|  |  |
| --- | --- |
|  |  |
| Audit Configuration Differentiation Utility | |
|  | Version: 0.1  Document Status: UNCLASSIFIED |

Document Control

|  |  |
| --- | --- |
| Author: | Steven P.Wang |
| File Name: | Audit Configuration Differentiation Utility |
| Path: | $/EST/eCSD/tools/AuditConfigDiffUtil/DiffProj |
| Created Date: | 06/03/2018 |
| Last Edited: | 19/04/2018 13:56 |
| Number of Pages: |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Endorsement | | | |
| Gihan Kannangara | Technical Team Lead | Tan Phan | Technical Team Lead |
|  | |  | |
| Approval | | | |
| Gihan Kannangara | Technical Team Lead | Tan Phan | Technical Team Lead |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Summary of Changes |
| 0.2 | 06/03/2018 | Steven P.Wang | Second draft |

# Introduction

The purpose of this document is to provide an overview of the Audit Configuration Differentiation Utility, its overall structure and instructions to use it.

This tool was built to differentiate two sets of Sterling configuration files. It outlines the difference between every pair of files set up to be compared then produce a report of all comparison in html. It should help the user finding the differences with ease.

## Audience

This document was written for:

| Team/Person | Interest |
| --- | --- |
| eCommerce Platform (IBM) | Responsible for implementation, testing and maintenance of the utility |
| SBR2 Support team | Running this tool to differentiate between two environments. |

# Functional Design

## 2.1 Context

This project was initiated by Tan Phan who is the team lead of the SBR2 small change team. The goal is to compare the configuration between two SBR2 environments. It will also help operators to monitor components across different environments. Hence additional information becomes available to explain the issues experienced by the SBR2 ecommerce platform. Many of the said issues were raised in PMRs. Hopefully the information generated by this utility may also accelerate the time to resolution.

From a high level, the tool finds the delta between two environments as illustrated below.

Prod1

MEIG

STERLING

VATS

Prod2

AUDIT CONFIG

AUDIT CONFIG

This tool is the triangle which compares the configuration files extracted from two environments or the delta by the Audit Configuration tool built by Matt Lyon.

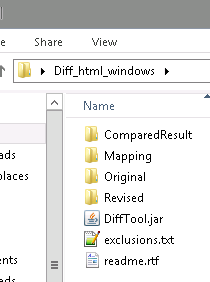
The tool was built by Steven P.Wang (ucy4b). Special thanks to Tan and Gihan who provided feedback and guidance and Matt whose interface of another project was modified to be used for this one. Last but not least, this tool was built around a java library from the following link. It saved time to not reinvent the wheel of the diff algorithm.

<https://code.google.com/archive/p/java-diff-utils/>

The said java library does the comparison on its own. However, it is not able to distinguish within which tag of the XML configuration file was the difference found. This tool meets such requirement by counting the line number of the xml tags. Then generate compared result in json format, which is ultimately read in by an html populated by an Angular js file.

## Functional Design

The basic file structure looks like the following. For execution, please find section 2.2.4.



“Compared Result” folder has the output of the tool. This includes the json outputs, html and the js file.

“Mapping” folder has the mapping files which indicate which two files are to be compared.

“Original” and “Revised” folder contain the raw configuration files from two environments respectively.

DiffTool.jar is the jar file producing the Compared Result.

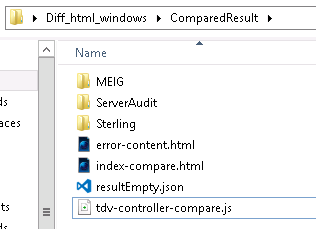
“exclusions.txt” contains the list of phrases are to be omitted from the comparison. This a blacklist.

“readme.rtf” is a text document recording development notes from the past.

[10/04/2018 new feature update]

“inclusions.txt” contains the list of phrases that must be presented for the comparison. This is a whitelist.

### 2.2.1 Compared Result

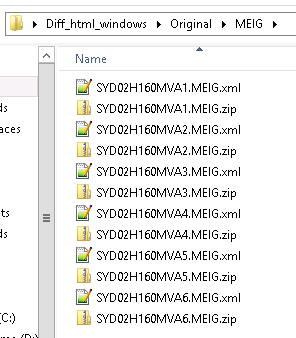


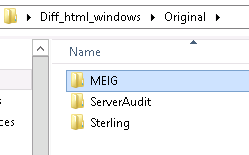
The folders store the compared output and the report. The rest of the files are copied into each folder after every comparison. They will also be modified during the process so each report would have distinctive content such as title, number of files compared and the actual result.

### 2.2.2 Mapping

This folder should have txt files with the same name as the folders containing the configuration files, if a mapping is to be defined. The content of the file uses a comma as a delimiter separating the name of the file from the Original and the Revised folder.

### 2.2.3 Original and Revised





Both Original and Revised folder should have the same content structure. Each folder should have files to be compared. Otherwise, there will be a pop up warning saying the folder is empty. At this moment, the report could process xml and zip (compare size) file under the MEIG folder, txt file under the ServerAudit folder and xml file under the Sterling folder.

When a mapping file on a folder is not presented, files are compared as the following.

When there are same number of files in both Original and Revised folder, the files are compared one to one in the order sorted by the system.

If there is less file in the Original folder, it will compare again every corresponding file with the Revised folder.

original has 1 and 2  
revised has a b c d e  
The new default behaviour of comparison will be 1a 2b 1c 2d 1e.

If there is less file in the Revised folder:  
original has 1 2 3 4  
revised has a b  
The new default behaviour of comparison will be 1a 2b.  
  
There will also be a popup warning when there are such miss matches, but it does not stop the report from generated.

### 2.2.4 DiffTool.jar

Running this file produces the output. It could be executed without an issue in a IBM laptop running either Windows or MacOS. However, there are issue running it in ATO even in the Dev box.

The jar could be executed in a command window or in a console. Double clicking works in a none ATO environment only.

java -jar DiffProj.jar

It could also take in two command line arguments pointing to other Original and Revised directories.

java -jar DiffProj.jar /Users/stevwang/Desktop/DiffProj\_html/Original /Users/stevwang/Desktop/DiffProj\_html/Revised

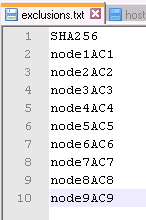
Or a third to where the mapping file is.

java -jar DiffProj.jar Original/ Revised/ Mapping/

This file is not capable to be copied alone to other directories for execution. It has dependencies on the folders in the same directory and the exclusion.txt file

Three reports may be generated at once at the moment, namely MEIG, Sterling and ServerAudit.

### 2.2.5 Exclusion.txt

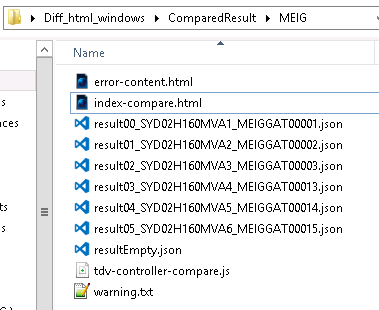


This file contains words to be excluded from the report output. One word per line.

### 2.2.5 readme.rtf

This file is the development log.

### Report Output



The output will be triggered to run automatically in a browser window after the execution of the DiffTool.jar file, assuming the execution is successful. It does not run in ATO net because of the restriction enforced on the browser. Please be sure that a browser is readily opened before running the jar file in the ATO Devnet environment. Otherwise two tabs of the same report would be opened. None ATO environments do not have these issues.

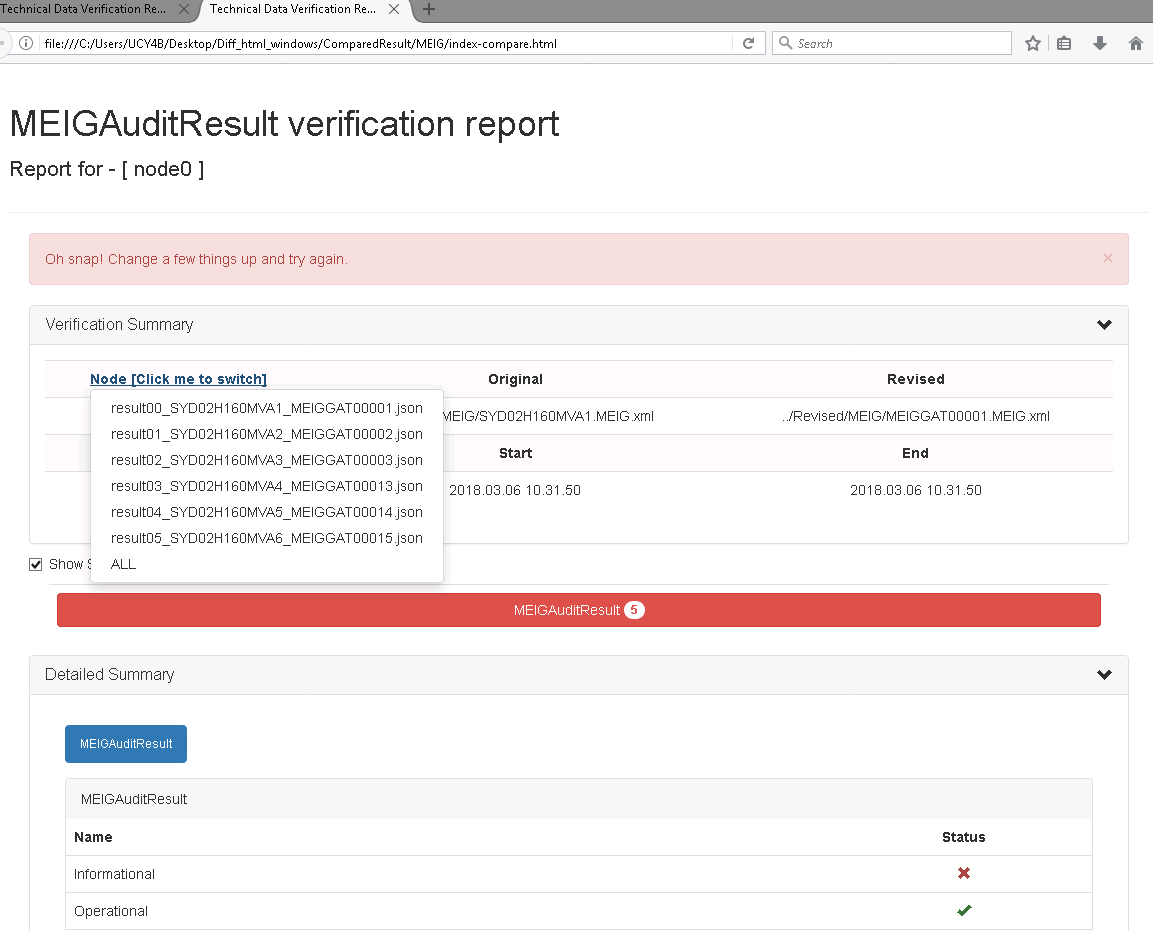
One may also run the report by executing the index-compare.html file in a browser.

Error-content.html controls the error pop up in the report.

The json files contain the report output.

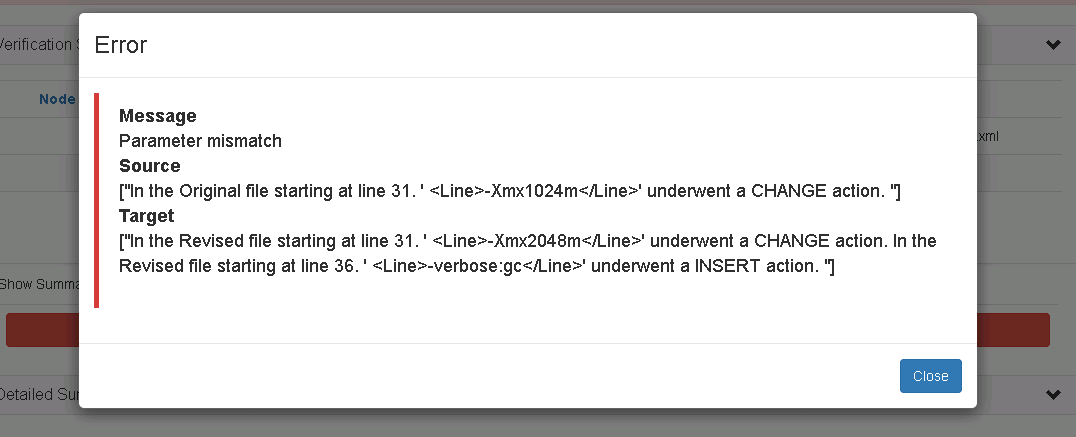
tdv-controller-compare.js reads the json files and populates the html file. Notice the resultEmpty.json file. It is required for the All option in the dropdown as elaborated below. The reason is because tdv-controller-compare.js uses AngularJS framework which read in file asynchronously. Reading multiple files at once really is not what AngularJS was designed to do. There was a time when the All option does not refresh the report completely, so this file was introduced to be processed upon selecting All.

The warning.txt file stores warning messages if the folder content of Original and Revised is different.

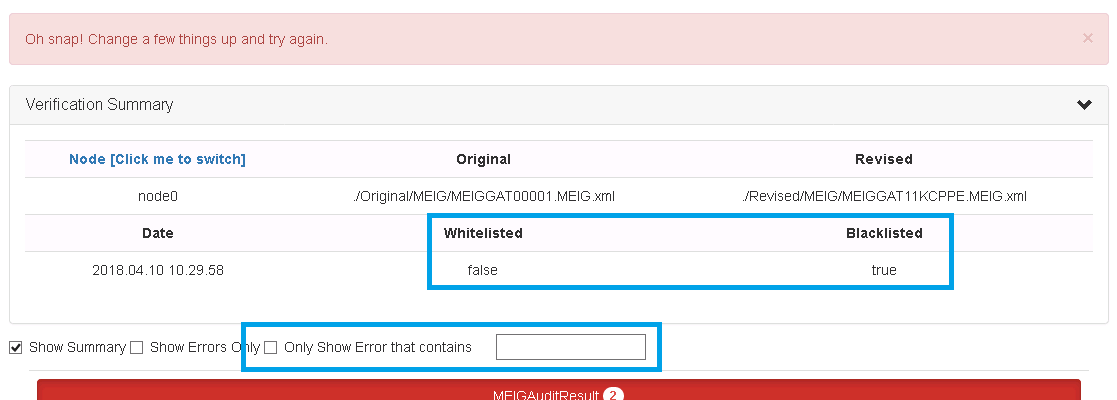


Users of this report could switch to different comparisons by click on the Node [Click me to switch] dropdown. The All option in the dropdown displays all the entries of every option in one single report.

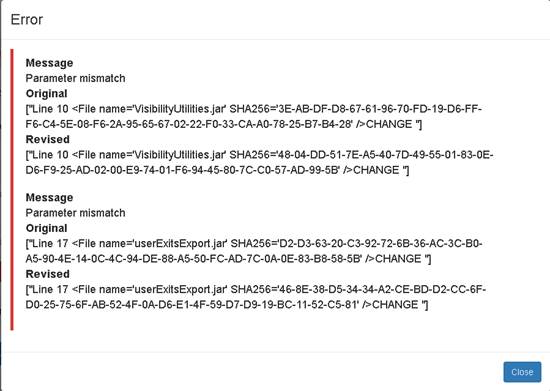
A green tick indicates this section is identical between two files. A red cross means there are differences. Clicking on the entry with a red cross invokes a pop up showing the differences.



[10/04/2018 new feature update]



1. Two fields are there to indicate whether words are whitelisted (as defined in inclusion.txt) or blacklisted from the result (as defined in exclusion.txt). Whitelisting is also a new feature. Only the words contained in the inclusion.txt will be compared.
2. This allows filtering when the report is displayed.
3. Error messages are no longer appended together. Each message has its own space. This improves readability greatly.



cid:image007.png@01D3D028.8C8D5C20

Now back to 2. Check the checkbox and enter a value. All errors will be filtered.



# Execution

Please find section 2.2.4

# Existing concerns and possible improvements

Functionally speaking, the tool has plenty functionalities to process the data in different ways in both Windows and MacOS. However, there are still room for further improvements.

Update on 28/02/2018 updated the algorithm reading in Sterling xml. The logic has been improved that it could handle scenarios where some tags are not presented in one or both files. This also means that it could find the number of closing tags, so it does not have to be counted and hardcoded manually. However, the developer must know a definite set of all possible tags in the xml file to make this work. The logic could handle tags it knows but not presented, but it is not smart enough to recognize tags that it does not know.

As articulated in 2.2.4, the jar file has some dependencies. Failing to comply would trigger java run time errors. It could be improved that having exceptions handling such scenarios.

Given ATO environment has some quirks, it is recommended to open up the report with firefox rather than the default IE. In Devnet, open a web browser and log in before running the tool.