

# 带SNMP的OSPF配置管理

## Contents

[Introduction](#)

[OSPF背景](#)

[进程定义](#)

[进程总管](#)

[进程目标](#)

[处理性能指示器](#)

[处理输入](#)

[进程输出](#)

[任务定义](#)

[初始化任务](#)

[重复任务](#)

[数据标识](#)

[一般数据特性](#)

[SNMP数据标识](#)

[RMON数据识别](#)

[系统日志数据识别](#)

[Cisco IOS CLI数据证明](#)

[数据收集](#)

[SNMP数据收集](#)

[RMON数据收集](#)

[系统日志数据收集](#)

[Cisco IOS CLI数据数据收集](#)

[数据表示](#)

[OSPF区域报告](#)

[OSPF接口报告](#)

[OSPF相邻报告](#)

[商用和公共互联网监控工具](#)

[SNMP轮询数据](#)

[示例数据收集算法](#)

[Related Information](#)

## [Introduction](#)

开放最短路径优先(OSPF)路由协议是由[RFC 2328 OSPF第2版](#)定义的。本文的目标是提供实现配置管理过程的enable (event)组织验证OSPF配置OSPF设计规划和周期地审核OSPF配置保证与刻意的设计的长期一致性的一个程序框架。

本文着重从ITU-T被定义的FCAPS (故障, 配置, 认为/库存、性能, 安全)型号的配置管理功能。配

置管理是由ITU-T M.3400定义的，当提供作用练习控制，识别，收集数据从和提供数据给NEs (网络单元)。

本文提供的信息在下述几个的主要部分引见。

[OSPF Background部分](#)在OSPF配置的重要方面提供OSPF技术概述包括背景信息。

[进程Definitions部分](#)提供用于的进程定义概述完成OSPF配置管理。进程详细资料被描述根据目标、性能指标、输入、输出和单个任务。

[任务定义](#)区分提供被选派的进程任务定义。每项任务被描述根据目标，任务输入，任务输出，需的资源完成任务和为任务实施者需要的工作技能。

[数据标识部分描述OSPF的数据标识。](#)数据标识考虑信息的来源或找出的地方。例如，信息由系统包含在简单网络管理协议(SNMP)管理信息库(MIB)， Syslog生成的日志文件或者可能由命令行界面(CLI)只获取的内部数据结构。

[本文的数据收集部分描述OSPF数据的集。](#)数据的收藏是密切相关对数据的位置。例如， SNMP MIB数据由几个机制收集例如陷阱、远程监控(RMON)警报和事件或者轮询。内部数据结构维护的数据收集由自动脚本或由用户手册记录到系统发出CLI命令然后记录输出的。

[Data Presentation部分](#)提供示例数据如何在报告格式被提交。在数据被识别并且收集后，分析。本文提供可能使用记录和比较OSPF配置数据的示例报告。

[商用和公共互联网监控工具](#)、[SNMP轮询数据](#)和[示例数据收集算法](#)部分在工具的发展提供信息实现OSPF配置管理过程。

## [OSPF背景](#)

OSPF是设计的内部网关协议在单个自控系统内使用。OSPF用途连接状态或Shortest Path First (SPF) -基础技术，与在路由协议或Bellman-Ford技术比较找到的距离矢量例如路由信息协议(RIP)。单个Link State Advertisement (LSA)描述OSPF路由域的部分，例如，整个自控系统。这些LSA被充斥在路由域中，形成链路状态数据库。在域的每个路由器有一个相同的链路状态数据库。链路状态数据库同步保持与可靠的扩散算法。从链路状态数据库，每个路由器通过计算一个最短路径树构件路由表，有是的树的根的计算机的路由器。此计算通常指dijkstra算法。

LSA是小的，并且每个LSA描述一小块OSPF路由域，特别地，单个路由器的邻近，单个穿透网络的邻近，单个区域间路由或者单个外部路由。

此表定义了OSPF主要特点：

功能	说明
邻接	当对OSPF路由器变得相邻时，以OSPF数据库交换信息包的形式，两路由器通过交换数据库汇总同步他们的链路状态数据库。邻接路由器通过可靠的扩散算法然后保持他们的链路状态数据库同步。由串行线路的路由器连接总是变得相邻。在多路访问网络(以太网)，所有路由器附有网络在指定路由器(DR)和备份指定路由(BDR)附近成为。
指	当DR在所有多路访问网络时选择，产生描述网络的本

定 路 由 器	地环境的网络LSA。它在扩散算法也扮演特殊作用，因为网络的所有路由器通过发送和接受LSA同步他们的链路状态数据库到/从DR在溢流过程中。
备 份 指 定 路 由	当当前DR消失时，BDR在多路访问网络决定加速DR的转换。当BDR接管时，不需要通过在局域网(LAN)的邻接进程。也BDR enable (event)进行的可靠的扩散算法在DR的缺乏在DR的失踪前被注意。
非 广 播 多 址 网 络 支 持	OSPF对待网络，例如帧中继公共数据网(PDN)，好象他们LAN。然而，更多的配置信息为路由器是需要的附有这些网络最初互相找到。
O S P F 配 置 管 理 区 域	OSPF允许自控系统是分类为的区域。这提供路由保护的一个额外的级别，以便路由在区域内保护免受所有信息外部到区域。并且，通过分裂一个自控系统到区域里，根据CPU周期，降低Dijkstra程序的费用。
虚 链 路	通过允许虚链路的配置，OSPF去除对区域布局的拓扑限制在自控系统。
路 由 协 议 交 换 的 认 证	在OSPF路由器收到一个路由协议信息包时候，能在进一步处理它前可选地验证信息包。
灵 活 路 由 度 量	在OSPF中，权值分配到outbound路由器接口。路径的费用是路径的组分接口的总和。默认情况下，路由度量从链路的带宽派生。它可以由系统管理员分配指示网络特性的所有组合例如延迟、带宽和费用。
相 等 代 价	当多个最佳开销路由到目的地存在时，OSPF查找并且使用他们对负载共享数据流对目的地。

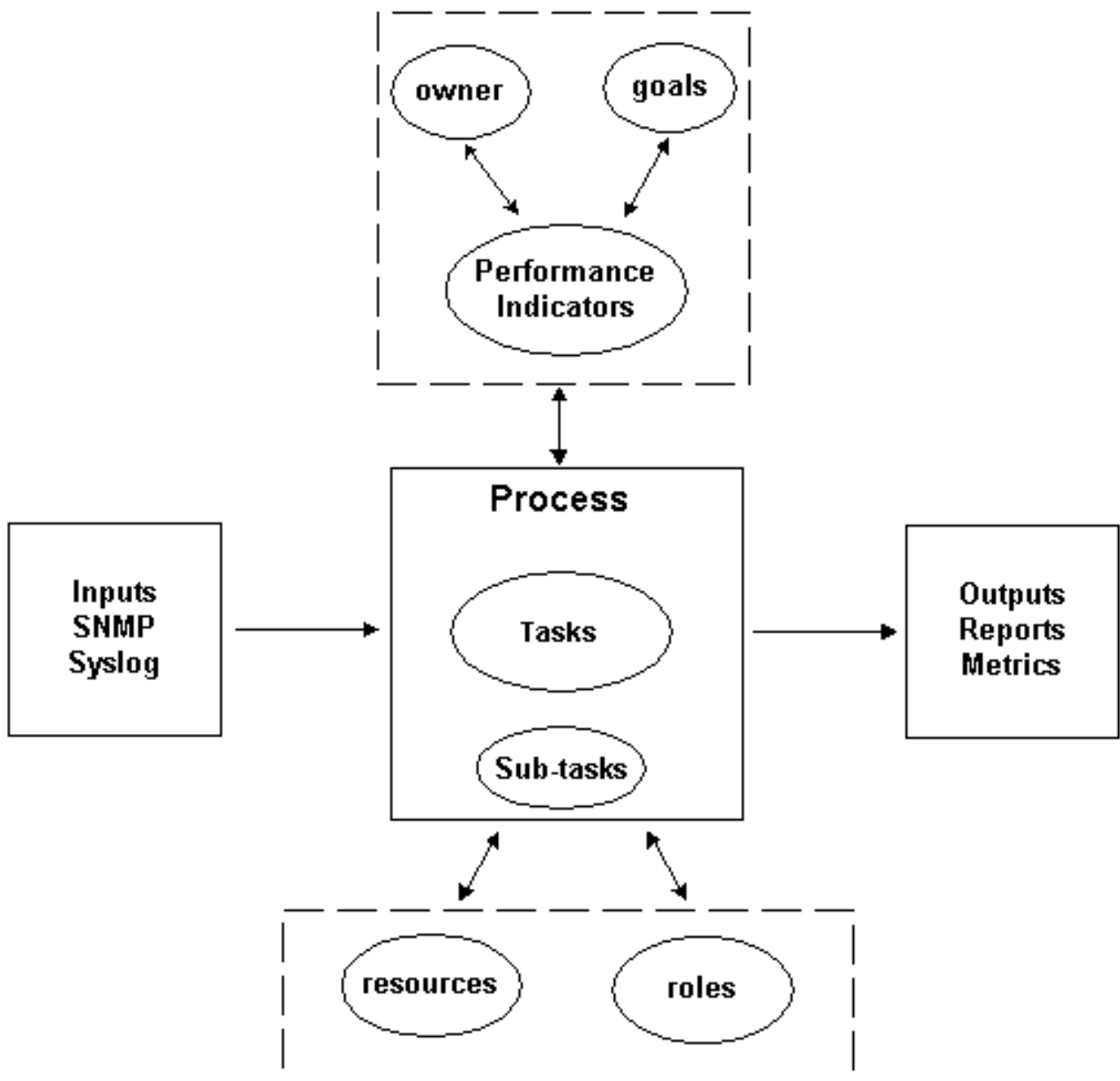
多重通道	
可变长度的子网支持	通过运载与每个建议的目的地的一个网络掩码支持可变长度子网掩码。
残余部分区域支持	要支持有的路由器内存不足，区域可以被配置作为残余部分。外部LSA没有被充斥到和在末端区域中。对外部目的地的路由在末端区域根据默认值独自地。

## 进程定义

进程定义是代理程序执行的一个连续的一系列动作、活动和更改目的在于满足目的或达到目标。

程序控制的是计划和调控的进程，以便于执行进程在一个有效和高效的方法。

如图解，这在下面图显示。



进程的输出必须依照是由组织定义的和根据业务目标的操作准则。如果进程依照套准则，进程被认为有效，因为可以被重复，测量，管理，并且对业务目标有用。如果活动以最少的力气进行，进程也被认为高效率。

## 进程总管

进程跨过多种组织边界。所以，有负责进程的定义的单独进程责任人是重要的。责任人是确定和报告的进程是否重要的是有效和高效的。如果进程无法是有效或高效的，进程总管驾驶进程的修改。进程的修改由更改控制和复核进程管理。

## 进程目标

进程目标设立设置方向和范围进程定义的。目标也用于定义使用测量进程的效果的权值。

此进程的目标是提供框架验证OSPF应用的配置的配置刻意的设计和提供机制周期地审核OSPF配置随着时间的推移保证一致性关于刻意的设计。

## 处理性能指示器

处理性能指示器用于测量进程定义的效果。性能指标应该是可测量和可计量的。如下所示的性能指标是数字或测量在时间之前。性能指标为OSPF配置管理进程被定义如下：

- 循环的所需的的时间的长度通过整个过程。
- 执行要求的频率为了主动地发现OSPF问题，在他们影响用户前。
- 与进程的执行产生关联的网络负载。
- 进程建议使用的纠正措施的数量。
- 由于进程实现的纠正措施的数量。
- 实现纠正措施的所需的的时间的长度。
- 实现纠正措施的所需的的时间的长度。
- 纠正措施积压。
- 停工期归因于OSPF相关问题。
- 在种子文件添加，删除，或修改的项目的数量。这是准确性和稳定性的征兆。

## 处理输入

处理输入用于定义标准和前提对于进程。许多次，处理输入的证明在外部条件提供信息。与OSPF配置管理有关下面提供输入列表。

- OSPF设计文档
- OSPF SNMP轮询收集的MIB数据
- 系统日志信息

## 进程输出

进程输出被定义如下：

- OSPF在本文的[Data Presentation部分](#)定义的配置报告
- OSPF对纠正措施的配置推荐执行

## 任务定义

以下部分定义了与OSPF配置管理和重复任务产生关联的初始化。

### 初始化任务

初始化任务一次在进程的实施时完成，并且不应该执行与进程的每迭代。

### 验证前提任务

在验证前提任务，如果确定任何一个任务不是被实施的也不提供充足的信息有效满足此程序需要，应该由进程总管描述和提交此事实给管理。表下面的分级显示事先需要的初始化任务。

前提任	说明
-----	----

<b>任务</b>	
<b>任务目的和输入</b>	<ol style="list-style-type: none"> <li>1. 验证OSPF设计文件存在，并且以下信息是可用的在网络设计文档：区域定义—名字、地址范围和区域类型局域边界路由器/自治系统边界路由器(ABR/ASBR)证明DR/BDR证明互联网注册(IR)节点和接口分配到区域</li> <li>2. 请使用一个SNMP标准配置模板验证SNMP在网络被配置。 <b>Note:</b> 这以后使用作为输入创建种子文件。</li> <li>3. 请使用一个Syslog标准配置模板验证Syslog在网络配置。</li> </ol>
<b>任务输出</b>	任务输出是关于前提任务的条件的一个状态报告。如果其中任一项支持的任务视为作为无效，进程总管应该提交请求安排支持进程更新。如果支持进程不可以是更新的，请进行在影响的一评估对此进程。
<b>任务角色</b>	网络工程师技能集

## 创建种子文件

OSPF配置管理进程要求使用种子文件取消需要对于网络发现功能。种子文件记录由OSPF进程管理的这一组路由器和也使用，当一个重要的协调与在组织的变更管理流程。例如，如果新节点被输入网络，他们需要被添加到OSPF种子文件。如果变动做对SNMP团体名字由于安全需求，那些修改在种子文件需要被反射。表下面的分级显示创建的种子文件进程。

<b>P r o c e s s</b>	<b>说明</b>
<b>任务目的</b>	创建将使用初始化OSPF配置管理软件的一个种子文件。种子文件的格式化取决于用于的资源实现OSPF配置管理进程。如果自定义脚本被开发，种子文件的格式是由软件设计定义的。如果使用网络管理系统(NMS)，种子文件的格式是由NMS文档定义的。
<b>任务输入</b>	<ol style="list-style-type: none"> <li>1. 安排种子文件的格局。</li> <li>2. 请使用OSPF设计文档识别以下数据：所有节点的IP地址SNMP团体字符串Telnet和CLI登录帐户和密码</li> <li>3. 日程表和联系人名称对于网络变更管理流程。</li> </ol>
<b>任务输出</b>	OSPF配置管理进程的一个种子文件。
<b>任</b>	• 商业NMS系统

任务资源	<ul style="list-style-type: none"> <li>按客户需要开发的软件系统</li> <li>手动程序—日志到每个网络单元和问题命令行里和记录输出。</li> </ul>
任务角色	<ul style="list-style-type: none"> <li>NMS —网络工程师、NMS管理员和NMS脚本技能集。</li> <li>自定义脚本—网络工程师和NMS脚本技能集。</li> <li>手动程序—网络工程师。</li> </ul>

## 重复任务

重复任务完成与进程的每迭代，并且确定他们的频率并且被修改为了改进性能指标。

## 维护种子文件

种子文件为OSPF配置管理进程的有效应用是重要。所以，必须积极地管理种子文件的当前状态。变成影响种子文件需要内容由OSPF配置管理进程责任人跟踪的网络。

Processes	说明
任务目的	<ol style="list-style-type: none"> <li>通过跟踪保存种子文件的货币，并且与控制网络移动的组织功能的交互作用，添加，更改，并且/或者网络配置修改。</li> <li>保持版本控制和备份的控制种子文件的。</li> </ol>
任务输入	<ol style="list-style-type: none"> <li>从变更管理的信息，例如移动、添加和更改，影响种子文件的内容。</li> <li>从影响种子文件的内容的工程/设计的信息。</li> </ol>
任务输出	<ol style="list-style-type: none"> <li>关于种子文件货币的状况的每周报告。</li> <li>描述种子文件备份的定义和文档位置和恢复程序。</li> </ol>
任务资源	<ul style="list-style-type: none"> <li>商业NMS系统</li> <li>按客户需要开发的软件系统</li> <li>手动程序—日志到每个网络单元和问题命令行里和记录输出。</li> </ul>
任务角色	<ul style="list-style-type: none"> <li>NMS —网络工程师、NMS管理员和NMS脚本技能集。</li> <li>自定义脚本—网络工程师和NMS脚本技能集。</li> <li>手动程序—网络工程师。</li> </ul>

## 执行OSPF扫描

用于的两个步骤执行OSPF扫描是：

1. 收集数据。



## 2. 分析数据。

根据如何使用进程，频率这两个步骤将变化。例如，此进程可以用于验证安装修改。在这种情况下，在更改和数据分析在更改以后执行确定更改前后的成功数据收集运行。

如果此进程用于验证OSPF配置管理设计记录，数据收集和分析频率依靠变化率在网络的。例如，如果有在网络上巨大数量的变化，设计验证每星期一次执行。如果有非常在网络的少量变化，设计验证没有更多比每月一次执行。

## [查看OSPF报告](#)

OSPF配置管理报告的格式依靠用于的资源实现OSPF配置管理进程。下面的表提供建议了按客户需要开发的报告格式。

<b>报告格式</b>	
<b>任务输入</b>	关于OSPF配置管理报告，请参阅在本文内的 <a href="#">Data Presentation部分</a> 。
<b>任务输出</b>	如果问题被找到在扫描报告和逻辑设计记录之间，必须做出决定项目是正确的，并且是不正确的。应该更正不正确项目。这可能包括设计记录或网络更改单的修改。
<b>任务资源</b>	<ul style="list-style-type: none"><li>• 商业NMS系统</li><li>• 按客户需要开发的软件系统</li><li>• 手工—日志到每个网络单元和问题命令行里和记录输出</li></ul>
<b>任务角色</b>	<ul style="list-style-type: none"><li>• NMS —网络工程师、NMS管理员和NMS脚本技能集。</li><li>• 自定义脚本—网络工程师和NMS脚本技能集。</li><li>• 手动程序—网络工程师。</li></ul>

## [数据标识](#)

### [一般数据特性](#)

下面的表描述可以被运用于OSPF配置管理的数据。

数据	说明
OSPF区域	描述的信息路由器的连接的区域包括： <ul style="list-style-type: none"><li>• 区域ID</li><li>• 区域认证</li><li>• SPF运行</li><li>• ABR的编号在区域</li><li>• ASBRs的编号在区域</li><li>• 区域LSA计数—在路由器间的一致性在区域</li></ul>

	<ul style="list-style-type: none"> <li>• 区域LSA校验和—在路由器间的一致性在区域</li> <li>• 频率信息包丢弃由于寻址错误每个区域</li> <li>• 频率路由进程的协议信息包丢弃每个区域</li> <li>• 频率路由信息包丢弃由于没有路由被找到的情况每个区域</li> </ul>
OSPF接口	<p>从OSPF的观点描述一个接口例如：</p> <ul style="list-style-type: none"> <li>• IP地址</li> <li>• 区域ID</li> <li>• 管理状态</li> <li>• OSPF度量分配到接口</li> <li>• OSPF计时器分配到接口</li> <li>• OSPF状态</li> </ul>
OSPF邻居状态	<p>描述一个OSPF邻居。</p> <ul style="list-style-type: none"> <li>• 邻接路由器ID</li> <li>• 相邻状态</li> <li>• 邻接事件—次数邻接关系更改了状态，或者错误出现。</li> <li>• 邻接重传队列—重传队列的当前长度。</li> </ul>

## [SNMP数据标识](#)

Cisco当前支持[RFC 1253 OSPF第2版MIB](#)。RFC 1253不包含OSPF的SNMP陷阱定义。OSPF MIB的新版本是[RFC 1850 OSPF版本2](#)。SNMP陷阱为在RFC 1850的OSPF被定义。OSPF MIB的Cisco的实施不支持RFC 1850。

请参见本文的[SNMP Polling Data部分](#)关于更详细的资料。

请参见MIB支持平台和代码版本的一张明确列表的[Cisco网络管理软件页](#)。

## [RMON数据识别](#)

没有为此程序需要的RMON特定资料。

## [系统日志数据识别](#)

一般来说，Syslog生成另外技术的服务特定的消息。虽然系统日志信息为故障和性能管理是适当，被提供的信息这里是参考。关于Cisco设备生成的OSPF系统日志信息示例，请参阅[OSPF错误信息](#)。

关于由设备的系统消息一张完全列表，请参见[消息和恢复流程](#)。

## [Cisco IOS CLI数据证明](#)

在OSPF配置管理过程中的此版本，没有需要的CLI数据。

# 数据收集

## SNMP数据收集

下面的表定义了SNMP数据收集不同的组件。

SNMP数据组件	定义
一般SNMP配置	参考 <a href="#">配置SNMP</a> 关于SNMP配置最佳实践的一般信息。
有服务特效的SNMP配置	没有对于此程序是必需的服务特定的SNMP配置。
SNMP MIB需求	<a href="#">请参阅上面数据标识部分。</a>
轮询集的SNMP MIB	SNMP轮询数据收集由一个商业系统例如 <a href="#">HP OpenView</a> 或由自定义脚本。 <a href="#">关于收集算法一进一步讨论，请参阅</a> 本文的 <a href="#">Example Data Collection Algorithms</a> 部分。
SNMP MIB陷阱集	在Cisco设备MIB支持的OSPF的当前版本不支持SNMP陷阱。没有对于此程序是必需的SNMP陷阱。

## RMON数据收集

没有在程序的此版本和资料需要的RMON配置。

## 系统日志数据收集

在本文的范围之外，一般Syslog配置指南是。参考[配置和排除Cisco Secure PIX防火墙故障用一一内部网络](#)欲知更多信息。

使用以下命令， OSPF特定需求通过配置OSPF路由器记录与系统消息的邻接更改解决：

```
OSPF_ROUTER(config)# ospf log-adj-changes
```

## Cisco IOS CLI数据数据收集

一般来说， Cisco IOS CLI提供最直接访问给NE包含的原始的信息。然而， CLI访问更好地适合故障检修程序和变更管理活动比对于全局配置管理如被定义由此程序。访问通过CLI不会为一个大型网络的管理扩展。在这些情况下，需要自动化的信息存取。

在OSPF配置管理过程中的此版本，没有需要的CLI配置和资料。

## 数据表示

## OSPF区域报告

下列是OSPF区域报告的一种示例格式。报告的格式取决于商业NMS的功能，如果使用一个，或者自定义脚本的有针对的输出。

区域	数据域	前运行	此运行
区域ID #1	认证		
	SPF运行		
	ABR计数		
	ASBR计数		
	LSA计数		
	LSA校验和		
	地址错误		
	路由丢弃		
	没有找到的路由		
区域ID #n	认证		
	SPF运行		
	ABR计数		
	ASBR计数		
	LSA计数		
	LSA校验和		
	地址错误		
	路由丢弃		
	没有找到的路由		

## OSPF接口报告

下列是OSPF接口报告的一种示例格式。实际上，报告的格式取决于商业NMS的功能，如果使用一个，或者自定义脚本的有针对的输出。

区域	设备	接口	数据域	前运行	此运行
区域ID #1	节点ID #1	接口ID #1	IP Address		
			区域ID		
			管理状态		
			OSPF状态		
			/费用的权值/计时器		
		接口ID #n	IP Address		
			区域ID		
			管理状态		
			OSPF状态		
			/费用的权值/计时器		

	节点 ID #n	接口 ID #1	IP Address		
			区域ID		
			管理状态		
			OSPF状态		
			/费用的权值/计时器		
		接口 ID #n	IP Address		
			区域ID		
			管理状态		
			OSPF状态		
			/费用的权值/计时器		
区域 ID #n	节点 ID #1	接口 ID #1	IP Address		
			区域ID		
			管理状态		
			OSPF状态		
			/费用的权值/计时器		
		接口 ID #n	IP Address		
			区域ID		
			管理状态		
			OSPF状态		
			/费用的权值/计时器		
	节点 ID #n	接口 ID #1	IP Address		
			区域ID		
			管理状态		
			OSPF状态		
			/费用的权值/计时器		
		接口 ID #n	IP Address		
			区域ID		
			管理状态		
			OSPF状态		
			/费用的权值/计时器		

## OSPF相邻报告

下列是OSPF相邻报告的一种示例格式。实际上，报告的格式取决于商业NMS的功能，如果使用一个，或者自定义脚本的有针对的输出。

区域	设备	相邻	数据域	前运行	此运行
区域ID #1	节点ID #1	邻居ID #1	路由器ID		

			路由器IP地址		
			状态		
			<a href="#">事件</a>		
			Retrans Que		
		邻居ID #n	路由器ID		
			路由器IP地址		
			状态		
			<a href="#">事件</a>		
	节点ID #n	邻居ID #1	路由器ID		
			路由器IP地址		
			状态		
			<a href="#">事件</a>		
		邻居ID #n	路由器ID		
			路由器IP地址		
			状态		
			<a href="#">事件</a>		
区域ID #n	节点ID #1	邻居ID #1	路由器ID		
			路由器IP地址		
			状态		
			<a href="#">事件</a>		
		邻居ID #n	路由器ID		
			路由器IP地址		
			状态		
			<a href="#">事件</a>		
	节点ID #n	邻居ID #1	路由器ID		
			路由器IP地址		
			状态		
			<a href="#">事件</a>		
		邻居ID #n	路由器ID		
			路由器IP地址		
			状态		
			<a href="#">事件</a>		

## 商用和公共互联网监控工具

商业工具在集和处理存在帮助系统日志信息和一般SNMP MIB变量集轮询的。

没有支持OSPF配置管理如被定义由此程序的商业或公共互联网监控工具知道。所以，需要本地定制脚本和程序。

## SNMP轮询数据

### 路由表RFC 1213

对象名称	对象说明
ipRouteDest	路由的目的地IP地址。与0.0.0.0的值的一个条目认为默认路由。多个路由到单个目的地在表里能出现，但是对这样多个条目的访问依靠网络管理协议定义的表存取机制在使用中。 : := {iprouteentry 1}对象标识符= 1.3.6.1.2.1.4.21.1.1
ipRouteMask	指示掩码是逻辑的对目的地地址在与在URL域的值比较前。对于不支持任意子网掩码的那些系统，代理程序修建值ipRouteMask通过确定对应的URL域的值是否属于A类， B或者C网络，使用以下掩码网络之一 : • A类= 255.0.0.0 • B类= 255.255.0.0 • C类= 255.255.255.0 如果ipRouteDest的值是0.0.0.0，默认路由，掩码值也是0.0.0.0。 <b>Note:</b> 所有IP路由子系统隐含地使用此机制。 : := {iprouteentry 11}对象标识符= 1.3.6.1.2.1.4.21.1.11
ipRouteNextHop	此路由下一跳的IP地址。一旦路由一定对认识到与广播价质的接口，此字段的值是在接口的代理程序的IP地址。 : := {iprouteentry 7}对象标识符= 1.3.6.1.2.1.4.21.1.7
ipRouteNextHopIndex	独特识别本地接口路由下一跳被到达的指标值。此接口是IfIndex值确定的同一个接口。 : := {iprouteentry 2}对象标识符= 1.3.6.1.2.1.4.21.1.2

### RFC 1213混杂对象

对象名称	对象说明
ipAddressEntry	独特识别接口可适用对条目的指标值。此接口是IfIndex值确定的同一个接口。 : := {ipaddentry 2}对象标识符= 1.3.6.1.2.1.4.20.1.2

tfl nd ex	
ipl nA dd rE rro rs	输入数据包的数量被丢弃的，因为在他们的IP头的IP地址是实体的一个无效的目的地字段。此计数包括无效的地址(0.0.0.0)和不支持的组地址(组E)。对于不是IP网关，并且不转发数据包的实体，计数器包括被丢弃的数据包，因为目的地地址不是一个本地地址。 {ip 5}对象标识符= 1.3.6.1.2.1.4.5
ip R ou tin g Di sc ar ds	被丢弃的有效路由条目的数量。丢弃这样条目的一个可能的来源是释放其他路由条目的缓冲空间。 {ip 23}对象标识符= 1.3.6.1.2.1.4.23
ip O ut N o R ou te s	IP数据包的数量被丢弃的，因为不可能发现路由传达他们给他们的目的地。 {ip 12}对象标识符= 1.3.6.1.2.1.4.12

### RFC 1253 OSPF区域表

对象名称	对象说明
ospfAreaId	独特识别区域的32位整数。区域ID 0.0.0.0使用OSPF骨干网。 : := {ospfareaentry 1}对象标识符= 1.3.6.1.2.1.14.2.1.1
ospfAuthType	为此区域指定的认证类型。另外的认证类型可能分配本地根据一个每区域基本类型。DEFAULT值是0。 : := {ospfareaentry 2}对象标识符= 1.3.6.1.2.1.14.2.1.2
OspfSpfRuns	使用此区域链路状态数据库，区域内路由表被计算的次数。对象标识符= 1.3.6.1.2.1.14.2.1.4
ospfAreaBdrRtrCount	ABR总数可及的在此区域内。这最初0，DEFAULT值和被计算在每张SPF通行证。 : := {ospfareaentry 5}对象标识符= 1.3.6.1.2.1.14.2.1.5
ospfASBdrRtrCount	ABSRs总数可及的在此区域内。这最初是0 (DEFAULT值)和被计算在每张SPF通行证。 : := {ospfareaentry 6}对象标识符= 1.3.6.1.2.1.14.2.1.6
ospfAreaLSACount	LSA总数在一个区域链路状态数据库的，不包括外部LSA。DEFAULT值是0。 : := {ospfareaentry 7}对象标识符=



	1.3.6.1.2.1.14.2.1.7
ospfAreaLSACKsumSum	在区域链路状态数据库包含的LSA的LS校验和的32位未签名的总和。此总和排除外部(LS类型5) LSA。总和可以用于确定是否有在路由器的链路状态数据库上的一个变化和比较两路由器链路状态数据库。DEFAULT值是0。 : := {ospfareaentry 8}对象标识符= 1.3.6.1.2.1.14.2.1.8

### RFC 1253 OSPF接口表

对象名称	对象说明
OspfIfIpAddress	OSPF接口的IP地址。对象标识符= 1.3.6.1.2.1.14.7.1.1
OspfIfEvents	次数OSPF接口更改了其状态，或者错误出现。对象标识符= 1.3.6.1.2.1.14.7.1.15
OspfIfState	OSPF接口状态。对象标识符= 1.3.6.1.2.1.14.7.1.12

### RFC 1253 OSPF相邻表

对象名称	对象说明
OspfNbrIpAddress	此相邻的IP地址。 : := {ospfnbrentry 1}对象标识符= 1.3.6.1.2.1.14.10.1.1
ospfNbrAddressLessIndex	IfIndex的对应的值在互联网标准的MIB的在没有一个IP地址的索引。在行创建，这可以从实例派生。 : := {ospfnbrentry 2}对象标识符= 1.3.6.1.2.1.14.10.1.2
ospfNbrRtrId	32位整数，表示作为IP地址，独特识别在自控系统的相邻路由器。DEFAULT值是0.0.0.0。 : := {ospfnbrentry 3}对象标识符= 1.3.6.1.2.1.14.10.1.3
ospfNbrState	关系的状态与相邻的。状态是： <ul style="list-style-type: none"> <li>• 在(1)下</li> <li>• 尝试(2)</li> <li>• init (3)</li> <li>• 双向(4)</li> <li>• exchangeStart (5)</li> <li>• 交换(6)</li> <li>• 加载(7)</li> <li>• 充分(8)</li> </ul> : := {ospfnbrentry 6}对象标识符= 1.3.6.1.2.1.14.10.1.6
ospfNbrEvents	次数邻接关系更改了状态，或者错误出现。DEFAULT值是0。 : := {ospfnbrentry 7}对象标识符= 1.3.6.1.2.1.14.10.1.7
ospfNbrLSRetransQLen	重传队列的当前长度。DEFAULT值是0。 : := {ospfnbrentry 8}对象标识符= 1.3.6.1.2.1.14.10.1.8

## 示例数据收集算法

在对本文的调查时，还原‘C’程序被开发了。使用有Visual C++版本5.0的， Microsoft Developer Studio 97程序，称为oscan，被写了。有提供SNMP功能Application Programming Interface (API)的两个特定库。那些库是snmpapi.lib和mgmtapi.lib

Microsoft API提供的功能在下表被组队到三个主要类别并且列出。

代理程序功能	管理器功能	实用程序函数
SnmpExtensionInit SnmpExtensionInitEx SnmpExtensionQuery SnmpExtensionTrap	SnmpMgrClose SnmpMgrGetTrap SnmpMgrOidToStr SnmpMgrOpen SnmpMgrRequest SnmpMgrStrToOid SnmpMgrTrapListen	SnmpUtilMemAlloc SnmpUtilMemFree SnmpUtilMemReAlloc SnmpUtilOidAppend SnmpUtilOidCmp SnmpUtilOidCpy SnmpUtilOidFree SnmpUtilOidNCmp SnmpUtilPrintAsnAny SnmpUtilVarBindCpy SnmpUtilVarBindListCpy SnmpUtilVarBindFree SnmpUtilVarBindListFree

oscan还原代码封装与如下所示的一套的Microsoft API另外的功能。

- snmpWalkStrOid
- snmpWalkAsnOid
- snmpWalkVarBind
- snmpWalkVarBindList

这些功能提供对多种SNMP MIB表的允许使用维护OSPF配置数据的通用的API。能将被获取的表的Object Identifier (OID)通过对oscan API与特定呼叫作用的表一起。呼叫在从表返回的数据上作用有智力对起作用。

### 主例程

首要任务创建将是OSCAN程序的目标节点的列表。为了避免“设备发现”问题，要求种子文件识别将被扫描的节点。种子文件提供信息例如IP地址和SNMP只读属性字段。

OSCAN程序需要维护几个内部数据结构存储从路由器收集的SNMP信息。一般来说，有收集的每张SNMP MIB表的一个内部数据结构。

```
Main
load node array based on information in the seed file.
while more entries in the node array
start SNMP session for this node
collect IP route table for this node
collect OSPF area table for this node
collect OSPF Neighbor table for this node
collect sysName for this node
collect OSPF Interface table for this node
end SNMP session for this node
end while
```

## IP路由表

必须保重，当访问与SNMP时的IP路由表，因为是简单的在此操作时超载路由器CPU。所以，OSCAN程序使用一个用户可配置延迟参数。参数提供每个SNMP请求之间的延迟。对于大环境，这意味着总时间收集信息可以是非常重大的。

路由表包含对感兴趣oscan是的四个信息：

- ipRouteDest
- ipRouteMask
- ipRouteNextHop
- ipRouteIfIndex

ipRouteDest标注路由表。所以，从SNMP获取请求返回的每个对象有ipRouteDest被添附对OID。

对象ipRouteIfIndex是整数该索引到IP地址表(ipAddrTable)。ipAddrTable被标注使用ipAdEntAddr对象(接口的IP地址)。为了获得接口的IP地址，需要一个四步进程：

1. 从路由表收集ipRouteIfIndex。
2. 使用模式匹配的，ipRouteIfIndex访问ipAddrTable。
3. 当找到时模式，请转换OID成字符串并且收集将是接口的IP地址的最后四个点分的十进制字段。
4. 存储接口的IP地址回到IP路由表。

访问的IP路由表一般算法如下所示。这时，存储ipRouteIfIndex的仅整数值。以后在进程，当收集接口信息，ipAddrTable被获取时和剩余信息中收集并且被放置到内部IP路由表。

```
OID List =  
ipRouteDestOID,  
ipRouteMaskOID,  
ipRouteNextHopOID,  
ipRouteIfIndexOID;
```

```
For each object returned by SNMP route table walk  
Sleep // user configurable polling delay.  
check varbind oid against OID list  
if OID is ipRouteDestOID  
add new entry in the internal route table array  
if OID is one of the others  
search internal route array for matching index value  
store information in array
```

收集的信息在类似于下面路由器CLI的熟悉输出的表里描述。

```
ROUTE TABLE  
*****  
Destination      Mask                GW                  Interface  
10.10.10.4        255.255.255.252    10.10.10.5         10.10.10.5  
10.10.10.16       255.255.255.252    10.10.10.6         10.10.10.5  
10.10.10.24       255.255.255.252    10.10.10.25        10.10.10.25  
10.10.10.28       255.255.255.252    10.10.11.2         10.10.11.1  
10.10.10.36       255.255.255.252    10.10.10.6         10.10.10.5  
10.10.11.0        255.255.255.0      10.10.11.1         10.10.11.1  
10.10.13.0        255.255.255.0      10.10.11.2         10.10.11.1
```

## OSPF区域表

从OSPF区域表的信息收集由扫描OSPF区域表(ospfAreaTable)和处理数据完成，当返回。ospfAreaTable的索引是ospfAreaId。ospfAreaId在与IP地址是相同的点分十进制格式存储。所以，使用处理和搜索ipRouteTable和ipRouteIndex的同样子例行程序可以重新使用这里。

有实际上不在OSPF区域表里在此部分包括的几数据项。例如，ipInAddrErrors，IpRoutingDiscards，并且ipOutNoRoute对象在MIB-2定义，但是没有与OSPF区域产生关联。这些对象与路由器产生关联。所以，这些计数器使用作为区域权值通过添加值每个节点在区域到区域计数器。例如，在OSPF区域报告，编号的信息包被丢弃的由于没有找到的路由实际上是所有丢弃的信息包的总和和路由器在该区域。这是提供区域的路由健康全视图的高级权值。

```
OID List =
ipInAddrErrorsOID,
ipRoutingDiscardsOID,
ipOutNoRouteOID,
areaIdOID,
authTypeOID,
spfRunsOID,
abrCountOID,
asbrCountOID,
lsaCountOID,
lsaChecksumOID;
```

```
For object returned from the SNMP walk of the Area Table
Sleep // user configurable polling delay.
check varbind oid against OID list.
if OID is ospfAreaId
add new entry in the internal route table array
if OID one of the others
search internal array for matching index value
store information in array
end of for loop
get ipInAddrErrors, ipRoutingDiscards, ipOutNoRoute
add values to overall Area counters
```

收集的信息在下表ASCII表描述。

```
AREAS
*****
AREA = 0.0.0.0AREA = 0.0.0.2
authType = 0authType = 0
spfRuns = 38spfRuns = 18
abrCount = 2abrCount = 1
asbrCount = 0asbrCount = 0
lsaCount = 11lsaCount = 7
lsaChecksumSum = 340985lsaChecksumSum = 319204
ipInAddrErrors = 0 ipInAddrErrors = 0
ipRoutingDiscards = 0ipRoutingDiscards = 0
ipOutNoRoutes = 0ipOutNoRoutes = 0
```

## [OSPF相邻表](#)

邻接表的索引是两值：

- ospfNbrIpAddr — ospfNbrIpAddr是相邻的IP地址。
- ospfNbrAddressLessIndex — ospfNbrAddressLessIndex可以是两值之一：对于有分配的一个IP地址的接口，它是零。对于没有分配的一个IP地址的接口，它被解释作为从互联网标准的

## MIB的IfIndex。

由于有索引的两值，您需要对返回的OIDs调整用于额外信息的前算法被添附。在做此调节以后，使用处理和搜索ipRouteTable和ipRouteIfIndex的同样子例行程序可以重新使用这里。

```
OID List =
ospfNbrIpAddrOID,
ospfNbrAddressLessIndexOID,
ospfNbrRtrIdOID,
ospfNbrStateOID,
ospfNbrEventsOID,
ospfNbrLSRetransQLenOID,
```

```
For object returned from the SNMP walk of the Neighbor Table
Sleep // user configurable polling delay.
check varbind OID against OID list.
if OID matches ospfNbrIpAddr
add new entry in the internal neighbor table array
if OID matches one of the others
search array for matching index value
store information in array
```

收集的信息在下表ASCII表描述。

```
NEIGHBORS
*****
NEIGHBOR #ONEIGHBOR #1
Nbr Ip Addr = 10.10.10.6Nbr Ip Addr = 10.10.11.2
Nbr Rtr Id = 10.10.10.17Nbr Rtr Id = 10.10.10.29
Nbr State = 8Nbr State = 8
Nbr Events = 6Nbr Events = 30
Nbr Retrans = 0Nbr Retrans = 0
```

## [Related Information](#)

- [OSPF配置指南](#)
- [与OSPF协议的RFC 1246经验](#)
- [RFC 1245 OSPF协议分析](#)
- [管理异步地生成的戒备的RFC 1224技术](#)
- [OSPF 支持页](#)
- [IP 路由支持页](#)
- [Technical Support - Cisco Systems](#)