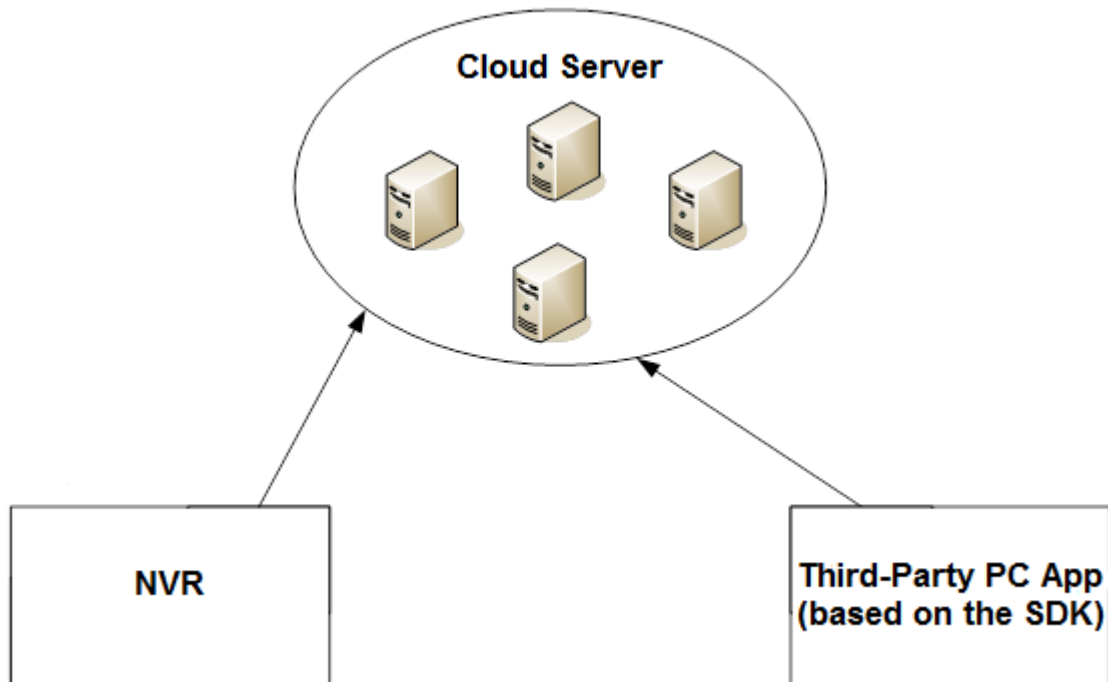


P2P NVR SDK Interface Guide for Third-Party PC Applications

Contents

1	Overview	3
2	Connection.....	3
3	Interfaces	4
3.1	Creating a Port Mapping Service	4
3.2	Destroying a Port Mapping Service.....	5
3.3	Adding a Port Mapping Entry (Using the Default IP and Port Number)	5
3.4	Adding a Port Mapping Entry (Using a Specific IP and Port Number)	6
3.5	Adding a Port Mapping Entry (Accessing a Specific Channel)	7
3.6	Deleting a Port Mapping Entry	8
3.7	Checking Connection Status	9
4	Example.....	9

1 Overview



The P2P NVR SDK (hereinafter referred to as the SDK) connects a third-party PC application to an NVR. The third-party PC application accesses the NVR by visiting the TCP listening port of the SDK. For the function calling procedure, see *P2P NVR SDK Guide*.

2 Connection

1. Initialization

Call the `NPC_F_MPI_MON_DNP_CreateDevNetPortServer` function to initialize the SDK.

This function will register with the server and go through the authentication process. The connection between the SDK and server is kept by heartbeat messages.

2. Creating a port mapping entry

Call the `NPC_F_MPI_MON_DNP_AddPort` function to create a port

mapping entry.

Calling this function will start a local listening port on the PC. Visiting this local listening port allows the access to the media port of the NVR.

3. Connecting the third-party PC application to the NVR

Connect the third-party PC application to the NVR using the following:

- Target IP: 127.0.0.1
- Target Port: the preceding local listening port on the PC

4. Deleting the port mapping

Call the `NPC_F_MPI_MON_DNP_DelPort` function to delete the port mapping after operations are performed.

3 Interfaces

3.1 Creating a Port Mapping Service

Function: `NPC_F_MPI_MON_DNP_CreateDevNetPortServer`

Description: This function creates a port mapping service to connect the third-party PC application to the NVR. When the authentication succeeds, this function returns a mapping service handle, which is used by follow-up functions.

Returned value: a mapping service handle; if the value is NULL, the creation fails

Parameters:

`in_pServerAddr`: server IP or domain

`in_usServerPort`: server port number

`in_pAuthId`: authentication ID

`in_pAuthPwd`: authentication password

3.2 Destroying a Port Mapping Service

Function: NPC_F_MPI_MON_DNP_DestroyDevNetPortServer

Description: Destroy a port mapping service.

Returned value:

Parameters:

in_hDevNetPort: service handle returned by the

NPC_F_MPI_MON_DNP_CreateDevNetPortServer function.

3.3 Adding a Port Mapping Entry (Using the Default IP and Port Number)

Function: NPC_F_MPI_MON_DNP_AddPort

Description: Add a port mapping entry.

Returned value: 0: succeeds; not 0: fails

Error code:

7: login to the server timed out

8: looking for an idle TCP listening port failed

9: parameter error

600001: parsing connection parameters failed

600002: creating a target failed

600004: creating a port mapping failed

600009: parameter error

630001: protocol error or system calling failed

630002: connecting to the NVR failed

630004: allocating resources failed

630006: accessing the server failed (timeout, disconnected, or server error)

630009: other cause

630101: incorrect username

630102: incorrect password

630104: connecting to the server

630105: disconnected from the server

630111: no permission

630112: NVR being offline

Parameters:

in_hDevNetPort: service handle returned by the

NPC_F_MPI_MON_DNP_CreateDevNetPortServer function

in_pUmid: UMID of the NVR, which is a unique ID

inout_pLocalPort: local listening port on the PC; this parameter is mandatory; if the value is 0, the PC assigns a port number

3.4 Adding a Port Mapping Entry (Using a Specific IP and Port Number)

Function: NPC_F_MPI_MON_DNP_AddPortByDestAddr

Description: Add a port mapping entry.

Returned value: 0: succeeds; not 0: fails

Error code:

7: login to the server timed out

8: looking for an idle TCP listening port failed

9: parameter error

600001: parsing connection parameters failed

600002: creating a target failed

600004: creating a port mapping failed

600009: parameter error

630001: protocol error or system calling failed

630002: connecting to the NVR failed

630004: allocating resources failed

630006: accessing the server failed (timeout, disconnected, or server error)

630009: other cause

630101: incorrect username

630102: incorrect password

630104: connecting to the server

630105: disconnected from the server

630111: no permission

630112: NVR being offline

Parameters:

in_hDevNetPort: service handle returned by the

NPC_F_MPI_MON_DNP_CreateDevNetPortServer function

in_pUmid: UMID of the NVR

in_pDestIpaddr: IP of the NVR

in_usDestPort: port of the NVR

inout_pLocalPort: local listening port on the PC; if the value is 0, the PC assigns a port number

3.5 Adding a Port Mapping Entry (Accessing a Specific Channel)

Function: NPC_F_MPI_MON_DNP_AddPortByChNo

Description: Add a port mapping entry.

Returned value: 0: succeeds; not 0: fails

Error code:

7: login to the server timed out

8: looking for an idle TCP listening port failed

9: parameter error

600001: parsing connection parameters failed

600002: creating a target failed

600004: creating a port mapping failed

600009: parameter error

630001: protocol error or system calling failed

630002: connecting to the NVR failed

630004: allocating resources failed

630006: accessing the server failed (timeout, disconnected, or server error)

630009: other cause

630101: incorrect username

630102: incorrect password

630104: connecting to the server

630105: disconnected from the server

630111: no permission

630112: NVR being offline

Parameters:

in_hDevNetPort: service handle returned by the

NPC_F_MPI_MON_DNP_CreateDevNetPortServer function

in_pUmid: UMID

in_iDestChNo: ID of the target channel; if the value is 0, the destination channel is the default channel

inout_pLocalPort: local listening port on the PC; if the value is 0, the PC assigns a port number

3.6 Deleting a Port Mapping Entry

Function: NPC_F_MPI_MON_DNP_DelPort

Description: Delete a port mapping entry.

Returned value:

Parameters:

in_hDevNetPort: service handle returned by the

NPC_F_MPI_MON_DNP_CreateDevNetPortServer function

in_usLocalPort: local listening port on the PC, which is started by the

NPC_F_MPI_MON_DNP_AddPort function

3.7 Checking Connection Status

Function: NPC_F_MPI_MON_DNP_CheckSrvConnState

Description: Check connection status.

Returned value: 0: disconnected; 1: connecting; 2: connected; 9: error

Parameters:

in_hDevNetPort: service handle returned by the

NPC_F_MPI_MON_DNP_CreateDevNetPortServer function

4 Example

```
#include "NPC_MPI_MON_DevNetPortServer.h"
```

```
NPC_HANDLE hDevNetPortServer = NULL;
```

```
int main(int argc, char* argv[])
```

```
{
```

```
    NPC_USHORT usLocalPort;
```

```
    NPC_CHAR sGetStrBuf[20];
```

```
    //INIT AREA
```

```
    {
```

```
        usLocalPort = 0;
```

```
    }
```

```
    //CODE AREA
```

```
    {
```

```
        //printf("Hello World!\n");
```

```
        //Create a port mapping service.
```

```
        hDevNetPortServer = NPC_F_MPI_MON_DNP_CreateDevNetPortServer(
```

```
            "app.P2P.cn",                //[IN] server IP or domain
```

```
            8300,                        //[IN] server port
```

```
            "xxxxxx",                   //[IN] authentication ID
```

```
            "xxxxxx"                     //[IN] authentication password
```

```
        );
```

```
        if ( hDevNetPortServer == NULL )
```

```
        {
```

```
            printf("The creation fails.\n");
```

```

        return 0;
    }
    printf("The creation succeeds.\n");

    usLocalPort = 23000;

    //App a port mapping entry.
    if ( NPC_F_MPI_MON_DNP_AddPort(
        hDevNetPortServer,          //[IN] service handle returned by the
        NPC_F_MPI_MON_DNP_CreateDevNetPortServer function
        "xxxxxxxxxxxx",            //[IN]UMID
        &usLocalPort                //[INOUT] local listening port on the PC; if the
        value is 0, the PC assigns a port number
        ) != 0 )
    {
        printf("The adding fails.\n");
        goto _NPC_CLEAR;
    }
    printf("The adding succeeds.\n");

    gets(sGetStrBuf);

    //Delete a port mapping entry.
    NPC_F_MPI_MON_DNP_DelPort(
        hDevNetPortServer,          //[IN] service handle returned by the
        NPC_F_MPI_MON_DNP_CreateDevNetPortServer function
        usLocalPort                //[IN] local listening port on the PC
    );
    printf("The deleting succeeds.\n");
}

//CODE AREA
_NPC_CLEAR:
{
    if ( hDevNetPortServer )
    {
        //Destroy a port mapping service.
        NPC_F_MPI_MON_DNP_DestroyDevNetPortServer(
            hDevNetPortServer       //[IN] service handle returned by the
            NPC_F_MPI_MON_DNP_CreateDevNetPortServer function
        );
        hDevNetPortServer = NULL;

        printf("The destroying succeeds.\n");
    }
}

```

```
    }  
}  
  
return 0;  
}
```