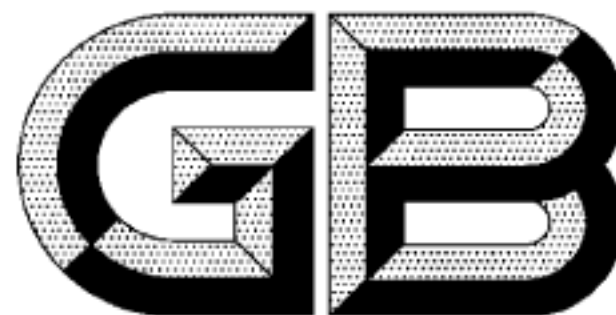


ICS 35.060
L 72



中华人民共和国国家标准

GB/T 35134—2017

物联网智能家居 设备描述方法

Smart home for internet of things—Device description method

2017-12-29 发布

2018-07-01 实施

中华人民共和国国家质量监督检验检疫总局 发布
中国国家标准化管理委员会

目 次

前言	III
1 范围	1
2 规范性引用文件	1
3 术语和定义	1
4 缩略语	2
5 设备的描述方法	2
6 设备功能对象类型	3
7 设备描述文件元素的定义域和编码	3
8 设备功能对象数据结构规范	4
附录 A (资料性附录) DDL 与 XML 格式对照	20
附录 B (资料性附录) 楼宇对讲机设备描述语言编写的设备描述文本示例	21
附录 C (资料性附录) 楼宇对讲机 XML 语言形式表达的设备描述文本示例	24

前 言

本标准按照 GB/T 1.1—2009 给出的规则起草。

本标准由中华人民共和国住房和城乡建设部提出。

本标准由全国智能建筑及居住区数字化标准化技术委员会(SAC/TC 426)归口。

本标准起草单位:广东瑞德智能科技股份有限公司、绿建智慧科技(北京)有限公司、中关村乐家智慧居住区产业技术联盟、秦皇岛尼特信息科技有限公司、厦门 ABB 振威电器设备有限公司、万科思自控信息(中国)有限公司、辽宁宜居尚雅科技有限公司、北京智慧通联科技有限公司、贵州天控自动化信息工程有限公司、成都清轻信息技术有限公司、成都思唯奇科技有限公司、成都博盛信息技术有限公司。

本标准主要起草人:汪军、卢仲宇、顾焯南、姚长标、张永刚、谢后贤、王钢、程卫东、林永强、洪佳雄、付伟、马立军、尚治宇、樊静静、邢树志、苑海娜、李黔蜀、沈燕。

物联网智能家居 设备描述方法

1 范围

本标准规定了物联网智能家居设备的描述方法、描述文件的格式要求、功能对象类型、描述文件元素的定义域和编码、描述文件的使用流程和功能对象数据结构。

本标准适用于智能家居系统中的所有家居设备,包括家用电器、照明系统、水电气热计量表、安全及报警系统和计算机信息设备、通信设备等。智能社区公共安全防范系统、公共设备监控系统、家庭信息采集及设备控制系统,以及所有面向家居设备的应用、服务的各种控制网络系统中的有关设备可参照使用。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件,仅注日期的版本适用于本文件。凡是不注日期的引用文件,其最新版本(包括所有的修改单)适用于本文件。

GB/T 1988 信息处理 信息交换用七位编码字符集

GB 2312 信息交换用汉字编码字符集 基本集

GB/T 35143—2017 物联网智能家居 数据和设备编码

3 术语和定义

下列术语和定义适用于本文件。

3.1

物联网智能家居 smart home for internet of things

以住宅为平台,融合建筑、网络通信、智能家居设备、服务平台,集系统、服务、管理为一体的高效、舒适、安全、便利、环保的居住环境。

3.2

智能家居设备 smart home device

具有网络通信功能,可自描述、发布并能与其他节点进行交互操作的家居设备。

3.3

智能家居系统 system of smart home

由智能家居设备通过某种网络通信协议,相互联结成为可交互控制管理的智能家居网络。

3.4

设备描述 device description

对设备自有功能和服务的表述。

3.5

设备描述方法 device description method

对设备自有功能和服务表述的方法。

3.6

设备描述文件 device description file

智能家居设备向智能家居系统发布的对自有功能和服务表述的自描述文件。

3.7

设备描述语言 device description language

设备功能对象数据结构描述的规定,对设备描述文件编写内容和格式化要求的详细规范。

4 缩略语

下列缩略语适用于本文件。

DDL 设备描述语言(Device Description Language)

XML 可扩展标记语言(Extensible Markup Language)

5 设备的描述方法

5.1 一般规定

物联网智能家居系统应实现对设备的管理和监控,向网关提供所有全部功能和服务信息。信息包括设备的功能、控制命令、交互接口及返回信息。

5.2 面向对象的设备描述内容

设备描述内容应以设备的功能为对象,应包含下列内容:

- a) 对象:与某一特定功能相关的所有数据元素的集合;
- b) 类:具有相同应用功能,执行类似的任务,归纳成标准数据结构,称为对象的“类”;
- c) 属性:对象执行的功能任务,称为对象的“属性”,属性可由若干层次组成。

5.3 设备描述文件的编码格式

设备描述文件按 GB/T 1988 和 GB 2312 的规定进行编译。

5.4 设备描述文件的界定符

字符串应由若干字段组成,字段之间通过一对界定符(起始标签和结束标签)界定:

- a) 对象界定符:“{/”和 “/}”;
- b) 部分(属性部分和操作部分)界定符:“[/”和 “/】”;
- c) 功能 Function 界定符:“</”和 “/>”;
- d) 基本界定符,如参数之间的分隔和功能内部之间的分隔,用“;”。

5.5 设备描述文件格式

设备描述文件的基本格式如下:

```
{/ Oi [ / Xj </B1;B2;B3;B4;B5/> /] /}
```

其中: Oi 是第 i 个对象类型标识

Xj 是该对象的第 j 个属性标识

B1:FunctionID(功能标识)

B2:FunctionName(功能名称)

B3:FunctionDescription(功能描述)

B4:FunctionType(功能执行方式)

B5:FunctionParameter(功能参数个数)

6 设备功能对象类型

6.1 一般规定

设备功能对象分为系统、基础、合成、关联和组合 5 种类型。对象分类描述文本结构及示例参见附录 A、附录 B、附录 C。

6.2 系统功能对象

系统功能对象应包括下列内容：

- a) System 对象：描述了设备发现和注册的基本工作模式；
- b) Device 对象：描述设备本身的详细信息，定义了设备的唯一标识码；
- c) File 对象：描述设备与系统间的数据传输过程。

6.3 基础功能对象

基础功能对象应包括下列内容：

- a) Analog Value 对象：描述节点中的模拟量及对模拟量的操作；
- b) Switch Value 对象：描述开关量，及对开关量的操作；
- c) Enum Value 对象：描述枚举量，及对枚举量的操作；
- d) Time 对象：描述时间数据，及对时间数据的操作；
- e) Event 对象：描述事件数据，及对事件数据的操作。

6.4 合成功能对象

合成功能对象应包括下列内容：

- a) Schedule 对象：描述时间触发型任务，包括设定时间及执行的任务；
- b) Action 对象：描述状态触发型的内部任务，包括触发状态及执行的内部任务。

6.5 关联功能对象

关联功能对象应包括下列内容：

- a) LoopAction 对象：描述外部事件触发的内部任务，包括外部的触发条件及内部执行的任务；
- b) ActionLoop 对象：描述内部活动触发的外部任务，包括内部触发条件及外部执行的任务。

6.6 组合功能对象

组合功能对象应包括下列内容：

- a) Group 对象：为一个辅助的功能对象，描述了设备功能的分组形式的使用；
- b) Combine Operation 对象：描述一系列功能的顺序的执行。

7 设备描述文件元素的定义域和编码

7.1 元素编码的定义域

7.1.1 数据类型

数据类型定义应包括下列内容：

- a) 字节类型：赋值范围为 0~255。

- b) 布尔类型:1 和 0,1 为真,0 为假。
- c) 文本类型应符合下列要求:
 - 1) 英文字符,最多 128 个字符(符合 GB/T 1988 的要求);
 - 2) 中文字符,最多 64 个汉字(符合 GB 2312 的要求);
 - 3) 文本中不得包含与 5.3 规定界定符相同的字符。
- d) 数组类型:数组长度范围为 0~255。

7.1.2 功能对象属性标识

功能对象通用属性标识应包括下列内容:

- a) 类的标识 Class ID:0~99 类的序号(0 为 System 类,唯一);
- b) 对象标识 Object ID:1~255(其中 ObjectID=1 为 System 类对象,是唯一的;其他的 ObjectID=2~99);
- c) 父对象组标识 Father Group:10~250 组标识,为 0 则未分组。

7.1.3 对象分组描述

对象分组描述应包括下列内容:

- a) Father Group 属性为类的分组,Group 类的 ObjectID;
- b) 同一组类用一个 Group 类来描述。

7.2 元素的定义域与编码规则

本条规定的编码规则,除对象类型、功能类型、元素类型按同类型的序号标识外。其他标识代码应按 GB/T 35143—2017 的规定执行。编码规则应符合下列要求:

- a) 用户不可更改,并应符合下列要求:
 - 1) ClassID:类的序号,由数字 1~99 组成;
 - 2) FunctionID:功能的序号,由 16 进制数组成。由于采用 8 位位组传送,只有一位数时,十位补 0,如 01,02,0A;
 - 3) Type:元素类型,由序号组成,在第 8 章表 1 属性说明表中定义。
- b) 用户(企业)自定义代码:
 - 1) ObjectID:对象标识,用户可以填入该元素对象的标识代码。缺省值为两位数字组成:父对象为 20,30...;则子对象为 21,22...;31,32...;
 - 2) FatherGroup:对象所隶属的组,用户可以填入该对象所属父对象组的标识代码。缺省值同上;
 - 3) ObjectName:对象名,用户可以填入本企业习惯的该功能对象名称,应为汉字或 GB 2312 的扩充。
- c) 布尔类型的数据:用 1 表示真;用 0 表示假。例如 FunctionReport,为 1 表示正常;为 0 表示不正常。

8 设备功能对象数据结构规范

8.1 设备描述语言的说明

设备描述语言用以描述资源对象的数据结构,应包括下列内容:

- a) 操作:
定义如下:

```

Function {
    ID;
    Name;
    Description;
    Type;
    ParamNum;
}

```

b) 属性:

按照规定的顺序和数据类型,使用文本方式表示。如表 1 所示。

表 1 属性说明表

名称	类型	说明		
ID	字节	标示 ID,在同一个类中不能重复		
Name	字符	名称		
Description	字符	实现功能的描述		
Type	字节	执行方式	值	说明
			1	网关主动向设备发送
			2	设备向网关返回
			3	设备自身执行
4	设备主动发送			
ParamNum	字节	参数个数		

8.2 设备描述语言对象的分类

8.2.1 系统功能对象

8.2.1.1 总则

系统功能对象包括 System 对象、Device 对象及 File 对象,并由相应的 System 类、Device 类、File 类实现数据描述。每个设备中,这三个对象必须实现,而且 System 对象和 Device 对象都仅有一个,File 对象必须存在至少一个设备文本传输类以实现设备描述文本的传输。

8.2.1.2 System 对象

System 对象描述了设备的基本工作模式,定义了设备统一的访问接口和访问方式,实现了与通信协议和通信设备无关的家庭网络设备的发现和注册功能。同时,System 对象描述了设备注册后对设备状态的查询及控制功能。

```

Class System {
//属性
    ClassID=0;
    AutoNotify;           //是否自动发送注册信息
                          //类型: Boolean
                          //1: 自动发送请求注册信息
}

```


//0:等待接收注册通知

//操作

```

RegisterNotify{
    ID:1
    Name:注册通知
    Description:系统向设备发送注册通知消息
    Type:1
    ParamNum:4    //为设备的唯一标识(UNID)码
};

PowerOnNotify{
    ID:2
    Name:设备注册
    Description:设备向系统发送请求注册消息
    Type:4
    ParamNum:4    //为设备的唯一标识(UNID)码
};

AssignID{
    ID:3
    Name:分配 ID
    Description:网关为设备分配 WorkID
    Type:1
    ParamNum:4    //为设备的唯一标识(UNID)码
};

CommReady{
    ID:4
    Name:设备在线状态
    Description:设备报告工作状态
    Type:2
    ParamNum:0
};

DeviceCheck{
    ID:5
    Name:检查设备状态
    Description:检查设备的当前状态
    Type:1
    ParamNum:0
};

Function SetStatus{
    ID:6
    Name:设置设备当前状态
    Description:
    Type:1
    ParamNum:1    //DeviceStatus
    
```

}

8.2.1.3 Device 对象

Device 对象描述了设备本身的详细信息,如以下类中的属性所示,并定义了设备型号的唯一标识码,同型号的设备使用相同的设备描述语言文本,以减少设备注册时的数据传输开销。

本条规定的产品代码、型号代码、版本号、产品类型代码、厂商代码,均应按 GB/T 35143—2017 第 5 章的规定执行。

```

Class Device {
//属性
    ClassID=1;
    DeviceName;           //字符:设备名称
    NationCode;          //字符:国家代码
    ProductID;           //字符:产品代码,
    ProductModel;        //字符:产品型号
    ProductVersion;      //产品版本号
                        //类型:字节 长度:2 字节
                        //高位:主版本号
                        //低位:副版本号
    Manufacturer;       //字符:制造商名称
    ProducingArea;      //字符:产地
}
//操作
    Function Check{
        ID:1
        Name:检查设备信息
        Description:
        Type:1
        ParamNum:0
    };
    Function ReportID{
        ID:2
        Name:报告设备标识字符序列
        Description:
        Type:2
        ParamNum:1       //设备标识字符序列
    };
}

```

8.2.1.4 File 对象

File 对象描述了设备与系统间的数据传输过程。设备在注册到系统时,一般情况下需要传输设备描述文本至系统,因此,必须至少实现一个描述设备描述文本传输的类。同时,File 对象还可以描述设备与系统间的其他数据传输过程,如图片、声音、资源文件等。

```

Class File{

```

```

//属性
ClassID=2;
ObjectID=1;
ObjectName=传输设备描述文本;
FileID; //文件的 ID
//类型:字节
//0:表示为设备描述文本文件

FileType; //数据传输类型
//类型:字节
//0:表示传输数据为设备描述文本
//1:表示为设备更新文件

//操作
Function DataCommand{
    ID:1
    Name:发送传输请求
    Description:发送请求传输数据消息
    Type:2
    ParamNum:0
};

Function DataCommandAck{
    ID:2
    Name:响应传输请求
    Description:发送数据传输响应消息
    Type:1
    ParamNum:2 //类型: Boolean
                //1:允许进行传输
                //2:拒绝进行传输
};

Function DataSend{
    ID:3
    Name:发送数据
    Description:发送数据
    Type:2
    ParamNum:2 //数据包号及数据
                //类型:字节
                //第一个字节为数据包号
                //以后为数据
};

Function DataSendAck{
    ID:4
    Name:接收数据验证
    Description:验证是否数据已经收到
    Type:1

```

```

        ParamNum:1           //接收到的数据包号
    };
    Function DataEnd{
        ID:5
        Name:数据传输完成
        Description:
        Type:1
        ParamNum:0
    };
}

```

8.2.2 基础功能对象

8.2.2.1 一般规定

基础功能对象包括 5 种,描述了设备的基本数据类型及其操作,并由这些基础功能对象组合,可以描述其他复杂的功能。

8.2.2.2 Analog Value 对象

Analog Value 对象描述了设备中的模拟量及对模拟量的操作。模拟量为具有上限和下限,并在其间具有连续值的数据,例如电视中的音量、亮度等。

```

Class AV {
//属性
    ClassID=3;
    ObjectID;
    ObjectName;
    MaxValue;           //最大值
                        //类型:字节 长度:2
    MinValue;           //最小值
                        //类型:字节 长度:2
    StepValue;          //步进值
                        //类型:字节 长度:2

    FatherGroup;
//操作
    Function Check{
        ID:1
        Name:查询当前值
        Description:
        Type:1
        ParamNum:0
    };
    Function Report{
        ID:2
        Name:报告当前值

```

```

        Description:
        Type:2
        ParamNum:1      //当前值
    };
Function AddOne{
    ID:3
    Name:当前值加一
    Description:
    Type:1
    ParamNum:0
};
Function MinusOne{
    ID:4
    Name:当前值减一
    Description:
    Type:1
    ParamNum:0
};
Function JumpTo{
    ID:5
    Name:赋当前值为
    Description:
    Type:1
    ParamNum:2      //所赋 AV 值
};
}

```

8.2.2.3 Switch Value 对象

Switch Value 对象描述了开关量,及对开关量的操作。开关量定义为只存在两种相反状态值的数据,如电源开关等。

```

Class SV {
//属性
    ClassID=4;
    ObjectID;
    ObjectName;
    OpenStatusName;      //开状态名称
                        //类型:字符
    CloseStatusName;    //关状态名称
                        //类型:字符

    FatherGroup;
//操作
    Function Check{
        ID:1

```



```

    Name:查询当前值
    Description:
    Type:1
    ParamNum:0
};
Function Report{
    ID:2
    Name:报告当前值
    Description:
    Type:2
    ParamNum:1 //当前值
};
Function ChangeStatus{
    ID:3
    Name:改变开关状态
    Description:
    Type:1
    ParamNum:1 //开关状态
                //类型: Boolean
                //1:开 0:关
};
}

```

8.2.2.4 Enum Valu 对象

Enum Value 对象描述了枚举量,及对枚举量的操作。枚举量定义为具有有限个确定的非连续值的数据,如空调的制热、制冷、除湿、通风就为一组枚举量。

```

Class EV {
//属性
    ClassID=5;
    ObjectID;
    ObjectName;
    ValueList {
    EnumValue; //枚举量的值
    EnumName; //枚举量的名称
    }
    FatherGroup;
//操作
    Function Check{
        ID:1
        Name:查询当前枚举值
        Description:
        Type:1
        ParamNum:0
    }
}

```

```

    };
    Function Report{
        ID:2
        Name:报告当前枚举量
        Description:
        Type:2
        ParamNum:1 //当前枚举量值
    };
    Function SetValue{
        ID:3
        Name:赋当前枚举量值为
        Description:
        Type:1
        ParamNum:1 //所赋枚举量的值
    };
}

```

8.2.2.5 Time 对象

Time 对象描述了时间数据,及对时间数据的操作。时间数据包括日期及时间,使用两种模式:时间跨度模式及时刻模式。

```

Class TimeValue {
//属性
    ClassID=6;
    ObjectID;
    ObjectName;
    YearMax;
    YearStep;
    MonthMax;
    MonthStep;
    DayMax;
    DayStep;
    HourMax;
    HourStep;
    MinuteMax;
    MinuteStep;
    SecondMax;
    SecondStep;
    TimeType; //Boolean:表示时间数据的类型
                //0:时间长度 1:时刻

    FatherGroup;
//操作
    Function Check{
        ID:1

```

```

    Name:查询时间
    Description:
    Type:1
    ParamNum:0
};
Function Report{
    ID:2
    Name:报告时间
    Description:
    Type:2
    ParamNum:6 //Year Month Date Hour Minute Second
};
Function SetTime{
    ID:3
    Name:设置时间
    Description:
    Type:1
    ParamNum:6 //Year Month Date Hour Minute Second
};
Function AddTime{
    ID:4
    Name:增加时间
    Description:
    Type:1
    ParamNum:6 //Year Month Date Hour Minute Second
};
}

```

8.2.2.6 Event 对象

Event 对象描述了事件数据,及对事件数据的操作。事件对象为设备内部触发而产生的操作,声称一种消息。例如报警信息等,并定义事件的级别以及类型。

本条规定的事件的级别以及类型的标识代码应按 GB/T 35143—2017 的 6.3、第 8 章、第 9 章各类故障类型级别的规定执行。

```

Class EventVaule{
//属性
    ClassID=7;
    ObjectID;
    ObjectName;
    EventLevel; //Byte:事件的级别
                //0:全局警报
                //1:通知
                //2:正常
    EventType; //Byte:事件的类型
}

```

```

//0:未知
//1:主控设备关闭
//2:断电
//3:报警信息
//4:设备关闭
//文本:事件信息
EventInfo;
EventDescription; //文本:事件的描述
//操作
Function Check{
    ID:1
    Name:查询事件
    Description:
    Type:1
    ParamNum:0
};
Function Report{
    ID:2
    Name:报告事件
    Description:
    Type:2
    ParamNum:0
};
Function SetEvent{
    ID:3
    Name:设置事件
    Description:
    Type:1
    ParamNum:0
};
Function RaiseEvent{
    ID:4
    Name:事件发生
    Description:
    Type:4
    ParamNum:4 // ClassID, ObjectID, Level, Info
}
}

```

8.2.3 合成功能对象

8.2.3.1 一般规定

合成功能对象描述了基本功能对象的合成使用方式,其中定义了两种使用较多的功能对象。

8.2.3.2 Schedule 对象

Schedule 对象描述了时间触发型任务,包括设定时间及执行的任务。

```

Class Schedule{
//属性
    ClassID=8;
    ObjectID;
    ObjectName;
    TimeObjectID;
    TaskObjectID;
    FatherGroup;
//操作
    Function Check{
        ID:1
        Name:检查时间及任务
        Description:
        Type:1
        ParamNum:0
    };
    Function Report{
        ID:1
        Name:报告时间及任务
        Description:
        Type:2
        ParamNum:0
    };
    Function EnableSchedule{
        ID:1
        Name:设定当前计划是否有效
        Description:
        Type:1
        ParamNum:1           //是否有效
    };
}

```

8.2.3.3 Action 对象

Action 对象描述了状态触发型的内部任务,包括触发状态及执行的内部任务。

```

Class Action {
//属性
    ClassID=9;
    ObjectID;
    ObjectName;
    ActionObejctID;

```



```

        TaskObjectID;
        FatherGroup;
    //操作
    Function Check{
        ID:1
        Name: 查询 Action 执行情况
        Description:
        Type:1
        ParamNum:0
    };
    Function Report{
        ID:2
        Name:报告 Action 执行情况
        Description:
        Type:2
        ParamNum:1
    };
    Function EnableAction{
        ID:3
        Name:设置 Action 是否有效
        Description:
        Type:1
        ParamNum:1           //是否有效
    };
}

```

8.2.4 关联功能对象

8.2.4.1 LoopAction 对象

LoopAction 对象描述了外部事件触发的内部任务,包括外部的触发事件及内部执行的任务。

当设备具有 LoopAction 对象时,如果接收到了系统中的 Event 对象消息,设备可以判断 Event 对象的级别,来确定是否需要执行相应的操作。

例如:一个火警 Event 消息被发送到系统中后,如果其他设备中具有针对警报级别的 LoopAction 对象,其对应的操作为一个执行关机操作的 SV 对象,那么接收设备就会立刻关机。而又如钥匙开门的消息发送时,具有相应级别的消息处理操作的设备可以实现开空调、开电视、开热水器等操作。

```

Class LoopAction{
    //属性
    ClassID=10;
    ObjectID;
    ObjectName;
    ExternalEventLevel;
    ExternalEventType;
    ExternalEventInfo;
}

```

```

    InternalObjectID;
//操作
    Function Check{
        ID:1
        Name: 检查 LoopAction 状态
        Description:
        Type:1
        ParamNum:0
    };
    Function Report{
        ID:2
        Name: 报告 LoopAction 状态
        Description:
        Type:2
        ParamNum:0
    };
    Function EnableLoopAction{
        ID:1
        Name: 是否执行
        Description:
        Type:1
        ParamNum:1
    };
}

```

8.2.4.2 ActionLoop 对象

ActionLoop 对象描述了内部活动触发的外部任务,包括内部触发条件及外部执行的任务。

```

Class ActionLoop{
//属性
    ClassID=11;
    ObjectID;
    ObjectName;
    ActionLoopLevel; //任务的等级
    ConditionObjectID; //内部状态对象
    ExternalUNID; //外部执行设备的标识
    ExternalObjectID;
    FatherGroup;
//操作
    Function Check{
        ID:1
        Name:检查执行条件
        Description:
        Type:1

```

```

        ParamNum:0
    };
    Function Report{
        ID:1
        Name:报告执行的条件
        Description:
        Type:2
        ParamNum:0
    };
    Function EnableActionLoop{
        ID:1
        Name:设置是否执行
        Description:
        Type:1
        ParamNum:0
    };
}

```

8.2.5 组合功能对象

8.2.5.1 Group 对象

Group 对象为一个辅助的功能对象,描述了设备功能的分组形式的使用。

```

Class Group {
//属性
    ClassID=12;
    ObjectID;
    ObjectName;
    GroupList {
        ObjectID;
    };
    FatherGroup;
//操作
    Function Check{};
    Function Report{};
}

```

8.2.5.2 Combine Operation 对象

Combine Operation 对象描述了一系列功能的顺序的执行。

(如果需要执行某些组合操作时,可能需要使用。如果将这种操作作为开关类,则不能体现为协议)

```

Class CO {
//属性
    ClassID=13;
    ObjectID;
}

```

```
ObjectName;  
OperationType;  
OperationList {  
    ObjectID;  
    FunctionID;  
  
};  
FatherGroup;  
//操作  
Function Operate{};           //开始操作  
}
```

附录 A
(资料性附录)
DDL 与 XML 格式对照

开关类示例及设备描述语言的语句说明见表 A.1。

表 A.1 开关类示例及设备描述语言的语句说明

设备描述文件语句	XML 格式语句及说明
{/ [4; 10; 待机; 30/]	<SwitchList> 类列表起始界定符(下面可能有几个相同的类) <Switch ClassID="4"> 开关类对象所属类的序号=4 <ObjectID>10</ObjectID> 对象标识号(控制码) <ObjectName>待机</ObjectName> 对象名 <FatherGroup>30</FatherGroup> 对象所隶属父对象标识号(入口码)
[/ </ 01; Check; 1; 1; /]	<FunctionList> 功能列表(下面有几个功能) <Function> 功能起始界定符 <FunctionID>01</FunctionID> 功能的序号 <FunctionName>Check</FunctionName> 功能名:检查状态 <IsCommand>1</IsCommand> 是否为控制命令 <IsNeedReturn>1</IsNeedReturn> 是否需要返回结果 </Function> 功能结束界定符
</ 02; Report; 1; 1; /]	<Function> 功能起始界定符 <FunctionID>02</FunctionID> 功能的序号 <FunctionName>Report</FunctionName> 功能名:报告状态 <IsCommand>1</IsCommand> 是否为控制命令 <IsNeedReturn>1</IsNeedReturn> 是否需要返回结果 </Function> 功能结束界定符
</ 03; ChangeStatus; 1; 1; /] /]	<Function> 功能起始界定符 <FunctionID>03</FunctionID> 功能的序号 <FunctionName>ChangeStatus</FunctionName> 功能名:改变状态 <IsCommand>1</IsCommand> 是否为控制命令 <IsNeedReturn>1</IsNeedReturn> 是否需要返回结果 </Function> 功能结束界定符 </FunctionList> 功能列表结束界定符
/] /]	</Switch> 开关类结束界定符 </SwitchList> 类列表结束界定符

附录 B

(资料性附录)

楼宇对讲机设备描述语言编写的设备描述文本示例

楼宇对讲机设备描述语言编写的设备描述文本示例：

```

{/
  [/0;1/]
  [/
    </01;注册通知;1;1;4/>
    </02;设备注册;1;4;4/>
    </03;分配 ID;1;1;4/>
    </04;设备在线状态;1;2;0/>
    </05;检查设备状态;1;1;0/>
    </06;设置设备当前状态;1;1;1/>
  /]
/}
{/
  [/1;楼宇对讲机;01;020301;01;0001;01;广东;/]
  [/
    </01;Check;1;1;0/>
    </02;Report;1;1;0/>
  /]
/}
{/
  [/4;0031;待机 ;30/]
  [/
    </01;Check;1;1;0/>
    </02;Report;1;1;0/>
    </03;ChangStatus;1;1;0/>
  /]
/}
{/
  [/4;1501;呼叫;30/]
  [/
    </01;Check;1;1;0/>
    </02;Report;1;1;0/>
    </03;ChangStatus;1;1;0/>
  /]
/}

```

```
{/
  [/4;1502;接听;30/]
  [/
    </01;Check;1;1;0/>
    </02;Report;1;1;0/>
    </03;ChangStatus;1;1;0/>
  /]
/}
{/
  [/4;1503;挂机;30/]
  [/
    </01;Check;1;1;0/>
    </02;Report;1;1;0/>
    </03;ChangStatus;1;1;0/>
  /]
/}
{/
  [/4;1504;抓拍;30/]
  [/
    </01;Check;1;1;0/>
    </02;Report;1;1;0/>
    </03;ChangStatus;1;1;0/>
  /]
/}
  [/4;1505;录音;30/]
  [/
    </01;Check;1;1;0/>
    </02;Report;1;1;0/>
    </03;ChangStatus;1;1;0/>
  /]
/}
{/
  [/4;1506;开锁;30/]
  [/
    </01;Check;1;1;0/>
    </02;Report;1;1;0/>
    </03;ChangStatus;1;1;0/>
  /]
/}
  [/5;1507;门锁状态;00;未开锁;01;已开锁;30/]
  [/
    </01;Check;1;1;0/>
    </02;Report;1;1;0/>
  /]
```

```
    </03;ChangStatus;1;1;0/>
  /]
/}
{/
  [/5;1508;门铃状态;00;空闲状态;01;振铃状态;02;通话状态;30/]
  [/
    </01;Check;1;1;0/>
    </02;Report;1;1;0/>
    </03;ChangStatus;1;1;0/>
  /]
/}
```

附录 C

(资料性附录)

楼宇对讲机 XML 语言形式表达的设备描述文本示例

楼宇对讲机 XML 语言形式表达的设备描述文本示例：

```

<? xml version = "1.0" ?>
<ObjectList>

  <System ClassID = "0">
    <System>
      <ObjectID>1</ObjectID>
      <FunctionList>
        <Function>
          <FunctionID>01</FunctionID>
          <FunctionName>注册通知</FunctionName>
          <IsCommand>1</IsCommand>
          <IsNeedReturn>1</IsNeedReturn>
          <ParaNum>4</ParaNum>
        </Function>
        <Function>
          <FunctionID>02</FunctionID>
          <FunctionName>设备注册</FunctionName>
          <IsCommand>1</IsCommand>
          <IsNeedReturn>4</IsNeedReturn>
          <ParaNum>4</ParaNum>
        </Function>
        <Function>
          <FunctionID>03</FunctionID>
          <FunctionName>分配 ID</FunctionName>
          <IsCommand>1</IsCommand>
          <IsNeedReturn>1</IsNeedReturn>
          <ParaNum>4</ParaNum>
        </Function>
        <Function>
          <FunctionID>04</FunctionID>
          <FunctionName>设备在线状态</FunctionName>
          <IsCommand>1</IsCommand>
          <IsNeedReturn>2</IsNeedReturn>
          <ParaNum>0</ParaNum>
        </Function>
        <Function>
          <FunctionID>05</FunctionID>

```

```

        <FunctionName>检查设备状态</FunctionName>
        <IsCommand>1</IsCommand>
        <IsNeedReturn>1</IsNeedReturn>
        <ParaNum>0</ParaNum>
    </Function>
    <Function>
        <FunctionID>06</FunctionID>
        <FunctionName>设置设备当前状态</FunctionName>
        <IsCommand>1</IsCommand>
        <IsNeedReturn>1</IsNeedReturn>
        <ParaNum>1</ParaNum>
    </Function>
</FunctionList>
</System>

<Device ClassID = "1">
    <ObjectID>1</ObjectID>
    <ObjectName>楼宇对讲机</ObjectName>
    <NationCode>01</NationCode>
    <ProductID>01</ProductID>
    <ProductModel>020301</ProductModel>
    <ProductVersion>0001</ProductVersion>
    <Manufacturer>01</Manufacturer>
    <ProducingArea>广东</ProducingArea>
    <FunctionList>
        <Function>
            <FunctionID>01</FunctionID>
            <FunctionName>Check</FunctionName>
            <IsCommand>1</IsCommand>
            <IsNeedReturn>1</IsNeedReturn>
            <ParaNum>0</ParaNum>
        </Function>
        <Function>
            <FunctionID>02</FunctionID>
            <FunctionName>Report</FunctionName>
            <IsCommand>1</IsCommand>
            <IsNeedReturn>1</IsNeedReturn>
            <ParaNum>0</ParaNum>
        </Function>
    </FunctionList>
</Device>
<SwitchList ClassID = "4">
    <Switch>

```



```

<ObjectID>10</ObjectID>
<ObjectName>待机</ObjectName>
<FatherGroup>30</FatherGroup>
<FunctionList>
  <Function>
    <FunctionID>01</FunctionID>
    <FunctionName>Check</FunctionName>
    <IsCommand>1</IsCommand>
    <IsNeedReturn>1</IsNeedReturn>
    <ParaNum>0</ParaNum>
  </Function>
  <Function>
    <FunctionID>02</FunctionID>
    <FunctionName>Report</FunctionName>
    <IsCommand>1</IsCommand>
    <IsNeedReturn>1</IsNeedReturn>
    <ParaNum>0</ParaNum>
  </Function>
  <Function>
    <FunctionID>03</FunctionID>
    <FunctionName>ChangeStatus</FunctionName>
    <IsCommand>1</IsCommand>
    <IsNeedReturn>1</IsNeedReturn>
    <ParaNum>0</ParaNum>
  </Function>
</FunctionList>
</Switch>
<Switch>
  <ObjectID>1501</ObjectID>
  <ObjectName>呼叫</ObjectName>
  <FatherGroup>30</FatherGroup>
  <FunctionList>
    <Function>
      <FunctionID>01</FunctionID>
      <FunctionName>Check</FunctionName>
      <IsCommand>1</IsCommand>
      <IsNeedReturn>1</IsNeedReturn>
      <ParaNum>0</ParaNum>
    </Function>
    <Function>
      <FunctionID>02</FunctionID>
      <FunctionName>Report</FunctionName>
      <IsCommand>1</IsCommand>

```

```

        <IsNeedReturn>1</IsNeedReturn>
        <ParaNum>0</ParaNum>
    </Function>
    <Function>
        <FunctionID>03</FunctionID>
        <FunctionName>ChangeStatus</FunctionName>
        <IsCommand>1</IsCommand>
        <IsNeedReturn>1</IsNeedReturn>
        <ParaNum>0</ParaNum>
    </Function>
</FunctionList>
</Switch>
<Switch>
    <ObjectID>1502</ObjectID>
    <ObjectName>接听</ObjectName>
    <FatherGroup>30</FatherGroup>
    <FunctionList>
        <Function>
            <FunctionID>01</FunctionID>
            <FunctionName>Check</FunctionName>
            <IsCommand>1</IsCommand>
            <IsNeedReturn>1</IsNeedReturn>
            <ParaNum>0</ParaNum>
        </Function>
        <Function>
            <FunctionID>02</FunctionID>
            <FunctionName>Report</FunctionName>
            <IsCommand>1</IsCommand>
            <IsNeedReturn>1</IsNeedReturn>
            <ParaNum>0</ParaNum>
        </Function>
        <Function>
            <FunctionID>03</FunctionID>
            <FunctionName>ChangeStatus</FunctionName>
            <IsCommand>1</IsCommand>
            <IsNeedReturn>1</IsNeedReturn>
            <ParaNum>0</ParaNum>
        </Function>
    </FunctionList>
</Switch>
<Switch>
    <ObjectID>1503</ObjectID>
    <ObjectName>挂机</ObjectName>

```

```
<FatherGroup>30</FatherGroup>
<FunctionList>
  <Function>
    <FunctionID>01</FunctionID>
    <FunctionName>Check</FunctionName>
    <IsCommand>1</IsCommand>
    <IsNeedReturn>1</IsNeedReturn>
    <ParaNum>0</ParaNum>
  </Function>
  <Function>
    <FunctionID>02</FunctionID>
    <FunctionName>Report</FunctionName>
    <IsCommand>1</IsCommand>
    <IsNeedReturn>1</IsNeedReturn>
    <ParaNum>0</ParaNum>
  </Function>
  <Function>
    <FunctionID>03</FunctionID>
    <FunctionName>ChangeStatus</FunctionName>
    <IsCommand>1</IsCommand>
    <IsNeedReturn>1</IsNeedReturn>
    <ParaNum>0</ParaNum>
  </Function>
</FunctionList>
</Switch>
<Switch>
  <ObjectID>1504</ObjectID>
  <ObjectName>抓拍</ObjectName>
  <FatherGroup>30</FatherGroup>
  <FunctionList>
    <Function>
      <FunctionID>01</FunctionID>
      <FunctionName>Check</FunctionName>
      <IsCommand>1</IsCommand>
      <IsNeedReturn>1</IsNeedReturn>
      <ParaNum>0</ParaNum>
    </Function>
    <Function>
      <FunctionID>02</FunctionID>
      <FunctionName>Report</FunctionName>
      <IsCommand>1</IsCommand>
      <IsNeedReturn>1</IsNeedReturn>
      <ParaNum>0</ParaNum>
    </Function>
  </FunctionList>
</Switch>
```

```

    </Function>
    <Function>
      <FunctionID>03</FunctionID>
      <FunctionName>ChangeStatus</FunctionName>
      <IsCommand>1</IsCommand>
      <IsNeedReturn>1</IsNeedReturn>
      <ParaNum>0</ParaNum>
    </Function>
  </FunctionList>
</Switch>
<Switch>
  <ObjectID>1505</ObjectID>
  <ObjectName>录音</ObjectName>
  <FatherGroup>30</FatherGroup>
  <FunctionList>
    <Function>
      <FunctionID>01</FunctionID>
      <FunctionName>Check</FunctionName>
      <IsCommand>1</IsCommand>
      <IsNeedReturn>1</IsNeedReturn>
      <ParaNum>0</ParaNum>
    </Function>
    <Function>
      <FunctionID>02</FunctionID>
      <FunctionName>Report</FunctionName>
      <IsCommand>1</IsCommand>
      <IsNeedReturn>1</IsNeedReturn>
      <ParaNum>0</ParaNum>
    </Function>
    <Function>
      <FunctionID>03</FunctionID>
      <FunctionName>ChangeStatus</FunctionName>
      <IsCommand>1</IsCommand>
      <IsNeedReturn>1</IsNeedReturn>
      <ParaNum>0</ParaNum>
    </Function>
  </FunctionList>
</Switch>
<Switch>
  <ObjectID>1506</ObjectID>
  <ObjectName>开锁</ObjectName>
  <FatherGroup>30</FatherGroup>
  <FunctionList>

```

```

    <Function>
      <FunctionID>01</FunctionID>
      <FunctionName>Check</FunctionName>
      <IsCommand>1</IsCommand>
      <IsNeedReturn>1</IsNeedReturn>
      <ParaNum>0</ParaNum>
    </Function>
  <Function>
    <FunctionID>02</FunctionID>
    <FunctionName>Report</FunctionName>
    <IsCommand>1</IsCommand>
    <IsNeedReturn>1</IsNeedReturn>
    <ParaNum>0</ParaNum>
  </Function>
  <Function>
    <FunctionID>03</FunctionID>
    <FunctionName>ChangeStatus</FunctionName>
    <IsCommand>1</IsCommand>
    <IsNeedReturn>1</IsNeedReturn>
    <ParaNum>0</ParaNum>
  </Function>
</FunctionList>
</Switch>
<EnumList ClassID = "5">
  <Enum>
    <ObjectID>1507</ObjectID>
    <ObjectName>门锁状态</ObjectName>
    <EnumValue>00</EnumValue>
    <EnumName>未开锁</EnumName>
    <EnumValue>01</EnumValue>
    <EnumName>已开锁</EnumName>
  <FatherGroup>30</FatherGroup>
  <EnumNum>5</EnumNum>
  <FunctionList>
    <Function>
      <FunctionID>01</FunctionID>
      <FunctionName>Check</FunctionName>
      <IsCommand>1</IsCommand>
      <IsNeedReturn>1</IsNeedReturn>
      <ParaNum>0</ParaNum>
    </Function>
    <Function>
      <FunctionID>02</FunctionID>

```

```

        <FunctionName>Report</FunctionName>
        <IsCommand>1</IsCommand>
        <IsNeedReturn>1</IsNeedReturn>
        <ParaNum>0</ParaNum>
    </Function>
    <Function>
        <FunctionID>03</FunctionID>
        <FunctionName>JumpTo</FunctionName>
        <IsCommand>1</IsCommand>
        <IsNeedReturn>1</IsNeedReturn>
        <ParaNum>1</ParaNum>
    </Function>
</FunctionList>
</Enum>
<Enum>
    <ObjectID>1508</ObjectID>
    <ObjectName>门铃状态</ObjectName>
    <EnumValue>00</EnumValue>
    <EnumName>空闲状态</EnumName>
    <EnumValue>01</EnumValue>
    <EnumName>振铃状态</EnumName>
    <EnumValue>02</EnumValue>
    <EnumName>通话状态</EnumName>
    <FatherGroup>30</FatherGroup>
    <EnumNum>4</EnumNum>
    <FunctionList>
        <Function>
            <FunctionID>01</FunctionID>
            <FunctionName>Check</FunctionName>
            <IsCommand>1</IsCommand>
            <IsNeedReturn>1</IsNeedReturn>
            <ParaNum>0</ParaNum>
        </Function>
        <Function>
            <FunctionID>02</FunctionID>
            <FunctionName>Report</FunctionName>
            <IsCommand>1</IsCommand>
            <IsNeedReturn>1</IsNeedReturn>
            <ParaNum>0</ParaNum>
        </Function>
        <Function>
            <FunctionID>03</FunctionID>
            <FunctionName>JumpTo</FunctionName>

```

```
        <IsCommand>1</IsCommand>
        <IsNeedReturn>1</IsNeedReturn>
        <ParaNum>1</ParaNum>
    </Function>
</FunctionList>
</Enum>
</EnumList>
</ObjectList>
```
