

管理OSPF与SNMP上下文的多个实例

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[规则](#)

[背景信息](#)

[意识的Snmp context](#)

[配置](#)

[验证](#)

[SNMPv2验证](#)

[SNMPv3验证](#)

[相关信息](#)

简介

描述如何使用Snmp context管理开放最短路径优先(OSPF)多个实例的本文为SNMPv2提供配置示例和SNMPv3。

先决条件

要求

本文档没有任何特定的要求。

使用的组件

本文档不限于特定的软件和硬件版本。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始(默认)配置。如果您使用的是真实网络,请确保您已经了解所有命令的潜在影响。

规则

有关文档规则的详细信息,请参阅 [Cisco 技术提示规则](#)。

背景信息

IETF定义的OSPF MIB ([RFC 1850](#))只设计与一OSPF程序/实例一起使用在一个给的路由器。

例如，有仅单个 *ospfRouterId* 对象，不是表他们。为了处理多个实例，[RFC 4750](#) 建议您使用 SNMPv3 上下文提供每实例视图。

意识的 Snmp context

在之前进行意识 IOS OSPF SNMP 代码的上下文，系统将选择更或较不随机的“默认”实例，当返回标量对象和一些表。在这些情况下，从其他实例的信息通过 SNMP 不是可用的。对于一些其他表，SNMP 将一起捣碎从所有实例的条目，不用任何方式辩明哪些是哪些。在许多情况下，这能导致模棱两可或重复的条目。它不是特别是良好的做法在 IP 地址和邻接路由器 ID 也许不是唯一的 PE-CE 配置方面。这做监听和故障排除不可能各自的 CE 的实例困难或。

使用当前上下文意识 IOS 代码(当上下文没有指定)时，标量对象的旧有行为仍然存在。唯一的更改是当前限制所有而不是某些表对“默认” OSPF 实例和标量一样。当供应时上下文，SNMP 查询可以被瞄准到特定 OSPF 实例，并且该实例的所有信息可以获取以一致和毫不含糊的方式。

如果使用 SNMPv3，上下文字符串可以直接地与投票一起提供。SNMPv2C 不提供上下文。然而，您在 IOS 配置里能映射 SNMP 团体字符串到上下文，并且这些上下文可以用于处理 SNMPv2 投票到特定 OSPF 实例。

配置

此配置示例根据 SNMPv2 :

```
路由器 1

Router1#

router ospf 1
  router-id 1.1.1.111
  log-adjacency-changes
  snmp context context1
!
router ospf 2
  router-id 4.4.4.111
  log-adjacency-changes
  snmp context context2
!--- Associates the SNMP context with the instance. !
snmp-server user u2 g2 v2c !--- Configures the user u2
to the SNMP group g2 and !--- specifies the group is
using the SNMPv2c security model. snmp-server group g2
v2c !--- Configures the SNMP group g2 and specifies !---
the group is using the SNMPv2c security model. snmp-
server group g2 v2c context context1 snmp-server group
g2 v2c context context2 snmp-server community public RO
!--- Community access string to permit access !--- to
the SNMP. snmp-server community cx1 RO snmp-server
community cx2 RO snmp-server context context1 snmp-
server context context2 snmp mib community-map cx1
context context1 security-name u2 !--- Associates the
SNMP community cx1 with !--- the context context 1. snmp
mib community-map cx2 context context2 security-name u2
```

此配置示例根据 SNMPv3 :

```
路由器 1
```

```
Router1#  
  
router ospf 1  
  router-id 1.1.1.111  
  log-adjacency-changes  
  snmp context context1  
!  
router ospf 2  
  router-id 4.4.4.111  
  log-adjacency-changes  
  snmp context context2  
!  
snmp-server user u1 g1 v3  
snmp-server group g1 v3 noauth  
snmp-server group g1 v3 noauth context context1  
snmp-server group g1 v3 noauth context context2  
snmp-server context context1  
snmp-server context context2
```

注意： 有关本文档所用命令的详细信息，请使用[命令查找工具](#)（[仅限注册用户](#)）。

验证

您能使用**snmpwalk**命令在所有客户端机器为了验证输出。

注意： [命令输出解释程序](#)（[仅限注册用户](#)）(OIT) 支持某些 **show** 命令。使用 OIT 可查看对 show 命令输出的分析。

SNMPv2验证

SNMPv2

```
linux>snmpwalk -c public -v2c irp-view14:7890 OSPF-  
MIB::ospfRouterId.0  
OSPF-MIB::ospfRouterId.0 = IPAddress: 4.4.4.111  
  
linux>snmpwalk -c cx1 -v2c irp-view14:7890 OSPF-  
MIB::ospfRouterId.0  
OSPF-MIB::ospfRouterId.0 = IPAddress: 1.1.1.111  
  
linux>snmpwalk -c cx2 -v2c irp-view14:7890 OSPF-  
MIB::ospfRouterId.0  
OSPF-MIB::ospfRouterId.0 = IPAddress: 4.4.4.111
```

SNMPv3验证

SNMPv3

```
linux>snmpwalk -u u1 -v3 irp-view14:7890 OSPF-  
MIB::ospfRouterId.0  
OSPF-MIB::ospfRouterId.0 = IPAddress: 4.4.4.111  
  
linux>snmpwalk -u u1 -v3 -n context1 irp-view14:7890  
OSPF-MIB::ospfRouterId.0  
OSPF-MIB::ospfRouterId.0 = IPAddress: 1.1.1.111
```

```
linux>snmpwalk -u u1 -v3 -n context2 irp-view14:7890
OSPF-MIB::ospfRouterId.0
OSPF-MIB::ospfRouterId.0 = IPAddress: 4.4.4.111
```

[相关信息](#)

- [使用 SNMP 管理 OSPF 配置](#)
- [技术支持和文档 - Cisco Systems](#)