# librbac.dll中接口：

MSG\_ID\_T **RBAC**\_Initialize() ：只针对界面锁定弹框参数初始化，ehmi登陆时调用；

VOID **RBAC**\_LoginLock() ：用户触发锁定界面时调用，弹出锁定弹框；

MSG\_ID\_T **RBAC**\_Login(char \*Privilege,uint8 LoginLimit,

bool bShowAOJ,int nDlgShowType,int nLoginType) ：程序登录接口

参数：

Privilege：系统权限，为空，不验证权限，否则验证登录用户的系统权限，有此权限才能登录成功；

LoginLimit：用户密码验证错误后，连续登录次数限值，如果为0，表示不限制；

bShowAOJ：弃用参数；

nDlgShowType：0表示小屏界面,1表示大屏登陆界面；

nLoginType：0表示登陆,1表示退出,2表示锁定,5表示注销

MSG\_ID\_T **RBAC**\_SpecLogin(char \*Privilege,char \*UserName,char \*UserNameDesc) ：显示用户登录窗口，输出用户名和用户描述，参数Privilege无效；ehmi中双席监督时有调用

# librbac.dll中接口：

MSG\_ID\_T **AOJ**\_Initialize ()、MSG\_ID\_T AOJ\_Download () ：无实际作用接口，不需要调用；

MSG\_ID\_T **AOJ**\_GetConsoleDesc (char \*ConsoleName,char \*ConsoleDesc) ：获取指定ConsoleName(主机名)对应的主机名描述(ConsoleDesc)，成功返回1，亦可验证此主机是否属于系统内的节点；

void **AOJ**\_GetList (AOJ\_D\_INFO\_TABLE &aojList,uint32 &AojCount)：获取所有权限区域列表；

MSG\_ID\_T **AOJ**\_GetDescription (AOJ\_D\_ASSIGNMENT AojId, char \*AojDesc) ：根据指定的权限区域id数组，输出对应的权限区域描述AojDesc，不同描述之间以点号'.'分割，成功返回1；

MSG\_ID\_T **AOJ**\_GetAssignment (char \*ConsoleName, AOJ\_D\_ASSIGNMENT AojId, char \*Supervision ) ：指定权限区域权限(对象授权表) Supervision，查找登录用户哪些权限区域有这样的权限，返回这些权限区域id数组AojId，其中Supervision可以是点号'.'分割的多个权限，它为NULL或空时，返回此用户所有的权限区域id集合，参数ConsoleName不为NULL时，此参数值若不是本地节点(本地主机名)会直接返回-1，否则返回AojId的个数；

MSG\_ID\_T **AOJ**\_CheckRelease (char \*ConsoleName, AOJ\_D\_ASSIGNMENT CurrentAojs, AOJ\_D\_ASSIGNMENT UnattendedAojs)、

MSG\_ID\_T **AOJ**\_InitAssignment (char \*ConsoleName, AOJ\_D\_ASSIGNMENT AojId, ROLE\_SUPERVISION \*AojSupervision, uint32 aojnum, bool bOperator)、MSG\_ID\_T **AOJ**\_ResetAssignment (char \*ConsoleName) ：

弃用接口；

MSG\_ID\_T **AOJ**\_Validate(uint32 AojValue, AOJ\_D\_ASSIGNMENT AojId)：AojValue在数组AojId中存在，则成功返回1；

MSG\_ID\_T **RBAC**\_GetUserRoleDescription(char \*Logger, char \*UserDesc, char \*RoleDesc, char \*RoleName)：获取登录用户名Logger或用户描述UserDesc或角色名RoleName或角色描述RoleDesc，参数可以为NULL；

MSG\_ID\_T **RBAC**\_GetRolePLevel (char \*RoleName, uint32 \*PLevel)：获取角色控制优先级，"role"不为NULL，获取指定角色名的Plevel，否则获取登录用户的PLevel，PLevel是一个元素为3的数组，下标0表示CCTV，1表示PIS，2表示PA；

void **RBAC**\_RequestOfUi(std::string &sendType,std::string &reqData,std::string &resultJson)：向服务端发送post请求，并获取结果resultJson，如librbac调用 ；

void **RBAC**\_OptRestFulObjInfoToUi(uint16 optType,std::string &valStr)：设置或者获取用户名、主机名等信息，提供给librbac使用；

int **RBAC**\_UpdateLocalInfo()：获取登录用户的用户名、角色名、权限区域信息，接口类型" updateinfo"；

MSG\_ID\_T **RBAC**\_InitUserInfo(char \*ConsoleName, char \*UserName, char \*Password)：隐式登录，命令方式登录程序调用，ConsoleName无效参数，UserName用户名，Password密码；

MSG\_ID\_T **RBAC**\_Logout(char \*ConsoleName)：注销登录用户信息，退出程序时调用，ConsoleName无效参数；

bool **RBAC**\_CheckPrivilege(char \*Privilege)：验证登录用户是否有某项系统权限，如调度员权限；

MSG\_ID\_T **RBAC**\_GetRoleDescription (char \*RoleName, char \*RoleDesc, char \*PrivilegeDesc=NULL, char \*AojDesc=NULL)：获取指定角色RoleName的角色描述，权限区域描述AojDesc(多个权限区域用点号分割)，PrivilegeDesc无效参数；

bool **RBAC**\_GetUserRoleAll(std::vector<User\_Info\_Stu> &userInfo,std::vector<Role\_Info\_Stu> &roleInfo)：获取用户表、角色表所有信息

int **RBAC**\_UpdateUserInfo(User\_Info\_Stu &userInfo)：修改用户信息，结构体中以"olduser"为准，与"user"不同则更新，其他字段不为空就更新，需要更改的选项不能为空，为空则跳过此字段；

int **RBAC**\_UpdateRoleInfo(Role\_Info\_Stu &roleInfo)：修改角色信息，以"oldrole"为准，与"role"不同则更新，其他字段不为空就更新，需要更改的选项不能为空，为空则跳过此字段，"plevel ":三个数按顺序 0-CCTV; 1-PIS; 2-PA；

int **RBAC**\_UpdateAojInfo(Aoj\_Info\_Stu &aojInfo)：修改权限区域信息，以"oldaojid"为准进行更新，"oldaojid"和"aojid "必须有值，且大于等于0，相同时不更新此字段，"aoj"，"aojdesc"为空则不更新对应字段；

BOOL **RBAC**\_ModifyAojAssignInfo(AojAssignment\_Info\_Stu &aojAssignInfo)：修改对象授权表信息 aojAssignInfo.opt=1插入,=2更改,=3删除，"aojid"不能为0，"almtype": 0表示不发报警信息；1表示角色修改报警；2表示aoj修改报警；

BOOL **RBAC**\_GetAssignedAojPvgInfo(AojAssignment\_Info\_Stu &aojAssignInfo,AssignedAojPvg\_List\_Stu &aojPvgList)：获取指定role或aojid的权限区域授权信息(对象授权表) aojAssignInfo.opt=1指定角色，=2指定aojid，AojAssignment\_Info\_Stu中只有opt、aojId、role有效；

int **RBAC**\_AddNewUser(User\_Info\_Stu &userInfo)：添加新用户

int **RBAC**\_AddNewRole(Role\_Info\_Stu &roleInfo)：添加新角色

int **RBAC**\_AddNewAoj(Aoj\_Info\_Stu &aojInfo)：添加新权限区域

int **RBAC**\_DeleteUser(std::string &userName,uint16 forceDel)：删除用户

int **RBAC**\_DeleteRole(std::string &roleName,uint16 forceDel)：删除角色

int **RBAC**\_DeleteAoj(uint32 aojId)：删除权限区域