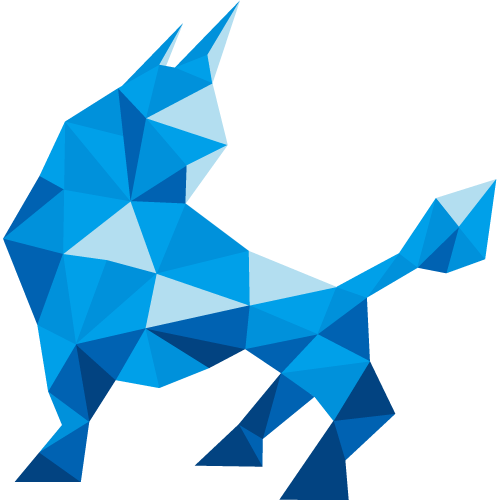
**麒麟负载均衡系统**

通讯接口定义



**湖南麒麟信息工程技术有限公司**

**文档修订记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 修订内容 | 修订人 | 修订日期 | 备注 |
| V1.0 | 初稿 | 文云川 | 2010-08-19 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[一、 基本定义 1](#_Toc283820501)

[协议 1](#_Toc283820502)

[参数类型 1](#_Toc283820503)

[复合类型定义 1](#_Toc283820504)

[函数调用 2](#_Toc283820505)

[通用函数定义 2](#_Toc283820506)

[二、 物理接口配置 2](#_Toc283820507)

[设置IP地址 2](#_Toc283820508)

[开启/关闭 3](#_Toc283820509)

[设置MAC地址 3](#_Toc283820510)

[设置MTU值 3](#_Toc283820511)

[设置ARP模式 3](#_Toc283820512)

[设置DHCP 4](#_Toc283820513)

[设置工作模式 4](#_Toc283820514)

[修改描述 4](#_Toc283820515)

[查看指定接口 5](#_Toc283820516)

[查看全部接口 5](#_Toc283820517)

[三、 汇聚接口配置 6](#_Toc283820518)

[设置IP地址 6](#_Toc283820519)

[开启/关闭 6](#_Toc283820520)

[设置MAC地址 7](#_Toc283820521)

[设置MTU值 7](#_Toc283820522)

[设置ARP模式 7](#_Toc283820523)

[设置DHCP 8](#_Toc283820524)

[设置汇聚模式 8](#_Toc283820525)

[设置链路检查 8](#_Toc283820526)

[设置从属接口 9](#_Toc283820527)

[添加汇聚接口 9](#_Toc283820528)

[删除汇聚接口 9](#_Toc283820529)

[修改描述 10](#_Toc283820530)

[查看指定接口 10](#_Toc283820531)

[查看全部接口 11](#_Toc283820532)

[四、 静态路由配置 11](#_Toc283820533)

[添加 11](#_Toc283820534)

[删除 12](#_Toc283820535)

[修改下一跳 12](#_Toc283820536)

[获取路由表 13](#_Toc283820537)

[获取路由表长度 14](#_Toc283820538)

[修改描述 14](#_Toc283820539)

[五、 策略路由配置 14](#_Toc283820540)

[添加 14](#_Toc283820541)

[插入 15](#_Toc283820542)

[删除 16](#_Toc283820543)

[修改下一跳 16](#_Toc283820544)

[添加规则 16](#_Toc283820545)

[删除规则 17](#_Toc283820546)

[获取路由表 17](#_Toc283820547)

[获取路由表长度 19](#_Toc283820548)

[修改描述 19](#_Toc283820549)

[六、 智能路由配置 19](#_Toc283820550)

[添加 19](#_Toc283820551)

[删除 20](#_Toc283820552)

[修改下一跳 20](#_Toc283820553)

[获取路由表 21](#_Toc283820554)

[获取路由表长度 21](#_Toc283820555)

[修改描述 22](#_Toc283820556)

[七、 SNAT配置 22](#_Toc283820557)

[添加 22](#_Toc283820558)

[插入 23](#_Toc283820559)

[替换 24](#_Toc283820560)

[删除 25](#_Toc283820561)

[开启/关闭 25](#_Toc283820562)

[修改描述 25](#_Toc283820563)

[获取列表 25](#_Toc283820564)

[获取列表长度 27](#_Toc283820565)

[八、 DNAT配置 27](#_Toc283820566)

[添加 27](#_Toc283820567)

[插入 28](#_Toc283820568)

[替换 29](#_Toc283820569)

[删除 29](#_Toc283820570)

[开启/关闭 30](#_Toc283820571)

[修改描述 30](#_Toc283820572)

[获取列表 30](#_Toc283820573)

[获取列表长度 31](#_Toc283820574)

[九、 ARP配置 32](#_Toc283820575)

[获取静态ARP表项 32](#_Toc283820576)

[获取动态ARP表项 32](#_Toc283820577)

[设置静态ARP表项 33](#_Toc283820578)

[删除ARP表项 33](#_Toc283820579)

[十、 ADSL配置 33](#_Toc283820580)

[获取全部接口 33](#_Toc283820581)

[获取指定接口 34](#_Toc283820582)

[添加 35](#_Toc283820583)

[删除 35](#_Toc283820584)

[设置 35](#_Toc283820585)

[拨号 36](#_Toc283820586)

[断开连接 36](#_Toc283820587)

[修改描述 36](#_Toc283820588)

[十一、 虚拟服务配置 36](#_Toc283820589)

[添加服务 36](#_Toc283820590)

[删除服务 37](#_Toc283820591)

[设置服务地址 37](#_Toc283820592)

[设置服务特征 38](#_Toc283820593)

[流量控制 39](#_Toc283820594)

[开启/关闭虚拟服务 39](#_Toc283820595)

[高可用配置 40](#_Toc283820596)

[修改描述 40](#_Toc283820597)

[名称转换为ID 41](#_Toc283820598)

[添加真实服务器 41](#_Toc283820599)

[删除真实服务器 42](#_Toc283820600)

[修改真实服务器 42](#_Toc283820601)

[开启/关闭真实服务器 43](#_Toc283820602)

[获取虚拟服务个数 43](#_Toc283820603)

[获取虚拟服务列表 43](#_Toc283820604)

[获取虚拟服务数据统计 45](#_Toc283820605)

[获取真实服务器数据统计 46](#_Toc283820606)

[十二、 高可用配置 46](#_Toc283820607)

[获取配置 46](#_Toc283820608)

[设置 47](#_Toc283820609)

[开启/关闭高可用 47](#_Toc283820610)

[十三、 智能DNS配置 48](#_Toc283820611)

[设置BIND 48](#_Toc283820612)

[开启/关闭BIND 48](#_Toc283820613)

[添加DNS策略 48](#_Toc283820614)

[删除DNS策略 49](#_Toc283820615)

[修改DNS策略 49](#_Toc283820616)

[获取BIND基本信息 50](#_Toc283820617)

[获取DNS策略信息 50](#_Toc283820618)

[获取BIND全部信息 51](#_Toc283820619)

[查询DNS策略ID 51](#_Toc283820620)

[十四、 ISP配置 52](#_Toc283820621)

[获取ISP数据列表 52](#_Toc283820622)

[获取ISP名字列表 52](#_Toc283820623)

[获取ISP信息 52](#_Toc283820624)

[获取ISP名字 53](#_Toc283820625)

[添加ISP 53](#_Toc283820626)

[删除ISP 53](#_Toc283820627)

[修改ISP 54](#_Toc283820628)

[十五、 黑名单设置 54](#_Toc283820629)

[添加规则 54](#_Toc283820630)

[删除规则 54](#_Toc283820631)

[修改规则 55](#_Toc283820632)

[获取规则列表 55](#_Toc283820633)

[获取规则列表长度 56](#_Toc283820634)

[修改规则描述 56](#_Toc283820635)

[十六、 其它配置 56](#_Toc283820636)

[域名服务器 56](#_Toc283820637)

[主机名称 57](#_Toc283820638)

[DOS攻击防护 57](#_Toc283820639)

[服务器探测 57](#_Toc283820640)

[系统状态 57](#_Toc283820641)

[系统时间 58](#_Toc283820642)

[修改登录密码 59](#_Toc283820643)

[配置文件 59](#_Toc283820644)

[设置日志服务器 59](#_Toc283820645)

[更新 60](#_Toc283820646)

[Licence管理 60](#_Toc283820647)

# 基本定义

## 协议

klbmanager程序对外提供XML-RPC通讯接口，每个调用都被映射为一个XML-RPC函数。

XML-RPC服务端口8888，URL的请求目标为RPC2。

## 参数类型

XML-RPC支持多种数据类型，在本程序中所使用的类型包括：

字符串类型：String

整数类型：Integer

布尔类型：Boolean

数组类型：Array

结构类型：Struct

二进制类型：Binary

后文将使用以上英文单词来定义所使用的数据类型。

## 复合类型定义

因Struct与Array类型都是复合类型，所以其定义采用以下方式进行描述：

Struct VirtualService //定义VirtualService结构

{

Integer ID //ID字段

Boolean Enabled //Enabled字段

String Description //Description字段

Array IP String //IP字段，数组元素为字符串类型

Array Servers Struct //Servers字段，数组元素为结构类型

{

String IP //IP字段

String Name //Name字段

}

Struct Monitor //Monitor字段，定义为结构类型

{

Integer Interval //Interval字段

Integer Timeout //Timeout字段

}

}

除非有特殊说明，否则Array的每个元素类型皆相同。

## 函数调用

每个XML-RPC函数调用都需要提供函数名以及一个数组类型的参数，用来传递可选的字符串密码以及可选的自定义操作参数。本节中称此数组类型的参数为Param。

调用完成后，返回一个自定义的结果。若无特别说明，则认为是返回Boolean值true。

调用失败，则会返回一个XML-RPC异常，包含一个错误码以及说明字符串。

可供调用的XML-RPC函数按所需参数可分为如下四类：

无密码、无操作参数：

Param为空。

有密码、无操作参数：

Param第一个元素为密码明文。

无密码、有操作参数：

Param第一个元素为操作参数。

有密码、有操作参数：

Param第一个元素为密码明文，第二个元素为操作参数。

## 通用函数定义

在本文之后的内容中，对XML-RPC函数的定义将包括：

a、函数名称

b、操作参数（之后将简称为参数）

c、是否使用密码

完成定义后，按照前一节函数调用的方式填充XML-RPC调用参数。

一般情况下，各个函数所定义的操作参数皆为结构类型。而其中的部分字段可以定义为可选项，即是说在构造XML-RPC结构类型的参数时，不填充此字段。有必要的话，程序将按其对应的默认值处理。

# 物理接口配置

## 设置IP地址

函数名：Ethernet.IP

密码：需要

参数：

Struct Ethernet

{

String Dev //目标接口

Array IP String //目标IP列表

}

详解：

IP字段：目标IP列表，每个元素指定一个IP及其掩码，使用/进行分隔。例如192.168.1.233/16

## 开启/关闭

函数名：Ethernet.Enabled

密码：需要

参数：

Struct Ethernet

{

String Dev //目标接口

Boolean Enabled //true为开启接口，false为关闭接口

}

## 设置MAC地址

函数名：Ethernet.Address

密码：需要

参数：

Struct Ethernet

{

String Dev //目标接口

String Address //指定的MAC地址

}

详解：

Address字段：所期望设定的MAC地址。若其值为空，则表示不指定特定地址，使用当前值或是默认值。格式为XX：XX：XX：XX：XX：XX。

## 设置MTU值

函数名：Ethernet.MTU

密码：需要

参数：

Struct Ethernet

{

String Dev //目标接口

Integer MTU //指定的MTU值，范围[68，1500]

}

## 设置ARP模式

函数名：Ethernet.Arp

密码：需要

参数：

Struct Ethernet

{

String Dev //目标接口

Integer Arp //指定的Arp模式代码

}

详解：

Arp字段：

0：开启

1：关闭

2：仅响应

3：代理模式

## 设置DHCP

函数名：Ethernet.Dhcp

密码：需要

参数：

Struct Ethernet

{

String Dev //目标接口

Boolean Dhcp //true为开启Dhcp，false为关闭Dhcp

}

## 设置工作模式

函数名：Ethernet.Link

密码：需要

参数：

Struct Ethernet

{

String Dev //目标接口

Boolean Detect //true为自动探测，false为手工设定

Boolean FullDuplex //true为全双工，false为半双工

Integer Speed //接口速率

}

详解：

Detect、FullDuplex、Speed字段皆为可选字段，但至少要填充一个。

Speed的取值范围为10、100、1000对应10Mbps、100Mbps、1000Mbps。

## 修改描述

函数名：Ethernet.Description

密码：需要

参数：

Struct Ethernet

{

String Dev //目标接口

String Description //指定的描述信息

}

## 查看指定接口

函数名：Ethernet.Get

密码：需要

参数：

Struct Ethernet

{

String Dev //目标接口

}

返回结果：

Struct Ethernet

{

String Dev //目标接口

String Description //描述信息

String Adsl //属于特定的ADSL接口

String Master //属于特定的汇聚接口

String Address //参考设置MAC地址一节

String CurrentAddress //当前的实际MAC地址

String RealAddress //启动时的MAC地址，理解为固有MAC。

Boolean Detect //参考设置工作模式一节

Boolean FullDuplex //参考设置工作模式一节

Integer Speed //参考设置工作模式一节

Integer MTU //参考设置MTU值一节

Array IP String //参考设置IP地址一节

Boolean Enabled //参考开启/关闭一节

Integer Arp //参考设置ARP模式一节

Boolean Dhcp //参考设置DHCP一节

Boolean Carrier //true为插入网线，false为拔出网线

}

## 查看全部接口

函数名：Ethernet.GetAll

密码：需要

参数：无

返回结果：

Array Result Struct

{

String Dev //接口名称

String Description //描述信息

String Adsl //属于特定的ADSL接口

String Master //属于特定的汇聚接口

String Address //参考设置MAC地址一节

String CurrentAddress //当前的实际MAC地址

String RealAddress //启动时的MAC地址，理解为固有MAC。

Boolean Detect //参考设置工作模式一节

Boolean FullDuplex //参考设置工作模式一节

Integer Speed //参考设置工作模式一节

Integer MTU //参考设置MTU值一节

Array IP String //参考设置IP地址一节

Boolean Enabled //参考开启/关闭一节

Integer Arp //参考设置ARP模式一节

Boolean Dhcp //参考设置DHCP一节

Boolean Carrier //true为插入网线，false为拔出网线

}

# 汇聚接口配置

## 设置IP地址

函数名：Bonding.IP

密码：需要

参数：

Struct Bonding

{

String Dev //目标接口

Array IP String //目标IP列表

}

详解：

IP字段：目标IP列表，每个元素指定一个IP及其掩码，使用/进行分隔。例如192.168.1.233/16

## 开启/关闭

函数名：Bonding.Enabled

密码：需要

参数：

Struct Bonding

{

String Dev //目标接口

Boolean Enabled //true为开启接口，false为关闭接口

}

## 设置MAC地址

函数名：Bonding.Address

密码：需要

参数：

Struct Bonding

{

String Dev //目标接口

String Address //指定的MAC地址

}

详解：

Address字段：所期望设定的MAC地址。若其值为空，则表示不指定特定地址，使用当前值或是默认值。格式为XX：XX：XX：XX：XX：XX。

## 设置MTU值

函数名：Bonding.MTU

密码：需要

参数：

Struct Bonding

{

String Dev //目标接口

Integer MTU //指定的MTU值，范围[68，1500]

}

## 设置ARP模式

函数名：Bonding.Arp

密码：需要

参数：

Struct Bonding

{

String Dev //目标接口

Integer Arp //指定的Arp模式代码

}

详解：

Arp字段：

0：开启

1：关闭

2：仅响应

3：代理模式

## 设置DHCP

函数名：Bonding.Dhcp

密码：需要

参数：

Struct Bonding

{

String Dev //目标接口

Boolean Dhcp //true为开启Dhcp，false为关闭Dhcp

}

## 设置汇聚模式

函数名：Bonding.Mode

密码：需要

参数：

Struct Bondnig

{

String Dev //目标接口

Integer Mode //指定模式

}

详解：

MODE字段：指定的汇聚模式，可选值为：

0：轮询模式。

1：主备模式。

2：IP地址异或模式。

3：广播模式。

4：支持802.3ad模式。

5：tlb平衡模式。

6：alb平衡模式。

## 设置链路检查

a、设置检查模式

函数名：Bonding.Check

密码：需要

参数：

Struct Bonding

{

String Dev //目标接口

Integer CheckMode //指定的链路检查模式

Integer Frequency //检测频率，正整数

}

详解：

CheckMode字段与Frequency为可选字段，但至少选择其中之一。

CheckMode字段的可选值为：

0：miimon检查模式

1：arp检查模式

b、arp模式检查IP地址

函数名：Bonding.CheckIP

密码：需要

参数：

Struct Bonding

{

String Dev //目标接口

Array CheckIP String //检查的IP地址列表

}

详解：

CheckIP字段为数组类型，每个元素以字符串格式保存一个需要检测的IP地址，格式为xxx.xxx.xxx.xxx。

## 设置从属接口

函数名：Bonding.SetSlaves

密码：需要

参数：

Struct Bonding

{

String Dev //目标接口

Array Slaves String //指定物理接口列表

}

## 添加汇聚接口

函数名：Bonding.Add

密码：需要

参数：

Struct Bonding

{

String Dev //期望添加的汇聚接口名称

}

详解：

Dev字段：格式为bondX（X为数字）

## 删除汇聚接口

函数名：Bonding.Delete

密码：需要

参数：

Struct Bonding

{

String Dev //待删除的汇聚接口名称

}

## 修改描述

函数名：Bonding.Description

密码：需要

参数：

Struct Bonding

{

String Dev //目标接口

String Description //指定的描述信息

}

## 查看指定接口

函数名：Bonding.Get

密码：需要

参数：

Struct Bonding

{

String Dev //目标接口

}

返回值：

Struct Bonding

{

String Dev //目标接口

Array Slaves String //参考设置从属接口一节

Integer Mode //参考设置汇聚模式一节

Integer CheckMode //参考设置链路检查一节

Integer Frequency //参考设置链路检查一节

Array CheckIP String //参考设置链路检查一节

String Description //描述信息

Array IP String //参考设置IP地址一节

Boolean Enabled //参考开启/关闭一节

Integer MTU //参考设置MTU值一节

String Address //参考设置MAC地址一节

String CurrentAddress //当前的实际MAC地址

Boolean Dhcp //参考设置DHCP一节

Integer Arp //参考设置ARP模式一节

Boolean Carrier //true为插入网线，false为拔出网线

}

## 查看全部接口

函数名：Bonding.GetAll

密码：需要

参数：无

返回值：

Array Result Struct

{

String Dev //接口名称

Array Slaves String //参考设置从属接口一节

Integer Mode //参考设置汇聚模式一节

Integer CheckMode //参考设置链路检查一节

Integer Frequency //参考设置链路检查一节

Array CheckIP String //参考设置链路检查一节

String Description //描述信息

Array IP String //参考设置IP地址一节

Boolean Enabled //参考开启/关闭一节

Integer MTU //参考设置MTU值一节

String Address //参考设置MAC地址一节

String CurrentAddress //当前的实际MAC地址

Boolean Dhcp //参考设置DHCP一节

Integer Arp //参考设置ARP模式一节

Boolean Carrier //true为插入网线，false为拔出网线

}

# 静态路由配置

## 添加

函数名：StaticRoute.Add

密码：需要

参数：

Struct Route

{

String Net //目标网络

Integer Metric //度量值，非负整数

String Description //描述信息

Integer GatePolicy //选择下一跳的算法

Array Gates Struct //描述下一跳的列表

{

String IP //网关地址

String Dev //下一跳的输出接口

Boolean Auto //是否自动获取网关

Integer Weight //权值，范围为[1，256]

}

}

详解：

Metric字段：可选项，不填视之为0。

Description字段：可选项。

GatePolicy字段：可选项，不填视之为0，可选值为：

0：rr算法

1：drr算法

2：random算法

3：wrandom算法

Gates字段：至少有一个元素，其元素的结构：

IP字段：可选项，格式为xxx.xxx.xxx.xxx。

Auto字段：可选项，若为true，则忽略IP字段。

Weight字段：可选项。

Dev字段：一般情况下必填。当Auto字段为false且填充IP字段的情况下，若Dev字段不填，则系统会尝试计算出相应的Dev。

## 删除

函数名：StaticRoute.Delete

密码：需要

参数：

Struct Route

{

Integer ID //待删除路由的ID

String Net //待删除的目标网络

Integer Metric //度量值，非负整数

}

详解：

三个字段皆是可选项，但至少ID、NET至少要填充一个。

ID字段的优先级高于Net、Metric字段。

通过Net、Metric来匹配当前路由表，确定相应的ID值。

## 修改下一跳

函数名：StaticRoute.Set

密码：需要

参数：

Struct Route

{

Integer ID //待修改路由的ID

String Net //待修改的目标网络

Integer Metric //度量值，非负整数

Integer GatePolicy //选择下一跳的算法

Array Gates Struct //描述下一跳的列表

{

String IP //网关地址

String Dev //下一跳的输出接口

Boolean Auto //是否自动获取网关

Integer Weight //权值，范围为[1，256]

}

}

详解：

参数字段可参考添加、删除两节中的同名字段。

## 获取路由表

函数名：StaticRoute.Get

密码：需要

参数：

Struct GetList

{

Boolean All //是否获取全部

Integer Start //获取的起始ID

Integer Count //希望获取的数量

}

返回值：

Array Result Struct

{

Integer ID //规则ID

String Net //目标网络

Integer Metric //度量值

String Description //描述信息

Integer GatePolicy //选择下一跳的算法

Boolean Status //路由的有效性

Array Gates Struct //描述下一跳的列表

{

String IP //网关地址

String Dev //下一跳的输出接口

Boolean Auto //是否自动获取网关

Integer Weight //权值，范围为[1，256]

Boolean Status //下一跳的有效性

}

}

详解：

参数的三个字段皆为可选项，但至少要填充一项。

All字段：不填则默认值为false

Start字段：不填则默认值为0

Count字段：不填则默认值为0

若All为true，则请求全部的路由规则，忽略Start与Count字段。

否则，请求从Start指定的ID号开始获取路由规则。

若Count不为0，则取Count条路由规则，否则，获取至最后一条路由规则。

返回值各个字段的意义可参考添加一节的同名字段。

ID字段：指定路由规则的序号，从0起编。

Status字段描述规则或下一跳的有效性，true为有效，false为无效。

## 获取路由表长度

函数名：StaticRoute.GetCount

密码：需要

参数：无

返回值：

Integer Result

详解：

Result为整个静态路由表条目的数量。

## 修改描述

函数名：StaticRoute.Description

密码：需要

参数：

Struct Route

{

Integer ID //指定的路由规则ID

String Description //描述信息

}

# 策略路由配置

## 添加

函数名：PolicyRoute.Add

密码：需要

参数：

Struct Route

{

String Description //描述信息

Integer GatePolicy //选择下一跳的算法

Array Gates Struct //描述下一跳的列表

{

String IP //网关地址

String Dev //下一跳的输出接口

Boolean Auto //是否自动获取网关

Integer Weight //权值，范围为[1，256]

}

}

详解：

Description字段：可选项。

GatePolicy字段：可选项，不填视之为0，可选值为：

0：rr算法

1：drr算法

2：random算法

3：wrandom算法

Gates字段：至少有一个元素，其元素的结构：

IP字段：可选项，格式为xxx.xxx.xxx.xxx。

Auto字段：可选项，若为true，则忽略IP字段。

Weight字段：可选项。

Dev字段：一般情况下必填。当Auto字段为false且填充IP字段的情况下，若Dev字段不填，则系统会尝试计算出相应的Dev。

## 插入

函数名：PolicyRoute.Insert

密码：需要

参数：

Struct Route

{

Integer ID //要插入的位置

String Description //描述信息

Integer GatePolicy //选择下一跳的算法

Array Gates Struct //描述下一跳的列表

{

String IP //网关地址

String Dev //下一跳的输出接口

Boolean Auto //是否自动获取网关

Integer Weight //权值，范围为[1，256]

}

}

详解：

ID字段：新规则要插入的位置，从0至当前规则数，可以理解为插入到当前同ID的规则之前。

其余各个字段可参考添加一节中的同名字段。

## 删除

函数名：PolicyRoute.Delete

密码：需要

参数：

Struct Route

{

Integer ID //待删除路由的ID

}

## 修改下一跳

函数名：PolicyRoute.Set

密码：需要

参数：

Struct Route

{

Integer ID //待修改路由的ID

Integer GatePolicy //选择下一跳的算法

Array Gates Struct //描述下一跳的列表

{

String IP //网关地址

String Dev //下一跳的输出接口

Boolean Auto //是否自动获取网关

Integer Weight //权值，范围为[1，256]

}

}

详解：

各个字段可参考添加一节中的同名字段。

## 添加规则

函数名：PolicyRoute.AddRule

密码：需要

参数：

Struct Route

{

Integer ID //待操作的路由ID

Array Rules Struct //定义规则列表

{

String SrcNet //匹配的来源网络

String DestNet //匹配的目标网络

Integer Protocol //匹配的通讯协议编号

String ProtocolStr //匹配的通讯协议名称

Integer SrcPort //匹配的源端口号

String SrcPortStr //匹配的源端口名称

Integer DestPort //匹配的目标端口号

String DestPortStr //匹配的目标端口名称

}

}

详解：

除ID字段外，其余各项皆为可选项，若类型为String，则默认值为空，若类型为Integer，则默认值为0。

规则中的String类型的字段若为空或Integer类型的字段为0，表示不匹配此项。

若ProtocolStr、SrcPortStr、DestPortStr不为空，则忽略对应的Protocol、SrcPort、DestPort项。

若ProtocolStr不为tcp、udp或者Protocol不为6、17，则不应填充SrcPort、SrcPortStr、DestPort、DestPortStr字段。

SrcNet、DestNet字段描述网络地址，格式为xxx.xxx.xxx.xxx/xx，例如：172.19.0.0/16。

## 删除规则

函数名：PolicyRoute.DeleteRule

密码：需要

参数：

Struct Route

{

Integer ID //待操作的路由ID

Array Rules Struct

{

Integer ID //待删除的规则ID

}

}

## 获取路由表

函数名：PolicyRoute.Get

密码：需要

参数：

Struct GetList

{

Boolean All //是否获取全部规则

Integer Start //获取的起始ID

Integer Count //希望获取的数量

}

返回值：

Array Result Struct

{

Integer ID //规则ID

String Description //描述信息

Integer GatePolicy //选择下一跳的算法

Boolean Status //路由的有效性

Array Gates Struct //描述下一跳的列表

{

String IP //网关地址

String Dev //下一跳的输出接口

Boolean Auto //是否自动获取网关

Integer Weight //权值，范围为[1，256]

Boolean Status //下一跳的有效性

}

Array Rules Struct

{

Integer ID //规则的ID

String SrcNet //匹配的来源网络

String DestNet //匹配的目标网络

Integer Protocol //匹配的通讯协议编号

String ProtocolStr //匹配的通讯协议名称

Integer SrcPort //匹配的源端口号

String SrcPortStr //匹配的源端口名称

Integer DestPort //匹配的目标端口号

String DestPortStr //匹配的目标端口名称

}

}

详解：

参数的三个字段皆为可选项，但至少要填充一项。

All字段：不填则默认值为false

Start字段：不填则默认值为0

Count字段：不填则默认值为0

若All为true，则请求全部的路由规则，忽略Start与Count字段。

否则，请求从Start指定的ID号开始获取路由规则。

若Count不为0，则取Count条路由规则，否则，获取至最后一条路由规则。

返回值各个字段的含义可参考添加、添加规则两节的同名字段。

ID字段：指定路由规则的序号，从0起编。

Status字段描述规则或下一跳的有效性，true为有效，false为无效。

## 获取路由表长度

函数名：PolicyRoute.GetCount

密码：需要

参数：无

返回值：

Integer Result

详解：

Result为整个策略路由表条目的数量。

## 修改描述

函数名：PolicyRoute.Description

密码：需要

参数：

Struct Route

{

Integer ID //指定的路由规则ID

String Description //描述信息

}

# 智能路由配置

## 添加

函数名：SmartRoute.Add

密码：需要

参数：

Struct Route

{

String Description //描述信息

Integer ISP //指定ISP的ID，0表示不指定

Integer GatePolicy //选择下一跳的算法

Array Gates Struct //描述下一跳的列表

{

String IP //网关地址

String Dev //下一跳的输出接口

Boolean Auto //是否自动获取网关

Integer Weight //权值，范围为[1，256]

}

}

详解：

Description字段：可选项。

ISP字段：指定此路由对应ISP编号；若为0，则表示不指定ISP；参见ISP配置节。

GatePolicy字段：可选项，不填视之为0，可选值为：

0：rr算法

1：drr算法

2：random算法

3：wrandom算法

Gates字段：至少有一个元素，其元素的结构：

IP字段：可选项，格式为xxx.xxx.xxx.xxx。

Auto字段：可选项，若为true，则忽略IP字段。

Weight字段：可选项。

Dev字段：一般情况下必填。当Auto字段为false且填充IP字段的情况下，若Dev字段不填，则系统会尝试计算出相应的Dev。

## 删除

函数名：SmartRoute.Delete

密码：需要

参数：

Struct Route

{

Integer ID //待删除路由的ID

}

## 修改下一跳

函数名：SmartRoute.Set

密码：需要

参数：

Struct Route

{

Integer ID //待修改路由的ID

Integer ISP //指定ISP的ID，0表示不指定

Integer GatePolicy //选择下一跳的算法

Array Gates Struct //描述下一跳的列表

{

String IP //网关地址

String Dev //下一跳的输出接口

Boolean Auto //是否自动获取网关

Integer Weight //权值，范围为[1，256]

}

}

详解：

各个字段可参考添加一节中的同名字段。

## 获取路由表

函数名：SmartRoute.Get

密码：需要

参数：

Struct GetList

{

Boolean All //是否获取全部规则

Integer Start //获取的起始ID

Integer Count //希望获取的数量

}

返回值：

Array Result Struct

{

Integer ID //规则ID

String Description //描述信息

Integer GatePolicy //选择下一跳的算法

Boolean Status //路由的有效性

Integer ISP //指定ISP的ID，0表示不指定

Array Gates Struct //描述下一跳的列表

{

String IP //网关地址

String Dev //下一跳的输出接口

Boolean Auto //是否自动获取网关

Integer Weight //权值，范围为[1，256]

Boolean Status //下一跳的有效性

}}

}

详解：

参数的三个字段皆为可选项，但至少要填充一项。

All字段：不填则默认值为false

Start字段：不填则默认值为0

Count字段：不填则默认值为0

若All为true，则请求全部的路由规则，忽略Start与Count字段。

否则，请求从Start指定的ID号开始获取路由规则。

若Count不为0，则取Count条路由规则，否则，获取至最后一条路由规则。

返回值各个字段的含义可参考添加节的同名字段。

ID字段：指定路由规则的序号，从0起编。

Status字段描述规则或下一跳的有效性，true为有效，false为无效。

## 获取路由表长度

函数名：SmartRoute.GetCount

密码：需要

参数：无

返回值：

Integer Result

详解：

Result为整个智能路由表条目的数量。

## 修改描述

函数名：SmartRoute.Description

密码：需要

参数：

Struct Route

{

Integer ID //指定的路由规则ID

String Description //描述信息

}

# SNAT配置

## 添加

函数名：Nat.Src.Add

密码：需要

参数：

Struct Nat

{

Boolean Enabled //是否启用本规则

String Description //描述信息

Struct Match

{

String SrcNet //匹配的来源网络

String DestNet //匹配的目标网络

Integer Protocol //匹配的通讯协议编号

String ProtocolStr //匹配的通讯协议名称

Integer SrcPort //匹配的源端口号

String SrcPortStr //匹配的源端口名称

Integer DestPort //匹配的目标端口号

String DestPortStr //匹配的目标端口名称

String Dev //匹配的输出接口

}

Struct Action

{

Boolean Except //是否排除匹配

Boolean Masquerade //是否动态IP转换

String StartIP //IP转换起始地址

String EndIP //IP转换结束地址

Integer StartPort //端口转换的起始范围

Integer EndPort //端口转换的结束范围

}

}

详解：

参数中的所有字段皆为可选项，String类型字段默认为空，Integer类型字段默认为0，Boolean字段类型默认为false。

Match字段中的String类型的字段若为空或Integer类型的字段为0，表示不匹配此项。

若ProtocolStr、SrcPortStr、DestPortStr不为空，则忽略对应的Protocol、SrcPort、DestPort项。

若ProtocolStr不为tcp、udp或者Protocol不为6、17，则不应填充SrcPort、SrcPortStr、DestPort、DestPortStr、StartPort、EndPort字段。

SrcNet、DestNet字段描述网络地址，格式为xxx.xxx.xxx.xxx/xx，例如：172.19.0.0/16。

若Except字段为true，则忽略Masquerade、StartIP、EndIP、StartPort、EndPort字段。

若Masquerade字段为true，则忽略StartIP、EndIP字段。

StartIP、EndIP字段格式为xxx.xxx.xxx.xxx。

## 插入

函数名：Nat.Src.Insert

密码：需要

参数：

Struct Nat

{

Integer ID //新规则插入的位置

Boolean Enabled //是否启用本规则

String Description //描述信息

Struct Match

{

String SrcNet //匹配的来源网络

String DestNet //匹配的目标网络

Integer Protocol //匹配的通讯协议编号

String ProtocolStr //匹配的通讯协议名称

Integer SrcPort //匹配的源端口号

String SrcPortStr //匹配的源端口名称

Integer DestPort //匹配的目标端口号

String DestPortStr //匹配的目标端口名称

String Dev //匹配的输出接口

}

Struct Action

{

Boolean Except //是否排除匹配

Boolean Masquerade //是否动态IP转换

String StartIP //IP转换起始地址

String EndIP //IP转换结束地址

Integer StartPort //端口转换的起始范围

Integer EndPort //端口转换的结束范围

}

}

详解：

各个字段含义可参考添加一节的同名字段。

## 替换

函数名：Nat.Src.Replace

密码：需要

参数：

Struct Nat

{

Integer ID //要替换的规则的位置

Boolean Enabled //是否启用本规则

String Description //描述信息

Struct Match

{

String SrcNet //匹配的来源网络

String DestNet //匹配的目标网络

Integer Protocol //匹配的通讯协议编号

String ProtocolStr //匹配的通讯协议名称

Integer SrcPort //匹配的源端口号

String SrcPortStr //匹配的源端口名称

Integer DestPort //匹配的目标端口号

String DestPortStr //匹配的目标端口名称

String Dev //匹配的输出接口

}

Struct Action

{

Boolean Except //是否排除匹配

Boolean Masquerade //是否动态IP转换

String StartIP //IP转换起始地址

String EndIP //IP转换结束地址

Integer StartPort //端口转换的起始范围

Integer EndPort //端口转换的结束范围

}

}

详解：

各个字段含义可参考添加一节的同名字段。

## 删除

函数名：Nat.Src.Delete

密码：需要

参数：

Struct Nat

{

Integer ID //待删除规则的ID

}

## 开启/关闭

函数名：Nat.Src.Enabled

密码：需要

参数：

Struct Nat

{

Integer ID //待操作规则的ID

Boolean Enabled //true启用规则，false停用规则。

}

## 修改描述

函数名：Nat.Src.Description

密码：需要

参数：

Struct Nat

{

Integer ID //待操作规则的ID

String Description //指定的描述信息

}

## 获取列表

函数名：Nat.Src.Get

密码：需要

参数：

Struct GetList

{

Boolean All //是否获取全部规则

Integer Start //获取的起始ID

Integer Count //希望获取的数量

}

返回值：

Array Result Struct

{

Integer ID //规则ID

Boolean Status //规则的有效性

Boolean Enabled //是否启用本规则

String Description //描述信息

Struct Match

{

String SrcNet //匹配的来源网络

String DestNet //匹配的目标网络

Integer Protocol //匹配的通讯协议编号

String ProtocolStr //匹配的通讯协议名称

Integer SrcPort //匹配的源端口号

String SrcPortStr //匹配的源端口名称

Integer DestPort //匹配的目标端口号

String DestPortStr //匹配的目标端口名称

String Dev //匹配的输出接口

}

Struct Action

{

Boolean Except //是否排除匹配

Boolean Masquerade //是否动态IP转换

String StartIP //IP转换起始地址

String EndIP //IP转换结束地址

Integer StartPort //端口转换的起始范围

Integer EndPort //端口转换的结束范围

}

}

详解：

参数的三个字段皆为可选项，但至少要填充一项。

All字段：不填则默认值为false

Start字段：不填则默认值为0

Count字段：不填则默认值为0

若All为true，则请求全部的SNAT规则，忽略Start与Count字段。

否则，请求从Start指定的ID号开始获取SNAT规则。

若Count不为0，则取Count条SNAT规则，否则，获取至最后一条SNAT规则。

返回值的字段意义可参考添加一节的同名字段。

ID字段：指定SNAT规则的序号，从0起编。

Status字段描述规则的有效性，true为有效，false为无效。

## 获取列表长度

函数名：Nat.Src.GetCount

密码：需要

参数：无

返回值：

Integer Result

# DNAT配置

## 添加

函数名：Nat.Dest.Add

密码：需要

参数：

Struct Nat

{

Boolean Enabled //是否启用本规则

String Description //描述信息

Struct Match

{

String SrcNet //匹配的来源网络

String DestNet //匹配的目标网络

Integer Protocol //匹配的通讯协议编号

String ProtocolStr //匹配的通讯协议名称

Integer SrcPort //匹配的源端口号

String SrcPortStr //匹配的源端口名称

Integer DestPort //匹配的目标端口号

String DestPortStr //匹配的目标端口名称

String Dev //匹配的输入接口

}

Struct Action

{

Boolean Except //是否排除匹配

String StartIP //IP转换起始地址

String EndIP //IP转换结束地址

Integer StartPort //端口转换的起始范围

Integer EndPort //端口转换的结束范围

}

}

详解：

参数中的所有字段皆为可选项，String类型字段默认为空，Integer类型字段默认为0，Boolean字段类型默认为false。

Match字段中的String类型的字段若为空或Integer类型的字段为0，表示不匹配此项。

若ProtocolStr、SrcPortStr、DestPortStr不为空，则忽略对应的Protocol、SrcPort、DestPort项。

若ProtocolStr不为tcp、udp或者Protocol不为6、17，则不应填充SrcPort、SrcPortStr、DestPort、DestPortStr、StartPort、EndPort字段。

SrcNet、DestNet字段描述网络地址，格式为xxx.xxx.xxx.xxx/xx，例如：172.19.0.0/16。

若Except字段为true，则忽略StartIP、EndIP、StartPort、EndPort字段。

StartIP、EndIP字段格式为xxx.xxx.xxx.xxx。

## 插入

函数名：Nat.Dest.Insert

密码：需要

参数：

Struct Nat

{

Integer ID //新规则插入的位置

Boolean Enabled //是否启用本规则

String Description //描述信息

Struct Match

{

String SrcNet //匹配的来源网络

String DestNet //匹配的目标网络

Integer Protocol //匹配的通讯协议编号

String ProtocolStr //匹配的通讯协议名称

Integer SrcPort //匹配的源端口号

String SrcPortStr //匹配的源端口名称

Integer DestPort //匹配的目标端口号

String DestPortStr //匹配的目标端口名称

String Dev //匹配的输入接口

}

Struct Action

{

Boolean Except //是否排除匹配

String StartIP //IP转换起始地址

String EndIP //IP转换结束地址

Integer StartPort //端口转换的起始范围

Integer EndPort //端口转换的结束范围

}

}

详解：

各个字段含义可参考添加一节的同名字段。

## 替换

函数名：Nat.Dest.Replace

密码：需要

参数：

Struct Nat

{

Integer ID //待修改规则的位置

Boolean Enabled //是否启用本规则

String Description //描述信息

Struct Match

{

String SrcNet //匹配的来源网络

String DestNet //匹配的目标网络

Integer Protocol //匹配的通讯协议编号

String ProtocolStr //匹配的通讯协议名称

Integer SrcPort //匹配的源端口号

String SrcPortStr //匹配的源端口名称

Integer DestPort //匹配的目标端口号

String DestPortStr //匹配的目标端口名称

String Dev //匹配的输入接口

}

Struct Action

{

Boolean Except //是否排除匹配

String StartIP //IP转换起始地址

String EndIP //IP转换结束地址

Integer StartPort //端口转换的起始范围

Integer EndPort //端口转换的结束范围

}

}

详解：

各个字段含义可参考添加一节的同名字段。

## 删除

函数名：Nat.Dest.Delete

密码：需要

参数：

Struct Nat

{

Integer ID //待删除规则的ID

}

## 开启/关闭

函数名：Nat.Dest.Enabled

密码：需要

参数：

Struct Nat

{

Integer ID //待操作规则的ID

Boolean Enabled //true启用规则，false停用规则。

}

## 修改描述

函数名：Nat.Dest.Description

密码：需要

参数：

Struct Nat

{

Integer ID //待操作规则的ID

String Description //指定的描述信息

}

## 获取列表

函数名：Nat.Dest.Get

密码：需要

参数：

Struct GetList

{

Boolean All //是否获取全部规则

Integer Start //获取的起始ID

Integer Count //希望获取的数量

}

返回值：

Array Result Struct

{

Integer ID //规则ID

Boolean Status //规则的有效性

Boolean Enabled //是否启用本规则

String Description //描述信息

Struct Match

{

String SrcNet //匹配的来源网络

String DestNet //匹配的目标网络

Integer Protocol //匹配的通讯协议编号

String ProtocolStr //匹配的通讯协议名称

Integer SrcPort //匹配的源端口号

String SrcPortStr //匹配的源端口名称

Integer DestPort //匹配的目标端口号

String DestPortStr //匹配的目标端口名称

String Dev //匹配的输入接口

}

Struct Action

{

Boolean Except //是否排除匹配

String StartIP //IP转换起始地址

String EndIP //IP转换结束地址

Integer StartPort //端口转换的起始范围

Integer EndPort //端口转换的结束范围

}

}

详解：

参数的三个字段皆为可选项，但至少要填充一项。

All字段：不填则默认值为false

Start字段：不填则默认值为0

Count字段：不填则默认值为0

若All为true，则请求全部的DNAT规则，忽略Start与Count字段。

否则，请求从Start指定的ID号开始获取DNAT规则。

若Count不为0，则取Count条DNAT规则，否则，获取至最后一条DNAT规则。

返回值的字段意义可参考添加一节的同名字段。

ID字段：指定DNAT规则的序号，从0起编。

Status字段描述规则的有效性，true为有效，false为无效。

## 获取列表长度

函数名：Nat.Dest.GetCount

密码：需要

参数：无

返回值：

Integer Result

# ARP配置

## 获取静态ARP表项

函数名：Arp.Get.Static

密码：需要

参数：无

返回值：

Array Arp Struct

{

String IP //IP地址

String MAC //物理地址

String Dev //对应接口名称

Integer Status //表项状态

}

详解：

Status字段有如下可选值：

0：无效表项

1：动态表项

2：静态表项

## 获取动态ARP表项

函数名：Arp.Get.Dynamic

密码：需要

参数：无

返回值：

Array Arp Struct

{

String IP //IP地址

String MAC //物理地址

String Dev //对应接口名称

Integer Status //表项状态

}

详解：

Status字段有如下可选值：

0：无效表项

1：动态表项

2：静态表项

## 设置静态ARP表项

函数名：Arp.Set

密码：需要

参数：

Struct Arp

{

String IP //IP地址

String MAC //物理地址

String Dev //对应接口名称

}

## 删除ARP表项

函数名：Arp.Delete

密码：需要

参数：

Struct Arp

{

String IP //IP地址

String Dev //对应接口名称

}

# ADSL配置

## 获取全部接口

函数名：Adsl.GetAll

密码：需要

参数：无

返回结果：

Array Result Struct

{

Integer ID //接口的ID

String Description //描述信息

String Dev //ADSL接口名称

String Ethernet //所使用的物理接口名称

String User //拨号用户名

String Password //拨号密码

Integer Timeout //无动作超时自动断线

String IP //获取的IP地址

String Dns //获取的Dns地址

String Gate //获取的网关地址

Integer MTU //MTU值

Integer Status //ADSL状态

Integer RX //收到的数据包个数

Integer TX //发送的数据包个数

Integer Time //已连接的时间

}

详解：

Status字段有如下可选值：

0：未拨号

1：正在拨号

2：连接已建立

3：无效的配置

## 获取指定接口

函数名：Adsl.Get

密码：需要

参数：

Struct Adsl

{

String Dev

}

返回结果：

Struct Result

{

Integer ID //接口的ID

String Description //描述信息

String Dev //ADSL接口名称

String Ethernet //所使用的物理接口名称

String User //拨号用户名

String Password //拨号密码

Integer Timeout //无动作超时自动断线

String IP //获取的IP地址

String Dns //获取的Dns地址

String Gate //获取的网关地址

Integer MTU //MTU值

Integer Status //ADSL状态

Integer RX //收到的数据包个数

Integer TX //发送的数据包个数

Integer Time //已连接的时间

}

详解：

Status字段参考获取全部接口一节解释。

## 添加

函数名：Adsl.Add

密码：需要

参数：

Struct Adsl

{

String Description //描述信息

String Dev //ADSL接口名称

String Ethernet //所使用的物理接口名称

String User //拨号用户名

String Password //拨号密码

Integer Timeout //无动作超时自动断线

Integer MTU //MTU值，范围[68，1492]

}

详解：

除Dev、Ethernet字段，其余字段皆为可选项。

String类型的字段默认值是空，Integer类型的字段默认值为0。例外：MTU字段的默认值为1492。

当Timeout字段为0时，若配置无误则会自动拨号。

Dev字段的格式为pppX（X为数字）

## 删除

函数名：Adsl.Delete

密码：需要

参数：

Struct Adsl

{

String Dev //待删除的ADSL接口名称

}

## 设置

数名：Adsl.Set

密码：需要

参数：

Struct Adsl

{

String Dev //待操作的ADSL接口名称

String User //拨号用户名

String Password //拨号密码

Integer Timeout //无动作超时自动断线

Integer MTU //MTU值，范围[68，1492]

}

详解：

各个字段含义可参考添加一节的同名字段。

## 拨号

函数名：Adsl.Dial

密码：需要

参数：

Struct Adsl

{

String Dev //要拨号的ADSL接口名称

}

## 断开连接

函数名：Adsl.Stop

密码：需要

参数：

Struct Adsl

{

String Dev //要断开连接的ADSL接口名称

}

## 修改描述

函数名：Adsl.Description

密码：需要

参数：

Struct Adsl

{

String Dev //待操作的ADSL接口名称

String Description //指定的描述信息

}

# 虚拟服务配置

## 添加服务

函数名：VirtualService.Add

密码：需要

参数：

Struct Service

{

Integer Mark //虚拟服务标记值，范围[0xF000，0xFFFF]

String Description //描述信息

String Name //虚拟服务名称

}

详解：

三个字段皆为可选字段。

若Mark字段不填充，则由程序自动生成一个虚拟服务标记值。

Name字段必须满足如下规则：

a、非纯数字。

b、由数字、字母、下划线、减号组成。

c、可以为空，表示此服务无名称，只能通过ID进行索引。

## 删除服务

函数名：VirtualService.Delete

密码：需要

参数：

Struct Service

{

Integer ID //虚拟服务ID

String Name //虚拟服务名称

}

详解：

ID与Name字段皆为可选字段，但至少选择填充一个。

Name字段默认值为空。

若Name字段值为空，表示不匹配Name字段，因此ID字段必须填充。

## 设置服务地址

函数名：VirtualService.Address

密码：需要

参数：

Struct Service

{

Integer ID //虚拟服务ID

String Name //虚拟服务名称

Array IP Struct

{

String IP //指定一个VIP地址

String Dev //指定对应的接口

}

Array TcpPorts Struct

{

Integer Min //描述端口范围一个子域的起始值

Integer Max //描述端口范围一个子域的结束值

}

Array UdpPorts Struct

{

Integer Min //描述端口范围一个子域的起始值

Integer Max //描述端口范围一个子域的结束值

}

}

详解：

Name字段与ID字段可参考删除服务一节。

IP、TcpPorts、UdpPorts字段皆为可选字段，若不填，则忽略此项，不进行修改。

IP字段定义虚拟服务的VIP列表，每一个元素定义一个VIP，其子字段为：

IP字段定义VIP地址，格式为xxx.xxx.xxx.xxx。

Dev字段为可选项，默认为空。描述对应VIP所属的接口，若值为空，则不属于任何接口。

TcpPorts字段与UdpPorts字段用来描述一个端口的范围，通过描述一串连续的子域区间，构成一个不连续的端口范围。

## 设置服务特征

函数名：VirtualService.Service

密码：需要

参数：

Struct Service

{

Integer ID //虚拟服务ID

String Name //虚拟服务名称

Integer Schedule //虚拟服务调度算法

Boolean Persistent //是否开启持久连接保持

Integer PersistentTimeout //持久连接超时时间，范围[60，86400]

String PersistentNetmask //持久连接范围掩码

Struct Monitor //真实服务器检查配置

{

Integer Interval //检查间隔，范围[1，360]

Integer Timeout //超时时间，范围[1，360]

Integer Retry //重试次数，范围[1，16]

Integer Type //检查类型

Integer Port //检查端口

}

}

详解：

Name字段与ID字段参考删除服务一节。

其余字段皆为可选字段，若不填，则忽略对应项，即使用原有值。

Schedule字段可选值为：

0：轮询调度。

1：加权轮询调度。

2：最小连接调度。

3：加权最小连接调度。

4：基于局部性的最少连接调度。

5：带复制的基于局部性的最小连接调度。

6：目标地址散列调度。

7：源地址散列调度。

8：最短期望延迟调度。

9：无需等待队列调度。

PersistentNetmask字段格式为xxx.xxx.xxx.xxx

Type字段的可选值为：

0：关闭检查

1：使用ping方式进行检查

2：使用TCP连接方式进行检查

## 流量控制

函数名：VirtualService.Traffic

密码：需要

参数：

Struct Service

{

Integer ID //虚拟服务ID

String Name //虚拟服务名称

Struct Traffic

{

Integer Up //上行流量，范围非负，单位kbps

Integer Down //下行流量，范围非负，单位kbps

}

}

详解：

Name字段与ID字段参考删除服务一节。

Up、Down字段皆为可选字段。若不填，则使用之前的值。若值为0，则对于相应的流量不进行限制。

## 开启/关闭虚拟服务

函数名：VirtualService.Enabled

密码：需要

参数：

Struct Service

{

Integer ID //虚拟服务ID

String Name //虚拟服务名称

Boolean Enabled //是否开启此虚拟服务

}

详解：

Name字段与ID字段参考删除服务一节。

## 高可用配置

函数名：VirtualService.HA

密码：需要

参数：

Struct Service

{

Integer ID //虚拟服务ID

String Name //虚拟服务名称

Integer HA //虚拟服务的高可用类型

}

详解：

Name字段与ID字段参考删除服务一节。

HA字段的可选值如下：

0：本地服务。

1：服务的主机为本机。

2：服务的主机为另外一台。

## 修改描述

函数名：VirtualService.Description

密码：需要

参数：

Struct Service

{

Integer ID //虚拟服务ID

String Name //虚拟服务名称

String Description //虚拟服务描述

}

详解：

Name字段与Description字段皆为可选值，若不选，则忽略此项。

将虚拟服务编号ID的服务名称与描述修改为Name与Description。

## 名称转换为ID

函数名：VirtualService.NameToID

密码：需要

参数：

Struct Service

{

String Name //虚拟服务名称

}

返回值：

Struct Service

{

Integer ID //虚拟服务ID

String Name //虚拟服务名称

}

## 添加真实服务器

函数名：VirtualService.Server.Add

密码：需要

参数：

Struct Service //虚拟服务

{

Integer ID //虚拟服务ID

String Name //虚拟服务名称

Array Servers Struct //真实服务器

{

String IP //服务器IP地址

String Name //服务器名称

Integer Action //转发方式

Integer Weight //服务器权值，范围[0，65535]

Boolean Enabled //是否启用此服务器

Integer MapPort //映射的端口号

}

}

详解：

虚拟服务中的Name、ID字段可参考删除服务一节。

真实服务器中除IP字段以外皆为可选字段，Name字段默认为空，Action字段默认为0，Weight字段默认为1，Enabled字段默认为true。

Mapport字段表示当虚拟服务为单IP单端口，真实服务器转发方式为网络地址转换方式时，可以选择映射的端口号，若为0，则表示与虚拟服务端口号一致。

Action字段的可选值为：

0：直接路由模式

1：网络地址转换方式

2：IP隧道模式

## 删除真实服务器

函数名：VirtualService.Server.Delete

密码：需要

参数：

Struct Service //虚拟服务

{

Integer ID //虚拟服务ID

String Name //虚拟服务名称

Array Servers Struct //真实服务器

{

Integer ID //服务器ID

String Name //服务器名称

}

}

详解：

虚拟服务中的Name、ID字段可参考删除服务一节。

真实服务器中的Name、ID字段类似于虚拟服务中的Name、ID字段。

## 修改真实服务器

函数名：VirtualService.Server.Set

密码：需要

参数：

Struct Service //虚拟服务

{

Integer ID //虚拟服务ID

String Name //虚拟服务名称

Array Servers Struct //真实服务器

{

Integer ID //服务器ID

String Name //服务器名称

String IP //服务器IP地址

Integer Action //转发方式

Integer Weight //服务器权值，范围[0，65535]

Boolean Enabled //是否启用此服务器

Integer MapPort //映射的端口号

}

}

详解：

虚拟服务中的Name、ID字段可参考删除服务一节。

真实服务器中的Name、ID字段类似于虚拟服务中的Name、ID字段。

Name、IP、Action、Weight、Enabled、Mapport字段皆为可选值，其含义可参考添加真实服务器一节。

## 开启/关闭真实服务器

函数名：VirtualService.Server.Enabled

密码：需要

参数：

Struct Service //虚拟服务

{

Integer ID //虚拟服务ID

String Name //虚拟服务名称

Array Servers Struct //真实服务器

{

Integer ID //服务器ID

String Name //服务器名称

Boolean Enabled //是否开启此服务器

}

}

详解：

各个字段含义可参考删除真实服务器一节。

## 获取虚拟服务个数

函数名：VirtualService.GetCount

密码：需要

参数：无

返回值：

Integer Result

## 获取虚拟服务列表

函数名：VirtualService.Get

密码：需要

参数：

Struct GetList

{

Boolean All //是否获取全部

Integer Start //获取的起始ID

Integer Count //希望获取的数量

}

返回值：

Array Result Struct

{

Integer ID //虚拟服务ID

Integer Mark //虚拟服务标记值，范围[0xF000，0xFFFF]

String Description //描述信息

String Name //虚拟服务名称

Boolean Enabled //是否开启此虚拟服务

Array IP Struct

{

String IP //指定一个VIP地址

String Dev //指定对应的接口

Boolean Status //此VIP的有效性

}

Array TcpPorts Struct

{

Integer Min //描述端口范围一个子域的起始值

Integer Max //描述端口范围一个子域的结束值

}

Array UdpPorts Struct

{

Integer Min //描述端口范围一个子域的起始值

Integer Max //描述端口范围一个子域的结束值

}

Integer Schedule //虚拟服务调度算法

Boolean Persistent //是否开启持久连接保持

Integer PersistentTimeout //持久连接超时时间，范围[60，86400]

String PersistentNetmask //持久连接范围掩码

Struct Monitor //真实服务器检查配置

{

Integer Interval //检查间隔，范围[1，360]

Integer Timeout //超时时间，范围[1，360]

Integer Retry //重试次数，范围[1，16]

Integer Type //检查类型

Integer Port //检查端口

}

Integer HA //虚拟服务的高可用类型

Boolean HAStatus //虚拟服务高可用状态

Array Servers Struct //真实服务器

{

Integer ID //服务器ID

String IP //服务器IP地址

String Name //服务器名称

Integer Action //转发方式

Integer Weight //服务器权值，范围[0，65535]

Boolean Enabled //是否启用此服务器

Integer MapPort //映射的端口号

Boolean Status //服务器状态

Integer Active //服务器活动连接数

Integer InActive //服务器非活动连接数

}

Struct Traffic //流量限制

{

Integer Up //上行流量，范围非负，单位kbps

Integer Down //下行流量，范围非负，单位kbps

}

}

详解：

参数的三个字段皆为可选项，但至少要填充一项。

All字段：不填则默认值为false

Start字段：不填则默认值为0

Count字段：不填则默认值为0

若All为true，则请求全部的虚拟服务信息，忽略Start与Count字段。

否则，请求从Start指定的ID号开始获取虚拟服务信息。

若Count不为0，则取Count条虚拟服务信息，否则，获取至最后一条虚拟服务信息。

各个字段的含义可参考本章各节内容。

HAStatus字段表示高可用状态，若HA为0则其值始终为true，否则，表示高可用模块决定是否启用对应的虚拟服务。

真实服务器的Status字段若为true则表示服务器正常，false表示服务器已死。

若真实服务器的Enabled字段为false，则对应的Status、Active、InActive字段无意义。

## 获取虚拟服务数据统计

函数名：Statistic.Service

密码：需要

参数：

Struct Statistic

{

Integer ID //虚拟服务ID

Integer From //起始时间

Integer To //结束时间

Integer Interval //时间间隔，正整数

}

返回值：

String Result

详解：

From、To字段的值为从1970-1-1 0：0：0至指定的本地时间之间的秒数。

Result值为一系列以分号分隔的整数，每个整数表示一个时间间隔之后的统计值。

Interval字段的可选值为60、3600、86400的正整数倍。

## 获取真实服务器数据统计

函数名：Statistic.Server

密码：需要

参数：

Struct Statistic

{

String IP //真实服务器IP地址

Integer From //起始时间

Integer To //结束时间

Integer Interval //时间间隔，正整数

}

返回值：

String Result

详解：

From、To字段的值为从1970-1-1 0：0：0至指定的本地时间之间的秒数。

Result值为一系列以分号分隔的整数，每个整数表示一个时间间隔之后的统计值。

Interval字段的可选值为60、3600、86400的正整数倍。

# 高可用配置

## 获取配置

函数名：HA.Get

密码：需要

参数：无

返回值：

Struct Result

{

Integer Keepalive //心跳间隔，范围[10，360000]

Integer Deadtime //死亡时间，范围[10，360000]

Integer Warntime //警告时间，范围[10，360000]

Integer Initdead //初始化时间，范围[10，360000]

Boolean Autoback //是否自动迁回资源

Integer Port //心跳端口

String IP //对方的IP

String Dev //输出心跳的接口

Boolean DevStatus //接口的状态，true为正常

String Hostname //对方的主机名

Boolean Enabled //是否开启高可用

Boolean Self //自己的资源状态，true为启用

Boolean Other //对方的资源状态，true为启用

Array Ping Array String //Ping检测组

Integer Sync //连接同步使用的ID，范围[0，255]

}

详解：

所有时间字段的值皆为毫秒单位。

## 设置

函数名：HA.Set

密码：需要

参数：

Struct HA

{

Integer Keepalive //心跳间隔，范围[10，360000]

Integer Deadtime //死亡时间，范围[10，360000]

Integer Warntime //警告时间，范围[10，360000]

Integer Initdead //初始化时间，范围[10，360000]

Boolean Autoback //是否自动迁回资源

Integer Port //心跳接口

String IP //对方的IP

String Dev //输出心跳的接口

String Hostname //对方的主机名

Array Ping Array String //Ping检测组

Integer Sync //连接同步使用的ID，范围[0，255]

}

详解：

所有字段皆为可选字段，若不选，则使用当前正在使用的值。

Ping字段为二维数组类型，每个元素为一个域名或IP地址。

Warntime要求为Keepalive的5倍以上。

Deadtime要求为Warntime的2倍以上。

Initdead要求为Deadtime的2倍以上。

Port字段若为0，将使得高可用无法启动。

IP字段若为空，将使得高可用无法启动。

Hostname字段若为空，将使得高可用无法启动。

所有时间字段的值皆为毫秒单位。

Sync为0时表示关闭连接同步。

在高可用启用的情况下不可进行设置。

## 开启/关闭高可用

函数名：HA.Enabled

密码：需要

参数：

Struct HA

{

Boolean Enabled //是否启用高可用

}

# 智能DNS配置

## 设置BIND

函数名：Bind.Service

密码：需要

参数：

Struct Bind

{

Boolean Reverse //是否开启反向解析

Integer Port //设置服务端口，范围[1,65535]

}

详解：

所有字段皆为可选字段，若不选，则使用当前正在使用的值。

## 开启/关闭BIND

函数名：Bind.Enabled

密码：需要

参数：

Struct Bind

{

Boolean Enabled //是否启用智能DNS

}

## 添加DNS策略

函数名：Bind.ARecord.Add

密码：需要

参数：

Struct ARecord

{

String Name //域名

Boolean Enabled //是否启用，默认启用

Boolean ReturnAll //是否返回全部地址，默认false

Integer TTL //生存期，默认值86400

Array Alias String //别名

Array Servers Struct //对应的服务器列表

{

Integer ISP //ISP的ID，若为0，表示不指定ISP

String IP

}

}

详解：

除Name字段外，其余字段皆为可选字段，若不选，则使用默认值。

## 删除DNS策略

函数名：Bind.ARecord.Delete

密码：需要

参数：

Struct ARecord

{

Integer ID //待删除的策略ID

String Name //待删除的策略的名称

}

详解：

两个字段皆为可选值，但至少选择一个。

若二者皆填充，则以ID字段优先。

## 修改DNS策略

函数名：Bind.ARecord.Set

密码：需要

参数：

Struct ARecord

{

Integer ID //待修改的策略的ID

String Name //新的域名

Boolean Enabled //是否启用，默认启用

Boolean ReturnAll //是否返回全部地址，默认false

Integer TTL //生存期，默认值86400

Array Alias String //别名

Array Servers Struct //对应的服务器列表

{

Integer ISP //ISP的ID，若为0，表示不指定ISP

String IP

}

}

详解：

同名字段可参考添加DNS策略一节。

需要注意的是，ID与Name字段至少填一个，若不填ID，则以Name查找DNS策略，此时Name字段不能作为新域名的名称。

## 获取BIND基本信息

函数名：Bind.Get.Service

密码：需要

参数：无

返回值：

Struct Bind

{

Integer Port //服务端口

Boolean Reverse //是否支持反向解析

Boolean Enabled //是否开启BIND

}

## 获取DNS策略信息

函数名：Bind.Get.ARecord

密码：需要

参数：

Struct ARecord

{

Integer ID //查询策略ID

String Name //查询策略的名称

}

返回值：

Struct ARecord

{

Integer ID //策略的ID

String Name //域名

Boolean Enabled //是否启用

Boolean ReturnAll //是否返回全部地址

Integer TTL //生存期

Array Alias String //别名

Array Servers Struct //对应的服务器列表

{

Integer ISP //ISP的ID，若为0，表示不指定ISP

String IP

}

}

详解：

ID、Name字段参考删除DNS策略一节。

## 获取BIND全部信息

函数名：Bind.Get.All

密码：需要

参数：无

返回值：

Struct Bind

{

Integer Port //服务端口

Boolean Reverse //是否支持反向解析

Boolean Enabled //是否开启BIND

Array A Struct

{

Integer ID //策略的ID

String Name //域名

Boolean Enabled //是否启用

Boolean ReturnAll //是否返回全部地址

Integer TTL //生存期

Array Alias String //别名

Array Servers Struct //对应的服务器列表

{

Integer ISP //ISP的ID，若为0，表示不指定ISP

String IP

}

}

}

## 查询DNS策略ID

函数名：Bind.Get.IDByName

密码：需要

参数：

Array ARecord Struct

{

String Name //待查找的域名

}

返回值：

Array ARecord Struct

{

Integer ID //对应域名的ID

String Name //待查找的域名

}

# ISP配置

## 获取ISP数据列表

函数名：ISP.Get.List.All

密码：需要

参数：无

返回值：

Array ISP Struct

{

Integer ID //ISP的ID

String Name //ISP的名字

Array Net String //ISP对应的网段列表

}

详解：

Net字段各个元素的格式形如192.168.0.0/16。

## 获取ISP名字列表

函数名：ISP.Get.List.Name

密码：需要

参数：无

返回值：

Array ISP Struct

{

Integer ID //ISP的ID

String Name //ISP的名字

}

## 获取ISP信息

函数名：ISP.Get.Item.All

密码：需要

参数：

Struct ISP

{

Integer ID

}

返回值：

Struct ISP

{

Integer ID //ISP的ID

String Name //ISP的名字

Array Net String //ISP对应的网段列表

}

详解：

Net字段各个元素的格式形如192.168.0.0/16。

## 获取ISP名字

函数名：ISP.Get.Item.Name

密码：需要

参数：

Struct ISP

{

Integer ID

}

返回值：

Struct ISP

{

Integer ID //ISP的ID

String Name //ISP的名字

}

## 添加ISP

函数名：ISP.Add

密码：需要

参数：

Struct ISP

{

String Name //ISP的名字

Array Net String //ISP对应的网段列表

}

详解：

Net字段各个元素的格式形如192.168.0.0/16。

## 删除ISP

函数名：ISP.Delete

密码：需要

参数：

Struct ISP

{

Integer ID //待删除ISP的ID

}

## 修改ISP

函数名：ISP.Set

密码：需要

参数：

Struct ISP

{

Integer ID //待修改的ISP的ID

String Name //ISP的名字

Array Net String //ISP对应的网段列表

}

详解：

Net字段各个元素的格式形如192.168.0.0/16。

# 黑名单设置

## 添加规则

函数名：BlackList.Add

密码：需要

参数：

Struct Rule

{

String SrcNet //匹配的来源网络

String DestNet //匹配的目标网络

Integer Protocol //匹配的通讯协议编号

String ProtocolStr //匹配的通讯协议名称

Integer SrcPort //匹配的源端口号

Integer DestPort //匹配的目标端口号

}

详解：

各个字段皆为可选字段。若不填充，则使用默认值。

## 删除规则

函数名：BlackList.Delete

密码：需要

参数：

Struct Rule

{

Integer ID //待删除的规则ID

}

## 修改规则

函数名：BlackList.Set

密码：需要

参数：

Struct Rule

{

Integer ID //待修改的规则ID

String SrcNet //匹配的来源网络

String DestNet //匹配的目标网络

Integer Protocol //匹配的通讯协议编号

String ProtocolStr //匹配的通讯协议名称

Integer SrcPort //匹配的源端口号

Integer DestPort //匹配的目标端口号

}

详解：

除ID字段以外，各个字段皆为可选字段。若不填充，则使用现有值。

## 获取规则列表

函数名：BlackList.Get

密码：需要

参数：

Struct GetList

{

Boolean All //是否获取全部

Integer Start //获取的起始ID

Integer Count //希望获取的数量

}

返回值：

Array Result Struct

{

Integer ID //规则ID

String SrcNet //匹配的来源网络

String DestNet //匹配的目标网络

Integer Protocol //匹配的通讯协议编号

String ProtocolStr //匹配的通讯协议名称

Integer SrcPort //匹配的源端口号

Integer DestPort //匹配的目标端口号

}

详解：

参数的三个字段皆为可选项，但至少要填充一项。

All字段：不填则默认值为false

Start字段：不填则默认值为0

Count字段：不填则默认值为0

若All为true，则请求全部的规则，忽略Start与Count字段。

否则，请求从Start指定的ID号开始获取规则。

若Count不为0，则取Count条规则，否则，获取至最后一条规则。

## 获取规则列表长度

函数名：BlackList.GetCount

密码：需要

参数：无

返回值：

Integer Result

详解：

Result为整个规则列表的长度。

## 修改规则描述

函数名：BlackList.Description

密码：需要

参数：

Struct Rule

{

Integer ID //指定的规则ID

String Description //描述信息

}

# 其它配置

## 域名服务器

获取列表：

函数名：Dns.Server.Get

密码：需要

参数：无

返回值：

Array Result String //每个元素一个DNS服务器IP。

设置：

函数名：Dns.Server.Set

密码：需要

参数：

Array Dns String //每个元素一个DNS服务器IP。

## 主机名称

获取：

函数名：Hostname.Get

密码：需要

参数：无

返回值：

String Result //主机名

设置：

函数名：Hostname.Set

密码：需要

参数：

String Hostname //主机名

## DOS攻击防护

获取：

函数名：AntiDos.Get

密码：需要

参数：无

返回值：

Boolean Result //是否开启

设置：

函数名：AntiDos.Set

密码：需要

参数：

Boolean Enabled //是否开启

## 服务器探测

函数名：Server.Check

密码：需要

参数：无

## 系统状态

查看CPU：

函数名：Status.Core

密码：需要

参数：无

返回值：

Struct Result

{

Array Core String //每个核心的频率

String Usage //CPU使用率

}

查看内存：

函数名：Status.Memory

密码：需要

参数：无

返回值：

Struct Result

{

Integer Total //总内存大小

Integer Free //剩余空间

}

详解：

内存大小单位为KB。

查看磁盘：

函数名：Status.Disk

密码：需要

参数：无

返回值：

Struct Result

{

Integer Total //总磁盘空间

Integer Free //剩余空间

Integer Used //已使用的空间

}

详解：

磁盘空间单位为MB。

## 系统时间

获取：

函数名：Time.Get

密码：需要

参数：无

返回值：

Struct Result

{

Integer Tick //据本地1970-1-1 0：0：0的秒数

Integer Zone //时区，范围[-14，12]

}

设置：

函数名：Time.Set

密码：需要

参数：

Struct Time

{

Integer Tick //据本地1970-1-1 0：0：0的秒数

Integer Zone //时区，范围[-14，12]

}

详解：

时区值，东八区值为-8，格林尼治时区为0。

## 修改登录密码

函数名：Password.Set

密码：需要

参数：

String Password //新密码

## 配置文件

保存：

函数名：Configure.Save

密码：需要

参数：无

重新载入：

函数名：Configure.Reload

密码：需要

参数：无

导出：

函数名：Configure.Export

密码：需要

参数：无

返回值：

String Result //内容由程序决定

导入：

函数名：Configure.Import

密码：需要

参数：

String Data //内容与导出的结果一致

## 设置日志服务器

获取IP：

函数名：Logger.Get

密码：需要

参数：无

返回值：

Struct Result

{

Boolean Enabled //是否开启Log功能

String Domain //远程日志服务器主机名或IP

}

设置IP：

函数名：Logger.Set

密码：需要

参数：

Struct Log

{

Boolean Enabled //是否开启Log功能

String Domain //远程日志服务器主机名或IP

}

详解：

两个字段皆为可选值，若都为空，则什么也不做。

## 更新

初始化：

函数名：Updater.Reset

密码：需要

参数：无

上传文件：

函数名：Updater.Upload

密码：需要

参数：

Binary Data //BASE64编码的字符串

更新：

函数名：Updater.Start

密码：需要

参数：无

详解：

整个更新过程分为三步：

1、初始化，准备上传更新包。

2、分块上传更新包，按顺序上传。

3、执行更新动作。

## Licence管理

查看注册状态：

函数名：Licence.IsExist

密码：需要

参数：无

返回值：Boolean Result //是否注册

获取机器码：

函数名：Licence.Machine

密码：需要

参数：无

返回值：

String Result //机器码字符串

导入Licence：

函数名：Licence.Import

密码：需要

参数：

String Code //注册用的Licence代码字符串

详解：

若导入Licence失败，则会抛出RPC异常。