LWARE 接口编程手册

# ifm介绍

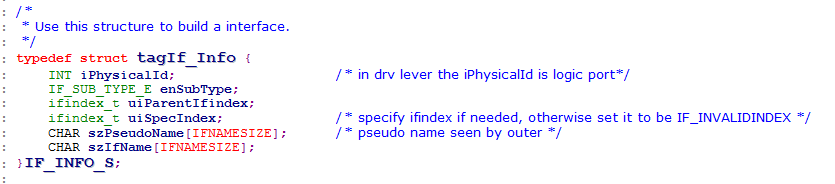
ifm 是用户态接口抽象层，此抽象层为编程用户提供统一的接口读取，设置，遍历，删除，创建等接口，屏蔽LINUX 设备层，方便用户开发。

# 接口API

## 接口创建函数

IF\_RET\_E IF\_CreatInterface( IN IF\_INFO\_S \* pstInfo, OUT ifindex\_t \* puiIfIndex );

入参IF\_INFO\_S 定义如下



enSubType 指定创建接口类型，具体定义在lw\_if\_pub.h中

uiParentIfindex 为父接口索引，存在父接口的，要填写此参数，例如L3VLAN和ETH接口往往存在父子关系。不存在则设置为IF\_INVALID\_IFINDEX

uiSpecIndex 为指定接口索引，如果此参数不为IF\_INVALID\_IFINDEX，系统则为用户创建的接口分配指定的索引，如果设置IF\_INVALID\_IFINDEX，则系统自动分配接口索引

szPseudoName 为伪接口名，用户可以随意设置，接口创建成功后，用户也可以更改

szIfName 为接口名，此接口名一旦设置后，后面将不能变更，其和ifindex都是此接口的

身份标志

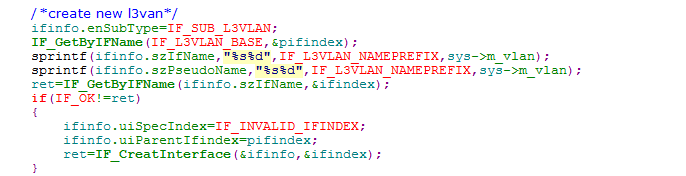
szIfName 的命名有一套规范，其中ETH网口，其名称是netdevice设备名，其他接口命名如下：

L3VLAN IF\_L3VLAN\_NAMEPREFIX+vid  
ETHPORT IF\_ETHPORT\_NAMEPREFIX+logicport

puiIfIndex 为出参，接口创建成功后，系统分配的接口索引

注意：此接口只能在master中使用，一般此接口只用在三层逻辑接口方面，其他物理接口系统会自动创建

举例：



## 接口删除函数

IF\_RET\_E IF\_DeleteInterface( IN ifindex\_t uiIfIndex );

uiIfIndex 为要删除接口的索引

注意： 如果此接口存在子接口，则要先删除字接口，否则无法删除

## 接口属性获取函数

IF\_RET\_E IF\_GetAttr( IN ifindex\_t uiIfindex, IN IF\_ATTR\_E enAttrId, INOUT VOID \* pValue ,IN UINT uiLen);

参数说明：

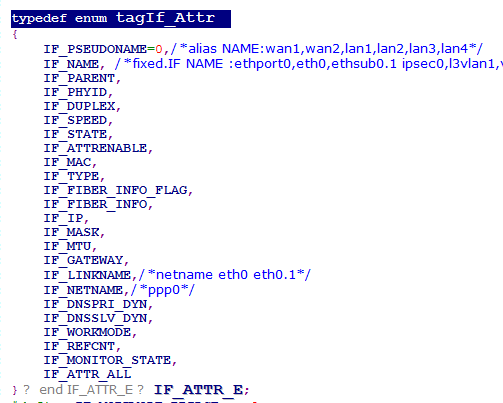
uiIfindex 接口索引

enAttrId 要获取的属性ID

pValue 返回值的BUFFER

uiLen BUFFER 的大小

可以获取的属性ID如下：

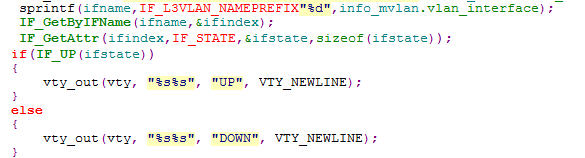


接口中存在都是实际生效的数据，例如网口可以配置为静态，也可以配置为动态获取

，其中的IP就是实际生效的IP信息

通过此接口可以获取接口状态

例如：



## 接口属性设置函数

IF\_RET\_E IF\_SetAttr( IN ifindex\_t uiIfindex, IN IF\_ATTR\_E iAttrId, IN VOID \* pValue,IN UINT uiLen );

参数说明：

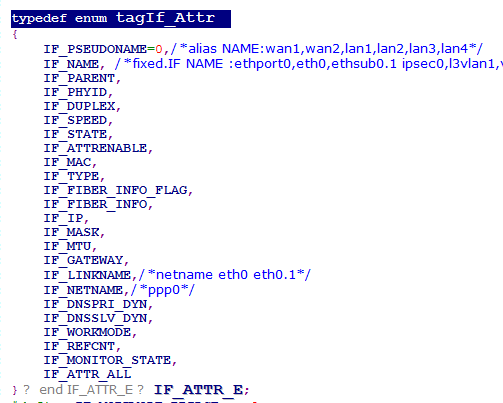
uiIfindex 接口索引

enAttrId 要获取的属性ID

pValue 设置值的BUFFER

uiLen BUFFER 的大小

可以设置的属性ID如下：



注意：

此接口只能在MASTER中调用，普通用户一般不要去设置。让master根据系统状态去设置。

## 接口遍历函数

#define IF\_GetFirstIfindex(enSubType,puiIfindex) IF\_GetFirstIfindexByDomain(enSubType,IF\_DOMAIN\_NORMAL,puiIfindex)

获取指定类型接口的首个接口

参数说明：

enSubType 要遍历的接口类型

puiIfindex 返回的接口索引

IF\_RET\_E IF\_GetNextIfindex( IN IF\_SUB\_TYPE\_E enSubType,INOUT ifindex\_t \* puiIfindex );

获取指定类型接口的下一个接口，

enSubType 要遍历的接口类型

puiIfindex 入参时，表示当前接口，出参时，表示获取的下一个接口

如果出参 puiIfindex 为IF\_INVALID\_IFINDEX，表示没有接口存在

举例：

static int timer\_handle\_vlanif\_monitor\_check(timer\_element\_t \*timer, void \*data)

{

INT iRet = 0;

ifindex\_t uiIfindex = 0;

UINT32 uiVid = 0;

UINT32 uiStatus=0;

IF\_STATUS\_E logstate = IFM\_MAX;

CHAR szBuf[BUF\_SIZE128]={0};

//CHAR szName[IFNAMESIZE];

CHAR szIfname[IFNAMESIZE];

CHAR szPseduname[BUF\_SIZE128];

iRet = IF\_GetFirstIfindex(IF\_SUB\_L3VLAN,&uiIfindex);

while(IF\_OK == iRet)

{

(VOID)cfg\_getval(uiIfindex,CONFIG\_IF\_ENABLE,szBuf,NULL,sizeof(szBuf));

if(strcmp(szBuf,ENUMTOSTR(IF\_MOD\_ENABLE))==0)

{

memset(szIfname,0,sizeof(szIfname));

if(IF\_OK!=IF\_GetAttr(uiIfindex,IF\_NAME,szIfname,sizeof(szIfname)))

{

iRet = IF\_GetNextIfindex(IF\_SUB\_L3VLAN,&uiIfindex);

continue;

}

sscanf(szIfname,IF\_L3VLAN\_NAMEPREFIX"%u",&uiVid);

#ifdef ONU\_STYLE

Drv\_GetVlanStatus((vlan\_id\_t)0, &uiStatus);

#else

Drv\_GetVlanStatus((vlan\_id\_t)uiVid, &uiStatus);

#endif

//printf("%s is %d.. \n",szPseduname,uiStatus);

if(UP == uiStatus)

{

/\*查看虚接口状态\*/

IF\_GetAttr(uiIfindex, IF\_STATE, &logstate, sizeof(IF\_STATUS\_E));

/\*虚接口UP，不做处理\*/

if(IF\_UP(logstate))

{

iRet = IF\_GetNextIfindex(IF\_SUB\_L3VLAN,&uiIfindex);

continue;

}

logstate = IFM\_NOIPGET;

IF\_SetAttr(uiIfindex, IF\_STATE, &logstate, sizeof(logstate));

memset(szPseduname,0,sizeof(szPseduname));

IF\_GetCliName(uiIfindex,szPseduname,sizeof(szPseduname));

ifm\_call\_notifiers(uiIfindex,IF\_NOTIFY\_PHYUP,NULL);

printf("%s is up.. \n",szPseduname);

}

else

{

/\*查看虚接口状态\*/

(VOID)IF\_GetAttr(uiIfindex, IF\_STATE, &logstate, sizeof(IF\_STATUS\_E));

/\*虚接口DOWN，不做处理\*/

if(!IF\_UP(logstate))

{

iRet = IF\_GetNextIfindex(IF\_SUB\_L3VLAN,&uiIfindex);

continue;

}

logstate = IFM\_PHYLINKDOWN;

IF\_SetAttr(uiIfindex, IF\_STATE, &logstate, sizeof(logstate));

/\*添加LOG信息\*/

memset(szPseduname,0,sizeof(szPseduname));

IF\_GetCliName(uiIfindex,szPseduname,sizeof(szPseduname));

ifm\_call\_notifiers(uiIfindex,IF\_NOTIFY\_PHYDOWN,NULL);

printf("%s is down.. \n",szPseduname);

}

}

iRet = IF\_GetNextIfindex(IF\_SUB\_L3VLAN,&uiIfindex);

}

return 0;

}

## 接口获取函数

#define IF\_GetByPseudoName(name,puiIfIndex) IF\_GetByPseudoNameDomain(name,IF\_DOMAIN\_NORMAL,puiIfIndex)

根据伪接口名获取接口索引

#define IF\_GetByIFName( name, puiIfIndex ) IF\_GetByIFNameDomain(name,IF\_DOMAIN\_NORMAL,puiIfIndex)

根据接口名获取索引

举例：



注意：

一般都是使用IF\_GetByIFName，如果用户自己定义了伪名，可以通过IF\_GetByPseudoName

获取

## 接口事件相关函数

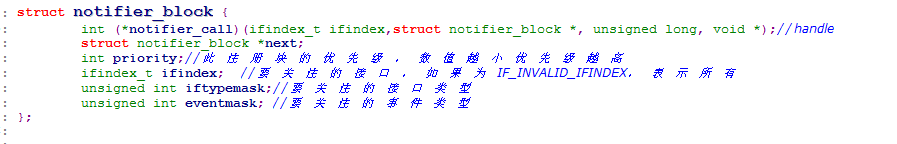
int ifm\_notifier\_register(struct notifier\_block \*nb);

注册关注事件

int ifm\_notifier\_unregister(struct notifier\_block \*nb);

注销关注事件

事件注册块定义如下：



举例：

例如根据接口变化去配置网络地址

