

# PS(Panasonic Security)-LOOKUP

---

## インターフェース仕様書 DLL 版

---

Edition 1.0 R12  
Dec. 12, 2017

パナソニック株式会社  
コネクティッドソリューションズ社

本書の内容について.

- ・本書の一部または全部を複製することを禁じます。
- ・本書の内容および仕様は予告無く変更することがありますがご了承ください。

## 変更履歴

Version	変更日	変更内容
1.0 R01	Dec. 21, 2011	初版
1.0 R02	Apr. 24, 2012	NV200 を対応機器に追加しました。
1.0 R03	Dec. 13, 2012	誤記訂正
1.0 R04	Mar. 21, 2013	2.1 ハードウェア環境の OS に Microsoft® Windows® 8 Pro を追加
1.0 R05	Jul. 18, 2013	誤記訂正
1.0 R06	Mar. 26, 2014	2.1 ハードウェア環境の OS に Microsoft® Windows® 8.1 Pro を追加 2.1 ハードウェア環境を更新
1.0 R07	Jul. 14, 2014	NV300 を対応機器に追加しました。
1.0 R08	Oct. 14, 2015	2.1 ハードウェア環境の OS に Microsoft® Windows® 10 Pro を追加
1.0 R09	Jan. 22, 2016	2.1 ハードウェア環境の OS に Microsoft® Windows Server® 2012 Standard を追加 2.1 ハードウェア環境の OS に Microsoft® Windows Server® 2012 R2 Standard を追加 2.1 ハードウェア環境の OS から Microsoft® Windows Server® 2003 Standard 64 ビット版を削除 2.1 ハードウェア環境の OS から Microsoft® Windows Server® 2003 Standard 32 ビット版を削除 2.1 ハードウェア環境の OS から Microsoft® Windows Server® 2003 Enterprise 64 ビット版を削除 2.1 ハードウェア環境の OS から Microsoft® Windows Server® 2003 Enterprise 32 ビット版を削除
1.0 R10	Feb. 8, 2017	NX400 を対応機器に追加しました。
1.0 R11	Jun. 19, 2017	社名変更 NX200 を対応機器に追加 2.1 ハードウェア環境の OS から Microsoft® Windows® XP Professional SP3 を削除 2.1 ハードウェア環境の OS から Microsoft® Windows Vista® Business SP2 32 ビット版を削除
1.0 R12	Dec. 12, 2017	2.1 ハードウェア環境を更新

## INDEX

1.	はじめに	1
1.1.	PS-LOOKUP について	1
1.2.	商標および登録商標について	1
1.3.	免責について	1
1.4.	本書での略称	2
1.5.	PS-LOOKUP の構成	2
1.6.	機能概要	3
1.7.	機能一覧	4
1.8.	機種別対応メソッド一覧	5
2.	動作環境	6
2.1.	ハードウェア環境	6
2.2.	開発環境	7
2.3.	サポートする パナソニック製品	7
3.	セットアップ	8
3.1.	製品構成	8
3.2.	インストール方法	10
3.3.	アンインストール方法	10
3.4.	制限事項	10
4.	ライブラリ概要	11
4.1.	機器への接続	11
4.2.	PS-LOOKUP を利用した機器検出の流れ	12
5.	DLL クラス・メソッド詳細	13
5.1.	クラス	13
5.1.1.	クラス定義	13
5.1.2.	クラス構造	13
5.1.2.1.	グローバル関数	14
5.1.2.2.	IPSLookup	14
5.1.2.3.	ILookupListener	14
5.2.	グローバル関数	15
5.2.1.	GetIPSLookup	15
5.2.2.	DeleteIPSLookup	17
5.3.	PSLookup グループ	19
5.3.1.	アプリケーション リスナー	19
5.3.1.1.	OnDevLookup	19
5.3.1.2.	SetDevLookupListener	21
5.3.1.3.	OnError	23
5.3.1.4.	SetErrListener	25
6.	操作手順/シーケンス	27
6.1.	PSLookup	27
6.1.1.	操作手順	27
6.1.2.	シーケンス	28
7.	エラーコード一覧	30

## 1. はじめに

### 1.1. PS-LOOKUP について

PS-LOOKUP はネットワーク上に接続されたパナソニック株式会社コネクティッドソリューションズ社製の映像監視セキュリティ製品(ネットワークカメラ、ネットワークインターフェースユニット、ネットワークディスクレコーダ(NV Series、NX Series))を検出し、アプリケーションに通知するために提供されるソフトウェアライブラリです。PS-LOOKUP は、ダイナミックリンクライブラリ(DLL)の形態で提供されます。

### 1.2. 商標および登録商標について

Microsoft®, Windows®は、米国Microsoft Corporation の米国およびその他の国における登録商標です。その他、本文中の社名や商品名は、各社の登録商標または商標です。

### 1.3. 免責について

・PS-LOOKUPはネットワーク上に接続されたパナソニック株式会社コネクティッドソリューションズ社製のセキュリティ製品(ネットワークカメラ、ネットワークインターフェースユニット、ネットワークディスクレコーダ(NV Series、NX Series))を検出し、アプリケーションに通知するインターフェースを提供するものであり、PS-LOOKUP 単独で犯罪などを防止するものではありません。

・付属のサンプルプログラムは PS-LOOKUP の使用方法を記述したものであり、実際の監視運用を目的として作成されたものではありません。

・弊社は、いかなる場合も以下に関して一切の責任を負わないものとします。

- (1) 本製品に関連して直接または間接に発生した、偶発的、特殊、または結果的損害・被害
- (2) お客様の故意や誤使用、不注意による障害または本製品の損傷など
- (3) お客様による本商品の逆コンパイル、逆アセンブルが行われた場合、それに起因するかどうかにかかわらず、発生した一切の故障または不具合
- (4) セキュリティデバイスから受信したデータ等のパソコンにおける消失、あるいは漏洩等によるいかなる損害、クレームなど
- (5) ネットワーク上からの不正アクセスなど悪意を持った第三者による画像データ、音声データ、認証情報(ユーザー名、パスワード)の漏えいなどによるいかなる損害、クレームなど

#### 1.4. 本書での略称

本書では、下記の通り略記しています。

Microsoft® Visual C++ 2005 を Visual C++ 2005 と記述しています。

#### 1.5. PS-LOOKUP の構成

Figure 1-1 に PS-LOOKUP の全体構成を示します。

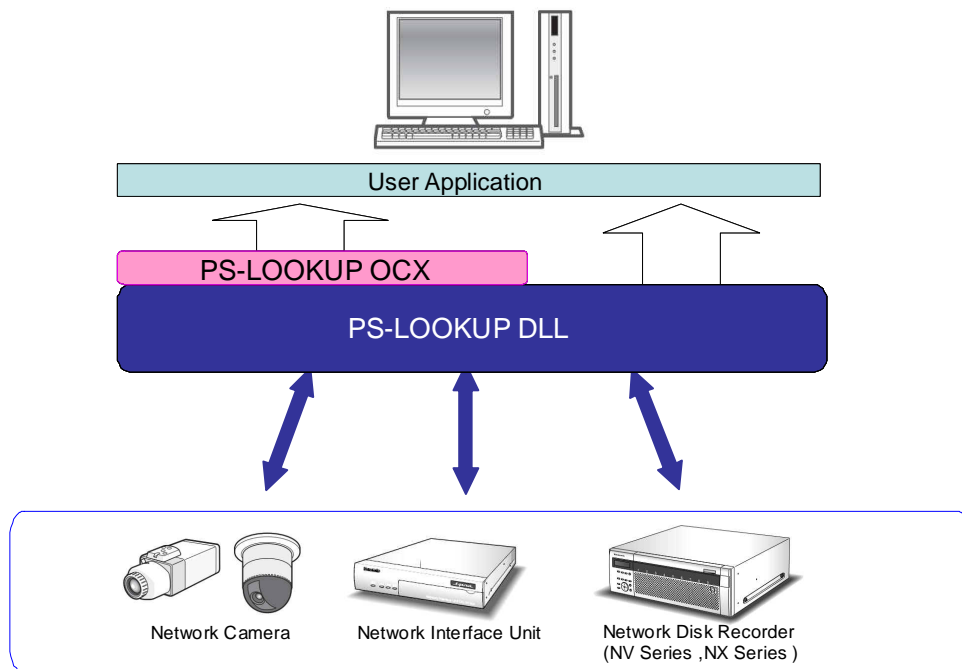


Figure 1-1 PS-LOOKUP structures

- \* 以降、Network Camera は Camera または NW カメラと表記します。
- \* 以降、Network Interface Unit は Encoder またはエンコーダーと表記します。
- \* 以降、Network Disk Recorder は NWDR と表記します。
- \* 以降、Network Disk Recorder (NV シリーズ)は NV Series と表記します。
- \* 以降、Network Disk Recorder (NX シリーズ)は NX Series と表記します。
- \* 以降、Digital Disk Recorder(HD300 シリーズ)は HD300 と表記します。
- \* 以降、Digital Disk Recorder (HD600 シリーズ、HD700 シリーズ)は HD600/700 と表記します。

## 1.6. 機能概要

Table 1-1 Function Overview

に PS-LOOKUP の機能概要を示します。

Table 1-1 Function Overview

No.	Overview	Reference
1	IPSLookup インスタンスの生成・破棄	グローバル関数
2	機器検出通知	PS Lookup グループ

## 1.7. 機能一覧

Table 1-2 List of Functions

に PS-LOOKUP の機能一覧を示します。

Table 1-2 List of Functions

### Method

No.	Class	Method	Overview	Reference
<i>グローバル関数</i>				
1	-	GetIPSLookup	IPSLookup インスタンスを生成します。	
2	-	DeleteIPSLookup	IPSLookup インスタンスを破棄します。	
<i>PSLookup グループ</i>				
1	IPSLookup	SetDevLookupListener	OnDevLookup リスナーを設定します。	
2	IPSLookup	SetErrListener	OnError リスナーを設定します。	

### Application Listener

No.	Class	Method	Overview	Reference
<i>PSLookup グループ</i>				
1	ILookupListener	OnDevLookup	機器検出情報をアプリケーションに通知します。	
2	ILookupListener	OnError	指定したアプリケーションにエラーを通知します。	

## 1.8. 機種別対応メソッド一覧

機種別の対応メソッド一覧を下記に示します。(IPSLookup クラスについて記載)

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 : NV200、NV300 のみサポートしています。

## 2. 動作環境

### 2.1. ハードウェア環境

PS-LOOKUP を使用するために必要な PC のスペックを示します。

**Table 2-1 OS**

OS	Microsoft® Windows® 7 Professional SP1 64 ビット版
	Microsoft® Windows® 7 Professional SP1 32 ビット版
	Microsoft® Windows® 8 Pro 32 ビット版(*1)
	Microsoft® Windows® 8 Pro 64 ビット版(*1)
	Microsoft® Windows® 8.1 Pro 32 ビット版(*1)
	Microsoft® Windows® 8.1 Pro 64 ビット版(*1)
	Microsoft® Windows® 10 Pro 32 ビット版
	Microsoft® Windows® 10 Pro 64 ビット版
	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 動作環境**

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

**Table 2-3 推奨環境**

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

(\*1) Modern UI には対応していません。

(\*2) 多画面表示を行う場合は、推奨環境をご使用ください。

## 2.2. 開発環境

PS-LOOKUP が対応しているアプリケーション開発ツールを以下に示します。

**Table 2-4 Development Environment**

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

## 2.3. サポートする パナソニック製品

同梱の「PS-API Supported Product List for Japanese」を参照してください。

### 3. セットアップ

#### 3.1. 製品構成

同梱の「PS-API Installation Guide for DLL」を参照してください。

Table 3-1 The Overview of Files

[DLL フォルダ]

フォルダ名	用途
PS-API¥ For Development	ヘッダーファイル/lib ファイル。 開発用です。 <b>再配布できません。</b>
PS-API¥ Redistributable	DLL ファイル。 再配布可能ファイルです。
PS-API¥ Setup	DirectShow フィルターのインストーラー。 開発用です。 <b>再配布できません。</b>
PS-API¥ Tool	テストツール。 本ツールを使用して、PS-API を使用した機器への接続が確認できます。 <b>再配布できません。</b>
PS-API¥ Document	取扱説明書。(本書) <b>再配布できません。</b>

フォルダ名	用途
PS-ALARM¥ For Development	ヘッダーファイル/lib ファイル。 開発用です。 <b>再配布できません。</b>
PS-ALARM¥ Redistributable	DLL ファイル。 再配布可能ファイルです。
PS-ALARM¥ Document	取扱説明書。 <b>再配布できません。</b>

フォルダ名	用途
PS-LOOKUP¥ For Development	ヘッダーファイル/lib ファイル。 開発用です。 <b>再配布できません。</b>
PS-LOOKUP¥ Redistributable	DLL ファイル。 再配布可能ファイルです。
PS-LOOKUP¥ Document	取扱説明書。 <b>再配布できません。</b>

[DLL-Sample フォルダ]

フォルダ名	用途
Sample Program	サンプルプログラム。 <b>再配布できません。</b>

### 3.2. インストール方法

同梱の「PS-API Installation Guide for DLL」を参照してください。

### 3.3. アンインストール方法

同梱の「PS-API Installation Guide for DLL」を参照してください。

### 3.4. 制限事項

- (1) システム内の機器のタイムゾーン設定は同一にしてください。
- (2) OS の休止、スタンバイ機能は使用しないでください。
- (3) PS-LOOKUP インスタンスはスレッドセーフではありません。
- (4) マルチバイト文字セットを使用してください。
- (5) 同時に受信できる機器接続情報数は、機器接続情報を受信する PC の性能に依存します。

## 4. ライブラリ概要

### 4.1. 機器への接続

アプリケーションがPS-LOOKUPを利用して機器を検出するには、SetDevLookupListener メソッドでリスナークラスを設定します。リスナークラス設定後は OnDevLookup イベントで機器検出情報が通知されます。機器検出を停止する場合は、SetDevLookupListener メソッドで NULL を設定してください。

\*登録中のリスナーオブジェクトを、アプリケーション側で破棄しないでください。

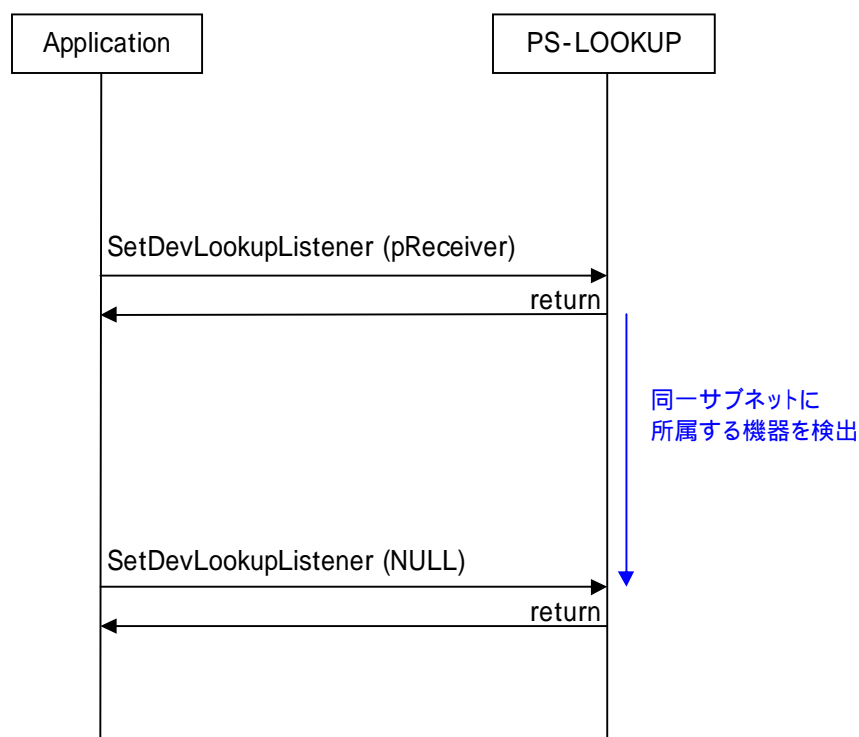


Figure 4-1 Connect to The Device

#### 4.2. PS-LOOKUP を利用した機器検出の流れ

例として、PS-LOOKUP インスタンス化 ~ 開始 ~ 機器検出 ~ 終了の流れを説明します。

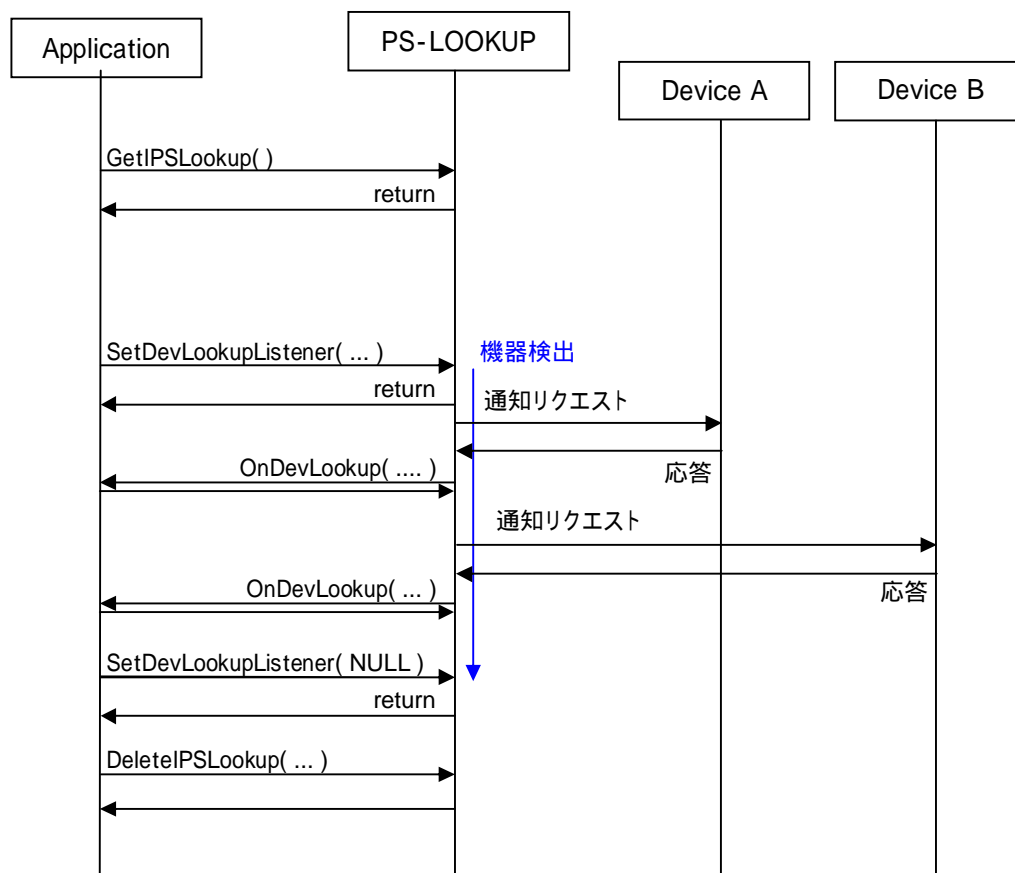


Figure 4-2 Steps to receive device information with PS-LOOKUP

## 5. DLL クラス・メソッド詳細

本章では、本ライブラリの詳細仕様について述べます。

### 5.1. クラス

#### 5.1.1. クラス定義

Table 5-1 Class Definition

No.	クラス名	概要
1	IPSLookup	PS-LOOKUP の機能をアプリケーションから使用するためのインターフェースクラス。ポートの設定、およびリスナークラスの登録を行います。
2	ILookupListener	PS-LOOKUP からの通知を受信するためのインターフェースクラス。アプリケーションは本クラスを継承し、利用したい通知メソッドを実装することで、通知に対する処理を記述することができます。 また、アプリケーションで実装したクラスのインスタンスを IPSLookup に登録することで通知を受信できます。

#### 5.1.2. クラス構造

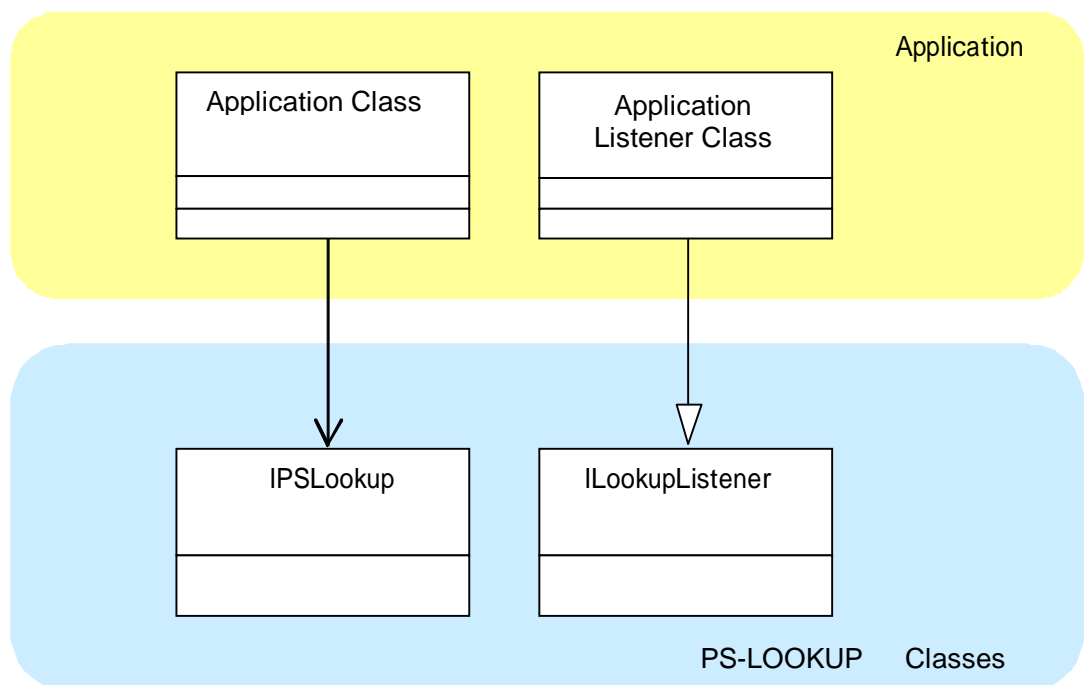


Figure 5-1 Class Diagram

#### 5.1.2.1. グローバル関数

```
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. グローバル関数

### 5.2.1. GetIPSLookup

<b>Class</b>	-
--------------	---

<b>Function</b>	<b>GetIPSLookup</b>
-----------------	---------------------

IPSLookup*	GetIPSLookup ( );
------------	-------------------

#### Description

IPSLookup のインスタンスを生成します。

#### Argument

なし

#### Return value

IPSLookup\*    生成した IPSLookup のポインター

#### Error

## Note

---

## Sequence

---

## Sample program code

---

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

## Reference

---

### 5.2.2. DeletelPSLookup

**Class** -

---

**Function** DeletelPSLookup

---

void DeletelPSLookup (IPSLookup\* ilookup)

---

#### Description

IPSLookup のインスタンスを破棄します。

---

#### Argument

ilookup	IPSLookup インスタンスのポインター	IPSLookup インスタンスのポインターを指定します。
---------	------------------------	-------------------------------

---

#### Return value

なし

---

#### Error

## Note

---

## Sequence

---

## Sample program code

---

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

## Reference

---

### 5.3. PSLookup グループ

#### 5.3.1. アプリケーション リスナー

##### 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

指定したアプリケーションに機器検出情報を通知します。

アプリケーションは、ILookupListener を継承したリスナークラスを作成し、OnDevLookup メソッドを実装する必要があります。

レコーダ(HD300,NWDR(NV Series、NX Series 以外), HD600/700)は、機器検出に対応していません。

#### Argument

macAddr	文字列(半角 24 文字以内)	検出された機器の MAC アドレス
ipAddr	文字列(半角 16 文字以内)	検出された機器の IPv4 アドレス
ipv6Addr	文字列(半角 64 文字以内)	検出された機器の IPv6 アドレス IPv6 アドレスの設定がない場合は、空文字(“”)が設定されます。
portNo	1 - 65535	検出された機器の HTTP ポート番号
camName	文字列(半角 32 文字以内)	検出された機器のカメラ名 カメラ名の設定がない場合は、空文字(“”)が設定されます。また、品番によってはカメラ名を設定していても取得できない場合があります。 機器から通知されるカメラ名は最大半角 15 文字です。
modelName	文字列(半角 16 文字以内)	検出された機器の品番

\* macAddr、ipAddr、ipv6Addr、camName、modelName は OnDevLookup 関数内でのみ使用してください。OnDevLookup 関数が終了すると、メモリは解放されます。

---

**Return value**

---

なし

---

**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

**Method**                SetDevLookupListener

long      SetDevLookupListener(  
   ILookupListener\*   pReceiver  
   );

#### Description

「5.3.1.1 OnDevLookup」を実装したリスナークラスのインスタンスを PS-LOOKUP に設定します。  
リスナーを設定すると、機器検出時に PS-LOOKUP は「5.3.1.1 OnDevLookup」を呼び出すことで、アプリケーションに機器検出情報を通知するようになります。  
レコーダ(HD300, NWDR(NV Series, NX Series 以外), HD600/700)は、機器検出に対応していません。

#### Argument

pReceiver	ポインター	設定するリスナークラスのポインター
	NULL	pReceiver に NULL を設定するとリスナークラスの登録を解除します。

#### Return value

0	成功
0 以外	エラー番号

#### Error

エラーは戻り値で定義されます。

#### Note

---

登録中のリスナーオブジェクトを、アプリケーション側で破棄しないでください。

#### 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 );
------	--

#### Description

指定したアプリケーションにエラーを通知します。  
アプリケーションは、ILookupListener を継承したリスナークラスを作成し、OnError メソッドを実装する必要があります。

#### Argument

errorCode	負の値	エラー番号
description	文字列	エラー詳細 * 本パラメーターはOnError関数内でのみ使用してください。OnError関数が終了すると、メモリは解放されます。

#### Return value

なし

#### Error

**Note**

---

**Sequence**

---

**Sample program code**

---

**Reference**

---

#### 5.3.1.4. SetErrListener

**Class** IPSLookup

**Listener** SetErrListener

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

#### Description

「5.3.1.3OnError」を実装したリスナークラスのインスタンスを PS-LOOKUP に設定します。  
リスナーを設定すると、エラー発生時に PS-LOOKUP は「5.3.1.3OnError」を呼び出すことで、アプリケーションにエラー情報を通知するようになります。

#### Argument

pReceiver	ポインター	設定するリスナークラスのポインター
	NULL	pReceiver に NULL を設定するとリスナークラスの登録を解除します。

#### Return value

0	成功
0 以外	エラー番号

#### Error

エラーは戻り値で定義されます。

#### Note

---

登録中のリスナーオブジェクトを、アプリケーション側で破棄しないでください。

#### Sequence

---

#### Sample program code

---

#### Reference

---

## 6. 操作手順/シーケンス

### 6.1. PSLookup

#### 6.1.1. 操作手順

##### 機器検出開始

手順	プロパティ/メソッド	パラメーター	説明
1	GetIPSLookup	-	IPSLookup のインスタンスを生成します。
2	SetDevLookupListener	リスナークラス (ILookupListener *)	リスナークラスのインスタンスを設定します。
-	(OnDevLookup)	MAC アドレス, IPv4 アドレス, IPv6 アドレス, ポート番号, カメラ名, 品番 (const char*, const char*, const char*, long, const char*, const char*)	機器を検出すると OnDevLookup メソッドでアプリケーションに通知します。

##### 機器検出停止

手順	プロパティ/メソッド	パラメーター	説明
3	SetDevLookupListener	NULL	NULL を設定し、リスナークラスの設定を解除します。
4	DeleteIPSLookup	IPSLookup のポインター (IPSLookup*)	インスタンスを破棄します。

## 6.1.2. シーケンス

### 機器検出開始

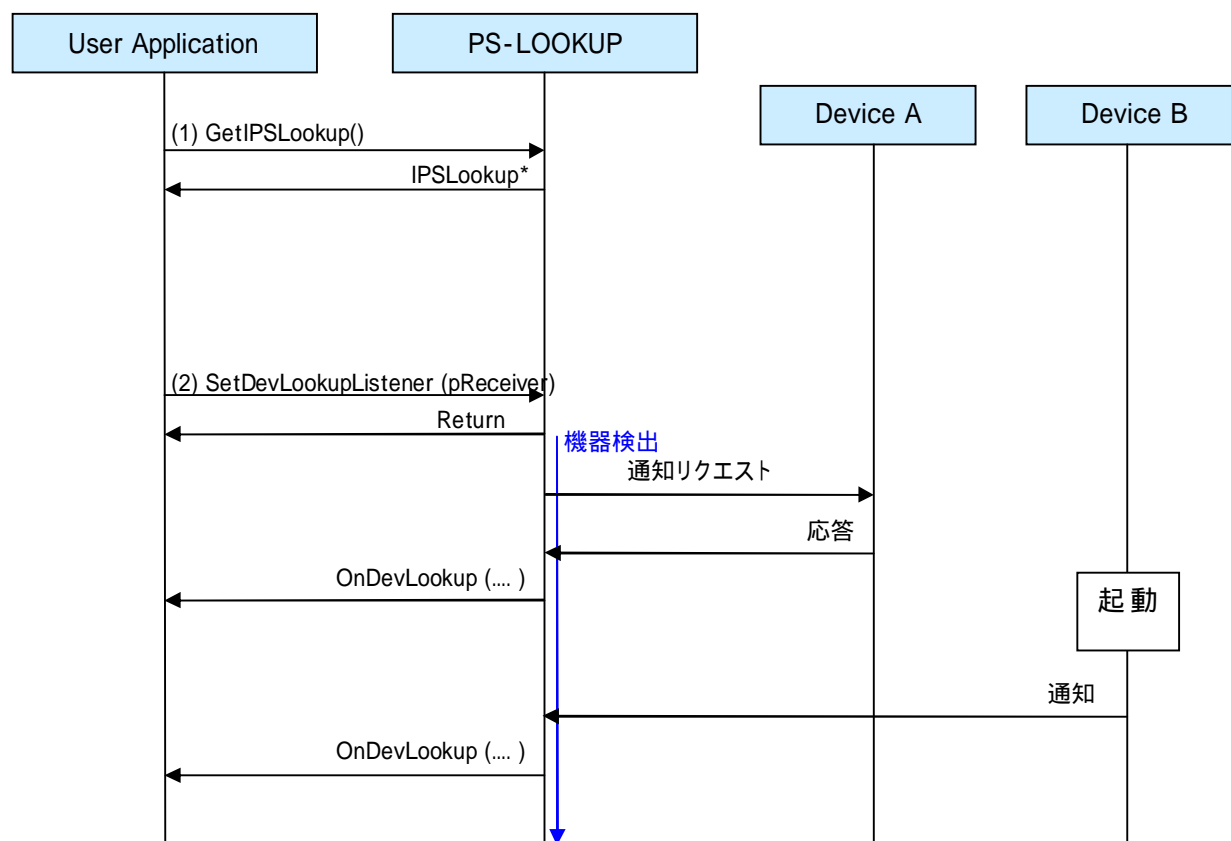


Figure 6-1 Start detecting Devices

## 機器検出停止

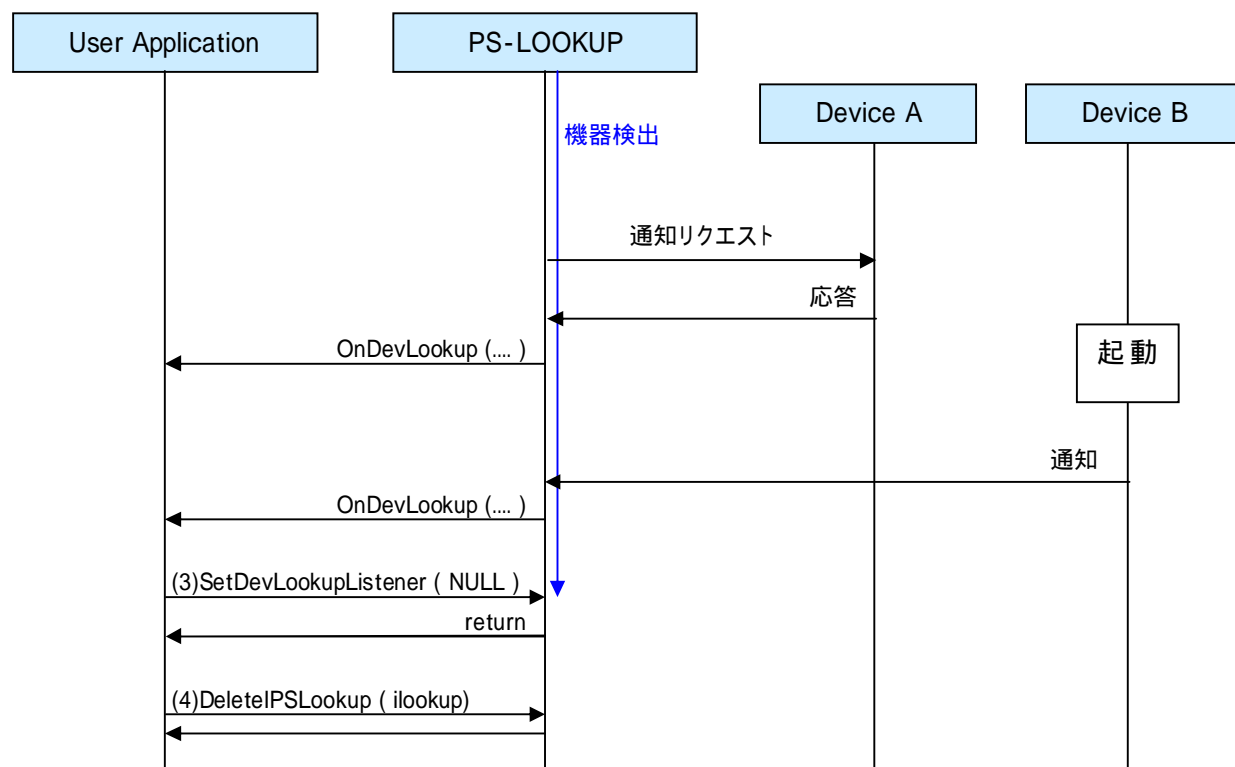


Figure 6-2 Stop detecting Devices

## 7. エラーコード一覧

エラーコードは次のフォーマットで定義されます。(10 進数、8 桁)

-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> 分類	A <sub>1</sub> 重要度	C <sub>1</sub> 発生元	M <sub>1</sub> M <sub>2</sub>	Z <sub>1</sub> Z <sub>2</sub> Z <sub>3</sub> コード	エラー内容
-6: PS-LOOKUP	1: エラー 2: 警告	1: PSLookup	(内部使用)	103	内部処理エラー (WSAStartup)
				104	内部処理エラー (アドレス情報取得)
				105	内部処理エラー (ソケット生成)
				106	内部処理エラー (ソケット bind)
				107	内部処理エラー (ソケット select)
				109	内部処理エラー (機器情報受信)
				110	内部処理エラー (機器情報リクエスト送信)
				201	SDK 内部エラー
				202	SDK 内部エラー
				203	SDK 内部エラー
				204	SDK 内部エラー
				401	機器検出実行中
				501	致命的エラー (スレッド生成失敗)
				502	致命的エラー (オブジェクト生成失敗)
				503	致命的エラー (メモリ確保失敗)