

網站偵測

【網站偵測】是一項網路監控的輔助功能，其主要目標是直接擷取和分析網站或網頁的資料，以判定其服務是否遭遇中斷或變異。此功能不僅對抽樣的網頁資料進行檢視，還具備對 DNS 伺服器中的網址進行解析的能力。它能執行網路連線狀態的偵測與驗證，測量網頁資料取得所需的時間，並對這些資訊進行深度分析與統計，以提供全面的網路連線品質監控。

壹、 網站偵測主要功能

- 網站內的網頁是否能順利取得資料
- 取得網站內的網頁資料花費時間
- 網頁原始碼比對

網站偵測通常發生情況

-----順利取得網站資料，網站回復狀態碼-----

【Watchdog】 ---取得網站資料--->順利取得網站資料--->網站資料回覆狀態碼

- 取得網頁的資料量低於最小資料量之警報值【自行定義數值】
- 取得網頁的資料量超出最高資料量之警報值【自行定義數值】

-----取得網站資料失敗，收取不到網站狀態碼-----

【Watchdog】 ---取得網站資料--->取得資料失敗【原因可能如下】

- 取得網頁的回應時間超出警報值
- 取得 DNS 失敗
- 網路無法連線
- 無法取得網頁

※若網站回應碼為【301】【302】【403】為轉址/無權限，則不會發出告警

由於網站資料取得不到，自然無法得到網站回覆的狀態碼，若希望當網站發生轉址時(網頁碼回應為 301、302、403)也發出告警，可以利用以下功能

貳、 網站偵測功能使用

一、 網頁索引

目的為

- 設定檢查的子頁
- 自定義【最高回應時間】、【最小資料量】以及【最大資料量】

A 級	B 級	C 級	檢 測 數 量	網 頁 索 引	原 碼 比 對	歸 屬 主 機	網 站 主 機	警 報
☐	☒	☐	1					
☐	☒	☐	1					

【最高回應時間】

取得網頁資料的回應時間超過定義的時間時發出告警

【最小資料量】

取得網頁的資料量低於最小資料量時發出告警

【最大資料量】

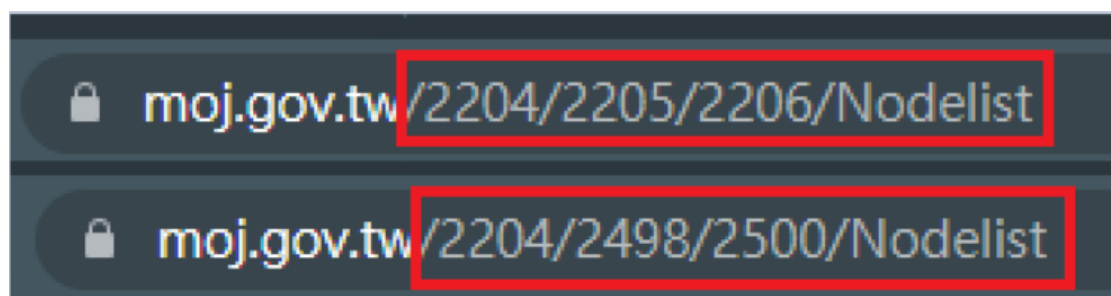
取得網頁的資料量高於最大資料量時發出告警

【檢查子頁】

例如我在【網站偵測】設定偵測的頁面為【www.moj.gov.tw】

我還希望抽樣偵測裡面的子頁，則可以點及【網頁索引】

並設先查看網頁子頁之網址，假設我要抽樣的網址如下



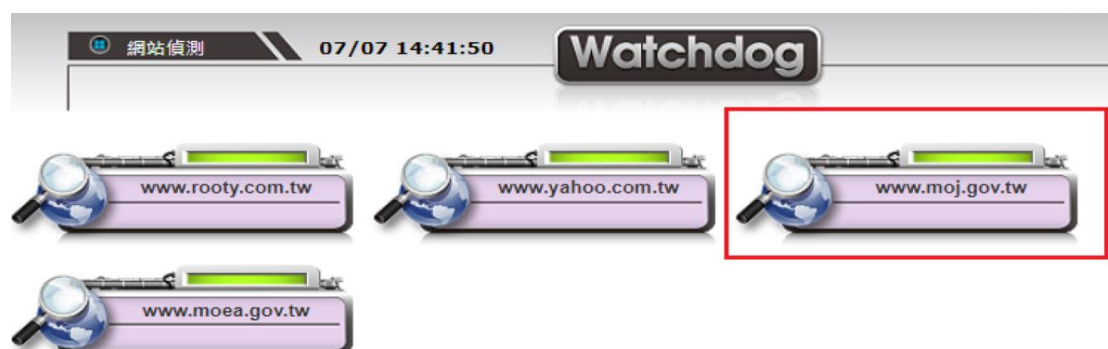
則將其紅框部分複製至【網頁索引欄位】，如

序號	啟用	暫停	刪除	編號	網頁索引
1	☒	☐	☐	00 ▾	/
2	☒	☐	☐	01 ▾	/2204/2205/2206/Nodelist
3	☒	☐	☐	02 ▾	/2204/2498/2500/Nodelist

系統便會對其母頁/子頁進行偵測
於文字型監控畫面會如下：

序號	圖標	狀態	網站位址 與 網頁路徑	通信埠	頁次	所屬IP位址	偵測時間	下次時間
1		正常	www.rooty.com.tw/	443	0	59.120.21.45	2023/07/07 14:38	2023/07/07 14:43
2		正常	www.yahoo.com.tw/	443	0	124.108.115.100	2023/07/07 14:38	2023/07/07 14:43
3		正常	www.moj.gov.tw/	443	0	163.29.130.159	2023/07/07 14:38	2023/07/07 14:43
4		正常	www.moj.gov.tw/2204/2205/2206/Nodelist	443	1	163.29.130.159	2023/07/07 14:38	2023/07/07 14:43
5		正常	www.moj.gov.tw/2204/2498/2500/Nodelist	443	2	163.29.130.159	2023/07/07 14:38	2023/07/07 14:43
6		正常	www.moea.gov.tw/	443	0	203.66.35.3	2023/07/07 14:38	2023/07/07 14:43

於圖形監控畫面會如下：



點擊設定的母頁後可看到其子頁偵測狀態

```

設備名稱:null
網站位址:www.moj.gov.tw
所屬IP位址:163.29.130.159
通信埠:443
啟動暫停:啟用
網頁伺服系統:null
警報時間:
----- 狀態/頁次/時間/訊息/網頁/回應時間/網頁資料量 -----
正常 0 2023/07/07 14:38 200 / 1.000 259362 _
正常 1 2023/07/07 14:38 200 /2204/2205/2206/Nodelist 0.350 215181 _
正常 2 2023/07/07 14:38 200 /2204/2498/2500/Nodelist 0.318 216407 _

```

若輸入為無效的網址，則會在監控畫面顯示異常狀態
如隨便設定一個網址

序號	啟用	暫停	刪除	編號	網頁索引
1	☒	☐	☐	00 ▾	/
2	☒	☐	☐	01 ▾	/2204/2205/2206/Nodelist
3	☒	☐	☐	02 ▾	/2204/2498/2500/Nodelist
4	☒	☐	☐	03 ▾	/fdfsdfsdfsdfsdfsdfs
5	☒	☐	☐	04 ▾	/191.111.165.5

經過檢測後，發現找不到該網址則會發出警報，可以看到下圖系統以發出警報，且可以進一步查看到系統查詢到【網頁資料量為 0】

3		正常	www.moj.gov.tw/	443	0		259362
4		正常	www.moj.gov.tw/2204/2205/2206/Nodelist	443	1		215181
5		正常	www.moj.gov.tw/2204/2498/2500/Nodelist	443	2	...	216407
6		警報	www.moj.gov.tw/fdfsdfsdfsdfsdfsdfs	443	3		0
7		警報	www.moj.gov.tw/191.111.165.5	443	4		0

※特別注意，當您設定子頁時，子頁的網址必須與母頁網址的開頭保持一致。如果開頭不同，即為該網址進行了轉址。在這種情況下，您需要在其他偵測頁面中獨立設定該母頁。

舉例來說，假設您的母頁網址是【tw.yahoo.com/】，並且您在首頁上點擊進入了新聞類別，則該新聞類別頁面的網址將變成【tw.news.yahoo.com/】。您會發現，新聞類別頁面的網址並未繼承母頁網址的開頭，因此，在設定母子頁面關係時，新聞類別頁面不能被視為是【tw.yahoo.com/】的子頁面。

在進行【網頁索引】設定時，請特別留意以上的規則，以確保您的設定能夠正確地反映出您想要追蹤的網頁結構。

二、 原碼比對

檢測【網頁原始碼】內的文字內容，以【一行】做為比對目標進行警報篩選

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN">
2 <html>
3 <head>
4 <meta http-equiv="Content-Type" content="text/html" ; charset="big5" />
5 <title>Watchdog 資訊維護監控系統 055-, 遠端主機, 監控系統, 代理偵測, 封包測試, IP通傳埠, 定時備案 2023 07</title>
6 <meta name="keywords" content="Watchdog 資訊維護監控系統 056-, 本機偵測與控制, 程式比對, 檔案數量, 執行程式, 硬碟使用, 目錄比對, 系統比對, 硬碟中的系統程式, 執行中的系統程式" />
7 <meta name="description" content="Watchdog 資訊維護監控系統 057-, 立即取得遠端主機資訊, 作業系統版本, CPU使用率, 記憶體使用, 硬碟使用, 網路卡設定, 網路狀態" />
```

可以點選原碼比對進入設定

A 級	B 級	C 級	檢 測 數 量	網 頁 索 引	原 碼 比 對	歸 屬 主 機	網 站 主 機	警 報
☐	☒	☐	1					
☐	☒	☐	0					

※選項說明

◆ 【設定欄位】

序號	啟用	存在	網頁編號	第一組字串&	第二組字串&	第三組字串	not 請輸入字串-1]]
00	☒	☐	00	watchdog首頁	color		
01	☐	☐	00				
02	☐	☐	00				
03	☐	☐	00				
04	☐	☐	00				
05	☐	☐	00				
06	☐	☐	00				

設定欄位有分成很多欄，若【網頁編號】皆為 00，代表每一欄皆會對【網頁索引設定為 00】的網頁進行設定的條件檢查，依欄位序號檢查，直到

- 1.【所有設定的欄位檢查完】
 - 或者
 - 2.【其中一欄條件觸發警報條件】
- 才停止

◆ 【存在選項】

【存在】選項打勾即代表，【存在設定字串則發告警】
【存在】選項不打勾即代表，【不存設定該字串則發告警】

◆ 【網頁編號】

即為【網頁索引】進入後設定的網頁編號，在設定【原碼比對】時檢查的頁面
例如:

我要在【www.rooty.test.com.tw】設定【原碼比對】條件，需在【網頁索引】
將其設定為【編號 02】的欄位，定義該網頁編號

啟用	暫停	刪除	編號	網頁索引
☒	☐	☐	00 ▼	/
☒	☐	☐	01 ▼	/www.rooty.com.tw
☒	☐	☐	02 ▼	/www.rooty.test.com.tw

之後再【原碼比對】的【網頁編號】輸入【02】，系統即會檢查
【www.rooty.test.com.tw】的網頁原始碼是否有符合設定條件

網站偵測 www.rooty.com.tw

/usr/rooty/wdog/Webpg/Cmp/www.rooty.com.tw

序號	啟用	存在	網頁編號	第一組字串&
00	☒	☐	00	watchdog首頁
01	☐	☐	02	
02	☐	☐	00	

◆ 【第 N 組字串】

這裡的設定將依據【存在】選項是否被勾選而產生不同的篩選流程。對於這兩種情境的具體說明和實例，您可以參見以下詳細介紹。

- 【不勾選存在】 / 不存在【第 N 組字串】觸發警報條件 / 存在字串視為正常

啟用	存在	網頁編號	第一組字串&	第二組字串&
☒	☐	00	監控系統	watchdog系統

在一行資料中，必須同時出現設定字串【監控系統】以及【watchdog 系統】以滿足條件。若設定的字串未在同一行內被偵測到，則系統將觸發警報。

例如：

```

17 <PRE style="padding-left:10px"><body background="/html/icons/wdog_meub
18 網頁原始碼比對-警報條件設定
19 </font><font color="#000000">
20 用途：
21 檢測網頁原始碼內的文字，以一行字串為比對目標，採用"正面或負面列表"方式
22 比對存在或符合之目標字串，若不存在或相同則條件成立(警報)
23 警報內容通常會用"第一個比對字串"文字為部份"警報內容"

```

設定

1. 第一組字串為【原始碼】
2. 第二組字串為【比對目標】
3. 不勾選存在

於此案例中，系統在第 21 行同時找到字串【原始碼】以及【比對項目】，因此不會觸發警報條件

➤ 【勾選存在】 / 存在【第 N 組字串】觸發警報條件 / 不存在字串視為正常

啟用	存在	網頁編號	第一組字串&	第二組字串&
☒	☒	00	監控系統	watchdog系統

在系統查詢網頁原始碼時，若找到同一行有字串【監控系統】以及【watchdog系統】則發送告警

例如:

```
17 <PRE style="padding-left:10px"><body background="/html/icons/wdog_meub
18 網頁原始碼比對-警報條件設定
19 </font><font color="#000000">
20 用途:
21 檢測網頁原始碼內的文字,以一行字串為比對目標,採用"正面或負面列表"方式
22 比對存在或符合之目標字串,若不存在或相同則條件成立(警報)
23 警報內容通常會用"第一個比對字串"文字為部份"警報內容"
```

設定

1. 第一組字串為【原始碼】
2. 第二組字串為【比對目標】
3. 勾選存在

於此案例中，系統在第 21 行同時找到字串【原始碼】以及【比對項目】，因此觸發警報條件發出告警

◆ 【not 排除字串】

將前面【第 N 組字串】的篩選結果進行二次篩選，【not 排除字串】在有【無勾選存在】時有完全不同的作用與結果，其中【not 排除字串】的設定前後沒有影響其作用，詳情請參考下面

啟用	存在	網頁編號	第一組字串&	第二組字串&	第三組字串	not 排除字串-1
☒	☒	00	viewport			scale

※使用【not 排除字串】前面至少需要有一個【第 N 組字串】
詳情請看後面【not 排除字串邏輯】

※使用【not 排除字串】會有兩種情況

一、 勾選【存在】 / 存在設定字串觸發警報條件

在警報條件已觸發的情況下，系統會對該行字串進行二次審查。如果該行字串包含任何一個【not 排除字串】，該行的警報狀態將被撤銷。經過這一步驟之後，如果仍有任何一行保持警報狀態，系統將自動發出警報。

【not 排除字串】前面必須至少設定一個【第 N 組字串】
因此篩選規則會變成以下

- 1.【第 N 組字串】篩選條件變為【存在設定字串即觸發警報條件】
- 2.在【第 N 組字串】觸發警報條件後，會檢查【not 排除字串】，若該行有【not 排除字串】設定的字串，則該行取消警報條件

例如:

```
33 var createClass = function () {  
34   function defineProperties(target, props) {  
35     for (var i = 0; i < props.length; i++) {  
36       var descriptor = props[i];  
37       descriptor.enumerable = descriptor.enumerable || false;  
38       descriptor.configurable = true;  
39       if ("value" in descriptor) descriptor.writable = true;  
40       Object.defineProperty(target, descriptor.key, descriptor);  
    }
```

設定

1. 第一組字串為【function】
2. 第二組字串為【props】
3. not 排除字串設定為【target】
4. 有勾選【存在】

系統於第 34 行偵測到

- 1.有【function】字串
 - 2.且同時有【props】字串
- 因此觸發警報條件

但因為有設定【not 排除字串】【target】

系統於同行 34 行找到【target】，因此取消該行警報條件

-----若有兩行以上同時擁有【function】以及【props】-----

例如

第 34 行 【function defineProperties(target, props)】

第 88 行 【function defineProperties(test, props)】

第 236 行 【function defineProperties(abc, props)】

若設定為

1. 第一組字串為【function】
2. 第二組字串為【props】
3. not 排除字串-1 設定為【target】
4. not 排除字串-2 設定為【abc】

系統於第 34、88、236 行偵測到同時有【function】以及【props】字串

→因此觸發警報條件

但因為有設定

1. 【not 排除字串-1 target】

系統於同行 34 行找到【target】，因此取消該行警報條件

第 34 行 ~~【function defineProperties(target, props)】~~ →取消警報

第 88 行 【function defineProperties(test, props)】 >發布警報

第 236 行 【function defineProperties(abc, props)】 >發布警報

2. 【not 排除字串-1 abc】

系統於同行 236 行找到【abc】，因此取消該行警報條件

第 34 行 ~~【function defineProperties(target, props)】~~ →取消警報

第 88 行 【function defineProperties(test, props)】 >發布警報

第 236 行 ~~【function defineProperties(abc, props)】~~ →取消警報

結果為兩個被觸發的警報條件被撤銷，然而，在第 88 行的情況下，由於在初步篩選階段已經觸發警報條件，並且沒有【not 排除字串】條件可以解除該警報，**如果仍有任何一行保持警報狀態，系統將自動發出警報**，因此監控系統還是會判定為需要發出警報。

【將其簡單的說】

當系統在網頁原始碼中找到含有我們設定的目標字串的行時，理應發出警報。然而，這些行將首先進入第二階段的篩選。在此階段，如果該行內含有我們設定的【not 排除字串】，那麼該行的警報狀態將被撤消。只有當所有含有目標字串的行都通過了此次篩選（即，所有行的警報狀態都已被撤消）時，系統才會認定為正常狀態。然而，只要有一行含有目標字串的行的警報狀態未被撤消，系統就會發出警報

二、 無勾選【存在】 / 不存在設定字串觸發警報條件

※若您打算使用【無勾選存在】的【not 排除字串】功能，強烈建議您仔細研讀並理解以下狀況的應用。此功能涉及較為複雜的邏輯關係，需要透過深入的思考來正確地設定條件，否則可能會導致預期外的警報觸發。

系統將對【同時存在設定字串】的行列行進行第二次的檢查，如果這些行中的任何一行包含了任一【not 排除字串】，那麼這一行就會被標記為觸發警報的條件。經過這一步驟的二次篩選後，只要【還有任何一行未被標記為觸發警報的行列存在】，系統就會認定為正常狀態，並不會發出警報

系統針對網頁原始碼進行搜尋，對指定的【第 N 組字串】進行偵測

- 如果系統無法在任何行中找到匹配的【第 N 組字串】，警報條件將被觸發，不進行【not 排除字串】第二次篩選。
- 如果系統找到含有【第 N 組字串】的行，則進入第二階段進行【not 排除字串】的檢查。在這一階段，如果該行包含了設定的【not 排除字串】，該行的警報條件將會被觸發

設定【不勾選存在】，會有幾種情況產生

➤ 整個網頁原始碼皆找不到設定之【第 N 組字串】

若整個網頁原始碼中都找不到設定的【第 N 組字串】，這種情況雖然觸發了警報條件，但由於沒有相應的目標字串可供進行二次篩選，【not 排除字串】的篩選機制在此時將無法發揮作用。換言之，系統在確認無法進行二次篩選後，將直接啟動警報通知程序，而不會進行進一步的篩選操作。

例如

1. 第一組字串為【元建資訊】
2. not 排除字串設定為【維修中】
3. 無勾選【存在】

假設網頁原始碼只有三行

第 1 行【descriptor.enumerable = descriptor.enumerable || false】

第 2 行【descriptor.configurable = true】

第 3 行【return function (Constructor, protoProps, staticProps)】

系統搜尋上述原始碼沒有含有【元建資訊】字串，因此觸發警報條件，接著要進行【not 排除字串】二次篩選，但因為原始碼沒有結果目標可以進行第二次【not 排除字串】篩選，因此系統會略過第二次【not 排除字串篩選】直接發送告警

➤ 若設定的字串為兩組(含)以上

【案例一】

假如原始碼為

第 1 行【descriptor.enumerable = descriptor.元建資訊 || 監控系統】

第 2 行【descriptor.監控系統 = 】

第 3 行【return function (元建資訊, 監控系統, staticProps)】

設定

第一組字串為【元建資訊】

第二組字串為【監控系統】

not 排除字串設定為【維修中】

無勾選【存在】

系統搜尋上述原始碼找到【第一行】以及【第三行】同時擁有設定的字串【元建資訊】以及【監控系統】，視為正常不觸發警報條件，接著系統進入第二階段【not 排除字串】檢查，發現第一行及第三行皆無【not 排除字串】【維修中】，因此這兩行也不觸發警報條件，因此此狀況系統不發出告警。

【案例二】

假如原始碼為

第 1 行【descriptor.維修中 = descriptor.元建資訊 || 監控系統】

第 2 行【descriptor.監控系統 = 】

第 3 行【return function (元建資訊, function, 監控系統)】

設定

第一組字串為【元建資訊】

第二組字串為【監控系統】

not 排除字串設定為【維修中】

無勾選【存在】

系統搜尋上述原始碼找到【第一行】以及【第三行】同時擁有設定的字串【元建資訊】以及【監控系統】，視為正常不觸發警報條件，接著系統進入第二階段【not 排除字串檢查】，發現第一行有【not 排除字串】【維修中】，因此第一行觸發警報條件，而第三行沒有【維修中】因此該行不觸發警報條件，只要還有任何一行未被標記為觸發警報的行列存在，系統就會認定為正常狀態，並不會發出警報

➤ 若設定的 **not** 排除字串為兩組(含)以上

【案例一】

假如原始碼為

第 1 行【descriptor.維修中 = descriptor.元建資訊 || 監控系統】

第 2 行【descriptor.監控系統 = 】

第 3 行【return function (元建資訊, function, 監控系統)】

設定

第一組字串為【元建資訊】

第二組字串為【監控系統】

not 排除字串-1 設定為【維修中】

not 排除字串-2 設定為【function】

無勾選【存在】

在系統進行原始碼掃描時，【第一行】和【第三行】都被認定為正常，因為它們同時包含了設定的字串【元建資訊】與【監控系統】，因此在此階段並不會觸發警報。然而，進入第二階段的【not 排除字串檢查】後，情況有所變化：第一行因包含【not 排除字串】之【維修中】而觸發警報條件，第三行則因含有【排除字串】之【function】而同樣觸發警報條件。此時，因為所有符合【同時存在設定字串】的行列皆觸發警報，系統判斷整體情況需要發出警報。

【案例二】

假設

第 1 行【descriptor. = 維修中 descriptor.元建資訊 || 監控系統】

第 2 行【descriptor.監控系統 = 維修中】

第 3 行【return function (元建資訊, function, 監控系統)】

第 4 行【meta name="keywords" content="Watchdog 資訊維運監控系統"】

第 5 行【<div 暫停使用 class="元建資訊 wweb_head_func 監控系統">】

設定

第一組字串為【元建資訊】

第二組字串為【監控系統】

not 排除字串-1 設定為【維修中】

not 排除字串-2 設定為【暫停使用】

無勾選【存在】

系統搜尋上述原始碼找到【第一行】、【第三行】以及【第五行】同時擁有設定的字串【元建資訊】以及【監控系統】，視為正常不觸發警報條件，接著系統進入第二階段【not 排除字串檢查】

發現

【第一行】

第 1 行【descriptor. = 維修中 descriptor.元建資訊 || 監控系統】

發現有【not 排除字串】【維修中】

【第五行】

第 5 行【<div 暫停使用 class="元建資訊 wweb_head_func 監控系統">】

發現有【not 排除字串】【暫停使用】

因此第一行及第五行由於【not 排除字串】而觸發警報條件，但第三行沒有【維修中】或【暫停使用】因此不觸發警報條件，但只要還有任何一行未被標記為觸發警報的行列存在，系統就會認定為正常狀態，並不會發出警報

【將其簡單的說】

當系統在網頁原始碼中找不到我們設定的目標字串時，就會直接發出警報。然而，如果找到了含有我們設定的目標字串的行，系統則會進一步檢查這些行。在此階段，若該行還包含【Not 排除字串】，則該行將被標記為觸發警報。只有在所有包含目標字串的行都被標記為觸發警報時，系統才會實際發出警報。換句話說，只要有一行包含目標字串且未被標記為觸發警報，系統就會視其為正常情況，並不會發出警報。

➤ 整合例子

第 1 行【<meta name="description" content="Watchdog 資訊維運監控系統" />】

第 2 行【<link href="/wweb/css/wweb_man.css" rel="stylesheet" type="text/css" />】

第 3 行【<div style="clear:both"></div>】

第 4 行【<div style="text-align:center">】

第 5 行【<div class="wweb_had_man" style="background-image】

以上為網頁原始碼

以下為【原碼比對】設定

序號	啟用	存在	網頁編號	第一組字串&	第二組字串&	第三組字串	not 排除字串-1
00	☒	☐	00	style	clear		
01	☒	☒	00	style	testests		
02	☒	☒	00	style	clear		both
03	☒	☐	00	style	text		center
04	☒	☐	00	style	text		type
05	☒	☐	00	description	name		資訊維運

系統會依序檢查【序號 00】至【序號 05】的設定欄

確認沒有警報才會逐步往下確認，若檢查到【序號 02】有觸發警報，則不會繼續往下檢查【序號 03】以及往後的設定規則

◆ 【序號 00】

設定規則：若【沒有同時存在】【style】及【clear】的字串則發出告警

搜尋結果：系統於第 3 行找到【style】及【clear】因此沒有觸發警報條件，繼續往下檢查【序號 01】

◆ 【序號 01】

設定規則：若【同時存在】【style】及【testests】的字串則發出告警

搜尋結果：系統找不到任何一行同時存在【style】及【testests】因此沒有觸發警報條件，繼續往下檢查【序號 02】

◆ 【序號 02】

設定規則：當找到包含【style】和【clear】的行時，系統將標記該行為警報。然後，系統將在這些被標記的行中進行第二階段的檢查，尋找【both】字串。若找到，則取消該行的警報狀態。

搜尋結果：系統在第 3 行找到了【style】和【clear】，並標記該行為警報。接著，系統在同一行中找到了【both】字串，因此取消了該行的警報狀態。最後，由於沒有其他同時包含【style】和【clear】且警報狀態未被取消的行，所以系統將狀態判定為正常，繼續往下檢查【序號 03】。

※【勾選存在】所觸發的警報條件，只要還有一條警報條件沒有被【not 排除字串】取消警報，即會發送告警

◆ 【序號 03】

設定規則：若在原始碼中找不到同時存在的【style】和【text】字串，系統將直接觸發警報。然而，如果找到這樣的字串，系統將進行第二階段的篩選。如果在該行找到【center】字串，該行就會被標記為警報。只有當所有包含【style】和【text】字串的行都被標記為警報時，系統才會觸發警報。如果在這些行中有任何一行沒有被標記為警報，則系統會視其為正常。

搜尋結果：系統在第 2 行和第 4 行找到了同時包含【style】和【text】字串的行。然後，它在這兩行中進行了第二階段的檢查，結果發現第 4 行包含了【center】字串，因此該行被標記為警報。然而，在第 2 行中，系統並未找到【center】字串，因此該行被視為正常。因此，儘管第 4 行被標記為警報，但由於第 2 行被視為正常，所以整體系統沒有觸發警報。

※【無勾選存在】的設定中，但只要還有任何一行未被標記為觸發警報的行列存在，系統就會認定為正常狀態，並不會發出警報

因此系統判定為正常，繼續往下檢查【序號 04】

◆ 【序號 04】

設定規則：與【序號 03】一樣

如果在原始碼中我們無法找到【style】和【text】這兩個字串同時存在，那麼系統會立即發出警報。但如果找到了這兩個字串，系統將進行進一步的檢查：如果在這樣的行中也發現了【type】字串，則該行會被標記為警報。但只有當所有包含【style】和【text】的行都被標記為警報時，系統才會真的發出警報。如果任何一行未被標記為警報，系統將認定為正常。

搜尋結果：系統在第 2 行和第 4 行找到了【style】和【text】字串同時存在。系統進一步檢查這兩行，發現第 2 行包含【type】字串，因此將該行標記為警報。但是，在第 4 行中，系統並未找到【type】字串，所以該行被認定為正常。因此，即使第 2 行被標記為警報，由於第 4 行被認定為正常，系統總體並未觸發警報。

※【無勾選存在】的設定中，但只要還有任何一行未被標記為觸發警報的行列存在，系統就會認定為正常狀態，並不會發出警報

因此系統判定為正常，繼續往下檢查【序號 05】

※請注意，【序號 03】和【序號 04】都是各自獨立的篩選過程，這意味著它們不會互相影響。所以，即使在【序號 03】的篩選過程中，第 4 行由於包含【center】字串而被標記為警報，這一結果也只會存在於【序號 03】的篩選結果中。

當我們進行到【序號 04】的篩選時，我們將使用【序號 04】的設定重新進行篩選，也就是說，即使在【序號 03】的篩選過程中第 4 行被標記為警報，這一結果並不會在【序號 04】的篩選過程中被納入考慮。每次篩選都是新的開始，並以該次設定的條件進行檢查。

※若於此案例要觸發警報請合併條件進行設定，如

序號	啟用	存在	網頁編號	第一組字串&	第二組字串&	第三組字串	not 排除字串-1	排除字串-2
00	☒	☐	00	style	clear			
01	☒	☒	00	style	testests			
02	☒	☒	00	style	clear		both	
03	☒	☐	00	style	text		center	type

◆ 【序號 05】

設定規則：如果在原始碼中找不到【description】和【name】兩字串共存的情況，警報會被觸發。然而，如果兩字串共存，系統會進一步檢查該行是否包含【資訊維運】這個字串，如果有，那麼該行就會被標記為警報。只有在所有含有【description】和【name】字串的行都不包含【資訊維運】時，系統才會認定為正常。

搜尋結果：當原始碼中沒有【description】和【name】這兩個字串同時存在時，系統會觸發警報。然而，在第一行，系統發現了這兩個字串。由於設定中包含【資訊維運】這個【not 排除字串】，系統開始檢查第一行是否含有此字串。結果發現第一行含有【資訊維運】，於是該行被標記為警報。由於沒有其他行同時包含【description】和【name】，所以系統判斷為異常並觸發警報。

且在網站的查看頁面會顯示警報資訊

回應時間(sec)	網頁資料量(Bytes)	訊息碼	
0.145	6641	0 page (description+name) compare error	
1.000	354427	200	
0.350	259604	200	
0.330	539060	200	

說明是【description】及【name】的字串的查找發生了問題

【not 排除字串邏輯】

為什麼使用【not 排除字串】必須前面加一個【第 N 組字串】

因為【not 排除字串】的使用存在一些前置條件。具體來說，【not 排除字串】的功能設計主要為了提供一個後續步驟，用以對【第 N 組字串】的初步篩選結果進行再驗證。其目的是驗證這個告警是否真的需要發送出去，或是該將某個本來被視為正常的狀態轉變為警報狀態。

舉兩個例子來說明

【狀況一：勾選存在 / 即為存在設定字串觸發警報條件】

當我們設定【第一組字串】為【test】，並將【not 排除字串】設定為【rooty】，同時【勾選存在】的選項。也就是當【test】字串存在時，系統會發出警報。然而，在警報發出之前，系統會進行第二階段的檢查，看看該行是否也存在另一特定的字串（例如【rooty】）。如果這個【not 排除字串】也存在，那麼該行的警報會被取消。因為【rooty】的存在可能使【test】的存在變得不那麼重要。但如果多行同時觸發警報，只要還有一行沒有因為【not 排除字串】的存在而取消警報，系統就會發出警報。

【狀況二：不勾選存在 / 即為不存在設定字串觸發警報條件】

當我們設定【第一組字串】為【test】，並將【not 排除字串】設定為【rooty】，同時【不勾選存在】的選項。在這種情況下，如果【test】字串不存在，則被視為異常狀態，換句話說，【test】字串的存在被視為正常狀態。然而，當系統找到【test】字串後，在確認該行為正常狀態之前，系統會再次檢查該行是否存在【rooty】這個【not 排除字串】。如果存在，則該行的正常狀態會被轉為警報狀態。

然而，如果有多行符合【test】字串的條件，經過第二次的【not 排除字串】檢查後，只要還有一行沒有觸發【rooty】的警報，系統還是會將整體狀態判定為正常，因此不會發出警報。

由於以上的原因，我們在使用【not 排除字串】時，至少設定一組【第 N 組字串】，這樣的設定才能確保【not 排除字串】的功能進行二次篩選