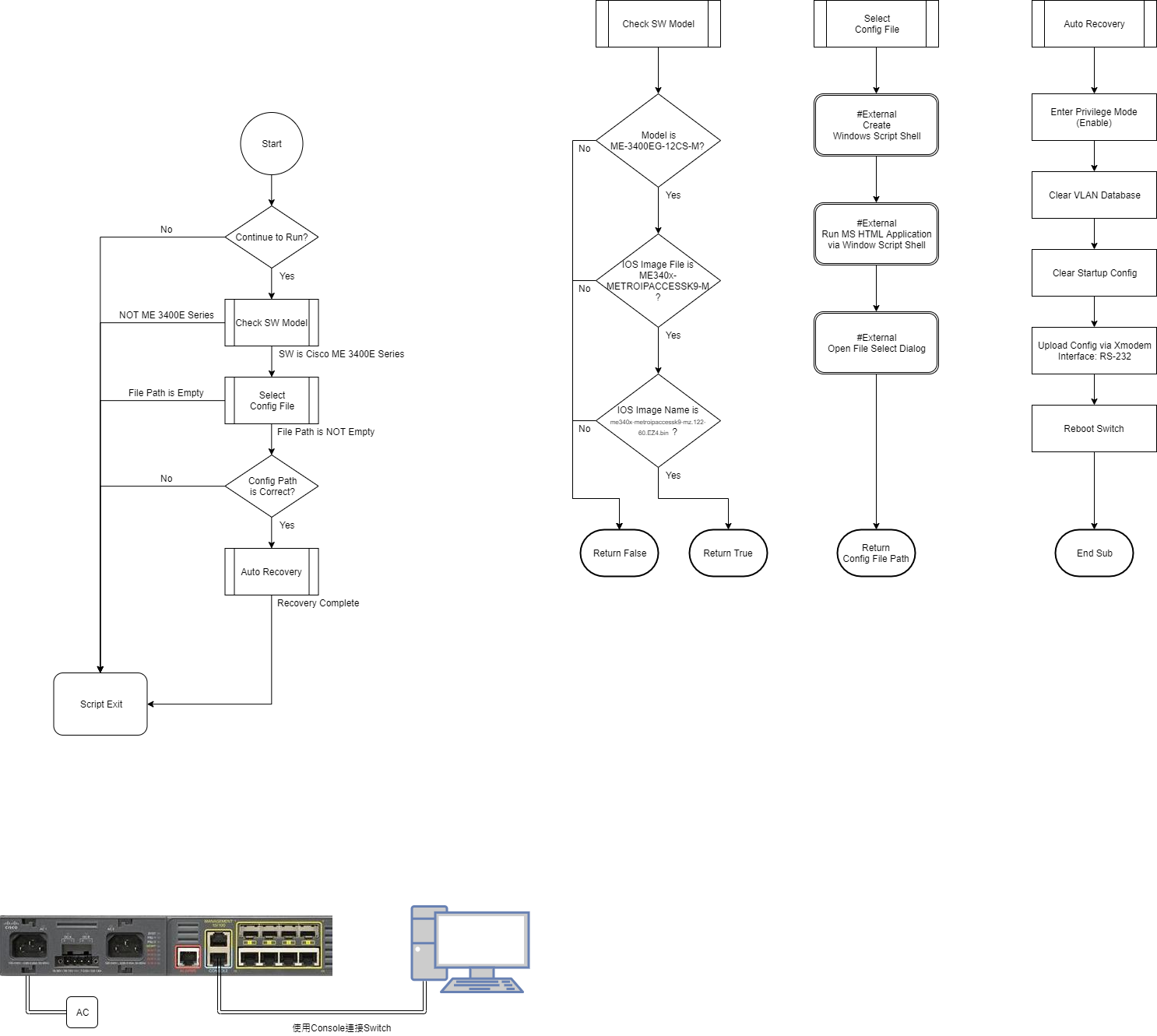
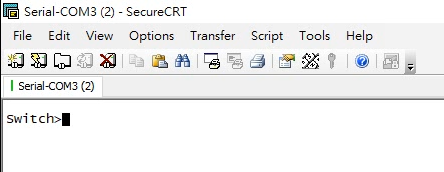
Cisco ME 3400E Series Configuration File 自動還原腳本使用說明

蔣政燁 2021/10/17 ver.01

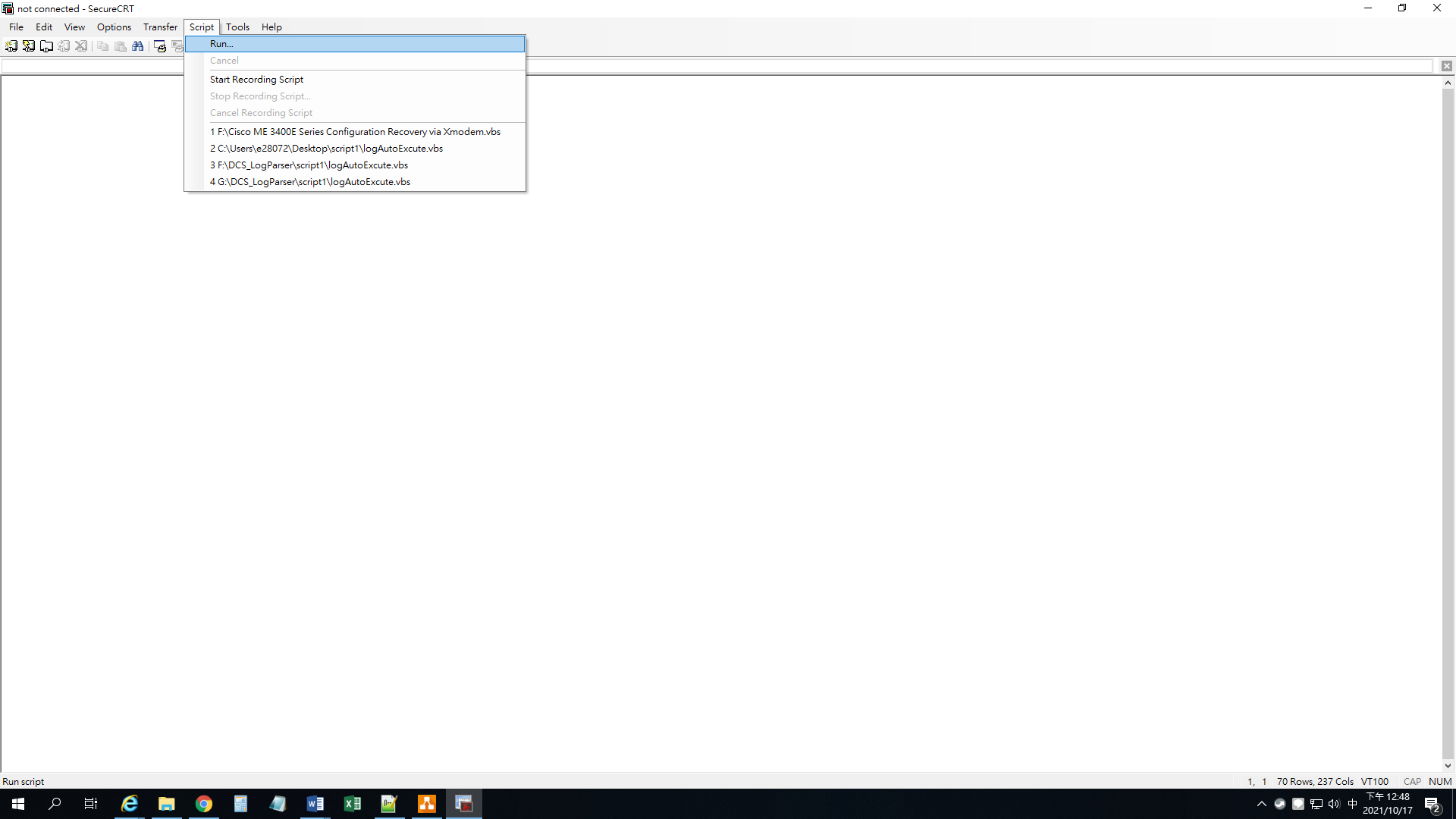
1. 備妥：
   1. 電腦1台（含SecureCRT v6.2.0以上版本）
   2. Rollover Cable + RJ-45 to RS-232轉換頭 或 Cisco 專用Console平行線
   3. RS-232 to USB轉換器
   4. 欲還原的交換機設定檔及自動還原腳本
   5. 新品交換機
2. 步驟
3. 將電腦以Console方式與交換機連接，並將交換機上電開機

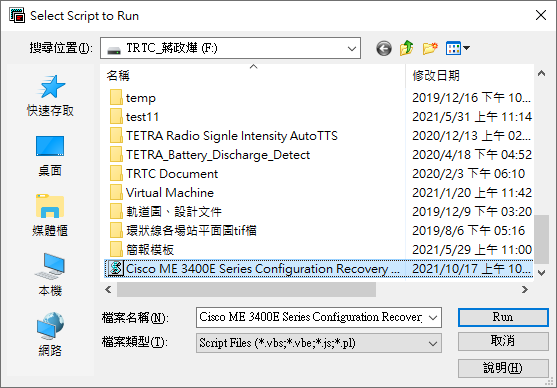


1. 使用SecureCRT以串列連線方式連入交換機

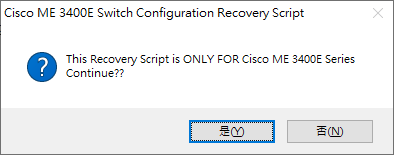


1. 開機完成後，畫面出現 [交換機名稱#] 或 [Switch>](空機) 提示字元時，點選功能選單Script 🡪 Run…，並選擇自動還原腳本

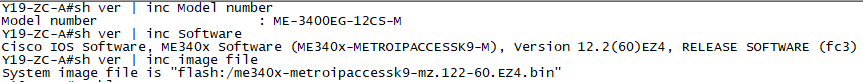




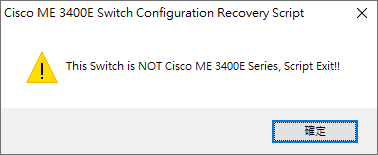
1. 出現「此腳本僅適用Cisco ME 3400E Series交換機」提示訊息，按下「是」繼續執行，按下「否」結束腳本



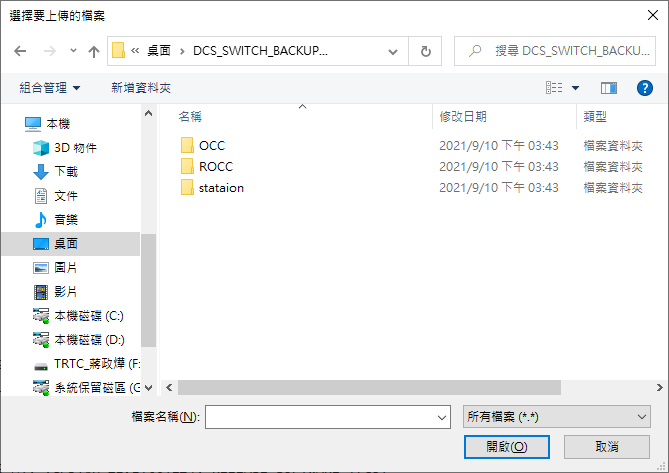
1. 自動還原腳本將檢查下列三項資訊是否相符
   1. 型號：ME-3400EG-12CS-M
   2. IOS 軟體系列：ME340x-METROIPACCESSK9-M
   3. IOS 檔案名稱：me340x-metroipaccessk9-mz.122-60.EZ4.bin



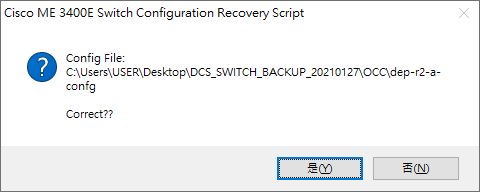
當其中任一項資訊不符時，腳本將會出現「此交換機並非Cisco ME 3400E Series」提示訊息並自動結束執行



1. 通過交換機型號檢查後，會出現檔案選擇視窗，請選擇要還原的交換機設定檔



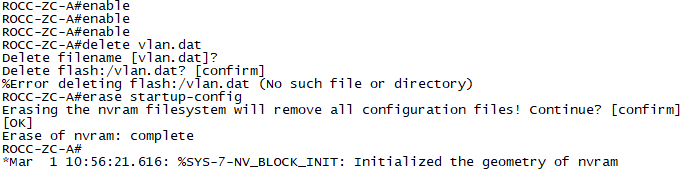
1. 因後續檔案上傳及重新啟動指令皆為自動執行，為避免誤選檔案，上傳非供Switch使用之設定檔，造成設備不正常運作。在選擇檔案後，會要求再次確認所選擇的檔案路徑是否正確



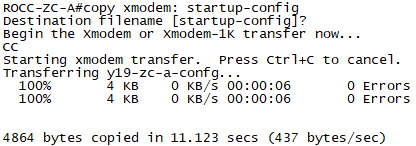
倘所選擇的檔案錯誤，請按下「否」結束執行，並回到第3步重新啟動腳本。

如選擇正確，請按下「是」繼續執行

1. 確認所選檔案無誤後，腳本將自動進入特權執行模式(Privilege / Enable Mode)，並清除既有VLAN Database、Startup Config



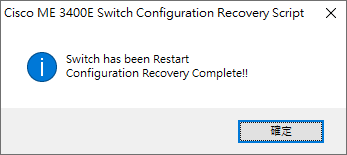
1. 清除完成後，會啟動Xmodem傳輸協定，將設定檔上傳至交換機



Cisco Switch 預設使用Xmodem-CRC進行循環冗餘校驗及檔案完整性驗證

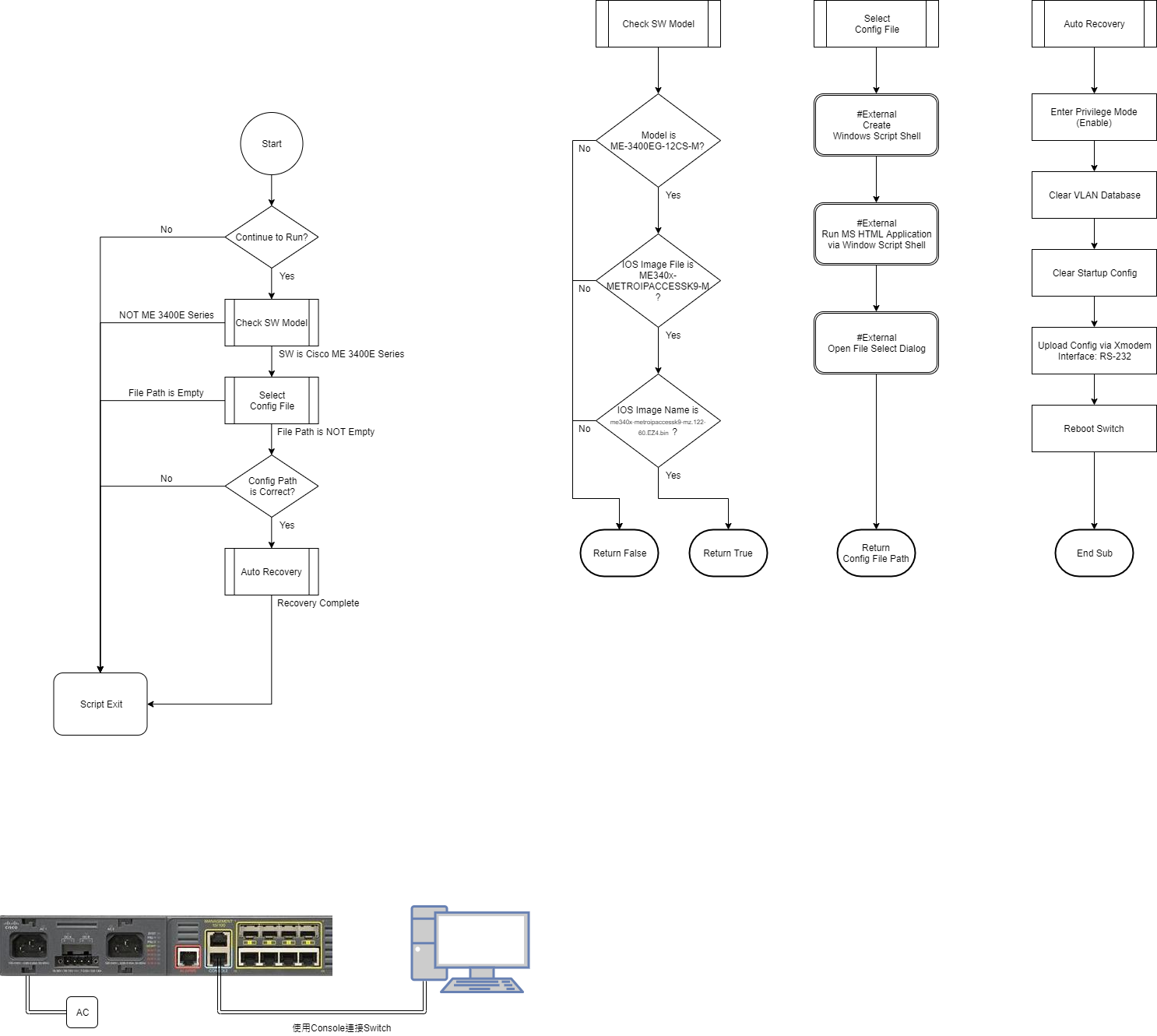
此外，SecureCRT預設Xmodem Block之傳輸單位長度為128 Bytes + 5 Bytes

1. 設定檔上傳完成後，交換機將自動重啟，並會出現還原完成之提示訊息

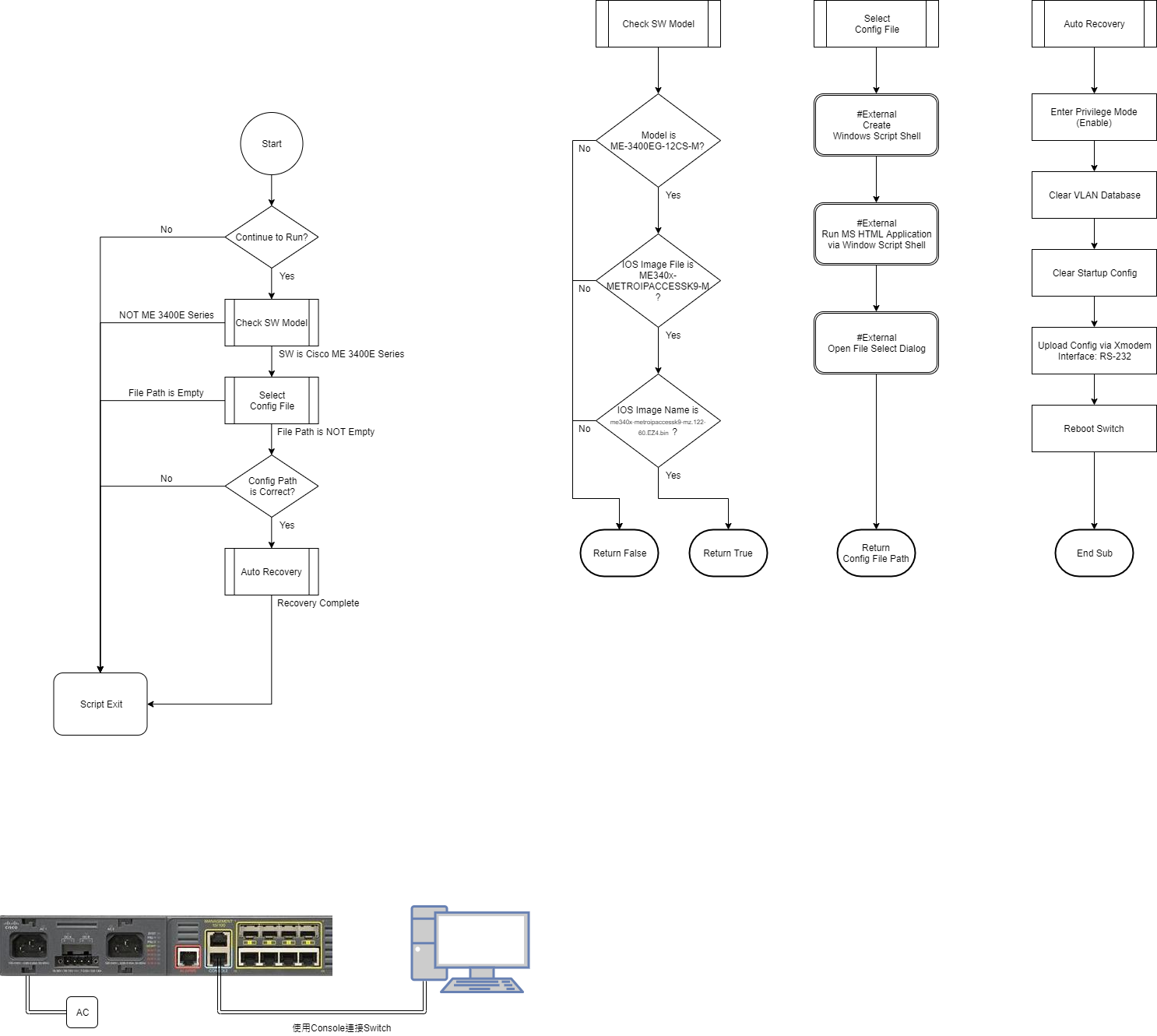


1. 腳本設計流程圖

主架構流程圖



模組流程圖



1. Xmodem 傳輸協定

Xmodem協定開發於1977年，為串列通訊中一種古老的泛用型非同步文件傳輸協定，可以做為Cisco設備上傳設定檔、IOS作業系統或救援模式的傳輸協定。

在Xmodem標準模式下，一個完整的資料塊(Block)長度固定為132 Bytes，其中包括128 Bytes的數據區塊及4 Bytes的控制區塊，封包格式如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SOH | Block No. | Block No.(反碼) | Data Block | Checksum |
| 1 Bytes | 1 Bytes | 1 Bytes | 128 Bytes | 1 Bytes |

其中資料塊均以SOH控制碼(^A, 0x01)為開頭，並附上區塊編號(0x01 ~ 0xFE)、反向區塊編號(0xFE ~ 0x01)。最後為校驗和，其值為數據區塊內容參照ASCII Table之Hex值總和除以256的餘數。

需注意的是，不論傳送資料的大小，每個資料塊中的數據區塊長度均固定為128 Bytes。意即，當傳輸至資料末端時，數據長度可能會不滿足 128 Bytes，此時協定將以控制碼Ctrl-Z(^Z, 0x1A)填補剩餘空間。

Xmodem協定亦有其他增強型變體，如Xmodem-CRC及Xmodem-1K，以及改良協定如Ymodem、Zmodem及Kermit等。

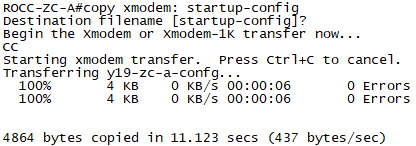
Cisco設備即預設使用Xmodem-CRC格式進行傳輸。

Xmodem-CRC與標準模式相差不多，僅將資料塊最後的「校驗和」改為「循環冗餘校驗」，資料塊長度則固定為133 Bytes。

其中 16bit CRC 可拆分為高、低位元組，意即CRC-H (8bit)、CRC-L (8bit)。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SOH | Block No. | Block No.(反碼) | Data Block | 16bit CRC |
| 1 Bytes | 1 Bytes | 1 Bytes | 128 Bytes | 2 Bytes |

該模式在VTY虛擬終端畫面上的顯式特徵為，於啟動傳輸時，Rx會向Tx發送字母C(0x43)，藉以向傳送端表示將使用CRC循環冗餘校驗。在標準校驗和模式下則改為發送控制碼NAK(^U, 0x15)



Xmodem-1K模式強制採用CRC循環冗餘校驗，數據區塊長度由128 Bytes提升至1024 Bytes，資料塊表頭以控制碼STX (^B, 0x02)取代控制碼SOH (^A, 0x01)。資料塊總長為1029 Bytes。

因為數據區塊長度的增加，Xmodem-1K的傳輸速率較CRC及標準模式來的有效率。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STX | Block No. | Block No.(反碼) | Data Block | 16bit CRC |
| 1 Bytes | 1 Bytes | 1 Bytes | 1024 Bytes | 2 Bytes |

與標準模式相同，當資料塊中的數據區塊長度不足1024 Bytes時，協定將以控制碼Ctrl-Z(^Z, 0x1A)進行填補，對於交換機或嵌入式單板電腦等具有記憶體容量限制之設備而言，此模式在特定情形下具有較高程度的記憶體浪費。

下表為Xmodem協定的控制碼定義：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 控制碼 | ASCII  Hex Code | 跳脫字元 | 用途 | 發送  角色 |
| SOH | 0x01 | ^A | 資料塊表頭，用以表示數據區塊為128 Bytes | Tx |
| STX | 0x02 | ^B | 資料塊表頭，用以表示數據區塊為1024 Bytes | Tx |
| ETX | 0x03 | ^C / Ctrl-C | Tx手動終止傳輸，Rx不須再傳送ACK | Tx |
| EOT | 0x04 | ^D | 傳輸結束 | Tx |
| ACK | 0x06 | ^F | 資料塊校驗無誤，可發送下一個資料塊 | Rx |
| NAK | 0x15 | ^U | 1. 在標準模式下等待接收，通知Tx使用校驗和  2. 資料塊校驗失敗，通知Tx重送 | Rx |
| CAN | 0x18 | ^X | Rx無條件終止傳輸，Tx不須再傳送EOT | Rx |
| SUB | 0x1A | ^Z / Ctrl-Z | 數據區塊末端填充用控制碼 | Tx |
| 字母C | 0x43 | N/A | 等待接收，通知Tx使用CRC循環冗餘校驗 | Rx |

Xmodem傳輸過程以Rx為主體，下列為Xmodem-CRC應答流程範例：

|  |  |  |  |
| --- | --- | --- | --- |
| Tx | 方向 | Rx | 說明 |
|  | << | C | 以CRC模式等待接收 |
| SOH 01 FE Data[128] CRC-H CRC-L | >> |  | 以128 Bytes長度傳送 |
|  | << | ACK | 校驗無誤 |
| SOH 02 FD Data[128] CRC-H CRC-L | >> |  |  |
|  | << | ACK |  |
| SOH 03 FC Data[128] CRC-H CRC-L | >> |  |  |
|  | << | NAK | 校驗失敗 |
| SOH 03 FC Data[128] CRC-H CRC-L | >> |  | 重送資料塊 |
|  | << | ACK |  |
| SOH 04 FB Data[100] 0x1A[28] CRC-H CRC-L | >> |  | 資料末端，以Ctrl-Z(^Z, 0x1A)填充 |
|  | << | ACK |  |
| EOT | >> |  | Tx傳輸結束 |
|  | << | ACK | Rx傳輸結束 |

1. Xmodem資料塊傳輸之驗證處理

標準校驗和模式資料塊範例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表頭 | Block No. | Block No.(反碼) | Data Block | Checksum |
| 1 Bytes | 1 Bytes | 1 Bytes | 128 Bytes | 1 Bytes |
| SOH, 0x01 | 0x03 | 0xFC | 0x41[50] 0x61[50] 0x1A[28] | 0x7C |

Rx在接收到以控制碼SOH / STX為開頭的資料塊後，將進行下列幾個步驟藉以確認資料傳輸完整性(以上述資料塊為計算範例)：

1. 確認資料塊序號完整性：

Block No. + Block No.(反碼)，其值應為0xFF。

0x03 + 0xFC = 0xFF

Block No. NOR Block No.(反碼)，其值應為0x00

0x03 NOR 0xFC = 0xFFFF FFFF FFFF FF00 (OverFlow)

因資料塊序號僅佔1個位元組，故其計算產生之溢位位元不計入驗證結果，最終值為0x00

倘上述兩項檢查不符預期，則傳送控制碼NAK(^U, 0x15)請求Tx重送資料塊

1. 檢查資料塊序號是否為期望值：

依上述資料塊範例，其前一資料塊序號為0x02，故次一序號期望值應為0x03，倘收到非序號0x03之資料塊時，Rx將發出控制碼CAN(^X, 0x18)終止傳輸。

倘收到之資料塊序號與前一資料塊序號相同，則Rx將忽略重複發送的Block，並向Tx發送控制碼ACK(^F, 0x06)。

1. 進行校驗和(Checksum)驗證

對數據區塊進行驗證，以標準模式校驗和(Checksum)為例，其計算方式如下所述：

|  |  |
| --- | --- |
| Data Block (128 Bytes) | |
| 數據內容 | 50個大寫A、50個小寫a、28個末端控制碼Ctrl-Z(^Z, 0x1A) |
| ASCII Hex Code | 0x41[50] 0x61[50] 0x1A[28] |

其數據總和值 = (0x41 \* 50) + (0x61 \* 50) + (0x1A \* 28)

= 0x0CB2 + 0x12F2 + 0x02D8

= 0x227C

總和值餘除256之Checksum為：0x227C Mod 256 = 0x7C

最終結果應與資料塊中最後一個位元組相同。倘驗證結果相同則傳送控制碼ACK(^F, 0x06)，不同則傳送控制碼NAK(^U, 0x15)。

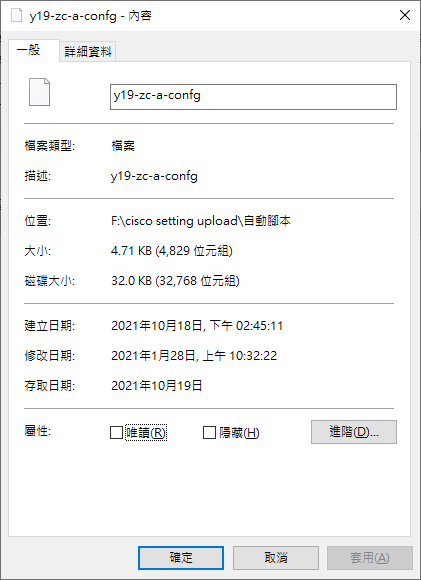
1. Xmodem數據區塊末端Ctrl-Z填充

Xmodem數據區塊在所乘載之資料未滿128 Bytes時(1K模式為1024 Bytes)，會以控制碼Ctrl-Z(^Z, 0x1A)進行填補，其附加長度計算方式如下：

標準模式：128 – [FileSize(Bytes) Mod 128] = 控制碼填補長度

1K模式：1024– [FileSize(Bytes) Mod 1024] = 控制碼填補長度

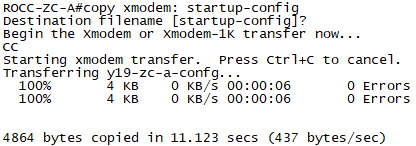
以Y19-ZC-A之Configuration File為例，原始檔案長度為4829 Bytes，透過Xmodem上傳至Cisco交換機後，其檔案末端將會被附加上35個控制碼Ctrl-Z(^Z, 0x1A)。

= 128 – [4829 Mod 128]

= 128 – 93

= 35個

另於Cisco交換機所顯示的傳輸資訊內，亦可檢視附加填充控制碼後的總資料長度

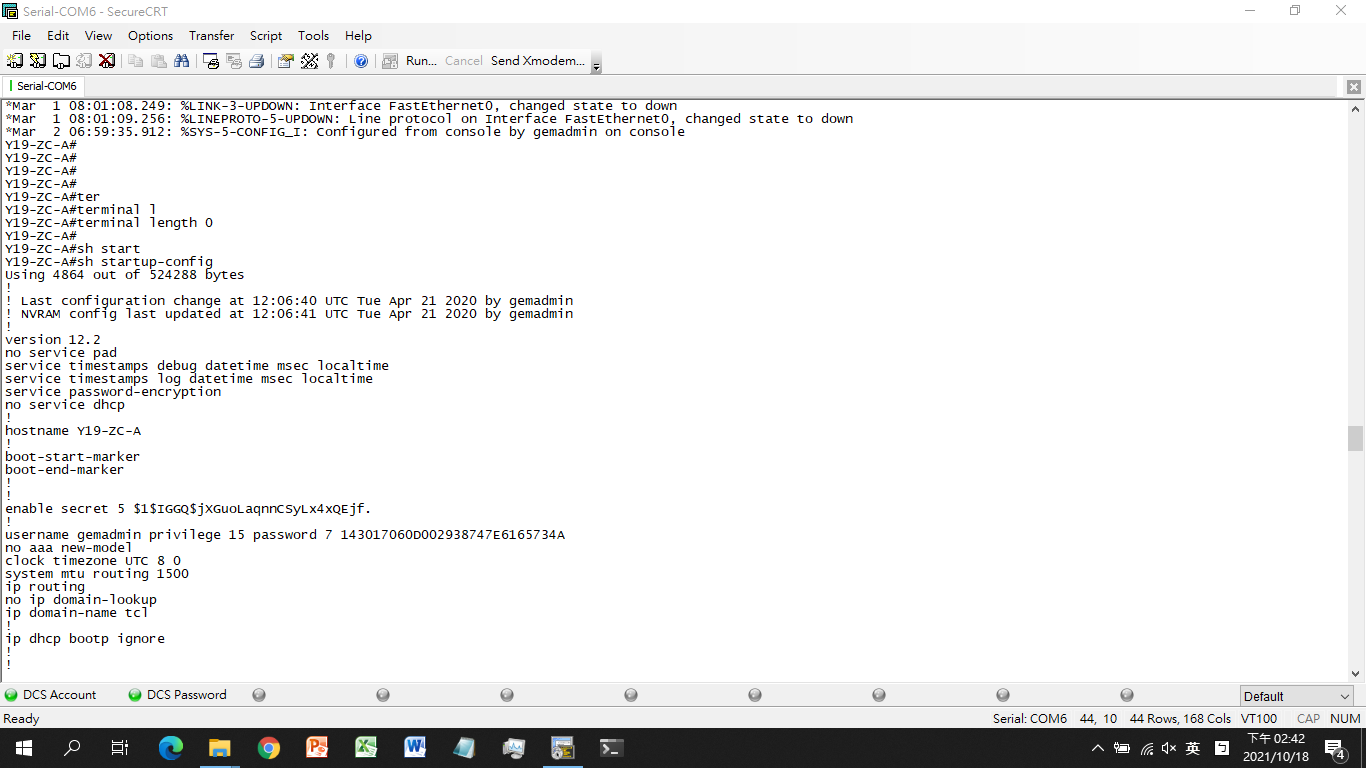


= 4829 Bytes + 35 Bytes

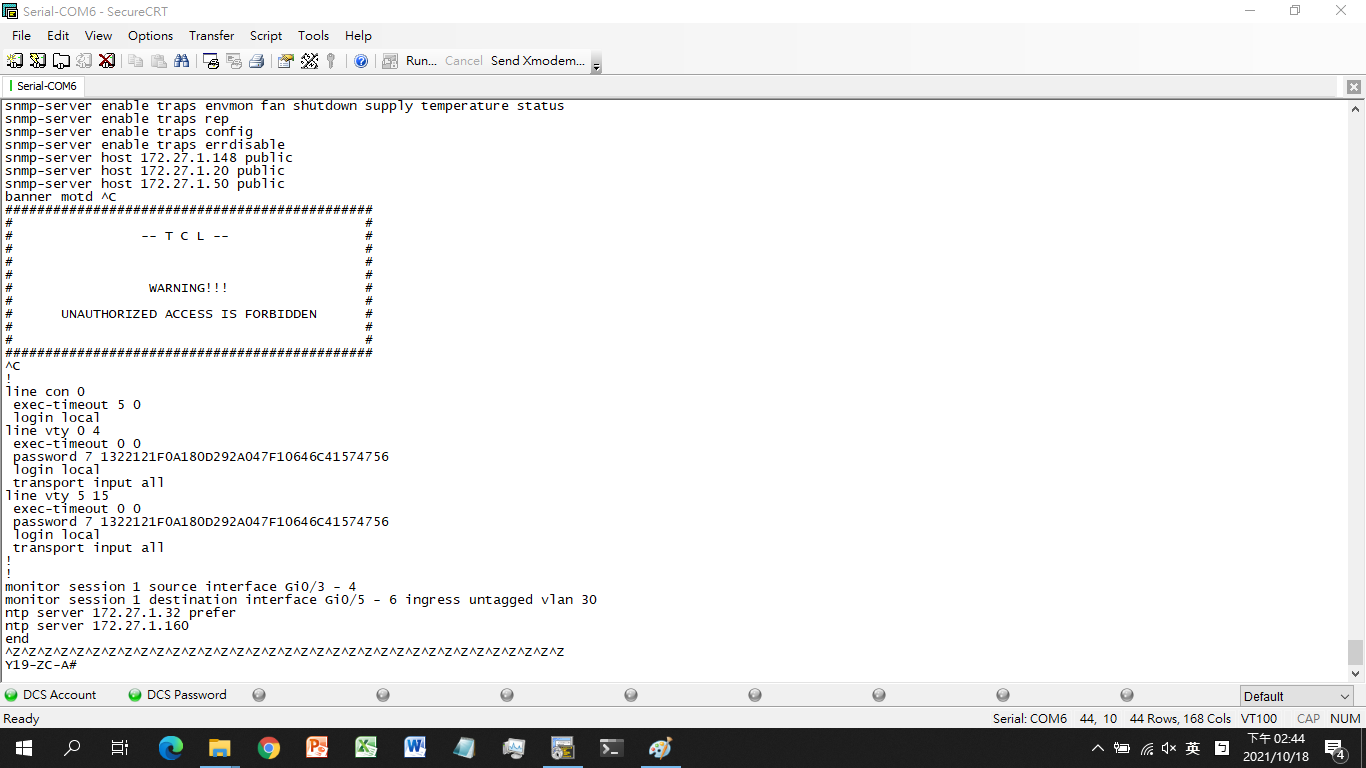
= 4864 Bytes

※總資料長度必為128之整數倍，1K模式則為1024之整數倍

於Cisco交換機檢視Startup Config，其檔案長度與傳輸長度相符

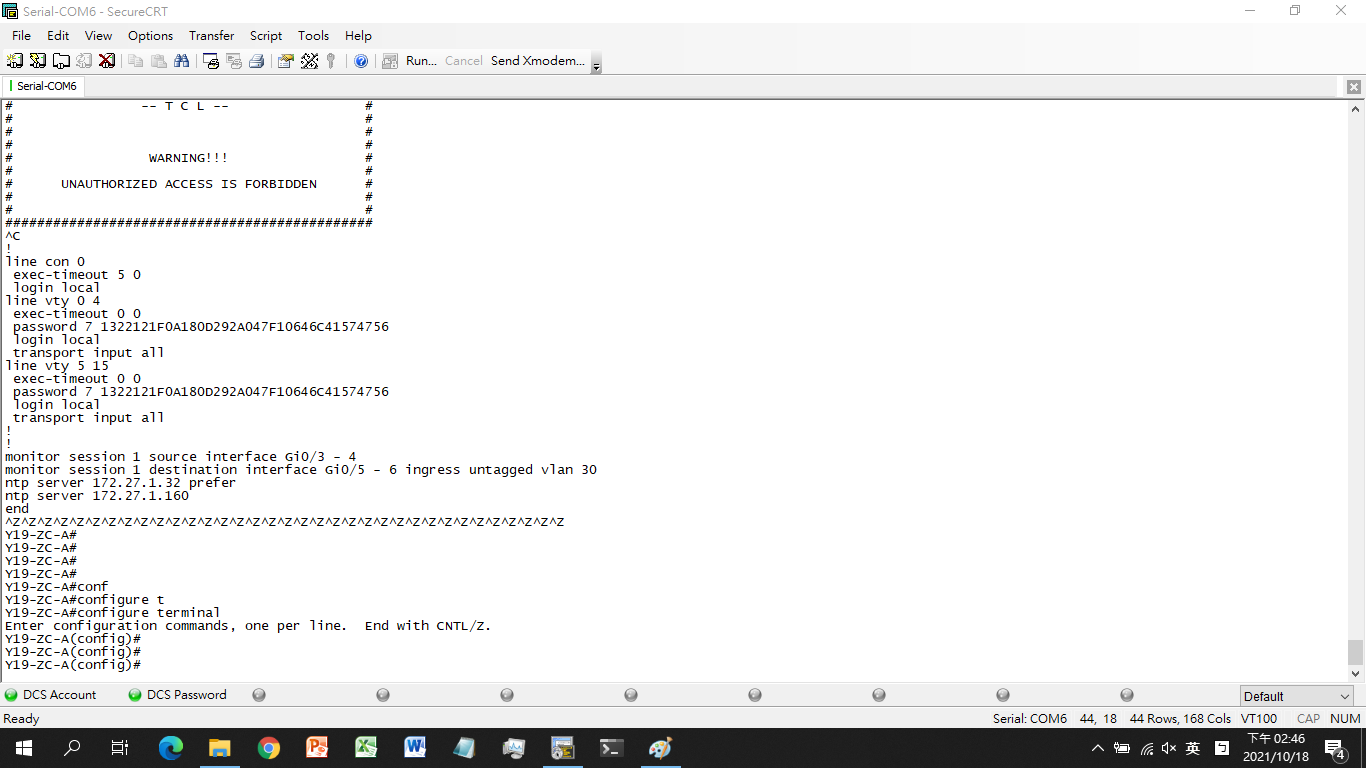


亦可於末端看到相等數量之^Z(此範例為35個^Z)



控制碼Ctrl-Z(^Z, 0x1A)雖被填充至Startup Config末端，但對於Cisco IOS而言，並不影響其Bootloader、POST、初始化及讀取Startup Config至Running Config之過程，其原因如下：

1. Cisco IOS為運作於IBM PowerPC處理器之Unix-like網路作業系統。在原生Unix作業系統中，Ctrl-Z被用於退出當前正在執行的互動式執行緒。
2. 在同屬Unix-like的CP/M作業系統(1974-1983)內，^Z(0x1A)被視為檔案末端及填充剩餘檔案空間(與Xmodem原理相同)
3. 在Startup Config內，填充控制碼前的最後一行指令為「end」。對於Cisco IOS而言，該指令為退出特權執行模式(Privilege / Enable Mode)。
4. 對Cisco IOS而言，Ctrl-Z亦為退出特權執行模式(Privilege / Enable Mode)之快速鍵。



不論上傳之Configuration File內附加了多少個Ctrl-Z控制碼，或是否包含「end」指令。基於上述C.及D.項，Cisco IOS在開機過程只要讀取到「end」或Ctrl-Z控制碼兩者其一(不論先後順序)，皆會視為Startup Config結束，故其末端填充行為不影響Cisco設備之正常運作。