
文件标识	
版本号	
密 级	

Agentbar 开发手册

编制		日期	
审核		日期	
批准		日期	

1 引言.....	3
2 开发简介.....	3
2.1 版本介绍.....	3
2.2 开发模式.....	3
2.3 示例.....	3
3 接口.....	5
3.1 属性.....	5
3.2 方法.....	7
3.3 事件.....	37
4 特殊应用场景编程介绍.....	59
4.1 Agentbar 发起外呼如何防骚扰.....	59
4.2 前端页面何时写话单.....	59

1 引言

Agentbar 是呼叫云的坐席客户端组件，它符合标准的 activex 接口，它是目前呼叫云唯一对外开放的接口，所有的二次开发都基于此接口。

2 开发简介

2.1 版本介绍

Agentbar 目前为 1.0 版本，四位版本号为 1.0.7.3。Agentbar 支持证书和自动下载安装。

2.2 开发模式

Agentbar 有两种开发模式，有界面模式和无界面模式。

1、有界面模式

有界面模式只需要把控件的高位设为 90，宽设为>800 即可，将会显示如下界面。



2、无界面模式

无界面模式只需要把控件的高位设为 0，宽 0 即可。

两种模式下共用一套接口，实现客户端与服务端的交互。

2.3 示例

下面是一个有界面模式的简单例子。

```
<script>

function Init()
{
    var ret = agentBar.Initial();
}
```

```
function UnInit()
{
    var ret = agentBar.UnInitial();
}
</script>
```

```
<OBJECT CLASSID="clsid:6D4733B4-8E49-495C-9F2B-27606CEB20E3" id="agent Bar"
codebase="CCAgentBar.cab#version=1,0,7,3" width=100% height = 90>
```

```
<param name=AgentID value="2002" />
<param name=Pass Word value="2002123" />
<param name=DN value="1889" />
<param name=StatusChangeType value=1 />
<param name=Auto Answer value=true />
<param name=ForceSignIn value=true />
<param name=ServiceList value="示例" />
<param name=MainCTIIP value="127.0.0.1" />
<param name=BackCTIIP value="127.0.0.1" />
<param name=CTIPort value=10000 />
<param name=TimeOut value=3000 />
<param name=IsInitial value=true />
<param name=IsSignIn value=true />
<param name=IsSIPPhone value=false />
<param name=SIPServerIP value="127.0.0.1" />
<param name=SIPServerPort value=5600 />
<param name=PhoneNum value="1998" />
<param name=PhonePwd value="1234" />
<param name=AgentTag value="" />
<param name=CustomerTag value="agt" />
<param name=ANICChangeType value=1 />
<param name=IsShowCallOut value=true />
<param name=IsCallOut value=true />
<param name=LogLevel value=3 />
<param name=EnableSeniorCall value=true />
```

</OBJECT>

<input id = "btnInit" type=button value="初始化" onclick="Init();" />

<input id = "btnUnInit" type=button value="反初始化" onclick="UnInit();" />

3 接口

3.1 属性

Agentbar 的属性较多，一般情况下，在开发中需要设置如下属性，其他属性使用默认值就可以了。

设置 CTI 服务器地址和端口 (MainCTIIP, CTIPort)，设置坐席基本信息 (AgentID, PassWord, DN, ServiceList)，如果是外呼型呼叫中心，将 StatusChangeType 设为 1，方便坐席操作。

属性	类型	设置&获取属性接口	说明
AgentID	string	获取属性: BSTR GetAgentID(void); 设置属性: void SetAgentID(LPCTSTR newVal);	坐席工号【string 类型均为长度限制，下同】
PassWord	string	获取属性: BSTR GetPassWord(void); 设置属性: void SetPassWord(LPCTSTR newVal);	密码
DN	string	获取属性: BSTR GetDN(void); 设置属性: void SetDN(LPCTSTR newVal);	坐席电话号码
StatusChangeType	int	获取属性: LONG GetStatusChangeType(void); 设置属性: void SetStatusChangeType(LONG newVal);	坐席挂机后的状态，0-接听就绪(默认)，1-外呼就绪
AutoAnswer	bool	获取属性: LONG GetAutoAnswer(void); 设置属性: void SetAutoAnswer(LONG newVal);	是否自动应答，true-是(默认)，false-否
ForceSignIn	bool	获取属性: LONG GetForceSignIn(void); 设置属性: void SetForceSignIn(LONG newVal);	坐席在别处已登录，是否提出后强制登陆，true-是(默认)，false-否
ServiceList	string	获取属性: BSTR GetServiceList(void); 设置属性: void SetServiceList(LPCTSTR newVal);	技能队列组合，以,分隔，如“投诉=3,咨询=1”
MainCTIIP	string	获取属性: BSTR GetMainCTIIP(void);	主服务器的 IP 地址

		设置属性: void SetMainCTIIP(LPCTSTR newVal);	
BackCTIIP	string	获取属性: BSTR GetBackCTIIP(void); 设置属性: void SetBackCTIIP(LPCTSTR newVal);	备服务器的 IP 地址 如果没有备用服务器, 则设置为空字符串, 缺省为空字符串
CTIPort	int	获取属性: LONG GetCTIPort(void); 设置属性: void SetCTIPort(LONG newVal);	服务器端口, 默认 10000
TimeOut	int	获取属性: LONG GetTimeOut(void); 设置属性: void SetTimeOut(LONG newVal);	API 函数调用超时, 默认 3000
IsInitial	bool	获取属性: LONG GetIsInitial(void); 设置属性: void SetIsInitial(LONG newVal);	是否初始化, 只读, true-是, false-否
IsSignIn	bool	获取属性: LONG GetIsSignIn(void); 设置属性: void SetIsSignIn(LONG newVal);	是否登录, 只读, true-是, false-否
IsSIPPhone	bool	获取属性: LONG GetIsSIPPhone(void); 设置属性: void SetIsSIPPhone(LONG newVal);	是否集成软电话, true-是, false-否 (默认)
SIPServerIP	string	获取属性: BSTR GetSIPServerIP(void); 设置属性: void SetSIPServerIP(LPCTSTR newVal);	SIP 服务器地址
SIPServerPort	int	获取属性: LONG GetSIPServerPort(void); 设置属性: void SetSIPServerPort(LONG newVal);	SIP 服务器端口, 默认 5060
PhoneNum	string	获取属性: BSTR GetPhoneNum(void); 设置属性: void SetPhoneNum(LPCTSTR newVal);	SIP 终端的账号
PhonePwd	string	获取属性: BSTR GetPhonePwd(void); 设置属性: void SetPhonePwd(LPCTSTR newVal);	SIP 终端的密码
AgentTag	string	获取属性: BSTR GetAgentTag(void); 设置属性: void SetAgentTag(LPCTSTR newVal);	呼坐席的出局码
CustomerTag	string	获取属性: BSTR GetCustomerTag(void); 设置属性: void SetCustomerTag(LPCTSTR newVal);	呼客户的出局码
ANICChangeType	int	获取属性: LONG GetANICChangeType(void); 设置属性: void SetANICChangeType(LONG newVal);	未使用
IsShowCallOut	bool	获取属性: LONG GetIsShowCallOut(void); 设置属性: void SetIsShowCallOut(LONG newVal);	允许显示外呼按钮, true-是(默认), false-否

IsCallOut	bool	获取属性: LONG GetIsCallOut(void); 设置属性: void SetIsCallOut(LONG newVal);	是否不拦截外呼, true-是(默认), false-否
LogLevel	int	获取属性: LONG GetLogLevel(void); 设置属性: void SetLogLevel(LONG newVal);	日志级别, 0-错误, 1-警告, 2-轨迹(默认), 3-调试
EnableSeniorCall	bool	获取属性: LONG GetEnableSeniorCall(void); 设置属性: void SetEnableSeniorCall(LONG newVal);	是否显示高级呼叫控制按钮, true-是(默认), false-否
AnyProperty	string	获取属性: BSTR GetAnyProperty(void); 设置属性: void SetAnyProperty(LPCTSTR newVal);	预留属性

3.2 方法

Agentbar 的方法较多, 但一般我们采用有界面的开发模式, 可以大大简化开发难度, 您只需要使用到 Initial、UnInitial、MakeCall 和部分监控质检等方法, 其他方法在使用时再查阅。

【状态设置方法】

3.2.1 Initial(初始化)

【功能】

坐席与 CTI 建初始化连接

【函数说明】

```
int Initial()
```

【参数说明】

无

【返回值】

Agentbar 返回错误值列表:

```
enum AGENTBARERROR
{
    AGENTBARERROR_SUCCESS,
    AGENTBARERROR_BGCC_CLIENTPROXYFAIL = 1000,
    AGENTBARERROR_BGCC_EXCEPTION,
    AGENTBARERROR_BAR_INITIAL,
    AGENTBARERROR_BAR_UNINITIAL,
```

```

AGENTBARERROR_BAR_UNSIGNIN,
AGENTBARERROR_BAR_WRONGSTATE,
AGENTBARERROR_BAR_OUTEVENTERROR,
AGENTBARERROR_BAR_ANSWERERROR,
AGENTBARERROR_JSON_READERROR,
AGENTBARERROR_SOFTPHONE_ERROR
};

```

ACD同时也会返回错误值，如下所示：

```

enum AcarResultT
{
    ArSuccess = 0,                // 接口调用成功
    ArAgentIdIsEmpty,             // 坐席工号为空
    ArAgentDnIsEmpty,             // 坐席分机为空
    ArSkillIsEmpty,               // 坐席技能为空
    ArCallerIdIsEmpty,            // 路由标识为空
    ArDestIdIsEmpty,              // 目标坐席工号为空
    ArDestDnIsEmpty,              // 目标坐席分机为空
    ArKeyIsEmpty,                 // 随路数据键为空
    ArValueIsEmpty,               // 随路数据值为空
    ArDigitalsIsEmpty,            // 二次拨号值为空
    ArFilenameIsEmpty,            // 文件名为空
    ArAgentIdNotExist,            // 坐席工号不存在
    ArAgentPasswordError,         // 坐席密码错误
    ArAgentSkillError,            // 坐席技能错误
    ArNotMaster,                  // 非主机
    ArAgentDnBeOccupied,          // 坐席分机被占用
    ArHasSigned,                  // 已签入
    ArNoSignin,                   // 未签入
    ArDestAgentIdNotExist,        // 目标坐席工号不存在
    ArDestAgentDnNotExist,        // 目标坐席分机不存在
    ArSkillNotExist,              // 技能不存在
    ArRouteRequestNotExist,       // 路由请求不存在
    ArAgentStatusIsWrong,         // 坐席状态错误
    ArImsError,                   // ims错误
    ArPasswordError,              // 密码错误
    ArFailed,                      // 其他错误
}

```

下同

【说明】

在初始化之前，必须设置如下属性：

MainCTIIP	string	主服务器的 IP 地址
-----------	--------	-------------

BackCTIIP	string	备服务器的 IP 地址 如果没有备用服务器，则设置为空字符串，缺省为空字符串
CTIPort	int	服务器端口，默认 10000
IsSIPPhone	bool	是否集成软电话，true-是，false-否(默认)
LogLevel	int	日志级别，0-错误，1-警告，2-轨迹(默认)，3-调试

Agentbar 自定义返回值说明：

表 3.2.1.2 返回值列表

枚举	值	出现情况
AGENTBARERROR_SUCCESS	0	主服务器的 IP 地址，返回此错误码；
AGENTBARERROR_BGCC_CLI ENTPROXYFAIL	1001	Agentbar Proxy 为空，返回此错误码；
AGENTBARERROR_BGCC_EXC EPTION	1002	调用方法时捕获异常时，返回此错误码；
AGENTBARERROR_BAR_INIT IAL	1003	在 Agentbar 已经初始化情况下，进行初始化操作，返回此错误码；
AGENTBARERROR_BAR_UNIN ITIAL	1004	在 Agentbar 未初始化的情况下，调用 Agentbar 反初始化方法，返回此错误码；
AGENTBARERROR_BAR_UNSI GNIN	1005	在 Agentbar 未签入的情况下，调用 Agentbar 方法时，返回此错误码；
AGENTBARERROR_BAR_WRON GSTATE	1006	出现该错误码情况较多，出现问题时可根据错误码及操作排查代码；
AGENTBARERROR_BAR_OUTE VENTERROR	1007	在解析、推送事件时捕获异常；
AGENTBARERROR_BAR_ANSW ERERROR	1008	没有集成软电话进行应答时；
AGENTBARERROR_JSON_REA DERROR	1009	在以下情况下返回此错误码： 1. 调用 GetAgentDetailsByAgentIDs 时； 2. 调用 GetAgentDetailsByAgentDNs 时；
AGENTBARERROR_SOFTPHON E_ERROR	1010	在以下情况下返回此错误码：（均为集成软电话的情况下） 1. 调用 Initial 时，初始化软电话失败； 2. 调用 UnInitial 时，反初始化软电话失败；

		3. 调用 SignIn 时，注册软电话失败； 4. 调用 SignOut 时，反注册软电话失败； 5. 调用 Answer 时，软电话应答失败；
--	--	---

3.2.2 UnInitial(反初始化)

【功能】

坐席与 CTI 断开连接

【函数说明】

```
int UnInitial()
```

【参数说明】

无

【返回值】

0: 成功

其他值: 失败

3.2.3 SignIn(坐席签入)

【功能】

坐席签入 CTI 服务器。

【函数说明】

```
int SignIn()
```

【参数说明】

无

【返回值】

0: 成功

其他值: 失败

【说明】

在签入之前，必须设置如下属性：

AgentID	string	坐席工号【string 类型均为长度限制，下同】
PassWord	string	密码
DN	string	坐席电话号码

StatusChangeType	int	坐席挂机后的状态，0-接听就绪(默认)，1-外呼就绪
AutoAnswer	bool	是否自动应答，true-是(默认)，false-否
ForceSignIn	bool	坐席在别处已登录，是否提出后强制登陆，true-是(默认)，false-否
ServiceList	string	技能队列组合，以;分隔，如“投诉=3;咨询=1”
TimeOut	int	API 函数调用超时，默认 3000
SIPServerIP	string	SIP 服务器地址
SIPServerPort	int	SIP 服务器端口，默认 5060
PhoneNum	string	SIP 终端的账号
PhonePwd	string	SIP 终端的密码
AgentTag	string	呼坐席的出局码
CustomerTag	string	呼客户的出局码
ANICChangeType	int	未使用
IsShowCallOut	bool	允许显示外呼按钮，true-是(默认)，false-否
IsCallOut	bool	是否不拦截外呼，true-是(默认)，false-否
EnableSeniorCall	bool	是否显示高级呼叫控制按钮，true-是(默认)，false-否
AnyProperty	string	预留属性

3.2.4 SignOut(坐席签出)

【功能】

坐席签出 CTI 服务器。

【函数说明】

```
int SignOut()
```

【参数说明】

无

【返回值】

0: 成功

其他值: 失败

3.2.5 SetIdle(示闲)

【功能】

将坐席状态设置为空闲状态。此时坐席可接 CTI 分配的来话。

【函数说明】

```
int SetIdle()
```

【参数说明】

无

【返回值】

0: 成功

其他值: 失败

3.2.6 SetBusy(示忙)

【功能】

将坐席状态设置为忙状态。若设置成功，CTI 将不再给此坐席分配来话。

【函数说明】

```
int SetBusy()
```

【参数说明】

无

【返回值】

0: 成功

其他值: 失败

3.2.7 SetRest(休息)

【功能】

将坐席状态设置为休息状态。若设置成功，CTI 将不再给此坐席分配来话。

【函数说明】

```
int SetRest(string reason)
```

【参数说明】

无

【返回值】

0: 成功

其他值：失败

3.2.8 SetAgentStatus(设置坐席状态)

【功能】

设置坐席状态。

【函数说明】

```
int SetAgentStatus(int AgentState, string Reason)
```

【参数说明】

参数	参数说明
AgentState	坐席状态编码 enum AgentStatusT { AsUnknownState, // 未知状态 AsUnLoginState, // 未登录 AsReadyState, // 空闲 AsBusyState, // 忙 AsRestState, // 休息 AsLockState, // 预占 AsAlertingState, // 振铃状态 AsHalfAlertingState, // 硬电话振铃 AsConnectingState, // 对方振铃状态 AsConnectedState, // 通话状态 AsHalfConnectedState, // 硬电话接通 AsInternalConnectedState, // 内部通话状态 AsSuspendedState, // 保持状态 AsConsultConnectingState, // 咨询呼叫状态 AsConsultConnectedState, // 咨询通话状态 AsConferenceState, // 会议状态 AsMonitorState, // 监听状态 AsInsertState, // 强插状态 AsMonitoredState, // 被监听状态 AsInsertedState // 被强插状态

	}
Reason	休息原因，设置从接口 GetRestReason 取到的所有休息原因中的一个

【返回值】

0：成功

其他值：失败

3.2.9 GetAgentStatus(取坐席状态)

【功能】

获取坐席状态。

【函数说明】

int GetAgentStatus ()

【参数说明】

【返回值】

```
enum AgentStatusT
{
    AsUnknownState = 0,
    AsUnLoginState,
    AsReadyState,
    AsBusyState,
    AsRestState,
    AsLockState,
    AsAlertingState,
    AsHarfAlertingState,
    AsConnectingState,
    AsConnectedState,
    AsHarfConnectedState,
    AsInternalConnectedState,
    AsSuspendedState,
    AsConsultConnectingState,
    AsConsultConnectedState,
    AsConferenceState,
    AsMonitorState,
    AsInsertState,
    AsMonitoredState,
    AsInsertedState
};

1000 以上为失败。
```

3.2.10 ResetSkill(重设技能)

【功能】

班长重新置坐席的技能。

【函数说明】

```
int ResetSkill(string SkillList)
```

【参数说明】

参数	参数说明
SkillList	重新设定技能

【返回值】

0: 成功

其他值: 失败

3.2.11 Reset(恢复初始态)

【功能】

在坐席状态出现混乱或者需要恢复成正常上班状态。

【函数说明】

```
int Reset()
```

【参数说明】

无

【返回值】

0: 成功

其他值: 失败

【呼叫控制方法】

3.2.12 MakeCall(外呼)

【功能】

外呼电话。

【函数说明】

```
int MakeCall(string dest, string showANI, string showDest, int isOffhook, int type)
```

【参数说明】

参数	参数说明
dest	目标坐席工号或者外线号码
showANI	显示给坐席的号码
showDest	显示给外线的号码
isOffhook	是否常摘机，0：否，1：是
type	外呼类型，0：内线，1：外线

【返回值】

0：成功

其他值：失败

3.2.13 Answer(应答)

【功能】

应答呼叫。

【数说明】

```
int Answer()
```

【参数说明】

无

【返回值】

0：成功

其他值：失败

3.2.14 ReleaseCall(释放)

【功能】

释放呼叫可释放各种媒体类型的呼叫。

【函数说明】

```
int ReleaseCall()
```

【参数说明】

无

【返回值】

0：成功

其他值：失败

3.2.15 Hold(保持)

【功能】

该方法把坐席正在通话的普通语音呼叫保持(只对语音呼叫有效)。坐席保持呼叫后，客户听到等待音，坐席进入占用态，系统不分配新的呼叫给坐席，但席可以执行呼出操作。

【函数说明】

```
int Hold()
```

【参数说明】

无

【返回值】

0：成功

其他值：失败

3.2.16 Retrieve(取保持)

【功能】

坐席取回被保持的普通语音呼叫进行通话(只对普通语音呼叫有效)。

【函数说明】

```
int Retrieve()
```

【参数说明】

无

【返回值】

0：成功

其他值：失败

3.2.17 Consult(咨询)

【功能】

坐席向另一号码发起咨询呼叫。

【函数说明】

```
int Consult(string consultNum, string showANI, string showDest, int consultType)
```

【参数说明】

参数	参数说明
ConsultNum	目标号码，String，最长 32。
showANI	显示给坐席的号码
showDest	显示给外线的号码
ConsultType	咨询类别，Integer。0：内部号码【坐席工号】，1：外部号码。

【返回值】

0：成功

其他值：失败

3.2.18 Reconnect(咨询接回)

【功能】

坐席取回咨询情况下被保持的普通语音呼叫进行通话，并挂断坐席和咨询方的通话（只对普通语音呼叫有效）。

【函数说明】

```
int Reconnect()
```

【参数说明】

无

【返回值】

0：成功

其他值：失败

3.2.19 Transfer(咨询转移)

【功能】

咨询后将呼转出到指定设备。

【函数说明】

```
int Transfer()
```

【参数说明】

无

【返回值】

0：成功

其他值：失败

3.2.20 SingleStepTransfer(单步转移)

【功能】

将呼叫转到指定设备。

【数说明】

```
int SingleStepTransfer(string transferNum, string showANI, string showDest,
int transferStyle, int transferType)
```

【参数说明】

参数	参数说明
TransferNum	目标号码
showANI	显示给坐席的号码
showDest	显示给外线的号码
TransferStyle	转移方式。0: 透传方式, 1: 非透传方式
TransferType	转移类别。0: 坐席工号, 1: 外部号码, 2: IVR 接入码。

【返回值】

0: 成功

其他值: 失败

3.2.21 Conference(咨询会议)

【功能】

坐席把一个保持呼叫加入到当前正在通话的呼叫中, 形成三方通话。

【函数说明】

```
int Conference ()
```

【参数说明】

无

【返回值】

0: 成功

其他值: 失败

3.2.22 SetCallData(设置呼叫数据)

【功能】

设置呼叫数据，将一块与呼叫相关联起来, 支持多种媒体类型。

【函数说明】

```
int SetCallData(string key, string CallData)
```

【参数说明】

参数	参数说明
key	关键字
CallData	待设置的呼叫数据内容

【返回值】

0: 成功

其他值: 失败

3.2.23 GetCallData(获取呼叫数据)

【功能】

查询呼叫数据，坐席取回与呼叫相关联的数据。持多种媒体类型。

【函数说明】

```
string GetCallData(string key)
```

【参数说明】

参数	参数说明
key	关键字，String，最长 32。
CallData	呼叫数据内容，String，最大长度 1023 字节。

【返回值】

呼叫数据: 成功

空串: 失败

【质检方法】

3.2.24 ForceOut(强制签出)

【功能】

将坐席强制签出。强制签出成功后，坐席必须再签入才能进行正常来话受理。

【函数说明】

```
int ForceOut(string AgentID)
```

【参数说明】

参数	参数说明
AgentID	被强制签出座席的工号。

【返回值】

0：成功。

其他值：失败

3.2.25 ForceIdle(强制示闲)

【功能】

将坐席强制置闲。

【函数说明】

```
int ForceIdle(string AgentID)
```

【参数说明】

参数	参数说明
AgentID	被强制置闲座席的工号。

【返回值】

0：成功。

其他值：失败

3.2.26 ForceBusy(强制示忙)

【功能】

将坐席强制置忙。

【函数说明】

```
int ForceBusy(string AgentID)
```

【参数说明】

参数	参数说明
AgentID	被强制置忙座席的工号。

【返回值】

0：成功。

其他值：失败

3.2.27 Listen(监听)

【功能】

质检员监听定坐席的通话。坐席听不见质检员讲话，质检员能听见坐席和用户的讲话。监听过程中质检员不能与坐席讲话，如要讲话，需调用 Insert 方法。如果要停止监听，可调用 ReleaseCall 方法。

【函数说明】

```
int Listen(string AgentID, string showANI, string showDest)
```

【参数说明】

参数	参数说明
AgentID	被监听坐席的工号。
showANI	显示给坐席的号码
showDest	显示给外线的号码

【返回值】

0：成功

其他值：失败

3.2.28 StopListen(停止监听)

【功能】

质检员发起监听后停止监听，也可以调用 ReleaseCall 方法停止监听。

【函数说明】

```
int StopListen(string AgentID)
```

【参数说明】

参数	参数说明
AgentID	被监听坐席的工号。

【返回值】

0：成功

其他值：失败

3.2.29 Insert(强插)

【功能】

质检员插指定坐席的话路，可与坐席通话。如果要停止插通，可调用 ReleaseCall 方法。

【函数说明】

```
int Insert(string agentID, string showANI, string showDest)
```

【参数说明】

参数	参数说明
AgentID	被监听坐席的工号。
showANI	显示给坐席的号码
showDest	显示给外线的号码

【返回值】

0: 成功

其他值: 失败

3.2.30 StopInsert(停止强插)

【功能】

质检员发起强插后停止强插，也可以调用 ReleaseCall 方法停止强插。

【函数说明】

```
int StopInsert(string agentID)
```

【参数说明】

参数	参数说明
AgentID	被监听坐席的工号。

【返回值】

0: 成功

其他值: 失败

3.2.31 SwitchInsertorListen(监听强插切换)

【功能】

质检员发起强插后停止强插，也可以调用 ReleaseCall 方法停止强插。

【函数说明】

```
int SwitchInsertorListen(string agentID, string showANI, string showDest)
```

【参数说明】

参数	参数说明
----	------

AgentID	被监听坐席的工号。
showANI	显示给坐席的号码
showDest	显示给外线的号码

【返回值】

0: 成功

其他值: 失败

3.2.32 Break(强拆)

【功能】

强拆，挂断其他话路。

【函数说明】

```
int Break (string agentID)
```

【参数说明】

参数	参数说明
AgentID	被挂断方号码

【返回值】

0: 成功

其他值: 失败

【媒体方法】

3.2.33 SendDTMF(二次拨号)

【功能】

坐席进行次拨号。在坐席呼出时，如果需要再次拨号，可调用此函数发送需要的拨号号码。如果需要发送多次，则可拱调用此函数多次来送拨号信息。

【函数说明】

```
int SendDTMF (string DTMFDigits)
```

【参数说明】

参数	参数说明
DTMFDigits	二次拨号的电话号码，最大长度为 32， String。

【返回值】

0: 成功
其他值: 失败

3.2.34 MuteOn(静音)

【功能】

静音。

【函数声明】

```
int MuteOn()
```

【参数说明】

无

【返回值】

0: 成功
其他值: 失败

3.2.35 MuteOff(静音)

【功能】

静音结束。

【函数声明】

```
int MuteOff()
```

【参数说明】

无

【返回值】

0: 成功
其他值: 失败

【查询方法】

3.2.36 GetAgentDetailByAgentID(通过 ID 获取座席信息)

【功能】

根据坐席工号取坐席信息。

【函数说明】

```
string GetAgentDetailByAgentID(string AgentID)
```

【参数说明】

参数	参数说明
AgentID	坐席工号

【返回值】

成功：坐席信息

时间是秒，GetAgentDetailByAgentID、GetAgentDetailByAgentDN、GetAgentDetailsByAgentIDs、GetAgentDetailsByAgentDNs、GetAgentDetailsBySkill相同，其他返回都是微妙

json:

```
struct AgentInfoT
{
    std::string agentId;        // 坐席ID
    std::string agentDn;        // 坐席号码
    int64_t loginTime;          // 登陆时间
    int32_t currStatus;         // 坐席当前状态
    int64_t currStatusTime;     // 当前状态时间
    int64_t answerTime;         // 通话开始时间
    int64_t currTime;           // 当前时间
    std::string origCaller;     // 原始主叫
    std::string origCallee;     // 原始被叫
    int32_t privQueueNum;       // 私有队列数
    int32_t callTimes;          // 此次登录呼叫次数
    int32_t answerTimes;        // 此次登录通话次数
    int32_t freeTotalTime;      // 此次登录空闲总时长
    int32_t busyTotalTime;      // 此次登录忙总时长
    int32_t resetTotalTime;     // 此次登录休息总时长
    int32_t callTotalTime;      // 此次登录后呼叫总时长
    int32_t answerTotalTime;    // 此次登录后通话总时长
}
```

错误：空字符串

3.2.37 GetAgentDetailByAgentDN(通过 DN 获取座席信息)

【功能】

根据坐席分机号码取坐席工号。

【函数说明】

```
string GetAgentDetailByAgentDN(string AgentDN)
```

【参数说明】

参数	参数说明
----	------

AgentDN	坐席分机号码
---------	--------

【返回值】

成功：坐席信息

json:

```
struct AgentInfoT
{
    std::string agentId;        // 坐席ID
    std::string agentDn;        // 坐席号码
    int64_t loginTime;          // 登陆时间
    int32_t currStatus;         // 坐席当前状态
    int64_t currStatusTime;     // 当前状态时间
    int64_t answerTime;         // 通话开始时间
    int64_t currTime;           // 当前时间
    std::string origCaller;     // 原始主叫
    std::string origCallee;     // 原始被叫
    int32_t privQueueNum;       // 私有队列数
    int32_t callTimes;          // 此次登录呼叫次数
    int32_t answerTimes;        // 此次登录通话次数
    int32_t freeTotalTime;      // 此次登录空闲总时长
    int32_t busyTotalTime;      // 此次登录忙总时长
    int32_t resetTotalTime;     // 此次登录休息总时长
    int32_t callTotalTime;      // 此次登录后呼叫总时长
    int32_t answerTotalTime;    // 此次登录后通话总时长
}
```

错误：空字符串

3.2.38 GetAgentDetailsByAgentIDs(通过 ID 批量获取座席信息)

【功能】

根据坐席工号取坐席信息。

【函数说明】

```
string GetAgentDetailByAgentIDs (string AgentIDs)
```

【参数说明】

参数	参数说明
AgentIDs	坐席工号列表

【返回值】

成功：坐席信息

json:数组

```
struct AgentInfoT
{
    std::string agentId;      // 坐席ID
    std::string agentDn;      // 坐席号码
    int64_t loginTime;        // 登陆时间
    int32_t currStatus;       // 坐席当前状态
    int64_t currStatusTime;   // 当前状态时间
    int64_t answerTime;       // 通话开始时间
    int64_t currTime;         // 当前时间
    std::string origCaller;    // 原始主叫
    std::string origCallee;   // 原始被叫
    int32_t privQueueNum;     // 私有队列数
    int32_t callTimes;        // 此次登录呼叫次数
    int32_t answerTimes;      // 此次登录通话次数
    int32_t freeTotalTime;    // 此次登录空闲总时长
    int32_t busyTotalTime;    // 此次登录忙总时长
    int32_t resetTotalTime;   // 此次登录休息总时长
    int32_t callTotalTime;    // 此次登录后呼叫总时长
    int32_t answerTotalTime;  // 此次登录后通话总时长
}
```

错误：空字符串

3.2.39 GetAgentDetailByAgentDNs(通过 DN 批量获取坐席信息)

【功能】

根据坐席分机号码取坐席工号。

【函数说明】

```
string GetAgentDetailByAgentDNs (string AgentDNs)
```

【参数说明】

参数	参数说明
AgentDNs	坐席分机号码，String，最长 32。

【返回值】

成功：坐席信息

json:数组

```
struct AgentInfoT
```

```
{
    std::string agentId;        // 坐席ID
    std::string agentDn;        // 坐席号码
    int64_t loginTime;          // 登陆时间
    int32_t currStatus;         // 坐席当前状态
    int64_t currStatusTime;     // 当前状态时间
    int64_t answerTime;         // 通话开始时间
    int64_t currTime;           // 当前时间
    std::string origCaller;     // 原始主叫
    std::string origCallee;    // 原始被叫
    int32_t privQueueNum;       // 私有队列数
    int32_t callTimes;          // 此次登录呼叫次数
    int32_t answerTimes;        // 此次登录通话次数
    int32_t freeTotalTime;      // 此次登录空闲总时长
    int32_t busyTotalTime;      // 此次登录忙总时长
    int32_t resetTotalTime;     // 此次登录休息总时长
    int32_t callTotalTime;      // 此次登录后呼叫总时长
    int32_t answerTotalTime;    // 此次登录后通话总时长
}
```

错误：空字符串

3.2.40 GetRecordFileName(查询录音文件名)

【功能】

获取录音文件名。

【函数说明】

```
string GetRecordFileName()
```

【参数说明】

【返回值】

录音文件名：成功

空串：失败

3.2.41 GetRestReason(查询休息原因)

【功能】

获取休息原因。

【函数说明】

```
string GetRestReason()
```

【参数说明】

【返回值】

休息原因串：成功

空串：失败

3.2.42 GetAgentNBySkill(当前技能登录座席数)

【功能】

当前技能登录座席数。

【函数说明】

```
int GetAgentNBySkill(string skill)
```

【参数说明】

参数	参数说明
skill	技能信息

【返回值】

成功：坐席数

错误：1000 以上

3.2.43 GetAgentDetailsBySkill(通过技能批量获取座席信息)

【功能】

通过技能批量获取座席信息

【函数说明】

```
string GetAgentDetailsBySkill(string skill)
```

【参数说明】

参数	参数说明
skill	技能信息

【返回值】

成功：

json:数组

```
struct AgentInfoT
```

```
{  
    std::string agentId;    // 坐席ID
```

```
std::string agentDn;      // 坐席号码
int64_t loginTime;        // 登陆时间
int32_t currStatus;       // 坐席当前状态
int64_t currStatusTime;   // 当前状态时间
int64_t answerTime;       // 通话开始时间
int64_t currTime;         // 当前时间
std::string origCaller;   // 原始主叫
std::string origCallee;   // 原始被叫
int32_t privQueueNum;     // 私有队列数
int32_t callTimes;        // 此次登录呼叫次数
int32_t answerTimes;      // 此次登录通话次数
int32_t freeTotalTime;    // 此次登录空闲总时长
int32_t busyTotalTime;    // 此次登录忙总时长
int32_t resetTotalTime;   // 此次登录休息总时长
int32_t callTotalTime;    // 此次登录后呼叫总时长
int32_t answerTotalTime;  // 此次登录后通话总时长
}
```

错误:

3.2.44 GetSkillWaitingNumber(查询队列中排队呼叫信息)

【功能】

查询队列中排队呼叫信息。

【函数说明】

```
string GetSkillWaitingNumber (string skill)
```

【参数说明】

参数	参数说明
Skill	技能信息

【返回值】

成功

Json:

```
queuenumber, freenumber
```

失败

3.2.45 GetSkillWaitingCallInfo(查询队列中排队呼叫信息)

【功能】

查询队列中排队呼叫信息。

【函数说明】

```
string GetSkillWaitingCallInfo(string skill)
```

【参数说明】

参数	参数说明
skill	技能信息，String，最长 32。

【返回值】

0: 成功

Json: 数组

```
struct QueueInfoT
{
    std::string callId;    // callid
    int64_t contactId;    // sessionid
    int64_t requestId;    // 路由请求 id
    std::string origCaller; // 原始主叫
    std::string origCallee; // 原始被叫
    int64_t startTime;    // 请求开始时间
    int64_t currTime;     // 当前时间
    int64_t changeTime;   // 路由调整时间
}
```

其他值: 失败

3.2.46 GetPrivateQueue(查询坐席私有队列信息)

【功能】

查询坐席私有队列信息。

【函数说明】

```
string GetPrivateQueue(string agentID)
```

【参数说明】

参数	参数说明
agentID	坐席工号

【返回值】

0: 成功

Json: 数组

```
struct QueueInfoT
{
    std::string callId;    // callid
    int64_t contactId;    // sessionid
    int64_t requestId;    // 路由请求 id
    std::string origCaller; // 原始主叫
    std::string origCallee; // 原始被叫
    int64_t startTime;    // 请求开始时间
    int64_t currTime;     // 当前时间
    int64_t changeTime;   // 路由调整时间
}
```

其他值：失败

3.2.47 GetSysInfo(查询系统相关数据)

【功能】

查询坐席私有队列信息。

【函数说明】

```
string GetSysInfo ()
```

【参数说明】

【返回值】

0：成功

```
struct SysInfoT
{
    int32_t totalAgents;    // 签入座席总数
    int32_t totalCalls;    // 系统呼叫总数
    int32_t ivrCalls;      // IVR 上呼叫总数
    int32_t agentCalls;    // 坐席呼叫总数
    int32_t queueCalls;    // 队列上呼叫总数
    int32_t privQueueCalls; // 私有队列呼叫总数
}
```

其他值：失败

3.2.48 GetCallingNo(取主叫信息)

【功能】

获取呼叫坐席当前处理呼叫的主叫号码。

【函数说明】

```
string GetCallingNo ()
```

【参数说明】

无

【返回值】

主叫：成功

空串：失败

3.2.49 GetOrgCallingNo(取原始主叫信息)

【功能】

获取呼叫坐席当前处理呼叫的原始主叫号码。

【函数说明】

```
string GetOrgCallingNo ()
```

【参数说明】

无

【返回值】

原始主叫：成功

空串：失败

3.2.50 GetCalledNo(取被叫信息)

【功能】

获取呼叫坐席当前处理呼叫的被叫号码。

【函数说明】

```
string GetCalledNo ()
```

【参数说明】

无

【返回值】

被叫：成功

空串：失败

3.2.51 GetOrgCalledNo(取原始被叫信息)

【功能】

获取呼叫坐席当前处理呼叫的原始被叫号码。

【函数说明】

```
string GetOrgCalledNo ()
```

【参数说明】

无

【返回值】

原始被叫：成功

空串：失败

【其他方法】

3.2.52 SynchronizeCTITime(同步 CTI 时间)

【功能】

坐席与 CTI 持时间同步。

【函数说明】

```
int SynchronizeCTITime()
```

【参数说明】

无

【返回值】

0：成功

其他值：失败

3.2.53 GetPromptByErrorCode(查询错误描述)

【功能】

根据错误查询错误描述字符串。

【函数说明】

```
string GetPromptByErrorCode(int ErrorCode)
```

【参数说明】

参数	参数说明
ErrorCode	错误码

【返回值】

错误描述字符串。

3.2.54 JumptheQueue(插队)

【功能】

对私有队列内的呼叫请求进行插队。

【函数说明】

```
Long JumptheQueue(int64_t requestId)
```

【参数说明】

参数	参数说明
requestId	路由请求 id

【返回值】

0: 成功

其他值: 失败

3.2.55 TransIVR(转接到 IVR)

【功能】

将当前通话转接到 IVR。

【函数说明】

```
Long TransIVR(string transferNum)
```

【参数说明】

参数	参数说明
transferNum	目标号码

【返回值】

0: 成功

其他值: 失败

3.2.56 TransIVREx(设置主叫转接到 IVR)

【功能】

将当前通话转接到 IVR。

【函数说明】

```
Long TransIVR(string callerNum, string transferNum)
```

【参数说明】

参数	参数说明
callerNum	给被叫显示的号码
transferNum	目标号码

【返回值】

0: 成功

其他值: 失败

3.3 事件

【坐席状态事件】

3.3.1 OnUnLogin(签出成功事件)

【事件说明】

签出成功事件。

【事件定义】

```
void OnUnLogin(int eventID, string callID, long contactID, long partyNum,  
string otherParty, int otherPartyAttribute, string originatingParty, string  
originalANI, string originalDNIS, int reason, long timestamp)
```

【参数说明】

参数	参数说明
eventID	事件 ID
callID	呼叫 ID

contactID	会话 ID
partyNum	呼叫方
otherParty	对方
otherParty Attribute	对方属性
originatingParty	发起方
originalANI	原始主叫
originalDNIS	原始被叫
reason	原因
timestamp	时间戳

reason 原因说明：3.3.1 至 3.3.19 节的原因码均为以下表中原因

参数	参数说明
CerSuccess=0	操作成功
CerInvokeComplete	调用执行完成
CerCallDestBusy	对方忙
CerCallNoAnswer	对方无应答
CerCallDestUnknown	对方状态未知
CerCallDestInvalid	对方号码无效
CerCallTimeout	呼叫超时
CerUnloginInitial	初始化
CerUnloginLogout	签出
CerUnloginSelfForceout	强制签出
CerUnloginOtherForceout	被班长或 ACD 超时无心跳签出
CerUnloginBackup	acd 主备倒换，先签出
CerUnloginResignin	坐席重签，先签出
CerUnloginElse	未定义其他状况
CerIdleSelfLogin	坐席签入，设置初始状态
CerIdleOtherLogin	坐席被签入，设置初始状态

CerIdleNormalSelfHangup	通话成功，本方挂断
CerIdleNormalOtherHangup	通话成功，对方挂断
CerIdleErrorSelfHangup	通话失败，本方挂断
CerIdleErrorOtherHangup	通话失败，对方挂断
CerIdleUnlock	坐席解锁
CerIdleIdle	坐席置闲
CerIdleSelfSet	坐席状态改变
CerIdleOtherSet	坐席状态被改变
CerIdleElse	未定义其他状况
CerTalkSelfCreat	本方发起通话
CerTalkOtherCreat	对方发起通话
CerTalkSelfRetrieve	本方通话接回
CerTalkOtherRetrieve	对方通话接回
CerTalkElse	未定义其他状况

【触发条件】

当坐席签出时发生。

3.3.2 OnReady(空闲事件)

【事件说明】

空闲事件。

【事件定义】

```
void OnReady(int eventID, string callID, long contactID, long partyNum, string otherParty, int otherPartyAttribute, string originatingParty, string originalANI, string originalDNIS, int reason, long timestamp)
```

【参数说明】

参数	参数说明
eventID	事件 ID
callID	呼叫 ID

contactID	会话 ID
partyNum	呼叫方
otherParty	对方
otherPartyAttribute	对方属性
originatingParty	发起方
originalANI	原始主叫
originalDNIS	原始被叫
reason	原因
timestamp	时间戳

【触发条件】

当坐席空闲时发生。

3.3.3 OnBusy(忙事件)

【事件说明】

忙事件。

【事件定义】

```
void OnBusy(int eventID, string callID, long contactID, long partyNum, string
otherParty, int otherPartyAttribute, string originatingParty, string originalANI,
string originalDNIS, int reason, long timestamp)
```

【参数说明】

参数	参数说明
eventID	事件 ID
callID	呼叫 ID
contactID	会话 ID
partyNum	呼叫方
otherParty	对方
otherPartyAttribute	对方属性
originatingParty	发起方
originalANI	原始主叫

originalDNIS	原始被叫
reason	原因
timestamp	时间戳

【触发条件】

当坐席忙时发生。

3.3.4 OnRest(休息事件)

【事件说明】

休息事件。

【事件定义】

```
void OnRest(int eventID, string callID, long contactID, long partyNum, string otherParty, int otherPartyAttribute, string originatingParty, string originalANI, string originalDNIS, int reason, long timestamp)
```

【参数说明】

参数	参数说明
eventID	事件 ID
callID	呼叫 ID
contactID	会话 ID
partyNum	呼叫方
otherParty	对方
otherParty Attribute	对方属性
originatingParty	发起方
originalANI	原始主叫
originalDNIS	原始被叫
reason	原因
timestamp	时间戳

【触发条件】

当坐席休息时发生。

3.3.5 OnLock(锁定坐席事件)

【事件说明】

锁定事件。

【事件定义】

```
void OnLock(int eventID, string callID, long contactID, long partyNum, string otherParty, int otherPartyAttribute, string originatingParty, string originalANI, string originalDNIS, int reason, long timestamp)
```

【参数说明】

参数	参数说明
eventID	事件 ID
callID	呼叫 ID
contactID	会话 ID
partyNum	呼叫方
otherParty	对方
otherPartyAttribute	对方属性
originatingParty	发起方
originalANI	原始主叫
originalDNIS	原始被叫
reason	原因
timestamp	时间戳

【触发条件】

当坐席锁定时发生。

3.3.6 OnAlerting(振铃事件)

【事件说明】

振铃事件。

【事件定义】

```
void OnAlerting(int eventID, string callID, long contactID, long partyNum, string otherParty, int otherPartyAttribute, string originatingParty, string originalANI, string originalDNIS, int reason, long timestamp)
```

【参数说明】

参数	参数说明
eventID	事件 ID
callID	呼叫 ID
contactID	会话 ID
partyNum	呼叫方
otherParty	对方
otherParty Attribute	对方属性
originatingParty	发起方
originalANI	原始主叫
originalDNIS	原始被叫
reason	原因
timestamp	时间戳

【触发条件】

来电触发此事件。

3.3.7 OnHarfAlerting(呼本方振铃事件)

【事件说明】

半振铃事件。

【事件定义】

```
void OnHarfAlerting(int eventID, string callID, long contactID, long partyNum,
string otherParty, int otherPartyAttribute, string originatingParty, string
originalANI, string originalDNIS, int reason, long timestamp)
```

【参数说明】

参数	参数说明
eventID	事件 ID
callID	呼叫 ID
contactID	会话 ID
partyNum	呼叫方

otherParty	对方
otherParty Attribute	对方属性
originatingParty	发起方
originalANI	原始主叫
originalDNIS	原始被叫
reason	原因
timestamp	时间戳

【触发条件】

本方来电触发此事件。

3.3.8 OnConnecting(外呼中事件)

【事件说明】

外呼中事件。

【事件定义】

```
void OnConnecting(int eventID, string callID, long contactID, long partyNum,
string otherParty, int otherPartyAttribute, string originatingParty, string
originalANI, string originalDNIS, int reason, long timestamp)
```

【参数说明】

参数	参数说明
eventID	事件 ID
callID	呼叫 ID
contactID	会话 ID
partyNum	呼叫方
otherParty	对方
otherParty Attribute	对方属性
originatingParty	发起方
originalANI	原始主叫
originalDNIS	原始被叫
reason	原因

timestamp	时间戳
-----------	-----

【触发条件】

对方振铃触发此事件。

3.3.9 OnConnected(通话事件)

【事件说明】

接通事件。

【事件定义】

```
void OnConnected(int eventID, string callID, long contactID, long partyNum,
string otherParty, int otherPartyAttribute, string originatingParty, string
originalANI, string originalDNIS, int reason, long timestamp)
```

【参数说明】

参数	参数说明
eventID	事件 ID
callID	呼叫 ID
contactID	会话 ID
partyNum	呼叫方
otherParty	对方
otherPartyAttribute	对方属性
originatingParty	发起方
originalANI	原始主叫
originalDNIS	原始被叫
reason	原因
timestamp	时间戳

【触发条件】

通话触发此事件。

3.3.10 OnHarfConnected(本方通话事件)

【事件说明】

半接通事件。

【事件定义】

```
void OnHarfConnected(int eventID, string callID, long contactID, long partyNum,
string otherParty, int otherPartyAttribute, string originatingParty, string
originalANI, string originalDNIS, int reason, long timestamp)
```

【参数说明】

参数	参数说明
eventID	事件 ID
callID	呼叫 ID
contactID	会话 ID
partyNum	呼叫方
otherParty	对方
otherPartyAttribute	对方属性
originatingParty	发起方
originalANI	原始主叫
originalDNIS	原始被叫
reason	原因
timestamp	时间戳

【触发条件】

本方通话触发此事件。

3.3.11 OnInternalConnected(内部通话事件)

【事件说明】

内部通话事件。

【事件定义】

```
void OnInternalConnected(int eventID, string callID, long contactID, long
partyNum, string otherParty, int otherPartyAttribute, string originatingParty,
string originalANI, string originalDNIS, int reason, long timestamp)
```

【参数说明】

参数	参数说明
eventID	事件 ID

callID	呼叫 ID
contactID	会话 ID
partyNum	呼叫方
otherParty	对方
otherPartyAttribute	对方属性
originatingParty	发起方
originalANI	原始主叫
originalDNIS	原始被叫
reason	原因
timestamp	时间戳

【触发条件】

内部通话触发此事件。

3.3.12 OnSuspended(保持事件)

【事件说明】

保持事件。

【事件定义】

```
void OnSuspended(int eventID, string callID, long contactID, long partyNum,
string otherParty, int otherPartyAttribute, string originatingParty, string
originalANI, string originalDNIS, int reason, long timestamp)
```

【参数说明】

参数	参数说明
eventID	事件 ID
callID	呼叫 ID
contactID	会话 ID
partyNum	呼叫方
otherParty	对方
otherPartyAttribute	对方属性

originatingParty	发起方
originalANI	原始主叫
originalDNIS	原始被叫
reason	原因
timestamp	时间戳

【触发条件】

保持触发此事件。

3.3.13 OnConsultConnecting(咨询中事件)

【事件说明】

咨询中事件。

【事件定义】

```
void OnConsultConnecting(int eventID, string callID, long contactID, long partyNum, string otherParty, int otherPartyAttribute, string originatingParty, string originalANI, string originalDNIS, int reason, long timestamp)
```

【参数说明】

参数	参数说明
eventID	事件 ID
callID	呼叫 ID
contactID	会话 ID
partyNum	呼叫方
otherParty	对方
otherPartyAttribute	对方属性
originatingParty	发起方
originalANI	原始主叫
originalDNIS	原始被叫
reason	原因
timestamp	时间戳

【触发条件】

咨询中触发此事件。

3.3.14 OnConsultConnected(咨询通话事件)

【事件说明】

咨询通话事件。

【事件定义】

```
void OnConsultConnected(int eventID, string callID, long contactID, long partyNum, string otherParty, int otherPartyAttribute, string originatingParty, string originalANI, string originalDNIS, int reason, long timestamp)
```

【参数说明】

参数	参数说明
eventID	事件 ID
callID	呼叫 ID
contactID	会话 ID
partyNum	呼叫方
otherParty	对方
otherPartyAttribute	对方属性
originatingParty	发起方
originalANI	原始主叫
originalDNIS	原始被叫
reason	原因
timestamp	时间戳

【触发条件】

咨询通话触发此事件。

3.3.15 OnConference(会议事件)

【事件说明】

会议事件。

【事件定义】

```
void OnConference(int eventID, string callID, long contactID, long partyNum, string otherParty, int otherPartyAttribute, string originatingParty, string
```

`originalANI, string originalDNIS, int reason, long timestamp)`

【参数说明】

参数	参数说明
eventID	事件 ID
callID	呼叫 ID
contactID	会话 ID
partyNum	呼叫方
otherParty	对方
otherPartyAttribute	对方属性
originatingParty	发起方
originalANI	原始主叫
originalDNIS	原始被叫
reason	原因
timestamp	时间戳

【触发条件】

会议触发此事件。

3.3.16 OnMonitor(监听事件)

【事件说明】

监听事件。

【事件定义】

```
void OnMonitor(int eventID, string callID, long contactID, long partyNum,
string otherParty, int otherPartyAttribute, string originatingParty, string
originalANI, string originalDNIS, int reason, long timestamp)
```

【参数说明】

参数	参数说明
eventID	事件 ID
callID	呼叫 ID
contactID	会话 ID
partyNum	呼叫方

otherParty	对方
otherParty Attribute	对方属性
originatingParty	发起方
originalANI	原始主叫
originalDNIS	原始被叫
reason	原因
timestamp	时间戳

【触发条件】

监听触发此事件。

3.3.17 OnInsert(强插事件)

【事件说明】

强插事件。

【事件定义】

```
void OnInsert(int eventID, string callID, long contactID, long partyNum,
string otherParty, int otherPartyAttribute, string originatingParty, string
originalANI, string originalDNIS, int reason, long timestamp)
```

【参数说明】

参数	参数说明
eventID	事件 ID
callID	呼叫 ID
contactID	会话 ID
partyNum	呼叫方
otherParty	对方
otherParty Attribute	对方属性
originatingParty	发起方
originalANI	原始主叫
originalDNIS	原始被叫
reason	原因

timestamp	时间戳
-----------	-----

【触发条件】

插入触发此事件。

3.3.18 OnMonitored(被监听事件)

【事件说明】

被监听事件。

【事件定义】

```
void OnMonitored(int eventID, string callID, long contactID, long partyNum,
string otherParty, int otherPartyAttribute, string originatingParty, string
originalANI, string originalDNIS, int reason, long timestamp)
```

【参数说明】

参数	参数说明
eventID	事件 ID
callID	呼叫 ID
contactID	会话 ID
partyNum	呼叫方
otherParty	对方
otherPartyAttribute	对方属性
originatingParty	发起方
originalANI	原始主叫
originalDNIS	原始被叫
reason	原因
timestamp	时间戳

【触发条件】

被监听触发此事件。

3.3.19 OnInserted(被强插事件)

【事件说明】

被强插事件。

【事件定义】

```
void OnInserted(int eventID, string callID, long contactID, long partyNum,
string otherParty, int otherPartyAttribute, string originatingParty, string
originalANI, string originalDNIS, int reason, long timestamp)
```

【参数说明】

参数	参数说明
eventID	事件 ID
callID	呼叫 ID
contactID	会话 ID
partyNum	呼叫方
otherParty	对方
otherPartyAttribute	对方属性
originatingParty	发起方
originalANI	原始主叫
originalDNIS	原始被叫
reason	原因
timestamp	时间戳

【触发条件】

被强插触发此事件。

【其他事件】

3.3.20 OnBeginRecord(启动录音事件)

【事件说明】

开始录音事件。

【事件定义】

```
void OnReady(string callID, long contactID, string filename, int reason, long
timestamp)
```

【参数说明】

参数	参数说明
callID	呼叫 ID

contactID	会话 ID
filename	录音文件
reason	原因
timestamp	时间戳

reason 原因说明

参数	参数说明
MerMediaUnknown=0	未知
MerMediaEnd=1	媒体操作结束
MerMediaOther=2	其他

【触发条件】

开始录音插触发此事件。

3.3.21 OnEndRecord(结束录音事件)

【事件说明】

停止录音事件。

【事件定义】

```
void OnEndRecord(string callID, long contactID, string filename, int reason, long timestamp)
```

【参数说明】

参数	参数说明
callID	呼叫 ID
contactID	会话 ID
filename	录音文件
reason	原因
timestamp	时间戳

reason 原因说明

参数	参数说明
MerMediaUnknown=0	未知

MerMediaEnd=1	媒体操作结束
MerMediaOther=2	其他

【触发条件】

停止录音插触发此事件。

3.3.22 OnInprivatequeue(进入私有队列)

【事件说明】

进入私有队列事件。

【事件定义】

```
void OnInprivatequeue(string callID, long contactID, string originalANI,  
string originalDNIS, int reason, long timestamp)
```

【参数说明】

参数	参数说明
callID	呼叫 ID
contactID	会话 ID
originalANI	原始主叫
originalDNIS	原始被叫
reason	原因 = 0
timestamp	时间戳

【触发条件】

进入队列触发此事件。

3.3.23 OnOutprivatequeue(离开私有队列)

【事件说明】

离开队列事件。

【事件定义】

```
void OnOutprivatequeue(string callID, long contactID, string originalANI,  
string originalDNIS, int reason, long timestamp)
```

【参数说明】

参数	参数说明
callID	呼叫 ID
contactID	会话 ID
originalANI	原始主叫
originalDNIS	原始被叫
reason	原因 = 0
timestamp	时间戳

【触发条件】

离开队列触发此事件。

3.3.24 OnCTIDisconnect(CTI 断连事件)

【事件说明】

CTI 宕机断连事件。

【事件定义】

```
void OnCTIDisconnect(string callID, long contactID, string originalANI,  
string originalDNIS, int reason, long timestamp)
```

【参数说明】

参数	参数说明
callID	呼叫 ID
contactID	会话 ID
originalANI	原始主叫
originalDNIS	原始被叫
reason	原因
timestamp	时间戳

reason:

参数	参数说明
EVENTREASON_IMS_DISCONNECT=1000	IMS 断链

EVENTREASON_ACD_DISCONNECT	ACD 断链
----------------------------	--------

【触发条件】

CTI 宕机时触发此事件。

3.3.25 OnMasterChanged(主备倒换事件)

【事件说明】

CTI 主备倒换事件。【未区分具体服务】

【事件定义】

```
void OnMasterChanged(string callID, long contactID, string originalANI,
string originalDNIS, int reason, long timestamp)
```

【参数说明】

参数	参数说明
callID	呼叫 ID
contactID	会话 ID
originalANI	原始主叫
originalDNIS	原始被叫
reason	原因
timestamp	时间戳

reason:

参数	参数说明
EVENTREASON_IMS_MASTERCHANGE	IMS 主备倒换
EVENTREASON_ACD_MASTERCHANGE	ACD 主备倒换
EVENTREASON_ACD_RELOGINSUCCESS	重签 ACD 成功

【触发条件】

CTI 主备倒换触发此事件。

3.3.26 OnOperationFailed (操作失败事件)

【事件说明】

操作失败

【事件定义】

```
void OnOperationFailed(string callID, long contactID, string originalANI,
string originalDNIS, int reason, long timestamp)
```

【参数说明】

参数	参数说明
callID	呼叫 ID
contactID	会话 ID
originalANI	原始主叫
originalDNIS	原始被叫
reason	原因
timestamp	时间戳

reason:

参数	参数说明
EVENTREASON_IMS_OPERATIONFAILED	IMS 操作错误
EVENTREASON_ACD_OPERATIONFAILED	ACD 操作错误
EVENTREASON_ACD_RELOGINFAILED	重签 ACD 失败

【触发条件】

CTI 主备倒换触发此事件。

3.3.27 OnBeforeOutCall(外呼前事件)

【事件说明】

外呼前事件，用于外呼校验。

【事件定义】

```
void OnBeforeOutCall(string otherParty, int otherPartyAttribute, string
originalANI, string originalDNIS, int reason, long timestamp)
```

【参数说明】

参数	参数说明
otherParty	对方
otherPartyAttribute	对方属性
originalANI	原始主叫
originalDNIS	原始被叫
reason	触发该事件的原因： 1：外呼外线 2：咨询外线 3：单步转移外线
timestamp	时间戳

【触发条件】

外线呼叫前插触发此事件，操作包括：外呼、咨询、单步转移

4 特殊应用场景编程介绍

4.1 Agentbar 发起外呼如何防骚扰

通过 agentbar 的界面发起外呼时，如何屏蔽当前号码。

- 1、系统在发起外呼前会发出 OnBeforeOutCall 事件，OnBeforeOutCall 事件将会携带外呼号码的参数
- 2、在事件的处理中查询外呼号码，是否需要屏蔽，如果不需要屏蔽，则设置属性 IsCallOut 为 true，否则设置为 false。
- 3、系统根据属性 IsCallOut 决定是否真正发起外呼

4.2 前端页面何时写话单

几乎所有的时间都有 reason 参数，根据这个参数确定如何写话单，如：

OnReady 事件，reason = CerIdleSelfLogin，写签入话单。

OnReady 事件，reason = CerIdleNormalSelfHangup，写挂断话单。

外呼模式下（StatusChangeType=1）：

OnBusy 事件，reason = CerIdleSelfLogin，写签入话单。

OnBusy 事件、OnReady 事件, reason = CerIdleNormalSelfHangup, 写挂断话单。

OnUnLogin 事件, reason= CerUnloginLogout, 写签出话单

OnConnected 事件, reason= EVENTREASON_TALK_CREATE, 写开始通话话单

OnInternalConnected 事件, reason= EVENTREASON_TALK_CREATE, 写开始通话话单