

# Gerenciar várias instâncias do OSPF com contextos SNMP

## Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Conventions](#)

[Informações de Apoio](#)

[Reconhecimento de contexto SNMP](#)

[Configuração](#)

[Verificar](#)

[Verificação SNMPv2](#)

[Verificação SNMPv3](#)

[Informações Relacionadas](#)

## [Introduction](#)

Este documento fornece configurações de exemplo para SNMPv2 e SNMPv3 que descrevem como usar contextos SNMP para gerenciar várias instâncias do OSPF (Open Shortest Path First).

## [Prerequisites](#)

## [Requirements](#)

Não existem requisitos específicos para este documento.

## [Componentes Utilizados](#)

Este documento não se restringe a versões de software e hardware específicas.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

## [Conventions](#)

Consulte as [Convenções de Dicas Técnicas da Cisco para obter mais informações sobre convenções de documentos](#).

## Informações de Apoio

A MIB OSPF definida pela IETF ([RFC 1850](#) [2]) foi projetada para funcionar com apenas um processo/instância OSPF em um determinado roteador.

Por exemplo, há apenas um único objeto *ospfRouterId*, não uma tabela deles. Para lidar com várias instâncias, o [RFC 4750](#) [3] sugere que você use contextos SNMPv3 para fornecer visualizações por instância.

## Reconhecimento de contexto SNMP

Antes de tornar o contexto do código SNMP do IOS OSPF consciente, o sistema escolheria uma instância "padrão" mais ou menos aleatória quando retornasse os objetos escalares e algumas tabelas. Nesses casos, as informações das outras instâncias não estavam disponíveis via SNMP. Para algumas outras tabelas, o SNMP reuniria as entradas de todas as instâncias sem nenhuma forma de descobrir qual era. Em muitos casos, isso pode levar a entradas ambíguas ou duplicadas. Especialmente não foi uma boa prática em configurações PE-CE em que os endereços IP e IDs de roteador vizinhos podem não ser exclusivos. Isso tornou o monitoramento e a solução de problemas de instâncias CE individuais difíceis ou impossíveis.

Com o código IOS sensível ao contexto atual (quando nenhum contexto é especificado), o comportamento antigo dos objetos escalares ainda existe. A única alteração é que agora ela limita todas as tabelas, em vez de apenas algumas, à mesma instância "padrão" do OSPF que as escalares. Quando contextos são fornecidos, as consultas SNMP podem ser direcionadas para uma instância OSPF específica, e todas as informações para essa instância podem ser recuperadas de maneira consistente e inequívoca.

Se SNMPv3 for usado, a string de contexto pode ser fornecida diretamente com a pesquisa. O SNMPv2c não fornece um contexto. No entanto, você pode mapear strings de comunidade SNMP para contextos na configuração do IOS, e esses contextos podem ser usados para direcionar pesquisas SNMPv2 para uma instância específica do OSPF.

## Configuração

Este exemplo de configuração é baseado em SNMPv2:

```
Roteador 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
```

Este exemplo de configuração é baseado em SNMPv3:

## Roteador 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
```

**Nota:** Use a [Command Lookup Tool](#) (somente clientes registrados) para obter mais informações sobre os comandos usados neste documento.

## [Verificar](#)

Você pode usar o comando **snmpwalk** em qualquer máquina cliente para verificar a saída.

**Observação:** a [Output Interpreter Tool](#) ([somente](#) clientes [registrados](#)) (OIT) suporta determinados comandos **show**. Use a OIT para exibir uma análise da saída do comando show.

## [Verificação SNMPv2](#)

### 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
```

## Verificação SNMPv3

### 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
```

## Informações Relacionadas

- [Gerenciamento de configuração do OSPF com SNMP](#)
- [Suporte Técnico e Documentação - Cisco Systems](#)