

# **PS(Panasonic Security)-LOOKUP**

---

## **Interface Specifications for DLL**

---

Edition 1.0 R12  
Dec. 12, 2017

Panasonic Corporation

With regard to the contents of this document.

- It is prohibited to reproduce part or all in this document.
- Panasonic Corporation reserves the right, at its discretion, to change, modify, add, or remove portions of the contents of this document at any time.

## Revision Record

Version	Revised Date	Content of Revision
1.0 R01	Dec. 21, 2011	First Edition
1.0 R02	Apr. 24, 2012	Added NV200 to supported device.
1.0 R03	Dec. 13, 2012	Correction of Typographical Error
1.0 R04	Mar. 21, 2013	2.1 Added Microsoft® Windows® 8 Pro to System Environment.
1.0 R05	Jul. 18, 2013	Correction of Typographical Error
1.0 R06	Mar. 26, 2014	2.1 Added Microsoft® Windows® 8.1 Pro to System Environment. 2.1 Update System Environment.
1.0 R07	Jul. 14, 2014	Added NV300 to supported device.
1.0 R08	Oct. 14, 2015	2.1 Added Microsoft® Windows® 10 Pro to System Environment.
1.0 R09	Jan. 22, 2016	2.1 Added Microsoft® Windows Server® 2012 Standard to System Environment. 2.1 Added Microsoft® Windows Server® 2012 R2 Standard to System Environment. 2.1 Deleted Microsoft® Windows Server® 2003 Standard 64 bit Edition to System Environment. 2.1 Deleted Microsoft® Windows Server® 2003 Standard 32 bit Edition to System Environment. 2.1 Deleted Microsoft® Windows Server® 2003 Enterprise 64 bit Edition to System Environment. 2.1 Deleted Microsoft® Windows Server® 2003 Enterprise 32 bit Edition to System Environment.
1.0 R10	Feb. 8, 2017	Added NX400 to supported device.
1.0 R11	Jun. 19, 2017	Change company name. Added NX200 to supported device. 2.1 Deleted Microsoft® Windows® XP Professional SP3 from System Environment. 2.1 Deleted Microsoft® Windows Vista® Business SP2 32 bit Edition from System Environment.
1.0 R12	Dec. 12, 2017	2.1 Update System Environment.

## **INDEX**

1. Preface .....	1
1.1. What's PS-LOOKUP .....	1
1.2. Trademarks and Registered Trademarks .....	1
1.3. Limitation of liability .....	1
1.4. Abbreviations .....	2
1.5. Structures of PS-LOOKUP .....	2
1.6. Overview of Functions .....	3
1.7. Function List .....	4
1.8. Compatible chart by models .....	5
2. System requirements .....	6
2.1. System Environment .....	6
2.2. Development Environment .....	7
2.3. Supported Panasonic Products .....	7
3. Setup .....	8
3.1. Product .....	8
3.2. Install .....	9
3.3. UnInstall .....	9
3.4. Restrictions .....	9
4. Overview of Library .....	10
4.1. Connect to the device .....	10
4.2. Steps to detect device with PS-LOOKUP .....	11
5. Details of DLL Class and Method .....	12
5.1. Class .....	12
5.1.1. Class Definition .....	12
5.1.2. Class Diagram .....	12
5.1.2.1. Global Function .....	13
5.1.2.2. IPSLookup .....	13
5.1.2.3. ILookupListener .....	13
5.2. Global Function .....	14
5.2.1. GetIPSLookup .....	14
5.2.2. DeleteIPSLookup .....	16
5.3. PSLookup Group .....	18
5.3.1. Application Listener .....	18
5.3.1.1. OnDevLookup .....	18
5.3.1.2. SetDevLookupListener .....	20
5.3.1.3. OnError .....	22
5.3.1.4. SetErrListener .....	24
6. Operation Procedure and Sequence .....	26
6.1. PSLookup .....	26
6.1.1. Operation Procedure .....	26
6.1.2. Sequence .....	27
7. Error Code List .....	29

## 1. Preface

### 1.1. What's PS-LOOKUP

PS-LOOKUP is the software library which is provided to help to develop an application software that is for detecting Panasonic Corporation made security products (Network cameras, Network Disk Recorders(NV series ,NX series ), Network Interface Units) and notify to application.

PS-LOOKUP is provided as the dynamic link library (DLL).

### 1.2. Trademarks and Registered Trademarks

Microsoft and Windows are registered trademarks of Microsoft Corporation in U.S. and/or other countries. Other names of companies and product contained in these operating instructions may be trademarks or registered trademarks of their respective owners.

### 1.3. Limitation of liability

- PS-LOOKUP provides interfaces for the application software to detect Panasonic Corporation made security products (Network cameras, Network Disk Recorders(NV series ,NX series ), Network Interface Units), and is not designed to protect against "theft" or "crime" independently.

- The provided sample programs are designed to instruct users how to use the SDK. They are not developed for the purpose of actual surveillance system.

- In not event shall Panasonic Corporation be liable to any party or any person, except for replacement or reasonable maintenance of the product, for the cases, including but not limited to below;

[1] Any damage and loss, including without limitation, direct or indirect, special, sequential or exemplary, arising out of or relating to the product;

[2] Personal injury or any damage caused by inappropriate use or neglect operation of the user;

[3] Any problems, consequential inconvenience, or loss or damage, arising out of the reverse compiling or reverse engineering of the product;

[4] Any loss or damage, or claims arising out from loss or leak of PS data including video data in the PC;

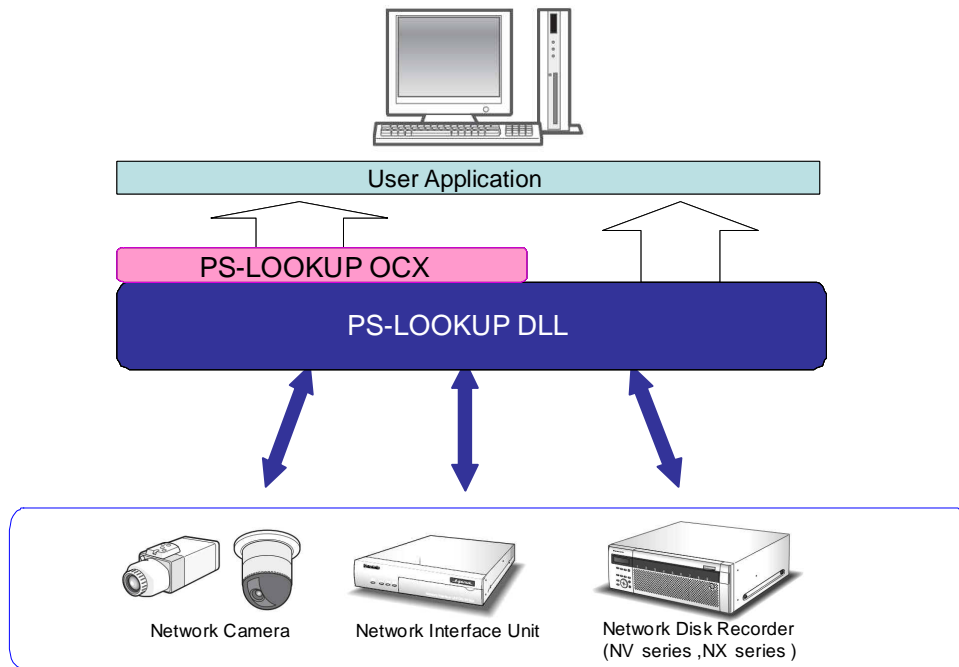
[5] Any claim or action for damages, brought by any person or organization being a photogenic subject, due to violation of privacy with the result of that surveillance camera's picture, including saved data, for some reason, becomes public or is used for the purpose other than surveillance;

#### 1.4. Abbreviations

The following abbreviations are used in these operating instructions.  
Microsoft® Visual C++ 2005 is described as Visual C++ 2005.

#### 1.5. Structures of PS-LOOKUP

Figure 1-1 shows the PS-LOOKUP structures.



**Figure 1-1 PS-LOOKUP Structures**

- \* Hereafter Network Camera is referred as Camera.
- \* Hereafter Network Interface Unit is referred as Encoder.
- \* Hereafter Network Disk Recorder is referred as NWDR.
- \* Hereafter Network Disk Recorder (NV series) is referred as NV series.
- \* Hereafter Network Disk Recorder (NX series) is referred as NX series.
- \* Hereafter Digital Disk Recorder (HD300 series) is referred as HD300.
- \* Hereafter Digital Disk Recorder (HD600 series, HD700 series) is referred as HD600/700.

## 1.6. Overview of Functions

Table 1-1 shows the overview of functions.

**Table 1-1 Function Overview**

No.	Overview	Reference
1	Create and Delete the IPSLookup instance	Global function
2	Notify an application of device detection	PSLookup Group

### 1.7. Function List

Table 1-2 shows the functions provided in this PS-LOOKUP.

**Table 1-2 List of Functions**

#### Method

No.	Class	Method	Overview	Reference
<i>Global Function</i>				
1	-	GetIPSLookup	Create the IPSLookup instance.	
2	-	DeleteIPSLookup	Delete the IPSLookup instance.	
<i>PSLookup Group</i>				
1	IPSLookup	SetDevLookupListener	Register OnDevLookup Listener function for an application.	
2	IPSLookup	SetErrListener	Register OnError Listener function for an application.	

#### Application Listener

No.	Class	Method	Overview	Reference
<i>PSLookup Group</i>				
1	ILookupListener	OnDevLookup	The OnDevLookup notification function is to pass an received device detection information from PS-LOOKUP to a specified application.	
2	ILookupListener	OnError	The OnError notification function is to pass an error code from PS-LOOKUP to a specified application.	

## 1.8. Compatible chart by models

The following list shows the compatible chart by models.

**Table 1-3 Compatible Chart by Models**

No.	Method	Camera	NWDR	HD300	HD600/700	Encoder	NX Series	remarks
	<i>PSLookup グループ</i>							
1	SetDevLookupListener	Yes	*1	-	-	Yes	Yes	
2	SetErrListener	Yes	*1	-	-	Yes	Yes	

\*1 : Only NV200 , NV300 support this function.

## 2. System requirements

### 2.1. System Environment

The following table shows the PC specification that is needed for using PS-LOOKUP.

**Table 2-1 OS**

OS	Microsoft® Windows® 7 Professional SP1 32 bit Edition
	Microsoft® Windows® 7 Professional SP1 64 bit Edition
	Microsoft® Windows® 8 Pro 32 bit Edition(*1)
	Microsoft® Windows® 8 Pro 64 bit Edition(*1)
	Microsoft® Windows® 8.1 Pro 32 bit Edition(*1)
	Microsoft® Windows® 8.1 Pro 64 bit Edition(*1)
	Microsoft® Windows® 10 Pro 32 bit Edition
	Microsoft® Windows® 10 Pro 64 bit Edition
	Microsoft® Windows Server® 2008 R2 Standard SP1
	Microsoft® Windows Server® 2008 R2 Enterprise SP1
	Microsoft® Windows Server® 2012 Standard
	Microsoft® Windows Server® 2012 R2 Standard

**Table 2-2 System requirements**

Processor(*2)	Intel® Core™2 Quad 2.66GHz or more
Memory	2.0GB or more
Hard drive	10GB or more
LAN	100Mbps or more

**Table 2-3 Recommended system requirements**

Processor(*2)	Intel® Core™i7-4790
Memory	8.0GB or more
Hard drive	10GB or more
LAN	100Mbps or more

**(\*1) Modern UI is not supported.**

**(\*2) When displaying video on multiple screens, please use the recommended system requirements.**

## 2.2. Development Environment

The following table shows the development environment list that is supported by PS-LOOKUP DLL.

**Table 2-4 Development Environment**

Target	Development Tool
PS-LOOKUP DLL	VisualC++ 2005 SP1

## 2.3. Supported Panasonic Products

Please refer to [**PS-API Supported Product List for English**] document.

### 3. Setup

#### 3.1. Product

Please refer to [PS-API Installation Guide for DLL] document.

**Table 3-1 The Overview of Files**

[DLL folder]

Directory Name	Objective
PS-API¥ For Development	Header files / lib files For development. <b>NOT REDISTRIBUTABLE.</b>
PS-API¥ Redistributable	DLL files Redistributable.
PS-API¥ Setup	Installer for DirectShow filters. For development. <b>NOT REDISTRIBUTABLE.</b>
PS-API¥ Tool	Test tool. By using this tool, you can confirm connecting to a target device with PS-API. <b>NOT REDISTRIBUTABLE.</b>
PS-API¥ Document	The users manual. (This document) <b>NOT REDISTRIBUTABLE.</b>

Directory Name	Objective
PS-ALARM¥ For Development	Header files / lib files For development. <b>NOT REDISTRIBUTABLE.</b>
PS-ALARM¥ Redistributable	DLL files Redistributable.
PS-ALARM¥ Document	The users manual. <b>NOT REDISTRIBUTABLE.</b>

Directory Name	Objective
PS-LOOKUP¥ For Development	Header files / lib files For development. <b>NOT REDISTRIBUTABLE.</b>
PS-LOOKUP¥ Redistributable	DLL files Redistributable.
PS-LOOKUP¥ Document	The users manual. <b>NOT REDISTRIBUTABLE.</b>

[DLL-Sample folder]

Directory Name	Objective
Sample Program	Sample Programs. <b>NOT REDISTRIBUTABLE.</b>

### 3.2. Install

Please refer to [**PS-API Installation Guide for DLL**] document.

### 3.3. UnInstall

Please refer to [**PS-API Installation Guide for DLL**] document.

### 3.4. Restrictions

- (1) Standardize the time zone setting of devices in your system.
- (2) Do not use the Power Schemes or System Standby mode of Windows®.
- (3) PS-LOOKUP instance is NOT thread safe.
- (4) Use Multibyte Character Set.
- (5) The maximum number of device detection that the PC can receive depends on the PC performance that receives device information.

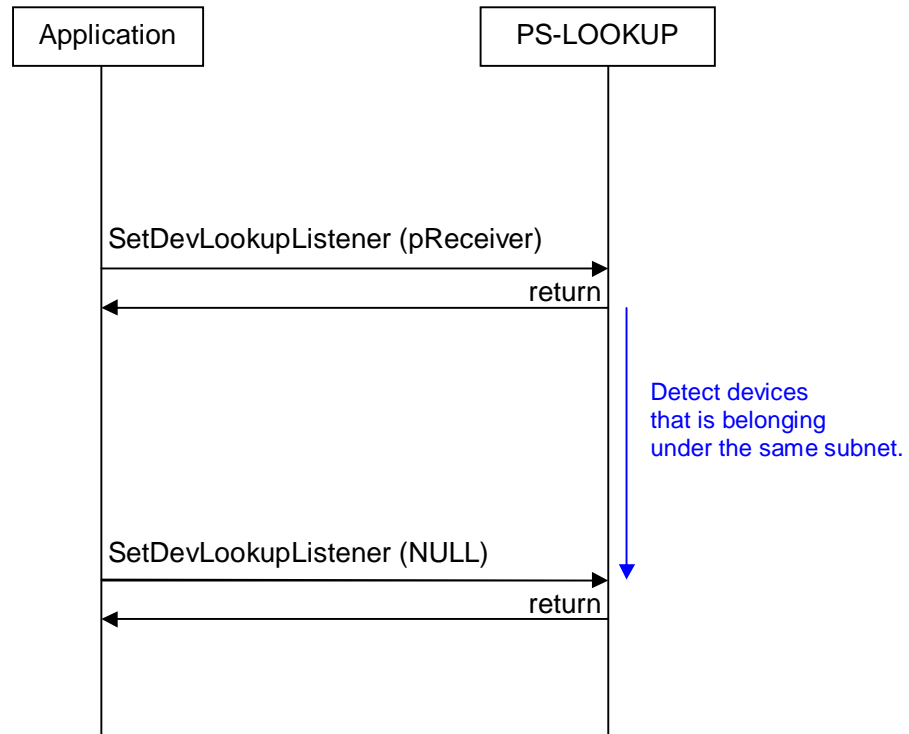
## 4. Overview of Library

### 4.1. Connect to the device

When the application detects devices by using PS-LOOKUP, it is necessary that SetDevLookupListener method is called to register the listener class.

After registered the listener class, detected device's information is notified via OnDevLookup method. When stop detecting devices, set NULL to the listener class by using SetDevLookupListener method.

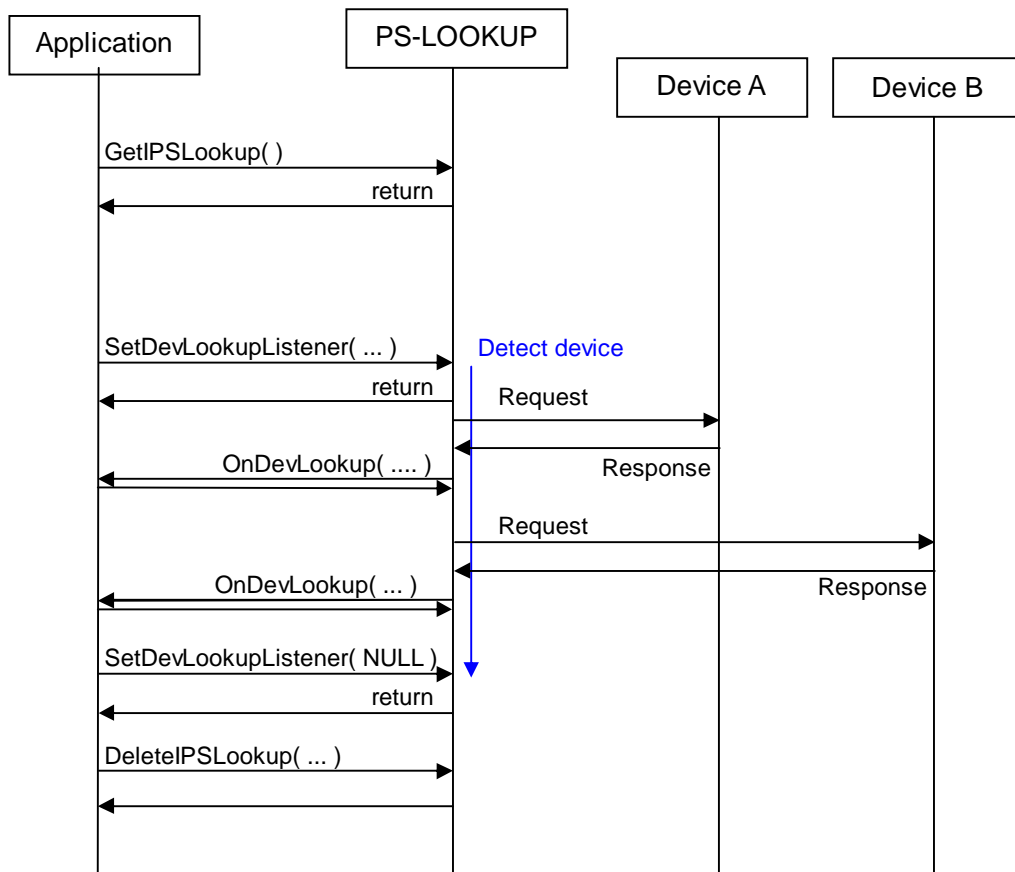
\* Don't delete the registered listener object by your application.



**Figure 4-1 Connect to The Device**

#### 4.2. Steps to detect device with PS-LOOKUP

The following chart shows the flow of creating instance, detecting devices, and stop.



**Figure 4-2 Steps to detect devices with PS-LOOKUP**

## 5. Details of DLL Class and Method

This chapter describes the detail specification of this library.

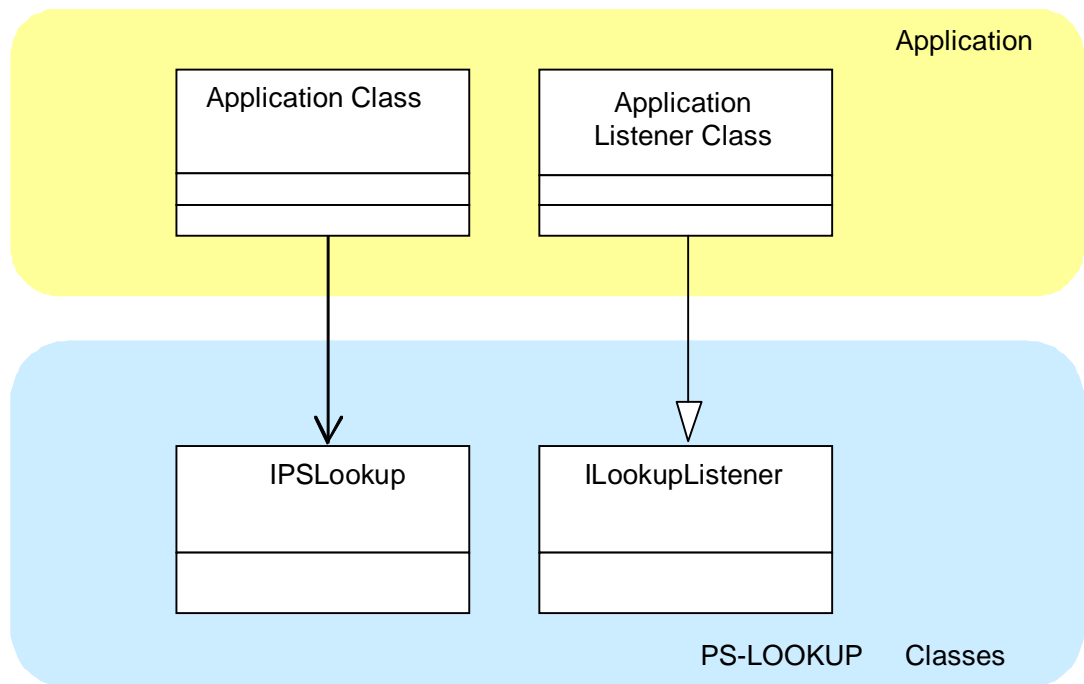
### 5.1. Class

#### 5.1.1. Class Definition

**Table 5-1 Class Definition**

No.	Class Name	Overview
1	IPSLookup	It is the interface class for using PS-LOOKUP functions from an application. Control a device by calling the method of IPSLookup.
2	ILookupListener	It is the interface class for receiving the notification from PS-LOOKUP. The application can implement the process for the notification by inheriting this class and by implementing the notification method. Also the application can receive the notification by registering the instance that is implemented by the application to IPSLookup.

#### 5.1.2. Class Diagram



**Figure 5-1 Class Diagram**

#### 5.1.2.1. Global Function

```
IPSLookup*    GetIPSLookup( );  
void          DeleteIPSLookup(IPSLookup*  ilookup);
```

#### 5.1.2.2. IPSLookup

```
long  SetDevLookupListener(ILookupListener* pReceiver);  
long  SetErrListener(ILookupListener* pReceiver);
```

#### 5.1.2.3. ILookupListener

```
virtual void  OnDevLookup (const char*  macAddr,  
                           const char*  ipAddr,  
                           const char*  ipv6Addr,  
                           long  portNo,  
                           const char*  camName,  
                           const char*  modelName) = 0;  
virtual void  OnError(long  errorCode, const char*  description) = 0;
```

## 5.2. Global Function

### 5.2.1. GetIPSLookup

<b>Class</b>	-
<b>Function</b>	<b>GetIPSLookup</b>
IPSLookup*	GetIPSLookup ( );

---

#### Description

Create the IPSLookup instance.

---

#### Argument

None

---

#### Return value

IPSLookup \* Pointer of created IPSLookup instance

---

#### Error

---

## Note

---

## Sequence

---

## Sample program code

---

[Visual C++ 2005] ..¥Sample Program¥PS-LOOKUP¥Visual C++ 2005¥301\_PSLookup

## Reference

---

### 5.2.2. DeleteIPSLookup

**Class** -

---

**Function** DeleteIPSLookup

---

void DeleteIPSLookup (IPSLookup \* IPSLookup)

---

#### Description

Delete the IPSLookup instance.

---

#### Argument

IPSLookup	Pointer of IPSLookup instance	Specify the pointer of the IPSLookup instance.
-----------	-------------------------------	--

---

#### Return value

None

---

#### Error

---

## Note

---

## Sequence

---

## Sample program code

---

[Visual C++ 2005] ..¥Sample Program¥PS-LOOKUP¥Visual C++ 2005¥301\_PSLookup

## Reference

---

### 5.3. PSLookup Group

#### 5.3.1. Application Listener

##### 5.3.1.1. OnDevLookup

Class	ILookupListener
-------	-----------------

Listener	OnDevLookup
----------	-------------

void	OnDevLookup ( const char* macAddr, const char* ipAddr, const char* ipv6Addr, long portNo, const char* camName, const char* modelName );
------	--

#### Description

Notify to detect device to the specified application.

Application needs to create the listener class that inherits ILookupListener and to implement OnDevLookup method.

This functions is not supported by recorders(HD300, NWDR except NV series and NX series, HD600/700).

#### Argument

macAddr	Character strings (24 or less characters)	The MAC address of a detected device.
ipAddr	Character strings (16 or less characters)	The IPv4 address of a detected device.
ipv6Addr	Character strings (16 or less characters)	The IPv6 address of a detected device. When there is no IPv6 address, this area is set to ""(empty).
portNo	1 – 65535	The HTTP port number of a detected device.
camName	Character strings (32 or less characters)	The camera name of a detected device. When there is no camera name setting, this area is set to "" (empty). Depending on the model, camera name cannot be acquired even if camera name is set. The maximum length of camera name is 15.
modelName	Character strings (16 or less characters)	The model of a detected device.

**\* Please use these parameters in only OnDevLookup function. When finish OnDevLookup function, memory for these parameters are released.**

**Return value**

---

None

**Error**

---

**Note**

---

**Sequence**

---

**Sample program code**

---

[Visual C++ 2005] ..¥Sample Program¥PS-LOOKUP¥Visual C++ 2005¥301\_PSLookup

**Reference**

---

### 5.3.1.2. SetDevLookupListener

**Class**                      **IPSLookup**

**Listener**                      **SetDevLookupListener**

```
long      SetDevLookupListener (
                                   ILookupListener*   pReceiver
                                   );
```

#### **Description**

Set the instance of listener class that implement “5.3.1.1 OnDevLookup “ .

After setting the listener, PS-LOOKUP can notify any device detection information by calling 5.3.1.1 OnDevLookup “, when PS-LOOKUP detect a device.

This functions is not supported by recorders(HD300, NWDR except NV series and NX series, HD600/700).

#### **Argument**

pReceiver	Pointer	Pointer for Listener class.
	NULL	If pReceiver is set to NULL, PS-LOOKUP unregister the Listener class.

#### **Return value**

0	Success to register an application listener.
Except 0	Error code

#### **Error**

Error is defined by the return value.

**Note**

---

Don't delete the registered listener object by your application.

**Sequence**

---

**Sample program code**

---

[Visual C++ 2005] ..¥Sample Program¥PS-LOOKUP¥Visual C++ 2005¥301\_PSLookup

**Reference**

---

#### 5.3.1.3. OnError

<b>Class</b>	<b>ILookupListener</b>		
<b>Listener</b>	<b>OnError</b>		
void	OnError( long   errorCode, const char*   description );		
<b>Description</b>			
Notify the error to the specified application. Application needs to create the listener class that inherits ILookupListener and to implement OnError method.			
<b>Argument</b>			
	errorCode	Negative value	Error code number
	description	Character strings	Detailed error description <b>* This parameter is valid in the inside of OnError function. When OnError function is finished, this memory is freed.</b>
<b>Return value</b>			
	None		
<b>Error</b>			

**Note**

---

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.3.1.4. SetErrListener

**Class**                      **IPSLookup**

**Listener**                  **SetErrListener**

```
long      SetErrListener(  
                                ILookupListener *   pReceiver  
                                );
```

#### **Description**

Set the instance of listener class that implement “5.3.1.3 OnError” .  
After setting the listener, PS-LOOKUP can notify any error information by calling “5.3.1.3 OnError”, when it happens.

#### **Argument**

pReceiver	Pointer	Pointer for Listener class.
	NULL	If pReceiver is set to NULL, PS-LOOKUP unregister the Listener class.

#### **Return value**

0	Success to register an application listener.
Except 0	Error code

#### **Error**

Error is defined by the return value.

**Note**

---

Don't delete the registered listener object by your application.

**Sequence**

---

**Sample program code**

---

**Reference**

---

## 6. Operation Procedure and Sequence

### 6.1. PSLookup

#### 6.1.1. Operation Procedure

##### Start detecting device

No.	Property / Method	Parameter	Description
1	GetIPSLookup	-	Create instance of IPSLookup.
2	SetDevLookupListener	Listener class (ILookupListener*)	Set the instance of listener class.
-	(OnDevLookup)	MAC address, IPv4 address, IPv6 address, port number, camera name, model (const char*, const char*, const char*, long, const char*, const char*)	Notify the specified application of device detection.

##### Stop detecting device

No.	Property / Method	Parameter	Description
3	SetDevLookupListener	NULL	Set NULL, and PS-LOOKUP unregister the Listener class.
4	DeleteIPSLookup	Pointer of IPSLookup instance (IPSLookup*)	Delete the IPSLookup instance.

### 6.1.2. Sequence

#### Start detecting device

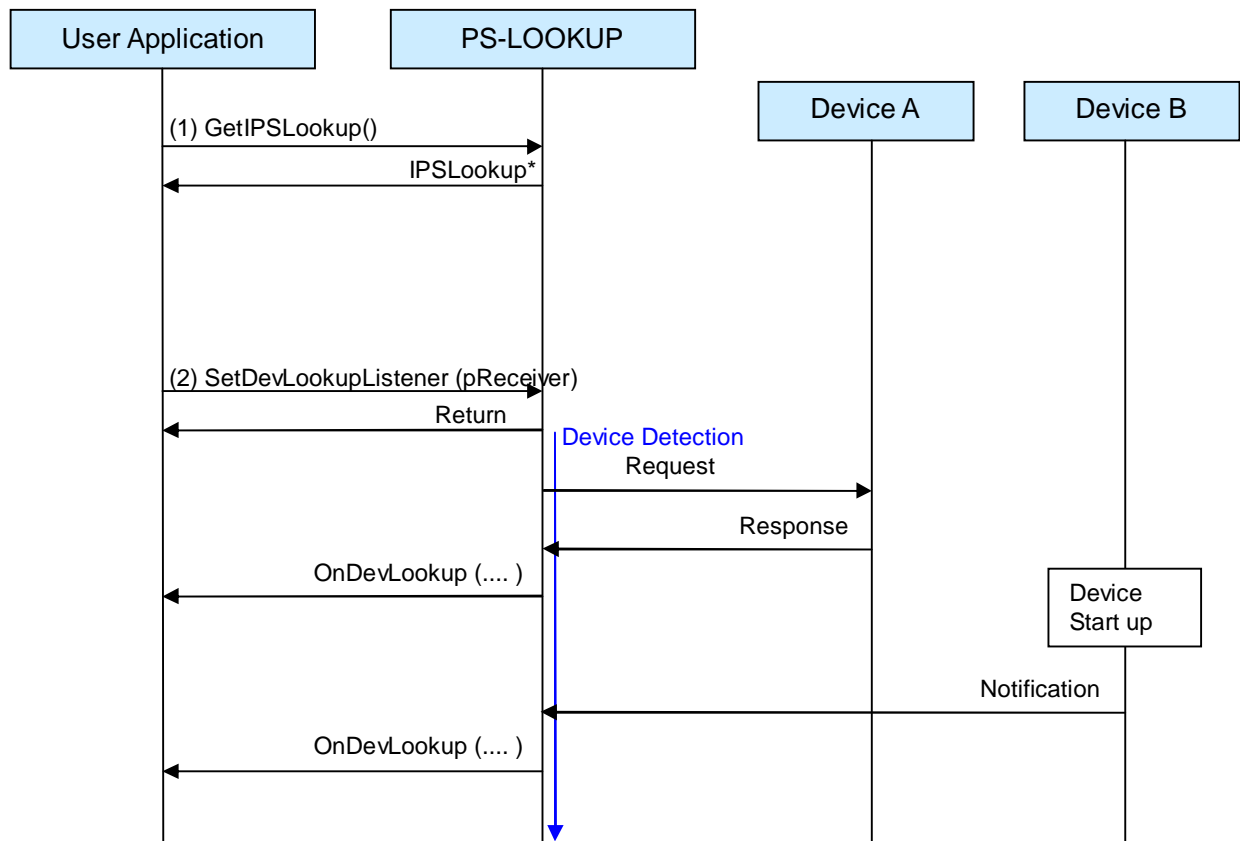


Figure 6-1 Start detecting device

## Stop detecting device

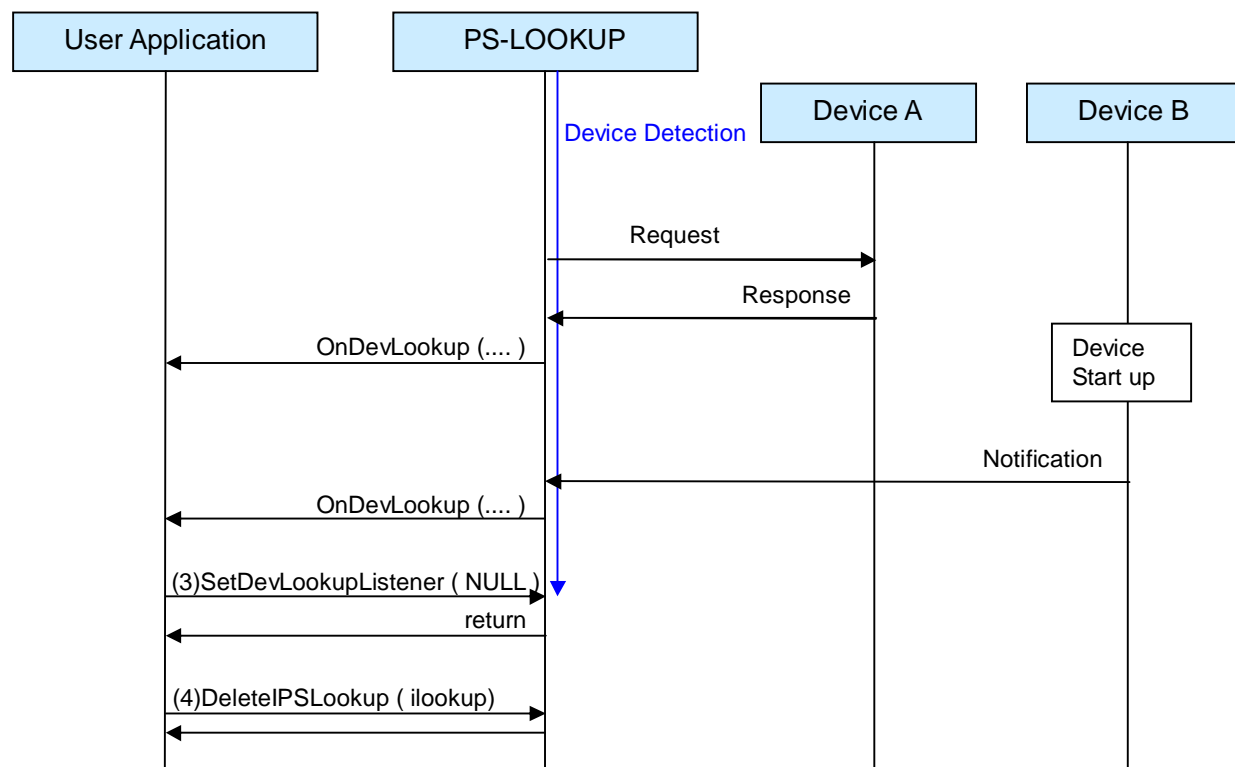


Figure 6-2 Stop detecting device

## 7. Error Code List

The error code is defined by the following format. (decimal, 8digits)

-D<sub>1</sub>A<sub>1</sub>C<sub>1</sub>M<sub>1</sub>M<sub>2</sub>Z<sub>1</sub>Z<sub>2</sub>Z<sub>3</sub>

-D <sub>1</sub> Category	A <sub>1</sub> Level	C <sub>1</sub> Source	M <sub>1</sub> M <sub>2</sub>	Z <sub>1</sub> Z <sub>2</sub> Z <sub>3</sub> Code	Error Description
-6 PS-LOOKUP	1:error 2:warning	1:PS-LOOKUP error	(internal use)	103	Internal process error (WSAStartup)
				104	Internal process error (Get address information)
				105	Internal process error (Create socket)
				106	Internal process error (Bind socket)
				107	Internal process error (Select socket)
				109	Internal process error (Receive device information)
				110	Internal process error (Sent Information Request)
				201	SDK internal error
				202	SDK internal error
				203	SDK internal error
				204	SDK internal error
				401	In detecting devices
				501	Fatal error (Create thread)
				502	Fatal error (Create object)
				503	Fatal error (Memory allocation)