令行输入source ~/.bashrc更新环境变量

4、检测java环境是否配置完成

控制台输入

java -version

配置成功的话会出现

java version XXXXXXX

控制台输入

javac -version

配置成功的话会出现

javac XXXXXX

### SDK环境配置

1. 解压SDK包

下载SDK包：<http://tools.android-studio.org/index.php/sdk>

拷贝并解压到任意位置（注意SDK的存放路径中不能有中文），打开解压后的文件夹，进入tools文件夹，命令行启动，输入android update sdk --no-ui进行SDK的安装和更新

2.添加用户环境变量

在控制台输入

sudo gedit ~/.bashrc

在打开的文档末尾加入如下内容：

（注意红色字体要修改成你的SDK存放路径，可进入SDK路径然后按命令行输入pwd查看路径，然后直接复制粘贴，保存关闭。）

export ANDROID\_HOME=/home/wuchaolin/SDK/android-sdk-linux

export PATH=$ANDROID\_HOME/tools:$ANDROID\_HOME/platform-tools:$PATH

export PATH=$ANDROID\_HOME/build-tools/23.0.2:$PATH

export PATH=$ANDROID\_HOME/tools:$PATH

3.检测SDK环境是否配置完成

控制台输入

adb version

配置成功的话会出现

Android Debug Bridge version XXXX

控制台输入

aapt v

配置成功的话会出现

Android Asset Packaging Tool , XXXXXXX

### 配置手机的rules

若不配置rules，在Ubuntu下adb devices无法正常连接新机型可能无法被识别。

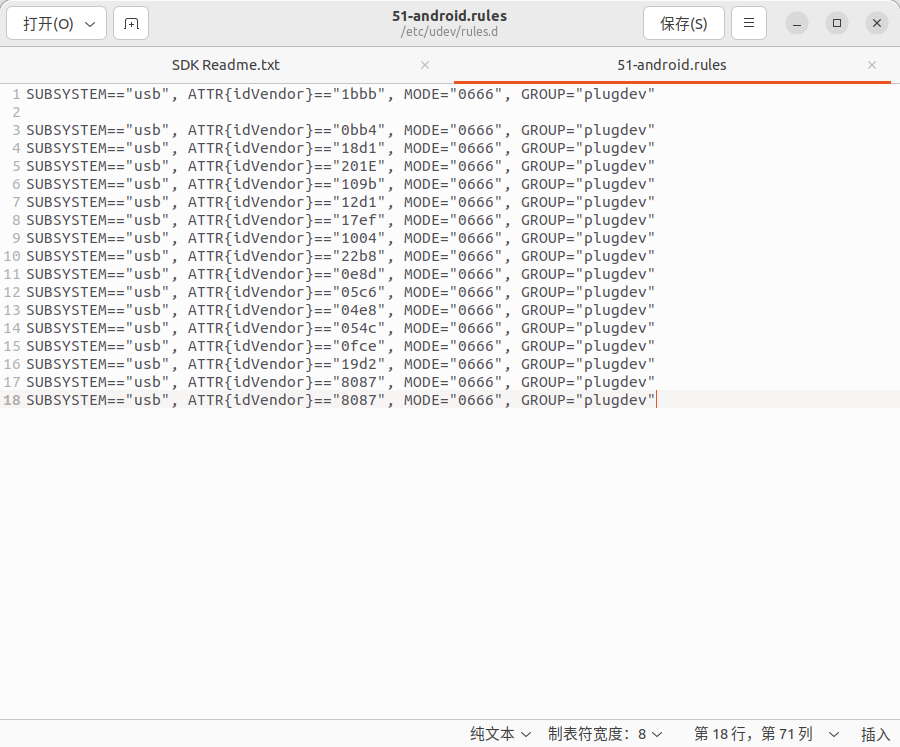
配置方法：

1、sudo mv /home/meizu/GMSTools/51-android.rules /etc/udev/rules.d/

1. sudo chmod -R 777 /etc/udev/rules.d/

adb devices获取idVendor填入即可，将如下改成对应的测试机的APP然后填入后粘贴到51-android.rules后保存

SUBSYSTEM=="usb", ATTR{idVendor}=="1bbb", MODE="0666", GROUP="plugdev"



（以下测试基于Ubuntu系统，以及以上基础配置环境都配置完成后才可进行测试）

# CTS测试指导：

（整个测试目录不能带有中文）

1. 下载与测试机器对应版本的测试包：

CTS 兼容性测试套件对应下载：

<https://source.android.com/compatibility/cts/downloads>

CTS测试媒体文件下载（最新的即可）：

<https://source.android.com/compatibility/cts/downloads#cts-media-files>

1. 测试机器需要进行如下配置：
2. 确保wifi可访问google服务且稳定，设置VPN，确保IPV6可用。  
    b、确保测试设备附近有可用的蓝牙设备。  
    c、如果不是刚刷机或者恢复出厂设置的的手机，需要设置Settings > Backup & reset > Factory data reset  
    d、语言要设置为英语，Settings > Language & input > Language  
    e、要打开位置，Settings > Location > On  
    f、Settings > Security > Screen lock > None  
    g、Settings > Developer options > USB debugging  
    h、Settings > Date & time > Use 24-hour format > Off  
    i、Settings > Developer options > Stay Awake > On  
    j、运行CTS测试媒体文件中的脚本：

sh copy\_images.sh和sh copy\_media.sh

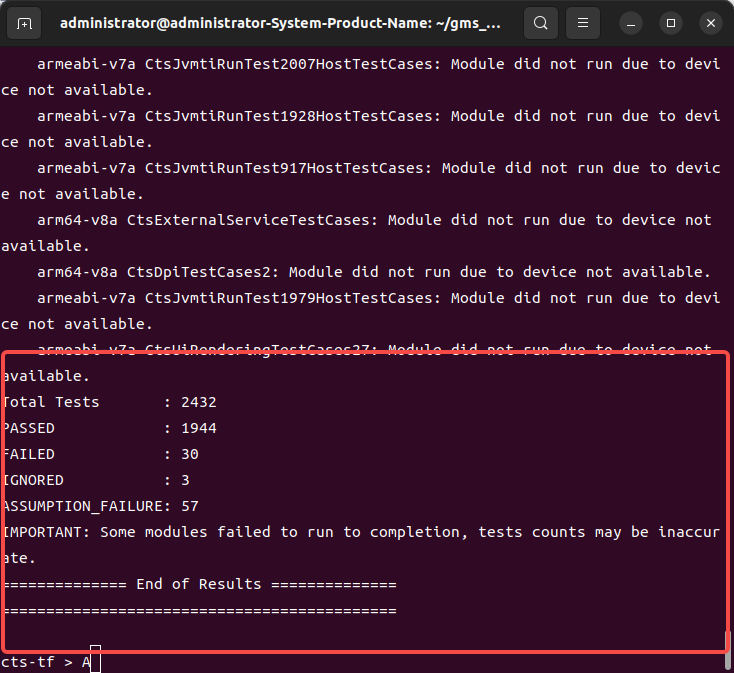
1. 进入解压好的CTS 兼容性测试套件中：

打开命令行进入/android11/android-cts-11\_r8-linux\_x86-arm/android-cts/tools

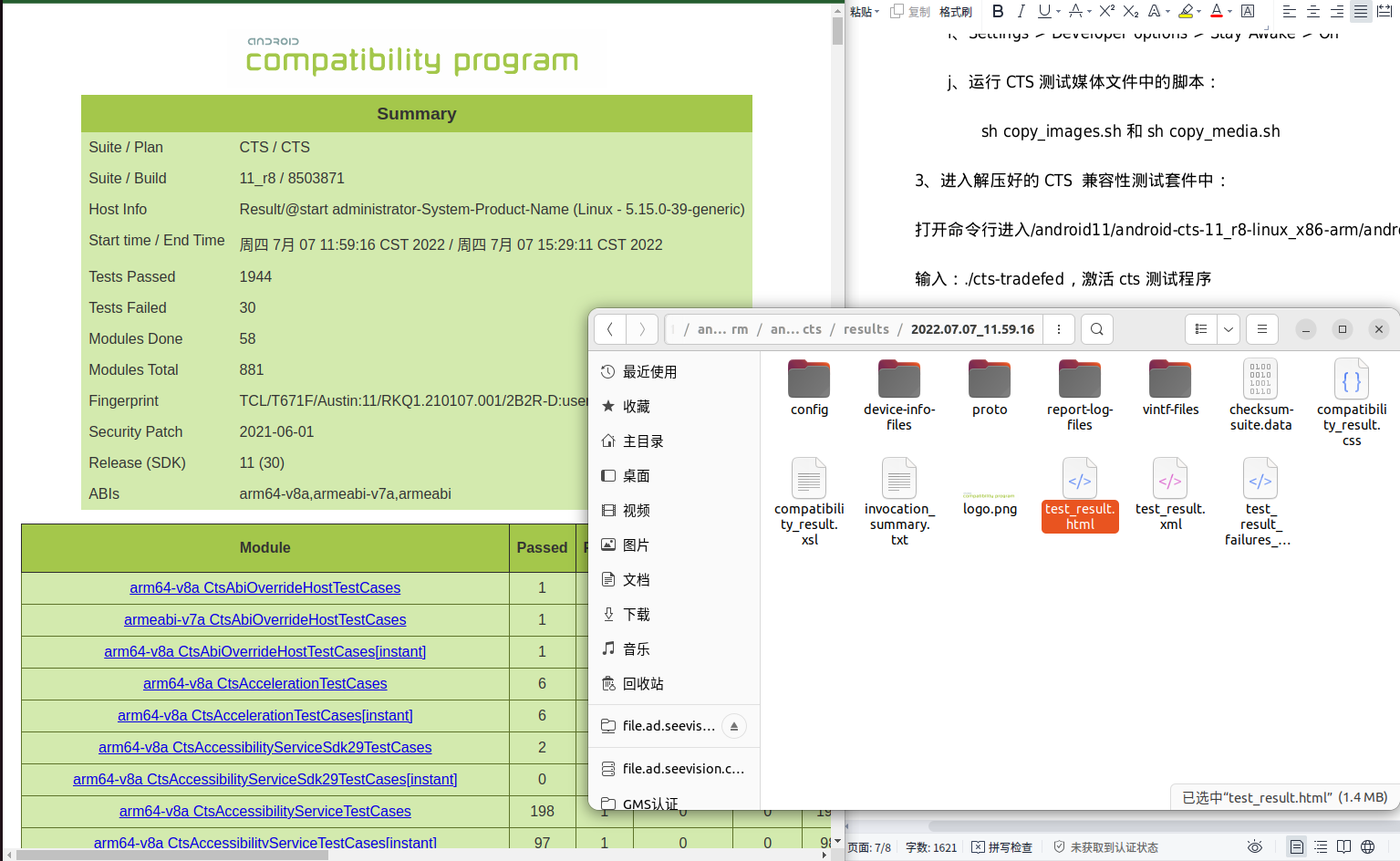
输入：./cts-tradefed，激活cts测试程序

再输入：run cts --plan CTS开始执行整个CTS的测试

测试结果：命令行输入l r查看输出的结果(没跑完或者不跑了，可以拔掉测试机器，等待一会看看是否出结果，重跑)



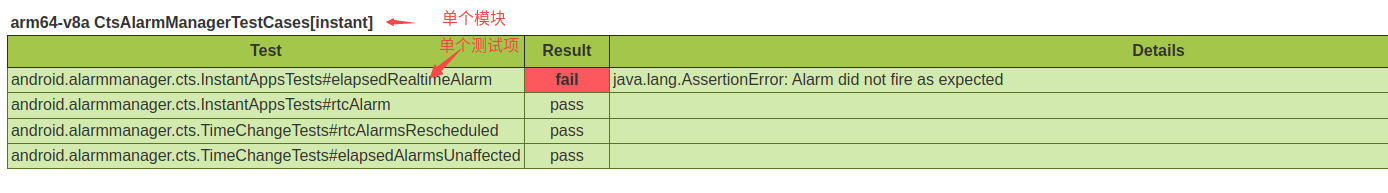
1. 测试输出结果查看：存放在android-cts/results文件夹下面，打开test\_result.html即可查看测试报告，cts log给到开发修复即可



1. 测试结果处理：

对于FAIL的测试项，进行但个模块或者单独测试项重跑，对于未跑完的，或结果输出异常的建议整个CTS测试重跑

1. 测试单个模块  
   run cts -m <模块名>  
   b、测试单个测试项  
   run cts -m <模块名> -t <test\_name>



**VTS测试指导:**

1. 环境配置：

拉取Android12源码：

配置repo：

<https://blog.csdn.net/qq_36554461/article/details/121328572>

配置git公钥SSH Keys：

<https://blog.csdn.net/qq_39196431/article/details/123797279>

<https://blog.csdn.net/yjj350418592/article/details/121640192>

拉取公司服务器Android12源代码：

更新Git Repo镜像：export REPO\_URL='http://mirrors.ad.seevision.cn/git/git-repo'

<http://mirrors.ad.seevision.cn/help/git-repo/>

repo init -u [ssh://xxx@gerrit.ad.seevision.cn:29418/aosp/platform/manifest](ssh://xxx@gerrit.ad.seevision.cn:29418/aosp/platform/manifest" \t "/home/administrator/文档\\x/_blank) -b android-12.0.0\_r32

repo sync -cd -j8 --no-tags

进行VTS测试：

ubuntu下配置好编译环境：

apt-get update -y && \

apt-get install -y build-essential python python3 git-core bc tzdata dosfstools mtools && \

apt-get install -y ccache doxygen gcc coreutils fakeroot squashfs-tools && \

apt-get install -y manpages-fr-extra liblz4-tool xz-utils cryptsetup-bin && \

apt-get install -y gnupg flex bison gperf zip unzip&& \

apt-get install -y curl zlib1g-dev gcc-multilib g++-multilib libc6-dev-i386 && \

apt-get install -y libncurses5 lib32ncurses5-dev x11proto-core-dev libx11-dev lib32z-dev && \

apt-get install -y libgl1-mesa-dev libxml2-utils xsltproc libssl-dev && \

apt-get install -y python3-pip python3-pexpect python3-markdown && \

apt-get install -y gawk wget diffstat texinfo chrpath socat cpio && \

apt-get install -y xz-utils debianutils iputils-ping libsdl1.2-dev xterm && \

apt-get install -y libmicrohttpd-dev libargtable2-dev clang-format ninja-build && \

apt-get install -y gzip ffmpeg && \

apt-get install -y cmake rsync xxd automake && \

apt-get install -y openjdk-8-jdk && \

sudo usermod -aG plugdev $LOGNAME

sudo apt-get install android-sdk-platform-tools-common

按照指导完成编译，再进行测试

(<https://blog.csdn.net/u010164190/article/details/122890815>)

sudo apt-get install python-dev

sudo apt-get install python-protobuf

sudo apt-get install protobuf-comiler

sudo apt-get install python-virtualenv

sudo apt-get install python-pip

source build/envsetup.sh

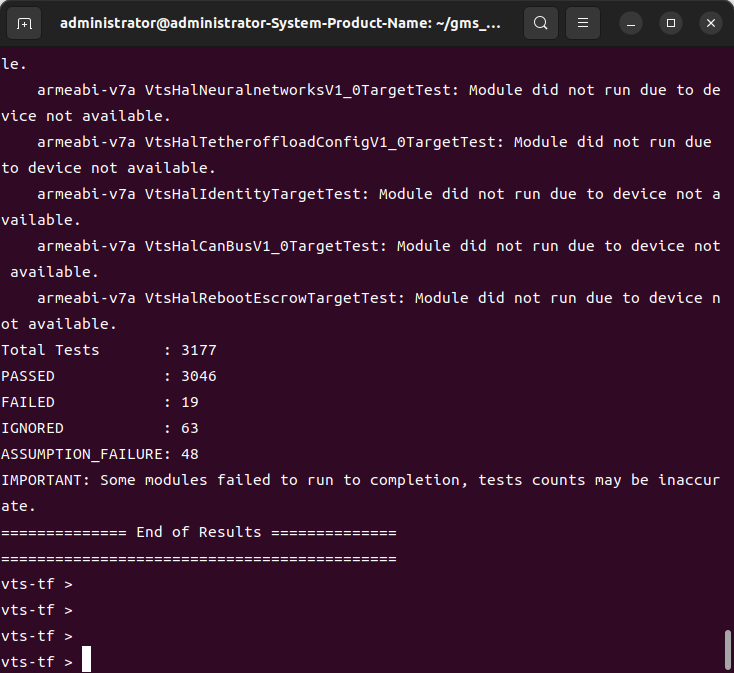
lunch aosp\_arm-eng(vs680)

make vts -j20

2、进行全局VTS测试：

./vts-tradefed

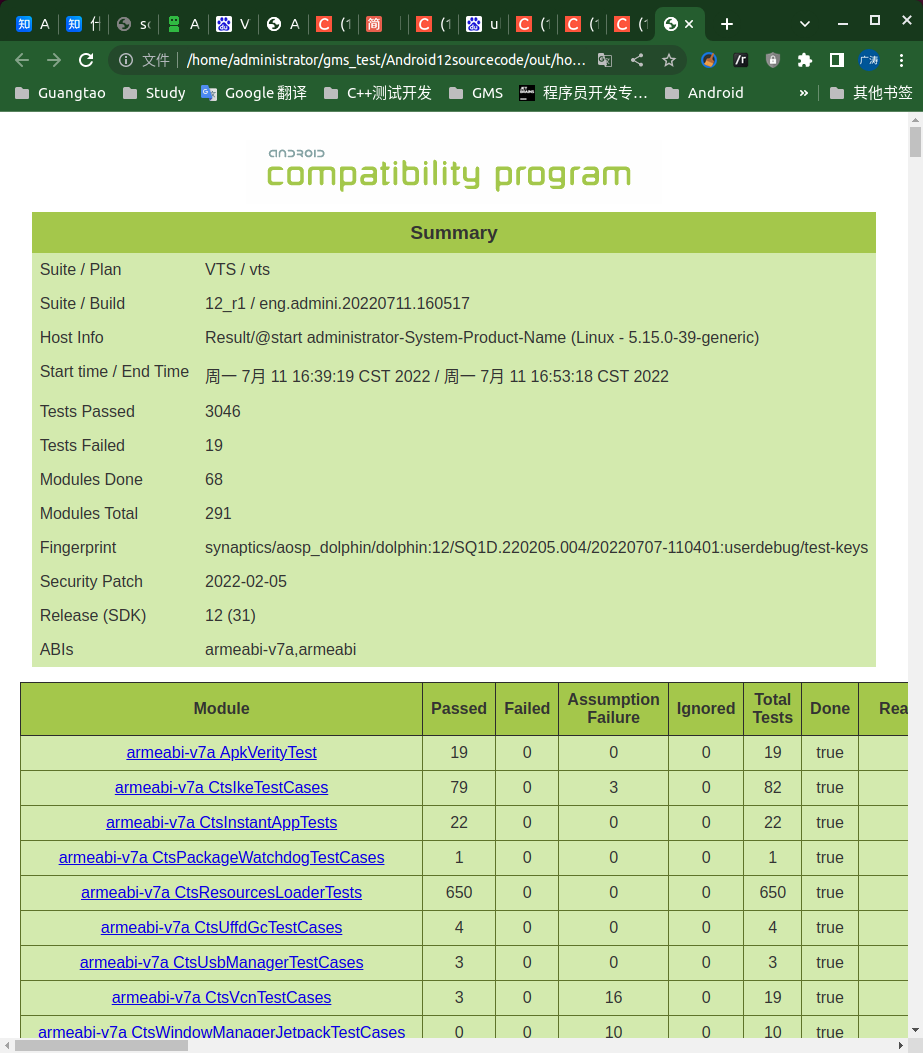
run vts



3、运行完成后，结果如下图：

进入目录，查看测试结果：

/home/administrator/gms\_test/Android12sourcecode/out/host/linux-x86/vts/android-vts/results/2022.07.11\_16.39.19



1. 如果遇到需要重新测试的模块可以进行单模快重跑：



4、可能遇到的VTS Fail项/VTS测试失败的可能原因：

1. 没有使用经过binderized的HAL
2. HAL中存在与接口规范不符的实现
3. Kernel接口问题
4. 没有添加相应的SEpolicy配置

GTS、STS、GVS、BTS