

Power Automate Desktop (PAD) 教學手冊

本教學聚焦「Power Automate for Desktop」(PAD) 桌面自動化，不再涵蓋雲端 Cloud Flow。內容以大量實務案例：Excel、Outlook、網頁資料擷取、迴圈 / 條件模式、OpenAI Web API 串接等，協助快速打造穩定 RPA。

目錄

1. 什麼是 Power Automate Desktop (定位與特性)
2. 安裝與環境準備 (帳號 / 安裝 / 首次設定)
3. 介面導覽與核心概念 (動作庫、變數、UI Elements、子流程、錯誤處理)
4. 第一個入門流程：記事本自動輸入
5. Excel 自動化實務 (讀 / 寫 / 篩選 / 批次處理)
6. Outlook 自動化實務 (讀信 / 篩選 / 下載附件 / 自動回覆)
7. 網頁資料擷取 (Web 錄製、動態元素、表格爬取)
8. 迴圈與條件判斷設計模式 (For Each / Loop / If / Switch / 重試)
9. 串接 OpenAI / Azure OpenAI Web API (文字摘要 / 關鍵字擷取)
10. 錯誤處理與穩定性 (On Error / 重試 / 超時 / 日誌)
11. 效能與最佳化 (元素快取、批次寫入、資源回收)
12. 安全與憑證保護 (機密、API Key、帳號)
13. 常見問題排除 (FAQ / Troubleshooting)
14. 進階技巧與建議學習路徑 附錄 A：常用動作與變數速查 附錄 B：範例流程骨架模板

1. 什麼是 Power Automate Desktop

Power Automate Desktop 為 Microsoft 提供的 RPA (Robotic Process Automation) 工具，透過「錄製 + 低程式化動作」模擬人類操作桌面應用與網頁：

- 模擬鍵盤滑鼠、輸入、複製貼上
- 操作 Windows / 應用程式 / 瀏覽器 / 檔案系統 / Excel / Outlook
- 資料擷取：網頁表格、文字、視窗控制項
- 整合 HTTP 呼叫，串接外部 API (例如 OpenAI)
- 支援有 UI 的老舊系統，無需 API 仍可自動化

核心價值：減少重複耗時的人工例行工作 → 標準化 → 降低錯誤 → 可追蹤。

2. 安裝與環境準備

1. 需求：Windows 10 以上 (含 Server), 具備工作 / 學校 Microsoft 帳號。
2. 下載：Microsoft 官網搜尋「Power Automate Desktop Download」。
3. 安裝選項：
 - (選) 安裝瀏覽器擴充 (Edge / Chrome) 供網頁錄製。
 - 啟用「UI 自動化」相關權限。
4. 首次啟動登入組織帳號。

5. (可選) 設定 Unattended 執行：需對應授權 + Windows 使用者可登入。

建議：建立專用虛擬機 (VM) 作為穩定執行環境，避免被手動操作干擾。

3. 介面導覽與核心概念

元件	說明	實務建議
Actions 動作庫	分類：變數、檔案、Excel、Outlook、Web、自動化	用搜尋快速定位；常用收藏星號
主流程 / 子流程	可模組化重複邏輯	分層：啟動 / 資料準備 / 執行 / 輸出
變數面板	即時查看目前變數值	命名： <code>var_</code> 前綴 + 類型，例如 <code>listInvoices</code>
UI Elements	儲存擷取的控制項定位資訊	避免純座標；必要時多策略 (Selector + OCR)
資料類型	Text, Number, List, DataTable, Boolean, Datetime, Dictionary	轉型時使用「轉換」動作避免錯誤
On Error	每個動作可自訂錯誤策略	重要步驟設定 Retry+紀錄Log

重要模式：

- 子流程拆分：輸入參數 / 輸出參數 → 可重用與測試。
- 資料結構：大量列資料優先使用 DataTable，再轉 List。
- 狀態紀錄：流程關鍵節點寫入 CSV / Excel 日誌。

4. 第一個入門流程：記事本自動輸入

目標：開啟記事本 → 輸入今日日期與一段文字 → 另存檔案。

步驟：

1. 新建流程：`NotepadIntro`。
2. 動作：Launch application → 路徑 `notepad.exe`。
3. 動作：Get current date and time → 變數 `var_now` 格式 `yyyy-MM-dd HH:mm`。
4. 動作：Send keys → 內容：`今日時間: %var_now%{ENTER}`這是第一個PAD流程。
5. 動作：Send keys → `^s` 觸發儲存。
6. 動作：Populate text field (或 Send keys) 輸入檔名：`log_%var_now%.txt`。
7. 動作：Press button (或 Send keys Enter) 。

加強：包一層 Try (Scope 子流程) → 若失敗寫入 `error_log.csv`。

5. Excel 自動化實務

情境一：讀取 Excel 「訂單」資料 → 篩選狀態為 Pending → 計算金額小計 → 產出彙總。

必備概念：

- 啟動 Excel：Launch Excel (可選是否可見)
- 開啟活頁簿：Open document
- 讀取資料表：Read from Excel worksheet (成 DataTable)
- DataTable → List：Get first free column/row / Convert

步驟 (簡化)：

1. Launch Excel (隱藏) → 儲存至變數 `ExcelInstance`。
2. Open document：檔案 `orders.xlsx`。
3. Read from Excel worksheet：輸出 `dt_orders`。
4. For each `row` in `dt_orders`：
 - If 條件：`%row['Status']% = 'Pending'`
 - 計算：`val_subtotal = CInt(row['Qty']) * CDb1(row['UnitPrice'])`
 - Append to list：`list_pending` 加入字典 {OrderId: `row['OrderID']`, Subtotal: `val_subtotal`}
5. 建 DataTable：Add data table column (OrderId), (Subtotal)
6. 將 `list_pending` 迴圈加入 DataTable。
7. Write to Excel worksheet：寫入新活頁簿 `pending_report.xlsx`。
8. 儲存 / 關閉所有 Excel。

情境二：批次填寫模板 (Mail Merge 類似)

- 讀取客戶清單 → 開啟 Word 模板不易？可使用記事本 / HTML + 變數 → 生成檔案。
- 或改由 Excel 寫入個別 Sheet 供後續人工檢閱。

最佳實務：

- 避免逐列即寫 → 改先在記憶體 DataTable 完成，再一次 Write。
- 大檔案拆批：每 1000 列輸出一檔。
- 釋放：流程結尾務必 Close Excel (防資源緊繃)。

錯誤處理：對 Read/Write 設 On Error → Retry 2 次；仍失敗 → 截圖 (Take screenshot) + 記錄列號。

6. Outlook 自動化實務

案例：每日下載主旨含「發票」且未讀之郵件附件 → 彙整寄件者 → 輸出列表。

步驟：

1. 動作：Retrieve emails (資料夾：收件匣，未讀 Only, Top=50)。
2. For each `mail` in `RetrievedEmails`：
 - If `Contains(toLower(%mail.Subject%), '發票')` 且 `mail.HasAttachments = True`
 - For each `att` in `mail.Attachments` → Save attachment (路徑：`C:\RPA\invoices\%att.Name%`)
 - Append to list：`list_senders` 加 `%mail.From%`
3. 去重：Use Remove duplicates from list → `list_senders_unique`
4. 建立文字檔：`today_senders.txt` → 寫入每行寄件者。
5. (選) Send email：自動回覆「已收到附件，將於 24 小時內處理」。

注意：

- 需已設定 Outlook Profile (桌面版) 。
- 避免一次抓太多→ 用日期過濾 (過去 1 天) 。
- 已處理郵件可移動至子資料夾 (Move email) 。

延伸：與 Excel 交叉比對寄件者是否為合法供應商；不在名單 → 另存警示資料夾。

7. 網頁資料擷取 (Web 爬取)

案例：登入網站 → 搜尋關鍵字 → 擷取結果表格 → 匯出 Excel。

步驟：

1. Launch new Edge / Chrome → 開新視窗 (乾淨 Session) 。
2. Navigate to Web page：網址登入頁。
3. 輸入帳號 / 密碼：Populate text field (UI 元素) → Click 登入。
4. 等待元素：Wait for UI element appear (搜尋框) 。
5. Populate text field：輸入 `%var_keyword%`。
6. Click：搜尋。
7. Extract data from web page → 選取表格 → 輸出 DataTable `dt_results`。
8. Write to Excel worksheet → `search_%var_keyword%.xlsx`。
9. Close browser。

動態元素技巧：

- 若元素 ID 會變：改用 XPath / CSS (在擷取設定中切換進階) 。
- 若無法定位：使用「滑鼠模擬」最後手段，搭配螢幕解析度固定。
- 翻頁：偵測「下一頁」按鈕是否 Enable；While 條件為 True 迴圈。

防阻擋：

- 增加隨機延遲 (Random number → Delay) 模擬人類。
- 避免極短頻率重複登入。

8. 迴圈與條件判斷設計模式

模式	說明	範例
For Each	對 List / DataTable 列處理	逐列計算金額
While / Loop	直到條件不成立	網頁翻頁直到無下一頁
If	單一條件分支	檢查附件存在
Else If (Nested If)	多情境	狀態分：New / Pending / Closed
Switch (模擬)	用多重 If + 結束	依文件類型選擇子流程
Retry	On Error + 計數器	連線失敗重試 3 次

範例：While 迴圈翻頁

1. 設 `bool_hasNext = True`
2. While `bool_hasNext = True` :
 - 擷取目前表格 → Append → 檢查「下一頁」按鈕是否存在
 - If 不存在 → `bool_hasNext = False` Else Click 下一頁 + 延遲 1~2 秒

重試範例 (API 呼叫) :

1. 設 `retryCount=0`
2. While `retryCount < 3` :
 - 呼叫 API → If 成功 → Break ; 若錯誤 → `retryCount = retryCount + 1` + Delay $2^{\text{retryCount}}$ 秒
3. If `retryCount = 3` → 記錄重大錯誤。

9. 串接 OpenAI / Azure OpenAI Web API

目標：讀取一組文字描述 (例如客服訊息) → 串接模型取得摘要 → 寫回 Excel。

前置：

- 取得 API Key (OpenAI 官方或 Azure OpenAI) 。
- 端點 (示例 OpenAI v1 Chat) : <https://api.openai.com/v1/chat/completions>
- 模型 : `gpt-4o-mini` (依實際可用) 。

安全存放：

- API Key 不寫死流程：啟動時使用「輸入對話框」或讀取加密文字檔。
- (進階) 可使用 Windows 憑證管理員 (Credential Manager + PowerShell) 間接讀取。

流程步驟 (簡化) :

1. 從 Excel 讀取待摘要欄位 → DataTable `dt_msgs`。
2. For each `row` in `dt_msgs` :
 - 準備 JSON 字串：

```
{
  "model": "gpt-4o-mini",
  "messages": [
    {"role": "system", "content": "你是一個協助產生50字以內摘要的助理"},
    {"role": "user", "content": "%row[ 'Message' ]%"}
  ],
  "temperature": 0.2
}
```

```
- 動作 : Invoke web service (POST)
* URL : `https://api.openai.com/v1/chat/completions`
* Headers : `Authorization: Bearer %var_apiKey%`, `Content-Type:
```

```
application/json`
  * Body：前述 JSON 文字（注意跳脫）
  - 解析回應：用 Parse JSON 或以 Retrieve value from JSON → 路徑
`choices[0].message.content`
  - 寫入 DataTable 新欄位 `Summary`。
```

3. 回寫 Excel（新增欄）。

錯誤與速率限制：

- 若 HTTP 回 429 → Delay 10 秒後重試（指數退避）。
- 若超過字數 → 可先截斷內容 `%Left(row['Message'], 1000)%`。

隱私建議：

- 嚴格資料避免外傳；可於內部只做關鍵字標註，不送完整內容。

10. 錯誤處理與穩定性

層級策略：

1. 動作：設定 On Error → Retry（漸進 delay）→ Failure 分支。
2. 子流程：統一輸出 `IsSuccess, ErrorMessage`。
3. 全域：主流程建立 `list_logs`，所有異常 Append（時間戳 + 模組 + 訊息）。
4. 結束：若 `list_logs` 非空 → 輸出 `error_log_<日期>.csv`。

截圖：關鍵 UI 失敗 → Take screenshot（加入失敗檔名 pattern）。

超時：等待元素時預設 Timeout 不宜過長，可自訂「重試多次 + 每次 5 秒」優於一次 60 秒。

11. 效能與最佳化

問題	作法
Excel 開啟慢	合併多次讀取為一次 Read；隱藏模式運行
頻繁 UI 等待	預判元素存在 → 若已存在略過等待
大量列表處理	先 Filter → 再 For Each
多次寫檔	累積緩衝於 List / DataTable 後一次寫入
API 速率限制	併發改為序列 + 延遲控制
重複登入	建立保持 Session 子流程（檢查已登入標記）

資源釋放：

- 結束前：關閉瀏覽器 / Excel / 釋放物件。
- 移除不再使用的大 DataTable（設定空 List）。

12. 安全與憑證保護

- API Key：不硬編，啟動時輸入或讀加密檔（自訂簡單 XOR / Base64 + 內部策略）。
- 帳密：使用 Windows Credential Manager；PAD 讀取後存於暫時變數，使用完清空。
- 檔案權限：將輸出 / Log 放在限制資料夾（NTFS 權限最小化）。
- 錄製時避免：畫面暴露敏感資訊（可先以假資料錄製再替換）。

13. 常見問題 (FAQ)

問題	可能原因	建議解法
找不到 UI 元素	視窗標題變 / DOM 動態	重新擷取 + 使用條件屬性；加入等待
Excel 卡住不關閉	未 Close 或例外中斷	加 Finally 子流程統一釋放
API 回 401	Key 錯誤 / Header 缺失	檢查 Authorization 格式 Bearer
瀏覽器常被登出	Session 過期	建立登入檢查；失敗再登入
執行速度忽快忽慢	網路 / 元素等待策略不佳	改成顯式 Wait for element
中文亂碼	編碼	指定 UTF-8（寫檔時選擇編碼）

14. 進階技巧與學習路徑

建議練習順序：

1. 基本 UI 操作（記事本 / 計算機）
2. Excel 批次處理 + 輸出報表
3. Outlook 郵件搜集 + 附件下載
4. 網頁登入 + 多頁爬取
5. OpenAI API 摘要加註分類
6. 整合所有模組成「每日自動報表流程」

延伸：

- 使用 Windows 工作排程（Task Scheduler）定時啟動流程。
- 與指令列 / PowerShell 混合（啟動外部腳本）。
- 節點化：建立共用子流程庫（Login、ExcelInit、ApiCall）。

附錄 A：常用動作速查

類別	動作	說明
Excel	Launch / Open / Read / Write / Close	啟動→操作→關閉完整生命週期
變數	Set variable / Increase variable	控制計數器、旗標
清單	Add item to list / Remove duplicates	產生結果集合
DataTable	Create / Add row / Convert	結構化批次資料

類別	動作	說明
流程控制	If / Loop / For each / Switch (模擬)	邏輯判斷
錯誤處理	On error / Take screenshot	錯誤捕捉與證據
Web	Launch new browser / Populate text field / Click link	UI 操作
Web 資料	Extract data from web page	表格爬取
Outlook	Retrieve emails / Save attachment / Send email	郵件處理
系統	Launch application / Run DOS command	外部程式
API	Invoke web service	HTTP REST 呼叫

命名建議：

- 子流程：Sub_<功能> e.g. Sub_Login, Sub_ExportExcel
- 變數：var_、List 用 list_、DataTable 用 dt_、布林is_、數字cnt_。

附錄 B：範例流程骨架模板

```
Main
├─ Sub_Init
│   ├── 設定日期 / 路徑 / 讀取設定檔
│   └─ 建立日誌結構 list_logs
├─ Sub_Login (可重試)
├─ Sub_ExtractData (翻頁 + 收集 DataTable)
├─ Sub_ProcessData (計算 / 清洗)
├─ Sub_CallAI (摘要 / 標籤)
├─ Sub_Output (Excel / CSV / 郵件)
└─ Sub_Finalize (關閉資源 / 輸出日誌)
```

總結

本手冊聚焦 PAD：以 Excel / Outlook / 網頁擷取 / 迴圈與錯誤控制 / OpenAI API 為核心情境。建議挑一個日常工作（如：每日收信下載 + 整理 + AI 摘要）作為整合實作，迭代加入穩定性與安全性後，再複製模式至更多流程。

(完)