**How to package calabash-perfecto – creating a gem file**

* Cd to where the code is
* Delete any old gem file
* Create a new gem file using the gemspec

Commands :

*cd C:\Users\rinat\dev\calabash-perfecto-0.1.0*

*del calabash-perfecto-0.1.0.gem*

*gem build calabash-perfecto.gemspec*

**Pre-requisites :**

* See calabash installation document (Hadas/Moshe)

**How to install calabash-perfecto – installing the gem**

Install the gem by running:

*gem install calabash-perfecto-0.1.0.gem*

**Verifying the installation**

To verify the installation run:

*gem list*

The list you get should include : calabash-perfecto (0.1.0)

Run :

>calabash-perfecto -help

Usage: calabash-perfecto [options] apk-file/iap-file

-v, --version Get version

-i, --install Install app before testing

-l, --location LOCATION Location in repository for uploading apk/ipa file

-a, --app APPNAME App name is APPNAME

-c, --cloud CLOUD Connect to cloud CLOUD

-u, --user USER Connect as USER

-p, --password PASSWORD Connect with PASSWORD

-d, --device DEVICE\_ID Connect to device with id DEVICE\_ID

-r, --report REPORT\_DIR Download reports to directory REPORT\_DIR

-h, --help Display this screen

**Setting up the certificate for uploading files to Perfecto cloud (ruby requirement needed to upload the apk/ipa file for installation)**

* Copy cacert.pem and install\_cert.rb to c:\temp
* Run this once :

*set SSL\_CERT\_FILE=c:\temp\cacert.pem*

*cd c:\temp*

*ruby install\_cert.rb*

**Setting up the feature files from the installation skeleton files**

The installation created a directory:

<YOUR\_RUBY\_INSTALLATION>\lib\ruby\gems\<VERSION>\gems\calabash-perfecto-0.1.0

Use the “features-skeleton” under this directory as your starting point by copying the files in this directory into your project’s “features” directory.

Important :

1. Do not override the my\_first.feature file if you have tests there you would like to use
2. Existing calabash-android files: If you have made changes to some of the hook files in order to change the default flow of calabash, you should merge the files, keeping the “calabash-android” parts as you have them, and adding the “calabash-perfecto” parts. This should be a rare scenario.
3. Existing calabash-ios files : would be separate from calabash-perfecto and remain untouched

**Configuring perfecto-calabash parameters :**

There are 3 ways to configure the perfecto-calabash parameters:

* <YOUR\_FEATURE\_DIR>\support\env.rb :

*# Default app file (for installing the app on the device)*

*#$PerfectoAppFile = "<APK\_OR\_IPA\_FILE>"*

*# Application name (for starting the app)*

*#$PerfectoAppName = "<YOUR\_APP\_NAME>"*

*# Where to upload the apk/ipa file to (directory in repository)*

$PerfectoUploadLocation = "PUBLIC:calabash"

*# Connection credentials*

*#$PerfectoCloud = "<YOUR\_PERFECTO\_CLOUD\_HERE>"*

*#$PerfectoUser = "<YOUR\_PERFECTO\_CLOUD\_USER\_HERE>"*

*#$PerfectoPassword= "<YOUR\_PERFECTO\_CLOUD\_PASSWORD\_HERE>"*

*#$PerfectoDevice = "<YOUR\_PERFECTO\_DEVICE\_ID\_HERE>"*

*# Should the app be re-installed before every scenario ?*

$PerfectoReinstallEveryTest = ***false***

*# Directory where reports should be downloaded to*

*#$PerfectoReportDir = "<YOUR\_REPORT\_DIR>"*

* Environment variables :
  1. PERFECTO\_PARAM\_INSTALL
  2. PERFECTO\_PARAM\_APPNAME
  3. PERFECTO\_PARAM\_CLOUD
  4. PERFECTO\_PARAM\_USER
  5. PERFECTO\_PARAM\_PASSWORD
  6. PERFECTO\_PARAM\_DEVICE
  7. PERFECTO\_APP\_FILE
  8. PERFECTO\_REPORT\_DIR
  9. PERFECTO\_UPLOAD\_LOCATION
* Command line args :

*Usage: calabash-perfecto [options] apk-file/iap-file*

*-v, --version Get version*

*-i, --install Install app before testing*

*-l, --location LOCATION Location in repository for uploading apk/ipa file*

*-a, --app APPNAME App name is APPNAME*

*-c, --cloud CLOUD Connect to cloud CLOUD*

*-u, --user USER Connect as USER*

*-p, --password PASSWORD Connect with PASSWORD*

*-d, --device DEVICE\_ID Connect to device with id DEVICE\_ID*

*-r, --report REPORT\_DIR Download reports to directory REPORT\_DIR*

*-h, --help Display this screen*

Note : Order of precedence is : command line option is strongest, then environment variable then env.rb file. So if any parameter is defined in multiple ways, the stronger option will override any other.

**How to create and run tests**

* Cd to where your feature file is
* Run :

perfecto-calabash [options]

* Note:
  + apk/ipa files are only necessary for re-installing. If app is already installed and you would like to use existing installation – no need for this option
  + Running tests for android and ios is the same
* Example 1: Specify app file, re-install every scenario, and specify app name (credentials configured in env.rb file or env vars)

*calabash-perfecto MyApp.apk –i –a “My app Name”*

* Example 2: Specify nothing – use env.rb file or env vars

*calabash-perfecto*

* Example 3: Give credentials and device on command line

*calabash-perfecto –c cloud.perfecto.com –u Rinat –p mypassword –d 12345ABC*

* Reports:
  + In case of failure, a report including screen shot will be generated in the reports directory

**Debug option**

To turn on debug messages :

set PERFECTO\_DEBUG=on

To turn off debug messages :

set PERFECTO\_DEBUG=