**.Net Tools Overview**

**SOURCE CODE LOCATION:** [PUNETETools/dotNetTools/Google\_TE\_O2O/](https://quickoffice-internal.googlesource.com/html-office/+/scratch/dotNet/PUNETETools/dotNetTools/Google_TE_O2O/)

**Environment Prerequisites:**

1. Platform: Windows 7 – 64 bit.
2. Microsoft office 2007
3. Dot net framework 4.0
4. ImageMagick-6.8.6 or above
5. 2007 Microsoft Office Add-in: Microsoft Save as PDF or XPS
6. Strawberry Perl
7. [OLE File Property Reader 2.1](http://www.microsoft.com/en-us/download/details.aspx?id=8422)
8. **Feature Classification Tool**

**Problem Statement:**

We have large number of office documents (Microsoft word,excel,power point) to test our product. When the development team add any specific feature or fix any bug specific to any feature to the product we need to test that feature with large set of documents with that feature. Finding documents with specific feature manually is time consuming task . It involves open the document manually check which features it has etc. This tool is developed to make it easy to find the documents with specific features.

**Steps for running Tool:**

**Input :** <FeatureClassificationTool.exe> <Input\_file\_folder>

**Output:** It inserts feature count in Database.

**Note:** We have hosted a local server from which anyone can download the feature specific files.

1. **MS Rendering Images Generator Tool**

**Problem Statement:**

For testing editors, one of the requirement was to extract images from Original and Round tripped files, compare the images and check for Visual Discrepancy after Roundtrip. For extracting images we have used .Net code and Office Interop libraries and Image magick.

**Steps For Running Tool:**

**Input :** <MsRenderingImagesGenerator.exe <Input\_file\_folder> <”XPS” or”Screen”>

**Output:** In output folder it generates the images (.png) in Screenshot folder

**Note:**

1) Input Argument <”XPS” or”Screen”>:

* XPS : Take screenshot by exporting to XPS first and generated images from that XPS file.
* Screen :Take screenshot by visible content on screen.

1. **Original Vs RT Visual Comparator Tool [O2O]:**

**Problem Statement:**

When testing editors, one of the important features is that after Roundtrip the document should not be corrupt and the visual fidelity of the original document should be preserved. Let's call this process as O2O (MS office to MS office comparison after Roundtrip). Initially the scope of Round Trip is just to open and save the document in our product without doing any changes to the document. Doing this manually is very time consuming task and it is not possible to process large set of data manually We needed automation framework\tools in place to let us run O2O on a set of documents daily and also run on a large set of documents a couple of times every week

**Steps For Running Tool:**

**Input :** <OriginalVsRtVisualComparator.exe> <Original\_MS\_Files> <Roundtrip\_Files> <Output\_folder> <”XPS” or ”Screen”>

**Output:** In output folder it generates the montage of mismatch pages also generates the html report, wherein we can have detailed information about mismatch percentage, test files and the montage images.

**Note:**

1) Please use absolute path of the exe when running the exe

2) Input Argument <”XPS” or”Screen”>:

* XPS : Take screenshot by exporting to XPS first and generated images from that XPS file.
* Screen :Take screenshot by visible content on screen.

1. **Office File Validator Tool**

**Problem Statement:**

We Needed a tool to identify the valid, password protected and corrupt MS office files from large set of real world files.

**Steps for running Tool:**

**Input :** <OfficeFilesValidityChecker.exe> <Input\_file\_folder>

**Output:** In output folder it segregates the files as per the category (i.e. corrupt files, password protected, dubious, mismatch extension and valid files).

1. **Office Files Segregator Tool**

**Problem Statement:**

We needed a tool to segregate test files according to their Office version in which they are created. To solve this problem we developed this tool using C# .Net and Binary stream reader.

**Steps for running Tool:**

**Input :** <FileSegregatorTool.exe> <Input\_file\_folder>

**Output:** This tool will segregate the files according to their office version and output directory will contain following sub-folders

* Valid Files
* Protected Files
* Extension Mismatch Files
* Files\_2011
* Files\_2007
* Files\_2003
* Files\_2008
* Files\_2013
* Other\_Office

1. **Spreadsheet Statistics Generator Tool**

**Problem Statement:**

We needed tool to analysis Spreadsheet file for performance and other sort of testing. Spreadsheet profiler tool analyses spreadsheets and gives count of total worksheets, used rows, columns, cells, and drawings.

**Steps for running Tool:**

**Input :** <spreadSheetProfiler.exe.exe> <Input\_file\_folder>

**Output:** The output folder at the same location of input folder, with following subfolders and files in it:

* + - * **Corrupt Files:** Files shows error/exception while opening or processing Spreadsheet.
      * **Valid Files:** This folder contains all the files that are processed successfully.
      * **qosp\_result.cvs:** This is the report file in CSV format which has data for spreadsheets per worksheet. The data captured per worksheet includes row count, column count, cell count, shape count and function count.

1. **Corruption Analysis Tool**

**Problem Statement:**

Developers needed a tool to analysis corrupt files for mostly occurring errors in xmls which are causing corruption. To solve this problem, we developed this tool using Open Xml Sdk.

**Steps for running Tool:**

**Input :** <CorruptionAnalysisTool.exe> <Corrupt\_file\_folder> <csv Path>

**Output:** This tool gives us a .csv file which will have all the missing tags/exceptions in xmls for each file

1. **Formula Analysis Tool**

**Problem Statement:**

We needed tool to analysis Spreadsheet files for most used functions from real word data. Tool used for product analysis and decide the priority for not supported function for implementation.

**Steps for running Tool:**

**Input :** <function\_ analysis.exe> <Input\_file\_folder>

**Output:** In output folder it segregates the files as per the category (i.e. valid, corrupt, Nonfunction files and Funtction\_analysis.csv).

1. **OfficeDocumentCreationDetails Tool**

**Problem Statement:**

We Needed a tool to identify the MS office binary files that are created in office version older than 2003 from large set of real world valid files.

**Steps for running Tool:**

**Input :** <OfficeDocumentCreationDetails.exe> <Input\_file\_folder>

**Output:** In output folder it generates two folders named “Office2003”, ”Officeolder” and a csv containing creation details i.e. File format, Application Name, Creation year and last saved year.

**Notes:**

* There are some files for which we aren’t able to get creation details.
* Valid only for binary file format as OOXML files aren’t ole documents.
* Office2003 folder will contain files created in Microsoft office 2003 and above.
* Officeolder folder will contain files created in older version than Office 2003.
* Files having file format less than 11 are created in older office version.
* For using this tool user need to install [OLE File Property Reader 2.1](http://www.microsoft.com/en-us/download/details.aspx?id=8422)