

应用场景

CCA开发过程中，每个开发人员的电脑上通常都有多个project，在编译UT，HST的时候，需要在不同的文件夹下面频繁切换

写了一个shell脚本，自动扫描硬盘下的工程，生成编译脚本buildagent，通过给该脚本传入简单的参数可以选择工程和编译类型

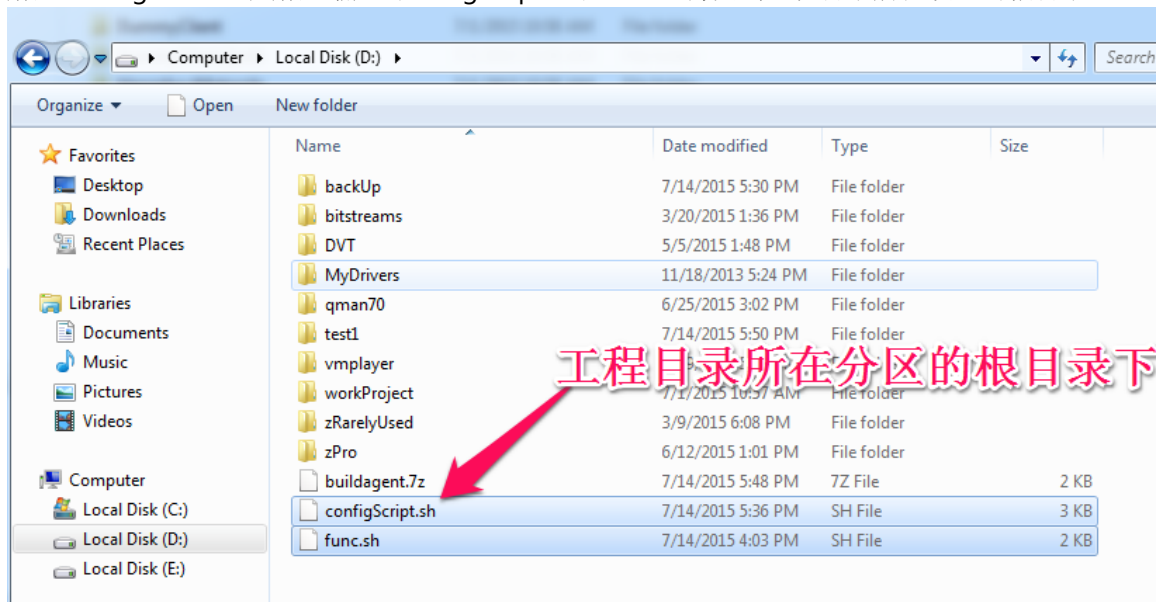
例如：

```
./buildagent --agent_sr14_1 --ut_cca
```

可以调用agent_sr14_1分支下的./test/UniversalTest/CCA/BuildAll编译CCA lib

配置脚本使用方法

1. 解压buildagent.7z，把解压缩后的configScript.sh和func.sh拷贝到工程目录所在硬盘的根目录

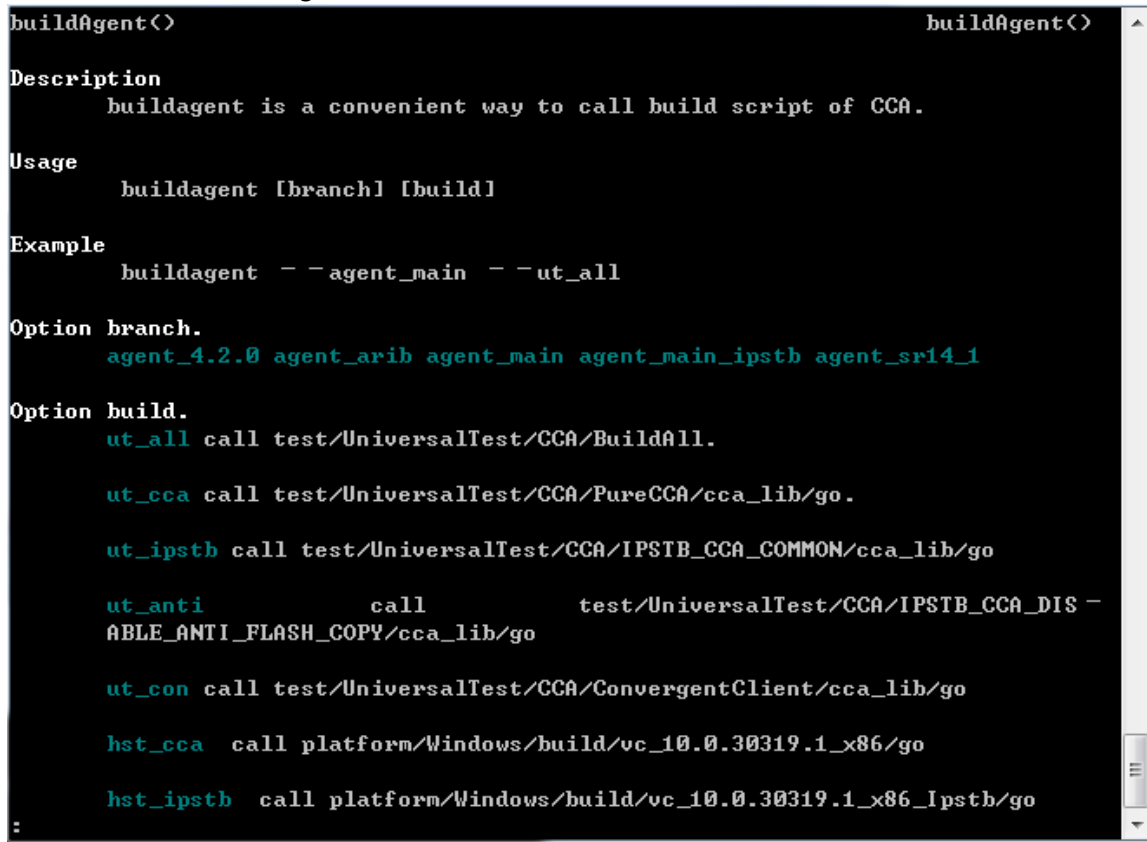


2. 在cygwin下运行configScript.sh（笔记本电脑运行该脚本比较慢，可以给配置脚本传入工程文件目录路径，例如 `./configScript.sh ./workProject`）



编译CCA Lib方法

- buildagent接受两个参数，第一个参数是branch选项，第二个参数是build脚本选项
- branch选项是根据当前工作目录下的工程文件名自动生成
- build脚本选项是固定的
- 可以调用命令man buildagent获取提示



```
buildAgent<>                                     buildAgent<>
Description
    buildagent is a convenient way to call build script of CCA.
Usage
    buildagent [branch] [build]
Example
    buildagent --agent_main --ut_all
Option branch.
    agent_4.2.0 agent_arib agent_main agent_main_ipsth agent_sr14_1
Option build.
    ut_all call test/UniversalTest/CCA/BuildAll.
    ut_cca call test/UniversalTest/CCA/PureCCA/cca_lib/go.
    ut_ipsth call test/UniversalTest/CCA/IPSTB_CCA_COMMON/cca_lib/go
    ut_anti call test/UniversalTest/CCA/IPSTB_CCA_DIS -
    ABLE_ANTI_FLASH_COPY/cca_lib/go
    ut_con call test/UniversalTest/CCA/ConvergentClient/cca_lib/go
    hst_cca call platform/Windows/build/vc_10.0.30319.1_x86/go
    hst_ipsth call platform/Windows/build/vc_10.0.30319.1_x86_ipsth/go
:
```

更新

如果从TFS上map到本地新的工程，需要重新运行configScript.sh完成配置