
Xenium LLMNR Responder Reference Manual

Release 5-alpha

Kaz Nishimura

Nov 19, 2022

Contents:

1	Internals of Xenium LLMNR Responder	1
1.1	Interface managers	1
1.2	Responders	2
2	Indices and tables	3
	Index	5

1.1 Interface managers

class interface_manager

This is an abstract class of interface managers.

interface_manager()

interface_manager(const *interface_manager*&) = delete
[protected] Constructs an interface manager object.

The copy constructor is deleted.

void operator=(const *interface_manager*&) = delete
The copy assignment operator is deleted.

virtual ~interface_manager()

Destructs an interface manager object.

int debug_level() const

Returns the current debug level of the interface manager object. The debug level controls the verbosity of debug logs to be emitted. The default debug level is 0.

void set_debug_level(int *debug_level*)

Sets the current debug level of the interface manager object.

void add_interface_listener(interface_listener **listener*)

void remove_interface_listener(interface_listener **listener*)

std::set<in_addr> in_addresses(unsigned int *interface_index*) const

std::set<in6_addr> in6_addresses(unsigned int *interface_index*) const

virtual void refresh(bool *maybe_asynchronous* = false) = 0

void remove_interfaces()
[protected]

void enable_interface(unsigned int *interface_index*)
[protected]

void disable_interface(unsigned int *interface_index*)
[protected]

```
void add_interface_address(unsigned int interface_index, int address_family, const void  
                           *address, size_t address_size)  
    [protected]
```

```
void remove_interface_address(unsigned int interface_index, int address_family, const  
                              void *address, size_t address_size)  
    [protected]
```

```
class rtnetlink_interface_manager : public interface_manager  
    RTNETLINK-based interface manager objects for Linux.
```

This implementation uses an RTNETLINK socket to communicate with the kernel.

```
rtnetlink_interface_manager()  
    Constructs an RTNETLINK-based interface manager object.
```

```
virtual ~rtnetlink_interface_manager()  
    Destructs an RTNETLINK-based interface manager object.
```

```
virtual void refresh(bool maybe_asynchronous = false) override
```

1.2 Responders

```
class responder  
    This is a class of LLMNR responders.
```

```
responder()
```

```
explicit responder(in_port_t port)  
    Constructs a responder object.
```

```
~responder()  
    Destructs a responder object.
```

CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`

I

`interface_manager` (*C++ class*), 1
`interface_manager::~interface_manager`
 (*C++ function*), 1
`interface_manager::add_interface_address`
 (*C++ function*), 1
`interface_manager::add_interface_listener`
 (*C++ function*), 1
`interface_manager::debug_level` (*C++ func-*
 tion), 1
`interface_manager::disable_interface` (*C++*
 function), 1
`interface_manager::enable_interface` (*C++*
 function), 1
`interface_manager::in6_addresses` (*C++ func-*
 tion), 1
`interface_manager::in_addresses` (*C++ func-*
 tion), 1
`interface_manager::interface_manager` (*C++*
 function), 1
`interface_manager::operator=` (*C++ function*),
 1
`interface_manager::refresh` (*C++ function*), 1
`interface_manager::remove_interface_address`
 (*C++ function*), 2
`interface_manager::remove_interface_listener`
 (*C++ function*), 1
`interface_manager::remove_interfaces` (*C++*
 function), 1
`interface_manager::set_debug_level` (*C++*
 function), 1

R

`responder` (*C++ class*), 2
`responder::~responder` (*C++ function*), 2
`responder::responder` (*C++ function*), 2
`rtnetlink_interface_manager` (*C++ class*), 2
`rtnetlink_interface_manager::~rtnetlink_interface_manager`
 (*C++ function*), 2
`rtnetlink_interface_manager::refresh` (*C++*
 function), 2
`rtnetlink_interface_manager::rtnetlink_interface_manager`
 (*C++ function*), 2