C2C Run

**TABLE OF CONTENTS**

[PREREQUISITES](#h.jdbeyu3w0gu5)

[SOURCE CODE LOCATION](#h.qlyearuookp)

[CODE ORGANIZATION/ARCHITECTURE](#h.piuivkvboshl)

[HOW TO SETUP](#h.hnrfkimhoy9)

[HOW TO RUN](#h.2zumfn54net9)

[TEST DATA USED](#h.bv3dbt4s5yw6)

[LINK TO REPORT](#h.dlhc03chk1pw)

[HOW TO DEBUG](#h.81l4m9az0q4)

[NOTES:](#h.2f5y0ve2whu6)

# 

# PREREQUISITES

The Quickoffice c2c (Chrome to Chrome) rendering test framework is used to compare rendering of office documents on chrome between two different builds.

Its supported the Ubuntu and MAC OS, following other software need to be install to run C2C:

* Git
* node
* grunt

# **SOURCE CODE LOCATION**

Get Source code [here](https://quickoffice-internal.googlesource.com/html-office/+/scratch/c2cTestFramework).

# **CODE ORGANIZATION/ARCHITECTURE**

Please refer the design document for C2CTestFramework

<https://docs.google.com/a/synerzip.com/document/d/1wM6ZlzxWBpfC0Q4Vqe0eSyibpvIBqk3E20etBUXFvF4/edit>

# **HOW TO SETUP**

Install following software on machine:

* Install git
  + On MAC - brew install git
  + On Ubuntu - sudo apt-get install git
* Install node
  + On MAC - install node
  + On Ubuntu - install it form link - <http://howtonode.org/how-to-install-nodejs>
* install grunt -brew npm install grunt
* Clone the HTML repo -

git clone <https://quickoffice-internal.googlesource.com/html-office>

On root directory fire following command to switch on

**scratch/c2cTestFramework** branch >git checkout scratch/c2cTestFramework

* Create a local property File “c2cprop.txt” and put it at any location. Refer sample property [file](https://docs.google.com/a/synerzip.com/file/d/0B14YezqF0VKPLWRaRjdSYkZfUTA/edit) -

Following are the properties name present in the property file -

*----------------------------------------------------------------------------------------------------*

***baseBuild****:* <QO build location against which baseline data is captured>

**newBuild**: <Compare QO build location which is to be tested, use same path which is used in the **baseBuild** for File open performance>

**rootDir**: <output dir where generated data and report is to be stored>

**msImageDir**: <location of Microsoft images for reference, these images are not required for file open performance. However this property file is require for C2C framework, hence you can put any dummy path >

**baseImageDir**: <location where baseline data is available,these images are not required for file open performance. However this property file is require for C2C framework, hence you can put any dummy path>

**testFileDir**: <location of test data>

------------------------------------------------------------------------------------------------------

*baseBuild:/Users/synerzip/base-html-office/crx*

*newBuild:/Users/synerzip/html-office/crx*

*rootDir:/Users/synerzip/C2CRoot/C2C*

*msImageDir:/Users/synerzip/msImages*

*baseImageDir: /Users/synerzip/blessedImages*

*testFilesDir:/Users/synerzip/testFiles*

# 

# **HOW TO RUN**

Change directory to report directory in source code which contain python script to run c2c run

cd ../html-office/crx/e2eTests/c2cTests/report

Use following command to run the C2C -

> python generate\_report.py <c2cprop.txt file path> ,

for running the end to end case.

> python generate\_report.py <c2cprop.txt file path> BLESSED

for getting the blessed images

> python generate\_report.py <c2cprop.txt file path> COMPARE

for comparing the images, here make sure that blessed images should be

there.

> python generate\_report.py <c2cprop.txt file path> REPORT

for getting the REPORT, here make sure the blessed and compared

images should be there.

# **TEST DATA USED**

|  |  |
| --- | --- |
| **File Format** | **Number of file**  **used** |
| docx | 247 |
| doc | 153 |
| xlsx | 95 |
| xls | 75 |
| pptx | 751 |
| ppt | 449 |

# **LINK TO REPORT**

Results are find [here](https://docs.google.com/a/synerzip.com/spreadsheet/ccc?key=0As9GKuLYU7jGdFhUS1F3Rjk2cFdEWlBoVld6NElnSGc&usp=sharing)

# **HOW TO DEBUG:-**

# To debug C2C code follow below step:-

# Put ‘debugger’ word in your code from where you start from debug.

# Once node command fire immediately press ‘option+command+C’. You will get another window.

# Click on ‘Sources’ label.

# Once execution stop at debugger, execute code step by step.

# **NOTES:**