|  |
| --- |
| **项目名称**：BiTIAS  **文件编号**：ZHKJ-YF-BiTIAS-BiVIEW-IDE-02-1  **文件版本**: V1.0  **ATS与ISCS系统界面接口规格书**  **浙江众合科技股份有限公司**  **研发中心** |
| |  |  |  |  | | --- | --- | --- | --- | | **编制** |  | **保密级别** | **普通（） 机密（√） 绝密（）** | | **审核** |  | **保存期限** | **短期（） 长期（√） 永久（）** | | **批准** |  | **发布日期** |  |   **本文件版权属于本公司所有，未经许可，不得复制引用** |

目录

[目录 iii](#_Toc528241763)

[0. 文档介绍 1](#_Toc528241764)

[0.1. 目的 1](#_Toc528241765)

[**0.2.** **文档范围** 1](#_Toc528241766)

[**0.3.** **文档上下文** 1](#_Toc528241767)

[**0.4.** **预期读者对象** 1](#_Toc528241768)

[0.5. 约定 1](#_Toc528241769)

[0.5.1. 需求条款约定 1](#_Toc528241770)

[0.5.2. 其它约定 1](#_Toc528241771)

[0.6. 术语与缩写解释 2](#_Toc528241772)

[0.7. 参考文档 2](#_Toc528241773)

[1. 概述 2](#_Toc528241774)

[2. 界面接口 2](#_Toc528241775)

[2.1. 界面对象接口 2](#_Toc528241776)

[2.1.1. AtsMmiUi类说明 2](#_Toc528241777)

[2.1.2. AtsTgpUi类说明 7](#_Toc528241778)

[2.1.3. ATS动态库加载器 8](#_Toc528241779)

[2.2. 信息交互接口 9](#_Toc528241780)

[2.2.1. 推图操作 9](#_Toc528241781)

[2.2.2. 顺控操作 11](#_Toc528241782)

[3. 用户权限 12](#_Toc528241783)

[4. 站场图跨屏显示 15](#_Toc528241784)

1. 文档介绍
   1. 目的

本文档为《ZHKJ-YF-BiTIAS-BiVIEW-SRS-01-1 BiVIEW子系统需求规格书》的附件，用于定义ATS系统与ISCS系统界面接口协议。

* 1. **文档范围**

本文档定义ATS系统与ISCS系统界面接口协议。

* 1. **文档上下文**

本文档作为《ZHKJ-YF-BiTIAS-BiVIEW-SRS-01-1 BiVIEW子系统需求规格书》的附件，不作为独立的文档进行上下游文件的追溯。

* 1. **预期读者对象**

ATS相关界面模块集成人员

* 1. 约定
     1. 需求条款约定

无

* + 1. 其它约定

无

* 1. 术语与缩写解释

|  |  |
| --- | --- |
| 缩写 | 中文 |
| ATS | 自动列车监控系统 |
| TIAS | 行车综合自动化系统 |
| ISCS | 综合监控系统 |

* 1. 参考文档

无

1. 概述

行车综合自动化系统TIAS集成ISCS系统和ATS系统人机界面，集中显示。ATS系统界面封装为动态库形式，提供给ISCS系统人机界面调用显示。

ATS系统包含站场图、派班计划、车辆调度、乘客调度和运行图等众多界面模块，其中站场图、派班计划、车辆调度和乘客调度界面模块共同封装入一个动态库文件（libAtsMmiUi），运行图单独封装为一个动态库文件（libAtsTgpUi）。下面主要对动态库文件接口进行介绍。

1. 界面接口
   1. 界面对象接口
      1. AtsMmiUi类说明

AtsMmiUi类主要用于创建站场图、派班计划、车辆调度和乘客调度界面模块，包括站场图、派班计划、列车信息、车辆光字牌、列车详情、车辆唤醒和乘客调界面。它主要接口如下：

**AtsMmiUi(QWidget \*pParent = 0, Qt::WindowFlags f = Qt::Widget, int screenNum = 1, const char\* cConsole = "", const char\* cStation = "");**

AtsMmiUi类的构造函数

|  |  |
| --- | --- |
| **参数** | **说明** |
| pParent | 父窗口 |
| f | 窗口类型 |
| screenNum | 屏幕个数若单屏显示站场图，则该参数值可不设置；若需跨多个屏显示站场图，则设置该参数值为相应的屏幕数 |
| cConsole | 控制台名称 |
| cStation | 车站名称，若显示线路全部站场图，则该参数值可不设置；若只需显示某个车站站场图界面，则设置该参数值为相应的车站名，例如显示杭州4号线近江站站场图界面，设置该参数值为“JinJiang” |
| **返回值** | **说明** |
| 无 | 无 |

**QWidget\* GetViewByParent(QWidget\* parent);**

获取“站场图”界面控件对象指针

|  |  |
| --- | --- |
| **参数** | **说明** |
| parent | 父窗口 |
| **返回值** | **说明** |
| QWidget\* | “站场图”界面控件对象指针 |

**QWidget\* GetDpUi(QWidget\* parent);**

获取“派班计划”界面控件对象指针

|  |  |
| --- | --- |
| **参数** | **说明** |
| parent | 父窗口 |
| **返回值** | **说明** |
| QWidget\* | “派班计划”界面控件对象指针 |

**QWidget\* GetVsTrainsList(QWidget\* parent) const;**

获取“列车信息”界面控件对象指针

|  |  |
| --- | --- |
| **参数** | **说明** |
| parent | 父窗口 |
| **返回值** | **说明** |
| QWidget\* | “列车信息”界面控件对象指针 |

**QWidget\* GetVsAlarmWin(QWidget\* parent) const;**

获取“车辆光字牌”界面控件对象指针

|  |  |
| --- | --- |
| **参数** | **说明** |
| parent | 父窗口 |
| **返回值** | **说明** |
| QWidget\* | “车辆光字牌”界面控件对象指针 |

**QWidget\* GetVsVehInfo(QWidget\* parent) const;**

获取“列车详情”界面控件对象指针

|  |  |
| --- | --- |
| **参数** | **说明** |
| parent | 父窗口 |
| **返回值** | **说明** |
| QWidget\* | “列车详情”界面控件对象指针 |

**QWidget\* GetVsWakeup(QWidget\* parent) const;**

获取“车辆唤醒”界面控件对象指针

|  |  |
| --- | --- |
| **参数** | **说明** |
| parent | 父窗口 |
| **返回值** | **说明** |
| QWidget\* | “车辆唤醒”界面控件对象指针 |

QWidget\* GetVsPassenger(QWidget\* parent);

获取“乘客调”界面控件对象指针

|  |  |
| --- | --- |
| **参数** | **说明** |
| parent | 父窗口 |
| **返回值** | **说明** |
| QWidget\* | “乘客调”界面控件对象指针 |

除此之外，为了便于其他程序显式调用动态链接库，提供获取类对象和各窗口对象的接口，主要有如下接口：

QWidget \*CreateGrObj(QWidget \*parent = 0, const int iScreenNum = 1, const char\* cConsole = "", const char\* cStation = "");

创建AtsMmiUi类对象。

|  |  |
| --- | --- |
| **参数** | **说明** |
| pParent | 父窗口 |
| screenNum | 屏幕个数若单屏显示站场图，则该参数值可不设置；若需跨多个屏显示站场图，则设置该参数值为相应的屏幕数 |
| cConsole | 控制台名称 |
| cStation | 车站名称，若显示线路全部站场图，则该参数值可不设置；若只需显示某个车站站场图界面，则设置该参数值为相应的车站名，例如显示杭州4号线近江站站场图界面，设置该参数值为“JinJiang” |
| **返回值** | **说明** |
| QWidget | AtsMmiUi类对象指针，该对象指针为全局变量 |

QWidget \*CreateGrObjWinObjOfAtsMmiUi(QWidget \*parent = 0);

创建"站场图"界面控件对象指针，如果全局AtsMmiUi类对象指针为空，则先创建AtsMmiUi类对象指针。

|  |  |
| --- | --- |
| **参数** | **说明** |
| parent | 父窗口 |
| **返回值** | **说明** |
| QWidget\* | “站场图”界面控件对象指针 |

QWidget \*CreateGrObjWinObjOfAtsDpUi(QWidget \*parent = 0);

创建“派班计划”界面控件对象指针，如果全局AtsMmiUi类对象指针为空，则先创建AtsMmiUi类对象指针。

|  |  |
| --- | --- |
| **参数** | **说明** |
| parent | 父窗口 |
| **返回值** | **说明** |
| QWidget\* | “派班计划”界面控件对象指针 |

QWidget \*CreateGrObjWinObjOfAtsVsTrainsList(QWidget \*parent = 0);

创建"车辆调度-列车信息"界面控件对象指针，如果全局AtsMmiUi类对象指针为空，则先创建AtsMmiUi类对象指针。

|  |  |
| --- | --- |
| **参数** | **说明** |
| parent | 父窗口 |
| **返回值** | **说明** |
| QWidget\* | “车辆调度-列车信息”界面控件对象指针 |

QWidget \*CreateGrObjWinObjOfAtsVsAlarmWin(QWidget \*parent = 0);

创建"车辆调度-光字牌"界面控件对象指针，如果全局AtsMmiUi类对象指针为空，则先创建AtsMmiUi类对象指针。

|  |  |
| --- | --- |
| **参数** | **说明** |
| parent | 父窗口 |
| **返回值** | **说明** |
| QWidget\* | “车辆调度-光字牌”界面控件对象指针 |

QWidget \*CreateGrObjWinObjOfAtsVsVehInfo(QWidget \*parent = 0);

创建"车辆调度-车辆详情"界面控件对象指针，如果全局AtsMmiUi类对象指针为空，则先创建AtsMmiUi类对象指针。

|  |  |
| --- | --- |
| **参数** | **说明** |
| parent | 父窗口 |
| **返回值** | **说明** |
| QWidget\* | “车辆调度-车辆详情”界面控件对象指针 |

QWidget \*CreateGrObjWinObjOfAtsVsWakeup(QWidget \*parent = 0);

创建"车辆调度-车辆唤醒"界面控件对象指针，如果全局AtsMmiUi类对象指针为空，则先创建AtsMmiUi类对象指针。

|  |  |
| --- | --- |
| **参数** | **说明** |
| parent | 父窗口 |
| **返回值** | **说明** |
| QWidget\* | “车辆调度-车辆唤醒”界面控件对象指针 |

QWidget \*CreateGrObjWinObjOfAtsVsPassenger(QWidget \*parent = 0);

创建"乘客调"界面控件对象指针，如果全局AtsMmiUi类对象指针为空，则先创建AtsMmiUi类对象指针。

|  |  |
| --- | --- |
| **参数** | **说明** |
| parent | 父窗口 |
| **返回值** | **说明** |
| QWidget\* | “乘客调”界面控件对象指针 |

* + 1. AtsTgpUi类说明

AtsTgpUi类封装运行图界面模块。它主要接口如下：

**AtsTgpUi (QWidget \*pParent = 0, Qt::WindowFlags f = Qt::Widget);**

AtsTgpUi类的构造函数

|  |  |
| --- | --- |
| **参数** | **说明** |
| pParent | 父窗口 |
| f | 窗口类型 |
| **返回值** | **说明** |
| 无 | 无 |

libAtsTgpUi动态库同样提供获取类对象和各窗口对象的接口，主要有如下接口：

QWidget \*CreateGrObj(QWidget \*parent = 0);

创建AtsTgpUi类对象指针

|  |  |
| --- | --- |
| **参数** | **说明** |
| pParent | 父窗口 |
| f | 窗口类型 |
| **返回值** | **说明** |
| QWidget\* | AtsTgpUi类对象指针，该对象指针为全局变量 |

QWidget \*CreateGrObjWinObjOfAtsTgpUi(QWidget \*parent);

获取"运行图"界面对象控件指针，如果全局AtsTgpUi类对象指针为空，则先创建AtsMmiUi类对象指针。

|  |  |
| --- | --- |
| **参数** | **说明** |
| pParent | 父窗口 |
| f | 窗口类型 |
| **返回值** | **说明** |
| Qwidget\* | “运行图”界面控件对象指针 |

* + 1. ATS动态库加载器

ATS提供动态库加载器AtsLoader，调用AtsLoader接口，可以加载ATS所有的界面动态库（目前ATS的libAtsMmiUi和libAtsTgpUi动态库）。加载之后，ATS系统的所有界面对象创建完成，当再次通过显式调用动态库方法获取各界面对象入口指针，不再创建新的界面对象，只是获取对象入口指针。加载器接口如下：

bool Load(const char\* cDllLibPath, QWidget \*parent = 0, const int iScreenNum = 1, const char\* cConsole = "", const char\* cStation = "");

加载动态库连接

|  |  |
| --- | --- |
| **参数** | **说明** |
| cDllLibPath | 动态库路径，动态库文件所在的路径 |
| parent | 父窗口 |
| iScreenNum | 屏幕个数，站场图界面将显示的屏幕个数 |
| cConsole | 控制台 |
| cStation | 车站 |
| **返回值** | **说明** |
| bool | 加载动态库是否成功 |

void Free();

释放动态库连接

|  |  |
| --- | --- |
| **参数** | **说明** |
| 无 | 无 |
| **返回值** | **说明** |
| 无 | 无 |

* 1. 信息交互接口

TIAS系统除了将ATS系统和ISCS系统集成显示之外，还增加了两个系统之间界面交互，主要包括ATS系统推图（调取导航图）操作和ATS系统顺控（供电区段上下电）操作。

* + 1. 推图操作

在ATS系统上操作设备图符， ISCS系统响应该操作并调取相应的导航图进行显示。两系统之间以QT信号槽方式进行数据交互。

信号槽接口描述如下：

void ATS2ISCSPopupGraphSignal(int iScreenNo, int iGraphNo, int iDispMode, int iGraphType, void\* pPara, int iParaLen);

ATS向ISCS发送推图消息信号

void OnATS2ISCSPopupGraphSignal(int iScreenNo, int iGraphNo, int iDispMode, int iGraphType, void\* pPara, int iParaLen);

ISCS响应ATS的推图信息

|  |  |
| --- | --- |
| **参数** | **说明** |
| screenNo | 屏幕编号，将推图显示在那个屏幕，默认值为最后屏幕编号 |
| iGraphNo | 图元编号，图元分为站场图和乘客调两种类型，其中站场图包含风机、乘客流、火灾、电扶梯和CCTV五种图元控件，每个图元编号为车站ID+图元控件类型 |
| iDispMode | 推图显示模式 |
| iGraphType | 图元类型 |
| pPara | 附加参数 |
| iParaLen | 附加参数长度 |
| **返回值** | **说明** |
| 无 | 无 |

站场图图元控件类型描述如下：

|  |  |  |
| --- | --- | --- |
| **常量** | **值** | **描述** |
| TIAS\_ATS\_CONTROL\_BAS | 0x0101 | 风机控件 |
| TIAS\_ATS\_CONTROL\_AFC | 0x0201 | 乘客流控件 |
| TIAS\_ATS\_CONTROL\_FIRE | 0x0301 | 火灾控件 |
| TIAS\_ATS\_CONTROL\_ESCA | 0x0401 | 电扶梯控件 |
| TIAS\_ATS\_CONTROL\_CCTV | 0x0501 | CCTV控件 |

乘客调包含的图元控件编号描述如下：

|  |  |  |
| --- | --- | --- |
| **常量** | **值** | **描述** |
| TIAS\_ATS\_CONTROL\_VEH\_CCTV | 0x0111 | 车辆CCTV |
| TIAS\_ATS\_CONTROL\_STA\_CCTV | 0x0211 | 车站CCTV |
| TIAS\_ATS\_CONTROL\_VEH\_PA | 0x0311 | 车辆PA |
| TIAS\_ATS\_CONTROL\_STA\_PA | 0x0411 | 车站PA |
| TIAS\_ATS\_CONTROL\_VEH\_PIS | 0x0511 | 车辆PIS |
| TIAS\_ATS\_CONTROL\_STA\_PIS | 0x0611 | 车站PIS |

显示模式描述如下：

|  |  |  |
| --- | --- | --- |
| **常量** | **值** | **描述** |
| TIAS\_GRAPH\_DISPMODE\_NORMAL | 0 | 在HMI主界面正常显示 |
| TIAS\_GRAPH\_DISPMODE\_NEWWINDOW | 1 | 在新弹出窗口显示 |

图元类型描述如下：

|  |  |  |
| --- | --- | --- |
| **常量** | **值** | **描述** |
| GRAPH\_TYPE\_NOPARA | 0 | 无参数的图形 |
| GRAPH\_TYPE\_PA\_TRAIN | 1 | 带PA列车参数的图形 |
| GRAPH\_TYPE\_PIS\_TRAIN | 2 | 带PIS列车参数的图形 |
| GRAPH\_TYPE\_CCTV\_TRAIN | 3 | 带CCTV列车参数的图形 |
| GRAPH\_TYPE\_CCTV\_STATION | 4 | 带CCTV车站参数的图形 |

* + 1. 顺控操作

在ATS系统上对供电区段进行上下电操作， ISCS系统响应该操作，执行上下电操作，并更新供电图显示。两系统之间以QT信号槽方式进行数据交互。

信号槽接口描述如下：

void ATS2ISCSSeqControlSignal(int iScadaIdx, int iCommandType);

ATS向ISCS发送执行顺控信号

void OnATS2ISCSSeqControlSignal(int iScadaIdx, int iCommandType);

ISCS响应ATS的执行顺控

|  |  |
| --- | --- |
| **参数** | **说明** |
| iScadaIdx | 供电区段索引 |
| iCommandType | 命令类型 |
| **返回值** | **说明** |
| 无 | 无 |

命令类型描述如下：

|  |  |
| --- | --- |
| **值** | **描述** |
| 0x1 | 上电 |
| 0x2 | 下电 |

1. 用户权限

用户权限由集成系统TIAS统一管理，ISCS系统负责向ATS系统提供用户登录和退出信息，ATS用户根据用户登录和退出信息来决定操作和浏览工作模式，操作模式下的操作规则管理有原ATS配置的方法实现。

ISCS系统与ATS系统之间通过QT信号槽方式进行用户信息交互。用户信息交互需要分两种情况，情况一是 ATS系统启动时向ISCS系统发送用户信息请求信号，ISCS收到ATS请求信号后，回复ATS发送用户信息；情况二是ISCS在进行用户登录或退出时，主动向ATS发送用户信息。



图 1 情况一：用户权限控制



图 2 情况二：用户权限控制

void ATS2ISCSUserInfoReqSignal();

ATS向ISCS发送用户信息请求信号

void OnATS2ISCSUserInfoReqSignal()

ISCS响应ATS的用户信息请求

|  |  |
| --- | --- |
| **参数** | **说明** |
| 无 | 无 |
| **返回值** | **说明** |
| 无 | 无 |

void ISCS2ATSUserInfoReplySignal(StructISCS2ATSUserInfoReply struUserInfo)；

ISCS向ATS发送用户信息信号

void OnISCS2ATSUserInfoReplySignal(StructISCS2ATSUserInfoReply struUserInfo);

ATS响应ISCS的用户信息

|  |  |
| --- | --- |
| **参数** | **说明** |
| struUserInfo | 用户信息结构体 |
| **返回值** | **说明** |
| 无 | 无 |

用户信息结构体如下：

|  |
| --- |
| typedef struct tagISCS2ATSUserInfoReply {  unsigned int iUserID; // 用户ID  char cUserName[255]; // 用户名  unsigned int iUserType; // 用户所属类别  unsigned char iLoginStatus; // 用户登录状态 1:login, 2:logout  tagISCS2ATSUserInfoReply(void)  {  iUserID = 0;  memset(cUserName, 0, sizeof(cUserName));  iUserType = 0;  iLoginStatus = 0;  }  } StructISCS2ATSUserInfoReply; |

1. 站场图跨屏显示

动态库站场图支持跨屏显示，可以在同一台多屏主机各屏幕上同时显示ATS站场图界面，并且任意两屏、三屏或多屏之间ATS站场图界面实现无缝动态拼接显示。调用多屏显示，在创建AtsMmiUi类对象时，将屏幕数量参数screenNum设置成所需要显示的屏幕数量，同时设置父窗口pParent的属性 “ScreenNo”的值（setProperty函数）。