RFC 209 - NetworkInterfaceInformationService

Draft

32 Pages

Abstract

This document defines the Java API that provides the information of network interfaces in an OSGi environment. The bundles can get not only information of network interfaces but notification when the configuration of network interfaces to be changed to use this API.

0 Document Information

0.1 License

DISTRIBUTION AND FEEDBACK LICENSE, Version 2.0

The OSGi Alliance hereby grants you a limited copyright license to copy and display this document (the “Distribution”) in any medium without fee or royalty. This Distribution license is exclusively for the purpose of reviewing and providing feedback to the OSGi Alliance. You agree not to modify the Distribution in any way and further agree to not participate in any way in the making of derivative works thereof, other than as a necessary result of reviewing and providing feedback to the Distribution. You also agree to cause this notice, along with the accompanying consent, to be included on all copies (or portions thereof) of the Distribution. The OSGi Alliance also grants you a perpetual, non-exclusive, worldwide, fully paid-up, royalty free, limited license (without the right to sublicense) under any applicable copyrights, to create and/or distribute an implementation of the Distribution that: (i) fully implements the Distribution including all its required interfaces and functionality; (ii) does not modify, subset, superset or otherwise extend the OSGi Name Space, or include any public or protected packages, classes, Java interfaces, fields or methods within the OSGi Name Space other than those required and authorized by the Distribution. An implementation that does not satisfy limitations (i)-(ii) is not considered an implementation of the Distribution, does not receive the benefits of this license, and must not be described as an implementation of the Distribution. "OSGi Name Space" shall mean the public class or interface declarations whose names begin with "org.osgi" or any recognized successors or replacements thereof. The OSGi Alliance expressly reserves all rights not granted pursuant to these limited copyright licenses including termination of the license at will at any time.

EXCEPT FOR THE LIMITED COPYRIGHT LICENSES GRANTED ABOVE, THE OSGi ALLIANCE DOES NOT GRANT, EITHER EXPRESSLY OR IMPLIEDLY, A LICENSE TO ANY INTELLECTUAL PROPERTY IT, OR ANY THIRD PARTIES, OWN OR CONTROL. Title to the copyright in the Distribution will at all times remain with the OSGi Alliance. The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted therein are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

THE DISTRIBUTION IS PROVIDED "AS IS," AND THE OSGi ALLIANCE (INCLUDING ANY THIRD PARTIES THAT HAVE CONTRIBUTED TO THE DISTRIBUTION) MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE DISTRIBUTION ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.
NEITHER THE OSGi ALLIANCE NOR ANY THIRD PARTY WILL BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR RELATING TO ANY USE OR DISTRIBUTION OF THE DISTRIBUTION.

Implementation of certain elements of this Distribution may be subject to third party intellectual property rights, including without limitation, patent rights (such a third party may or may not be a member of the OSGi Alliance). The OSGi Alliance is not responsible and shall not be held responsible in any manner for identifying or failing to identify any or all such third party intellectual property rights.

The Distribution is a draft. As a result, the final product may change substantially by the time of final publication, and you are cautioned against relying on the content of this Distribution. You are encouraged to update any implementation of the Distribution if and when such Distribution becomes a final specification.

The OSGi Alliance is willing to receive input, suggestions and other feedback (“Feedback”) on the Distribution. By providing such Feedback to the OSGi Alliance, you grant to the OSGi Alliance and all its Members a non-exclusive, non-transferable,

worldwide, perpetual, irrevocable, royalty-free copyright license to copy, publish, license, modify, sublicense or otherwise distribute and exploit your Feedback for any purpose. Likewise, if incorporation of your Feedback would cause an implementation of the Distribution, including as it may be modified, amended, or published at any point in the future ("Future Specification"), to necessarily infringe a patent or patent application that you own or control, you hereby commit to grant to all implementers of such Distribution or Future Specification an irrevocable, worldwide, sublicenseable, royalty free license under such patent or patent application to make, have made, use, sell, offer for sale, import and export products or services that implement such Distribution or Future Specification. You warrant that (a) to the best of your knowledge you have the right to provide this Feedback, and if you are providing Feedback on behalf of a company, you have the rights to provide Feedback on behalf of your company; (b) the Feedback is not confidential to you and does not violate the copyright or trade secret interests of another; and (c) to the best of your knowledge, use of the Feedback would not cause an implementation of the Distribution or a Future Specification to necessarily infringe any third-party patent or patent application known to you. You also acknowledge that the OSGi Alliance is not required to incorporate your Feedback into any version of the Distribution or a Future Specification.

I HEREBY ACKNOWLEDGE AND AGREE TO THE TERMS AND CONDITIONS DELINEATED ABOVE.

0.2 Trademarks

OSGi™ is a trademark, registered trademark, or service mark of the OSGi Alliance in the US and other countries. Java is a trademark, registered trademark, or service mark of Oracle Corporation in the US and other countries. All other trademarks, registered trademarks, or service marks used in this document are the property of their respective owners and are hereby recognized.

0.3 Feedback

This document can be downloaded from the OSGi Alliance design repository at https://github.com/osgi/design The public can provide feedback about this document by opening a bug at https://www.osgi.org/bugzilla/.

0.4 Table of Contents

0.5 Terminology and Document Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY" and "OPTIONAL" in this document are to be interpreted as described in 10.1.

```
Source code is shown in this typeface.
```

0.6 Revision History

The last named individual in this history is currently responsible for this document.

Revision	Date	Comments
Initial	Nov 18, 2013	Initial version Shigekuni Kondo, NTT Corporation, kondo.shigekuni@lab.ntt.co.jp
0.2	Feb 10, 2014	Based on the last meeting, the section 5 has changed. Changed the design to service repository model. Shigekuni Kondo, NTT Corporation, kondo.shigekuni@lab.ntt.co.jp

1 Introduction

Java standard APIs (i.e. java.net.NetworkInterface, java.net.InetAddress) provide functions that allow IP network interface information, such as the IP address and MAC address to be obtained.

However, the bundle that wants to get network interface information has to monitor whether the information has changed or not for a certain period of time. Changes in network interface can be pushed to the bundles concerned, the need for polling by bundles can be eliminated.

In addition, some information cannot be obtained via Java standard APIs.

This RFC

defines the Java API that provides the information of network interfaces in an OSGi environment. The bundles can get not only information of network interfaces but notification when the configuration of network interfaces to use this API.

2 Application Domain

There are many bundles that use the IP network to communicate with other networked devices. In paticular, since a Residential Gateway (RGW) may have a number of network interfaces, each bundle running on the RGW needs to obtain an IP address and confirm whether the network interface associated with the allocated IP address suits the bundle's requirements or not.

For example, a protocol adapter needs the IP address of a network interface on the wide area network side to communicate with an external server. UPnP device service bundle needs the IP address that can be used to communicate with devices in a local area network.

These bundles can acquire information about the network interface via the following Java standard APIs.

- java.net.NetworkInterface
- *java.net.InetAddress*

3 Problem Description

Many application bundles on the RGW provide services on IP networks. For example, a protocol adapter for DMT Admin Service, a http server established by HTTP Service bundle and UPnP device service bundle use IP networks. In those cases, the bundles need to get information about the network interface on the RGW such as IP address, MAC address, network interface name, and so on.

The information about the network interface can be obtained by using Java standard APIs which are java.net.NetworkInterface and java.net.InetAddress. However, these APIs fail to provide the features needed by the bundles when they use the IP network in the following situations:

[Problem 1] There is no feature that sends a notification when information of the network interface (i.e. IP address) changes during runtime, e.g. the connection status or the assigned IP address.

[Problem 2] There is no feature that can acquire the subnet mask of the network interface.

[Problem 3] Operating System specific bundles must be prepared because some information about network interface depends on the Operating System.

If these functions were available, it would be very useful for bundles that need to use the IP network. However, a standard API does not exist at this time, so it must be prepared for each environment.

3.1 Use Cases

Use case 1

The TR-069 protocol adapter bundle on a RGW needs to communicate with an Auto Configuration Server (ACS). The ACS needs to know the public IP address of the Residential Gateway to send a UDP packet to the protocol adapter bundle for a connection request. In this case, the bundle has to provide the IP address to the ACS when the bundle is started or the IP address has changed.

Use case 2

When an HTTP Service bundle is available, at least one HTTP server is expected to run. When the HTTP server needs to be assigned to a specific network interface, the HTTP Service bundle has to know the information of the network interface. In addition, the HTTP Service bundle needs to know when the IP address of the network interface being managed changes.

Use case 3

The UPnP Device Service bundle needs to create the DatagramSocket for receiving and sending M-search messages. In the case of devices such as Residential Gateway, which has multi network interfaces, the UPnP bundle has to create a DatagramSocket that is bound to an appropriate local IP address. Therefore, the UPnP bundle needs to know the current IP address of the network interface and the replacement IP address.

Use case 4

An application bundle wants to obtain the subnet mask of the IP address to cover the situation in which the bundle needs to execute the Wake-up-On-LAN process.

Use case 5

An application wants to obtain information about available network services, such as available DNS Server, Log Server, NTP Server, or network characteristics, such as domain names, ARP cache timeouts, broadcast address, etc. For this, the local DHCP server can be queried to get those information.

Use case 6

A device running an OSGi framework in an mixed IPv4/IPv6 environment needs to get specific information about the network interface(s) in order to provide, for example, different services for the IPv4 and IPv6 environments.

4 Requirements

[REQ_1] The solution MUST provide means to send notifications to interested bundles whenever the information of network interface has changed.(i.e. The bundle is notified the information of IP address change from Network Interface Information Service implemented bundle)

[REQ_2] The solution MUST provide an API that can obtain information from a multiple network interfaces. Each network interface can provide information about multiple addresses. (An application bundle needs to know whether the network interface is a LAN interface or a WAN interface.).

[REQ_3] The solution MUST provide a mechanism that can provide the network interface information needed regardless of the Operating System type.

[REQ_4] The solution MUST provide the means of configuring network interface type. It will be defined for each environment (i.e. “LAN”, “WAN” that is bound to each logical interface) .

[REQ_5] The solution MUST provide an API that can obtain the subnet mask of each IP address.

[REQ_6] The solution MUST support both IPv4 and IPv6 environments (mixed or separately) and the corresponding characteristics, for example IPv4 and IPv6 addresses, multi-prefixes, multicast etc. .

[REQ_7] The solution SHOULD support the retrieval of MAC addresses for network interfaces.

[REQ_8] The solution MAY provide an API that allows alteration of network interface configurations.

[REQ_9] The solution MAY provide an API that can obtain the capability of network interface. (e.g. the physical type of network interface, list of BOOTP/DHCP command options, DNS server address, Default Gateway address, etc.)

5 Technical Solution

5.1 Intoroduction

When the IP address is changed, the bundles utilize IP address information (i.e. Http Service bundle running HTTP Servers) is necessary to detect the fact of the change. In case of using the standard Java API, such as java.net.InetAddress java.net.NetworkInterface, processing to confirm the IP address at regular intervals from the bundle itself is required. Since this is a process common to all bundles which are necessary to detect the change of IP address information, provision of services to notify a change of IP address is very effective.

Therefore the API provides the change notification feature for each network interface information (including the IP address information) is investigated in this RFC document. In addition to that, this RFC defines APIs which provide the functionalities to obtain the network interface information and the information of IP address which is bound it, to create and remove a logical network interface and to add, change and remove an IP address.

The name of the network interface is information dependent on the operating system. In order to be able to bundle implementation that uses the Network Interface Information Service is not aware of the differences in the operating system, the mechanism of identifying network interface is necessary in a format that does not depend on the operating system. This RFC also defines it.

5.2 Entities

- Network Interface
  Available and activated network interfaces provided in the execution environment. In this specification, the unit of the network interface is the logical interface, not the physical interface.

- NwIfInfo
  The OSGi service that provides information related to the Network Interface. This service provides functionalities corresponding to “java.net. NetworkInterface”.

- NwIfInetAddress
  The OSGi service that provides information of IP addresses available on execution environment on a Network Interface Information Service bundle is running.

- NwIfInfoAdmin
  The OSGi Service. This interface provides the functionality to create new logical network interface. IP address can be configured via created NwIfInfo Service.

- NwIfInfoPermission
  This class represents execute authority of the bundle which registers NwIfInfo Admin Service, NwIfInfo service and NwIfInetAddress service.

- NwIfInfoException
  Exception class that represents a processing failure of NwIfInfo.

- NwIfType
  The identifier of the network interface to be defined in a manner. It is independent of the operating system. This identifier string is not specified in this specification. The Network Interface Information Service bundle provider should define this identifier string. This identifier is used by user bundle to specify the network interface to be monitored.

- IPAdressType
  An identifier indicating the type of IP address (i.e. IPv4_PRIVATE , IPv6_GROVAL). This identifier is defined in this specification. This identifier is used by user bundle to specify the network interface to be monitored.

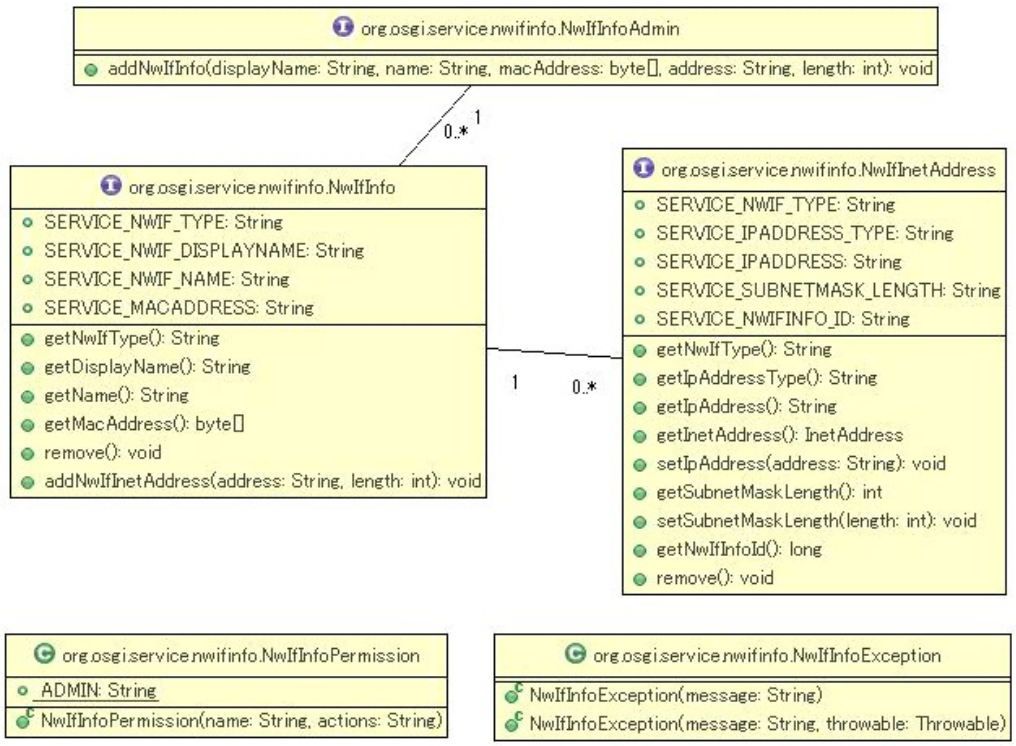

Fig.1 Class structure of Network Interface Information Service

<Network Interface Information Bundle>

To register three kind of services.

NwIfInfo service provides network interface information, this bundle registers this service each logical interface.

NwIfInetAddress service provides each IP address information, this bundle registers this service each IP address.

NwIfInetAddress service is associated with specific NwIfInfo service.

When information of network interface is changed, service properties of NwIfInfo setvice and NwIfInetAddress service will be modified. NwIfInfioAdmin service provides the functionality to create new logical network interface.

<User bundle>

Tracking necessary NwIfInfo service and NwIfInetAddress service (using filter).This bundle can be notified the change of network interface information via Service Event.

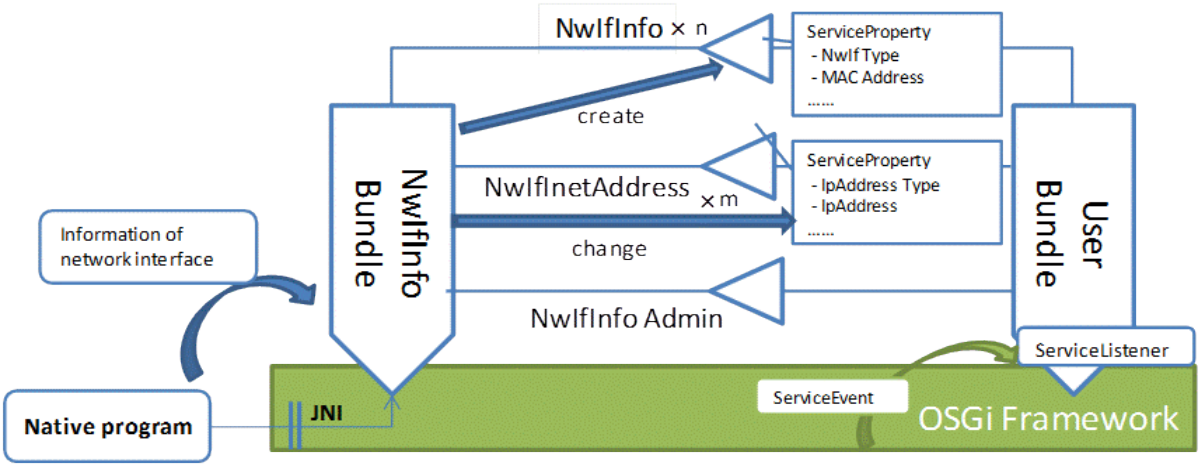

Fig.2 Overview of Network Interface Information Service

5.3 NwIfInfo Service

NwIfInfo is an interface that provides information about available network interfaces that are provided by the execution environment on the Network Interface Information Service bundle is running.

NwIfInfo service is registered to the service repository with service properties which are shown in the following table.

Table 1. Service properties of NwIfInfo Service

The key of service property	Description
service.nwif.type	Required property. Network interface type is set to a value.
service.macaddress	Required property. MAC address is set to a value.
service.nwif.name	Required property. Network interface name is set to a value.
service.nwif.displayname	Required property. Network interface display name is set to a value.

When a network interface becomes available, NwIfInfo service associated with the network interface is registered to service repository. If the network interface becomes unavailable, the corresponding NwIfInfo service is unregistered.

When the attribute values of the network interface are set to the service property changes, NwIfInfo service is updated. NwIfInfo interface provides a method corresponding to java.net.NetworkInterface in order to provide information on the network interface associated with. However, it does not provide in this interface method corresponding to the Static method. In addition to that, because InetAddress object or NetworkInterface object is registered in the service repository, the method to get those objects does not provide in NwIfInfo interface. NwIfInfo provides a method to retrieve the value of an attribute of a network interface.

NwIfInfo provides the remove method to delete the network interface associated with it. However, network interface that can be removed by the remove method is limited to the network interface generated via Network Interface Information Service. The network interface provided in the execution environment from the beginning cannot be deleted in the remove method. It is in order not to disturb the behavior of the non-OSGi application.

5.4 NwIfInfoAdmin Service

This interface provides the functionality to create new logical network interface. After the creating new logical network interface, NwIfInfo service will be registered. IP address can be added and configured via NwIfInfo service and NwIfInetAddress service.

5.5 NwIfInetAddress Service

NwIfInetAddress interface provides information of IP addresses available on execution environment on a Network Interface Information Service bundle is running.

NwIfInetAddress service is registered to the service repository with service properties which are shown in the following table.

Table 2. Service properties of NwIfInetAddress Service

The key of service property	Description
service.nwif.type	Required property. Network interface type is set to a value.
service.ipaddress.type	Required property. IP address type is set to a value.
service.ipaddress	Required property. IP address String is set to a value.
service.subnetmask.length	Required property. subnet mask length of the required properties IPv4, or IPv6 prefix length is set to a value.
service.nwifinfo.id	Required property. Service ID of the NwIfInfo servuice corresponding to the network interface binding this IP address is set to a value.

NwIfInetAddress service is registered to service repository for each available IP address.

When the associated IP addresses is deleted, or the network interface that the IP address is bound becomes unavailable, the NwIfInetAddress service is unregistered. When the associated IP address has been changed, NwIfInetAddress service is updated. The user bundle can detect the change of IP address by monitoring the registration or unregistering, updating of NwIfInetAddress service.

Because IP addresses are bound to the network interface of any, Service ID of the associated NwIfInfo service and its network interface type are set to service property.

NwIfInetAddress service is registered after the associated NwIfInfo service is registered. On the other hand, in case of unregistering services, after NwIfInetAddress all related services are unregistered, associated NwIfInfo service is unregistered.

NwIfInetAddress provides the remove method to delete the IP address associated with it. In addition to that NwIfInetAddress provides the Setter method to change the subnet mask (or prefix length) and IP address.

5.6 NwIfInfoPermission

This class represents execute authority of the bundle which registers NwIfInfo service and NwIfInetAddress service.

The name is a string using the network interface type and the IP address type. There are three types of name as below. "Str1" and "Str2" is used to represent string example.

<Name1>
It is consisted of Str1 and Str2 which is a dot-separated string (i.e. Str1.Str2). "*" is not included in the name. "Network interface type" is described as Str1 and "IP address type" is described as Str2. This name represents all NwIfInfo service and NwIfInetAddress Service that match the "Network interface type" represented Str1 and "IP address type" represented Str2.

<Name2>
The name is consisted of Str1 and "*" which is a dot-separated string (i.e. Str1.*).: "Network interface type" is described as Str1. This name represents all NwIfInfo service and NwIfInetAddress Service that match the "Network interface type" represented Str1.

<Name3>
The name is consisted of only "*". This name represents all NwIfInfo service and NwIfInetAddress Service.

The action string is defined only “ADMIN”. It means the permission to execute the Add, Set and Delete method for each information.

There is no need for this permission to execute of the Get method for each information.

5.7 NwIfInfoException

Exception class that represents a processing failure of NwIfInfo.

5.8 IP address type and Network interface type

5.8.1 Network interface type

In order to identify the network interface, it is possible to use the network interface name.

However, since the network interface name is an identifier that is dependent on the operating system, if network interface name is used as identifier, it is necessary to implement the user bundle being aware of the operating system. Therefore, in this specification, “network interface type” which is independent of the operating system is used to identify the network interface. The network interface type sting itself is not defined in this specification. It should be provided by the platform provider on which Network Interface Information Service bundle is running. For example, Network interface type “LAN” indicates the network interface to connect to a local area network, Network interface type “WAN” indicates the network interface to connect to the Internet. If a bundle wants to obtain the information of the network interface which connects to the Internet, the bundle is able to get it to obtain NwIfInfo service which is set "SERVICE_NWIF_TYPE = WAN" to service property from service repository.

5.8.2 IP address type

This spec defines “IP address type” to be narrowed down the IP address by user bunde of following.

Table 3. IP Address Type

IP Address Type	Discription	Condition
IPV4_GLOBAL	IP address type which means IPv4 global address.	- Implementation subclass of InetAddress instance is equivalent to Inet4Address. - InetAddress.isAnyLocalAddress () is false. - InetAddress.isLinkLocalAddress () is false. - InetAddress.isLoopbackAddress () is false. - InetAddress.isMulticastAddress () is false. - InetAddress.isSiteLocalAddress () is false.
IPV4_PRIVATE	IP address type which means IPv4 private address.	- Implementation subclass of InetAddress instance is equivalent to Inet4Address. - InetAddress.isAnyLocalAddress () is false. - InetAddress.isLinkLocalAddress () is false. - InetAddress.isLoopbackAddress () is false. - InetAddress.isMulticastAddress () is false. - InetAddress.isSiteLocalAddress () is true.
IPV4_LOOPBACK	IP address type which means IPv4 loopback address.	- Implementation subclass of InetAddress instance is equivalent to Inet4Address. - InetAddress.isAnyLocalAddress () is false. - InetAddress.isLinkLocalAddress () is false. - InetAddress.isLoopbackAddress () is true. - InetAddress.isMulticastAddress () is false. - InetAddress.isSiteLocalAddress () is false.
IPV6_GLOBAL	IP address type which means IPv6 global address.	- Implementation subclass of InetAddress instance is equivalent to Inet6Address. - InetAddress.isAnyLocalAddress () is false. - InetAddress.isLinkLocalAddress () is false. - InetAddress.isLoopbackAddress () is false. - InetAddress.isMulticastAddress () is false. - InetAddress.isSiteLocalAddress () is false
IPV6_ LINKLOCAL	IP address type which means IPv6 linklocal address.	- Implementation subclass of InetAddress instance is equivalent to Inet6Address. - InetAddress.isAnyLocalAddress () is false. - InetAddress.isLinkLocalAddress () is true. - InetAddress.isLoopbackAddress () is false. - InetAddress.isMulticastAddress () is false. - InetAddress.isSiteLocalAddress () is false.
IPV6_ LOOPBACK	IP address type which means IPv6 loopback address.	- Implementation subclass of InetAddress instance is equivalent to Inet6Address.

		- InetAddress.isAnyLocalAddress () is false. - InetAddress.isLinkLocalAddress () is false. - InetAddress.isLoopbackAddress () is true. - InetAddress.isMulticastAddress () is false. - InetAddress.isSiteLocalAddress () is false.
IPV6_ SITELOCAL	IP address type which means IPv6 sitelocal address.	- Implementation subclass of InetAddress instance is equivalent to Inet6Address. - InetAddress.isAnyLocalAddress () is false. - InetAddress.isLinkLocalAddress () is false. - InetAddress.isLoopbackAddress () is false. - InetAddress.isMulticastAddress () is false. - InetAddress.isSiteLocalAddress () is true.

If a bundle which wants to check for the IP address of the IPv4 global, the bundle is able to confirm to obtain NwIfInetAddress service which is set "SERVICE_IPADDRESS_TYPE = IPV4_GLOBAL" to service property from service repository.

5.9 Usage

T.B.D.

6 Data Transfer Objects

T.B.D.

7 Javadoc

OSGi Javadoc

14/02/06 17:54

Package Summary		Page
org.osgi.service.nwifinfo		17

Package org.osgi.service.nwifinfo

Interface Summary		Page
NwIfInetAddress	This interface represents an IP address information.	*18*
NwIfInfo	This interface represents network interface information.	*23*
NwIfInfoAdmin	This interface represents a administrator of network interface information.	*27*

Class Summary		Page
NwIfInfoPermission	This class represents execute authority of the bundle which registers NwIfInfo service and NwIfInetAddress service.	*29*

Exception Summary		Page
NwIfInfoException	Exception class that represents a processing failure of NwIfInfo.	*28*

Interface NwIfInetAddress

org.osgi.service.nwifinfo

```
public interface NwIfInetAddress
```

This interface represents an IP address information.
IP address information service is set the following information as service property.

1. SERVICE_NWIF_TYPE : Network Interface Type
2. SERVICE_IPADDRESS_TYPE : IP Address Type
3. SERVICE_IPADDRESS : IP Address
4. SERVICE_SUBNETMASK_LENGTH : Subnet Mask Length(IPv4) or Prefix Length(IPv6)
5. SERVICE_NWIFINFO_ID : Service ID of the interface information service to which it belongs

---

Field Summary		Page
String	**SERVICE_IPADDRESS** The key string of "service.ipaddress" service property.	19
String	**SERVICE_IPADDRESS_TYPE** The key string of "service.ipaddress.type" service property.	19
String	**SERVICE_NWIF_TYPE** The key string of "service.nwif.type" service property.	19
String	**SERVICE_NWIFINFO_ID** The key string of "service.nwifinfo.id" service property.	19
String	**SERVICE_SUBNETMASK_LENGTH** The key string of "service.subnetmask.length" service property.	19

Method Summary		Page
InetAddress	**getInetAddress**() Returns the InetAddress object of this IP address.	20
String	**getIpAddress**() Returns the IP address of "service.ipaddress" service property value.	20
String	**getIpAddressType**() Returns the IP address type of "service.ipaddress.type" service property value.	20
long	**getNwIfInfoId**() Returns the "service.nwifinfo.id" service property value.	21
String	**getNwIfType**() Returns the network interface type of "service.nwif.type" service property value.	19
int	**getSubnetMaskLength**() Returns the "service.subnetmask.length" service property value.	21
void	**remove**() Remove IP address of the relevant network interface from the environment.	21
void	**setIpAddress**(String address) Set the IP address.	20
void	**setSubnetMaskLength**(int length) Set the Subnet Mask Length(IPv4) or Prefix Length(IPv6) The Subnet Mask Length(IPv4) or Prefix Length(IPv6) of the relevant network is changed.	21

Field Detail

SERVICE_NWIF_TYPE

```
public static final String SERVICE_NWIF_TYPE = "service.nwif.type"
```

The key string of "service.nwif.type" service property. Network Interface Type is specified.

SERVICE_IPADDRESS_TYPE

```
public static final String SERVICE_IPADDRESS_TYPE = "service.ipaddress.type"
```

The key string of "service.ipaddress.type" service property. IP Address Type is specified.

SERVICE_IPADDRESS

```
public static final String SERVICE_IPADDRESS = "service.ipaddress"
```

The key string of "service.ipaddress" service property. IP Address is specified.

SERVICE_SUBNETMASK_LENGTH

```
public static final String SERVICE_SUBNETMASK_LENGTH = "service.subnetmask.length"
```

The key string of "service.subnetmask.length" service property. Subnet Mask Length(IPv4) or Prefix Length(IPv6) is specified.

SERVICE_NWIFINFO_ID

```
public static final String SERVICE_NWIFINFO_ID = "service.nwifinfo.id"
```

The key string of "service.nwifinfo.id" service property. Service ID of the interface information service to which it belongs is specified.

Method Detail

getNwIfType

```
String getNwIfType()
```

Returns the network interface type of "service.nwif.type" service property value.

Returns:

Network Interface Type

---

getIpAddressType

```
String getIpAddressType()
```

Returns the IP address type of "service.ipaddress.type" service property value.

Returns:
IP Address Type

---

getIpAddress

```
String getIpAddress()
```

Returns the IP address of "service.ipaddress" service property value.

Returns:
IP Address string

---

getInetAddress

```
InetAddress getInetAddress()
```

Returns the InetAddress object of this IP address.

Returned object is created from "service.ipaddress" service property value.

Returns:
InetAddress

---

setIpAddress

```
void setIpAddress(String address)
          throws NwIfInfoException
```

Set the IP address.

The IP address of the relevant network interface is changed.
If the operation fails for some reason, NwIfInfoException is thrown.
If a security manager exists, NwIfInfoPermission [.,ADMIN] must be required.
If the caller does not have the appropriate NwIfInfoPermission SecurityException is thrown.

Service property value of "service.ipaddress" is changed to set value.
ServiceEvent.MODIFIED is fired with the change of service properties.

Throws:
`NwIfInfoException` - In case that the operation fails for some reason.
`SecurityException` - If the caller does not have the appropriate NwIfInfoPermission, and the Java Runtime Environment supports permissions.

---

getSubnetMaskLength

```
int getSubnetMaskLength()
```

Returns the "service.subnetmask.length" service property value.

Returns:
Subnet Mask Length(IPv4) or Prefix Length(IPv6)

---

setSubnetMaskLength

```
void setSubnetMaskLength(int length)
                  throws NwIfInfoException
```

Set the Subnet Mask Length(IPv4) or Prefix Length(IPv6)

The Subnet Mask Length(IPv4) or Prefix Length(IPv6) of the relevant network is changed.
If the operation fails for some reason, NwIfInfoException is thrown.
If a security manager exists, NwIfInfoPermission [.,ADMIN] must be required.
If the caller does not have the appropriate NwIfInfoPermission SecurityException is thrown.

Service property value of "service.subnetmask.length" is changed to set value.
ServiceEvent.MODIFIED is fired with the change of service properties.

Throws:
NwIfInfoException - In case that the operation fails for some reason.
SecurityException - If the caller does not have the appropriate NwIfInfoPermission, and the Java Runtime Environment supports permissions.

---

getNwIfInfoId

```
long getNwIfInfoId()
```

Returns the "service.nwifinfo.id" service property value.

Returns:
Service ID of the interface information service to which it belongs

---

remove

```
void remove()
     throws NwIfInfoException
```

Remove IP address of the relevant network interface from the environment.

Remove IP address of the relevant network interface from the environment and unregister this service.
If the operation fails for some reason, NwIfInfoException is thrown.
If a security manager exists, NwIfInfoPermission [.,ADMIN] must be required.
If the caller does not have the appropriate NwIfInfoPermission SecurityException is thrown.

IP address represented this service is removed. After that, this service is unregistered.

ServiceEvent.UNREGISTERING is fired with the unregistering this service.

Throws:

`NwIfInfoException` - In case that the operation fails for some reason.
`SecurityException` - If the caller does not have the appropriate NwIfInfoPermission, and the Java Runtime Environment supports permissions.

Interface NwIfInfo

org.osgi.service.nwifinfo

```
public interface NwIfInfo
```

This interface represents network interface information. Network interface information service is set the following information as service property.

1. SERVICE_NWIF_TYPE : Network Interface Type
2. SERVICE_NWIF_DISPLAYNAME : Network Interface Display Name
3. SERVICE_NWIF_NAME : Network Interface Name
4. SERVICE_MACADDRESS : MAC Address

Field Summary		Page
String	**SERVICE_MACADDRESS** The key string of "service.macaddress" service property.	24
String	**SERVICE_NWIF_DISPLAYNAME** The key string of "service.nwif.displayname" service property.	24
String	**SERVICE_NWIF_NAME** The key string of "service.nwif.name" service property.	24
String	**SERVICE_NWIF_TYPE** The key string of "service.nwif.type" service property.	23

Method Summary		Page
void	**addNwIfInetAddress**(String address, int length) Adding IP address.	26
String	**getDisplayName**() Returns the network interface display name of "service.nwif.displayname" service property value.	24
byte[]	**getMacAddress**() Returns the MAC address of "service.macaddress" service property value.	25
String	**getName**() Returns the network interface name of "service.nwif.name" service property value.	25
String	**getNwIfType**() Returns the network interface type of "service.nwif.type" service property value.	24
void	**remove**() Remove Network Interface represented this service from the environment.	25
void	**setDisplayName**(String name) Set the network interface display name The display name of the relevant network is changed.	24

Field Detail

SERVICE_NWIF_TYPE

```
public static final String SERVICE_NWIF_TYPE = "service.nwif.type"
```

The key string of "service.nwif.type" service property. Network Interface Type is specified.

SERVICE_NWIF_DISPLAYNAME

```
public static final String SERVICE_NWIF_DISPLAYNAME = "service.nwif.displayname"
```

The key string of "service.nwif.displayname" service property. Network Interface Display Name is specified.

SERVICE_NWIF_NAME

```
public static final String SERVICE_NWIF_NAME = "service.nwif.name"
```

The key string of "service.nwif.name" service property. Network Interface Name is specified.

SERVICE_MACADDRESS

```
public static final String SERVICE_MACADDRESS = "service.macaddress"
```

The key string of "service.macaddress" service property. MAC Address is specified.

Method Detail

getNwIfType

```
String getNwIfType()
```

Returns the network interface type of "service.nwif.type" service property value.

Returns:
Network Interface Type

---

getDisplayName

```
String getDisplayName()
```

Returns the network interface display name of "service.nwif.displayname" service property value.

Returns:
Network Interface Display Name

---

setDisplayName

```
void setDisplayName(String name)
```

Set the network interface display name

The display name of the relevant network is changed.

Service property value of "service.nwif.displayname" is changed to set value. ServiceEvent.MODIFIED is fired with the change of service properties.

---

getName

```
String getName()
```

Returns the network interface name of "service.nwif.name" service property value.

Returns:
Network Interface Name

---

getMacAddress

```
byte[] getMacAddress()
```

Returns the MAC address of "service.macaddress" service property value.

Returns:
MAC Address

---

remove

```
void remove()
     throws NwIfInfoException
```

Remove Network Interface represented this service from the environment.

Remove Network Interface represented this service from the environment and unregister this service. If the operation fails for some reason, NwIfInfoException is thrown. If a security manager exists, NwIfInfoPermission [.,ADMIN] must be required. If the caller does not have the appropriate NwIfInfoPermission SecurityException is thrown.

Network Interface represented this service is removed. Then, the service represents IP address of the relevant network interface is unregistered. After that, this service is unregistered. ServiceEvent.UNREGISTERING is fired with the unregistering this service.

Throws:
`NwIfInfoException` - In case that the operation fails for some reason.
`SecurityException` - If the caller does not have the appropriate NwIfInfoPermission, and the Java Runtime Environment supports permissions.

---

addNwIfInetAddress

```
void addNwIfInetAddress(String address,
                        int length)
                 throws NwIfInfoException
```

Adding IP address.

Adding IP address of the relevant network interface represented this service to the environment. IP address type is defined from the IP address. If the operation fails for some reason, NwIfInfoException is thrown. If a security manager exists, NwIfInfoPermission [.,ADMIN] must be required. If the caller does not have the appropriate NwIfInfoPermission SecurityException is thrown.

IP address is added. After that NwIfInetAddress service is registered. ServiceEvent.REGISTERING is fired with the registering the service.

Parameters:

`address` - IP Address

`length` - Subnet Mask Length(IPv4) or Prefix Length(IPv6)

Throws:

`NwIfInfoException` - In case that the operation fails for some reason.

`SecurityException` - If the caller does not have the appropriate NwIfInfoPermission, and the Java Runtime Environment supports permissions.

Interface NwIfInfoAdmin

org.osgi.service.nwifinfo

public interface **NwIfInfoAdmin**

This interface represents a administrator of network interface information.

Method Summary		Page
void	**addNwIfInfo**(String nwIfType, String displayName, String name, byte[] macAddress, String address, int length) Method to add the interface information.	27

Method Detail

addNwIfInfo

```
void addNwIfInfo(String nwIfType,
                 String displayName,
                 String name,
                 byte[] macAddress,
                 String address,
                 int length)
          throws NwIfInfoException
```

Method to add the interface information.

A network interface information is added to operating environment.
IP address type is defined from the IP address.
If the MAC address bound network interface is not exist, NwIfInfoException is thrown.
If the operation fails for some reason, NwIfInfoException is thrown.
If a security manager exists, NwIfInfoPermission [.,ADMIN] must be required.
If the caller does not have the appropriate NwIfInfoPermission SecurityException is thrown.

Network Interface Information Service bundle will detect adding network interface information, and register the network interface information service.
After that, the event of ServiceEvent.REGISTERING will be fired.

Parameters:

`nwIfType` - Network Interface Type
`displayName` - Network Interface Display Name
`name` - Network Interface Name
`macAddress` - MAC Address
`address` - String of IP Address
`length` - Subnet Mask Length(IPv4) or Prefix Length(IPv6)

Throws:

`NwIfInfoException` - In case that the operation fails for some reason.
`SecurityException` - If the caller does not have the appropriate NwIfInfoPermission, and the Java Runtime Environment supports permissions.

Class NwIfInfoException

org.osgi.service.nwifinfo

```
java.lang.Object
  └java.lang.Throwable
      └java.lang.Exception
          └org.osgi.service.nwifinfo.NwIfInfoException
```

All Implemented Interfaces:

Serializable

```
public class NwIfInfoException
extends Exception
```

Exception class that represents a processing failure of NwIfInfo.

Constructor Summary	Page
`NwIfInfoException(String message)` Constructor.	28
`NwIfInfoException(String message, Throwable throwable)` Constructor.	28

Constructor Detail

NwIfInfoException

```
public NwIfInfoException(String message)
```

Constructor.

Parameters:

`message` - Exception message

NwIfInfoException

```
public NwIfInfoException(String message,
                         Throwable throwable)
```

Constructor.

Parameters:

`message` - Exception message

Class NwIfInfoPermission

org.osgi.service.nwifinfo

```
java.lang.Object
  └java.security.Permission
      └java.security.BasicPermission
          └org.osgi.service.nwifinfo.NwIfInfoPermission
```

All Implemented Interfaces:
Guard, Serializable

```
public class NwIfInfoPermission
extends BasicPermission
```

This class represents execute authority of the bundle which registers NwIfInfo service and NwIfInetAddress service. This class extends BasicPermission, BasicPermission.implies(java.security.Permission), BasicPermission.equals(Object), BasicPermission.hashCode() and BasicPermission.newPermissionCollection() don't need to be overridden in this class.

The name is a string using the network IF information type and the IP address type. There are three types of name as below. "Str1" and "Str2" is used to represent string example. The length of Str1 and Str2 are greater or equal 1, "." and "*" are not included in the strings.

1. name 1: It is consisted of Str1 and Str2 which is a dot-separated string (i.e. Str1.Str2). "*" is not included in the name. "Network Interface Type" is described as Str1 and "IP Address Type" is described as Str2. This name represents all NwIfInfo service and NwIfInetAddress Service that match the "Network Interface Type" represented Str1 and "IP Address Type" represented Str2.
2. name 2: The name is consisted of Str1 and "*" which is a dot-separated string (i.e. Str1.*).: "Network Interface Type" is described as Str1. This name represents all NwIfInfo service and NwIfInetAddress Service that match the "Network Interface Type" represented Str1.
3. name 3: The name is consisted of only "*". This name represents all NwIfInfo service and NwIfInetAddress Service.

action is a string of below.

1. ADMIN Permission to execute the Add, Set and Delete method for each information

There is no need for this permission to execute of the Get method for each information.

Field Summary		Page
static String	**ADMIN** A string that represents the execution of authority Add, Set and Delete method of the information.	30

Constructor Summary	Page
NwIfInfoPermission(String name, String actions) Constructor.	30

Field Detail

ADMIN

```
public static final String ADMIN = "admin"
```

A string that represents the execution of authority Add, Set and Delete method of the information.

Constructor Detail

NwIfInfoPermission

```
public NwIfInfoPermission(String name,
                          String actions)
```

Constructor.

Parameters:

`name` - The name of access authority
`actions` - Action

8 Considered Alternatives

T.B.D

9 Security Considerations

T.B.D.

10 Document Support

10.1 References

[1]. Bradner, S., Key words for use in RFCs to Indicate Requirement Levels, RFC2119, March 1997.

[2]. Software Requirements & Specifications. Michael Jackson. ISBN 0-201-87712-0

10.2 Author's Address

Name	Shigekuni KONDO
Company	NTT Corporation
Address	1-1, Hikari-no-oka, Yokosuka-shi, 238-0847, Kanagawa, Japan
Voice	+81 46 859 3444
e-mail	kondo.shigekuni@lab.ntt.co.jp

10.3 Acronyms and Abbreviations

10.4 End of Document