

# CA Performance Management Data Aggregator

進階使用者憑證指南

2.4





本文件包含內嵌說明系統與文件 (以下稱為「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」為 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，若您為「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」的製造商為 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 © 2014 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為相關公司所有。

## CA Technologies 產品參考資料

本文件提及下列 CA Technologies 產品：

- CA Performance Management Data Aggregator (Data Aggregator)
- Data Collector
- CA Performance Center

## 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：簡介 9

關於本指南.....	9
裝置支援如何運作.....	10
透過自身憑證進行裝置支援.....	11
自訂憑證案例.....	12
先決條件.....	12

## 第 2 章：下載結構描述和範例檔案 15

## 第 3 章：建立自訂元件 17

建立元件 XML 範本.....	17
瞭解元件 XML 結構.....	18
基本內容.....	19
ItemSyncDefinition.....	20
自訂元件不支援的內容.....	22
匯入自訂元件.....	23
驗證自訂元件結果.....	24
更新自訂元件.....	24

## 第 4 章：建立自訂度量系列 27

指南結構.....	27
建立度量系列 XML 範本.....	28
瞭解度量系列 XML 結構.....	29
基本內容.....	33
ComponentFacets.....	35
ItemFacets.....	35
SourceFacetTypes 屬性.....	36
AttributeGroup (度量系列) .....	37
BaselineDefinitions .....	43
Expressions.....	45
階層 .....	46
ComponentReconciliation.....	47
ItemReconciliation .....	48

---

MatchAlgorithms .....	49
ReconfigDetectionAttr .....	52
自訂度量系列不支援的內容 .....	52
匯入自訂度量系列 .....	53
驗證自訂度量系列結果 .....	54
其他度量支援 .....	54
更新自訂度量系列 .....	59

## 第 5 章：建立自訂廠商憑證 61

建立廠商憑證 XML 範本 .....	61
瞭解廠商憑證 XML 結構 .....	62
基本內容 .....	64
AttributeGroup (廠商憑證) .....	66
ExpressionGroup .....	71
HierarchyList .....	75
IndexTagList .....	76
匯入自訂廠商憑證 .....	77
使用 REST 用戶端匯入自訂廠商憑證 .....	78
使用自訂廠商憑證安裝程式 .....	79
驗證自訂廠商憑證結果 .....	80
篩選支援 .....	81
多重 MIB 表格支援 .....	84
AttributeGroup (多個 MIB 表格) .....	88
UseIndex .....	89
IndexTagList (多個 MIB 表格) .....	89
更新自訂廠商憑證 .....	90

## 附錄 A：廠商憑證運算式：運算式運算子、函數與全域變數 93

運算式運算子 .....	93
函數及全域變數 .....	95
availabilityWithSysUptime 函數 .....	95
mapModel 函數 .....	96
mapVendor 函數 .....	97
snmpConstArrayMap 函數 .....	98
mvelInfo 函數 .....	100
mvelWarn 函數 .....	101
mvelError Function .....	102
mvelDebug 函數 .....	104
mvelDebug 函數 .....	105

---

snmpCounter64 函數.....	106
snmpGetUpSinceTime 函數.....	107
snmpMax 函數.....	108
snmpObjectIDToASCIIString 函數.....	109
snmpOIDParser 函數.....	109
snmpOctetStringFloat 函數.....	111
snmpProtectedDiv 函數.....	112
snmpRound 函數.....	113
snmpStringParser 函數.....	114
snmpSvcs Function.....	115
storePortReconfig 函數.....	116
全域變數.....	117

## 附錄 B：疑難排解 119

疑難排解：無法建立廠商憑證.....	119
疑難排解：不支援度量系列.....	120
疑難排解：度量系列不完整.....	120
疑難排解：廠商憑證運算式錯誤.....	121



# 第 1 章：簡介

---

本節包含以下主題：

[關於本指南](#) (位於 p. 9)

[裝置支援如何運作](#) (位於 p. 10)

[透過自身憑證進行裝置支援](#) (位於 p. 11)

[自訂憑證案例](#) (位於 p. 12)

[先決條件](#) (位於 p. 12)

## 關於本指南

Data Aggregator 提供的廠商自我憑證介面可讓您擴充 CA Performance Management 監控支援。自我憑證可讓您以廠商 MIB 中的全新或自訂元件和度量補充 Data Aggregator 中現有的度量系列。

您可以建立廠商憑證或修改現有的裝置或元件支援。本指南討論如何使用自訂度量系列及自訂廠商支援來增加新的技術支援。

本指南提供了建立自訂廠商憑證與度量系列的進階案例。需要有 XML 和結構描述檔案的基本知識。

**附註：**Data Aggregator 提供了建立自訂廠商憑證與度量系列的基本與進階方法。基本方法的程序較為簡單，其中包含為現有支援技術 (度量系列) 增加廠商支援、使用使用者介面。此方法可滿足許多使用者的需求。進階方法則以原廠憑證格式為基礎，且提供一組完整的功能。本指南將說明進階憑證方法。如需基本憑證方法的詳細資訊，請參閱《*Data Aggregator 基本自我憑證指南*》。

本指南完全遵循特定範例。身為系統管理員的您可以監控框架轉送永久虛擬電路 (PVC)。立即可用的 Data Aggregator 不支援監控框架轉送 PVC，但您可以使用自我憑證來產生此支援。在本指南中，我們提供了定義元件、建立自訂度量系列以及定義自訂廠商憑證的相關步驟，以便監控框架轉送永久虛擬電路 (PVC)。

## 裝置支援如何運作

Data Aggregator 使用度量系列及廠商憑證支援廠商裝置。這些元件將共同決定 Data Aggregator 如何收集裝置的配置及操作度量。瞭解裝置支援如何在 Data Aggregator 中運作有助於您判斷您的裝置是否在 Data Aggregator 中獲得適當的支援。如果未獲得支援，瞭解這個程序有助於您調整設定，達到所需的結果。

**附註：**如有需要，您可自訂度量系列、廠商憑證或兩者，新增廠商裝置的支援。

Data Aggregator 使用下列配置功能來支援裝置：

1. **搜索設定檔**--決定 Data Aggregator 會在環境中搜索哪些項目，一般會依據 IP 位址的範圍進行。搜索程序將識別找到的各個項目所屬的「類型」。
2. **裝置集合**--將清查項目組織為相關項目的群組。依據項目類型及 IP 位址，項目將自動新增到裝置集合。
3. **監控設定檔**--控制裝置集合的輪詢率，並決定輪詢哪些度量系列。監控設定檔可輪詢一或多個度量系列。

**附註：**為協助確保系統不會出現輪詢流量超載的情形，請使用監控設定檔調整不同組度量的輪詢率。

4. **度量系列**--控制監控設定檔收集哪些度量。度量系列與一或多個廠商憑證相關聯，廠商憑證按照優先順序列出。

**附註：**在監控設定檔中重複使用度量系列有助於確保一致的資料報告。

5. **廠商憑證**--將廠商 MIB 的屬性對應到度量系列中的度量。另外，對於從項目收集的度量，決定如何設定格式，以用於 CA Performance Center UI 及報告中。對於項目提供的度量可能不同，端視項目的廠商而定。無論廠商為何，對應這些值都能確保以一致方式報告度量值。多個廠商憑證可以與單一度量系列產生關聯。在這種情況下，Data Aggregator 會使用廠商憑證分級清單來對應度量值。Data Aggregator 可使用與輪詢的項目相符的最高優先廠商憑證，計算度量值。

**附註：**SNMP MIB 之類的 MIB 可匯入系統中，做為構成廠商憑證的一部份。

### 範例：對於路由器裝置的支援

執行搜索設定檔時，Data Aggregator 將尋找項目，並識別為路由器。路由器管理的項目將自動新增到「所有路由器」裝置集合中。此裝置集合與「路由器」監控設定檔相關聯，此設定檔使用「CPU」及「記憶體」度量系列，搜索裝置的 CPU 及記憶體元件。這些度量系列也會判斷計算這些元件的度量值時，要使用的廠商憑證。依據此監控設定檔，Data Aggregator 將每 5 分鐘輪詢路由器，取得這些度量系列中的度量資料。例如，CPU 度量系列包含 CPU 閑置使用率、CPU 系統使用率，以及 CPU 良好使用率。最後，與度量系列相關聯的廠商憑證將決定如何計算原始度量資料，以及如何對於原始度量資料設定一致的格式。Data Aggregator 儲存對於路由器收集的度量資料，這些資料將由 CA Performance Center 用於 UI 及報告中。

## 透過自身憑證進行裝置支援

Data Aggregator 使用預先定義的憑證來支援常見廠商裝置。憑證可指定如何收集裝置的配置及操作度量。Data Aggregator 使用下列的憑證方法：

- 度量系列
- 廠商憑證

Data Aggregator 未針對您的裝置提供預先定義的憑證時，您如何收集資料？您可以自行認證裝置的支援。

**附註：**度量系列及廠商憑證屬於全域 (亦即並非租戶感知)。如需有關租戶的詳細資訊，請參閱《CA Performance Center 管理員指南》。

Data Aggregator 的自行認證支援可讓您建立自訂廠商憑證、自訂度量系列或兩者。依下列說明，決定您需要的方法：

- **僅廠商憑證**—一組要預設輪詢的度量，但是 Data Aggregator 不支援裝置廠商 MIB 使用這些度量。例如，Data Aggregator 提供 CPU 度量系列收集 CPU 使用率之類的資料。但您想要收集 Bargain Server Company 製造之伺服器的 CPU 資料。您可以使用製造商提供的 MIB，建立伺服器 CPU 的自訂廠商憑證。
- **僅度量系列**—預設支援裝置廠商 MIB，但是不輪詢 MIB 支援的某些度量。例如，廠商 MIB 支援程序的度量，但是 Data Aggregator 不提供收集該度量資料的「程序」度量系列。
- **兩種方法**--Data Aggregator 不支援裝置廠商 MIB 或其度量時，建立度量系列及廠商憑證。

## 自訂憑證案例

CA Performance Management 在您的網路基礎結構中支援立即可用的通用廠商、度量和元件。技術憑證入口網站會依照 Data Aggregator 版本、廠商憑證和度量系列，列出各項預設憑證：

<http://serviceassurance.ca.com/im/>

不過，您可以建立自訂 CA Performance Management 憑證，充分發揮基礎結構管理功能。當您需要下列項目的 CA Performance Management 支援時自訂：

- 新技術：
  - 定義新元件。
  - 建立自訂度量系列。
- 適用於現有技術的新度量：
  - 從預先定義的 (原廠) 度量系列建立自訂度量系列。
  - 定義自訂度量系列的自訂廠商憑證。
- 新廠商：
  - 建立原廠度量系列的自訂廠商憑證。
  - 建立自訂度量系列的自訂廠商憑證。

## 先決條件

下列資訊說明處理廠商憑證 XML 的先決條件。

- 需要 CA Performance Management Data Aggregator 2.2.00 版或更新版本。
- 需具備「憑證」的豐富知識 (如《CA Performance Management Data Aggregator 自我憑證指南》所述)，以及使用廠商憑證精靈的經驗。
- XML 編輯器。

建議使用以結構描述驗證 XML 的 XML 編輯器。遵循 XML 編輯器的指示來使用 XSD 檔案。

**附註：**本指南中的範例是使用 XML Notepad 2007 for Windows (可從 Microsoft® 下載中心免費下載) 加以建立和驗證。

- MIB 瀏覽器。例如，支援免費的 iReasoning 版本。

- 對 CA Performance Management Data Aggregator 伺服器的連線。
- 若要支援本指南中使用的範例，需具備 FRAMERELAY-RFC1315-MIB 和 IF-MIB。



## 第 2 章：下載結構描述和範例檔案

---

在建立自訂度量系列和廠商憑證 XML 檔案前，請下載和檢閱相關的結構描述 XSD 檔案和範例 XML 檔案。需要結構描述才能驗證自己的 XML 檔。請先檢閱範例檔案，然後再建立自己的檔案，才能確保 XML 內容正確無誤。

Data Aggregator 提供的結構描述檔案有元素類型、發生和允許長度的詳細資訊。這類檔案也包含可提供更多資訊的注釋，例如允許的字元和命名慣例。

在網頁瀏覽器的 [網址] 欄位中輸入下列 URL，可以下載結構描述和範例 XML 檔案。*Hostname* 是 Data Aggregator 主機，而預設 *port* 為 8581。

- <http://hostname:port/resource/xsd/IMDBCertificationFacet.xsd>
- <http://hostname:port/resource/xsd/ComponentFacet.xsd>
- <http://hostname:port/resource/xsd/ItemSyncDefinition.xsd>
- <http://hostname:port/resource/xsd/SNMPCertificationFacet.xsd>
- <http://hostname:port/resource/xsd/CertificationFacet.xsd>
- <http://hostname:port/resource/xsd/CertificationFacet.xsd>
- <http://hostname:port/resource/xsd/webservices.xsd>
- <http://hostname:port/resource/xsd/basewebservices.xsd>
- <http://hostname:port/resource/xsd/datamodel.xsd>



# 第 3 章：建立自訂元件

---

本節包含以下主題：

- [建立元件 XML 範本](#) (位於 p. 17)
- [瞭解元件 XML 結構](#) (位於 p. 18)
- [匯入自訂元件](#) (位於 p. 23)
- [驗證自訂元件結果](#) (位於 p. 24)
- [更新自訂元件](#) (位於 p. 24)

## 建立元件 XML 範本

若要建立範例 XML 檔案，以便做為建立自訂元件時的範本，請使用慣用的 REST 用戶端。

從擷取現有的元件定義清單開始。然後可以驗證是否已支援您所需的元件。

**請依循下列步驟：**

1. 透過與 Data Aggregator 伺服器的連線來設定 REST 用戶端。
2. 針對 REST 用戶端中的 Data Aggregator Web 服務輸入下列 URL：  
`http://da_hostname:8581/typecatalog/components`  
所有可用元件的清單隨即顯示。
3. 驗證是否已支援您所需的元件。

如果尚未支援您所需的元件，請檢視及匯出類似您要建立之自訂元件的單一元件定義。

**請依循下列步驟：**

1. 若要擷取類似您所需元件的特定元件，請輸入下列 URL：  
`http://da-hostname:8581/typecatalog/components/name`  
***name***  
現有、特定元件的名稱，例如 NormalizedCPUInfo。
2. 選取 [方法] 索引標籤上的 GET。

3. 執行此方法。

傳回的 XML 包含元件定義。

您可以使用此 XML 做為建立新元件時的範本。

4. 將元件 XML 複製到文字檔中，然後視需要進行修改。如需 XML 結構的範例，請參閱[瞭解元件 XML](#) (位於 p. 18)。

## 瞭解元件 XML 結構

裝置元件可定義與裝置相關聯的元件項目類別。雖已提供許多原廠定義的元件，但通常會針對自訂度量系列定義自訂元件。元件可以定義選用 **ItemSyncDefinition**，以將元件項目同步處理至 CA Performance Center。您可以接著檢視 [清查] 清單、群組和內容頁面中的元件。

以下的元件定義 XML 範例可支援框架轉送 PVC 範例。已增加新元件 frPVC：

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Auto-generated by the type catalog local manager.-->
<DataModel namespace="http://im.ca.com/inventory"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="ComponentFacet.xsd">
  <FacetType name="frPVC">
    <Documentation>A Frame Relay PVC</Documentation>
    <FacetOf namespace="http://im.ca.com/core" name="Item" />
    <Component>true</Component>
    <ItemSyncDefinition itemTypeName="component"
      itemSubtypeName="frpvc" itemTypeLabel="FrameRelayPVC"
      itemTypeLabelPlural="FrameRelayPVCs" categorize="false"
      groupBy="false" context="true">
      </ItemSyncDefinition>
    </FacetType>
  </DataModel>
```

將 **ItemSyncDefinition** 中的內容設定為 “true”，以啓用自訂內容頁面的連結。您可以從 [清查]、[裝置元件] 清單中顯示的 frPVC 裝置元件導覽到此頁面。將它設定為 “true” 也可讓您選取度量系列做為「內容」，而使自訂度量系列可用於 [動態趨勢] 圖表類型中。如需詳細資訊，請參閱 [ItemSyncDefinition](#) (位於 p. 20)。

## 基本內容

自訂元件的基本內容有助於與您建立的其他自訂元件相區別。

### **FacetType/name**

指定元件名稱。每個元件都必須有唯一名稱，以便在系統內部進行識別。小心選擇最不可能與未來類似元件發生命名衝突的名稱。例如，定義命名方式，有助於確保元件名稱是唯一的。

**附註：**此名稱不會對外公開。若要在使用者介面中顯示元件名稱，請使用 `ItemSyncDefinition`、`itemTypeLabel` 和 `itemTypeLabelPlural` 元素。

**可以更新：**否

**可能值：**英數字元和底線。不允許使用點和破折號。

### 文件

指定元件的內部註解。若要充分利用註解，建議您說明為何及何時新增或變更元件。

**可以更新：**是

**可能值：**純文字

**更新效果：**無

**更新的生效時間：**立即

**更新生效所需執行的動作：**無

### 元件

確定此項目是元件。

**可以更新：**否

**可能值：**true

### **FacetOf**

確定此元件是項目。

**可以更新：**否

**可能值：** `namespace="http://im.ca.com/core" name="Item"`

## ItemSyncDefinition

**ItemSyncDefinition** 屬性是選用的。此屬性指定元件項目如何同步處理並顯示在 CA Performance Center 中。如果未指定元件項目，則元件項目就不會顯示在 CA Performance Center 清查清單中 (例如，裝置元件)。但是，仍會在自訂檢視中進行報告。

### ItemSyncDefinition/itemTypeName

指定項目類型。對於自訂元件，此值必須為「元件」。

**可以更新：**否

**可能值：**元件

### ItemSyncDefinition/itemSubtypeName

指定 CA Performance Center 中元件的內部名稱。對所有元件而言，此值必須是唯一的。使用可避免未來與原廠和自訂元件發生衝突的命名慣例，例如使用表示貴組織的前置詞 'acmeFan'。

**可以更新：**否

**可能值：**所有元件中唯一的英數字元

### ItemSyncDefinition/itemTypeLabel

指定顯示此類型的單一元件時使用的使用者介面標籤。例如，此值使用於 [清查]、[裝置元件] UI [類型] 欄中。

**可以更新：**是

**可能值：**所有元件中唯一的純文字

**更新效果：**標籤會顯示在 CA Performance Center 清查使用者介面中。

**更新的生效時間：**允許發生重新同步並在 15 分鐘內完成更新。

**更新生效所需執行的動作：**無

### ItemSyncDefinition/itemTypeLabelPlural

指定顯示此類型的多個元件時要使用的使用者介面標籤。使用 [清查] 功能表 (請參閱 **groupBy**) 和 [群組] 名稱 (請參閱 **categorize**)。

**可以更新：**是

**可能值：**所有元件中唯一的純文字

**更新效果：**標籤會顯示在 CA Performance Center 清查 UI 中。

**更新的生效時間：**允許發生重新同步並在 15 分鐘內完成更新。

**更新生效所需執行的動作：**無

### ItemSyncDefinition/categorize

指示 CA Performance Center 在 [清查]、[所有項目] 之下建立清查群組。此群組包含此元件類型的所有項目。此群組的名稱為 '*{itemTypeLabelPlural}*'。

**附註：**建立的清查群組無法使用於報告儀表板。此群組僅適用於清查目的。如果選取群組進行報告，則不會顯示任何資料。其他以裝置為基礎的清查群組 (在 [清查]、[所有項目] 之下) 可用於報告，例如 [路由器] 和 [伺服器]。不過，以元件為基礎的清查群組則無法用於報告 (例如，[裝置元件])。

**可以更新：**是

**可能值：**true、false

**更新效果：**在 CA Performance Center 中建立或移除 [清查] 群組。若為 CA Performance Center 群組管理員，在 [管理群組] 頁面上，群組會建立在 [清查]、[所有項目]、*{itemTypeLabelPlural}* 之下。

**更新的生效時間：**允許發生重新同步並在 30 分鐘內完成更新。

**更新生效所需執行的動作：**項目通常會在 30 分鐘內顯示在群組中。如未顯示，手動重新同步 Data Aggregator 資料來源。按一下 [重新同步] 確認按鈕之前，務必選取 [執行完整重新同步]。

### ItemSyncDefinition/groupBy

指示 CA Performance Center 建立清查功能表項目 (在 [清查] 之下)，以便檢視此元件項目類型的所有項目。功能表的名稱為 '*{itemTypeLabelPlural}*'。設定檢視內容時，此屬性也會造成元件類型顯示在 [內容類型] 下拉式功能表中。若為 false，元件會列在 [清查]、[裝置元件] 表格 (其類型為 '*{itemTypeLabel}*' ) 中。groupBy 內容不會建立群組 (請參閱 **categorize**)。

**可以更新：**是

**可能值：**true、false

**更新效果：**若為 true 則會建立功能表項目，否則會在值為 false 時移除。

**更新的生效時間：**允許發生重新同步並允許在 15 分鐘內完成更新。

**更新生效所需執行的動作：**無

#### **ItemSyncDefinition/context**

讓每個元件項目名稱成為 [清查] 元件檢視中的內容超連結，若按一下，則會導覽至個別元件內容頁面。

**可以更新：**是/否

**可能值：**純文字

**更新效果：**讓每個元件項目名稱成為 [清查] 元件檢視中的內容超連結。

**更新的生效時間：**允許發生重新同步並允許在 15 分鐘內完成更新。

**更新生效所需執行的動作：**無

#### **移除 ItemSyncDefinition**

需要特定程序才能完整移除 ItemSyncDefinition。

**請依循下列步驟：**

1. 移除 ItemSyncDefinition 區段，包括 <ItemSyncDefinition> 開始和結束標記。
2. 將元件變更套用到 Data Aggregator 之後，請以管理員的身分登入 CA Performance Center。
3. 選取 [資料來源] 頁面上的 DataAggregator 資料來源。
4. 按一下 [重新同步] 按鈕。
5. 勾選 [執行完整重新同步] 選項，然後按一下 [重新同步] 確認按鈕。

重新同步程序隨即開始。允許在 15–30 分鐘內同步處理變更。

此程序完成時，會移除 ItemSyncDefinition 為此元件定義的所有 CA Performance Center 行為。

#### **自訂元件不支援的內容**

自訂元件不支援下列內容：

- 屬性
- Web 服務
- ItemSyncDefinition/isDeviceComponent\*

- ItemSyncDefinition/mapped\*
- ItemSyncDefinition/ItemProperty

\* 這些內容可以顯示在您的 XML 中，但無法設定為 true。

## 匯入自訂元件

將自訂元件匯入 Data Aggregator 之後，新元件即可用於支援自訂度量系列。

Data Aggregator 提供一項 Web 服務來支援新元件的匯入。使用任何 REST 用 戶端。建議搭配使用 RESTClient 與 CA Performance Management。如果尚未 安裝，可以從 <http://code.google.com/p/rest-client/> 取得 RESTClient GUI JAR 檔 案。

使用 RESTClient 時，請考量下列資訊：

- 若要加以啟動，請連按兩下 JAR 檔案。
- 當您 POST XML 時，確定 Charset 已設定為 UTF-8。  
若要檢視並確認此設定，請按一下 [編輯 Content-Type 和 Charset] 按鈕。
- 您也可以使回應內文自動縮排，如下所述：
  1. 按一下 [工具] 功能表上的 [選項]。
  2. 選取 [Etc.] 索引標籤。
  3. 選取 [使回應內文自動縮排] 並按一下 [確定]。

請依循下列步驟：

1. 輸入 `http://da-hostname:8581/typecatalog/components` 做為 URL。
2. 選取 POST 做為方法。
3. 選取「application/xml」做為 Body 設定中的「Body Content-type」。  
**重要！無法設定 Content-type 會造成 404 錯誤。**
4. 將自訂元件 XML 複製並貼到 [內文] 欄位中。
5. 執行此方法。

自訂元件隨即匯入。如果未發生錯誤，[HTTP 回應] 區段中的 [狀態] 欄位 隨即顯示：

HTTP/1.1 200 OK

**附註：**有其他傳回碼指出在更新自訂元件時發生錯誤。修正此錯誤並執行其他 POST 來重試更新元件。

**重要！** 為了避免資料遺失，每次建立或更新廠商憑證、度量系列或元件時，請務必備份您的部署目錄。

## 驗證自訂元件結果

匯入自訂元件 XML 之後，請驗證結果。在我們的範例中，我們會驗證 frPVC 元件是否匯入成功。

**請依循下列步驟：**

1. 在 REST 用戶端中輸入下列 URL：

`http://da_hostname:8581/typecatalog/components/frPVC`

頁面上會顯示自訂框架轉送元件 XML。

您也可以搜索新元件。同步處理之後，您可以接著使用 [清查]、[裝置元件] 清單來檢視元件。

## 更新自訂元件

您可以更新現有的自訂元件。

**附註：**如需在更新元件時更新標記或屬性之效果的詳細資訊，請檢閱自訂元件 XML 詳細資料。請特別檢閱特定屬性說明。

**請依循下列步驟：**

1. 在 URL 欄位中輸入下列網址：

`http://da_hostname:8581/typecatalog/components/name`

**名稱**

要更新的自訂元件名稱。

2. 選取 [方法] 索引標籤中的 PUT。

3. 將更新後的元件 XML 複製並貼到 [內文] 索引標籤上的 [編輯] 欄位中，並將 Content-type 設定為 application/xml。

**重要！** 無法設定 Content-type 會造成 404 錯誤。

4. 按一下 [URL] 欄位旁邊的 [移至] 按鈕。

自訂元件隨即更新。如果未發生錯誤，[HTTP 回應] 區段中的 [狀態] 欄位隨即顯示：

HTTP/1.1 200 OK

**附註：**有其他傳回碼指出在更新自訂元件時發生錯誤。修正此錯誤並執行其他 PUT 來重試更新元件。

**重要！** 為了避免資料遺失，每次建立或更新廠商憑證、度量系列或元件時，請務必備份您的部署目錄。

**更多資訊：**

[瞭解元件 XML 結構](#) (位於 p. 18)



# 第 4 章：建立自訂度量系列

---

匯入成功時，新度量系列即可用於支援自訂廠商憑證。度量系列也可與監控設定檔相關聯，以供搜索和輪詢。

本節包含以下主題：

[指南結構](#) (位於 p. 27)

[建立度量系列 XML 範本](#) (位於 p. 28)

[瞭解度量系列 XML 結構](#) (位於 p. 29)

[匯入自訂度量系列](#) (位於 p. 53)

[驗證自訂度量系列結果](#) (位於 p. 54)

[其他度量支援](#) (位於 p. 54)

[更新自訂度量系列](#) (位於 p. 59)

## 指南結構

本指南說明必要的 XXX。TO COME

VCs:

- 建立 vcs
- 編輯 vcs
- 匯入 vc
- 匯出 vc

原廠度量系列將定義要監控的最常用度量屬性。不過，要收集新度量屬性的資料時，可建立自訂度量系列。例如，如果沒有度量系列可供收集程序資料，您可以建立一個度量系列。然後，使用新的度量系列，建立監測程序的度量屬性所用的廠商憑證。

## 建立度量系列 XML 範本

使用慣用的 REST 用戶端來建立範例 XML 檔案，以便做為建立自訂度量系列時的範本。

從擷取現有的度量系列清單開始。然後可以驗證是否已支援您所需的度量系列。

**請依循下列步驟：**

1. 透過與 Data Aggregator 伺服器的連線來設定 REST 用戶端。
2. 針對 REST 用戶端中的 Data Aggregator Web 服務輸入下列 URL。

`http://da_hostname:8581/typecatalog/metricfamilies`

所有可用的度量系列清單隨即顯示。

3. 驗證是否已支援您所需的度量系列。

如果尚未支援您所需的度量系列，請檢視及匯出類似您要建立之自訂度量系列的度量系列。

**請依循下列步驟：**

1. 若要擷取類似您所需度量系列的特定度量系列，請輸入下列 URL：

`http://da_hostname:8581/typecatalog/metricfamilies/name`

**name**

特定、現有度量系列的名稱，例如 NormalizedCPUInfo。

2. 選取 [方法] 索引標籤中的 GET。
3. 執行此方法。

傳回的 XML 包含度量系列資訊。

您可以使用此 XML 做為建立自訂度量系列時的範本。

4. 將度量系列 XML 複製到文字檔中，然後視需要進行修改。如需 XML 結構的範例，請參閱[瞭解度量系列 XML](#) (位於 p. 29)。

## 瞭解度量系列 XML 結構

度量系列定義對特定技術要收集和報告的一組度量。這些度量經過標準化，因此無論廠商 (資料來源) 為何，報告均一致。並非所有廠商都會為度量系列中的每個度量提供值，且並非所有度量都是必要的。如果廠商未提供值，則度量為 "null"。此外，任何以 null 度量為基礎的報告檢視都是空的。

度量系列也會定義在搜索期間擷取的屬性，如項目名稱和索引。此外，也已定義可重新調整元件比對的搜索規則。您可在監控設定檔中包含度量系列。監控設定檔中的這組度量系列可決定要針對與此設定檔相關聯的各裝置集合中的裝置，收集哪些度量。

以下的度量系列範例可支援框架轉送 PVC 範例。請注意，範例自訂元件 frPVC 如何納入 ComponentFacets 區段中 (為了舉例，以粗體顯示)：

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Auto-generated by the type catalog local manager. -->
<DataModel namespace="http://im.ca.com/normalizer"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="IMDBCertificationFacet.xsd">
  <FacetType name="frPVCInfo"
  descriptorClass="com.ca.im.core.datamodel.certs.NormalizedFacetDescriptorImpl">
    <Documentation>Frame Relay Permanent Virtual
    Circuit</Documentation>
    <FacetOf namespace="http://im.ca.com/core" name="Item" />
    <AttributeGroup name="AttributeGroup" external="true"
    list="true">
      <Documentation />
      <Attribute name="Indexes" type="ObjectID[]">
        <Documentation />
        <Polled>false</Polled>
        <Baseline>false</Baseline>
        <IsDbColumn>false</IsDbColumn>
        <Variance>false</Variance>
        <StandardDeviation>false</StandardDeviation>
        <Minimum>false</Minimum>
        <Maximum>false</Maximum>
        <WriteOnPoll>false</WriteOnPoll>
        <RollupStrategy />
        <AttributeDisplayName />
        <Percentile>0</Percentile>
      </Attribute>
      <Attribute name="Names" type="String">
        <Documentation>The name of the frame relay
        circuit</Documentation>
        <Polled>false</Polled>
        <Baseline>false</Baseline>
        <IsDbColumn>false</IsDbColumn>
        <Variance>false</Variance>
        <StandardDeviation>false</StandardDeviation>
        <Minimum>false</Minimum>
        <Maximum>false</Maximum>
        <WriteOnPoll>false</WriteOnPoll>
        <RollupStrategy />
        <AttributeDisplayName />
        <Percentile>0</Percentile>
      </Attribute>
      <Attribute name="Description" type="String">
        <Documentation>A description for the frame relay
        circuit</Documentation>
```

```
<Polled>false</Polled>
<Baseline>false</Baseline>
<IsDbColumn>false</IsDbColumn>
<Variance>false</Variance>
<StandardDeviation>false</StandardDeviation>
<Minimum>false</Minimum>
<Maximum>false</Maximum>
<WriteOnPoll>false</WriteOnPoll>
<RollupStrategy />
<AttributeDisplayName />
<Percentile>0</Percentile>
</Attribute>
<Attribute name="BECNIn" type="Double">
    <Documentation>Backward congestion since the virtual
circuit was created</Documentation>
    <Polled>true</Polled>
    <Baseline>false</Baseline>
    <IsDbColumn>true</IsDbColumn>
    <Variance>false</Variance>
    <StandardDeviation>false</StandardDeviation>
    <Minimum>false</Minimum>
    <Maximum>false</Maximum>
    <WriteOnPoll>false</WriteOnPoll>
    <RollupStrategy>Sum</RollupStrategy>
    <AttributeDisplayName />
    <Percentile>0</Percentile>
</Attribute>
<Attribute name="FECNIn" type="Double">
    <Documentation>Forward congestion since the virtual
circuit was created</Documentation>
    <Polled>true</Polled>
    <Baseline>false</Baseline>
    <IsDbColumn>true</IsDbColumn>
    <Variance>false</Variance>
    <StandardDeviation>false</StandardDeviation>
    <Minimum>false</Minimum>
    <Maximum>false</Maximum>
    <WriteOnPoll>false</WriteOnPoll>
    <RollupStrategy>Sum</RollupStrategy>
    <AttributeDisplayName />
    <Percentile>0</Percentile>
</Attribute>
<Attribute name="FramesIn" type="Double">
    <Documentation>Frames received since the virtual circuit
was created</Documentation>
    <Polled>true</Polled>
    <Baseline>false</Baseline>
    <IsDbColumn>true</IsDbColumn>
    <Variance>false</Variance>
```

```
<StandardDeviation>false</StandardDeviation>
<Minimum>false</Minimum>
<Maximum>false</Maximum>
<WriteOnPoll>false</WriteOnPoll>
<RollupStrategy>Sum</RollupStrategy>
<AttributeDisplayName />
<Percentile>0</Percentile>
</Attribute>
<Attribute name="FramesOut" type="Double">
    <Documentation>Frames sent since the virtual circuit was
created</Documentation>
    <Polled>true</Polled>
    <Baseline>false</Baseline>
    <IsDbColumn>true</IsDbColumn>
    <Variance>false</Variance>
    <StandardDeviation>false</StandardDeviation>
    <Minimum>false</Minimum>
    <Maximum>false</Maximum>
    <WriteOnPoll>false</WriteOnPoll>
    <RollupStrategy>Sum</RollupStrategy>
    <AttributeDisplayName />
    <Percentile>0</Percentile>
</Attribute>
<Attribute name="BytesIn" type="Double">
    <Documentation>Bytes received since the virtual circuit was
created</Documentation>
    <Polled>true</Polled>
    <Baseline>false</Baseline>
    <IsDbColumn>true</IsDbColumn>
    <Variance>false</Variance>
    <StandardDeviation>false</StandardDeviation>
    <Minimum>false</Minimum>
    <Maximum>false</Maximum>
    <WriteOnPoll>false</WriteOnPoll>
    <RollupStrategy>Sum</RollupStrategy>
    <AttributeDisplayName />
    <Percentile>0</Percentile>
</Attribute>
<Attribute name="BytesOut" type="Double">
    <Documentation>Bytes sent since the virtual circuit was
created</Documentation>
    <Polled>true</Polled>
    <Baseline>false</Baseline>
    <IsDbColumn>true</IsDbColumn>
    <Variance>false</Variance>
    <StandardDeviation>false</StandardDeviation>
    <Minimum>false</Minimum>
    <Maximum>false</Maximum>
    <WriteOnPoll>false</WriteOnPoll>
```

```
<RollupStrategy>Sum</RollupStrategy>
<AttributeDisplayName />
<Percentile>0</Percentile>
</Attribute>
</AttributeGroup>
<Attribute name="SourceFacetTypes" cached="true" list="true" persistent="true" type="QName">
    <Documentation />
</Attribute>
<DisplayName>Frame Relay PVC</DisplayName>
<Expressions>
    <ExpressionGroup destCert="{http://im.ca.com/core}Item">
        <Expression destAttr="Name">Names</Expression>
    </ExpressionGroup>
    <ExpressionGroup
destCert="{http://im.ca.com/inventory}DeviceComponent">
        <Expression destAttr="IndexList">Indexes</Expression>
    </ExpressionGroup>
</Expressions>
<TableName>FR_PVC_INFO</TableName>
<ComponentFacets>
    <Facet>{http://im.ca.com/inventory}frPVC</Facet>
</ComponentFacets>
<Protocol>IMDB</Protocol>
<Normalized>true</Normalized>
</FacetType>
</DataModel>
```

## 基本內容

自訂度量系列的基本內容，有助於與您建立的其他自訂度量系列相區別。

決定基本內容時，請考量下列限制：

- 各度量系列的 `FacetType/name`、`DisplayName` 和 `TableName` 內容必須是唯一的。
- `Protocol` 標記一律為 `IMDB`。
- `Normalized` 標記一律為 `true`。
- 設定 `FacetType/descriptorClass` 內容以及所有的 `DataModel` 和 `FacetOf` 內容。

#### **FacetType/name**

指定度量系列名稱。對每個度量系列而言，此名稱必須是可在系統內部進行識別的唯一名稱。小心選擇最不可能與未來類似度量系列發生命名衝突的名稱。例如，定義命名方式，有助於確保度量系列名稱是唯一的。

**附註：**此名稱不會對外公開。若要在使用者介面中顯示度量系列名稱，請使用 `DisplayName` 元素。

**可以更新：**否

**可能值：**英數字元和底線。不允許使用點和破折號。在所有度量系列中必須是唯一的。

#### **DisplayName**

指定使用者介面中顯示的度量系列名稱。

**可以更新：**是

**可能值：**純文字 在所有度量系列中必須是唯一的。

**更新效果：**變更為系統管理員使用者介面中的名稱。

**更新的生效時間：**立即

**更新生效所需執行的動作：**重新整理使用者介面。

#### **文件**

指定度量系列的外部說明。若要充分利用註解，建議您說明為何及何時新增或變更度量系列。

**可以更新：**是

**可能值：**純文字

**更新效果：**無

#### **TableName**

指定用於儲存度量系列所收集之度量的資料庫表格名稱。

**可以更新：**是

**可能值：**大寫英數字元和底線 必須以字母開頭。在所有度量系列中必須是唯一的。

**範例：**PROCESS\_STATS

**更新效果：**輪詢資料儲存在新的資料庫表格集中。

**重要！** 當您更新 `TableName` 時，會遺失舊的輪詢資料。舊的報告檢視會受損。

**更新的生效時間：**立即 CA Performance Center 載入新的 MIB 檔案時，最久延遲 5 分鐘的時間才可建立新檢視。

**更新生效所需執行的動作：**必須重新建立檢視

## ComponentFacets

ComponentsFacets 區段列出在搜索期間建立的 **Facet**。搜索作業會將項目識別為裝置元件，或在兩個項目之間建立階層關係。

### Facet

指定在元件搜索期間附加至元件項目的 **Facet**。

**可以更新：**是

**可能值：**Facet 的 QName

**更新效果：**如果元件 **Facet** 已同步處理至 CA Performance Center，則可在 CA Performance Center 中看見此元件。

**更新的生效時間：**重新搜索

**更新生效所需執行的動作：**刪除裝置並重新搜索。

**更多資訊：**

[階層 \(位於 p. 46\)](#)

[ComponentReconciliation \(位於 p. 47\)](#)

## ItemFacets

**重要！** ItemFacets 是一個有可能變更的新區段，以支援未來複雜的度量系列結構。不建議使用該區段。

ItemFacets 區段列出在搜索期間建立並將項目識別為裝置的 **Facet**。

### Facet

指定在搜索期間附加至項目的 **Facet**。

**可以更新：**是

**可能值：**Facet 的 QName

**更新效果：**元件會針對指定的 Facet 顯示在 REST 服務上。如果元件 Facet 已同步處理至 CA Performance Center，則可在 CA Performance Center 中看見此元件。

**更新的生效時間：**重新搜索

**更新生效所需執行的動作：**刪除裝置並重新搜索。

**範例：**

```
<ItemFacets>
  <Facet>{http://im.ca.com/inventory}Host</Facet>
  <Facet>{http://im.ca.com/inventory}Device</Facet>
  <Facet>{http://im.ca.com/inventory}ConsolidatedAndDiscoveredMetricFamilyHistory</Facet>
  <Facet>{http://im.ca.com/core}Syncable</Facet>
  <!-- The IPDomainID attribute will be filled in by discovery -->
  <Facet>{http://im.ca.com/core}IPDomainMember</Facet>
</ItemFacets>
```

**更多資訊：**

[ItemReconciliation](#) (位於 p. 48)

## SourceFacetTypes 屬性

搜索需要 SourceFacetTypes 屬性，所以必須加以定義。

使用下列必要值：

- 名稱：SourceFacetTypes
- 類型：QName
- 快取：true
- 持續性：true
- 清單：true

**範例：** <Attribute name="SourceFacetTypes" type="QName" cached="true" persistent="true" list="true" />

**可以更新：**否

## AttributeGroup (度量系列)

AttributeGroup 是項目搜索屬性和度量屬性的集合。項目搜索屬性如同項目說明，都是在搜索期間設定。度量屬性是在輪詢期間收集。下列資訊說明 AttributeGroup 區段中使用的元素。

將 AttributeGroup/list 和 AttributeGroup/external 內容設定為 true。這些內容指定每個屬性都代表從外部來源取得的值清單。自訂下列 XML 元素：

### AttributeGroup/name

指定屬性群組名稱。符合 "<FacetType/name>Group" 命名方式。

**可以更新：**是

**可能值：**純文字

**更新效果：**無

### 文件

(選用) 指定屬性群組的說明。

**可以更新：**是

**可能值：**純文字

**更新效果：**無

## 一般屬性 (度量系列)

所有度量系列的一般屬性如下：

### Attribute/name

指定唯一的內部名稱。對於度量而言，此名稱也會用於命名資料庫欄。

**附註：**此名稱不會對外公開。若要在使用者介面中顯示屬性名稱，請使用 AttributeDisplayName 元素。

**可以更新：**是

**可能值：**英數字元和底線。

**更新效果：**對於度量，此屬性的值會儲存在對應於更新後名稱的新資料庫欄中。使用者會失去針對此度量 (具有較舊的名稱) 收集的歷史資料。自訂報告對於此度量的報告失敗。

**更新的生效時間：**下一次輪詢

**更新生效所需執行的動作：**無

#### **Attribute/type**

指出此屬性的資料類型。最常用的資料類型是 Int、Long、Double、String 或 ObjectId。資料庫會將度量屬性儲存為浮點。因此，這些屬性必須使用數值類型 (建議使用 Double)。其他類型使用於項目屬性。

**可以更新：**是

**可能值：**Boolean、Int、Long、Double (浮點)、BigInteger、String、DateTime、IPAddress、MACAddress、IPSubnet、OctetString (16 進位表示法)、ObjectId、ItemId、QName (完整名稱)

**附註：**類型名稱不區分大小寫，例如，"boolean" 與 Boolean 相同。

**更新效果：**對度量而言，沒有效果。所有度量都會以浮點形式儲存在資料庫中。若為項目屬性，則必須刪除並重新搜索裝置。

**更新的生效時間：**對於度量而言，在下一次輪詢時。若為項目屬性，則在重新搜索時。

**更新生效所需執行的動作：**對度量而言，不需任何動作。若為項目屬性，刪除裝置並重新搜索。

#### **AttributeDisplayName**

指定運算子及管理員介面中出現的值。

**可以更新：**是

**可能值：**英數字元、空格和底線。

**更新效果：**度量會在度量系列 UI 和自訂報告中反映更新後的 AttributeDisplayName。

**更新的生效時間：**立即

**更新生效所需執行的動作：**無

#### **文件**

在使用者介面中顯示屬性說明。當您將游標移至屬性名稱上方時，文件也會顯示在工具提示中。

**可以更新：**是

**可能值：**純文字

**更新效果：**將游標移至屬性名稱上方可顯示更新後的文件。

**更新的生效時間：**立即

**更新生效所需執行的動作：**無

**Polled**

指出是否輪詢屬性。如果設定為 `false`，則只有在搜索期間存取屬性。

**可以更新：**是

**可能值：**`true`、`false`

**更新效果：**如果設定為 `false`，在沒有其他輪詢的屬性/度量在其運算式中使用此屬性/度量的對應 OID 時，則不會輪詢該 OID，如果設定為 `true`，則會輪詢此屬性/度量的對應 OID。

**更新的生效時間：**下一次輪詢

**更新生效所需執行的動作：**無

**IsDbColumn**

將其值儲存於資料庫表格中。`IsDbColumn` 使用於度量屬性。在 `Polled` 設定為 `true` 時，將 `IsDbColumn` 值設定為 `true`。

**可以更新：**是

**可能值：**`true`、`false`

**更新效果：**如果設定為 `false`，此屬性/度量的資料不會儲存在資料庫中。如果設定為 `true`，此屬性/度量的資料就會儲存在資料庫中。

**更新的生效時間：**立即

**更新生效所需執行的動作：**無

## 搜索屬性

對於大多數屬性，只有在搜索期間擷取的值會儲存在資料庫中。不會執行進一步的輪詢或處理，例如基準的評估。

所有度量系列都必須有 `Indexes` 和 `Names` 屬性。`Descriptions` 屬性是選用的。

```
<Attribute name="Indexes" type="ObjectID[]" />
<Attribute name="Names" type="String" />
<Attribute name="Descriptions" type="String" />
```

支援階層的度量系列必須包含下列屬性：

```
<Attribute name="ItemUniqueIDs" type="String" />
<Attribute name="ParentUniqueIDs" type="String" />
```

## 輪詢和基準屬性

下列資訊說明輪詢和基準屬性元素：

### 基準

指出是否計算此屬性的平均值。如果設定為 true，必須定義對應的 BaselineList 定義。

**附註：**Baseline 屬性要求將 StandardDeviation 屬性設定為 true。

**可以更新：**是

**可能值：**true、false

**更新效果：**值為 true 時會計算基準值。

**更新的生效時間：**下一次輪詢

**更新生效所需執行的動作：**無

### 最大值

指出是否在彙總期間計算此屬性的上限。在資料庫表格中建立「max\_」欄。若已定義 RollupStrategy，也必須定義此屬性。

**可以更新：**是

**可能值：**true、false

**更新效果：**值為 true 時可提供「最大值」的計算及其報告欄位。

**更新的生效時間：**下一次輪詢

**更新生效所需執行的動作：**無

### 最小值

指出是否在彙總期間計算此屬性的下限。在資料庫表格中建立「min\_」欄。若已定義 RollupStrategy，也必須定義此屬性。

**可以更新：**是

**可能值：**true、false

**更新效果：**值為 true 時可提供「最小值」的計算及其報告欄位。

**更新的生效時間：**下一次輪詢

**更新生效所需執行的動作：**無

**StandardDeviation**

指出是否在彙總期間計算此屬性的標準差。在資料庫表格中建立「std\_」欄。若已定義 RollupStrategy，也必須定義此屬性。

**可以更新：**是

**可能值：**true、false

**更新效果：**值為 true 時可提供「標準差」的計算及其報告欄位。

**更新的生效時間：**下一次輪詢

**更新生效所需執行的動作：**無

**DeviationFromBaseline**

要求將 Baseline 屬性設定為 true。提供額外兩個使用基準資料計算的報告欄位：「平均基準」和「誤差百分比」。這些欄位不適用於建立自訂檢視。資料庫表格不會變更。

**可以更新：**是

**可能值：**true、false

**更新效果：**值為 true 時可提供「平均基準」和「誤差百分比」欄位，以便開發內部報告。

**更新的生效時間：**立即

**更新生效所需執行的動作：**無

**百分位數**

指出是否在彙總期間計算此屬性的第 95 個百分位數。在資料庫表格中建立「pct\_」欄。若已定義 RollupStrategy，也必須定義此屬性。

**可以更新：**是

**可能值：**0 (零)、95

**更新效果：**值為 95 可提供「第 95 個百分位數」的計算及其報告欄位。值為零表示不執行計算，且無法使用此報告欄位。

**更新的生效時間：**下一次輪詢

**更新生效所需執行的動作：**無

### **RollupStrategy**

指定在個別輪詢值的彙總期間內各個週期執行的作業。如果 **Polled** 和 **IsDbColumn** 均設為 **true**，則需要此元素。

**可以更新**：是

**可能值**：Sum (計數器的總和)、Avg (量測計的平均值)

**更新效果**：指定的策略用於執行彙總計算。

**更新的生效時間**：下一次輪詢

**更新生效所需執行的動作**：無

### **Rate**

提供額外一個報告欄位「平均速率」，該欄位會計算為 AVG (度量值 / 時間)。資料庫表格不會變更。

**備註**：Rate 適用於報告，但不能在監控設定檔事件規則時使用。

**可以更新**：是

**可能值**：true、false

**更新效果**：提供「平均速率」欄位以供報告。

**更新的生效時間**：立即

**更新生效所需執行的動作**：無

### **Units**

指定報告中使用的 units 標籤名稱。顯示的實際標籤會根據報告的語言設定進行轉譯。

**可以更新**：是

**可能值**：Percent、Packets、PacketsPerSecond、DiscardedPackets、DiscardedPacketsPerSecond、ErroredPackets、ErroredPacketsPerSecond、Bits、BitsPerSecond、Bytes、BytesPerSecond、Microseconds、Milliseconds、UnixTime

**更新效果**：指定的 units 標籤會顯示在報告中。

**更新的生效時間**：立即

**更新生效所需執行的動作**：無

## 範例：使用輪詢和基準屬性元素

```

<Attribute name="Utilization" type="double">
  <AttributeDisplayName>Utilization</AttributeDisplayName>
  <AttributeAbbreviation>Utilization</AttributeAbbreviation>
  <IsDbColumn>true</IsDbColumn>
  <Baseline>true</Baseline>
  <Minimum>true</Minimum>
  <Maximum>true</Maximum>
  <RollupStrategy>Avg</RollupStrategy>
  <StandardDeviation>true</StandardDeviation>
  <DeviationFromBaseline>true</DeviationFromBaseline>
  <Percentile>95</Percentile>
  <Polled>true</Polled>
  <Units>Percent</Units>
</Attribute>

```

## BaselineDefinitions

BaselineDefinitions 區段包含要對於此度量系列計算的基準定義。必須在 AttributeGroup 區段中針對每個度量指定基準定義，而其 Baseline 內容設定為 true。

您可以定義兩個基準類型：每小時 (必要) 和每日 (選用)。每小時基準用於處理事件以及在報告中顯示基準。每日基準用於在時間範圍等於或大於一個月的報告中顯示基準。

下列資訊說明使用的基準元素：

### 名稱

指定度量的基準定義類型。此類型為每小時或每天。

可以更新：否

可能值：HourlyBaseline、DailyBaseline

### ID

指定不再使用的值。不過，必須將此欄位指定為正整數，而且在此度量系列的所有每小時和每日基準定義中必須是唯一的。

可以更新：是

可能值：任何唯一的正整數。

更新效果：無

### **PerformanceMetric**

指定計算基準的度量名稱 (區分大小寫)。將度量屬性的 Polled 和 Baseline 內容設定為 true。

**可以更新：**是

**可能值：**有效的度量名稱 (區分大小寫)。

**更新效果：**執行度量的基準計算。

**更新的生效時間：**下一次基準計算 (每小時或每日)。

**更新生效所需執行的動作：**無

### **期間**

指定基準計算類型，亦即每小時或每日基準。將 HourlyBaseline 名稱的值指定為「1 小時」，將 DailyBaseline 名稱的值指定為「1 天」。

**可以更新：**是

**可能值：**1 小時、1 天

**更新效果：**每小時或每天執行基準計算。

**更新的生效時間：**下一次基準計算 (每小時或每日)。

**更新生效所需執行的動作：**無

### **時間範圍**

指定不再使用的值。不過，必須分別針對每小時基準和每日基準將此欄位指定為「30 天」和「90 天」。

**可以更新：**否

**可能值：**30 天、90 天

### **StartDate、EndDate、DaysOfWeek**

指定更多未使用的值，但必須將其指定為 0 (零)。

**可以更新：**否

**可能值：**0

## Expressions

Expressions 區段是由用於元件搜索的 ExpressionGroup 標記組成。在元件搜索期間，會計算元件項目內容 (例如 `IndexList`、`Name` 和 `Description`) 的值。此計算會使用支援度量系列運算式的廠商憑證運算式。

**附註：**請勿混淆度量系列與廠商憑證 ExpressionGroup 標記。

下列 DestCert URI 的 ExpressionGroup 標記必須存在：

DestCert	DestAttr
{http://im.ca.com/core}Item	名稱
{http://im.ca.com/core}Item	說明
{http://im.ca.com/inventory} DeviceComponent	IndexList

### ExpressionGroup/name

(選用) 指定運算式群組名稱。

**可以更新：**是

**可能值：**純文字

**更新效果：**無

### ExpressionGroup/destCert

指定包含要填入之 destAttrs 的元件 Facet。除了 Item 和 DeviceComponent Facet 以外，Facet 名稱通常來自 ComponentFacets 區段。

**可以更新：**是

**可能值：**定義於 ComponentFacets 中的 Facet，或 Item、DeviceComponent Facet。

**更新效果：**變更可允許的運算式 destAttr

**更新的生效時間：**元件重新搜索

**更新生效所需執行的動作：**無

#### **ExpressionGroup/Expression**

指定元件 Facet 屬性的運算式。

**可以更新**：是

**可能值**：任何有效的度量

**更新效果**：變更可允許的運算式 destAttr

**更新的生效時間**：元件重新搜索

**更新生效所需執行的動作**：無

#### **ExpressionGroup/Expression/destAttr**

指定元件 Facet 屬性名稱。

**可以更新**：是

**可能值**：該元件 Facet 中任何有效的屬性。

**更新效果**：變更屬性名稱

**更新的生效時間**：元件重新搜索

**更新生效所需執行的動作**：無

## 階層

在不同度量系列 (例如 Interface 和 CBQoS 類別對映) 的項目之間，可以定義階層或上層-子系關係。在度量系列定義中，必須使用下列各項在子度量系列中指定 Hierarchy：

- ComponentFacets 中的 Hierarchy QName
- Hierarchy ExpressionGroup 中的 ItemUniqueId 和 ParentUniqueId destAttr 值
- AttributeGroup 中的 ItemUniqueIDs 和 ParentUniqueIDs 屬性

支援運算式[定義於廠商憑證](#) (位於 p. 75) 中。

下列 DestCert URI 的 Hierarchy ExpressionGroup 標記必須存在：

<b>DestCert</b>	<b>DestAttr</b>
{http://im.ca.com/inventory}Hierarchy	ItemUniqueId

DestCert	DestAttr
{http://im.ca.com/inventory}Hierarchy y	ParentUniqueID

### 範例：

```

<ComponentFacets>
    <Facet>{http://im.ca.com/inventory}QoSClassMap</Facet>
    <Facet>{http://im.ca.com/inventory}Hierarchy</Facet>
</ComponentFacets>

<ExpressionGroup name="Hierarchy"
destCert="{http://im.ca.com/inventory}Hierarchy">
    <Expression destAttr="ItemUniqueID">ItemUniqueIDs</Expression>
    <Expression destAttr="ParentUniqueID">ParentUniqueIDs</Expression>
</ExpressionGroup>

<AttributeGroup name="QosCosGroup" list="true" external="true">
    <Attribute name="ItemUniqueIDs" type="String" />
    <Attribute name="ParentUniqueIDs" type="String" />
    ...
</AttributeGroup>

```

### 更多資訊：

[ComponentFacets](#) (位於 p. 35)

## ComponentReconciliation

下列資訊定義元件搜索中使用的元件重新調整邏輯。首先，此資訊可判斷系統是否已搜索此元件。然後，重新調整邏輯會決定要更新現有元件還是建立新元件。

```

<ComponentReconciliation>
    <MatchAlgorithms>
        <ExactMatch>
            <MatchAttribute name="{http://im.ca.com/inventory}Port.Type"/>
            <MatchAttribute name="{http://im.ca.com/core}Item.Description"/>
        </ExactMatch>
        <BestOfMatch leastMatchCount="3">
            <MatchAttribute name="{http://im.ca.com/inventory}Port.Type" required="true"/>
            <MatchAttribute name="{http://im.ca.com/inventory}Port.Alias"/>
            <MatchAttribute name="{http://im.ca.com/core}Item.Description"/>
            <MatchAttribute name="{http://im.ca.com/inventory}Port.MACAddress"/>
            <MatchAttribute name="{http://im.ca.com/inventory}DeviceComponent.IndexList"/>
        </BestOfMatch>
    </MatchAlgorithms>
</ComponentReconciliation>

```

每一度量系列可以有多個已排序的演算法。如果度量系列未定義重新調整演算法，則會套用具有比對屬性 `Item.Name` 的預設重新調整演算法。

**更多資訊：**

[ComponentFacets](#) (位於 p. 35)  
[MatchAlgorithms](#) (位於 p. 49)

## ItemReconciliation

**重要！** `ItemReconciliation` 是一個有可能變更的新區段，以支援未來複雜的度量系列結構。不建議使用該區段。

下列資訊定義項目搜索中使用的元件重新調整邏輯。此邏輯可判斷系統是否已搜索項目。根據這項判斷，更新現有項目或建立新項目。項目重新調整類似於元件重新調整。不過，項目重新調整使用於不是元件的項目，例如虛擬主機。`ItemFacets` 會新增到任何新項目或任何相符項目 (如果 `Facet` 不存在的話)。

**範例：**

```
<ItemReconciliation>
  <SourceAgentScopedReconciliation>
    <MatchAlgorithms>
      <ExactMatch>
        <MatchAttribute
          name="http://im.ca.com/inventory}SourceAgentInfo.SourceAgentIndexes" />
      </ExactMatch>
    </MatchAlgorithms>
  </SourceAgentScopedReconciliation>
  <GlobalScopedReconciliation matchDevices="true" />
</ItemReconciliation>
```

### SourceAgentScopedReconciliation

定義用於重新調整項目的比對演算法。

**可以更新：**是

**更新效果：**變更項目重新調整邏輯。

**更新的生效時間：**重新搜索

**更新生效所需執行的動作：**更新度量系列或變更廠商憑證優先順序。

### GlobalScopedReconciliation

定義無法針對來源代理程式重新調整項目時使用的比對演算法。

**GlobalScopedReconciliation** 演算法用於尋找針對其他代理程式建立,但符合潛在新項目的項目。如果 `matchDevices` 內容設定為 `true`, 則會使用系統預設 (內建, 未顯示在 XML 中) `ComponentReconciliationInfo` 比對演算法。比對演算法是以裝置主要 IP 位址和主機名稱為基礎。

**可以更新**：是

**更新效果**：變更項目重新調整邏輯。

**更新的生效時間**：重新搜索

**更新生效所需執行的動作**：更新度量系列或變更廠商憑證優先順序。

**更多資訊**：

[ItemFacets](#) (位於 p. 35)

[MatchAlgorithms](#) (位於 p. 49)

## MatchAlgorithms

元件重新調整和項目重新調整會使用比對演算法。支援以下兩個比對演算法：

- **ExactMatch**—必須符合所有指定的屬性，才能判定項目與新資料相符。
- **BestOfMatch**—使用者必須使用 "`leastMatchCount`" 值，指定要符合的屬性數目下限。此外，每個屬性都有「必要的」重要內容。如果必要的內容設定為 `true`，則必須符合該屬性才可視為相符項目。

針對一個度量系列提供多個演算法時，此演算法具有比對優先順序。演算法的順序可決定優先順序。最上方的演算法具有最高的優先順序。最下方的演算法具有最低的優先順序。

每個演算法都必須至少有一個相符屬性。當資料符合多個具有相同演算法的項目時，則具有最多相符屬性的項目勝出。當多個相符項目有同樣多的相符屬性時，則會從這些項目中隨機挑選勝出者。

## 範例：重新調整的運作方式

針對一個度量系列提供兩個比對演算法：alg1 和 alg2。Alg1 的優先順序高於 alg2。此度量系列有三個現有的元件項目：1、2 和 3。重新搜尋度量系列可找到三個項目：A、B 和 C。我們現在套用兩個演算法，以判斷哪個項目是新的、已變更和未變更。

重新調整中繼資料	新資料	現有元件
<ComponentReconciliation>		
<MatchAlgorithms>	A	1
<MatchAlgorithm1>	B	2
<MatchAttribute name="attr1"/>	C	3
<MatchAttribute name="attr2"/>		
</MatchAlgorithm1>		
<MatchAlgorithm2>		
<MatchAttribute name="attr1"/>		
<MatchAttribute name="attr3"/>		
<MatchAttribute name="attr4"/>		
</MatchAlgorithm2>		
</MatchAlgorithms>		
</ComponentReconciliation>		

<MatchAlgorithm1> 和 <MatchAlgorithm2> 可以是 <ExactMatch> 或 <BestOfMatch>。這兩個比對演算法的順序顯示 MatchAlgorithm1 的優先順序高於 MatchAlgorithm2。

### 案例 1：唯一 1 對 1 相符項目

項目 A 符合項目 1，而項目 1 沒有任何其他相符項目。

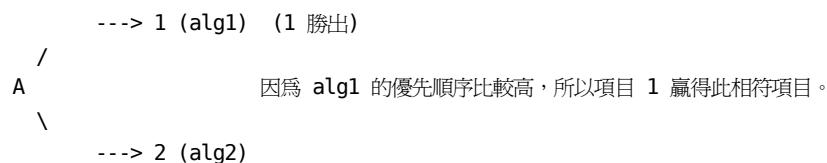
A -----> 1

此範例是最簡單的案例。此相符項目是唯一的，所以它是否也符合 alg1 或 alg2 並不構成影響。項目 A 符合項目 1。

良好的比對演算法可產生更多個唯一相符項目。

### 案例 2：一個項目有多個相符項目

項目 A 符合項目 1 (依據 alg1)，而且也符合 2 (依據 alg2)。



### 案例 3：多個項目符合具有不同演算法的相同項目

項目 A 符合 1 (依據 alg1)，而項目 B 也符合項目 1 (依據 alg2)。

A -----> 1 (alg1) (A 勝出)  
 B -----> 1 (alg2)

因為 alg1 的優先順序比較高，所以項目 A 勝出。

### 案例 4：多個項目符合具有相同演算法的相同項目，但符合的屬性數目不同

A 和 B 均符合 1 (依據 alg1)。

A -----> 1 (alg1, 相符屬性數目：2) (A 勝出)  
 B -----> 1 (alg1, 相符屬性數目：1)

因為 A 有比較多的相符屬性，所以 A 勝出。

如果相符屬性數目一樣，則會隨機挑選勝出者並產生警告。

### 案例 5：混合比對 1

```

alg1
A -----> 1
 / alg2(相符屬性計數：3)
B
 \ alg2(相符屬性計數：2)
 -----> 2
  
```

A 符合 1，因為這是以較高優先順序的演算法進行比對。

B 符合 2，因為 1 已符合 A。

### 案例 6：混合比對 2

```

-----> 3
 / alg1
A           ==> A 崴得 3，因為 alg1 的比對優先順序比較高
 \ alg2
-----> 1
 / alg2
B           ==> B 崴得 2，因為 alg1 的比對優先順序比較高
 \ alg1
-----> 2
 / alg2
C           ==> C 沒有相符項目，因為 2 已符合 B，而 3 已符合 A
 \ alg2
-----> 3
  
```

項目 C 被視為新元件。1 被視為不符的項目。

比對案例 1 (唯一相符) 越多，比對演算法就越好。

可以更新：是

更新效果：變更元件重新調整邏輯。

更新的生效時間：重新搜索

更新生效所需執行的動作：更新度量系列或變更廠商憑證優先順序。

更多資訊：

[ComponentReconciliation](#) (位於 p. 47)

[ItemReconciliation](#) (位於 p. 48)

## ReconfigDetectionAttr

此元素定義用於進行變更偵測的度量系列屬性。您可以在監控設定檔上啓用變更偵測。Data Aggregator 只會輪詢該屬性，以檢查目標裝置是否已變更，而非進行完整的重新搜索。此功能對效能有幫助，並可減少網路流量。

可以更新：是

可能值：度量系列屬性的完整名稱。指定的度量系列屬性旗標應將 cached、persistent 和 external 旗標設定為 true。

更新效果：變更為元件重新配置偵測。

更新的生效時間：在重新搜索之後。

更新生效所需執行的動作：更新度量系列或變更廠商憑證優先順序。

## 自訂度量系列不支援的內容

自訂度量系列不支援下列內容：

- 差異
- RollupExpression

**重要！**如果以匯出的原廠度量系列做為範本，請移除所有不支援的元素，或在自訂度量系列中將其設定為 false。

## 匯入自訂度量系列

若要讓自訂度量系列生效，請將其匯入 Data Aggregator 中。

CA Performance Center 尚未提供使用者介面來直接支援匯入以 IMDBCertificationFacet.xsd 結構描述為基準的度量系列。因此，運用 Data Aggregator 公開的 Web 服務來完成此工作。

使用任何 REST 用戶端。建議搭配使用 RESTClient 與 CA Performance Management。如果尚未安裝，可以從 <http://code.google.com/p/rest-client/> 取得 RESTClient GUI JAR 檔案。

使用 RESTClient 時，請考量下列資訊：

- 若要加以啟動，請連接兩下 JAR 檔案。
- 當您 POST XML 時，確定 Charset 已設定為 UTF-8。  
若要檢視並確認此設定，請按一下 [編輯 Content-Type 和 Charset] 按鈕。
- 您也可以使回應內文自動縮排，如下所述：
  1. 按一下 [工具] 功能表上的 [選項]。
  2. 選取 [Etc.] 索引標籤。
  3. 選取 [使回應內文自動縮排] 並按一下 [確定]。

請依循下列步驟：

1. 輸入 `http://da-hostname:8581/typedefatalog/metricfamilies` 做為 URL。
2. 選取 POST 做為方法。
3. 選取「application/xml」做為 Body 設定中的「Body Content-type」。  
**重要！無法設定 Content-type 會造成 404 錯誤。**
4. 將自訂度量系列 XML 複製並貼到 [內文] 索引標籤中。
5. 執行此方法。

自訂度量系列隨即匯入。如果未發生錯誤，[HTTP 回應] 區段中的 [狀態] 欄位隨即顯示：

HTTP/1.1 200 OK

**附註：**有其他傳回碼指出在更新自訂度量系列時發生錯誤。修正此錯誤並執行其他 POST 來重試更新度量系列。

**重要！** 為了避免資料遺失，每次建立或更新廠商憑證、度量系列或元件時，請務必備份您的部署目錄。

## 驗證自訂度量系列結果

匯入自訂度量系列 XML 之後，請驗證結果。在我們的範例中，我們會驗證框架轉送度量系列是否匯入成功。

請依循下列步驟：

1. 在 REST 用戶端中輸入下列 URL：

```
http://da_hostname:8581/typecatalog/metricfamilies/name
```

*name*

自訂度量系列的名稱。在此範例中，這是 FrameRelay。

頁面上會顯示自訂框架轉送度量系列 XML。

您也可以驗證自訂度量系列是否已列在 Data Aggregator 資料來源的 [監控配置] 功能表上的 [度量系列] 索引標籤。

## 其他度量支援

有些時候，您會想將其他度量新增至現有的自訂度量系列。延續我們的範例，您想要新增 Bits In 和 Bits Out 度量。

以下範例顯示更新後的 frPVCInfo 自訂度量系列外觀。在此程序中進行的變更會以粗體表示：

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Auto-generated by the type catalog local manager.-->
<DataModel namespace="http://im.ca.com/normalizer"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="IMDBCertificationFacet.xsd">
  <FacetType name="frPVCInfo"
    descriptorClass="com.ca.im.core.datamodel.certs.NormalizedFacetDescriptorImpl">
    <Documentation>Frame Relay Permanent Virtual
    Circuit</Documentation>
    <FacetOf namespace="http://im.ca.com/core" name="Item" />
    <AttributeGroup name="AttributeGroup" external="true"
      list="true">
      <Documentation />
      <Attribute name="Indexes" type="ObjectID[]">
        <Documentation />
        <Polled>false</Polled>
        <Baseline>false</Baseline>
        <IsDbColumn>false</IsDbColumn>
        <Variance>false</Variance>
        <StandardDeviation>false</StandardDeviation>
        <Minimum>false</Minimum>
        <Maximum>false</Maximum>
        <WriteOnPoll>false</WriteOnPoll>
        <RollupStrategy />
        <AttributeDisplayName />
        <Percentile>0</Percentile>
      </Attribute>
      <Attribute name="Names" type="String">
        <Documentation>The name of the frame relay
        circuit</Documentation>
        <Polled>false</Polled>
        <Baseline>false</Baseline>
        <IsDbColumn>false</IsDbColumn>
        <Variance>false</Variance>
        <StandardDeviation>false</StandardDeviation>
        <Minimum>false</Minimum>
        <Maximum>false</Maximum>
        <WriteOnPoll>false</WriteOnPoll>
        <RollupStrategy />
        <AttributeDisplayName />
        <Percentile>0</Percentile>
      </Attribute>
      <Attribute name="Description" type="String">
        <Documentation>A description for the frame relay
        circuit</Documentation>
```

```
<Polled>false</Polled>
<Baseline>false</Baseline>
<IsDbColumn>false</IsDbColumn>
<Variance>false</Variance>
<StandardDeviation>false</StandardDeviation>
<Minimum>false</Minimum>
<Maximum>false</Maximum>
<WriteOnPoll>false</WriteOnPoll>
<RollupStrategy />
<AttributeDisplayName />
<Percentile>0</Percentile>
</Attribute>
<Attribute name="BECNIn" type="Double">
    <Documentation>Backward congestion since the virtual
circuit was created</Documentation>
    <Polled>true</Polled>
    <Baseline>false</Baseline>
    <IsDbColumn>true</IsDbColumn>
    <Variance>false</Variance>
    <StandardDeviation>false</StandardDeviation>
    <Minimum>false</Minimum>
    <Maximum>false</Maximum>
    <WriteOnPoll>false</WriteOnPoll>
    <RollupStrategy>Sum</RollupStrategy>
    <AttributeDisplayName />
    <Percentile>0</Percentile>
</Attribute>
<Attribute name="FECNIn" type="Double">
    <Documentation>Forward congestion since the virtual
circuit was created</Documentation>
    <Polled>true</Polled>
    <Baseline>false</Baseline>
    <IsDbColumn>true</IsDbColumn>
    <Variance>false</Variance>
    <StandardDeviation>false</StandardDeviation>
    <Minimum>false</Minimum>
    <Maximum>false</Maximum>
    <WriteOnPoll>false</WriteOnPoll>
    <RollupStrategy>Sum</RollupStrategy>
    <AttributeDisplayName />
    <Percentile>0</Percentile>
</Attribute>
<Attribute name="FramesIn" type="Double">
    <Documentation>Frames received since the virtual circuit
was created</Documentation>
    <Polled>true</Polled>
    <Baseline>false</Baseline>
    <IsDbColumn>true</IsDbColumn>
    <Variance>false</Variance>
```

```
<StandardDeviation>false</StandardDeviation>
<Minimum>false</Minimum>
<Maximum>false</Maximum>
<WriteOnPoll>false</WriteOnPoll>
<RollupStrategy>Sum</RollupStrategy>
<AttributeDisplayName />
<Percentile>0</Percentile>
</Attribute>
<Attribute name="FramesOut" type="Double">
<Documentation>Frames sent since the virtual circuit was
created</Documentation>
<Polled>true</Polled>
<Baseline>false</Baseline>
<IsDbColumn>true</IsDbColumn>
<Variance>false</Variance>
<StandardDeviation>false</StandardDeviation>
<Minimum>false</Minimum>
<Maximum>false</Maximum>
<WriteOnPoll>false</WriteOnPoll>
<RollupStrategy>Sum</RollupStrategy>
<AttributeDisplayName />
<Percentile>0</Percentile>
</Attribute>
<Attribute name="BytesIn" type="Double">
<Documentation>Bytes received since the virtual circuit
was created</Documentation>
<Polled>true</Polled>
<Baseline>false</Baseline>
<IsDbColumn>true</IsDbColumn>
<Variance>false</Variance>
<StandardDeviation>false</StandardDeviation>
<Minimum>false</Minimum>
<Maximum>false</Maximum>
<WriteOnPoll>false</WriteOnPoll>
<RollupStrategy>Sum</RollupStrategy>
<AttributeDisplayName />
<Percentile>0</Percentile>
</Attribute>
<Attribute name="BytesOut" type="Double">
<Documentation>Bytes sent since the virtual circuit was
created</Documentation>
<Polled>true</Polled>
<Baseline>false</Baseline>
<IsDbColumn>true</IsDbColumn>
<Variance>false</Variance>
<StandardDeviation>false</StandardDeviation>
<Minimum>false</Minimum>
<Maximum>false</Maximum>
<WriteOnPoll>false</WriteOnPoll>
```

```
        <RollupStrategy>Sum</RollupStrategy>
        <AttributeDisplayName />
        <Percentile>0</Percentile>
    </Attribute>
    <Attribute name="BitsIn" type="Double">
        <Documentation>Bits received since the virtual circuit
was created</Documentation>
        <Polled>true</Polled>
        <Baseline>false</Baseline>
        <IsDbColumn>true</IsDbColumn>
        <Variance>false</Variance>
        <StandardDeviation>false</StandardDeviation>
        <Minimum>false</Minimum>
        <Maximum>false</Maximum>
        <WriteOnPoll>false</WriteOnPoll>
        <RollupStrategy>Sum</RollupStrategy>
        <AttributeDisplayName />
        <Percentile>0</Percentile>
    </Attribute>
    <Attribute name="BitsOut" type="Double">
        <Documentation>Bits sent since the virtual circuit was
created</Documentation>
        <Polled>true</Polled>
        <Baseline>false</Baseline>
        <IsDbColumn>true</IsDbColumn>
        <Variance>false</Variance>
        <StandardDeviation>false</StandardDeviation>
        <Minimum>false</Minimum>
        <Maximum>false</Maximum>
        <WriteOnPoll>false</WriteOnPoll>
        <RollupStrategy>Sum</RollupStrategy>
        <AttributeDisplayName />
        <Percentile>0</Percentile>
    </Attribute>
</AttributeGroup>
<Attribute name="SourceFacetTypes" cached="true"
list="true" persistent="true" type="QName">
    <Documentation />
</Attribute>
<DisplayName>Frame Relay PVC</DisplayName>
<Expressions>
    <ExpressionGroup destCert="{http://im.ca.com/core}Item">
        <Expression destAttr="Name">Names</Expression>
    </ExpressionGroup>
    <ExpressionGroup>
        <Expression destAttr="IndexList">Indexes</Expression>
    </ExpressionGroup>
</Expressions>
```

```

<TableName>FR_PVC_INFO</TableName>
<ComponentFacets>
  <Facet>{http://im.ca.com/inventory}frPVC</Facet>
</ComponentFacets>
<Protocol>IMDB</Protocol>
<Normalized>true</Normalized>
</FacetType>
</DataModel>

```

我們假設在匯入自訂度量系列之後進行這些變更。因此，在您進行變更後，更新度量系列。更新度量系列之後，您還必須更新與此度量系列相關聯的廠商憑證。

**更多資訊：**

[更新自訂廠商憑證](#) (位於 p. 90)  
[更新自訂度量系列](#) (位於 p. 59)

## 更新自訂度量系列

您可以更新現有的自訂度量系列。在此範例中，您先前已在自訂度量系列 **frPVCInfo** 中新增度量。立即更新度量系列，讓變更生效。

**附註：**如需在更新自訂度量系列時更新標記或屬性之效果的詳細資訊，請檢閱自訂度量系列 XML 詳細資料。請特別檢閱特定屬性說明。

**請依循下列步驟：**

1. 在 URL 欄位中輸入下列網址：

`http://da_hostname:8581/typecatalog/metricfamilies/name  
名稱`

要更新的自訂度量系列名稱。在我們的範例中，度量系列的名稱為 **frPVCInfo**。

2. 若要確認度量系列位於該 URL，請選取 [方法] 索引標籤中的 GET。

**附註：**如果此度量系列不存在，您可以[匯入自訂度量系列](#) (位於 p. 53)。

3. 確認度量系列位於該 URL 之後，請選取 [方法] 索引標籤中的 PUT。

4. 將更新後的自訂度量系列 XML 複製並貼到 [內文] 索引標籤 ([編輯] 欄位) 中，並將 Content-type 設定為 application/xml。

**重要！** 無法設定 Content-type 會造成 404 錯誤。

5. 按一下 [URL] 欄位旁邊的 [移至] 按鈕。

自訂度量系列隨即更新。如果未發生錯誤，[HTTP 回應] 區段中的 [狀態] 欄位會顯示下列訊息：

HTTP/1.1 200 OK

有其他傳回碼指出在更新自訂度量系列時發生錯誤。修正此錯誤並執行其他 PUT 來重試更新度量系列。

**重要！** 為了避免資料遺失，每次建立或更新廠商憑證、度量系列或元件時，請務必備份您的部署目錄。

6. [接著，更新與此度量系列相關聯的自訂廠商憑證](#) (位於 p. 90)。

**更多資訊：**

[瞭解度量系列 XML 結構](#) (位於 p. 29)

[其他度量支援](#) (位於 p. 54)

# 第 5 章：建立自訂廠商憑證

---

匯入成功時，新的廠商憑證會自動新增到指定度量系列的優先順序清單末端。您可以修改此優先順序清單，視需要提高您的廠商憑證的優先順序。搜尋指定度量系列的裝置支援時會使用廠商憑證。

**附註：**Data Aggregator 提供了建立自訂廠商憑證與度量系列的基本與進階方法。基本方法的程序較為簡單，其中包含為現有支援技術 (度量系列) 增加廠商支援、使用使用者介面。此方法可滿足許多使用者的需求。進階方法則以原廠憑證格式為基礎，且提供一組完整的功能。本指南將說明進階憑證方法。如需基本憑證方法的詳細資訊，請參閱《*Data Aggregator 自我憑證指南*》。

本節包含以下主題：

- [建立廠商憑證 XML 範本](#) (位於 p. 61)
- [瞭解廠商憑證 XML 結構](#) (位於 p. 62)
- [匯入自訂廠商憑證](#) (位於 p. 77)
- [驗證自訂廠商憑證結果](#) (位於 p. 80)
- [篩選支援](#) (位於 p. 81)
- [多重 MIB 表格支援](#) (位於 p. 84)
- [更新自訂廠商憑證](#) (位於 p. 90)

## 建立廠商憑證 XML 範本

使用慣用的 REST 用戶端來建立範例 XML 檔案，以便做為建立自訂廠商憑證時的範本。

從擷取現有的廠商憑證清單開始。然後可以驗證是否已支援您所需的廠商憑證。

**請依循下列步驟：**

1. 透過與 Data Aggregator 伺服器的連線來設定 REST 用戶端。
2. 針對 REST 用戶端中的 Data Aggregator Web 服務輸入下列 URL。  
`http://da_hostname:8581/typecatalog/certifications/snmp`  
所有可用的廠商憑證清單隨即顯示。
3. 驗證是否已支援您所需的廠商憑證。

如果尚未支援您所需的廠商憑證，請檢視及匯出類似您要建立之自訂廠商憑證的廠商憑證。

**請依循下列步驟：**

1. 輸入下列 URL，以擷取類似您所需度量系列的特定廠商憑證：

`http://da_hostname:8581/typecatalog/certifications/snmp/name`

廠商憑證的名稱，例如 CiscoCPUMibRev。

2. 選取 [方法] 索引標籤中的 GET。
3. 執行此方法。

傳回的 XML 包含廠商憑證資訊。

您可以使用此 XML 做為建立自訂廠商憑證時的範本。

4. 將廠商憑證 XML 複製到文字檔中，然後視需要進行修改。如需 XML 結構的範例，請參閱瞭解廠商憑證 XML (位於 p. 62)。

## 瞭解廠商憑證 XML 結構

廠商憑證會將廠商和裝置專用資料對應至度量系列中定義的效能度量及配置資料。將不同來源的這些資料對應至「標準化」度量系列值，有助於 Data Aggregator 對於所有裝置廠商統一報告這些資料。

以下是自訂廠商憑證 XML 範例可支援框架轉送 PVC 的範例。請注意，範例自訂度量系列 frPVCInfo 如何納入 ExpressionGroup 區段中 (為了舉例，以粗體顯示)：

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Auto-generated by the type catalog local manager.-->
<DataModel namespace="http://im.ca.com/certifications/snmp"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="SNMPCertificationFacet.xsd">
  <FacetType name="frPVCInfoCustom"
  descriptorClass="com.ca.im.core.datamodel.certs.CertificationFacet
  DescriptorImpl">
    <Documentation>Frame Relay PVC Vendor
    Certification</Documentation>
    <FacetOf namespace="http://im.ca.com/core" name="Item" />
    <AttributeGroup name="AttributeGroup" external="true"
    list="true">
      <Documentation />
      <Attribute name="INDEX" type="ObjectID">
        <Documentation />
        <Source>1.3.6.1.2.1.10.32.2.1.4</Source>
        <IsIndex>true</IsIndex>
        <IsKey>false</IsKey>
        <NeedsDelta>false</NeedsDelta>
      </Attribute>
      <Attribute name="frCircuitReceivedBECNs" type="Long">
        <Documentation />
        <Source>1.3.6.1.2.1.10.32.2.1.5</Source>
        <IsIndex>false</IsIndex>
        <IsKey>true</IsKey>
        <NeedsDelta>true</NeedsDelta>
      </Attribute>
      <Attribute name="frCircuitSentFrames" type="Long">
        <Documentation />
        <Source>1.3.6.1.2.1.10.32.2.1.6</Source>
        <IsIndex>false</IsIndex>
        <IsKey>true</IsKey>
        <NeedsDelta>true</NeedsDelta>
      </Attribute>
      <Attribute name="frCircuitSentOctets" type="Long">
        <Documentation />
        <Source>1.3.6.1.2.1.10.32.2.1.6</Source>
        <IsIndex>false</IsIndex>
        <IsKey>true</IsKey>
        <NeedsDelta>true</NeedsDelta>
      </Attribute>
      <Attribute name="frCircuitReceivedFrames" type="Long">
        <Documentation />
        <Source>1.3.6.1.2.1.10.32.2.1.8</Source>
```

```
<IsIndex>false</IsIndex>
<IsKey>true</IsKey>
<NeedsDelta>true</NeedsDelta>
</Attribute>
<Attribute name="frCircuitReceivedOctets" type="Long">
<Documentation />
<Source>1.3.6.1.2.1.10.32.2.1.9</Source>
<IsIndex>false</IsIndex>
<IsKey>true</IsKey>
<NeedsDelta>true</NeedsDelta>
</Attribute>
</AttributeGroup>
<Protocol>SNMP</Protocol>
<DisplayName>Frame Relay PVC Certification</DisplayName>
<Expressions>
<ExpressionGroup
destCert="http://im.ca.com/normalizer}frPVCInfo"
name="frPVCInfoDS">
<Expression destAttr="Indexes">INDEX</Expression>
<Expression destAttr="Names">"Frame Relay " +
INDEX</Expression>
<Expression
destAttr="FECNIn">frCircuitReceivedFECNs</Expression>
<Expression
destAttr="BECNIn">frCircuitReceivedBECNs</Expression>
<Expression
destAttr="FramesIn">frCircuitReceivedFrames</Expression>
<Expression
destAttr="FramesOut">frCircuitSentFrames</Expression>
<Expression
destAttr="BytesIn">frCircuitReceivedOctets</Expression>
<Expression
destAttr="BytesOut">frCircuitSentOctets</Expression>
</ExpressionGroup>
</Expressions>
<MIB>RFC1315-MIB</MIB>
</FacetType>
</DataModel>
```

## 基本內容

自訂廠商憑證的基本內容，有助於與您建立的自訂廠商憑證相區別。此外，這些內容指出您從哪個廠商 MIB 收集度量資料。

決定基本內容時，請考量下列限制：

- 各廠商憑證的 `FacetType/name` 和 `FacetType/DisplayName` properties 內容必須是唯一的。
- `Protocol` 標記一律為 SNMP。
- 設定 `FacetType/descriptorClass` 內容以及所有的 `DataModel` 和 `FacetOf` 內容，如先前圖例中的範例 XML 所示。

#### **FacetType/name**

唯一識別廠商憑證。

**建議：**符合 "`<MibName><TableName>Mib`"。

**可以更新：**否

**可能值：**英數字元和底線。不允許使用點和破折號。

`FacetType` 區段顯示特定廠商憑證。當廠商憑證公開廠商專用裝置的各個層面 (例如 MIB-2 實作的 TCP 和 UDP 統計資料) 時，相同的 XML 文件可以包含多個 `FacetType` 區段。

`FacetType` 區段包含一些基本內容。例如，此區段包含廠商 MIB 的名稱，其後為一或多個 `AttributeGroup` 區段。這些 `AttributeGroup` 區段定義此憑證使用來自 MIB 的哪些屬性。最後，有一或多個 `ExpressionGroup` 區段可將 `AttributeGroup` 區段中的屬性對應到在度量系列中指定的度量。

#### **FacetType/Documentation**

說明使用廠商憑證認證的項目。

**建議：**納入有關廠商、MIB 名稱和表格名稱的詳細資料。

**可以更新：**是

**可能值：**純文字

**更新效果：**無

#### **FacetType/MIB**

指定 DEFINITIONS 子句在 ASN.1 檔案中定義的 MIB 名稱。

**建議：**符合 "<MibName>"

**可以更新：**是

**可能值：**純文字

**更新效果：**變更為系統管理員使用者介面的 [廠商憑證] 索引標籤中的 [SNMP MIB 名稱] 欄。

**更新的生效時間：**立即

**更新生效所需執行的動作：**重新整理使用者介面。

#### **FacetType/DisplayName**

指定 CA Performance Center 中顯示的廠商憑證名稱。

**建議：**開頭為廠商名稱並包含 MIB 和功能資訊。

**可以更新：**是

**可能值：**純文字

**更新效果：**變更為系統管理員使用者介面中的名稱。

**更新的生效時間：**立即

**更新生效所需執行的動作：**重新整理使用者介面。

### **AttributeGroup (廠商憑證)**

下列範例說明自訂廠商憑證的 AttributeGroup 區段。此區段可識別廠商 MIB 中用來對應原始裝置資料的特定表格屬性 (變數 OID)。此資料會對應至度量系列中定義的效能度量及配置資料。

您可將 AttributeGroup/list 和 AttributeGroup/external 內容設定為 true，如先前圖例中的範例 XML 所示。這些內容指定每個屬性都代表從外部來源 (MIB 表格) 取得的值清單。下列資訊摘要說明可自訂的 XML 元素。

#### **AttributeGroup/name**

指定屬性群組名稱。

**建議：**符合 "<FacetType/name>Group"。

**可以更新：**是

**可能值：**純文字

**更新效果：**無

#### **文件**

(選用) 指定屬性群組的說明。

**可以更新：**是

**可能值：**純文字

**更新效果：**無

#### **UseIndex**

指定要做為此屬性群組之索引的屬性名稱，以供聯結多個 MIB 表格。

**建議：**設定為 AttributeGroup/name 內容的值。

**可以更新：**是

**可能值：**純文字

**更新效果：**無

#### **更多資訊：**

[AttributeGroup \(多個 MIB 表格\)](#) (位於 p. 88)

[IndexTagList](#) (位於 p. 76)

## 一般屬性 (廠商憑證)

所有廠商憑證的一般屬性如下：

### Attribute/name

指定屬性名稱。

**建議：**設定為 OBJECT-TYPE 子句在 ASN.1 檔案中定義的 MIB 變數名稱。

**可以更新：**是

**可能值：**英數字元和底線。不允許使用點和破折號。

**更新效果：**您必須更新任何參照此屬性的運算式。

**更新的生效時間：**立即

**更新生效所需執行的動作：**無

### Attribute/type

指定屬性的資料類型。

**建議：**使用最符合 SYNTAX 子句在 ASN.1 檔案中定義之變數類型的屬性類型。

**可以更新：**是

**可能值：**Boolean、Int、Long、Double、BigInteger、String、DateTime、IPAddress、MACAddress、IPSubnet、OctetString、ObjectID

**更新效果：**輪詢的 SNMP 資料會轉換為此類型。

**更新的生效時間：**下一次輪詢

**更新生效所需執行的動作：**無

### 文件

(選用) 指定屬性的說明，其記載 MIB 變數的語意 (例如單位)。

**建議：**使用從 MIB ASN.1 檔案取得的說明。

**可以更新：**是

**可能值：**純文字

**更新效果：**無

**IsKey**

(選用) 使用旗標來指出 MIB 變數是否為決定表格支援的關鍵。若將多個欄位指定為關鍵，則所有欄位會一起被視為複合關鍵。

**預設值** : false

**建議** : 如果這是元件搜索的關鍵 MIB 物件，則設定為 true。

**可以更新** : 是

**可能值** : true、false

**更新效果** : 元件可變更為新的廠商憑證。

**更新的生效時間** : 在元件重新搜索之後。

**更新生效所需執行的動作** : 更新度量系列或變更廠商憑證優先順序。

**IsIndex**

(選用) 使用旗標來指出此變數是否為 MIB 表格的索引。

**預設值** : false

**建議** : 對於索引屬性設定為 true。

**可以更新** : 是

**可能值** : true、false

**更新效果** : 元件索引可能變更。

**更新的生效時間** : 在元件重新搜索之後。

**更新生效所需執行的動作** : 更新度量系列或變更廠商憑證優先順序。

**NeedsDelta**

(選用) 使用旗標來指出是否要區別(也就是，儲存這次和上次輪詢 Counters 之間的差異) MIB 變數。

**預設值** : false

**建議** : 如果此變數定義為 MIB 中的 Counter、Counter32、Counter64 或 TimeTicks 數量，則設定為 true。

**可以更新** : 是

**可能值** : true、false

**更新效果** : 輪詢的資料會變更。

**更新的生效時間** : 下一次輪詢

**更新生效所需執行的動作** : 無

## 來源

指定屬性的 ObjectId。

**建議：**設定為 OBJECT-TYPE 定義的完整 MIB 變數 OID。

**可以更新：**是

**可能值：**以點分隔的數字 (例如，1.3.6.1.4.1...)

**更新效果：**從指定的 OID 輪詢資料。

**更新的生效時間：**下一次輪詢

**更新生效所需執行的動作：**無

## 版本

指定廠商憑證的版本。如果在升級期間安裝廠商憑證，安裝程式會檢查憑證的版本。如果版本比安裝程式套件中的版本更新，安裝程式不會變更現有的廠商憑證。

**預設值：**1.0

**可以更新：**是

**可能值：**以點分隔的數字 (例如，1.3.6.1.4.1...)

**更新效果：**版本屬性已更新。

**更新的生效時間：**立即

**更新生效所需執行的動作：**無

## 作者

指定廠商憑證的建立者。

**預設值：**"Custom"

**可以更新：**是

**可能值：**任何英數字元字串。

**更新效果：**作者屬性已更新。

**更新的生效時間：**立即

**更新生效所需執行的動作：**無

屬性清單可指定此廠商憑證支援時度量系列所收集的資料集。通常，此資料分為兩類：

- 僅在搜索期間收集之裝置元件的配置資料 (例如名稱或索引)。
- 每個輪詢週期收集的效能資料。

更多資訊：

[多重 MIB 表格支援](#) (位於 p. 84)

[篩選支援](#) (位於 p. 81)

## 配置資料屬性

名稱為 INDEX 且類型為 ObjectId 的屬性會對應至目標度量系列的 Indexes 屬性。您可以將 Source 標記的值設定為任何變數 OID。不過，您通常會使用表格的 INDEX 子句中所列的其中一個變數。例如，請考量 MIB-2 的介面表格中的 ifIndex。此變數做為相同 MIB 表格中其他變數的索引。此外，此屬性的 IsIndex 標記 (通常還有 IsKey 標記) 已設定為 true。

在此範例中，ifDesc 或 ifType 等屬性會在介面上提供更多配置資訊。因此，這類屬性適合用於目標度量系列的 Names 和 Description 屬性。

## 效能資料屬性

這類屬性提供目標度量系列中效能度量的原始資料。請考量下列幾點：

- 您可以將其中一個屬性直接對應到度量系列效能度量，或者
- 可以在運算式中使用此屬性以及其他屬性來計算度量值。

## ExpressionGroup

ExpressionGroup 對應屬性的方式如下：

- 從 AttributeGroup (定義如何從 SNMP MIB 取得度量)
- 至度量系列中指定的度量 (定義如何在資料庫中定義屬性)

從裝置接收 MIB 值或在執行某些正規化作業之後，您可以在資料庫中儲存該 MIB 值。例如，正規化作業包含除以或乘以 1024 以便進行 KB 轉換。

### ExpressionGroup/name

(選用) 指定運算式群組名稱。

可以更新：是

可能值：純文字

更新效果：無

#### **ExpressionGroup/destCert**

指定包含要填入之 destAttrs 的度量系列。

**可以更新**：是

**可能值**：任何有效的度量系列

**更新效果**：變更可允許的運算式 destAttr。

**更新的生效時間**：立即

**更新生效所需執行的動作**：無

#### **ExpressionGroup/Filter**

(選用) 指定要搜索的元件。 使用 Filter 可減少受管理的元件數目。

**可以更新**：是

**可能值**：使用可用屬性的布林值 MVEL 運算式

**更新效果**：變更搜索的元件。

**更新的生效時間**：在元件重新搜索之後。

**更新生效所需執行的動作**：更新度量系列或變更廠商憑證優先順序。

### **ExpressionGroup 中的篩選器**

以下清單將說明篩選範例。

#### **範例 1：**

```
<Filter> hrStorageType.toString() == "1.3.6.1.2.1.25.2.1.4" &&
          hrStorageSize != 0
</Filter>
```

在此範例中，資料儲存庫中只會建立大小  $\neq 0$  且 storageType == hrStorageFixedDisk 的磁碟分割執行個體。

#### **範例 2：**

```
<Filter> (jnxOperatingCPUIndex.toString() == "9.2.0.0" ||
          jnxOperatingCPUIndex.toString() == "9.1.0.0" ) &&
          (jnxOperatingState != 1 || jnxOperatingCPU > 0 )
</Filter>
```

**範例 3：**

```
<Filter> (rttMonCtrlAdminRttType==9) && ( !(rttMonCtrlAdminOwner.toString() contains "Network Health") )
```

**範例 4：**

```
<Filter> (snmpOIDParser(sonetIndex,2,2).toString() == "1") </Filter>
```

此範例使用廠商憑證公用程式功能來剖析 OID。

**附註：**如需功能的完整清單，請參閱本指南中的附錄。

## Expression/destAttr 度量

下列資訊說明 Expression/destAttr 度量：

### Indexes

指定使用 **ObjectID** 的廠商憑證屬性來定義 MVEL 運算式，以提供 **Indexes** 度量系列屬性的值。

**建議：**設定為 **INDEX**。

**可以更新：**是

**可能值：**具有 `<IsIndex>true</IsIndex>` 的任何屬性。

**更新效果：**元件索引可能變更。

**更新的生效時間：**在元件重新搜索之後

**更新生效所需執行的動作：**更新度量系列或變更廠商憑證優先順序。

### Names

指定使用廠商憑證屬性來收集配置資料。此配置資料有助於定義 MVEL 運算式，以提供 **Names** 度量系列屬性的值。

**建議：**盡量納入所需的資訊，以便唯一識別執行個體。

**可以更新：**是

**可能值：**使用可用屬性的 MVEL 運算式

**更新效果：**元件名稱變更

**更新的生效時間：**在元件重新搜索之後

**更新生效所需執行的動作：**更新度量系列或變更廠商憑證優先順序。

## 說明

(選用) 指定使用廠商憑證屬性來收集配置資料。此配置資料有助於定義 MVEL 運算式，以提供 **Descriptions** 度量系列的值。並非所有度量系列都支援 **Descriptions** 屬性。

**建議：**盡量納入可用於說明執行個體的資訊。

**可以更新：**是

**可能值：**使用可用屬性的 MVEL 運算式

**更新效果：**元件說明變更

**更新的生效時間：**元件重新搜索

**更新生效所需執行的動作：**更新度量系列或變更廠商憑證優先順序。

## 其他度量

指定使用廠商憑證屬性來收集配置或效能資料。此資料用於定義 MVEL 運算式，以提供度量系列屬性的值。

**可以新增：**是，如果 **destAttr** 存在度量系列中。

**可以更新：**是

**可能值：**使用可用屬性的 MVEL 運算式，其可產生符合目的地屬性類型的值。

**更新效果：**輪詢值變更

**更新的生效時間：**下一次輪詢

**更新生效所需執行的動作：**無

度量系列會公開 URI (例如，

`{http://im.ca.com/normalizer}FamilyName.AttributeName`)，其分別在 **ExpressionGroup** 中參照。**ExpressionGroup/destCert** 內容已設定為 URI (例如，`{http://im.ca.com/normalizer}FamilyName`)，而 **Expression/destAttr** 設定為 **AttributeName**。

**更多資訊：**

[配置資料屬性 \(位於 p. 71\)](#)

## HierarchyList

下列資訊定義階層行為。

### Hierarchy/ParentFacet

指定用於尋找候選上層項目的 Facet QName。

可以更新：是

可能值：任何有效的 Facet

更新效果：變更階層結構。

更新的生效時間：在元件重新搜索之後

更新生效所需執行的動作：更新度量系列或變更廠商憑證優先順序。

### Hierarchy/ParentAttribute

指定用於識別特定上層項目的屬性 QName。

可以更新：是

可能值：任何有效的屬性 QName

更新效果：變更階層結構。

更新的生效時間：在元件重新搜索之後

更新生效所需執行的動作：更新度量系列或變更廠商憑證優先順序。

### Hierarchy/ChildAttribute

指定子項目的屬性 QName，用於比對上層項目的 ParentAttribute。

可以更新：是

可能值：任何有效的屬性 QName

更新效果：變更階層結構。

更新的生效時間：在元件重新搜索之後

更新生效所需執行的動作：更新度量系列或變更廠商憑證優先順序。

## IndexTagList

若要輪詢多個 MIB 表格中的屬性,每一 MIB 表格都需要含有這些屬性的屬性群組。索引標記清單提供的機制可讓兩個屬性群組 (或 MIB 表格) 與不同索引發生關聯。群組會發生關聯,讓某個表格的其中一個項目 (列) 會連結至第二個表格中的對應列。

### PrimaryTag

參照主要屬性群組 (也就是,使用 `ObjectID` 類型定義索引屬性的群組)。此元素的值必須等於主要群組之屬性群組的 'UseIndex' 標記。

**可以更新** : 是

**可能值** : 對應至主要屬性群組之屬性群組的 'UseIndex' 標記。

**更新效果** : 變更索引

**更新的生效時間** : 在元件重新搜索之後

**更新生效所需執行的動作** : 更新度量系列或變更廠商憑證優先順序。

### IndexTag

定義如何讓主要群組 (或 MIB 表格) 的列與次要群組的列發生關聯。此元素指定兩個群組的屬性必須相符,才能讓列發生關聯。

### IndexTag/Name

參照次要群組 (或 MIB 表格)。此元素的值必須等於您嘗試與主要屬性群組發生關聯之次要屬性群組的 'UseIndex' 標記。

**可以更新** : 是

**可能值** : 次要屬性群組的 'UseIndex' 標記。

**更新效果** : 變更索引

**更新的生效時間** : 在元件重新搜索之後

**更新生效所需執行的動作** : 更新度量系列或變更廠商憑證優先順序。

**IndexTag/PrimaryKeyExpression**

指定含有主要群組屬性的 MVEL 運算式，或對應至任何先前定義之 **IndexTag** 的屬性群組。計算值會與 'ThisTagKeyExpression' 進行比對。如果有相符項目，則會連結兩個屬性群組 (或 MIB 表格) 的列。然後，這些屬性可以一起使用於支援 **destAttr** (或 **Metric**) 的 'Expression'。

**可以更新**：是

**可能值**：有效的 MVEL 運算式

**更新效果**：變更索引

**更新的生效時間**：在元件重新搜索之後

**更新生效所需執行的動作**：更新度量系列或變更廠商憑證優先順序。

**IndexTag/ThisTagKeyExpression**

指定含有次要屬性群組之屬性的 MVEL 運算式。計算值會與 'PrimaryKeyExpression' 進行比對。如果有相符項目，則會連結兩個群組 (或 MIB 表格) 的列。然後，這些屬性可以一起使用於支援 **destAttr** (或 **Metric**) 的 'Expression'。

**可以更新**：是

**可能值**：有效的 MVEL 運算式

**更新效果**：變更索引

**更新的生效時間**：在元件重新搜索之後

**更新生效所需執行的動作**：更新度量系列或變更廠商憑證優先順序。

## 匯入自訂廠商憑證

若要讓廠商憑證生效，請將其匯入 Data Aggregator 中。有兩種方法可匯入廠商憑證：

- [使用 REST 用戶端匯入自訂廠商憑證](#) (位於 p. 78)。
- [使用自訂廠商憑證安裝程式](#) (位於 p. 79)

## 使用 REST 用戶端匯入自訂廠商憑證

使用任何 REST 用戶端匯入廠商憑證。建議使用 RESTClient 對 CA Performance Management 中的 REST 式 Web 服務執行 GET、POST、PUT 及其他方法。如果尚未安裝，可以從 <http://code.google.com/p/rest-client/> 取得 RESTClient GUI JAR 檔案。

使用 RESTClient 時，請考量下列資訊：

- 若要加以啓動，請連按兩下 JAR 檔案。
- 當您 POST XML 時，確定 Charset 已設定為 UTF-8。  
若要檢視並確認此設定，請按一下 [編輯 Content-Type 和 Charset] 按鈕。
- 您也可以使回應內文自動縮排，如下所述：
  1. 按一下 [工具] 功能表上的 [選項]。
  2. 選取 [Etc.] 索引標籤。
  3. 選取 [使回應內文自動縮排] 並按一下 [確定]。

請依循下列步驟：

1. 輸入 `http://da-hostname:8581/typecatalog/certifications/snmp` 做為 URL。
2. 選取 [方法] 索引標籤上的 POST。
3. 選取「application/xml」做為 Body 設定中的「Body Content-type」。  
**重要！** 無法設定 Content-type 會造成 404 錯誤。
4. 將自訂廠商憑證 XML 複製並貼到 [內文] 索引標籤中。
5. 執行此方法。

隨即會匯入廠商憑證。如果未發生錯誤，[HTTP 回應] 區段中的 [狀態] 欄位隨即顯示：

HTTP/1.1 200 OK

**附註：**有其他傳回碼指出在更新自訂元件時發生錯誤。修正此錯誤並執行其他 POST 來重試更新元件。

**重要！** 為了避免資料遺失，每次建立或更新廠商憑證、度量系列或元件時，請務必備份您的部署目錄。

## 使用自訂廠商憑證安裝程式

使用 Data Aggregator 上的 CA Performance Management 憑證安裝程式匯入憑證。安裝程式可讓您透過三種方法安裝憑證：

- 單一憑證
- ZIP 檔案形式的憑證集合
- 目錄中的所有憑證 (必須直接指向包含憑證的資料夾)

請依循下列步驟：

1. 下載並執行安裝程式。

安裝程式可於下列 URL 取得：

<http://<DA>:8581/cert/install.htm>

<DA>

Data Aggregator 的主機名稱。

**附註：**預設 REST 連接埠是 8581。

2. 選取安裝程式的語言。
3. 相對應的提示出現時，輸入下列資訊：

- DA 主機名稱
- DA REST 連接埠

4. 選取您要安裝單一憑證、憑證的集合 (ZIP) 還是目錄中的所有憑證。

選取從目錄安裝時憑證所在的資料夾。憑證安裝程式不會搜尋任何子目錄中的憑證。

5. 在安裝程序期間，安裝程式會檢查各個憑證的版本。

如果 Data Aggregator 指出已經安裝憑證，憑證安裝程式會嘗試升級憑證。如果安裝程式無法升級憑證，則會出現錯誤，指出失敗的原因。

6. 安裝程式安裝好憑證時，會顯示遇到的所有錯誤。

您可以檢閱安裝程式在電腦的主目錄中建立的記錄檔所含的錯誤詳細資料。

**附註：**您也可以使用 Linux 命令列或 UI 安裝程式安裝憑證。

## 驗證自訂廠商憑證結果

匯入自訂廠商憑證 XML 之後，請驗證結果。在我們的範例中，我們會驗證 `frPVCInfoCustom` 自訂廠商憑證是否匯入成功。

### 請依循下列步驟：

1. 在 REST 用戶端中輸入下列 URL：

`http://da_hostname:8581/typecatalog/certifications/snmp/name`

*name*

自訂廠商憑證的名稱。在此範例中，這是 `frPVCInfoCustom`。

頁面上會顯示 `frPVCInfoCustom` 自訂廠商憑證 XML。

2. 在 REST 用戶端中輸入下列 URL：

`http://da-hostname:8581/rest/vendorpriorities`

3. 在顯示的 XML 文件中搜尋廠商憑證。

4. 驗證 `MetricFamilyVendorPriority` XML 是否同時列出自訂廠商憑證和度量系列。

您現在即可使用 Data Aggregator 使用者介面來驗證自訂廠商憑證。

### 請依循下列步驟：

1. 確認您的自訂廠商憑證可用於 Data Aggregator 中：

- a. 對於 Data Aggregator 資料來源，按一下 [監控配置] 功能表上的 [廠商憑證]。

確認您的自訂廠商憑證已列出 (或搜尋您的廠商憑證)。

- b. 從廠商憑證清單中選取您的自訂廠商憑證。

確認您的自訂度量系列已顯示在 [度量系列] 檢視中。

- c. 在 [度量系列] 檢視中按一下您的度量系列。

- d. 按一下 [廠商憑證優先順序] 索引標籤。

確認自訂廠商憑證與 [所有可管理的裝置] 裝置集合已產生關聯。

2. 針對您的自訂廠商憑證支援的度量系列，配置監控設定檔和裝置集合：

- a. 確認搜索建立了預期和適當的項目。

- b. 確認項目輪詢並收集您在度量系列 XML 中指定的度量資料。

- c. 驗證收集的資料正確無誤。
- d. 如果已配置項目同步處理，則確認項目資料已正確地與 CA Performance Center 同步處理。
- e. 再次執行搜索，並確認資訊正確更新。

## 篩選支援

ExpressionGroup 區段的 Filter 標記用於在元件搜索程序上套用篩選。

繼續進行我們的範例，您只想要監控使用中的框架轉送 PVC。您可以將 Filter 標記套用到 frPVCInfoCustom 自訂廠商憑證，以監控這些特定元件。Filter 會新增至 ExpressionGroup。

**附註：**您可以在 Filter 運算式中使用 MVEL 語法。如需 MVEL 語法、廠商憑證公用程式函數和廠商憑證全域變數的詳細資訊，請參閱本指南的附錄。

以下範例顯示更新後的 frPVCInfoCustom 自訂廠商憑證外觀。在此程序中進行的變更會以粗體表示。

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Auto-generated by the type catalog local manager.-->
<DataModel namespace="http://im.ca.com/certifications/snmp"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="SNMPCertificationFacet.xsd">
  <FacetType name="frPVCInfoCustom"
    descriptorClass="com.ca.im.core.datamodel.certs.CertificationFacet
    DescriptorImpl">
    <Documentation>Frame Relay PVC Vendor
    Certification</Documentation>
    <FacetOf namespace="http://im.ca.com/core" name="Item" />
    <AttributeGroup name="AttributeGroup" external="true"
      list="true">
      <Documentation />
      <Attribute name="INDEX" type="ObjectID">
        <Documentation />
        <Source>1.3.6.1.2.1.10.32.2.1.4</Source>
        <IsIndex>true</IsIndex>
        <IsKey>false</IsKey>
        <NeedsDelta>false</NeedsDelta>
      </Attribute>
      <Attribute name="frCircuitReceivedFECNs" type="Long">
        <Documentation />
        <Source>1.3.6.1.2.1.10.32.2.1.4</Source>
        <IsIndex>false</IsIndex>
        <IsKey>true</IsKey>
        <NeedsDelta>true</NeedsDelta>
      </Attribute>
      <Attribute name="frCircuitReceivedBECNs" type="Long">
        <Documentation />
        <Source>1.3.6.1.2.1.10.32.2.1.5</Source>
        <IsIndex>false</IsIndex>
        <IsKey>true</IsKey>
        <NeedsDelta>true</NeedsDelta>
      </Attribute>
      <Attribute name="frCircuitSentFrames" type="Long">
        <Documentation />
        <Source>1.3.6.1.2.1.10.32.2.1.6</Source>
        <IsIndex>false</IsIndex>
        <IsKey>true</IsKey>
        <NeedsDelta>true</NeedsDelta>
      </Attribute>
      <Attribute name="frCircuitSentOctets" type="Long">
        <Documentation />
        <Source>1.3.6.1.2.1.10.32.2.1.6</Source>
        <IsIndex>false</IsIndex>
```

```
    <IsKey>true</IsKey>
    <NeedsDelta>true</NeedsDelta>
  </Attribute>
  <Attribute name="frCircuitReceivedFrames" type="Long">
    <Documentation />
    <Source>1.3.6.1.2.1.10.32.2.1.8</Source>
    <IsIndex>false</IsIndex>
    <IsKey>true</IsKey>
    <NeedsDelta>true</NeedsDelta>
  </Attribute>
  <Attribute name="frCircuitReceivedOctets" type="Long">
    <Documentation />
    <Source>1.3.6.1.2.1.10.32.2.1.9</Source>
    <IsIndex>false</IsIndex>
    <IsKey>true</IsKey>
    <NeedsDelta>true</NeedsDelta>
  </Attribute>
  <Attribute name="frCircuitState" type="int">
    <Documentation />
    <Source>1.3.6.1.2.1.10.32.2.1.3</Source>
    <IsIndex>false</IsIndex>
    <IsKey>false</IsKey>
    <NeedsDelta>false</NeedsDelta>
  </Attribute>
</AttributeGroup>
<Protocol>SNMP</Protocol>
<DisplayName>Frame Relay PVC Certification</DisplayName>
<Expressions>
  <ExpressionGroup
destCert="{http://im.ca.com/normalizer}frPVCInfo"
name="frPVCInfoDS">
    <Filter>(frCircuitState==2)</Filter>
    <Expression destAttr="Indexes">INDEX</Expression>
    <Expression destAttr="Names">"Frame Relay " +
INDEX</Expression>
    <Expression
destAttr="FECNIn">frCircuitReceivedFECNs</Expression>
    <Expression
destAttr="BECNIn">frCircuitReceivedBECNs</Expression>
    <Expression
destAttr="FramesIn">frCircuitReceivedFrames</Expression>
    <Expression
destAttr="FramesOut">frCircuitSentFrames</Expression>
    <Expression
destAttr="BytesIn">frCircuitReceivedOctets</Expression>
    <Expression
destAttr="BytesOut">frCircuitSentOctets</Expression>
    <Expression
destAttr="BitsIn">frCircuitReceivedOctets*8</Expression>
```

```
<Expression
destAttr="BitsOut">frCircuitSentOctets*8</Expression>
  </ExpressionGroup>
</Expressions>
<MIB>RFC1315-MIB</MIB>
</FacetType>
</DataModel>
```

我們假設在匯入自訂廠商憑證之後進行這些變更。因此，在您進行變更後，更新廠商憑證。

**更多資訊：**

[更新自訂廠商憑證](#) (位於 p. 90)

## 多重 MIB 表格支援

認證廠商 MIB 時，您必須從兩個以上的表格收集特定度量系列的原始資料。

CA Performance Management 提供需要存取多個 MIB 表格之廠商憑證的支援。此支援是以單一表格憑證之 XML 文件結構的增強功能為基礎，允許使用通用索引鍵 (索引) 聯結從多個表格收集而來的資料。繼續進行我們的範例，您只想要監控使用中的框架轉送 PVC。修改 frPVCInfoCustom 自訂廠商憑證，以新增多個 MIB 表格支援。

在此範例中，可以使用 ifXTable 中的 ifName MIB 物件與可提供此 PVC 之資料連結連線識別碼 (DLCI) 的 frCircuitDlci 物件的組合，來命名框架轉送 PVC。此種命名慣例很適合用於判斷 PVC 置於哪個框架轉送介面上。

修改自訂廠商憑證，以新增下列資訊：

- 將新的屬性新增至現有的 AttributeGroup，以代表 frCircuitDlci MIB 物件。
- 您要使用的 ifName MIB 物件來自不包含在自訂廠商憑證中的 MIB。增加新 AttributeGroup (在此例中為 ifXTable)，然後增加新屬性 (ifName)。

以下範例顯示更新後的 frPVCInfoCustom 自訂廠商憑證外觀。在此程序中進行的變更會以粗體表示：

```

<?xml version="1.0" encoding="UTF-8" ?>
<!-- Auto-generated by the type catalog local manager.-->
<DataModel namespace="http://im.ca.com/certifications/snmp"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="SNMPCertificationFacet.xsd">
  <FacetType name="frPVCInfoCustom"
  descriptorClass="com.ca.im.core.datamodel.certs.CertificationFacet
  DescriptorImpl">
    <Documentation>Frame Relay PVC Vendor
    Certification</Documentation>
    <FacetOf namespace="http://im.ca.com/core" name="Item" />
    <AttributeGroup name="ifXTableGroup" external="true"
    list="true">
      <Documentation>This pulls data from the ifXTable so that the
      ifName corresponding to the PVC can be referenced</Documentation>
      <UseIndex>ifXIndexTag</UseIndex>
      <Attribute name="ifXTableIndex" type="ObjectID">
        <Documentation />
        <IsKey>false</IsKey>
        <IsIndex>true</IsIndex>
        <Source>1.3.6.1.2.1.31.1.1.1</Source>
        <Polled>false</Polled>
      </Attribute>
      <Attribute name="ifName" type="OctetString">
        <Documentation />
        <IsKey>false</IsKey>
        <IsIndex>false</IsIndex>
        <Source>1.3.6.1.2.1.31.1.1.1</Source>
        <Polled>false</Polled>
      </Attribute>
      </AttributeGroup>
      <IndexTagList>
        <PrimaryTag>PVCIndexTag</PrimaryTag>
        <IndexTag>
          <Name>ifXIndexTag</Name>
        </IndexTag>
      </IndexTagList>
      <PrimaryKeyExpression>snmpOIDParser(INDEX,1,1)</PrimaryKeyExpression>
    <ThisTagKeyExpression>ifXTableIndex</ThisTagKeyExpression>
    </IndexTag>
  </IndexTagList>
  <AttributeGroup name="AttributeGroup" external="true"
  list="true">
    <Documentation />
    <UseIndex>PVCIndexTag</UseIndex>
  
```

```
<Attribute name="INDEX" type="ObjectID">
  <Documentation />
  <Source>1.3.6.1.2.1.10.32.2.1.4</Source>
  <IsIndex>true</IsIndex>
  <IsKey>false</IsKey>
  <NeedsDelta>false</NeedsDelta>
</Attribute>
<Attribute name="frCircuitReceivedFECNs" type="Long">
  <Documentation />
  <Source>1.3.6.1.2.1.10.32.2.1.4</Source>
  <IsIndex>false</IsIndex>
  <IsKey>true</IsKey>
  <NeedsDelta>true</NeedsDelta>
</Attribute>
<Attribute name="frCircuitReceivedBECNs" type="Long">
  <Documentation />
  <Source>1.3.6.1.2.1.10.32.2.1.5</Source>
  <IsIndex>false</IsIndex>
  <IsKey>true</IsKey>
  <NeedsDelta>true</NeedsDelta>
</Attribute>
<Attribute name="frCircuitSentFrames" type="Long">
  <Documentation />
  <Source>1.3.6.1.2.1.10.32.2.1.6</Source>
  <IsIndex>false</IsIndex>
  <IsKey>true</IsKey>
  <NeedsDelta>true</NeedsDelta>
</Attribute>
<Attribute name="frCircuitSentOctets" type="Long">
  <Documentation />
  <Source>1.3.6.1.2.1.10.32.2.1.6</Source>
  <IsIndex>false</IsIndex>
  <IsKey>true</IsKey>
  <NeedsDelta>true</NeedsDelta>
</Attribute>
<Attribute name="frCircuitReceivedFrames" type="Long">
  <Documentation />
  <Source>1.3.6.1.2.1.10.32.2.1.8</Source>
  <IsIndex>false</IsIndex>
  <IsKey>true</IsKey>
  <NeedsDelta>true</NeedsDelta>
</Attribute>
<Attribute name="frCircuitReceivedOctets" type="Long">
  <Documentation />
  <Source>1.3.6.1.2.1.10.32.2.1.9</Source>
  <IsIndex>false</IsIndex>
  <IsKey>true</IsKey>
  <NeedsDelta>true</NeedsDelta>
</Attribute>
```

```

<Attribute name="frCircuitState" type="int">
  <Documentation />
  <Source>1.3.6.1.2.1.10.32.2.1.3</Source>
  <IsIndex>false</IsIndex>
  <IsKey>false</IsKey>
  <NeedsDelta>false</NeedsDelta>
</Attribute>
<Attribute name="frCircuitDlci" type="int">
  <Documentation />
  <Source>1.3.6.1.2.1.10.32.2.1.2</Source>
  <IsIndex>false</IsIndex>
  <IsKey>false</IsKey>
  <NeedsDelta>false</NeedsDelta>
</Attribute>
</AttributeGroup>
<Protocol>SNMP</Protocol>
<DisplayName>Frame Relay PVC Certification</DisplayName>
<Expressions>
  <ExpressionGroup
destCert="{http://im.ca.com/normalizer}frPVCInfo"
name="frPVCInfoDS">
    <Filter>(frCircuitState==2)</Filter>
    <Expression destAttr="Indexes">INDEX</Expression>
    <Expression destAttr="Names">isdef(ifName)?
(isdef(frCircuitDlci) ? ifName + " DCLI:" + frCircuitDlci : "Frame
Relay " + INDEX) : "Frame Relay " + INDEX</Expression>
    <Expression
destAttr="FECNIn">frCircuitReceivedFECNs</Expression>
    <Expression
destAttr="BECNIn">frCircuitReceivedBECNs</Expression>
    <Expression
destAttr="FramesIn">frCircuitReceivedFrames</Expression>
    <Expression
destAttr="FramesOut">frCircuitSentFrames</Expression>
    <Expression
destAttr="BytesIn">frCircuitReceivedOctets</Expression>
    <Expression
destAttr="BytesOut">frCircuitSentOctets</Expression>
    <Expression
destAttr="BitsIn">frCircuitReceivedOctets*8</Expression>
    <Expression
destAttr="BitsOut">frCircuitSentOctets*8</Expression>
    </ExpressionGroup>
  </Expressions>
  <MIB>RFC1315-MIB</MIB>
</FacetType>
</DataModel>

```

我們假設在匯入自訂廠商憑證之後進行這些變更。因此，在您進行變更後，[更新廠商憑證](#) (位於 p. 90)。

## AttributeGroup (多個 MIB 表格)

每個表格都必須位於自己的 AttributeGroup 區段中。該表格上的每個屬性都會新增為該 AttributeGroup 的子項目。

請參閱下列各節以取得下列資訊：

- AttributeGroup 資訊 – 有關用於定義主要和次要表格屬性之 XML 元素的詳細資料。
- UseIndex 和 IndexTagList 資訊 – 有關用於聯結主要和次要屬性群組之 XML 元素的詳細資料。

在此案例中，主要屬性群組代表您要以更多資訊「擴充」的表格。次要屬性群組包含主要屬性群組的「擴充」資訊。

主要 AttributeGroup 包含的屬性用以識別做為次要 AttributeGroup 中通用「索引鍵」的 MIB 表格變數。

次要 AttributeGroup 包含所有 MIB 表格變數的屬性定義，其帶有主要表格的「擴充」資訊。此外，有一個屬性用以識別符合主要 AttributeGroup 中通用「索引鍵」的變數。

**更多資訊：**

[AttributeGroup \(廠商憑證\)](#) (位於 p. 66)

[UseIndex](#) (位於 p. 89)

[IndexTagList \(多個 MIB 表格\)](#) (位於 p. 89)

## UseIndex

每個 AttributeGroup 都有 UseIndex 標記。UseIndex 標記可讓您將 OID 群集在一個通用名稱之下。然後，此通用名稱會與指定的變數產生關聯，以做為個別 MIB 表格中的通用索引鍵 (索引)。

下列資訊摘要說明可自訂的 XML 元素：

### AttributeGroup/UseIndex

分別唯一識別 IndexTagList 區段中使用的主要和次要標記名稱。

**建議：**設定為 AttributeGroup/name 內容的值。

更多資訊：

[AttributeGroup \(多個 MIB 表格\)](#) (位於 p. 88)

## IndexTagList (多個 MIB 表格)

IndexTagList 區段是一個可讓兩個屬性群組 (或 MIB 表格) 與不同索引發生關聯的機制。當群組有所關聯時，一個項目會有多個表格中的多個索引 ID。

IndexTagList 區段包含所有聯結資訊，包括每個次要屬性群組的 IndexTag 區段。

### IndexTagList/PrimaryTag

定義主要屬性群組 (或 MIB 表格)。設定為主要 AttributeGroup 的 UseIndex 內容值。

### IndexTag/Name

定義次要屬性群組。設定為次要 AttributeGroup 的 UseIndex 內容值。

### IndexTag/PrimaryKeyExpression

指定用以產生主要表格中通用索引鍵的運算式。考慮使用 MVEL 函數，從指定的主要表格索引 Attribute 衍生通用索引鍵。

### IndexTag/ThisTagKeyExpression

指定用以產生次要表格中通用索引鍵的運算式。考慮使用 MVEL 函數，從指定的次要表格索引 Attribute 衍生通用索引鍵。

多重表格方法可支援兩個以上的表格鏈結。多重表格聯結中有兩種關係存在：

- **主要 → 次要 #1, 主要 → 次要 #2**  
索引標記清單中次要表格的順序不構成影響。
- **主要 → 次要 #1 → 次要 #2**  
將次要表格 #1 列在次要表格 #2 之前，如此一來就會合併表格。

主要表格中的一或多列可以合法對應至次要表格中的同一列。系統會依序搜尋次要表格上的索引鍵，第一個相符項目勝出。

**更多資訊：**

[多重 MIB 表格支援](#) (位於 p. 84)

[AttributeGroup \(多個 MIB 表格\)](#) (位於 p. 88)

## 更新自訂廠商憑證

您可以更新現有的自訂廠商憑證。繼續進行我們的範例，您先前稍微變更了自訂度量系列和自訂廠商憑證。

您在度量系列中新增了度量。您在廠商憑證中新增了 MIB 表格，並且納入自訂廠商憑證的篩選。立即更新廠商憑證，讓變更生效。

**附註：**如需更新標記或屬性之效果的詳細資訊，請檢閱自訂廠商憑證 XML 詳細資料。請特別檢閱特定屬性說明。

**請依循下列步驟：**

1. 在 URL 欄位中輸入下列網址：

`http://da_hostname:8581/typecatalog/certifications/snmp/CustomVendorCertName`

***CustomVendorCertName***

要更新的自訂廠商憑證名稱。在我們的範例中，廠商憑證的名稱為 `frPVCInfoCustom`。

2. 若要確認廠商憑證位於該 URL，請選取 [方法] 索引標籤中的 GET。
3. 確認廠商憑證位於該 URL 之後，請選取 [方法] 索引標籤中的 PUT。

4. 將更新後的自訂度量系列 XML 複製並貼到 [內文] 索引標籤 ([編輯] 欄位) 中，並將 Content-type 設定為 application/xml。

**重要！** 無法設定 Content-type 會造成 404 錯誤。

5. 按一下 [URL] 欄位旁邊的 [移至] 按鈕。

自訂廠商憑證隨即更新。如果未發生錯誤，[HTTP 回應] 區段中的 [狀態] 欄位會顯示下列訊息：

HTTP/1.1 200 OK

**附註：**有其他傳回碼指出在更新自訂廠商憑證時發生錯誤。修正此錯誤並執行其他 PUT 來重試更新廠商憑證。

**重要！** 為了避免資料遺失，每次建立或更新廠商憑證、度量系列或元件時，請務必備份您的部署目錄。

**更多資訊：**

[瞭解廠商憑證 XML 結構](#) (位於 p. 62)

[更新自訂度量系列](#) (位於 p. 59)



# 附錄 A：廠商憑證運算式：運算式運算子、函數與全域變數

---

此區段說明廠商憑證運算式中可使用的一些運算式運算子、函數及全域變數。

您可以在廠商憑證運算式中使用 MVEL 語法。MVEL 是 Java 環境中可公開使用的可嵌入運算式語言，所用的語法接近 Java。MVEL 支援類似 Java 運算式的運算式。

您能夠使用運算子建立運算式、使用括弧控制優先順序，而且您能夠以分號控制陳述式。如需 MVEL 語言的詳細參考，請參閱 <http://mvel.codehaus.org>。

MVEL 語法具有廠商憑證公用程式函數，以及廠商憑證全域變數，您也可用於廠商憑證運算式中。

若要瞭解如何使用函數、運算子及全域變數，可使用 CA Performance Center 中的 [廠商憑證] 索引標籤。

本節包含以下主題：

[運算式運算子](#) (位於 p. 93)  
[函數及全域變數](#) (位於 p. 95)

## 運算式運算子

此區段說明廠商憑證運算式中可使用的運算子。

您可以在廠商憑證運算式中使用 MVEL 語法。MVEL 是 Java 環境中可公開使用的可嵌入運算式語言，所用的語法接近 Java。MVEL 支援類似 Java 運算式的運算式。

您能夠使用運算子建立運算式、使用括弧控制優先順序，而且以分號控制陳述式。如需 MVEL 語言的詳細參考，請參閱 <http://mvel.codehaus.org>。

下表摘要說明可用的運算子：

附註：在 XML 文件中，使用 XML Named Entities (XNE) 呈現方式。

運算子	XNE	說明	範例
=		指派	<code>a = 1</code>
==		等於	<code>"fred" == "fred"</code>
!=		不等於	<code>"fred" != "tom"</code>
>	&gt;	大於	<code>1 &gt; 0 為 true</code>
<	&lt;	小於	<code>0 &lt; 1 為 true</code>
>=		大於或等於	<code>1 &gt;= 0 為 true</code>
<=		小於或等於	<code>1 &lt;= 1 為 true</code>
contains		確認左邊的值是否包含右邊的值	<code>"tomcat" contains "cat"</code>
isdef		測試是否定義變數	<code>isdef a</code>
+		新增	<code>1 + 1</code>
+		加	<code>"one" + "two"</code>
-		減	<code>2 - 1</code>
*		乘	<code>2 * 2</code>
/		除	<code>4 / 2</code>
%		模數	<code>5 % 2</code>
&&	&&	邏輯 AND	<code>(x&gt;-1) &amp;&amp; (x&lt;1)</code>
		邏輯 OR	<code>(x&gt;-1) &amp;&amp; (x&lt;1)</code>
&	&	AND 位元運算	<code>17 &amp; 0xF</code>
		OR 位元運算	<code>4   1</code>
^		排除 OR 位元運算	<code>5 ^ 1</code>
!		否定	<code>! True</code>
?		三元運算子	<code>年齡 &gt; 17 ? "allow" : "deny"</code>

## 函數及全域變數

此區段說明廠商憑證運算式中可使用的一些運算式運算子、函數及全域變數。

您可以在廠商憑證運算式中使用 MVEL 語法。MVEL 是 Java 環境中可公開使用的可嵌入運算式語言，所用的語法接近 Java。MVEL 支援類似 Java 運算式的運算式。

您能夠使用運算子建立運算式、使用括弧控制優先順序，而且您能夠以分號控制陳述式。如需 MVEL 語言的詳細參考，請參閱 <http://mvel.codehaus.org>。

MVEL 語法具有廠商憑證公用程式函數，以及廠商憑證全域變數，您也可用於廠商憑證運算式中。

若要瞭解如何使用函數與全域變數，可使用 CA Performance Center 中的 [廠商憑證] 索引標籤。

### availabilityWithSysUptime 函數

此函數將使用 sysUptime 及輪詢期間計算可用性的百分比，授予寬限期。使用此函數可認證裝置。

#### 語法

此函數的格式如下：

```
Object availabilityWithSysUptime (Long sysUpTime, Long duration)
```

#### 參數

##### sysUpTime

系統的網路管理部份上次重新初始化之後的時間 (單位為釐秒)。

##### 期間

輪詢期間時間，單位為秒。使用全域變數 \_rspDuration。如需詳細資訊，請參閱進階範例。

#### 傳回值

傳回可用性的百分比 (0–100)，傳遞無效資料時則傳回「null」。

### 範例

對於 `sysUpTime` 的 30000 及輪詢期間的 300，下列運算式產生下列結果：

**運算式：**

```
availabilityWithSysUptime (sysUpTime, duration)
```

**結果：**

100

對於 `sysUpTime` 的 6000 及輪詢期間的 300，同一個運算式產生下列結果：

**結果：**

20

對於 `sysUpTime` 的 30005 及輪詢期間的 300，同一個運算式產生下列結果：

**結果：**

100

### 進階範例

下列運算式取自「系統統計資料」廠商憑證：

```
Availability=availabilityWithSysUptime(sysUpTime,_rspDuration)
```

## mapModel 函數

此函數使用 `objectID (sysObjectID)` 的值，並將系統 OID 對應至模型名稱字串。使用此函數可認證裝置。

### 語法

此函數的格式如下：

```
String mapModel ( ObjectID sysObjectID )
```

### 參數

#### **sysObjectID**

要剖析的物件 ID 值。

## 傳回值

傳回對應的模型名稱包含在其中的字串。

## 範例

對於 OID 值的 1.3.6.1.4.1.9.1.223，下列運算式產生下列結果：

**運算式：**

`mapModel (oid )`

**結果：**

`Cisco7204VXR`

對於 OID 值的 1.3.6.1.4.1，同一個運算式產生下列結果：

**結果：**

`不明的 1.3.6.1.4.1`

## 進階範例

下列運算式取自「系統統計資料」廠商憑證：

`Model=mapModel (sysObjectID)`

## mapVendor 函數

此函數使用 `objectID (sysObjectID)` 的值，並將系統 OID 對應至廠商名稱字串。使用此函數可認證裝置。

## 語法

此函數的格式如下：

`String mapVendor ( ObjectID sysObjectID )`

## 參數

### sysObjectID

要剖析的物件 ID 值。

### 傳回值

傳回對應的廠商名稱包含在其中的字串。

### 範例

對於 OID 值的 1.3.6.2.1.2.2636.0，下列運算式產生下列結果：

**運算式：**

```
mapVendor (oid )
```

**結果：**

Juniper

對於 OID 值的 1.3.6.2.1.2.1234567.0，同一個運算式產生下列結果：

**結果：**

不明

### 進階範例

下列運算式取自「系統統計資料」廠商憑證：

```
Model=mapVendor (sysObjectID)
```

## snmpConstArrayMap 函數

此函數將值 (索引) 對應到常數的集合 (陣列)。必要時，此函數會將輸入值四捨五入為最接近的整數值。然後，它會使用整數值做為顯示成  $c_0$ 、 $c_1$  到  $c_{n-1}$  的常數 (陣列) 集合索引。 $c$  值必須是整數。運算式經過剖析，並傳回  $cx$  時，此函數會檢查這些值。如果值不在 0 至  $n-1$  (含) 的範圍內，結果將是 0 (不出現錯誤訊息)。使用此函數可認證裝置。

### 語法

此函數的格式如下：

```
Integer snmpConstArrayMap(Double index, Integer[] array)
```

**參數****索引**

做為陣列索引的 Double 值。

**陣列**

整數值的任何範圍。

**傳回值**

傳回陣列的整數值。 null 的整數值將傳回 null。

**範例**

對於 2 的索引以及 {5, 6, 7, 8, 9, 4} 的陣列，下列運算式將產生下列結果：

**運算式：**

`snmpConstArrayMap (index, array)`

**結果：**

7

對於 4.88 的索引以及 {5, 6, 7, 8, 9, 4} 的陣列，下列運算式將產生下列結果：

**運算式：**

`snmpConstArrayMap (value, array)`

**結果：**

4

**進階範例**

下列運算式取自「一般數據機」廠商憑證：

```
SpeedOut=snmpConstArrayMap(mdmCsFinalTxLinkRate,{0,110,300,600,1200,2400,4800,720
0,9600,12000,14000,16000,19000,38000,75,450,0,57000,21000,24000,26000,28000,0,310
00,33000,25333,26666,28000,29333,30666,32000,33333,34666,36000,37333,38666,40000,
41333,42666,44000,45333,46666,48000,49333,50666,52000,53333,54666,56000,57333,586
66,60000,61333,62666,64000})
```

## mvelInfo 函數

此函數會以輸入參數填入 karaf.log 檔案的 INFO 層級。使用此函數可記錄傳回您認為不正確結果的報告中的輪詢值。

Data Collector 的輪詢記錄只會顯示輪詢屬性的值，而報告只會顯示計算的結果。mvelInfo 函數可讓您檢視輸入輪詢值，以判斷計算錯誤的地方。

### 語法

此函數的格式如下：

```
String mvelInfo (Array objects)
```

### 參數

#### **objects**

物件陣列會記錄在 Data Collector 的 karaf.log 檔案的 INFO 層級之下。

### 傳回值

Null

### 範例

下列運算式會在 karaf.log 檔案中記錄 cpmCPUTotal5minRev。

#### 運算式：

```
mvelInfo(["cpmCPUTotal5minRev=", cpmCPUTotal5minRev])
```

#### 結果：

Null

#### 結果 (karaf.log)：

MVEL 資訊：cpmCPUTotal5minRev=15

### 進階範例

```
mvelInfo(["cpmCPUTotal5minRev=", cpmCPUTotal5minRev, " cpmCPUTotal10minRev=",  
cpmCPUTotal10minRev]); cpmCPUTotal10minRev;
```

結果：

12

結果 (karaf.log)：

MVEL 資訊：cpmCPUTotal5minRev=15 cpmCPUTotal10minRev=12

## mvelWarn 函數

此函數會以輸入參數填入 karaf.log 檔案的 WARN 層級。使用此函數可記錄傳回您認為不正確結果的報告中的輪詢值。

Data Collector 的輪詢記錄只會顯示輪詢屬性的值，而報告只會顯示計算的結果。mvelWarn 函數可讓您檢視輸入輪詢值，以判斷計算錯誤的地方。

### 語法

此函數的格式如下：

```
String mvelWarn (Array objects)
```

### 參數

#### objects

物件陣列會記錄在 Data Collector 的 karaf.log 檔案的 WARN 層級之下。

### 傳回值

Null

### 範例

下列運算式會在 karaf.log 檔案中記錄 cpmCPUTotal5minRev。

運算式：

```
mvelWarn(["cpmCPUTotal5minRev=", cpmCPUTotal5minRev])
```

結果：

Null

結果 (karaf.log)：

MVEL 警告：cpmCPUTotal5minRev=15

### 進階範例

```
mvelWarn(["cpmCPUTotal5minRev=", cpmCPUTotal5minRev, " cpmCPUTotal10minRev=", cpmCPUTotal10minRev]); cpmCPUTotal10minRev;
```

結果：

12

結果 (karaf.log)：

MVEL 警告：cpmCPUTotal5minRev=15 cpmCPUTotal10minRev=12

## mvelError Function

此函數會以輸入參數填入 karaf.log 檔案的 ERROR 層級。使用此函數可記錄傳回您認為不正確結果的報告中的輪詢值。

Data Collector 的輪詢記錄只會顯示輪詢屬性的值，而報告只會顯示計算的結果。mvelError 函數可讓您檢視輸入輪詢值，以判斷計算錯誤的地方。

### 語法

此函數的格式如下：

```
String mvelError (Array objects)
```

## 參數

### objects

物件陣列會記錄在 Data Collector 的 karaf.log 檔案的 ERROR 層級之下。

## 傳回值

Null

## 範例

下列運算式會在 karaf.log 檔案中記錄 cpmCPUTotal5minRev。

### 運算式：

```
mvelError(["cpmCPUTotal5minRev=", cpmCPUTotal5minRev])
```

### 結果：

Null

### 結果 (karaf.log)：

MVEL 錯誤：cpmCPUTotal5minRev=15

## 進階範例

```
mvelError(["cpmCPUTotal5minRev=", cpmCPUTotal5minRev, " cpmCPUTotal10minRev=", cpmCPUTotal10minRev]); cpmCPUTotal10minRev;
```

### 結果：

12

### 結果 (karaf.log)：

MVEL 錯誤：cpmCPUTotal5minRev=15 cpmCPUTotal10minRev=12

## mvelDebug 函數

此函數會以輸入參數填入 karaf.log 檔案的 DEBUG 層級。使用此函數可記錄傳回您認為不正確結果的報告中的輪詢值。

Data Collector 的輪詢記錄只會顯示輪詢屬性的值，而報告只會顯示計算的結果。mvelDebug 函數可讓您檢視輸入輪詢值，以判斷計算錯誤的地方。

### 語法

此函數的格式如下：

```
String mvelDebug (Array objects)
```

### 參數

#### objects

物件陣列會記錄在 Data Collector 的 karaf.log 檔案的 DEBUG 層級之下。

### 傳回值

Null

### 範例

下列運算式會在 karaf.log 檔案中記錄 cpmCPUTotal5minRev。

#### 運算式：

```
mvelDebug([“cpmCPUTotal5minRev=”, cpmCPUTotal5minRev])
```

#### 結果：

Null

#### 結果 (karaf.log)：

MVEL 傳錯：cpmCPUTotal5minRev=15

## 進階範例

```
mvelDebug(["cpmCPUTotal5minRev=", cpmCPUTotal5minRev, " cpmCPUTotal10minRev=", cpmCPUTotal10minRev]); cpmCPUTotal10minRev;
```

結果：

12

結果 (karaf.log)：

MVEL 傳錯：cpmCPUTotal5minRev=15 cpmCPUTotal10minRev=12

## mvelDebug 函數

此函數會以輸入參數填入 karaf.log 檔案的 TRACE 層級。使用此函數可記錄傳回您認為不正確結果的報告中的輪詢值。

Data Collector 的輪詢記錄只會顯示輪詢屬性的值，而報告只會顯示計算的結果。mvelTrace 函數可讓您檢視輸入輪詢值，以判斷計算錯誤的地方。

### 語法

此函數的格式如下：

```
String mvelTrace (Array objects)
```

### 參數

#### objects

物件陣列會記錄在 Data Collector 的 karaf.log 檔案的 TRACE 層級之下。

### 傳回值

Null

### 範例

下列運算式會在 karaf.log 檔案中記錄 cpmCPUTotal5minRev。

運算式：

```
mvelTrace(["cpmCPUTotal5minRev=", cpmCPUTotal5minRev])
```

結果：

Null

結果 (karaf.log)：

MVEL 追蹤：cpmCPUTotal5minRev=15

### 進階範例

```
mvelTrace(["cpmCPUTotal5minRev=", cpmCPUTotal5minRev, " cpmCPUTotal10minRev=", cpmCPUTotal10minRev]); cpmCPUTotal10minRev;
```

結果：

12

結果 (karaf.log)：

MVEL 追蹤：cpmCPUTotal5minRev=15 cpmCPUTotal10minRev=12

## snmpCounter64 函數

此函數評估兩個 32 位元數值，並傳回包含 64 位元表示的值。使用此函數可認證裝置。hiVal 左移 32 個位元，而且新增 lowVal，結果將放在 64 位元傳回變數中。

### 語法

此函數的格式如下：

```
Object snmpCounter64 (Long hiVal, Long lowVal)
```

## 參數

### hiVal

代表高序位位元的 32 位元數值。

### lowVal

代表低序位位元的 32 位元數值。

## 傳回值

傳回兩個 32 位元數值的 64 位元表示，或在 32 位元值輸入為 null 時傳回「null」。

## 範例

對於 hiVal 的 88 及 lowVal 的 558，下列運算式產生下列結果：

### 運算式：

```
snmpCounter64 (hiVal, lowVal)
```

### 結果：

377957122606

## 進階範例

下列運算式取自「Cisco CBQos ClassMap」廠商憑證：此憑證包含許多 snmpMax 範例：

```
PrePolicyPackets=snmpMax(0,snmpCounter64(cbQosCMPrePolicyPktOverflow,cbQosCMPrePolicyPkt))
```

## snmpGetUpSinceTime 函數

此函數會根據目前狀況後的秒數，傳回系統的啓動時間。

## 語法

此函數的格式如下：

```
snmpGetUpSinceTime(Long upTime)
```

## 參數

### upTime

目前狀況開始後的秒數。您可以從下列 OID 取得系統運作時間：  
1.3.6.1.2.1.1.3.0。在傳入之前，請將它轉換為厘秒。

## 傳回值

以目前狀況後的總秒數形式，傳回裝置的啓動時間。

## snmpMax 函數

此函數傳回兩個 64 位元值中較大的值。使用此函數可認證裝置。

## 語法

此函數的格式如下：

```
Object snmpMax(BigInteger val1, BigInteger val2)
```

## 參數

### val1

第一個 64 位元 BigInteger 值。

### val2

第二個 64 位元 BigInteger 值。

## 傳回值

傳回傳入的兩個 BigInteger 值上限，或在 BigInteger 值輸入為 null 時傳回「null」。

## 範例

對於 val1 的  $2^{32}$  及 val2 的 10，下列運算式產生下列結果：

運算式：

```
snmpMax (val1, val2)
```

結果：

$2^{32}$

對於 val1 的 5864 及 val2 的 134556890，同一個運算式產生下列結果：

結果：

134556890

### 進階範例

下列運算式取自「Cisco CBQos ClassMap」廠商憑證：此憑證包含許多 snmpMax 範例：

```
PrePolicyPackets=snmpMax(0,snmpCounter64(cbQosCMPrePolicyPktOverflow,cbQosCMPrePolicyPkt))
```

## snmpObjectIDToASCIIString 函數

此函數會將 SNMP OID 值轉換為其字串表示法。所有前置或結尾空格都會被移除。

### 語法

此函數的格式如下：

```
snmpObjectIDToASCIIString( Object Id oid )
```

### 參數

#### oid

要轉換為字串的物件 ID。

## snmpOIDParser 函數

此函數使用 objectID (OID) 的值，並依據 startIndex 及 endIndex 值剖析出 OID 的子集。索引從 1 開始。如果 endIndex 為 -1，則前往 OID 的末端。使用此函數可認證裝置。

### 語法

此函數的格式如下：

```
ObjectID snmpOIDParser( ObjectID OID, Integer startIndex, Integer endIndex )
```

## 參數

### OID

要剖析的物件 ID (OID) 值。

### startIndex

開始剖析的索引整數值。

### endIndex

停止剖析的索引整數值。

## 傳回值

傳回剖析的子集 ObjectID (OID)。

## 範例

對於 OID 值的 1.2.3.4.5.6.7.8.9.10、startIndex 值的 1，以及 endIndex 值的 5，下列運算式產生下列結果：

運算式：

```
snmpOIDParser(oid, startIndex, endIndex )
```

結果：

1.2.3.4.5

對於 OID 值的 1.2.3.4.5.6.7.8.9.10、startIndex 值的 6，以及 endIndex 值的 -1，同一個運算式產生下列結果：

結果：

6.7.8.9.10

## 進階範例

下列運算式取自「Cisco CBQos ClassMap」廠商憑證：

```
ItemUniqueIDs=snmpOIDParser(cbQosConfigIndex, 2, 2)
```

## snmpOctetStringFloat 函數

此函數將 SNMP 八位元字串轉換為浮點值。使用此函數可認證裝置。SNMP 八位元字串是七位元 ASCII 字串。

### 語法

此函數的格式如下：

```
Object snmpOctetStringFloat(byte[] octetString)
```

### 參數

#### octetString

SNMP 八位元字串。

### 傳回值

傳回轉換後的字串值，或在函數無法轉換字串時傳回「null」。

### 範例

對於 octetString 的 {0x33, 0x33, 0x2E, 0x33, 0x33}，下列運算式產生下列結果：

#### 運算式：

```
snmpOctetStringFloat (octetString)
```

#### 結果：

33.33

對於 octetString 的 {0x36, 0x36, 0x36}，同一個運算式產生下列結果：

#### 結果：

666.0

## snmpProtectedDiv 函數

此函數除兩個 Double 值，並傳回除法運算後的 Double 結果。如果被除數或除數為 null 或 0.0，傳回值為 0.0。使用此函數可保護運算式不除以 null 或 0。Data Repository 可以包含 null 或 0 值，例如在輪詢失敗時。在此情況下，可使用此函數避免除以 0 的例外。

### 語法

此函數的格式如下：

```
Double snmpProtectedDiv(Double val1, Double val2)
```

### 參數

#### val1

為被除數。這是將除以 val2 的 Double 值 (浮點數)。*(Double* 是 Java 資料類型。)

#### val2

為除數。這是 Double 值 (浮點數)。*(Double* 是 Java 資料類型。)

### 傳回值

如果被除數或除數為 null 或 0.0 (*Double* 是 Java 資料類型)，傳回 Double 或 0.0 的除法運算結果。

### 範例

對於 val1 的 7.2 及 val2 的 2，下列運算式產生下列結果：

運算式：

```
snmpProtectedDiv(val1, val2)
```

結果：

3.6

對於 val1 的 7.2 及 val2 的 null 或 0.0，下列運算式產生下列結果：

結果：

0.0

### 進階範例

下列運算式取自廠商憑證：

```
Utilization=snmpProtectedDiv((cpuStatsUser + cpuStatsSys),(cpuStatsUser +  
cpuStatsSys + (isdef(cpuStatsWait)?cpuStatsWait:0) + cpuStatsIdle))*100
```

## snmpRound 函數

此函數將數值四捨五入為最接近的整數值。

### 語法

此函數的格式如下：

```
Long snmpRound(Double dNumber)
```

### 參數

#### dNumber

將四捨五入的 Double 值 (浮點數) (Double 是 Java 資料類型)。

### 傳回值

傳回 Long 值，這是與 dNumber 中提供的值最接近的整數值 (Long 是 Java 資料類型)。

### 範例

對於 dNumber 的 3.5，下列運算式產生下列結果：

#### 運算式：

```
snmpRound(dNumber)
```

#### 結果：

對於 dNumber 的 3.4，同一個運算式產生下列結果：

結果：

3

### 進階範例

下列運算式取自「Cisco IPSLA 抖動精確值統計資料」廠商憑證：

```
PathAvailability=snmpRound(rttMonJitterStatsNumOfRTT / (rttMonJitterStatsNumOfRTT + rttMonJitterStatsPacketLossSD + rttMonJitterStatsPacketLossDS + rttMonJitterStatsPacketOutOfSequence + rttMonJitterStatsPacketMIA + rttMonJitterStatsPacketLateArrival + rttMonJitterStatsError + rttMonJitterStatsBusies + 1/100) * 100)
```

## snmpStringParser 函數

此函數僅針對內部使用而撰寫。它可剖析來自 CA Application Insight Module (AIM) 的 IP 位址。CA 只在內部類別中測試此函數；因此，可能不支援其他類型的類別。

### 語法

此函數的格式如下：

```
snmpStringParser(Delimiter, Type to convert to, String to parse 1, String to parse 2)
```

### 參數

#### 要轉換的目標類型

提供的字串所要剖析成爲的類別類型。

#### 要剖析的字串 1 和 2。

要剖析的 IP 位址。這兩個字串可讓您提供 IPv4 和 IPv6 格式的位址。

### 傳回值

傳回轉換後的字串值，或在函數無法剖析字串時傳回「null」。

## snmpSvcs Function

此函數從代理程式的 sysObjectID、sysService 及 ipForwarding MIB 變數取得值，並決定 SNMP 代理程式支援哪些服務。例如，Router/Switch/Repeater/Host 可以是支援的服務，如 SNMP MIB RFC 1213 所定義。

函數傳回的結果將經過如下評估，因為自訂裝置類型的優先順序高於系統類型：

- 如果在 DeviceTypes 檔案中對應 sysObjectID OID，則將從該檔案傳回服務。
- 如果並未在 DeviceTypes 檔案中對應 sysObjectID OID，則 sysServices 及 ipForwarding 用來傳回支援的服務。

### 語法

此函數的格式如下：

```
DeviceService[] snmpSvcs(0bjectID sysObjectID, Integer sysServices, Integer
ipForwarding)
```

### 參數

#### sysObjectID

要剖析的物件 ID 值。

#### sysServices

各個位元代表不同服務的整數，例如 switch/repeater/host。

#### ipForwarding

此實體做為資料包轉送的 IP 閘道或 IP 主機。此實體接受轉送的資料包，但是傳送的資料包未送達此實體。

### 傳回值

傳回下列一或多個裝置服務的清單：

- ROUTER
- REPEATER
- SWITCH
- HOST
- UNKNOWN\_TYPE

## 範例

對於 sysServices 值 8、ipForwarding 值 0，以及 DeviceTypes 檔案中找不到的 sysObjectID，下列運算式產生下列結果：

**運算式：**

```
snmpSvcs(sysObjectID,sysServices,ipForwarding)
```

**結果：**

```
DeviceService[HOST]
```

## 進階範例

下列運算式取自「系統統計資料」廠商憑證：

```
Services=snmpSvcs(sysObjectID,isdef(sysServices)?sysServices:0,isdef(ipForwarding)?ipForwarding:0)
```

## storePortReconfig 函數

此函數傳回 XML 代表 ifNumber、ifTableLastChange、ifStackLastChange 值的字串。必要時，您可以使用 XML 追蹤項目變更，並重新搜索裝置的介面。

### 語法

此函數的格式如下：

```
String storePortReconfig ( Integer ifNumber, Long ifTableLastChange, Long ifStackLastChange )
```

### 參數

#### **ifNumber**

裝置的連接埠數。

#### **ifTableLastChange**

最新連接埠表格變更的日期及時間，單位為毫秒，從 1970 年 1 月 1 日 (GMT) 的開始日期及時間起算。

#### **ifStackLastChange**

最新連接埠堆疊變更的日期及時間，單位為毫秒，從 1970 年 1 月 1 日 (GMT) 的開始日期及時間起算。

## 傳回值

傳回 XML 使用下列範例所示格式的字串。

## 範例

對於 `ifNumber` 的 5、`ifTableLastChange` 的 123456，以及 `ifStackLastChange` 的 234567，下列運算式產生下列結果：

運算式：

```
storePortReconfig ( ifNumber, ifTableLastChange, ifStackLastChange )
```

結果：

```
<ReconfigData>
  <ReconfigValue name="ifNumber" value="5"/>
  <ReconfigValue name="ifTableLastChange" value="123456"/>
  <ReconfigValue name="ifStackLastChange" value="234567"/>
</ReconfigData>
```

## 全域變數

Data Aggregator 支援下列全域變數：

### \_rspDuration

包含目前輪詢週期期間的 Long (Java 資料類型)，單位為秒數。

附註：「系統統計資料」廠商憑證包含 `_rspDuration` 的使用範例。

### \_rspTimestamp

1970 年 1 月 1 日 (GMT) 之後目前輪詢週期開始時的時間戳記包含在其中的 Long (Java 資料類型)，單位為毫秒。



# 附錄 B：疑難排解

---

本節包含以下主題：

[疑難排解：無法建立廠商憑證](#) (位於 p. 119)

[疑難排解：不支援度量系列](#) (位於 p. 120)

[疑難排解：度量系列不完整](#) (位於 p. 120)

[疑難排解：廠商憑證運算式錯誤](#) (位於 p. 121)

## 疑難排解：無法建立廠商憑證

**徵狀：**

我嘗試建立廠商憑證，但是出現失敗的錯誤訊息。

**解決方法：**

開啟 Data Aggregator 安裝目錄中的 karaf 記錄檔，並按照下列步驟進行：

1. 找出 MIB 名稱字串，或您選取的度量系列名稱。
2. 檢閱例外的堆疊追蹤，並找出 CertManagerException 及錯誤的原因。發生例外之後，會顯示錯誤的原因。

**範例：**運算式剖析器無法處理 Token 之後的 ++，如下所示：

```
Caused by: com.ca.im.dm.certmgr.interfaces.CertManagerException: Tech Cert:  
{http://im.ca.com/normalizer}NormalizedCPUInfo, Unable to compile expression:  
[Error: expected end of statement but encountered: e]  
[Near : {... stemID ++ extremeSystemBoardID ....}]
```

3. 依據提供的原因修正錯誤。確認符合下列需求：
  - 運算式群組不可包含混合的純量及表格項目。
  - 運算式必須包含有效的語法。
  - 對於度量系列變數，至少定義一個運算式。
  - 至少定義兩個度量系列變數，尤其「名稱」及「索引」是必要的變數 (對於僅純量度量則非必要)。
  - 運算式中使用的廠商憑證變數必須來自選擇的 MIB 表格 (在使用者介面中有效)。

## 疑難排解：不支援度量系列

### 徵狀：

我建立監控設定檔輪詢一個裝置集合的度量系列。不過，在 [輪詢的度量系列] 表格中，其中一個度量系列出現「不支援」的狀態。

### 解決方法：

若要更正問題，請按照下列步驟進行：

1. 驗證輪詢的裝置可回應 SNMP 查詢。
2. 按一下不支援的度量系列，導覽至該度量系列。
3. 驗證廠商憑證支援度量系列。如果未定義任何廠商憑證，請建立自訂廠商憑證。
4. 驗證裝置上支援所有主要的廠商憑證屬性。若支援主要廠商憑證屬性，請導覽回裝置，並選取已新增自訂廠商憑證的度量系列，然後按一下 [更新度量系列]。

隨即更新您的裝置配置。

## 疑難排解：度量系列不完整

### 徵狀：

我成功匯入自訂度量系列，但是後來發現度量定義不完整。例如，<Name> 內容的長度上限為 32 個字元。如果超過這個限制，會導致同步處理問題。

### 解決方法：

小心刪除自訂度量系列，如下所示：

1. 找出 `IMDataAggregator/apache-karaf-2.3.0/deploy` 目錄。
2. 刪除先前為度量系列建立並部署的 XML 檔。這些檔案的名稱如下所示：
  - `im.ca.com-normalizer-<technology>.xml`
  - `im.ca.com-inventory-<technology>.xml`

若適用，也一併刪除對於廠商憑證建立的檔案：

  - `im.ca.com-certifications-snmp-<vendor>.xml`

3. 執行下列命令重新啟動 Data Aggregator：

```
service dadaemon restart
```

Data Aggregator 重新啟動後，確認先前匯入的度量系列或廠商憑證未出現在 CA Performance Center 中。另外，也將刪除所有先前對於此自訂憑證所搜尋的元件。

4. 按一下 CA Performance Center 中的 [管理] > [資料來源]。

5. 選取 Data Aggregator，然後按一下 [重新同步] 按鈕。

Data Aggregator 與 CA Performance Center 之間其餘度量系列的元件將同步處理。

6. 編輯並更正自訂度量系列 XML 檔。

7. 匯入更正的度量系列 XML 檔。

## 疑難排解：廠商憑證運算式錯誤

### 徵狀：

MVEL 編譯程式可能無法提供無效運算式的評估例外 (錯誤)。對於某語法錯誤，會發生這種情況，包括但不限於遺失或未關閉的括弧，以及多個星號。

編譯不正確的運算式時，只有在使用適當的變數執行運算式評估時，才會出現錯誤狀況。未填入預期運算式目標的資料庫欄。

### 解決方法：

使用下列步驟，啓動 ExpressionEvaluator 的偵錯記錄：

1. 找出 IMDataAggregator/apache-karaf-2.3.0/etc 目錄。

2. 開啓 org.ops4j.pax.logging.cfg 檔案，並建立下列項目：

```
log4j.logger.com.ca.im.core.expressionevaluator=DEBUG
```

3. 執行下列命令重新啟動 Data Aggregator：

```
service dadaemon restart
```

4. 在 IMDataAggregator/apache-karaf-2.3.0/data/log 目錄的 karaf.log 檔案中，尋找評估例外。