**管理端与X00服务间的请求规约**

目录

[1 一般概念 4](#_Toc475372435)

[2 请求分类 4](#_Toc475372436)

[3 新增数据结构 5](#_Toc475372437)

[3.1 写法说明示例 5](#_Toc475372438)

[3.2 获得识别记录的条件GetRecogRecordCondT 5](#_Toc475372439)

[3.3 识别记录SrvRecogRecordT 6](#_Toc475372440)

[3.4 获得黑名单识别记录的条件GetRecogRecordBlacklistCondT 7](#_Toc475372441)

[3.5 带属性的识别记录SrvRecogRecordPropertyT 9](#_Toc475372442)

[3.6 获得人员的条件GetPersonCondT 10](#_Toc475372443)

[3.7 人员信息SrvPersonBaseT 12](#_Toc475372444)

[3.8 获得当前连接设备条件SrvConnectDevCondT 13](#_Toc475372445)

[3.9 设备信息SrvConnectDevT 14](#_Toc475372446)

[3.10 人员与设备的绑定关系DevPersonBindT 14](#_Toc475372447)

[3.11 符合条件的人员与设备绑定关系DevCondPersonBindT 15](#_Toc475372448)

[3.12 具有某种相同属性的对象ObjectT 17](#_Toc475372449)

[3.13 获取扩展识别记录GetSrvRecogRecordExT 18](#_Toc475372450)

[3.14 扩展识别记录SrvRecogRecordExT 19](#_Toc475372451)

[3.15 人员注册图像EnrolledImageT 20](#_Toc475372452)

[3.16 获得人员的黑名单条件GetPersonBlacklistCondT 22](#_Toc475372453)

[3.17 人员属性设置PersonPropertyT 24](#_Toc475372454)

[3.18 人员及属性信息SrvPersonPropertyT 25](#_Toc475372455)

[3.19 属性结构PropertyT 26](#_Toc475372456)

[3.20 身份证人员信息IdCardPersonInfoT 27](#_Toc475372457)

[3.21 身份证人员信息回应IdCardRespT 28](#_Toc475372458)

[4 请求列表 29](#_Toc475372459)

[5 请求及回应 31](#_Toc475372460)

[5.1 一般回应格式 31](#_Toc475372461)

[5.2 获得人员列表 32](#_Toc475372462)

[5.2.1 请求 32](#_Toc475372463)

[5.2.2 回应 32](#_Toc475372464)

[5.3 获得服务分页识别记录 34](#_Toc475372465)

[5.3.1 请求 34](#_Toc475372466)

[5.3.2 回应 34](#_Toc475372467)

[5.4 获得服务分页带黑名单条件识别记录 35](#_Toc475372468)

[5.4.1 请求 35](#_Toc475372469)

[5.4.2 回应 35](#_Toc475372470)

[5.5 获取当前与服务连接的设备请求 37](#_Toc475372471)

[5.5.1 请求 37](#_Toc475372472)

[5.5.2 回应 37](#_Toc475372473)

[5.6 设置人员与设备间绑定关系 38](#_Toc475372474)

[5.6.1 请求 38](#_Toc475372475)

[5.6.2 回应 39](#_Toc475372476)

[5.7 获取人员与设备间的绑定关系 39](#_Toc475372477)

[5.7.1 请求 39](#_Toc475372478)

[5.7.2 回应 39](#_Toc475372479)

[5.8 设置符合条件的人员与设备间绑定关系 40](#_Toc475372480)

[5.8.1 请求 40](#_Toc475372481)

[5.8.2 回应 40](#_Toc475372482)

[5.9 获取某种属性的对象列表 41](#_Toc475372483)

[5.9.1 请求 41](#_Toc475372484)

[5.9.2 回应 41](#_Toc475372485)

[5.10 获取扩展识别记录 42](#_Toc475372486)

[5.10.1 请求 42](#_Toc475372487)

[5.10.2 回应 42](#_Toc475372488)

[5.11 获取人员注册图像 43](#_Toc475372489)

[5.11.1 请求 43](#_Toc475372490)

[5.11.2 回应 44](#_Toc475372491)

[5.12 获得黑名单条件人员列表 44](#_Toc475372492)

[5.12.1 请求 44](#_Toc475372493)

[5.12.2 回应 45](#_Toc475372494)

[5.13 设置人员属性 46](#_Toc475372495)

[5.13.1 请求 46](#_Toc475372496)

[5.13.2 回应 46](#_Toc475372497)

[5.14 设置属性（暂不启用） 47](#_Toc475372498)

[5.14.1 请求 47](#_Toc475372499)

[5.14.2 回应 47](#_Toc475372500)

[5.15 获取属性列表（暂不启用） 48](#_Toc475372501)

[5.15.1 请求 48](#_Toc475372502)

[5.15.2 回应 48](#_Toc475372503)

[5.16 新增人员身份证信息 49](#_Toc475372504)

[5.16.1 请求 49](#_Toc475372505)

[5.16.2 回应 49](#_Toc475372506)

[5.17 修改人员身份证信息 50](#_Toc475372507)

[5.17.1 请求 50](#_Toc475372508)

[5.17.2 回应 50](#_Toc475372509)

[5.18 获取人员身份证详细信息 51](#_Toc475372510)

[5.18.1 请求 51](#_Toc475372511)

[5.18.2 回应 51](#_Toc475372512)

# 一般概念

1. 管理端与服务端之间的请求和设备与服务端之间的请求分开。
2. 管理端与服务端连接的端口缺省是9998；设备与服务端连接端口缺省是9999。
3. 一般来说，都是由管理端在需要的时候连接服务端，并主动发送请求到服务端。
4. 管理端与服务端之间不需要保持长连接，不需要与服务端之间发送保活请求。
5. 所有请求和回应的数据的最大长度不应超过1MB。

# 请求分类

人员相关：

~~新增、修改、删除（一期不做）~~

获取现有人员；

~~获取指定设备上现有人员；~~

设备相关：

获取与服务连接的设备（包括设备详情）；

~~（一期不做）~~

~~控制设备重启~~

~~控制设备关机~~

~~修改设备IP~~

识别记录相关：

获取服务上符合要求的识别记录

绑定相关：

设置人员与设备绑定

设置人员与设备解绑

属性相关：

获取符合某种属性的列表（指部门等现有属性）

获取人员注册图像

特殊要求：

人员黑名单属性设置及取消，及相关人员和识别记录的查询

# 新增数据结构

## 写法说明示例

int32 占4个字节的整数

char[64] 固定长度为64的字符数组

UUID 占16个字节

BYTE[变长] 长度未定的unsigned char数组，具体长度由其他字段给出

"0" 字符0

0 数字0

null 数据库中字段的null

## 获得识别记录的条件GetRecogRecordCondT

结构总长度固定，为264字节。

struct GetRecogRecordCondT

{

int32 pageIndex; // 页面序号

int32 countInPage; // 一次获取的记录个数

int32 success; // 识别是否成功

char[64] deviceSn; // 设备序列号

char[64] personName; // 人员姓名

char[64] workSn; // 人员工号

char[20] idCard; // 身份证号

char[20] startTime; // 起始时间

char[20] endTime; // 结束时间

};

字段详细说明（长度单位：字节）

|  |  |  |
| --- | --- | --- |
| 内容 | 长度 | 说明 |
| pageIndex | 4 | 页面序号，从0开始，如果发来的index<0，自动设置为0 |
| countInPage | 4 | 一次获取的记录个数，最多支持1024条记录；如果这里的参数值超过1024，则返回的记录条数最多仍然是1024 |
| success | 4 | 是否识别成功，查询条件之一。1：成功识别的记录；0：未成功识别的记录（目前不支持）；-1：全部 |
| deviceSn | 64 | 设备序列号，查询条件之一，支持模糊查询。如果为空，则忽略此查询条件 |
| personName | 64 | 人员姓名，查询条件之一，支持模糊查询。如果为空，则忽略此查询条件 |
| workSn | 64 | 人员工号，查询条件之一，支持模糊查询。如果为空，则忽略此查询条件 |
| idCard | 20 | 身份证号，查询条件之一，支持模糊查询。如果为空，则忽略此查询条件 |
| startTime | 20 | 起始时间，查询条件之一。如果为空，则忽略此查询条件；如不为空，则条件为人员创建时间>=startTime |
| endTime | 20 | 结束时间，查询条件之一。如果为空，则忽略此查询条件；如不为空，则条件为人员创建时间<startTime |

相关请求[REQ\_GET\_PAGE\_RECOG\_RECORD](#_请求)。

## 识别记录SrvRecogRecordT

结构总长度固定，为252字节。

struct SrvRecogRecordT

{

char[64] deviceSn; // 设备序列号

char[20] idenTime; // 识别时间

char[64] personName; // 人员姓名

char[64] workSn; // 人员工号

char[20] idCard; // 身份证号

int32 eyeFlag; // 眼睛

UUID personId; // 人员ID

};

字段详细说明（长度单位：字节）

无内容的字段置为全0。

|  |  |  |
| --- | --- | --- |
| 内容 | 长度 | 说明 |
| deviceSn | 64 | 设备序列号，标识是在哪台设备上识别的 |
| idenTime | 20 | 识别时间，样例：2016-10-01 18:35:30 |
| personName | 64 | 人员姓名 |
| workSn | 64 | 人员工号 |
| idCard | 20 | 身份证号 |
| eyeFlag | 4 | 眼睛，哪只眼睛识别通过的，目前无效，置为0 |
| personId | 16 | 人员id |

相关请求[ACK\_GET\_PAGE\_RECOG\_RECORD](#_回应)。

## 获得黑名单识别记录的条件GetRecogRecordBlacklistCondT

结构总长度固定，为268字节。

struct GetRecogRecordBlacklistCondT

{

int32 pageIndex; // 页面序号

int32 countInPage; // 一次获取的记录个数

int32 success; // 识别是否成功

char[64] deviceSn; // 设备序列号

char[64] personName; // 人员姓名

char[64] workSn; // 人员工号

char[20] idCard; // 身份证号

char[20] startTime; // 起始时间

char[20] endTime; // 结束时间

int32 blacklist; // 黑名单

};

字段详细说明（长度单位：字节）

|  |  |  |
| --- | --- | --- |
| 内容 | 长度 | 说明 |
| pageIndex | 4 | 页面序号，从0开始，如果发来的index<0，自动设置为0 |
| countInPage | 4 | 一次获取的记录个数，最多支持1024条记录；如果这里的参数值超过1024，则返回的记录条数最多仍然是1024 |
| success | 4 | 是否识别成功，查询条件之一。1：成功识别的记录；0：未成功识别的记录（目前不支持）；-1：全部 |
| deviceSn | 64 | 设备序列号，查询条件之一，支持模糊查询。如果为空，则忽略此查询条件 |
| personName | 64 | 人员姓名，查询条件之一，支持模糊查询。如果为空，则忽略此查询条件 |
| workSn | 64 | 人员工号，查询条件之一，支持模糊查询。如果为空，则忽略此查询条件 |
| idCard | 20 | 身份证号，查询条件之一，支持模糊查询。如果为空，则忽略此查询条件 |
| startTime | 20 | 起始时间，查询条件之一。如果为空，则忽略此查询条件；如不为空，则条件为人员创建时间>=startTime |
| endTime | 20 | 结束时间，查询条件之一。如果为空，则忽略此查询条件；如不为空，则条件为人员创建时间<startTime |
| blacklist | 4 | 表示是否查询属于黑名单的识别记录  为0，表示查询全部符合条件的识别记录（不论是否属于黑名单的人员）  为1，表示只查询现阶段为黑名单上人员的识别记录 |

相关请求[REQ\_GET\_SRV\_RECOG\_RECORD\_BLACKLIST](#_请求_9)。

## 带属性的识别记录SrvRecogRecordPropertyT

结构总长度固定，为256字节。

struct SrvRecogRecordPropertyT

{

char[64] deviceSn; // 设备序列号

char[20] idenTime; // 识别时间

char[64] personName; // 人员姓名

char[64] workSn; // 人员工号

char[20] idCard; // 身份证号

int32 eyeFlag; // 眼睛

UUID personId; // 人员ID

int32 property; // 人员属性

};

字段详细说明（长度单位：字节）

无内容的字段置为全0。

|  |  |  |
| --- | --- | --- |
| 内容 | 长度 | 说明 |
| deviceSn | 64 | 设备序列号，标识是在哪台设备上识别的 |
| idenTime | 20 | 识别时间，样例：2016-10-01 18:35:30 |
| personName | 64 | 人员姓名 |
| workSn | 64 | 人员工号 |
| idCard | 20 | 身份证号 |
| eyeFlag | 4 | 眼睛，哪只眼睛识别通过的，目前无效，置为0 |
| personId | 16 | 人员id |
| property | 4 | 人员属性 |

相关请求[ACK\_GET\_PAGE\_RECOG\_RECORD\_BLACKLIST](#_回应_6)。

## 获得人员的条件GetPersonCondT

结构总长度固定，为332字节。

（20161115增加部门名称）

struct GetPersonCondT

{

int32 pageIndex; // 页面序号

int32 countInPage; // 一页记录个数

int32 irisStatus; // 注册状态

int32 bindingStatus; // 绑定状态

char[64] deviceSn; // 设备序列号

char[64] personName; // 人员姓名

char[64] workSn; // 人员工号

char[20] idCard; // 身份证号

char[64] departName; // 部门名称

char[20] startTime; // 起始时间

char[20] endTime; // 结束时间

};

字段详细说明（长度单位：字节）

|  |  |  |
| --- | --- | --- |
| 内容 | 长度 | 说明 |
| pageIndex | 4 | 页面序号，从0开始，如果发来的index<0，自动设置为0 |
| countInPage | 4 | 一次获取的人员记录个数，最多支持1024条记录；如果这里的参数值超过1024，则返回的记录条数最多仍然是1024 |
| irisStatus | 4 | 是否已注册虹膜，查询条件之一。1：已注册；0：未注册；-1：全部  如输入其他数字，自动设置为-1 |
| bindingStatus | 4 | 是否已绑定设备，查询条件之一。1：已绑定；0：未绑定；-1：全部  如输入其他数字，自动设置为-1  如输入-1，deviceSn的查询条件自动忽略；  如输入0，且deviceSn不为空，则表示查询未和该设备绑定的人员；  如输入0，且deviceSn为空，则表示查询未和任何设备绑定的人员；  如输入1，且deviceSn不为空，则表示查询已和该设备绑定的人员；  如输入1，且deviceSn为空，则表示查询已和任何设备绑定的人员； |
| deviceSn | 64 | 设备序列号，查询条件之一，支持模糊查询。如果为空，则忽略此查询条件 |
| personName | 64 | 人员姓名，查询条件之一，支持模糊查询。如果为空，则忽略此查询条件 |
| workSn | 64 | 人员工号，查询条件之一，支持模糊查询。如果为空，则忽略此查询条件 |
| idCard | 20 | 身份证号，查询条件之一，支持模糊查询。如果为空，则忽略此查询条件 |
| departName | 64 | 部门名称，查询条件之一，支持模糊查询。如果为空，则忽略此查询条件；  如果部门名称为64个空格，则表示查询在数据库中，部门名称为NULL（未设置）的条件 |
| startTime | 20 | 起始时间，查询条件之一。如果为空，则忽略此查询条件；如不为空，则条件为人员创建时间>=startTime |
| endTime | 20 | 结束时间，查询条件之一。如果为空，则忽略此查询条件；如不为空，则条件为人员创建时间<endTime |

相关请求[REQ\_GET\_SRV\_PERSON\_LIST](#_请求_1)。

## 人员信息SrvPersonBaseT

结构总长度固定，为404字节。

struct SrvPersonBaseT

{

UUID personId; // 人员ID

int32 hasFeature; // 是否已注册

int32 hasBinding; // 是否已绑定

char[64] personName; // 人员姓名  
 char[64] workSn; // 人员工号

int32 sex; // 性别

char[20] idCard; // 身份证号

char[64] departName; // 部门名称

char[16] cardNumber; // 卡号

char[128] memo; // 备注

char[20] createTime; // 创建时间

};

字段详细说明（长度单位：字节）

|  |  |  |
| --- | --- | --- |
| 内容 | 长度 | 说明 |
| personId | 16 | 人员ID，UUID |
| hasFeature | 4 | 是否已注册，1——已注册，0——未注册 |
| hasBinding | 4 | 是否已绑定，1——已绑定，0——未绑定 |
| personName | 64 | 人员姓名 |
| workSn | 64 | 人员工号 |
| sex | 4 | 性别，1——男性，2——女性 |
| idCard | 20 | 身份证号 |
| departName | 64 | 部门名称 |
| cardNumber | 16 | 卡号 |
| memo | 128 | 备注 |
| createTime | 20 | 人员创建时间（写入服务数据库的时间） |

相关请求[ACK\_GET\_SRV\_PERSON\_LIST](#_响应获得服务现有人员列表)。

## 获得当前连接设备条件SrvConnectDevCondT

结构总长度固定，为8字节。

struct SrvConnectDevCondT

{

int32 pageIndex; // 页面序号

int32 countInPage; // 一页记录个数

}

字段详细说明（长度单位：字节）

|  |  |  |
| --- | --- | --- |
| 内容 | 长度 | 说明 |
| pageIndex | 4 | 页面序号，从0开始，如果发来的index<0，自动默认为0 |
| countInPage | 4 | 一次获取的设备个数，最多支持1024条记录；如果这里的参数值超过1024，则返回的记录条数最多仍然是1024 |

相关请求[REQ\_GET\_CONNECT\_DEVICE](#_请求_3)。

## 设备信息SrvConnectDevT

结构总长度固定，为96字节。

struct SrvConnectDevT

{

char[16] ip; // 设备IP

char[16] gateway; // 设备网关

char[64] deviceSn; // 设备序列号

};

字段详细说明（长度单位：字节）

|  |  |  |
| --- | --- | --- |
| 内容 | 长度 | 说明 |
| ip | 16 | 设备IP，仅为辅助标识，不一定唯一 |
| geteway | 16 | 设备网关，目前无此功能，全置为0 |
| deviceSn | 64 | 设备序列号，作为设备的唯一标识 |

相关请求[ACK\_GET\_CONNECT\_DEVICE](#_回应获取当前与服务连接的设备)。

## 人员与设备的绑定关系DevPersonBindT

结构总长度固定，为84字节。

struct DevPersonBindT

{

int32 binding; // 绑定(1)或解绑(0)

char[64] deviceSn; // 设备序列号

UUID personId; // 人员ID

};

字段详细说明（长度单位：字节）

|  |  |  |
| --- | --- | --- |
| 内容 | 长度 | 说明 |
| binding | 4 | 1——设置人员与设备绑定  0——设置人员与设备解绑 |
| deviceSn | 64 | 绑定或解绑的设备的设备序列号  当binding=0（解绑），且deviceSn=""时，表示将人员与当前绑定设备全部解绑。目前无全部绑定功能(added at 20161108) |
| personId | 16 | 人员ID，UUID |

相关请求[REQ\_BIND\_PERSON\_DEVICE](#_请求_2)。

## 符合条件的人员与设备绑定关系DevCondPersonBindT

符合条件的人员与指定设备绑定或解绑。

结构总长度固定，为392字节。

struct DevCondPersonBindT

{

int32 binding; // 绑定(1)或解绑(0)

char[64] bindingDeviceSn; // 此次进行绑定操作的设备序列号

// 后面都为人员的查询条件

int32 irisStatus; // 注册状态

int32 bindingStatus; // 绑定状态

char[64] deviceSn; // 设备序列号

char[64] personName; // 人员姓名

char[64] workSn; // 人员工号

char[20] idCard; // 身份证号

char[64] departName; // 部门名称

char[20] startTime; // 起始时间

char[20] endTime; // 结束时间

};

字段详细说明（长度单位：字节）

|  |  |  |
| --- | --- | --- |
| 内容 | 长度 | 说明 |
| binding | 4 | 1——设置人员与设备绑定  0——设置人员与设备解绑 |
| bindingDeviceSn | 64 | 此次进行绑定或解绑操作的设备序列号  注意与后面的查询条件中的deviceSn要区分开，两者意义不一样 |
| irisStatus | 4 | 是否已注册虹膜，查询条件之一。1：已注册；0：未注册；-1：全部  如输入其他数字，自动设置为-1 |
| bindingStatus | 4 | 是否已绑定设备，查询条件之一。1：已绑定；0：未绑定；-1：全部  如输入其他数字，自动设置为-1  如输入-1，deviceSn的查询条件自动忽略；  如输入0，且deviceSn不为空，则表示查询未和该设备绑定的人员；  如输入0，且deviceSn为空，则表示查询未和任何设备绑定的人员；  如输入1，且deviceSn不为空，则表示查询已和该设备绑定的人员；  如输入1，且deviceSn为空，则表示查询已和任何设备绑定的人员； |
| deviceSn | 64 | 设备序列号，查询条件之一，支持模糊查询。如果为空，则忽略此查询条件 |
| personName | 64 | 人员姓名，查询条件之一，支持模糊查询。如果为空，则忽略此查询条件 |
| workSn | 64 | 人员工号，查询条件之一，支持模糊查询。如果为空，则忽略此查询条件 |
| idCard | 20 | 身份证号，查询条件之一，支持模糊查询。如果为空，则忽略此查询条件 |
| departName | 64 | 部门名称，查询条件之一，支持模糊查询。如果为空，则忽略此查询条件  如果部门名称为64个空格，则表示查询在数据库中，部门名称为NULL（未设置）的条件 |
| startTime | 20 | 起始时间，查询条件之一。如果为空，则忽略此查询条件；如不为空，则条件为人员创建时间>=startTime |
| endTime | 20 | 结束时间，查询条件之一。如果为空，则忽略此查询条件；如不为空，则条件为人员创建时间<endTime |

相关请求[REQ\_BIND\_COND\_PERSON\_DEVICE](#_请求_4)。

## 具有某种相同属性的对象ObjectT

结构总长度固定，为68字节。

struct ObjectT

{

int32 objectId; // 对象id

char[64] objectName; // 对象名称

};

字段详细说明（长度单位：字节）

|  |  |  |
| --- | --- | --- |
| 内容 | 长度 | 说明 |
| objectId | 4 | 对象id，对某些对象来说是无意义的，可设置为0 |
| objectName | 64 | 对象名称 |

相关请求[ACK\_GET\_ATTRIBUTE\_OBJECT\_LIST](#_回应_1)。

使用示例，获取系统中的部门名称，回应中包括一系列的部门id+部门名称

## 获取扩展识别记录GetSrvRecogRecordExT

结构总长度固定，为80个字节。

获取扩展的识别记录请求结构。

（20161205 增加查询属性的相关字段，属性查询条件中可设置黑名单等查询条件）

struct GetSrvRecogRecordExT

{

int32 lastRecogIndex;

int32 countInPage;

int32 type;

int32 devSnValid;

char[64] deviceSn;

};

字段详细说明（长度单位：字节）

|  |  |  |
| --- | --- | --- |
| 内容 | 长度 | 说明 |
| lastRecogIndex | 4 | 最后一条识别记录。新获取的识别记录以这个index为基础。 |
| countInPage | 4 | 一次获取的识别记录条数，最多不超过50条记录 |
| type | 4 | 获取识别记录的类型  bit0表示是否需要增加一条随机记录  0：不需要增加一条随机记录  1：需要增加一条随机记录  bit1表示是否获取只属于黑名单的记录  0：不论是否属于黑名单的记录都可以  1：只获取属于当前黑名单中人员的识别记录  例如：  type=0表示不论是否属于黑名单的记录都可以，且不需要增加一条随机记录；  type=1表示不论是否属于黑名单的记录都可以，且需要增加一条随机记录；  type=2表示只获取属于当前黑名单中人员的识别记录，且不需要增加一条随机记录；  type=3表示只获取属于当前黑名单中人员的识别记录，且需要增加一条随机记录（也属于黑名单中人员的识别记录）； |
| devSnValid | 4 | deviceSn查找条件是否有效。1：有效；0：无效 |
| deviceSn | 64 | 设备序列号。如果devSnValid=1，则表示需要发送在deviceSn确定的设备上识别出来的识别记录 |

相关请求[REQ\_GET\_SRV\_RECOG\_RECORD\_EX](#_请求_5)。

## 扩展识别记录SrvRecogRecordExT

结构总长度固定，为396个字节。

用于公安项目显示识别记录，要求人员信息尽量齐全。

struct SrvRecogRecordExT

{

int32 logId; // 识别序列号

char[64] deviceSn; // 设备序列号

char[20] idenTime; // 识别时间

char[16] deviceIp; // 设备IP

char[36] personIdStr; // 人员ID String

int32 eyeFlag; // 眼睛

char[64] personName; // 人员姓名  
 char[64] workSn; // 人员工号

int32 sex; // 性别

char[20] idCard; // 身份证号

char[64] departName; // 部门名称

char[16] cardNumber; // 卡号

char[20] createTime; // 人员创建时间

};

字段详细说明（长度单位：字节）

|  |  |  |
| --- | --- | --- |
| 内容 | 长度 | 说明 |
| logId | 4 | 识别记录在数据库表中的index |
| deviceSn | 64 | 识别记录产生的设备序列号 |
| deviceIp | 16 | 识别记录产生的设备IP。  注意：由于识别记录表中没有IP，所以此IP是服务记录的设备序列号曾经对应的IP，不一定是现在的设备IP |
| idenTime | 20 | 识别时间 |
| personIdStr | 36 | 人员UUID的string |
| eyeFlag | 4 | 识别通过的是左眼还是右眼，目前无效，设置为0 |
| personName | 64 | 人员姓名 |
| workSn | 64 | 人员工号 |
| sex | 4 | 性别，1——男性，2——女性 |
| idCard | 20 | 身份证号 |
| departName | 64 | 部门名称 |
| cardNumber | 16 | 卡号 |
| createTime | 20 | 人员创建时间（写入服务数据库的时间） |

相关请求[ACK\_GET\_SRV\_RECOG\_RECORD\_EX](#_回应_2)。

## 人员注册图像EnrolledImageT

包括人员注册时的最佳左眼图像（如果有的话），最佳右眼图像（如果有的话）和照片图像（如果有的话）

struct EnrolledImageT

{

char[36] personIdStr;

int32 leftEyeSize;

int32 leftEyeType;

BYTE[变长] leftEyeImage;

int32 rightEyeType;

int32 rightEyeSize;

BYTE[变长] rightEyeImage;

int32 photoSize;

int32 photoType;

BYTE[ 变长] photoImage;

};

字段详细说明（长度单位：字节）

|  |  |  |
| --- | --- | --- |
| 内容 | 长度 | 说明 |
| personIdStr | 36 | 人员UUID的string |
| leftEyeSize | 4 | 左眼最佳注册图像的大小； 0：表示无左眼图像；非0：左眼图像大小 |
| leftEyeType | 4 | 左眼图像类型；0：无图；1：bmp；2：jpg；无论是否有左眼图像，总是占位 |
| leftEyeImage | 变长 | 实际左眼注册图像，如果有，则长度为leftEyeSize的值；如没有左眼注册图像，则不占位 |
| rightEyeSize | 4 | 右眼最佳注册图像的大小； 0：表示无右眼图像；非0：右眼图像大小 |
| rightEyeType | 4 | 右眼图像类型；0：无图；1：bmp；2：jpg；无论是否有右眼图像，总是占位 |
| rightEyeImage | 变长 | 实际右眼注册图像，如果有，则长度为rightEyeSize的值；如没有右眼注册图像，则不占位 |
| photoSize | 4 | 照片图像的大小； 0：表示无照片图像；非0：照片图像大小 |
| photoType | 4 | 照片图像类型；0：无图；1：bmp；2：jpg；无论是否有照片图像，总是占位 |
| photoImage | 变长 | 实际照片图像，如果有，则长度为photoSize的值；如没有照片图像，则不占位 |

对应请求[ACK\_GET\_PERSON\_IMAGES](#_回应_3)。

## 获得人员的黑名单条件GetPersonBlacklistCondT

结构总长度固定，为336字节。

（20161201为新疆需求增加查询属于黑名单人员的功能，在原来查询人员条件的基础上，增加是否为黑名单的查询条件）

struct GetPersonBlacklistCondT

{

int32 pageIndex; // 页面序号

int32 countInPage; // 一页记录个数

int32 irisStatus; // 注册状态

int32 bindingStatus; // 绑定状态

char[64] deviceSn; // 设备序列号

char[64] personName; // 人员姓名

char[64] workSn; // 人员工号

char[20] idCard; // 身份证号

char[64] departName; // 部门名称

char[20] startTime; // 起始时间

char[20] endTime; // 结束时间

int32 blacklist; // 是否属于黑名单的人员

};

字段详细说明（长度单位：字节）

|  |  |  |
| --- | --- | --- |
| 内容 | 长度 | 说明 |
| pageIndex | 4 | 页面序号，从0开始，如果发来的index<0，自动设置为0 |
| countInPage | 4 | 一次获取的人员记录个数，最多支持1024条记录；如果这里的参数值超过1024，则返回的记录条数最多仍然是1024 |
| irisStatus | 4 | 是否已注册虹膜，查询条件之一。1：已注册；0：未注册；-1：全部  如输入其他数字，自动设置为-1 |
| bindingStatus | 4 | 是否已绑定设备，查询条件之一。1：已绑定；0：未绑定；-1：全部  如输入其他数字，自动设置为-1  如输入-1，deviceSn的查询条件自动忽略；  如输入0，且deviceSn不为空，则表示查询未和该设备绑定的人员；  如输入0，且deviceSn为空，则表示查询未和任何设备绑定的人员；  如输入1，且deviceSn不为空，则表示查询已和该设备绑定的人员；  如输入1，且deviceSn为空，则表示查询已和任何设备绑定的人员； |
| deviceSn | 64 | 设备序列号，查询条件之一，支持模糊查询。如果为空，则忽略此查询条件 |
| personName | 64 | 人员姓名，查询条件之一，支持模糊查询。如果为空，则忽略此查询条件 |
| workSn | 64 | 人员工号，查询条件之一，支持模糊查询。如果为空，则忽略此查询条件 |
| idCard | 20 | 身份证号，查询条件之一，支持模糊查询。如果为空，则忽略此查询条件 |
| departName | 64 | 部门名称，查询条件之一，支持模糊查询。如果为空，则忽略此查询条件；  如果部门名称为64个空格，则表示查询在数据库中，部门名称为NULL（未设置）的条件 |
| startTime | 20 | 起始时间，查询条件之一。如果为空，则忽略此查询条件；如不为空，则条件为人员创建时间>=startTime |
| endTime | 20 | 结束时间，查询条件之一。如果为空，则忽略此查询条件；如不为空，则条件为人员创建时间<endTime |
| blacklist | 4 | 是否获取只属于黑名单的人员  0：忽略此查询条件  1：只在当前黑名单中的人员中查询 |

相关请求[REQ\_GET\_SRV\_PERSON\_BLACKLIST](#_请求_6)。

## 人员属性设置PersonPropertyT

结构总长度固定，为112字节。

struct PersonPropertyT

{

char[32] type; // 属性类型

char[64] property; // 属性设置

UUID personId; // 人员ID

};

字段详细说明（长度单位：字节）

|  |  |  |
| --- | --- | --- |
| 内容 | 长度 | 说明 |
| type | 32 | 属性类型，目前可接受类型包括——  "property"：设置人员属性 |
| property | 64 | 具体的属性设置  当属性类型为"property"时  字符串将转为int类型（4字节长度）的数字设置到对应的数据表中  对于黑名单的设置需求来说，  为"1"表示设置对应人员进入黑名单  为"0"表示从黑名单中删除该人 |
| personId | 16 | 人员ID，UUID |

相关请求[REQ\_SET\_PERSON\_PROPERTY](#_请求_7)。

## 人员及属性信息SrvPersonPropertyT

结构总长度固定，为408字节。

struct SrvPersonPropertyT

{

UUID personId; // 人员ID

int32 hasFeature; // 是否已注册

int32 hasBinding; // 是否已绑定

char[64] personName; // 人员姓名  
 char[64] workSn; // 人员工号

int32 sex; // 性别

char[20] idCard; // 身份证号

char[64] departName; // 部门名称

char[16] cardNumber; // 卡号

char[128] memo; // 备注

char[20] createTime; // 创建时间

int32 property; // 人员属性

};

字段详细说明（长度单位：字节）

|  |  |  |
| --- | --- | --- |
| 内容 | 长度 | 说明 |
| personId | 16 | 人员ID，UUID |
| hasFeature | 4 | 是否已注册，1——已注册，0——未注册 |
| hasBinding | 4 | 是否已绑定，1——已绑定，0——未绑定 |
| personName | 64 | 人员姓名 |
| workSn | 64 | 人员工号 |
| sex | 4 | 性别，1——男性，2——女性 |
| idCard | 20 | 身份证号 |
| departName | 64 | 部门名称 |
| cardNumber | 16 | 卡号 |
| memo | 128 | 备注 |
| createTime | 20 | 人员创建时间（写入服务数据库的时间） |
| property | 4 | 人员属性  如无属性，则为0；  有属性，则为实际属性值 |

相关请求[ACK\_GET\_SRV\_PERSON\_BLACKLIST](#_回应_4)。

## 属性结构PropertyT

结构总长度固定，共68字节。

struct PropertyT

{

int32 property;

char[64] name;

}

字段详细说明（长度单位：字节）

|  |  |  |
| --- | --- | --- |
| 内容 | 长度 | 说明 |
| property | 4 | 属性数值 |
| name | 64 | 属性对应的名称 |

例如，黑名单是一种属性。

相关请求[REQ\_ SET\_ PROPERTY](#_请求_8)及[ACK\_GET\_PROPERTY\_LIST](#_回应_5)。

## 身份证人员信息IdCardPersonInfoT

结构总长度不固定，不算照片数据(photo)，长度为1744。

struct IdCardPersonInfoT

{

char[256] name; // 姓名

int32 sex; // 性别

char[32] ethnicity; // 民族

char[20] dateOfBirth; // 出生日期

char[1024] residentialAddress; // 住址

char[20] idCard; // 身份证号

char[256] authority; // 签发机关

char[20] beginOfValidity; // 有效期限开始日期

char[20] endOfValidity; // 有效期限结束日期

int32 errorNo;

char[64] errorString; // 错误描述

char[20] photoType; // 照片图片类型

int32 photoLength; // 照片长度

char[变长] photo; // 照片数据

}

字段详细说明（长度单位：字节）

|  |  |  |
| --- | --- | --- |
| 内容 | 长度 | 说明 |
| name | 256 | 人员姓名 |
| sex | 4 | 性别 |
| ethnicity | 32 | 民族 |
| dateOfBirth | 20 | 出生日期 |
| residentialAddress | 1024 | 住址 |
| idCard | 20 | 身份证号 |
| authority | 256 | 签发机关 |
| beginOfValidity | 20 | 有效期限开始日期 |
| endOfValidity | 20 | 有效期限结束日期 |
| errorNo | 4 | 错误号（当为增加或更新人员身份证号时，设置为0即可；当为获取人员身份证号回应的数据时，接收方需判断errorNo是否为0，如果是0，表示对应的身份证详细信息有效，否则无效） |
| errorString | 64 | 错误描述（当为增加或更新人员身份证号时，设置为全空白即可；当为获取人员身份证号回应的数据时，是错误描述） |
| photoType | 20 | 照片类型，如'bmp'，'jpg'。照片最好为完整的图片数据格式，不要为裸格式 |
| photoLength | 4 | 照片长度，photo的长度 |
| photo | xx | 照片数据，长度为photoLength |

此数据结构对应从身份证上读出的人员信息。

相关请求[REQ\_ADD\_PERSON\_IDCARD](#_请求_10)、[REQ\_UPDATE\_PERSON\_IDCARD](#_请求_11)及[ACK\_GET\_PERSON\_IDCARD](#_回应_7)。

## 身份证人员信息回应IdCardRespT

结构总长度固定，共88字节。

struct IdCardRespT

{

char[20] idCard; // 身份证号;

int32 errorNo; // 错误号

char[64] errorString; // 错误描述

}

字段详细说明（长度单位：字节）

|  |  |  |
| --- | --- | --- |
| 内容 | 长度 | 说明 |
| idCard | 20 | 身份证号 |
| errorNo | 4 | 错误号 |
| errorString | 64 | 错误描述 |

错误号和错误描述对应关系如下：

|  |  |
| --- | --- |
| 错误号 | 错误描述 |
| 0 | 成功 |
| 1 | 没有此身份证号 |
| 2 | 此身份证号已存在 |

相关请求[REQ\_ SET\_ PROPERTY](#_请求_8)及[ACK\_GET\_PROPERTY\_LIST](#_回应_5)。

# 请求列表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 获得人员列表 | REQ\_GET\_SRV\_PERSON\_LIST | 0x0231 | 要求获得服务端人员列表 | 管理端->服务器端 |
| 响应获得人员列表 | ACK\_GET\_SRV\_PERSON\_ LIST | 0x4231 | 响应获得服务端人员列表 | 服务器端->管理端 |
| 获得黑名单条件人员列表 | REQ\_GET\_SRV\_PERSON\_BLACKLIST | 0x0232 | 要求获得服务端黑名单条件人员列表 | 管理端->服务器端 |
| 响应获得黑名单条件人员列表 | ACK\_GET\_SRV\_PERSON\_BLACKLIST | 0x4232 | 响应获得服务端黑名单条件人员列表 | 服务器端->管理端 |
|  |  |  |  |  |
| 获取识别记录 | REQ\_GET\_SRV\_RECOG\_RECORD | 0x0451 | 要求获得服务端保存的识别记录 | 管理端->服务器端 |
| 响应获取识别记录 | ACK\_GET\_SRV\_RECOG\_RECORD | 0x4451 | 响应获得服务端保存的识别记录请求 | 服务器端->管理端 |
| 按页获取识别记录 | REQ\_GET\_PAGE\_RECOG\_RECORD | 0x0452 | 要求按页获得服务端保存的识别记录 | 管理端->服务器端 |
| 响应按页获取识别记录 | ACK\_GET\_PAGE\_RECOG\_RECORD | 0x4452 | 响应按页获得服务端保存的识别记录请求 | 服务器端->管理端 |
| 获取扩展识别记录 | REQ\_GET\_SRV\_RECOG\_RECORD\_EX | 0x0453 | 要求获得服务端保存的扩展的识别记录 | 管理端->服务器端 |
| 响应获取扩展识别记录 | ACK\_GET\_SRV\_RECOG\_RECORD\_EX | 0x4453 | 响应获得服务端保存的扩展的识别记录请求 | 服务器端->管理端 |
| 按页获取黑名单条件识别记录 | REQ\_GET\_PAGE\_RECOG\_RECORD\_BLACKLIST | 0x0454 | 要求按页获得服务端保存的识别记录，带有黑名单条件 | 管理端->服务器端 |
| 响应按页获取黑名单条件识别记录 | ACK\_GET\_PAGE\_RECOG\_RECORD\_BLACKLIST | 0x4454 | 响应按页获得服务端保存的识别记录请求，带有黑名单条件 | 服务器端->管理端 |
|  |  |  |  |  |
| 获取当前与服务连接的设备 | REQ\_GET\_CONNECT\_DEVICE | 0x0b01 | 要求获取当前与服务端连接的设备列表 | 管理端->服务器端 |
| 响应获取当前与服务连接的设备 | ACK\_GET\_CONNECT\_DEVICE | 0x4b01 | 回应当前与服务端连接的设备列表 | 服务器端->管理端 |
|  |  |  |  |  |
| 设置人员与设备间的绑定关系 | REQ\_BIND\_PERSON\_DEVICE | 0x0b02 | 管理端向服务器端设置人员与设备的绑定关系 | 管理端->服务器端 |
| 回应设置人员与设备间的绑定关系 | ACK\_BIND\_PERSON\_DEVICE | 0x4b02 | 回应人员与设备的绑定关系 | 服务器端->管理端 |
| 获取某个人与设备的绑定关系 | REQ\_GET\_PERSON\_BIND\_INFO | 0x0b03 | 管理端向服务端要求获取某个指定人员与哪些设备绑定的信息 | 管理端->服务器端 |
| 响应获取某个人与设备的绑定关系 | ACK\_GET\_PERSON\_BIND\_INFO | 0x4b03 | 服务端向管理端回应某个人员与哪些设备绑定的设置 | 服务器端->管理端 |
| 设置符合条件的人员与设备间的绑定关系 | REQ\_BIND\_COND\_PERSON\_DEVICE | 0x0b04 | 管理端向服务器端设置查询人员条件，符合条件的人员有服务器端设置与设备的绑定关系 | 管理端->服务器端 |
| 回应设置符合条件的人员与设备间的绑定关系 | ACK\_BIND\_ COND\_PERSON\_DEVICE | 0x4b04 | 回应符合条件人员与设备的绑定关系 | 服务器端->管理端 |
|  |  |  |  |  |
| 设置人员属性 | REQ\_SET\_PERSON\_PROPERTY | 0x0b11 | 管理端向服务器端设置人员属性 | 管理端->服务器端 |
| 回应设置人员属性 | ACK\_SET\_PERSON\_PROPERTY | 0x4b11 | 回应人员与设备属性设置 | 服务器端->管理端 |
| ~~设置某种属性~~ | ~~REQ\_SET\_PROPERTY~~ | ~~0x0b12~~ | ~~管理端向服务器端设置某种属性~~ | ~~管理端->服务器端~~ |
| ~~响应设置某种属性~~ | ~~ACK\_SET\_PROPERTY~~ | ~~0x4b12~~ | ~~服务器端向管理端返回设置某种属性~~ | ~~服务器端->管理端~~ |
| ~~获取全部属性列表~~ | ~~REQ\_GET\_ PROPERTY\_LIST~~ | ~~0x0b13~~ | ~~管理端向服务器端获取全部属性列表~~ | ~~管理端->服务器端~~ |
| ~~响应获取全部属性列表~~ | ~~ACK\_GET\_PROPERTY\_LIST~~ | ~~0x4b13~~ | ~~服务器端向管理端返回全部属性列表~~ | ~~服务器端->管理端~~ |
|  |  |  |  |  |
| 获取某种属性的对象列表 | REQ\_GET\_ATTRIBUTE\_OBJECT\_LIST | 0x0c01 | 管理端向服务器端获取有某种相同属性的对象列表 | 管理端->服务器端 |
| 响应获取某种属性的对象列表 | ACK\_GET\_ATTRIBUTE\_OBJECT\_LIST | 0x4c01 | 服务器端向管理端返回符合要求的对象列表 | 服务器端->管理端 |
| 获取人员注册图像 | REQ\_GET\_PERSON\_IMAGES | 0x0c02 | 管理端向服务器端获取人员的注册图像，包括眼睛注册图像和照片 | 管理端->服务器端 |
| 响应获取人员注册图像 | ACK\_GET\_PERSON\_IMAGES | 0x4c02 | 服务器端向管理端人员的注册图像 | 服务器端->管理端 |
| 获取某种属性 | REQ\_SET\_PROPERTY | 0x0c01 | 管理端向服务器端获取有某种相同属性的对象列表 | 管理端->服务器端 |
| 响应获取某种属性的对象列表 | ACK\_GET\_ATTRIBUTE\_OBJECT\_LIST | 0x4c01 | 服务器端向管理端返回符合要求的对象列表 | 服务器端->管理端 |
| 获取某种属性的对象列表 | REQ\_GET\_ATTRIBUTE\_OBJECT\_LIST | 0x0c01 | 管理端向服务器端获取有某种相同属性的对象列表 | 管理端->服务器端 |
| 响应获取某种属性的对象列表 | ACK\_GET\_ATTRIBUTE\_OBJECT\_LIST | 0x4c01 | 服务器端向管理端返回符合要求的对象列表 | 服务器端->管理端 |
|  |  |  |  |  |
| 增加人员身份证信息 | REQ\_ADD\_PERSON\_IDCARD | 0x0207 | 客户端向服务器发送新增的人员信息 | 客户端->服务器端 |
| 响应增加人员身份证信息 | ACK\_ADD\_PERSON\_IDCARD | 0x4207 | 服务器向客户端回应新增的人员信息 | 服务器端->客户端 |
| 修改人员身份证信息 | REQ\_UPDATE\_PERSON\_IDCARD | 0x0208 | 客户端向服务器发送修改的人员信息 | 客户端->服务器端 |
| 响应修改人员身份证信息 | ACK\_UPDATE\_PERSON\_IDCARD | 0x4208 | 服务器向客户端回应修改的人员信息 | 服务器端->客户端 |
|  |  |  |  |  |
| 获取人员身份证信息 | REQ\_GET\_PERSON\_IDCARD | 0x0209 | 管理端向服务器要求获取人员身份证信息 | 管理端->服务器端 |
| 响应获取人员身份证信息 | ACK\_GET\_PERSON\_IDCARD | 0x4209 | 服务器向管理端回应人员身份证信息 | 服务器端->管理端 |
|  |  |  |  |  |

# 请求及回应

## 一般回应格式

大部分请求的回应帧格式如没有特殊说明，则均按照如下形式。

请求帧：

如果错误码=0，表示对应的请求成功，格式为

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 错误码 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | (2 Bytes) | 0xaaaa  (2 Bytes) | 4+length  (4 Bytes) | 0  (4Bytes) | 假设数据长度为length | (2 Bytes) |

如果错误码<0，表示对应的请求失败，并给出错误描述，格式为

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 错误码 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | (2 Bytes) | 0xaaaa  (2 Bytes) | 4+260  (4 Bytes) | 实际值  (4Bytes) | ErrorString  (260Bytes) | (2 Bytes) |

请求帧字节长度固定为：276Bytes。

## 获得人员列表

### 请求

请求方向：管理端→服务器端。

此请求用于管理端向服务器端获取当前服务数据库中已有的人员信息。

请求名称：REQ\_GET\_SRV\_PERSON\_LIST (0x0231)

请求帧：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | 0x0231  (2 Bytes) | 0xaaaa  (2 Bytes) | 332  (4 Bytes) | GetSrvPersonCondT  (332Bytes) | (2 Bytes) |

请求帧字节长度固定为：344Bytes。

其中从服务获得人员列表条件的数据结构参见[GetSrvPersonCondT](#_从服务获得人员列表的条件GetSrvPersonT)。

### 回应

请求方向：服务器端→管理端。

此请求用于服务器端响应管理端要求获取当前服务数据中已有的人员信息。

请求名称：ACK\_GET\_SRV\_PERSON\_LIST (0x4231)

请求帧：

当错误码为0时（表示请求成功执行，回应中包括人员的信息）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 错误码 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | 0x4231  (2 Bytes) | 0xaaaa  (2 Bytes) | 4+传送的数据长度  (4 Bytes) | 0  (4Bytes) | recordTotalCount+  SrvPersonBaseT数组  (4+  4+404Bytes\*个数) | (2 Bytes) |

请求帧字节长度不是固定值，以传送的实际数据长度为准。如果查询到的结果为0条，则返回的错误码仍为0（没有错误），但是传送的数据中，数组个数为0。

**实际传送的数据结构**为：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| recordTotalCount | len | SrvPersonBaseT | SrvPersonBaseT | …… |
| 4Bytes  按照本次的查询条件，一共有多少条记录。注意：是全部的记录条数，不是本次返回的记录条数。 | 4Bytes  后面跟着的SrvPersonBaseT个数 | 404Bytes | 404Bytes | …… |

其中，len表示后面的人员数据长度，即有多少个人员的数据。

其中人员信息的数据结构参见[SrvPersonBaseT](#_管理端从服务端获得的人员信息SrvPersonBaseT)。

当错误码不等于0，表示请求未成功执行，请求帧格式参见“一般回应格式”。

注意：由于每次获取的人员数量有限（最多1024个），所以如果要获取全部人员，可能需要经过多次请求。

获取的人员列表自动按照人名排序。

如果获取人员列表的条件中，设备序列号(deviceSn)有效，则服务收到此请求后向设备发送查询请求，根据设备返回结果作为回应发送给管理端；

如果获取人员列表的条件中，设备序列号(deviceSn)无效，则服务收到此请求后查询服务数据库，将查询结果作为回应发送给管理端。

隐患：服务与设备间获得人员信息中没有查询条件、没有排序、没有分页。

## 获得服务分页识别记录

### 请求

请求方向：管理端→服务器端

此请求用于管理端向服务器端要求获得指定的分页的识别记录。

请求名称：REQ\_GET\_PAGE\_RECOG\_RECORD (0x0452)

请求帧：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | 0x0452  (2 Bytes) | 0xaaaa  (2 Bytes) | 264  (4 Bytes) | GeSrvRecogRecordCondT  (264Bytes) | (2 Bytes) |

请求帧字节长度固定为264Bytes。

从服务获得指定识别记录请求的数据结构参见[GetSrvRecogRecordCondT](#_从服务获得识别记录GetSrvRecogRecordT)。

### 回应

请求方向：服务器端→管理端

此请求用于服务器端响应管理端要求获得服务端数据库中保存的指定识别记录的请求，按照分页返回，可能成功或失败。如果成功，则回应中包含符合要求的识别记录（个数可能为0）。

请求名称：ACK\_GET\_PAGE\_RECOG\_RECORD (0x4452)

请求帧：

如果响应成功，错误码为0，发送数据为符合要求的识别记录数组

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 错误码 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | 0x4452  (2 Bytes) | 0xaaaa  (2 Bytes) | 4+传送的数据长度  (4 Bytes) | 0  (4Bytes) | recordTotalCount+  SrvRecogRecordT数组  (4+4+252Bytes\*个数) | (2 Bytes) |

请求帧字节长度不是固定值，以传送的实际数据长度为准。

**实际传送的数据结构**为：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| recordTotalCount | len | SrvRecogRecordT | SrvRecogRecordT | …… |
| 4Bytes  按照本次的查询条件，一共有多少条记录。注意：是全部的记录条数，不是本次返回的记录条数。 | 4Bytes  后面跟着的SrvRecogRecordT个数 | 252Bytes | 252Bytes | …… |

单独一条识别记录的数据结构参见[SrvRecogRecordT](#_识别记录RecogRecordT)。

如果响应不成功，错误码不等于0，请求帧格式参见“一般回应格式”。

获得的识别记录是服务端数据库中保存的识别记录。

获得的识别记录自动按照识别时间逆序排列，即最新识别的识别记录在最前面。

## 获得服务分页带黑名单条件识别记录

### 请求

请求方向：管理端→服务器端

此请求用于管理端向服务器端要求获得指定的分页的识别记录。

请求名称：REQ\_GET\_PAGE\_RECOG\_RECORD\_BLACKLIST (0x0454)

请求帧：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | 0x0454  (2 Bytes) | 0xaaaa  (2 Bytes) | 268  (4 Bytes) | GeSrvRecogRecordBlacklistCondT  (268Bytes) | (2 Bytes) |

请求帧字节长度固定为280Bytes。

从服务获得指定识别记录请求的数据结构参见[GetSrvRecogRecordBlacklistCondT](#_获得属性识别记录的条件GetRecogRecordPropertyCo)。

### 回应

请求方向：服务器端→管理端

此请求用于服务器端响应管理端要求获得服务端数据库中保存的指定识别记录的请求，按照分页返回，可能成功或失败。如果成功，则回应中包含符合要求的识别记录（个数可能为0）。

请求名称：ACK\_GET\_PAGE\_RECOG\_RECORD\_BLACKLIST (0x4454)

请求帧：

如果响应成功，错误码为0，发送数据为符合要求的识别记录数组

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 错误码 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | 0x4454  (2 Bytes) | 0xaaaa  (2 Bytes) | 4+传送的数据长度  (4 Bytes) | 0  (4Bytes) | recordTotalCount+  SrvRecogRecordPropertyT数组  (4+4+256Bytes\*个数) | (2 Bytes) |

请求帧字节长度不是固定值，以传送的实际数据长度为准。

**实际传送的数据结构**为：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| recordTotalCount | len | SrvRecogRecordPropertyT | SrvRecogRecordPropertyT | …… |
| 4Bytes  按照本次的查询条件，一共有多少条记录。注意：是全部的记录条数，不是本次返回的记录条数。 | 4Bytes  后面跟着的SrvRecogRecordPropertyT个数 | 256Bytes | 256Bytes | …… |

单独一条识别记录的数据结构参见[SrvRecogRecordPropertyT](#_带属性的识别记录SrvRecogRecordPropertyT)。

如果响应不成功，错误码不等于0，请求帧格式参见“一般回应格式”。

获得的识别记录是服务端数据库中保存的识别记录。

获得的识别记录自动按照识别时间逆序排列，即最新识别的识别记录在最前面。

## 获取当前与服务连接的设备请求

### 请求

请求方向：管理端→服务器端

此请求用于管理端向服务要求获取当前与服务处于连接状态的设备。

请求名称：REQ\_GET\_CONNECT\_DEVICE (0x0b01)

请求帧：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | 0x0b01  (2 Bytes) | 0xaaaa  (2 Bytes) | 8  (4 Bytes) | SrvConnectDevCondT (8Bytes) | (2 Bytes) |

请求帧字节长度固定为：12Bytes。

获得当前与服务连接的设备请求数据结构参见[SrvConnectDevCondT](#_获得当前连接设备条件SrvConnectDevCondT)。

### 回应

响应方向：服务器端→管理端

此响应为服务器端回应管理端要求获取当前与服务处于连接状态的设备的请求。

请求名称：ACK\_GET\_CONNECT\_DEVICE (0x4b01)

请求帧：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 错误码 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | 0x4b01  (2 Bytes) | 0xaaaa  (2 Bytes) | 4+  传送的数据长度  (4 Bytes) | 0  (4Bytes) | recordTotalCount+  设备信息数组  (4+4Bytes+96Bytes\*个数) | (2 Bytes) |

请求帧字节长度不是固定值，以传送的实际数据长度为准。

**实际传送的数据结构**为：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| recordTotalCount | len | SrvConnectDevT | SrvConnectDevT | …… |
| 4Bytes  按照本次的查询条件，一共有多少条记录。注意：是全部的记录条数，不是本次返回的记录条数。 | 4Bytes  后面跟着的SrvConnectDevT个数 | 96Bytes | 96Bytes | …… |

其中，len表示后面的设备数组长度，即有多少设备的数据。

单个设备的数据结构参见[SrvConnectDevT](#_设备信息SrvConnectDevT)。

获得的设备是当前与服务处于连接状态的设备。

## 设置人员与设备间绑定关系

### 请求

请求方向：管理端→服务器端

此请求用于管理端向服务端设置人员与设备间的绑定关系，可以绑定也可以解绑。

请求名称：REQ\_BIND\_PERSON\_DEVICE (0x0b02)

请求帧：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | 0x0b02  (2 Bytes) | 0xaaaa  (2 Bytes) | 传送的数据长度  (4 Bytes) | 绑定信息数组  (4Bytes+84Bytes\*个数) | (2 Bytes) |

请求帧字节长度不是固定值，以传送的实际数据长度为准。

单个绑定信息的数据结构参见[DevPersonBindT](#_人员与设备的绑定关系DevPersonBindT)。

**实际传送的数据结构**为：

typedef list<DevPersonBindT> lstBinding;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| len | DevPersonBindT | DevPersonBindT | DevPersonBindT | …… |
| 4Bytes | 84Bytes | 84Bytes | 84Bytes | …… |

其中，len表示后面的绑定信息数组长度，即有多少个绑定的数据。

注意：当DevPersonBindT结构中，binding=0且deviceSn为空字符串时，表示指定人员与当前绑定设备全部解绑。

### 回应

请求方向：服务器端→管理端

此请求用于服务端回应管理端设置人员与设备间的绑定关系。

请求名称：ACK\_BIND\_PERSON\_DEVICE (0x4b02)

请求帧格式参见“[一般回应格式](#_一般回应格式)”。

## 获取人员与设备间的绑定关系

### 请求

请求方向：管理端→服务器端

此请求用于管理端向服务端要求获取某个指定人员与哪些设备当前处于绑定状态。

请求名称：REQ\_GET\_PERSON\_BIND\_INFO (0x0b03)

请求帧：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | 0x0b03  (2 Bytes) | 0xaaaa  (2 Bytes) | 16  (4 Bytes) | 指定人员的uuid  (16 Bytes) | (2 Bytes) |

请求帧字节长度是固定值，目前只能获取单个人员的绑定关系。

### 回应

请求方向：服务器端→管理端

此请求用于服务端回应管理端人员与设备的绑定关系。

请求名称：ACK\_ GET\_PERSON\_BIND\_INFO (0x4b03)

请求帧：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 错误码 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | 0x4b03  (2 Bytes) | 0xaaaa  (2 Bytes) | 4+  传送的数据长度  (4 Bytes) | 0  (4Bytes) | personId+  设备序列号数组  (16+4Bytes+64Bytes\*个数) | (2 Bytes) |

请求帧字节长度不是固定值，以传送的实际数据长度为准。

**实际传送的数据结构**为：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| personId | len | deviceSn | deviceSn | …… |
| 指定人员的personId(UUID)  与请求中的uuid一致 | 4Bytes  后面跟着的设备序列号个数 | 64Bytes | 64Bytes | …… |

其中，len表示后面的设备序列号数组长度，即有多少设备的数据。

## 设置符合条件的人员与设备间绑定关系

### 请求

请求方向：管理端→服务器端

此请求用于管理端向服务端设置符合条件的人员与设备间的绑定关系，可以绑定也可以解绑。

请求名称：REQ\_BIND\_COND\_PERSON\_DEVICE (0x0b04)

请求帧：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | 0x0b04  (2 Bytes) | 0xaaaa  (2 Bytes) | 392  (4 Bytes) | DevCondPersonBindT  (392Bytes) | (2 Bytes) |

请求帧字节长度固定值，为392字节。

符合条件的人员与设备绑定信息的数据结构参见[DevCondPersonBindT](#_符合条件的人员与设备绑定关系DevCondPersonBindT)。

### 回应

请求方向：服务器端→管理端

此请求用于服务端回应管理端设置人员与设备间的绑定关系。

请求名称：ACK\_ BIND\_COND\_PERSON\_DEVICE (0x4b04)

请求帧格式参见“[一般回应格式](#_一般回应格式)”。

## 获取某种属性的对象列表

### 请求

请求方向：管理端→服务器端

此请求用于管理端向服务端要求获取某个指定属性的对象列表，例如，获取当前存在的部门名称。

请求名称：REQ\_GET\_ATTRIBUTE\_OBJECT\_LIST (0x0c01)

请求帧：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | 0x0c01  (2 Bytes) | 0xaaaa  (2 Bytes) | 64  (4 Bytes) | 指定属性名称  (64Bytes) | (2 Bytes) |

请求帧字节长度是固定值，目前有效名称包括：depart。

### 回应

请求方向：服务器端→管理端

此请求用于服务端回应管理端要求获取的对象列表。

请求名称：ACK\_ GET\_ATTRIBUTE\_OBJECT\_LIST (0x4c01)

请求帧：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 错误码 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | 0x4c01  (2 Bytes) | 0xaaaa  (2 Bytes) | 4+  传送的数据长度  (4 Bytes) | 0  (4Bytes) | 对象数组  (4Bytes+68Bytes\*个数) | (2 Bytes) |

请求帧字节长度不是固定值，以传送的实际数据长度为准。

**实际传送的数据结构**为：

|  |  |  |  |
| --- | --- | --- | --- |
| len | ObjectT | ObjectT | …… |
| 4Bytes  后面跟着的对象个数 | 68Bytes | 68Bytes | …… |

其中，len表示后面的对象数组长度，即有多少对象的数据。

单个对象的结构定义见[ObjectT](#_有某种相同属性的对象ObjectT)。

请求属性：

depart —— 返回人员表中，所有部门名称，部门id均为0（无效）。如果有部门名称在数据库中为NULL（未设置），则返回列表中不会有这样的部门。

## 获取扩展识别记录

### 请求

请求方向：管理端→服务器端

此请求用于管理端向服务端要求获取符合要求的识别记录列表，识别记录中包含的信息比较全面。

请求名称：REQ\_GET\_SRV\_RECOG\_RECORD\_EX (0x0453)

请求帧：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | 0x0453  (2 Bytes) | 0xaaaa  (2 Bytes) | 80  (4 Bytes) | 指定属性名称  (80Bytes) | (2 Bytes) |

请求帧字节长度是固定值92个字节，对应的请求结构为[GetSrvRecogRecordExT](#_获取扩展识别记录GetSrvRecogRecordExT)。

此请求与REQ\_GET\_SRV\_RECOG\_RECORD(0x0451)的差别在于

1、按照不同的设置，可以多请求一个随机的识别记录，或者可以获取属于黑名单的识别记录，具体请求条件请见请求结构中的详细描述

2、如多请求了一个随机的识别记录，则此随机的识别记录在返回的识别记录数组中的最后

3、返回的识别记录中人员信息较为全面

### 回应

请求方向：服务器端→管理端

此请求用于服务端回应管理端符合要求的识别记录列表。

请求名称：ACK\_GET\_SRV\_RECOG\_RECORD\_EX (0x4453)

请求帧：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 错误码 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | 0x4453  (2 Bytes) | 0xaaaa  (2 Bytes) | 4+  传送的数据长度  (4 Bytes) | 0  (4Bytes) | 扩展识别记录数组  SrvRecogRecordExT  (4Bytes+396Bytes\*个数) | (2 Bytes) |

请求帧字节长度不是固定值，以传送的实际数据长度为准。

**实际传送的数据结构**为：

|  |  |  |  |
| --- | --- | --- | --- |
| len | SrvRecogRecordExT | SrvRecogRecordExT | …… |
| 4Bytes  后面跟着的对扩展识别记录个数 | 396Bytes | 396Bytes | …… |

其中，len表示后面的设备序列号数组长度，即有多少对象的数据。

单个扩展识别记录数据结构为[SrvRecogRecordExT](#_扩展识别记录SrvRecogRecordExT)。

## 获取人员注册图像

### 请求

请求方向：管理端→服务器端

此请求用于管理端向服务端要求获取人员注册的图像，包括左右眼最佳注册图像和照片。

请求名称：REQ\_GET\_PERSON\_IMAGES (0x0c02)

请求帧：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | 0x0c02  (2 Bytes) | 0xaaaa  (2 Bytes) | 36  (4 Bytes) | person UUID string  (36Bytes) | (2 Bytes) |

请求帧字节长度是固定值16个字节，为需要获取图像的人员uuid。

### 回应

请求方向：服务器端→管理端

此请求用于服务端回应管理端符合要求的人员的注册图像。

请求名称：ACK\_GET\_PERSON\_IMAGES (0x4c02)

请求帧：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 错误码 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | 0x4c02  (2 Bytes) | 0xaaaa  (2 Bytes) | 4+传送的数据长度  (4 Bytes) | 0  (4Bytes) | EnrolledImageT  (以实际情况为为准) | (2 Bytes) |

请求帧字节长度不是固定值，以传送的实际数据长度为准。

实际传送的数据结构参见[EnrolledImageT](#_人员注册图像EnrolledImageT)。

## 获得黑名单条件人员列表

### 请求

请求方向：管理端→服务器端。

此请求用于管理端向服务器端获取当前服务数据库中已有的人员信息（增加是否属于黑名单的查询条件）。

请求名称：REQ\_GET\_SRV\_PERSON\_BLACKLIST (0x0232)

请求帧：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | 0x0232  (2 Bytes) | 0xaaaa  (2 Bytes) | 372  (4 Bytes) | GetSrvPersonBlacklistCondT  (336Bytes) | (2 Bytes) |

请求帧字节长度固定为：412Bytes。

其中从服务获得人员列表条件的数据结构参见[GetSrvPersonBlacklistCondT](#_获得人员的附加条件GetPersonAttachCondT)。

### 回应

请求方向：服务器端→管理端。

此请求用于服务器端响应管理端要求获取当前服务数据中已有的人员信息（增加了人员属性）。

请求名称：ACK\_GET\_SRV\_PERSON\_BLACKLIST (0x4232)

请求帧：

当错误码为0时（表示请求成功执行，回应中包括人员的信息）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 错误码 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | 0x4232  (2 Bytes) | 0xaaaa  (2 Bytes) | 4+传送的数据长度  (4 Bytes) | 0  (4Bytes) | recordTotalCount+  SrvPersonPropertyT数组  (4+  4+408Bytes\*个数) | (2 Bytes) |

请求帧字节长度不是固定值，以传送的实际数据长度为准。如果查询到的结果为0条，则返回的错误码仍为0（没有错误），但是传送的数据中，数组个数为0。

**实际传送的数据结构**为：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| recordTotalCount | len | SrvPersonPropertyT | SrvPersonPropertyT | …… |
| 4Bytes  按照本次的查询条件，一共有多少条记录。注意：是全部的记录条数，不是本次返回的记录条数。 | 4Bytes  后面跟着的SrvPersonPropertyT个数 | 408Bytes | 408Bytes | …… |

其中，len表示后面的人员数据长度，即有多少个人员的数据。

其中带属性的人员信息数据结构参见[SrvPersonPropertyT](#_人员及属性信息SrvPersonPropertyT)。

当错误码不等于0，表示请求未成功执行，请求帧格式参见“一般回应格式”。

注意：由于每次获取的人员数量有限（最多1024个），所以如果要获取全部人员，可能需要经过多次请求。

获取的人员列表自动按照人名排序。

## 设置人员属性

### 请求

请求方向：管理端→服务器端

此请求用于管理端向服务端设置人员的某种属性，具体设置内容见数据结构中的约定描述。

请求名称：REQ\_ SET\_PERSON\_PROPERTY (0x0b11)

请求帧：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | 0x0b11  (2 Bytes) | 0xaaaa  (2 Bytes) | 传送的数据长度  (4 Bytes) | 属性设置数组  (4Bytes+112Bytes\*个数) | (2 Bytes) |

请求帧字节长度不是固定值，以传送的实际数据长度为准。

单个人员属性的数据结构参见[PersonPropertyT](#_人员属性设置PersonPropertyT)。

**实际传送的数据结构**为：

typedef list<[PersonPropertyT](#_人员属性设置PersonPropertyT)> lstProperty;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| len | [PersonPropertyT](#_人员属性设置PersonPropertyT) | [PersonPropertyT](#_人员属性设置PersonPropertyT) | [PersonPropertyT](#_人员属性设置PersonPropertyT) | …… |
| 4Bytes | 112Bytes | 112Bytes | 112Bytes | …… |

其中，len表示后面的人员属性数组长度，即有多少个人员属性设定的数据。

目前接受的属性包括：黑名单设置。

### 回应

请求方向：服务器端→管理端

此请求用于服务端回应管理端设置人员属性。

请求名称：ACK\_ SET\_PERSON\_PROPERTY (0x4b11)

请求帧格式参见“[一般回应格式](#_一般回应格式)”。

## 设置属性（暂不启用）

### 请求

请求方向：管理端→服务器端

此请求用于管理端向服务端设置某种属性。

请求名称：REQ\_ SET\_PROPERTY (0x0b12)

请求帧：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | 0x0b12  (2 Bytes) | 0xaaaa  (2 Bytes) | 传送的数据长度  (4 Bytes) | 属性设置数组  (4Bytes+68Bytes\*个数) | (2 Bytes) |

请求帧字节长度不是固定值，以传送的实际数据长度为准。

单个设置属性的数据结构参见[PropertyT](#_属性结构PropertyT)。

**实际传送的数据结构**为：

typedef list<[PropertyT](#_人员属性设置PersonPropertyT)> lstProperty;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| len | [PropertyT](#_人员属性设置PersonPropertyT) | [PropertyT](#_人员属性设置PersonPropertyT) | [PropertyT](#_人员属性设置PersonPropertyT) | …… |
| 4Bytes | 68Bytes | 68Bytes | 68Bytes | …… |

其中，len表示后面的人员属性数组长度，即有多少个人员属性设定的数据。

在实际使用中，如果结构中的property数值在数据库中已经存在，则对应的name改写为请求结构中的name；如果结构中的property数值在数据库中不存在，则创建一条新的属性记录。

### 回应

请求方向：服务器端→管理端

此请求用于服务端回应管理端设置属性。

请求名称：ACK\_SET\_PROPERTY (0x4b12)

请求帧格式参见“[一般回应格式](#_一般回应格式)”。

## 获取属性列表（暂不启用）

### 请求

请求方向：管理端→服务器端

此请求用于管理端向服务端获取所有属性列表。

请求名称：REQ\_ GET\_PROPERTY\_LIST (0x0b13)

请求帧：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 校验 |
| 0x5353  (2 Bytes) | 0x0b13  (2 Bytes) | 0xaaaa  (2 Bytes) | 0  (4 Bytes) | (2 Bytes) |

请求帧字节长度是固定值，为12字节。

### 回应

请求方向：服务器端→管理端

此请求用于服务端回应管理端设置人员属性。

请求名称：ACK\_GET\_PROPERTY\_LIST (0x4b13)

请求帧：

当错误码为0时（表示请求成功执行，回应中包括人员的信息）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 错误码 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | 0x4b13  (2 Bytes) | 0xaaaa  (2 Bytes) | 4+传送的数据长度  (4 Bytes) | 0  (4Bytes) | 属性数组  (4+68Bytes\*个数) | (2 Bytes) |

请求帧字节长度不是固定值，以传送的实际数据长度为准。如果查询到的结果为0条，则返回的错误码仍为0（没有错误），但是传送的数据中，数组个数为0。

单个设置属性的数据结构参见[PropertyT](#_属性结构PropertyT)。

**实际传送的数据结构**为：

typedef list<[PropertyT](#_人员属性设置PersonPropertyT)> lstProperty;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| len | [PropertyT](#_人员属性设置PersonPropertyT) | [PropertyT](#_人员属性设置PersonPropertyT) | [PropertyT](#_人员属性设置PersonPropertyT) | …… |
| 4Bytes | 68Bytes | 68Bytes | 68Bytes | …… |

其中，len表示后面的属性设置数组长度，即有多少个属性设定的数据。

## 新增人员身份证信息

### 请求

请求方向：客户端→服务器端

此请求用于客户端向服务端新增人员的身份证信息。

请求名称：REQ\_ ADD\_PERSON\_IDCARD(0x0207)

请求帧：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | 0x0207  (2 Bytes) | 0xaaaa  (2 Bytes) | 传送的数据长度  (4 Bytes) | 单个人员身份证信息  ??Bytes | (2 Bytes) |

请求帧字节长度不是固定值，以传送的实际数据长度为准。

单个人员身份证信息的数据结构参见[IdCardPersonInfoT](#_身份证人员信息IdCardPersonInfoT)。

### 回应

请求方向：服务器端→管理端

此请求用于服务端回应管理端设置属性。

请求名称：ACK\_ADD\_PERSON\_IDCARD(0x4207)

请求帧：

当错误码为0时（表示请求成功执行，回应中包括人员的信息）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 错误码 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | 0x4207  (2 Bytes) | 0xaaaa  (2 Bytes) | 4+传送的数据长度  (4 Bytes) | 0  (4Bytes) | 回应数据  88Bytes | (2 Bytes) |

请求帧字节长度是固定值，为104字节

单个回应的数据结构参见[IdCardRespT](#_身份证人员信息回应IdCardRespT)。

## 修改人员身份证信息

### 请求

请求方向：客户端→服务器端

此请求用于客户端向服务端修改人员的身份证信息。

请求名称：REQ\_ UPDATE\_PERSON\_IDCARD(0x0208)

请求帧：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | 0x0208  (2 Bytes) | 0xaaaa  (2 Bytes) | 传送的数据长度  (4 Bytes) | 单个人员身份证信息  ??Bytes | (2 Bytes) |

请求帧字节长度不是固定值，以传送的实际数据长度为准。

单个人员身份证信息的数据结构参见[IdCardPersonInfoT](#_身份证人员信息IdCardPersonInfoT)。

### 回应

请求方向：服务器端→客户端

此请求用于服务端回应客户端修改人员身份证信息。

请求名称：ACK\_UPDATE\_PERSON\_IDCARD(0x4208)

请求帧格式参见“[新增人员身份证信息回应](#_回应_8)”。

## 获取人员身份证详细信息

### 请求

请求方向：管理端→服务器端

此请求用于管理端向服务端要求获取人员身份证详细信息。

请求名称：REQ\_GET\_PERSON\_IDCARD(0x0209)

请求帧：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | 0x0209  (2 Bytes) | 0xaaaa  (2 Bytes) | 传送的数据长度  (4 Bytes) | 人员身份证号  20Bytes | (2 Bytes) |

请求帧字节长度是固定值，为32字节。

### 回应

请求方向：服务器端→管理端

此请求用于服务端回应管理端获取人员身份证详细信息请求，返回人员身份证详细信息。

请求名称：ACK\_GET\_PERSON\_IDCARD(0x4209)

请求帧：

当错误码为0时（表示请求成功执行，回应中包括人员身份证的详细信息）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 同步头 | 请求码 | 请求子码 | 实际数据长度 | 错误码 | 传送的数据 | 校验 |
| 0x5353  (2 Bytes) | 0x4b13  (2 Bytes) | 0xaaaa  (2 Bytes) | 4+传送的数据长度  (4 Bytes) | 0  (4Bytes) | 单个人员身份证信息  ??Bytes | (2 Bytes) |

请求帧字节长度不是固定值，以传送的实际数据长度为准。错误码为0人员查询到身份证数据，具体需要查看单个人员身份证信息中的errorNo。

单个人员身份证信息的数据结构参见[IdCardPersonInfoT](#_身份证人员信息IdCardPersonInfoT)。