

EasyDarwin 开源流媒体平台系列文档

EasyDarwin 开源流媒体平台 交互协议



版本 v1.0

EasyDarwin 开源流媒体社区

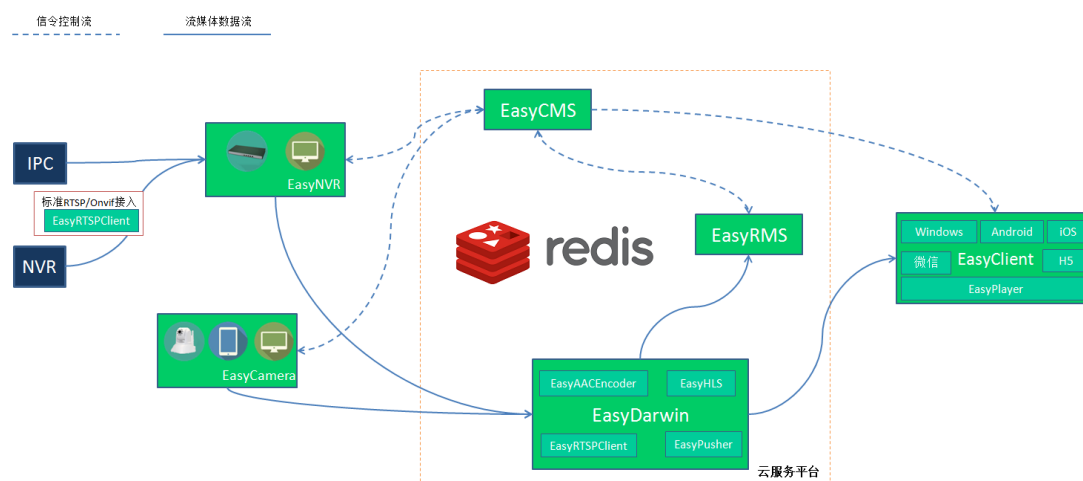
2016 年 5 月

目录

| | |
|---|----|
| Part I 系统架构..... | 3 |
| 1. 整体架构..... | 3 |
| 2. 模块划分..... | 4 |
| Part II 交互协议..... | 5 |
| 1. 协议设计原则..... | 5 |
| 2. 消息头部分..... | 5 |
| 3. 消息的数据部分..... | 5 |
| 3.1 客户端发送请求消息时的 Header 标签..... | 5 |
| 3.2 服务器响应客户端消息时的 Header 标签..... | 6 |
| 4. 消息类型定义..... | 6 |
| 4.1 EasyNVR 注册到 EasyCMS..... | 7 |
| 4.2 EasyCMS 向 EasyNVR 请求通道摄像机实时码流推送..... | 8 |
| 4.3 EasyCMS 请求停止 EasyNVR 通道摄像机实时码流推送..... | 10 |
| 4.4 EasyClient 向 EasyCMS 请求设备列表信息..... | 11 |
| 4.5 EasyClient 向 EasyCMS 请求具体设备信息..... | 12 |
| 4.6 EasyClient 向 EasyCMS 请求设备的码流..... | 13 |
| 4.7 EasyDarwin 向 EasyCMS 请求停止码流推送..... | 13 |
| 4.8 设备向 CMS 上传快照..... | 14 |
| 4.9 EasyClient 向 EasyCMS 请求控制云台..... | 15 |
| 4.10 EasyCMS 向设备请求控制云台..... | 15 |
| 4.11 EasyClient 向 EasyCMS 请求控制预置位..... | 17 |
| 4.12 EasyCMS 向设备请求控制预置位..... | 17 |
| 4.13 设备向 EasyDarwin 推送流媒体..... | 19 |
| Part III 附录..... | 20 |
| 1. 状态码定义..... | 20 |
| 2. 对象地址编码..... | 21 |
| Part IV 更新日志..... | 22 |

Part I 系统架构

1. 整体架构



2. 模块划分

- EasyCMS 中心管理服务器：
负责所有设备的接入与管理，各个服务单元的接口接入 (EasyCamera、EasyNVR、EasyClient)；
- EasyDarwin 流媒体服务器：
负责流媒体分发、转码、HLS发布等功能；
- EasyCamera
对接与EasyCMS的信令交互、对接与EasyDarwin的流媒体交互、对接与摄像机方案的交互 (Arm Linux内部调用)；
- EasyNVR
对接与EasyCMS的信令交互、对接与EasyDarwin的流媒体交互、对接与各通道的摄像机的Onvif/RTSP交互；

Part II 交互协议

1. 协议设计原则

客户端与服务器通讯过程中，信息以消息为载体进行传输，每条消息都包含有消息头和消息的数据部分构成：

- 消息头部分以 HTTP 协议构成，以\r\n\r\n 结尾；
- 数据部分采用 json 文本协议，保证协议高可读性和扩展性。

2. 消息头部分

HTTP 头部分为 HTTP 请求头部与 HTTP 响应头部，以客户端建立会话举例：

```
GET http://www.easydarwin.org/ HTTP/1.1
```

```
CSeq: 1
```

```
Content-Length: 216
```

```
<json 数据部分>
```

服务器响应：

```
HTTP/1.1 200 OK
```

```
CSeq: 1
```

```
Content-Length: 158
```

```
<json 数据部分>
```

3. 消息的数据部分

json 部分细分为两大标签<Header></Header>和<Body></Body>

3.1 客户端发送请求消息时的 Header 标签

```
"Header": {  
    "Version": "1.0",  
    "AppType": "EasyCamera",  
    "TerminalType": "ARM_Linux",  
    "CSeq": "1",  
    "MessageType": "MSG_CS_REGISTER_REQ"  
}
```

格式说明:

Version: 协议版本;

TerminalType: 客户端应用程序类型 ARM_Linux、Android、IOS、WEB、PC 等等;

AppType: 终端类型, 设备采集端 (EasyNVR/EasyCamera/EasyPusher)、服务端 (EasyCMS、EasyDarwin/EasyRMS)、客户端 (EasyClient /EasyPlayer);

MessageType: 消息类型, 如登录消息、列表请求消息等等, 详情请参照[“消息类型定义”](#);

CSeq: 请求消息的序列号, 该字段在请求和响应消息中成对出现, 目的用于确认请求消息所对应的响应消息。同一客户端在同一次会话过程中不应出现 CSeq 值重复的情况, 服务器对该条命令响应的时候返回的 CSeq 值与客户端传递过来 CSeq 值的相同。

3.2 服务器响应客户端消息时的 Header 标签

```
“Header”: {
    “Version”: “1.0”,
    “AppType”: “EasyCamera”,
    “TerminalType”: “ARM_Linux”,
    “CSeq”: “1”,
    “MessageType”: “MSG_SC_REGISTER_ACK”
}
```

4. 消息类型定义

| 定义 | 描述 |
|---|---------------------------------|
| MSG_DS_REGISTER_REQ | 设备向 EasyCMS 服务器发送的注册请求(周期性发送) |
| MSG_SD_REGISTER_ACK | 设备注册响应 |
| MSG_SD_PUSH_STREAM_REQ | EasyCMS 向设备请求实时码流推送到 EasyDarwin |
| MSG_DS_PUSH_STREAM_ACK | 推流请求响应 |
| MSG_SD_STREAM_STOP_REQ | EasyCMS 向设备请求停止实时码流推送 |
| MSG_DS_STREAM_STOP_ACK | 停止命令响应 |
| <i>http://[ip]:[port]/api/getdevicelist</i> | 客户端向 EasyCMS 请求设备列表 |
| MSG_SC_DEVICE_LIST_ACK | 在线设备列表响应 |
| <i>http://[ip]:[port]/api/getdeviceinfo?device=[Serial]</i> | 客户端向 EasyCMS 请求设备详细信息 |
| MSG_SC_DEVICE_INFO_ACK | 设备具体信息响应 |
| <i>http://[ip]:[port]/api/getdevicestream?device=001002000001&channel=0&protocol=RTSP&reserve=1</i> | 客户端向 EasyCMS 请求设备的码流 |
| MSG_SC_GET_STREAM_ACK | 携带流媒体地址响应 |
| <i>http://[ip]:[port]/api/freedevicestream?device=001002000001&channel=01&protocol=RTSP&reserve=1</i> | 客户端向 EasyCMS 请求释放设备码流 |
| MSG_SC_FREE_STREAM_ACK | 释放请求响应 |
| MSG_DS_POST_SNAP_REQ | 设备向 EasyCMS 上传视频快照 |

| | |
|---|-------------------------|
| MSG_SD_POST_SNAP_ACK | 快照响应 |
| <i>http://[ip]:[port]/api/ ptzcontrol?device=001002000001&channel=0&protocol=onvif&reserve=1 &actiontype=single&command=down&speed=5&protocol=onvif</i> | 客户端向 EasyCMS 请求控制设备 ptz |
| MSG_SC_PTZ_CONTROL_ACK | ptz 控制请求回应 |
| MSG_SD_CONTROL_PTZ_REQ | EasyCMS 向设备请求控制 ptz |
| MSG_DS_CONTROL_PTZ_ACK | 控制 ptz 响应 |
| <i>http://[ip]:[port]/api/ presetcontrol?device=001001000058&channel=0&command=goto&preset=1&protocol=onvif</i> | 客户端向 EasyCMS 请求控制设备预置位 |
| MSG_SC_PRESET_CONTROL_ACK | 预置位控制请求回应 |
| MSG_SD_CONTROL_PRESET_REQ | EasyCMS 向设备请求控制预置位 |
| MSG_DS_CONTROL_PRESET_ACK | 控制预置位响应 |

4.1 EasyNVR 注册到 EasyCMS

1) 请求: MSG_DS_REGISTER_REQ

```
{
  "EasyDarwin": {
    "Header": {
      "Version": "1.0",
      "AppType": "EasyNVR",
      "TerminalType": "ARM_Linux",
      "CSeq": "1",
      "MessageType": "MSG_DS_REGISTER_REQ"
    },
    "Body": {
      "Serial": "001002000001",
      "Name": "EasyNVR Model 3535",
      "Tag": "demo",
      "Token": "a939cd77c7cb4af6b2bd8f2090562b91",
      "ChannelCount": "16",
      "Channels":
      [
        {
          "Channel": "1",
          "Name": "Camera01",
          "Status": "online/offline"
        },
        {
          "Channel": "2",
          "Name": "Camera02",
```

```

        "Status": "online/offline"
    },
    .....
    {
        "Channel": "16",
        "Name": "Camera16",
        "Status": "online/offline"
    }
]
}
}
}
}

```

格式说明:

Serial: 设备序列号;

Name: 设备名称;

Token: 设备授权码, 也就是密码, 加密后传输;

ChannelCount: EasyNVR 当前接入的摄像机通道数量;

Channels: 各个通道接入的摄像机信息, 包括摄像机序列号、摄像机名称的数组;

注意: 如果 AppType 为 EasyCamera, 那么只需要携带 Serial、Name、Token 属性;

2) 响应: MSG_SD_REGISTER_ACK

```

{
    "EasyDarwin": {
        "Header": {
            "Version": "1.0",
            "CSeq": "1",
            "MessageType": "MSG_SD_REGISTER_ACK",
            "ErrorNum": "200",
            "ErrorString": "Success OK!"
        }
        "Body": {
            "Serial": "001002000001"
        }
    }
}

```

心跳间隔默认 30S, 超时 90S;

服务器 90S 后如果没有接收到客户端心跳, 则认为客户端掉线;

客户端 90S 后如果没有接收到服务器响应, 则认为与服务器链接断开, 重新建立连接。

4.2 EasyCMS 向 EasyNVR 请求通道摄像机实时码流推送

1) 请求: MSG_SD_PUSH_STREAM_REQ

```

{
    "EasyDarwin": {
        "Header": {

```



```

        "Version": "1.0",
        "MessageType": "MSG_SD_PUSH_STREAM_REQ",
        "CSeq": "2"
    },
    "Body": {
        "Serial": "001002000001",
        "Channel": "0",
        "Protocol": "RTSP",
        "Server_IP": "8.8.8.8",
        "Server_Port": "554",
        "StreamID": "QWERASDFGHJKZCVV",
        "Reserve": "1",
        "From": "ASDFGHJKLZXCVBNM",
        "To": "ASDFGHJKLQWERTYUIO",
        "Via": "ZXCVBNMASDFGHJKL"
    }
}
}
}

```

格式说明:

Serial: 设备序列号;

Channel: 摄像机通道号;

Protocol: 指定流媒体传输协议, 如 RTSP、自定义协议等等;

Server_IP: 推送码流到的流媒体服务器地址;

Server_Port: 推送码流到的流媒体服务器端口;

StreamID: EasyCMS 为控制拉流和推流的合法性生成唯一的 StreamID 并存储在 Redis 上, 由 EasyDarwin 进行判断是否合法;

From: EasyCMS 接收 Client 访问的 SessionID;

To: EasyCMS 向 Device 发送报文的 SessionID;

Via: EasyCMS 的 ServiceID;

Reserve: 预留不同的码流, 0流畅, 1标清, 2高清, 3超清;

如果设备端都是推送RTSP的话, 就不用再加Protocol字段了, 如果加了, 停止推流也应该加上。

2) 响应: MSG_DS_PUSH_STREAM_ACK

```

{
    "EasyDarwin": {
        "Header": {
            "Version": "1.0",
            "CSeq": "2",
            "MessageType": "MSG_DS_PUSH_STREAM_ACK",
            "ErrorNum": "200",
            "ErrorString": "Success OK"
        }
    }
    "Body": {
        "Serial": "001002000001",
        "Channel": "1",

```

```

        "Reserve": "1",
        "Server_IP": "9.9.9.9",
        "Server_Port": "554",
        "From": "ASDFGHJKLZXCVBNM",
        "To": "ASDFGHJKLQWERTYUIO",
        "Via": "ZXCVBNMASDFGHJKL"
    }
}
}

```

Server_IP 和 Server_Port 在此返回的是设备实际推流的 IP 和端口。

4.3 EasyCMS 请求停止 EasyNVR 通道摄像机实时码流推送

1) 请求: MSG_SD_STREAM_STOP_REQ

```

{
    "EasyDarwin": {
        "Header": {
            "Version": "1.0",
            "MessageType": "MSG_SD_STREAM_STOP_REQ",
            "CSeq": "3"
        },
        "Body": {
            "Serial": "001002000001",
            "Channel": "1",
            "Protocol": "RTSP", //请求推流有，那么停止推流也应该有
            "Reserve": "1", //码流类型
            "From": "ASDFGHJKLZXCVBNM",
            "To": "ASDFGHJKLQWERTYUIO",
            "Via": "ZXCVBNMASDFGHJKL"
        }
    }
}

```

2) 响应: MSG_DS_STREAM_STOP_ACK

```

{
    "EasyDarwin": {
        "Header": {
            "Version": "1.0",
            "CSeq": "3",
            "MessageType": "MSG_DS_STREAM_STOP_ACK",
            "ErrorNum": "200",
            "ErrorString": "Success OK"
        }
        "Body": {
            "Serial": "001002000001",

```

```
    "Channel": "1",
    "Protocol": "RTSP", //请求推流有，那么停止推流也应该有
    "Reserve": "1", //码流类型
    "From": "ASDFGHJKLZXCVBNM",
    "To": "ASDFGHJKLQWERTYUIO",
    "Via": "ZXCVBNMASDFGHJKL"
  }
}
}
```

4.4 EasyClient 向 EasyCMS 请求设备列表信息

1) 请求: RESTful

http://[ip]:[port]/api/getdevicelist

2) 响应: MSG_SC_DEVICE_LIST_ACK

```
"EasyDarwin": {
  "Header": {
    "Version": "1.0",
    "MessageType": "MSG_SC_DEVICE_LIST_ACK",
    "ErrorNum": "200",
    "CSeq": "4",
    "ErrorString": "Success OK"
  },
  "Body": {
    "DeviceCount": "2",
    "Devices":
    [
      {
        "AppType": "EasyNVR",
        "Serial": "001002000001",
        "Name": "DeviceName_001",
        "Tag": "DeviceTag001",
        "TerminalType": "ARM_Linux"
      },
      {
        "AppType": "EasyCamera",
        "Serial": "000000000002",
        "Name": "DeviceName_002",
        "Tag": "DeviceTag002",
        "TerminalType": "ARM_Linux"
      }
    ]
  }
}
```

```
}

```

4.5 EasyClient 向 EasyCMS 请求具体设备信息

1) 请求: RESTful

http://[ip]:[port]/api/getdeviceinfo?device=[Serial]

格式说明: Serial: 设备序列号

2) 响应: MSG_SC_DEVICE_INFO_ACK

```
{
  "EasyDarwin": {
    "Header": {
      "Version": "1.0",
      "MessageType": "MSG_SC_DEVICE_INFO_ACK",
      "ErrorNum": "200",
      "CSeq": "5",
      "ErrorString": "Success OK"
    },
    "Body": {
      "Serial": "001002000001",
      "ChannelCount": "16",
      "Channels":
      [
        {
          "Channel": "1",
          "Name": "Camera01",
          "SnapURL": "XXXXXXXXXXXXXXXXX. JPEG",
          "Status": "online/offline"
        },
        {
          "Channel": "2",
          "Name": "Camera02",
          "SnapURL": "XXXXXXXXXXXXXXXXX. JPEG",
          "Status": "online/offline"
        },
        .....
        {
          "Channel": "16",
          "Name": "Camera16",
          "SnapURL": "XXXXXXXXXXXXXXXXX. JPEG",
          "Status": "online/offline"
        }
      ]
    }
  }
}
```

```
}
```

4.6 EasyClient 向 EasyCMS 请求设备的码流

1) 请求: RESTful

`http://[ip]:[port]/api/getdevicestream?device=001002000001&channel=01&protocol=RTSP
&reserve=1`

2) 响应: MSG_SC_GET_STREAM_ACK

```
{  
  "EasyDarwin": {  
    "Header": {  
      "Version": "1.0",  
      "MessageType": "MSG_SC_GET_STREAM_ACK",  
      "ErrorNum": "200",  
      "CSeq": "6",  
      "ErrorString": "Success OK"  
    }  
    "Body": {  
      "Serial": "001002000001",  
      "Channel": "0",  
      "URL": "live stream url",  
      "Protocol": "RTSP",  
      "reserve": "1"  
    }  
  }  
}
```

推流拉流都使用随机生成的 token 来进行合法性认证, 假设生成的 token 为 QWERASDFGHIJKZCVV, 则 URL 为 "rtsp://120.118.23.45:554/001002000001/01.sdp?token= QWERASDFGHIJKZCVV"

4.7 EasyDarwin 向 EasyCMS 请求停止码流推送

1) 请求: MSG_CS_FREE_STREAM_REQ

```
{  
  "EasyDarwin": {  
    "Header": {  
      "Version": "1.0",  
      "AppType": "EasyDarwin",  
      "TerminalType": "Arm_Linux",  
      "CSeq": "7",  
      "MessageType": "MSG_CS_FREE_STREAM_REQ"  
    }  
    "Body": {  
      "Serial": "001002000001",  
      "Channel": "1",  
      "Protocol": "RTSP",  

```

```
        "Reserve": "1"
    }
}
}
```

2) 响应: MSG_SC_FREE_STREAM_ACK

```
"EasyDarwin": {
    "Header": {
        "Version": "1.0",
        "MessageType": "MSG_SC_FREE_STREAM_ACK",
        "ErrorNum": "200",
        "CSeq": "7",
        "ErrorString": "Success OK"
    }
    "Body": {
        "Serial": "001002000001",
        "Channel": "1",
        "Protocol": "RTSP",
        "Reserve": "1"
    }
}
}
```

4.8 设备向 CMS 上传快照

1) 请求: MSG_DS_POST_SNAP_REQ

```
{
    "EasyDarwin": {
        "Header": {
            "Version": "1.0",
            "CSeq": "8",
            "MessageType": "MSG_DS_POST_SNAP_REQ"
        },
        "Body": {
            "Serial": "001002000001",
            "Channel", "01",
            "Type": "JPEG/IDR",
            "Time": "2015-07-20 12:55:30",
            "Image": "Base64Encode_Image_Data",
            "Reserve": "width=1080&height=720&codec=28"
        }
    }
}
```

2) 响应: MSG_SD_POST_SNAP_ACK

```
"EasyDarwin": {
  "Header": {
    "Version": "1.0",
    "CSeq": "8",
    "MessageType": "MSG_SD_POST_SNAP_ACK",
    "ErrorNum": "200",
    "ErrorString": "Success OK"
  }
  "Body": {
    "Serial": "001002000001",
    "Channel": "1"
  }
}
```

4.9 EasyClient 向 EasyCMS 请求控制云台

1) 请求: RESTful

`http://[ip]:[port]/api/ptzcontrol?device=001001000058&channel=0&actiontype=single&command=down&speed=5&protocol=onvif`

device: 设备序列号;

channel: 通道;

protocol: 控制模式, 包括 Onvif、SDK;

actiontype: ptz 控制模式, 分为连续 (Continuous)、单步 (Single);

command: ptz 控制命令包括转动、变焦等,

stop/up/down/left/right/zoomin/zoomout/focusin/focusout/aperturein/apertureout;

speed: ptz 控制速度。

2) 响应: MSG_SC_PTZ_CONTROL_ACK

```
"EasyDarwin": {
  "Body": {
    "Channel": "0",
    "Protocol": "ONVIF",
    "Reserve": "1",
    "Serial": "001001000058"
  },
  "Header": {
    "CSeq": "1",
    "ErrorNum": "200",
    "ErrorString": "Success OK",
    "MessageType": "MSG_SC_PTZ_CONTROL_ACK",
    "Version": "1.0"
  }
}
```

4.10 EasyCMS 向设备请求控制云台

1) 请求: MSG_SD_CONTROL_PTZ_REQ

```
{
  "EasyDarwin": {
    "Body": {
      "ActionType": "SINGLE",
      "Channel": "0",
      "Command": "DOWN",
      "From": "f6a221eec46b47dea8ae1a2bd11f8d02",
      "Protocol": "ONVIF",
      "Reserve": "1",
      "Serial": "001001000058",
      "Speed": "5",
      "To": "245d6ec33cd247b7b7524219552db4d8",
      "Via": "27823d2e8b6b4032b453d435a16b7be8"
    },
    "Header": {
      "CSeq": "1",
      "MessageType": "MSG_SD_CONTROL_PTZ_REQ",
      "Version": "1.0"
    }
  }
}
```

格式说明:

Serial: 设备序列号;

Channel: 摄像机通道号;

Protocol: 指定 ptz 控制方式, ONVIF 协议或者设备 SDK;

ActionType: ptz 控制类型, 包括连续或者单步;

Command: ptz 控制命令, 包括停止, 上下左右旋转, 变焦等等;

Speed: ptz 控制速度;

From: EasyCMS 接收 Client 访问的 SessionID;

To: EasyCMS 向 Device 发送报文的 SessionID;

Via: EasyCMS 的 ServiceID;

Reserve: 预留。

2) 响应: MSG_DS_CONTROL_PTZ_ACK

```
{
  "EasyDarwin": {
    "Body": {
      "Channel": "0",
      "From": "245d6ec33cd247b7b7524219552db4d8",
      "Protocol": "ONVIF",
      "Reserve": "1",
      "Serial": "001001000058",
      "To": "f6a221eec46b47dea8ae1a2bd11f8d02",
      "Via": "27823d2e8b6b4032b453d435a16b7be8"
    },
  },
}
```



```

    "Header": {
      "CSeq": "1",
      "ErrorNum": "200",
      "ErrorString": "Success OK",
      "MessageType": "MSG_DS_CONTROL_PTZ_ACK",
      "Version": "1.0"
    }
  }
}

```

4.11 EasyClient 向 EasyCMS 请求控制预置位

1) 请求: RESTful

http://[ip]:[port]/api/

presetcontrol?device=001001000058&channel=0&command=goto&preset=1&protocol=onvif

device: 设备序列号;

channel: 通道;

protocol: 控制模式, 包括 Onvif、SDK;

command: 预置位控制命令包括转到、设置、删除等, goto/set/remove;

preset: 预置位编号。

2) 响应: MSG_SC_PRESET_CONTROL_ACK

```

"EasyDarwin": {
  "Body": {
    "Channel": "0",
    "Protocol": "ONVIF",
    "Reserve": "1",
    "Serial": "001001000058"
  },
  "Header": {
    "CSeq": "1",
    "ErrorNum": "200",
    "ErrorString": "Success OK",
    "MessageType": "MSG_SC_PRESET_CONTROL_ACK",
    "Version": "1.0"
  }
}

```

4.12 EasyCMS 向设备请求控制预置位

1) 请求: MSG_SD_CONTROL_PRESET_REQ

```

{
  "EasyDarwin": {
    "Body": {
      "Channel": "0",
      "Command": "GOTO",
      "From": "f6a221eec46b47dea8ae1a2bd11f8d02",
      "Protocol": "ONVIF",

```

```
    "Reserve": "1",
    "Serial": "001001000058",
    "Preset": "3",
    "To": "245d6ec33cd247b7b7524219552db4d8",
    "Via": "27823d2e8b6b4032b453d435a16b7be8"
  },
  "Header": {
    "CSeq": "1",
    "MessageType": "MSG_SD_CONTROL_PRESET_REQ",
    "Version": "1.0"
  }
}
```

格式说明:

Serial: 设备序列号;

Channel: 摄像机通道号;

Protocol: 指定预置位控制方式, ONVIF 协议或者设备 SDK;

Command: 预置位控制命令, 包括转到、设置、删除等;

Preset: 预置位编号;

From: EasyCMS 接收 Client 访问的 SessionID;

To: EasyCMS 向 Device 发送报文的 SessionID;

Via: EasyCMS 的 ServiceID;

Reserve: 预留。

2) 响应: MSG_DS_CONTROL_PRESET_ACK

```
{
  "EasyDarwin": {
    "Body": {
      "Channel": "0",
      "From": "245d6ec33cd247b7b7524219552db4d8",
      "Protocol": "ONVIF",
      "Reserve": "1",
      "Serial": "001001000058",
      "To": "f6a221eec46b47dea8ae1a2bd11f8d02",
      "Via": "27823d2e8b6b4032b453d435a16b7be8"
    },
    "Header": {
      "CSeq": "1",
      "ErrorNum": "200",
      "ErrorString": "Success OK",
      "MessageType": "MSG_DS_CONTROL_PRESET_ACK",
      "Version": "1.0"
    }
  }
}
```

4.13 设备向 EasyDarwin 推送流媒体

标准 RTSP 推送流程: **ANNOUNCE/SETUP/PLAY/RTP**.

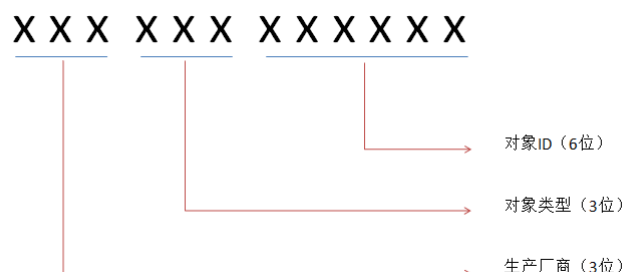
Part III 附录

1. 状态码定义

| 响应码 | 报文描述 | 定义 |
|-----|-------------------------------|----------------------------------|
| 200 | Success OK | 成功 |
| 201 | Success Created | 成功创建 |
| 202 | Success Accepted | 已接受用于处理，但处理尚未完成 |
| 204 | Success No Content | 已接收请求，但不存在要回送的信息 |
| 206 | Success Partial Content | 已接收请求，但要回送的信息不完整 |
| 301 | Redirect Permanent Moved | 请求的数据具有新的位置且更改是永久的。 |
| 302 | Redirect Temp Moved | 请求的数据临时具有不同 URI |
| 303 | Redirect See Other | 可在另一 URI 下找到对请求的响应 |
| 305 | Use Proxy | 必须通过位置字段中提供的代理来访问请求的资源 |
| 400 | Client Bad Request | 请求中有语法问题，或不能满足请求 |
| 401 | Client Unauthorized | 未授权客户端访问数据 |
| 402 | Payment Required | 需要付款,表示计费系统已有效 |
| 403 | Client Forbidden | 禁止，即使有授权也不需要访问 |
| 404 | Not Found | 服务器找不到给定的资源 |
| 405 | Method Not Allowed | 请求的方法不支持 |
| 407 | Proxy Authentication Required | 代理认证请求，客户机首先必须使用代理认证自身 |
| 408 | Request Timeout | 请求超时 |
| 409 | Conflict | 请求冲突 |
| 412 | Precondition Failed | 前提条件失败 |
| 415 | Unsupported Media Type | 服务器拒绝服务请求，因为不支持请求实体的格式 |
| 500 | Server Internal Error | 内部错误,因为意外情况，服务器不能完成请求 |
| 501 | Server Not Implemented | 未执行,服务器不支持请求 |
| 502 | Server Bad Gateway | 错误网关,服务器接收到来自上游服务器的无效响应 |
| 503 | Server Unavailable | 由于临时过载或无法获得服务护,服务器无法处理请求 |
| 505 | RTSP Version Not Supported | 不支持的 RTSP 版本 |
| 600 | Memcache Not Found | 找不到 Memcache 服务器 |
| 601 | Database Not Found | 找不到 Database 服务器 |
| 602 | User Not Found | 找不到用户信息（该用户不存在） |
| 603 | Device Not Found | 找不到设备信息（该设备不存在或者没有与请求用户绑定） |
| 604 | Session Not Found | 找不到会话信息(Session 过期或者不存在) |
| 605 | Service Not Found | 找不到请求的服务模块 |
| 620 | Password Error | 密码错误 |

| | | |
|-----|------------------|----------|
| 621 | Parse Error | XML 解析失败 |
| 622 | Permission Error | 没有权限 |
| | | |

2. 对象地址编码



所有设备Serial都是用12位字符串表示，前3位表示生产厂商，第4位-第6位表示对象类型，第7位-第12位表示对象ID；

接入商根据需求在不断的扩大，目前分配的为：

001: EasyDarwin

类型有如：

001: EasyCamera

002: EasyNVR

...

例如：EasyDarwin的ID为000001的EasyCamera设备

 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |

Part IV 更新日志

| 更新内容 | 版本号 | 更新日期 | 作者 |
|------------|--------|------------|--------|
| 1、定义基本通讯协议 | V1.0.0 | 2014-08-25 | Babosa |
| | | | |