

# PolyLinkCTI 项目

---

## 座席端 OCX 控件开发模块

### 概要设计说明书

## 修改履历

序号	修改时间	修改人	修改内容
1	2012-8-30	Jack	新建
2	2012-11-2	Jack	添加控件属性、方法、事件
3	2015-12-24	Jie. Zhao	添加成员属性 SoftPhoneAppProcessName 方法 DoShowKeyPad()
4	2016-7-6	Baron	整理修改整个文件。
5	2016-7-7	Jie. Zhao	修改排版，删除废弃接口 <b>PredictCallOutBridgeRingEvent</b> ，新增错误码

## 目 录

1	引言	4
1.1	目的 .....	4
1.2	目标读者 .....	4
1.3	软件名称 .....	4
1.4	统一命名规则 .....	4
1.5	定义、首字母缩写词和缩略语 .....	5
1.6	参考资料 .....	6
2	PlAgentBar 控件结构	6
2.1	模块在系统中的位置 .....	6
3	接口模块设计	7
3.1	整体功能概述 .....	7
3.2	功能函数接口类 .....	7

# 1 引言

## 1.1 目的

座席端的 OCX 控件是座席客户端软件和 CTI 之间互相交互的 OCX 控件。不仅通过与交换机模块交互实现电话接听、电话呼出、来电显示等基本功能，而且还实现了保持、转接、三方通话、监听、强插、强拆、监控等呼叫中心特有的功能。

## 1.2 目标读者

CTI 需求分析人员、开发人员、测试人员和产品相关人员。

## 1.3 软件名称

现定义本文所要开发的座席端 OCX 控件的名称为：PLAgentBar。

## 1.4 统一命名规则

名词	英文名	解释
登入	Login	座席端连接到 CTI，即通常所说的座席登入
登出	Logout	座席端断开与 CTI 的连接，即通常所说的座席登出
签入	SignIn	座席端 PLAgentBar 连接到 CTI
签出	SignOut	座席端 PLAgentBar 断开连接到 CTI
转接	Transfer	座席 A 接听了客户的来电后，客户说要找座席 B，座席 A 使用“转接”操作将来电转到座席 B
静音	Mute	跟保持(hold)很类似，但不播放背景音乐。
分机	Extension	座席的物理电话机实体。
PSTN 话机	Phone	PSTN 话机
呼出	callout	座席拨打一个电话，此电话可以是内部分机，也可以是外部固话或者手机。
挂断	Hangup	座席在跟客户通话中才能执行挂断操作，挂断后双方通话结束。
保持	Hold	座席在跟客户通话的过程中可以把当前通话保持，保持后，客户一方只能听到背景音乐，但是听不到座席一方的任何声音。如果要继续通话需要取回保持。
取回保持	Retrieve	取回保持的电话。可恢复正常通话。
接听	Answer	当有来电时才可以使用。接听成功后座席跟客户建立会话。
强插	Join	三方通话的一种，这种方式用于座席和客户产生纠纷或者座席无法解决问题时，班长可以直接插进他们之间的通话中，此时任意一方都能听到其他两人的声音。
密语	whisper	三方通话的一种。属于班长的监控功能之一。这种方式跟插话的区别在于：密语后客户不能听到班长和座席的声音。
咨询	Consult	三方通话的一种。属于班长的监控功能之一。这种方式

		发生在座席跟客户通话中时，由座席发起向第三方请求协助，此时跟客户的通话处于保持状态，当与第三方通话结束后，再取消保持，继续跟客户通话。
监听	Monitor	班长的监控功能之一。班长能够听到座席当前的通话，但座席跟客户完全听不到班长的任何声音，也察觉不到有人在监听。
强拆	forceHangup	班长的监控功能之一。班长强行挂断指定座席的通话。
抢接	Rop	班长的监控功能之一。当班长发现某座席有来电并且正在响铃时，而座席并没有接听，此时班长可以使用抢接功能接听到这个电话。
振铃	Ring	CTI 需要将来电分配给指定座席接听，会调用该命令控制 IPPBX 呼叫该坐席，之后座席的电话会响铃。

## 1.5 定义、首字母缩写词和缩略语

缩略语	英文全名	中文释义
UE	User Equipment	用户终端设备
CPS	Calls Per Second	每秒呼叫数
API	Application Programming Interface	是一些预先定义的函数，目的是提供应用程序与开发人员基于某软件或硬件的以访问一组例程的能力，而又无需访问源码，或理解内部工作机制的细节。
FS	FreeSwitch	一个开源的 PBX 系统
OCX	Object Linking and Embedding (OLE) Control eXtension	对象类别扩充组件

## 1.6 参考资料

CTI_Server_Requirements.docx	CTI 需求分析文档，SVN 的 Doc 目录下
开发进度计划表	Svn://192.17.1.16/CTI-R1/Doc/CallCenter.mpp
概要设计文档	Svn://192.17.1.16/CTI-R1/Doc/CTIPreDesign.docx

## 2 PIAgentBar 控件结构

### 2.1 模块在系统中的位置

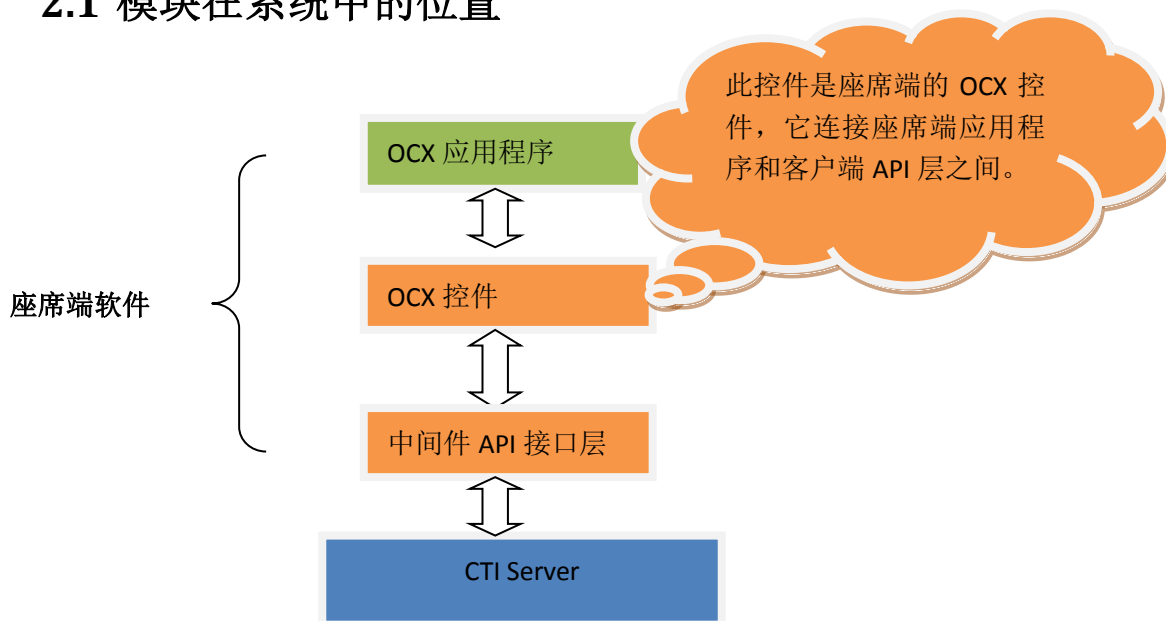


图 2.1-1 PIAgentBar 在系统中的位置图

该控件内嵌在座席客户端软件内，以图形化的操作界面给座席人员以最有效方式工作。座席分为普通座席和班长座席。普通座席可以通过座席端 OCX 控件进行接听、呼出、咨询等，班长座席除了具有普通座席的功能外，还提供监听、强插、强拆等班长席特有功能。

## 3 接口模块设计

### 3.1 整体功能概述

AgentBar 是使用 C#2008 开发的 ActiveX 控件，主要包括控件的属性、方法和事件三大部分。

### 3.2 功能函数接口类

#### 3.2.1 类名

AgentBar

#### 3.2.2 成员属性

◆ **string AgentID**

座席号

◆ **string AgentExten**

分机号

◆ **string AgentPwd**

坐席登陆密码

◆ **string cuID**

电话唯一标识，当前电话的 Callid，即一通电话的唯一标识。

◆ **string WebUrl**

弹屏网址，暂不用。

◆ **string ServerIP**

服务器 IP 地址

◆ **string ServerPort**

服务器端口号

◆ **string AgentName**

座席姓名

◆ **bool BindExten**

是否绑定分机

◆ **string AgentStatus**

座席登录后初始状态

◆ **bool GradeSwitch**

是否可以评分

◆ **string AgentGroupID**

坐席组 ID

◆ **string AgentGroupName**

坐席组名称

◆ **string RoleID**

角色 ID

◆ **string HeartBeatTimeout**

心跳超时时间，此值以服务器的设置为准，客户端设置的值将被覆盖。

◆ **string SigninIntervalAfterSoftPhoneRegistered**

软电话上线后等待多久再签入。这是因为签入之前为防止 ctiserver 速度慢导致未检测到分机上线而导致签入失败，所以先 sleep 下。默认 50 毫秒。

◆ **bool SoftPhoneEnable**

是否使用内置软电话

◆ **string SipNum**

内置软电话登录的分机号

◆ **string SipPwd**

内置软电话分机密码

◆ **string SipServer**

内置软电话服务器 ip 地址

◆ **int SipPort**

内置软电话服务器端口号

◆ **int SipRegistTime**

内置软电话注册时间超时间隔

◆ **bool SipAutoAnswer**

内置软电话是否自动接听

◆ **bool SoftphoneAutoAnswer**

内置软电话不自动接听登录前提下，设置电话呼出时内置软电话是否自动接听

◆ **bool bindSoftPhoneLogin**

是否绑定内置软电话登录

◆ **string SipLocalNum**

内置软电话本机使用号码

◆ **bool IsMonitorOfflineAgent**

是否监控离线坐席

◆ **string SaltKey**

密钥，用于来电弹屏加密

◆ **string DefaultAccessNum**

指定外呼时选择的默认接入号。

◆ **list CalloutHistory**

外呼时呼叫号码历史列表。

◆ **int RefreshReportStatisInterval**

现场监控屏历史数据部分刷新闻隔，默认 60 秒。

◆ **Agent\_Status AgentStateAfterHangup**

坐席电话挂断后需要置为什么状态。如果不设置当跟客户通话结束后会默认置为后处理状态（包括呼入呼出），而内部呼叫通话结束后置为空闲状态。

◆ **Bool SipAutoSignIn**

使用中间件登录时，内置软电话是否连同中间件自动登录，默认 false。

◆ **Bool SipPhoneOnLineWarning**

中间件登录系统时，内置软电话正在运行的情况下，是否提示正在运行，重新登录，默认 false。

◆ **string NoAnswerCallsURL**

未接来电数据获取 url。

◆ **string SoftPhoneAppProcessName**

设置内置软电话的进程名，默认值为 VideoTelephone



### 3.2.3 成员方法

#### ◆ **int DoConnect ()**

功能说明:

连接到 CTI Server 。

必填属性:

ServerIP      服务器 IP 地址

Port          端口号

返回:

返回 0 代表连接成功, 其他代表失败。

#### ◆ **Bool DoDisconnect()**

功能说明:

断开与 CTI Server 之间的连接。

返回:

返回 true 代表连接成功, false 失败。

#### ◆ **Bool DoHeartBeat()**

功能说明:

发送心跳包。发送周期由签入成功时服务器送的时间为准。一般是 30 秒

必填属性:

AgentID      座席号

返回:

返回 true 代表连接成功, false 失败。

#### ◆ **Bool DoSignIn()**

功能说明:

发起座席签入的请求。如果 AgentExten 为空则不绑定座席, 不为空则会绑定这个分机。

必填属性:

AgentID      座席号

AgentPwd     座席密码

AgentExten   座席分机

BindExten    是否绑定分机

AgentStatus   登录后的初始状态。

WebURL       web 页面的 URL 地址

ExtenIsOutbound 分机是否外线

返回:

返回 true 代表连接成功, false 失败。

#### ◆ **Bool DoSignOut()**

功能说明:

同 DoSignIn 函数对应。从 CTIServer 上签出。

必填属性:

AgentID      座席号

返回:

返回 true 代表连接成功, false 失败。

◆ **Bool DoSignInWithConnect ()**

功能说明:

带连接的签入, 签入前会先连接到 CTIServer。如果 AgentExten 为空则不绑定座席, 不为空则会绑定这个分机。

必填属性:

ServerIP	服务器 IP 地址
Port	端口号
AgentID	座席号
AgentPwd	坐席登陆密码
AgentStatus	登录后的初始状态。

可选属性:

AgentExten	座席分机
BindExten	是否绑定分机
WebURL	web 页面的 URL 地址 (暂时不用)

返回:

返回 true 代表连接成功, false 失败。

◆ **Bool DoCallOut(string CalledID, string displayNum)**

功能说明:

呼叫一个外线号码。

必填属性:

AgentID	座席号, 必填
---------	---------

传入参数:

CalledID	电话号码, 必填
displayNum	呼出时使用的号码, 如果为空则使用默认路由。

**注: 坐席的状态必须是 AGENT\_STATUS\_IDLE 或者 AGENT\_STATUS\_CALL\_OUT。**

◆ **Bool DoCallOutForeignId (string CalledID, string displayNum, string customerForeignId)**

功能说明:

呼叫一个外线号码(只提供接口)。

必填属性:

AgentID	座席号, 必填
---------	---------

传入参数:

CalledID	电话号码, 必填
displayNum	呼出时使用的号码, 如果为空则使用默认路由
customerForeignId	客户外来 ID

**注: 坐席的状态必须是 AGENT\_STATUS\_IDLE 或者 AGENT\_STATUS\_CALL\_OUT。**

◆ **Bool DoCallAgent(string AgentNum)**

功能说明:

呼叫座席号码。

**必填属性:**

AgentID      座席号, 必填

**传入参数:**

AgentNum      座席工号, 必填

**返回:**

返回 true 代表成功, false 失败。

**◆ Bool DoHangUp ()****功能说明:**

挂断电话。

**必填属性:**

AgentID      座席号, 必填

**返回:**

返回 true 代表成功, false 失败。

**◆ Bool DoGetAgentDefineStatus ()****功能说明:**

获得座席自定义状态。

**必填属性:**

AgentID      座席号, 必填

**返回:**

返回 true 代表连接成功, false 失败。

**◆ bool DoGetAgentWebSiteInfo()****功能说明:**

获得座席网址收藏夹列表。

**必填属性:**

AgentID      座席号, 必填

**传入参数:**

无

**返回:**

返回 true 代表获取成功, false 失败。

**◆ Bool DoHold ()****功能说明:**

保持。

**必填属性:**

AgentID      座席号, 必填

**返回:**

返回 true 代表成功, false 失败。

**◆ Bool DoStopHold ()****功能说明:**

取消保持。

必填属性:

AgentID          座席号, 必填

返回:

返回 true 代表成功, false 失败。

◆ **Bool DoMute ()**

功能说明:

静音。

必填属性:

AgentID          座席号, 必填

返回:

返回 true 代表成功, false 失败。

◆ **Bool DoStopMute ()**

功能说明:

取消静音。

必填属性:

AgentID          座席号, 必填

返回:

返回 true 代表成功, false 失败。

◆ **Bool DoConsult (string destAgentID,bool trdisOutbound)**

功能说明:

咨询。

必填属性:

AgentID          座席号, 必填

AgentExten      座席分机, 必填

destAgentID      被咨询的座席号, 必填

trdisOutbound    被咨询者是否是外线, 必填

返回:

返回 true 代表成功, false 失败。

◆ **Bool DoStopConsult ()**

功能说明:

取消咨询。

必填属性:

AgentID          座席号, 必填

AgentExten      座席分机, 必填

返回:

返回 true 代表成功, false 失败。

◆ **Bool DoConsultTransfer ()**

功能说明:

询问转。

**必填属性:**

AgentID      座席号, 必填  
AgentExten   座席分机, 必填

**返回:**

返回 true 代表成功, false 失败。

**◆ Bool DoListen (string destAgentID)****功能说明:**

监听。

**必填属性:**

AgentID      座席号, 必填  
AgentExten   座席分机, 必填

**传入参数:**

destAgentID 被监听的座席工号, 必填

**返回:**

返回 true 代表成功, false 失败。

**◆ Bool DoWhisper (string destAgentID)****功能说明:**

咨询。

**必填属性:**

AgentID      座席号, 必填  
AgentExten   座席分机, 必填

**传入参数:**

destAgentID 被密语的座席工号, 必填

**返回:**

返回 true 代表成功, false 失败。

**◆ bool DoForceChangeStatusEvent (string strAgentID, string status)****功能说明:**

强制更改某坐席状态。

**必填属性:**

AgentID      座席号, 必填

**传入参数:**

strAgentID   被更改的坐席工号  
status        被更改后的坐席状态

**返回:**

返回 true 代表获取成功, false 失败。

**◆ Bool DoTransferAgent (string destAgentID , string strOutBoundFlag))****功能说明:**

转接到座席。

**必填属性:**

AgentID      座席号, 必填

AgentExten 座席分机，必填

传入参数：

destAgentID 被密语的座席工号，必填

strOutBoundFlag 转接外线，值为 true 或 false

返回：

返回 true 代表成功，false 失败。

◆ **Bool DoTransferIvr (string ivrNum)**

功能说明：

转接到座席。

必填属性：

AgentID 座席号，必填

AgentExten 座席分机，必填

传入参数：

ivrNum ivr 号，必填

返回：

返回 true 代表成功，false 失败。

◆ **Bool DoTransferQueue (string queueNum)**

功能说明：

转接到队列。

必填属性：

AgentID 座席号，必填

AgentExten 座席分机，必填

传入参数：

queueNum 要转到的队列号，必填

返回：

返回 true 代表成功，false 失败。

◆ **bool DoTransferIvrProfile (string ivrProfileNum)**

功能说明：

转接到 ivr profile。

必填属性：

AgentID 座席号，必填

AgentExten 座席分机，必填

传入参数：

ivrProfileNum ivr profile 号码

返回：

返回 true 代表获取成功，false 失败。

◆ **Bool DoSetAgentDefineStatus (int targetStatus, int IsNeedApproval)**

功能说明：

设置座席的状态。

必填属性：

AgentID            座席号，必填

传入参数：

targetStatus       设置的座席状态，必填

isNeedApproval    是否需要申请，0：需要申请 1:不需要申请

返回：

返回 true 代表成功，false 失败。

◆ **Bool DoBargein (string strAgentID)**

功能说明：

插话。

必填属性：

AgentID           座席号，必填

AgentExten        座席分机，必填

传入参数：

strAgentID        被插话的座席工号，必填

返回：

返回 true 代表成功，false 失败。

◆ **Bool DoForceHangup (string strAgentID)**

功能说明：

强拆。

必填属性：

AgentID           座席号，必填

AgentExten        座席分机，必填

传入参数：

strAgentID        被强拆的座席工号，必填

返回：

返回 true 代表成功，false 失败。

◆ **Bool DoHeartBeat ()**

功能说明：

心跳检测。

必填属性：

AgentID           座席号，必填

返回：

返回 true 代表成功，false 失败。

◆ **Bool DoCheck ()**

功能说明：

测试电话是否正常使用，即环回测试。如果正常则会收到一个来电，接通话能够听到自己的声音。

必填属性：

AgentID           座席号，必填

AgentExten        分机号，必填

返回:

返回 true 代表连接成功, false 失败。

◆ **Bool DoGrade (string language)**

功能说明:

评分。

必填属性:

AgentID 座席号, 必填

AgentExten 分机号, 必填

传入参数:

language 评分时使用的语言, 目前支持有中英文两种。

返回:

返回 true 代表连接成功, false 失败。

◆ **Bool DoGetAccessNumbers ()**

功能说明:

获得接入号。

必填属性:

AgentID 座席号, 必填

返回:

返回 true 代表连接成功, false 失败。

◆ **Bool DoThreeWay (string trdAgentNum, bool trdIsOutbound)**

功能说明:

三方邀请。

必填属性:

AgentID 座席号, 必填

AgentExten 分机号, 必填

传入参数:

trdAgentNum 第三方号码, 必填, 当为坐席时此为坐席工号

trdIsOutbound 第三方是否是外线

返回:

返回 true 代表连接成功, false 失败。

◆ **Bool DoThreeWayCancel ()**

功能说明:

取消三方邀请。

必填属性:

AgentID 座席号, 必填

AgentExten 分机号, 必填

返回:

返回 true 代表连接成功, false 失败。

◆ **Bool DoGetOnlineAgent(string specificNum,int numType)**



**功能说明:**

获得所有在线座席列表。

**必填属性:**

AgentID 座席号, 必填

AgentExten 座席分机, 必填

当 numType 为 1 或 2 时, specificNum 不能为空。

**传入参数:**

specificNum 坐席工号或所有坐席 (此时值为空) 或某坐席组名称

numType 类型 (1: 指定的坐席工号 2: 某个坐席组名称 3: 所有坐席)

**返回:**

返回 true 代表连接成功, false 失败。

**◆ Bool DoGetIvrList()****功能说明:**

获得所有 IVR 列表。

**必填属性:**

AgentID 座席号, 必填

AgentExten 座席分机, 必填

**返回:**

返回 true 代表连接成功, false 失败。

**◆ Bool DoGetQueueList()****功能说明:**

获得所有队列列表。

**必填属性:**

AgentID 座席号, 必填

**返回:**

返回 true 代表连接成功, false 失败。

**◆ bool DoGetIvrProfileList ()****功能说明:**

获得 ivr profile 列表。

**必填属性:**

AgentID 座席号, 必填

AgentExten 座席分机, 必填

**返回:**

返回 true 代表获取成功, false 失败。

**◆ Bool DoGetDefinedRoleRights ()****功能说明:**

获得自定义的角色和权限。

**必填属性:**

AgentID 座席号, 必填

**返回:**

返回 true 代表连接成功，false 失败。

◆ **Bool DoGetAgentGroupList ()**

功能说明：

获得所有座席组列表。

必填属性：

AgentID          座席号，必填

返回：

返回 true 代表连接成功，false 失败。

◆ **Bool DoGetAgentsOfQueue (string QueueNum)**

功能说明：

获得某队列的座席列表。

必填属性：

AgentID          座席号，必填

传入参数：

QueueNum        队列号码，一次只能查询一个队列

返回：

返回 true 代表连接成功，false 失败。

◆ **Bool DoGetAgentsOfAgentGroup(string AgentGroupNum)**

功能说明：

获得某坐席组的座席列表。

必填属性：

AgentID          座席号，必填

传入参数：

AgentGroupNum   坐席组号码，一次只能查询一个坐席组

返回：

返回 true 代表连接成功，false 失败。

◆ **Bool DoGetAgentsMonitorInfo (string AgentsStr)**

功能说明：

获得一批坐席的基本监控信息。

必填属性：

AgentID          座席号，必填

传入参数：

AgentsStr        可以是一个或一批座席号码，多个用逗号隔开

返回：

返回 true 代表连接成功，false 失败。

◆ **Bool DoGetDetailCallInfo (string AgentNum)**

功能说明：

获得某个坐席的详细监控信息。

必填属性：

AgentID      座席号，必填

传入参数：

AgentNum      座席号码

返回：

返回 true 代表连接成功，false 失败。

◆ **Bool DoGetCustomerOfQueue (string QueueNum)**

功能说明：

获得某个队列的客户排队监控信息。

必填属性：

AgentID      座席号，必填

传入参数：

QueueNum      队列号码

返回：

返回 true 代表连接成功，false 失败。

◆ **bool DoGetCustomerOfMyQueue ()**

功能说明：

获取本坐席所在的队列的监控信息。含队列名称、队列号码、队列等待人数、最大等待时长等。

必填属性：

AgentID      座席号，必填

返回：

返回 true 代表获取成功，false 失败。

◆ **bool DoGetQueueStatis (string queue\_num)**

功能说明：

获取队列监控统计信息。包括：队列名称、队列号码、队列排队数、坐席空闲数、坐席离开数、坐席通话数、坐席离线数

必填属性：

AgentID      座席号，必填

传入参数：

queue\_num      要监控的队列号码

返回：

返回 true 代表获取成功，false 失败。

◆ **bool DoGetAllQueueStatis ()**

功能说明：

获取系统中所有队列监控统计信息。包括：排队数、最大排队时长、坐席空闲数、坐席离开数、坐席通话数、坐席离线数，用于现场监控屏。

必填属性：

AgentID      座席号，必填

返回：

返回 true 代表获取成功，false 失败。

**◆ bool DoGetReportStatisInfo()****功能说明:**

获取系统中当天报表统计信息。用于现场监控屏。对应事件为: **GetReportStatisInfoEvent**

**必填属性:**

无

**返回:**

返回 true 代表获取成功, false 失败。

**◆ bool DoApplyChangeStatus (string targetStatus)****功能说明:**

发起一个(离开)申请。

**必填属性:**

AgentID          座席号, 必填

**传入参数:**

targetStatus    目标状态的值。7: 表示离开, 必填

**返回:**

返回 true 代表获取成功, false 失败。

**◆ bool DoApplyApproval (string apply\_agent\_id,string targetStatus,string passFlag)****功能说明:**

审批一个申请。

**必填属性:**

AgentID          座席号, 必填

**传入参数:**

Apply\_agent\_id    申请者座席号, 必填

targetStatus    目标状态的值。7 表示离开, 必填

passFlag          审批是否通过, 1:通过 0:不通过

**返回:**

返回 true 代表获取成功, false 失败。

**◆ bool DoGetAgentGroupStatusMaxNum (string targetAgentGroupNameLstStr, string targetAgentGroupIdLstStr, string targetStatus)****功能说明:**

获取坐席组的申请离开阈值。

**必填属性:**

AgentID          座席号, 必填

**传入参数:**

targetAgentGroupNameLstStr    要获取的坐席组名称列表, 多个用逗号隔开, 必填

targetAgentGroupIdLstStr        要获取的坐席组 id 列表, 多个用逗号隔开, 必填

targetStatus    目标状态的值。7 表示离开, 必填

**返回:**

返回 true 代表获取成功, false 失败。

◆ **bool DoSetAgentGroupStatusMaxNum(string targetAgentGroupNameLstStr, string targetAgentGroupIdLstStr, string targetStatus)**

功能说明：

设置坐席组申请离开人数阈值。

必填属性：

AgentID 座席号，必填

传入参数：

targetAgentGroupNameLstStr 要获取的坐席组名称列表，多个用逗号隔开，必填

targetAgentGroupIdLstStr 要获取的坐席组 id 列表，多个用逗号隔开，必填

targetStatus 目标状态的值。7 表示离开，必填

返回：

返回 true 代表获取成功，false 失败。

◆ **bool DoGetChangeStatusApplyList(string targetStatus)**

功能说明：

获取申请离开记录列表。审批者刚登录时需要获取。

必填属性：

AgentID 座席号，必填

传入参数：

targetStatus 目标状态的值。7 表示离开，必填

返回：

返回 true 代表获取成功，false 失败。

◆ **bool DoClickAgentBarButtonByName (string btnName)**

功能说明：

通过 agentBar 上按钮名称触发按钮事件。

必填属性：

btnName 按钮名称。必填。AgentBar 中各个按钮名称如下：

按钮名称	解释
tsbCancelApplication	取消申请
tsbAnswer	接听
tsbKeyPad	拨号盘
tsbCallAgent	内部呼叫
tsbCallOut	外呼
tsbHangUp	挂断
tsbTransfer	转接
tsbGrade	评分
tsbHold	保持
tsbStopHold	取消保持
tsbThreeWay	三方
tsbThreeWayCancel	取消三方
tsbConsult	询问
tsbStopConsult	取消询问

tsbConsultTransfer	询问转
tsbListen	监听
tsbWhisper	密语
tsbBargein	插话
tsbForceHangup	强拆
tsddbAfterHangup	挂断后是否转为空闲
tsbCheck	环回测试
tsbMonitor	监控
tsbApprove	离开审批

返回:

返回 true 代表获取成功, false 失败。

#### ◆ bool DoShowKeyPad()

功能说明:

调用内置软键盘的接口

#### ◆ CallStatus GetCallStatus ()

功能说明:

获取当前电话的状态

返回:

返回 CallStatus. NO\_CALL 代表未有电话或电话已挂断

返回 CallStatus. CALL\_IN 代表电话是呼入型的

返回 CallStatus. CALL\_OUT 代表电话是呼出型的

### 3.2.4 请求响应

#### ◆ ServerResponse (String agentID,Int retCode, String reason)

功能说明:

发送的请求命令后如果失败总会收到一个响应失败事件。

传出参数:

AgentID	座席号
retCode	返回的错误码, 如果是 0 代表成功, 否则是个错误代码, 详见错误代码文档。根据错误码可以知道具体错误原因。
Reason	错误原因。即错误描述, 不过是全英文的。

错误码和错误描述:

### 3.2.5 结构体

#### ◆ 服务器中规定的状态值,具体值由服务器定义

```
public enum Agent_Status
{
    AGENT_STATUS_UNKNOWN=-2,
    AGENT_STATUS_OFFLINE=-1,
    AGENT_STATUS_IDLE = 0,
```

```

AGENT_STATUS_RING = 1,
AGENT_STATUS_TALKING = 2,
AGENT_STATUS_HOLD = 3,
AGENT_STATUS_ACW = 4,
AGENT_STATUS_CAMP_ON = 5,
AGENT_STATUS_BUSY = 6,
AGENT_STATUS_LEAVE = 7,
AGENT_STATUS_CALL_OUT = 8,
AGENT_STATUS_MONITOR = 9,
AGENT_STATUS_CALLING_OUT=10,
AGENT_STATUS_MUTE=11,
AGENT_STATUS_RESTORE=999,          //恢复原状态
}

```

- 说明：**
- 1、坐席发起手动外呼的时候，坐席的状态必须是AGENT\_STATUS\_IDLE或者AGENT\_STATUS\_CALL\_OUT。
  - 2、客户主动来电服务器分配坐席的时候，只会分配给坐席状态为空闲的坐席。
  - 3、坐席间的所有电话呼叫，需要根据CTI\_Web上面的配置，设置哪些状态下可以进行坐席间的电话呼叫。

#### ◆ 客户端中需使用的状态

```

public enum Agent_State
{
    AGENT_STATUS_UNKNOWN,
    AGENT_STATUS_SIGNIN,
    AGENT_STATUS_IDLE,
    AGENT_STATUS_OFFLINE,
    AGENT_STATUS_CONSULTING,
    AGENT_STATUS_THREEWAYTING,
    AGENT_STATUS_TALKING_CUSTOMER,
    AGENT_STATUS_TALKING_CONSULT,
    AGENT_STATUS_ANSWER_CONSULT,
    AGENT_STATUS_TALKING_THREEWAY,
    AGENT_STATUS_TALKING_INTERNAL,
    AGENT_STATUS_TALKING_EAVESDROP,
    AGENT_STATUS_TALKING_WHISPER,
    AGENT_STATUS_TALKING_BARGEIN,
    AGENT_STATUS_CAMP_ON,
    AGENT_STATUS_ANSWER_CALLEE,          //收到被叫answer
    AGENT_STATUS_BRIDGE_CALLEE,         //收到被叫bridge
}

```

#### ◆ 队列中等待的客户信息

```

public struct Customer_Info_Struct
{
    public string call_type;          //呼叫类型
}

```

```
public string callcenter_num;    //接入号
public string customer_num;     //客户号码
public string customer_status;  //客户状态
public string customer_type;    //客户类型
public string customer_uuid;    //客户 uuid
public string enter_queue_time; //进队列时间
public string queue_num;        //队列号码
public string queue_name;       //队列名称
public int orderId;             //排序
public string queue_customer_amount; //队列等待人数
public ulong wait_time;        //等待时长
public string customer_enter_channel; //进线通道
public string exclusive_agent_num; //专属坐席号码
public string exclusive_queue_num; //专属队列号码
}
```

#### ◆ 电话的状态

```
public enum CallStatus
{
    NO_CALL = 0,           //没有电话或电话已挂断
    CALL_IN = 1,           //电话是呼入型
    CALL_OUT = 2,          //电话是呼出型
}
```

### 3.2.6 触发事件

#### ◆ CallInEvent (string callerID, string makeStr, string callType, Boolean bNeedArea, String outExtraParamsFromIvr)

功能说明:

普通来电事件。

传出参数:

callerID 来电号码

makeStr 随录数据, 是一个字符串, 里面包含了很多参数信息, 如: 工号、被叫号码、队列号码等以#分隔。

显示格式如:

```
customer_num=8008123456#callcenter_num=61003456#access_num_name=接入号
1#agent_num=1001#relation_uuid=123334ad3834343bea#area_id=12#area_name=江苏-苏州
#cust_grade=0#customer_num_format_local=123456789#customer_num_format_national=123
456789#customer_num_format_e164=0123456789#customer_num_phone_type=+8612345678
9 #;
```

解释如下:

customer\_num: 客户号码

callcenter\_num: 接入号

access\_num\_name: 接入号名称



agent\_num: 已分配的坐席工号  
relation\_uuid: 唯一标识  
area\_id: 地区 id  
area\_name: 地区名称, 省市已“-”分隔  
cust\_grade: 客户等级 0: 普通客户 1: vip  
callType 呼叫类型  
CTI\_CALL\_TYPE\_COMMON\_CALL\_IN: 普通呼入  
~~CTI\_CALL\_TYPE\_MANUAL\_CALL\_OUT: 手动呼出~~  
CTI\_CALL\_TYPE\_PREDICT\_CALL\_OUT: 预测式外呼  
bNeedArea 是否需要提示来电归属地(默认是 true)  
customer\_num\_format\_local: 本地电话  
customer\_num\_format\_national: 区号 + 本地电话  
customer\_num\_format\_e164: 国家码 + 区号 + 本地电话  
customer\_num\_phone\_type: 电话类型, 固话: fixed\_line, 手机: mobile  
outExtraParamsFromIvr ivr 随录数据 (传输时使用 base64 加密)

~~◆ PredictCallOutBridgeRingEvent (string callerID, string make\_str, string callType, Boolean bNeedArea, string outExtraParamsFromIvr)~~

~~功能说明:~~

~~预测式外呼来电事件。~~

~~传出参数:~~

~~参上。~~

注: 该事件已经废弃, 预测式外呼的分配响铃事件合并至 **CallInEvent**, 通过 callType 区分是普通呼入还是预测式外呼的分配响铃。

◆ **CallOutRingEvent** (string customerNum, string callcenterNum, string accessNumName, string callType, string areaID, string areaName, string makeStr)

功能说明:

呼出响铃事件。

传出参数:

customerNum	客户号码
callcenterNum	呼叫中心号码
accessNumName	接入号
callType	呼叫类型
areaID	号码归属地 ID
areaName	号码归属地名
makeStr	是一个字符串, 里面包含了很多参数信息, 如: 本地电话等

显示格式如:

```
#customer_num_format_local=123456789#customer_num_format_national=123456789#  
customer_num_format_e164=0123456789#customer_num_phone_type=+86123456789 #  
#customerForeignId=123456#;
```

◆ **InternalCall\_CallInEvent** (string agentid, string callerid, string callType)

功能说明:

内部呼叫主叫来电事件。

传出参数:

agentid	本地座席工号。
callerid	主叫座席工号。
callType	呼叫类型。

#### ◆ InternalCallerRingEvent (string agentid,string callerid,string callType)

功能说明:

内部呼叫被叫来电事件。

传出参数:

agentid	本地座席工号。
callerid	主叫座席工号。
callType	呼叫类型。

#### ◆ ThreeWayCallRingEvent (string agentID, string customerNum, string callcenterNum, string accessNumName, string callerAgentNum, string callType, string areaID, string areaName, string custGrade, string outExtraParamsFromIvr, string makeStr)

功能说明:

三方来电事件。

传出参数:

callerID	来电号码
customerNum	客户号码
callcenterNum	接入号
accessNumName	接入号名称
callerAgentNum	发起三方坐席
callType	呼叫类型
areaID	号码归属地 ID
areaName	号码归属地名称
custGrade	客户等级 0: 普通 1: VIP
outExtraParamsFromIvr	ivr 随录数据 (传输时使用 base64 加密)
customer_num_format_local	本地电话
customer_num_format_national	区号 + 本地电话
customer_num_format_e164	国家码 + 区号 + 本地电话
customer_num_phone_type	电话类型, 固话: fixed_line, 手机: mobile

#### ◆ HangupEvent (string hangupReason, int retCode)

功能说明:

挂断事件。主要指本地坐席。

传出参数:

retCode	事件结果。0 成功 <0 失败。
hangupReason	挂断原因, 包括如下几种:
NO_ANSWER	未接
NO_USER_RESPONSE	拒接
CALL_REJECTED	拒接

ORIGINATOR\_CANCEL: 对方挂断  
NORMAL\_CLEARING: 通话正常结束  
NO\_ROUTE\_DESTINATION: 转接

◆ **BridgeEvent ()**

功能说明:

接通事件。是双通。单通是 answerEvent(), 当询问成功后主叫和被叫都会收到此事件。

◆ **CheckExtenEvent (string agentid)**

功能说明:

环回测试。

传出参数:

agentid                      本地座席工号

◆ **ThreeWayeeHangupEvent (string agentid, string hangupReason)**

功能说明:

三方时, 被邀请者挂断事件。

传出参数:

agentid                      本地座席工号  
hangupReason                挂断原因

◆ **ConsulteeHangupEvent (string agentid, string hangupReason)**

功能说明:

询问时, 被询问者挂断事件。

传出参数:

agentid                      本地座席工号  
hangupReason                挂断原因

◆ **ConsultCallInEvent (string agentID, string customerNum, string callcenterNum, string accessNumName, string consulterAgentNum, string callType, string areaID, string areaName, string custGrade, string outExtraParamsFromIvr, string makeStr)**

功能说明:

来电询问事件。

传出参数:

agentID                      本地坐席 ID  
customerNum                客户号码  
callcenterNum              呼叫中心号码  
accessNumName              接入号  
consulterAgentNum          被咨询的坐席 ID  
callType                    呼叫类型  
areaID                      号码归属地 ID  
areaName                    号码归属地名  
custGrade                   客户等级 0: 普通客户 1: VIP  
outExtraParamsFromIvr    ivr 随录数据 (传输时使用 base64 加密)

- ◆ **TransferBlindCallInEvent**(string agentID, string customerNum, string callcenterNum, string accessNumName, string consulterAgentNum, string callType, string areaID, string areaName, string custGrade, string outExtraParamsFromIvr, string makeStr)

功能说明:

盲转事件。

传出参数:

agentID	本地坐席工号
customerNum	客户号码
callcenterNum	呼叫中心号码
accessNumName	接入号
consulterAgentNum	被盲转的坐席工号
callType	呼叫类型。包含以下几种类型:
areaID	号码归属地 ID
areaName	号码归属地名
custGrade	客户等级 0: 普通客户 1: VIP
outExtraParamsFromIvr	ivr 随录数据 (传输时使用 base64 加密)

- ◆ **EavesdropEvent** (string agentid, string reason, string retCode)

功能说明:

监听请求的结果事件。

传出参数:

agentid	本地座席工号
reason	监听失败原因,retCode 不为 0 时有效
retCode	错误码, 0 表示成功, 否则是错误码。

- ◆ **EavesdropCallRingEvent** (string agentID, string customerNum, string callcenterNum, string accessNumName, string desAgentID, string callType, string areaID, string areaName, string custGrade, string outExtraParamsFromIvr, string makeStr)

功能说明:

监听请求的结果事件。

传出参数:

agentID	本地坐席工号
customerNum	客户号码
callcenterNum	呼叫中心号码
accessNumName	接入号
desAgentID	目标坐席 ID
callType	呼叫类型。包含以下几种类型:
areaID	号码归属地 ID
areaName	号码归属地名
custGrade	客户等级 0: 普通客户 1: VIP
outExtraParamsFromIvr	ivr 随录数据 (传输时使用 base64 加密)

- ◆ **ForceHangupEvent** (string agentid, string reason,int retCode)

功能说明:

强拆的结果事件。

传出参数:

agentid	本地座席工号
reason	监听失败原因
retCode	错误码, 0 表示成功, 否则失败。

◆ **WhisperEvent (string agentid, string reason,int retCode)**

功能说明:

密语请求的结果事件。

传出参数:

agentid	本地座席工号
reason	监听失败原因
retCode	错误码, 0 表示成功, 否则失败。

◆ **BargeinEvent (string agentid, string reason,int retCode)**

功能说明:

插话请求的结果事件。

传出参数:

agentid	本地座席工号
reason	监听失败原因
retCode	错误码, 0 表示成功, 否则失败。

◆ **WhisperCallRingEvent (string agentID, string customerNum, string callcenterNum, string accessNumName, string desAgentID, string callType, string areaID, string areaName, string custGrade, string outExtraParamsFromIvr, string makeStr)**

功能说明:

发起密语后振铃的事件。

传出参数:

参上

◆ **BargeinCallRingEvent (string agentID, string customerNum, string callcenterNum, string accessNumName, string desAgentID, string callType, string areaID, string areaName, string custGrade, string outExtraParamsFromIvr, string makeStr)**

功能说明:

发起强插后振铃的事件。

传出参数:

同上

◆ **ForceHangupCallRingEvent (string agentID, string customerNum, string callcenterNum, string accessNumName, string desAgentID, string callType, string areaID, string areaName, string custGrade, string outExtraParamsFromIvr, string makeStr)**

功能说明:

发起强拆后振铃的事件。

传出参数:

同上

#### ◆ GetAgentOnlineEvent (List<Agent\_Online\_Struct> agent\_online)

功能说明:

获取在线座席的请求结果事件。(当使用 javascript 接收时对应的事件为:  
[JSGetAgentOnlineEvent](#)(string jsonArgsStr), 详见“针对 javascript 接收事件的调整说明”)

传出参数:

Agent\_online                      目前所有在线的座席列表, 其中 Agent\_Online\_Struct 表示的是在线座席结构体。

定义如下:

```
public struct Agent_Online_Struct
{
    public string agentgroup_name;           //座席组名称
    public string agentgroup_num;           //座席组id
    public string agentName;                //座席姓名
    public string agentNum;                 //座席工号
    public string agentStatus;              //座席状态
    public bool bindExten;                  //是否绑定分机
    public string flag;                     //状态标识。-99: 座席不存在或不可用 -1:离
线 0:在线
    public string roleNum;                  //角色id
    public string status_change_time;        //座席状态改变时间
    public string customer_enter_channel;    //客户进线通道
}
```

#### ◆ GetAccessNumberEvent (string[] accessNumbers,string default\_access\_number,List<string> callHistory)

功能说明:

获得所有接入号列表和历史外呼记录的事件。对应 javascript 的事件为  
“[JSGetIvrListEvent](#)”。

传出参数:

accessNumbers      ,              接入号列表  
default\_access\_number ,              默认接入号  
callHistory              ,              外呼历史号码列表

#### ◆ GetIvrListEvent (Dictionary<string,string> ivr\_list)

功能说明:

获得所有 IVR 列表的事件。对应 javascript 的事件为 “JSGetAccessNumberEvent”。

传出参数:

ivr\_list                      ivr 列表, key 是 ivr 号, value 是 ivr 名称

#### ◆ GetQueueListEvent (Dictionary<string,string> queue\_list)

功能说明:

获得所有队列列表的事件。对应 javascript 的事件为 “[JSGetQueueListEvent](#)”。

传出参数:

queue\_list                      队列列表, 其中 key 是队列号, value 是队列名称

◆ **GetRoleNameEvent (string roleName)**

功能说明:

获得角色名称事件。

传出参数:

roleName                      角色名称

注: 此事件没有对应的请求接口, 签入成功后会自动触发此事件。

◆ **GetAgentGroupListEvent (Dictionary<string, string> agent\_group\_list)**

功能说明:

获得我的座席组列表事件。对应 javascript 的事件为 “[JSGetAgentGroupListEvent](#)”。

传出参数:

agent\_group\_list              座席组列表.其中 key 是组编号, value 是座席组名称

◆ **GetAllAgentGroupListEvent (Dictionary<string, string> agent\_group\_list)**

功能说明:

获得所有座席组列表事件。对应 javascript 的事件为 “[JSGetAllAgentGroupListEvent](#)”。

传出参数:

agent\_group\_list              座席组列表.其中 key 是组编号, value 是座席组名称

◆ **GetAgentsOfQueueEvent (string agent\_list,string queue\_num)**

功能说明:

获得某队列中的座席列表。

传出参数:

agent\_list                      座席列表, 包含一批座席工号, 以逗号隔开  
queue\_num                      队列号码

◆ **GetAgentsOfAgentGroupEvent (string agent\_list,string agentgroup\_num)**

功能说明:

获得某坐席组中的座席列表。

传出参数:

agent\_list                      座席列表, 包含一批座席工号, 以逗号隔开  
agentGroup\_num                座席组号码

◆ **GetAgentsMonitorInfoEvent(List<Agent\_Online\_Struct> agent\_monitor\_info , string current\_time, Dictionary<string, string> group\_list,Dictionary<string, string> role\_list)**

功能说明:

获得一批座席的基本监控信息。对应 javascript 的事件接口为

### [“JSGetAgentsMonitorInfoEvent”](#)

#### 传出参数:

agent_monitor_info	座席监控信息, 包括工号、姓名、状态等
current_time	当前时间
group_list	座席组列表
role_list	角色列表

### ◆ GetDetailCallInfoEvent (string targetAgentNum, string call\_type,List<Leg\_Info\_Struct> leg\_info, List<Relation\_Info\_Struct> relation\_info)

#### 功能说明:

获得某座席的详细通话信息。对应 javascript 的事件接口为 [“JSGetDetailCallInfoEvent”](#)

#### 传出参数:

targetAgentNum	要查看的座席工号
callType	呼叫类型(普通呼入、预测式外呼、内部呼叫)
leg_Info	每个电话 channel 信息
relation_info	关系列表。如: 监听、三方、询问等

### ◆ GetCustomerOfQueueEvent (string queueNumLstStr, string current\_time,List<Customer\_Info\_Struct> customer\_list)

#### 功能说明:

获得多个队列的客户排队监控信息。对应 javascript 的事件接口为 [“JSGetCustomerOfQueueEvent”](#)

#### 传出参数:

queueNumLstStr	多个队列号码, 以逗号隔开
current_time	当前时间
customer_list	客户排队信息。

### ◆ AddCustomerToQueueEvent (string call\_type, string callcenter\_num, string customer\_num, string customer\_status, string customer\_type, string customer\_uuid, string enter\_queue\_time, string exclusive\_agent\_num,string exclusive\_queue\_num, string queue\_num, string current\_time,string queue\_name, string queue\_customer\_amount, string early\_queue\_enter\_time,string early\_queue\_enter\_time\_all, string customer\_enter\_channel)

#### 功能说明:

客户进入排队事件。

#### 传出参数:

call_type	呼叫类型
callcenter_num	呼叫中心号码
customer_status	客户状态
customer_type	客户类型
customer_uuid	客户 uuid
enter_queue_time	客户进入队列时间
exclusive_agent_num	专属坐席号码



exclusive\_queue\_num 专属坐席所在的队列  
queue\_num 队列号码  
current\_time 当前时间  
queue\_name 队列名称  
queue\_customer\_amount 队列排队等待人数  
early\_queue\_enter\_time 本队列中客户最早进入队列的时间(时间戳格式)  
early\_queue\_enter\_time\_all 所有队列中客户最早进去队列的时间(时间戳格式)  
customer\_enter\_channel 客户进线通道

◆ **UpdateCustomerOfQueueEvent** (string call\_type, string callcenter\_num, string customer\_num, string customer\_status, string customer\_type, string customer\_uuid, string enter\_queue\_time, string exclusive\_agent\_num, string exclusive\_queue\_num, string queue\_num)

功能说明:

客户在队列中的状态发生改变事件。

传出参数:

call\_type 呼叫类型  
callcenter\_num 呼叫中心号码  
customer\_status 客户状态  
customer\_type 客户类型  
customer\_uuid 客户 uuid  
enter\_queue\_time 客户进入队列时间  
exclusive\_agent\_num 专属坐席号码  
exclusive\_queue\_num 专属坐席所在的队列  
queue\_num 队列号码  
current\_time 当前时间

◆ **DelCustomerFromQueueEvent** (string customer\_uuid, String queue\_num, String current\_time, String early\_queue\_enter\_time, String early\_queue\_enter\_time\_all String reason)

功能说明:

客户退出队列事件。

传出参数:

customer\_uuid 客户 uuid  
queue\_num 队列号码  
current\_time 服务器当前时间  
early\_queue\_enter\_time 本队列中客户最早进去队列的时间。  
early\_queue\_enter\_time\_all 所有队列中客户最早进去队列的时间。  
reason 退出原因。如下几种:  
1: 服务成功

- 2: 客户主动挂机
- 3: 客户排队主动按键离开
- 4: 超时离开
- 5: 超过最大人数

◆ **WarnAgentForceChangeStatusEvent** (string executorAgentID)

功能说明:

坐席状态被强制更改的通知事件。

传出参数:

executorAgentID 更改者或执行者的工号

◆ **SignInResponse** (string AgentID, int retCode, string Reason)

功能说明:

签入命令响应事件。

传出参数:

AgentID	座席号
retCode	命令是否成功 0 表示成功, 小于 0 则表示错误码。
Reason	错误原因

◆ **SignOutResponse** (string AgentID, int retCode, string Reason)

功能说明:

签出命令响应事件。

传出参数:

AgentID	座席号
BlnSuccess	命令是否成功
Reason	错误原因

◆ **SignInEvent** (string AgentID, bool blnSuccess, string Reason)

功能说明:

签入结果事件。

传出参数:

AgentID	座席号
BlnSuccess	命令是否成功
Reason	错误原因

◆ **SignOutEvent** (string AgentID, int retCode, string Reason)

功能说明:

签出结果事件。

传出参数:

AgentID	座席号
retCode	命令是否成功 0 表示成功, <0 代表错误码。
Reason	错误原因

◆ **KickOutEvent** (string AgentID, int retCode, string Reason)

功能说明:

被踢下线的事件。

传出参数:

AgentID	座席号
retCode	命令是否成功 0 表示成功, <0 代表错误码。
Reason	错误原因

◆ **SockDisconnectEvent (string Reason, int retCode)**

功能说明:

socket 断开连接的事件。

传出参数:

retCode	命令是否成功 0 表示成功, <0 代表错误码。
Reason	错误原因。

◆ **WarnAgentResigninEvent ()**

功能说明:

由于坐席信息更改而提醒坐席重新签入的事件。

◆ **SysThrowExceptionEvent (string reason,int retCode)**

功能说明:

系统异常事件。

传出参数:

reason	异常信息描述
retCode	错误码

◆ **GetWebSiteInfoEvent (string agentID, int retCode, string reason, Dictionary<string, string> web\_site\_info)**

功能说明:

获取报表地址事件。对应 javascript 的事件接口为“[JSGetWebSiteInfoEvent](#)”

传出参数:

agentID	坐席工号
retCode	0 成功 <0 表示错误码
reason	错误信息描述
web_site_info	网址字典

◆ **CheckExtenEvent (string AgentID)**

功能说明:

检测分机是否正常事件。

传出参数:

AgentID	座席号
---------	-----

◆ **AgentStatusChangeEvent ( string agent\_num,string old\_status,string new\_status,bool is\_bind\_exten,string customer\_enter\_channel)**

功能说明:

坐席状态更改事件。

### 传出参数:

Agent\_num 座席工号  
 Old\_status 更改之前的状态  
 New\_status 更改后的状态  
 Is\_bind\_exten 是否绑定了分机,0:未绑定 1: 绑定  
 customer\_enter\_channel 客户进线通道

### ◆ GetQueueStatisLstEvent(string queueNumLstStr,string current\_time,string List<Queue\_Statis\_Struct> queue\_statis\_list)

#### 功能说明:

获取队列监控统计事件。对应 javascript 事件接口为“[JSGetQueueStatisLstEvent](#)”

#### 传出参数:

queueNumLstStr 队列号码，多个用逗号隔开  
 current\_time 服务器当前时间  
 queue\_statis\_list 队列统计信息 list 结构体

```
public struct Queue_Statis_Struct
{
    public string agent_busy_amount; //忙碌座席数
    public string agent_callout_amount; //呼出坐席数
    public string agent_calling_amount; //呼出中坐席数
    public string agent_hold_amount; //保持坐席数
    public string agent_mute_amount; //静音坐席数
    public string agent_idle_amount; //空闲坐席数
    public string agent_leave_amount; //离开坐席数
    public string agent_no_exten_amount; //未绑定分机的座席数
    public string agent_offline_amount; //不在线坐席数
    public string agent_online_amount; //在线坐席数
    public string agent_post_treatment; //后处理座席数
    public string agent_ring_amount; //响铃座席数
    public string agent_talking_amount; //通话座席数
    public string agentNumLstStr; //坐席列表字符串
    public string early_queue_enter_time; //本队列中最早进入队列时间
    public string early_queue_enter_time_all; //所有队列中最早进入队列时间
    public string min_enter_queue_time; //未用
    public string queue_name; //队列名称
    public string queue_num; //队列号码
    public string queue_wait_people_amount; //队列中等待的人数
}
```

### ◆ GetAllQueueStatisEvent(string current\_time,string List<Queue\_Statis\_Struct> queue\_statis\_list)

#### 功能说明:

获取所有队列监控统计事件。对应 javascript 事件接口为“[JSGetAllQueueStatisEvent](#)”

**传出参数:**

current\_time      服务器当前时间  
queue\_statist\_list   队列统计信息 list 结构体

**◆ GetCustomerOfMyQueueEvent (string queueNumLstStr,string current\_time,string List<Customer\_Info\_Struct> customer\_list)****功能说明:**

获取本坐席所在队列中等待的客户明细事件。所有等待的客户信息都放在 customer\_list 对象中。对应 javascript 的事件接口为“[JSGetCustomerOfMyQueueEvent](#)”。

**传出参数:**

queueNumLstStr   本坐席所在队列号码，多个用逗号隔开  
current\_time      服务器当前时间  
customer\_list      队列中等待的客户信息 list 结构体详见结构体说明。  
其中可用字段:

```
call_type;                //呼叫类型  
callcenter_num;          //接入号  
customer_num;            //客户号码  
customer_status;         //客户状态  
customer_type;           //客户类型  
customer_uuid;           //客户 uuid  
enter_queue_time;        //进队列时间  
queue_num;               //队列号码  
customer_enter_channel;   //客户进线通道
```

**◆ GetIvrProfileListEvent (Dictionary<string, string> ivr\_profile\_list)****功能说明:**

获取 IVR Profile 列表事件。

**传出参数:**

ivr\_profile\_list   ivr profile 列表，key: ivr profile 号码，value:ivr profile 名称

**◆ GetReportStatisInfoEvent(string AgentID,int retCode,string reason, Dictionary<string,string> reportStatisInfo)****功能说明:**

获取报表统计事件。此数据属于报表统计数据，并不是实时的，目前 15 分钟更新一次。主要用于大屏幕监控等。对应 javascript 的事件接口为“[JSGetReportStatisInfoEvent](#)”

**传出参数:**

AgentID, 工号  
retCode, 错误码  
reason, 错误描述  
reportStatisInfo, 报表统计信息，  
目前含以下信息（可扩展）：  
todayAbandon, 当天放弃量  
todayCallin, 当天呼入量  
todayCallinAnswer, 当天进入队列并被坐席接听的电话量  
todayCallout, 当天呼出量

todayCalloutAnswer, 当天呼出接通量

todayEnterQueue, 当天直接进入队列的数量

todayEnterQueueOther, 当天非直接(即通过转入)进入队列的数量

◆ **ApplyChangeStatusEvent (string agentID, int retCode, string reason)**

功能说明:

申请离开事件。

传出参数:

agentID , 工号

retCode , 返回码

reason , 错误原因

◆ **ApplyChangeStatusCancelEvent(string agentID, string apply\_agentid, string targetStatus, int retCode, string reason)**

功能说明:

取消申请离开事件。

传出参数:

agentID , 工号

apply\_agentid , 申请者工号

targetStatus , 目标状态, 7:离开

retCode , 返回码

reason , 错误原因

◆ **ApplyChangeStatusDistributeEvent(string AgentID, string apply\_agentid, string targetStatus, string apply\_agentName, string apply\_agent\_groupID, string apply\_agent\_groupName, string apply\_time, string applyType, int retCode, string reason)**

功能说明:

申请离开审批者事件, 会通知给所有有此工号审批权限的在线审批者, 也会通知申请者自己。

传出参数:

agentID , 工号

apply\_agentid , 申请者工号

targetStatus , 目标状态, 7:离开

apply\_agentName , 申请者姓名

apply\_agent\_groupID , 申请者坐席组 id

apply\_agent\_groupName , 申请者坐席组名称

apply\_time , 申请时间

applyType , 申请类型 1 坐席 2 系统

retCode , 返回码

reason , 错误原因

◆ **ApproveChangeStatusDistributeEvent(string AgentID, string apply\_agentid, string apply\_agent\_groupID, string targetStatus, string approveResult, string approve\_time, int retCode, string reason)**

#### 功能说明:

申请离开审批事件, 会通知给所有有此工号审批权限的在线审批者, 也会通知申请者自己。

#### 传出参数:

agentID , 工号  
 apply\_agentid , 申请者工号  
 targetStatus , 目标状态, 7:离开  
 approveResult , 审批结果。1: 通过 0: 不通过  
 approve\_time , 审批时间  
 retCode , 返回码  
 reason , 错误原因

#### ◆ ApproveChangeStatusTimeoutDistributeEvent (string AgentID, string apply\_agentid, string apply\_agent\_groupID, string targetStatus, string timeoutType)

#### 功能说明:

申请或审批超时事件, 会通知给所有有此工号审批权限的在线审批者, 也会通知申请者自己。

#### 传出参数:

agentID , 工号  
 apply\_agentid , 申请者工号  
 targetStatus , 目标状态, 7:离开  
 apply\_agent\_groupid , 审批结果。1: 通过 0: 不通过  
 timeoutType , 1 申请超时 2 审批超时

#### ◆ GetAgentGroupStatusMaxNumEvent (string AgentID, string agentGroupNameLstStr, Dictionary<string, string> dicAgentGroupStatusMaxNum)

#### 功能说明:

获取坐席组离开人数阈值。对应 javascript 的接口事件为 [“JSGetAgentGroupStatusMaxNumEvent”](#)

#### 传出参数:

agentID , 工号  
 agentGroupNameLstStr , 坐席组名称列表, 多个用逗号隔开。  
 dicAgentGroupStatusMaxNum, 各个坐席组的离开人数最大阈值字典。

#### ◆ GetChangeStatusApplyListEvent (string agentID, List<Apply\_Change\_Status> apply\_agent\_list, int retCode, string reason)

#### 功能说明:

获取所有需要本坐席审批或允许本坐席查看的坐席离开申请记录列表。对应 javascript 接口事件为 [“JSGetChangeStatusApplyListEvent”](#)

#### 传出参数:

agentID , 工号  
 apply\_agent\_list , 坐席离开申请记录列表。  
 retCode , 返回码  
 reason , 错误原因

**◆ AllGetwaysFullEvent()**

功能说明:

呼叫外线时线路满线事件。

传出参数:

无。

**◆ NoAnswerCallAlarmEvent ()**

功能说明:

未接来电提示事件。

传出参数:

无。

**◆ QueueTransferBoundEvent(string agentID, string customer\_num, string queue\_num, string transfee\_num, string access\_num, string tranfer\_time)**

功能说明:

客户电话队列转外线事件。

传出参数:

agentID: 坐席 ID

customer\_num: 客户电话号码

queue\_num: 客户由哪个队列转入外线

transfee\_num: 转接的号码

access\_num: 客户呼入的接入号

tranfer\_time: 转接的时间

**◆ RecordStart(string AgentExten, string AgentID, string Agent\_call\_uuid, string CalleeNum, string CallerNum, string FilePath)**

功能说明:

录音开始事件。

传出参数:

**AgentExten:** 坐席分机

**AgentID:** 坐席工号

**Agent\_call\_uuid:** 坐席 uuid

**CalleeNum:** 被叫号码

**CallerNum:** 主叫号码

**FilePath:** 录音文件存储路径

**◆ RecordStop(string AgentID, string Agent\_call\_uuid, , string FilePath)**

功能说明:

录音开始事件。

传出参数:

**AgentID:** 坐席工号

**Agent\_call\_uuid:** 坐席 uuid



**FilePath:** 录音文件存储路径

### 3.2.7 针对 Javascript 接收事件的调整说明

如果使用的是 Javascript 来接收事件，由于事件中的参数类型有些是字典、列表等类型，而这些在 javascript 里是没有的，故对于此类事件则不能使用上述接口，而要使用其他的接口，我们会将这些新增的接口名称前统一命名为：JS + 旧接口名称，并且参数为 json 格式的字符串，格式统一如下：

```
{
    "第一个参数名称": "[{结构体 1}, {结构体 2}, {结构体 3}],
    "第二个参数名称": "{ \"key1\": \"value1\", \"key2\": \"value2\", \"key3\": \"value3\" }",
    "第三个参数名称": "13800000000"
}
```

举例如下：

有个学生信息的结构体如下：

Public Struct stu

```
{
    String name;
    String sex;
    String age;
}
```

如果是 C# 系统则可以调用的事件接口如下：

getAllStudents(list< stu > m\_stu, string agent\_num, string current\_time)

如果是 javascript 则调用的接口则是：

JSgetAllStudents(string strArgsJson)

其中 strArgsJson 内容格式如下：

```
{
    "m_stu": [{ "name": "张三", "sex": "男", "age": "20"},
               { "name": "李四", "sex": "女", "age": "18"},
               { "name": "王五", "sex": "男", "age": "24"} ],
    "agent_num": "1001",
    "current_time": "2015-9-24 17:23:00"
}
```

针对 Javascript 新增加的事件接口有如下几个：

#### ◆ JSGetAgentOnlineEvent(string jsonArgsStr)

功能说明：

获取在线座席的事件。

传出参数：

jsonArgsStr ， json 格式的字符串，其中内容是将接口“[GetAgentOnlineEvent](#)”里的所有参数转换为一个 json 格式的字符串。

#### ◆ JSGetAccessNumberEvent(string jsonArgsStr)

功能说明：

获取所有接入号列表和外呼历史的事件。

传出参数:

jsonArgsStr , json 格式的字符串, 其中内容是将接口“[GetAgentOnlineEvent](#)”里的所有参数转换为一个 json 格式的字符串。

◆ **JSGetIvrListEvent(string jsonArgsStr)**

功能说明:

获取所有 ivr 列表的事件。

传出参数:

jsonArgsStr , json 格式的字符串, 其中内容是将接口“[GetIvrListEvent](#)”里的所有参数转换为一个 json 格式的字符串。

◆ **JSGetQueueListEvent(string jsonArgsStr)**

功能说明:

获取所有 ivr 列表的事件。

传出参数:

jsonArgsStr , json 格式的字符串, 其中内容是将接口“[GetQueueListEvent](#)”里的所有参数转换为一个 json 格式的字符串。

◆ **JSGetAgentGroupListEvent(string jsonArgsStr)**

功能说明:

获取所有 ivr 列表的事件。

传出参数:

jsonArgsStr , json 格式的字符串, 其中内容是将接口“[GetAgentGroupListEvent](#)”里的所有参数转换为一个 json 格式的字符串。

◆ **JSGetAllAgentGroupListEvent(string jsonArgsStr)**

功能说明:

获取所有 ivr 列表的事件。

传出参数:

jsonArgsStr , json 格式的字符串, 其中内容是将接口“[GetAllAgentGroupListEvent](#)”里的所有参数转换为一个 json 格式的字符串。

◆ **JSGetAgentsMonitorInfoEvent(string jsonArgsStr)**

功能说明:

获取一批坐席的基本监控信息事件。

传出参数:

jsonArgsStr , json 格式的字符串, 其中内容是将接口“[GetAgentsMonitorInfoEvent](#)”里的所有参数转换为一个 json 格式的字符串。

◆ **JSGetDetailCallInfoEvent (string jsonArgsStr)**

功能说明:

获取一批坐席的基本监控信息事件。

传出参数:

jsonArgsStr ， json 格式的字符串，其中内容是将接口“[GetDetailCallInfoEvent](#)”里的所有参数转换为一个 json 格式的字符串。

◆ **JSGetCustomerOfQueueEvent (string jsonArgsStr)**

功能说明：

获得多个队列的客户排队监控信息。

传出参数：

jsonArgsStr ， json 格式的字符串，其中内容是将接口“[GetCustomerOfQueueEvent](#)”里的所有参数转换为一个 json 格式的字符串。

◆ **JSGetCustomerOfMyQueueEvent (string jsonArgsStr)**

功能说明：

获取本坐席所在队列中等待的客户明细事件。

传出参数：

jsonArgsStr ， json 格式的字符串，其中内容是将接口“[GetCustomerOfMyQueueEvent](#)”里的所有参数转换为一个 json 格式的字符串。

◆ **JSGetQueueStatisLstEvent (string jsonArgsStr)**

功能说明：

获取队列监控统计事件。

传出参数：

jsonArgsStr ， json 格式的字符串，其中内容是将接口“[GetQueueStatisLstEvent](#)”里的所有参数转换为一个 json 格式的字符串。

◆ **JSGetAllQueueStatisEvent (string jsonArgsStr)**

功能说明：

获取所有队列监控统计事件。

传出参数：

jsonArgsStr ， json 格式的字符串，其中内容是将接口“[GetAllQueueStatisEvent](#)”里的所有参数转换为一个 json 格式的字符串。

◆ **JSGetWebSiteInfoEvent (string jsonArgsStr)**

功能说明：

获取报表地址事件。

传出参数：

jsonArgsStr ， json 格式的字符串，其中内容是将接口“[GetWebSiteInfoEvent](#)”里的所有参数转换为一个 json 格式的字符串。

◆ **JSGetReportStatisInfoEvent (string jsonArgsStr)**

功能说明：

获取报表统计事件。此数据属于报表统计数据，并不是实时的，目前 15 分钟更新一次。主要用于大屏幕监控等。

传出参数：

jsonArgsStr ， json 格式的字符串，其中内容是将接口“[GetReportStatisInfoEvent](#)”里的所有参数转换为一个 json 格式的字符串。

#### ◆ JSGetAgentGroupStatusMaxNumEvent (string jsonArgsStr)

功能说明:

获取坐席组离开人数阈值。

传出参数:

jsonArgsStr , json 格式的字符串, 其中内容是将接口 [“GetAgentGroupStatusMaxNumEvent”](#) 里的所有参数转换为一个 json 格式的字符串。

#### ◆ JSGetChangeStatusApplyListEvent (string jsonArgsStr)

功能说明:

取所有需要本坐席审批或允许本坐席查看的坐席离开申请记录列表。

传出参数:

jsonArgsStr , json 格式的字符串, 其中内容是将接口 [“GetChangeStatusApplyListEvent”](#) 里的所有参数转换为一个 json 格式的字符串。

### 3.2.8 错误码

指的是以上 retCode 的值及其解释。

```
<add key="-1" value="XML 解析错误" />
<add key="-2" value="XML 根节点的 tag 名称不是 “request” ” />
<add key="-3" value="cmdType 字段为空" />
<add key="-4" value="agentID 字段为空" />
<add key="-5" value="cmdType 类型未知" />
<add key="-6" value="请不要重复签入" />
<add key="-7" value="工号与登陆的 socket 号不相符" />
<add key="-8" value="agentPwd 字段为空" />
<add key="-9" value="bindExten 字段不为 yes 或 no" />
<add key="-10" value="bindExten 字段为 yes 但 agentExten 字段为空" />
<add key="-11" value="分机状态错误,请检查该分机是否注册在线" />
<add key="-12" value="分机已被其他座席绑定" />
<add key="-13" value="登陆初始座席状态为空" />
<add key="-14" value="登陆初始座席状态不合法" />
<add key="-15" value="目标座席状态为空" />
<add key="-16" value="目标座席状态不合法" />
<add key="-17" value="不能手动切换至目标座席状态" />
<add key="-18" value="customerNum 字段为空" />
<add key="-19" value="customerDisplayNum 字段为空" />
<add key="-20" value="查找座席角色失败" />
<add key="-21" value="权限检查失败,请确认您是否有该权限" />
<add key="-22" value="customerDisplayNum 字段不合法" />
<add key="-23" value="没有找到座席, 可能没有签入" />
<add key="-24" value="座席状态错误" />
<add key="-25" value="座席没有绑定分机" />
<add key="-26" value="绑定的分机状态错误" />
```

```
<add key="-27" value="目标座席状态错误" />
<add key="-28" value="主叫和被叫相同" />
<add key="-29" value="未找到主叫座席，可能没有签入" />
<add key="-30" value="未找到被叫座席，可能没有签入" />
<add key="-31" value="主叫座席状态错误" />
<add key="-32" value="被叫座席状态错误" />
<add key="-33" value="主叫座席未绑定分机" />
<add key="-34" value="被叫座席未绑定分机" />
<add key="-35" value="主叫座席绑定的分机状态错误" />
<add key="-36" value="被叫座席绑定的分机状态错误" />
<add key="-37" value="language 字段不是 zh 或 en" />
<add key="-38" value="查找座席所在座席组失败" />
<add key="-39" value="IVR 号码为空" />
<add key="-40" value="IVR 号码非法" />
<add key="-41" value="队列号码为空" />
<add key="-42" value="队列号码非法" />
<add key="-43" value="后台查询数据库失败" />
<add key="-44" value="座席组号码为空" />
<add key="-45" value="座席组号码列表字段为空" />
<add key="-46" value="座席组列表为空" />
<add key="-47" value="队列管理模块指针为空" />
<add key="-48" value="查找座席所属队列列表失败" />
<add key="-49" value="未找到目标座席，可能没有签入" />
<add key="-50" value="目标座席座席状态错误" />
<add key="-51" value="该座席没有注册" />
<add key="-52" value="该座席还未启用" />
<add key="-53" value="登陆密码错误" />
<add key="-54" value="使座席下线失败" />
<add key="-55" value="初始化座席相关参数失败" />
<add key="-56" value="锁分机失败" />
<add key="-57" value="手动呼出失败" />
<add key="-58" value="改变座席状态失败" />
<add key="-59" value="输入参数错误" />
<add key="-60" value="电话管理模块操作失败" />
<add key="-61" value="坐席组号码不存在" />
<add key="-62" value="获得自定义网址失败" />
<add key="-63" value="IVR Profile 号码为空" />
<add key="-64" value="IVR Profile 号码非法" />
<add key="-65" value="是否是外线号码标识为空" />
<add key="-66" value="忙转的外线号码为空" />
<add key="-67" value="对方坐席状态无法被强制改变" />
<add key="-68" value="此坐席不属于任何坐席组！" />
<add key="-69" value="已申请的数量已达到允许的最大阈值，请稍后再试！" />
<add key="-70" value="您已经申请过，请不要重复申请！" />
```

```
<add key="-71" value="审批者目前不在线，请稍后再试！" />
<add key="-72" value="未找到您所在的坐席组！" />
<add key="-73" value="申请者的工号为空！" />
<add key="-74" value="审批结果不能为空！" />
<add key="-75" value="此审批已经被处理！" />
<add key="-76" value="此申请的处理状态不正确！" />
<add key="-77" value="坐席组名称不能为空！" />
<add key="-78" value="坐席组离开阈值无效！" />
<add key="-79" value="设置坐席组离开阈值失败！" />
<add key="-80" value="此申请或审批超时,操作无效！" />
<add key="-81" value="坐席电话 uud 为空！" />
<add key="-82" value="客户电话 uuid 为空！" />
<add key="-83" value="web 服务调用失败！" />
<add key="-84" value="旧密码错误！" />
<add key="-1000" value="客户端接收信息线程异常退出" />
<add key="-1001" value="与网络断开连接！" />
<add key="-1002" value="网络异常错误！" />
<add key="-2000" value="系统异常错误！" />
<add key="-2001" value="系统异常错误！" />
```