VNDP VIVOTEK Network Development Platform

VIVOTEK NETWORK DEVELPMENT PLATFORM

Discover Register Mechanism Module

Version 3.1.0.1

2017/01/20

© 2017 VIVOTEK Inc. All Right Reserved

VIVOTEK may make changes to specifications and product descriptions at any time, without notice.

The following is trademarks of VIVOTEK Inc., and may be used to identify VIVOTEK products only: VIVOTEK. Other product and company names contained herein may be trademarks of their respective owners.

No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from VIVOTEK Inc.

TABLE OF CONTENTS

VIVOTEK CONFIDENTIAL 2017.01.20

1. Overview

1.1 Introduction

This document describes the properties and methods supported by the VIVOTEK Discover Register Mechanism (DRM) module.

1.2 Getting Started with DRM Module

The main usage of DRM module is to discover all available network cameras or video servers from VIVOTEK in the LAN.

1.3 File Structure

Table 1-1 File Structure

File	Description
doc\VNDP_DRMControl_API.pdf	This manual document
lib\d_DRMControl.lib	The dynamic linking library
lib\DRMControl.dll	The dynamic runtime library
inc\DRMControl.h	Header file

VIVOTEK CONFIDENTIAL 2017.01.20

2. Programmer's Guide

2.1 Using DRM Module

- Starts DRM module.
- Discovers all cameras and servers in the LAN.
- Gets information of available network cameras or video servers in the LAN via a callback function.

VIVOTEK CONFIDENTIAL
2017.01.20

3. Sample Code

3.1 Discover Device

Description

Discovers the devices in the LAN at first, and than searches a certain device by MAC address.

Sample Code

Step 1. Prepare a callback function for DRM control to notify devices

```
DWORD __stdcall DRMControlCallback(
HANDLE hParentObject, EDRMControl_CallbackType eCallbackType, void* pvCallbackData)
{
  tagTDRMServiceInfo* ptDRMServiceInfo = NULL;
  switch (eCallbackType)
  {
    case DRMControl_Callback_DiscoveryResult :
    ptDRMServiceInfo= (tagTDRMServiceInfo*)pvCallbackData;
    printf(“IP address: %-15s “,ptDRMServiceInfo->acIP);
    printf(“Server Type: %s “,ptDRMServiceInfo->acMachineType);
    printf(“CameraName:%s\n”,ptDRMServiceInfo->acCameraName);
    printf(“MAC address:”);
    for(int i = 0; i < 6; i++)
    printf(“%02x”,ptDRMServiceInfo->abyMac[i]);
    printf(“ HTTP port: %-4d “,ptDRMServiceInfo->sHTTPPort);
    printf(“FTP port: %-4d “,ptDRMServiceInfo->sFTPPort);
    printf(“Language: %s\n\n”,ptDRMServiceInfo->acLanguage);
    break;
  ...
  }
}
```

Step 2. Startup Windows Sockets module

```
WSADATA wsaData;
if (WSAStartup(0x202, &wsaData) == SOCKET_ERROR)
```

VIVOTEK CONFIDENTIAL 2017.01.20

```
{
  fprintf(stderr,”WSAStartup error %d\n”,WSAGetLastError());
  return -1;
}
else
{
 printf(“WSAStartup successed.\n”);
}
```

Step 3. Create DRMControl

```
HANDLE hDRMControl = NULL;
TDRMControlCreateOptions tDRMOption;
memset(&tDRMOption, 0, sizeof(tDRMOption));
tDRMOption.dwMaxDiscoveryItem = 100; // Set maximum discovery items
tDRMOption.dwAliasIP = TRUE; // Search with 169.254.x.x interface, too.
tDRMOption.usDRMControlPort = 9000; // Assign a free port
DRMControl_Create(&hDRMControl, &tDRMOption);
```

Step 4. Set callback function

```
DRMControl_SetCallback(hDRMControl, (DWORD) NULL, DRMControlCallback);
```

Step 5. Start the DRMControl

```
DRMControl_Start(hDRMControl);
```

Step 6: Start to discover devices in the LAN.

```
printf(“Start to discover the devices in the LAN\n”);
DRMControl_Discovery(hDRMControl); // discovers all devices in the LAN
Sleep(10000);
printf(“Discover device by MAC\n”);
DRMControl_DiscoveryByMAC(hDRMControl,”0002D100089C”); // search for this MAC only
Sleep(10000);
```

STEP 7: Release the module

```
DRMControl_Stop(hDRMControl);
DRMControl_Close(&hDRMControl);
WSACleanup();
```

VIVOTEK CONFIDENTIAL 2017.01.20

4. API Reference

This chapter describes the API function calls for the DRM module.

4.1 Enumeration

The enumeration used is depicted here.

VIVOTEK CONFIDENTIAL
2017.01.20

4.1.1 EDRMControl_CallbackType

The enumeration presents DRMControl callback type.

```
typedef enum {
                    DRMControl_Callback_DiscoveryResult     = 1
                    DRMControl_Callback_Stopped             = 4
                    DRMControl_Callback_Error               = 5
} EDRMControl_CallbackType;
```

Values

DRMControl_Callback_DiscoveryResult

Discover device and information acquired.

DRMControl_Callback_Stopped

DRM module stopped.

DRMControl_Callback_Error

Search device error.

Remarks

Requirements

DRMControl.h

VIVOTEK CONFIDENTIAL 2017.01.20

4.2 Data Structure

The data structure is depicted here.

VIVOTEK CONFIDENTIAL
2017.01.20

4.2.1 TDRMControlCreateOptions

This structure collects the settings of the DRM object. When creating DRM object, fill this structure to setup.

```
typedef struct {
                                        DWORD               dwMaxDiscoveryItem ;
                                        unsigned short      usDRMControlPort ;
                                        DWORD               dwAliasIP;
} TDRMControlCreateOptions;
```

Members

dwMaxDiscoveryItem

The maximal items that the controller can handle.

usDRMControlPort

Port number which DRM object is going to use for discover devices in the LAN.

dwAliasIP

Discover devices with the 169.254.x.x interface, too. The value should be TRUE or FALSE.

Remarks

Requirements

DRMControl.h

VIVOTEK CONFIDENTIAL 2017.01.20

4.2.2 tagTDRMServiceInfo

This structure defines of the information of the device searched from LAN. The information will callback in the callback function.

```
typedef struct {
                                        char            acMachineType[44];
                                        BYTE            abyMac[6] ;
                                        char            acIP[20];
                                        char            acServiceName[44];
                                        short           sHTTPPort;
                                        short           sFTPPort;
                                        char            acLanguage[6];
                                        DWORD           dwEZversion;
                                        char            acCameraName[97];
} tagTDRMServiceInfo;
```

VIVOTEK CON
2017.01.2

Members

acMachineType[44]

The firmware version of the device.

abyMac[6]

MAC address of the device.

acIP[20]

IP address of the device.

acServiceName[44]

Not used.

sHTTPPort

Port number of the web server of the device.

sFTPPort

FTP port of the ftp server of the device.

acLanguage[6]

lauguange of the deivce.

dwEZversion

This parameter is for EZ Installation. Please reference the related document.

acCameraName

The camera name, only some newer camera support this field.

Remarks

Requirements

DRMControl.h

VIVOTEK CONFIDENTIAL 2017.01.20

4.3 Callback Function

The Callback function is depicted here.

VIVOTEK CONFIDENTIAL 2017.01.20

4.3.1 FDRMControlCallback

This define the type of the callback function.

Syntax

```
typedef DWORD (*FDRMControlCallback)
                                        HANDLE                      hInstance,
                                        EDRMControl_CallbackType    eDRMControlCallbackType,
                                        void                        *pvParam
;
```

Members

hInstance

[in] the bypass external 32-bit data to callback function which is the callback context (the first parameter) used in the DRMControl_SetCallback.

eDRMControlCallbackType

[in] The callback type of the callback function.

pvParam

[in] If the callback type is DRMControl_Callback_DiscoveryResult defined in EDRMControl_CallbackType, the pvParam points to the structure tagTDRMServiceInfo that contains the information of found device.In other callback types, pvParam points to NULL.

Remarks

Requirements

DRMControl.h

VIVOTEK CONFIDENT
2017.01.20

4.4 API Definition

The API definition is depicted here.

VIVOTEK CONFIDENTIAL
2017.01.20

4.4.1 DRMControl_Create

This function create the DRM Module. You must call this function before using this module.

Syntax

```
SCODE DRMControl_Create (
                                        HANDLE                          *phObject,
                                        TDRMControlCreateOptions        *pDRMControlInitOptions
);
```

Parameters

phPacketMaker

[out] The pointer to receive the handle of DRM object.

pDRMControlInitOptions

[in] Pointer to the structure that contain the setting to create DRM object.

Return Values

S_OK

Create this module ok.

S_FAIL

Create this module failed.

Remarks

Requirements

DRMControl.h

See Also

TDRMControlCreateOptions

VIVOTEK CONFIDENTIAL 2017.01.20

4.4.2 DRMControl_Close

Call this function to release the DRM object.

Syntax

```
SCODE DRMControl_Close (
                                    HANDLE                  *phObject
);
```

Parameters

phPacketMaker

[in] The address of the pointer to the DRM object, returned by **DRMControl_Create**.

Return Values

S_OK

Release the object successfully.

S_FAIL

Failed to release the object.

Remarks

Requirements

DRMControl.h

See Also

```
typedef DWORD (*FDRMControlCallback)(DWORD dwInstance, EDRMControl_CallbackType
eDRMControlCallbackType,void *pvParam);
```

VIVOTEK CONFIDENTIAL
2017.01.20

4.4.3 DRMControl_SetCallback

Create a channel to display video or play sound.

Syntax

```
SCODE DRMControl_SetCallback    (
                                        HANDLE                  hObject,
                                        HANDLE                  dwInstance,
                                        FDRMControlCallback     fDRMControlCallback
);
```

Parameters

hObject

[in] The handle of the DRM object created by DRMControl_Create.

dwInstance

[in] The instance pass to the DRM module. Usually the object might used in the callback function.

fDRMControlCallback

[in] Pointer to the callback function.

Return Values

S_OK

Set callback function OK.

S_FAIL

Set callback function failed.

Remarks

Requirements

DRMControl.h

VIVOTEK CONFIDENTIAL 2017.01.20

4.4.4 DRMControl_Start

Start DRM module.

Syntax

```
SCODE DRMControl_Start (
                                                      HANDLE                          hDRMObject
);
```

Parameters

hDRMObject

[in] The handle of DRM Object created by DRMControl_Create.

Return Values

S_OK

Start the DRM module successfully.

S_FAIL

Fail to start DRM module.

Remarks

Requirements

DRMControl.h

See Also

VIVOTEK CONFIDENTIAL 2017.01.20

4.4.5 DRMControl_Stop

Stop DRM module.

Syntax

```
SCODE DRMControl_Stop (
                                        HANDLE                  hDRMObject
);
```

Parameters

hDRMObject

[in] The handle of DRM Object created by DRMControl_Create.

Return Values

S_OK

Stop the DRM module successfully.

S_FAIL

Fail to stop DRM module.

Remarks

Requirements

DRMControl.h

See Also

VIVOTEK CONFIDENTIAL 2017.01.20

4.4.6 DRMControl_Discovery

Call this function to search all the devices in the LAN.

Syntax

```
SCODE DRMControl_ Discovery (
                                        HANDLE                    hDRMObject
);
```

Parameters

hDRMObject

[in] The handle of DRM Object created by DRMControl_Create.

Return Values

S_OK

The discover message send out successfully.

S_FAIL

Fail to send out discover message.

Remarks

Once this API is called, the module will ignore any previous search and only callback the result of current search. The best way to use this API is to call it once, wait the result in the callback function about 2~5 seconds.

Requirements

DRMControl.h

See Also

VIVOTEK CONFIDENTIAL 2017.01.20

4.4.7 DRMControl_DiscoveryByMAC

Call this function to search the device of the specific MAC address in the LAN.

Syntax

```
SCODE DRMControl_DiscoveryByMAC (
                                        HANDLE                  hDRMObject
                                        char*                   acMACAddress
);
```

Parameters

hDRMObject

[in] The handle of DRM Object created by DRMControl_Create.

acMACAddress

[in] The pointer to the string of MAC address. The MAC address must be a 12 byte string.

Return Values

S_OK

The discover message send out successfully.

S_FAIL

Fail to send out discover message.

Remarks

Once this API is called, the module will ignore any previous search and only callback the result of current search. The best way to use this API is to call it once, wait the result in the callback function about 2~5 seconds.

Requirements

DRMControl.h

See Also

VIVOTEK CONFIDENTIAL 2017.01.20