

# CA 性能管理 Data Aggregator

超级用户认证指南

2.4



本文档包括内嵌帮助系统和以电子形式分发的材料（以下简称“文档”），其仅供参考，CA 随时可对其进行更改或撤销。

未经 CA 事先书面同意，不得擅自复制、转让、翻印、透露、修改或转录本文档的全部或部分内容。本文档属于 CA 的机密和专有信息，不得擅自透露，或除以下协议中所允许的用途，不得用于其他任何用途：(i) 您与 CA 之间关于使用与本文档相关的 CA 软件的单独协议；或者 (ii) 您与 CA 之间单独的保密协议。

尽管有上述规定，但如果您为本文档中所指的软件产品的授权用户，则您可打印或提供合理数量的本文档副本，供您及您的雇员内部用于与该软件相关的用途，前提是所有 CA 版权声明和标识必须附在每一份副本上。

打印或提供本文档副本的权利仅限于此类软件所适用的许可协议的有效期内。如果该许可因任何原因而终止，您应负责向 CA 书面证明已将本文档的所有副本和部分副本已退还给 CA 或被销毁。

在所适用的法律允许的范围内，CA 按照“现状”提供本文档，不附带任何保证，包括但不限于商品适销性、适用于特定目的或不侵权的默示保证。CA 在任何情况下对您或其他第三方由于使用本文档所造成的直接或间接的损失或损害都不负任何责任，包括但不限于利润损失、投资受损、业务中断、信誉损失或数据丢失，即使 CA 已经被提前明确告知这种损失或损害的可能性。

本文档中涉及的任何软件产品的使用均应遵照有关许可协议的规定且根据本声明中的条款不得以任何方式修改此许可协议。

本文档由 CA 制作。

仅提供“有限权利”。美国政府使用、复制或透露本系统受 FAR Sections 12.212、52.227-14 和 52.227-19(c)(1) - (2) 以及 DFARS Section 252.227-7014(b)(3) 的相关条款或其后续条款的限制。

版权所有 © 2014 CA。保留所有权利。此处涉及的所有商标、商品名称、服务标识和徽标均归其各自公司所有。

## CA Technologies 产品引用

本文档引用以下 CA Technologies 产品：

- CA 性能管理 Data Aggregator (Data Aggregator)
- Data Collector
- CA Performance Center

## 联系技术支持

要获取在线技术帮助以及办公地址、主要服务时间和电话号码的完整列表，请联系技术支持：<http://www.ca.com/worldwide>。



# 目录

---

<b>第 1 章： 简介</b>	<b>9</b>
关于本指南.....	9
设备支持的工作原理.....	10
通过自行认证提供设备支持 .....	11
自定义认证场景.....	12
先决条件.....	12
<b>第 2 章： 正在下载架构和示例文件</b>	<b>15</b>
<b>第 3 章： 正在创建自定义组件</b>	<b>17</b>
创建组件 XML 模板.....	17
正在了解组件 XML 结构.....	18
基本属性.....	19
ItemSyncDefinition.....	20
不支持自定义组件的属性.....	22
导入自定义组件 .....	22
验证自定义组件结果.....	23
更新自定义组件.....	24
<b>第 4 章： 创建自定义度量标准系列</b>	<b>25</b>
指南结构.....	25
创建度量标准系列 XML 模板 .....	26
正在了解度量标准系列 XML 结构 .....	26
基本属性.....	31
ComponentFacets .....	32
ItemFacets.....	33
SourceFacetTypes 属性.....	34
AttributeGroup（度量标准系列） .....	34
BaselineDefinitions .....	41
表达式.....	42
层次结构.....	44
ComponentReconciliation .....	45
ItemReconciliation .....	45
MatchAlgorithms .....	46
ReconfigDetectionAttr .....	49
自定义度量标准系列不支持的属性.....	49
导入自定义度量标准系列 .....	50

---

验证自定义度量标准系列结果 .....	51
其他度量标准支持 .....	51
更新自定义度量标准系列 .....	56

## 第 5 章：正在创建自定义供应商认证 59

创建供应商认证 XML 模板 .....	59
正在了解供应商认证 XML 结构 .....	60
基本属性 .....	62
AttributeGroup (供应商认证) .....	64
ExpressionGroup .....	69
HierarchyList .....	72
IndexTagList .....	73
导入自定义供应商认证 .....	75
使用 REST 客户端导入自定义供应商认证 .....	75
使用自定义供应商认证安装程序 .....	76
验证自定义供应商认证结果 .....	77
筛选支持 .....	78
多个 MIB 表支持 .....	81
AttributeGroup (多个 MIB 表) .....	84
UseIndex .....	85
IndexTagList (多个 MIB 表) .....	86
更新自定义供应商认证 .....	86

## 附录 A：供应商认证表达式：表达式运算符、函数和全局变量 89

表达式运算符 .....	89
函数和全局变量 .....	91
availabilityWithSysUptime 函数 .....	91
mapModel 函数 .....	92
mapVendor 函数 .....	93
snmpConstArrayMap 函数 .....	94
mvelInfo 函数 .....	95
mvelWarn 函数 .....	96
mvelError 函数 .....	98
mvelDebug 函数 .....	99
mvelDebug 函数 .....	100
snmpCounter64 函数 .....	101
snmpGetUpSinceTime 函数 .....	102
snmpMax 函数 .....	103
snmpObjectIDToASCIIString 函数 .....	104
snmpOIDParser 函数 .....	104
snmpOctetStringFloat 函数 .....	106
snmpProtectedDiv 函数 .....	107

---

snmpRound 函数 .....	108
snmpStringParser 函数 .....	109
snmpSvcs 函数 .....	109
storePortReconfig 函数 .....	111
全局变量 .....	112

## 附录 B：故障排除

113

故障排除：无法创建供应商认证 .....	113
故障排除：度量标准系列不受支持 .....	114
故障排除：度量标准系列不完整 .....	114
故障排除：供应商认证表达式不正确 .....	115



# 第 1 章：简介

---

此部分包含以下主题：

- [关于本指南 \(p. 9\)](#)
- [设备支持的工作原理 \(p. 10\)](#)
- [通过自行认证提供设备支持 \(p. 11\)](#)
- [自定义认证场景 \(p. 12\)](#)
- [先决条件 \(p. 12\)](#)

## 关于本指南

Data Aggregator 提供供应商自行认证接口，以便您扩展 CA 性能管理 监视 支持。自我认证可让您通过供应商 MIB 的全新或自定义组件和度量标 准，在 Data Aggregator 中补充现有度量标准系列。

您可以创建供应商认证或修改现有设备或组件支持。本指南讨论了如何 通过自定义度量标准系列以及自定义供应商支持来添加新的技术支持。

本指南解决了创建自定义供应商认证和度量标准系列的高级场景。要求 掌握 XML 和架构文件的基本知识。

**注意：** Data Aggregator 提供了用于创建自定义供应商认证和度量标准系 列的基本方法和高级方法。基本方法比较简单，包括使用用户界面为现 有支持技术（度量标准系列）添加供应商支持。此方法满足许多用户 的要求。而高级方法是以工厂认证格式为基础，并且提供一套完整 的功能。本指南对高级认证方法进行了说明。有关基本认证方法的信 息，请参见《*Data Aggregator 基本自行认证指南*》。

本指南将始终遵循具体示例。作为管理员，您希望监视框架转接永久虚 拟电路 (PVC)。开箱即用的 Data Aggregator 不支持监视框架转接 PVC，但 您可以使用自行认证来生成此支持。在本指南中，我们将提供多个步骤 来定义组件、创建自定义度量标准系列和定义自定义供应商认证，以便 监视框架转接永久虚拟电路 (PVC)。

## 设备支持的工作原理

**Data Aggregator** 使用度量标准系列和供应商认证支持供应商设备。这些组件通过协同工作来确定 **Data Aggregator** 收集设备的配置和运行度量标准的方式。了解设备支持在 **Data Aggregator** 中的工作原理有助于您确定您的设备是否在 **Data Aggregator** 中得到相应支持。如果没有，了解此过程可以帮助您调整设置以得到需要的结果。

**注意：**如有必要，您可以自定义度量标准系列、供应商认证，或同时自定义二者以添加对供应商设备的支持。

**Data Aggregator** 使用以下配置功能支持设备：

1. **发现配置文件**—通常是根据一系列 IP 地址，确定 **Data Aggregator** 在您的环境中发现的项。发现过程为其发现的每个项标识“类型”。

2. **设备集合**—将您的清单组织到包含相关项的组。根据项类型和 IP 地址，自动将项添加到设备集合中。

3. **监视配置文件**—控制设备集合的轮询比率，并确定要轮询的度量标准系列。监视配置文件可以轮询一个或多个度量标准系列。

**注意：**要确保您的系统没有超负荷轮询流量，请使用监控配置文件为不同的度量标准集调整轮询速度。

4. **度量标准系列**—控制为监视配置文件收集的度量标准。度量标准系列与一个或多个供应商认证相关联，这些供应商认证按优先级顺序列出。

**注意：**在监视配置文件中重复使用度量标准系列有助于帮助确保一致的数据报告。

5. **供应商认证**—将供应商 MIB 中的属性映射到度量标准系列中的度量标准。此外，还确定如何设置从某个项收集的度量标准的格式以在 CA Performance Center UI 和报告中使用。根据项供应商的不同，为该项提供的度量标准可能有所不同。映射这些值可确保报告的度量标准值保持一致，无论供应商是谁。多个供应商认证可以与单个度量标准系列相关联。在此类情况下，**Data Aggregator** 使用供应商认证的分级列表映射度量标准值。**Data Aggregator** 将会使用与已轮询项匹配的最高优先级供应商认证计算度量标准值。

**注意：**可以将 MIB（如 SNMP MIB）导入系统并将其作为构建供应商认证的一部分进行编译。

### 示例：路由器设备支持

在运行您的发现配置文件时，Data Aggregator 查找并将项标识为路由器。该路由器受管项将自动添加到“所有路由器”设备集合中。此设备集合与路由器监视配置文件相关联，该配置文件使用 CPU 和内存度量标准系列在设备上发现 CPU 和内存组件。这些度量标准系列也用于确定在为这些组件计算度量标准值时使用的供应商认证。基于此监控配置文件，Data Aggregator 每 5 分钟针对这些度量标准系列中的度量标准数据轮询您的路由器。例如，CPU 度量标准系列包含 CPU 空闲使用率、CPU 系统使用率和 CPU 有效使用率。最后，与度量标准系列相关联的供应商认证确定如何一致地计算原始度量标准数据并设置其格式。Data Aggregator 存储为您的路由器收集的度量标准数据，CA Performance Center 在 UI 和报告中使用该度量标准数据。

## 通过自行认证提供设备支持

Data Aggregator 使用预定义的认证支持常用供应商设备。认证指定如何为设备收集配置度量标准和操作度量标准。Data Aggregator 使用以下方法进行认证：

- 度量标准系列
- 供应商认证

当 Data Aggregator 不为设备提供预定义认证时，您如何收集数据？您可以为设备自行认证支持。

**注意：**度量标准系列和供应商认证是全局性的（即，不是承租方可识别的）。有关承租方的详细信息，请参阅《CA Performance Center 管理员指南》。

Data Aggregator 中的自行认证支持使您可以创建自定义供应商认证和/或自定义度量标准系列。确定您需要哪种方法，如下所示：

- **仅供应商认证**—默认情况下会轮询您希望的一组度量标准，但对于这些度量标准，Data Aggregator 不支持您的设备供应商 MIB。例如，Data Aggregator 提供了一个 CPU 度量标准系列来收集数据（如 CPU 使用率）。但是，您希望收集 Bargain Server 公司生产的服务器的 CPU 数据。使用制造商提供的 MIB，您可以为服务器 CPU 创建自定义供应商认证。
- **仅度量标准系列**—默认情况下会包含对设备供应商 MIB 的支持，但不会轮询某些 MIB 支持的度量标准。例如，供应商 MIB 支持进程的度量标准，但 Data Aggregator 不提供“进程”度量标准系列来收集该度量标准数据。

- 两种方法—在 Data Aggregator 不为设备供应商 MIB 或其度量标准提供支持时，同时创建度量标准系列和供应商认证。

## 自定义认证场景

开箱即用的 CA 性能管理 支持您网络基础架构中的公用供应商、度量标准和组件。技术认证门户列出 Data Aggregator 版本、供应商认证和度量标准系列的即用型认证：

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

但您可以通过创建自定义 CA 性能管理 认证来最大化您的基础架构管理权限。在您需要下列 CA 性能管理 支持时进行自定义：

- 新技术：
  - 定义新组件。
  - 创建自定义度量标准系列。
- 现有技术的新度量标准：
  - 从预定义（工厂）度量标准系列创建自定义度量标准系列。
  - 定义自定义度量标准系列的自定义供应商认证。
- 新供应商：
  - 创建工厂度量标准系列的自定义供应商认证。
  - 创建自定义度量标准系列的自定义供应商认证。

## 先决条件

下列信息描述了使用供应商认证 XML 的先决条件。

- CA 性能管理 Data Aggregator 要求 2.2.00 或更高版本。
- 要求具备 CA 性能管理 Data Aggregator 《自行认证指南》中所述有关“认证”的扎实知识以及使用供应商认证向导的丰富经验。
- XML 编辑器。

建议采用可验证 XML 架构的 XML 编辑器。遵循 XML 编辑器指南，使用 XSD 文件。

**注意：**本指南中的示例是通过 XML Notepad 2007 for Windows 创建和验证的，可从 Microsoft® 下载中心免费下载此工具。

- MIB 浏览器。例如，支持 iReasoning 免费版。

- CA 性能管理 Data Aggregator 服务器连接。
- 为了支持我们在本指南中使用的示例，要求具备 FRAMERELAY - RFC1315 - MIB 和 IF-MIB。



## 第 2 章： 正在下载架构和示例文件

---

在您创建自定义度量标准系列和供应商认证 `xml` 文件之前，请下载并审阅相关的架构 `XSD` 文件和示例 `xml` 文件。您需要此架构来创建个人 `XML` 文件。在创建个人示例文件之前审阅这些文件有助于确保 `XML` 内容的准确性。

`Data Aggregator` 提供的架构文件中包含有关元素类型、匹配项和允许长度的详细信息。文件中也包含可提供更多信息的批注，比如允许字符和命名约定。

通过在 `Web` 浏览器地址字段中输入以下 `URL` 来下载架构和示例 `xml` 文件。主机名是 `Data Aggregator` 主机，默认端口是 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：正在创建自定义组件

---

此部分包含以下主题：

- [创建组件 XML 模板 \(p. 17\)](#)
- [正在了解组件 XML 结构 \(p. 18\)](#)
- [导入自定义组件 \(p. 22\)](#)
- [验证自定义组件结果 \(p. 23\)](#)
- [更新自定义组件 \(p. 24\)](#)

## 创建组件 XML 模板

如要创建可用作创建自定义组件模板的示例 XML 文件，请使用预定义 REST 客户端。

先从检索现有组件定义的列表开始。之后可以验证您所需的组件是否受到支持。

**遵循这些步骤：**

1. 将 REST 客户端设置为与 Data Aggregator 服务器连接。
2. 对于 REST 客户端中的 Data Aggregator Web 服务 API，输入下列 URL：  
`http://da_hostname:8581/typecatalog/components`  
将会显示所有适用的组件列表。
3. 验证您需要的组件是否受到支持。

如果您需要的组件不受支持，请查看和导出类似于您所要创建自定义组件的单个组件定义。

**遵循这些步骤：**

1. 如要检索类似于您所需组件的具体组件，请输入下列 URL：

`http://da-hostname:8581/typecatalog/components/name`

***name***

是现有具体组件的名称，比如 NormalizedCPUInfo。

2. 在“方法”选项卡上选择“GET”。
3. 运行该方法。

已返回的 XML 包含有组件定义。

您可以将此 XML 用作模板来创建新组件。

4. 根据需要，将组件 XML 复制到文本文件，再对其进行修改。有关 XML 结构的示例，请参见“[了解组件 XML \(p. 18\)](#)”。

## 正在了解组件 XML 结构

定义与设备关联的组件项类的设备组件。将提供许多工厂定义的组件，但自定义组件通常为自定义度量标准系列定义的。可以定义可选 **ItemSyncDefinition** 的组件，使组件项同步到 CA Performance Center。之后您可以在清单、列表、组以及上下文页面中查看组件。

下列是支持我们框架转接 PVC 示例的组件定义 XML 示例。已添加新组件 frPVC：

```
<?xml version="1.0" encoding="UTF-8"?>
<!- -由类型目录本地管理器自动生成。-->
<DataModel namespace="http://im.ca.com/inventory"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="ComponentFacet.xsd">
  <FacetType name="frPVC">
    <Documentation>框架转接 PVC</Documentation>
    <FacetOf namespace="http://im.ca.com/core" name="Item" />
    <Component>true</Component>
    <ItemSyncDefinition itemTypeName="component"
      itemSubtypeName="frpvc" itemTypeLabel="FrameRelayPVC"
      itemTypeLabelPlural="FrameRelayPCVs" categorize="false"
      groupBy="false" context="true">
    </ItemSyncDefinition>
  </FacetType>
</DataModel>
```

将 **ItemSyncDefinition** 的上下文设置为“True”，以启用对自定义上下文页面的链接。您可以从在清单、设备组件列表中显示的 frPVC 设备组件导航至本页。将其设置为“True”，还可使您选择度量标准系列作为“上下文”，从而使您的自定义度量标准系列在“动态趋势”图表类型中可用。有关更多信息，请参阅 [ItemSyncDefinition \(p. 20\)](#)。

## 基本属性

自定义组件的基本属性可帮助您将其与您创建的其他自定义组件分开来。

### FacetType/name

指定组件名称。每个组件必须具有在系统内部用于标识的唯一名称。请谨慎选择名称，以尽量减少与未来类似组件的命名冲突。例如，定义命名方案有助于确保这些组件的名称保持唯一。

**注意：**绝不能对外公开此名称。要在用户界面中显示组件名称，请使用 `ItemSyncDefinition`、`itemTypeLabel` 和 `itemTypeLabelPlural` 元素。

**可否更新：**否

**可能的值：**字母数字和下划线。不允许使用点和虚线。

### 文档

指定组件的内部注释。为使这些注释生效，我们建议您说明添加或更改组件的原因及时间。

**可否更新：**是

**可能的值：**纯文本

**更新效果：**无

**更新何时生效：**立即生效

**更新生效所需的操作：**无

### 组件

声明此项是组件。

**可否更新：**否

**可能的值：**True

### FacetOf

声明此组件是项。

**可否更新：**否

**可能的值：**namespace="<http://im.ca.com/core>" name="Item"

## ItemSyncDefinition

ItemSyncDefinition 属性是可选的。此属性指定组件项如何在 CA Performance Center 中同步和显示。如果未指定组件项，它们将不会在 CA Performance Center 清单列表中显示（例如，设备组件）。但仍可以在自定义视图上对其进行报告。

### ItemSyncDefinition/itemTypeName

指定项目类型。对于自定义组件，此值必须是“组件”。

**可否更新：** 否

**可能的值：** 组件

### ItemSyncDefinition/itemSubtypeName

指定 CA Performance Center 中组件的内部名称。此值对于所有组件必须是唯一的。采用可避免与未来工厂和自定义组件产生冲突的命名约定，比如采用可表示贵组织的前缀“acmeFan”。

**可否更新：** 否

**可能的值：** 字母数字，对所有组件保持唯一

### ItemSyncDefinition/itemTypeLabel

指定显示此类单个组件时使用的用户界面标签。例如，此值将用在设备组件 UI“类型”列的清单中

**可否更新：** 是

**可能的值：** 纯文本，对所有组件保持唯一

**更新效果：** 标签将显示在 CA Performance Center 清单用户界面中。

**更新何时生效：** 允许发生重新同步及多达 15 分钟的时间完成更新。

**更新生效所需的操作：** 无

### ItemSyncDefinition/itemTypeLabelPlural

指定显示此类多个组件时使用的用户界面标签。供清单菜单（参见 **groupBy**）和组名称（参见 **分类**）。

**可否更新：** 是

**可能的值：** 纯文本，对所有组件保持唯一

**更新效果：** 标签将显示在 CA Performance Center 清单 UI 中。

**更新何时生效：** 允许发生重新同步及多达 15 分钟的时间完成更新。

**更新生效所需的操作：** 无

### ItemSyncDefinition/categorize

指示 CA Performance Center 在所有项的清单下创建清单组。此组包含该组件类型的所有项。该组被命名为 '*{itemTypeLabelPlural}*'。

**注意：**已创建的清单组不可用于报告显示板。相反，此组仅可供清单使用。如果选择该组进行报告，则不会显示数据。而其他基于设备的清单组（所有项的清单之下）可用于报告，比如“路由器”和“服务器”。但基于组件的清单组不可用于报告，比如“设备组件”。

**可否更新：**是

**可能的值：**True、False

**更新效果：**创建或移除 CA Performance Center 中的清单组。对于 CA Performance Center 组管理员，在“管理组”页面上，该组 (*{itemTypeLabelPlural}*) 将创建于所有项的清单之下。

**更新何时生效：**允许发生重新同步及多达 30 分钟的时间完成更新。

**更新生效所需的操作：**项通常在 30 分钟内显示在组中显示。如未重新同步，请手动重新同步 Data Aggregator 数据源。在单击重新同步确认按钮之前，务必选择“执行全面重新同步”。

### ItemSyncDefinition/groupBy

指示 CA Performance Center 创建清单菜单项（在清单之下），以查看此组件项类型的所有项。菜单被命名为 "*{itemTypeLabelPlural}*"。在设置视图上下文时，此属性也将使组件类型显示在上下文类型下拉菜单中。如果是 False，组件将会显示在具有 *{itemTypeLabel}* 类型的设备组件表的清单中。groupBy 属性不会创建组（参见分类）。

**可否更新：**是

**可能的值：**True、False

**更新效果：**如果是 True，将会创建菜单项，如果是 False，将会删除菜单项。

**更新何时生效：**允许发生重新同步及多达 15 分钟的时间完成更新。

**更新生效所需的操作：**无

### ItemSyncDefinition/context

在清单组件视图中，将每个组件项名称变为上下文超链接，在单击时，导航到单个组件上下文页面。

**可否更新：**是/否

**可能的值：**纯文本

**更新效果：**在清单组件视图中，将每个组件项名称变为上下文超链接。

**更新何时生效：**允许发生重新同步及多达 15 分钟的时间完成更新。

**更新生效所需的操作：**无

### 移除 ItemSyncDefinition

完全移除 ItemSyncDefinition 需要具体的程序。

遵循这些步骤:

1. 移除“ItemSyncDefinition”部分，包括 <ItemSyncDefinition> 开始和结束标记。
2. 在应用对 Data Aggregator 的组件更改之后，以管理员的身份登录到 CA Performance Center。
3. 在“数据源”页面上选择 DataAggregator 数据源。
4. 单击“重新同步”按钮。
5. 单击“执行全面重新同步”选项，再单击“重新同步”确认按钮。

重新同步过程将会开始。允许多达 15-30 分钟的时间，以便对重新同步进行更改。

在此过程完成时，所有为组件定义的 ItemSyncDefinition CA Performance Center 行为都将被移除。

### 不支持自定义组件的属性

自定义组件不支持下列属性:

- 属性
- Web 服务
- ItemSyncDefinition/isDeviceComponent\*
- ItemSyncDefinition/mapped\*
- ItemSyncDefinition/itemProperty

\* 这些属性可能会出现在您的 XML 中，但无法将其设置为 True。

## 导入自定义组件

在您将自定义组件导入到 Data Aggregator 之后，将会有新组件来支持自定义度量标准系列。

Data Aggregator 提供了支持导入新组件的 Web 服务。使用任何 REST 客户端。我们建议使用带有 CA 性能管理的 RESTClient。如未安装，可从 <http://code.google.com/p/rest-client/> 获取 RESTClient GUI JAR 文件。

在使用 RESTClient 时，请注意下列信息：

- 如要启动，请双击 JAR 文件。
- 在使用 POST 发送 XML 时，务必将字符集设置为 UTF-8。  
要查看和验证此设置，请单击“编辑内容 - 类型和字符集”按钮。
- 您还可以自动缩进响应正文，如下所示：
  1. 单击“工具”菜单上的“选项”。
  2. 选择“Etc.”选项卡。
  3. 选择“自动缩进响应正文”，单击“确定”。

**遵循这些步骤：**

1. 将 `http://da-hostname:8581/typecatalog/components` 作为 URL 输入。
2. 选择“POST”作为方法。
3. 在“正文”设置中选择“application/xml”作为“正文内容类型”。  
**重要说明！ 无法设置内容类型导致产生 404 错误。**
4. 将您的自定义组件 XML 复制粘贴到“正文”字段。
5. 运行该方法。

您的自定义组件已导入。如果无错误发生，“HTTP 响应”部分中的“状态”字段将会显示：

`HTTP/1.1 200 OK`

**注意：**如有任何其他返回代码，表示在更新您的自定义组件时发生错误。更正错误，通过再次执行 POST 来重试更新组件。

**重要说明！** 为避免可能产生的数据丢失，请一定要在每次创建或更新供应商认证、度量标准系列或组件时都备份您的 `deploy` 目录。

## 验证自定义组件结果

在您导入自定义组件 XML 之后，请验证结果。在我们的示例中，我们将验证是否已成功导入 frPVC 组件。

**遵循这些步骤：**

1. 在 REST 客户端中输入以下 URL：

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

一个页面将会显示自定义框架转接组件 XML。

您还能发现新组件。在同步之后，您可以使用设备组件列表的清单来查看组件。

## 更新自定义组件

您可以更新现有自定义组件。

**注意：**有关在更新组件时更新标记或属性效果的信息，请查看自定义组件 XML 详细信息。尤其是，请查看具体的属性描述。

**遵循这些步骤：**

1. 在 URL 字段输入下列地址：

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

**名称**

是要更新的自定义组件名称。

2. 在“方法”选项卡中选择“PUT”。
3. 将已更新的组件 XML 复制粘贴到“正文”选项卡的“编辑”字段，将内容类型设置为 application/xml。

**重要说明！ 无法设置内容类型导致产生 404 错误。**

4. 单击 URL 字段旁的“转到”按钮。

您的自定义组件已更新。如果无错误发生，“HTTP 响应”部分中的“状态”字段将会显示：

`HTTP/1.1 200 OK`

**注意：**如有任何其他返回代码，表示在更新您的自定义组件时发生错误。更正错误，通过再次执行 PUT 来重试更新组件。

**重要说明！ 为避免可能产生的数据丢失，请一定要在每次创建或更新供应商认证、度量标准系列或组件时都备份您的 deploy 目录。**

**详细信息：**

[正在了解组件 XML 结构 \(p. 18\)](#)

# 第 4 章： 创建自定义度量标准系列

---

在成功导入之后，将会有新的度量标准系列来支持自定义供应商认证。度量标准系列还可用于关联监视配置文件，以进行发现和轮询。

此部分包含以下主题：

- [指南结构 \(p. 25\)](#)
- [创建度量标准系列 XML 模板 \(p. 26\)](#)
- [正在了解度量标准系列 XML 结构 \(p. 26\)](#)
- [导入自定义度量标准系列 \(p. 50\)](#)
- [验证自定义度量标准系列结果 \(p. 51\)](#)
- [其他度量标准支持 \(p. 51\)](#)
- [更新自定义度量标准系列 \(p. 56\)](#)

## 指南结构

此指南解释必要的 XXX。 TO COME

VC:

- 创建 vc
- 编辑 vc
- 导入 vc
- 导出 vc

工厂度量标准系列定义要监视的最常见的度量标准属性。但是，当您要为新度量标准属性收集数据时，可以创建一个自定义度量标准系列。例如，如果不存在用于收集进程数据的度量标准系列，则可以创建一个。然后使用新的度量标准系列创建供应商认证，以监视进程的度量标准属性。

## 创建度量标准系列 XML 模板

使用首选 REST 客户端来创建示例 xml 文件，您可以将其用作模板来创建自定义度量标准系列。

先从检索现有度量标准系列的列表开始。之后可以验证您所需的度量标准系列是否受到支持。

**遵循这些步骤:**

1. 将 REST 客户端设置为与 Data Aggregator 服务器连接。
2. 对于 REST 客户端中的 Data Aggregator Web 服务 API，输入下列 URL:

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

将会显示所有适用度量标准系列的列表。

3. 验证您所需的度量标准系列是否受到支持。

如果您所需的度量标准系列不受支持，请查看和导出类似于您所要创建自定义度量标准系列的度量标准系列。

**遵循这些步骤:**

1. 如要检索类似于您所需度量标准系列的具体度量标准系列，请输入下列 URL:

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

**name** 是具体现有度量标准系列的名称，比如 NormalizedCPUInfo。

2. 在“方法”选项卡中选择“GET”。
3. 运行该方法。

已返回的 XML 包含度量标准系列信息。

您可以将此 XML 用作模板来创建自定义度量标准系列。

4. 根据需要，将度量标准系列 XML 复制到文本文件，再对其进行修改。  
有关 XML 结构的示例，请参见“[了解度量标准系列 XML \(p. 26\)](#)”。

## 正在了解度量标准系列 XML 结构

度量标准系列定义了要为指定技术收集和报告的度量标准集合。这些度量标准已进行过规范化处理，因此无论供应商（数据源）为何，报告都是统一的。并非所有的供应商都对度量标准系列中的每个度量标准提供了值，而且也不需要所有的度量标准。如果供应商未提供值，则度量标准为“空”。此外，任何基于空度量标准的报告视图都为“空”。

度量标准系列还定义了发现过程中捕获的属性，比如项名称和索引。此外，还会定义用以调节组件匹配的发现规则。您可以在配置文件中添加度量标准系列。对于每个与配置文件关联的设备收集中的设备，监视配置文件中的度量标准系列集合确定了要收集哪些度量标准，

下列是支持我们框架转接 PVC 示例的度量标准系列 XML 示例。请注意我们的示例自定义组件 frPVC 如何包含在 ComponentFacets 部分当中(粗体显示，以供说明)：

```
<?xml version="1.0" encoding="UTF-8" ?>
<!--由类型目录本地管理器自动生成。 -->
<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.NormalizedFacet
    DescriptorImpl">
    <Documentation>框架转接永久虚拟电路</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>框架转接电路名称</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>有关框架转接电路的描述</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>虚拟电路创建之后向后拥塞</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>总和</RollupStrategy>
    <AttributeDisplayName />
    <Percentile>0</Percentile>
</Attribute>
<Attribute name="FECNIn" type="Double">
    <Documentation>虚拟电路创建之后向前拥塞</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>总和</RollupStrategy>
    <AttributeDisplayName />
    <Percentile>0</Percentile>
</Attribute>
<Attribute name="FramesIn" type="Double">
    <Documentation>虚拟电路创建之后接收的框架</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>总和</RollupStrategy>
    <AttributeDisplayName />
    <Percentile>0</Percentile>
</Attribute>
<Attribute name="FramesOut" type="Double">
    <Documentation>虚拟电路创建之后发送的框架</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>总和</RollupStrategy>
<AttributeDisplayName />
<Percentile>0</Percentile>
</Attribute>
<Attribute name="BytesIn" type="Double">
<Documentation>虚拟电路创建之后接收的字节</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>总和</RollupStrategy>
<AttributeDisplayName />
<Percentile>0</Percentile>
</Attribute>
<Attribute name="BytesOut" type="Double">
<Documentation>虚拟电路创建之后发送的字节</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>总和</RollupStrategy>
<AttributeDisplayName />
<Percentile>0</Percentile>
</Attribute>
</AttributeGroup>
<Attribute name="SourceFacetTypes" cached="true"
list="true" persistent="true" type="QName">
<Documentation />
</Attribute>
<DisplayName>框架转接 PVC </DisplayName>
<Expressions>
<ExpressionGroup
destCert="http://im.ca.com/core}Item">
<Expression destAttr="Name">名称</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>

```

## 基本属性

自定义度量标准系列的基本属性可帮助您将其与您创建的其他自定义度量标准系列区分开来。

在您确定基本属性时，请注意下列限制条件：

- 各个度量标准系列的 FacetType/name、DisplayName 和 TableName 属性必须是唯一的。
- 协议标记始终是 IMDB。
- 规范化标记始终是 True。
- 设置 FacetType/descriptorClass 属性及全部 DataModel 和 FacetOf 属性。

### FacetType/name

指定度量标准系列名称。对于各个度量标准系列，其名称必须是唯一的，以便在系统内部进行标识。请谨慎选择名称，以尽量减少与未来类似度量标准系列的命名冲突。例如，定义命名方案有助于确保这些度量标准系列的名称保持唯一。

**注意：** 绝不能对外公开此名称。如要在用户界面中显示度量标准系列名称，请使用 DisplayName 元素。

**可否更新：** 否

**可能的值：** 字母数字和下划线。不允许使用点和虚线。在所有度量标准系列中必须保持唯一。

### DisplayName

指定要在用户界面中显示的度量标准系列名称。

**可否更新：** 是

**可能的值：** 纯文本。在所有度量标准系列中必须保持唯一。

**更新效果：** 更改为管理员用户界面中显示的名称。

**更新何时生效：** 立即生效

**更新生效所需的操作：** 刷新用户界面。

## 文档

指定度量标准系列的外部描述。为使这些注释生效，我们建议您说明添加或更改度量标准系列的原因及时间。

**可否更新：** 是

**可能的值：** 纯文本

**更新效果：** 无

## TableName

指定用于存储度量标准系列所收集度量标准的数据库表名称。

**可否更新：** 是

**可能的值：** 大写字母数字和下划线。必须以字母开头。在所有度量标准系列中必须保持唯一。

**示例：** PROCESS\_STATS

**更新效果：** 轮询数据存储在全新的数据库表集合中。

**重要说明！** 在您更新 TableName 时，旧有的轮询数据将会丢失。旧有的报告视图将会断开。

**更新何时生效：** 立即生效。在创建新视图之前，CA Performance Center 加载新 MIB 文件时将会有长达 5 分钟的延迟。

**更新生效所需的操作：** 必须重新创建视图。

## ComponentFacets

ComponentsFacets 部分列出了发现期间创建的方面。发现标识了作为设备组件的项或创建了各项之间的层次结构关系。

### Facet

指定在组件发现期间依附于组件项的方面。

**可否更新：** 是

**可能的值：** 方面的 QName

**更新效果：** 如果将组件方面同步到 CA Performance Center，则组件将在 CA Performance Center 中可见。

**更新何时生效：** 重新发现时

**更新生效所需的操作：** 删除设备，重新发现。

详细信息：

[层次结构 \(p. 44\)](#)  
[ComponentReconciliation \(p. 45\)](#)

## ItemFacets

**重要说明！ ItemFacets 是可能会发生更改以支持未来复杂度量标准系列结构的新部分。 禁止使用。**

ItemFacets 列出了发现期间创建的可将项标识为设备的方面。

### Facet

指定在发现期间依附于项的方面。

**可否更新：**是

**可能的值：**方面的 QName

**更新效果：**组件在特定方面的 REST 服务上可见。如果将组件方面同步到 CA Performance Center，则组件将在 CA Performance Center 中可见。

**更新何时生效：**重新发现时

**更新生效所需的操作：**删除设备，重新发现。

### 示例：

```
<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>
  <! --IPDomainID 属性可通过发现进行填充 -->
  <Facet>{http://im.ca.com/core}IPDomainMember</Facet>
</ItemFacets>
```

详细信息：

[ItemReconciliation \(p. 45\)](#)

## SourceFacetTypes 属性

SourceFacetTypes 是发现所必需，而且必须对其进行定义的属性。

使用这些所需值：

- 名称：SourceFacetTypes
- 类型：QName
- 缓存：True
- 持续：True
- 列表：True

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

可否更新：否

## AttributeGroup（度量标准系列）

AttributeGroup 是项发现属性和度量标准属性的集合。项发现属性是在发现期间设置的，比如项描述。度量标准属性在轮询期间收集的。下列信息描述了您在 AttributeGroup 部分中使用的元素。

将 AttributeGroup/list 和 AttributeGroup/external 属性设置为 True。这些属性指定的各个属性代表从外部源获取的值列表。自定义下列 XML 元素：

### AttributeGroup/name

指定属性组名称。符合“<FacetType/name>Group”命名规则。

可否更新：是

可能的值：纯文本

更新效果：无

### 文档

(可选) 指定属性组的说明。

可否更新：是

可能的值：纯文本

更新效果：无

## 常规属性（度量标准系列）

所有度量标准系列的常规属性如下所示：

### Attribute/name

指定唯一的内部名称。对于度量标准，此名称还可用于命名数据库列。

**注意：** 绝不能对外公开此名称。要在用户界面中显示属性名称，请使用 AttributeDisplayName 元素。

**可否更新：** 是

**可能的值：** 字母数字和下划线。

**更新效果：** 对于度量标准，此属性的值将存储在与已更新名称相对应的新数据库列中。用户将失去为此度量标准收集的历史数据（具有原名称）。对此度量标准的自定义报告将会失败。

**更新何时生效：** 下次轮询时

**更新生效所需的操作：** 无

### Attribute/type

表明此属性的数据类型。最常用的数据类型是 Int、Long、Double、String 或 ObjectId。数据库将以浮点的形式存储度量标准属性。因此，这些属性必须采用数字类型（我们建议采用 Double 型）。其他类型用于项属性。

**可否更新：** 是

**可能的值：** Boolean、Int、Long、Double（浮点）、BigInteger、String、DateTime、IPAddress、MACaddress、IPSubnet、OctetString（十六进位表达式）、ObjectId、ItemID、QName（限定名称）

**注意：** 类型名称将不区分大小写，例如，“boolean”与“Boolean”相同。

**更新效果：** 对于度量标准，无。所有度量标准都将以浮点形式存储在数据库中。对于项属性，必须删除和重新发现设备。

**更新何时生效：** 对于度量标准，下次轮询时 对于项属性，执行重新发现。

**更新生效所需的操作：** 对于度量标准，无 对于项属性，删除设备，重新发现。

### **AttributeDisplayName**

指定运算符和管理员界面中显示的值。

**可否更新:** 是

**可能的值:** 字母数字、空格和下划线。

**更新效果:** 度量标准在度量标准系列 UI 和自定义报告中体现了已更新的 AttributeDisplayName。

**更新何时生效:** 立即生效

**更新生效所需的操作:** 无

### **文档**

在用户界面中显示属性说明。在您将光标悬停在属性名称上时，工具提示中也将显示文档。

**可否更新:** 是

**可能的值:** 纯文本

**更新效果:** 将光标悬停在属性名称上来显示已更新的文档。

**更新何时生效:** 立即生效

**更新生效所需的操作:** 无

### **已轮询**

表明该属性是否已轮询。如果将其设置为 `False`，则仅能在发现期间对其进行访问。

**可否更新:** 是

**可能的值:** `True`、`False`

**更新效果:** 如果设置为 `False`，那么如果没有其他轮询 attribute/metric 在表达式中使用此 OID，则不会轮询与 attribute/metric 相对应的 OID。如果设置为 `True`，则会轮询与 attribute/metric 相对应的 OID。

**更新何时生效:** 下次轮询时

**更新生效所需的操作:** 无

**IsDbColumn**

在数据库表中存储其值。 **IsDbColumn** 用于度量标准属性。如果将 **Polled** 设置为 **True**，请将 **IsDbColumn** 值设置为 **True**。

**可否更新:** 是

**可能的值:** **True**、**False**

**更新效果:** 如果设置为 **False**，那么 **attribute/metric** 的数据将不会存储于数据库中。如果设置为 **True**，那么 **attribute/metric** 的数据将会存储于数据库中。

**更新何时生效:** 立即生效

**更新生效所需的操作:** 无

## 发现属性

对许多属性而言，仅有发现期间检索的值会存储于数据库中。将不会执行其他轮询或处理，比如基准评估。

所有度量标准系列必须存在“索引”和“名称”属性。“描述”属性是可选的。

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

支持层次结构的度量标准系列必须包含这些属性：

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

## 轮询和基准属性

下列信息描述了轮询和基准属性元素：

### 基准

指示是否要计算此属性的平均值。如果将其设置为 **True**，则必须定义相应的 **BaselineList** 定义。

**注意:** 基准属性要求将 **StandardDeviation** 属性设置为 **True**。

**可否更新:** 是

**可能的值:** **True**、**False**

**更新效果:** 为 **True** 时计算基准值。

**更新何时生效:** 下次轮询时

**更新生效所需的操作:** 无

### 最大值

指示是否要在累加期间计算此属性的最大值。在数据库表中创建“max\_”列。如果定义 RollupStrategy，那么还必须定义此属性。

**可否更新:** 是

**可能的值:** True、False

**更新效果:** True 提供了“最大值”的计算和报告字段。

**更新何时生效:** 下次轮询时

**更新生效所需的操作:** 无

### 最小值

指示是否要在累加期间计算此属性的最小值。在数据库表中创建“min\_”列。如果定义 RollupStrategy，那么还必须定义此属性。

**可否更新:** 是

**可能的值:** True、False

**更新效果:** True 提供了“最小值”的计算和报告字段。

**更新何时生效:** 下次轮询时

**更新生效所需的操作:** 无

### StandardDeviation

指示是否要在累加期间计算此属性的标准偏差。在数据库表中创建“std\_”列。如果定义 RollupStrategy，那么还必须定义此属性。

**可否更新:** 是

**可能的值:** True、False

**更新效果:** True 提供了“标准偏差”的计算和报告字段。

**更新何时生效:** 下次轮询时

**更新生效所需的操作:** 无

**DeviationFromBaseline**

要求将基准属性设置为 `True`。提供了两个额外的报告字段，即通过基准数据计算的“平均基准”和“偏差百分率”。这些字段不可用于构建自定义视图。未对数据库表进行任何更改。

**可否更新:** 是

**可能的值:** `True`、`False`

**更新效果:** `True` 提供了内部报告开发的“平均基准”和“偏差百分率”字段。

**更新何时生效:** 立即生效

**更新生效所需的操作:** 无

**百分位**

指示是否要在累加期间计算此属性的第 95 百分位。在数据库表中创建 '`pct_`' 列。如果定义 `RollupStrategy`，那么还必须定义此属性。

**可否更新:** 是

**可能的值:** `0`（零）、`95`

**更新效果:** 值 `95` 提供了“第 95 百分位”的计算和报道字段。零表示未执行任何计算，而且报告字段不可用。

**更新何时生效:** 下次轮询时

**更新生效所需的操作:** 无

**RollupStrategy**

在逐个累加轮询值期间，指定每个周期执行的运算。如果将轮询和 `IsDbColumn` 设置为 `True`，则需要此元素。

**可否更新:** 是

**可能的值:** `Sum`（计数器的总和），`Avg`（计量器的平均值）

**更新效果:** 指定的策略用于执行累加计算。

**更新何时生效:** 下次轮询时

**更新生效所需的操作:** 无

## 速率

提供了额外的报告字段，即作为 AVG（度量值/时间）计算的“平均速率”。未对数据库表进行任何更改。

**注意：**速率用于报告，但不在监视配置文件事件规则时使用。

**可否更新：**是

**可能的值：**True、False

**更新效果：**提供了报告的“平均速率”字段。

**更新何时生效：**立即生效

**更新生效所需的操作：**无

## 单位

指定报告中使用的单位标签名称。显示的实际标签将根据报告的语言设置进行转换。

**可否更新：**是

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

**更新效果：**指定的单元标签将显示在报告中。

**更新何时生效：**立即生效

**更新生效所需的操作：**无

## 示例：使用轮询和基准属性元素

```
<Attribute name="Utilization" type="double">
  <AttributeDisplayName>利用率</AttributeDisplayName>
  <AttributeAbbreviation>利用率</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>百分比</Units>
</Attribute>
```

## BaselineDefinitions

**BaselineDefinitions** 部分中包含有用以计算度量标准系列的基准定义。对于基准属性被设为 **True** 的 **AttributeGroup** 部分中的各个度量标准，必须为其指定基准定义。

您可以定义两种类型的基准：每小时（要求）和每日（可选）。每小时基准适用于事件处理以及在报告中显示基准。每日基准适用于在具有 1 个月或更长时间范围的报告中显示基准。

下列信息描述了使用的基准属性元素：

### 名称

指定度量标准的基准定义类型。其类型是每小时或每日基准。

**可否更新：** 否

**可能的值：** `HourlyBaseline`、`DailyBaseline`

### ID

指定不再使用的值。但必须将此字段指定为正整数，而且在此度量标准系列的所有每小时和每日基准定义中必须是唯一的。

**可否更新：** 是

**可能的值：** 任何唯一的正整数。

**更新效果：** 无

### PerformanceMetric

指定需要计算基准的度量标准名称（区分大小写）。将度量标准属性的“轮询”和“基准”属性设置为 **True**。

**可否更新：** 是

**可能的值：** 有效的度量标准名称（区分大小写）。

**更新效果：** 为度量标准执行基准计算。

**更新何时生效：** 下次基准计算时（每小时或每日）。

**更新生效所需的操作：** 无

### 时间

指定基准计算的类型，即每小时或每日基准。为 HourlyBaseline 名称指定值“1 小时”，为 DailyBaseline 名称指定值“1 天”。

**可否更新：**是

**可能的值：**1 小时、1 天

**更新效果：**将会每小时或每日执行基准计算。

**更新何时生效：**下次基准计算时（每小时或每日）

**更新生效所需的操作：**无

### 窗口

指定不再使用的值。但必须将此字段的每小时基准指定为“30 天”，每日基准指定为“90 天”。

**可否更新：**否

**可能的值：**30 天、90 天

### StartDate、EndDate、DaysOfWeek

指定更多不使用的值，但必须将其指定为 0（零）。

**可否更新：**否

**可能的值：**0

## 表达式

“表达式”部分由用于组件发现的 ExpressionGroup 标记构成。在组件发现过程中，将会计算组件项属性的值（比如 IndexList、名称和说明）。支持度量标准系列表达式的供应商认证表达式用于此计算。

**注意：**请勿混淆度量标准系列和供应商认证 ExpressionGroup 标记。

以下 DestCert URI 的 ExpressionGroup 标记必须存在：

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

**ExpressionGroup/name**

(可选) 指定表达式组名称。

**可否更新:** 是

**可能的值:** 纯文本

**更新效果:** 无

**ExpressionGroup/destCert**

指定包含 destAttrs 的组件方面以进行填充。除项和 DeviceComponent 方面之外，方面名称通常来自于 ComponentFacets 部分。

**可否更新:** 是

**可能的值:** 在 ComponentFacets 中定义的方面，或项和 DeviceComponent 方面。

**更新效果:** 允许表达式 destAttr 变更

**更新何时生效:** 组件重新发现时

**更新生效所需的操作:** 无

**ExpressionGroup/Expression**

为组件方面属性指定表达式。

**可否更新:** 是

**可能的值:** 任何有效的度量标准

**更新效果:** 允许表达式 destAttr 变更

**更新何时生效:** 组件重新发现时

**更新生效所需的操作:** 无

**ExpressionGroup/Expression/destAttr**

指定组件方面属性名称。

**可否更新:** 是

**可能的值:** 来自该组件方面的任何有效属性。

**更新的效果:** 更改属性名称

**更新何时生效:** 组件重新发现时

**更新生效所需的操作:** 无

## 层次结构

可定义不同度量标准系列项之间的层次结构或父子关系，例如，界面和 CBQoS Classmap。在度量标准系列定义中，必须在子度量标准系列中指定层次结构：

- ComponentFacets 中的层次结构 QName
- 层次结构 ExpressionGroup 中的 ItemUniqueId 和 ParentUniqueId destAttr 值
- AttributeGroup 中的 ItemUniqueIDs 和 ParentUniqueIDs 属性

支持表达式是在[供应商认证中定义的](#) (p. 72)。

下列 DestCert URI 的层次结构 ExpressionGroup 标记必须存在：

DestCert	DestAttr
{http://im.ca.com/inventory}Hierarchy	ItemUniqueId
{http://im.ca.com/inventory}Hierarchy	ParentUniqueId

[示例：](#)

```
<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>
```

[详细信息：](#)

[ComponentFacets](#) (p. 32)

## ComponentReconciliation

下列信息定义了用于组件发现的组件调节逻辑。首先，此信息将确定系统是否已发现此组件。其次，调节逻辑将确定是要更新现有组件或是要创建新组件。

```
<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>
```

每个度量标准系列可能有多个命令算法。如果度量标准系列未定义调节算法，则会应用具有匹配属性 `Item.Name` 的默认调节算法。

**详细信息：**

[ComponentFacets](#) (p. 32)  
[MatchAlgorithms](#) (p. 46)

## ItemReconciliation

**重要说明！** `ItemReconciliation` 是可能会发生更改以支持未来复杂度量标准系列结构的新部分。禁止使用。

下列信息定义了用于项发现的项调节逻辑。此逻辑将确定系统是否已发现项。基于此确定，将会更新现有项或创建新项。项调节类似于组件调节。但项调节可用于项而非组件，比如虚拟主机。`ItemFacets` 将被添加到任何新项或任何匹配项（如果方面不存在）。

**示例：**

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

### **SourceAgentScopedReconciliation**

定义用于调节项的匹配算法。

**可否更新:** 是

**更新效果:** 项调节逻辑变更。

**更新何时生效:** 重新发现时

**更新生效所需的操作:** 更新度量标准系列或更改供应商认证优先顺序。

### **GlobalScopedReconciliation**

定义在无法调节源代理项时使用的匹配算法。

**GlobalScopedReconciliation** 算法用于查找为其他代理创建但匹配潜在新项的项。如果将 `matchDevices` 属性设置为 `True`，则会使用系统默认（内置、在 XML 中不可见）`ComponentReconciliationInfo` 匹配算法。匹配算法基于设备主 IP 地址和主机名。

**可否更新:** 是

**更新效果:** 项调节逻辑变更。

**更新何时生效:** 重新发现时

**更新生效所需的操作:** 更新度量标准系列或更改供应商认证优先顺序。

**详细信息:**

[ItemFacets \(p. 33\)](#)

[MatchAlgorithms \(p. 46\)](#)

## **MatchAlgorithms**

组件调节和项调节将采用匹配算法。两个匹配算法都将受到支持：

- **ExactMatch** - 必须匹配所有指定的属性以描述匹配新数据的项。
- **BestOfMatch** - 用户必须指定通过使用“`leastMatchCount`”值来匹配的最小属性数量。此外，每个属性都带有“必要的”主要属性。如果将必要的属性设置为 `True`，则要将必须匹配的属性视为匹配。

如果对度量标准系列提供多个算法，则算法将拥有匹配优先级。算法的顺序确定了优先顺序。处于顶端的算法拥有最高的优先级。底端的算法拥有最低的优先级。

每个算法必须拥有至少 1 个匹配属性。如果数据通过同样的算法匹配多个项，那么拥有最佳匹配的项将会胜出。如果多个匹配项拥有相同数量的匹配属性，将从这些项中随机挑选胜出者。

### 示例：调节的工作方式

对度量标准系统提供有两个匹配算法：alg1 和 alg2。Alg1 拥有比 alg2 更高的优先级。度量标准系列拥有三个现有组件项：1、2 和 3。重新发现度量标准系列将会查找三个条目：A、B 和 C。现在，我们将采用两种算法来确定哪个条目是新条目、已更改条目或未更改条目。

```
调节元数据 新数据 现有组件
<ComponentReconciliation>          A      1
  <MatchAlgorithms>                 B      2
    <MatchAlgorithm1>                C      3
      <MatchAttribute name="attr1"/>
      <MatchAttribute name="attr2"/>
    </MatchAlgorithm1>

    <MatchAlgorithm2>
      <MatchAttribute name="attr1"/>
      <MatchAttribute name="attr3"/>
      <MatchAttribute name="attr4"/>
    </MatchAlgorithm2>

  </MatchAlgorithms>
</ComponentReconciliation>
```

<MatchAlgorithm1> 和 <MatchAlgorithm2> 可能是 <ExactMatch> 或 <BestOfMatch>。两种匹配算法的顺序说明 MatchAlgorithm1 拥有比 MatchAlgorithm2 更高的优先级。

#### 情况 1：唯一的 1 对 1 匹配

条目 A 匹配项 1，项 1 没有任何其他匹配。

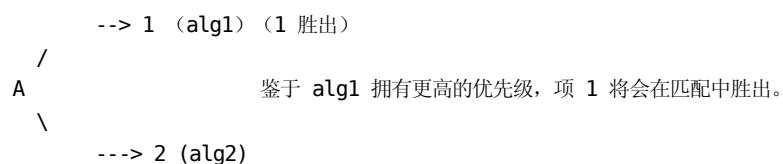
A -----> 1

此示例是最简单的案例。此匹配是唯一的，因此即使其匹配 alg1 或 alg2 也无关紧要。条目 A 匹配项 1。

良好的匹配算法可产生更多的唯一匹配。

#### 情况 2：一个条目拥有多种匹配

通过 alg1，条目 A 将匹配项 1，通过 alg2，还将匹配项 2。



### 情况 3：通过不同的算法将多个条目与同样的项进行匹配

通过 alg1，条目 A 将匹配 1，通过 alg2 项目，B 还将匹配项 1。

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

鉴于 alg1 拥有更高的优先级，条目 A 将会胜出。

### 情况 4：通过同样的算法但不同的匹配属性数量将多个条目与同样的项进行匹配

通过 alg1，A 和 B 都将匹配 1。

A -----> 1 (alg1, # 匹配属性数量: 2) (A 胜出)  
B -----> 1 (alg1, # 匹配属性数量: 1)

鉴于 A 拥有更多的匹配属性，A 将胜出。

如果匹配属性的数量相同，则会随机挑选胜出者，还将生成警告。

### 情况 5：混合匹配 1

```
alg1  
A -----> 1  
/ alg2 (匹配属性计数: 3)  
B  
/ alg2 (匹配属性计数: 2)  
-----> 2
```

A 匹配 1，因为其是通过更高优先级算法进行匹配的。

B 匹配 2，因为 1 已匹配 A。

### 情况 6：混合匹配 2

```
-----> 3  
/ alg1  
A ==> A 对 3 胜出，因为 alg1 拥有更高的匹配优先级  
\ alg2  
-----> 1  
/ alg2  
B ==> B 对 2 胜出，因为 alg1 拥有更高的匹配优先级  
\ alg1  
-----> 2  
/ alg2  
C ==> C 没有匹配，因为 2 已匹配 B, 3 已匹配 A  
\ alg2  
-----> 3
```

条目 C 将被视为新组件。1 将被视为未匹配项。

情况 1（唯一匹配）匹配的越多，匹配算法将越有效。

可否更新：是

更新效果：组件调节逻辑变更。

**更新何时生效:** 重新发现时

**更新生效所需的操作:** 更新度量标准系列或更改供应商认证优先顺序。

**详细信息:**

[ComponentReconciliation](#) (p. 45)

[ItemReconciliation](#) (p. 45)

## ReconfigDetectionAttr

此元素定义用于更改检测的度量标准系列。您可以在监视配置文件上实现检测变更。Data Aggregator 可提取仅用于检查目标设备是否变更而非执行全面重新发现的属性。此特性有助于提升性能和减少网络流量。

**可否更新:** 是

**可能的值:** 度量标准系列属性的全名。指定的度量标准系列属性标志要将缓存、持续和外部标志设置为 True。

**更新效果:** 组件配置检测变更。

**更新何时生效:** 重新发现后。

**更新生效所需的操作:** 更新度量标准系列或更改供应商认证优先顺序。

## 自定义度量标准系列不支持的属性

自定义度量标准系列不支持下列属性：

- 方差
- RollupExpression

**重要说明！** 如果您将导出的工厂度量标准系列用作模板，请移除所有不受支持的元素或在自定义度量标准系列中将其设置为 True。

## 导入自定义度量标准系列

要使您的自定义度量标准系列生效，请将其导入到 Data Aggregator。

CA Performance Center 尚未提供用户界面，以支持直接基于 IMDBCertificationFacet.xsd 架构的度量标准系列导入。因此，要充分利用 Data Aggregator 暴露的 Web 服务来完成此任务。

使用任何 REST 客户端。我们建议使用带有 CA 性能管理的 RESTClient。如未安装，可从 <http://code.google.com/p/rest-client/> 获取 RESTClient GUI JAR 文件。

在使用 RESTClient 时，请注意下列信息：

- 如要启动，请双击 JAR 文件。
- 在使用 POST 发送 XML 时，务必将字符集设置为 UTF-8。  
要查看和验证此设置，请单击“编辑内容 - 类型和字符集”按钮。
- 您还可以自动缩进响应正文，如下所示：
  1. 单击“工具”菜单上的“选项”。
  2. 选择“Etc.”选项卡。
  3. 选择“自动缩进响应正文”，单击“确定”。

**遵循这些步骤：**

1. 将 `http://da-hostname:8581/typecatalog/metricfamilies` 作为 URL 输入。
2. 选择“POST”作为方法。
3. 在“正文”设置中选择“`application/xml`”作为“正文内容类型”。
- 重要说明！无法设置内容类型导致产生 404 错误。**
4. 将您的自定义度量标准系列 XML 复制粘贴到“正文”选项卡中。
5. 运行该方法。

将导入您的自定义度量标准系列。如果无错误发生，“HTTP 响应”部分中的“状态”字段将会显示：

`HTTP/1.1 200 OK`

**注意：**如有任何其他返回代码，表示在更新您的自定义度量标准系列时发生错误。更正错误，通过再次执行 POST 来重试更新度量标准系列。

**重要说明！**为避免可能产生的数据丢失，请一定要在每次创建或更新供应商认证、度量标准系列或组件时都备份您的 `deploy` 目录。

## 验证自定义度量标准系列结果

在您导入自定义度量标准系列 XML 之后，请验证结果。在我们的示例中，我们将验证是否已成功导入框架转接度量标准系列。

遵循这些步骤：

1. 在 REST 客户端中输入以下 URL：

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

***name***

是您的自定义度量标准系列名称。在此示例中，名称是 FrameRelay。

一个页面将会显示自定义框架转接度量标准系列 XML。

您还可以验证自定义度量标准系列是否列在 Data Aggregator 数据源的监视配置菜单中“度量标准系列”选项卡上。

## 其他度量标准支持

可能会出现这样的情况，您希望将其他度量标准添加到现有自定义度量标准系列。继续我们的示例，您希望添加“Bits In”和“Bits Out”度量标准。

下列是更新的 frPVCInfo 自定义度量标准系列外观示例。我们在此程序中进行的更改以粗体显示：

```
<?xml version="1.0" encoding="UTF-8" ?>
<!--由类型目录本地管理器自动生成。-->
<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.NormalizedFacet
    DescriptorImpl">
    <Documentation>框架转接永久虚拟电路</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>框架转接电路名称</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>有关框架转接电路的描述</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>虚拟电路创建之后向后拥塞</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>总和</RollupStrategy>
    <AttributeDisplayName />
    <Percentile>0</Percentile>
</Attribute>
<Attribute name="FECNIn" type="Double">
    <Documentation>虚拟电路创建之后向前拥塞</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>总和</RollupStrategy>
    <AttributeDisplayName />
    <Percentile>0</Percentile>
</Attribute>
<Attribute name="FramesIn" type="Double">
    <Documentation>虚拟电路创建之后接收的框架
</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>总和</RollupStrategy>
    <AttributeDisplayName />
    <Percentile>0</Percentile>
</Attribute>
<Attribute name="FramesOut" type="Double">
    <Documentation>虚拟电路创建之后发送的框架
</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>总和</RollupStrategy>
<AttributeDisplayName />
<Percentile>0</Percentile>
</Attribute>
<Attribute name="BytesIn" type="Double">
    <Documentation>虚拟电路创建之后接收的字节
</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>总和</RollupStrategy>
    <AttributeDisplayName />
    <Percentile>0</Percentile>
</Attribute>
<Attribute name="BytesOut" type="Double">
    <Documentation>虚拟电路创建之后发送的字节
</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>总和</RollupStrategy>
    <AttributeDisplayName />
    <Percentile>0</Percentile>
</Attribute>
<Attribute name="BitsIn" type="Double">
    <Documentation>虚拟电路创建之后接收的位</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>总和</RollupStrategy>
    <AttributeDisplayName />
    <Percentile>0</Percentile>
</Attribute>
<Attribute name="BitsOut" type="Double">
    <Documentation>虚拟电路创建之后发送的位</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>总和</RollupStrategy>
<AttributeDisplayName />
<Percentile>0</Percentile>
</Attribute>
</AttributeGroup>
<Attribute name="SourceFacetTypes" cached="true"
list="true" persistent="true" type="QName">
    <Documentation />
    </Attribute>
    <DisplayName>框架转接 PVC </DisplayName>
    <Expressions>
        <ExpressionGroup
destCert="{http://im.ca.com/core}Item">
            <Expression destAttr="Name">名称</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>

```

我们假设您在导入自定义度量标准系列之后进行这些更改。因此，在您进行更改之后，请更新度量标准系列。在您更新度量标准系列之后，您还必须更新度量标准系列与之关联的供应商认证。

#### 详细信息：

[更新自定义供应商认证 \(p. 86\)](#)  
[更新自定义度量标准系列 \(p. 56\)](#)

## 更新自定义度量标准系列

您可以更新现有自定义度量标准系列。在此示例中，您此前对自定义度量标准系列添加度量标准 frPVCInfo。现在，要更新度量标准系列，以使更改生效。

**注意：**有关在更新自定义度量标准系列时更新标记或属性效果的信息，请查看自定义度量标准系列 XML 详细信息。尤其是，请查看具体的属性描述。

**遵循这些步骤：**

1. 在 URL 字段输入下列地址：

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

是要更新的自定义度量标准系列名称。有关我们的示例，度量标准系列的名称是 frPVCInfo。

2. 如要验证 URL 上存在的度量标准系列，请在“方法”选项卡中选择“GET”。

**注意：**如果度量标准系列不存在，您可以[导入自定义度量标准系列](#)(p. 50)。

3. 在验证 URL 上存在的度量标准系列后，请在“方法”选项卡中选择“PUT”。

4. 将更新的自定义度量标准系列 XML 复制粘贴到“正文”选项卡(“编辑”字段)，将内容类型设置为 application/xml。

**重要说明！无法设置内容类型导致产生 404 错误。**

5. 单击 URL 字段旁的“转到”按钮。

将更新您的自定义度量标准系列。如果无错误发生，“HTTP 响应”部分中的“状态”字段将会显示下列信息：

HTTP/1.1 200 OK

如有任何其他返回代码，表示在更新您的自定义度量标准系列时发生错误。更正错误，通过再次执行 PUT 来重试更新度量标准系列。

**重要说明！**为避免可能产生的数据丢失，请一定要在每次创建或更新供应商认证、度量标准系列或组件时都备份您的 deploy 目录。

6. [接下来，更新度量标准系列与之关联的自定义供应商认证。](#) (p. 86)

详细信息：

[正在了解度量标准系列 XML 结构 \(p. 26\)](#)  
[其他度量标准支持 \(p. 51\)](#)



# 第 5 章：正在创建自定义供应商认证

---

成功导入后，新的供应商认证将被自动添加到指定度量标准系列的优先列表末尾。在需要时，您可以修改优先列表，将供应商认证移动到更高的优先级。在发现指定度量标准系列的设备支持时，将会使用供应商认证。

**注意：**Data Aggregator 提供了用于创建自定义供应商认证和度量标准系列的基本方法和高级方法。基本方法比较简单，包括使用用户界面为现有支持技术（度量标准系列）添加供应商支持。此方法满足许多用户的要求。而高级方法是以工厂认证格式为基础，并且提供一套完整的功能。本指南对高级认证方法进行了说明。有关基本认证方法的信息，请参见《*Data Aggregator 自行认证指南*》。

此部分包含以下主题：

- [创建供应商认证 XML 模板 \(p. 59\)](#)
- [正在了解供应商认证 XML 结构 \(p. 60\)](#)
- [导入自定义供应商认证 \(p. 75\)](#)
- [验证自定义供应商认证结果 \(p. 77\)](#)
- [筛选支持 \(p. 78\)](#)
- [多个 MIB 表支持 \(p. 81\)](#)
- [更新自定义供应商认证 \(p. 86\)](#)

## 创建供应商认证 XML 模板

使用首选 REST 客户端来创建示例 xml 文件，您可以将其用作模板来创建自定义供应商认证。

先从检索现有供应商认证的列表开始。之后可以验证您所需的供应商认证是否受到支持。

遵循这些步骤：

1. 将 REST 客户端设置为与 Data Aggregator 服务器连接。
2. 对于 REST 客户端中的 Data Aggregator Web 服务 API，输入下列 URL：  
`http://da_hostname:8581/typecatalog/certifications/snmp`  
将会显示所有适用供应商认证的列表。
3. 验证您所需的供应商认证是否受到支持。

如果您所需的供应商认证不受支持，请查看和导出类似于您所要创建自定义供应商认证的供应商认证。

**遵循这些步骤:**

1. 输入下列的 URL，以便检索类似于您所需度量标准系列的具体供应商认证：

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

是您的供应商认证名称，比如 CiscoCPUMibRev。

2. 在“方法”选项卡中选择“GET”。

3. 运行该方法。

已返回的 XML 包含供应商认证信息。

您可以将此 XML 用作模板来创建自定义供应商认证。

4. 根据需要，将供应商认证 XML 复制到文本文件，再对其进行修改。有关 XML 结构的示例，请参见“[了解供应商认证 XML \(p. 60\)](#)”。

## 正在了解供应商认证 XML 结构

供应商认证会将供应商和设备特定数据映射到度量标准系列中定义的性能度量标准和配置数据中。将此数据从各个源映射到“规范化”度量标准系列值，以便 Data Aggregator 在不考虑设备供应商的情况下统一报告数据。

下列是支持我们框架转接 PVC 示例的自定义供应商认证 XML 示例。请注意我们的示例度量标准系列 frPVCInfo 如何包含在 ExpressionGroup 部分当中（粗体显示，以供说明）：

```
<?xml version="1.0" encoding="UTF-8" ?>
<!--由类型目录本地管理器自动生成。-->
<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.CertificationFa
  cetDescriptorImpl">
    <Documentation>框架转接 PVC 供应商认证</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>框架转接 PVC 认证</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>

```

## 基本属性

自定义供应商认证的基本属性可帮助您将其与您创建的其他自定义供应商认证区分开来。此外，这些属性还指示您从哪个供应商 MIB 收集度量标准数据。

在您确定基本属性时，请注意下列限制条件：

- 各个供应商认证的 `FacetType/name` 和 `FacetType/DisplayName` 属性必须是唯一的。
- 协议标记始终是 `SNMP`。
- 设置 `FacetType/descriptorClass` 属性及所有 `DataModel` 和 `FacetOf` 属性，如上图中的示例 XML 所示。

**FacetType/name**

唯一标识供应商认证。

**建议:** 符合“<MibName><TableName>Mib”。

**可否更新:** 否

**可能的值:** 字母数字和下划线。不允许使用点和虚线。

FacetType 部分显示了特别的供应商认证。如果供应商认证暴露了供应商特定设备（比如来自 MIB-2 实施的 TCP 和 UDP 统计）的各个方面，那么同样的 XML 文档可包含多个 FacetType 部分。

FacetType 部分中包含某些基本属性。例如，本部分包含供应商 MIB 的名称，接着是一个或多个 AttributeGroup 部分。这些 AttributeGroup 部分定义了认证从 MIB 中使用了哪个属性。最终，一个或多个 ExpressionGroup 部分会将来自 AttributeGroup 部分的属性映射到度量标准系列中指定的度量标准。

**FacetType/Documentation**

描述通过供应商认证认证了哪些内容。

**建议:** 包括有关供应商 MIB 名称和表名称的详细信息。

**可否更新:** 是

**可能的值:** 纯文本

**更新效果:** 无

**FacetType/MIB**

指定了 DEFINITIONS 条款在 ASN. 文件中定义的 MIB 名称。

**建议:** 符合“<MibName>”

**可否更新:** 是

**可能的值:** 纯文本

**更新效果:** 更改到管理员用户界面“供应商认证”选项卡中的“SNMP MIB 名称”列。

**更新何时生效:** 立即生效

**更新生效所需的操作:** 刷新用户界面。

#### **FacetType/DisplayName**

指定供应商认证的名称，如 CA Performance Center 中所示。

**建议：**以供应商名称开始，包含 MIB 和功能信息。

**可否更新：**是

**可能的值：**纯文本

**更新效果：**管理员用户界面中的名称变更。

**更新何时生效：**立即生效

**更新生效所需的操作：**刷新用户界面。

### **AttributeGroup（供应商认证）**

下列示例说明了您的自定义供应商认证 AttributeGroup 部分。本部分标识了用于映射原始设备数据的供应商 MIB 中特定表的属性（变量 OID）。此数据将被映射到性能度量标准和度量标准系列中定义的配置数据。

您要将 AttributeGroup/list 和 AttributeGroup/external 属性设置为 True，如上图中的示例 XML 所示。这些属性指定的各个属性代表从外部源获取的值列表（MIB 表）。下列信息概述了用于自定义的 XML 元素。

#### **AttributeGroup/name**

指定属性组名称。

**建议：**符合“<FacetType/name>组”。

**可否更新：**是

**可能的值：**纯文本

**更新效果：**无

#### **文档**

（可选）指定属性组的说明。

**可否更新：**是

**可能的值：**纯文本

**更新效果：**无

**UseIndex**

指定属性的名称，以便用作连接多个 MIB 表的属性组索引。

**建议：**设为 AttributeGroup/name 属性的值。

**可否更新：**是

**可能的值：**纯文本

**更新效果：**无

**详细信息：**

[AttributeGroup（多个 MIB 表）\(p. 84\)](#)

[IndexTagList \(p. 73\)](#)

## 常规属性（供应商认证）

所有供应商认证的常规属性如下所示：

**Attribute/name**

指定属性名称。

**建议：**如果设置为 MIB 变量名称，ASN.1 文件中将会定义 OBJECT-TYPE 条款。

**可否更新：**是

**可能的值：**字母数字和下划线。不允许使用点和虚线。

**更新效果：**您必须更新任何引用此属性的表达式。

**更新何时生效：**立即生效

**更新生效所需的操作：**无

**Attribute/type**

指定此属性的数据类型。

**建议：**使用最匹配 SYNTAX 条款在 ASN.1 文件中所定义变量类型的属性类型。

**可否更新：**是

**可能的值：**Boolean、Int、Long、Double、BigInteger、String、DateTime、IPAddress、MACaddress、IPSubnet、OctetString、ObjectID

**更新效果：**轮询 SNMP 数据将会转化为此类型。

**更新何时生效：**下次轮询时

**更新生效所需的操作：**无

## 文档

(可选) 指定属性的描述, 用以记录 MIB 变量的语义 (比如单位)。

**建议:** 使用取自 MIB ASN. 文件的描述。

**可否更新:** 是

**可能的值:** 纯文本

**更新效果:** 无

## IsKey

(可选) 使用标志来指示 MIB 变量是否是确定对表支持的秘钥。如果将多个字段指定为秘钥, 可将所有字段视为复合秘钥。

**默认:** False

**建议:** 如果是组件发现的关键 MIB 对象, 则将其设置为 True。

**可否更新:** 是

**可能的值:** True、False

**更新效果:** 组件可能变更为新的供应商认证。

**更新何时生效:** 组件重新发现后。

**更新生效所需的操作:** 更新度量标准系列或更改供应商认证优先顺序。

## IsIndex

(可选) 使用标志来指示此变量是否是 MIB 表的索引。

**默认:** False

**建议:** 将索引属性设置为 True。

**可否更新:** 是

**可能的值:** True、False

**更新效果:** 组件索引可能变更。

**更新何时生效:** 组件重新发现后。

**更新生效所需的操作:** 更新度量标准系列或更改供应商认证优先顺序。

**NeedsDelta**

(可选) 使用标志来指示是否存储 MIB 变量（也即是存储计数器当前和上次轮询之间的差异）。

**默认:** False

**建议:** 如果变量被定义为 Counter、Counter32、Counter64 或 MIB 中的 TimeTicks 数量，则要将其设置为 True。

**可否更新:** 是

**可能的值:** True、False

**更新效果:** 轮询数据变更。

**更新何时生效:** 下次轮询时

**更新生效所需的操作:** 无

**源**

指定属性的 ObjectId。

**建议:** 设置为 OBJECT-TYPE 定义的完全限定 MIB 变量 OID。

**可否更新:** 是

**可能的值:** 以点分隔的数字（例如，1.3.6.1.4.1...）

**更新效果:** 从指定的 OID 轮询数据。

**更新何时生效:** 下次轮询时

**更新生效所需的操作:** 无

**版本**

指定供应商认证的版本。如果在升级期间安装供应商认证，安装程序则会检查认证的版本。如果其版本高于安装程序包的版本，安装程序则不会更改现有的供应商认证。

**默认值:** 1.0

**可否更新:** 是

**可能的值:** 以点分隔的数字（例如，1.3.6.1.4.1...）

**更新的效果:** 更新版本属性。

**更新何时生效:** 立即生效

**更新生效所需的操作:** 无

## 作者

指定供应商认证的创建者。

**默认值:** “自定义”

**可否更新:** 是

**可能的值:** 任意字母数字字符串。

**更新的效果:** 更新作者属性。

**更新何时生效:** 立即生效

**更新生效所需的操作:** 无

可指定通过供应商认证支持而收集度量标准系列数据集的属性列表。通常，这种数据将分为两类：

- 仅在发现时收集的设备组件配置数据（比如名称或索引）。
- 性能数据是在每个轮训周期内收集的。

**详细信息:**

[多个 MIB 表支持 \(p. 81\)](#)

[筛选支持 \(p. 78\)](#)

## 配置数据属性

具有名称 INDEX 和类型 ObjectId 的属性将被映射到目标度量标准系列的索引属性。您可以将源标记的值设置为任何变量 OID。但您通常要使用 INDEX 条款表中所列的其中某个变量。例如，请注意 MIB-2 界面表中的 ifIndex。此变量可作为相同 MIB 表中其他变量的索引。此外，还要将此属性的 IsIndex 标记（通常还有 IsKey 标记）设置为 True。

在此示例中，诸如 ifDesc 或 ifType 等属性提供了有关界面的更多配置信息。因此，这些属性可用于目标度量标准系列的“名称”和“描述”属性。

## 性能数据属性

这些属性提供了目标度量标准系列中的性能度量标准原始数据。请注意以下几点：

- 您可以直接将其中的某个属性映射到度量标准性能度量标准，或
- 您可以将表达式中的属性与其他属性同时使用来计算度量标准的值。

## ExpressionGroup

ExpressionGroup 映射属性如下：

- 从 AttributeGroup (定义如何从 SNMP MIB 获取度量标准)
- 到度量标准系列中的度量标准 (定义如何在数据库中如何存储属性)

您可以在从设备接收值时或在执行某些规范化操作后，将 MIB 值存储在数据库中。例如，规格化操作包括乘以或除以 1024，以进行千字节的转换。

### ExpressionGroup/name

(可选) 指定表达式组名称。

**可否更新:** 是

**可能的值:** 纯文本

**更新效果:** 无

### ExpressionGroup/destCert

指定包含 destAttrs 的度量标准系列以进行填充。

**可否更新:** 是

**可能的值:** 任何有效的度量标准

**更新效果:** 允许表达式 destAttr 变更

**更新何时生效:** 立即生效

**更新生效所需的操作:** 无

### ExpressionGroup/Filter

(可选) 指定要发现哪些组件。使用筛选来减少待管理的组件数量。

**可否更新:** 是

**可能的值:** 采用适用属性的 Boolean MVEL 表达式

**更新的效果:** 更改要发现哪些组件。

**更新何时生效:** 组件重新发现后。

**更新生效所需的操作:** 更新度量标准系列或更改供应商认证优先顺序。

## 在 ExpressionGroup 中筛选

以下列表描述了筛选示例。

### 示例 1:

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

在此示例中，仅分割在数据存储库中创建的大小  $> 0$ 、`storageType == hrStorageFixedDisk` 的实例。

### 示例 2:

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

### 示例 3:

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

### 示例 4:

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

本示例采用供应商认证实用函数来解析 OID。

**注意：**有关完整的函数列表，请参见本指南中的附录。

## Expression/destAttr 度量标准

下列信息描述了 Expression/destAttr 度量标准：

### 索引

指定以使用 `ObjectID` 的供应商认证属性来定义 MVEL 表达式，从而对索引度量标准系列属性提供值。

**建议：** 设置为 INDEX。

**可否更新：** 是

**可能的值：** 带有 `<IsIndex>true</IsIndex>` 的任何属性。

**更新效果：** 组件索引可能变更。

**更新何时生效：** 组件重新发现后

**更新生效所需的操作：** 更新度量标准系列或更改供应商认证优先顺序。

### 名称

指定以使用供应商认证属性来收集配置数据。此配置数据可帮助定义 MVEL 表达式，以便对“名称”度量标准系列属性提供值。

**建议：** 包含有尽可能多的信息来唯一标识实例。

**可否更新：** 是

**可能的值：** 采用适用属性的 String MVEL 表达式

**更新效果：** 组件名称变更

**更新何时生效：** 组件重新发现后

**更新生效所需的操作：** 更新度量标准系列或更改供应商认证优先顺序。

### 说明

(可选) 指定以使用供应商认证属性来收集配置数据。此配置数据可帮助定义 MVEL 表达式，以便对“描述”度量标准系列属性提供值。并非所有度量标准系列都支持“描述”属性。

**建议：** 包括尽可能多的信息以便描述实例。

**可否更新：** 是

**可能的值：** 采用适用属性的 String MVEL 表达式

**更新效果：** 组件描述变更

**更新何时生效：** 组件重新发现时

**更新生效所需的操作：** 更新度量标准系列或更改供应商认证优先顺序。

### 其他度量标准

指定以使用供应商认证属性来收集配置或性能数据。此数据可帮助定义 MVEL 表达式，以便对度量标准系列属性提供值。

**可否添加:** 是，前提是度量标准系列中存在 destAttr。

**可否更新:** 是

**可能的值:** 采用适用属性的 MVEL 表达式，生成匹配目标属性类型的值。

**更新效果:** 轮询值变更

**更新何时生效:** 下次轮询时

**更新生效所需的操作:** 无

度量标准系列暴露 URI（比如，  
`{http://im.ca.com/normalizer}FamilyName.AttributeName`），可在 ExpressionGroup 中单独引用。ExpressionGroup/destCert 属性将被设置为 URI（例如，`{http://im.ca.com/normalizer}FamilyName`），Expression/destAttr 属性将被设置为 *AttributeName*。

**详细信息:**

[配置数据属性 \(p. 68\)](#)

## HierarchyList

下列信息定义了层次结构行为。

### Hierarchy/ParentFacet

指定用于查找候选父项的方面 QName。

**可否更新:** 是

**可能的值:** 任何有效的方面

**更新效果:** 更改层次结构构造。

**更新何时生效:** 组件重新发现后

**更新生效所需的操作:** 更新度量标准系列或更改供应商认证优先顺序。

**Hierarchy/ParentAttribute**

指定用于识别具体父项的属性 QName。

**可否更新:** 是

**可能的值:** 任何有效的属性 QName

**更新效果:** 更改层次结构构造。

**更新何时生效:** 组件重新发现后

**更新生效所需的操作:** 更新度量标准系列或更改供应商认证优先顺序。

**Hierarchy/ChildAttribute**

指定用于匹配父项上 ParentAttribute 的子项上的属性 QName。

**可否更新:** 是

**可能的值:** 任何有效的属性 QName

**更新效果:** 更改层次结构构造。

**更新何时生效:** 组件重新发现后

**更新生效所需的操作:** 更新度量标准系列或更改供应商认证优先顺序。

**IndexTagList**

如要从多个 MIB 表轮询属性，我们需要每个 MIB 表属性组中都包含这些属性。索引标记列表提供了将两个属性组（或 MIB 表）与不同索引关联起来的机制。将各组关联起来之后，一个表的项（行）将会与第二个表的对应行联系起来。

**PrimaryTag**

引用主要属性组（即组将采用 ObjectId 类型来定义索引属性）。此元素的值必须等于主要组的属性组 'UseIndex' 标记。

**可否更新:** 是

**可能的值:** 属性组的 'UseIndex' 标记对应于主要属性组。

**更新效果:** 索引变更

**更新何时生效:** 组件重新发现后

**更新生效所需的操作:** 更新度量标准系列或更改供应商认证优先顺序。

**IndexTag**

定义如何将主要组（或 MIB 表）的行与次要组中的行关联起来。此元素通过指定必须匹配的两组属性来关联行。

### **IndexTag/Name**

引用次要组（或 MIB 表）。此元素的值必须等于您试图与主要属性组进行关联的次要属性组 'UseIndex' 标记。

**可否更新：** 是

**可能的值：** 次要属性组的 'UseIndex' 标记。

**更新效果：** 索引变更

**更新何时生效：** 组件重新发现后

**更新生效所需的操作：** 更新度量标准系列或更改供应商认证优先顺序。

### **IndexTag/PrimaryKeyExpression**

指定包含主要属性组属性或对应于任何此前定义 IndexTag 属性组的 MVEL 表达式。计算的值匹配 'ThisTagKeyExpression'。如果匹配，则两个属性组（或 MIB 表）的行将会关联起来。之后可在支持 destAttr（或度量标准）的 'Expression' 中共同使用。

**可否更新：** 是

**可能的值：** 有效的 MVEL 表达式

**更新效果：** 索引变更

**更新何时生效：** 组件重新发现后

**更新生效所需的操作：** 更新度量标准系列或更改供应商认证优先顺序。

### **IndexTag/ThisTagKeyExpression**

指定包含次要属性组属性的 MVEL 表达式。计算的值匹配 'PrimaryKeyExpression'。如果匹配，则两个组（或 MIB 表）的行将会关联起来。之后可在支持 destAttr（或度量标准）的 'Expression' 中共同使用。

**可否更新：** 是

**可能的值：** 有效的 MVEL 表达式

**更新效果：** 索引变更

**更新何时生效：** 组件重新发现后

**更新生效所需的操作：** 更新度量标准系列或更改供应商认证优先顺序。

## 导入自定义供应商认证

要使您的供应商认证生效，请将其导入到 Data Aggregator。有两种导入供应商认证的方式：

- [使用 REST 客户端导入自定义供应商认证 \(p. 75\)](#)。
- [使用自定义供应商认证安装程序 \(p. 76\)](#)

### 使用 REST 客户端导入自定义供应商认证

使用任意 REST 客户端导入供应商认证。我们建议采用 RESTClient 来执行 GET、POST、PUT，采用其他方法来执行 CA 性能管理中的 RESTful Web 服务。如未安装，可从 <http://code.google.com/p/rest-client/> 获取 RESTClient GUI JAR 文件。

在使用 RESTClient 时，请注意下列信息：

- 如要启动，请双击 JAR 文件。
- 在使用 POST 发送 XML 时，务必将字符集设置为 UTF-8。  
要查看和验证此设置，请单击“编辑内容 - 类型和字符集”按钮。
- 您还可以自动缩进响应正文，如下所示：
  1. 单击“工具”菜单上的“选项”。
  2. 选择“Etc.”选项卡。
  3. 选择“自动缩进响应正文”，单击“确定”。

#### 遵循这些步骤：

1. 将 `http://da-hostname:8581/typecatalog/certifications/snmp` 作为 URL 输入。
2. 在“方法”选项卡上选择“POST”。
3. 在“正文”设置中选择“application/xml”作为“正文内容类型”。  
**重要说明！** 无法设置内容类型导致产生 404 错误。
4. 将您的自定义供应商认证 XML 复制粘贴到“正文”选项卡中。
5. 运行该方法。

即已导入您的供应商认证。如果无错误发生，“HTTP 响应”部分中的“状态”字段将会显示：

HTTP/1.1 200 OK

**注意:** 如有任何其他返回代码, 表示在更新您的自定义组件时发生错误。更正错误, 通过再次执行 POST 来重试更新组件。

**重要说明!** 为避免可能产生的数据丢失, 请一定要在每次创建或更新供应商认证、度量标准系列或组件时都备份您的 `deploy` 目录。

## 使用自定义供应商认证安装程序

使用 Data Aggregator 上的 CA 性能管理 认证安装程序来导入认证。通过安装程序可以以三种方式安装认证:

- 单个认证
- 采用 ZIP 文件形式的认证集合
- 所有认证位于一个目录 (必须直接指向包含认证的文件夹)

**遵循这些步骤:**

1. 下载并运行安装程序。

以下 URL 提供此安装程序:

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

**<DA>**

您的 Data Aggregator 的主机名。

**注意:** 默认 REST 端口为 8581。

2. 选择安装程序的语言。
3. 出现相应的提示时输入以下信息:

- DA 主机名
- DA REST 端口

4. 选择您是想安装单个认证、认证集合 (ZIP), 还是所有认证位于一个目录。

从目录安装时, 选择您的认证所在的文件夹。认证安装程序不会在任何子目录中搜索认证。

5. 安装程序在安装过程期间检查每个认证的版本。

如果 Data Aggregator 表示已安装某认证, 认证安装程序会尝试升级该认证。如果安装程序无法升级该认证, 会出现一条错误来指定失败的原因。

6. 当安装程序已经完成认证的安装时，会显示所有遇到的错误。

您可以在安装程序在计算机主目录中创建的日志文件中查看错误详细信息。

**注意：**您还可以使用 Linux 命令行或 UI 安装程序来安装认证。

## 验证自定义供应商认证结果

在您导入自定义供应商认证 XML 之后，请验证结果。在我们的示例中，我们将验证是否已成功导入 `rPVCInfoCustom` 自定义供应商认证。

**遵循这些步骤：**

1. 在 REST 客户端中输入以下 URL:

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

是您的自定义供应商认证名称。在此示例中，名称是 `frPVCInfoCustom`。

一个页面将显示自定义 `frPVCInfoCustom` 供应商认证 XML。

2. 在 REST 客户端中输入以下 URL:

```
http://da_hostname:8581/rest/vendorpriorities
```

3. 搜索显示在 XML 文档中的供应商认证。

4. 验证 `MetricFamilyVendorPriority` XML 是否列出您的自定义供应商认证和度量标准系列。

您现在可以使用 Data Aggregator 用户界面来验证您的自定义供应商认证。

**遵循这些步骤：**

1. 验证您的自定义供应商认证在 Data Aggregator 中是否可用：

- a. 在 Data Aggregator 数据源的“监视配置文件”菜单中单击“供应商认证”。

验证您的自定义供应商认证是否已列出（或搜索您的供应商认证）。

- b. 从供应商认证的列表中选择您的自定义供应商认证。

验证您的自定义度量标准系列是否显示在“度量标准系列”视图中。

- c. 单击“度量标准系列”视图中的度量标准系列。
- d. 单击“供应商认证优先顺序”选项卡。

验证您的自定义供应商认证是否与所有可管理设备的设备集合相关联。
2. 配置您的自定义供应商认证所支持度量标准系列的监视配置文件和设备集合：
  - a. 验证发现是否创建预期和适当的项。
  - b. 验证这些项是否轮询和收集您在度量标准系列 XML 中指定的度量标准数据。
  - c. 验证收集的数据是否正确。
  - d. 如果已配置项同步，则要验证项数据是否与 CA Performance Center 正确同步。
  - e. 再次运行发现并验证是否已正确更新信息。

## 筛选支持

**ExpressionGroup** 部分的筛选标记用于在组件发现过程中应用筛选。

继续我们的示例，您只希望监视活动的框架转接 PVC。您可以通过将筛选标记应用到您的 `frPVCInfoCustom` 自定义供应商认证，以监视这些具体组件，来执行此操作。筛选将被添加到 **ExpressionGroup**。

**注意：**您可以在“筛选”表达式中使用 MVEL 语法。注意：有关 MVEL 语法、供应商认证实用函数和供应商认证全局变量的更多信息，请参见本指南的附录。

下列是更新的 frPVCInfoCustom 自定义供应商认证外观示例。我们在此程序中进行的更改以粗体显示。

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-由类型目录本地管理器自动生成。-->
<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.CertificationFa
    cetDescriptorImpl">
    <Documentation>框架转接 PVC 供应商认证</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>框架转接 PVC 认证</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>
```

我们假设在导入自定义供应商认证之后进行这些更改。因此，在您进行更改之后，请更新供应商认证。

详细信息：

[更新自定义供应商认证 \(p. 86\)](#)

## 多个 MIB 表支持

在认证供应商 MIB 时，存在您必须从两个或多个表中收集特定度量标准系列原始数据的情况。

CA 性能管理 可对需要访问多个 MIB 表的供应商认证提供支持。此支持是基于单表认证 XML 文档结构的增强，允许连接通过公用密钥（索引）从多个表中收集的数据。继续我们的示例，您只希望监视活动的框架转接 PVC。修改您的 frPVCInfoCustom 自定义供应商认证来添加多个 MIB 表支持。

在此示例中，可采用 ifXTable 中的 ifName MIB 对象以及对 PVC 提供数据链路连接标识符 (DLCI) 的 frCircuitDlci 对象组合来命名框架转接 PVC。这种命名约定适用于确定 PVC 在哪个框架转接界面分层。

修改您的自定义供应商认证，以添加下列信息：

- 将新的属性添加到现有 AttributeGroup，以便表示 frCircuitDlci MIB 对象。
- 您希望使用的 ifName MIB 对象来自未包含在自定义供应商认证中的 MIB。添加新的 AttributeGroup（在此情况下，ifXTable），再添加新的属性（ifName）。

下列是更新的 frPVCInfoCustom 自定义供应商认证外观示例。我们在此程序中进行的更改以粗体显示：

```
<?xml version="1.0" encoding="UTF-8" ?>
<!--由类型目录本地管理器自动生成。-->
<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.CertificationFa
    cetDescriptorImpl">
    <Documentation>框架转接 PVC 供应商认证</Documentation>
    <FacetOf namespace="http://im.ca.com/core" name="Item" />
    <AttributeGroup name="ifXTableGroup" external="true"
      list="true">
      <Documentation>将从 ifXTable 提取数据，以便引用对应于 PVC 的
      ifName</Documentation>
      <UseIndex>ifXIndexTag</UseIndex>
      <Attribute name="ifXTableIndex" type="ObjectID">
        <Documentation />
        <IsKey>false</IsKey>
        <IsIndex>true</IsIndex>
        <Source>1.3.6.1.2.1.31.1.1.1.1</Source>
        <Polled>false</Polled>
      </Attribute>
      <Attribute name="ifName" type="OctetString">
        <Documentation />
        <IsKey>false</IsKey>
        <IsIndex>false</IsIndex>
        <Source>1.3.6.1.2.1.31.1.1.1.1</Source>
        <Polled>false</Polled>
      </Attribute>
      <AttributeGroup>
        <IndexTagList>
          <PrimaryTag>PVCIndexTag</PrimaryTag>
          <IndexTag>
            <Name>ifXIndexTag</Name>
          <PrimaryKeyExpression>snmpOIDParser(INDEX,1,1)</PrimaryKeyExpres
          sion>
          <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>
        </AttributeGroup>
      </IndexTagList>
    </AttributeGroup>
  </FacetOf>
</DataModel>
```

```
</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>框架转接 PVC 认证</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(frCircuitDlc) ? ifName + " DCLI:" + frCircuitDlc : "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>
```

我们假设在导入自定义供应商认证之后进行这些更改。因此，在您进行更改之后，[请更新供应商认证 \(p. 86\)](#)。

## AttributeGroup (多个 MIB 表)

各个表必须访问其各自的 AttributeGroup 部分。可将表中的各个属性添加为 AttributeGroup 的子项。

有关下列信息，请参考这些部分：

- **AttributeGroup** 信息 - 有关用于定义主要和次要表属性的 XML 元素详细信息。
- **UseIndex** 和 **IndexTagList** 信息 - 有关连接主要和次要属性组的 XML 元素详细信息。

在此场景中，主要属性组表示您希望通过更多信息进行“扩展”的表。次要属性组包含主要属性组的“扩展”信息。

主要 **AttributeGroup** 包含有标识 MIB 表变量的属性（充当访问次要 **AttributeGroup** 的公用“秘钥”）。

次要 **AttributeGroup** 包含有所有 MIB 表变量的属性定义（执行主要表的“扩展”信息）。此外，还存在标识变量的属性（匹配主要 **AttributeGroup** 的公用“秘钥”）。

**详细信息：**

[AttributeGroup \(供应商认证\) \(p. 64\)](#)

[UseIndex \(p. 85\)](#)

[IndexTagList \(多个 MIB 表\) \(p. 86\)](#)

## UseIndex

每个 **AttributeGroup** 都有指定的 **UseIndex** 标记。**UseIndex** 标记可使您根据通用名称对 OID 进行分组。此通用名称之后将与指定的变量关联起来，充当访问各个 MIB 表的公用密钥（索引）。

下列信息概述了用于自定义的 XML 元素：

### AttributeGroup/UseIndex

唯一标识用在 **IndexTagList** 部分中的主要和次要标记名称（分别标识）。

**建议：** 设为 **AttributeGroup/name** 属性的值。

**详细信息：**

[AttributeGroup \(多个 MIB 表\) \(p. 84\)](#)

## IndexTagList（多个 MIB 表）

IndexTagList 部分提供了将两个属性组（或 MIB 表）与不同索引关联起来的机制。如果将组关联起来，一个项将从多个表中获取多个索引 ID。

IndexTagList 部分中包含所有的连接信息，其中包括每个次要属性组的 IndexTag 部分。

### IndexTagList/PrimaryTag

定义主要属性组（或 MIB 表）。设置为主要 AttributeGroup 的 UseIndex 属性值。

### IndexTag/Name

定义次要属性组。设置为次要 AttributeGroup 的 UseIndex 属性值。

### IndexTag/PrimaryKeyExpression

指定表达式以便在主要表中生成公用密钥。考虑采用 MVEL 函数从指定的主要表索引属性中获取公用密钥。

### IndexTag/ThisTagKeyExpression

指定表达式以便在次要表中生成公用密钥。考虑采用 MVEL 函数从指定的次要表索引属性中获取公用密钥。

多个表方法支持超过两个表的链接。多个表连接中存在两种类型的关系：

- **主要 → 次要 #1, 主要 → 次要 #2**  
次要表在索引标记列表中的排序无关紧要。
- **主要 → 次要 #1 → 主要 → 次要 #2**  
列表次要表 #1 位于次要表 #2 之前主要是因为表的合并方式。

主要表中的一个或多个行可以合法地映射到次要表中的相同行。次要表上的密钥是按顺序搜索的，以第一个匹配的密钥为准。

详细信息：

[多个 MIB 表支持 \(p. 81\)](#)  
[AttributeGroup（多个 MIB 表） \(p. 84\)](#)

## 更新自定义供应商认证

您可以更新现有自定义供应商认证。继续我们的示例，您之前可能对自定义度量标准系列和自定义供应商认证作出过某些更改。

对您的度量标准系列，添加度量标准。对您的供应商认证，添加 MIB 表，还包含了对自定义供应商认证的筛选。现在，要更新供应商认证，以使更改生效。

**注意：**有关更新标记或属性生效的信息，请查看自定义供应商认证 XML 详细信息。尤其是，请查看具体的属性描述。

#### 遵循这些步骤：

1. 在 URL 字段输入下列地址：

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

**CustomVendorCertName**

是要更新的自定义供应商认证名称。有关我们的示例，供应商认证的名称是 `frPVCInfoCustom`。

2. 如要验证 URL 上存在的供应商认证，请在“方法”选项卡中选择“GET”。
3. 在验证 URL 上存在的供应商认证后，请在“方法”选项卡中选择“PUT”。
4. 将更新的自定义度量标准系列 XML 复制粘贴到“正文”选项卡（“编辑”字段），将内容类型设置为 `application/xml`。

**重要说明！无法设置内容类型导致产生 404 错误。**

5. 单击 URL 字段旁的“转到”按钮。

您的自定义供应商认证已更新。如果无错误发生，“HTTP 响应”部分中的“状态”字段将会显示下列信息：

`HTTP/1.1 200 OK`

**注意：**如有任何其他返回代码，表示在更新您的自定义供应商认证时发生错误。更正错误，通过再次执行 PUT 来重试更新供应商认证。

**重要说明！为避免可能产生的数据丢失，请一定要在每次创建或更新供应商认证、度量标准系列或组件时都备份您的 deploy 目录。**

#### 详细信息：

[正在了解供应商认证 XML 结构 \(p. 60\)](#)  
[更新自定义度量标准系列 \(p. 56\)](#)



# 附录 A：供应商认证表达式：表达式运算符、函数和全局变量

---

本节介绍您可以在供应商认证表达式中使用的一些表达式运算符、函数和全局变量。

在供应商认证表达式中可以使用 MVEL 语法。MVEL 是适用于 Java 环境的公开嵌入式表达式语言，其语法与 Java 类似。MVEL 支持类似于 Java 表达式的表达式。

您可以使用运算符创建表达式，您可以使用大括号控制优先级，您还可以使用分号终止语句。有关 MVEL 语言的详细参考，请参阅  
<http://mvel.codehaus.org>。

MVEL 语言具有供应商认证实用工具函数和您还可以在供应商认证表达式中使用的供应商认证全局变量。

要学习使用函数、操作符和全局变量，您可以使用 CA Performance Center 的“供应商认证”选项卡。

此部分包含以下主题：

[表达式运算符 \(p. 89\)](#)

[函数和全局变量 \(p. 91\)](#)

## 表达式运算符

本节说明供应商认证表达式中可使用的运算符。

在供应商认证表达式中可以使用 MVEL 语法。MVEL 是适用于 Java 环境的公开嵌入式表达式语言，其语法与 Java 类似。MVEL 支持类似于 Java 表达式的表达式。

您可以使用运算符创建表达式，您可以使用大括号控制优先级，您还可以使用分号终止语句。有关 MVEL 语言的详细参考，请参阅  
<http://mvel.codehaus.org>。

下表概述可用的运算符：

**注意：**在 XML 文档中，使用 XML 命名实体 (XNE) 表现方式。

运算符	XNE	说明	示例
=		赋值	a = 1
==		等于	"fred" == "fred"
!=		不等于	"fred" != "tom"
>	&gt;	大于	1 > 0 为真
<	&lt;	小于	0 < 1 为真
>=		大于或等于	1 >= 0 为真
<=		小于或等于	1 <= 1 为真
包含		验证左侧的值是否包含右侧的值	"tomcat" contains "cat"
isdef		测试是否定义了变量	isdef a
+		添加	1 + 1
+		连接	"one " + "two"
-		减	2 - 1
*		乘	2 * 2
/		除	4 / 2
%		取模	5 % 2
&&	&amp;&amp;	逻辑与	(x>-1) && (x<1)
		逻辑或	(x<-1)    (x>1)
&	&amp;	按位与运算	17 & 0xF
		按位或运算	4   1
^		按位异或运算	5 ^ 1
!		非	! True
?		三元运算符	age > 17 ? "allow" : "deny"

## 函数和全局变量

本节介绍您可以在供应商认证表达式中使用的一些表达式运算符、函数和全局变量。

在供应商认证表达式中可以使用 MVEL 语法。MVEL 是适用于 Java 环境的公开嵌入式表达式语言，其语法与 Java 类似。MVEL 支持类似于 Java 表达式的表达式。

您可以使用运算符创建表达式，您可以使用大括号控制优先级，您还可以使用分号终止语句。有关 MVEL 语言的详细参考，请参阅 <http://mvel.codehaus.org>。

MVEL 语言具有供应商认证实用工具函数和您还可以在供应商认证表达式中使用的供应商认证全局变量。

要学习使用函数和全局变量，您可以使用 CA Performance Center 中的“供应商认证”选项卡。

### availabilityWithSysUptime 函数

该函数使用 sysUptime 和轮询持续时间计算可用性百分比，以允许宽限期。使用该函数来认证设备。

#### 语法

该函数使用以下格式：

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

#### 参数

##### sysUpTime

自上次重新初始化系统的网络管理部分到现在经过的时间（以百分之一秒为单位）。

##### duration

轮询持续时间（秒）。使用全局变量 \_rspDuration。有关详细信息，请参阅高级示例。

#### 返回值

返回可用性百分比 (0—100)，或者在无效数据通过时返回“null”。

## 示例

当 sysUpTime 的值为 30000 且轮询持续时间为 300 时，下列表达式将生成以下结果：

**表达式:**

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

**结果:**

```
100
```

当 sysUpTime 的值为 6000 且轮询持续时间为 300 时，同一表达式将生成以下结果：

**结果:**

```
20
```

当 sysUpTime 的值为 3000 且轮询持续时间为 300 时，同一表达式将生成以下结果：

**结果:**

```
100
```

## 高级示例

下列表达式取自“系统统计信息”供应商认证：

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

## mapModel 函数

该函数使用 objectID (sysObjectID) 的值，并将系统 OID 映射到模型名称字符串中。使用该函数来认证设备。

### 语法

该函数使用以下格式：

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

### 参数

#### sysObjectID

要进行解析的对象 ID 值。

## 返回值

返回包含映射的模型名称的字符串。

## 示例

当 OID 的值为 1.3.6.1.4.1.9.1.223 时，下列表达式将生成以下结果：

### 表达式：

```
mapModel (oid )
```

### 结果：

Cisco7204VXR

当 OID 的值为 1.3.6.1.4.1 时，同一表达式将生成以下结果：

### 结果：

未知的 1.3.6.1.4.1

## 高级示例

下列表达式取自“系统统计信息”供应商认证：

```
Model=mapModel (sysObjectID)
```

## mapVendor 函数

该函数使用 objectID (sysObjectID) 的值，并将系统 OID 映射到供应商名称字符串中。使用该函数来认证设备。

## 语法

该函数使用以下格式：

```
String mapVendor ( ObjectID sysObjectID )
```

## 参数

### sysObjectID

要进行解析的对象 ID 值。

## 返回值

返回包含映射的供应商名称的字符串。

## 示例

当 OID 的值为 1.3.6.2.1.2.2636.0 时，下列表达式将生成以下结果：

**表达式：**

```
mapVendor (oid )
```

**结果：**

```
Juniper
```

当 OID 的值为 1.3.6.2.1.2.1234567.0 时，同一表达式将生成以下结果：

**结果：**

```
未知
```

## 高级示例

下列表达式取自“系统统计信息”供应商认证：

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

## snmpConstArrayMap 函数

该函数将值（索引）映射到一组常数值（数组）。如有必要，该函数会将输入值四舍五入到最接近的整数值。然后，它将该整数值用作常量值（数组）集的索引，这些常量值显示为 c0、c1，直至 cn-1。c 值必须是整数。该函数在解析表达式时检查这些值，并返回 cx。如果值不在 0 到 n-1（包含）之间的域中，则结果为 0（不显示错误消息）。使用该函数来认证设备。

## 语法

该函数使用以下格式：

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

## 参数

**index**

用作数组中的索引的 Double 值。

**array**

整数值的任何范围。

## 返回值

从数组返回整数值。 空索引值将返回“null”。

## 示例

当索引为 2 且数组为 {5, 6, 7, 8, 9, 4} 时，下列表达式将生成以下结果：

### 表达式:

```
snmpConstArrayMap (index, array)
```

### 结果:

7

当索引为 4.88 且数组为 {5, 6, 7, 8, 9, 4} 时，下列表达式将生成以下结果：

### 表达式:

```
snmpConstArrayMap (value, array)
```

### 结果:

4

## 高级示例

下列表达式取自“常规调制解调器”供应商认证：

```
SpeedOut=snmpConstArrayMap(mdmCsFinalTxLinkRate,{0,110,300,600,1200,2400,4800,7200,9600,12000,14000,16000,19000,38000,75,450,0,57000,21000,24000,26000,2800,0,31000,33000,25333,26666,28000,29333,30666,32000,33333,34666,36000,37333,38666,40000,41333,42666,44000,45333,46666,48000,49333,50666,52000,53333,54666,56000,57333,58666,60000,61333,62666,64000})
```

## mvelInfo 函数

此函数将通过输入参数对 karaf.log 文件的 INFO 级别进行填充。使用此函数记录轮询值（来自您认为其返回结果不准确的报告）。

Data Collector 的轮询日志仅显示轮询属性的值，而该报告仅显示计算结果。mvelInfo 函数允许您查看输入轮询值，以确定哪个位置的计算出错。

## 语法

该函数使用以下格式：

字符串 mvelInfo (数组对象)

**参数**

**对象**

对象数组将记录在 Data Collector karaf.log 文件的 INFO 级别之下。

**返回值**

空

**示例**

下列表达式记录 karaf.log 文件中的 cpmCPUTotal5minRev。

**表达式:**

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

**结果:**

空

**结果 (karaf.log):**

```
MVEL info: cpmCPUTotal5minRev=15
```

**高级示例**

```
mvelInfo([ "cpmCPUTotal5minRev=" , cpmCPUTotal5minRev, " cpmpCPUTotal10minRev=" ,  
cpmpCPUTotal10minRev]); cpmpCPUTotal10minRev;
```

**结果:**

12

**结果 (karaf.log):**

```
MVEL info: cpmCPUTotal5minRev=15 cpmpCPUTotal10minRev=12
```

## mvelWarn 函数

此函数将通过输入参数对 karaf.log 文件的 WARN 级别进行填充。使用此函数记录轮询值（来自您认为其返回结果不准确的报告）。

Data Collector 的轮询日志仅显示轮询属性的值，而该报告仅显示计算结果。mvelWarn 函数允许您查看输入轮询值，以确定哪个位置的计算出错。

## 语法

该函数使用以下格式：

字符串 `mvelWarn` (数组对象)

## 参数

### 对象

对象数组将记录在 Data Collector karaf.log 文件的 WARN 级别之下。

## 返回值

空

## 示例

下列表达式记录 karaf.log 文件中的 `cpmCPUTotal5minRev`。

### 表达式:

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

### 结果:

空

### 结果 (karaf.log):

```
MVEL warn: cpmCPUTotal5minRev=15
```

## 高级示例

```
mvelWarn([ "cpmCPUTotal5minRev=" , cpmCPUTotal5minRev , " cpmpCPUTotal10minRev=" ,  
cpmpCPUTotal10minRev ]); cpmpCPUTotal10minRev;
```

### 结果:

12

### 结果 (karaf.log):

```
MVEL warn: cpmCPUTotal5minRev=15 cpmpCPUTotal10minRev=12
```

## mvelError 函数

此函数将通过输入参数对 karaf.log 文件的 ERROR 级别进行填充。使用此函数记录轮询值（来自您认为其返回结果不准确的报告）。

Data Collector 的轮询日志仅显示轮询属性的值，而该报告仅显示计算结果。mvelError 函数允许您查看输入轮询值，以确定哪个位置的计算出错。

### 语法

该函数使用以下格式：

字符串 mvelError (数组对象)

### 参数

#### 对象

对象数组将记录在 Data Collector karaf.log 文件的 ERROR 级别之下。

### 返回值

空

### 示例

下列表达式记录 karaf.log 文件中的 cpmCPUTotal5minRev。

#### 表达式:

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

#### 结果:

空

#### 结果 (karaf.log):

```
MVEL error: cpmCPUTotal5minRev=15
```

## 高级示例

```
mvelError([“cpmCPUTotal5minRev=”, cpmCPUTotal5minRev, “cpmCPUTotal10minRev=”,  
cpmCPUTotal10minRev]); cpmCPUTotal10minRev;
```

结果:

12

结果 (**karaf.log**):

```
MVEL error: cpmCPUTotal5minRev=15 cpmCPUTotal10minRev=12
```

## **mvelDebug** 函数

此函数将通过输入参数对 **karaf.log** 文件的 **DEBUG** 级别进行填充。使用此函数记录轮询值（来自您认为其返回结果不准确的报告）。

**Data Collector** 的轮询日志仅显示轮询属性的值，而该报告仅显示计算结果。**mvelDebug** 函数允许您查看输入轮询值，以确定哪个位置的计算出错。

### 语法

该函数使用以下格式：

字符串 **mvelDebug** (数组对象)

### 参数

#### 对象

对象数组将记录在 **Data Collector karaf.log** 文件的 **DEBUG** 级别之下。

### 返回值

空

## 示例

下列表达式记录 karaf.log 文件中的 cpmCPUTotal5minRev。

表达式:

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

结果:

空

结果 (karaf.log):

```
MVEL debug: cpmCPUTotal5minRev=15
```

## 高级示例

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

结果:

12

结果 (karaf.log):

```
MVEL debug: cpmCPUTotal5minRev=15 cpmCPUTotal10minRev=12
```

## mvelDebug 函数

此函数将通过输入参数对 karaf.log 文件的 TRACE 级别进行填充。使用此函数记录轮询值（来自您认为其返回结果不准确的报告）。

Data Collector 的轮询日志仅显示轮询属性的值，而该报告仅显示计算结果。mvelTrace 函数允许您查看输入轮询值，以确定哪个位置的计算出错。

## 语法

该函数使用以下格式:

字符串 mvelTrace (数组对象)

## 参数

### 对象

对象数组将记录在 Data Collector karaf.log 文件的 TRACE 级别之下。

## 返回值

空

## 示例

下列表达式记录 karaf.log 文件中的 cpmCPUTotal5minRev。

### 表达式:

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

### 结果:

空

### 结果 (karaf.log):

```
MVEL trace: cpmCPUTotal5minRev=15
```

## 高级示例

```
mvelTrace([ "cpmCPUTotal5minRev=" , cpmCPUTotal5minRev , " cpnCPUTotal10minRev=" ,  
cpnCPUTotal10minRev]); cpnCPUTotal10minRev;
```

### 结果:

12

### 结果 (karaf.log):

```
MVEL trace: cpmCPUTotal5minRev=15 cpnCPUTotal10minRev=12
```

## snmpCounter64 函数

该函数评估两个 32 位数值，并返回包含 64 位表述的值。使用该函数来认证设备。将 hiVal 转换为 32 位，添加 lowVal，并将结果置于 64 位返回变量中。

## 语法

该函数使用以下格式：

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

## 参数

### hiVal

表示高阶数位的 32 位数值。

### lowVal

表示低阶数位的 32 位数值。

## 返回值

返回两个 32 位数值的 64 位表述，或者在任意一个 32 位值输入为空时返回“null”。

## 示例

当 hiVal 的值为 88 且 lowVal 的值为 558 时，下列表达式将生成以下结果：

### 表达式:

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

### 结果:

377957122606

## 高级示例

下列表达式取自“Cisco CBQos 类映射”供应商认证。此认证包含许多 snmpMax 示例：

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

## snmpGetUpSinceTime 函数

此函数将基于距离当前时间的秒数来返回系统打开的时间。

## 语法

该函数使用以下格式：

```
snmpGetUpSinceTime (启动时间)
```

## 参数

### upTime

从当前时间开始以来的秒数。您可以从下列 OID: 1.3.6.1.2.1.1.3.0 获取系统启动时间。在时间过去之前，将其转换为厘秒。

## 返回值

返回设备自当前时间以来开机所用的总秒数。

## snmpMax 函数

该函数将返回两个 64 位值中的较大值。 使用该函数来认证设备。

### 语法

该函数使用以下格式：

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

### 参数

#### val1

第一个 64 位 BigInteger 值。

#### val2

第二个 64 位 BigInteger 值。

## 返回值

返回通过的两个 BigInteger 值中的最大值，或者在其中一个 BigInteger 值输入为空时返回“null”。

### 示例

当 val1 的值为  $2^{32}$  且 val2 的值为 10 时，下列表达式将生成以下结果：

#### 表达式:

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

#### 结果:

$2^{32}$

当 val1 的值为 5864 且 val2 的值为 134556890 时，同一表达式将生成以下结果：

结果：

134556890

### 高级示例

下列表达式取自“Cisco CBQos 类映射”供应商认证。此认证包含许多 snmpMax 示例：

```
PrePolicyPackets=snmpMax(0,snmpCounter64(cbQosCMPrePolicyPktOverflow,cbQosCMP  
rePolicyPkt))
```

## snmpObjectIDToASCIIString 函数

此函数可将 SNMP OID 值转换成字符串表示形式。任何前导或尾随空格都将被移除。

### 语法

该函数使用以下格式：

```
snmpObjectIDToASCIIString(对象 ID oid)
```

### 参数

**oid**

要转换为字符串的对象 ID。

## snmpOIDParser 函数

该函数使用 objectID (OID) 的值，并基于 startIndex 和 endIndex 值解析出 OID 子集。索引以 1 为基础。如果 endIndex 是 -1，将转至 OID 的结尾。使用该函数来认证设备。

### 语法

该函数使用以下格式：

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

## 参数

### OID

要解析的对象 ID (OID) 值。

### startIndex

索引的整数值，达到该值时将开始解析。

### endIndex

索引的整数值，达到该值时将停止解析。

## 返回值

返回已解析的子集 ObjectID (OID)。

## 示例

当 OID 的值为 1.2.3.4.5.6.7.8.9.10、startIndex 的值为 1 且 endIndex 的值为 5 时，下列表达式将生成以下结果：

### 表达式:

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

### 结果:

1.2.3.4.5

当 OID 的值为 1.2.3.4.5.6.7.8.9.10、startIndex 的值为 6 且 endIndex 的值为 -1 时，同一表达式将生成以下结果：

### 结果:

6.7.8.9.10

## 高级示例

下列表达式取自“Cisco CBQos 类映射”供应商认证：

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

## snmpOctetStringFloat 函数

该函数将 SNMP 八进制字符串转换为浮点值。 使用该函数来认证设备。  
SNMP 八进制字符串是七位 ASCII 字符串。

### 语法

该函数使用以下格式：

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

### 参数

#### octetString

SNMP 八进制字符串。

### 返回值

返回转换的字符串值，或者在该函数无法转换字符串时返回“null”。

### 示例

当 octetString 的值为 {0x33, 0x33, 0x2E, 0x33, 0x33} 时，下列表达式将生成以下结果：

#### 表达式:

```
snmpOctetStringFloat (octetString)
```

#### 结果:

33.33

当 octetString 的值为 {0x36, 0x36, 0x36} 时，同一表达式将生成以下结果：

#### 结果:

666.0

## snmpProtectedDiv 函数

该函数除以两个 Double 值，然后将相除结果作为 Double 值返回。如果被除数或者除数为空或 0.0，则返回值为 0.0。使用此函数可防止表达式的除数为空或为 0。[assign the value for cadr in your book] 可以包含空或零值（如在轮询失败时）。在这种情况下，您可以使用此函数避免除数为零的异常。

### 语法

该函数使用以下格式：

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

### 参数

#### val1

是被除数。这是要被 val2 除的 Double 值（浮点数）。（*Double* 是 Java 数据类型。）

#### val2

是除数。这是 Double 值（浮点数）。（*Double* 是 Java 数据类型。）

### 返回值

如果被除数或除数为空值或为 0.0（*Double* 是一种 Java 数据类型），则将相除结果作为 Double 值或 0.0 返回。

### 示例

当 val1 的值为 7.2 且 val2 的值为 2 时，下列表达式将生成以下结果：

#### 表达式:

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

#### 结果:

3.6

当 val1 的值为 7.2 且 val2 的值为空或 0.0 时，下列表达式将生成以下结果：

#### 结果:

0.0

### 高级示例

下列表达式取自供应商认证：

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

## snmpRound 函数

该函数将数值四舍五入到最接近的整数值。

### 语法

该函数使用以下格式：

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

### 参数

#### dNumber

将四舍五入的 Double 值（浮点数）（*Double* 是一种 Java 数据类型）。

### 返回值

返回 Long 值，它是最接近 dNumber 中提供的值的整数值（*Long* 是一种 Java 数据类型）。

### 示例

当 dNumber 的值为 3.5 时，下列表达式将生成以下结果：

#### 表达式:

```
snmpRound(dNumber)
```

#### 结果:

4

当 dNumber 的值为 3.4 时，同一表达式将生成以下结果：

**结果：**

3

## 高级示例

下列表达式取自“Cisco IPSLA 抖动精度统计”供应商认证：

```
PathAvailability=snmpRound(rttMonJitterStatsNumOfRTT /
(rtMonJitterStatsNumOfRTT + rtMonJitterStatsPacketLossSD +
rtMonJitterStatsPacketLossDS + rtMonJitterStatsPacketOutOfSequence +
rtMonJitterStatsPacketMIA + rtMonJitterStatsPacketLateArrival +
rtMonJitterStatsError + rtMonJitterStatsBusies + 1/100) * 100)
```

## snmpStringParser 函数

此函数仅为内部使用而编写。它将解析接收自 CA Application Insight Module (AIM) 的 IP 地址。CA 仅使用内部类对此函数进行过测试；因此，其他类型的类将受到支持。

### 语法

该函数使用以下格式：

`snmpStringParser (分隔符, 要转换的类型, 解析 1 的字符串, 解析 2 的字符串)`

### 参数

#### 要转换的类型

解析供应字符串的类型的类。

#### 解析 1 和 2 的字符串

要解析的 IP 地址。两个字符串可使您能够提供 IPv4 和 IPv6 格式的地址。

### 返回值

返回转换的字符串值，或在该函数无法解析字符串时返回“null”。

## snmpSvcs 函数

该函数从代理的 sysObjectOID、sysService 和 ipForwarding MIB 变量中获取值，然后确定 SNMP 代理支持哪些服务。例如，Router/Switch/Repeater/Host 可能是受支持的服务，如 SNMP MIB RFC 1213 中所定义。

由于自定义设备类型的优先级高于系统的设备类型优先级，因此评估该函数中的返回服务如下所示：

- 如果将 sysObjectID OID 映射在 DeviceTypes 文件中，那么返回服务来自此文件。
- 如果没有将 sysObjectID OID 映射在 DeviceTypes 文件中，那么 sysServices 和 ipForwarding 可用于返回支持服务。

## 语法

该函数使用以下格式：

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

## 参数

### **sysObjectID**

要进行解析的对象 ID 值。

### **sysServices**

每个数位各表示不同服务的整数，如 switch/repeater/host。

### **ipForwarding**

整数值，其中该实体在转发数据报时用作 IP 网关或 IP 主机。该实体接收转发的数据报，但是转发的数据报不会发送给该实体。

## 返回值

返回下列一个或多个设备服务的列表：

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

## 示例

当 sysServices 的值为 8, ipForwarding 的值为 0 且未在设备类型文件中找到 sysObjectID 时, 下列表达式将生成以下结果:

**表达式:**

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

**结果:**

```
DeviceService[HOST]
```

## 高级示例

下列表达式取自“系统统计信息”供应商认证:

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

## storePortReconfig 函数

该函数返回包含表示 ifNumber、ifTableLastChange、ifStackLastChange 的值的 XML 的字符串。需要时, 您可以使用 XML 跟踪设备变更以及重新发现在设备上的接口。

### 语法

该函数使用以下格式:

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

### 参数

#### ifNumber

设备上的端口数目。

#### ifTableLastChange

最新端口表更改的日期和时间 (以毫秒为单位), 根据 1970 年 1 月 1 日 GMT 的开始日期和时间来计算。

#### ifStackLastChange

最新端口堆栈更改的日期和时间 (以毫秒为单位), 根据 1970 年 1 月 1 日 GMT 的开始日期和时间来计算。

## 返回值

返回包含采用以下示例中所示格式的 XML 的字符串。

## 示例

当 `ifNumber` 的值为 5、`ifTableLastChange` 的值为 123456 且 `ifStackLastChange` 的值为 234567 时，下列表达式将生成以下结果：

**表达式：**

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

**结果：**

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

## 全局变量

Data Aggregator 支持以下全局变量：

### **\_rspDuration**

包含当前轮询周期持续时间（秒）的 Long 值（Java 数据类型）。

**注意：**“系统统计信息”供应商认证包含 `_rspDuration` 的使用示例。

### **\_rspTimestamp**

包含自 1970 年 1 月 1 日 GMT 起当前轮询周期（毫秒）开始的时间戳的 Long 值（Java 数据类型）。

# 附录 B：故障排除

---

此部分包含以下主题：

[故障排除：无法创建供应商认证 \(p. 113\)](#)

[故障排除：度量标准系列不受支持 \(p. 114\)](#)

[故障排除：度量标准系列不完整 \(p. 114\)](#)

[故障排除：供应商认证表达式不正确 \(p. 115\)](#)

## 故障排除：无法创建供应商认证

### 症状：

我尝试创建供应商认证，但收到创建失败的错误消息。

### 解决方案：

打开 Data Aggregator 安装目录中的 karaf 日志文件，然后执行以下步骤：

1. 查找您选择的 MIB 名称字符串或度量标准系列的名称。
2. 复查异常的堆栈跟踪，并查找 CertManagerException 和错误原因。错误原因因异常而异。

**示例：**表达式解析程序不应在 ++ 之后出现令牌，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. 基于提供的原因，修正错误。验证是否满足下列要求：

- 表达式组不能同时包含标量和表条目。
- 表达式必须包含有效的语法。
- 至少为度量标准系列变量定义了一个表达式。
- 至少定义了两个度量标准系列变量；尤其需要注意的是，名称和索引是必需的（仅限标量度量标准除外）。
- 用于表达式的供应商认证变量必须来自选定的 MIB 表（在用户界面中有效）。

## 故障排除：度量标准系列不受支持

### 症状：

我创建了监视配置文件以在设备集合上轮询度量标准系列。但是，在“轮询的度量标准系列”表中，其中一个度量标准系列处于“不受支持”状态。

### 解决方案：

要更正问题，请执行以下步骤：

1. 确认轮询的设备对 SNMP 查询作出响应。
2. 单击不受支持的度量标准系列即可导航到该位置。
3. 确认供应商认证支持度量标准系列。如果未定义供应商认证，请创建一个自定义供应商认证。
4. 确认所有关键供应商认证属性在设备上受支持。如果支持所有关键供应商认证属性，请导航回该设备，并选择已为其添加自定义供应商认证的度量标准系列，然后单击“更新度量标准系列”。

将更新您的设备配置。

## 故障排除：度量标准系列不完整

### 症状：

我已成功导入自定义度量标准系列，但之后发现了有缺陷的度量标准定义。例如，`<Name>` 属性的最大长度为 32 个字符。如果超出此限制，则会导致同步问题。

### 解决方案：

删除自定义度量标准系列时务必谨慎，如下所示：

1. 找到 `IMDataAggregator/apache-karaf-2.3.0/deploy` 目录。
2. 删除为度量标准系列创建和部署的 XML 文件。这些文件的名称如下：
  - `im.ca.com-normalizer-<technology>.xml`
  - `im.ca.com-inventory-<technology>.xml`

如果适用，请同时删除为供应商认证创建的文件：

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

3. 通过运行以下命令重新启动 Data Aggregator:

```
service dadaemon restart
```

重新启动 Data Aggregator 之后，请确认先前导入的度量标准系列或供应商认证不再显示在 CA Performance Center 中。此外，针对该自定义认证所有以前发现的组件都将删除。

4. 在 CA Performance Center 中单击“管理”，“数据源”。

5. 选择 Data Aggregator，然后单击“重新同步”按钮。

剩余度量标准系列的组成内容将在 Data Aggregator 和 CA Performance Center 之间同步。

6. 编辑并更正您的自定义度量标准系列 XML 文件。

7. 导入改正后的度量标准系列 xml 文件。

## 故障排除：供应商认证表达式不正确

### 症状：

MVEL 编译器可能不会为错误表达式提供评估异常（错误）。针对某些语法错误可能会出现此情况，包括但不限于缺少括号或开括号以及多个星号。

不正确的表达式已被编译，在使用适当的变量执行表达式评估之前不会显示错误情况。不会填充作为预期表达式的目标的数据库列。

### 解决方案：

使用下列步骤为 ExpressionEvaluator 打开调试登录：

1. 找到 IMDataAggregator/apache-karaf-2.3.0/etc 目录。

2. 打开 org.ops4j.pax.logging.cfg 文件并创建以下条目：

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

3. 通过运行以下命令重新启动 Data Aggregator:

```
service dadaemon restart
```

4. 在 IMDataAggregator/apache-karaf-2.3.0/data/log 目录的 karaf.log 文件中搜索评估异常。