VIVOTEK NETWORK DEVELOPMENT PLATFORM

Server Dependent Resource

Version 4.1.0.8

2009/8/31

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.

TABLE OF CONTENTS

1. Overview

2.PROGRAMMER'S GUIDE

3.API Reference

1. Overview

1.1. Introduction

This document describes how to use SrvDepResouce module to get Server Dependent information.

The SrvDepResource module don’t maintain/update the device information any more(after VIVOTEK IP7131 camera). We recommend you use ServerManager module strongly.

File Structure

FILE	DESCRIPTION
doc\VNDP_SrvDepResource_API.pdf	This manual
lib\ d_SrvDepResource.lib	The dynamic linking library
lib\SrvDepResource.dll	The dynamic runtime library
inc\SrvDepResource.h	Header file
Inc\SrvTypeDef.h	Common definition file

2. PROGRAMMER'S GUIDE

2.1. Using SrvDepResource Module

Getting Server Dependent Information

You can get the server dependent information by server's friendly name (call SrvDepResource_GetParamForServerByFriendlyName) or by server's firmware version (call SrvDepResource_GetParamForServer).

Enumerating Supported Servers

Call SrvDepResource_EnumerateSupportingMode to acquire a list of friendly names of the supported servers. Remember to call SrvDepResource_FreeFriendlyNameList to free the allocated memory.

Supported Server Friendly Name

Please use the SrvDepResource_EnumerateSupportingMode function to retrieve all the models currently supported.

3. API Reference

This chapter contains the API function calls for the Server Dependent Resource.

3.1. Data Structure

The data structure is depicted here.

TSrvDepResource_ServerInfo

This structure indicates server information.

```
typedef struct
{
char                zServerName[20];
DWORD               dwCamNum;
DWORD               dwComNum;
DWORD               dwDiNum;
DWORD               dwDoNum;
DWORD               dwVideoCodec;
DWORD               dwAudioCodec;
char                zVideoUrl[40];
char                zRxUrl[40];
char                zTxUrl[40];
char                zQuadUrl[40];
char                zPtzUrl[40];
char                zRecallUrl[40];
char                zPresetUrl[40];
char                zGetDiUrl[40];
char                zSetDoUrl[40];
char                zUartUrl[40];
DWORD               dwMDWinNum;
char                zDefaultVSize[10];
DWORD               dwDefaultQuality;
DWORD               dwDefaultProtocol;
DWORD               dwDefaultMediaType;
DWORD               dwFlag;
} TSrvDepResource_ServerInfo;
```

Members

zServerName

server friendly name

dwCamNum
the camera number

dwComNum
the com port number

dwDiNum
the DI number

dwDoNum
the DO number

dwVideoCodec
the video codec

dwAudioCodec
the audio codec.

zVideoUrl
the URL for getting video stream

zRxUrl
the URL to receive media stream

zTxUrl
the URL to transmit media stream

zQuadUrl
the URL for getting quad-video stream

zPtzUrl
the URL for PTZ control

zRecallUrl
the URL for recalling preset position

zPresetUrl
the URL for setting preset position

zGetDiUrl
the URL for getting DI status

zSetDoUrl
the URL for setting DO

zUartUrl
the Uart URL

dwMDWinNum
the maximum number of the motion detection windows

zDefaultVSize
the default Vsize (only valid for 2K series)

dwDefaultQuality
the default video quality

dwDefaultProtocol
the default protocol type

dwDefaultMediaType
the default media type

dwFlag
the ability to support camera control

Remarks

There are 2 options for dwFlag.

SRV_FLAG_PTZ_SPEED
The server supports adjusting PTZ speed.

SRV_FLAG_PTZ_ADV
The server supports adjusting PTZ focus.

Requirements

SrvDepResource.h

TSrvDepResource_SysInfo

This structure indicates server information.

```
typedef struct
{
BYTE                    sModel[SDS_MAX_MODEL_LEN + 1];
BYTE                    szHostName[SDS_MAX_HOSTNAME_LEN + 1];
BYTE                    aszLocation[SDS_MAX_CHANNEL_NUM][SDS_MAX_
                        LOCATION_LEN + 1];
                        aaszPreset[SDS_MAX_CHANNEL_NUM][SDS_MAX_
BYTE                    PRESET_NUM][SDS_MAX_PRESET_LEN];
                        szRTSPAccessName[SDS_MAX_RTSP_ACC_NAME
BYTE                    + 1];
                        adwPTZCap[SDS_MAX_CHANNEL_NUM];
DWORD                   dwCapVersion;
DWORD                   dwEvent ;
DWORD                   dwRTSPPort;
DWORD                   dwCamNum
DWORD                   dwUartNum;
DWORD                   dwDINum;
DWORD                   dwDONum;
DWORD                   dwVideoCodec;
DWORD                   dwAudioCodec;
DWORD                   dwMotionMethod;
DWORD                   dwStreamingProtocol;
DWORD
} TSrvDepResource_ServerInfo;
```

Members

sModel
the model name of the server

szHostName
the host name of the server

aszLocation
the array of location of server

aaszPreset
the array of preset of server

adwPTZCap
the DWORD array of server

dwCapVersion
the video codec

dwEvent
the audio codec.

dwCamNum
the URL for getting video stream

dwUartNum
the URL to receive media stream

dwDINum
the digital input number of server

dwDONum
the digital output number of server

dwVideoCodec
the video codec of server

dwAudioCode
the audio codec of server

dwMotionMethod
the motion method of server

dwStreamingProtocol
the streaming protocol of server

Remarks

There are 2 options for dwFlag.

SRV_FLAG_PTZ_SPEED
The server supports adjusting PTZ speed.

SRV_FLAG_PTZ_ADV
The server supports adjusting PTZ focus.

Requirements

SrvDepResource.h

3.2. Enumeration

The enumeration used is depicted here.

TsdrProtocolType

This enumeration indicates the protocol types.

```
typedef enum
{
eptHTTP,

eptTCP,

eptUDP

eptMULTICAST
} TsdrProtocolType;
```

Values

eptHTTP
HTTP

eptTCP
TCP

eptUDP
UDP

eptMULTICAST
Multi-Cast

Remarks

Requirements

SrvTypeDef.h

TsdrMediaType

This enumeration indicates the request media types.

```
typedef enum
{
emtAudio                 = 1,
emtVideo                 = 2,
emtTransmitAudio         = 4
} TsdrMediaType;
```

Values

emtAudio

the media is received audio

emtVideo

the media is received video

emtTransmitAudio

the media is transmitted audio

Remarks

Requirements

SrvTypeDef.h

TsdrVideoCodec

This enumeration indicates the video codec types.

```
typedef enum
{
eVCodecNone                    = 0x0000,

eVCodecMJPEG                   = 0x0001,

eVCodecH263                    = 0x0002,

eVCodecMPEG4                   = 0x0004
} TsdrVideoCodec;
```

Values

eVCodecNone
no video codec

eVCodecMJPEG
Motion JPEG

eVCodecH263
H.263

eVCodecMPEG4
MPEG-4

Remarks

Requirements

SrvTypeDef.h

TsdrAudioCodec

This enumeration indicates the audio codec types.

```
typedef enum
{
eACodecNone                = 0x0000,
eACodecG7221               = 0x0100,
eACodecG729A               = 0x0200,
eACodecAAC                 = 0x0400,
eACodecGAMR                = 0x0800,
} TsdrAudioCodec;
```

Values

eACodecNone
no audio codec

eACodecG7221
G.722.1

eACodecG729A
G.729A

eACodecAAC
AAC

eACodecGAMR
GAMR

Remarks

Requirements

SrvTypeDef.h

EOptSysInfo

This enumeration indicates the items in server to be set. To set these options, please call DataBroker_SetConnectionExtraOption.

```
typedef enum
{
eoptszHostName          = 0x00000001,
eoptLocation            = 0x00000002,
eoptPreset              = 0x00000004,
eoptRTSPAccessName      = 0x00000008,
eoptPTZCap              = 0x00000010,
eoptCapVersion          = 0x00000020,
eoptEvent               = 0x00000040,
eoptRTSPPort            = 0x00000080,
eoptCamNum              = 0x00000100,
eoptUartNum             = 0x00000200,
eoptDINum               = 0x00000400,
eoptDONum               = 0x00000800,
eoptVideoCodec          = 0x00001000,
eoptAudioCodec          = 0x00002000,
eoptMotionMethod        = 0x00004000,
eoptStreamingProtocol   = 0x00008000,
} TsdrAudioCodec;
```

Values

eoptszHostName
Reserved for future use.

eoptLocation
Reserved for future use.

eoptPreset
Reserved for future use.

eoptRTSPAccessName
Set the RTSP access name.

eoptPTZCap
Reserved for future use.

eoptCapVersion
Reserved for future use.

eoptEvent

Reserved for future use.

eoptRTSPPort

Set RTSP port.

eoptCamNum

Reserved for future use.

eoptUartNum

Reserved for future use.

eoptDINum

Reserved for future use.

eoptDONum

Reserved for future use.

eoptVideoCodec

Set video codec.

eoptAudioCodec

Set audio codec.

eoptMotionMethod

Reserved for future use.

eoptStreamingProtocol

Reserved for future use.

Remarks

Requirements

SrvTypeDef.h

ESysPTZCapability

This enumeration indicates the PTZ capablity types.

```
typedef enum
{
ptzCapLensBuiltIn             = 0x00000001,
ptzCapPan                     = 0x00000002,
ptzCapTilt                    = 0x00000004
ptzCapZoom                    = 0x00000008
ptzCapFocus                   = 0x000000010
ptxCapIris                    = 0x000000020
} ESysPTZCapability;
```

Values

ptzCapLensBuiltIn
Lens built in

ptzCapPan
th capability to pan the camera

ptzCapTilt
the capability to pan the camera

ptzCapZoom
the capability to zoom the camera

ptzCapFocus
the capability to focus the camera

ptzCapIris
the capability of iris of the camera

Remarks

Requirements

SrvTypeDef.h

ESysCapEvent

This enumeration indicates the event capablity types.

```
typedef enum
{
eventCapVideEvents          = 0x00000001,
eventCapSysinfo             = 0x00000002,
} ESysCapEvent;
```

Values

eventCapVideoEvents
the capability of video event

eventCapSysinfo
the capability of sysinfo

Remarks

Requirements

SrvTypeDef.h

ESysCapStreaming

This enumeration indicates the streaming capablity types.

```
typedef enum
{
streamCapHttpSingleJpeg    = 0x00000001,
streamCapHttpMpeg          = 0x00000002,
streamCap3K_Http           = 0x00000004,
streamCap3K_TCP            = 0x00000008,
streamCap3K_UDP            = 0x00000010,
streamCap6K_HTTP           = 0x00000020,
streamCap6K_UDP            = 0x00000040,
streamCap6K_HTTTALK        = 0x00000080,
streamRTSP_RTP_UDP         = 0x00000100,
streamRTSP_RTP_TCP         = 0x00000200,
streamRTSP_RTP_HTTP        = 0x00000400,
streamRTSP_RTP_            = 0x00000800,
MCAST
                           = 0x00001000,
streamRTP_MCAST
} ESysCapEvent;
```

Values

streamingCapHttpSingleJpeg
the capability of single jpeg via http protocol

streamingCapHttpMpeg
the capability of motion jpeg via http protocol

streamingCap3k_Http
the streaming capability of 3k type server via http protocol

StreamingCap3k_TCP
the streaming capability of 3k type server via tcp protocol

StreamingCap3k_UDP
the streaming capability of 3k type server via udp protocol

StreamingCap3k_HTTP
the streaming capability of 3k type server via http protocol

StreamingCap6k_HTTP
the streaming capability of 6k type server via http protocol

StreamingCap6k_UDP
the streaming capability of 3k type server via udp protocol

StreamingCap6k_HTTP_TALK
the talk capability of 6k type server via http protocol

StreamingRTSP_RTP_UDP
the streaming capability of rtsp type server via udp protocol

StreamingRTSP_RTP_TCP
the streaming capability of rtsp type server via tcp protocol

StreamingRTSP_RTP_HTTP
the streaming capability of rtsp type server via http protocol

StreamingRTSP_RTP_MCAST
the streaming capability of rtsp type server via multi cast protocol

StreamingRTP_MCAST
the streaming capability of rtp via multi cast

Remarks

Requirements

SrvTypeDef.h

3.3. API Definition

The API definition is depicted here.

SrvDepResource_GetParamForServerByFriendlyName

Get server information by given the server friendly name.

Syntax

```
SCODE SrvDepResource_GetParamForServerByFriendlyName (
        char* pzFriendlyName,
        TSrvDepResource_ServerInfo *ptServerInfo      );
```

Parameters

***pzFriendlyName**

[int] a pointer to the string of server friendly name

***ptServerInfo**

[out] a pointer to receive the TSrvDepResource_ServerInfo object.

Return Values

S_OK

Get server information successfully.

S_FAIL

Can’t find the information of specific server.

Remarks

Requirements

SrvDepResource.h

See Also

SrvDepResource_GetParamForServer

Get server information by given the server’s firmware version.

Syntax

```
SCODE
SrvDepResource_GetParamForServer(     char* pzServerType,
                                      TSrvDepResource_ServerInfo*
                                      ptServerInfo
                                                                  );
```

Parameters

***pzServerType**

[in] the pointer to the string of server’s firmware version.

***ptServerInfo**

[out] a pointer to receive the TSrvDepResource_ServerInfo object.

Return Values

S_OK

Get server information successfully.

S_FAIL

Can’t find the information of specific server.

Remarks

Requirements

SrvDepResource.h

See Also

SrvDepResource_GetMappingFriendlyName

Given the server’s firmware version and acquire the server friendly name.

Syntax

```
SCODE
SrvDepResource_GetMappingFriendlyName (      char* pzServerType,
                                             char* pzFriendlyName   );
```

Parameters

***pzServerType**

[in] a pointer to the string of the server’s firmware version.

***pzFriendlyName**

[out] a pointer to the string of the server friendly name.

Return Values

S_OK

Mapping to friendly name successfully.

S_FAIL

Failed to map to friendly name.

Remarks

Requirements

SrvDepResource.h

See Also

SrvDepResource_EnumerateSupportingMode

Get the enumeration of the supporting server friendly name.

Syntax

```
SCODE
SrvDepResource_EnumerateSupportingMode (    char*** pFriendlyNameList,
                                            DWORD* dwServerNum        );
```

Parameters

*****pFriendlyNameList**

[out] the pointer to an array of server friendly name list.

***dwServerNum**

[out] the pointer to the number of server friendly name in the list.

Return Values

S_OK

Get the enumeration of supporting server friendly name successfully.

ERR_OUT_OF_MEMORY

Out of memory.

Remarks

Requirements

SrvDepResource.h

See Also

SrvDepResource_FreeFriendlyNameList

Free the server friendly name list.

Syntax

```
SCODE
SrvDepResource_FreeFriendlyNameList (      char*** pFriendlyNameList      );
```

Parameters

*****pFriendlyNameList**

[in] the pointer to the array of server friendly name list

Return Values

S_OK

Free the list successfully.

Remarks

Requirements

SrvDepResource.h

See Also

SrvDepResource_GetVersionInfo

Get the version information of the SrvDepResource library.

Syntax

```
SCODE
SrvDepResource_ GetVersionInfo (      BYTE *byMajor,
                                      BYTE *byMinor,
                                      BYTE *byBuild,
                                      BYTE *byRevision     );
```

Parameters

***byMajor**

[out] the pointer to the Major byte of the version.

***byMinor**

[out] the pointer to the Minor byte of the version.

***byBuild**

[out] the pointer to the Build byte of the version.

***byRevision**

[out] the pointer to the Revision byte of the version.

Return Values

S_OK

Get the version information successfully.

Remarks

Requirements

SrvDepResource.h

See Also

SrvDepResource_ParseSysInfo

Parse the received system information from server.

Syntax

```
SCODE
SrvDepResource_ ParseSysInfo (    BYTE * bySysinfoData,
                                  TSrvDepResource_SysInfo
                                  *ptTarget                   );
```

Parameters

***bySysinfoData**

[in] the pointer to the byte of the original sysinfo data.

***ptTarget**

[out] the pointer to the data structure of parsed sysinfo data.

Return Values

S_OK

Parse the sysinfo successfully.

1.

Remarks

The pbySysinfoData is actually a char string,and it needs to contain the '\0' at end of string, we define BYTE here to avoid unicode issue. Under windows CE environment, it must be a char too.

Requirements

SrvDepResource.h

See Also

SrvDepResource_SetDataFilePath

If the server's information is not in SrvDepResource's table, it will retrieve the information from server and save to an XML file, Data.xml. The default path is the same as AP. This function can set the XML's file path.

Syntax

```
SCODE
SrvDepResource_SetDataFilePath (     TCHAR *tXMLDataPath    );
```

Parameters

***tXMLDataPath**

[in] file path to be set.

Return Values

S_OK

Set file path successfully.

Remarks

Requirements

SrvDepResource.h

See Also