

CA Spectrum®

監看使用者指南

9.3 版



本文件包含內嵌說明系統與文件 (以下稱爲「文件」) 僅供您參考之用，且 CA 得隨時予以變更或撤銷。

未經 CA 事先書面同意，任何人不得對本「文件」之任何部份或全部內容進行影印、傳閱、再製、公開、修改或複製。此「文件」爲 CA 之機密與專屬資訊，您不得予以洩漏或用於任何其他用途，除非 (i) 您與 CA 已另立協議管理與本「文件」相關之 CA 軟體之使用；或 (ii) 與 CA 另立保密協議同意使用之用途。

即便上述，若您爲「文件」中所列軟體產品之授權使用者，則可列印或提供合理份數之「文件」複本，供您以及您的員工內部用於與該軟體相關之用途，但每份再製複本均須附上所有 CA 的版權聲明與說明。

列印或提供「文件」複本之權利僅限於軟體的相關授權有效期間。如果該授權因任何原因而終止，您有責任向 CA 以書面證明該「文件」的所有複本與部份複本均已經交還 CA 或銷毀。

在相關法律許可的情況下，CA 係依「現狀」提供本文件且不做任何形式之保證，其包括但不限於任何針對商品適銷性、適用於特定目的或不侵權的暗示保證。在任何情況下，CA 對於您或任何第三方由於使用本文件而引起的直接、間接損失或傷害，其包括但不限於利潤損失、投資損失、業務中斷、商譽損失或資料遺失，即使 CA 已被明確告知此類損失或損害的可能性，CA 均毋須負責。

「文件」中提及之任何軟體產品的使用均須遵守相關授權協議之規定，本聲明中任何條款均不得將其修改之。

此「文件」的製造商爲 CA。

僅授與「有限權利」。美國政府對其之使用、複製或公開皆受 FAR 條款 12.212，52.227-14 與 52.227-19(c)(1) - (2) 與 DFARS 條款 252.227-7014(b)(3) 中所設之相關條款或其後續條約之限制。

Copyright © 2013 CA. All rights reserved. 本文提及的所有商標、商品名稱、服務標章和公司標誌均爲相關公司所有。

CA Technologies 產品參考資料

本指南涉及 CA Spectrum®。

連絡技術支援

如需線上技術協助及完整的地址清單、主要服務時間以及電話號碼，請洽「技術支援」，網址為：<http://www.ca.com/worldwide>。

目錄

第 1 章：使用監看	7
關於監看.....	7
臨界值及監看子檢視.....	8
建立監看.....	9
建立和編輯警報原因說明.....	13
複製警報.....	13
警報指令檔範例.....	14
啓動或停用監看.....	16
從命令列啓動及停用監看.....	17
編輯監看.....	17
可繼承的監看編輯.....	18
複製監看.....	18
刪除監看.....	18
顯示監看資訊.....	19
產生多個監看的報告.....	19
第 2 章：事件管理	21
事件.....	21
配置事件.....	21
事件代碼.....	23
事件格式檔案.....	24
事件變數.....	24
第 3 章：監看運算式公式規則	27
監看運算式.....	27
根詞.....	28
常數.....	34
轉換運算子.....	34
常值的資料類型.....	35
屬性.....	36
執行個體識別碼.....	36
監看定義屬性.....	38
監看目的地屬性.....	38

臨界值參考及重設相容性.....	38
------------------	----

附錄 A： 監看類型範例 **41**

第一個監看案例.....	41
第二個監看案例.....	43
變更時評估監看.....	45
輪詢的臨界值監看.....	46
第一個隨需監看案例.....	47
第二個隨需監看案例.....	49
第三個隨需監看案例.....	51
第四個隨需監看案例.....	53
第一個可用性及測試監看案例.....	54
第二個可用性及測試監看案例.....	55
第三個可用性及測試監看案例.....	57
第四個可用性及測試監看案例.....	58

第 1 章：使用監看

本節包含以下主題：

[關於監看](#) (位於 p. 7)

[臨界值及監看子檢視](#) (位於 p. 8)

[建立監看](#) (位於 p. 9)

[啓動或停用監看](#) (位於 p. 16)

[編輯監看](#) (位於 p. 17)

關於監看

*監看*是一種為模型屬性新增臨界值的機制。監看可讓您以高度細節監控網路元素，例如路由器。監看也提供可搭配其他 CA Spectrum 工具進行網路分析的最新資料。

監看管理包含兩個主要元件：SpectroSERVER 中執行監控功能的智慧型電路，以及 OneClick 管理介面。從模型的 [資訊] 索引標籤上的 [臨界值及監看] 檢視即可進行監看管理。

您可以對於任何類型的屬性動態套用監看，也可以根據產生事件和警報的臨界值監控屬性。您也可以將監看的內容複製到其他監看。如此一來，您可以既保留第一個監看的資訊，同時又將新的資訊增加到第二個監看中。

您可以對任何模型類型的內部及外部屬性設定監看。您也可以對屬性設定多個監看。例如，您可以對裝置設定兩個封包速率臨界值監看。一個臨界值在值超過 10,000 時產生黃色警報，另一個臨界值則對超過 15,000 的值產生紅色警報。

附註：建議您先熟悉 CA Spectrum 管理作業後再設定監看。如需詳細資訊，請參閱《*資料庫管理指南*》、《*管理指南*》及《*SpectroSERVER 效能管理指南*》。熟悉 OneClick 介面也有幫助。

監看功能可讓您執行下列工作：

- 對任何類型的屬性動態套用監看。
- 根據臨界值監控屬性及產生事件和警報。

- 違反監看或監看重設時執行指令檔。
- 產生監看報告。

附註：在一個 CA Spectrum 版本中建立的監看可移轉到較新的版本。

臨界值及監看子檢視

您可以在 OneClick 中建立、配置和管理監看。從 [臨界值及監看] 子檢視中的表格，即可檢視和配置監看。

附註：您可以從模型的 [資訊] 索引標籤存取 [臨界值及監看] 子檢視。

[監看] 表格會顯示該模型上定義的各個監看的資訊。[監看狀態] 欄以色彩代碼顯示監看狀況，如下所示：

灰色

指出監看沒有在使用。監看尚未啟動，因此目前並未執行。

藍色

指出監看的初始狀態。監看已啟動，但是尚未第一次執行。

綠色

指出監看已啟動且執行中，未發生任何違規。

黃色

指出已違反監看臨界值。

紅色

指出監看無法評估。會有文字說明原因。

工具列按鈕可讓您執行下列工作：

- 啟動
- 停用
- 建立
- 編輯
- 複製
- 刪除
- 顯示監看資訊


- 列印監看資訊
- 匯出 [監看] 表格

建立監看

您可以建立監看來監控網路模型的所選屬性。

請依循下列步驟：

1. 按一下模型的 [資訊] 索引標籤，並展開 [臨界值及監看] 子檢視。
2. 展開 [監看] 子檢視。
[監看] 表格隨即出現。

3. 在 [監看] 子檢視中，按一下工具列上的  (建立新監看)。
[建立監看] 對話方塊會隨即開啓。

附註：會預設選取 [運算式] 索引標籤。

4. 輸入新監看的名稱，並從 [資料類型] 清單選取資料類型。

附註：此資料類型是監看目的地屬性。監看運算式必須評估為此資料類型。

5. 結合運算子、函數及模型類型屬性，建立監看運算式：
 - 按一下運算式區域左邊的按鈕，將運算子或函數插入游標處的監看運算式。
 - 按一下 [屬性] 按鈕，並選取所要的屬性來插入屬性。

附註：將 `PollingStatus` 屬性設定為 `True` 可啓動監看。

6. 按一下 [內容] 索引標籤，並指定下列設定：

預設為啓動

指定所有繼承監看的模型，其監看是否預設為啓動。如果未設定此選項，請以手動方式啓動所需模型的監看。

重要！ 為輪詢的監看設定 [預設為啓動] 會對 SpectroSERVER 的效能造成不利影響。

隨需評估

只在讀取監看屬性時評估監看運算式。

變更時評估

在監看運算式的屬性有變更時評估監看運算式。屬性必須設定「記憶體」或「資料庫」旗標。

依輪詢評估

在各個輪詢問隔後評估監看。

輪詢間隔

指定輪詢頻率 (以秒為單位)。只有在選取 [依輪詢評估] 時，才會啓用此欄位。

附註：如果將輪詢問隔設定為 0，監看不會啓動。

要建立監看的模型類型 *模型類型*

指定要建立監看的模型類型。按一下 [瀏覽] 選取不同的上層模型類型。

附註：依預設會對所選模型的模型類型建立監看。不過，您可以為監看選取不同的上層模型類型，以便讓其他衍生出來的模型類型繼承監看。

成為可繼承

讓監看可供其模型類型衍生自所選模型類型的模型使用。

7. 如果要對監看附加臨界值，請按一下 [臨界值] 索引標籤，並選取 [附加臨界值] 核取方塊。
隨即會啓用 [比較]、[通知] 及 [指令檔] 選項。
8. 指定下列 [比較] 選項。[比較] 選項所指定的臨界值設定會指出監看通知的調節界限或限制：

臨界值違規狀況值

指定臨界值違規的運算子。例如，選取「大於」。如果範例傳回的值超過臨界值屬性中的值，將視為違反臨界值。對於臨界值違規值，請指定常數，或按一下 [屬性] 選取屬性。

臨界值重設狀況值

決定何時將 [已違規] 的臨界值狀態重設為 [正常]。對於臨界值違規後的各個後續範例，將會使用比較根詞以相反方向比較此重設值與監看運算式。也就是說，如果比較根詞為 \geq (大於或等於)，一旦範例傳回低於此重設值的值，狀態就會重設為 [正常]。當 [臨界值比較] 重設為 $=$ 或 \neq 時，重設值並不適用，因為如果該值大於或小於於監看運算式 (不論方向為何)，都會將監看視為違規。在此情況下，此選項會停用；如果沒有停用，就必須加以輸入。對於臨界值重設值，請指定常數，或按一下 [屬性] 按鈕選取屬性。

9. 指定下列 [通知] 選項。[通知] 選項可讓您指定違反臨界值時所傳送之通知的類型。

無通知

避免在違反臨界值時產生通知。

產生事件

指定 CA Spectrum 事件管理系統在違反或重設監看時產生事件。選取此選項將啟用事件代碼的欄位，如下所示：

- 當違反臨界值時 — 指定違反臨界值時產生的事件。
- 當臨界值重設時 — 指定重設臨界值時產生的事件。
- 當停用/所有執行個體重設時 — 指定重設此監看的所有執行個體或停用違規的監看時產生的事件。

您可以進一步配置這些事件來產生或清除警報，並參與事件規則和程序。並非所有欄位都需要事件；只有包含有效事件代碼的欄位，才會產生事件。

產生警報

指定在違反監看臨界值時直接產生警報。另外也會產生與警報相關聯的一般事件 (0x480004)。如果監控經過重設，並清除警報，將會產生一般重設事件 (0x480005)。選取此選項將啟用下列欄位：

- 嚴重性 — 指定 [輕微]、[嚴重] 或 [重大] 的警報嚴重性。
- 說明 — 指出警報原因說明。依預設會提供一般說明。[瀏覽] 按鈕可讓您選取不同的原因說明或建立新的原因說明。新建立的說明儲存在 OneClick 伺服器的 `$INSTALL/custom/Events/CsPCause` 目錄中。如果您有多個 OneClick 伺服器，請以手動方式複製檔案。

- 可清除的使用者 — 設定時，可讓您手動清除監看警報。
- 使用者清除警報時重設監看 — 如果手動清除警報，即使監看尚未達到重設值，仍然重設監看。後續違規將再次產生警報。只有在選取 [可清除的使用者] 時，才可使用此選項。

10. 指定下列 [指令檔] 選項。 [指令檔] 選項可讓您指定違反或清除監看臨界值時將執行的指令檔。

達到臨界值時執行指令檔

啓用指令檔執行。選取會觸發指令檔執行的臨界值條件。

對每一個執行個體執行

如果已啓用，而且監看運算式包含清單屬性，則會對符合臨界值條件的各個執行個體執行指令檔。

指令檔檔案

指定 SpectroSERVER 上要執行的指令檔檔案。如果未指定目錄路徑，將使用預設目錄 `<$SPECROOT>/SS-Tools/SwScript`。您可以在檔案 `<$SPECROOT>/SS/.watchrc` 中設定 `sw_script_path`，變更預設目錄。

附註：指令檔將由啓動 SpectroSERVER 的使用者執行。因此，使用者需要指令檔的存取和執行權限。如果在建立或修改監看時偵測到權限問題，會出現錯誤訊息。如果稍後變更權限，而且在嘗試執行指令檔時偵測到問題，則會產生事件。

11. 按一下 [範圍] 按鈕，指定監看的目的地範圍。在 OneClick 環境中，您可以將監看散佈到分散式伺服器環境中的多個範圍。依預設會在所有範圍中建立新的監看。

對話方塊會隨即開啓，顯示範圍的清單。

12. 將不要建立監看的範圍移到右邊。

附註：建立監看時如果沒有範圍可用，您可以稍後再將監看新增至該範圍。使用 [\[編輯監看\] 對話方塊](#) (位於 p. 17)，並按一下 [範圍]。

13. 按一下 [確定]。

隨即會建立監看。




更多資訊：

[屬性](#) (位於 p. 36)

建立和編輯警報原因說明

對於因為臨界值違規所產生的警報，您可以建立和編輯原因說明。臨界值違規的通知是透過 CA Spectrum 事件及警報設施傳遞。這些功能支援使用外部指令檔介面，透過電話、呼叫器或電子郵件傳遞通知。



請依循下列步驟：

1. 選取監看，並按一下  (編輯)。
監看便會在 [編輯監看] 對話方塊中開啓。
2. 選取 [臨界值] 索引標籤，並選取 [產生警報]。
隨即會顯示警報選項。
3. 按一下 [瀏覽]。
[選取警報原因代碼] 對話方塊會隨即開啓。底端有警報詳細資料。
4. 按一下  (建立) 建立新的警報原因，或從清單中選取原因，然後按一下  (編輯) 編輯原因。
隨即有對話方塊開啓，供您建立或編輯警報原因。
附註： [原因代碼] 是唯讀欄位。
5. 必要時在欄位中輸入資訊，並按一下 [確定]。
便會建立或修改警報原因說明。

複製警報

使用 [複製] 功能，可建立多個彼此略有不同的警報原因說明。

請依循下列步驟：

1. 選取 [監看] 表格中的監看，並按一下  (編輯)。
監看便會在 [編輯監看] 對話方塊中開啓。
2. 選取 [臨界值] 索引標籤上的 [產生警報]，並按一下 [瀏覽]。
[選取警報原因代碼] 對話方塊會隨即開啓。
3. 從清單中選取警報原因說明，並按一下  (複製)。
[複製警報原因] 對話方塊會隨即開啓。

4. 編輯原因代碼。
5. (選擇性) 編輯其他警報內容。
6. 按一下 [確定]。
隨即建立新的警報原因說明。

警報指令檔範例

以下是您可建立作為通知的 UNIX 指令檔範例。您可以指定對於臨界值監看設定或清除警報時所要執行的指令檔。此範例指令檔的名稱為 *sw_alm_script*，包含在安裝 CA Spectrum 時的 *SS-Tools/SwScript* 目錄中。

```
#!/bin/sh
#####
這個範例指令檔可供使用者用來對監看的臨界值違規（監看狀態為 #VIOLATED）執行重要的工作，或在已經違規的監看變成正常（監看狀態為 #NORMAL）時執行重要的工作。
#
# 在以上的各個案例中，將會執行使用者指定的指令檔（在建立監看時指定），此指令檔提供 20 個引數。這些引數以遞增排序顯示如下：
#
# 1) DATE - 監看違規的日期。
# 2) TIME - 監看違規的時間。
# 3) MTYPE_NAME - 監看的模型類型。
# 4) MODEL_HANDLE - 模型的模型控制代碼。
# 5) MODEL_NAME - 監看的模型。
# 6) INSTANCE - 其監看違規或重設的執行個體（連接埠、電路板等）。
# 7) ALARM_ID - 警報的 ID。
# 8) CONDITION - 警報的條件。
# 9) CAUSE_CODE - 警報的原因代碼。
# 10) REPAIR_PERSON - 獲派進行疑難排解的修復人員。
# 11) ALARM_STATUS - 警報的狀態。
# 12) SCRIPT_TYPE - 如果是在違規時執行指令檔，這會設為「VIOLATED」，如果是在重設時執行指令檔，這會設為「NORMAL」。
# 13) WATCH_NAME - 對其執行指令檔之監看的名稱。
# 14) WATCH_CREATOR - 監看的建立者名稱。
# 15) WATCH_SRC - 監看運算式的公式。
# 16) WATCH_SRC_VAL - 監看運算式公式的值。
# 17) WATCH_REF - 臨界值參考的公式。
# 18) WATCH_REF_VAL - 臨界值參考的值。
# 19) WATCH_RES - 臨界值重設的公式。
# 20) WATCH_RES_VAL - 臨界值重設的值。
#####
```



```
## 將引數值儲存於日後可使用的變數。
## 請記住，DATE 是第一個引數，WATCH_RES_VAL 是最後一個引數。
# 重要附註：
## 請特別留意變數「SCRIPT_TYPE」。如果監看出現違規狀態，SCRIPT_TYPE 會設為「VIOLATED」。如果監看出現正常狀態，SCRIPT_TYPE 會設為「NORMAL」。您可以據此來決定如何處理監看違規和監看正常的情況。
while [ "$#" -ne 0 ]
do
case "$#"
in
20) DATE=`echo "$1"`;;
19) TIME=`echo "$1"`;;
18) MTYPE_NAME=`echo "$1"`;;
17) MODEL_HANDLE=`echo "$1"`;;
16) MODEL_NAME=`echo "$1"`;;
15) INSTANCE=`echo "$1"`;;
14) ALARM_ID=`echo "$1"`;;
13) CONDITION=`echo "$1"`;;
12) CAUSE_CODE=`echo "$1"`;;
11) REPAIR_PERSON=`echo "$1"`;;
10) ALARM_STATUS=`echo "$1"`;;
9) SCRIPT_TYPE=`echo "$1"`;;
8) WATCH_NAME=`echo "$1"`;;
7) WATCH_CREATOR=`echo "$1"`;;
6) WATCH_SRC=`echo "$1"`;;
5) WATCH_SRC_VAL=`echo "$1"`;;
4) WATCH_REF=`echo "$1"`;;
3) WATCH_REF_VAL=`echo "$1"`;;
2) WATCH_RES=`echo "$1"`;;
1) WATCH_RES_VAL=`echo "$1"`;;
esac
shift
done
```

```
## 例如，撰寫可用來傳送郵件的訊息。  
## 所撰寫的訊息也可用於其他任何用途。  
message()  
{  
  echo "Watch Status Notification"  
    echo ""  
    echo "Watch Status is $1"  
    echo ""  
  
  echo "Date:                $DATE"  
  echo "Time:                $TIME"  
  echo "MType Name:         $MTYPE_NAME"  
  echo "Model Handle:       $MODEL_HANDLE"  
  echo "Model Name:         $MODEL_NAME"  
  echo "Instance:          $INSTANCE"  
  echo "Alarm ID:           $ALARM_ID"  
  echo "Condition:          $CONDITION"  
  echo "Cause Code:         $CAUSE_CODE"  
  echo "Repair Person:      $REPAIR_PERSON"  
  echo "Alarm Status:       $ALARM_STATUS"  
  echo "Watch Name:         $WATCH_NAME"  
  echo "Watch Creator:      $WATCH_CREATOR"  
  echo "Watch Expression:   $WATCH_SRC"  
  echo "Watch Expression Value: $WATCH_SRC_VAL"  
  echo "Watch Reference:    $WATCH_REF"  
  echo "Watch Reference Value: $WATCH_REF_VAL"  
  echo "Watch Reset:        $WATCH_RES"  
  echo "Watch Reset Value: $WATCH_RES_VAL"}  
## 將撰寫的訊息寄送給相關的使用者。  
## <USER LIST> 是郵件收件者的清單。  
message "$SCRIPT_TYPE" | mail <USER LIST>
```

啓動或停用監看

監看會預設為啓動還是停用，視其建立方式而定。

可新增到系統的監看並無數量或格式的限制。如此的彈性也表示，若不小心使用，將會耗盡系統資源，而導致效能低落。即使單一個監看只會使用最少的平台資源，並產生極少的網路流量，仍建議您停用或刪除不必要的監看。另外，請您謹慎設定參考大量裝置屬性的監看。

若要啓動或停用單一模型的監看，請從 [監看] 表格選取監看，並按一下  (啓動監看) 或  (停用監看)。您也可以選取多個監看來一次啓動或停用。

您可以使用「屬性編輯器」，啓動或停用多個模型上的多個監看。選取監看屬性並將其移到「屬性編輯器」的右側面板時，會啓用 [啓動監看] 及 [停用監看] 選項。

附註：如需「屬性編輯器」的詳細資訊，請參閱《*模型化和管理 IT 基礎架構管理指南*》。

從命令列啓動及停用監看

除了 OneClick 主控台之外，您也可以從命令列啓動或停用監看，並檢視資料設定檔。

下列命令列出模型類型適用的監看資料：

```
show watch [mh=model_handle] [lh=landscape_handle]
```

下列命令更新監看狀態：


```
update action=action_code [watch=watch_id] [mh=modelhandle]
```

附註：[watch=watch_id] 引數僅適用於啓動 (0x00480003) 和停用 (0x00480004) 動作。

編輯監看

您可以編輯監看，以變更其運算式、內容及臨界值。

請依循下列步驟：

1. 按一下模型的 [資訊] 索引標籤，並展開 [臨界值及監看] 子檢視。
2. 展開 [監看] 子檢視。
3. 選取 [監看] 表格中的監看，並按一下  (編輯)。
監看便會在 [編輯監看] 對話方塊中開啓。
4. 必要時變更監看運算式、內容及臨界值，並按一下 [確定]。

附註：如需監看運算式、內容及臨界值的詳細資訊，請參閱[建立監看](#) (位於 p. 9)。

隨即會編輯監看。

可繼承的監看編輯

編輯可繼承的監看時，預設會修改所選模型類型的監看定義。您可以為監看選取不同的上層模型類型。請使用 [內容] 索引標籤的 [模型類型] 區段中出現的 [瀏覽] 按鈕。


如果對衍生自監看原始模型類型的模型類型修改監看定義，則變更只會對該類型及其衍生類型的模型生效。監看的原始模型類型不會變更。

例如，監看原先是在 Gen_IF_Port 上建立，而且可繼承。Serial_IF_Port 衍生自 Gen_IF_Port，因此 Serial_IF_Port 模型會繼承該監看。如果在 Gen_IF_Port 模型類型修改監看定義，則變更會傳播至 Serial_IF_Port 模型。不過，如果在 Serial_IF_Port 修改監看定義，新版定義會覆寫 Gen_IF_Port 上的定義。只有 Serial_IF_Port 模型會受到影響，Gen_IF_Port 模型則會保留原始的監看定義。由於繼承規則的緣故，您可以分別重新定義或覆寫衍生模型類型的可繼承監看。

複製監看

使用 [複製] 功能，可建立多個彼此略有不同的監看。

請依循下列步驟：

1. 選取 [監看] 表格中的監看，並按一下  (複製監看)。
監看會隨即開啓。
2. 輸入新監看的名稱、視需要變更其他任何資訊，然後按一下 [確定]。
隨即會複製監看並建立新的監看。

刪除監看

如果不再需要相關的臨界值監控，可刪除監看。

請依循下列步驟：

1. 選取 [監看] 表格中的監看，並按一下  (刪除)。


如果監看存在於多個範圍，會有對話方塊開啓。左邊顯示監看存在的範圍及將刪除監看的範圍。

2. (選擇性) 若要保留某範圍的監看，請將該範圍移到右邊。
3. 按一下 [確定]。
便會從左邊列出的範圍刪除監看。

顯示監看資訊

您可以顯示監看資訊，以檢視監看報告。您也可以列印這些資訊，並將資訊匯出為文字檔或 HTML 檔。

請依循下列步驟：

1. 選取 [監看] 表格中的監看，並按一下  (資訊)。
[監看報告] 對話方塊便會顯示監看資訊。
2. (選擇性) 按一下 [列印] 可列印監看資訊。
3. (選擇性) 按一下 [匯出] 可將該資訊儲存為文字檔或 HTML 檔。
便會列印或匯出監看。

產生多個監看的報告

您可以從 [OneClick 管理] 頁面產生多個監看的報告。

請依循下列步驟：

1. 在瀏覽器中開啓 OneClick 首頁，並按一下 [管理] 索引標籤。
[管理頁面] 隨即開啓。
2. 按一下 [監看報告] 連結。
[產生監看報告] 對話方塊隨即開啓。其中顯示 [選取範圍] 清單及 [選取模型類型] 清單。
3. 選取要產生報告的範圍及模型類型，並按一下 [產生報告]。
便會產生報告。

第 2 章：事件管理

本節包含以下主題：

[事件](#) (位於 p. 21)

[配置事件](#) (位於 p. 21)

事件

爲了說明 CA Spectrum 進階事件處理，請設想下列案例。爲了進行測試，您想要暫時允許高子件計數。您想要在高子件計數情形持續一分鐘以上時才產生警報，並且在重設監看時清除任何這類警報。CA Spectrum 可使用 EventPair 事件規則執行下列工作：


- 在 ChildLimit 監看違規時產生初始事件。
- 隱藏警報，除非初始事件發生後的 60 秒內並未緊接著發生重設事件。
- 在重設或停用此監看的所有執行個體時產生其他事件。

附註：如需事件規則及建立事件的詳細資訊，請參閱《*事件配置使用者指南*》。

配置事件

下列程序是事件配置的範例。接續前一個主題中的範例，第一個事件是「初始子件計數過高」(0xffffffff)，第二個事件是「重設」(0xffffffffe)。如果違反子件計數臨界值，而且並未在 60 秒內加以重設，將會產生「子件計數過高」警報 (事件/警報 0xffffffffd)。

請依循下列步驟：

1. 從 [工具] 功能表依序選取 [公用程式]、[事件配置]。
[事件配置] 頁面會顯示 [導覽]、[內容] 及 [詳細資料] 面板。
2. 在 [導覽] 面板中按一下  (建立事件)。
[建立事件] 對話方塊隨即出現，顯示填入預設值的事件代碼。

3. (選擇性) 編輯事件代碼，並輸入事件訊息，然後按一下 [確定]。
隨即建立事件，事件的詳細資料會出現在右邊的 [內容] 及 [詳細資料] 面板中。
4. 建立並儲存初始子件計數過高事件 (0xffffffff)。
此事件會觸發 EventPair 規則，在特定條件下產生警報及重設事件 (0xffffffffe)。監看會使用重設事件評估子件計數條件的持續性，並且在重設監看時清除 EventPair 規則所產生的任何警報。
5. 建立並儲存事件「0xffffffd」，以及子件計數持續相當高時用來產生通知的相對應警報。
6. 對於 60 秒內未收到重設事件 (0xffffffffe) 的情況下產生「0xffffffd」事件及警報的初始「子件計數過高」事件 (0xffffffff)，建立並儲存事件組規則。
重要！ 使用事件規則處理事件時，如果更新 SpectroSERVER，處理作業不會完成。叫用此功能時處理的所有事件規則將予以清空。
7. 依序按一下 [檔案]、[全部儲存]。
隨即會儲存新事件。
8. 開啓監看進行編輯。
9. 在 [臨界值] 索引標籤上，選取 [通知] 下的 [產生事件]。
事件代碼欄位隨即出現。
10. 在 [當違反臨界值時] 欄位中輸入先前建立的初始「子件計數過高」事件 (0xffffffff)。
11. 在 [當臨界值重設時] 欄位中輸入先前建立的警報清除「重設」事件 (0xffffffffe)，並按一下 [確定]。
事件隨即與監看產生關聯。

如果 LAN_802_3 容器包含 5 個 (含) 以上的模型，將產生初始「子件計數過高」事件，並評估事件規則。只有在初始事件發生後的 60 秒內未收到重設事件時，才會產生警報。如果重設監看，將清除監看產生的「子件計數過高」警報。

此範例的 EventDisp 項目 (若需要) 如下：

```
# 確定子件計數過高 (0xffffffff)
# 在 60 秒內重設 (0xffffffe)
# 否則產生警報事件 (0xffffffd)0xffffffff E 50 R Aprisma.EventPair, 0xffffffe,
0xffffffd, 60

# 產生持續子件計數過高的警報 0xffffffd E 50 A 1,0xffffffd

# 如果子件計數低於臨界值，請清除任何現有警報 0xffffffe E 50 C 0xffffffd
```

事件代碼

CA Spectrum 事件記錄中會產生各個事件的事件代碼。

附註： 監看建立作業不會產生事件。

下列清單包含事件代碼及其說明：

0x0048000a

無法啟動模型的監看

0x00480000

修改監看

0x00480001

損毀監看

0x00480002

啟動監看

0x00480003

停用監看

0x00480004

違反監看臨界值

0x00480005

重設監看的臨界值狀態 (具有變數重設值)

0x00480006

監看執行個體的違規

0x00480007

重設監看執行個體的臨界值狀態 (具有變數重設值)

0x00480010

無效的可能原因訊息

0x00480012

記錄間隔衝突

0x00480013

執行指令檔時發生錯誤 (僅對於臨界值監看)

0x00480014

重設模型的監看臨界值，具有原因

事件格式檔案

您可以使用現有的事件格式檔案 Event00480004 及 Event00480005 (位於 `<$SPECROOT>/SG-Support/CsEvFormat`) 作為建立自己的監看相關事件格式檔案所用的範本。

附註：如需詳細資訊，請參閱《*事件配置使用者指南*》。

事件變數

監看臨界值違規 (0x00480004) 及重設 (0x00480005) 的事件有變數繫結。這將套用於直接產生警報的監看及產生事件之監看。清除事件有一個變數繫結 ID #2，其中包含監看名稱。

下表顯示繫結變數及其內容：

變數繫結 ID	內容
2	監看名稱
4	臨界值參考值
5	臨界值重設值 (僅限重設事件)
6	比較字串值
7	計算的監看值

變數繫結 ID	內容
8	如果監看有執行個體，即為執行個體物件識別碼 (OID)，否則為空白
9	內部；完成個別事件訊息的字串
10	內部；完成個別事件訊息的字串
11	內部；完成個別事件訊息的字串

第 3 章： 監看運算式公式規則

本節包含以下主題：

[監看運算式](#) (位於 p. 27)

[屬性](#) (位於 p. 36)

[臨界值參考及重設相容性](#) (位於 p. 38)

監看運算式

監看運算式會定義監看所監控的對象。監看運算式可以是簡單的屬性名稱，也可以是複雜的公式。監看運算式包含透過數學符號及布林值運算子彼此相互關聯的屬性及常數值，這些項目稱為根詞。

運算式將按照下列的根詞優先順序由左至右進行評估：

優先順序	根詞
1	.#.
2	()
3	DELTA () COUNTER_DELTA ()
4	!
5	&
6	* /
7	+ -
8	= != >= <= < >

更多資訊：

[屬性](#) (位於 p. 36)

根詞

根詞一般用於有左右兩端的運算式。例如，在 $A+B$ 中， $+$ 是定義左邊的 A 和右邊的 B 兩者之間關係的根詞。括弧用於指出有多個組成部份的運算式進行評估時的順序。

可用的根詞如下：

- $+$
- $-$
- $*$
- $/$
- $==$
- $!=$
- $>$
- $<$
- $>=$
- $<=$
- $!$ (not)
- $&$ (and)
- $,$
- $|$
- $($
- $)$
- True
- FALSE
- TIME
- DELTA
- MIN
- MAX
- INTEGER
- REAL
- UNSIGNED

- COUNTER_DELTA
- UNSIGNED64

您可以按照功能及適用性將根詞分組。下列根詞僅適用於文字字串之間或數值運算式中：

- == 等於
- >= 大於或等於
- <= 小於或等於
- != 不等於
- > 大於
- < 小於

附註：這些根詞不支援屬性類型「物件 ID」、「IP 位址」及「布林值」。

下列根詞僅適用於數值運算式：

- + 加
- - 減
- * 乘
- / 除

這些根詞不支援屬性類型「文字字串」、「物件 ID」、「IP 位址」及「布林值」。支援的類型將按照下列順序進行評估：

- 實數
- 時間刻度、日期、量測計、計數器
- 列舉整數

如果元素來自兩個不同層級，將按照較高的層級評估元素。例如， $5 + 5$ 經評估為整數，而 $5 + 5.1$ 經評估為實數。

附註：CA Spectrum 模型類型使用屬性類型「計數器」來儲存不帶正負號的長整數。如果將不帶正負號的根詞錯誤用於運算式，例如 "Text String + UNSIGNED(5)"，產生的錯誤訊息屬於「計數器」類型。

下列根詞僅適用於布林值運算式：

- ! logical not
- & and
- | or

下列根詞由文字組成，而非由符號組成：

TIME

表示目前時間，這是格林威治標準時間 1970 年 1 月 1 日 00:00 以來的秒數。

DELTA

計算取樣頻率間隔的屬性變更。例如，名稱爲 int1 且間隔爲 30 秒的整數屬性監看會產生 DELTA 值，如下所示：

間隔	值
0	100, 0
30	1000 900
60	1000 0
90	-1000 -2000
120	2000 3000

DELTA 根詞支援下列屬性類型：

- 整數
- 列舉
- 實數
- 短
- 時間刻度
- 日期
- 量測計
- 計數器

在值絕不會減少的情況下 (例如時間刻度及計數器)，請使用 COUNTER_DELTA。

您可以 DELTA 運算式中的屬性為例，如下所示：

- DELTA (If_In_Octets.2)
- DELTA (If_Out_Octets.#)
- 也支援 DELTA (TIME)

附註：DELTA (TIME) 是目前的時間與監看的最後一次評估時間之間的時間差；僅適用於涉及內部屬性的計算。若要產生涉及外部屬性的速率計算，請使用裝置的時間，例如 DELTA (XfIDELTA (Sys_Up_Time)，其中 Sys_Up_Time 是裝置的時間計數器。

COUNTER_DELTA

計算其值應該是增加且不帶正負號整數之屬性的變更。值必須重設為零，但是此根詞的結果始終都是不帶正負號的正整數。負數屬性值將被視為不帶正負號的大值。

COUNTER_DELTA 根詞支援與 DELTA 相同的屬性值，而且您也可以和針對 DELTA 所做的說明一樣，以 COUNTER_DELTA 運算式中的屬性為例。COUNTER_DELTA (TIME) 也受支援。

ATTR

指出根詞後出現在括號中的十六進位數字是屬性識別碼。如以下範例所示，在出現重複名稱的情況下，此根詞可讓屬性獲得唯一識別：

ATTR (<attr_id>)
ATTR (<attr_id>.<instance id>)
ATTR (<attr_id>.#

<attr_id> 值是「0x」或「0X」後接 1 至 8 個十六進位數字。

True

定義布林值常數 True。

FALSE

定義布林值常數 False。

MAX

識別以逗號區隔且以括號括住的兩個運算式之中較大的值，例如 (attribute x+1, attribute y-5)。這些運算式可能由運算元、屬性或根詞以任意方式組合而成，並且這些組合會產生數值 (非文字、字串、八位元、布林值)。

MIN

識別以逗號區隔且以括號括住的兩個運算式之中較小的值，例如 (attribute x+1, attribute y-5)。這些運算式可能由運算元、屬性或根詞組合而成，並且這些組合會產生數值 (非文字、字串、八位元、布林值)。

您可以使用下列根詞 (未出現在快顯選取功能表中) 指出監看所監控的清單屬性執行個體。

。

指出後續項目會識別清單屬性的特定執行個體。

例如，If_In_Octets.2 會指定屬性 If_In_Octets 的第二個執行個體。對於 IP 位址表而言，識別碼可能是整個位址，而非單一數字。

如果指定的屬性執行個體不存在，即使對於已啟動監看的其中一個模型也是如此，監看將會失敗。在某些模型存在執行個體而其他模型不存在執行個體的情況下，存在執行個體的模型，其監看將會成功，不存在執行個體的模型，其監看則會失敗。

.#

指出「監看詳細資料檢視」中顯示的目前「執行個體指定元」值，會決定要監控哪些執行個體。

例如，如果「執行個體指定元」值是 ALL，則運算式 If_In_Octets.# 會將監看套用至 If_In_Octets 屬性的所有執行個體。不過，如果「執行個體指定元」值是 RANGE (1-3)，則只會監控執行個體 1、2 及 3。

在指定執行個體範圍的情況下，即使範圍中的一個執行個體未出現在其中一個模型中，監看仍將失敗。相反地，範圍中的所有執行個體都有出現的任何模型，監看將會成功。

資料類型

下表顯示可指派到不同屬性類型的資料類型值。

附註：雖然 CA Spectrum 不支援資料類型為 OCTET STRING 的屬性監看，但是您可以使用 TEXT STRING 資料類型監看這類屬性。

運算式結果類型	可接受的目的地屬性類型
BOOLEAN	任何數值類型
TEXT STRING	文字字串
INTEGER	整數、列舉、實數
ENUMERATION	整數、列舉、實數
REAL	整數、列舉、實數
DATE	實數、日期、時間刻度、計數器、量測計、Counter64
TIME TICKS	實數、日期、時間刻度、計數器、量測計、Counter64
COUNTER	實數、日期、時間刻度、計數器、量測計、Counter64
GAUGE	實數、日期、時間刻度、計數器、量測計、Counter64
OBJECT Id	OBJECT Id
IP 位址	IP 位址
COUNTER64	COUNTER64，實數

例如，假設整數屬性的名稱是 Int1，則允許運算式 $\text{Int1} = 50.5$ (需要轉換)，但是不允許 $\text{Int1} = \text{"a string"}$ 。

這些規則是將值指派給布林值和文字字串屬性最彈性的方法。任何運算式均可評估為 0 和 1，而且可寫入布林值屬性。例如，假設 Bool1 是布林值屬性，運算式 `Bool1 = 500 * 50 + 450` 會將 TRUE 指派給 Bool1。運算式 `Bool1 = (500 * 50 + 450) * 0` 會將 FALSE 指派給 Bool1。同樣地，任何運算式均可評估為文字字串。假設文字字串屬性的名稱為 str1，運算式 `str1 = 500 + 50` 會將字串 "550" 指派給 str1。

附註：文字字串屬性類型接受有引號的文字字串，以及沒有引號的算術運算式，例如 `500 + 50`，這個運算式會產生字串 "550"。

常數

與屬性值和運算子一樣，您可以直接在運算式欄位中輸入常數。您可以在運算式公式中輸入下列類型的常數：

- 不帶正負號的整數，以一或多個數字串連一起來表示，具有正值。
- 實數，以零個或多個數字、一個點，以及一或多個數字串連一起來表示。例如，.7、1.7 或 23.24 (但 7 不行)。
- 帶正負號的整數，以一或多個數字串連一起來表示，具有正值或負值。
- 文字字串，以任何字元串連一起並以雙引號括住來表示。例如，"a string"、"5.25" 或 ""。

轉換運算子

轉換會迫使一種資料類型變成另一種資料類型。執行轉換時，所轉換的值可能會失去某個部份。來源資料類型的範圍可放入目的地資料類型的範圍時，則不需要轉換。

例如，您不需要轉換就可以將整數指派給實數，因為 $-1.79769 \text{ e}+308 \leq -2,147,483,648$ 且 $2,147,483,647 \leq 1.79769 \text{ e}+308$ 。不過，將整數指派給計數器時，則必須使用轉換，因為兩者的範圍不重疊。

轉換適用於所涉及的資料類型重疊的範圍。例如，將整數 5 轉換到計數器能夠達到預期的行為效果。不過，負數不在計數器的範圍內。因此，將 -5 轉換到計數器，將由於電腦呈現數字方式的緣故，而產生不帶正負號 (正數) 的 4,294,967,291。相反地，將大於 2,147,483,647 的計數器值轉換到整數，則會產生負數。

您可以使用下列運算子，將括號中的運算式結果轉換為選取的資料類型。

運算子	轉換的結果
UNSIGNED	一般而言，會將 27 當作是帶正負號的整數；不過，“UNSIGNED (27)” 等同於不帶正負號的整數 27。如果負數出現在此運算子之後的運算式中，則會等同於不帶正負號。範例：UNSIGNED (-5) 等同於 5。數字 -5 經 UNSIGNED 後的值為 4294967291。
INTEGER	運算式中接在此運算子後的任何實數將四捨五入為最接近的整數。範例：INTEGER (2.4) 等同於 2，INTEGER (2.6) 等同於 3。
REAL	範例：REAL (3) 等同於 3.0。

常值的資料類型

監看運算式中的常值會與一連串的範圍進行比較，以決定其資料類型。下表的 [類型] 欄是常值的實際資料類型，此類型能夠直接與欄儲存格中列出的其他類型互換。

下表顯示如何決定常值的資料類型：

最小值	最大值	類型
-2,147,483,648	2,147,483,647	整數、布林值*、列舉
0	4,294,967,295	計數器、日期**、量測計、時間刻度
0	18,446,744,073,709,551,615	Counter64
-1.79769 e+308	1.79769 e+308	實數

* 您也可以對布林值資料類型使用常數 TRUE 及 FALSE。任何非零值均等同於 TRUE。

** 從 1970 年 1 月 1 日 00:00:00 UTC(0) 以來的秒數。

屬性

您在監看運算式公式中輸入的屬性必須一律以字母為開頭，並包含字母、數字和底線字元 (_) 的組合。以單引號括住屬性以及與根詞 (TRUE、FALSE、TIME、DELTA、COUNTER_DELTA) 相同的名稱。您也可以使用 ATTR 根詞以屬性 ID 指定屬性。

[監看運算式](#) (位於 p. 27) 可包含單一個屬性。若要建立多重屬性運算式，請從按鈕調色盤選取運算子，然後選取另一個屬性，依此類推。

如果在運算式中指定清單屬性，則必須包含執行個體資訊。若要指定執行個體，請附加「.」，後面接著執行個體 ID 號碼 (例如，ifInOctets.2)。或者，附加「.#」 (例如，ifInOctets.#)，並且在 [執行個體] 欄位中選取 [全部] 或 [範圍]。對於範圍，請在 [開始] 和 [結束] 欄位中指定高低 ID。在 [屬性選取器] 對話方塊的 [類型] 欄中，清單屬性均以「[]」表示。

除了輸入作為監看運算式一部份的屬性之外，您也可以動態建立在儲存監看資訊及目的地屬性時所需的屬性。

執行個體識別碼

在監看運算式公式中使用清單屬性時，您可以使用執行個體識別碼來指定特定物件。

清單屬性需要執行個體 ID 才能完整識別物件。您可以指定特定執行個體、一個範圍的執行個體，或監看運算式中屬性的所有可能執行個體。建立 OID 時，最多可使用兩個不同的子 OID：標準 SpectroSERVER OID 機制或監看執行個體指定元。SpectroSERVER 會從 OID 前置詞及 OID 參考 (選擇性) 建立 OID。在 [運算式] 頁面上指定監看運算式的執行個體時，執行個體會附加到 OID，如下所示：

OID = OID 前置詞 + OID 參考 + 監看執行個體

如果使用「執行個體指定元」指定範圍，則會計算範圍中各個執行個體的結果。如果指定的範圍大於實際的執行個體，則只會計算實際執行個體的結果。也就是說，如果您指定 1 至 10 的範圍，而且只有 3 個執行個體，將只會計算 3 個結果。

例如，您要對路由器設定 If_In_Octets 的監看。您提供 1 至 3 的範圍作為「執行個體指定元」。由於 CA Spectrum 資料庫用於 If_In_Octets 的 OID 前置詞是 1.3.6.1.2.1.2.2.1.10，因此會指派下列 OID：

- OID = 1.3.6.1.2.1.2.2.1.10 + .1
- OID = 1.3.6.1.2.1.2.2.1.10 + .2
- OID = 1.3.6.1.2.1.2.2.1.10 + .3

如果您使用 ALL 作為「執行個體指定元」，將動態決定 If_In_Octets 物件的所有執行個體。您也可以監看運算式中指定執行個體，或者混用物件的特定執行個體與 RANGE 或 ALL 執行個體指定元。使用 Instance_of 根詞（「.」）指定物件的執行個體，將覆寫該物件的執行個體指定元。

例如，在下列公式中，Instance_of 根詞（後接一個值）可用來指定 If_In_Errors 物件的執行個體號碼 3，而 .# 根詞會指出應該將目前的執行個體指定元用於 If_In_Octets 物件。

If_In_Errors.3 / If_In_Octets.#

假設目前的執行個體指定元是 ALL，而且 If_In_Errors 與 If_In_Octets 分別是 OID 為 1.3.6.1.2.1.2.2.1.14 及 1.3.6.1.2.1.2.2.1.10 的清單屬性。將會使用下列 OID 執行各個執行個體的除法運算：

If_In_Errors OID = 1.3.6.1.2.1.2.2.1.14 + .3
If_In_Octets OID = 1.3.6.1.2.1.2.2.1.10 + .1

If_In_Errors OID = 1.3.6.1.2.1.2.2.1.14 + .3
If_In_Octets OID = 1.3.6.1.2.1.2.2.1.10 + .2

If_In_Errors OID = 1.3.6.1.2.1.2.2.1.14 + .3
If_In_Octets OID = 1.3.6.1.2.1.2.2.1.10 + .n

請注意 Instance_of 根詞如何覆寫 If_In_Errors 屬性 OID 的執行個體指定元。Instance_of 根詞不會允許以各個可能的執行個體作為選取 ALL 之結果的各個屬性，而是允許指定特定的執行個體。不過，執行個體指定元仍可決定特定監看的執行個體號碼。

模型的 `INSTANCE_ID` 及執行個體指定元會套用至監看運算式中的各個清單屬性。如果監看運算式包含使用不同執行個體的屬性，嘗試新增監看時會傳回錯誤。

例如，如果一個輸入的公式所包含的屬性使用電路板號碼作為執行個體，而另一個公式使用電路板連接埠作為執行個體，則會傳回錯誤。

監看定義屬性

建立模型的監看時，會建立屬性來儲存監看詳細資料。屬性的名稱是由電腦產生的。產生方式是使用模型所屬的模型類型下，監看定義群組中的主動資料庫開發人員 ID (登錄或預設)。

監看目的地屬性

建立模型的監看時，也將建立與監看相同名稱及資料類型的目的地屬性。此屬性類似於其他任何模型類型屬性。監看運算式會在讀取此屬性時自動進行評估。評估的結果就是讀取作業的結果。

臨界值參考及重設相容性

對於臨界值監看，臨界值及重設運算式所屬的類型必須與監看運算式相容。

下表顯示結果類型及對應的可接受類型：

結果類型	可接受類型
BOOLEAN	布林值
TEXT STRING	文字字串
INTEGER	整數、列舉、實數
ENUMERATION	整數、列舉、實數
REAL	實數
DATE	實數、日期、時間刻度、計數器、量測計、Counter64
TIME TICKS	實數、日期、時間刻度、計數器、量測計、Counter64
COUNTER	實數、日期、時間刻度、計數器、量測計、Counter64

結果類型	可接受類型
GAUGE	實數、日期、時間刻度、計數器、量測計、Counter64
OBJECT Id	OBJECT Id
IP 位址	IP_ADDR
COUNTER64	Counter64，實數

附錄 A：監看類型範例

本節包含以下主題：

- [第一個監看案例](#) (位於 p. 41)
- [第二個監看案例](#) (位於 p. 43)
- [變更時評估監看](#) (位於 p. 45)
- [輪詢的臨界值監看](#) (位於 p. 46)
- [第一個隨需監看案例](#) (位於 p. 47)
- [第二個隨需監看案例](#) (位於 p. 49)
- [第三個隨需監看案例](#) (位於 p. 51)
- [第四個隨需監看案例](#) (位於 p. 53)
- [第一個可用性與測試監看案例](#) (位於 p. 54)
- [第二個可用性與測試監看案例](#) (位於 p. 55)
- [第三個可用性與測試監看案例](#) (位於 p. 57)
- [第四個可用性與測試監看案例](#) (位於 p. 58)

第一個監看案例

下列參數會建立 Hub_CSI_IRBM 的監看。此監看每隔 60 秒檢查一次傳輸的框架總數以及收到的衝突總數。如果超過一百萬的臨界值，便會產生輕微警報，並顯示「過多衝突」訊息。「臨界值重設值」指出當衝突總數回落到 500,000 時，臨界值狀態會從 [已違規] 重設為 [正常] (並清除警報)。

您可以建立此案例的監看，如下所示：

1. 建立類型為「計數器」的非使用中監看，並命名為 HubColls_v。在運算式中指派 1000000 的衝突總數。
2. 建立類型為「計數器」的非使用中監看，並命名為 HubColls_r。在運算式中指派 500000 的衝突總數。
3. 建立類型為「計數器」的使用中監看，並命名為 HubColls，然後為其指派值，如下列監看範例所示。

範例：監看

第一個監看包含下列參數：

- 名稱：HubColls_v
- 資料類型：計數器

- 運算式：
 - 運算式：1000000
- 執行個體：無
- 內容
 - 預設啓動：非使用中
- 臨界值：無

第二個監看包含下列參數：

- 名稱：HubColls_r
- 資料類型：計數器
- 運算式：
 - 運算式：500000
- 執行個體：無
- 內容
 - 預設啓動：非使用中
- 臨界值：無

第三個監看包含下列參數：

- 名稱：HubColls
- 資料類型：計數器
- 運算式
 - 運算式：Hub_Trans_Coll+Hub_Rec_Colls
- 執行個體：無
- 內容
 - 預設啓動：使用中
 - 評估：依輪詢
 - 輪詢間隔：00:01:00
- 臨界值
 - 附加臨界值 (檢查) 臨界值違規狀況值：> HubColls_v (建立計數器監看，名稱爲 HubColls_v，非使用中，運算視爲 1000000)。

- 臨界值重設狀況值：HubColls_r (建立計數器監看，名稱爲 HubColls_r，非使用中，運算式爲 500000)。
- 建立 HubColls 監看前，必須先建立 HubColls_v 及 HubColls_r 屬性。
- 產生警報 (檢查) 嚴重性：輕微警報說明：CollsExceeded (建立新的警報，名稱爲 CollsExceeded，訊息是「過多衝突」。)

第二個監看案例

網路管理員想要監控伺服器的磁碟使用率水準，但是不要監控 CD-ROM 光碟機。CD-ROM 光碟機在沒有 CD 時一般會顯示 0% 使用率，但在將完整的 CD 放入光碟機時則會顯示接近 100% 的使用率。不過，這類資訊對於管理員而言毫無用處。

爲了檢查磁碟使用率，管理員想要使用 RFC2790App，這能夠將裝置類型傳回爲 OID。各個 OID 均對應於不同的裝置類型。此 MIB 中必須根據儲存區類型檢查的任何運算式，必須使用本身的 OID 進行比較。(您可比較字串與 OID)。

下表顯示 RFC2790App 的裝置類型/OID 對應。

裝置類型	OID
hrStorage	1.3.6.1.2.1.25.2
hrStorageTypes	1.3.6.1.2.1.25.2.1
hrStorageOther	1.3.6.1.2.1.25.2.1.1
hrStorageRam	1.3.6.1.2.1.25.2.1.2
hrStorageVirtualMemory	1.3.6.1.2.1.25.2.1.3
hrStorageFixedDisk	1.3.6.1.2.1.25.2.1.4
hrStorageRemovableDisk	1.3.6.1.2.1.25.2.1.5
hrStorageFloppyDisk	1.3.6.1.2.1.25.2.1.6
hrStorageCompactDisk	1.3.6.1.2.1.25.2.1.7
hrStorageRamDisk	1.3.6.1.2.1.25.2.1.8
hrStorageFlashMemory	1.3.6.1.2.1.25.2.1.9
hrStorageNetworkDisk	1.3.6.1.2.1.25.2.1.10

管理員必須對 RFC2790App 模型建立兩個新的監看。第一個監看設定用以在第二個監看中排除 CD-ROM 光碟機 (OID 1.3.6.1.2.1.25.2.1.7) 不予監控的布林值。第二個監看會監控主機中其他類型的儲存裝置本身的磁碟使用率 (以百分比表示)。在此範例中，如果使用率超過 90%，就會傳送警報。

範例：監看

第一個監看包含下列參數：

- 名稱：isNotCDROM
- 資料類型：布林值
- 運算式：hrStorageType.# != 1.3.6.1.2.1.25.2.1.7
- 執行個體：全部
- 內容
 - 預設啓動：使用中
 - 評估：隨需
- 臨界值：無

第二個監看包含下列參數：

- 名稱：PctDiskUsed
- 資料類型：實數
- 運算式
 - 運算式： $\text{REAL}(\text{hrStorageUsed.\#}) / \text{REAL}(\text{hrStorageSize.\#}) * 100 * \text{REAL}(\text{isNotCDROM.\#})$
 - 執行個體：全部
- 內容
 - 預設啓動：使用中
 - 評估：依輪詢
 - 輪詢間隔：30 秒
- 臨界值
 - 臨界值違規狀況值 \geq \langle 設定為可接受的磁碟使用率水準 \rangle

- 警報
 - 警報嚴重性：輕微
 - 警報說明：DiskUtilAlarm
 - 使用者可清除警報
 - 使用者清除警報時，不重設監看。
- 指令檔：無

變更時評估監看

下列監看是變更時評估 (EoC) 監看的範例。這些監看可確定檢視的編輯時間。因此，管理員可執行搜尋，顯示 LAN 的確切編輯時間，以及新模型是在何時新增至這些 LAN。這兩個屬性會在變更時評估，而且這兩個屬性都不太會經常變更。

範例：變更時評估監看

第一個監看包含下列參數：

- 名稱：Child_Count_Watch
- 資料類型：整數
- 運算式
 - 運算式：Child_Count
 - 執行個體：無
- 內容
 - 預設啟動：使用中
 - 評估：依 EoC
 - 輪詢間隔：無
- 臨界值：無

第二個監看包含下列參數：

- 名稱：Edit_Count
- 資料類型：整數

- 運算式
 - 運算式：EDIT_COUNT
 - 執行個體：無
- 內容
 - 預設啓動：使用中
 - 評估：依 EoC
- 臨界值：無

輪詢的臨界值監看

在下列範例中，會建立輪詢的臨界值監看，以在每次 LAN_802_3 中的子件數目超過 5 時產生輕微警報。子件計數回落到低於 5 時，監看會隨即清除警報。

首先，建立屬性監看 ChildLimit_v (任意名稱)，並設定運算式「5」。建立監看會詳細說明此監看的建立情形。然後建立輪詢的臨界值監看，命名為 Child_Limit。

範例：輪詢的臨界值監看

第一個監看包含下列參數：

- 名稱：ChildLimit_v
- 資料類型：計數器
- 運算式：5
- 內容：無
- 臨界值：無

第二個監看包含下列參數：

- 名稱：ChildLimit
- 資料類型：整數
- 運算式
 - 運算式：Child_Count
- 執行個體：無

- 內容
 - 預設啓動：使用中
 - 評估：依輪詢
 - 輪詢間隔：00:01:00
- 臨界值
 - 附加臨界值 (檢查)
 - 臨界值違規狀況值：> ChildLimit_v
 - 臨界值重設狀況值：<= ChildLimit_v
 - 產生警報 (檢查)
 - 嚴重性：輕微
 - 警報說明：ExceedChildLimit (建立名稱爲 ExceedChildLimit 的新警報，訊息是「僅允許 5 個子件」)。

第一個隨需監看案例

只要向隨需監看要求資訊，就會評估隨需監看。

下列的簡單隨需監看範例會搭配輪詢的臨界值監看，後者會在 Cisco 路由器 (模型類型：Rtr_CiscoIGS) 開始記憶體不足時產生警報。屬性 freeMem 小於 1500，超過 2500 時將清除。

範例：簡單隨需監看搭配輪詢的臨界值監看

第一個監看包含下列參數：

- 名稱：MemLow
- 資料類型：整數
- 運算式
 - 運算式：1500
 - 執行個體：無

- 內容
 - 預設啓動：非使用中
 - 評估：隨需
- 臨界值：無

第二個監看包含下列參數：

- 名稱：MemHigh
- 資料類型：整數
- 運算式
 - 運算式：2500
 - 執行個體：無
- 內容
 - 預設啓動：非使用中
 - 評估：隨需
- 臨界值：無

第三個監看包含下列參數：

- 名稱：Mem_Good
- 資料類型：整數
- 運算式
 - 運算式：freeMem
 - 執行個體：無
- 內容
 - 預設啓動：使用中
 - 評估：依輪詢
 - 輪詢間隔：1 分鐘
- 臨界值
 - 臨界值違規狀況值：<= MemLow
 - 臨界值重設狀況值：> MemHigh

- 警報
 - 警報嚴重性：嚴重
 - 警報說明：Rtr+Memory
- 指令檔：無

第二個隨需監看案例

只要向隨需監看要求資訊，就會評估隨需監看。

下列監看是用來顯示當 Cisco 路由器 (模型類型：Rtr_Cisco) 的效能逐漸降低時，由網路管理員以色彩標示的警報情況。輕微警報的值為 50 至 59；嚴重警報的值為 60 至 69，而重大警報的值超過 70。

範例：隨需監看

第一個監看包含下列參數：

- 名稱：True_Ref
- 資料類型：布林值
- 運算式
 - 運算式：1
 - 執行個體：無
- 內容
 - 預設啟動：非使用中
 - 評估：隨需
- 臨界值：無

第二個監看包含下列參數：

- 名稱：Minor_Busy
- 資料類型：布林值
- 運算式
 - 運算式：((busyPer >= 50) & (busyPer <60))
 - 執行個體：無

- 內容
 - 預設啓動：使用中
 - 評估：依輪詢
 - 輪詢間隔：1 分鐘
- 臨界值
 - 臨界值違規狀況值： $> \text{True_Ref}$
 - 臨界值重設狀況值： $\leq \text{True_Ref}$
- 警報：
 - 警報嚴重性：輕微
 - 警報說明：RtrBusy1
 - 指令檔：無

第三個監看包含下列參數：

- 名稱：Major_Busy
- 資料類型：布林值
- 運算式
 - 運算式： $((\text{busyPer} \geq 60) \& (\text{busyPer} < 70))$
 - 執行個體：無
- 內容
 - 預設啓動：使用中
 - 評估：依輪詢
 - 輪詢間隔：1 分鐘
- 臨界值
 - 臨界值：臨界值違規狀況值： $== \text{True_Ref}$
 - 臨界值重設狀況值： $!= \text{True_Ref}$
- 警報
 - 警報嚴重性：嚴重
 - 警報說明：RtrBusy 2
 - 指令檔：無

第四個監看包含下列參數：

- 名稱：Critical_Busy
- 資料類型：布林值
- 運算式
 - 運算式：(busyPer >= 70)
 - 執行個體：無
- 內容
 - 預設啟動：使用中
 - 評估：依輪詢
 - 輪詢間隔：60
- 臨界值
 - 臨界值違規狀況值：== True_Ref。
 - 臨界值重設狀況值：!= True_Ref。
- 警報：
 - 警報嚴重性：重大
 - 警報說明：RtrBusy3
 - 指令檔：無

第三個隨需監看案例

只要向隨需監看要求資訊，就會評估隨需監看。

對於此案例，網路管理員想要在「框架轉送」連結失敗，致使撥號備份線路啟動時看到警報。撥號備份是 ISDN 介面類型 21 (基本 ISDN 服務)。下列監看會確認是否有在任何 ISDN 連接埠收到流量。如果有流量，將對 Cisco 路由器 (模型類型：Rtr_Cisco) 發出輕微警報。

範例：隨需監看

第一個監看包含下列參數：

- 名稱：True_Ref
- 資料類型：布林值

- 運算式
 - 運算式：1
 - 執行個體：無
- 內容
 - 預設啓動：非使用中
 - 評估：隨需
- 臨界值：無

第二個監看包含下列參數：

- 名稱：ISDN_Backup
- 資料類型：布林值
- 運算式
 - 運算式：((COUNTER_DELTA (ifInOctets.#) > 0) & (ifType.# == 21))
 - 執行個體：全部
- 內容
 - 預設啓動：使用中
 - 評估：依輪詢
 - 輪詢間隔：1 分鐘
- 臨界值
 - 臨界值違規狀況值：== True_Ref
 - 臨界值重設狀況值：!= True_Ref
- 警報：
 - 警報嚴重性：輕微
 - 警報說明：Backup_Active
- 指令檔：無

第四個隨需監看案例

只要向隨需監看要求資訊，就會評估隨需監看。

對於此案例，監看會監控 ISDN 介面模型上的 `ifOutOctets` 值，以查看該值是否增加或是否為非零值。此值可指出介面是否使用中。您可以按照您的需求，修改輪詢、記錄、警報嚴重性及警報文字。第二個監看會使用 `true` 或 `false` (1/0) 指標：1 表示介面正在傳送 `ifOutOctets`，而 0 表示介面沒有傳送。

範例：隨需監看

第一個監看包含下列參數：

- 名稱：isISDN
- 資料類型：布林值
- 運算式
 - 運算式：`(X_ifType == 21)`
 - 執行個體：無
- 內容
 - 預設啟動：使用中
 - 評估：隨需
 - 可繼承：False
- 臨界值
 - 臨界值：無

第二個監看包含下列參數：

- 名稱：ISDN_Up
- 資料類型：整數
- 運算式
 - 運算式：`MAX(0, MIN(1, INTEGER((((COUNTER_DELTA (X_OutOctets) * INTEGER(isISDN)) > 0))))`
 - 執行個體：無

- 內容
 - 預設啟動：使用中
 - 評估：依輪詢
 - 輪詢間隔：0+00:05:00
 - 可繼承：False
- 臨限值：
 - 臨限值違規狀況值：== 1
 - 臨限值重設狀況值：!= 1
- 警報
 - 警報嚴重性：輕微
 - 警報說明：ErrorTholdAlarm
 - 使用者可清除警報
 - 使用者清除警報時，不重設監看。
- 指令檔：無

第一個可用性及測試監看案例

此處說明的範例監看是要用於可用性及測試。

針對此案例，您建立了兩個監看，一個用來建立名稱為 `WatchLoad_v` 且運算式值為 `.75` 的屬性，另一個監看則使用 `CPUloadRate` 屬性。在下列範例中，您建立了名稱為 `WatchLoad_Alarm` 的警報說明，指出「CPU 負載已超過 75%」。

範例：可用性及測試監看

第一個監看包含下列參數：

- 名稱：`WatchLoad_v`
- 資料類型：實數
- 運算式
 - 運算式：`.75`
 - 執行個體：無

- 內容
 - 預設啓動：非使用中
 - 評估：隨需
- 臨界值：無

第二個監看包含下列參數：

- 名稱：CPU_Load
- 資料類型：實數
- 運算式
 - 運算式：CPUloadRate
 - 執行個體：無
- 內容
 - 預設啓動：使用中
 - 評估：依輪詢
 - 輪詢間隔：1 分鐘
- 臨界值
 - 臨界值違規狀況值：> WatchLoad_v
 - 臨界值重設狀況值：<= WatchLoad_v
- 警報
 - 警報嚴重性：輕微
 - 警報說明：WatchLoad_Alarm
 - 指令檔：無

第二個可用性及測試監看案例

此處說明的範例監看是要用於可用性及測試。

在此範例中，您建立了 EoC 監看，以在容器模型的「複合狀況」屬性超過 4 時產生輕微警報。您首先建立名稱爲 `conditionCheck_ref` 的參考屬性，並將該屬性用於名稱爲 `ConditionCheck` 的監看中。然後使用 [警報說明] 對話方塊建立名稱爲 `ConditionCheckAlarm` (或其他適合的字串) 的警報說明。

參考屬性有下列參數：

- 名稱：ConditionCheck_ref
- 資料類型：整數
- 運算式
 - 運算式：4
 - 執行個體：無
- 內容：無
- 臨界值：無

範例：可用性及測試監看

此監看包含下列參數：

- 名稱：ConditionCheck
- 資料類型：整數
- 運算式
 - 運算式：Composite_Condition
 - 執行個體：無
- 內容
 - 預設啓動：使用中
 - 評估：EoC
- 臨界值
 - 臨界值違規狀況值：> ConditionCheck_ref
 - 臨界值重設狀況值：<= ConditionCheck_ref
- 警報：
 - 警報嚴重性：輕微
 - 警報說明：ConditionCheckAlarm
- 指令檔：無

第三個可用性及測試監看案例

此處說明的範例監看是要用於可用性及測試。

在此範例中，如果負載值持續 90 分鐘低於 10%，便會建立警報。若要配置此監看，請將監看運算式變更爲使用所需的外部屬性，並將臨界值設定爲等於所需持續期間的連續輪詢數除以輪詢間隔的值。

範例：可用性及測試監看

第一個監看包含下列參數：

- 名稱：Watch_Load_Under_10_Pct
- 資料類型：整數
- 運算式
 - 運算式：MAX (0, MIN (1, (<此爲低於 10% 負載的測試運算式>)))
 - 執行個體：無
- 內容
 - 預設啓動：非使用中
 - 評估：隨需
- 臨界值：無

第二個監看包含下列參數：

- 名稱：Watch_TimeTicker_LoadUnder10Pct
- 資料類型：整數
- 運算式
 - 運算式： $(\text{Watch_TimeTicker_LoadUnder10Pct} + 1) * \text{Watch_Load_Under_10_Pct}$
 - 執行個體：無
- 內容
 - 預設啓動：非使用中
 - 評估：隨需
- 臨界值：*所需的連續輪詢數*

第四個可用性及測試監看案例

此處說明的範例監看是要用於可用性及測試。

第一個監看會監控 CPU 的持續參考使用量值 (此範例為 80%)。

第二個監看會在 CPU 使用量持續一段時間達到特定程度 (80%) 時觸發警報。此監看計算這段時間的方式，是使用臨界值 (3) 乘以輪詢間隔 (5 分鐘)。因此，如果 CPU 使用量持續 15 分鐘超過 80%，此監看會違反臨界值，並觸發警報。您可以按照您的需求調整這些值。

範例：可用性及測試監看

第一個監看包含下列參數：

- 名稱：CPU_Duration_Over_80
- 資料類型：整數
- 運算式
 - 運算式：MAX(0, MIN(1, INTEGER((INTEGER(cpqHoCpuUtilMin.#) >= 80))))
 - 執行個體：全部
- 內容
 - 預設啟動：使用中
 - 評估：隨需
 - 可繼承：False
- 臨界值：無

第二個監看包含下列參數：

- 名稱：CPU_Time_Duration
- 資料類型：整數
- 運算式
 - 運算式：((CPU_Time_Duration.# + 1) * CPU_Duration_Over_80.#)
 - 執行個體：全部

- 內容
 - 預設啓動：使用中
 - 評估：依輪詢
 - 輪詢間隔：0 + 00:05:00
 - 可繼承：False
- 臨界值
 - 臨界值違規狀況值： ≥ 3
 - 臨界值重設狀況值： < 3
- 警報
 - 警報嚴重性：輕微
 - 警報說明：ErrorTholdAlarm
 - 使用者可清除警報
 - 使用者清除警報時，不重設監看。
 - 指令檔：無