ToneThink. soft

UltrCRM_Webservice

接口调用说明

2013 年

一、概述

1．1、CallThink 呼叫中心简介

CallThink 呼叫中心系统是针对企业级客服中心而设计的，一个以电话通信、计算机网络集成为核心的智能通信平台，它将交换技术、CTI技术、网络技术有机的结合起来，和企业的管理、企业的文化、企业的工作流程通过这个平台，以信息技术的方式得以实施。它自1998年推向市场以来，以其功能齐全、性能稳定、价格合理、使用方便等特点，已广泛应用于大中型企业的技术支持、售后服务、产品订购、报刊发行等领域。

CallThink 呼叫中心系统主要由三部分组成：

1、 前端通信平台

2、 客户业务应用（CRM）

3、 质量检验和业务统计

前端通信平台，包含了排队机、CTI服务器、IVR服务器、传真服务器、全程录音设备等几个部分，是客服中心实施的基础。

在后端主要由客户关系管理（CRM）以及支撑客户业务的数据库系统、业务处理系统等组成，该部分将用户的业务流程和管理理念，通过计算机网络技术实现。

质量检验通过技术手段，可以对业务员的工作量、服务态度、技能水平、工作效率等业务能力进行考察；通过实时监测，录音监听等手段，可以抽查业务员处理业务的每一个细节，为掌控整个客服中心的服务水平技术上的支持。

业务统计报表可以从业务量、工作效率、系统使用率、座席繁忙率、用户地域分布、客户资源挖掘等多方面着手，提供给管理人员详尽、真实的第一手材料，为合理使用人力、物力资源，提高客服中心的接通率提供依据。

CallThink 呼叫中心平台是一个开放式智能系统，通过系统提供的API开发包，可以满足所有业务开发的需求。针对IVR语音业务，提供SmarTele流程生成器平台，可以满足用户定制语音流程的各种需求；针对CRM业务开发，提供“CRM客户端控件ATClient.ocx”，本手册将对客户端控件的使用作详细的说明。

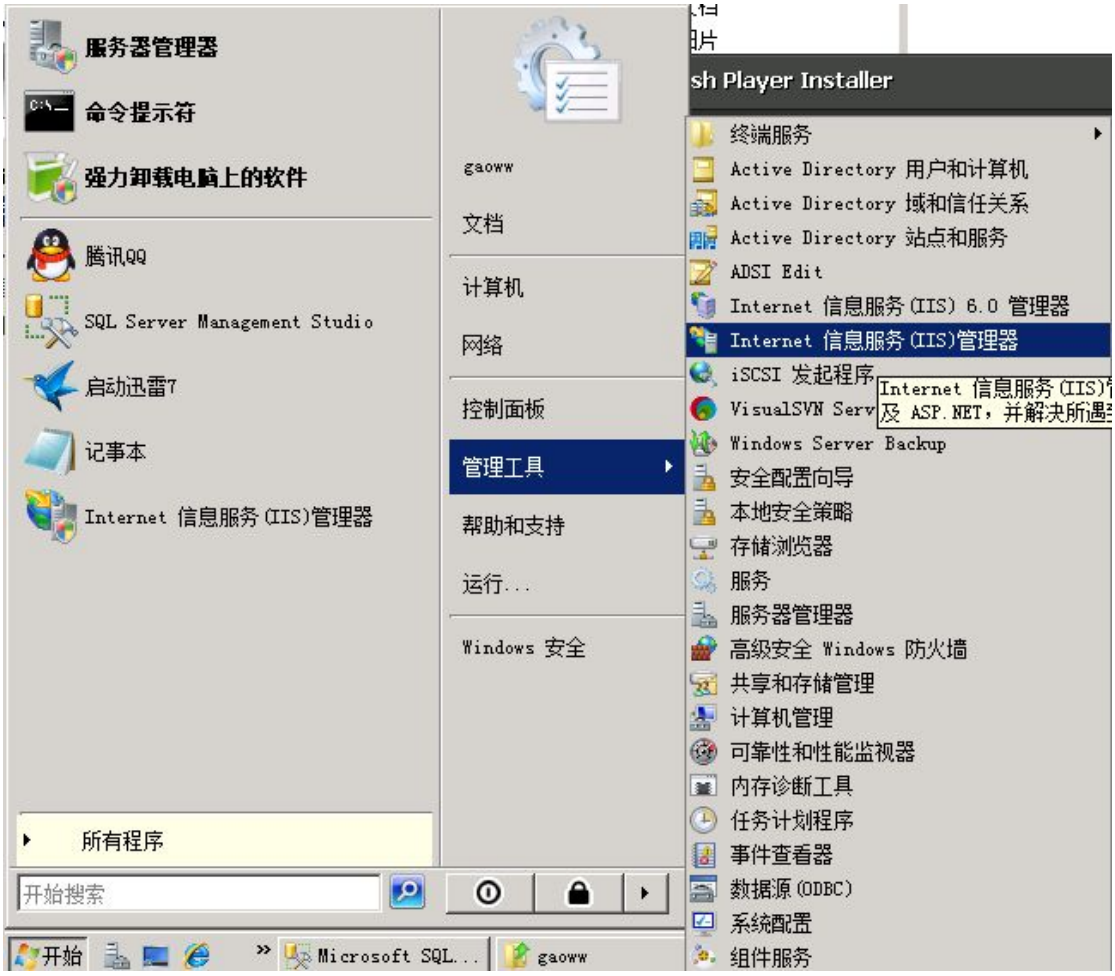
二、使用 UltraCRM_Webservice 开发业务系统

2.1 开发流程概述

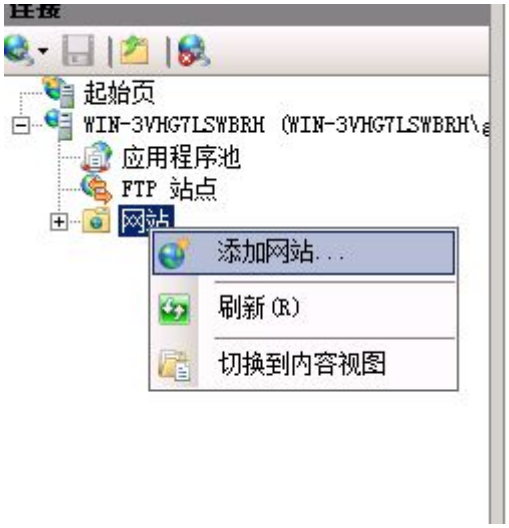
2.1.1、部署 UltraCRM_Webservice 站点

1） 将安装盘“\SDK\UltraCRM_Webservice”目录拷到站点部署的服务器上，建立 iis 站点

2）点击操作系统菜单中的“开始”->“管理工具”->“Internet 信息服务（IIS）管理器”，如下图所示



3) 右键选择“网站”->添加网站，如下图所示



4）在打开的“添加网站”界面中设置网站的名称，物理路径，ip 地址和端口，将“应用程序池”选择为“Classic .NET AppPool”，设置好后选择确定

4、 web.config 参数设置

```
<appSettings>
  <add key="cfgConn_callthink" value="Data Source='127.0.0.1';initial catalog=CALLTHINK;password=UltraTel@5266;persist security inf
  <add key="cfgConn_crm" value="Data Source='127.0.0.1';initial catalog=CALLTHINK_CRM;password=UltraTel@5266;persist security info=
  <add key="cfgConn_cdr" value="Data Source='127.0.0.1';initial catalog=CALLTHINK_CDR;password=UltraTel@5266;persist security info=
  <add key="cfgConn_Talentel_log" value="Data Source='127.0.0.1';initial catalog=TALENTEL_LOG;password=UltraTel@5266;persist securi
  <add key="cfgConn_rpt" value="Data Source='127.0.0.1';initial catalog=CALLTHINK_RPT;password=UltraTel@5266;persist security info=

  <!--需要手动设置UltraCTI-SOA服务器IP-->
  <add key="cfgSOAServer_ip" value="127.0.0.1"/>
```

参数说明：

cfgConn_callthink:callthink 数据库连接参数

cfgConn_crm：callthink_crm 数据库连接参数

cfgConn_cdr：callthink_cdr 数据库连接参数

cfgConn_Talentel_log：talentel_log 数据库连接参数

cfgConn_rpt：callthink_rpt 数据库连接参数

注：以上 5 个参数是给 UltraCRM_Webservice.asmx 中的接口使用的

cfgSOAServer_ip：CTI 服务器的 IP 地址

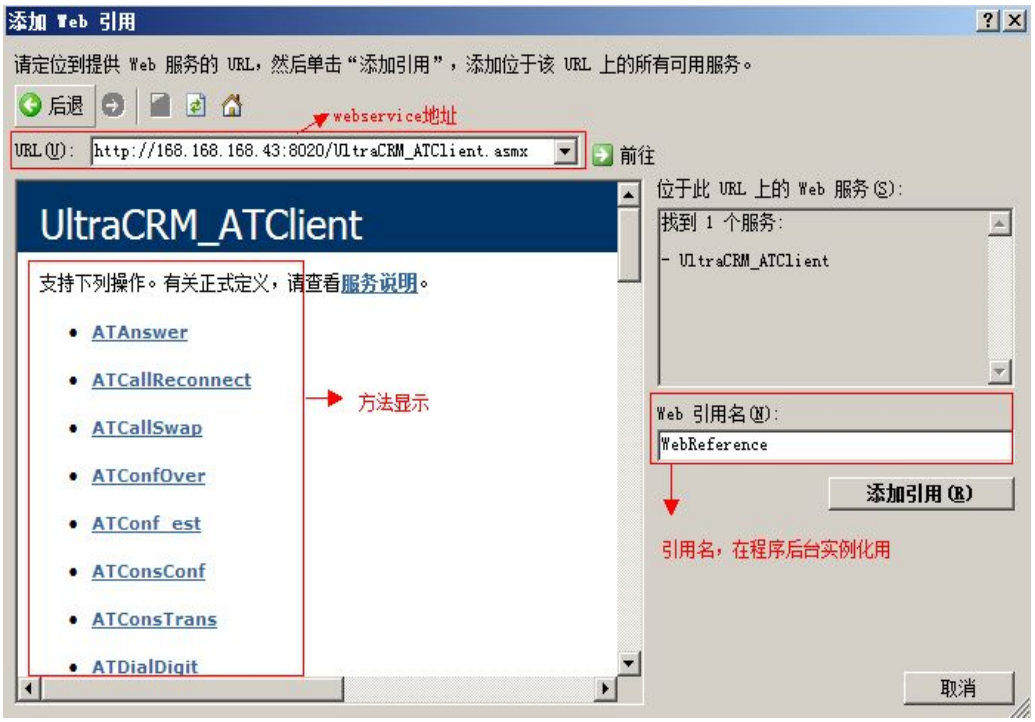
注：cfgSOAServer_ip 参数是给 UltraCRM_ATClient.asmx 中软电话的接口使用的

2.1.2、添加 webservice 引用

注：在 UltraCRM_Webservice 接口中一共有两个接口地址，一个是 http://部署的服务器ip/UltraCRM_ATClient.asmx，这个接口是软电话操作接口的 url 地址；另一个是 http://部署的服务器 ip/UltraCRM_Webservice.asmx，这个接口是查询呼叫中心数据库接口的 url 地址。

下面以“UltraCRM_ATClient.asmx”为例说明。

首先在需要使用 UltraCRM_Webservice 的项目中添加 web 引用，以 asp.net 为例:



2.1.3、程序后台调用 WebService

```
public partial class _Default : System.Web.UI.Page
{
    //实例化WebService
    WebReference.UltraCRM_ATClient WebService_ATClient = new WebReference.UltraCRM_ATClient();

    protected void Page_Load(object sender, EventArgs e)
    {
       //初始化
    }
    //按钮点击事件
    protected void btnSetBusy(object sender, EventArgs e)
    {
        //获取座席员呼叫信息
        string strCallInfo = WebService_ATClient.ATGetCallInfo("8643", "");
        //获取中继线号码
        string strTrunk_list = WebService_ATClient.ATGetTrunk_list();
        //保持呼叫 -1: 失败 1: 成功
        int nHoldCall = WebService_ATClient.ATHoldCall("8643");
        //发起呼叫 -1: 失败 1: 成功
        int nPlaceCall = WebService_ATClient.ATPlaceCall("8643", "");
    }
}
```

2.2 典型应用之.NET

1)前台代码

```
<tr>
                    <td align="left" colspan="3">
                        <span id="D">挂断电话：</span>
                    </td>
                </tr>
                <tr id="trATHangup">
                    <td width="30%" align="left">
                        座席工号：<input type="text" id="txtATHangup_Uid" value='8615' size='12' runat="server"
/>
                    </td>
                    <td align="left" width="30%">
                        呼叫标识：<input type="text" id="txtATHangup_Callid" value='' size='12' runat="server" />
                    </td>
                    <td width="80" align="left">
                        <input name="btnATHangup" onserverclick="btnATHangup_Click" runat="server" type="button"
                            value="挂断" id="btnATHangup" />
                    </td>
                </tr>
                <tr>
                    <td align="left" colspan="4">
                        <span id="C">获取传真发件箱中的记录：</span>
                    </td>
                </tr>
                <tr id="trFax_Sent">
                    <td width="30%">
                        开 始 日 期 ： <input type="text" id="txtFax_Sent_Sdate" value="20100101" size="20"
runat="server" />
                    </td>
                    <td width="30%" align="left">
                        结 束 日 期 ： <input type="text" id="txtFax_Sent_Edate" value='20130101' size='12'
runat="server" />
                    </td>
                    <td align="left" width="30%">
                        查询工号：<input type="text" id="txtFax_Sent_Uid" value='8615' size='12' runat="server"
/>
                    </td>
                    <td width="80" align="left">
                        <input name="btnFax_Sent" onserverclick="btnFax_Sent_Click" runat="server" type="button"
                            value="获取传真发件箱中的记录" id="btnFax_Sent" />
                    </td>
                </tr>
```

2)后台代码

```
public partial class _Default : System.Web.UI.Page
    {
        //实例化WebService
        WebReference.UltraCRM_ATClient WebService_ATClient = new WebReference.UltraCRM_ATClient();
        WebReference1.UltraCRM_Webservice WebService1 = new WebReference1.UltraCRM_Webservice();

        protected void Page_Load(object sender, EventArgs e)
        {
            //初始化
        }
        //挂断
        protected void btnATHangup_Click(object sender, EventArgs e)
        {
            int nResult = WebService_ATClient.ATHangup(txtATHangup_Uid.Value, txtATHangup_Callid.Value);
            if (nResult == 1)
                TextArea.Value = "成功";
            else if (nResult == -1)
                TextArea.Value = "失败";
        }

        //获取传真发件箱记录
        protected void btnFax_Sent_Click(object sender, EventArgs e)
        {
            TextArea.Value = "";
            //通过调用WebService方法，获取查询到的信息---json格式
            string json_Fax_Sent = WebService1.Fax_Sent(txtFax_Sent_Sdate.Value, txtFax_Sent_Edate.Value,
txtFax_Sent_Uid.Value);

            //以下是把json数据转换成 DataTable 数据

            //取出表名
            Regex rg = new Regex(@"(?<={)[^:]+(?=:\[)", RegexOptions.IgnoreCase);
            string strName = rg.Match(json_Fax_Sent).Value;
            DataTable tb = null;
            //去除表名
            json_Fax_Sent = json_Fax_Sent.Substring(json_Fax_Sent.IndexOf("[") + 1);
            json_Fax_Sent = json_Fax_Sent.Substring(0, json_Fax_Sent.IndexOf("]"));

            //获取数据
            rg = new Regex(@"(?<={)[^}]+(?=})");
            MatchCollection mc = rg.Matches(json_Fax_Sent);
```

```
            for (int i = 0; i < mc.Count; i++)
            {
                string strRow = mc[i].Value;
                string[] strRows = strRow.Split(',');

                //创建表
                if (tb == null)
                {
                    tb = new DataTable();
                    tb.TableName = strName;
                    foreach (string str in strRows)
                    {
                        DataColumn dc = new DataColumn();
                        string[] strCell = str.Split(':');
                        dc.ColumnName = strCell[0].ToString().Replace("\"", "");
                        tb.Columns.Add(dc);
                    }
                    tb.AcceptChanges();
                }

                //增加内容
                DataRow dr = tb.NewRow();
                for (int r = 0; r < strRows.Length; r++)
                {
                    dr[r] = strRows[r].Split(':')[1].Trim().Replace(", ", ",").Replace(": ", ":").Replace("/",
"").Replace("\"", "");
                }
                tb.Rows.Add(dr);
                tb.AcceptChanges();
            }

            GridView1.DataSource = tb;
            GridView1.DataBind();

        }

        //获取座席员呼叫信息
        protected void btnATGetCallInfo_Click(object sender, EventArgs e)
        {
            string strCallInfo = WebService_ATClient.ATGetCallInfo(txtATGetCallInfo_Uid.Value, "");
            TextArea.Value = strCallInfo;
        }
```

三、软电话操作说明

注：使用软电话接口前需要引用 http://部署的服务器 ip/UltraCRM_ATClient.asmx

ATAnswer 方法

```
public int ATAnswer(string strUid, string strCallid);
```

说明：当有电话呼入时，执行此函数，可以自动应答电话，一般用于软应答

参数：

strUid：座席员工号

strCallid ：呼入到分机的电话呼叫标识

返回值：

-1：失败 1：成功

备注：

如果座席分机是 数字话机，或 IP 分机，可以使用耳麦，实现软摘机；如果是模拟分机，需要人工摘机。

ATCallReconnect 方法

```
public int ATCallReconnect(string strUid, string strCallid)
```

说明： 在一个分机上有两个呼叫，挂断活动呼叫，取回保持呼叫

参数：

strUid:座席员工号

strCallid: 活动的呼叫标识

返回值： -1：失败 1：成功

备注：

ATCallSwap 方法

```
public int ATCallSwap(string strUid,string strCallid);
```

说明：在一个分机上两个呼叫之中切换通话

参数：

strUid:座席员工号

strCallid :被保持的呼叫标识

返回值：

-1：失败 1：成功

备注：

主叫 A 与分机 B 正在通话时，通过执行 ATHoldCall()，主叫 A 进入 Holding 状态，分机 B 呼叫分机 C，建立通话。 执行此函数后，分机 C 进入 Holding 状态，主叫 A 与分机 B 通话

ATConf_est 方法

```
public int ATConf_est(string strExt,string strExt_dest);
```

说明：建立电话会议-单步实现。

参数：

strExt：当前分机号码

strExt_dest ： 要加入会议的目标分机或外线号码

返回值：

-1：失败 1：成功

备注：

主叫 A 与分机 B 正在通话时，执行此函数后，主叫 A、分机 B（strExt）、分机 C（strExt_dest），一起进入会议。

ATConsConf 方法

```
public int ATConsConf(string strUid,string strExt_dest);
```

说明：发起会议协商，与 ATConfOver 配合使用

参数：

strUid ： 座席员工号

strExt_dest ： 要加入会议的目标分机或外线号码

返回值：

-1：失败 1：成功

备注：

有两种情形：

5、 主叫 A 与分机 B 正在通话时，执行此函数后，主叫方 A 听到保留音乐，分机 B 呼叫分机 C（strExt_dest），建立通话后，可以协商相关事宜。

6、 主叫 A、分机 B 与分机 C 建立会议，正在通话时，执行此函数后，主叫 A、分机 B 听到保留音乐，分机 B 呼叫分机 D（strExt_dest），建立通话后，可以协商相关事宜。

ATConfOver 方法

```
public int ATConfOver(string strUid,string strCallid);
```

说明：完成会议 将一个座席上的两个呼叫，合在一起形成会议

参数：

strUid：座席员工号

strCallid ：被保持的呼叫标识

返回值：

-1：失败 1：成功

备注：

执行 ATConfOver 函数后，主叫方、该分机与 strExt_dest 一起进入会议。

ATConsTrans 方法

```
public int ATConsTrans(string strUid,string strExt_dest);
```

说明：转移协商

参数：

strUid ：座席员工号

strExt_dest ：转接的目标分机号码

返回值：

-1：失败 1：成功

备注：

分机正在通话时，执行此函数后，主叫方听到保留音乐，该分机呼叫 strExt_dest，建立通话后，可以协商相关事宜。

ATTranOver 方法

```
public int ATTranOver(string strUid,string strCallid);
```

说明：转移完成，与 ATConsTrans 配合使用

参数：

strUid ：座席员工号

strCallid ：被保持的呼叫标识

返回值

-1：失败 1：成功

备注

执行 ATConsTrans 函数后，主叫方听到保留音乐，该分机与 strExt_dest 通话，执行此函数后，该分机听忙音，主叫方与 strExt_dest 通话。

ATDialDigit 方法

```
public int ATDialDigit(string strUid,string strDigit);
```

说明：代替话机号指令（注意：与发送 DTMF 是不同的）

参数:

strUid ： 座席员工号

strDigit ： 话机上按键的位置号，具体数值需要咨询 PBX 技术人员

返回值:

-1：失败 1：成功

备注:

该函数主要配合软电话功能使用，模拟用户在话机上按键动作。

例如:

1、用户摘机，听到拨号音，执行此函数后可以替代拨号。 如果是数字话机，还可以发送功能键码。

2、座席分机通话状态，可以向对方发送按键码。

ATDiscCall 方法

```
public int ATDiscCall(string strUid,string strExt,string strCallid);
```

说明: 强拆电话-分机正在通话时，执行此函数，可以挂断电话。

参数:

strUid ： 座席员工号

strExt ： 正在处于通话状态的、要被强拆的分机号码

strCallid：呼叫标识：可为空

返回值:

-1：失败 1：成功

ATGetCallInfo 方法

```
public string ATGetCallInfo(string strUid, string strKey)
```

说明: 获取座席员呼叫信息

参数:

strUid：座席员工号

strKey：

返回值:

CALLID=呼叫标识;CALLER=主叫;CALLED=被叫号码;DIR=呼叫方向;STATUS=呼叫状态;EXT=座席分机号;TRUNK=中继号;INFO=;UID=座席工号;TIME=呼叫开始时间;TRANS_TYPE=呼叫转移标志;

例如:

CALLID=081155150040;CALLER=643;CALLED=660;DIR=4;STATUS=05;EXT=643;TRUNK=;INFO=;UID=8643;TIME=11:55:13;TRANS_TYPE=0;

备注:

1)DIR:呼叫方向

0：表示未知 1：表示外线呼入 2：表示外线呼出 3：表示内线呼入 4：表示内线呼出

2)STATUS:呼叫状态

01：空闲状态 02：摘机状态 03：振铃状态 04：回铃状态 05：通话状态 06：断开状态 07：

占用状态

3) TRANS_TYPE:呼叫转移标志

0：一次呼叫 1：转入，2：转出 3：转入+转出 4：代接 5：代接+转出 6：被代接 7：申请协商转接

ATGetTrunk_list 方法

```
public string ATGetTrunk_list()
```

说明：获取中继线号码

返回值:

TRUNK1| TRUNK2| TRUNK3| TRUNK4...

例如：

1|10|1001|1002|1003|1004|1005|1006|1007

ATGetUidInfo 方法

```
public string ATGetUidInfo(string strUid)
```

说明：获取座席员信息

参数:

strUid：座席员工号

返回值:

UID=座席工号;NAME=座席姓名;EXT=座席分机号;STATUS=当前座席状态;CALLS=呼叫状态;ROUPS=ACD 组号;REG_GROUP=ALL;LTIME=登录 CTI 时间

例如：

UID=8643;NAME=付智伟;EXT=643;STATUS=01;CALLS=01;GROUPS=15;REG_GROUP=ALL;LTIME=20130408 083539

备注：

1）STATUS:当前座席状态

00：已注销 01：可工作 02：置忙 03：事后处理 04：离席

2）CALLS:呼叫状态

01：空闲状态 02：摘机状态 03：振铃状态 04：回铃状态 05：通话状态 06：断开状态 07：占用状态

ATGetUidInfo_byExt 方法

```
public string ATGetUidInfo_byExt(string strExt)
```

说明：通过座席分机号获取坐席员信息

参数:

strExt：座席分机号

返回值:

UID=座席工号;NAME=座席姓名;EXT=座席分机号;STATUS=当前座席状态;CALLS=呼叫状态;ROUPS=ACD 组号;REG_GROUP=ALL;LTIME=登录 CTI 时间

例如：

UID=8643;NAME=付智伟;EXT=643;STATUS=01;CALLS=01;GROUPS=15;REG_GROUP=ALL;LTIME=20130408 083539

备注：

1）STATUS:当前座席状态

00：已注销 01：可工作 02：置忙 03：事后处理 04：离席

2）CALLS:呼叫状态

01：空闲状态 02：摘机状态 03：振铃状态 04：回铃状态 05：通话状态 06：断开状态 07：占用状态

ATGetUid_Online 方法

```
public string ATGetUid_Online(string strGroup);
```

说明：读取指定组的所有在线座席员工号

参数：

strGroup ：指定 ACD 组，空-读取所有组

返回值：

UID1|UID2|UID3|UID4...

ATHangup 方法

```
public int ATHangup(string strUid,string strCallid);
```

说明：分机正在通话时，执行此函数，可以挂断电话，一般用于软挂机。

参数：

strUid ：座席员工号

strCallId ：呼叫标识：可为空

返回值：

-1：失败 1：成功

备注：

如果座席分机是 数字话机，或 IP 分机，可以使用耳麦，实现软挂机；如果是模拟分机，需要人工挂机。

ATHoldCall 方法

```
public int ATHoldCall(string strUid);
```

说明：保持呼叫

参数：

strUid：座席员工号

返回值：

-1：失败 1：成功

备注：

分机正在通话时，执行此函数，主叫方听到保留音乐，该分机听到拨号音，可以建立另外一个呼叫

ATInsert 方法

```
public int ATInsert(string strUid,string strExt,int nType);
```

说明：强插会议或强插监听

参数：

strUid：座席员工号

strExt：正在处于通话状态的、要被加入会议的分机号码

nType：0-表示取消强插 1-表示强插会议，2-表示强插监听

返回值：

-1：失败 1：成功

备注： 主叫A与分机C（strExt_dest）正在通话时，执行此函数后，主叫A、分机C、本座席一起进入会议。

ATJoinIVR_toCnf 方法

```
public int ATJoinIVR_toCnf(string strUid,int nCh,string strExt,string strInfo);
```

说明：座席发送请求->CTI（将坐席分机的呼叫转到空闲的IVR端口上）

参数：

strUid :座席员工号

nCh :通道号，0-表示由 CTI 选定 IVR 端口

strExt:指定 IVR 分机号

strInfo :业务的数据包，格式如下：

"AC_SWITCHIVR;CALLID=呼叫标识;EXT=座席分机;IVRFILE=IVR 流程文件名;NODE=IVR 流程节点号;IVRMSG= 传入 IVR 的附加信息（自定义）；"

返回值：

-1：表示操作失败，1：表示成功

ATPickCall 方法

```
public int ATPickCall(string strUid,string strExt,string strCallid);
```

说明：代接电话

参数：

strUid：座席员工号

strExt：代接分机号码

strCallid：代接呼叫标识：可为空

返回值：

-1：失败 1：成功

备注：

如果某个分机正在振铃，执行此函数，可以代接电话。

ATPlaceCall 方法

```
public int ATPlaceCall(string strUid,string strCallid);
```

说明：发起呼叫

参数：

strUid：座席员工号

strCallid：被叫号码

返回值：

-1：失败 1：成功

ATRetriveCall 方法

```
public int ATRetriveCall(string strUid,string strCallid);
```

说明：取回呼叫，与ATHoldCall配合使用

参数：

strUid ：座席员工号

strCallid：呼叫标识：可为空

返回值：

-1：失败 1：成功

备注：

分机上有保持呼叫，执行此函数，主叫方停止播放音乐，该分机与主叫方恢复通话

ATSendDtmf 方法

```
public int ATSendDtmf(string strUid,string strDTMF);
```

说明：座席分机通话状态，向对方发送DTMF号码串

参数：

strUid ：座席员工号

strDTMF：必须为 0-9、A-D、*、#

返回值：

-1：失败 1：成功

ATSendMsg 方法

```
public int ATSendMsg(string strUid,string strUid_to,string strInfo);
```

说明：座席员之间发送消息

参数：

strUid ：发送座席员工号

strUid_to ：接收座席员工号

strInfo ：消息内容

返回值:

0:成功

ATSendOEMCommand 方法

```
public int ATSendOEMCommand(string strUid,string strUid_to,string strSubKey,string strMsg);
```

说明：座席员之间发送消息设备间传递消息

参数：

strUid ：发送座席员工号

strUid_to：接收者

strSubKey：业务代码 ，内容如下：

1）AGENTTOMCI：向 mci 程序发送消息，一般用于发送短信、email

2）AGENTTOIVR：向 ivr 程序发送消息

3）AGENTTOAGENT：向坐席发送消息

strMsg：消息内容 ，如下： 当 strCommand 为 AGENTTOMCI 时，strinfo 的内容可以为：

1） EMAIL_SEND;TO=XX;CC=XX;BCC=XX;SUBJ=XX;BODY=XX;FILE=XX;

向 mci 提交发送邮件消息，to：收件人地址，cc：抄送人地址；bcc：暗抄送人地址；subj：邮件主题；body：邮件正文；file：邮件附件存放在服务器上的全路径

2） SMS_SEND;UID=XX;TEL=XX;SUBJ=XX;BODY=XX;

向 mci 提交发送短信的消息：uid：发送短信座席工号；tel：接收短信手机号码;subj：短信主题；body：短信正文 当 strCommand 为 AGENTTOIVR 时，需要跟相应人员再订立 strinfo 的具体内容

ATSendOEMCommand_Route 方法

```
public int ATSendOEMCommand_Route(string strCmd,string strGhid,string strInfo);
```

说明：发送多媒体消息到座席或分组，并可以按路由规则转发消息到分组

参数：

strCmd ：消息的路由点，可以自定义，当消息按 ACD 路由策略由 CTI 控制转发到某个座席时，需要将此变量内容设置到 ACD 路由策略中，由 cti 根据路由策略将消息路由到相应座席。

strGhid：接收消息的座席工号，如果为空或座席忙，则按 ACD 路由策略执行，可以转发消息到组或其他座席

strInfo：座席端接收到的消息内容，自定义，与客户端配合使用。

ATSetAfterWorking 方法

```
public int ATSetAfterWorking(string strUid,int nState,string strCause);
```

说明：设置座席员进入事后处理状态

参数

strUid ： 座席员工号

nState ： 0-取消事后处理 1-进入事后处理状态

strCause：原因

返回值

ATSetBusy 方法

```
public int ATSetBusy(string strUid,int nBusy_state,string strCause);
```

说明：将指定座席员置忙

参数：

strUid ： 座席员工号

nBusy_state ： 0-取消置忙 1-置忙

strCause ： 置忙原因

返回值：

0：失败 1：成功

ATSetLeaveSeat 方法

```
public int ATSetLeaveSeat(string strUid,int nState,string strCause);
```

说明：设置座席员离席/取消离席状态

参数：

strUid ： 座席员工号

nState ： 0-取消离席 1-进入离席状态

strCause ： 原因

返回值：

0：失败 1：成功

ATSetReady 方法

```
public int ATSetReady(string strUid, int nState)
```

说明：设置座席员准备好/未准备好状态

参数：

strUid：座席工号

nState：0-未准备好状态 1-准备好状态，可以接听 ACD 电话

返回值：0：失败 1：成功

备注：

ATTranCall 方法

```
public int ATTranCall(string strExt,string strExt_dest);
```

说明：电话呼叫转移-单步转移。

参数：

strExt：被转移分机号

strExt_dest：目标分机号

返回值：

备注：

单步呼叫转移是指直接将一个分机上的呼叫转移到另外一个分机上，被转移的分机就处于断开状态（数字分机）或空闲状态（模拟分机）。 这是一种比较快速的转接方法，不像一般的转接要先 HOLD，然后再呼叫目标分机，最后再完成转接。 这种转接方式受到交换机类型的限制，SIEMENS 和 AVAYA 交换机可以实现这项功能。

ATTranCall_toIVR 方法

```
public int ATTranCall_toIVR(string strUid,int nCh,string strExt,string strInfo);
```

说明：座席发送请求->CTI（将坐席分机的呼叫转到空闲的IVR端口上）

参数：

strUid ： 座席员工号

nCh ： 通道号，0-表示由 CTI 选定 IVR 端口

strExt ： 指定 IVR 分机号

strInfo ： 业务的数据包 ， 格式如下：

"AC_SWITCHIVR;CALLID=呼叫标识;EXT=座席分机;IVRFILE=IVR 流程文件名;NODE=IVR 流程节点号;IVRMSG= 传入 IVR 的附加信息（自定义）；"

返回值

-1：表示操作失败，1：表示成功

四、数据库接口说明

注：使用软电话接口前需要引用 http://部署的服务器 ip/ UltraCRM_Webservice.asmx

VMS_Recv 方法

```
public string VMS_Recv(string strSdate, string strEdate, string strGhid)
```

说明：获取留言信箱中的记录

参数：

strSdate：查询开始日期 格式：yyyyMMdd

strEdate：查询结束日期 格式：yyyyMMdd

strGhid：查询的工号(可以为空)

返回值：

0:没有记录；-1 输入的开始日期或结束日期不正确；json 格式数据：成功

json 格式数据：

[{\"CALLID\":\"201003021713393\",\"CALLER\":\"82015268\",\"CALLED\":\"\",\"GHID\":\"8656\",\"FILENAME\":\"D:\\\\ToneThink\\\\IVR_DATA\\\\\\\\\\VMSDATA\\\\RVMS\\\\20100302\\\\R17121603.wav\",\"SDATE\":\"20100302\",\"STIME\":\"171339\",\"CHNUM\":3}]

备注：

CALLID:呼叫标识	CALLER：主叫号码	CALLED：被叫号码
GHID：座席工号	FILENAME：文件路径	SDATE：日期
STIME：时间	CHNUM：通道号	

Fax_Sent 方法

```
public string Fax_Sent(string strSdate, string strEdate, string strGhid)
```

说明：获取传真发件箱中的记录

参数：

strSdate：查询开始日期 格式：yyyyMMdd

strEdate：查询结束日期 格式：yyyyMMdd

strGhid：查询的工号(可以为空)

返回值：

0:没有记录；-1 输入的开始日期或结束日期不正确；json 格式数据：成功

json 格式数据：

[{"UNAME":"","FAX":"111","GHID":"8615","SDATE":"20110627","STIME":"141016","SUBJ":"","FILENAME":"d:\\ToneThink\\ivr_data\\faxdata\\sfax\\8615\\test.txt","FILENAME1":"","FILENAME2":""}]

备注：

UNAME:传真接收单位　　FAX：传真号码　　GHID：座席工号
SDATE：发送日期　　STIME：发送时间　　SUBJ：传真主题
FILENAME：传真文件 1　　FILENAME1：传真文件 2　　FILENAME2：传真文件 3

Fax_Recv 方法

```
public string Fax_Recv(string strSdate, string strEdate, string strGhid)
```

说明：获取传真收件箱中的记录

参数：

strSdate：查询开始日期 格式：yyyyMMdd
strEdate：查询结束日期 格式：yyyyMMdd
strGhid：查询的工号(可以为空)

返回值：

0:没有记录；-1 输入的开始日期或结束日期不正确；json 格式数据：成功

json 格式数据：

[{"CALLER":"65120797","GHID":"8615","SDATE":"20100906","STIME":"145314","FILENAME":"d:\\ToneThink\\ivr_data\\FAXDATA\\RFAX\\20100906\\R14522204.tif"}]

备注：

CALLER:对方传真号码　　GHID：传真信箱号码　　SDATE：接收日期
STIME：接收时间　　FILENAME：传真文件

Sms_Sent 方法

```
public string Sms_Sent(string strSdate, string strEdate, string strGhid)
```

说明：获取短信发件箱中的记录

参数：

strSdate：查询开始日期 格式：yyyyMMdd
strEdate：查询结束日期 格式：yyyyMMdd
strGhid：查询的工号(可以为空)

返回值：

0:没有记录；-1 输入的开始日期或结束日期不正确；json 格式数据：成功

json 格式数据：

[{"CALLID":"201108121252265160","TEL_TO":"13521192996","UNAME":"","SUBJECTS":"工作流消息提醒","BODYS":"有一个新工单等待分派，编号为【201108121252072422】，请签收！","SDATE":"20110812","STIME":"125313","GHID":"8615"}]

备注：

CALLID:短信标识　　TEL_TO：收信人号码　　UNAME：收信人
SUBJECTS：短信主题　　BODYS：短信内容　　SDATE：发送日期
STIME：发送时间　　GHID：发送人工号

Sms_Recv 方法

```
public string Sms_Recv(string strSdate, string strEdate, string strGhid)
```

说明：获取短信收件箱中的记录

参数：

strSdate：查询开始日期 格式：yyyyMMdd
strEdate：查询结束日期 格式：yyyyMMdd
strGhid：查询的工号(可以为空)

返回值：

0:没有记录；-1 输入的开始日期或结束日期不正确；json 格式数据：成功

json 格式数据：

[{"CALLID":"201111031741242383","TEL_FROM":"13002275575","UNAME":"","SUBJECTS":"","BODYS":"北京泰康回馈客户，本周六日在五星酒店设宴讲解泰康养老规划和长短期理财产品，特邀您和家人参加，来电 15699858590 预订贵宾席位或回复短信","SDATE":"20111103","STIME":"174124","GHID":""}]

备注：

CALLID:短信标识　　TEL_FROM：发信人号码　　UNAME：发信人
SUBJECTS：短信主题　　BODYS：短信内容　　SDATE：接收日期
STIME：接收时间　　GHID：接收人工号

Email_Sent 方法

```
public string Email_Sent(string strSdate, string strEdate, string strGhid)
```

说明：获取Email发件箱中的记录

参数：

strSdate：查询开始日期 格式：yyyyMMdd
strEdate：查询结束日期 格式：yyyyMMdd
strGhid：查询的工号(可以为空)

返回值：

0:没有记录；-1 输入的开始日期或结束日期不正确；json 格式数据：成功

json 格式数据：

[{"CALLID":"2010102512345 67","FROM_ADDR":"bjqx@allytel.com.cn","TO_ADDR":"gaoww@allytel.com.cn","CC_ADDR":"","BCC_ADDR":"","SUBJECTS":"邮件测试","BODYS":"","ATTACH1":"","ATTACH2":"","ATTACH3":"","ATTACH4":"","ATTACH5":"","FILE_EBX":"\\emaildata\\semail_IBX\\20101025161745516 0.ibx","GHID":"8615"}]

备注：

CALLID:邮件标识　　FROM_ADDR：发件人　　TO_ADDR：收件人
CC_ADDR：抄送　　BCC_ADDR：密送　　SUBJECTS：主题
BODYS：内容　　ATTACH1：附件 1　　ATTACH2：附件 2

ATTACH3:附件 3　　ATTACH4：附件 4　　ATTACH5：附件 5
EBX：ibx 路径　　GHID：发件人工号

Email_Recv 方法

```
public string Email_Recv(string strSdate, string strEdate, string strGhid)
```

说明：获取Email收件箱中的记录

参数：

strSdate：查询开始日期 格式：yyyyMMdd
strEdate：查询结束日期 格式：yyyyMMdd
strGhid：查询的工号(可以为空)

返回值：

0:没有记录；-1 输入的开始日期或结束日期不正确；json 格式数据：成功

json 格式数据：

[{"CALLID":"201110091648130000","FROM_ADDR":"f3000@imail.com","TO_ADDR":"werwrwer","CC_ADDR":"","BCC_ADDR":"","SUBJECTS":"Fw:[***SPAM*** Score/Req: 21.4/5.0] tech 你好 建.站","BODYS":"Reply-To: \"f3000\" <f3000@imail.com>\r\nFrom: \"f3000\"<f3000@imail.com>\r\nTo:href=3D\
","ATTACH1":"","ATTACH2":"","ATTACH3":"","ATTACH4":"","ATTACH5":"","FILE_EBX":".\\emaildata\\8615\\20111009164812.obx","GHID":"8615"}]

备注：

CALLID:邮件标识　　FROM_ADDR：发件人　　TO_ADDR：收件人
CC_ADDR：抄送　　BCC_ADDR：密送　　SUBJECTS：主题
BODYS：内容　　ATTACH1：附件 1　　ATTACH2：附件 2
ATTACH3:附件 3　　ATTACH4：附件 4　　ATTACH5：附件 5
FILE_EBX：obx 路径　　GHID：工号

Get_Caller_Area 方法

```
public string Get_Caller_Area(string strCaller)
```

说明：获取主叫号码所在地区

参数：

strCaller：主叫号码（模糊查询，必填）

返回值：

0:没有记录；-1 输入条件不正确；json 格式数据：成功

json 格式数据：

[{"CALLER":"02012345678","PROV":"广东","CITY":"广州"}]

Get_Customer_Info 方法

```
public string Get_Customer_Info(string Caller, string Userid, string UserName)
```

说明：获取客户资料(三个条件不能同时为空)

参数：

Caller：联系电话或家庭电话或手机号码（模糊查询）

Userid：客户编号

UserName：客户名称(模糊查询)

返回值：

0:没有记录；-1 输入条件不正确；json 格式数据：成功，json 内容为实际数据库表中的字段

json 格式数据：

[{"USERID":"200406081449020001","UNAME":"北京强讯公司","PYM":"BJQXGS","UTYPE":1,"TEL":"82015268","TEL_HOME":"82015266","FAX":"82015270","ADDRESS":"北京市西城区北三环中路甲 29 号华尊大厦 B 座601","RELATION":0,"PROV":"北京","CITY":"北京","MOBILENO":"","EMAIL":"","ZIP":"","PERSON":"MA","BDATE":"","SDATE":"20080902","STIME":"151346","LASTEXT":"8610","GHID":"8619","MEMO":""}]

备注：

USERID:用户标识	UNAME：客户名称	PYM：拼音码
UTYPE：客户类型	TEL：电话号码	TEL_HOME：家庭电话
FAX:传真号码	ADDRESS：详细地址	RELATION：关系
PROV：省份	CITY：城市	MOBILENO：手机
EMAIL:邮箱	ZIP：邮编	PERSON：联系人
BDATE：建立日期	SDATE：最新服务日期	STIME：最新服务时间
LASTEXT：上次服务工号	GHID：座席工号	MEMO：备注

Get_Person_Info 方法

```
public string Get_Person_Info(string strCaller, string Userid, string UserName)
```

说明：获取联系人资料(三个条件不能同时为空)

参数：

strCaller：联系电话或家庭电话或手机号码（模糊查询）

Userid：客户编号

UserName：联系人姓名(模糊查询)

返回值：

0:没有记录；-1 输入条件不正确；json 格式数据：成功，json 内容为实际数据库表中的字段

json 格式数据：

[{"USERID":"200406081449020001","UNAME":"北京强讯公司","PYM":"BJQXGS","UTYPE":1,"TEL":"82015268","TEL_HOME":"82015266","FAX":"82015270","ADDRESS":"北京市西城区北三环中路甲 29 号华尊大厦 B 座601","RELATION":0,"PROV":"北京","CITY":"北京","MOBILENO":"","EMAIL":"","ZIP":"","PERSON":"MA","BDATE":"","SDATE":"20080902","STIME":"151346","LASTEXT":"8610","GHID":"8619","MEMO":"","INFO":""}]

备注：

USERID:用户标识	UNAME：客户名称	PYM：拼音码

UTYPE：客户类型　TEL：电话号码　TEL_HOME：家庭电话
FAX:传真号码　ADDRESS：详细地址　RELATION：关系
PROV：省份　CITY：城市　MOBILENO：手机
EMAIL:邮箱　ZIP：邮编　PERSON：联系人
BDATE：建立日期　SDATE：最新服务日期　STIME：最新服务时间
LASTEXT：上次服务工号　GHID：座席工号　MEMO：备注
INFO:客户信息

Get_Talentel_log_Info 方法

```
public string Get_Talentel_log_Info(string strSdate, string strEdate, string strCaller, string strExt, string strCallid, string strGhid)
```

说明：获取录音信息

参数：

strSdate：查询开始日期 格式：yyyyMMdd(必填)
strEdate：查询结束日期 格式：yyyyMMdd(必填)
strCaller：主叫号码（模糊查询，可为空）
strExt：座席分机号（模糊查询，可为空）
strCallid：呼叫标识（精确查询，可为空）
strGhid：座席工号（模糊查询，可为空）

返回值：

0:没有记录；-1 输入条件不正确；json 格式数据：成功，json 内容为实际数据库表中的字段

json 格式数据：

[{"CALLER":"","CALLED":"603","MSG_TYPE":1,"SDATE":"20130301","STIME":"131839","LEN":134,"FILE_PATH":"D:\\recwav\\20130301\\ch038\\R131839.wav","OPID":"603","UCID":"011318290004"}]

备注：

CALLER:主叫号码　CALLED：被叫号码　MSG_TYPE：呼叫方向
SDATE：录音日期　STIME：起始时间　LEN：时长(秒)
FILE_PATH:录音文件路径　OPID：工号　UCID：标识

注：呼叫方向：0-空,1-呼入,2-呼出,3-内线呼入,4-内线呼出,11-未接

Get_Call_Info 方法

```
public string Get_Call_Info(string strSdate, string strEdate, string strCaller, string strExt, string strCallid, string strGhid)
```

说明：获取话单信息

参数：

strSdate：查询开始日期 格式：yyyyMMdd(必填)
strEdate：查询结束日期 格式：yyyyMMdd(必填)
strCaller：主叫号码（模糊查询，可为空）

strExt：座席分机号（模糊查询，可为空）

strCallid：呼叫标识（精确查询，可为空）

strGhid：座席工号（模糊查询，可为空）

返回值：

0:没有记录；-1 输入条件不正确；json 格式数据：成功，json 内容为实际数据库表中的字段

json 格式数据：

[{"CALLID":"051037520044","CALLER":"615","CALLED":"604","EXT":"615","DIRECTION":4,"DEST_TYPE":3,"GHID":"8615","OP_NAME":"高维薇","GROUPS":"","SDATE":"20130105","STIME":"103721","ETIME":"103726","LEN_RING":5,"LEN_TALK":0,"CALLTYPE":0,"QN_TYPE":0,"QN_DEST":"","QN_REASON":0,"QN_RESULT":0}]

备注：

CALLID:呼叫标识　　CALLER：主叫号码　　CALLED：被叫号码

EXT：座席分机号　　GHID：座席工号　　OP_NAME：座席员姓名

GROUPS：ACD 组号　　SDATE：通话日期　　STIME：起始时间

ETIME：结束时间　　LEN_RING：振铃时长(秒)　　LEN_TALK：通话时长(秒)

DIRECTION：呼叫方向：

0：未知 1：外线呼入 2：外线呼出 3：内线呼入 4：内线呼出

DEST_TYPE：呼入时，目标类型：

0：未知 1：路由点 2：IVR 3：座席分机 4：其它分机 5： 外线号码

CALLTYPE：呼叫话单类型：

0：正常通话话单 10：表示外线排队等待的话单

QN_TYPE：排队类型：

1：表示在分组上排队 2：表示在坐席上排队 3：表示在分机上排队 4：表示在交换机的 UCD 组中排队或 IVR 端口上排队

QN_DEST：排队目标，根据 QN_TYPE 决定：

QN_TYPE=1，表示组号 2：工号 3：分机号 4：IVR 端口号

QN_REASON：排队开始原因：

1：表示全忙进入排队 2：表示无人接听进入排队 3：表示其它原因进入排队

QN_RESULT： 排队结束原因：

1：表示成功转入 2：表示挂断，排队失败 3：表示其它原因 21：超时

Groups_Info 方法

```
public string Groups_Info(string strGroup, int nType)
```

说明：获取分组名称

参数：

strGroup：分组的编号或者是 ALL（必填）.

nType：分组类型，0-ACD 组，1-业务组（必填）

返回值：

strGroup 如果是 ALL,则返回所有组的组号和组名，返回格式为 1-技术部|2-销售部|...

strGroup 如果不是 ALL，则返回指定组的组名称

Group_Add 方法

```
public int Group_Add(string strGroup, string strGname, int nType)
```

说明：增加分组

参数：

strGroup：分组的编号（必填）

strGname：分组的名称（必填）

nType：分组类型，0-ACD 组，1-业务组（必填）

返回值：

1:成功；0：必填项没有输入完整；-1 失败

Group_Del 方法

```
public int Group_Del(string strGroup, int nType)
```

说明：删除分组，同时删除组员

参数：

strGroup：分组的编号（必填）

nType：分组类型，0-ACD 组，1-业务组（必填）

返回值：

1:成功；0：必填项没有输入；-1 失败

备注：

删除组号为 strGroup 的组；同时删除的 trGroup 的所有组成员

Group_Member 方法

```
public string Group_Members(string strGroup, int nType)
```

说明：获取某一分组坐席人员工号

参数：

strGroup：分组的编号（必填）

nType：分组类型，0-ACD 组，1-业务组（必填）

返回值：

成功：GHID1|GHID2|GHID3|；失败：””

Group_Member_Del 方法

```
public int Group_Member_Del(string strGroup, string strGhid, int nType)
```

说明：删除指定分组所属座席;当strGhid为空时，删除所有组员，否则只删除指定工号

参数：

strGroup：分组的组号（必填）

strGhid：要删除的组成员工号，当strGhid为空时，删除所有组员，否则只删除指定工号（非必填）

nType：分组类型，0-ACD 组，1-业务组（必填）

返回值：

1：成功；0：必填项没有输入或者没有要删除的数据；-1：失败

Group_Member_Add 方法

```
public int Group_Member_Add(string strGroup, string strGhid, int nType)
```

说明：向指定分组增加坐席

参数：

strGroup：分组的组号（必填）

strGhid：要增加的组成员工号（必填）

nType：分组类型，0-ACD 组，1-业务组（必填）

返回值：

1：成功；0：必填项没有输入完整；-1：失败

Agent_Role 方法

```
public string Agent_Role (string strGhid)
```

说明：获取坐席角色，返回角色名称

参数：

strGhid：CTS_OPIDK 工号（必填）

返回值：

返回此工号对应的角色名称

Agent_Add 方法

```
public int Agent_Add(string strGhid_Info)
```

说明：增加坐席在CTS_OPIDK表中，且默认在CTS_OPIDK_INFO中增加数据

参数：

strGhid_Info：坐席信息（必填）

strGhid_Info 内容：GHID=座席工号;NAME=座席姓名;PASS=座席密码;ROLES=角色编号;UTYPE=座席类型;EXT=分机号码;MOBILE=手机号码;

strGhid_Info 说明：EXT、MOBILE 为非必填项，其余的为必填项

返回值：

1：成功；0：必填项没有输入完整；-1：失败

Agent_Del 方法

```
public int Agent_Del(string strGhid)
```

说明：删除坐席，同时删除所属组成员

参数：

strGhid：坐席工号（必填）

返回值：

1：成功；0：必填项没有输入完整或者没有要删除的数据；-1：失败