METRIC CONSULTING ENGINEERS & ARCHITECTS CO.,LTD.

รายงานสรุปการทำงาน: Automation Script สำหรับอัปเดต Clash Report (v2.0)

ภาพรวมของสคริปต์ (Overview)
เอกสารฉบับนี้อธิบายการทำงานของ Python Script ที่พัฒนาขึ้นเพื่ออำนวยความสะดวกในการจัดทำรายงาน Clash Detection โดยระบบจะทำการดึงข้อมูล Viewpoint และรูปภาพจากไฟล์ HTML ที่ Export จากโปรแกรม Navisworks นำมากลั่นกรองและเขียนลงในเทมเพลต Excel แบบอัตโนมัติ โดยยังคงรักษารูปแบบ (Format) เดิมของเอกสารไว้ทุกประการ และสับเปลี่ยนมาใช้เครื่องยนต์ **xlwings** เพื่อไม่ให้รูปเดิมสูญหาย

1. อธิบายการทำงานในแต่ละส่วนของโค้ด

สคริปต์นี้ถูกแบ่งการทำงานออกเป็น 4 กระบวนการหลัก ดังนี้:

1.1 ส่วนติดต่อผู้ใช้ (GUI - Graphical User Interface)

- ใช้ไลบรารี `customtkinter` ในการสร้างหน้าต่างโปรแกรมที่ทันสมัย กระชับ และเป็นระเบียบ
- **การปรับฟอร์มหน้าต่าง:** ปรับ Layout ให้เรียงลำดับขั้นตอน 1, 2, 3 ลงมาด้านล่างอย่างชัดเจนตามฟอร์มแรกสุด มีการซ่อน Path เต็มให้แสดงผลเฉพาะชื่อไฟล์สั้นๆ เพื่อให้หน้าต่างดูเรียบร้อย ไม่ยืดตัวองค์ประกอบ

1.2 การดึงและทำความสะอาดข้อมูล (Data Parsing)

- ใช้ `glob` สแกนหาไฟล์ HTML ทั้งหมด และใช้ `BeautifulSoup` ในการค้นหาและแยกแท็ก HTML
- ระบบจะค้นหา `div class="viewpoint"` เพื่อดึงข้อมูลชื่อ View (Title) และตำแหน่งของรูปภาพ (Image Path)
- **การตัดข้อมูลพิกัด:** มีการนำคำสั่งดึง Camera Position ออกทั้งหมดตามความต้องการ เพื่อความรวดเร็วและสะอาดตาของตารางรายงาน

1.3 การจัดการ Excel ด้วยเครื่องยนต์ xlwings (Data Validation)

- **ระบบป้องกันข้อมูลซ้ำ:** โปรแกรมจะตรวจสอบรายชื่อพาดหัวจากคอลัมน์ C ก่อน หากมีอยู่แล้วจะข้ามทันทีเพื่อกรอกเฉพาะรายการใหม่
- **การแยกหมวดหมู่ด้วยเงื่อนไขใหม่ (Else = ETC.):** มีระบบตรวจสอบตัวอักษรนำหน้าชื่อ Viewpoint หากขึ้นต้นด้วย **"C"** จะระบุเป็น CLASH CHECK, ขึ้นต้นด้วย **"V"** เป็น VISUAL CHECK และกรณีอื่นๆ นอกเหนือจากนี้ทั้งหมด (Else) จะจัดกลุ่มลงในหมวด **ETC.**

1.4 การฝังรูปภาพลงในเซลล์แบบ 100% (Smart Formatting)

- **รักษารูปภาพเดิมไม่ให้สูญหาย:** เปลี่ยนมาใช้ `xlwings` ซึ่งสั่งเปิดแอปพลิเคชัน Excel จริงในโหมดเบื้องหลัง ทำให้วัตถุรูปภาพหรือกราฟแท่งที่มีอยู่เดิมในตารางไม่ถูกทำลายหรือสูญหายตอนบันทึกไฟล์
- **ระบบฝังและคำนวณสัดส่วนรูปภาพ (Embed Images):** โปรแกรมจะแปลงพิกัดรูปภาพเป็น Absolute Path เพื่อความเสถียร จากนั้นใช้ `PIL (Pillow)` หาขนาดภาพจริงมาย่อขยายให้อยู่ในกรอบสูง 250px พร้อมคำนวณและปรับขยายความสูงของแถว (Row Height) และความกว้างคอลัมน์ G (Column Width) ให้พอดีกับสัดส่วนภาพจริงโดยอัตโนมัติ ไม่ล้นทับเซลล์อื่น
- **คัดลอกสไตล์ตารางเดิม:** ใช้คำสั่งคัดลอกรูปแบบตาราง (Border, Font, Fill) จากบรรทัดด้านบนลงมาแปะทับแถวใหม่ ส่งผลให้ตารางรายงานสวยงามเป็นเนื้อเดียวกัน

2. คำสั่งในการ Compile เป็น Executable File (.exe)

เนื่องจากระบบมีการย้ายเครื่องยนต์ไปจับคู่กับ xlwings ในการจัดทำระบบฝังรูป จึงต้องปรับใช้คำสั่งการแพ็คไฟล์ดังนี้:

```
python -m PyInstaller --noconsole --onefile --collect-all customtkinter --collect-all xlwings
ClashUpdater.py
```

3. Python Source Code (ฉบับสมบูรณ์)

```
import customtkinter as ctk
from tkinter import filedialog, messagebox
import openpyxl
import xlwings as xw
from PIL import Image as PILImage
from bs4 import BeautifulSoup
import os
import glob
import shutil

ctk.set_appearance_mode("System")
ctk.set_default_color_theme("blue")

def browse_excel():
    filepath = filedialog.askopenfilename(
        title="เลือกไฟล์ Excel Clash Report",
        filetypes=(("Excel files", "*.xlsx"), ("All files", "*.*"))
    )
    if filepath:
        lbl_excel_path.configure(text=os.path.basename(filepath), text_color="#2b6cb0")
        excel_path_var.set(filepath)

        try:
```

```
            wb = openpyxl.load_workbook(filepath, read_only=True)
            sheets = wb.sheetnames
            if sheets:
                sheet_combo.configure(values=sheets)
                sheet_combo.set(sheets[0])
        except Exception as e:
            messagebox.showerror("Error", f"ไม่สามารถอ่านไฟล์ Excel ได้:\n{e}")

def browse_html_folder():
    folderpath = filedialog.askdirectory(title="เลือกโฟลเดอร์ที่เก็บไฟล์ HTML")
    if folderpath:
        lbl_html_folder.configure(text=os.path.basename(folderpath), text_color="#2b6cb0")
        html_folder_var.set(folderpath)

def run_process():
    excel_path = excel_path_var.get()
    html_folder = html_folder_var.get()
    target_sheet = sheet_combo.get()

    if not excel_path or not os.path.exists(excel_path):
        messagebox.showwarning("คำเตือน", "กรุณาเลือกไฟล์ Excel ที่ถูกต้อง")
        return
    if not target_sheet or target_sheet == "ยังไม่ได้เลือกไฟล์ Excel":
        messagebox.showwarning("คำเตือน", "กรุณาเลือก Sheet เป้าหมาย")
        return
    if not html_folder or not os.path.exists(html_folder):
        messagebox.showwarning("คำเตือน", "กรุณาเลือกโฟลเดอร์ HTML")
        return

    try:
        html_files = glob.glob(os.path.join(html_folder, "*.html"))
        if not html_files:
            messagebox.showinfo("ข้อมูล", "ไม่พบไฟล์ .html ในโฟลเดอร์ที่เลือก")
            return

        new_clashes = []
        for html_path in html_files:
            html_dir = os.path.dirname(html_path)

            with open(html_path, 'r', encoding='utf-8') as f:
                soup = BeautifulSoup(f, 'html.parser')

            viewpoints = soup.find_all('div', class_='viewpoint')

            for vp in viewpoints:
                title_tag = vp.find('h2')
                title = title_tag.text.strip() if title_tag else ''

                img_tag = vp.find('img')
                image_src = img_tag['src'] if img_tag else ''
```

```
        full_img_path = ""
        if image_src:
            full_img_path = os.path.join(html_dir, image_src)

        if title:
            new_clashes.append({
                'title': title,
                'image_path': full_img_path
            })

base, ext = os.path.splitext(excel_path)
save_path = f"{base}_Updated{ext}"
shutil.copy(excel_path, save_path)

app = xw.App(visible=False)

try:
    wb = app.books.open(save_path)
    ws = wb.sheets[target_sheet]

    max_row = ws.range('C' + str(ws.cells.last_cell.row)).end('up').row

    existing_titles = set()
    for row in range(1, max_row + 1):
        val = ws.range(f'C{row}').value
        if val:
            existing_titles.add(str(val).strip())

    item_no = 1
    for row in range(max_row, 0, -1):
        val = ws.range(f'A{row}').value
        if isinstance(val, (int, float)):
            item_no = int(val) + 1
            break

    current_row = max_row + 1
    added_count = 0

    for clash in new_clashes:
        if clash['title'] not in existing_titles:

            if clash['title'].startswith('C'):
                folder_name = "CLASH CHECK"
            elif clash['title'].startswith('V'):
                folder_name = "VISUAL CHECK"
            else:
                folder_name = "ETC."

            if current_row > 1:
```

```
                        ws.range(f'{current_row-1}:{current_row-1}').copy()
                        ws.range(f'{current_row}:{current_row}').paste(paste='formats')

                    ws.range(f'A{current_row}').value = item_no
                    ws.range(f'B{current_row}').value = folder_name
                    ws.range(f'C{current_row}').value = clash['title']
                    ws.range(f'H{current_row}').value = "NEW"

                    if clash['image_path'] and os.path.exists(clash['image_path']):
                        try:
                            abs_image_path =
os.path.normpath(os.path.abspath(clash['image_path']))

                            with PILImage.open(abs_image_path) as pil_img:
                                orig_width, orig_height = pil_img.size

                            target_height_px = 250
                            ratio = orig_width / orig_height

                            target_height_pt = target_height_px * 0.75
                            target_width_pt = target_height_pt * ratio

                            ws.range(f'{current_row}:{current_row}').row_height =
target_height_pt + 10

                            needed_width = ((target_height_px * ratio) - 5) / 7
                            current_width = ws.range('G:G').column_width
                            if needed_width > current_width:
                                ws.range('G:G').column_width = needed_width + 2

                            cell_left = ws.range(f'G{current_row}').left + 5
                            cell_top = ws.range(f'G{current_row}').top + 5

                            ws.pictures.add(
                                abs_image_path,
                                left=cell_left,
                                top=cell_top,
                                width=target_width_pt,
                                height=target_height_pt
                            )

                        except Exception as img_err:
                            filename = os.path.basename(clash['image_path'])
                            ws.range(f'G{current_row}').value = f"{filename} (Err:
{str(img_err)})"
                            ws.range(f'{current_row}:{current_row}').row_height = 15
                    else:
                        ws.range(f'{current_row}:{current_row}').row_height = 15

                    existing_titles.add(clash['title'])
```

```
                    current_row += 1
                    item_no += 1
                    added_count += 1

            wb.save()
            wb.close()

        finally:
            app.quit()

        messagebox.showinfo("สำเร็จ", f"อัปเดตเรียบร้อย!\n\nเพิ่มตรวจสอบและบันทึก Clash ใหม่ลงเซลล์จำนวน
{added_count} รายการ\nไฟล์ถูกบันทึกที่:\n{save_path}")

    except Exception as e:
        messagebox.showerror("Error", f"เกิดข้อผิดพลาดระหว่างทำงาน:\n{e}")

# ... (GUI Layout Configured for Compact Form) ...
```